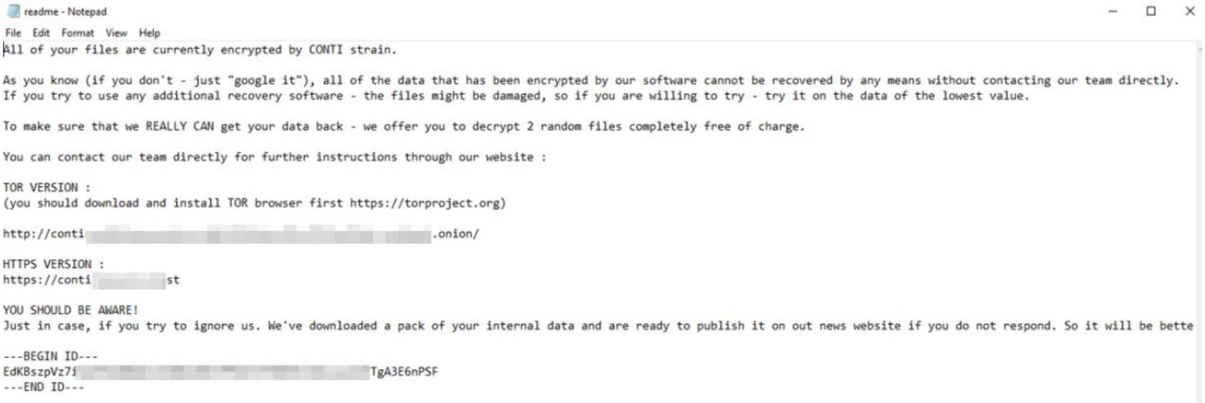
TryHackMe: Conti

The following writeup covers the room Conti, which is an intermediate room that involves investigating an exchange server that was compromised with Conti ransomware. This challenge involves using Splunk to investigate a series of queries and answer all the relevant questions. I really enjoyed this room and hope my writeup can be of use to someone out there.

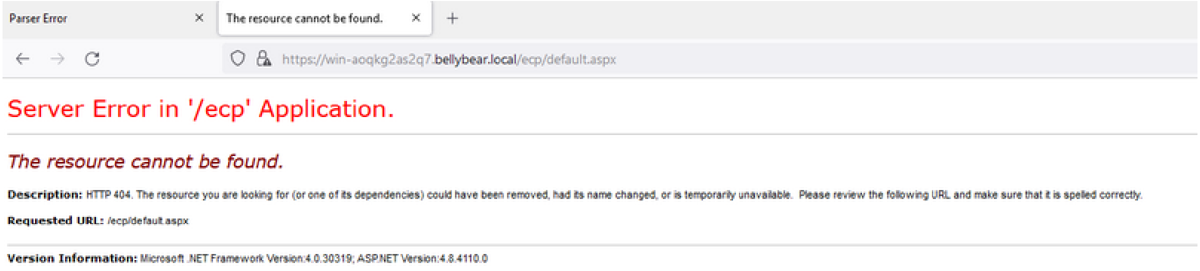
Scenario: Some employees from your company reported that they can't log into Outlook. The Exchange system admin also reported that he can't log in to the Exchange Admin Centre. After initial triage, they discovered some weird readme files settled on the Exchange server. Below is a copy of the ransomware note:



```
All of your files are currently encrypted by CONTI strain.

As you know (if you don't - just "google it"), all of the data that has been encrypted by our software cannot be recovered by any means without contacting our team directly.
If you try to use any additional recovery software - the files might be damaged, so if you are willing to try - try it on the data of the lowest value.

To make sure that we REALLY CAN get your data back - we offer you to decrypt 2 random files completely free of charge.

You can contact our team directly for further instructions through our website :

TOR VERSION :
(you should download and install TOR browser first https://torproject.org)

http://conti[redacted].onion/

HTTPS VERSION :
https://conti[redacted]st

YOU SHOULD BE AWARE!
Just in case, if you try to ignore us. We've downloaded a pack of your internal data and are ready to publish it on out news website if you do not respond. So it will be bette

---BEGIN ID---
EdKBszpVz7i[redacted]TgA3E6nPSF
---END ID---
```

Below are the error messages that the exchange admin and employees see when they try to access anything related to exchange or outlook:

Exchange Control Panel:



https://win-aoqkg2as2q7.bellybear.local/ecp/default.aspx

Server Error in '/ecp' Application.

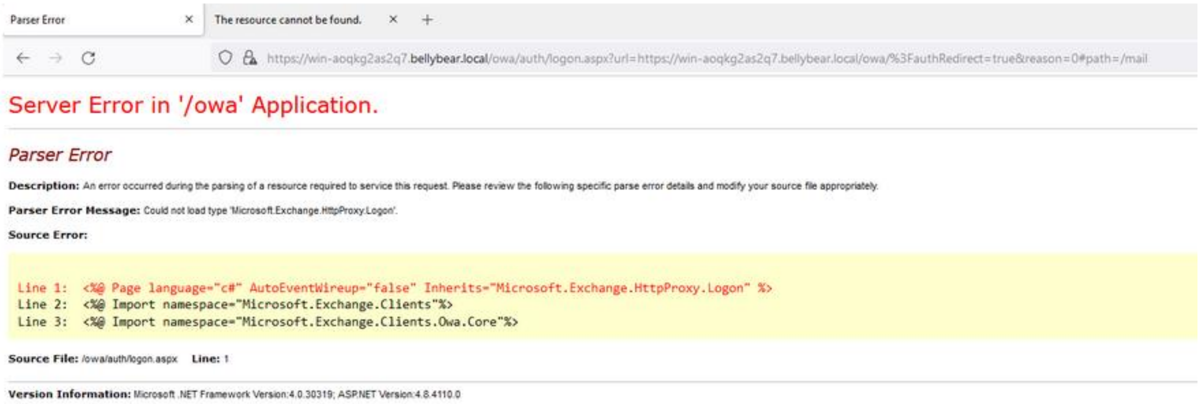
The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

Requested URL: /ecp/default.aspx

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.4110.0

Outlook Web Access:



https://win-aoqkg2as2q7.bellybear.local/owa/auth/logon.aspx?url=https://win-aoqkg2as2q7.bellybear.local/owa/%3FauthRedirect=true&reason=0#path=/mail

Server Error in '/owa' Application.

Parser Error

Description: An error occurred during the parsing of a resource required to service this request. Please review the following specific parse error details and modify your source file appropriately.

Parser Error Message: Could not load type 'Microsoft.Exchange.HttpProxy.Logon'.

Source Error:

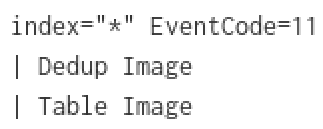
```
Line 1:  <%@ Page language="c#" AutoEventWireup="false" Inherits="Microsoft.Exchange.HttpProxy.Logon" %>
Line 2:  <%@ Import namespace="Microsoft.Exchange.Clients"%>
Line 3:  <%@ Import namespace="Microsoft.Exchange.Clients.Owa.Core"%>
```

Source File: /owa/auth/logon.aspx   **Line:** 1

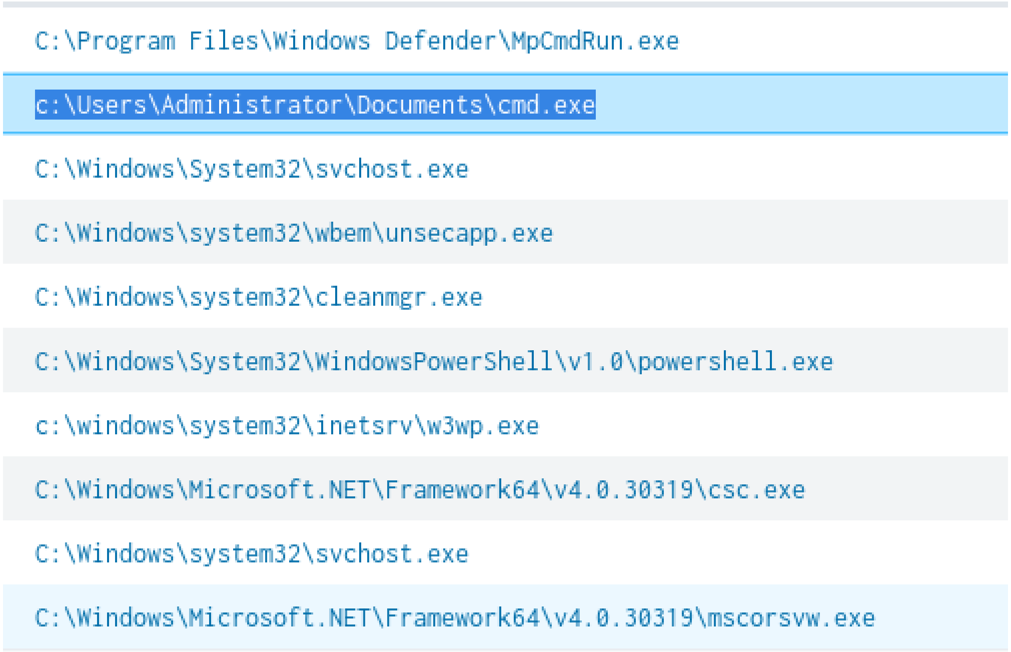
Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.4110.0

Can you identify the location of the ransomware?

To start, navigate to the search and reporting app within Splunk. This question is asking us to find the location of the ransomware, if you Google what event ID is related to file creation you can see that it is 11. We can leverage this to search for file creation logs. I did this by entering the following query:



```
index="*" EventCode=11
| Dedup Image
| Table Image
```



```
C:\Program Files\Windows Defender\MpCmdRun.exe
c:\Users\Administrator\Documents\cmd.exe
C:\Windows\System32\svchost.exe
C:\Windows\system32\wbem\unsecapp.exe
C:\Windows\system32\cleanmgr.exe
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
c:\windows\system32\inetsrv\w3wp.exe
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe
C:\Windows\system32\svchost.exe
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsvw.exe
```

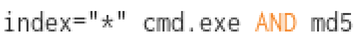
This produces 10 events, if you check the hint it says to look for a normal Windows binary in an unusual binary, cmd.exe being in the highlighted path is extremely suspicious and this is the answer.

What is the Sysmon event ID for the related file creation event?

We have already found the answer in the previous question, event ID 11 is for file creation.

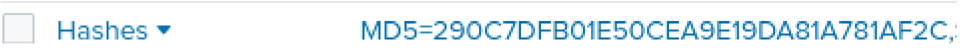
Can you find the MD5 hash of the ransomware?

I originally answer this question in a weird way, I simply searched for the identified suspicious binary (cmd.exe) and the string md5:

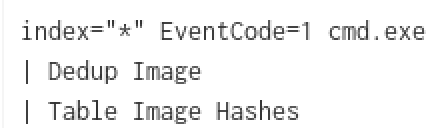


```
index="*" cmd.exe AND md5
```

In the first event I was able to find the md5 hash for the binary:



□ Hashes ▾ MD5=290C7DFB01E50CEA9E19DA81A781AF2C, Then I realised from investigating the log that you can also find the answer by searching for the event ID 1 (Process creation) and looking for the cmd.exe process like as follows:



```
index="*" EventCode=1 cmd.exe
| Dedup Image
| Table Image Hashes
```

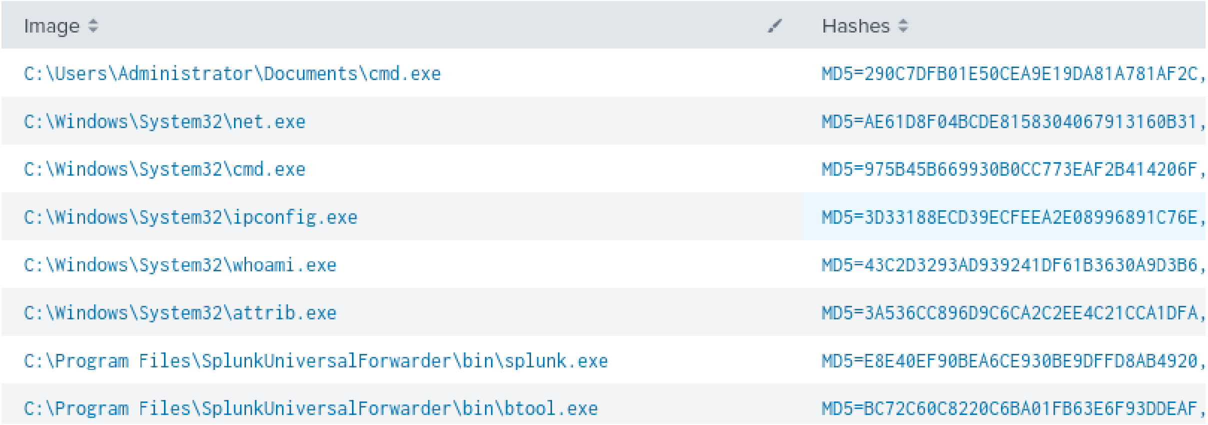
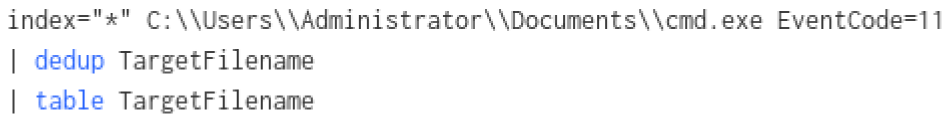


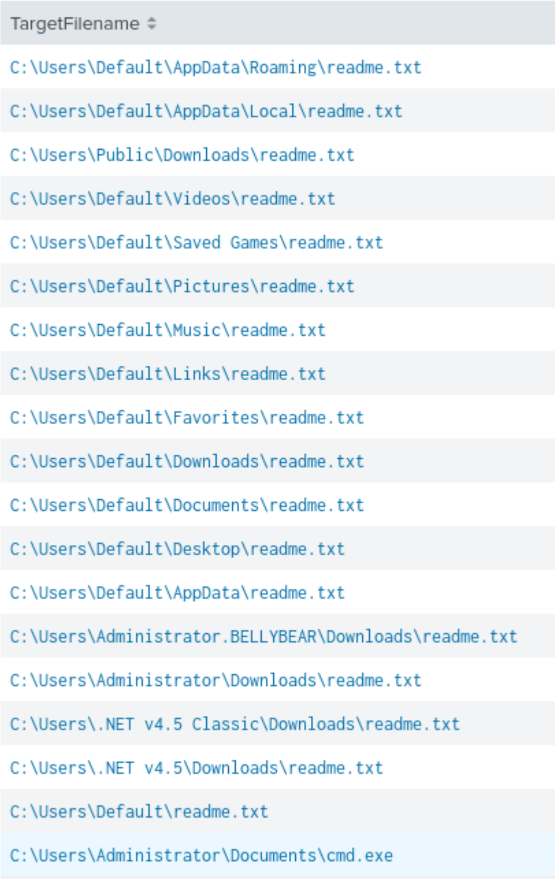
Image	Hashes
C:\Users\Administrator\Documents\cmd.exe	MD5=290C7DFB01E50CEA9E19DA81A781AF2C,
C:\Windows\System32\net.exe	MD5=AE61D8F04BCDE8158304067913160B31,
C:\Windows\System32\cmd.exe	MD5=975B45B669930B0CC773EAF2B414206F,
C:\Windows\System32\ipconfig.exe	MD5=3D33188ECD39ECFEEA2E08996891C76E,
C:\Windows\System32\whoami.exe	MD5=43C2D3293AD939241DF61B3630A9D3B6,
C:\Windows\System32\attrib.exe	MD5=3A536CC896D9C6CA2C2EE4C21CCA1DFA,
C:\Program Files\SplunkUniversalForwarder\bin\splunk.exe	MD5=E8E40EF90BEA6CE930BE9DFFD8AB4920,
C:\Program Files\SplunkUniversalForwarder\bin\btool.exe	MD5=BC72C60C8220C6BA01FB63E6F93DDEAF,

What file was saved to multiple folder locations?

To find what file was saved to multiple folders, we can assume that it probably has something to do with the malicious binary was have already identified. So let's use the following query:



```
index="*" C:\\Users\\Administrator\\Documents\\cmd.exe EventCode=11
| dedup TargetFilename
| table TargetFilename
```

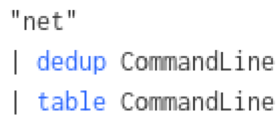


TargetFilename
C:\Users\Default\AppData\Roaming\readme.txt
C:\Users\Default\AppData\Local\readme.txt
C:\Users\Public\Downloads\readme.txt
C:\Users\Default\Videos\readme.txt
C:\Users\Default\Saved Games\readme.txt
C:\Users\Default\Pictures\readme.txt
C:\Users\Default\Music\readme.txt
C:\Users\Default\Links\readme.txt
C:\Users\Default\Favorites\readme.txt
C:\Users\Default\Downloads\readme.txt
C:\Users\Default\Documents\readme.txt
C:\Users\Default\Desktop\readme.txt
C:\Users\Default\AppData\readme.txt
C:\Users\Administrator.BELLYBEAR\Downloads\readme.txt
C:\Users\Administrator\Downloads\readme.txt
C:\Users\.NET v4.5 Classic\Downloads\readme.txt
C:\Users\.NET v4.5\Downloads\readme.txt
C:\Users\Default\readme.txt
C:\Users\Administrator\Documents\cmd.exe

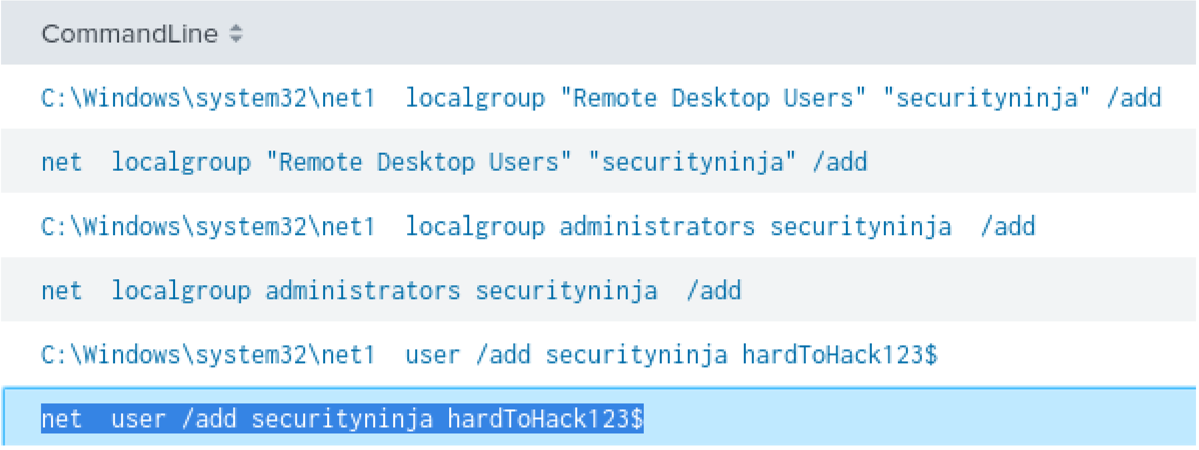
The output makes it obvious that the readme.txt file is the answer. All of these file creations were made from the cmd.exe image that we identified to be the ransomware.

What was the command the attacker used to add a new user to the compromised system?

There are a couple ways to do this, I found the answer by searching for the string “net” and looking at the CommandLine field:

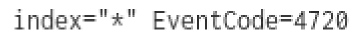


```
"net"
| dedup CommandLine
| table CommandLine
```

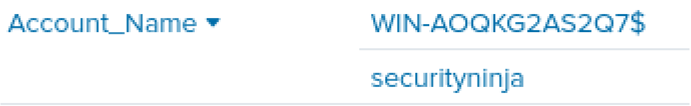


CommandLine
C:\Windows\system32\net1 localgroup "Remote Desktop Users" "securityninja" /add
net localgroup "Remote Desktop Users" "securityninja" /add
C:\Windows\system32\net1 localgroup administrators securityninja /add
net localgroup administrators securityninja /add
C:\Windows\system32\net1 user /add securityninja hardToHack123$
net user /add securityninja hardToHack123$

As you can see, someone entered the net user /add command and created the user securityninja with the password hardToHack123$. You can also find the answer by searching for event ID 4720:



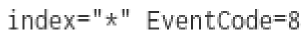
```
index="*" EventCode=4720
```



Account_Name	WIN-AOQKG2AS2Q7$
	securityninja

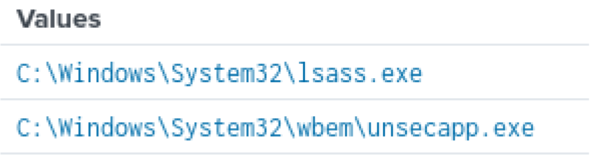
The attacker migrated the process for better persistence. What is the migrated process image (executable), and what is the original process image (executable) when the attacker got on the system?

As stated in the hint, we need to be investigating event ID 8, which is a Sysmon event ID for CreateRemoteThread. This event detects when a process created a thread in another process, this is a technique often used by malware to inject code and hide in other legitimate processes. If you filter for this event code like as follows:



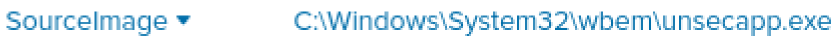
```
index="*" EventCode=8
```

Take a look at the TargetImage field:



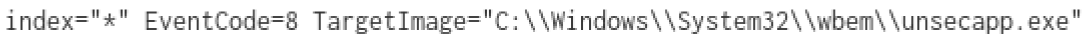
Values
C:\Windows\System32\lsass.exe
C:\Windows\System32\wbem\unsecapp.exe

Something is trying to create a remote thread in lsass.exe which is extremely suspicious, if you investigate the source image of this, you can determine that it is unsecapp.exe:

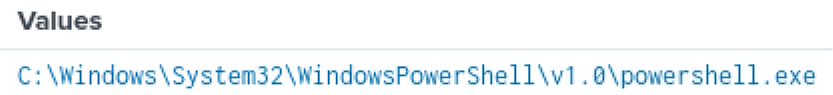


SourceImage	C:\Windows\System32\wbem\unsecapp.exe

If you investigate the source image of this binary, you will discover that it is PowerShell:



```
index="*" EventCode=8 TargetImage="C:\\Windows\\System32\\wbem\\unsecapp.exe"
```



Values

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

The answer is simply the full path for PowerShell followed by unsecapp.

The attacker also retrieved the system hashes. What is the process image used for getting the system hashes?

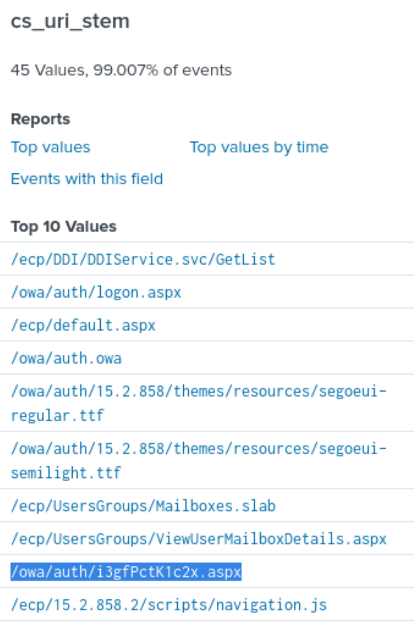
If you have a fundamental understanding of Windows, you will be aware that credential dumping tools like Mimikatz will access the lsass.exe process to dump credentials. Therefore, a process like unsecapp accessing lsass is extremely suspicious. Therefore, the answer is C:\Windows\System32\lsass.exe.

What is the web shell the exploit deployed to the system?

I really struggled with this question and needed to seek some outside assistance. However, I was eventually able to find the answer by filtering for the file extension .aspx, as it is a common web shell file type:

```
*.aspx*
```

If you investigate the cs_uri_stem field, you can identify the web shell:



cs_uri_stem

45 Values, 99.007% of events

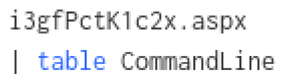
Reports

Top 10 Values

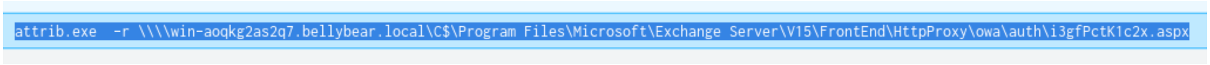
```
/ecp/DDI/DDIService.svc/GetList
/owa/auth/logon.aspx
/ecp/default.aspx
/owa/auth.owa
/owa/auth/15.2.858/themes/resources/segoeui-regular.ttf
/owa/auth/15.2.858/themes/resources/segoeui-semilight.ttf
/ecp/UsersGroups/Mailboxes.slab
/ecp/UsersGroups/ViewUserMailboxDetails.aspx
/owa/auth/i3gfPctK1c2x.aspx
/ecp/15.2.858.2/scripts/navigation.js
```

What is the command line that executed this web shell?

To identify the command line that executed this web shell, we can simply search for the web shell name and investigate the CommandLine field:



```
i3gfPctK1c2x.aspx
| table CommandLine
```



```
attrib.exe  -r \\\\win-aoqkg2as2q7.bellybear.local\C$\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\i3gfPctK1c2x.aspx
```

What three CVEs did this exploit leverage?

To be completely honest, I could not be bothered to research CVEs used by conti, so I just read someone else's writeup to find the answer:

CVE-2020-0796,CVE-2018-13374,CVE-2018-13379

The Conti room presented an engaging and insightful challenge that involved investigating a compromised Exchange server affected by Conti ransomware. Using Splunk to navigate through various legs, I was able to correctly answer all of the question. This challenge emphasised the importance of understanding event IDs and leveraging Splunk's search capabilities to identify and trace malicious activities. This room helped me reinforced my skills in incident response and forensic analysts. Overall, this was a fun challenge and valuable learning experience.