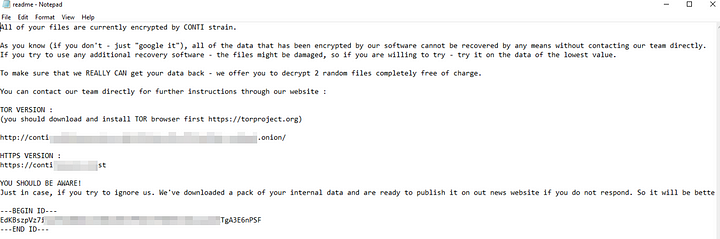
### **Conti — TryHackMe**

**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 Center. After initial triage, they discovered some weird readme files settled on the Exchange server.

Below is a copy of the ransomware note.



**Warning**: Do **NOT** attempt to visit and/or interact with any URLs displayed in the ransom note.

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**:

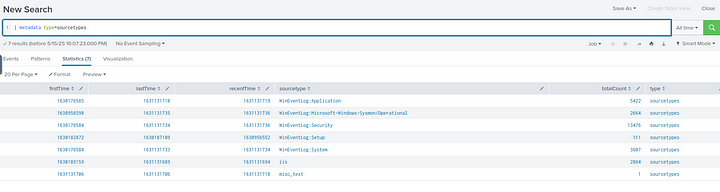


**Outlook Web Access**:

#### **Tasks**

As always before going ahead with the investigation, let us look at the source types available to us in the data.

| metadata type=sourcetypes



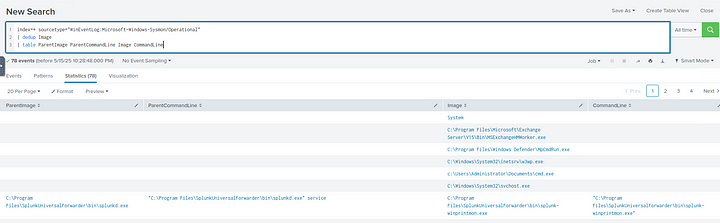
1. **Can you identify the location of the ransomware?**

First let us look at the Sysmon logs, specifically for the ParentAImage, ParentCommandLine, Image and CommandLine fields.

index=\* sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational"

| dedup Image

| table ParentImage ParentCommandLine Image CommandLine

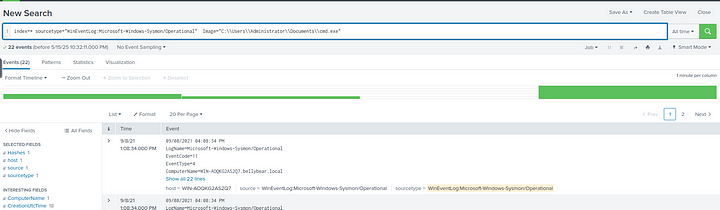


Now, we need to look for any unusual Commands or Images.

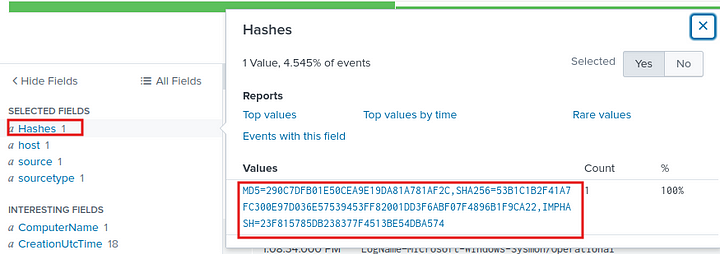


As we can see in the Image field an Image called cmd.exe which is in an unusual location. Let us further investigate this by obtaining the hash of this Image.

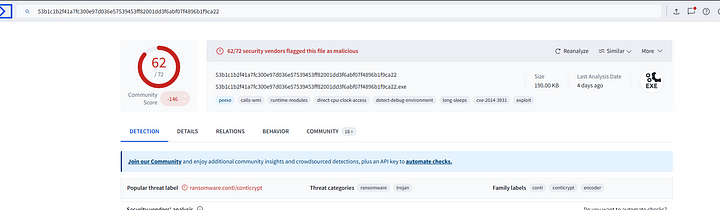
index=\* sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" Image="C:\\Users\\Administrator\\Documents\\cmd.exe"



Now, we can look at the hashes field to obtain the hash of this Image.



Let us copy this hash and submit to VirusTotal to see if this is malicious.

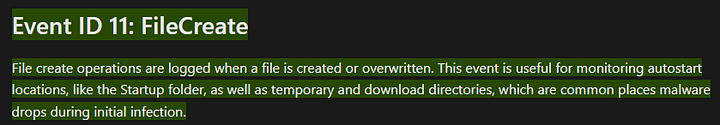


From VirusTotal resuslts it is certain that this file is malicious.

**Answer:** c:\Users\Administrator\Documents\cmd.exe

**2. What is the Sysmon event ID for the related file creation event?**

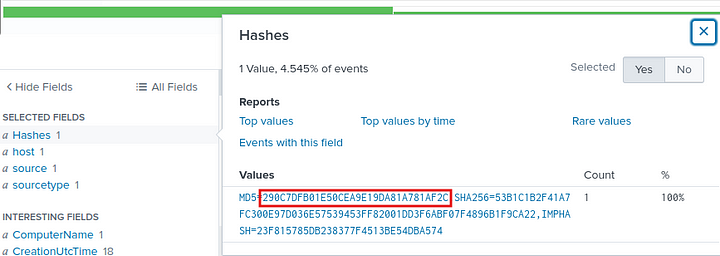
The Event Id for file creation is 11.



**Answer:** 11

**3. Can you find the MD5 hash of the ransomware?**

We already found this in question 1.



**Answer:** 290c7dfb01e50cea9e19da81a781af2c

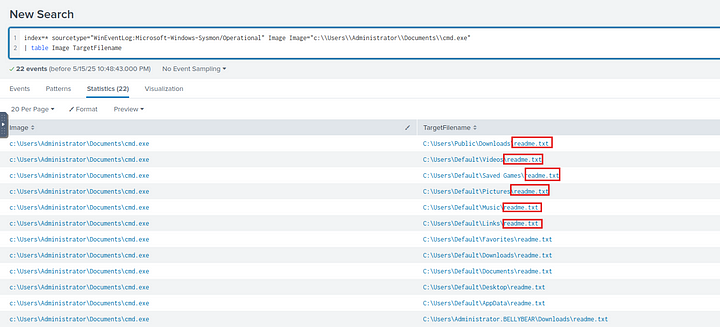
**4. What file was saved to multiple folder locations?**

We need to find the file name that was created by the ransomware, which was stored in different locations. Without looking at the results we can say that it will be the ransomware note that will be left after the ransomware attack, as this will contain the details about who to contact or how much to pay.

But let us look for the file in splunk. We can use the following filter.

index=\* sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" Image="c:\\Users\\Administrator\\Documents\\cmd.exe"

| table Image TargetFilename

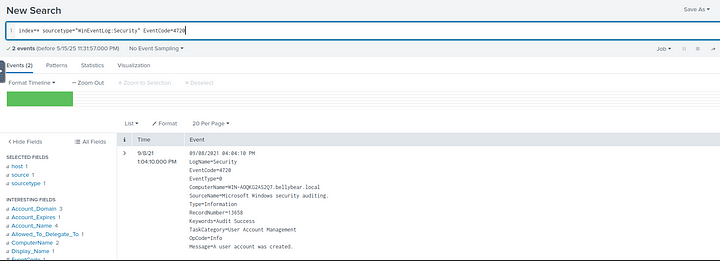


**Answer:** readme.txt

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

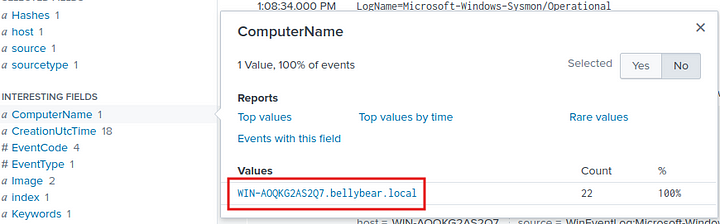
Let us check the Windows Security logs for the event id 4720 which is the event id for user creation.

index=\* sourcetype="WinEventLog:Security" EventCode=4720



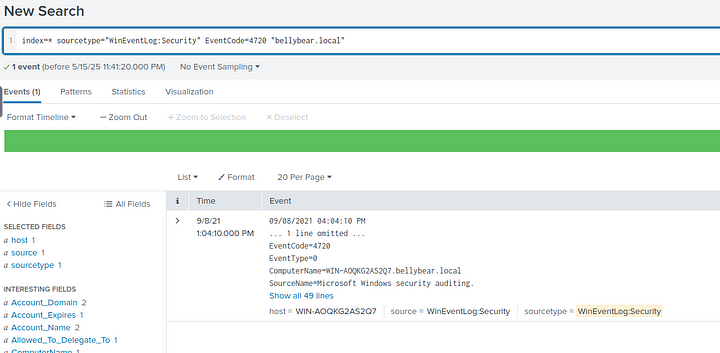
We got two events.

In the previous tasks we can notice that the account that was compromised was bellybear.local

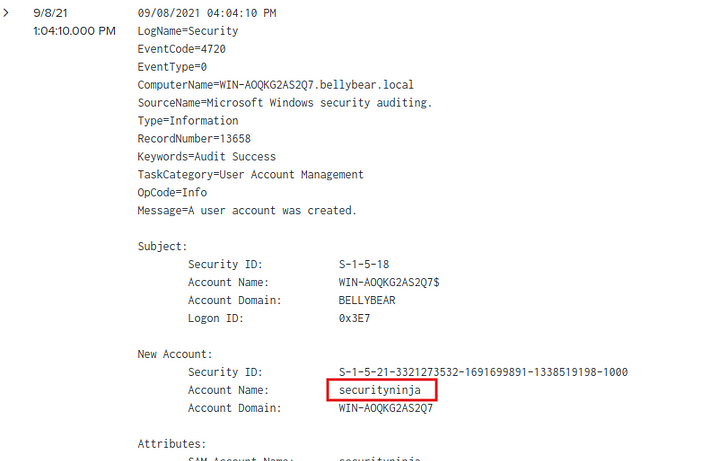


So in the returned events, let us look for this Account.

index=\* sourcetype="WinEventLog:Security" EventCode=4720 "bellybear.local"



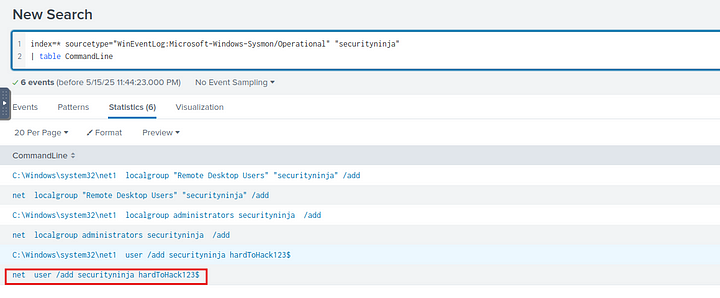
We got one event. If we examine this event, we can find the user name of the new account that was created.



The new user created was securityninja. Now we can go back to sysmon logs and look for this username.

index=\* sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" "securityninja"

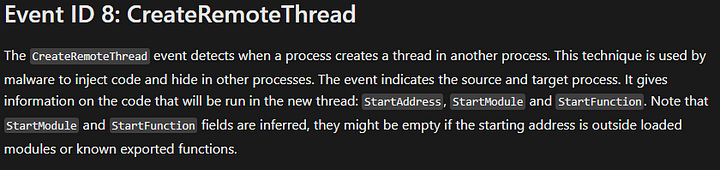
| table CommandLine



**Answer:** net user /add securityninja hardToHack123$

**6. 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?**

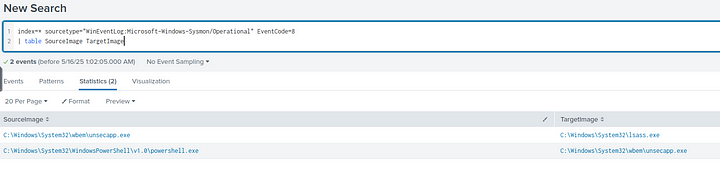
The event id relating to this question will be the sysmon event id 8.



So let us filter events on this id. And look for SourceImage and TargetImage fields specifically.

index=\* sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" EventCode=8

| table SourceImage TargetImage



We can see how the process had changed.

**Answer:** C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe,C:\Windows\System32\wbem\unsecapp.exe

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

In the above screenshot we can see that the process unsecapp.exe had migrated to lsass.exe. LSASS is **Local Security Authority Subsystem Service** which store a lot of sensitive data like encrypted passwords, NT hashes, LM hashes etc. So we can say that the attacker retrieved the hashes from this image.



**Answer:** C:\Windows\System32\lsass.exe

**8. What is the web shell the exploit deployed to the system?**

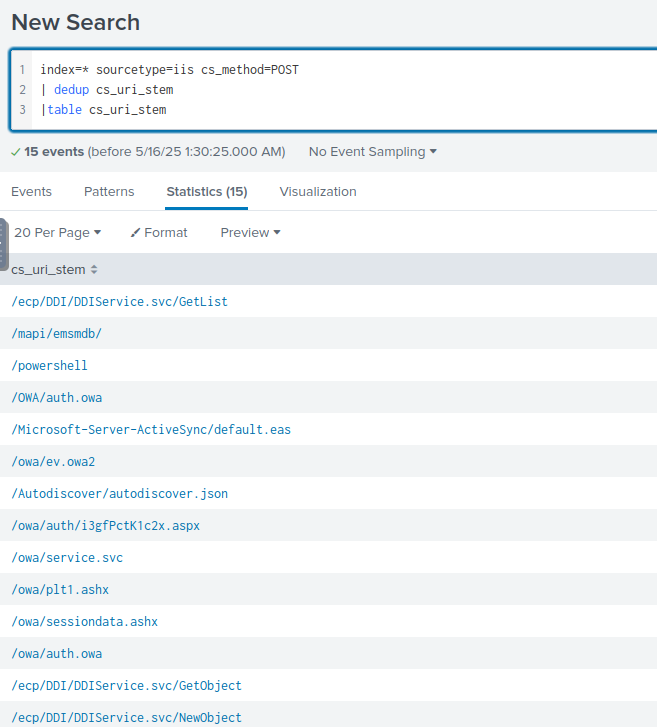
As per the hint, we need to look for POST requests in iis logs.

index=\* sourcetype=iis cs\_method=POST

| dedup cs\_uri\_stem

|table cs\_uri\_stem

Here cs\_uri\_stem will contain the exact resource requested in the log.



In the above screenshot we can see a.aspx file which is often used as a web shell in iis environments. Also the name of the file seems to not mean anything.

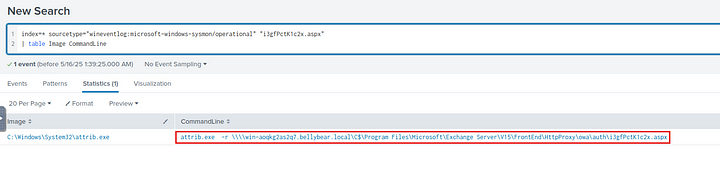
**Answer:** i3gfPctK1c2x.aspx

**9. What is the command line that executed this web shell?**

We can search for Sysmon logs, with this file in them and list the command line to find out the actual command that executed the web shell.

index=\* sourcetype="wineventlog:microsoft-windows-sysmon/operational" "i3gfPctK1c2x.aspx"

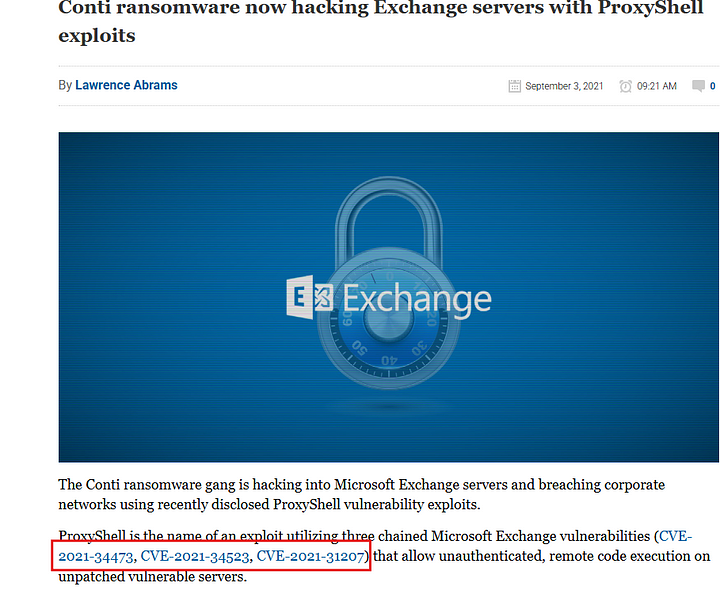
| table Image CommandLine



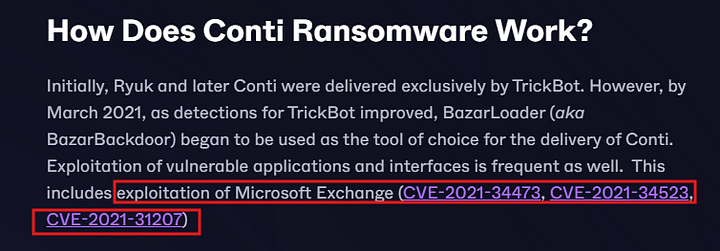
**Answer:** attrib.exe -r \\\\win-aoqkg2as2q7.bellybear.local\C$\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\ auth\i3gfPctK1c2x.aspx

**10. What three CVEs did this exploit leverage? Provide the answer in ascending order.**

While googling about Conti ransomware effecting Microsoft exchange servers, I came across an article from bleepingcomputer which lists out the CVE’s that this ransomware leverages to infect the Microsoft Exchange Servers.

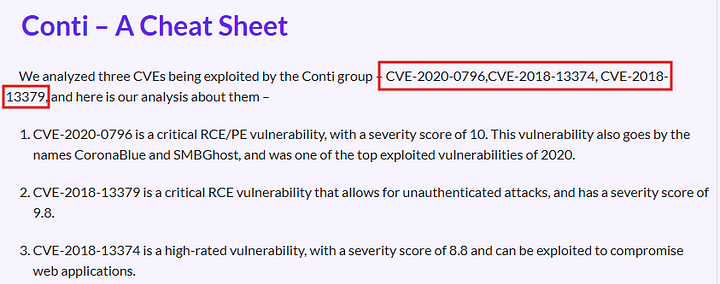


I also found an article from sentinel one.



But unfortunately these are not the answers TryHackMe is expecting.

With further research, I found an article from securin.io. And the CVE’s listed in there were accepted as right answer. I must admit, I am not fully convinced that these are the exact CVE’s that are exploited in this instance.



**Answer:** CVE-2020–0796, CVE-2018–13374, CVE-2018–13379

This is the end of this walkthorugh.