

Conti

<https://tryhackme.com/room/contiransomwarehgh>

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.

The screenshot shows a Windows Notepad window with the title "readme - Notepad". The content of the note is as follows:

```
File Edit Format View Help
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 our news website if you do not respond. So it will be better

---BEGIN ID---
EdKBszpVz7j[REDACTED] TgA3E6nPSF
---END ID---
```

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

Read the latest on the Conti ransomware [here](#).

Connect to OpenVPN or use the AttackBox to access the attached Splunk instance.

Splunk Interface Credentials:

Username: bellybear

Password: password!!!

Splunk URL: http://MACHINE_IP:8000

Can you identify the location of the ransomware?

Check the Q2 and it mentioned that Sysmon event created and code is 11. write your query

1 index** EventCode=11

✓ 104 events (before 5/6/25 11:25:41.000 AM) No Event Sampling ▾

Events (104) Patterns Statistics Visualization

Format Timeline ▾ — Zoom Out + Zoom to Selection X Deselect

	Time	Event
>	9/8/21 1:08:43.000 PM	09/08/2021 04:08:43 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=11 EventType=4 ComputerName=WIN-AOQKG2AS2Q7.bellybear.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=3232 Keywords=None TaskCategory=File created (rule: FileCreate) OpCode=Info Message=File created: RuleName: EXE

◀ Hide Fields ▶ All Fields

SELECTED FIELDS

- a host 1
- a source 1
- a sourcetype 1

INTERESTING FIELDS

- a ComputerName 1
- a CreationUtcTime 87
- # EventCode 1
- # EventType 1
- a Image 10
- a Index 1
- a Keywords 1
- # Linecount 1
- a LogName 1
- a Message 100+

for our search for the tip we got something

Explain common Windows binary located in an unusual location.



A common Windows binary located in an unusual location is often a red flag in cybersecurity, particularly in malware analysis or incident response. A good example is:

❖ Example: cmd.exe located in C:\Users\Public\ or C:\ProgramData\ instead of its legitimate path, which is C:\Windows\System32\cmd.exe.

⚠ Why it's suspicious:

- Windows binaries like cmd.exe, powershell.exe, rundll32.exe, regsvr32.exe, etc., are frequently abused by attackers ("living off the land").
- When found outside their usual directories (e.g., System32 or SysWOW64), it may indicate:
 - Renamed or relocated malicious copies.
 - Persistence mechanisms using fake trusted binaries.
 - Evasion of security monitoring tools that whitelist binaries by location.

🔍 Detection Tip:

Use tools like Sysinternals' Autoruns or run PowerShell scripts that check hashes and file locations of known binaries to detect anomalies.

Would you like a list of commonly abused binaries (LOLBins) and where they should or should not be found?



We need to check images if we can see anything.

The screenshot shows the Splunk interface for a search titled "Image". The left sidebar lists various fields, with "index 1" and "Keywords 1" highlighted by red arrows. The main pane displays a table of "Top 10 Values" for the "Image" field, ordered by count. The table includes columns for "Count" and "%". A red arrow points to the row for "cmd.exe" at index 18, which has a count of 18 and a percentage of 17.308%. Below the table, a detailed event log entry is shown, starting with a timestamp of 9/8/21 at 1:08:35 PM.

Top 10 Values	Count	%
C:\Windows\System32\cleanmgr.exe	27	25.962%
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsvw.exe	26	25%
C:\Windows\System32\svchost.exe	18	17.308%
c:\Users\Administrator\Documents\cmd.exe	18	17.308%
C:\Program Files\Windows Defender\MpCmdRun.exe	6	5.769%
c:\windows\system32\inetsrv\w3wp.exe	4	3.846%
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	2	1.923%
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe	1	0.962%
C:\Windows\system32\svchost.exe	1	0.962%
C:\Windows\system32\wbem\unsecapp.exe	1	0.962%

In Splunk, the "Image" section often refers to the image path of a process or executable as recorded in event logs—particularly in Windows Event Logs or Sysmon logs. It's not about visual images, but rather executable file paths.

💡 Here's what it means:

- **Image** = Full path to the executable file (.exe) that started a process or was involved in an event.
- This field is most common in:
 - Windows Security Logs (Event ID 4688: A new process has been created)
 - Sysmon Logs (Event ID 1: Process Create)
 - EDR or SIEM data sources

🔍 Example:

In a Sysmon event:

json

CopyEdit

"Image": "C:\Windows\System32\cmd.exe"

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

11

Can you find the MD5 hash of the ransomware?

Basicly remove the EcentCode =11 o and add md5 in the search bar

New Search

1 index=* Image="c:\Users\Administrator\Documents\cmd.exe" md5

1 event (before 5/6/25 11:03:40.000 AM) No Event Sampling

Enable event sampling to run the search and return a random set of events.

Events (1) Patterns Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect

1 millisecond

List ▾ Format 20 Per Page ▾

Selected Fields

host 1 source 1 sourcetype 1

Interesting Fields

CommandLine 1 Company 1 ComputerName 1 CurrentDirectory 1 Description 1 EventCode 1 EventType 1 FileVersion 1 Hashes 1

Time Event

> 9/8/21 04:05:32 PM ... 29 lines omitted ... TerminalSessionId: 0 IntegrityLevel: System Hashes: MD5: 290C70B0E50CEA9E19DA81A781AF26, SHA256=53B1C1B2F41A7FC300E97D036E57539453FF82001DD3F6ABF07F4896B1F9CA22, IMPHASH=23F8157850B238377F4513BE54D8A574 ParentProcessGuid: {72893ba8-1628-6139-7c02-000000000000} Show all 37 lines

host = WIN-AOOKG2AS2Q7 source = WinEventLog:Microsoft-Windows-Sysmon/Operational sourcetype = WinEventLog:Microsoft-Windows-Sysmon/Operational

virustotal.com/gui/file/53b1c1b2f41a7fc300e97d036e57539453ff82001dd3f6abf07f4896b1f9ca22

62/72 security vendors flagged this file as malicious

Community Score -146

Size 190.00 KB Last Analysis Date 25 days ago

PE executable

Detection Details Relations Behavior Community 18+

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label ransomware.conti/conticrypt Threat categories ransomware trojan Family labels conti conticrypt encoder

Security vendors' analysis

Acronis (Static ML)	Suspicious	AhnLab-V3	Ransomware/Win.Conti.R372647
Alibaba	Ransom:Win32/ContiCrypt.b7e00109	AliCloud	RansomWare
ALYac	Trojan.Ransom.Conti	Antly-AVL	Trojan(Ransom)/Win32.Conti
Arcabit	Trojan.Ransom.Conti.135	Arctic Wolf	Unsafe
Avast	Win32:Conti-B [Ransom]	AVG	Win32:Conti-B [Ransom]
Avira (no cloud)	HEUR/AGEN.1366989	BitDefender	Gen:Variant.Ransom.Conti.135
Bkav Pro	W32.AIDetectMalware	ClamAV	Win.Ransomware.Conti-9826703-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	CTX	Exe.ransomware.conti

What file was saved to multiple folder locations?

"Image="c:\\Users\\Administrator\\Documents\\cmd.exe" EventCode=11", and after that click on TargetFileName on the interesting fields section, we can see below result.

In Splunk, the field TargetFileName typically refers to the full path and name of a file involved in a file-related event—most commonly from Windows logs, Sysmon (System Monitor), or endpoint detection tools.

Why it's important in security:

- Used to detect:
 - File creation in suspicious locations (e.g., TEMP, Downloads, ProgramData)
 - Known malware filenames or hashes
 - Unexpected file writes by sensitive processes

🛠 Example Splunk search:

spl

CopyEdit

```
index=sysmon EventCode=11 TargetFileName="*\Temp\*"
```

The screenshot shows a Splunk search interface. On the left, there is a sidebar with a list of fields, one of which, 'a TargetFilename', is highlighted with a red arrow. The main pane displays a table of 'Top 10 Values' for 'TargetFilename'. Red arrows point from the table rows to the right, indicating specific entries: 'C:\Users\Administrator\Downloads\readme.txt', 'C:\Users\Default\Downloads\readme.txt', 'C:\Users\Default\AppData\Local\readme.txt', 'C:\Users\Default\AppData\Roaming\readme.txt', 'C:\Users\Default\Desktop\readme.txt', 'C:\Users\Default\Documents\readme.txt', and 'C:\Users\Default\Downloads\readme.txt'. The top of the table shows the following header: **Time** Event. Below the table, the event details are shown: 1:08:23.000 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=11 EventType=4 ComputerName=WIN-AOQKG2AS2Q7.bellybear.local

Value	Count	%
C:\Users\NET v4.5 Classic\Downloads\readme.txt	1	5.556%
C:\Users\NET v4.5\Downloads\readme.txt	1	5.556%
C:\Users\Administrator.BELLYBEAR\Downloads\readme.txt	1	5.556%
C:\Users\Administrator\Downloads\readme.txt	1	5.556%
C:\Users\Default\AppData\Local\readme.txt	1	5.556%
C:\Users\Default\AppData\Roaming\readme.txt	1	5.556%
C:\Users\Default\Desktop\readme.txt	1	5.556%
C:\Users\Default\Documents\readme.txt	1	5.556%
C:\Users\Default\Downloads\readme.txt	1	5.556%

The screenshot shows the Splunk interface with the following details:

- Message**: A modal window showing 22 events before 5/6/25 11:42:51.000 AM.
- Events (22)**: The main tab selected, showing a timeline and zoom options.
- Selected Fields** (highlighted with red arrows):
 - a host 1
 - a source 1
 - a sourcetype 1
 - a ComputerName 1
 - a CreationUtcTime 18
 - # EventCode 4
 - # EventType 1
 - a Image 2
 - a index 1
 - a Keywords 1
 - # linecount 4
 - a LogName 1
 - a Message 22
 - a OpCode 1
 - a ProcessGuid 1
 - # ProcessId 1
 - a punct 3
- Top 10 Values** table (highlighted with red arrows):

	Count	%
Dns query: RuleName: - UtcTime: 2021-09-08 20:05:33.923 ProcessGuid: {72893ba8-178c-6139-b402-000000000c00} ProcessId: 15540 QueryName: WIN-AOQKG2AS2Q7 QueryStatus: 0 QueryResults: 10.10.10.6; Image: C:\Users\Administrator\Documents\cmd.exe	1	4.545%
File created: RuleName: DefaultUserModified UtcTime: 2021-09-08 20:05:45.888 ProcessGuid: {72893ba8-178c-6139-b402-000000000c00} ProcessId: 15540 Image: c:\Users\Administrator\Documents\cmd.exe TargetFilename: C:\Users\Default\readme.txt CreationUtcTime: 2021-09-08 20:05:45.887	1	4.545%
File created: RuleName: DefaultUserModified UtcTime: 2021-09-08 20:08:23.543 ProcessGuid: {72893ba8-178c-6139-b402-000000000c00} ProcessId: 15540 Image: c:\Users\Administrator\Documents\cmd.exe TargetFilename: C:\Users\Default\AppData\readme.txt CreationUtcTime: 2021-09-08 20:08:23.543	1	4.545%
File created: RuleName: DefaultUserModified UtcTime: 2021-09-08 20:08:23.548 ProcessGuid: {72893ba8-178c-6139-b402-000000000c00} ProcessId: 15540 Image: c:\Users\Administrator\Documents	1	4.545%

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

Filter out using CommandLine and wildcard on add to check all commands containing *add*. Below is the result command.

The screenshot shows the Splunk "Select Fields" interface with the following details:

- Select Fields** section:
 - Field: CommandLine (selected with a checked checkbox).
 - Reports:
 - Top values
 - Events with this field
 - Rare values
- Event Coverage** table:

# of Values	Event Coverage	Type
1	4.55%	String
- Selected Fields** table:

	Count	%	Type
host	1	100%	String
source	1	100%	String
sourcetype	1	100%	String
Company	1	4.55%	String
ComputerName	1	100%	String
CreationUtcTime	18	81.82%	String
CurrentDirectory	1	4.55%	String
Description	1	4.55%	String
DestinationHostname	2	9.09%	String
Destinationip	2	9.09%	String
Destinationport	1	9.09%	String

CommandLine="*/add*"

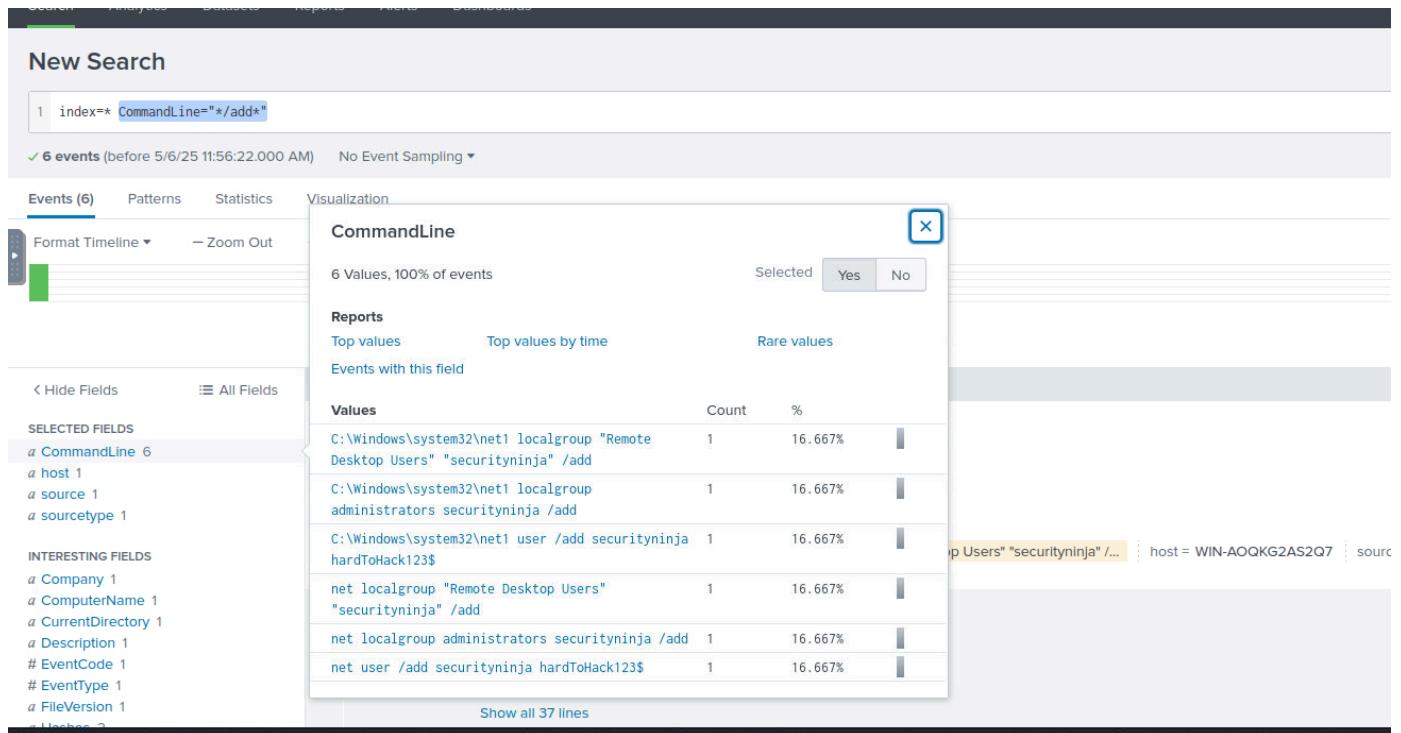
The query snippet `CommandLine="/add"` is filtering events in Splunk where the `CommandLine` field contains the string `/add`.

"`/add`":

- The asterisks (*) act as wildcards in Splunk (they represent "zero or more characters").
- So this means: **the CommandLine must contain the string `/add` anywhere in it.**

Why this is often monitored:

- The `/add` switch is commonly used with the `net` command to add user accounts or groups.
- This pattern is often monitored in detection rules to catch potential privilege escalation or unauthorized account creation.



`net user /add securityninja hardToHack123$`

- ✓ This is the most direct command that:

Adds a new user account named `securityninja`

Sets the password to `hardToHack123$`

The other commands extend privileges or add the user to groups, but this one is the initial creation of the user account.

Breakdown of Each Command:

1. ✓ `net user /add securityninja hardToHack123$`

- **Purpose:** Create a new local user account named `securityninja` with the password `hardToHack123$`
- **Explanation:**

This is the core action that introduces the new user into the system.

- **Impact:** Account now exists locally on the system.

2. ~~net~~ net localgroup administrators securityninja /add

- **Purpose:** Add the new user to the local Administrators group.

- **Explanation:**

Grants the user full administrative privileges.

- **Impact:** The attacker now has full control over the system using this account.

3. ~~net~~ net localgroup "Remote Desktop Users" "securityninja" /add

- **Purpose:** Allow the user to connect to the system via Remote Desktop Protocol (RDP).

- **Explanation:**

This is required if the attacker wants to log in graphically via RDP.

- **Impact:** Enables remote access with GUI.

4. ~~C:\Windows\system32\~~ net1 user /add securityninja hardToHack123\$

- **Purpose:** Same as Command #1 but using net1 instead of net.

- **Explanation:**

net1 is a backup or alternative binary to net, sometimes used to bypass basic monitoring that watches net.exe.

- **Impact:** Redundant but useful for stealth.

5. ~~C:\Windows\system32\~~ net1 localgroup administrators securityninja /add

- **Purpose:** Same as Command #2, but using net1.

- **Explanation:**

Also for stealth or if net.exe is being monitored or restricted.

6. ~~C:\Windows\system32\~~ net1 localgroup "Remote Desktop Users" "securityninja" /add

- **Purpose:** Same as Command #3, but using net1.

- **Explanation:**

Again, an alternate way to ensure remote access is granted.

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?

Add EventCode =8 in search bar

Splunk > enterprise Apps ▾

Search Analytics Datasets Reports Alerts Dashboards

New Search

```
1 index=* EventCode=8
```

✓ 2 events (before 5/6/25 12:05:09.000 PM) No Event Sampling ▾

Events (2) Patterns Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect

List ▾ Format 20 Per Page ▾

Time	Event
9/8/21 12:55:30.000 PM	<code>09/08/2021 03:55:30 PM</code> <code>LogName=Microsoft-Windows-Sysmon/Operational</code> <code>EventCode=8</code> <code>EventType=4</code> <code>ComputerName=WIN-AOQKG2AS2Q7.bellybear.local</code> <code>User=NOT_TRANSLATED</code> <code>Sid=S-1-5-18</code> <code>SidType=0</code> <code>SourceName=Microsoft-Windows-Sysmon</code> <code>Type=Information</code> <code>RecordNumber=2915</code> <code>Keywords=None</code> <code>TaskCategory=CreateRemoteThread detected (rule: CreateRemoteThread)</code> <code>OpCode=Info</code> <code>Message=CreateRemoteThread detected:</code>
9/8/21 12:54:12.000 PM	<code>09/08/2021 03:54:12 PM</code> <code>LogName=Microsoft-Windows-Sysmon/Operational</code> <code>EventCode=8</code> <code>EventType=4</code> <code>ComputerName=WIN-AOQKG2AS2Q7.bellybear.local</code> <code>User=NOT_TRANSLATED</code> <code>Sid=S-1-5-18</code> <code>SidType=0</code> <code>SourceName=Microsoft-Windows-Sysmon</code> <code>Type=Information</code> <code>RecordNumber=2839</code> <code>Keywords=None</code> <code>TaskCategory=CreateRemoteThread detected (rule: CreateRemoteThread)</code> <code>OpCode=Info</code> <code>Message=CreateRemoteThread detected:</code> <code>RuleName: -</code> <code>UtcTime: 2021-09-08 19:54:12.665</code> <code>SourceProcessGuid: {72893ba8-1458-6139-0702-000000000c00}</code> <code>SourceProcessId: 1580</code> <code>SourceImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</code> <code>TargetProcessGuid: {72893ba8-1125-6139-5d00-000000000c00}</code> <code>TargetProcessId: 5016</code> <code>TargetImage: C:\Windows\System32\wbem\unsecapp.exe</code> <code>NewThreadId: 12464</code> <code>StartAddress: 0x000001BFEE130000</code> <code>StartModule: -</code> <code>StartFunction: -</code>

Fields

- **Source Image (Original Process):**

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

- **Target Image (Migrated Process):** `C:\Windows\System32\wbem\unsecapp.exe`

- **Details:**

- **Original Process:** `powershell.exe` is being used by the attacker, which is common in exploitation or post-exploitation activities. Attackers frequently use PowerShell for scripting malicious actions.
- **Migrated Process:** `unsecapp.exe` is a legitimate Windows process involved in WMI (Windows Management Instrumentation). However, in this case, it is being used by the attacker to move

their malicious code into a more stealthy process that is less likely to be flagged.

The key here is that the attacker migrated from `powershell.exe` to `unsecapp.exe`, and this is typically done for **better persistence** and to evade detection. `powershell.exe` is often closely monitored, and by injecting code into `unsecapp.exe`, the attacker can keep their activities hidden behind a legitimate process.

Answer:

- **Migrated Process Image (Executable):** `C:\Windows\System32\wbem\unsecapp.exe` (Event 2)
 - **Original Process Image (Executable):**
`C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe` (Event 2)
-

Why This Event is Key:

- **Persistence and Stealth:** The attacker migrated from a high-visibility process (`powershell.exe`) to a less suspicious one (`unsecapp.exe`). This is often done to maintain control while avoiding detection by security tools that might flag the more obvious PowerShell activity.

This is the correct event because the **process migration** to `unsecapp.exe` reflects the attacker's intent to use a legitimate system process for persistence, while reducing the likelihood of detection.

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

Using the EventCode=8 filter alone, check the TargetImage and it will show two events, if we refer to the output from the search used in question 6, we can see that a second process migration takes place between `unsecapp.exe` and `lsass.exe`

splunk>enterprise Apps ▾

Search Analytics Datasets Reports Alerts Dashboards

3 Message

New Search

1 index=* EventCode=8

✓ 2 events (before 5/6/25 12:05:09.000 PM) No Event Sampling

Enable event sampling to run the search and return a random set of events.

Events (2) Patterns Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselected

List ▾ ✎ Format 20 Per Page ▾

Time	Event
9/8/21 12:55:30.000 PM	09/08/2021 03:55:30 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=8 EventType=4 ComputerName=WIN-AOQKG2AS2Q7.bellybear.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=2915 Keywords=None TaskCategory=CreateRemoteThread detected (rule: CreateRemoteThread) OpCode=Info Message=CreateRemoteThread detected:
9/8/21 12:54:12.000 PM	09/08/2021 03:54:12 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=8

Selected Fields

- host 1
- source 1
- sourcetype 1

Interesting Fields

- ComputerName 1
- # EventCode 1
- # EventType 1
- Index 1
- Keywords 1
- # linecount 1
- LogName 1
- Message 2
- # NewThreads 1

Hide Fields

All Fields

ids

List ▾ ✎ Format 20 Per Page ▾

Time	Event
9/8/21 12:55:30.000 PM	09/08/2021 03:55:30 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=8 EventType=4 ComputerName=WIN-AOQKG2AS2Q7.bellybear.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=2915 Keywords=None TaskCategory=CreateRemoteThread detected (rule: CreateRemoteThread) OpCode=Info Message=CreateRemoteThread detected: RuleName: - UtcTime: 2021-09-08 19:55:30.770 SourceProcessGuid: {72893ba8-1125-6139-5d00-000000000000} SourceProcessId: 5016 SourceImage: C:\Windows\System32\wbem\unsecapp.exe TargetProcessGuid: {72893ba8-111d-6139-0c00-000000000000} TargetProcessId: 672 TargetImage: C:\Windows\System32\lsass.exe NewThreadId: 13980 StartAddress: 0x000001D471950000 StartModule: - StartFunction: - Collapse
9/8/21 12:54:12.000 PM	09/08/2021 03:54:12 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=8

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

To answer this question, I started by changing my SourceType from Sysmon to IIS events, since it collects events related to web pages. Next, I filtered IIS events for POST requests and common web shell file types (.php, .asp, .aspx, .jsp):



Web shells are malicious scripts that attackers upload to a web server to gain remote control and execute arbitrary commands. They are often disguised as legitimate files to avoid detection. Here are some common file types and extensions used for web shells:

Common Web Shell File Types

1. PHP Files

- Extensions: ` .php` , ` .php5` , ` .phtml`
- Example Web Shells: ` c99.php` , ` r57.php` , ` b374k.php`

2. ASP and ASP.NET Files

- Extensions: ` .asp` , ` .aspx` , ` .ashx`
- Example Web Shells: ` cmdasp.asp` , ` aspxspy.aspx`

3. JSP (Java Server Pages) Files

- Extensions: ` .jsp` , ` .jspx`
- Example Web Shells: ` cmd.jsp` , ` webshell.jsp`

4. CFM (ColdFusion Markup) Files

- Extensions: ` .cfm` , ` .cfml`



The screenshot shows the Splunk 8.2.2 interface with a search bar containing the query: `index=* sourcetype=iis cs_method=POST | search .php OR .asp OR .aspx OR .jsp`. The results pane displays 187 events. A modal window titled "CS_URI_STEM" is open, showing a table of top values by count and percentage. One row in the table is highlighted: `/owa/auth/13gPck1Kc2x.aspx` with a count of 4 and a percentage of 2.13%. The table also includes columns for "Values", "Count", and "%". The bottom of the table shows log entries corresponding to the highlighted URL.

Values	Count	%
/ecp/001/001Service.svc/GetList	171	91.44%
/owa/auth.owa	8	4.27%
/owa/auth/13gPck1Kc2x.aspx	4	2.13%
/ecp/001/001Service.svc/NewObject	2	1.07%
/ecp/001/001Service.svc/GetObject	1	0.53%
/owa/lang.owa	1	0.53%

```
index=* sourcetype=iis cs_method=POST
| search .php OR .asp OR .aspx OR .jsp
```

Try looking in the IIS logs for POST requests.

Try looking in the IIS logs for POST requests. Logs mean use sourcetype=iis

The screenshot shows a Splunk search interface with the following search command:

```
1 index=* sourcetype=iis cs_method=POST
2 | search *.php OR *.asp OR *.aspx OR *.jsp
```

Results: 187 events (before 5/29/24 1:51:23.000 PM) No Event Sampling

Event details for sourcetype=iis:

Value	Count	%
iis	187	100%

Log entries for sourcetype=iis:

- /001Service.svc/GetList ActivityCorrelationID=34a08616-c3ab-0094-fd76-cc3f52f82808&workFlow=GetCount&ua=&schema=Notification&m\$ExchEcpCanary=Aelf v_c_x2Mmk0.&CorrelationID=<empty>;&cafeReqId=c14cb7c-151b-4a68-a234-5868ea526a5; 443 bellybear\Administrator fe80::50c7:e3a5:fed7:dc195 Mozilla refox/91.0 https://win-a0qkg2as2q7.bellybear.local/ecp/default.aspx 200 0 0 1724
- 2AS2Q7 : source = C:\inetpub\logs\LogFiles\W3SVC1u_ex210908.log : sourcetype = iis /001Service.svc/GetList ActivityCorrelationID=80260cce-f05e-37ed-9f82-00fee7f53ae6&workFlow=GetCount&ua=&schema=Notification&m\$ExchEcpCanary=Aelf v_c_x2Mmk0.&CorrelationID=<empty>;&cafeReqId=d6d0f99-5817-e4e8-939a-d653f6f258; 443 bellybear\Administrator fe80::50c7:e3a5:fed7:dc195 Mozilla refox/91.0 https://win-a0qkg2as2q7.bellybear.local/ecp/default.aspx 200 0 0 15139
- 2AS2Q7 : source = C:\inetpub\logs\LogFiles\W3SVC1u_ex210908.log : sourcetype = iis /001Service.svc/GetList ActivityCorrelationID=b5a8acf-bd28-dc2c-ba31-8312e798885d&workFlow=GetCount&ua=&schema=Notification&m\$ExchEcpCanary=Aelf v_c_x2Mmk0.&CorrelationID=<empty>;&cafeReqId=f899b4a-c438-4802-b887-7a29f7514c5d; 443 bellybear\Administrator fe80::50c7:e3a5:fed7:dc195 Mozilla refox/91.0 https://win-a0qkg2as2q7.bellybear.local/ecp/default.aspx 200 0 0 16835
- CS_url Stem = /ecp/DDIService.svc/GetList : host = WIN-AOQKG2AS2Q7 : source = C:\inetpub\logs\LogFiles\W3SVC1u_ex210908.log : sourcetype = iis

index=* sourcetype=iis cs_method=POST | search .php OR .asp OR .aspx OR .jsp sourcetype=iis

The term "cs_method" means "Client to Server Method" in English.

The screenshot shows a Splunk search interface with the following search command:

```
1 index=* sourcetype=IIS cs_method=POST
```

Results: 862 events (before 5/29/24 9:48:16.000 PM) No Event Sampling

Event details for sourcetype=IIS:

Value	Time	Event
9/8/21 1:07:45.000 PM	2021-09-08 20:07:45 fe80::50c7:e3a5:fed7:dc195 POST /ecp/DDIService.svc/GetList ActivityCorrelationID=34a08616-c3ab-0094-fd76-cc3f52f82808&workFlow=GetCount&ua=&schema=Notification&m\$ExchEcpCanary=Aelf v_c_eReqId=c14cb7c-151b-4a68-a234-5868ea526a5; 443 bellybear\Administrator fe80::50c7:e3a5:fed7:dc195 Mozilla /5.0+(Windows+NT+10.0;+Win64;+rv:91.0)+Gecko/20100101+Firefox/91.0 https://win-a0qkg2as2q7.bellybear.local/ecp/default.aspx 200 0 0 1724	
9/8/21 1:07:25.000 PM	2021-09-08 20:07:25 fe80::50c7:e3a5:fed7:dc195 POST /api/emsmdb/mailboxId=9b329a78-b738-41df-bc61-4c59076bd3b@bellybear.local&CorrelationID=<empty>;&cafeReqId=503e55c3-a52c-4ada-9e64-306874c3d786; 443 BELLYBEAR\HealthMailboxd7fe2ca fe80::50c7:e3a5:fed7:dc195 MapiHttpClient - 200 0 0 2931	

The screenshot shows a Splunk search interface with the following search command:

```
1 index=** sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" *aspx
```

Results: 1 event (before 5/29/24 9:59:02.000 PM) No Event Sampling

Event details for sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational":

Value	Time	Event
attrib.exe -r \\\win-a0qkg2as2q7.bellybear.local\CS\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\i3gfPctk1c2x.	2021-09-08 20:07:25 fe80::50c7:e3a5:fed7:dc195	AttribInfoR:4c8a4985-a32e-48cd-b95d-1864038e3c5:3;RT:Disconnect;CI:5b3ddc12-e0dc-41b2-802d-36b279c0b436:1;CID:e68a0d1e-4a20-4043-a3e8-af2e6f4e9 e8&cafeReqId=503e55c3-a52c-4ada-9e64-306874c3d786; 443 BELLYBEAR\HealthMailboxd7fe2ca fe80::50c7:e3a5:fed7:dc195 MapiHttpClient - 200 0 0 2931

CommandLine details:

Value	Count	%
attrib.exe -r \\\win-a0qkg2as2q7.bellybear.local\CS\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\i3gfPctk1c2x.	1	100%

What is the command line that executed this web shell?

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

Now we need to find out which command is used to run the web shell. We can find this by searching for the name of the .aspx file in Splunk. By checking the results we can find the attrib.exe command used to run the web shell.

The screenshot shows a Splunk search interface with the following details:

- Search bar: index=* sourcetype="WinEventLog:Microsoft-Windows-Sysmon/Operational" *aspx
- Results: 1 event (before 5/29/24 9:59:02.000 PM) No Event Sampling
- Time range: All time
- Event details:
 - Selected Fields: Hashes 1, host 1, Image 1, source 1, sourcetype 1
 - Interesting Fields: CommandLine 1, Company 1
 - CommandLine value: attrib.exe -r \\\win-a0qkg2as2q7.bellybear.local\C\$\\Program Files\\Microsoft\\Exchange Server\\V15\\FrontEnd\\HttpProxy\\owa\\auth\\i3gfPctK1c2x.aspx
 - Event timestamp: 2024-05-29T09:59:02.000Z
 - Event source: WinEventLog:Microsoft-Windows-Sysmon/Operational

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

<https://www.securin.io/articles/is-conti-ransomware-on-a-roll/>

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

Conti - A Cheat Sheet

We analyzed three CVEs being exploited by the Conti group - **CVE-2020-0796**, **CVE-2018-13374**, **CVE-2018-13379**, and here is our analysis about them -

- **CVE-2020-0796** is a critical RCE/PE vulnerability, with a severity score of 10. This vulnerability also goes by the names CoronaBlue and SMBGhost, and was one of the top exploited vulnerabilities of 2020.
- **CVE-2018-13379** is a critical RCE vulnerability that allows for unauthenticated attacks, and has a severity score of 9.8.
- **CVE-2018-13374** is a high-rated vulnerability, with a severity score of 8.8 and can be exploited to compromise web applications.
- **CWE Weakness Categories**
 - **CVE-2020-0796** is categorised under a weakness leading to improper input validation, **CWE-20**
 - **CVE-2018-13379** falls under a weakness leading to improper limitation of a pathname to a restricted directory, **CWE-22**
 - **CVE-2018-13374** belongs to **CWE-732** that leads to assignment of incorrect permissions for critical resources
- **CVE-2020-0796** is present in two Microsoft products - **Windows 10** and **Windows Server 2016**, while the other two exist in Fortinet's **FortiOS**.
- A [patch](#) for **CVE-2020-0796** has been available since March 2020, while vendors recommend upgrading to the latest version of **FortiOS** for the other two vulnerabilities.