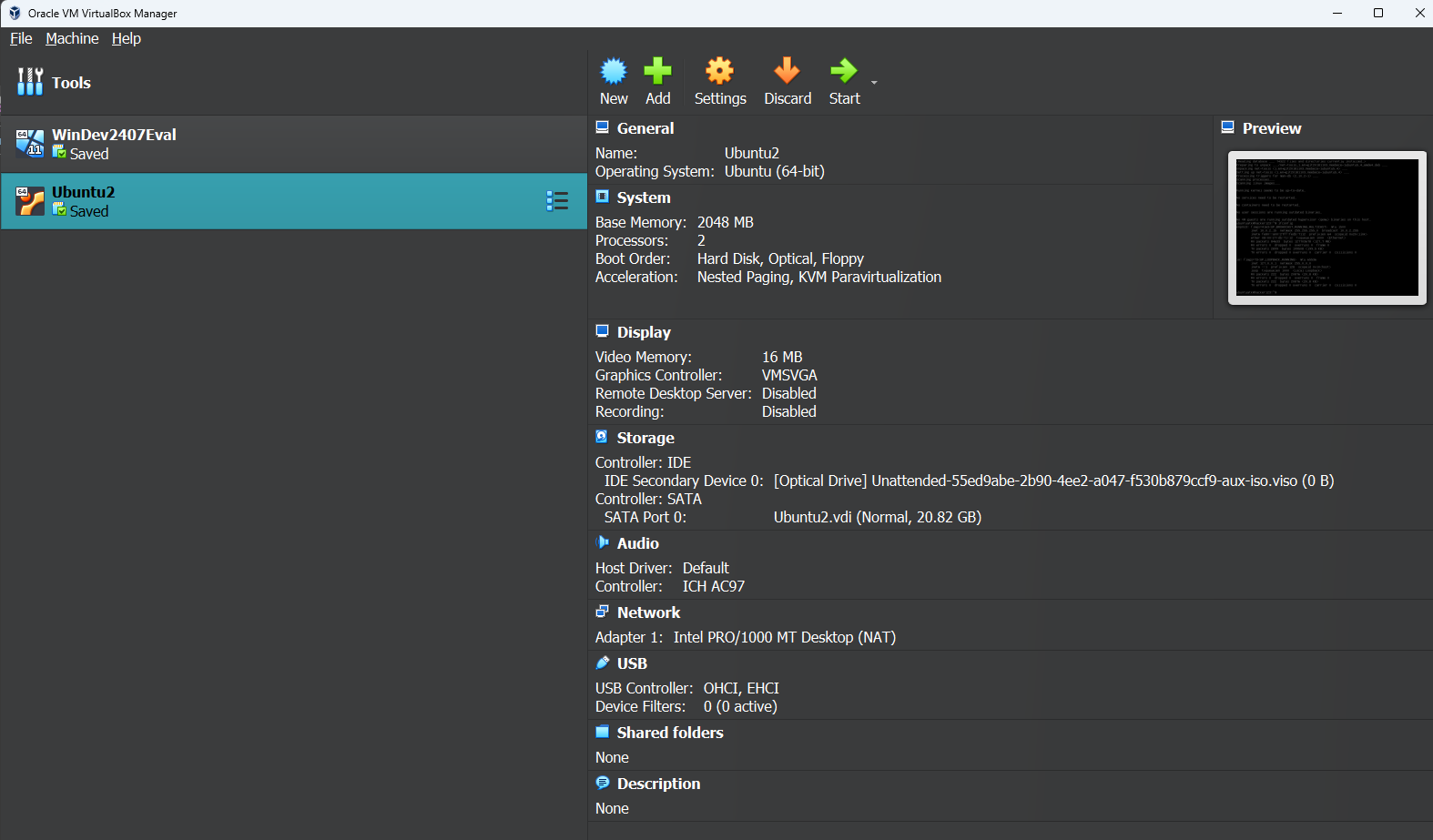
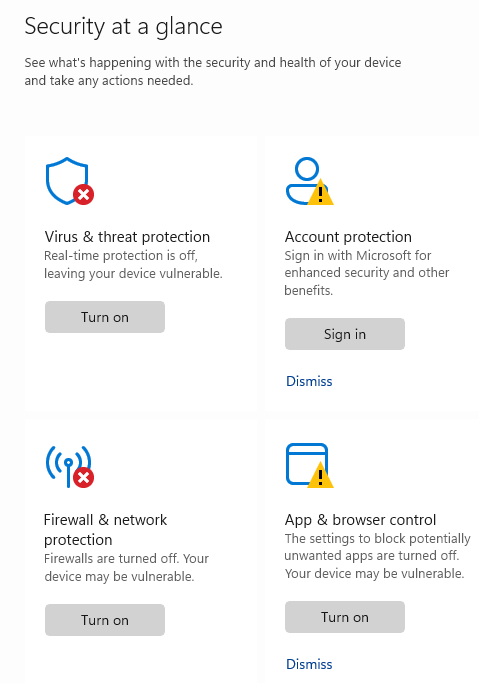
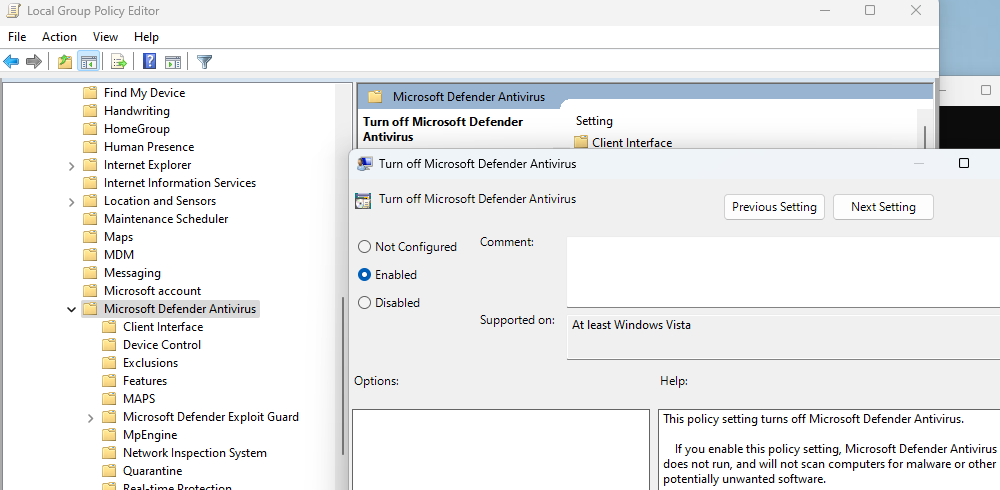
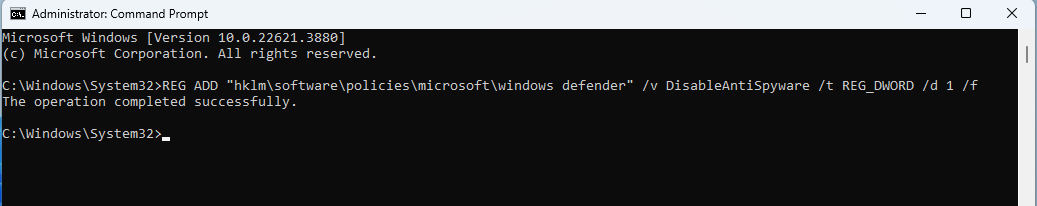
LimaCharlie and Sliver SOC Home Lab



Windows VM (Victim) Setup

1. Downloaded VM and installed W11 VM file.
2. Turned off Windows Defender temp and permanently. I used this guide to help turn it off permanently: https://windowsreport.com/disable-windows-defender-windows-11/  
3. I then went into the Registry to update the Windows boot settings. I changed the start value of all these to ‘4’:

Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Sense

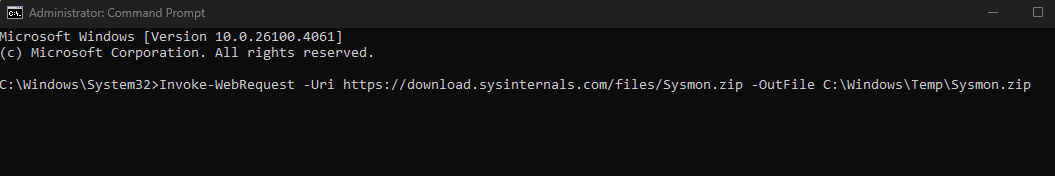
Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\WdBoot

Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\WinDefend

Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\WDNisDrv

Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\WdNisSvc

Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\WdFilter

1. Fasdf
2. Installed Sysmon through powershell
3. Unzipped the Sysmon.exe file and ran Sysmon on the Windows machine. Using the following commands:

#Extracts the Sysmon.zip archive to the Sysmon folder in Temp

Expand-Archive -LiteralPath C:\Windows\Temp\Sysmon.zip -DestinationPath C:\Windows\Temp\Sysmon

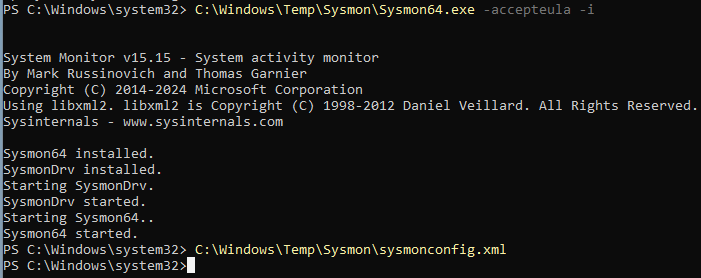
#Downloads the Sysmon configuration file from GitHub to the Sysmon folder

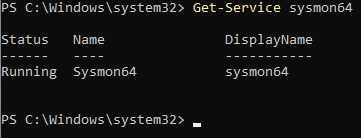
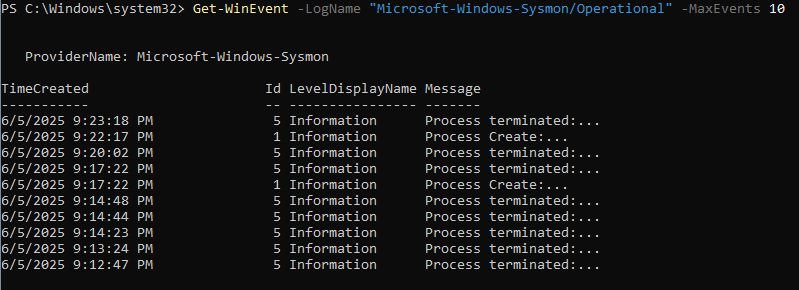
Invoke-WebRequest -Uri https://raw.githubusercontent.com/SwiftOnSecurity/sysmon-config/master/sysmonconfig-export.xml -OutFile C:\Windows\Temp\Sysmon\sysmonconfig.xml

#Installs and configures Sysmon using the downloaded config file, automatically accepting the EULA

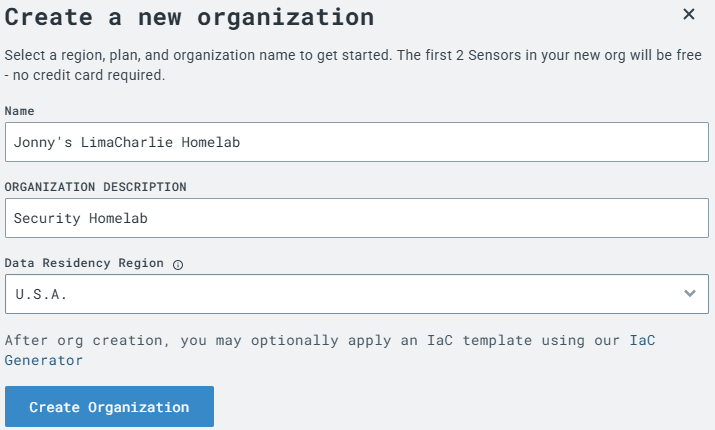
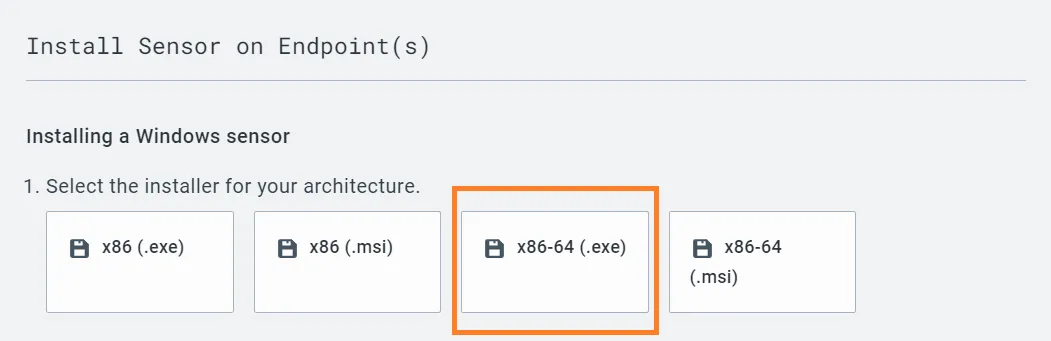
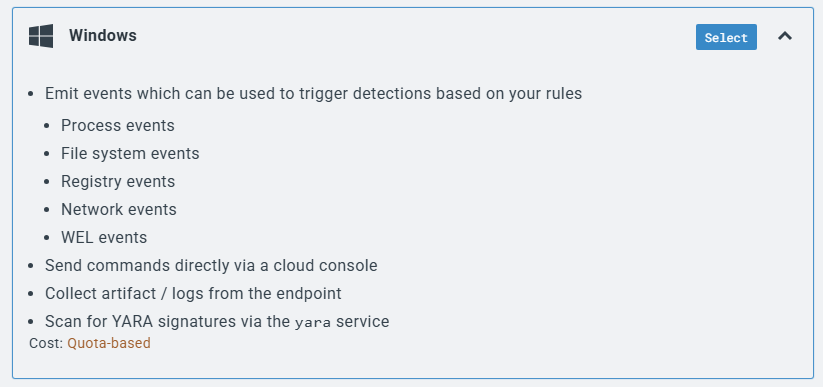
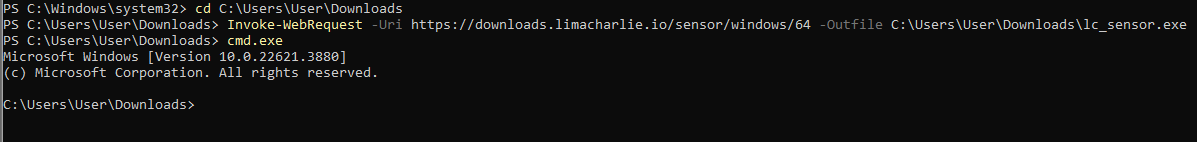
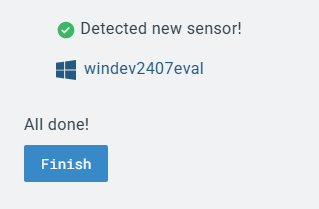
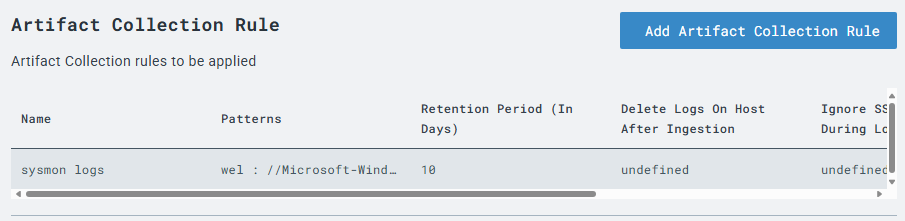
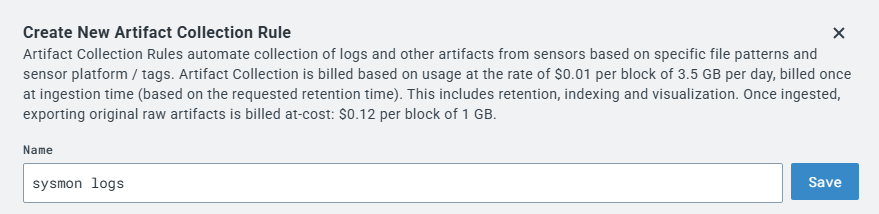
C:\Windows\Temp\Sysmon\Sysmon64.exe -accepteula -i

C:\Windows\Temp\Sysmon\sysmonconfig.xml



1. Confirmed the Sysmon service was running 
2. Confirmed that is was generating logs with at least 10 log results 

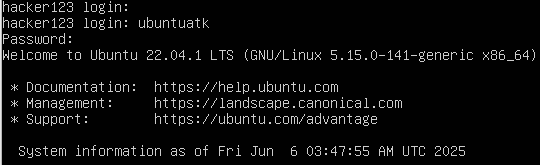
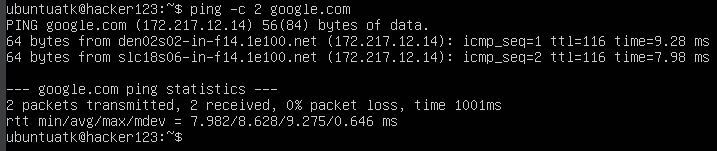
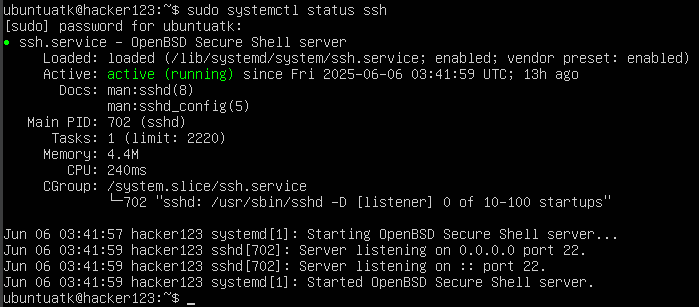
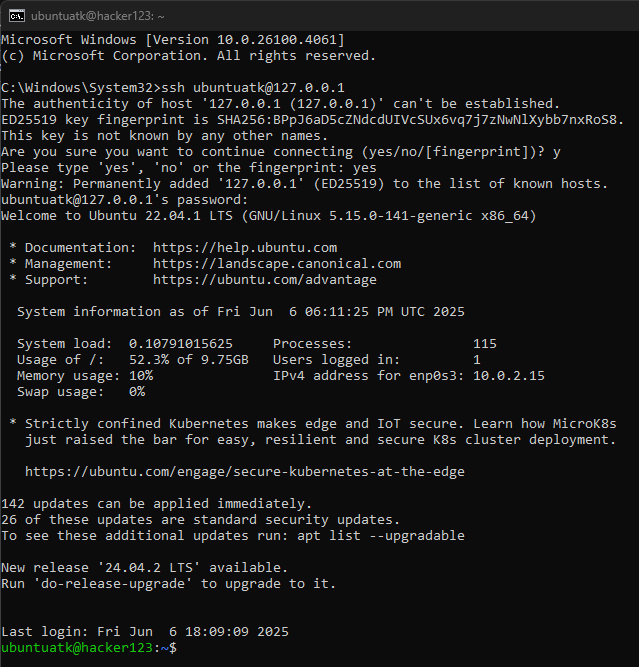
LimaCharlie Account and Sensor Setup

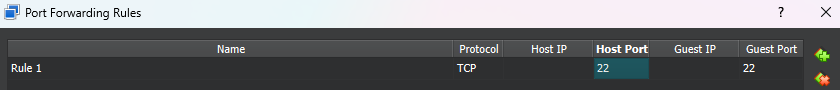
1. Created a LimaCharlie Account 
2. Added a new sensor for Windows (downloaded within Windows VM). Add sensors -> select Windows -> create new -> provide description as Lab VM -> select Create -> select Lab VM -> select the .exe file 
3. Ran the following commands to open the new .exe sensor file within the Windows VM. The first command is to change into the ‘Downloads’ folder and the second command to invoke the .exe file and integrate the new LimaCharlie sensor. 
4. Pasted in the LimaCharlie sensor key into the Windows VM and ran the command to install the LimaCharlie Agent. I had to make sure ‘lc\_sensor.exe’ was added to the first part of the command. 
5. Back on the LimaCharlie site, I added a new Artifact Collection Rule. The pattern command shown below is a directive that tells LimaCharlie what Windows Event Logs (WEL) to collect from the endpoint (sensor), specifically targeting Sysmon logs. It monitors and collects all event entries to centralize Sysmon telemetry from endpoints that can be queried, analyzed, or used for detection rules. 

Ubuntu VM (Attacker) Setup

1. I created a separate VM for Ubuntu.

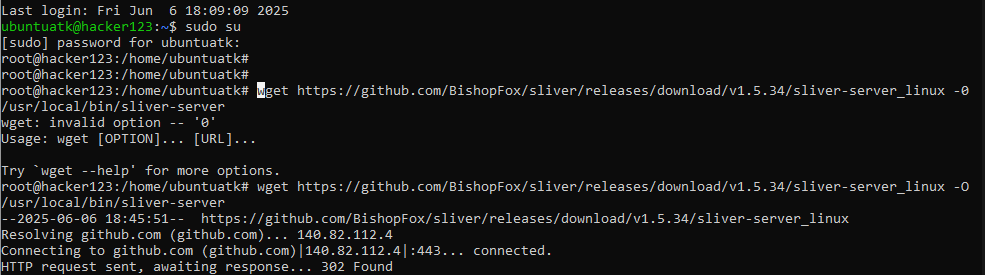
Errors: After installation, I ran into some errors stating that it was missing some packages for ‘ifconfig’ and some other network commands. Command to install network package: ‘sudo apt install net-tools'

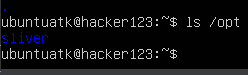
1. Logged into attack machine 
2. Ran a quick ping command to check for network connection. I used the ‘-c’ flag and ‘2’ argument to only include two echo requests. 
3. Checked if the ssh service was running on my Ubuntu VM by running the command ‘sudo systemctl status ssh’ 
4. From my local Windows machine, I used the Command Prompt (CMD) to SSH and logged into the Ubuntu VM. 

Errors: I ran into an issue with SSH because VirtualBox needs a port forwarding rule to connect (SSH) into the VM itself. I created a port forwarding rule on the VM to use port 22 (SSH) between my local machine and the Ubuntu VM. 

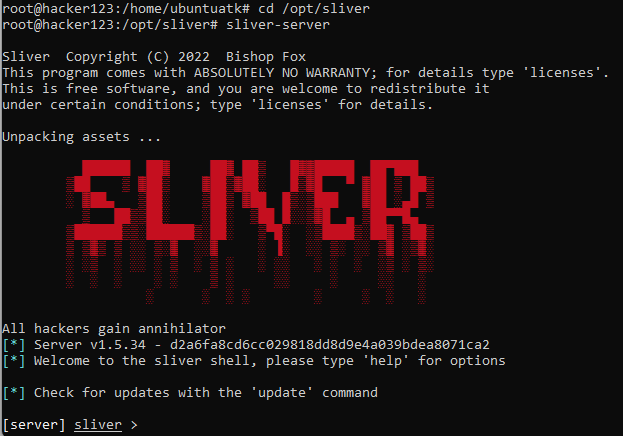
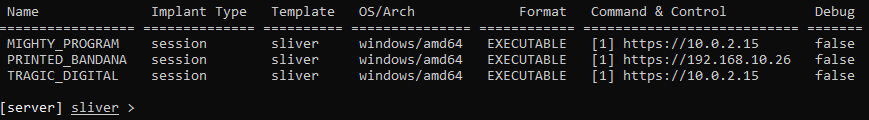
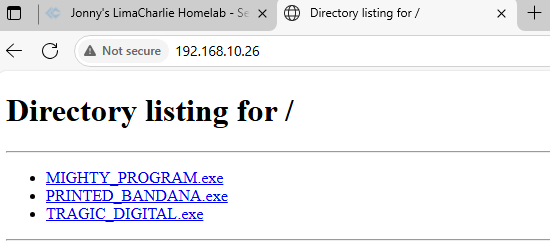
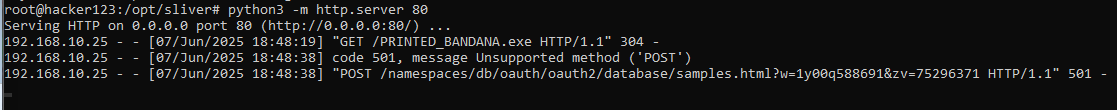
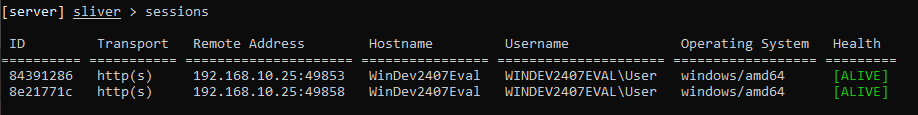
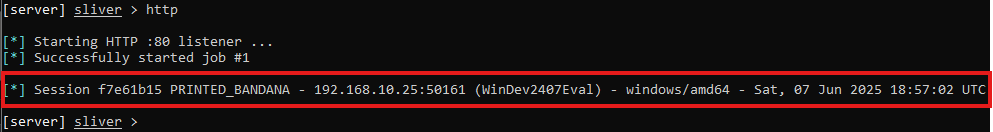
1. I then rooted into the attacker VM using the following commands (**this step is to simulate how an attacker would remote into a victim’s endpoint via SSH via open ports and install a C2 server in their machine**):

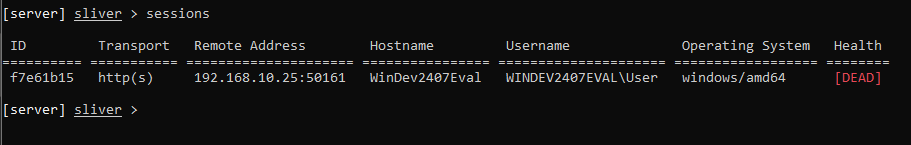
#Downloads Sliver Linux server binary  
wget https://github.com/BishopFox/sliver/releases/download/v1.5.34/sliver-server\_linux -O /usr/local/bin/sliver-server  
#Changes the file permission to become executable  
chmod +x /usr/local/bin/sliver-server  
#Installs mingw-w64 for additional capabilities  
apt install -y mingw-w64



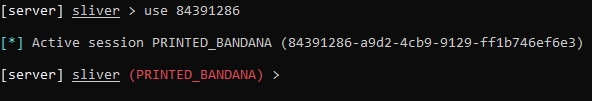
1. Ran another command to create a new directory for Sliver C2: mkdir -p /opt/sliver
2. Checked back on the Ubuntu VM to see if the new directory showed 

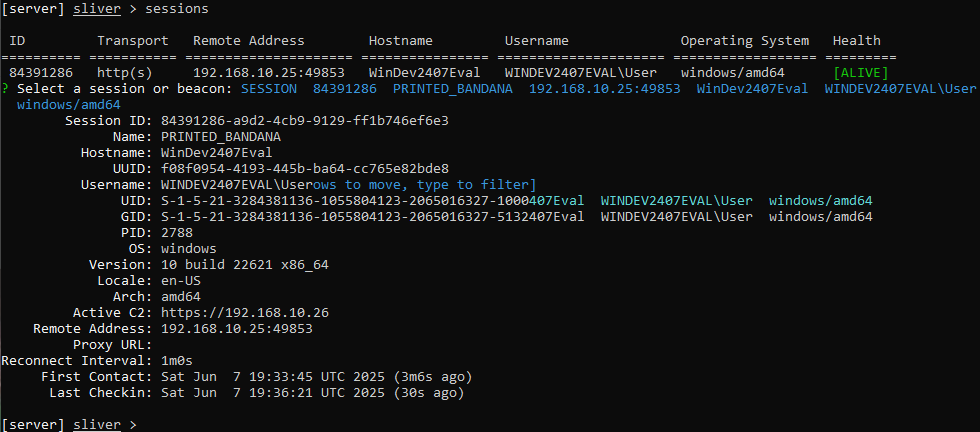
Generating & Monitoring Telemetry

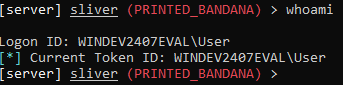
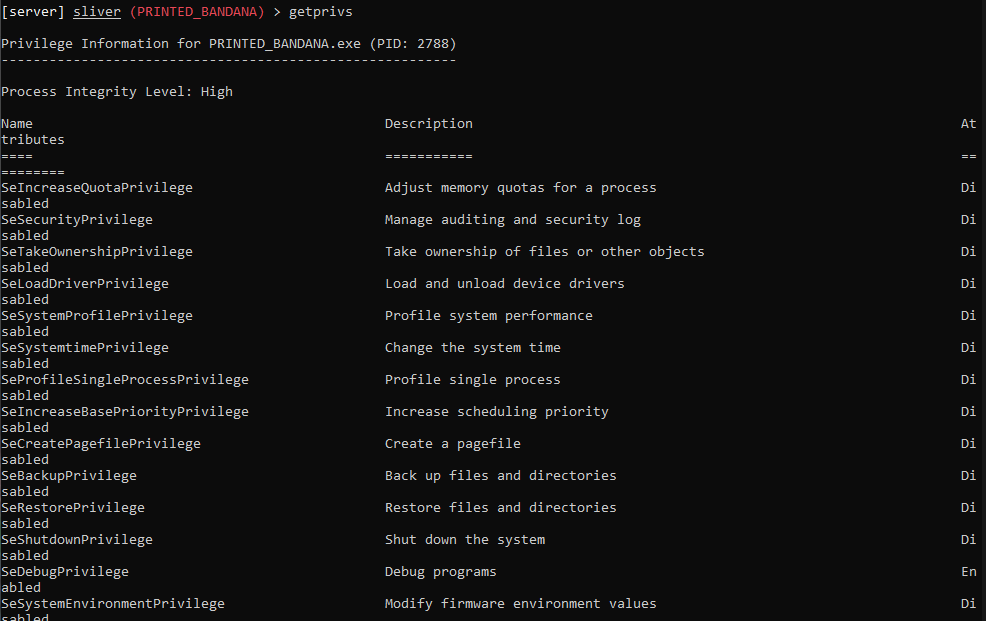
1. From my local machine, I rooted into the Ubuntu VM again via SSH to launch the Sliver C2 server. I first changed the root directory using cd /opt/sliver then launched the package with sliver-server.
2. Next, I ran cd generate –http [Ubuntu\_VM\_IP] --save /opt/sliver to create payload data then checked implants created. 
3. From my SSH session, I ran this command to transfer the created payload to the Windows victim VM. 
4. AI then opened a browser on the Windows VM and went to ‘http://192.168.10.26’ to download the Sliver payload on the victim machine. 
5. I took a snapshot to save a clean version of the Windows VM.
6. I opened the file and noticed my SSH session was outputting these commands: 
7. I closed that process and then reopened Sliver again to start the Sliver HTTP listener to catch reverse shell from the Windows VM.
8. Back on the Windows VM, I ran the following command: 
9. Looking back at the Sliver session, I noticed that I got a reverse shell of the Windows VM. 

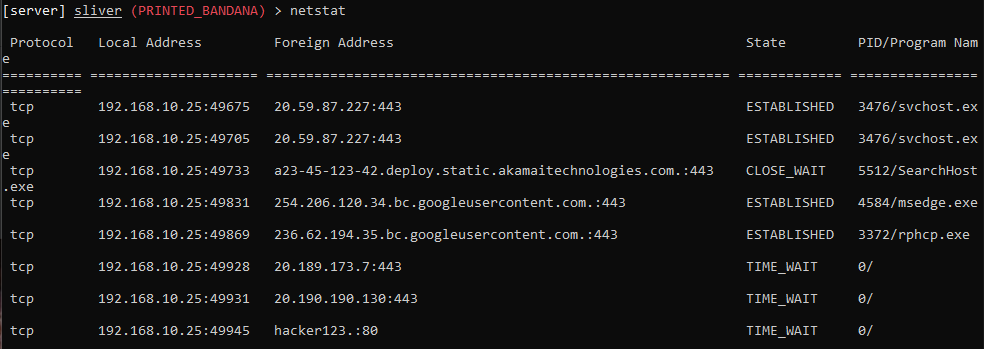
Errors: I kept running into an issue with keeping the Sliver session alive on the Windows VM. It looks like Windows Defender kept turning back on and deleting the payload file. This would keep turning off the session from my Sliver session 

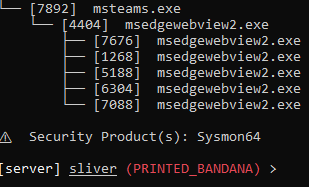
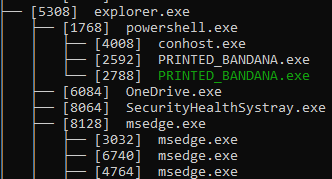
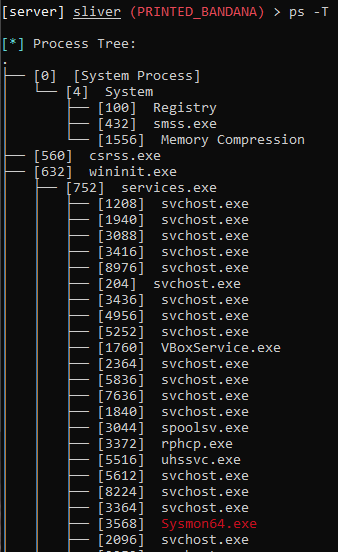
1. With a live payload session on the Windows VM, I ran some comands to see telemetry. Notice that Sliver cleverly highlights its own process in green and any detected countermeasures (defensive tools) in red:

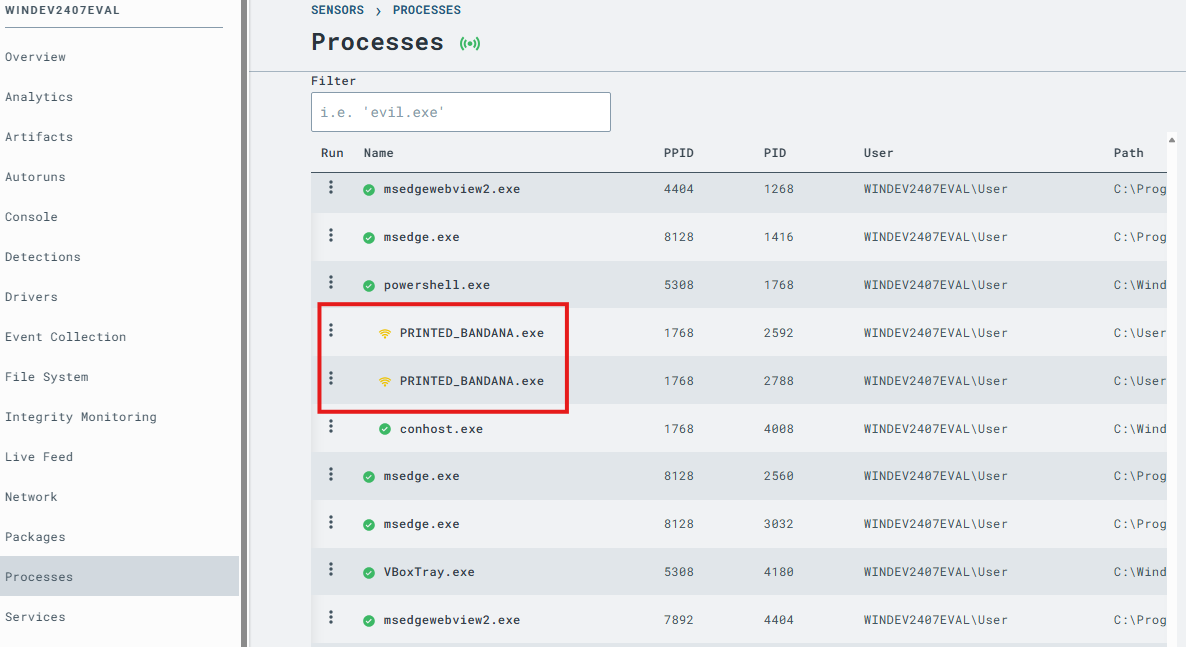
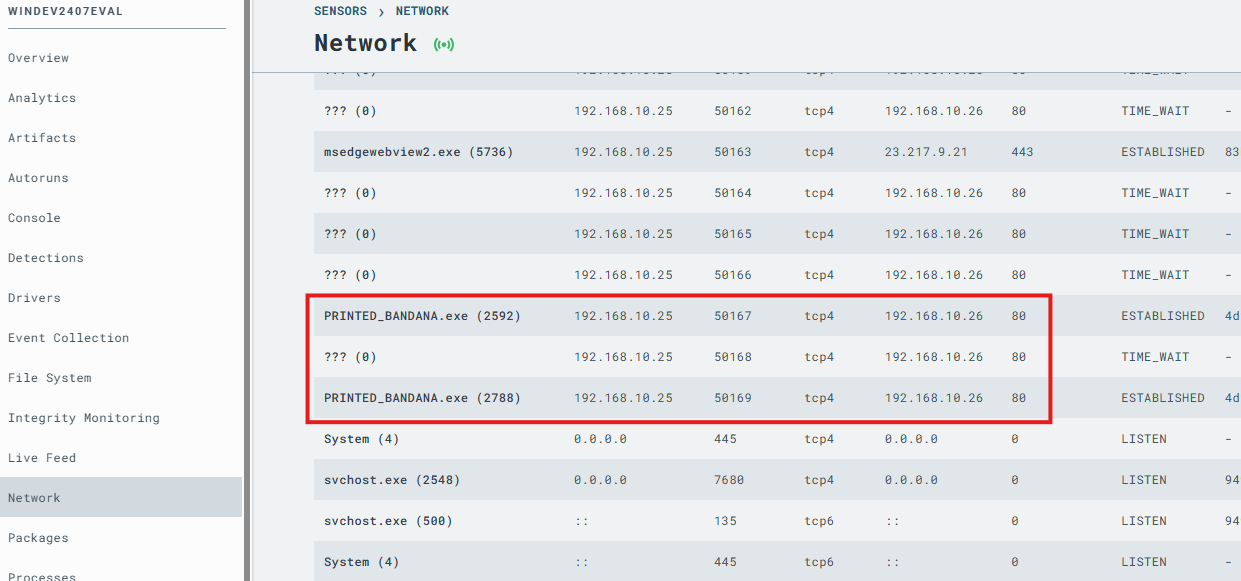
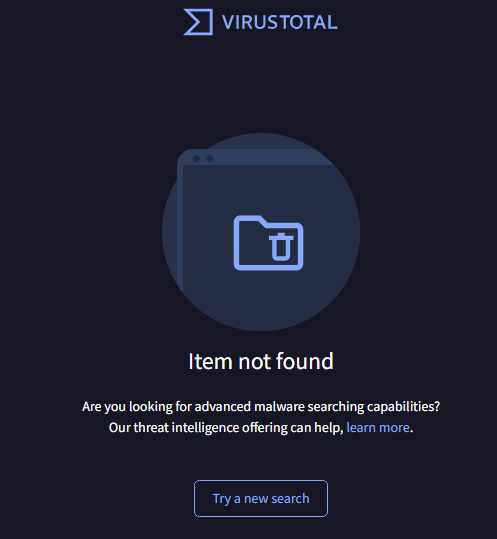


info #(basic info about session)  
  
whoami #(user of the session)

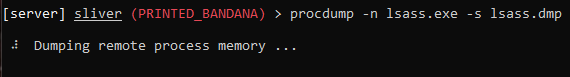
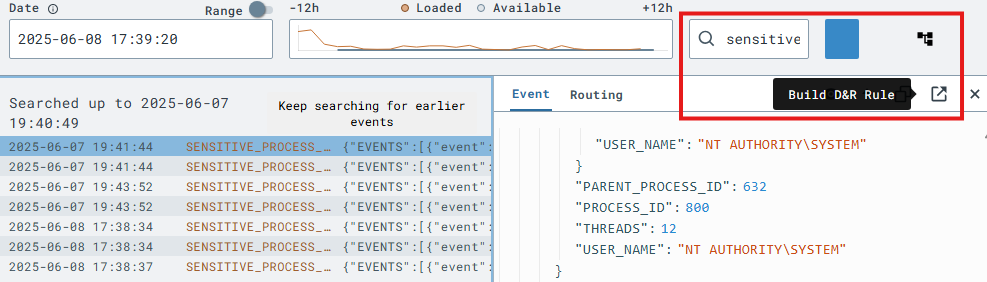
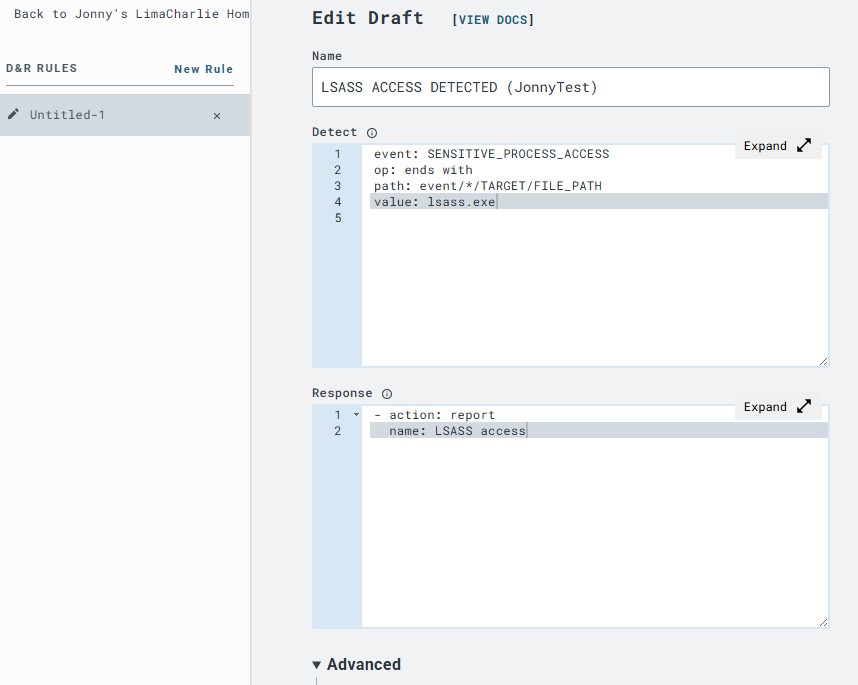
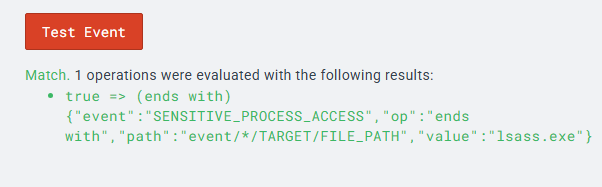
  
  
getprivs #(privilages of the user)  
  
pwd #(working directory)

  
  
netstat #(network connections on host i.e. windows)  
  
ps -T #(running processes on VM)

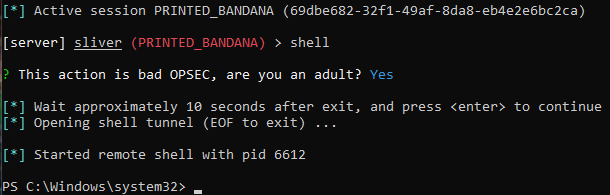
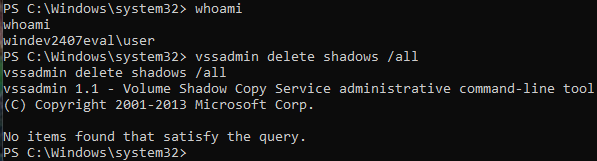
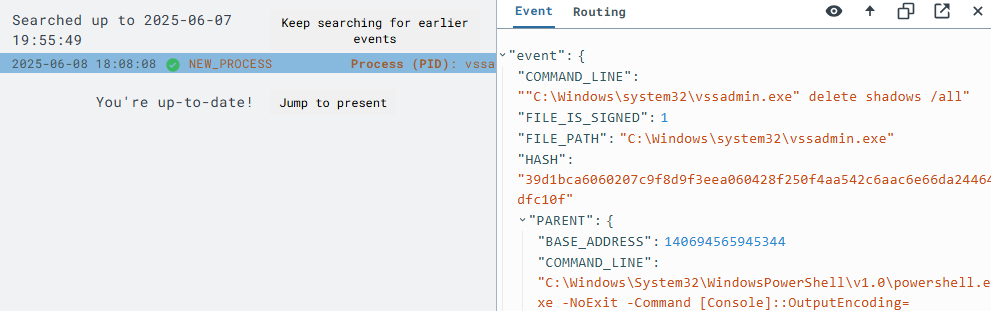
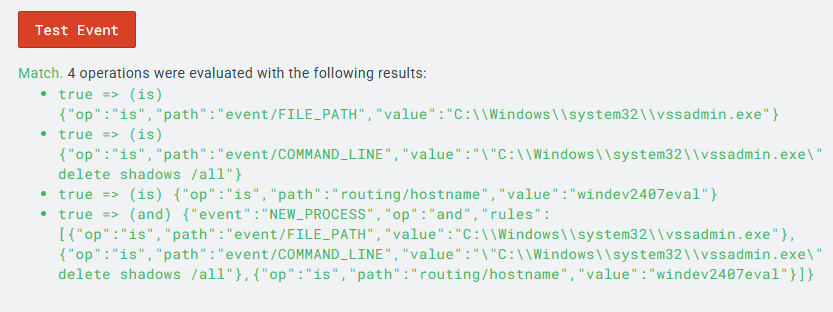
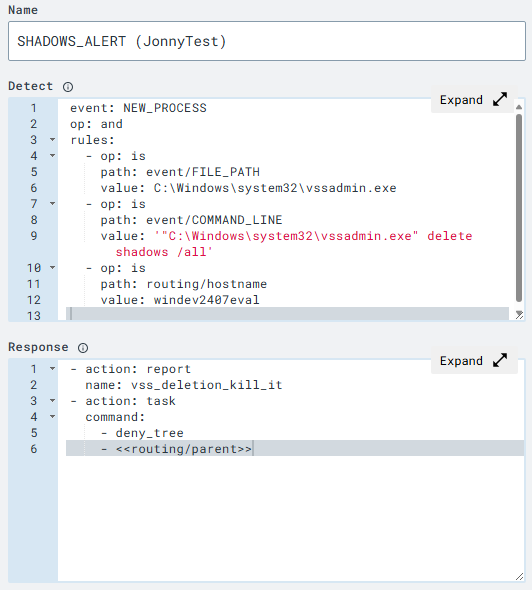
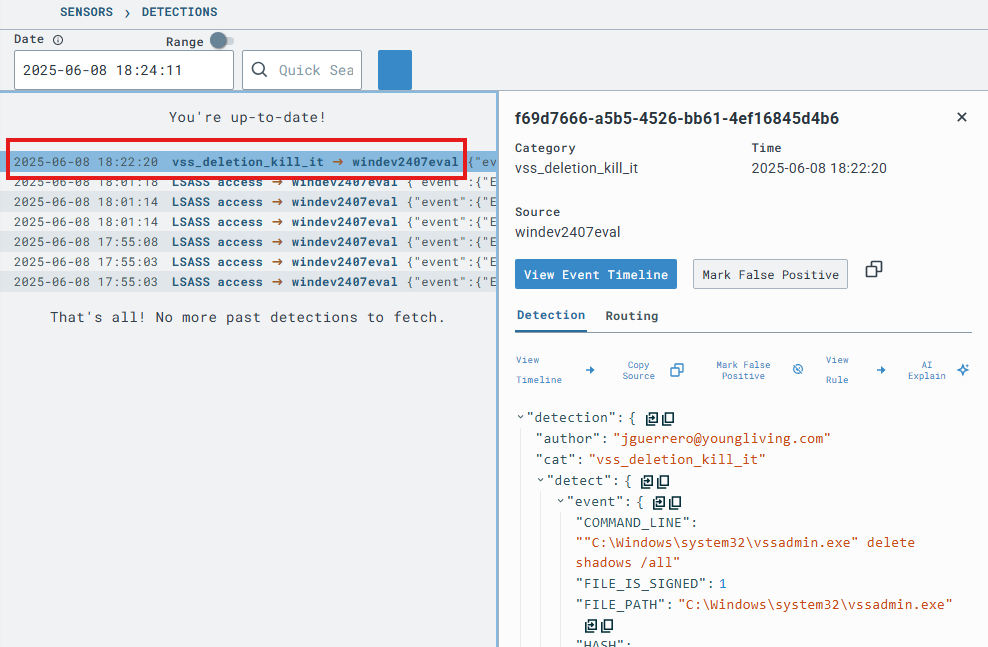
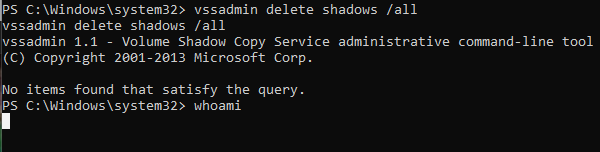


1. Next, I checked back on LimaCharlie to view some telemetry through the web portal. Sensors -> Active Windows Sensor
2. Under the ‘Processes’ tab I could see the unsigned processes that showed suspicious activity in the logs. In our case, our C2 implant is not signed & also is active on the network. Now in network section we see that the process CROWDED is communicating with 192.168.0.134 which is suspicious. 
3. Under the ‘Network’ tab, I could also see that the payload was communicating between the two different IP addresses from my VM machines 
4. Under the ‘File System’ tab, I navigated to the payload file and ran its hash through VirusTotal 
5. Under the ‘Timeline’ tab, I could see a few ‘SENSITIVE\_PROCESS’ detections from LimaCharlie which indicated that Sliver C2 was looking to see something it wasn’t authorized to see.
6. Asdfasdf
7. Fdasf

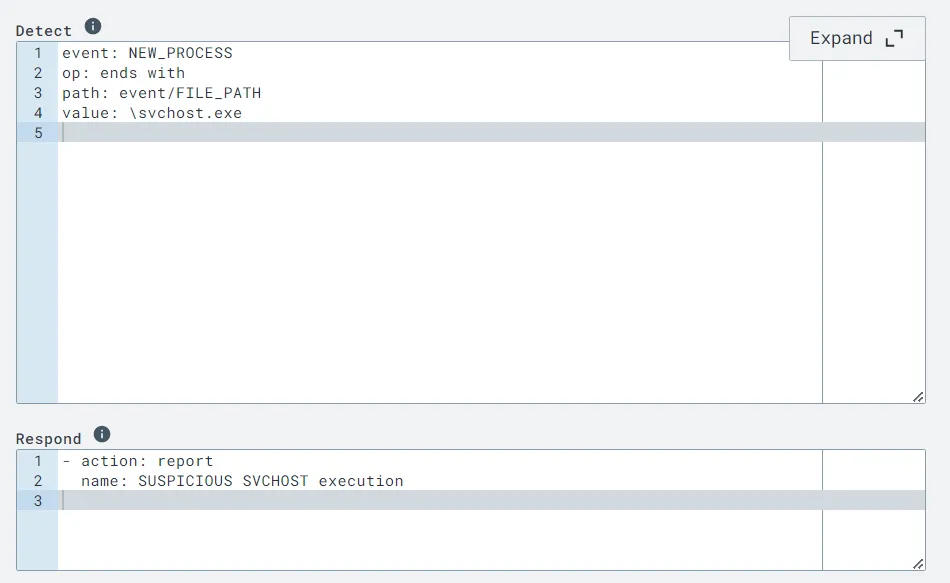
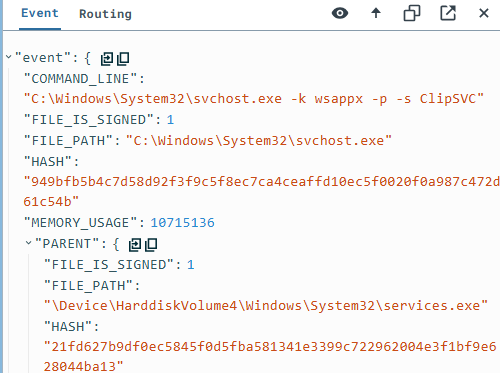
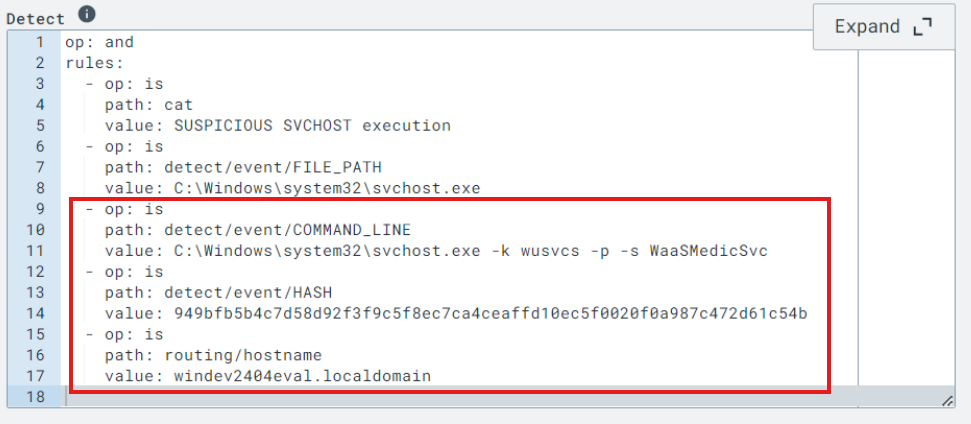
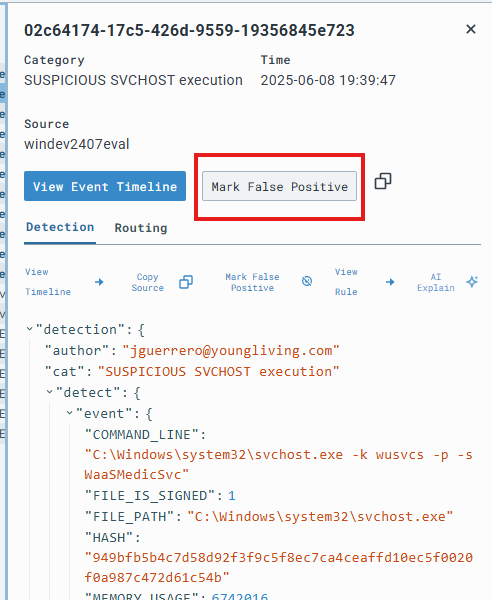
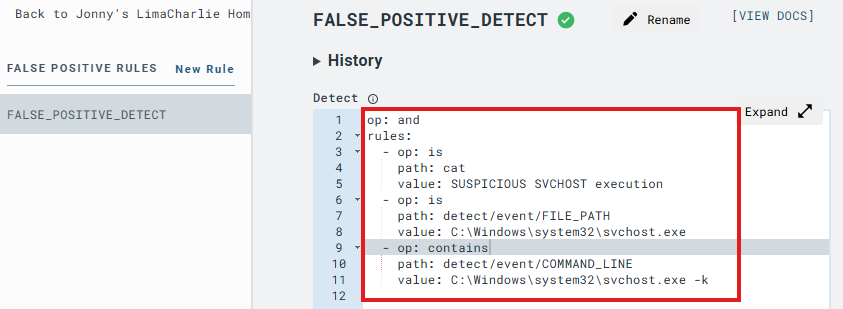
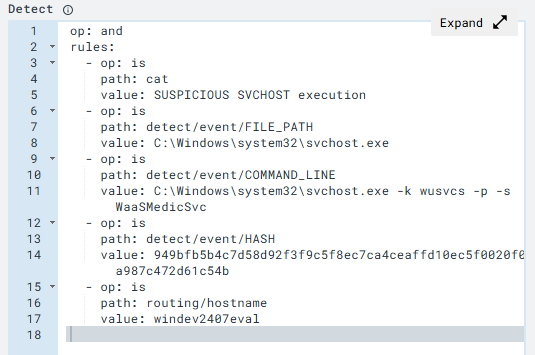
Detecting Attacks & Generating an Alert in LimaCharlie

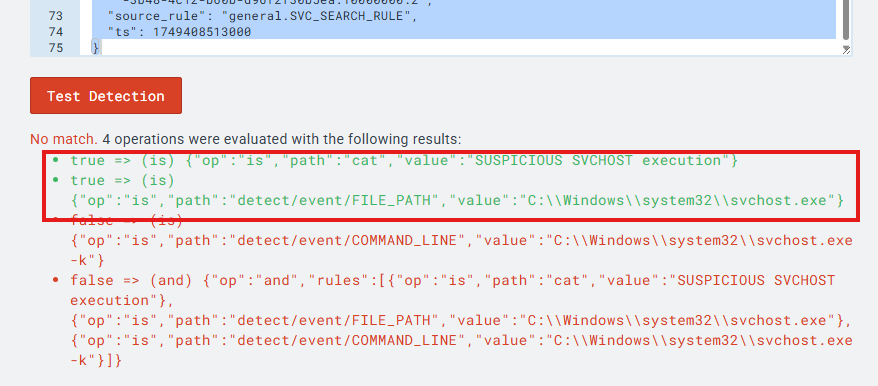
1. To emulate adversaries attack, I ran the following command to dump credentials from my SSH session: 
2. From LimaCharlie, I went to timeline & search “SENSITIVE\_PROCESS\_ACCESS” & select one & select create D&R rule. 
3. I created new detection & response rules to capture the LSASS activity from the Windows VM 
4. I tested the new detection rules from LimaCharlie. 
5. I reran the procdump command from my SSH session. Under the ‘Detections’ tab I confirmed that my new detection rule was capturing the procdump commands from my SSH session. 
6. asdfadsf

Detecting Attack & Blocking it

1. From the SSH session, I dropped a shell from sliver. 
2. I ran the following command to delete shadow copies in the VM 
3. I searched for the shadows deletion event in LimaCharlie and created a new detection rule to capture the event 
4. I added the following rule command under the response rule section. 
5. I reran the command to delete shadows and noticed the new rule appeared under the Detections tab. 
6. I tried running the command again and noticed that the command would hand when trying to use ‘whoami’ confirming the new detection rule had blocked the attempt. 

Tuning False Positives

1. I created a new rule in LimaCharlie to create a lot of noise. The rule I set up creates an alert whenever svchost process is initiated.
2. I selected a new detection from the Timeline and used the D&R creation tool to setup the new detection rule 
3. I created another rule to detect False Positives. This rule will work in conjunction with above rule to suppress false positives. At most times system runs the svchost with -k argument which specifies its a shared process & bad actors most of the times don't use it also when svchost runs from system32 its legit. 
4. I went back to the same detection rule and added the following new lines to the detection and response sections. 
5. I opened the detection and marked it as a ‘False Positive’ to test the new detection. 
6. Detection rule before and after configurations. I kept the category as is as “sus svchost execution”, file path to be from “system32” as that's the non-suspicious but in command line we will only keep contains svchost.exe -k. we will delete hash section as it might change & lastly we want this rule for all hosts so we delete last one as well. 
7. I went under the Automations tab to view the new False Postive Rule.
8. I went under the Target Detection section to paste in a random event to see if it would block the detection. 

Errors: My detection script wasn’t grabbing commands that include svchost.exe as the op value was set to ‘is’ instead of ‘contains’. This made the detection think it had to be an exact path value instead of detecting ALL values that had that path name. 

1. The new detection rule could be used to block any detections that have ‘Suspicious SVCHOST as the Category name, file path as system32, and if the command line includes ‘svchost.exe’.

Lessons Learned

When logging back into Sliver, I had to rerun the http listener command to view active session from the Windows VM. 