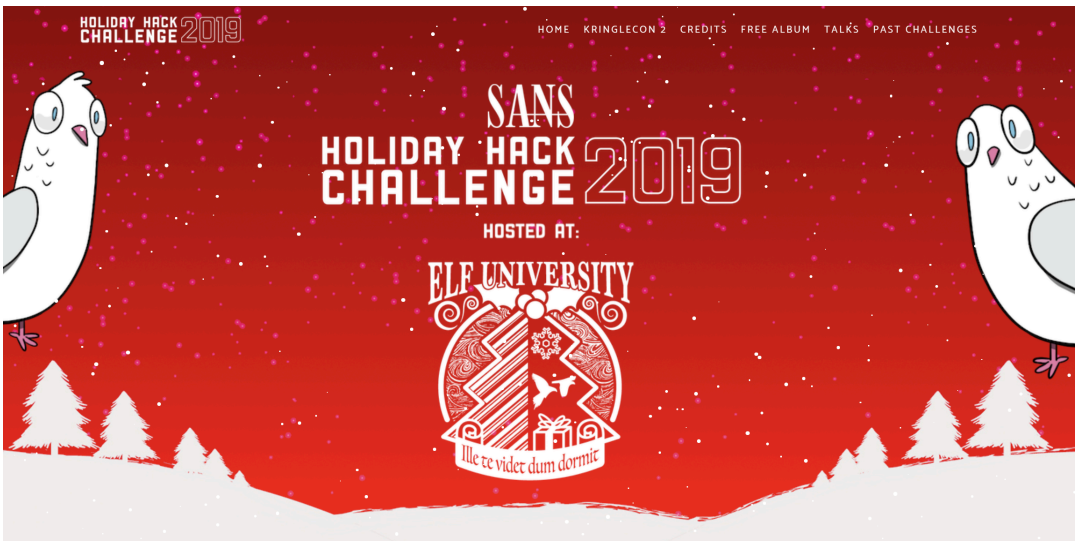


Introduction

SANS holiday hack challenge known as “KringleCon” is a cybersecurity conference developed by SANS to provide a wealth of hands on workshops to teach about cyber security and the many disciplines.

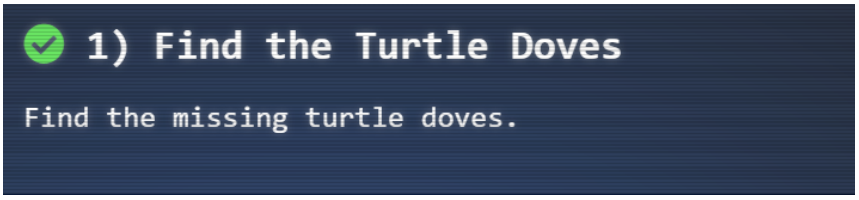
There are many challenges, with a range of activities such as low level shell escaping, basic threat hunting, and eventually moving across to SQL injection and malware analysis.

Below is our main objective board, and challenges 3 - 12 contain secondary objectives that are completed to earn hints for how to complete the main objective. Because we are doing this for education purposes, we will be going through all the challenges.

Challenges Index:

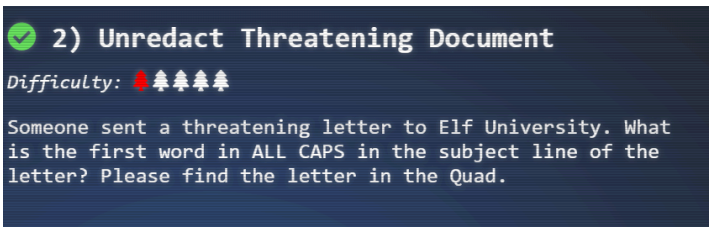
This is the starter challenge to get you familiar with the map.

Simply walking into the courtyard will reveal Santa and the objective will be completed.



This is another starter challenge to get you familiar with the map.

You will find the two turtle doves in the north most building by the fireplace.



Summary

A challenge that will yet again require us to explore the campus, and utilize file forensics

1. Exploring the campus further will eventually lead to finding a letter
2. The letter has been redacted to hide the subject line and body contents
3. Utilizing a tool called pdf wondershare, we can bypass the redaction and read confidential content

We find the file in the top left corner of the campus behind a tree.

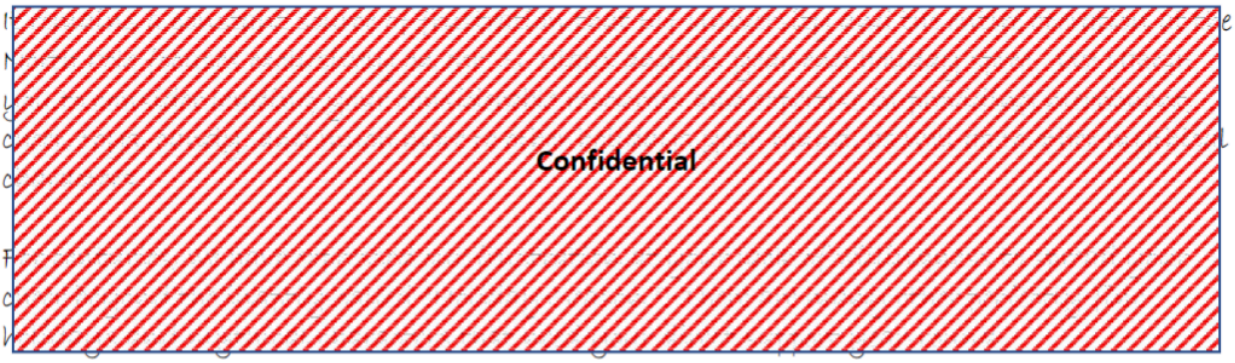
Opening The document provides us with the following

Date: February 28, 2019

To the Administration, Faculty, and Staff of Elf University
17 Christmas Tree Lane
North Pole

From: A Concerned and Aggrieved Character

Attention All Elf University Personnel,



If you do not accede to our demands, we will be forced to take matters into our own hands. We do not make this threat lightly. You have less than six months to act demonstrably.

Sincerely,

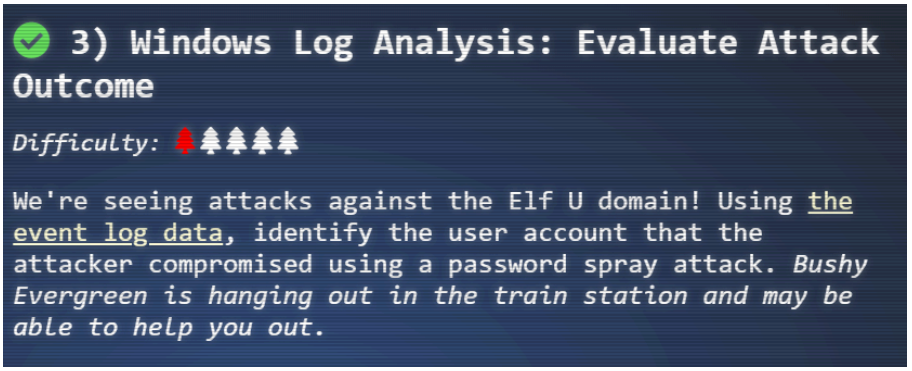
--A Concerned and Aggrieved Character

You can download pdf wondershare from
https://pdf.wondershare.com/

This will allow us to edit the pdf document and bypass insecure redaction, by copying and pasting the plain text.

The answer is the first word in the subject line in all capitals.

DEMAND



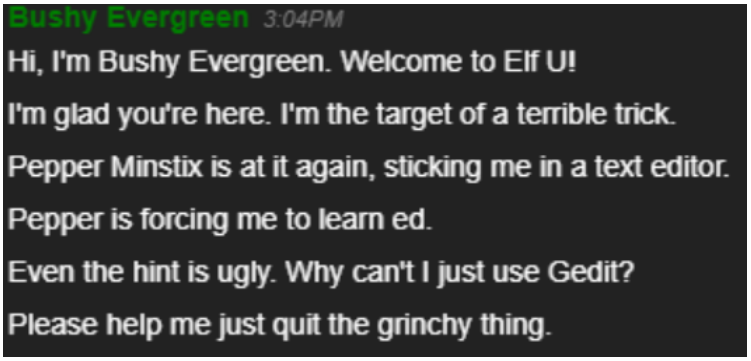
Summary

A challenge that requires log analysis

1. Complete bushy evergreens secondary objective.
    a. Text editor escaping
2. Complete Primary Objective
    a. Threat hunting event logs

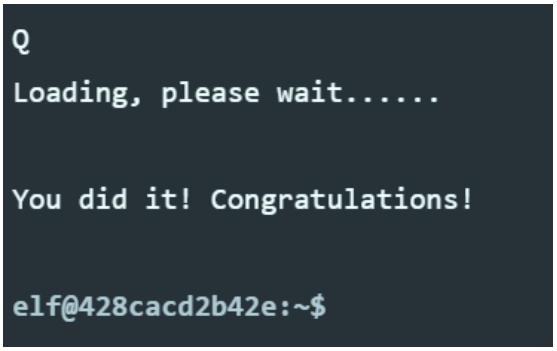
Secondary Objective: Escape Ed - CranPi

Upon talking to Bushy, we learn that Pepper forced Bushy to learn how to use the ed text editor and has left Bushy stuck.



Upon opening the terminal we are provided with the following output:

This seems simple enough, we just have to exit the terminal. So the first thing we try is typing q and pressing enter and sure enough it works. Just like vi editor.



Primary Objective:
Windows Log Analysis: Evaluate Attack Outcome

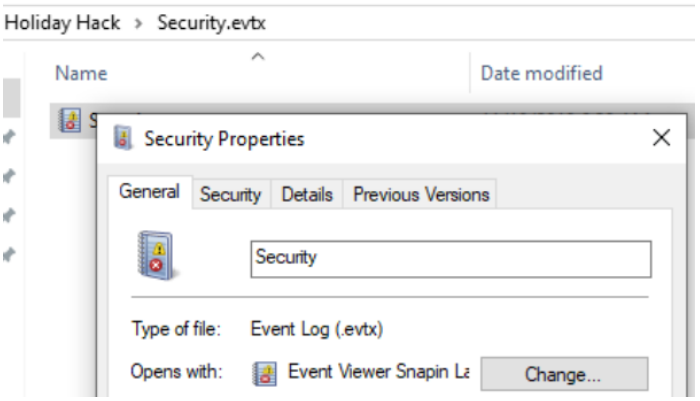
Now that we have completed the escape Ed Terminal, we can talk to Bushy again for more hints.

In this challenge, it is important to research event ID's to learn about some of the more important ID's when threat hunting.

Bushy evergreen provides a hint about a tool called "Deep Blue CLI" which is a password spraying tool.

Taking the time to research this tool, will also bring to your attention to Event ID 4776 - Credential validation, which we look for and find in the system log files.

Let's open the system even file.



4624	Logon
4672	Special Logon
4776	Credential Validation
4625	Logon
4648	Logon
4625	Logon
4648	Logon
4625	Logon

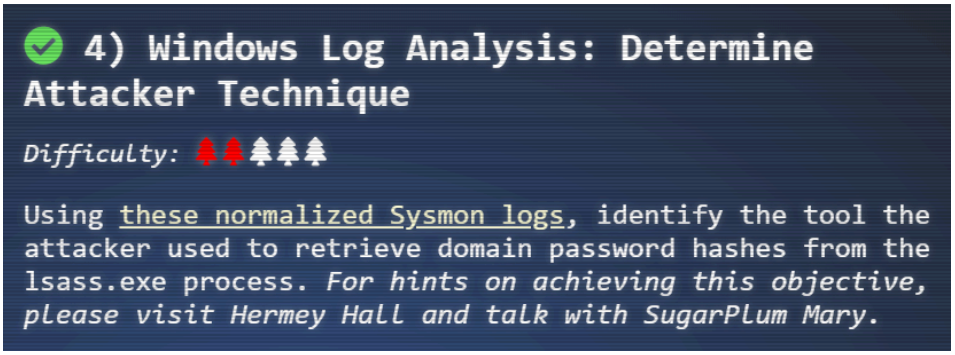
EventData

PackageName	MICROSOFT_AUTHENTICATION
TargetUserName	supatree
Workstation	WORKSTATION
Status	0x0

Event ID 4776 Credential Validation is found down the list.

Expanding the details tab, give us the target username and objective answer.

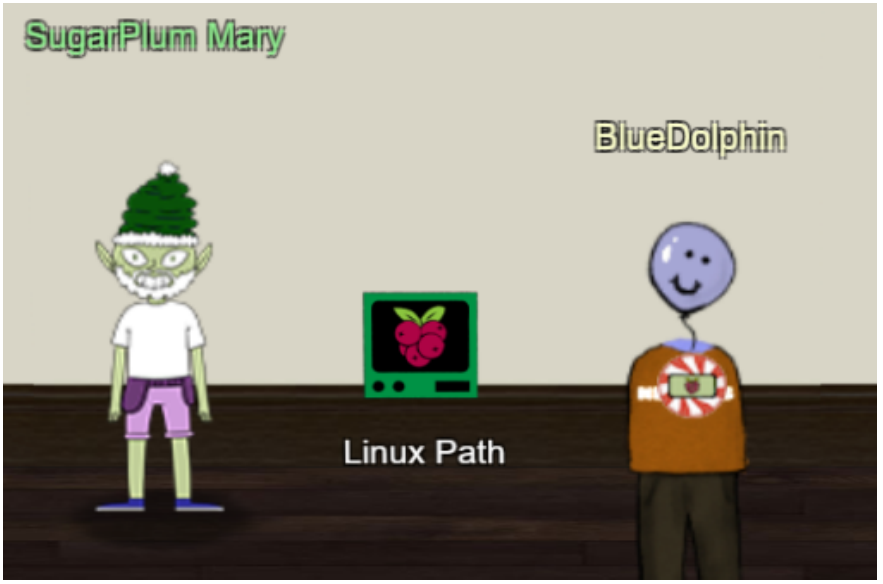
Supatree



Summary

A challenge that requires log analysis

1. Secondary objective
    a. Linux file paths
2. Primary objectives
    a. Threat hunting log analyses



Talking to sugar plum Mary gives us the dialogue with some context for this secondary challenge.

Oh me oh my - I need some help!

I need to review some files in my Linux terminal, but I can't get a file listing.

I know the command is *ls*, but it's really acting up.

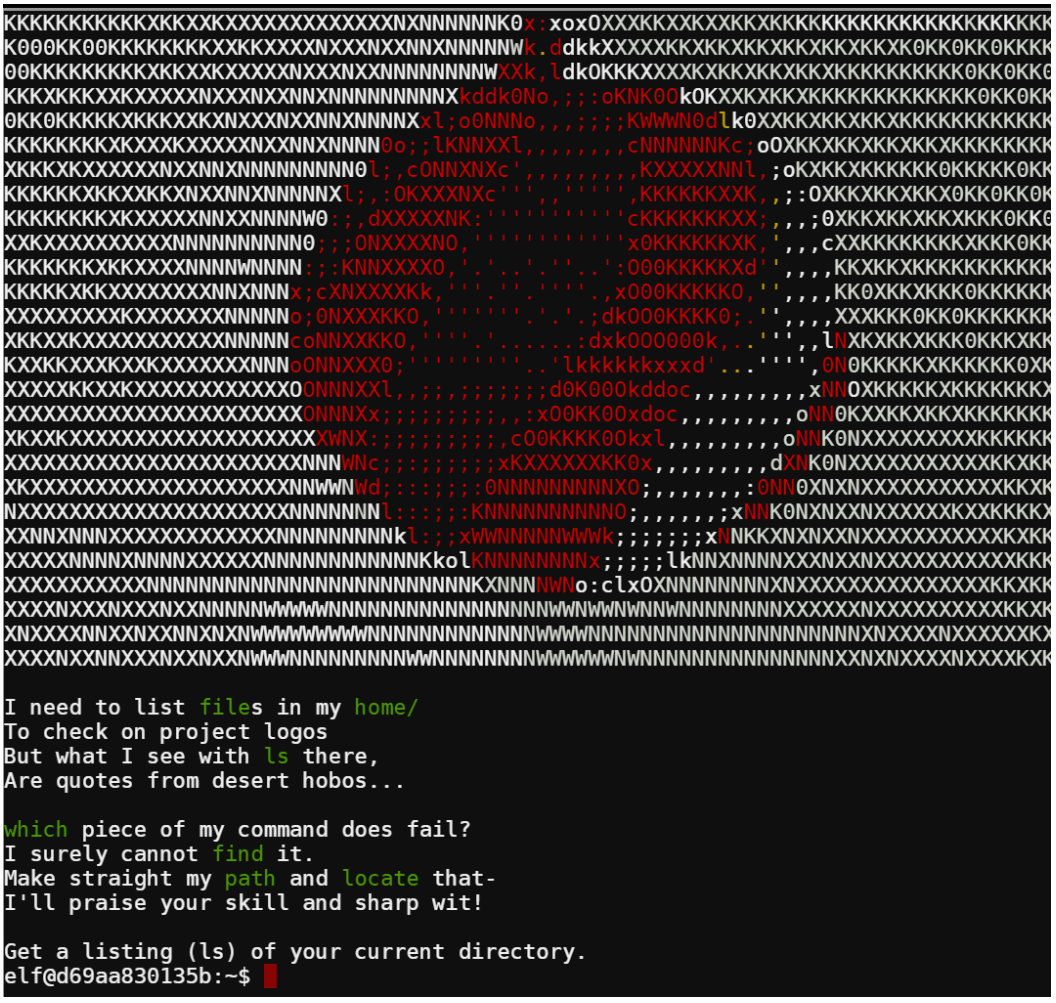
Do you think you could help me out? As you work on this, think about these questions:

1. Do the words in green have special significance?

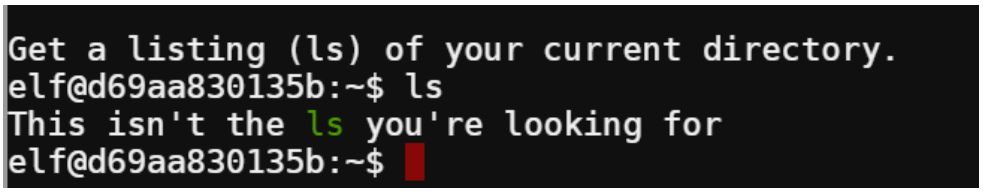
2. How can I find a file with a specific name?

3. What happens if there are multiple executables with the same name in my $PATH?

At first, this seems easy, as we only need to list some files on Linux for Mary. Opening the terminal provides us with the following output.



Trying to run **LS** provides us with the following output.



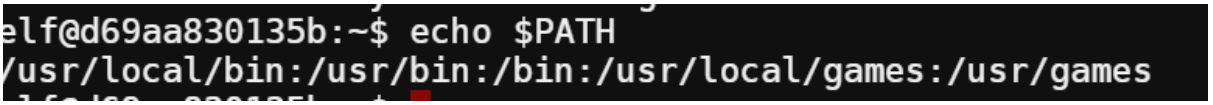
So another binary was executed and we were denied functionality of the command. Mary had hinted for us to pay attention to the words in green, and then posed the question “What happens if there are multiple executables with the same name in the **$PATH**?”.

A PATH is an environment variable that the majority of unix operating systems use to tell the shell which directories to search for executable files in when command line arguments are passed by the user.

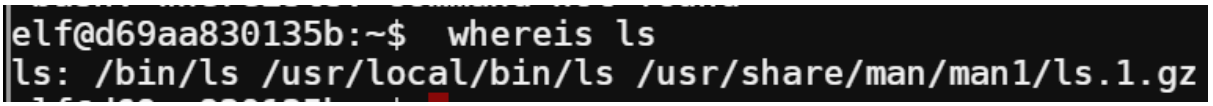
The user’s PATH consists of colon-separated paths in a plain text file. Anytime a user types a command-line argument that is not found in the shell it searches those paths, unless the path is specified in the command line argument.

We can echo the path variable to find those absolute paths with

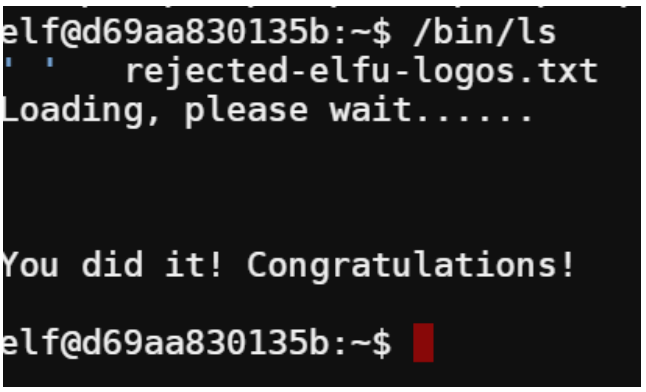
Echo $PATH



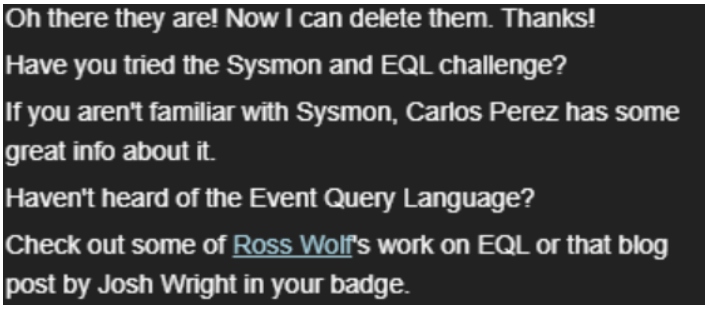
Let’s try and find the location of the ls command now with the **whereis** command.



We run ls from the first found path /bin/ and get the flag.

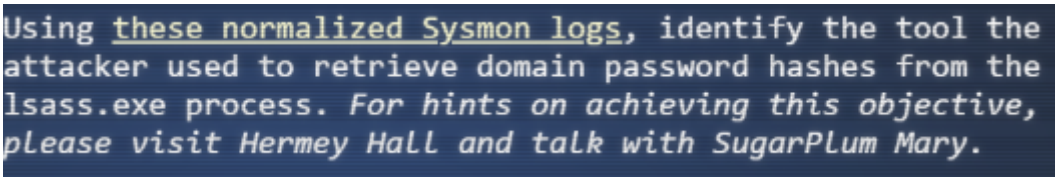


Upon completing this challenge we can talk to sugar plum mary to continue with the main objective.



Main Objective

For this next part, we have to identify the tools used to retrieve the domain password hashes from lsass.exe process.



Upon downloading the files we see it is in a JSON format.
Well, sugar plum mary provided us with a hint about using Sysmon by Carlos Perez, EQL Threat Hunting as well as some of Ross Wolf's stuff on EQL.

After some reading and research, we download the tools and query for ntdsutil which is a common way to back up and AD environment, and within that backup are password hashes.

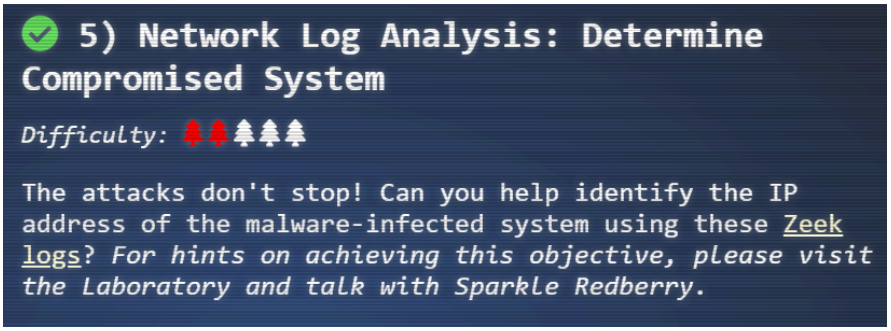
We query the file for ntdsutil and get the following

```
"command_line": "ntdsutil.exe  \"ac i ntds\" ifm \"create full c:\\hive\" q q
"event_type": "process",
"logon_id": 999,
"parent_process_name": "cmd.exe",
"parent_process_path": "C:\\Windows\\System32\\cmd.exe",
"pid": 3556,
"ppid": 3440,
"process_name": "ntdsutil.exe",
"process_path": "C:\\Windows\\System32\\ntdsutil.exe",
"subtype": "create",
"timestamp": 132186398470300000,
"unique_pid": "{7431d376-dee7-5dd3-0000-0010f0c44f00}",
"unique_ppid": "{7431d376-dedb-5dd3-0000-001027be4f00}",
"user": "NT AUTHORITY\\SYSTEM",
"user_domain": "NT AUTHORITY",
"user_name": "SYSTEM"
```


That is our answer, ntdsutil.

Another way to solve this was to simply open the .json file in notepad ++ and walkthrough the files until you found ntdsutil at the bottom.


```
"command_line": "ntdsutil.exe  \"ac i ntds\" ifm \"create full c:\\hive\" q q",
"event_type": "process",
"logon_id": 999,
"parent_process_name": "cmd.exe",
"parent_process_path": "C:\\Windows\\System32\\cmd.exe",
"pid": 3556,
"ppid": 3440,
"process_name": "ntdsutil.exe",
"process_path": "C:\\Windows\\System32\\ntdsutil.exe",
"subtype": "create",
"timestamp": 132186398470300000,
"unique_pid": "{7431d376-dee7-5dd3-0000-0010f0c44f00}",
"unique_ppid": "{7431d376-dedb-5dd3-0000-001027be4f00}",
"user": "NT AUTHORITY\\SYSTEM",
"user_domain": "NT AUTHORITY",
"user_name": "SYSTEM"
```

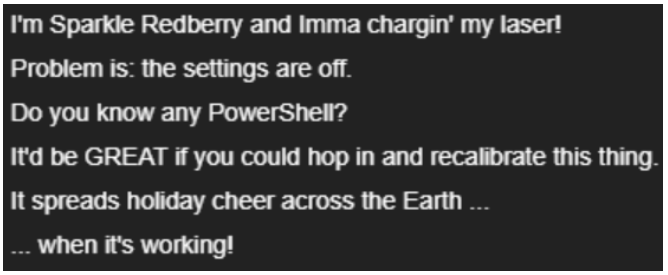
Summary

A challenge that requires further log analysis

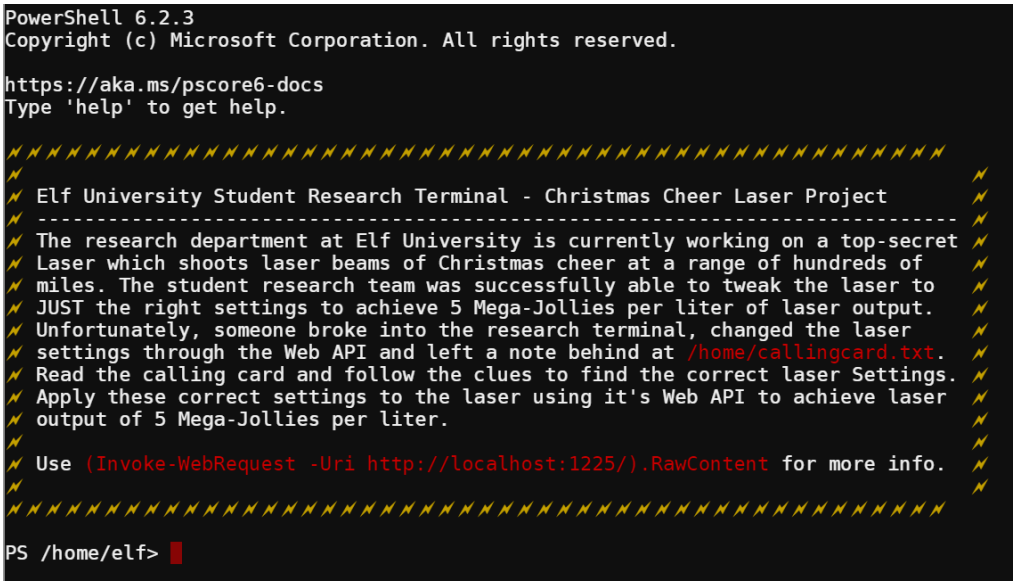
1. Complete Sparkle Redberry's secondary challenge
    a. Powershell challenges

2. Complete Main Objective
    a. Threat hunting with logs

We talk with Sparkle Redberry who informs us she is having issues with her laser that is configured in Powershell.



Let's open the terminal, and see what we receive for output.



The information on the terminal informs us that we need to recalibrate this laser and change the settings to achieve 5 Mega-Jollies Per liter of laser output. We also are informed that someone left behind a note at /home/callingcard.txt with information about the laser settings.

It is also explained to us that we can execute (Invoke-WebRequest -Uri http://localhost:1225/).RawContent to view the settings and access the web api to change laser settings.

We run that command and receive the following.

```
-------------------------------------------------
Christmas Cheer Laser Project Web API
-------------------------------------------------
Turn the laser on/off:
GET http://localhost:1225/api/on
GET http://localhost:1225/api/off

Check the current Mega-Jollies of laser output
GET http://localhost:1225/api/output

Change the lense refraction value (1.0 - 2.0):
GET http://localhost:1225/api/refraction?val=1.0

Change laser temperature in degrees Celsius:
GET http://localhost:1225/api/temperature?val=-10

Change the mirror angle value (0 - 359):
GET http://localhost:1225/api/angle?val=45.1

Change gaseous elements mixture:
POST http://localhost:1225/api/gas
POST BODY EXAMPLE (gas mixture percentages):
O=5&H=5&He=5&N=5&Ne=20&Ar=10&Xe=10&F=20&Kr=10&Rn=10
-------------------------------------------------
```

This is the settings page, displaying all possible and adjustable variables for the powershell laser. We need to find all 5; mega-jollies, lense refraction, laser temperature, mirror angle, and gaseous elements.
Now we view the calling card file mentioned above.
The calling card gives us a hint.

```
PS /home/elf> type /home/callingcard.txt
What's become of your dear laser?
Fa la la la la, la la la la
Seems you can't now seem to raise her!
Fa la la la la, la la la la
Could commands hold riddles in hist'ry?
Fa la la la la, la la la la
Nay! You'll ever suffer myst'ry!
Fa la la la la, la la la la
PS /home/elf>
```

It is suggested we check the history.
Checking the history of our shell commands gives us the next hint.

We were provided with the hint earlier to review the sans cheat sheet here

https://blogs.sans.org/pen-testing/files/2016/05/PowerShellCheatSheet_v41.pdf

We use the get- history command.

```
Id CommandLine
-- -----------
 1 Get-Help -Name Get-Process
 2 Get-Help -Name Get-*
 3 Set-ExecutionPolicy Unrestricted
 4 Get-Service | ConvertTo-HTML -Property Name, Status > C:\services.htm
 5 Get-Service | Export-CSV c:\service.csv
 6 Get-Service | Select-Object Name, Status | Export-CSV c:\service.csv
 7 (Invoke-WebRequest http://127.0.0.1:1225/api/angle?val=65.5).RawContent
 8 Get-EventLog -Log "Application"
 9 I have many name=value variables that I share to applications system wide. At a co…
10 (Invoke-WebRequest -Uri http://localhost:1225/).RawContent
11 (Invoke-WebRequest -Uri http://localhost:1225/api/output).RawContent
12 (Invoke-WebRequest -Uri http://localhost:1225/home/callingcard.txt).RawContent
13 type callingcard.txt
14 type /home/callingcard.txt
```

We notice our mirror angle at line 7 and a hint at line 9 that is cut off, so let's fix this with a formating flag.

Get-history -Id 9 |FL

```
PS /home/elf> get-history -Id 9 | FL

Id                 : 9
CommandLine        : I have many name=value variables that I share to applications
                     system wide. At a command I will reveal my secrets once you Get my
                     Child Items.
ExecutionStatus    : Completed
StartExecutionTime : 11/29/19 4:57:16 PM
EndExecutionTime   : 11/29/19 4:57:16 PM
Duration           : 00:00:00.6090308
```

Okay, the only thing I can think of is an environment variable, so let's use Powershell to dump them.

Get-childitem Env:

```
PS /home/elf> get-childitem Env:

Name                           Value
----                           -----
_                              /bin/su
DOTNET_SYSTEM_GLOBALIZATION_I… false
HOME                           /home/elf
HOSTNAME                       c17f64dd7fb7
LANG                           en_US.UTF-8
LC_ALL                         en_US.UTF-8
LOGNAME                        elf
MAIL                           /var/mail/elf
PATH                           /opt/microsoft/powershell/6:/usr/local/sbin:/usr/local/b…
PSModuleAnalysisCachePath      /var/cache/microsoft/powershell/PSModuleAnalysisCache/Mo…
PSModulePath                   /home/elf/.local/share/powershell/Modules:/usr/local/sha…
PWD                            /home/elf
RESOURCE_ID                    e4999d6c-8cff-4054-95a2-6062f566264e
riddle                         Squeezed and compressed I am hidden away. Expand me from…
SHELL                          /home/elf/elf
SHLVL                          1
TERM                           xterm
USER                           elf
userdomain                     laserterminal
USERDOMAIN                     laserterminal
username                       elf
USERNAME                       elf
```

We do see a hint here but it needs to be formatted.
So we run get-childitem Env: | FL

```
Name  : riddle
Value : Squeezed and compressed I am hidden away. Expand me from my prison and I will
        show you the way. Recurse through all /etc and Sort on my LastWriteTime to
        reveal im the newest of all.
```

Lest do some research for a query on objects by last write time.
We eventually come across the below command.

```
Get-ChildItem -Path /etc/ -Recurse | Sort-Object LastWriteTime -Descending
```

```
   Directory: /etc/apt

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-r---          1/31/20   4:48 PM        5662902 archive
```

Great we expand this archive with expand-archive /etc/apt/archive

```
   Directory: /home/elf/archive/refraction

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
------          11/7/19 11:57 AM            134 riddle
------          11/5/19  2:26 PM        5724384 runme.elf
```

We output the riddle file and receive:

```
Very shallow am I in the depths of your elf home.
identity:

25520151A320B5B0D21561F92C8F6224
```

We then change the permissions and run the runme.elf file.

```
PS /home/elf/archive/refraction> chmod +x ./runme.elf
PS /home/elf/archive/refraction> ./runme.elf
refraction?val=1.867
PS /home/elf/archive/refraction>
```

Great there is our refraction value.

Now we need to run the md5 hash that was provided to use get- against our depth directory.

```
gci -File -recurse | Get-FileHash -Algorithm md5 | where Hash -eq "25520151A320B5B0D21561F92C8F6224" | format-list
```

And we get our file location.

```
Algorithm : MD5
Hash      : 25520151A320B5B0D21561F92C8F6224
Path      : /home/elf/depths/produce/thhy5hll.txt
```

```
PS /home/elf/depths/produce> type ./thhy5hll.txt
temperature?val=-33.5

I am one of many thousand similar txt's contained within the deepest of /home/elf/depths.
 Finding me will give you the most strength but doing so will require Piping all the Full
Name's to Sort Length.
PS /home/elf/depths/produce>
```

For the final setting of **GAS** we just need to extract all files in the depths location and sort by length.

```
Get process information to include Username identification. Stop Process to show me you're skilled and in
this order they must be killed:
bushy
alabaster
minty
holly

Do this for me and then you /shall/see .
```

If you initially try to read the /shall/see file, you will get a permission denied error.

However, if we stop the process in the specified order we can then cat the /shall/see the file to receive our next clue below.

“Get the .xml children of /etc - an event log to be found. Group all .Id's and the last thing will be in the Properties of the lonely unique event Id.”

We are then required to parse the .xml files in /etc and sort them by event id.

This provides us with a .xml file containing our gas value.

We can then update our laser with the web api and we get the flag.

Main objective

The attacks don't stop! Can you help identify the IP address of the malware-infected system using these Zeek logs? *For hints on achieving this objective, please visit the Laboratory and talk with Sparkle Redberry.*

For this challenge we are asked to investigate Zeek logs to help identify the IP address of the malware-infected sysem.

The following is the hint from sparkle redberry upon completing our secondary object.

You got it - three cheers for cheer!

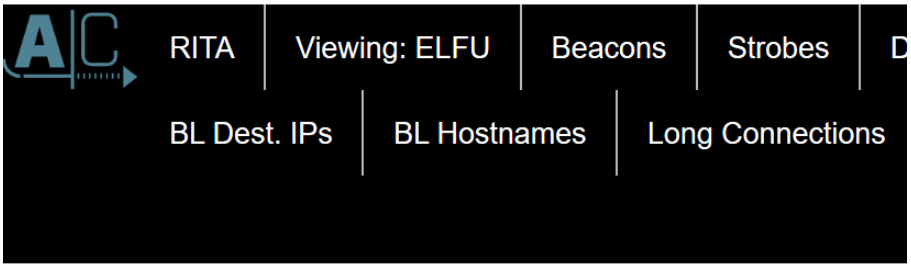
For objective 5, have you taken a look at our Zeek logs?

Something's gone wrong. But I hear someone named Rita can help us.

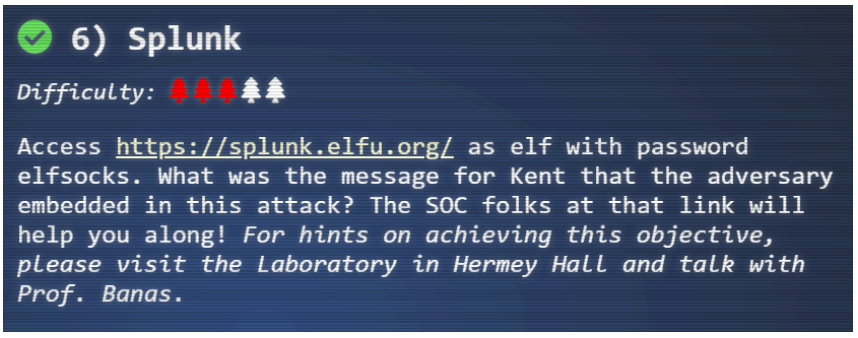
Can you and she figure out what happened?

Although the intended solution is to download, install and run Rita while passing the zeeklogs in input, I managed to just browse down into the beacons.html log file within, where I found an IP address with an enormous amount of connections. This

was indeed our target.



Score	Source	Destination	Connections
0.998	192.168.134.130	144.202.46.214	7660

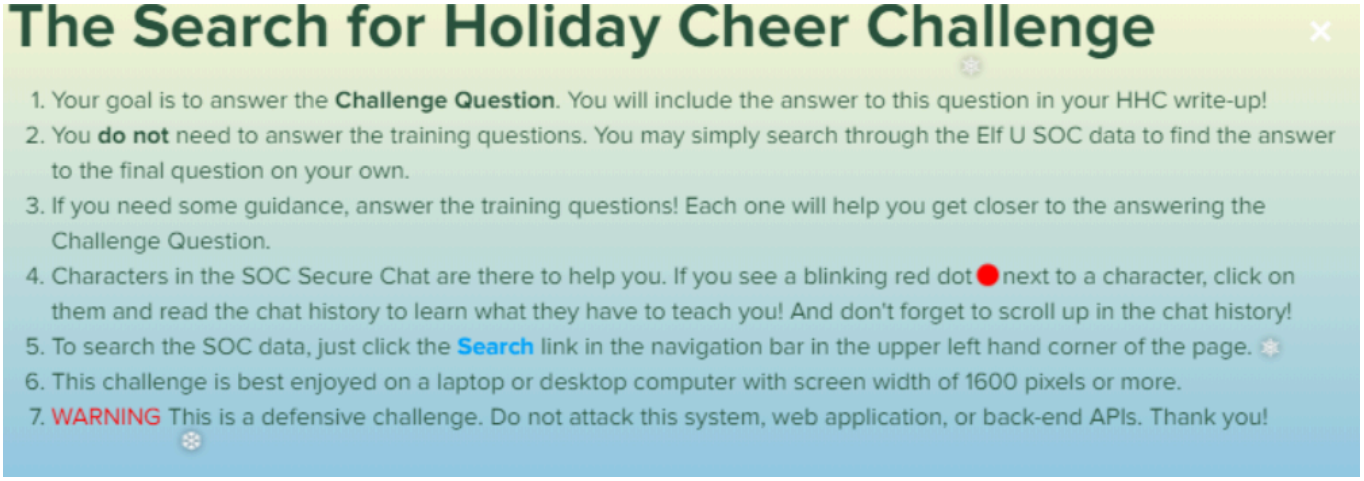


If you do not already know, Splunk is a tool for log analyses and threat hunting in short. If you have never used Splunk beforem it is a very user-friendly application.

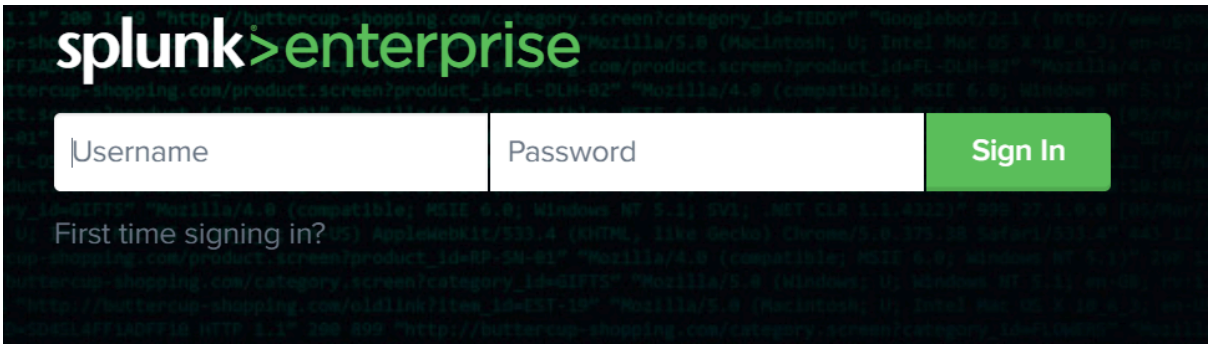
Let's visit Prof. Banas in the laboratory.

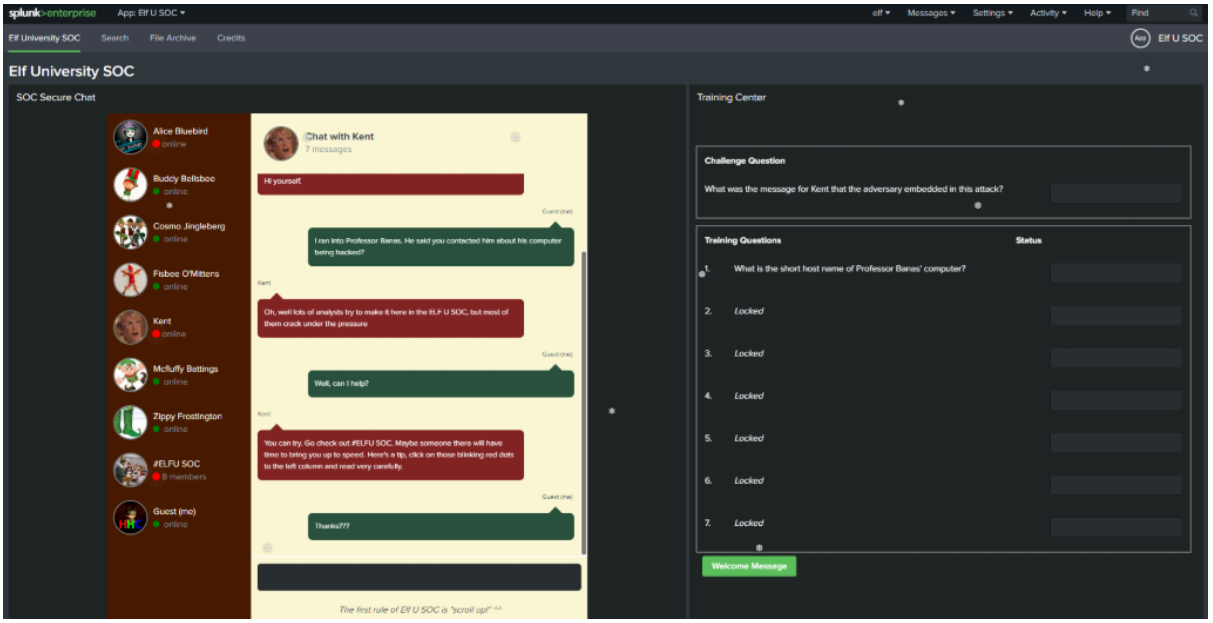
He directs us to the website mentioned above.

We are prefaced with this banner about the secondary objective of this challenge.

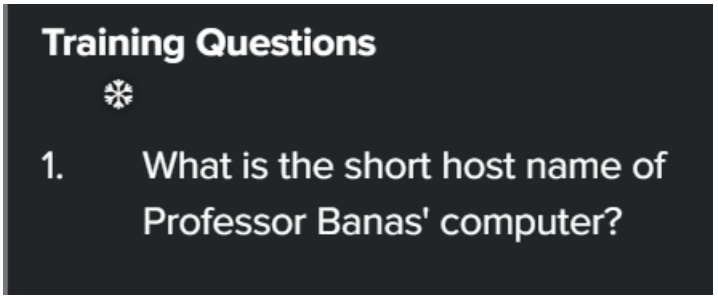


Lest go forward and connect to the Elf Soc Plunk website

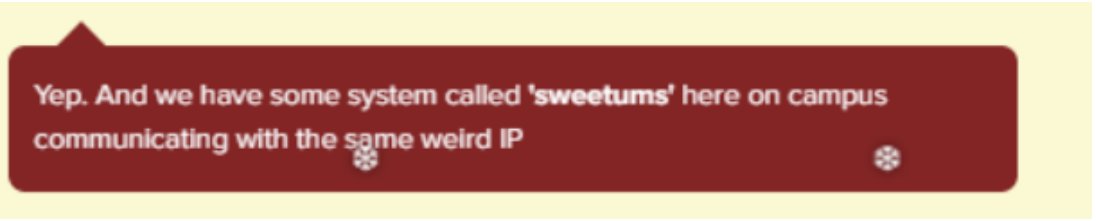


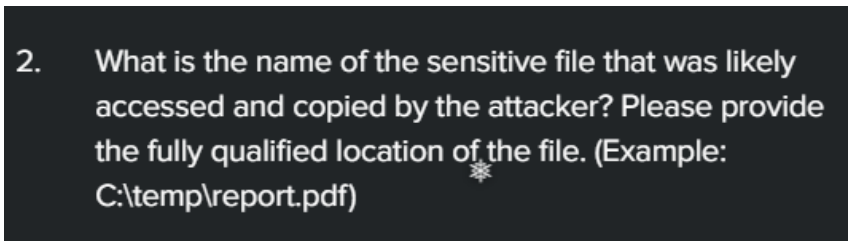


You will see here, this is really cool Elf University Soc centre, where multiple security analysts are working away and communicating in real time. Thef left side is the team communication web app and on tje right are the 7 secondary questions and the main objective question. We are going to start with the training exercise as it is intended to teach you basic skills and familiartiy with splunk and threat hunting.



Our first question is simple and is disaplyed in the chat box with our co-worker alice.





At this point, the chat with Alice on the left-hand side unfolds and you learn that these attackers are trying to get to santa by constantly trying to attack him and they may have found some of his data. So we know our target here was Santa, and thus our Splunk queries are foind