D.B Cooper Challenge

D.B Cooper Challenge

Introduction

We embarked on the D.B Cooper Challenge, starting with the following resources:

- **GitHub Account**: cooperghost230
- **Twitter Account**: cooperghost230

Initial Clues

On Twitter, we found two Base64 encoded messages:

```
Welcome to the hunt. Many have stepped onto this path, but only a few will make it to the end.
Every clue is a trap. Every step is a test. Keep your eyes sharp, your mind sharper.
You are not alone. Many teams are chasing the same answers, walking the same path.
The game has begun. Are you ready? #CooperGhost230
```

A QR code on the same Twitter account led us to a Google Drive link containing an audio file in Morse code. Decoding the Morse code revealed the message:

```
THIS IS NOT THE PATH TO HEAVEN MY CHILD YOU ARE GOING DIRECTLY TO HELL
```

GitHub Investigation

Further investigation of the GitHub commits and logs led us to a Discord server: https://discord.gg/jPCKcNpJ. We analyzed the server's profile picture and soundboard audios but found nothing significant.

Location OSINT

A teammate, Kushal, managed to OSINT the location near the Bahrain F1 circuit. The brand in question was `lululemon`. Given Lewis Hamilton's recent move to Ferrari, we focused on the "Lewis" River near Skamania County, Washington, which hinted at Lewis Hamilton and the Bahrain F1 circuit.

We searched for parks near the Bahrain circuit and found the Al Areen Wildlife Park:

[Visit Al Areen Wildlife Park]

Zip File and Hidden Git

We found the exact location, and using the coordinates as a password, we opened a zip file containing `secret_message.mp4`. Standard video steganography techniques yielded no results. However, further investigation revealed a hidden `.git` file.

```
stark@stark:~/pico/Where_am_i$ git log --oneline
313d196 (HEAD -> master) Here is a message for you all
aa81545 Hello Strangers
ae399c7 [illegible]
4b0fb6f One piece is real
217307b Let the madness begin
b9a2029 Flight got canceled
fe571d1 How is thsi flight
b32135e This time probably i have to take some flight to find him
86ef395 Oh he is lost again
caba232 Oh he is lost again
24f430b how is zoro the first one to arrive
stark@stark:~/pico/Where_am_i$ _
```

The commit message was a ROT8000 encoded link to a rickroll, which we avoided clicking.

Reverting Git

Using `git checkout`, we reverted the `.git` and found several files:

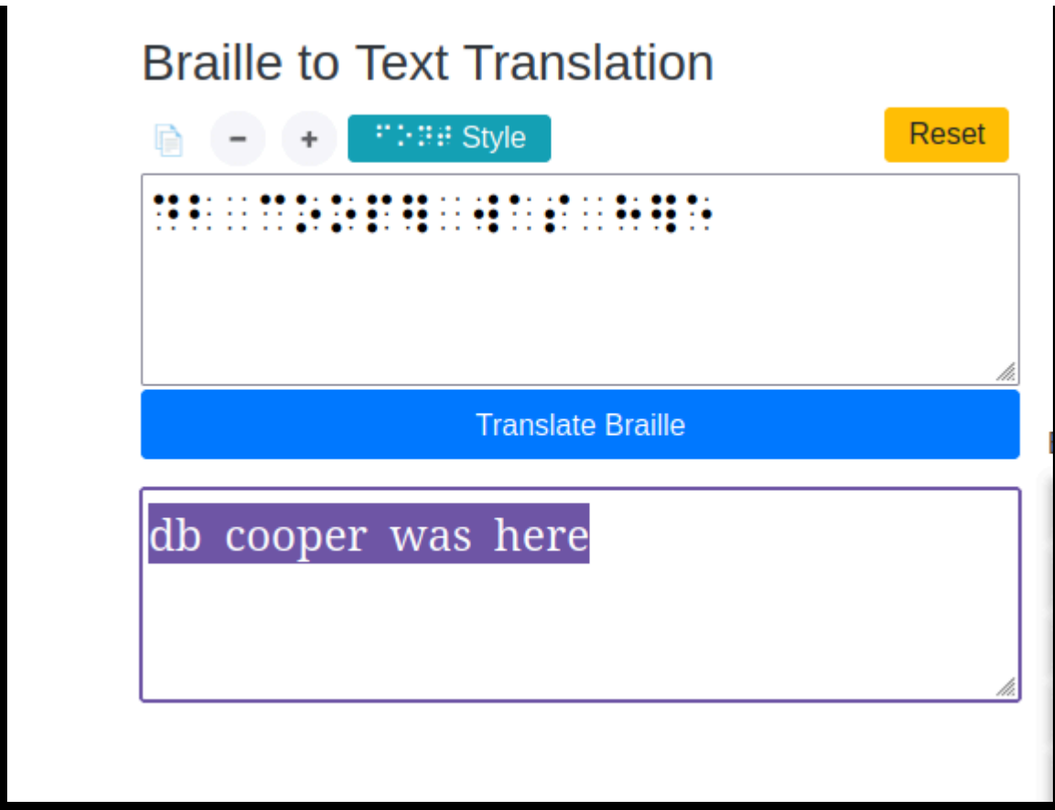
```
stark@stark:~/pico/Where_am_i$ git checkout 24f430b
Note: switching to '24f430b'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 24f430b how is zoro the first one to arrive
stark@stark:~/pico/Where_am_i$ ls
ZORO.jpeg
```

Size: 117 x 60

A file had this,

N721AF https://www.flightaware.com/live/flight/N721AF/history/20250215/1530ZZ/KGUC/KBVU

N721EF https://www.flightaware.com/resources/registration/N721EB



Of course zoro....

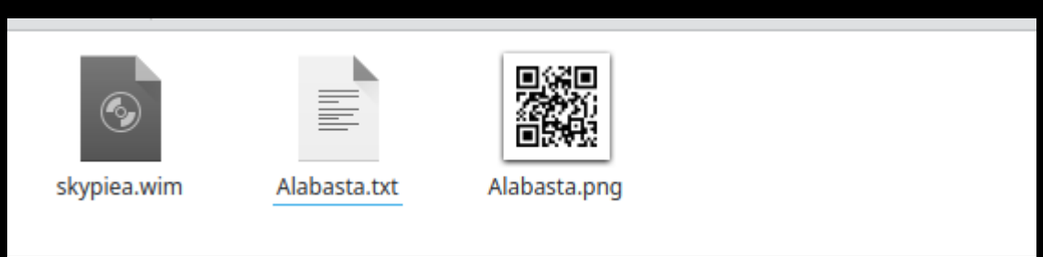
Arlong Park Brotli File

We extracted an Arlong Park brotli compressed file, which upon decompression, yielded a `.rar`, a `.jpeg`, and a `secret.txt`. The `secret.txt` contained a cipher at the bottom.

Using `steghide` we got `h3ll0w0rld`, we extracted further data:

```
stark@stark:~/pico/Arlong Park$ steghide extract -sf Arlong_Park.jpeg
Enter passphrase:
wrote extracted data to "temp.txt".
stark@stark:~/pico/Arlong Park$ cat temp.txt
stark@stark:~/pico/Arlong Park$ _
```

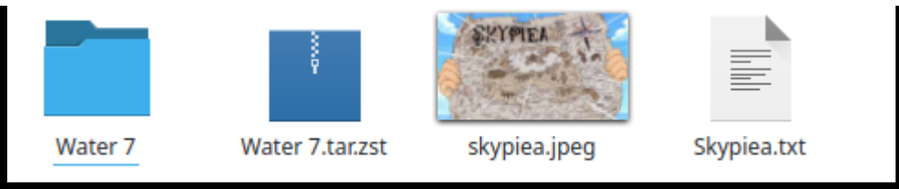
The `Alabasta.rar` extracted using `h3ll0w0rld` as pass, which revealed a PNG QR code:



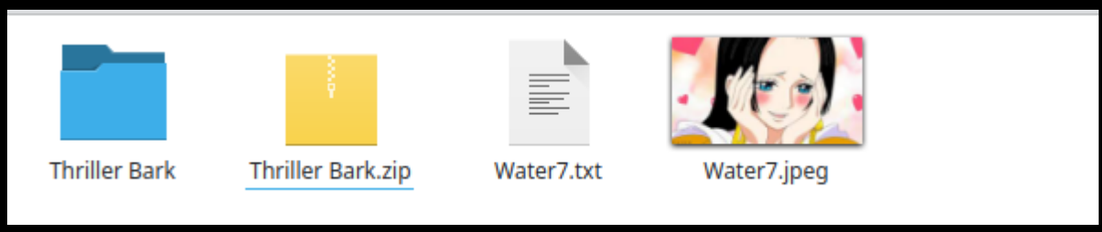
The QR code gave the coordinates `123.456, 456.789`, marking the first poneglyph.

Further Extraction

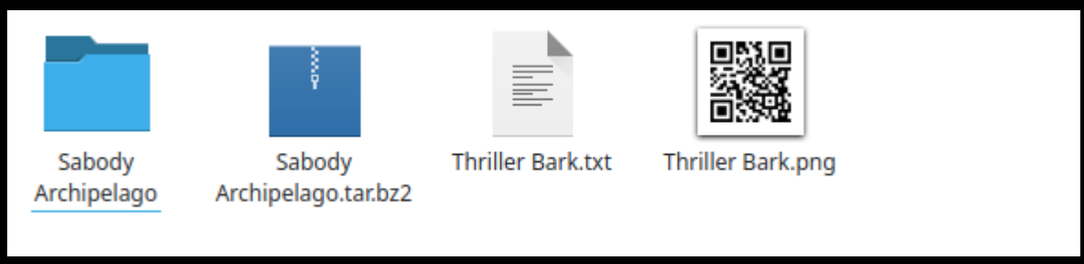
Using `wimextract`, we opened `skypiea.wim`:



The `Water7.tar.zst` file required no password:



Using `exiftool` on `Water7.jpeg`, we found the password `cutie_pie!` in the description:



This revealed another poneglyph: `789.123, 321.654`.
Then another poneglyph down the lane: `106.168 172.253`

Final Steps

We continued extracting files and found more poneglyphs:

```
123.456 456.789
789.123 321.654
789.123 321.654
106.168 172.253
```

Now for the final island,

> makeki: "GABABABABA!!"

```
"Joy Boy, huh...?!"
"We were just too early!!"
```

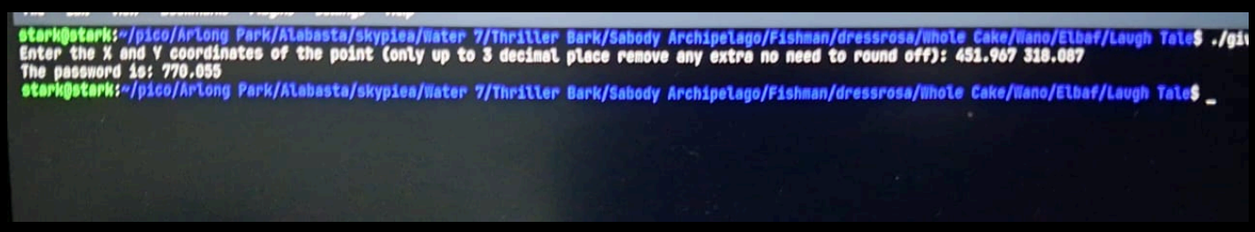
> "Ahahahaha!"
> "What a funny story!!"

"They laughed... so hard... They laughed like crazy..."

Averaging these coordinates gave us `x: 451.9675, y: 318.0875`. Entering these into the provided executable yielded the password for the `YOU WON!!.pdf`.

Conclusion

The challenge was both fun and challenging. Key learnings include the importance of validating approaches with challenge creators early on to avoid unnecessary rabbit holes. Despite some distractions, we successfully navigated through the clues to uncover the final solution.



Final Coordinates: `x: 451.9675, y: 318.0875`

Password for `YOU WON!!.pdf`: Derived from the executable using the averaged coordinates. Could have skipped finding last poneglyph by seeing the decompiled binary :cat:

```
local_18 = 451.9675;
local_14 = 318.0875;
std::operator<<((basic_ostrean
```
