1.The Beginning:

Straight forward . Flag found in given image

Flag: sctf(JOurney\_beg1ns)

2. From Blasts to Time Warps

Pretty straight forward. Found text written in reverse order in the given image

### 3. L1 Timegate

This one was a little tricky . I immediately identified the code as Morse . So I looked online for a Morse to Text converter . After that It was pretty simple

Flag : sctf{reson4anc3\_ridg3}

### 4. L2 Time Stamp

This was the most tricky and took me the most time by far . I started looking at the image and tried finding the details from the time it was uploaded to drive . However after several attempts I felt that it was useless as The vice captain was in a future timeline so entering OCT 11 2023 seemed useless . I then downloaded the image and reverse searched it . I then remembered how we used to inspect web pages to check for flags in the earlier minor competitions I had participated in .

So I opened the image using note pad . It looked like useless info at first but when I Ctrl F the word “sctf” I found the date immediately

Flag: sctf{5THJUNE2080}

### 5. L3 Wormhole

After doing the previous one this one was a piece of cake . Did the same thing again and found the key .

Flag: sctf{T3mp0r4l\_R1ft}

### 6. Airlock

This was a little confusing . What I could make out of the code was that there was a base64 code which needed to be decoded . the checkPassword method XOR-encodes the input password with the byte array representation of the key "scavenger\_\_hunt" and then base64 encodes the result.

So I had to reverse engineer the code a bit

import base64

expected = "LBAXBw1VHlQLLSw6G0EDRyI="

key = "\_scavenger\_\_hunt\_"

expected\_bytes = base64.b64decode(expected)

password\_bytes = [expected\_byte ^ key\_byte for expected\_byte, key\_byte in zip(expected\_bytes, key.encode())]

password = "".join(chr(byte) for byte in password\_bytes)

print(password)

### 7. Radiation Shield

A little bit of googling later I found what I had in front of me was a brainfuck program . Found a complier for that and got a base64 code . Converted that to ASCII to get the key

Flag: sctf{20solarflare75}

### 8. Robotic Arm

The key with a little bit of googling was TELEGONUS , the son of ODYSSESS .

Wrote a code in Python to Convert the key to a repeating key of the same length as the decimal code . Perform XOR decryption . Convert the result to a string .

import base64

# The decimal code

decimal\_code = [55, 119, 2, 117, 29, 33, 58, 44, 10, 12, 23, 38, 36, 0, 25, 126, 13, 97, 18, 48, 22, 3, 126, 37, 44, 18, 21, 33, 31, 125, 125, 119, 0, 10, 57, 106]

key = "TELEGONUS"

key = key.encode()

key = key \* (len(decimal\_code) // len(key)) + key[:len(decimal\_code) % len(key)]

decoded\_code = [decimal\_code[i] ^ key[i] for i in range(len(decimal\_code))]

decoded\_string = ''.join([chr(dec) for dec in decoded\_code])

print(decoded\_string)

We get a base64 code and we decode it .

### 9. Enter The Ship

Complied the sql file to get a list of passwords . Decoded the base64 code and got a decimal value . Converted it to ASCII and got a message saying that the 4th pass in the table was correct

Flag: sctf{p@ssw0rd\_h@ck}

### 10. Fix Coms

Converted morse code to text . Got a key = TIME . Complied the braindead code . Got a Cipher . Through trial and error I found out that it was the Vigenere Cipher . Decoded it using the KEY

Flag: sctf{TemporalConundrum}

11.The SOS

The task was bit confusing at first . I found the SOS codes in the handbook . Also found the contact for the Temporal Experts .I applied base64 decoder to the next message and got Rot13 as an answer . Apllied ROT13 to the 3rd Temporal Expert contact and got a message “Vigenere with Trek” . As per the instruction , applied all three to the SOS message and got the flag .

Flag : sctf{sos\_isnt\_save\_our\_souls}

12.

Couldn’t figure this one out . Tried removing individual planet images and look at the bin file ( as given in the hint ) but couldn’t find anything in all three