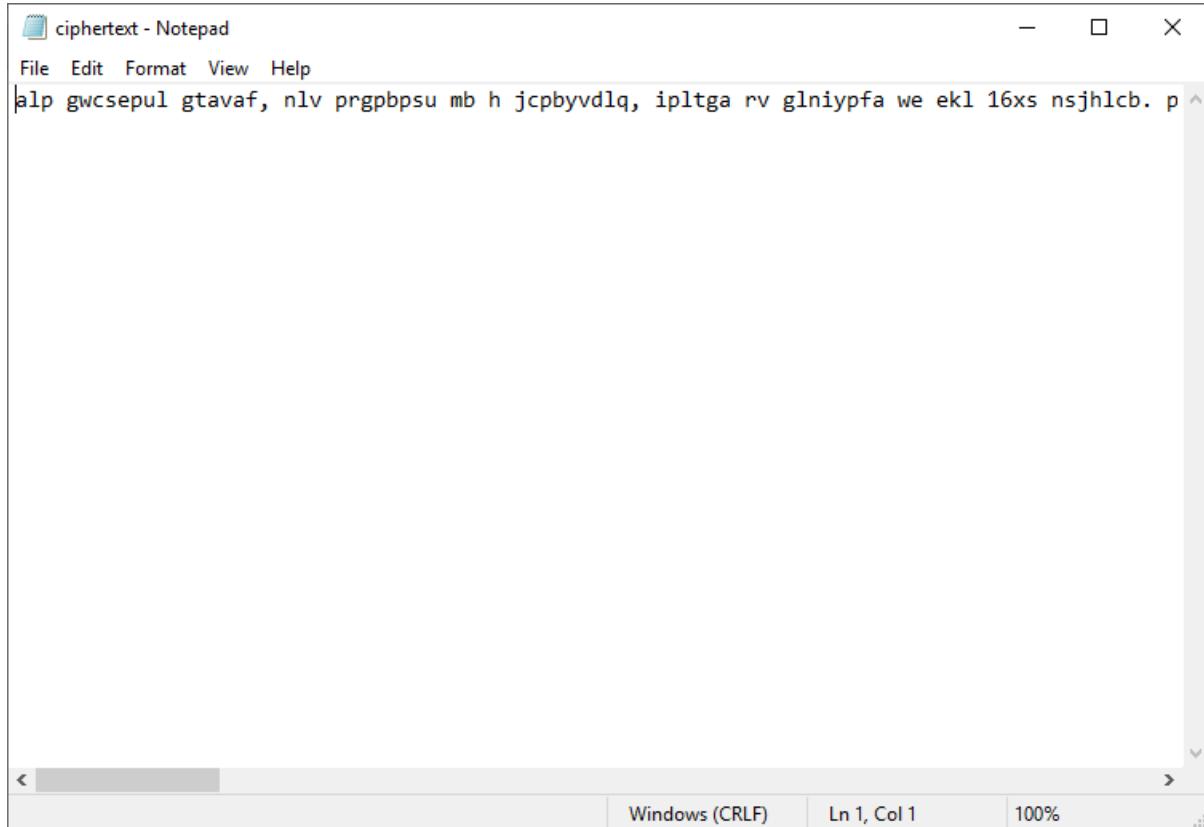


Teng Wei Xian B031610131 3BITZ

## 1.0 Classic, yet complicated!

1. Download the Zip file and open the cipertext.txt. It contains some cipertext, using Caesar cipher doesn't show the result, so I use classic vigenere.



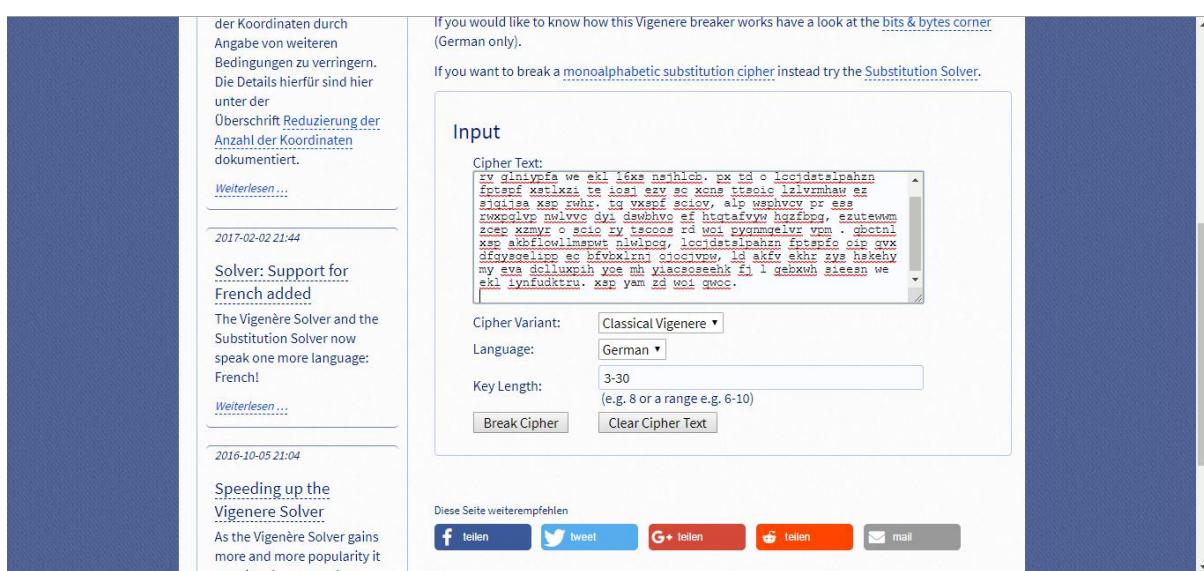
ciphertext - Notepad

File Edit Format View Help

```
alp gwcsepul gtavaf, nlv prgpbpsu mb h jcpbyvd1q, ipltga rv glniypfa we ekl 16xs nsjhlcb. p
```

Windows (CRLF) Ln 1, Col 1 100%

2. Search classic vigenere decode on Google and choose <https://www.guballa.de/vigenere-solver>. Copy the ciphertext and click break cipher.



The Vigenère Solver and the Substitution Solver now speak one more language: French!

Cipher Text:

```
alp gwcsepul gtavaf, nlv prgpbpsu mb h jcpbyvd1q, ipltga rv glniypfa we ekl 16xs nsjhlcb. p
```

Cipher Variant: Classical Vigenere ▾

Language: German ▾

Key Length: 3-30 (e.g. 8 or a range e.g. 6-10)

Break Cipher Clear Cipher Text

3. Then the result will be shown. Answer is **HTB{helloworld}**.

## Result

---

[Clear text](#) [\[hide\]](#)

Clear text using key "helloworld":

```
the vigenere cipher, was invented by a frenchman, blaise de  
vigenere in the 16th century. it is a polyalphabetic cipher  
because it uses two or more cipher alphabets to encrypt the data.  
in other words, the letters in the vigenere cipher are shifted by  
different amounts, normally done using a word or phrase as the  
encryption key . unlike the monoalphabetic ciphers, polyalphabetic  
ciphers are not susceptible to frequency analysis, as more than  
one letter in the plaintext can be represented by a single letter  
in the encryption. the key is the flag.
```

---

[Details](#) [\[show\]](#)

---

[Key length statistics](#) [\[show\]](#)

---

[Histogram](#) [\[show\]](#)

---

*Runtime: 0.010 seconds*

## 2.0 Deceitful Batman

1. From the cipher text we could guess is baconian cipher because it contains two repeated character.



2. Using online tools <https://www.dcode.fr/bacon-cipher>, we enter the code. We can see the there is THEFLAGISNAPIER. So the flag will be **HTB{NAPIER}**.

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:  
e.g. type scrabble

Results

A=A, B=N (αβ1) THEFLAGISNAPIER  
A=A, B=N (αβ2) SHEFKAGIRMAOIEQ  
A=N, B=A (αβ2) NY??Y?ZXOTT?RX?P  
A=N, B=A (αβ1) O???X??ZPU?S2?Q

famous

Cryptography • Substitution Cipher • Bacon Cipher

Sponsored ads

Download Chrome Browser

Install Offline, Device-based Group Policies & More. Deploy Chrome MSI

Google

Baconian Cipher Decoder

★ BACON CIPHERTEXT  
NAANAAAANNAANAAAANANAAAANAAAANAAAANAAAANAAAANNA  
ANAAAAAAANAAA

See also: Uppercase Lowercase Writing

Bacon Encoder

★ BACON PLAIN TEXT  
dCode Bacon

LETTER 1 A  
LETTER 2 B

ENCRYPT

See also:

Using dCode - you accept cookies for statistic and advertising purposes. OK ...

Feedback

Support

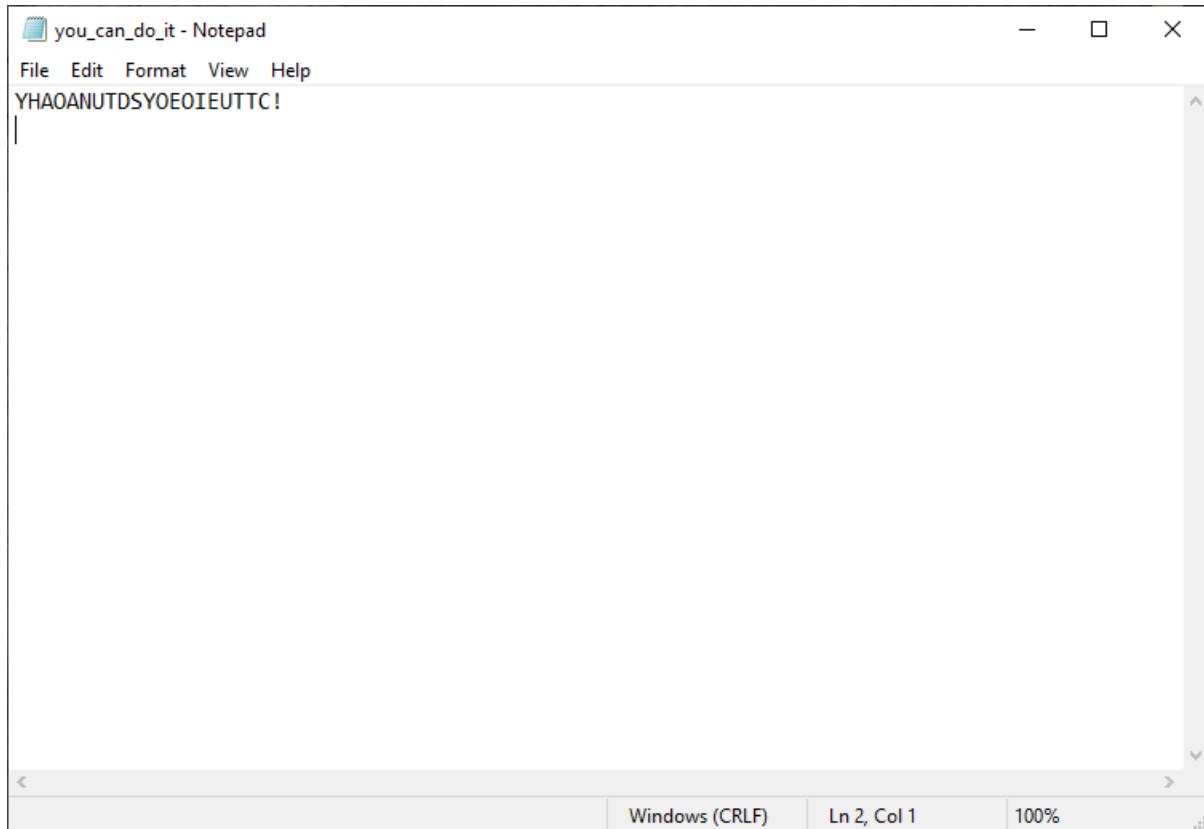
Similar tools

- ★ Baconian Cipher Decoder
- ★ Bacon Encoder
- ★ How to encrypt using Bacon cipher
- ★ How to decrypt Bacon cipher
- ★ How to recognize a Bacon ciphertext?
- ★ What are the variants of the Bacon cipher?
- ★ When Bacon cipher have been invented?

- ★ Uppercase Lowercase Writing
- ★ Caesar Cipher
- ★ Letter Number (A1Z26) A=1, B=2, C=3
- ★ Morse Code
- ★ Alphabetical Substitution
- ★ Binary Code
- ★ ROT Cipher
- ★ Polybius Cipher
- ★ Enigma Machine
- ★ Shift Cipher
- ★ All Tools ★

### 3.0 You Can Do It!

1. From the ciphertext below, we can conclude that it actually quite related to the title, so I think it just a scrambled of letters in sentence.



2. By using Caesar Box Cipher, we can rearrange the letter. Hence the answer will be **HTB{YOUSEETHATYOUCANDOIT!}**

Search for a tool  
★ SEARCH A TOOL ON dCODE BY KEYWORDS:  
e.g. type sudoku GO

Results

(7) YOUSEETHATYOUCANDOIT.  
(8) YOUSEET·HATYOU·ANDOIT·  
(9) YOUSEET·HATYOU·ANDOIT···  
(10) YAAUDYEIUTHONTSOEETC  
(4) YNYEHUOUATEODOTASIC  
(5) YADEHNOSOTAUYTOTOC  
(3) YTIIHDEASUOYTAOTNECUO·  
(2) YYHDAEOAOAINEUUTTDTSC  
(6) YADEU·HNSOT·AUYIT·OTOECD

SPONSORED SEARCHES

block cipher decoder

box cipher

CAESAR BOX CIPHER

Cryptography · Transposition Cipher · Caesar Box Cipher

Sponsored ads

SPONSORED SEARCHES

block cipher decoder

box cipher

Caesar Box Decoder

★ CAESAR BOX CIPHERTEXT  
YHAOANUTDSYOEOTIEUTTC !

KEEP PUNCTUATION AND SPACES

SIZE (WIDTH) OF THE BOX

TRY ALL POSSIBLE SIZES (BRUTE-FORCE ATTACK)

DECRYPT CAESAR BOX

See also: Scytale Cipher — Caesar Cipher — Transposition Cipher

Caesar Box Encoder

★ CAESAR BOX PLAIN TEXT  
dCode Caesar Box

Summary

Version Française

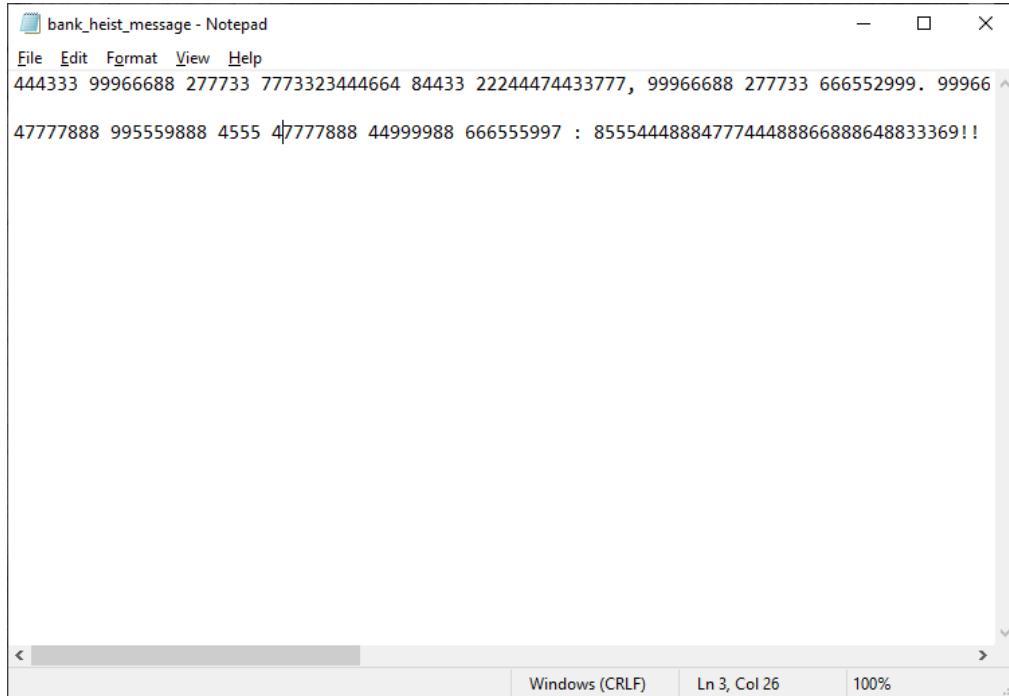
Feedback

Similar tools

- ★ Caesar Cipher
- ★ Transposition Cipher
- ★ Scytale Cipher
- ★ Rail Fence (Zip-Zag) Cipher
- ★ ADFGVX Cipher
- ★ Columnar Transposition Cipher
- ★ Double Transposition Cipher

## 4.0 Bank Heist

1. As from the description show that it is using a phone. So, we can conclude that using a multi-tap cipher.



bank\_heist\_message - Notepad  
File Edit Format View Help  
444333 99966688 277733 7773323444664 84433 22244474433777, 99966688 277733 666552999. 99966  
47777888 995559888 4555 47777888 44999988 666555997 : 8555444888477744488866888648833369!!

2. Using online tools multi-tap cipher sms mode, the message decoded are [IIGF YOU ARE READING THE CIPHER YOU ARE OKAY YOUR SHARE OF THE HEIST IS IN YOUR HOUSE THE KEY TO THE LOCK IS BELOW GO TO PARIS]



Search for a tool  
★ SEARCH A TOOL ON DCODE BY KEYWORDS:  
e.g. type caesar GO

Results  
IIGF YOU ARE READING THE CIPHER YOU ARE OKAY  
YOUR SHARE OF THE HEIST IS IN YOUR HOUSE THE  
KEY TO THE LOCK IS BELOW GO TO PARIS

SPONSORED SEARCHES  
a mobile phones best deal mobile phone online

MULTI-TAP CIPHER (SMS MODE ABC)  
Communication System > Telecom > Multi-tap Cipher (SMS Mode ABC)

Sponsored ads  
SPONSORED SEARCHES  
a mobile phones best deal mobile phone online

Multi-tap Decoder/Translator  
★ MULTI-TAP MOBILE PHONE CIPHERTEXT  
444333 99966688 277733 7773323444664 84433 22244474433777,  
99966688 277733 666552999. 99966688777 777744277733 666333  
84433 44334447778 444777 44466 99966688777 446668877733.  
84433 5533999 8666 84433 5556662255 444777 22335556669.  
4666 8666 727774447777.

★ DICTIONARY dCode ENGLISH Dictionary (common words)  
★ BRUTEFORCE ALL POSSIBILITIES

DECRYPT MULTI-TAP

See also: T9 Cipher (SMS)  
T9 vs Multitap  
Multitap should not be confused with T9 predictive text. DCODE is written 3222666333 in Multitap and 32633 in T9.

Go to: T9 Cipher (SMS)

3. The second line of ciphertext will be using FRENCH Dictionary to decode. And it gives out [GSV XLWV GL GSV HZU OLXP TLIVGRIVNVMGUFMW].

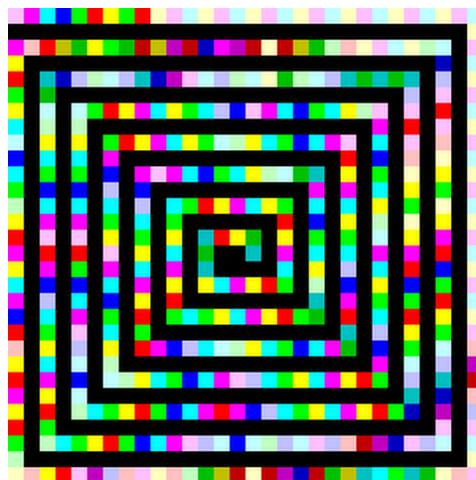
The screenshot shows the dCode search interface. In the search bar, 'Multi-tap Cipher (SMS Mode ABC)' is entered. Below the search bar, several sponsored search results are listed, including 'a mobile phones' and 'best deal mobile phone online'. On the right side of the page, there is a detailed section for 'Multi-tap Decoder/Translator' showing a ciphertext input field containing '47777888 995559888 4555 47777888 44999988 666555997 : 85554448884774488866888648833369 !'. Below this, there are dropdown menus for 'DICTIONARY' set to 'dCode FRENCH Dictionary (full - all words)' and 'BRUTEFORCE ALL POSSIBILITIES'. A large yellow button labeled 'DECRYPT MULTI-TAP' is prominently displayed.

4. Take the cipher and user Atbash Mirror Decoder. As we can see the flag will be **HTB{GORETIREMENTFUND!!}**

The screenshot shows the dCode search interface again. This time, the search term is 'Atbash Mirror Cipher'. The results page features a large banner for a 'RM 45 REBATE' with a 'PROMO CODE BONIA45TH'. Below the banner, the ciphertext 'THE CODE TO THE SAF LOCK GORETIREMENTFUND' is shown. To the right, there is a detailed section for 'Atbash Decoder' with a ciphertext input field containing 'GSV XLWV GL GSV HZU OLXP TLIVGRIVNVMGUFMW'. There are dropdown menus for 'ALPHABET' (set to 'ABCDEFGHIJKLMNOPQRSTUVWXYZ') and 'USE HEBRAIC ALPHABET' (unchecked). A yellow button labeled 'DECRYPT ATBASH' is present. Below this, there is a note about other substitution ciphers: 'See also: Alphabetical Substitution – Caesar Cipher – ROT-13 Cipher'.

## 5.0 Art

1. Download the image, is look like a portable pixmap. Use npiet online to decode.

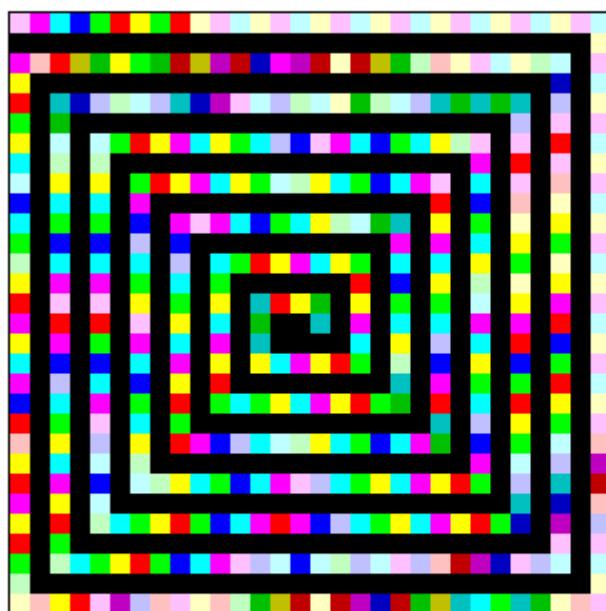


2. Therefore the flag is **HTB{p137\_m0ndr14n}**

Info: upload status: Ok

Info: found picture width=300 height=300 and code1 size=10

Uploaded picture (shown with a small border): **art.png**



Info: executing: npiet -e 1000000 art.png

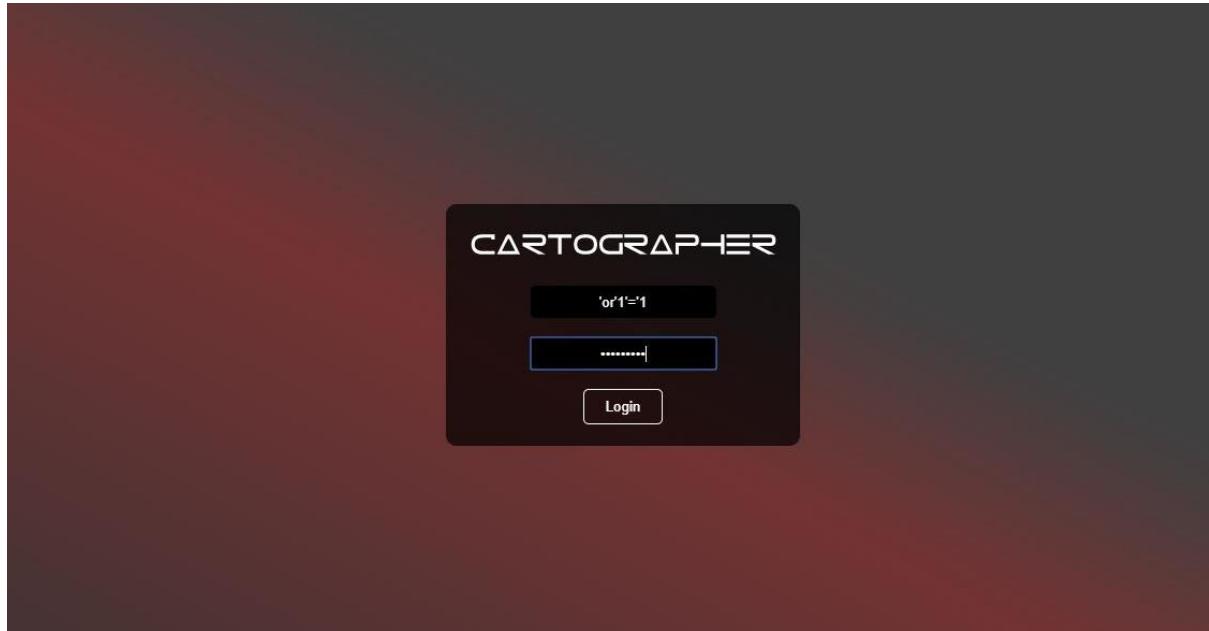
---

HTB{p137\_m0ndr14n}? ? \$□? ? □□18? 32464? ? ? 8? ? ?

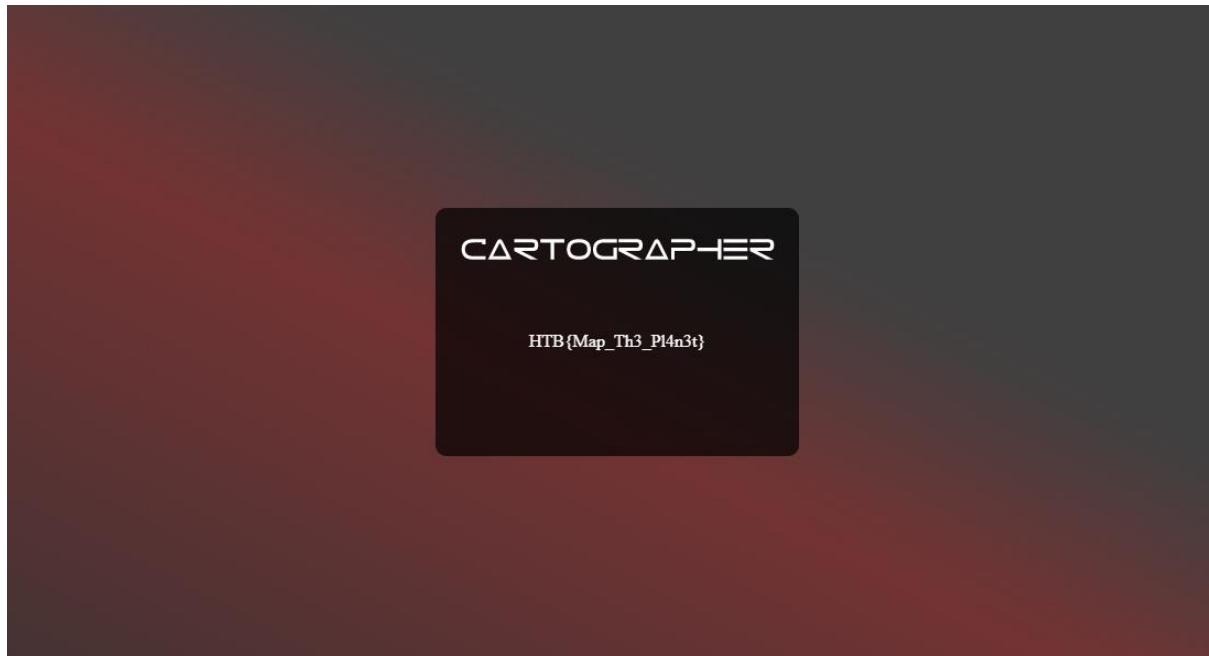
---

## 6.0 Cartographer

1. Login using 'or'1'='1 by SQL injection.



2. At the URL inject info=flag. Flag=HTB{Map\_Th3\_Pl4n3t}.



## 7.0 HDC

1. First inspect the code.

**MDS encrypt this string**

E2G7wuY0IfdfnWukv2u9

Too slow!

MD5  Submit

Styles Computed Event Listeners >

```
<html>
  <head></head>
  <body style="background-color:powderblue;">
    <h1 align="center">MDS encrypt this string</h1>
    <h3 align="center">E2G7wuY0IfdfnWukv2u9</h3>
    <p align="center">Too slow!</p>
    <center>
      <form action="" method="post"> == $0
        <input type="text" name="hash" placeholder="MDS" align="center">
        <br>
        <input type="submit" value="Submit">
      </form>
    </center>
  </body>
</html>
```

Filter :hov .cls + ^

Inherited from center

center { user agent stylesheet

text-align: -webkit-center;

Inherited from html

html { user agent stylesheet

color: -internal-root-color;

Highlights from the Chrome 74 update

Highlight all nodes affected by CSS property

Hover over a CSS property like padding or margin in the Styles pane to highlight all nodes affected by that declaration.

Lighthouse v4 in the Audits panel

Featuring a new "tap targets" audit for checking that mobile links and buttons are properly sized, and a new UI for PWA reports.

WebSocket binary message viewer

2. Write a python code and run so that can extract the flag. The flag is **HTB{N1c3\_ScrIptInG\_B0i!}.**

Online Python compiler, Online Python IDE, and... Code Python, compile Python, run Python, and host your programs an...

run ▶ share ↗ + new repl languages Sign up

Files main.py saved

```
main.py
1 import requests
2 import hashlib
3 import re
4
5
6
7 url="http://docker.hackthebox.eu:34044/"
8
9 r=requests.session()
10 out=r.get(url)
11 out=re.search('<h3 align="center'>.*?</h3>',out.text)
12 out=re.search('<^|>|<|.....',out[0])
13 out=re.search("[^|>|<].....",out[0])
14
15 out=hashlib.md5(out[0].encode('utf-8')).hexdigest()
16
17 print("sending md5 :-{}".format(out))
18
19 data={'hash': out}
20 out = r.post(url = url, data = data)
21
22 print(out.text)
```

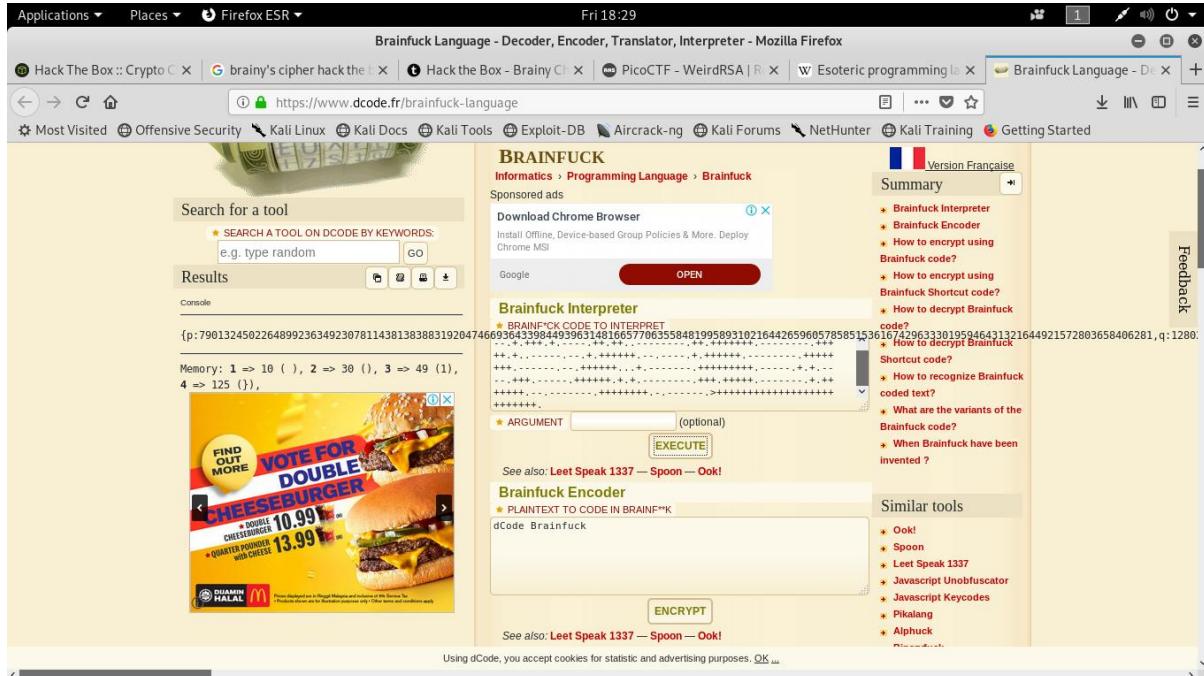
Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux

```
sending md5 :-0e29676fc392f9041da05dcc91df5f5
<html>
  <head>
    <title>embed five for life</title>
  </head>
  <body style="background-color:powderblue;">
    <h1 align="center">MDS encrypt this string</h1><h3 align="center">HTB{N1c3_ScrIptInG_B0i!}</h3><p align="center">Too slow!</p>
    <center>
      <form action="" method="post">
        <input type="text" name="hash" placeholder="MDS" align="center">
        <br>
        <input type="submit" value="Submit">
      </form>
    </center>
  </body>
</html>
```

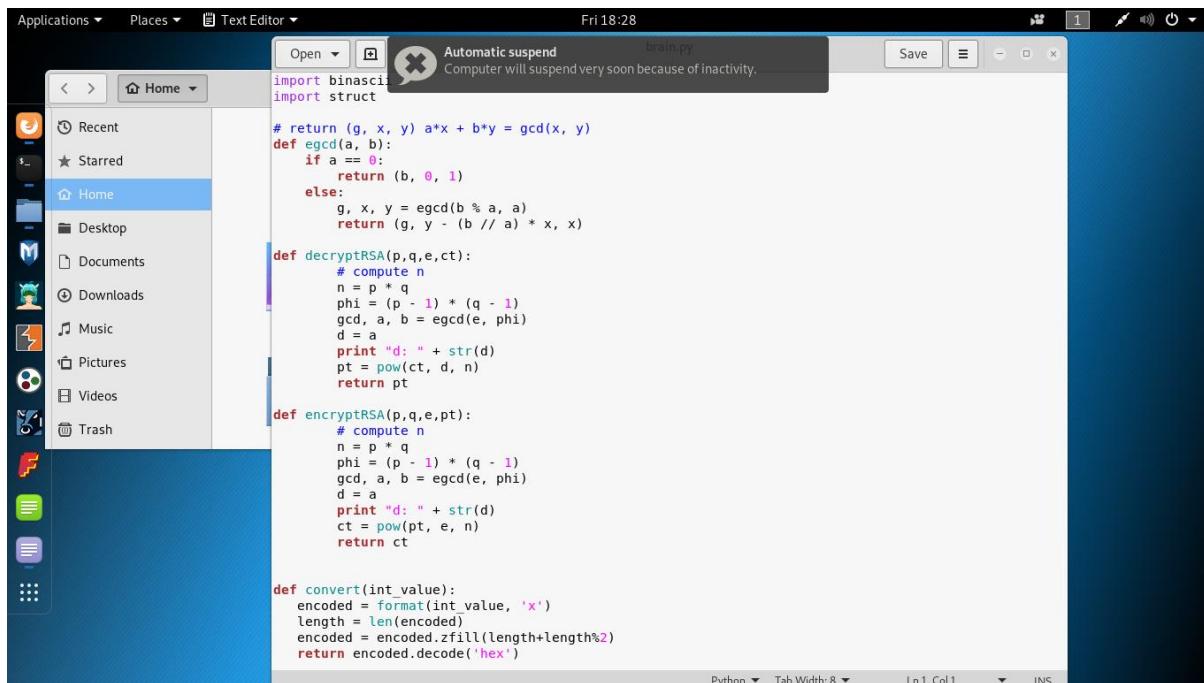
KeyboardInterrupt

## 8.0 Brainy's Cipher

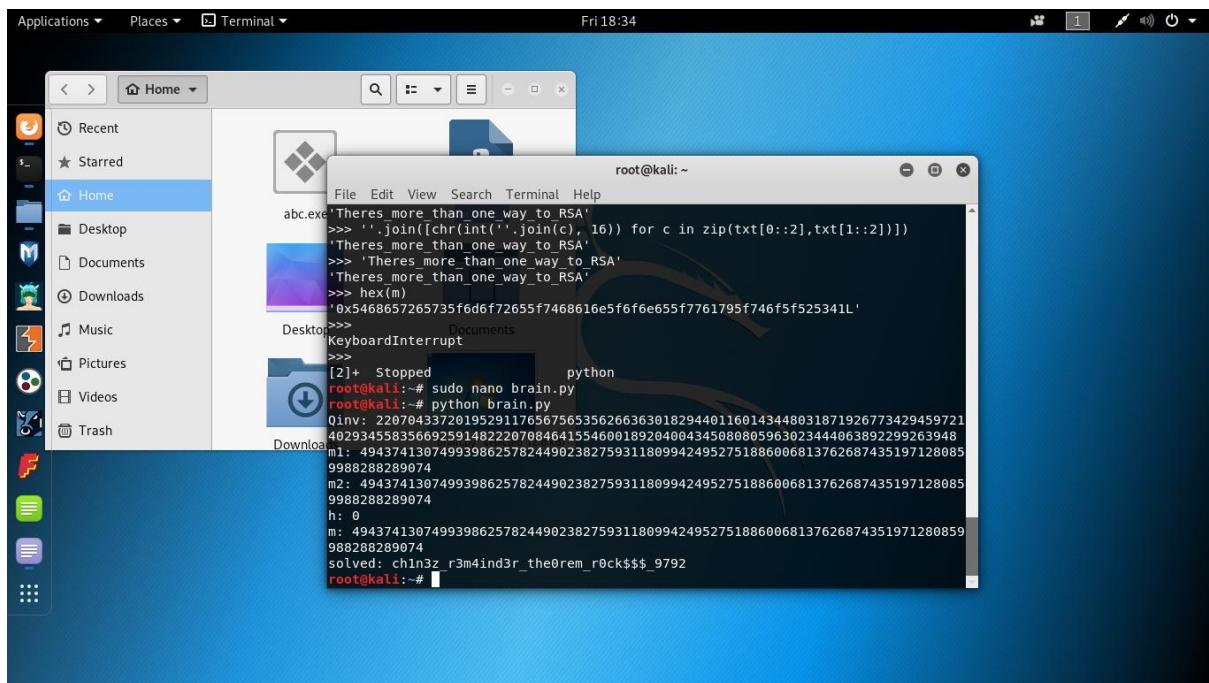
1. Using BrainFuck algorithms to decode the ciphertext.



2. Using the coding below to decode.



3. Then we get the flag. **HTB{ch1n3z\_r3m4ind3r\_th30rem\_r0ck\$\$\$\_9792}**.



## 9.0 Keys

1. From the ciphertext, we could guess it is a Fernet algorithms.

Enter the token and key then pop: **HTB{N0t\_A\_Flg!}**

Fri 18:42

Fernet (Decode) - Mozilla Firefox

Applications ▾ Places ▾ Firefox ESR ▾

Hack The Box :: Crypto C × Hack The Box - Keys - Cr × Fernet (Decode) × +

https://asecuritysite.com/encryption/ferdecode

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

[ Log On ]

bill's A security site.com + profsims.com - Networksims

HOME TST CHA ENC CODE IP FUN SUB DIGF CIS COM DB ABOUT NETSIM

**Fernet (Decode)**

[Back] Fernet is a symmetric encryption method which makes sure that the message encrypted cannot be manipulated/read without the key. It uses URL safe encoding for the keys. Fernet uses 128-bit AES in CBC mode and PKCS7 padding, with HMAC using SHA256 for authentication. The IV is created from os.random(). This page decodes the token. Generate a token here: [Fernet]

Token: gAAAAABaDDCRPXCPdGdcBKfqEFz9zvnaiLUbWhqXqScTTYwfZJcz-WhH7rf\_fYHo67GzJAdkrwATuMptY-nJmU-eYG3HKL09WDLm027sex1-R85CZEFcU=

Key: hBU9lesroX\_veFoHz-xUcaz4\_ymH-D8p28IP\_4rtjq0=

Determine

Decoded: Flag : HTB{N0t\_A\_Flg!}

Date created: Wed Nov 15 12:18:25 2017

Current time: Fri Jun 14 11:40:43 2019

=====Analysis=====

## 10.0 Da Vinci

1. Download the file, It has 3 image. First we take the picture with passphrase which is TOM.

### Steganographic Decoder

This form decodes the payload that was hidden in a JPEG image or a WAV or AU audio file using the [encoder form](#). When you submit, you will be asked to save the resulting payload file to disk. This form may also help you guess at what the payload is and its file type...

Select a JPEG, WAV, or AU file to decode:

Choose File | monalisa.jpg

Password (may be blank):

TOM

- View raw output as MIME-type `text/plain`  
 Guess the payload  
 Prompt to save (you must guess the file type yourself.)

Submit

To use this form, you must first [encode a file](#).

These pages use the [steghide](#) program to perform steganography, and the files generated are fully compatible with steghide.

Please send comments or questions to [Alan Eliasen](#)

[Back to Alan's Home Server](#)

2. It give the key with md5 encoded.

Hey Filippos,  
This is my secret key for our folder.... (key:020e60c6a84db8c5d4c2d56a4e4fe082)  
I used an encryption with 32 characters. hehehehehe! No one will find it! ;)  
Decrypt it... It's easy for you right?  
Don't share it with anyone...plz!

if you are reading that, call me!  
I need your advice for my new CTF challenge!

Kisses,  
-Luc1f3r

3. After decoded, it show the password is lenardo.

## MD5 Decryption

Enter your MD5 hash below and cross your fingers :

Decrypt

Found : **leonardo**

(hash = 020e60c6a84db8c5d4c2d56a4e4fe082)

## How it works?

MD5 is a 128-bit encryption algorithm, which generates a hexadecimal hash of 32 characters, regardless of the input word size.

This algorithm is not reversible, it's normally impossible to find the original word from the MD5.

Our tool uses a huge database in order to have the best chance of cracking the original word.

Just enter the hash in the MD5 decoder in the form above to try to decrypt it!

Words in the database: 1,154,869,660,066



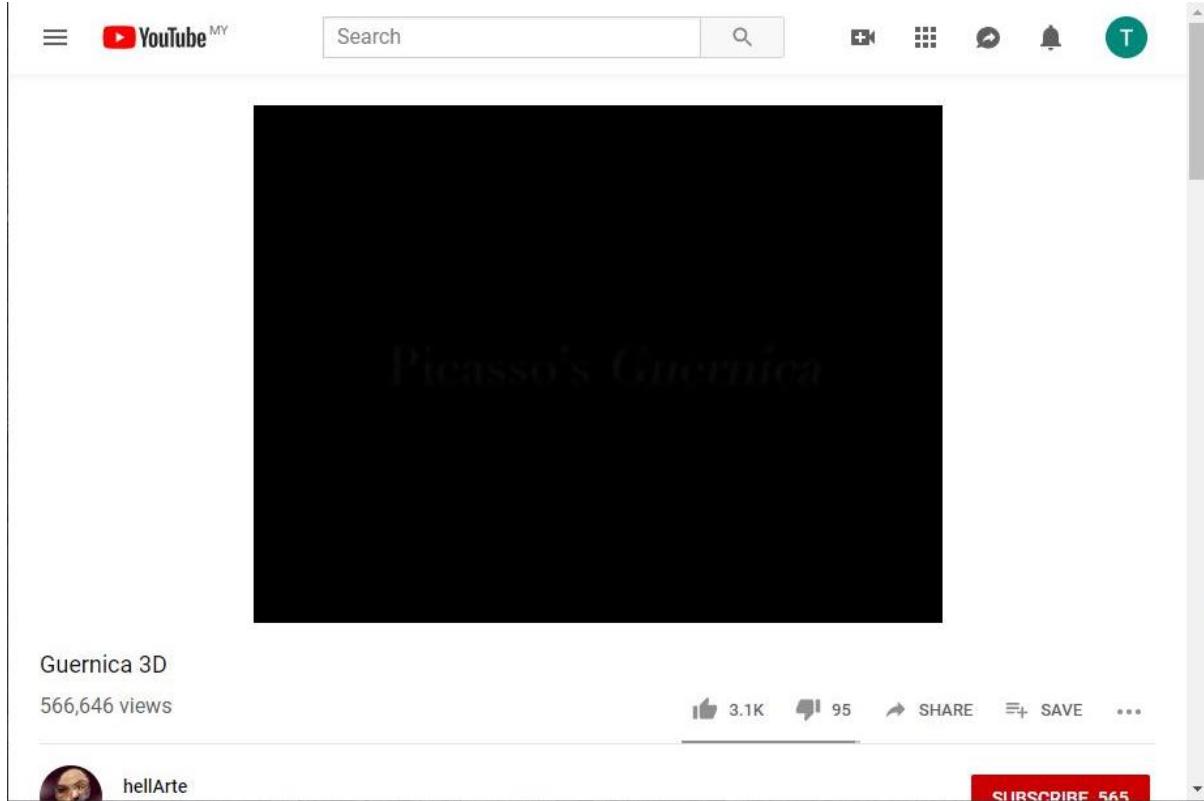
4. For the Plan image, drop to HxD and there is a youtube link.

The screenshot shows the HxD hex editor interface with the file 'Plans.jpg' open. The left pane displays the hex dump of the file, and the right pane shows the corresponding ASCII and decoded text. A specific line of text is highlighted in blue, revealing a YouTube URL: 'AyÙhttps://www.youtTube.com/watch?v=jclNfx4c5LQ.' This indicates that the image file contains a reference to a video on YouTube.

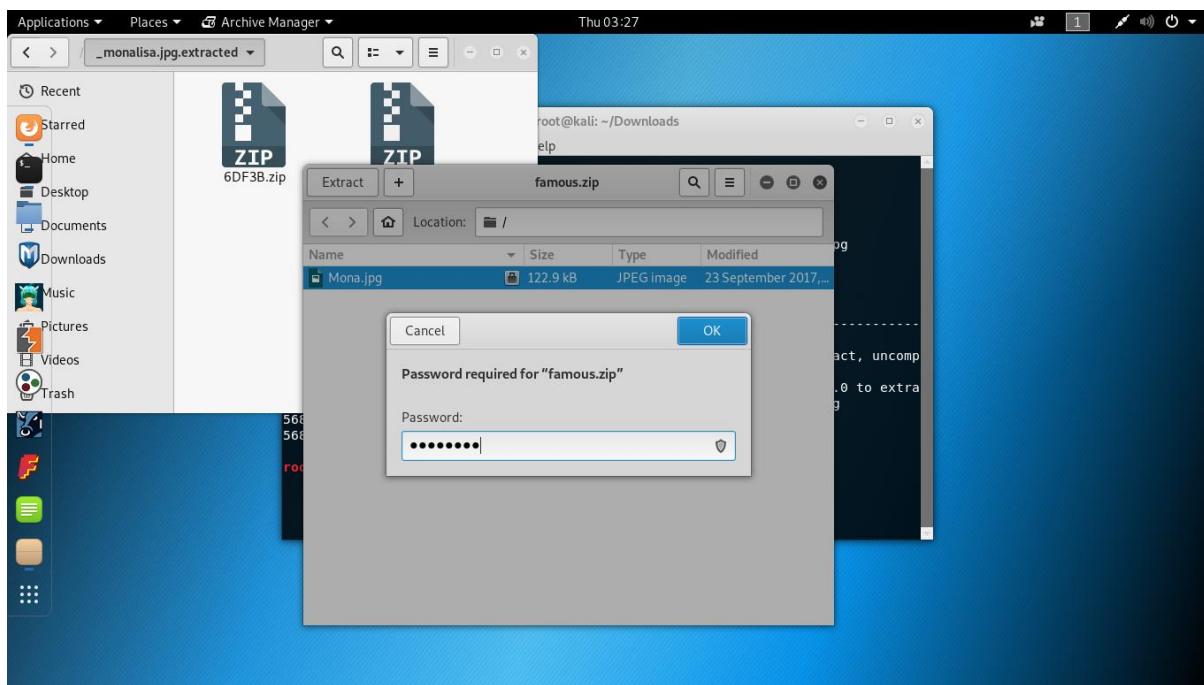
Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00015260	4C 67 12 84 6C A1 42 B3 34 B2 C1 61 27 5E 97 BD	Ig.,,1;B'4fÁa'^_Á
00015270	55 8C B0 45 52 DB 98 0B F8 D7 57 50 65 A4 95 10	UG°ERÚ.,ø×Wþe×•.
00015280	75 63 AD FA 57 2C 05 F5 FD 71 07 FC 6B 5D 5D 40	uc.úW,.öýq.úk]}@
00015290	B7 12 C6 3A 88 06 81 94 47 89 91 73 31 E3 5D 5D	.E:..,"Gh'slájj]
000152A0	41 FF D9 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79	AyÙhttps://www.y
000152B0	6F 75 74 75 62 65 2E 63 6F 6D 2F 77 61 74 63 68	utTube.com/watch
000152C0	3F 76 3D 6A 63 31 4E 66 78 34 63 35 4C 51 0B	?v=jclNfx4c5LQ.

Offset(h): 152A3 | Block(h): 152A3-152CE | Length(h): 2C | Overwrite

5. At the youtube link, Guernica is another key.



6. After steghide the monalisa, it has 2 zip files in it. Extract the famous.zip using leonardo. Output is Mona.jpg.



7. Extract the Mona.jpg with steghide again. Then it will give u a base64 encoded keys.

```
root@kali: ~/Downloads
File Edit View Search Terminal Help
Preparing to unpack .../libmcrypt4_2.5.8-3.4_amd64.deb ...
unpacking libmcrypt4 (2.5.8-3.4) ...
Selecting previously unselected package libmhash2:amd64.
Unpacking libmhash2:amd64 (0.9.9.9-7+b1) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-13_amd64.deb ...
Unpacking steghide (0.5.1-13) ...
Setting up libmhash2:amd64 (0.9.9.9-7+b1) ...
Processing triggers for libc-bin (2.28-2) ...
Processing triggers for man-db (2.8.5-1) ...
Setting up libmcrypt4 (2.5.8-3.4) ...
Setting up steghide (0.5.1-13) ...
Processing triggers for libc-bin (2.28-2) ...
root@kali:~/Downloads# steghide extract -sf Mona.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
root@kali:~/Downloads# steghide extract -sf Mona.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
root@kali:~/Downloads# steghide extract -sf Mona.jpg
Enter passphrase:
wrote extracted data to "key".
root@kali:~/Downloads#
```

8. Take the key and decoded 3 times. **HTB{M0n@\_L1z@\_!sD3@D}**

The screenshot shows a Firefox browser window with the address bar set to <https://www.base64decode.org>. The main content area displays a form titled "Decode from Base64 format". The input field contains the base64 encoded string: SFRCe00wbkBTDf6QF8hc19EM0BEfQ==. Below the input field, there are two dropdown menus: "ASCII" and "Source charset". There is also a checkbox labeled "Live mode OFF" which is checked. A green "DECODE" button is visible. To the right of the form, there is an advertisement for PhpStorm. The background of the browser window features a green pattern of various icons related to software development and technology.

## 11.0 Unified

1. The code is using Unicode Text Steganography Encoders/Decoders. So decode the unknown character to get flag **HTB{tr1th3m1u5\_1499}**.

The screenshot shows a web page titled "Unicode Text Steganography Encoders/Decoders". The page has a sidebar on the left with a green background, featuring a logo for "IronGeek .Com", a "REGISTER NOW" button, and a "Donate" button with payment method icons (Visa, MasterCard, American Express, Discover). Below the sidebar, there's a message: "Help IronGeek.com pay for bandwidth and research equipment!" with a "Donate" button.

The main content area contains several text input fields and labels:

- Cover Text To Use:** A text area containing "HTB{tr1th3m1u5\_1499}" with a note: "0 characters".
- Input (output if decoding):** A text area containing "HTB{tr1th3m1u5\_1499}" with a note: "20 characters to encode".
- Stegotext (input if decoding):** A text area containing a series of small, mostly illegible characters with a note: "65 real characters (not in bytes)".

Below these fields are three radio buttons: "Encode" (selected), "Decode", and "Reset". There are also two options: "Distribute Tag in Spaces" (selected) and "Put all Tags at end".

On the right side of the page, there are several green rounded rectangular boxes with white text, which are likely ads or related links:

- Ads by Google
- unicode steganograph
- anography text dec
- online text generat
- end message online
- trim a string

At the bottom of the main content area, there is a note: "This one is more complex than the one below, and uses better looking Homoglyphs. Mostly it uses whitespace for encoding. Got some ideas and suggestion from [Mick Douglas](#) on this one. It uses 7 bit ASCII to save space."

## 12.0 Pusheen Loves Graphs

1. Since it mention only IDA can read, so open with IDA.

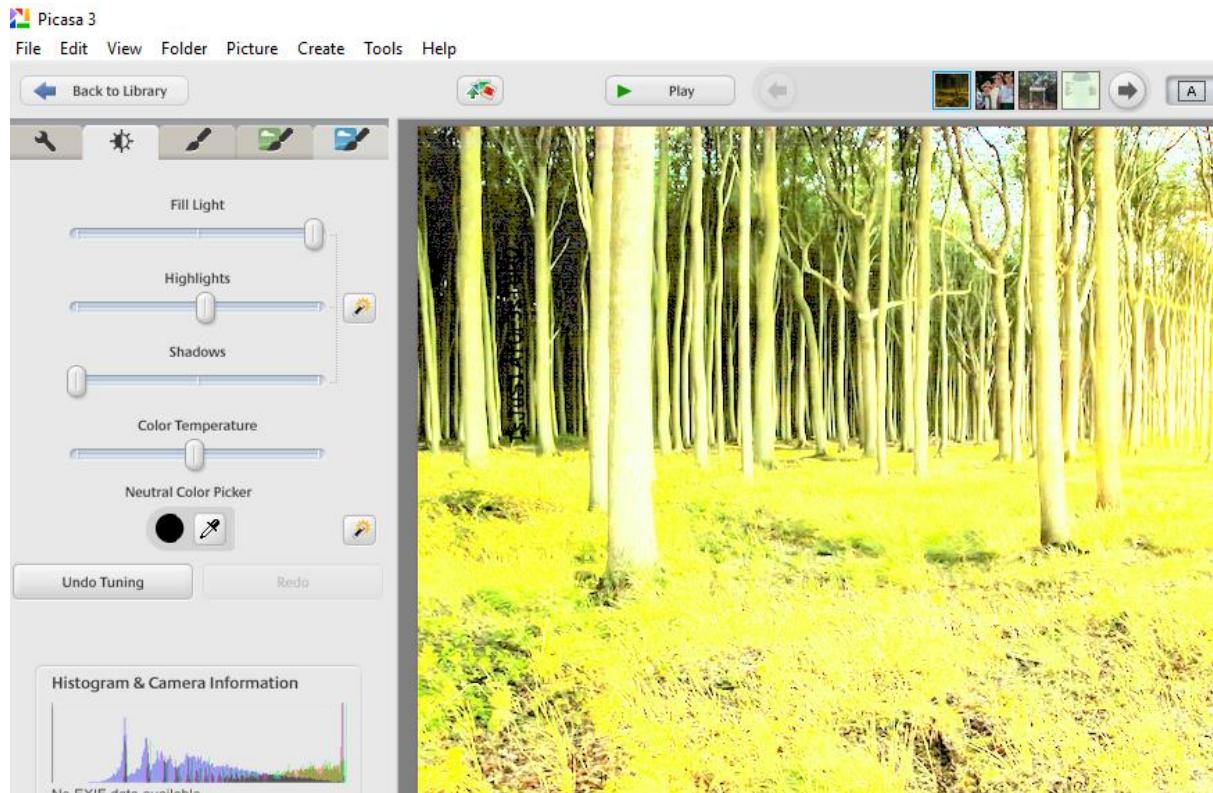
When loading the executable file in IDA it will throw an error on the maximum amount of nodes (in my case they were 1,000), then I raised it to 2,000 and then to 20,000 given that 2,000 was not enough.

Then it show the result below. **HTB{fUn\_w17h\_CFGz}**.



## 13.0 Forest

1. Using Picasa 3, adjust the light to highest, we can view some word on the pic. which is IsJuS1Af0r3sTbR0.



2. Use the phrase above to decode the forest.jpg.

```
Gur sberfg vf n pbzcyrk rpbflfgrz pbafvfgvat znvayl bs gerrf gung ohssre gur rnegu
naq fhccbeg n zlevnq bs yvsr sbezf. Gur gerrf uryc perngr n fcrpvny raivebazrag
juvpu, va gheax, nssrpgf gur xvaqf bs navznyf naq cynagf gung pna rkvg va gur sberfg.
Gerrf ner na vzcbeagnag pbzzbarag bs gur raivebazrag. Gurl pyrna gur nve, pbby vg ba
ubg qnlf, pbafreir urng ng avtug, naq npg nf rkpryyrag fbhaqnofbeoref.
UGO{NzNm1aTfXvyYmMOe0}
```

3. Use Caesar cipher to decode the flag: **HTB{AmAzInGsKilLzZBr0}**.

The screenshot shows the Cryptii Caesar cipher tool interface. On the left, under 'Plaintext', the flag **HTB{AmAzInGsKilLzZBr0}** is entered. In the center, the 'Caesar cipher' settings are displayed: SHIFT is set to 13, and the ALPHABET is abcdefghijklmnopqrstuvwxyz. Under CASE STRATEGY, 'Maintain case' is selected. FOREIGN CHARS options are 'Include' and 'Ignore'. Below these settings, a message indicates 'Decoded 22 chars in 0.07ms'. On the right, under 'Ciphertext', the decoded flag is shown as **UGO{NzNm1aTfXvyYmMoë0}**. The top right corner of the interface features a small advertisement for Spaghettifest 2017.

Caesar cipher: Encode and decode online

## 14.0 Blackhole

1. Download the file, unzip it. It will show a hawking file. Using HxD we could know it is a JIFF file which will use steganography. Upload the file to decoder using phrase hawking.

2. Decode it twice using base64 decoder to get a {}.

RWZxYnRxeiBjDxH4dW15IFRTaxD1enMgaW1IIG16IFF6c3h1ZXQgZnRxYW RxZnVvbXggYnRrZXVvd  
 WVmLCBvYWV5YXhhc3ViZiwgbXpwIG1NzRhZCwgxaXRhIGltZS BwdWRxb2ZhZCBhcIBkcWVxbWRvd  
 CBtZiBmdHeGt3F6ZmRxIHJhZCBGdHFhZH FmdW9teCBPYWV5YXhhc2sgbWYgZnRxIEd6dWhxZGV  
 1ZmsgYXlgT215bmR1chNxIg1mIGZ0cSBmdXlxlGFyIHR1ZSBwcW1mdC4gVHEgaW1IIGZ0cSBY29t  
 ZXVteiBCZGfycWVIYWQgYXlgWW1mdHF5bWZ1b2UgbWYgZnRxIEd6dWhxZGV1ZmsgYXlgT215bm  
 R1chNxIg5xZmlxcXogMTk3OSBtenAgMjAwOS4gVG1pd3V6cyBtb3R1cWhxcCBvYXl5cWRvdW14IGV  
 nb29xZWUgaXVm dCBlcWhxZG14IGlhZHdIIGFyI GJhYmd4bWQgZw91cXpvcSB1eiBpdHVvdCB0cSBw  
 dWVz2VlcWUgdHVIIGFpeiBmdHFhZH VxZSBtenAgb2FleWF4YXNrI HV6IH Nxe nFkbXgulFR1ZSBuYW  
 F3IE0gTmR1cXlgVHViZmFkayBhc iBGdXlxIg1iYnFtZHFwIGF6IGZ0cSBOZH VmdWV0IEVnenBtayBGd  
 XlxSBUcWVmLWVx eHhxZCB4dWVmIhJhZCbtIGRxb2FkcC1uZHFTd3V6cyAyMzcg aXFxd2UulFRtaXd  
 1enMgaW1IIG0gcnF4eGFpIGFyIGZ0cSBEY Wt eCBFYW91cWZrLCBtlHh1cmFmdXlxIhxeW5xZCBhcIB  
 mdHEgQmF6ZnVydW9teCBNb21wcXlrIGFyIEVvdXF6b3F1LCBtenAgbSBkcW91YnVxemYgYXlgZnRxIE  
 JkcWV1cHF6ZnVteCBz cXBteCBhc iBSZHFxcGF5LCBmdHEgdHVzdHFIZiBvdWh1eHVteiBtaW1kcCB1  
 eiRmdHEaR3n17nFwlFVmbW7x7S4nVXaaMiAwMiwnVG1nd3V6cvRnbWLIaZG16d3FwlHnn eW5xZCA

**For encoded binaries (like images, documents, etc.) upload your data via the [file decode form](#) below.**

Source charset.

Decodes in real-time when you type or paste (supports only unicode charsets).

[◀ DECODE ▶](#)

Decodes your data into the textarea below.

### Download Chrome Browser

Install Offline, Device-based Group Policies & More. Deploy Chrome MSI Google

[OPEN](#)

Efbqtz luxxumy Tmiwuzs ime mz Qzsxuet ftqadqfuomx btkeuouef, oae yaxasuef, mzp mqftad, ita ime  
 pudqofad ar dqe qmdot mf ftq Oqzf dq rad Ftqadqfuomx Oaeyaxask mf ftq Gzuhqdeufk ar Omyndupsg mf  
 ftq fuyq ar tue pqmft. Tq ime ftq Xqomeumz Bd ar qeead ar Ymftqymfuoe mf ftq Gzuhqdeufk ar  
 Omyndupsg nqf iqqz 1979 mzp 2009. Tmiwuzs motuqhqp oayyq doumx egoogee iuft eqh gdmx iadwe ar  
 babg xm d eo uqzqoq uz ituo t q pue ogee e tue aiz ftq aduqe mzp oae yaxask uz sqzqdmx. Tue naaw M  
 Nd uqr T ue fad k ar Fuyq mb bqm d qp az ftq Ndu fu et Egzpmk Fuyq neqf-eqxxqd xuef rad m dqoadp-  
 ndqm wuzs 237 iq qwe. Tmiwuzs ime m rqxxai ar ftq Dakmx Eaouqfk, m xurqfuvg vqynq d ar ftq  
 Bazfuruomx Mompqyk ar Eouqzqoe, mzp m dqoubugzf ar ftq Bdgeupqzfumx Yqpmx ar Rdqqpav, ftq  
 tustqef ouhuxumz mimdp uz ftq Gzufqp Ef mfae. Uz 2002, Tmiwuzs ime dmzwqp zgynq d 25 uz ftq NNOle  
 baxx ar ftq 100 Sdqmfqef Ndufaze.

3. Caesar cipher will do the rest. The flag is:

**HTB{N3veR\_l3T\_tH3\_b4sTaRd5\_G3t\_Y0u\_d0wN}.**



## 15.0 Longbottom's Locker

1. Extract the three file. Extract the file in the socute.jpg.

The terminal window shows the command `binwalk -e socute.jpg` being run. The output shows the extraction of several files, including a JPEG image, a Zip archive, and a folder named `donotshare`. The terminal then attempts to run a Python script named `hello.py`, which fails due to missing files. Finally, the command `python hello.py` is run successfully.

```
root@kali:~# cd Downloads
root@kali:~/Downloads# binwalk -e socute.jpg
root@kali:~/Downloads# nano hello.py
root@kali:~/Downloads# python hello.py
Traceback (most recent call last):
  File "hello.py", line 3, in <module>
    f = open ('/root/Downloads/donotshare')
IOError: [Errno 2] No such file or directory: '/root/Downloads/donotshare'
root@kali:~/Downloads# nano hello.py
root@kali:~/Downloads# python hello.py
Traceback (most recent call last):
  File "hello.py", line 3, in <module>
    f = open ('/root/Downloads/_socute.jpg.extracted/donotshare')
IOError: [Errno 2] No such file or directory: '/root/Downloads/_socute.jpg.extracted/donotshare'
root@kali:~/Downloads# nano hello.py
root@kali:~/Downloads# python hello.py
Traceback (most recent call last):
  File "hello.py", line 3, in <module>
    f = open ('/root/Downloads/_socute.jpg.extracted/donotshare.txt')
IOError: [Errno 2] No such file or directory: '/root/Downloads/_socute.jpg.extracted/donotshare.txt'
root@kali:~/Downloads# nano hello.py
root@kali:~/Downloads# python hello.py
root@kali:~/Downloads# python hello.py
```

2. Write a python script to extract.

The terminal window shows the creation of a Python script named `hello.py` using the `nano` editor. The script uses the `pickle` module to read a file named `donotshare` from the `Downloads` directory. It then iterates through the contents of the file, concatenating characters into a string `outstr`. Finally, it prints `outstr` to the console. The terminal also shows a file browser displaying the extracted files: `ACOSX`, `donotshare`, and `_socute.jpg.extracted`.

```
root@kali:~# nano 3.2
GNU nano 3.2
import pickle
f = open ('/root/Downloads/donotshare')
o = pickle.load(f)
outstr = ''
for line in o:
    for char, n in line:
        outstr += char*n
    outstr += '\n'

print outstr
[Read 14 lines]
```

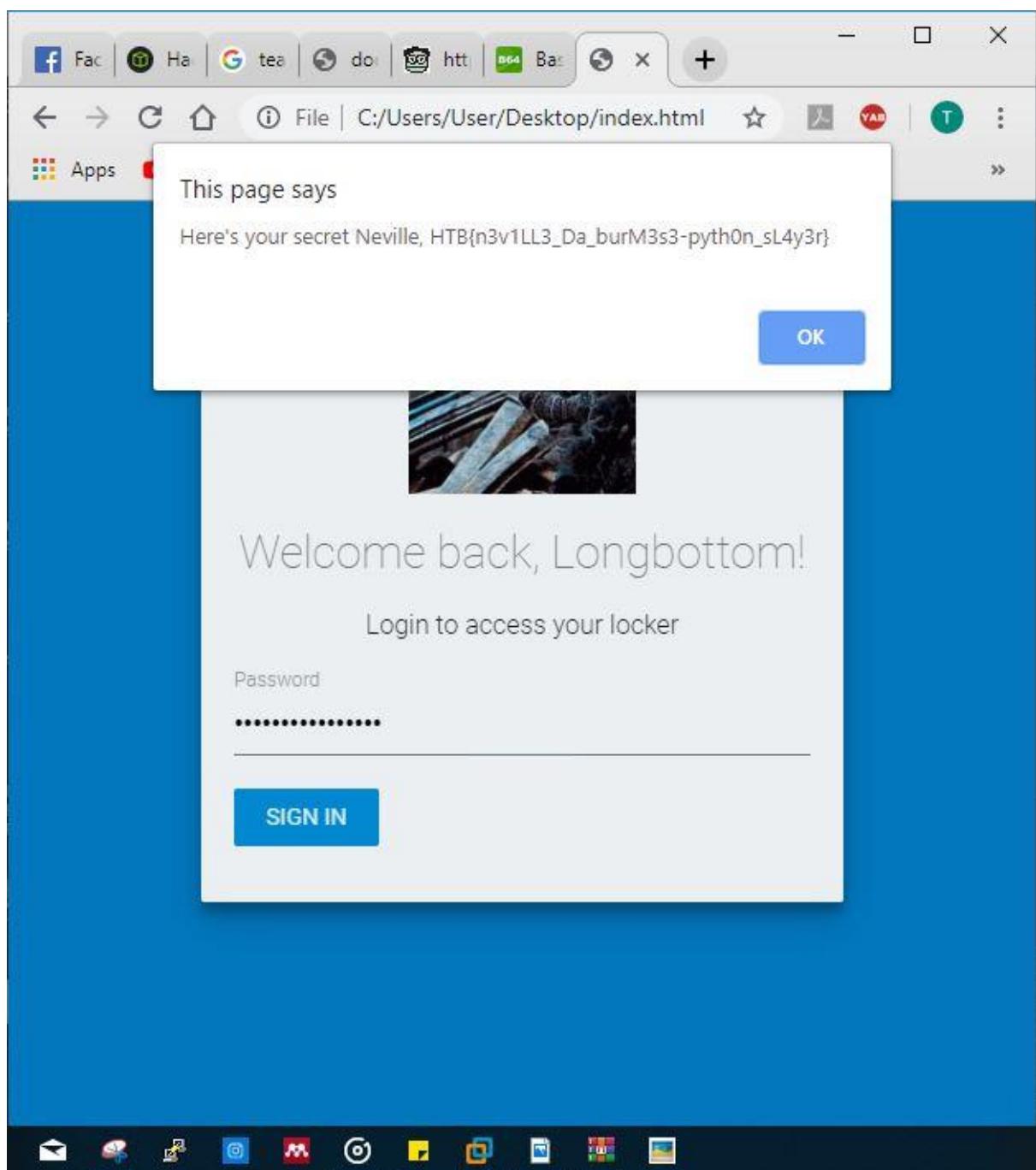
### 3. Run the script.

The screenshot shows a terminal window titled "root@kali: ~/Downloads" running on Kali Linux. The terminal displays the following command-line session:

```
rootkali:~/Downloads# nano hello.py
rootkali:~/Downloads# python hello.py
Traceback (most recent call last):
  File "hello.py", line 3, in <module>
    f = open ('/root/Downloads/donotshare')
IOError: [Errno 2] No such file or directory: '/root/Downloads/donotshare'
rootkali:~/Downloads# nano hello.py
rootkali:~/Downloads# python hello.py
Traceback (most recent call last):
  File "hello.py", line 3, in <module>
    f = open ('/root/Downloads/ socute.jpg.extracted/donotshare')
IOError: [Errno 2] No such file or directory: '/root/Downloads/ socute.jpg.extracted/donotshare'
rootkali:~/Downloads# nano hello.py
rootkali:~/Downloads# python hello.py
Traceback (most recent call last):
  File "hello.py", line 3, in <module>
    f = open ('/root/Downloads/ socute.jpg.extracted/donotshare.txt')
IOError: [Errno 2] No such file or directory: '/root/Downloads/ socute.jpg.extracted/donotshare.txt'
rootkali:~/Downloads# nano hello.py
rootkali:~/Downloads# python hello.py
Traceback (most recent call last):
  File "hello.py", line 3, in <module>
    f = open('/root/Downloads/donotshare.txt')
IOError: [Errno 2] No such file or directory: '/root/Downloads/donotshare.txt'
rootkali:~/Downloads# nano hello.py
rootkali:~/Downloads# python hello.py
```

The terminal then displays a large amount of binary data consisting of repeating patterns of 'd8888b.' and 'Y88b' characters.

4. Use the key below and enter to the index.html. The flag is: **HTB{n3v1LL3\_Da\_burM3s3-pyth0n\_sL4y3r}**.

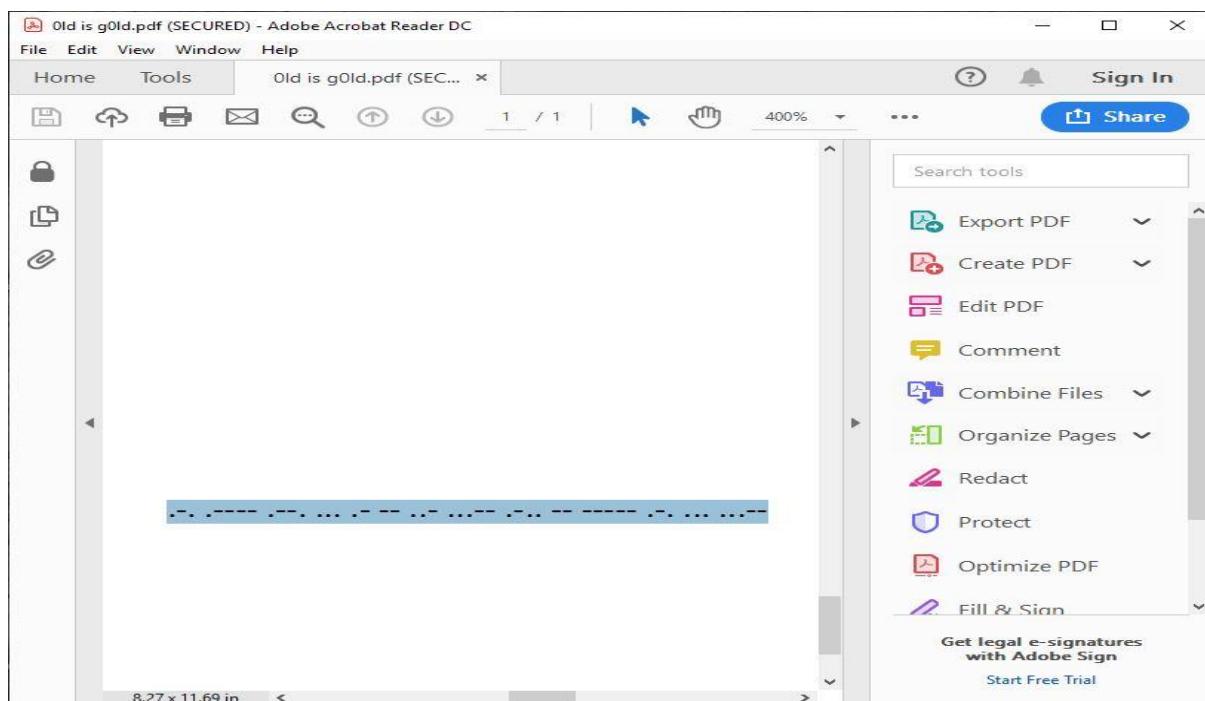


## 16.0 Old\_is\_g0ld

1. Extract it. It requires password to unlock pdf file. Use pdfcrack in kali and get password jumanji69.

A screenshot of a Kali Linux desktop environment. The desktop background features the classic Kali logo. A terminal window is open in the center, showing the command 'pdfrack -f "Old is g0ld.pdf" -w rockyou.txt' being run by root. The terminal displays the progress of the PDF cracking process, including file version information, security handler details, and a list of password attempts with their respective speeds. The desktop interface includes a top bar with 'Applications', 'Places', and 'Terminal' buttons, and a dock on the left containing icons for various applications like a web browser, file manager, and terminal.

2. Unlock the pdf then we can see tiny morse code on below.



3. Decode the morse code. The flag is: **HTB{R1PSAMU3LM0RS3}**.

The screenshot shows a web application titled "Morse Code Translator" from the website "SCPhillips.com". The main interface has a dark header with the site name and a navigation bar with links like "Blog", "Morse Code", "Units", "Dance", and "CV". Below the header, there's a secondary navigation bar with tabs for "Morse", "Translator" (which is active), "Training", "Audio Decoder", "Gaze Decoder", "Keyer", "The Code", "Timing", "Alphabets", and "FAQ".

The main content area is titled "Translate a Message". It features two text input fields: "Input" containing Morse code and "Output" showing the decoded text "R1PSAMU3LM0RS3". Below these fields are several buttons: "Translate" (orange), "Sound" (checkbox checked), "Light" (checkbox unchecked), and three other small green and blue buttons. To the right of the main content is a sidebar with a Microsoft advertisement for Office 365 Home. The ad includes the text "Akses dunia anda" and "Kongsi langganan Office 365 Home dalam kalangan 6 orang pada berbilang peranti". At the bottom of the page, there are links for "Send your message to a friend" and "Advanced Controls".

## 17.0 Inferno

### 1. Extract the content. Decode with base64.

The screenshot shows a web application for decoding base64 encoded data. The main area displays a yellow McDonald's promotional banner for the "McD ELECTIONS" event. Below the banner, there is a text area containing the decoded content of the banner image, which appears to be a series of encoded characters. There are also input fields for ASCII and source charset, and a "DECODE" button. A sidebar on the right contains links for "Other tools" and "URL Decode".

2. Decode again with Malbolge decoder. The flag is  
**HTB{!1t\_1s\_just\_M4lb0lg3\_l4ngu4g3!}.**

The screenshot shows a terminal window within a web-based development environment. The terminal window has tabs for "Hello World", "Library", "About the project", and "New program". The "Terminal" tab is active, displaying the command "HTB{!1t\_1s\_just\_M4lb0lg3\_l4ngu4g3!}" and its output. The "Program code:" tab shows the Malbolge encoded program. The bottom of the window includes social media sharing icons (g+, f, t, in, r) and a copyright notice: "© 2019 PROMYK".

18.0 fsociety

1. Use fcrackzip to get the password. Use the password to open the fsociety.zip

The screenshot shows a Kali Linux desktop environment. On the left, there's a vertical dock with various icons. The main workspace has a terminal window titled "root@kali: ~/Downloads" and a Firefox browser window titled "Hack the Box Challenge: fsociety - Mozilla Firefox".

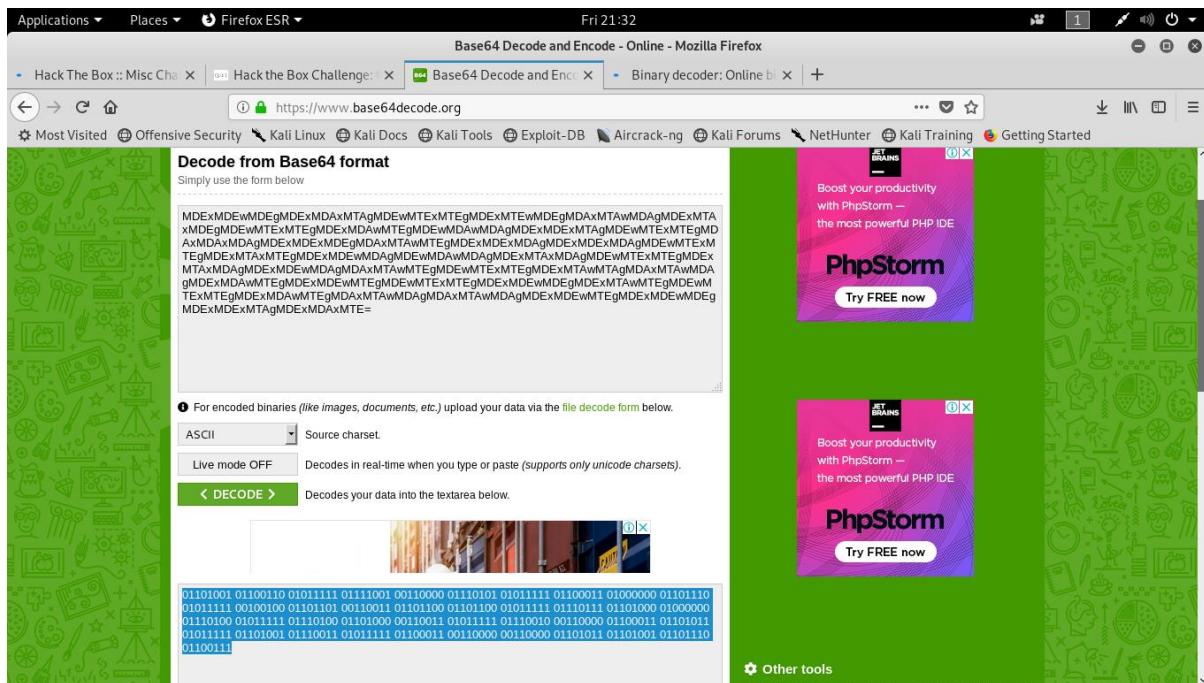
**Terminal Window Content:**

```
root@kali:~# fcrackzip -u -D -p 'rockyou.txt' fsociety.zip
skipping 'fsociety.zip': No such file or directory
no usable files found
root@kali:~# cd Downloads
root@kali:~/Downloads# fcrackzip -u -D -p 'rockyou.txt' fsociety.zip
First thing's first, we need to do what we can to make sure that can use
If you're following my articles by putting up a password protected zip file,
not, please refer to the "Old is gold" guide.
Since I haven't used fcrackzip before, I'll start by looking at the man
pages for it. The first thing I do is check the man pages for it.
man fcrackzip
Since I'm going to leverage rockyou.txt, I'll use the -D, -u and -p switches (case-sensitive).
root@kali:~/Downloads# fcrackzip -u -D -p 'rockyou.txt' fsociety.zip
I don't recall how long the first time took, but when I reran it to write this guide, it
```

**Firefox Browser Window Content:**

The Firefox window displays a blog post from "postpnedramblings" about cracking a zip file using fcrackzip. It provides a step-by-step guide, mentioning the use of the "rockyou.txt" wordlist and the "-D", "-u", and "-p" options.

2. In the ssh text file, there is base64 encoded text. So, decode it. It shows all the binary number.



3. Use binary translate to text. The flag is: HTB{if\_y0u\_\$m3ll\_wh@t\_th3\_r0ck\_is\_c00king}

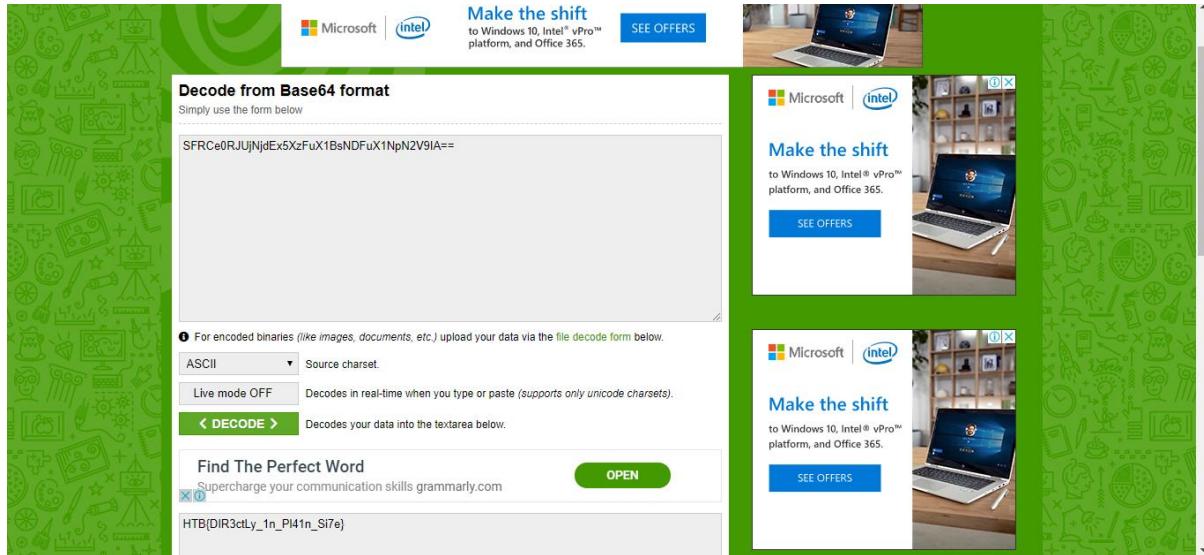
The screenshot shows a Firefox browser window with the title "Binary decoder: Online binary to text translator — Cryptii - Mozilla Firefox". The address bar shows the URL <https://cryptii.com/pipes/binary-decoder>. The page displays a binary-to-text conversion tool. On the left, under the "Bytes" view, there is a large block of binary code. On the right, under the "Text" view, the binary code is converted into the readable string: "if\_y0u\_c@n\_\$m3ll\_wh@t\_th3\_r0ck\_is\_c00king". The interface includes dropdown menus for "FORMAT" (set to "Binary") and "GROUP BY" (set to "Byte"). A status message at the bottom left says "Transferring data from srv.carbonads.net...".

## 19. misDIRection

1. Download the file. Follow the number and arrange the folder accordingly.

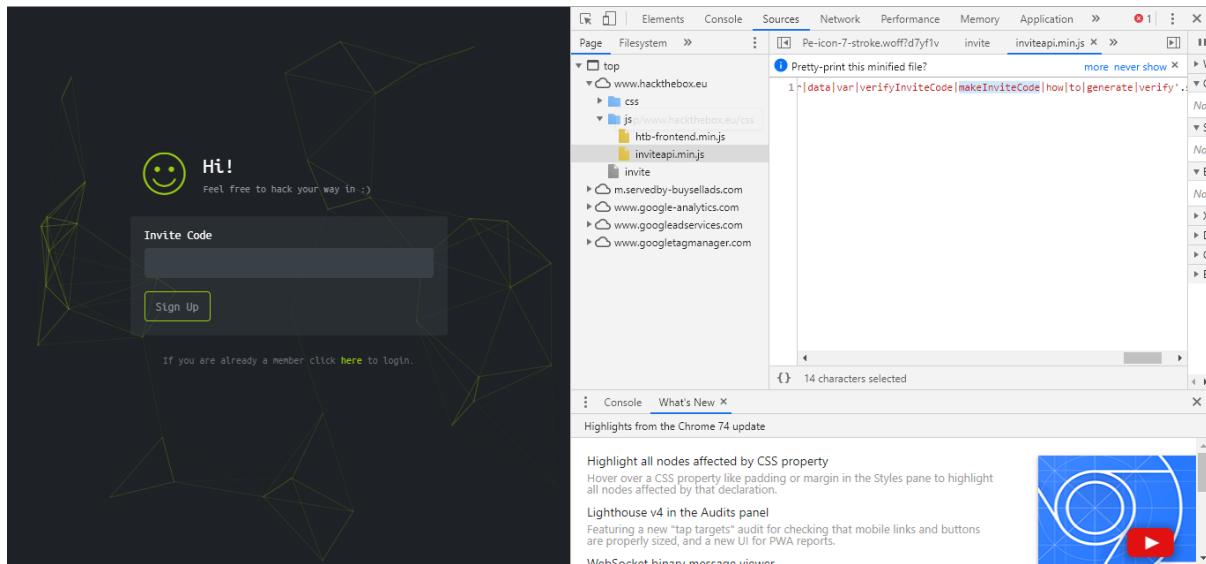
SFRCe0RJUjNjdEx5XzFuX1BsNDFuX1NpN2V9IA==

2. Decode by base64. The flag is: **HTB{DIR3ctLy\_In\_Pl41n\_Si7e}**



## 20.0 Hack for Invitation Code.

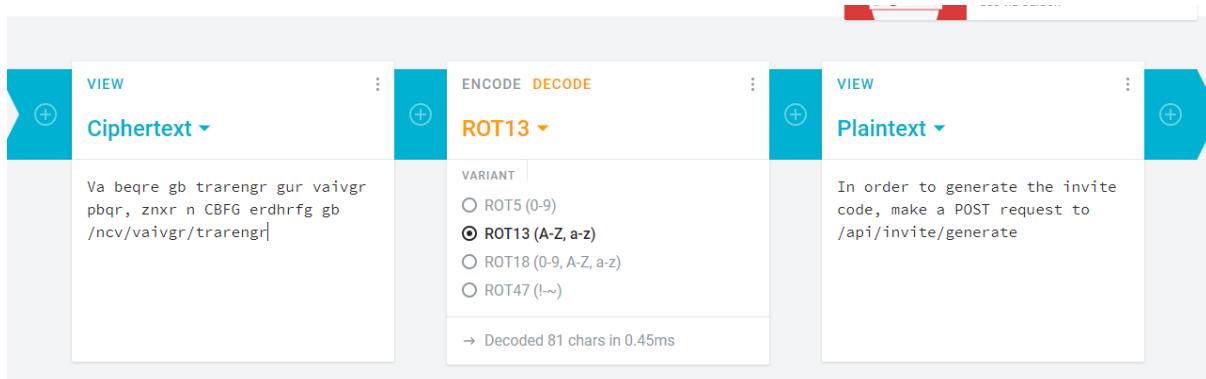
1. View the invite.js, we can see there is makeInviteCode.



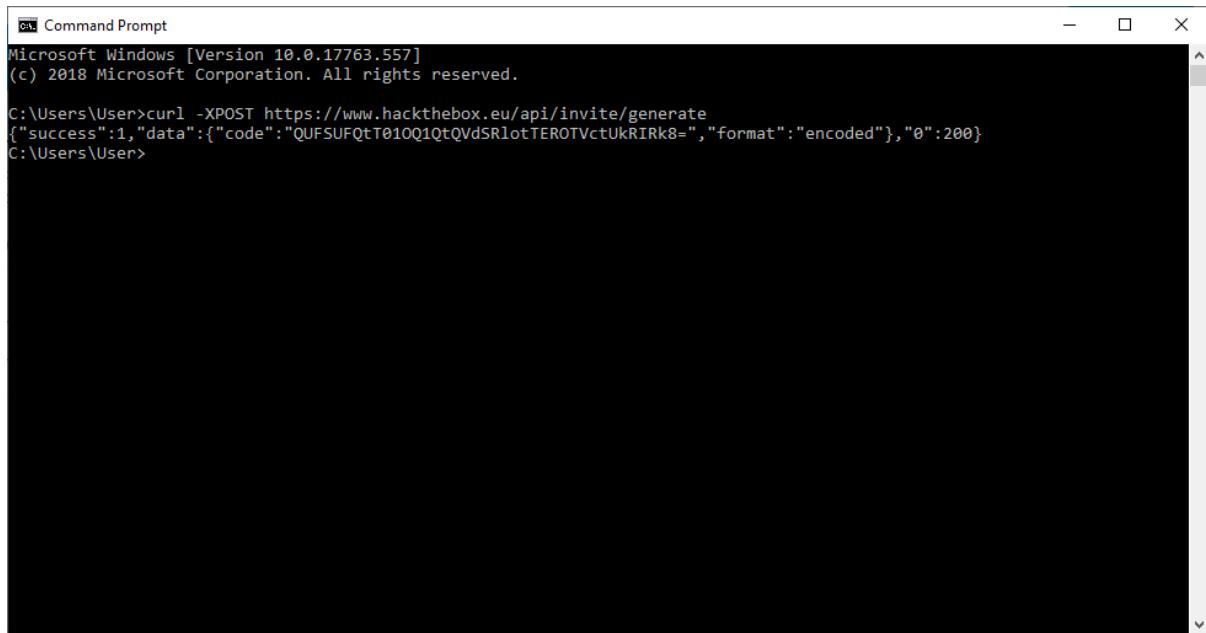
2. At the console, run makeInviteCode(). It will show a message with rot13 encoded.



3. Decode and get the message.



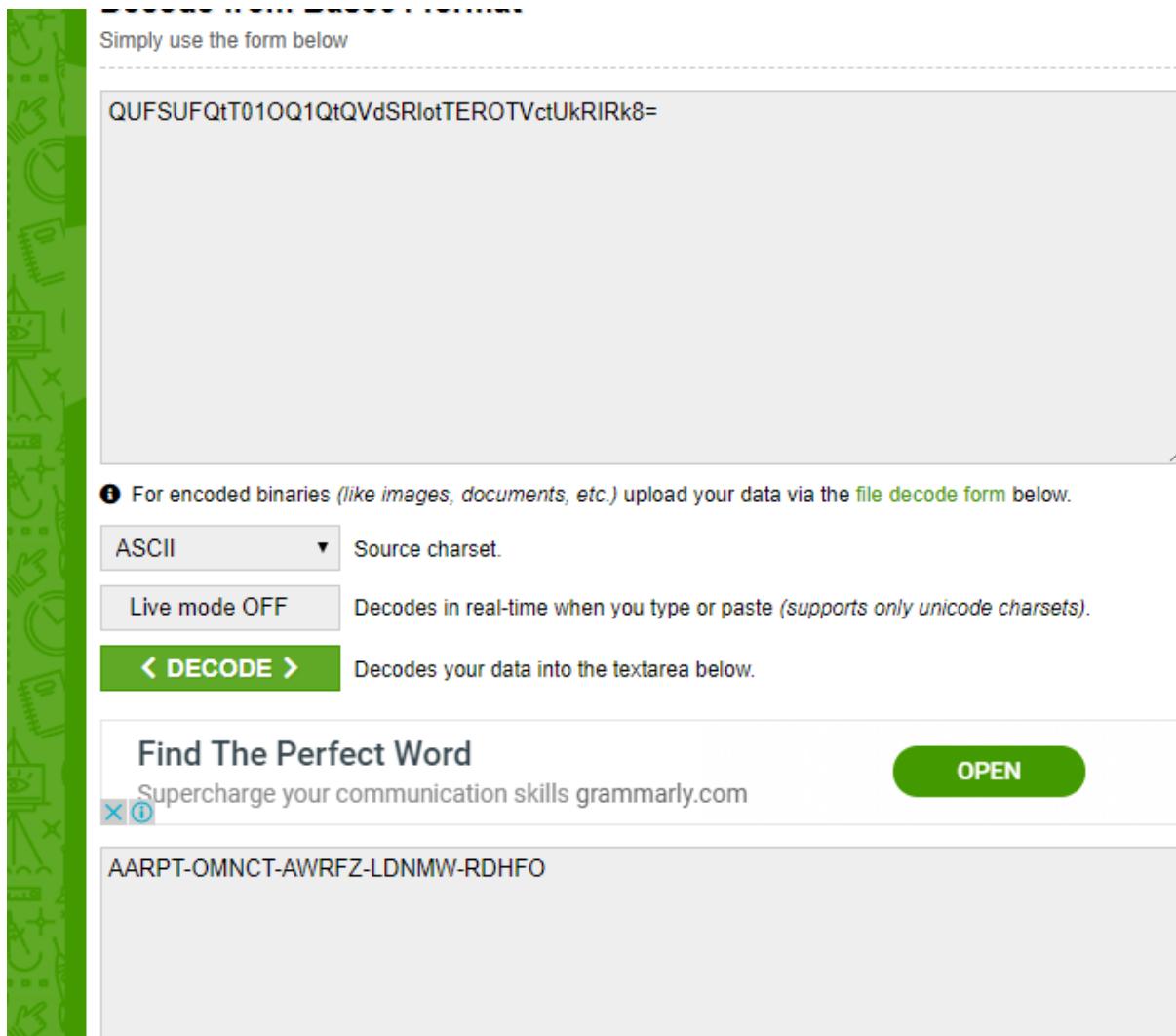
4. Use terminal to make a POST request.



```
Command Prompt
Microsoft Windows [Version 10.0.17763.557]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>curl -XPOST https://www.hackthebox.eu/api/invite/generate
{"success":1,"data":{"code":"QUFSUFQtT01OQ1QtQVdSRIotTEROTVctUkRIRk8=","format":"encoded"},"o":200}
C:\Users\User>
```

5. Decode the message and get the invitation code.



Simply use the form below

QUFSUFQtT01OQ1QtQVdSRIotTEROTVctUkRIRk8=

For encoded binaries (like images, documents, etc.) upload your data via the [file decode form](#) below.

ASCII ▾ Source charset.

Live mode OFF Decodes in real-time when you type or paste (supports only unicode charsets).

**< DECODE >** Decodes your data into the textarea below.

**Find The Perfect Word** Supercharge your communication skills [grammarly.com](#)

**OPEN**

AARPT-OMNCT-AWRFZ-LDNMW-RDHFO

6. Successful.

