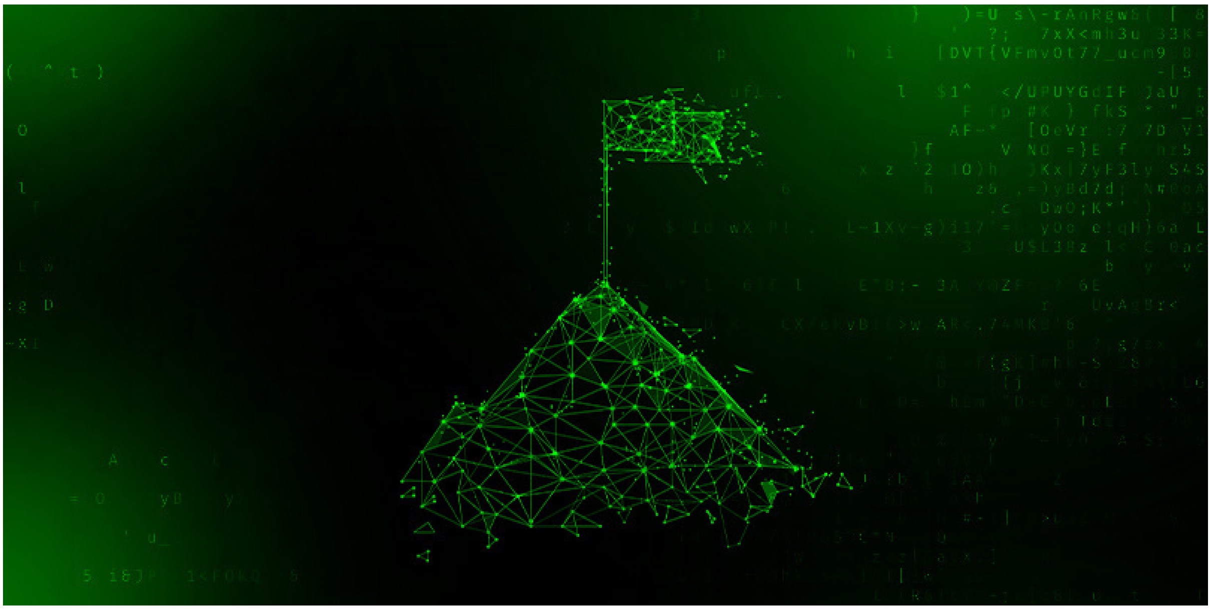
II. 3524 Software Security

Antonin Vannier

PROJECT 2

CTF



isep

École d'ingénieurs du numérique

Summary

CTF 1 - Cryptology “Unrecognized hash”

Challenge description

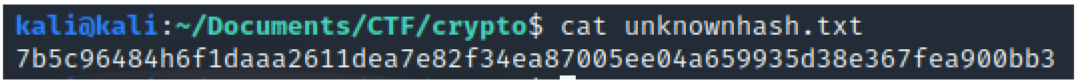
Scenario : You are given a hash, but its format seems unknown. Find the corresponding uncracked password.

Documents given : unknownhash.txt, readme.txt and challengevalidation.zip

Validation : In order to validate this challenge, try to open the challengevalidation.zip by issuing the MD5 of the password.

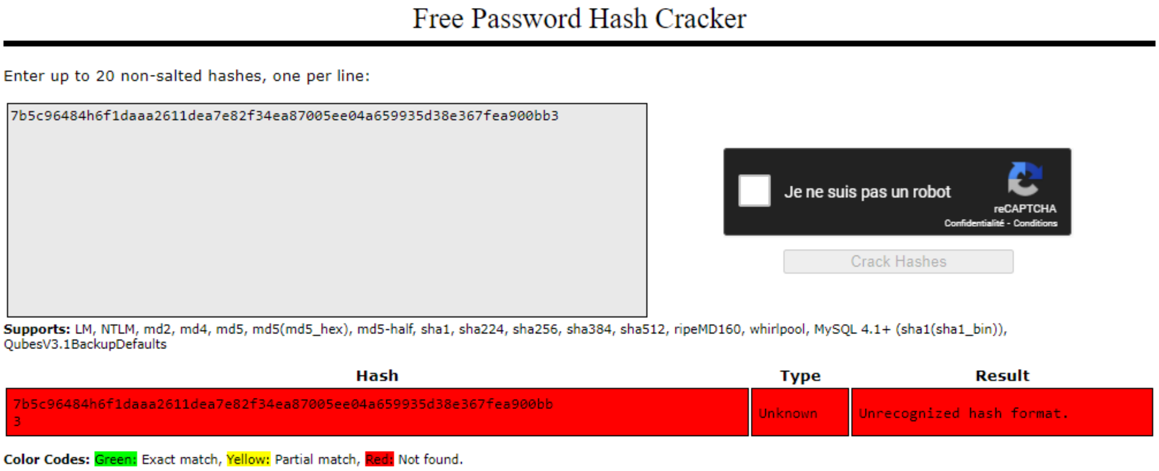
Challenge solution

Here is what the hash looks like :



```
kali@kali:~/Documents/CTF/crypto$ cat unknownhash.txt
7b5c96484h6f1daaa2611dea7e82f34ea87005ee04a659935d38e367fea900bb3
```

If you try to crack the Hash using well known websites, such as CrackStation.net, you obtain this result :



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
7b5c96484h6f1daaa2611dea7e82f34ea87005ee04a659935d38e367fea900bb3
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
7b5c96484h6f1daaa2611dea7e82f34ea87005ee04a659935d38e367fea900bb 3	Unknown	Unrecognized hash format.

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

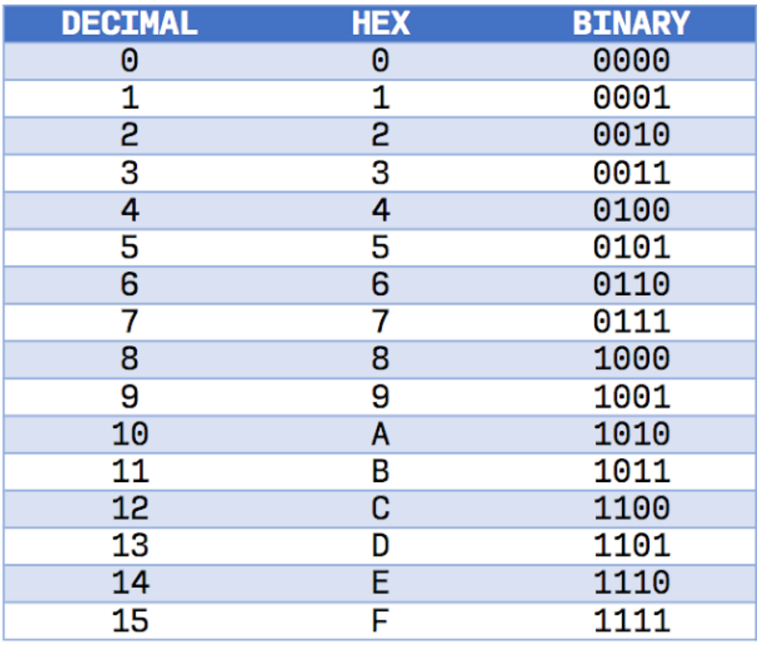
The format of the hash is unrecognized.

A hint here is to look at the length of the hash.

You know that hash formats have a specific length, for instance it is 32 digits for MD5 hash, and 64 digits for SHA256 hash.

If you count the number of digits in unknownhash.txt, you see that there are 65 digits.

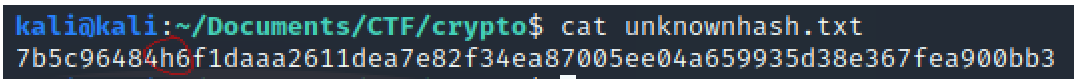
Now if you look closely at every digit, you will see that something is wrong.
Our hash here is written in hexadecimal.



DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

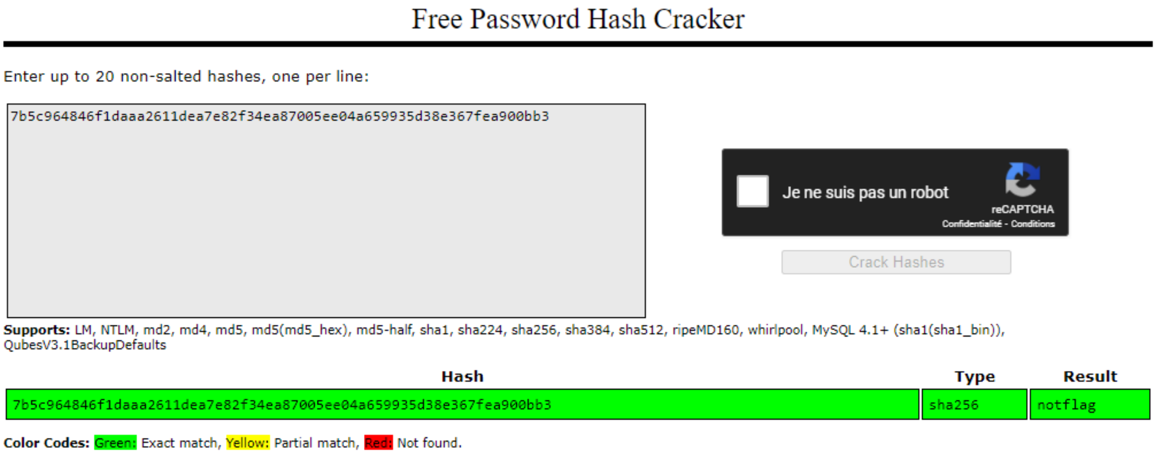
In hexadecimal, numbers stop at 9 and letters stop at F.

But here, in the unrecognized hash given, you can see an ‘h’.



```
kali@kali:~/Documents/CTF/crypto$ cat unknownhash.txt
7b5c96484h6f1daaa2611dea7e82f34ea87005ee04a659935d38e367fea900bb3
```

If you remove it and try to crack it again, it will now work.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

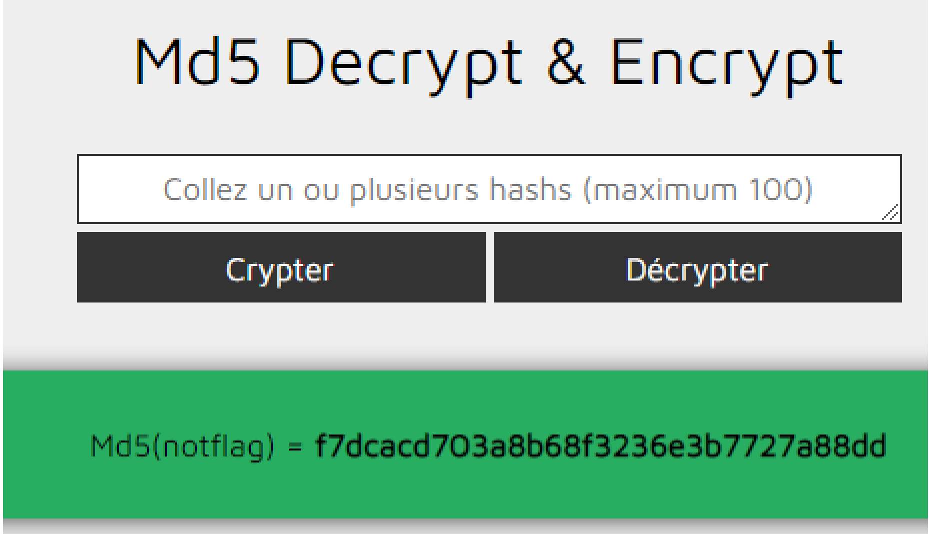
```
7b5c964846f1daaa2611dea7e82f34ea87005ee04a659935d38e367fea900bb3
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
7b5c964846f1daaa2611dea7e82f34ea87005ee04a659935d38e367fea900bb3	sha256	notflag

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

The clear text equivalent of the hash is “notflag”.
To end the challenge, you just have to encrypt this string using MD5 :


Md5 Decrypt & Encrypt

Md5(notflag) = f7dcacd703a8b68f3236e3b7727a88dd

CTF 2 - Cryptography “Secret message”

Challenge description

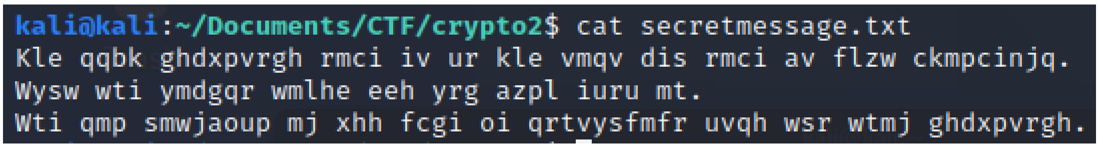
Scenario : You need to decipher the secret message to move on to the next challenge !

Documents : You are given a zip containing a secretmessage.txt and a readme.txt

Validation : In order to validate this challenge, you have to open the document of the third challenge.

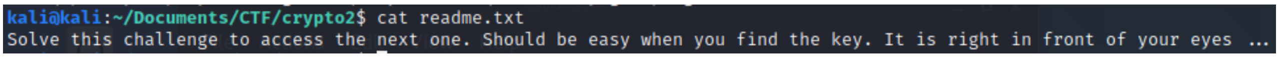
Challenge solution

Here is what looks like the secret message.



```
kali@kali:~/Documents/CTF/crypto2$ cat secretmessage.txt
Kle qqbk ghdxpvrgh rmci iv ur kle vmqv dis rmci av flzw ckmpcinjq.
Wysw wti ymdgqr wmlhe eeh yrg azpl iuru mt.
Wti qmp smwjaoup mj xhh fcgi oi qrtvysfmfr uvqh wsr wtmj ghdxpvrgh.
```

You can guess that it has been ciphered using Caesar’s cipher or something similar.

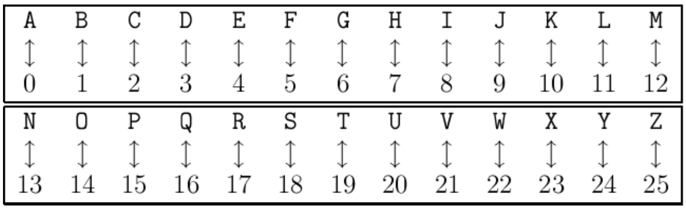


```
kali@kali:~/Documents/CTF/crypto2$ cat readme.txt
Solve this challenge to access the next one. Should be easy when you find the key. It is right in front of your eyes ...
```

The readme.txt gives you a hint because it says that you have to find the “key”. You can now be sure that the message was ciphered using the Vigenère cipher.

The Vigenère cipher uses a key to cipher messages.

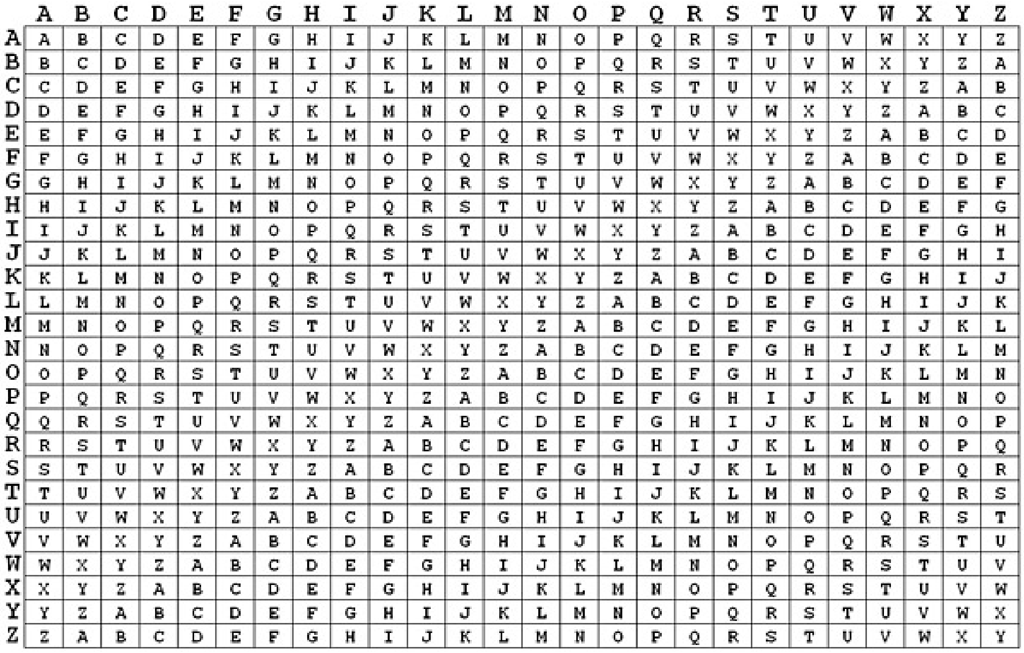
Every letter of the alphabet is associated with a number.



A	B	C	D	E	F	G	H	I	J	K	L	M
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
13	14	15	16	17	18	19	20	21	22	23	24	25

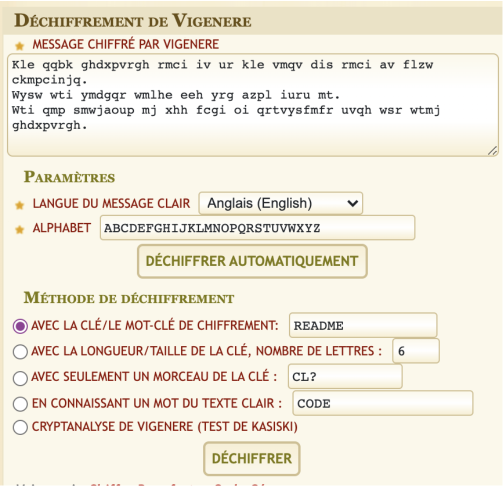
To cipher a message, you have to add up the equivalent number of the message that you want to cipher with the key, letter by letter.
You can also use a table and cross the letter of the message and the letter of the key to obtain the ciphered letter.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Nowadays, powerful online tools allow you to easily crack this algorithm, and even without the key you can have the unciphered message.

The readme.txt said that the key is right in front of you, because the key used to cipher the message was “README”.



MESSAGE CHIFFRÉ PAR VIGENERE

```
Kle qqbk ghdxpvrgh rmci iv ur kle vmqv dis rmci av flzw
ckmpcinjq.
Wysw wti ymdgqr wmlhe eeh yrg azpl iuru mt.
Wti qmp smwjaoup mj xhh fcgi oi qrtvysfmfr uvqh wsr wtmj
ghdxpvrgh.
```

LANGUE DU MESSAGE CLAIR: Anglais (English)

ALPHABET: ABCDEFGHIJKLMNOPQRSTUVWXYZ

● AVEC LA CLÉ/LE MOT-CLÉ DE CHIFFREMENT: README

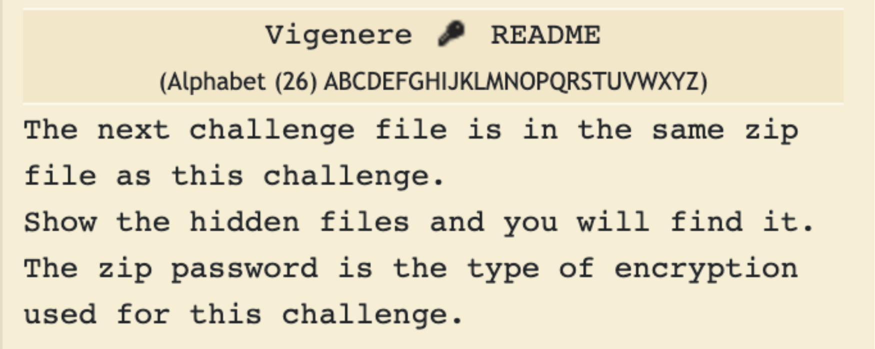
○ AVEC LA LONGUEUR/TAILLE DE LA CLÉ, NOMBRE DE LETTRES : 6

○ AVEC SEULEMENT UN MORCEAU DE LA CLÉ : CL?

○ EN CONNAISSANT UN MOT DU TEXTE CLAIR : CODE

○ CRYPTANALYSE DE VIGENERE (TEST DE KASISKI)

If you decode the message, you obtain :

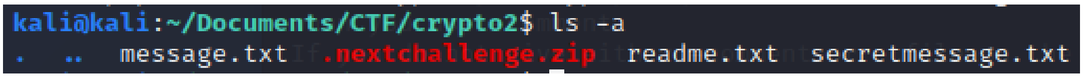


Vigenere 🔑 README

(Alphabet (26) ABCDEFGHIJKLMNOPQRSTUVWXYZ)

```
The next challenge file is in the same zip
file as this challenge.
Show the hidden files and you will find it.
The zip password is the type of encryption
used for this challenge.
```

So you just need to make a ‘ls -a’ to see the file.



```
kali@kali:~/Documents/CTF/crypto2$ ls -a
.  ..  message.txt  .nextchallenge.zip  readme.txt  secretmessage.txt
```

The password of the archive is Vigenere.

CTF 3 - Steganography

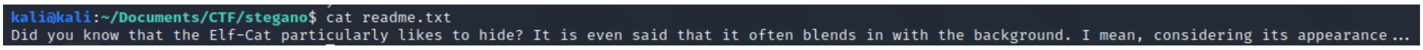
Challenge description

Scenario : You are given a basic pdf file and you need to find hidden information.

Documents : (you need to complete CTF 2 in order to have these) ElfCat.pdf and readme.txt.
Validation : Find the flag (format “flag {...}”)

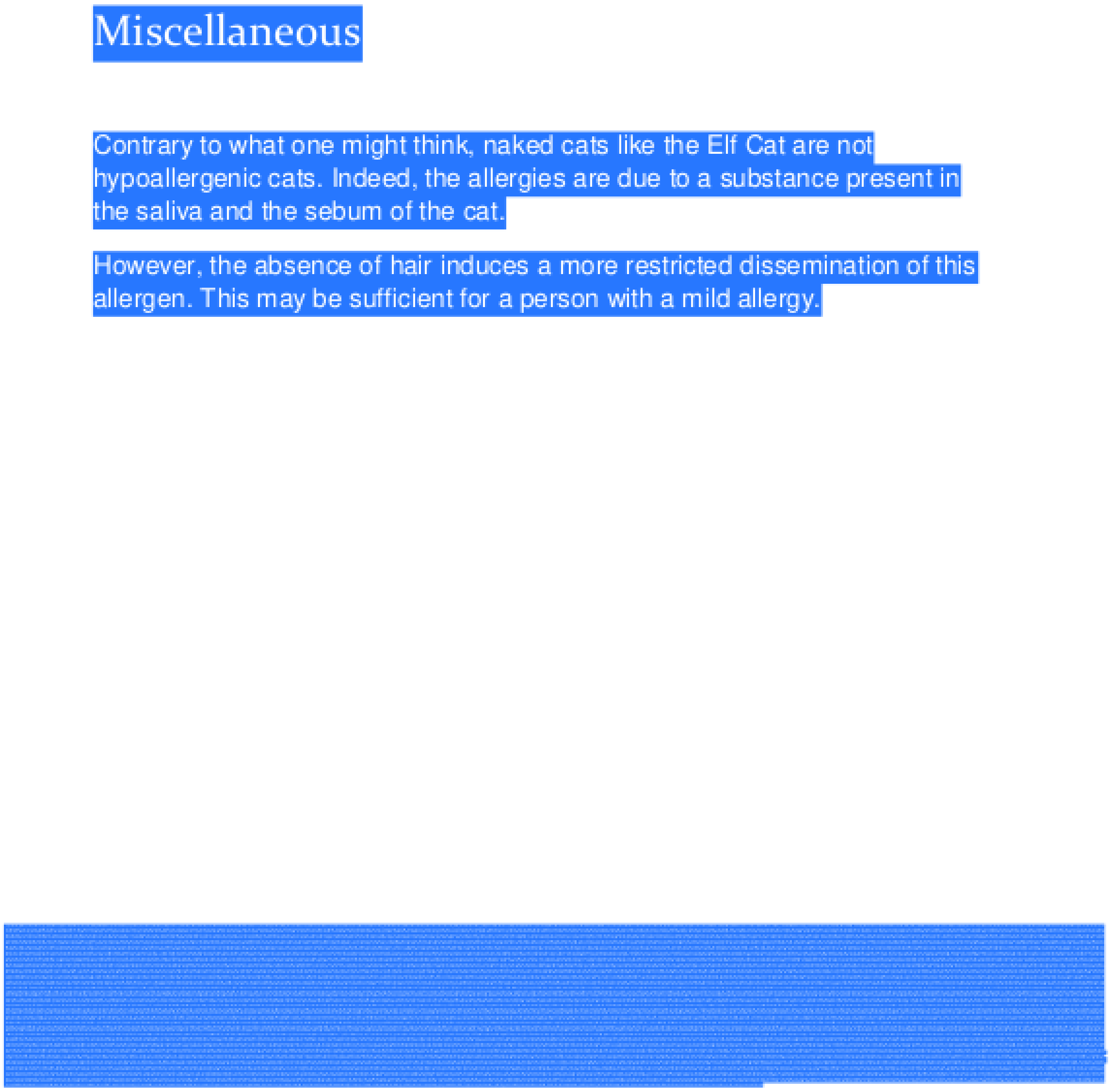
Challenge solution

When you first open the pdf file, you don’t see anything relevant. But a hint is given in the readme.txt :



```
kali@kali:~/Documents/CTF/stegano$ cat readme.txt
Did you know that the Elf-Cat particularly likes to hide? It is even said that it often blends in with the background. I mean, considering its appearance...
```

"if often blends in with the background". Something must be hidden in this pdf. By making a ctrl+a and by scrolling to the last page, you see that an unreadable string appears.



Miscellaneous

Contrary to what one might think, naked cats like the Elf Cat are not hypoallergenic cats. Indeed, the allergies are due to a substance present in the saliva and the sebum of the cat.

However, the absence of hair induces a more restricted dissemination of this allergen. This may be sufficient for a person with a mild allergy.

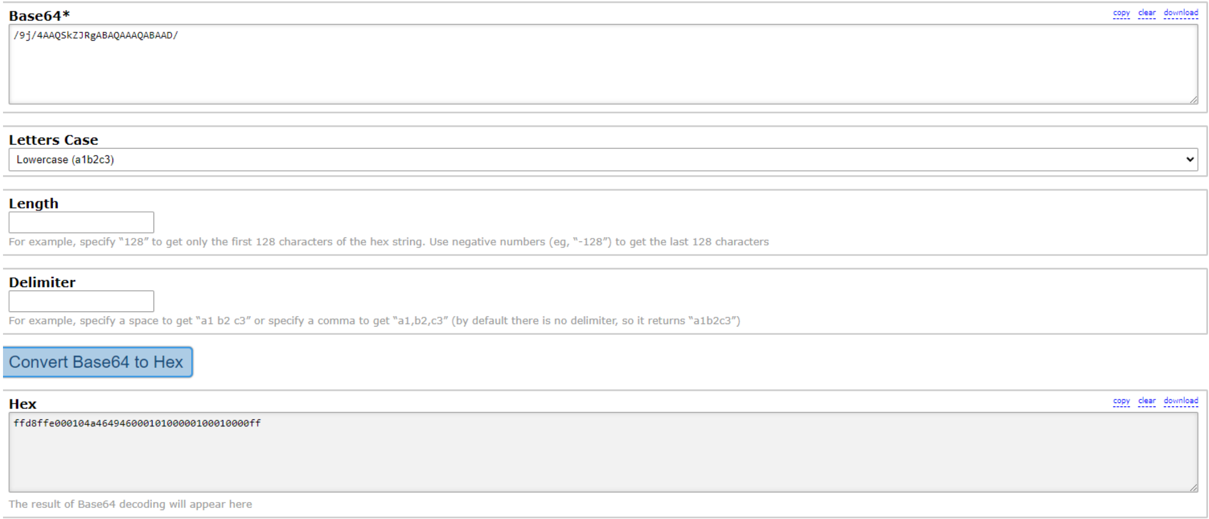
If you now copy everything and remove all the text concerning elf cat, you can now start to study the string that you got.

You can notice that there are numbers, letters and special numeric characters such as "/", "+" or "-". The string is therefore in base 64.
The thing is that the string is very very long, either it is a huge encoded text, or maybe it can be something else, encoded in base 64.

By searching a bit on the internet, you can find information concerning the beginning of the string that can give you a clue of its nature.

For instance, JPEG file, encoded in hexadecimal, starts with FFD8.

So let’s try to check if our base64 string starts with the same. You can easily convert your base64 into hexadecimal.



Base64*

```
/9j/4AAQSkZJRgABAQAAAQABAAD/
```

Letters Case

Lowercase (a1b2c3)

Length

For example, specify "128" to get only the first 128 characters of the hex string. Use negative numbers (eg, "-128") to get the last 128 characters

Delimiter

For example, specify a space to get "a1 b2 c3" or specify a comma to get "a1,b2,c3" (by default there is no delimiter, so it returns "a1b2c3")

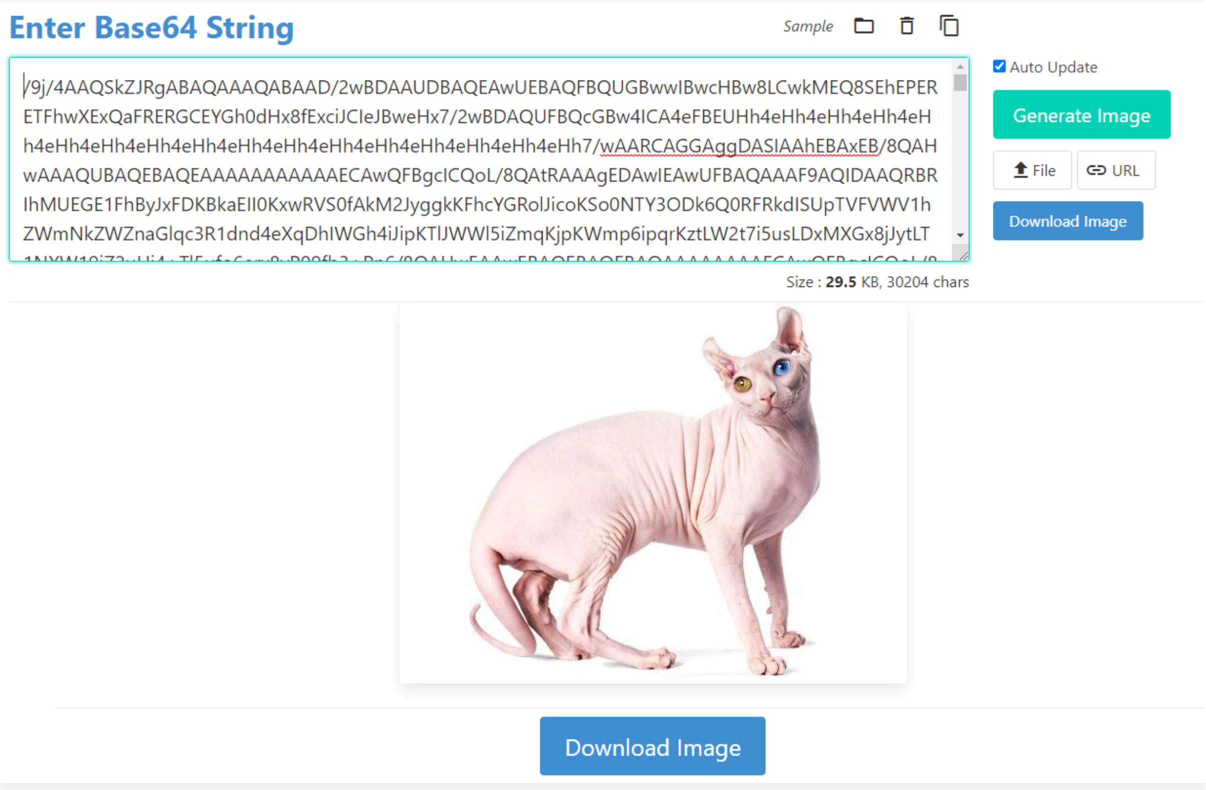
Hex

```
ffd8ffe000104a46494600010100000100010000ff
```

The result of Base64 decoding will appear here

And, yes, the hexadecimal starts with FFD8; so your base64 string is most likely an image !

Now, you need to find a tool to convert base64 text into image.



Enter Base64 String

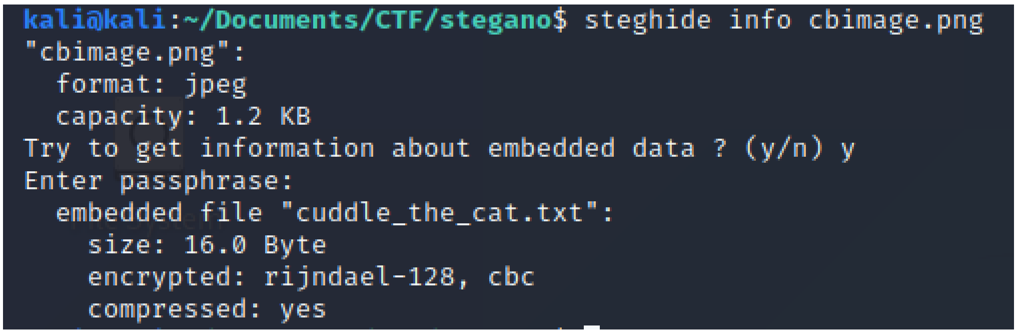
/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAUDBAQEAwUEBAQFBQUGBwwIBwcHBw8LCwkMEQ8SEhEPER
ETFhwXExQaFRERGCEYGh0dHx8fExciJCIeJBweHx7/2wBDAQUFBQcGBw4ICA4eFBEUHh4eHh4eHh4eHh4eH
h4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh4eHh7/wAARCAGGAggDASIAAhEBAxEB/8QAH
wAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBR
IhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1h
ZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT

Size : **29.5** KB, 30204 chars

It is an image of an elf cat, so you must be in the right direction.

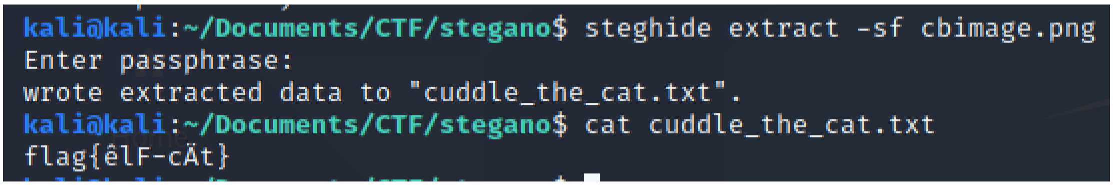
You know that you need to find a flag to validate the challenge. You now need to check if there isn’t any embedded data in your image.

You can do it using steghide.



```
kali@kali:~/Documents/CTF/stegano$ steghide info cbimage.png
"cbimage.png":
  format: jpeg
  capacity: 1.2 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "cuddle_the_cat.txt":
    size: 16.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
```

You see that there is an embedded file called “cuddle_the_cat.txt” in your image. You can now extract it, still using steghide :



```
kali@kali:~/Documents/CTF/stegano$ steghide extract -sf cbimage.png
Enter passphrase:
wrote extracted data to "cuddle_the_cat.txt".
kali@kali:~/Documents/CTF/stegano$ cat cuddle_the_cat.txt
flag{êlF-cÄt}
```

And you have the flag !