BITSCTF 2017: fanfie

8 February 2017 (https://codisec.com/bitsctf-2017-fanfie/) Robert Tomkowski (https://codisec.com/author/robert-tomkowski/) — No Comments (https://codisec.com/bitsctf-2017-fanfie/#respond)

CTF: BITSCTF 2017
Points: 20
Category: Crypto

DESCRIPTION

> Brute and get the base 32 format of flag.
> encrypted.txt: `MZYVMIWLGBL7CIJOGJQVOA3IN5BLYC3NHI`

This task is worth 20 points, but only 8 teams have solved it during ctf and I really wonder why.

Before we start, I assume that everyone knows how base32 works: **link (http://www.herongyang.com/Encoding/Base32-Encoding-Algorithm.html)**.

SOLUTION

Task description tells us that flag is converted to base32 and somehow encrypted.

From other tasks we know flag format and when we compare it with ciphertext length, we can assume that plaintext looks like this: `BITSCTF{**********}`.

Let's encode first five letters of flag (one block of base32), `BITSC` to base32: `IJEVIU2D`.

Compare first 5 letters of base32 plaintext and ciphertext:

```
I J E V I U 2 D
M Z Y V M I W L
```

We can notice that every letter in ciphertext decodes to distinct letter in plaintext (with `M` decoding twice to `I`), so we can guess that this is kind of monoalphabetic substitution cipher.

Let's look for any patterns in ciphertext alphabet.

Our alphabet(all base32 letters):

```
 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  2  3  4  5  6  7
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

Encrypting alphabet

```
 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  2  3  4  5  6  7
 ?  ?  ?  L  Y  ?  ?  ?  M  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  I  V  ?  ?  ?  J  W  ?  ?  ?  ?  ?
        11 24          12                                    8 21          9 22
```

So:

```
3  -> 11
4  -> 24
8  -> 12
20 ->  8
21 -> 21
25 ->  9
26 -> 22
```

When we look closely we can see that this is encrypted with **affine cipher (https://en.wikipedia.org/wiki/Affine_cipher)**, with `a = 13` and `b = 4`.

By the way, after finding that pattern we realize that title of this task is anagram of **affine** word.

So our encrypted alphabet will look like this:

```
 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  2  3  4  5  6  7
 4 17 30 11 24  5 18 31 12 25  6 19  0 13 26  7 20  1 14 27  8 21  2 15 28  9 22  3 16 29 10 23
 E  R  6  L  Y  F  S  7  M  Z  G  T  A  N  2  H  U  B  O  3  I  V  C  P  4  J  W  D  Q  5  K  X
```

Now we can get our plaintext:

```
MZYVMIWLGBL7CIJOGJQVOA3IN5BLYC3NHI -> IJEVIU2DKRDHWUZSKZ4VSMTUN5RDEWTNPU
```

After that we need to add b32 padding and finally we can read our solution:

Python

```
import base64
base64.b32decode('IJEVIU2DKRDHWUZSKZ4VSMTUN5RDEWTNPU======')
-> BITSCTF{S2VyY2tob2Zm}
```

Easter egg:
flag is base64 of string 'Kerckhoff':

```
$ echo -n "S2VyY2tob2ZZm" | base64 -d
Kerckhoff
```

Kerckhoffs's principle (https://en.wikipedia.org/wiki/Kerckhoffs's_principle).

Tagged with: crypto (https://codisec.com/tag/crypto/)

LEAVE A REPLY

Required fields are marked

Your comment*

Your name*

Your e-mail*

Your website

POST COMMENT

Recent Posts

BITSCTF 2017: fanfie (https://codisec.com/bitsctf-2017-fanfie/) 8 February 2017
6th place on SECCON 2016 finals (https://codisec.com/6th-place-seccon-2016-finals/) 3 February 2017
Veles release 2017.1 (https://codisec.com/veles-release-2017-1/) 26 January 2017
Binary visualization explained (https://codisec.com/binary-visualization-explained/) 13 January 2017
Binary data visualization (https://codisec.com/binary-data-visualization/) 2 December 2016

Categories

CTF 2016 (https://codisec.com/category/ctf-2016/) (16)
- BackdoorCTF 2016 (https://codisec.com/category/ctf-2016/backdoorctf16/) (4)
- DefCamp CTF 2016 (https://codisec.com/category/ctf-2016/defcamp-ctf-2016/) (1)
- Tokyo Westerns/MMA CTF 2nd 2016 (https://codisec.com/category/ctf-2016/tokyo-westernsmma-ctf-2nd-2016/) (4)
- TU CTF 2016 (https://codisec.com/category/ctf-2016/tu-ctf-2016/) (1)
- TUMCTF 2016 (https://codisec.com/category/ctf-2016/tumctf-2016/) (2)
- WhiteHat contest 11 (https://codisec.com/category/ctf-2016/whitehat11/) (4)

CTF 2017 (https://codisec.com/category/ctf-2017/) (1)
- BITSCTF (https://codisec.com/category/ctf-2017/bitsctf/) (1)

News (https://codisec.com/category/news/) (8)
Veles (https://codisec.com/category/veles/) (3)

Tags

crypto (https://codisec.com/tag/crypto/) forensic (https://codisec.com/tag/forensic/) image (https://codisec.com/tag/image/) misc (https://codisec.com/tag/misc/) network (https://codisec.com/tag/network/) png (https://codisec.com/tag/png/) pwn (https://codisec.com/tag/pwn/) python (https://codisec.com/tag/python/) RE (https://codisec.com/tag/re/) sound (https://codisec.com/tag/sound/) stegano (https://codisec.com/tag/stegano/) steganography (https://codisec.com/tag/steganography/)