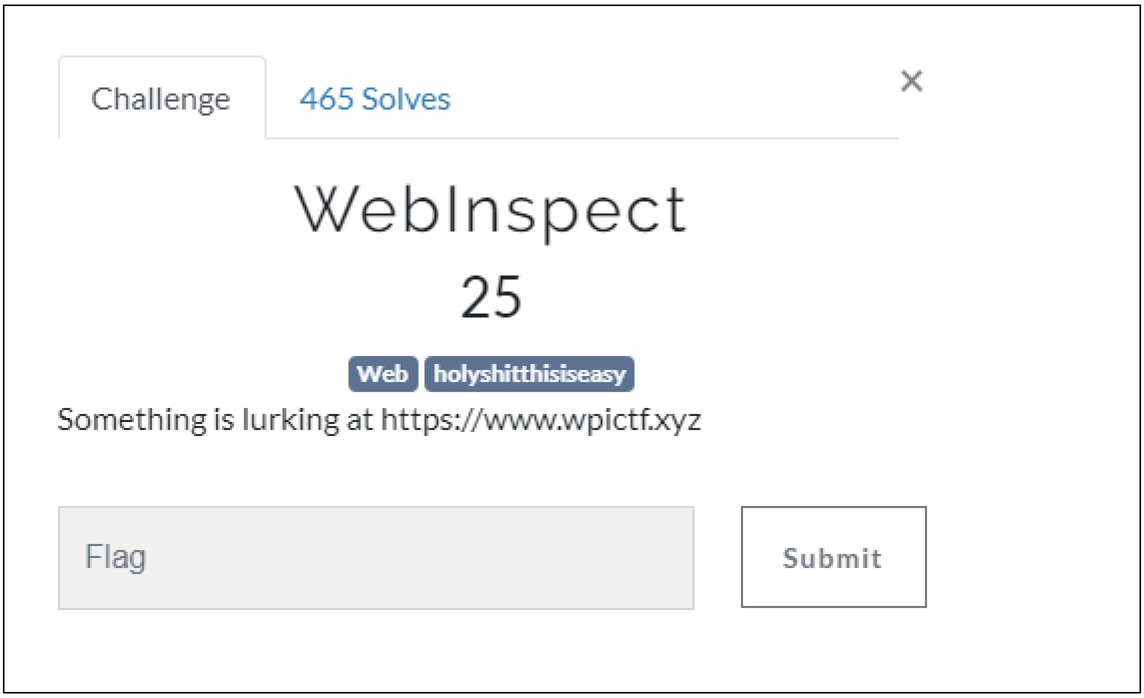
TENESYS

IAM9ROOT

AKINARI

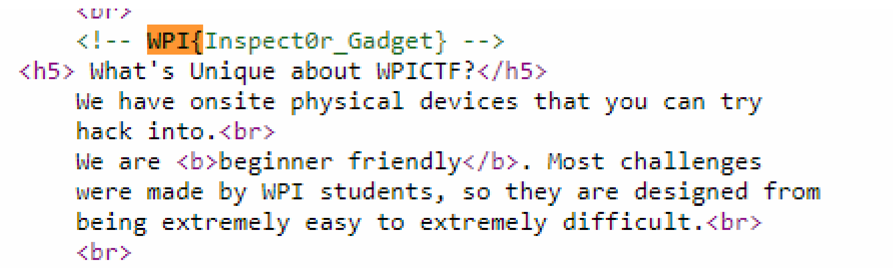
TEKNOKRAT AND SYSTEM SECURITY TENESYS 2019

1. Webinspect – 25 point – Web –



Something is lurking at https://www.wpictf.xyz

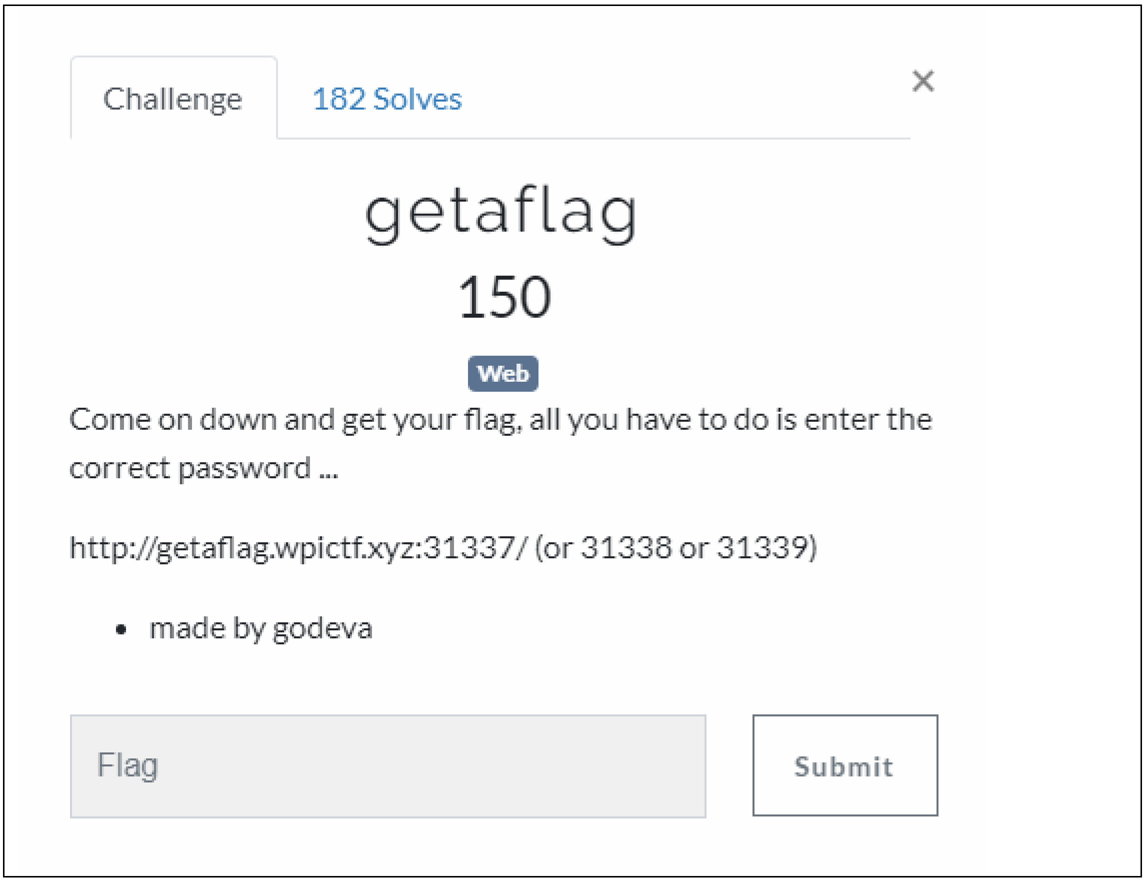
- Didapatkan sebuah website
- Lalu lihat view source + search “WPI{“



```
        <br>
        <!-- WPI{Inspect0r_Gadget} -->
    <h5> What's Unique about WPICTF?</h5>
        We have onsite physical devices that you can try
        hack into.<br>
        We are <b>beginner friendly</b>. Most challenges
        were made by WPI students, so they are designed from
        being extremely easy to extremely difficult.<br>
        <br>
```

- Dan didapatkan flag : **WPI{Inspect0r_Gadget}**

2. getaflag – 150 point – Web –



getaflag

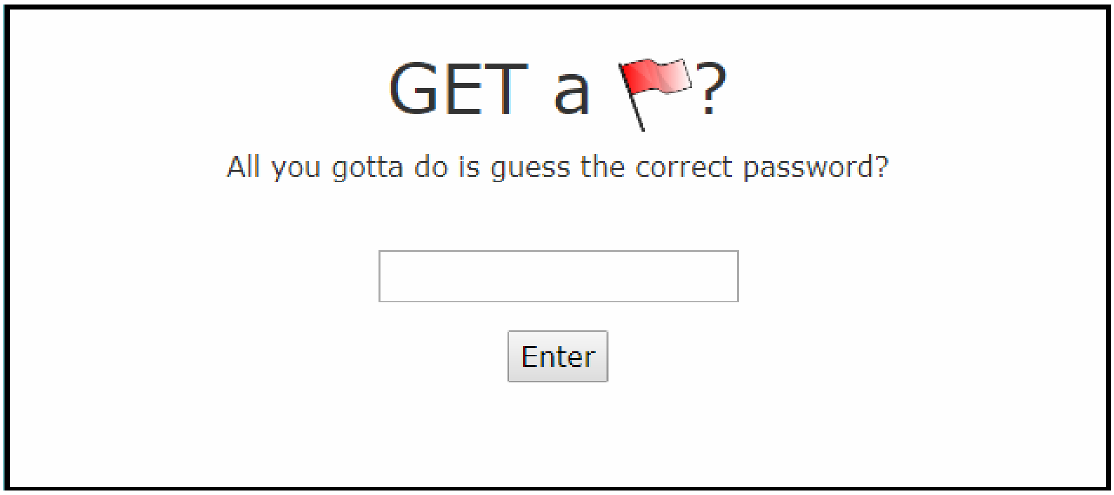
150

Come on down and get your flag, all you have to do is enter the correct password ...

http://getaflag.wpictf.xyz:31337/ (or 31338 or 31339)

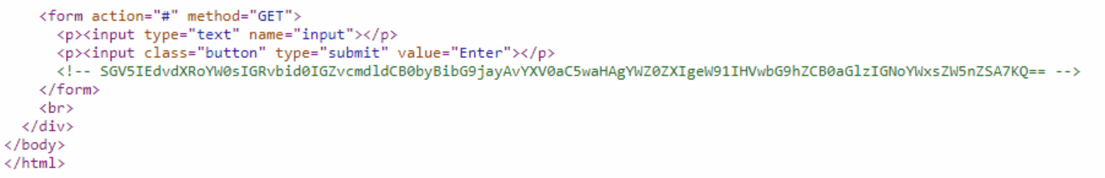
- made by godeva

- Didapatkan sebuah website


GET a 🚩?

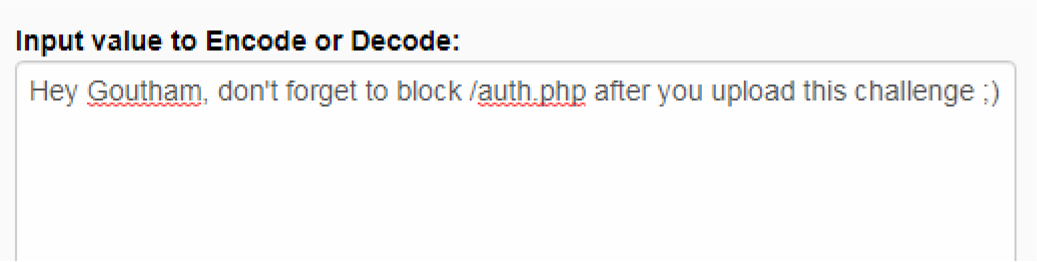
All you gotta do is guess the correct password?

- Lalu di cek terlebih dahulu view sourcenya dan ada base64



```
    <form action="#" method="GET">
      <p><input type="text" name="input"></p>
      <p><input class="button" type="submit" value="Enter"></p>
      <!-- SGV5IEdvdXRoYW0sIGRvbid0IGZvcmdldCB0byBibG9jayAvYXV0aC5waHAgYWZ0ZXIgeW91IHVwbG9hZCB0aGlzIGNoYWxsZW5nZSA7KQ== -->
    </form>
    <br>
  </div>
</body>
</html>
```

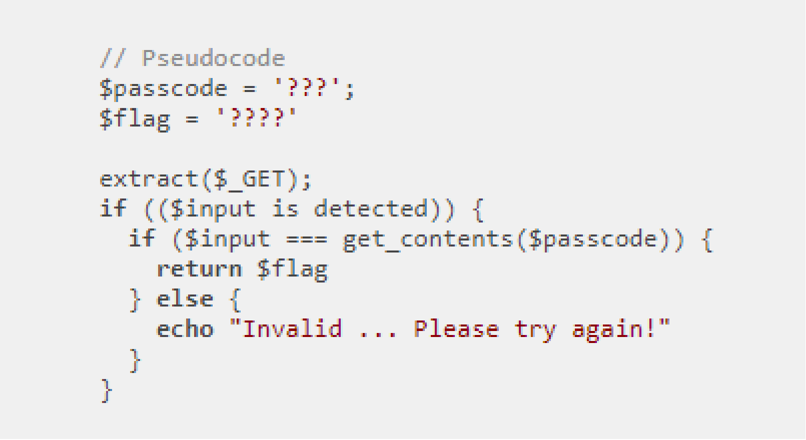
- Lalu didecode dan didapatkan auth.php



Input value to Encode or Decode:

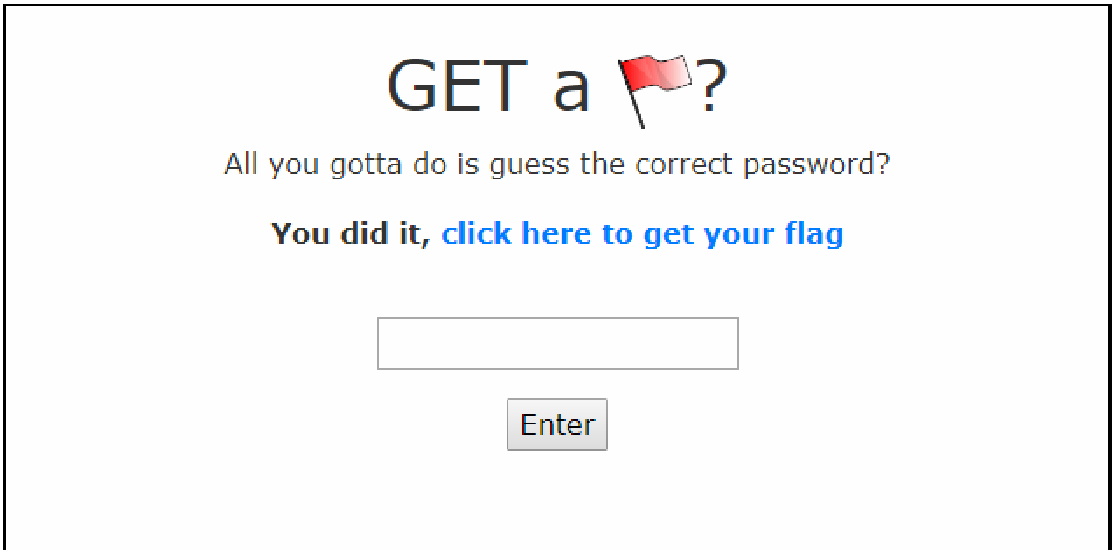
Hey Goutham, don't forget to block /auth.php after you upload this challenge ;)

- Lalu akses http://getaflag.wpictf.xyz:31337/auth.php



```
// Pseudocode
$passcode = '???';
$flag = '????'

extract($_GET);
if (($input is detected)) {
  if ($input === get_contents($passcode)) {
    return $flag
  } else {
    echo "Invalid ... Please try again!"
  }
}
```

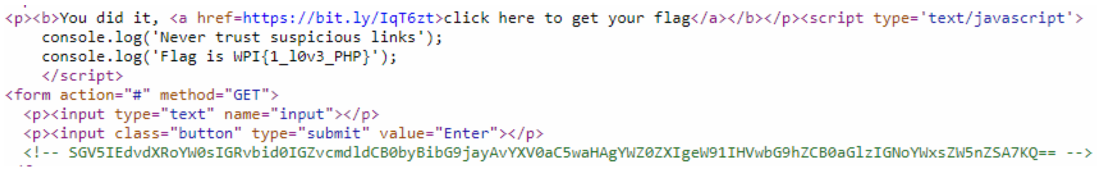
- Lalu akses http://getaflag.wpictf.xyz:31337/?input=&passcode=



All you gotta do is guess the correct password?

You did it, click here to get your flag

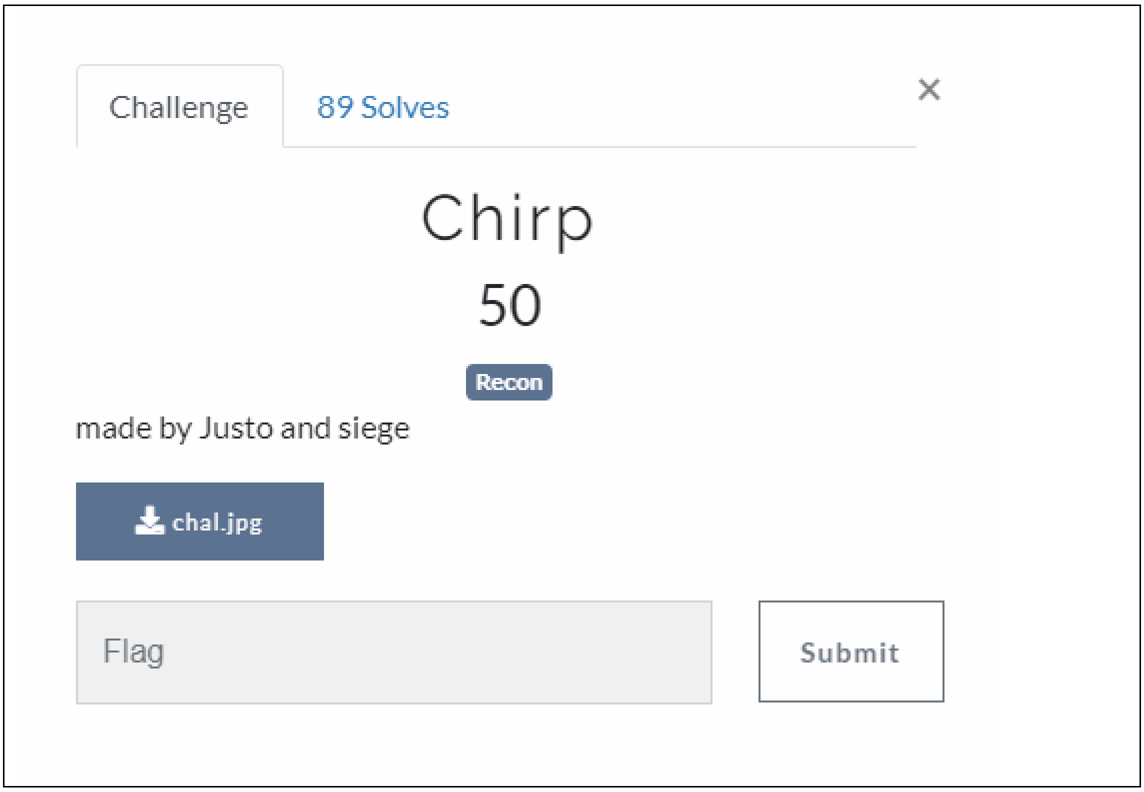
- Lalu lihat view source kembali



```
<p><b>You did it, <a href=https://bit.ly/IqT6zt>click here to get your flag</a></b></p><script type='text/javascript'>
    console.log('Never trust suspicious links');
    console.log('Flag is WPI{1_l0v3_PHP}');
    </script>
<form action="#" method="GET">
  <p><input type="text" name="input"></p>
  <p><input class="button" type="submit" value="Enter"></p>
  <!-- SGV5IEdvdXRoYW0sIGRvbid0IGZvcmdldCB0byBibG9jayAvYXV0aC5waHAgYWZ0ZXIgeW91IHVwbG9hZCB0aGlzIGNoYWxsZW5nZSA7KQ== -->
```

- Dan didapatkan flag : **WPI{1_l0v3_PHP}**

3. Chirp – 50 point – Recon –

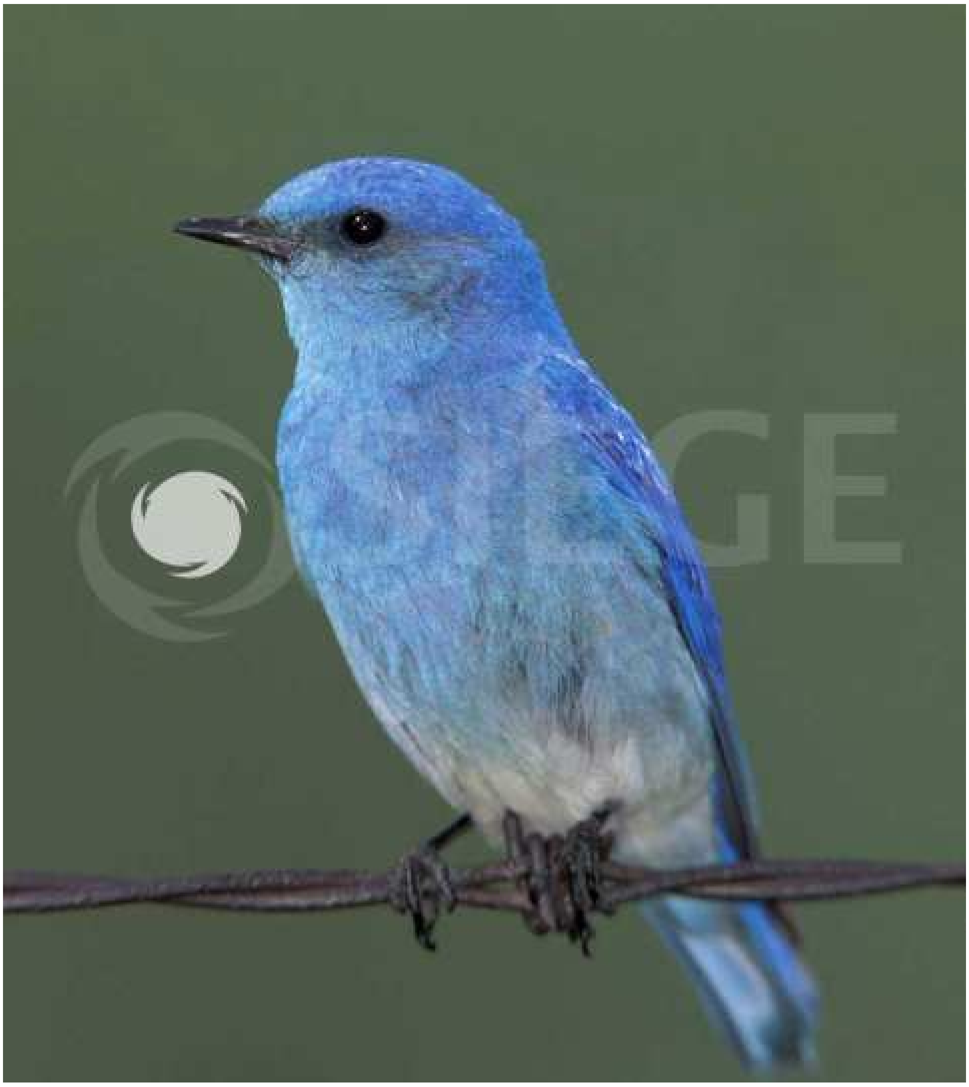


Chirp

50

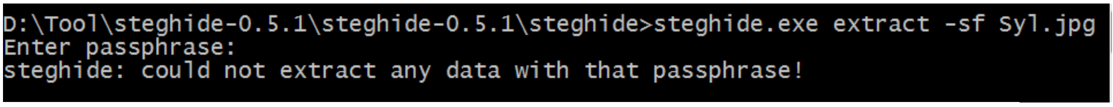
made by Justo and siege

- Didapatkan sebuah gambar



-

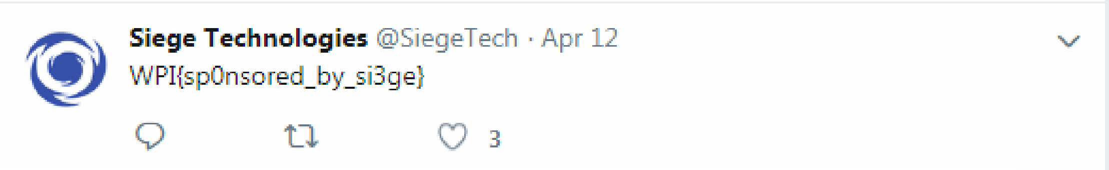
- Asumsi pertama ini adalah sebuah stegano, lalu dicoba menggunakan Steghide dan ternyata ada sebuah password



```
D:\Tool\steghide-0.5.1\steghide-0.5.1\steghide>steghide.exe extract -sf Syl.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

-

- Lalu saya masukkin beberapa pass dengan nama nama bluebird lah , berbau dengan event tersebutlah dan hasilnya nihil
- Dan ku tersadar bahwa ini adalah sebuah recon
- Lalu ku asumsikan lagi bahwa ini adalah sebuah bluebird alias twitter lalu ku buka twitter sponsor acara @SiegeTech

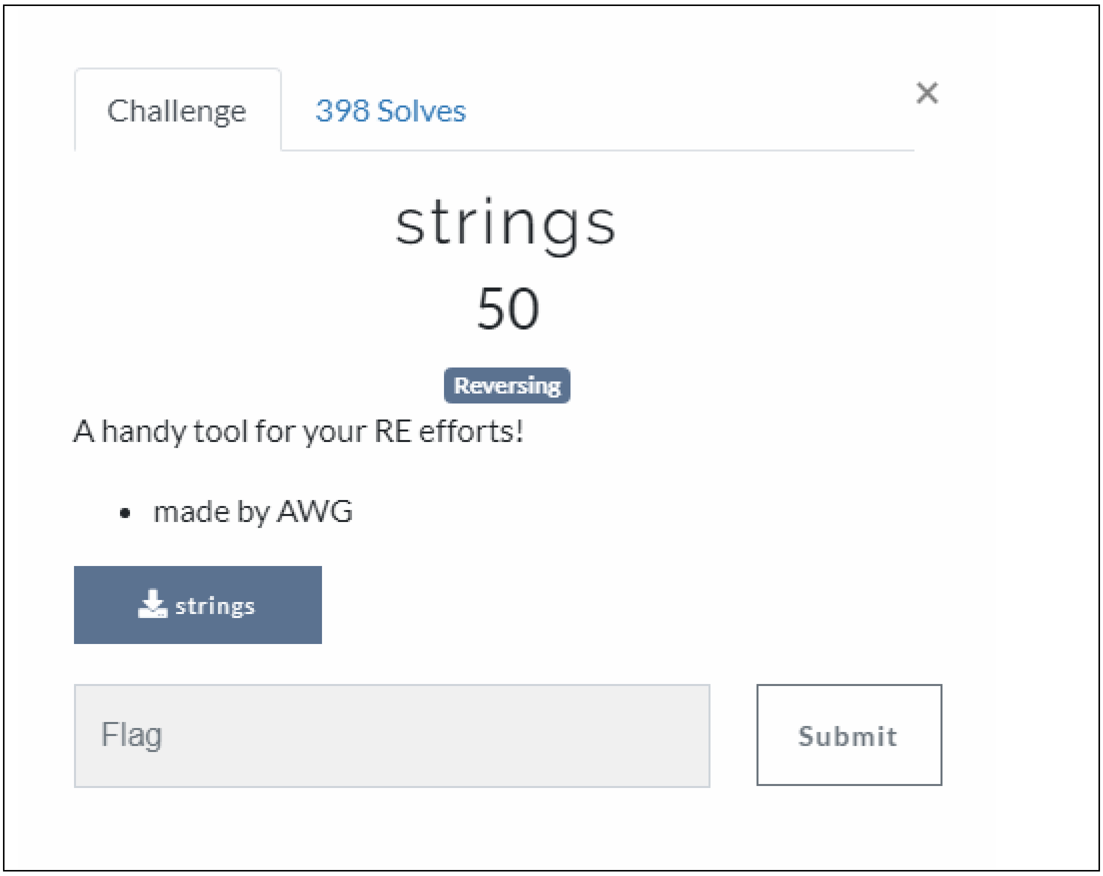


Siege Technologies @SiegeTech · Apr 12

WPI{sp0nsored_by_si3ge}

- Dengan senang hatinya itu ku masukkan ke pass steghide gambar tersebut dan nothing juga
- Lalu dicoba disolve ternyata itu flagnya ☹
- Dan didapatkan flag : **WPI{sp0nsored_by_si3ge}**

4. Strings – 50 point – Reversing –



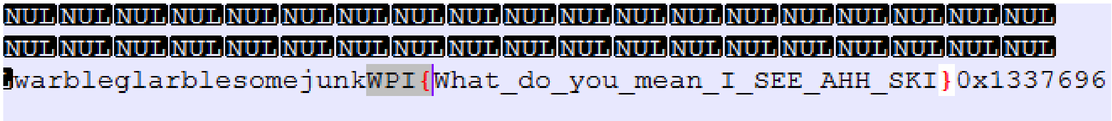
strings

50

A handy tool for your RE efforts!

- made by AWG

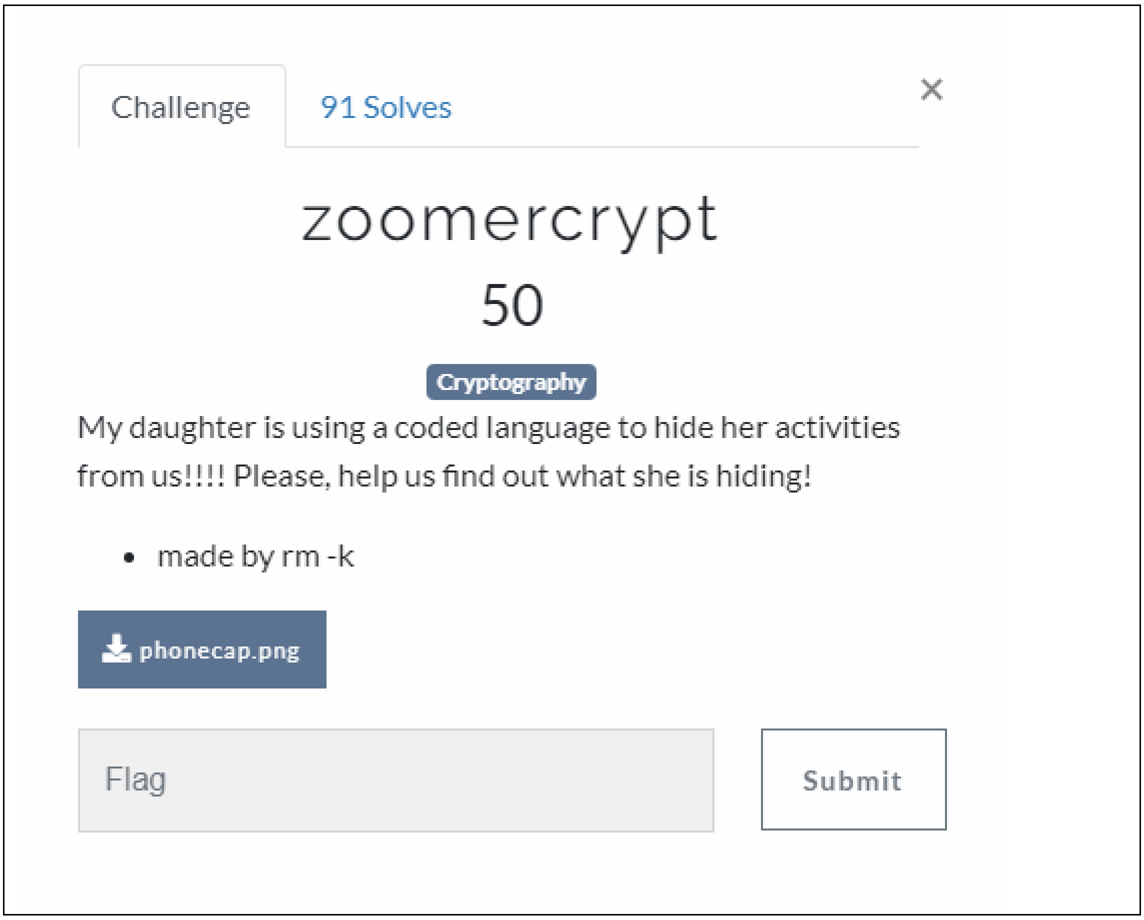
- Didapatkan sebuah file ELF
- Lalu dilihat strings nya



```
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
warbleglarblesomejunkWPI{What_do_you_mean_I_SEE_AHH_SKI}0x1337696
```

- Dan didapatkan flag : **WPI{What_do_you_mean_I_SEE_AHH_SKI}**

5. zoomercrypt – 50 point – Cryptography –



zoomercrypt

50

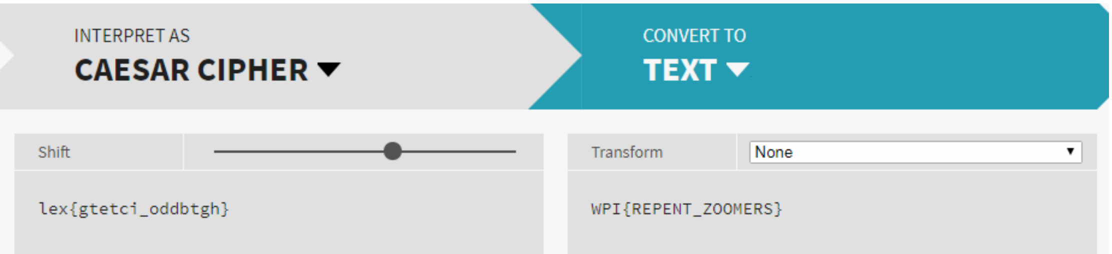
Cryptography

My daughter is using a coded language to hide her activities from us!!!! Please, help us find out what she is hiding!

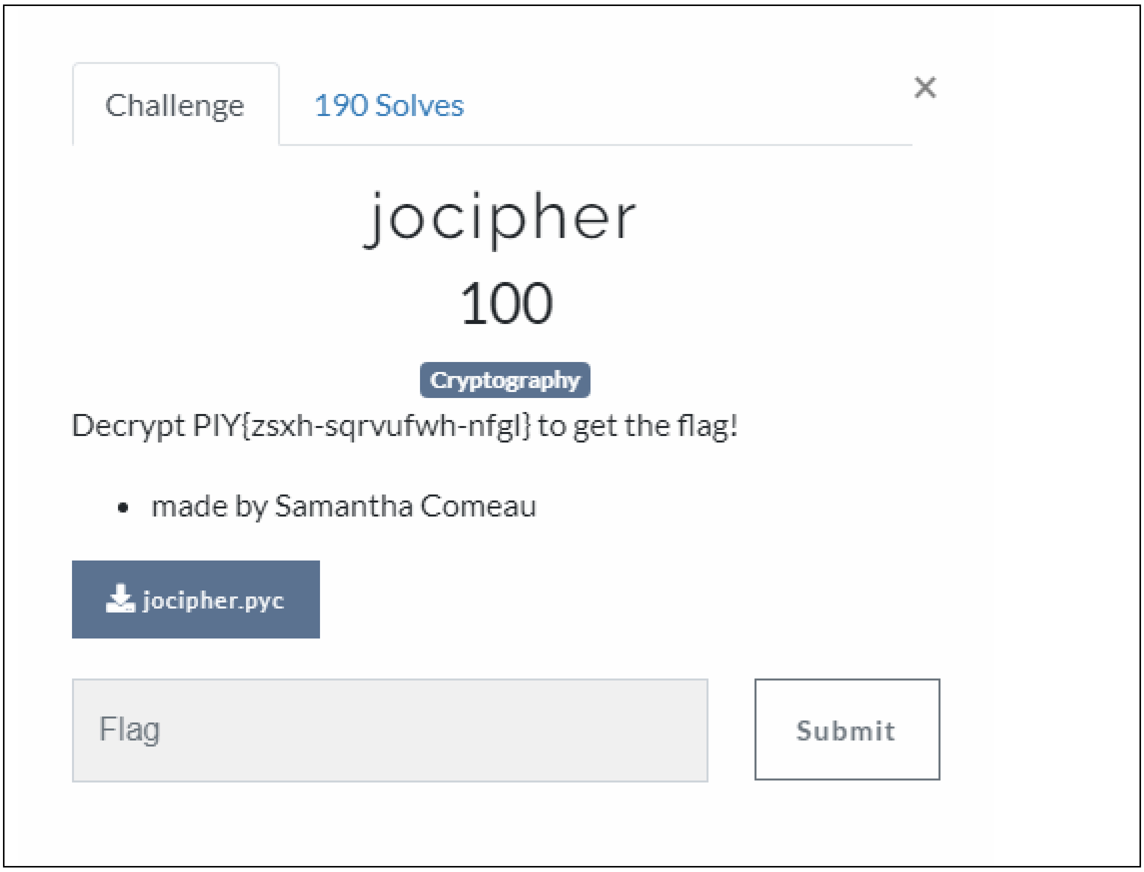
- made by rm -k

- Didapatkan sebuah gambar emoji
- Lalu emoji diubah menjadi Unicode
- 1F60B 1F604 1F617 {1F606 1F613 1F604 1F613 1F602 1F608_1F60E 1F603 1F603 1F601 1F613 1F606 1F607}
- Lalu lihat Dictionary Emoji berikut :

```
{
    'a': 'ðŸ˜€',  # value 1F600
    'b': 'ðŸ˜ ',  # value 1F601
    'c': 'ðŸ˜‚',  # value 1F602
    'd': 'ðŸ˜ƒ',  # value 1F603
    'e': 'ðŸ˜„',  # value 1F604
    'f': 'ðŸ˜…',  # value 1F605
    'g': 'ðŸ˜†',  # value 1F606
    'h': 'ðŸ˜‡',  # value 1F607
    'i': 'ðŸ˜ˆ',  # value 1F608
    'j': 'ðŸ˜‰',  # value 1F609
    'k': 'ðŸ˜Š',  # value 1F60A
    'l': 'ðŸ˜‹',  # value 1F60B
    'm': 'ðŸ˜Œ',  # value 1F60C
    'n': 'ðŸ˜ ',  # value 1F60D
    'o': 'ðŸ˜Ž',  # value 1F60E
    'p': 'ðŸ˜ ',  # value 1F60F
    'q': 'ðŸ˜ ',  # value 1F610
    'r': 'ðŸ˜‘',  # value 1F611
    's': 'ðŸ˜’',  # value 1F612
    't': 'ðŸ˜“',  # value 1F613
    'u': 'ðŸ˜”',  # value 1F614
    'v': 'ðŸ˜•',  # value 1F615
    'w': 'ðŸ˜–',  # value 1F616
    'x': 'ðŸ˜—',  # value 1F617
    'y': 'ðŸ˜˜',  # value 1F618
    'z': 'ðŸ˜™'   # value 1F619
}
```

- Didapatkan lex{gtetci_oddbtgh}
- Lalu di Caesar cipher shift 15 atau Rot 15
- 
  INTERPRET AS CAESAR CIPHER
  CONVERT TO TEXT
  Shift
  Transform None
  lex{gtetci_oddbtgh}
  WPI{REPENT_ZOOMERS}
- Dan didapatkan flag : **WPI{REPENT_ZOOMERS}**

6. Jocipher – 100 point – Cryptography –



jocipher

100

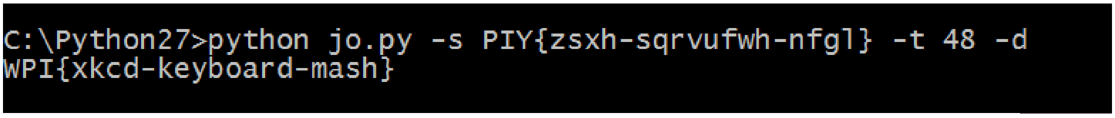
Decrypt PIY{zsxh-sqrvufwh-nfgl} to get the flag!

- made by Samantha Comeau

- Didapatkan sebuah file pyc
- Lalu di decompile
- Lalu running script berikut ini :

```
import argparse, re
num = ''
first = ''
second = ''
third = ''

def setup():
    global first
    global num
    global second
    global third
    num += '1'
    num += '2'
    num += '3'
    num += '4'
    num += '5'
    num += '6'
    num += '7'
    num += '8'
    num += '9'
    num += '0'
    first += 'q'
    first += 'w'
    first += 'e'
    first += 'r'
    first += 't'
    first += 'y'
    first += 'u'
    first += 'i'
    first += 'o'
    first += 'p'
    second += 'a'
    second += 's'
    second += 'd'
    second += 'f'
    second += 'g'
    second += 'h'
    second += 'j'
    second += 'k'
    second += 'l'
    third += 'z'
    third += 'x'
    third += 'c'
    third += 'v'
    third += 'b'
    third += 'n'
    third += 'm'


def encode(string, shift):
    result = ''
    for i in range(len(string)):
        char = string.lower()[i]
        if char in num:
            new_char = num[(num.index(char) + shift) %
len(num)]
            result += new_char
        elif char in first:
            new_char = first[(first.index(char) +
shift) % len(first)]
            if string[i].isupper():
                result += new_char.upper()
            else:
                result += new_char
        elif char in second:
            new_char = second[(second.index(char) +
shift) % len(second)]
            if string[i].isupper():
                result += new_char.upper()
            else:
                result += new_char
        elif char in third:
            new_char = third[(third.index(char) +
shift) % len(third)]
            if string[i].isupper():
                result += new_char.upper()
            else:
                result += new_char
        else:
            result += char

    print result
    return 0


def decode(string, shift):
    result = ''
    shift = -1 * shift
    for i in range(len(string)):
        char = string.lower()[i]
        if char in num:
            new_char = num[(num.index(char) + shift) %
len(num)]
            result += new_char
        elif char in first:
            new_char = first[(first.index(char) +
shift) % len(first)]
            if string[i].isupper():
                result += new_char.upper()
            else:
                result += new_char
        elif char in second:
            new_char = second[(second.index(char) +
shift) % len(second)]
            if string[i].isupper():
                result += new_char.upper()
            else:
                result += new_char
        elif char in third:
            new_char = third[(third.index(char) +
shift) % len(third)]
            if string[i].isupper():
                result += new_char.upper()
            else:
                result += new_char
        else:
            result += char

    print result
    return 0


def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--string', '-s', type=str,
required=True, help='the string to encode or decode')
    parser.add_argument('--shift', '-t', type=int,
required=True, help='the shift value to use')
    parser.add_argument('--encode', '-e',
required=False, action='store_true', help='encode the
string')
    parser.add_argument('--decode', '-d',
required=False, action='store_true', help='decode the
string')
    args = parser.parse_args()
    setup()
    p = re.compile('[a-zA-Z0-9\\-{}]')
    if p.match(args.string) is not None:
        if args.encode:
            ret = encode(args.string, args.shift)
        else:
            if args.decode:
                ret = decode(args.string, args.shift)
        if ret is not 0:
            print 'Sorry, this cipher only uses the [a-
zA-Z0-9\\-{}]'
    else:
        print 'Sorry, this cipher only uses the [a-zA-
Z0-9\\-{}]'
    return


if __name__ == '__main__':
    main()
```

- Masukkan cipher dan shiftnya



```
C:\Python27>python jo.py -s PIY{zsxh-sqrvufwh-nfgl} -t 48 -d
WPI{xkcd-keyboard-mash}
```

-
- Dan didapatkan flag : **WPI{xkcd-keyboard-mash}**