

Writeup IT RACE CTF lvl 3

Ditulis oleh petani_hacker(s)



Pwning The World

Daftar Isi

| | |
|--------------------------------|-----------|
| Kategori Forensik | 3 |
| Lorem is not Ipsum (13 poin) | 3 |
| Binary Typ0 (45 poin) | 4 |
| Kategori Web | 6 |
| AJAX XAJA (25 poin) | 6 |
| Compare Us (40 poin) | 6 |
| Not Heart Bleed (43 poin) | 8 |
| Square Them (60 poin) | 10 |
| Kategori Programming | 13 |
| Pixel Racist (70 poin) | 13 |
| Scazzy (88 poin) | 15 |
| Kategori Reversing | 19 |
| Brute Self (35 poin) | 19 |
| module_2 (90 poin) | 22 |
| Kategori Recon | 29 |
| Ping Me (50.1 poin) | 29 |
| Copy Pasta (100.2 poin) | 29 |
| Kategori Crypto | 31 |
| As Beautiful As Ruby (55 poin) | 31 |
| Yarpchiever (78 poin) | 34 |
| Kategori Misc | 39 |
| Print The Flag (21 poin) | 39 |

Kategori Forensik

Lorem is not Ipsum (13 poin)

Soal:

<http://task-00000100.itrace.systems/loremisnotipsum.tar.gz>

Hint:

-

Solusi:

Mencari pattern yang digunakan untuk menulis flag dengan simbol “{“, “_“, “}”,

```
+ lorem strings * | grep -r "{"  
48.loremipsum>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vel magna metus. Fusce blandit, lorem  
m a aliquet condimentum, magna ligula molestie nibh, sodales semper quam neque eget nisi. Sed non faucibus libero  
, sit amet aliquam eros. Duis imperdiet nunc auctor neque auctor facilisis. Praesent lacinia libero mattis, posue  
re risus et, feugiat magna. Fusce pretium { ornare. Suspendisse porttitor vestibulum semper. Mauris non euismod n  
isi. Suspendisse ultrices orci a porta finibus.  
+ lorem strings * | grep -r "_"  
54.loremipsum:Sed ac orci dolor. Quisque non dui et semper condimentum. Sed vel urna dictum, fringilla tellus  
sit amet, dapibus metus. Aenean ligula leo, porttitor nec vestibulum tristique, auctor id justo. Mauris porttito  
r turpis enim. Vivamus arcu felis, dictum _ sagittis in, ornare eget ante. Vivamus pulvinar accumsan dolor, nec e  
leifend dui euismod eu. Etiam vitae aliquet lorem. Pellentesque sit amet turpis ex. Donec eu mauris justo. Sed ve  
l enim ut metus sodales dignissim. Sed tincidunt quis massa id gravida. Donec vestibulum efficitur nisi.  
+ lorem strings * | grep -r "}"  
60.loremipsum:Sed ac orci dolor. Quisque non dui et semper condimentum. Sed vel urna dictum, fringilla tellus  
sit amet, dapibus metus. Aenean ligula leo, porttitor nec vestibulum tristique, auctor id justo. Mauris porttito  
r turpis enim. Vivamus arcu felis, dictum vitae sagittis in, ornare eget ante. Vivamus pulvinar } dolor, nec elei  
fend dui euismod eu. Etiam vitae aliquet lorem. Pellentesque sit amet turpis ex. Donec eu mauris justo. Sed vel e  
nim ut metus sodales dignissim. Sed tincidunt quis massa id gravida. Donec vestibulum efficitur nisi.
```

Dapat dilihat flag berada file lorem ipsum no 48 - 60. Lalu dicari perbedaan setiap file menggunakan website diffchecker.com, lalu bandingkan file no 1 dengan no 48-60. Contoh file no 1 dibandingkan dengan no 48, lihat di teks yang berwarna hijau ada simbol “{”

| | |
|--|--|
| 1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vel magna metus. Fusce blandit, lorem a aliquet condimentum, magna ligula molestie nibh, sodales semper quam neque eget nisi. Sed non faucibus libero, sit amet aliquam eros. Duis imperdiet nunc auctor neque auctor facilisis. Praesent lacinia libero mattis, posuere risus et, feugiat magna. Fusce pretium maximus ornare. Suspendisse porttitor vestibulum semper. Mauris non euismod nisi. Suspendisse ultrices orci a porta finibus. | 1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vel magna metus. Fusce blandit, lorem a aliquet condimentum, magna ligula molestie nibh, sodales semper quam neque eget nisi. Sed non faucibus libero, sit amet aliquam eros. Duis imperdiet nunc auctor neque auctor facilisis. Praesent lacinia libero mattis, posuere risus et, feugiat magna. Fusce pretium { ornare. Suspendisse porttitor vestibulum semper. Mauris non euismod nisi. Suspendisse ultrices orci a porta finibus. |
|--|--|

File no 1 dengan no 49, ada kata “si”

| | |
|--|---|
| 9. Sed ac orci dolor. si non dui et semper condimentum. Sed vel urna dictum, fringilla tellus sit amet, dapibus metus. Aenean ligula leo, porttitor nec vestibulum tristique, auctor id justo. Mauris porttitor turpis enim. Vivamus arcu felis, dictum vitae sagittis in, ornare eget ante. Vivamus pulvinar accumsan dolor, nec eleifend dui euismod eu. Etiam vitae aliquet lorem. Pellentesque sit amet turpis ex. Donec eu mauris justo. Sed vel enim ut metus sodales dignissim. Sed tincidunt quis massa id gravida. Donec vestibulum efficitur nisi. | 9. Sed ac orci dolor. Quisque non dui et semper condimentum. Sed vel urna dictum, fringilla tellus sit amet, dapibus metus. Aenean ligula leo, porttitor nec vestibulum tristique, auctor id justo. Mauris porttitor turpis enim. Vivamus arcu felis, dictum vitae sagittis in, ornare eget ante. Vivamus pulvinar accumsan dolor, nec eleifend dui euismod eu. Etiam vitae aliquet lorem. Pellentesque sit amet turpis ex. Donec eu mauris si Sed vel enim ut metus sodales dignissim. Sed tincidunt quis massa id gravida. Donec vestibulum efficitur nisi. |
|--|---|

File no 1 dengan no 50, ada kata “mi”

| | |
|--|---|
| 1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vel magna metus. Fusce blandit, lorem a aliquet condimentum, magna ligula molestie nibh, sodales semper quam neque eget nisi. Sed non faucibus libero, sit amet aliquam eros. Duis imperdiet nunc auctor neque auctor facilisis. Praesent lacinia libero mattis, posuere risus et, feugiat magna. Fusce pretium maximus ornare. Suspendisse porttitor vestibulum semper. Mauris non euismod nisi. Suspendisse ultrices orci a porta finibus. | 1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla vel magna metus. Fusce blandit, lorem a aliquet condimentum, magna ligula molestie nibh, sodales semper quam neque eget nisi. Sed non faucibus libero, sit amet aliquam eros. Duis imperdiet nunc auctor neque auctor facilisis. Praesent lacinia libero mattis, posuere risus et, feugiat magna. Fusce pretium maximus ornare. mi porttitor vestibulum semper. Mauris non euismod nisi. Suspendisse ultrices orci a porta finibus. |
|--|---|

Lalu lakukan sampai no 60, dan didapatkan flagnya,

Flag : ITRACE{similiriti_similikiti}

Binary Typ0 (45 poin)

Soal:

<http://task-00000101.itrace.systems/raws.pcapng>

Hint:

-

Solusi:

Buka file raws.pcapng menggunakan wireshark, lalu follow TCP stream,
Pada bagian filter ubah menjadi "tcp.stream eq 6" lalu didapatkan source di bawah ini:

```
GET /t HTTP/1.1
Host: task-00000101.itrace.systems
User-Agent: curl/7.47.0
Accept: */*

HTTP/1.1 200 OK
Date: Sun, 09 Oct 2016 14:02:14 GMT
Server: Apache
Upgrade: h2
Connection: Upgrade
Last-Modified: Sun, 09 Oct 2016 13:35:57 GMT
ETag: "5982d40-a11b-53e6eb9283528"
Accept-Ranges: bytes
Content-Length: 41243
Cache-Control: max-age=172800
Expires: Tue, 11 Oct 2016 14:02:14 GMT
Strict-Transport-Security: max-age=31536000

ff d8 ff e0 0 10 fg hi jk lm 0 1 1 1 0 48 0 48 0 0 ff fe 0 3a 46 6c
61 67 20 69 73 3a 20 49 54 52 41 43 45 7b 74 68 31 35 5f 69 35 5f
68 34 68 34 68 34 5f 6a 75 73 74 6b 69 64 64 69 6e 67 20 2e 20 6e
6f 74 20 74 68 69 73 20 3a 70 ff db 0 43 0 3 2 2 3 2 2 3 3 3 3 4 3
3 .....
```

Setelah dianalisis hex **ff d8 ff e0 0 10** merupakan header file gambar dengan format JPEG, dan **fg hi jk lm** merupakan hex yang typo, lalu ganti hex typo tersebut dengan byte yang seharusnya **"4a 46 49 46"** (JFIF) dan gabungkan semua source hex dari tcp.stream eq 6 sampai tcp.stream eq 13 didapatkan gambar di bawah ini:



Didapatkan flagnya

Flag : ITRACE{FROM.0-F}

Kategori Web

AJAX XAJA (25 poin)

Soal:

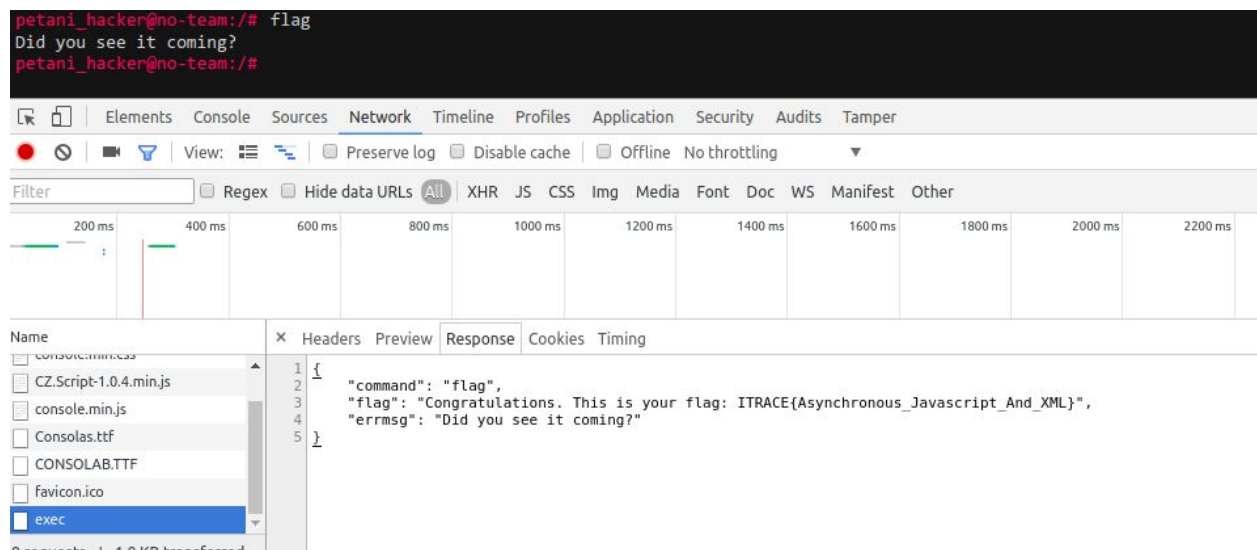
command: flag

Hint:

-

Solusi:

Ketik 'flag' di console platform CTF lalu lihat tab **network** di **developer tools**. Lihat respon dari request Ajax. Flag memang tidak ditampilkan di web, tapi data response Ajax-nya mengandung flag.



Dan didapatkan flagnya.

Flag: ITRACE{Asynchronous_Javascript_And_XML}

Compare Us (40 poin)

Soal:

<http://task-00000001.itrace.systems/compare-us.php>

Hint:

-

Solusi:

```
<?php
error_reporting(0);
include 'flag.php';
$check=thepassword();
parse_str($_SERVER['QUERY_STRING']);
$A=$_GET['key'];
if ctype_xdigit($A){
    $e=implode('',array_map(function($i,$A){return
chr(hexdec($A{$i+$i}).$A{$i+($i+1)}));},list($m,$n,$o)=range(0,2),array($A,$A,$A)));
    if($e<1 && $e>0 && $e!==0){
        if((int)(substr($A,strlen($e)*2)+0) < -1){
            if($check==$_GET['password']){
                echo flag();
            } else {
                echo 'Bad.';
            }
        } else {
            echo 'Bad.';
        }
    } else {
        echo 'Bad.';
    }
} else {
    echo 'Bad.';
}
echo "<pre>";
echo htmlentities(highlight_string(file_get_contents(__FILE__)));
echo "</pre>";
```

Di web soal diberikan source codenya. Input parameter “key” yang akan jadi variabel \$A harus memenuhi beberapa syarat. Syarat pertama `ctype_xdigit($A)` yang artinya input harus merupakan digit hex (0-9 dan a-f). Syarat kedua, ada perhitungan variabel \$e =

```
implode('',array_map(function($i,$A){return
chr(hexdec($A{$i+$i}).$A{$i+($i+1)}));},list($m,$n,$o)=range(0,2),array($A,$A,$A)));
```

Bisa dicek itu ngapain, dengan membuat script sederhana:

```
<?php

$A = "4142434445";
$e = implode('',array_map(function($i,$A){return
chr(hexdec($A{$i+$i}).$A{$i+($i+1)}));},list($m,$n,$o)=range(0,2),array($A,$A,$A)));
echo $e;
```

Dan outputnya adalah “ABC”. Berarti dia melakukan decode dari hex namun hasilnya hanya sepanjang 3 karakter. Selanjutnya `$e<1 && $e>0 && $e!==0`, berarti input kalau di-decode harus berupa angka di antara 0 sampai 1, misalnya 0.5 (decode hexa = **302e35**).

Selanjutnya `(int)(substr($A,strlen($e)*2)+0) < -1`, kita tahu `strlen($e)` pasti bernilai 3, sehingga itu sama saja `(int)(substr($A,6)+0) < -1`. Nilai \$A sejauh ini adalah **302e35** (6 karakter), kalau di-substring 6 berarti dia mencari string lanjutannya, yang nanti dijumlahkan dengan nol (PHP otomatis mengubah tipenya jadi angka) dan di-cast jadi integer. Coba-coba:

3. Berhubungan dengan UTF-8

Sesuai nama soalnya, sepertinya berhubungan dengan heart bleed. Target dicoba dengan mengirimkan request 10 karakter.

```
$ curl -s -X POST --data "request=`python -c "print 'A'*10`"
'http://task-00001110.itrace.systems/request.php'


```
{
 "method": "POST",
 "request": "AAAAAAAAAA",
 "Accept-Charset": "UTF-8",
 "Content-Length": 64,
 "Respond-Text": "AAAAAAAAA....."
}%
```


```

Respon dari server tetap 10 karakter. Akan dicoba lagi mengirim 64 karakter.

```
$ curl -s -X POST --data "request=`python -c "print 'A'*64`"
'http://task-00001110.itrace.systems/request.php'


```
{
 "method": "POST",
 "request": "AA",
 "Accept-Charset": "UTF-8",
 "Content-Length": 64,
 "Respond-Text": "AA"
}%
```


```

Jika mengirim lebih dari 64 maka akan mengirim pesan error "Data is too long"

```
$ curl -s -X POST --data "request=`python -c "print 'A'*65`"
'http://task-00001110.itrace.systems/request.php'
Data is too long%
```

Akhirnya dicoba mengirimkan karakter emoji 😊 ke server target.

```
$ curl -s -X POST --data "request=`python -c "print '😊'*64`"
'http://task-00001110.itrace.systems/request.php'
Data is too long%
```

Hmm... data too long, artinya request yang dikirimkan lebih dari 64 byte. Setelah lihat list emoji dari konversinya ke UTF-8¹ ternyata emoji karakter tersebut setara 4 byte. Sehingga, agar request yang dikirim pas 64 byte kita harus mengirim 64 / 4 yaitu 16 karakter emoji.

```
$ curl -s -X POST --data "request=`python -c "print '😊'*16`"
'http://task-00001110.itrace.systems/request.php'
```

¹ <http://apps.timwhitlock.info/emoji/tables/unicode>

[illegible]

Dan didapatkan sebuah respon kumpulan heksadesimal, agar proses decoding lebih mudah kami buat kode inline bash sederhana untuk menyelesaikan soal tersebut.

```
$ curl -s -X POST --data "request=`python -c "print '🤪'*16`"
'http://task-00001110.itrace.systems/request.php' | grep -Eo "0x[a-f0-9]{2}" | xxd -p -r
LOOKING FOR A FLAG? THEN THIS IS FOR YOU: ITRACE{Y4Y_TO_345Y}%
```

Flag: ITRACE{Y4Y_TO_345Y}

Square Them (60 poin)

Soal:

<http://task-00000010.itrace.systems/square.php>

Hint:

—

Solusi:

Terdapat sebuah soal programming dimana kita diminta generate 9 angka yang jumlah kolom dan barisnya adalah N. Dimana N adalah nomer yang dikasih dari soal.

More Example:

| | | |
|---|----|---|
| 3 | 10 | 5 |
| 8 | 6 | 4 |
| 7 | 2 | 9 |

Sum each Rows: 18
Sum each Cols: 18
Numbers (From Left to Right, First row to the last) : 3,10,5,8,6,4,7,2,9
POST numbers=3,10,5,8,6,4,7,2,9

Dibuat sebuah solver-nya menggunakan library Z3 Constraint Solver²

```
magicsq.py

#!/usr/bin/python

from z3 import *
import sys

def solve_magic(sum):
    c = sum / 3
    a = Int('a')
    b = Int('b')

    s = Solver()
    s.add(0 < a)
    s.add(a < b)
    s.add(b < c - a)
    s.add(b != 2 * a)

    if s.check() == sat:
        m = s.model()
        a = int(str(m[a]))
        b = int(str(m[b]))
        square = [
            c - b,          c + (a + b),          c - a,
            c - (a - b),    c,                    c + (a - b),
            c + a,          c - (a + b),          c + b
        ]
        assert square[0] + square[1] + square[2] == sum
        assert square[0] + square[3] + square[6] == sum
        assert square[0] + square[4] + square[8] == sum
        return square

print 'numbers='+','.join(str(x) for x in solve_magic(int(sys.argv[1])))
```

Program tersebut menerima argumen sebuah angka yang diberikan dari soal, misal 18

```
$ python magicsq.py 18
numbers=3,10,5,8,6,4,7,2,9
```

Selanjutnya membuat script otomatis untuk submit ke soal.

```
square.sh

#!/bin/bash

SESSION='PHPSESSID=o9mqvjfv376h986in5edkn7u84'
NUM=$(curl -s -X POST --cookie $SESSION 'http://task-00000010.itrace.systems/square.php' |
tail -n 5 | head -n 1 | grep -Eo '[0-9]' | tr -d '\n')
DATA=$(python magicsq.py $NUM)

while true; do
```

² <https://github.com/Z3Prover/z3>

```

RSP=$(curl -s -X POST --data $DATA --cookie $SESSION
'http://task-00000010.itrace.systems/square.php')
SOLVD=$(echo $RSP | grep -Eo 'ITRACE{.+}')
echo $RSP
if [[ $SOLVD ]]; then
    break
fi
NUM=$(echo $RSP | grep -Eo '[0-9]' | tr -d '\n')
DATA=$(python magicsq.py $NUM)
done

```

Lalu jalankan

```

$ ./square.sh
{"errmsg":"Good job","nextsum":261}
{"errmsg":"Good job","nextsum":222}
{"errmsg":"Good job","nextsum":273}
{"errmsg":"Good job","nextsum":291}
{"errmsg":"Good job","nextsum":222}
{"errmsg":"Good job","nextsum":288}
{"errmsg":"Good job","nextsum":210}
{"errmsg":"Good job","nextsum":246}
{"errmsg":"Good job","nextsum":351}
{"errmsg":"Good job","nextsum":252}
Congratulation. This is your Flag: ITRACE{m4g1c_squ4r3_is_s0_m4th}

```

Flag: ITRACE{m4g1c_squ4r3_is_s0_m4th}

Kategori Programming

Pixel Racist (70 poin)

Soal:

<http://task-00001001.itrace.systems/racist.php>

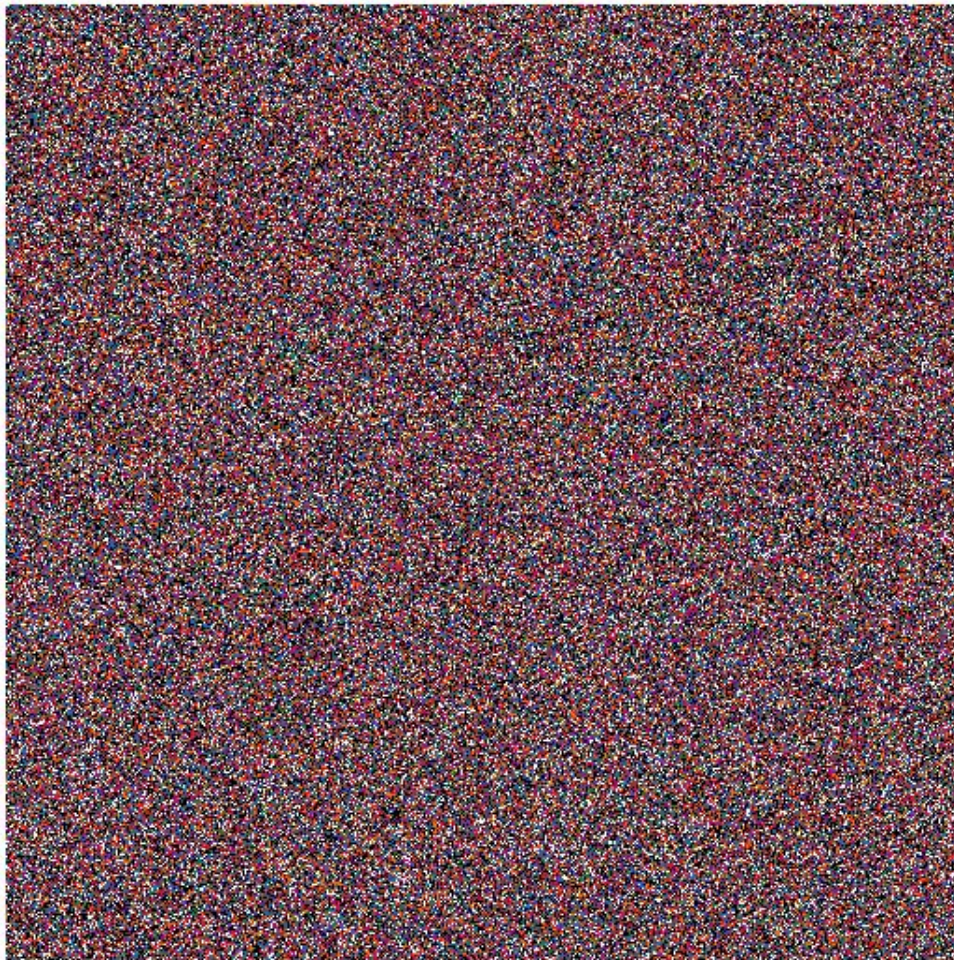
Hint:

-

Solusi:

Diberikan sebuah soal programming dimana kita harus mencari sebuah warna yang kemunculannya paling sedikit.

Find the color with the least amount. Ex. post: "color=fff000"



Submit dengan mengirim Post Request dengan parameter "color=(jawaban)".

Buatlah sebuah solvernya dengan python

```
ppc.py

#!/usr/bin/python

import collections
from PIL import Image

counter = collections.defaultdict(int)
im = Image.open("target.png")
pix = im.load()
w, h = im.size
tupl = []

for r in range(w):
    for c in range(h):
        tupl.append("".join([hex(x)[2:4].zfill(2) for x in pix[r,c]]))

for word in tupl:
    counter[word] += 1

print "color="+sorted(counter.iteritems(), key=lambda x: x[::-1])[0][0]
```

Buat sebuah script auto-submit dengan bash, sebelumnya ambil cookie terlebih dahulu dari browser.

```
ppc.sh

#!/bin/bash

i=0
SESSION="PHPSESSID=qedntnb92f1l6g3klko17855a3"

while true; do

# counter
echo $i; i=$((i+1))

# download image
curl 'http://task-00001001.itrace.systems/image.php' --cookie $SESSION -s > target.png

# get least amount color
DATA=$(python ppc.py)

echo $DATA

curl -s --cookie $SESSION -X POST --data $DATA
'http://task-00001001.itrace.systems/racist.php'

done
```

Didapatkan flagnya pada iterasi ke-3

```
0
color=5d5f7d
<p style="text-align:center">
  <br /><br />
Find the color with the least amount. Ex. post: "color=fff000"Good. Next.<br /><br /></p>
1
color=19315f
<p style="text-align:center">
  <br /><br />
Find the color with the least amount. Ex. post: "color=fff000"Good. Next.<br /><br /></p>
2
color=044d96
<p style="text-align:center">
  <br /><br />
Find the color with the least amount. Ex. post: "color=fff000"Good. Next.<br /><br /></p>
3
color=0bb341
<p style="text-align:center">
  <br /><br />
Find the color with the least amount. Ex. post: "color=fff000"Good. Next.Congratz. This is your flag: ITRACE{y0u_6uy5_4r3_4wes0m3}</p>
4
color=0bb341
<p style="text-align:center">
  <br /><br />
Find the color with the least amount. Ex. post: "color=fff000"Good. Next.<br /><br /></p>
```

Flag: ITRACE{y0u_6uy5_4r3_4wes0m3}

Scazzy (88 poin)

Soal:

Scazzy, bot berjenis kelamin perempuan. Suka menggoda. Tapi kalo dirayu jual mahal.

<https://web.telegram.org/#/im?p=@itrace>

Hint:

-

Solusi:

Sewaktu kita buka web Telegram itu, ada bot “Scazzy” yang menyapa channel @itrace. Kita bisa PM ke bot Scazzy tersebut, dan dia akan menjelaskan instruksinya. Jadi via PM bukan public channel.

Kita bisa membalas dengan “/start”, dan dia akan memberi pertanyaan hitung-hitungan. Kita harus menjawab dengan “/answer <jawaban>”.

| | | |
|----|--|------------|
| SC | Scazzy Please begin with /start | 6:24:50 PM |
| MA | Muhammad /start | 6:25:38 PM |
| SC | Scazzy Reply your answer started with /answer. Ex: /answer 123 | 6:25:42 PM |
| | Solve this: 10001101 x 0xEF - 0x6A in less than 1 minute | 6:25:44 PM |
| MA | Muhammad /answer 33593 | 6:25:44 PM |
| SC | Scazzy Solve this: 0x6E add 0xC0 add 0xA6 add 0xF1 x 0xB2 ÷ 0xC6 in less than 1 minute | 6:25:51 PM |
| MA | Muhammad /answer 684 | 6:25:51 PM |

Kita diharuskan menjawab pertanyaannya dalam waktu kurang dari semenit (di tengah acara diupdate jadi kurang dari 20 detik). Jadi sepertinya kita harus bikin program buat mengotomatisasinya.

Butuh instal:

- Telegram CLI <https://github.com/vysheng/tg>
- Telegram Python client <https://github.com/luckydonald/pytg>

Instal dan baca dokumentasinya.

Buat fungsi untuk solve soal dari Scazzy:

```
# -*- coding: utf-8 -*-

import re

def solve(text):
    text = re.search(r'Solve this: (.*) in less than', text).group(1)
    text = text.replace(u' x ', u' * ')      # ubah x jadi *
    text = text.replace(u' - ', u' - ')      # ubah - jadi -
    text = text.replace(u' add ', u' + ')     # ubah add jadi +
    text = text.replace(u' ÷ ', u' / ')       # ubah ÷ jadi /
    text = re.sub(r'([01]{8})', r'0b\1', text) # cari biner 8 digit & tambahkan 0b didepan
    return eval(text)                        # let python calculate it unsafely
```

Sekaligus script untuk secara otomatis menerima dan mengirim pesan ke bot. Script ini dimodifikasi dari tutorial/dokumentasi library **pytg**:

```
from pytg import Telegram
from pytg.utils import coroutine

from pytg.sender import Sender
from pytg.receiver import Receiver
receiver = Receiver(host="localhost", port=4458)
sender = Sender(host="localhost", port=4458)

sender.send_msg("Scazzy", u'/start')

@coroutine #
def main_loop():
    while 1:
        msg = (yield) # it waits until it got a message, stored now in msg.
        if msg[u'event'] == u'message':
            text = msg[u'text']
            if 'Solve this' in text:
                answer = solve(text)
                print "Question =", text
                print "Answer =", answer
                sender.send_msg("Scazzy", u'/answer ' + str(answer))
            #else:
            #    print text

receiver.start()
receiver.message(main_loop())
```

Di akhir acara didapatkan flagnya (percakapan muncul otomatis di web Telegram waktu script berjalan):

SC

Scazzy

5:28:12 AM

Solve this: $01101111 \div 00100001$ add $10000101 - 0x46 - 0x2D \times 0x19$ add $0xE7 \times 0xFD - 0x6E \times 01011100$ add $0x63 \times 10111101 - 11111110 \div 0x67 \div 11111110 - 0x32 \div 11110001 \div 0xD7$ add 10111001 add $00110100 \times 0x4B$ add $0xAC \times 0x7B \times 0x35 - 0xEA \div 0x92 \times 0xDA - 10011001 \div 11010001 - 01111110 \div 10011011 \times 0x1A - 0x6F \div 0x2A - 11100010 \div 0x1F \times 01010110 \times 0xB3$ add $0x63 \times 01110010 - 01101000$ add $11001101 \div 0xD2 \div 01101100 \div 01011000 \div 0x7D \times 0xBD \times 0x49 \div 00110100 \div 10100100 - 00110111 \div 10001101 - 00010001 - 10010010 - 0x9C - 11101001 \times 0xA5$ add 01001101 add 11011011 add 00011001 in less than 20 seconds.

MA

Muhammad

5:28:12 AM

/answer 1056089

SC

Scazzy

5:28:18 AM

Congratulations. This is your flag:
ITRACE{b0t_tele6r4m_i5_m0d3rn_typ3_of_ircs_b0t}

Flag: ITRACE{b0t_tele6r4m_i5_m0d3rn_typ3_of_ircs_b0t}

Kategori Reversing

Brute Self (35 poin)

Soal:

<http://task-00000011.itrace.systems/password.tar.gz>

Hint:

-

Solusi:

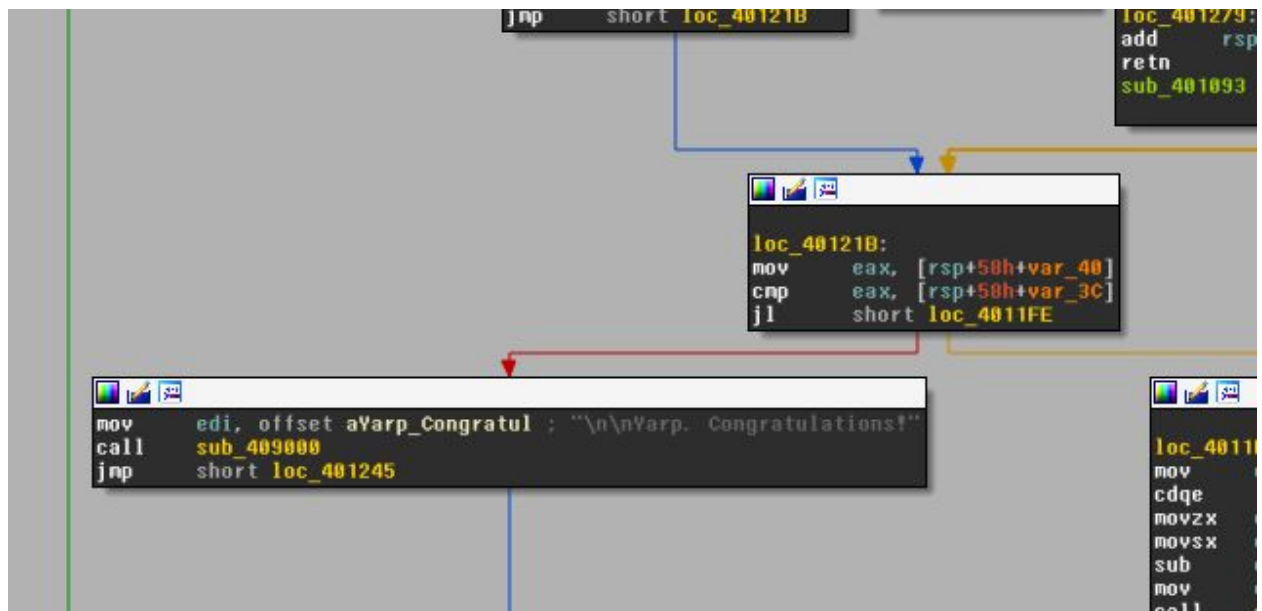
Pertama-tama dicek terlebih dahulu format file soal dengan command 'file'.

```
$ file password
password: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
statically linked, for GNU/Linux 2.6.24,
BuildID[sha1]=636e9ba2b787af74e02a1060fa0e599d0cd8844f, stripped
```

Analisis string dengan menggunakan IDA Pro 64 bit .

| | Address | Length | Type | String |
|--|----------------|----------|------|--|
| | .rodata:000... | 00000008 | C | LD_PRELOAD |
| | .rodata:000... | 00000010 | C | LD_LIBRARY_PATH |
| | .rodata:000... | 00000005 | C | Bad. |
| | .rodata:000... | 00000008 | C | Flag is |
| | .rodata:000... | 00000019 | C | \n\nYarp. Congratulations! |
| | .rodata:000... | 0000000D | C | libc-start.c |
| | .rodata:000... | 00000017 | C | FATAL: kernel too old\n |
| | .rodata:000... | 0000000D | C | /dev/urandom |
| | .rodata:000... | 0000002D | C | __ehdr_start.e_phentsize == sizeof *_dl_phdr |

Kunjungi alamat yang memanggil string tersebut. Lalu lihat flow graph-nya



Setelah mundur-mundur terdapat sebuah array mencurigakan.

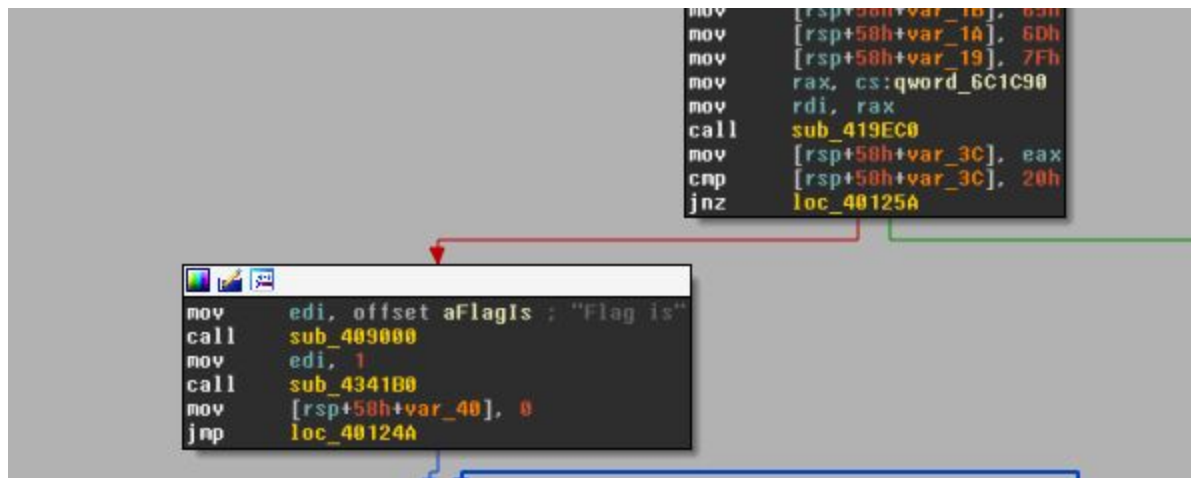
```

loc_4010CA:
mov     rax, [rsp+50h+var_50]
mov     rax, [rax+0]
mov     cs:qword_6C1C90, rax
mov     [rsp+50h+var_38], 40h
mov     [rsp+50h+var_37], 56h
mov     [rsp+50h+var_36], 54h
mov     [rsp+50h+var_35], 43h
mov     [rsp+50h+var_34], 45h
mov     [rsp+50h+var_33], 47h
mov     [rsp+50h+var_32], 7Bh
mov     [rsp+50h+var_31], 64h
mov     [rsp+50h+var_30], 36h
mov     [rsp+50h+var_2F], 66h
mov     [rsp+50h+var_2E], 61h
mov     [rsp+50h+var_2D], 65h
mov     [rsp+50h+var_2C], 32h
mov     [rsp+50h+var_2B], 66h
mov     [rsp+50h+var_2A], 35h
mov     [rsp+50h+var_29], 37h
mov     [rsp+50h+var_28], 61h
mov     [rsp+50h+var_27], 6Ah
mov     [rsp+50h+var_26], 33h
mov     [rsp+50h+var_25], 66h
mov     [rsp+50h+var_24], 33h
mov     [rsp+50h+var_23], 70h
mov     [rsp+50h+var_22], 30h
mov     [rsp+50h+var_21], 61h
mov     [rsp+50h+var_20], 33h
mov     [rsp+50h+var_1F], 50h
mov     [rsp+50h+var_1E], 61h
mov     [rsp+50h+var_1D], 72h
mov     [rsp+50h+var_1C], 36h
mov     [rsp+50h+var_1B], 65h
mov     [rsp+50h+var_1A], 6Dh
mov     [rsp+50h+var_19], 7Fh
mov     rax, cs:qword_6C1C90
mov     rdi, rax
call    sub_419EC0
mov     [rsp+50h+var_3C], eax
cmp     [rsp+50h+var_3C], 20h
jnz     loc_40125A

```

Berikut array-nya

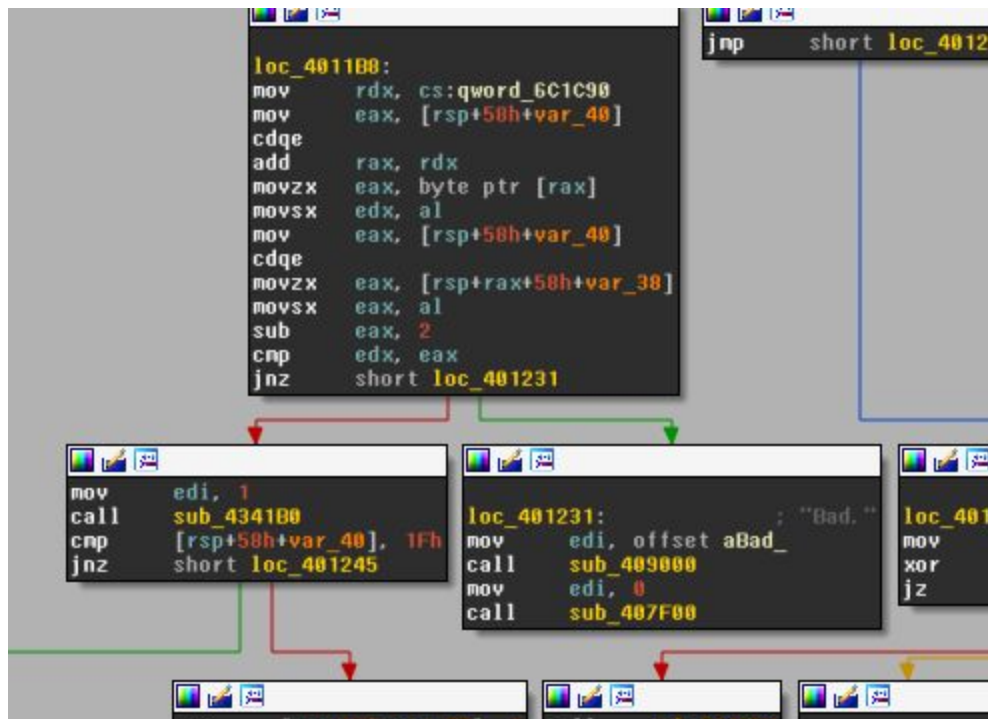
```
0x4B 0x56 0x54 0x43 0x45 0x47 0x7D 0x64 0x36 0x66 0x61 0x65 0x32 0x66  
0x35 0x37 0x61 0x6A 0x33 0x66 0x33 0x70 0x38 0x61 0x33 0x50 0x61 0x72  
0x36 0x65 0x6D 0x7F
```



Perhatikan instruksi `cmp [rsp+58h+var_3C], 20h` seperti dia melakukan perbandingan suatu variable dengan 20h (32) seperti nya variable tersebut adalah jumlah input.

```
$ ./password $(python -c "print 'A'*31")  
Bad.  
$ ./password $(python -c "print 'A'*32")  
Flag is  
Bad.
```

Ternyata program melakukan print 'Flag is' namun masih salah. Lanjut ke instruksi lainnya.



var_38 (array 'mencurigakan' tadi) disimpan di eax, lalu di operasikan 'sub eax 2' dan dibandingkan dengan **edx** jika salah maka masuk ke kondisi 'Bad'. Tanpa berpikir panjang kami langsung buat inline script Python untuk melakukan pengurangan nilai 2 dari array tadi.

```
>>> """.join([chr(int(x, 16)-2) for x in "0x4B 0x56 0x54 0x43 0x45 0x47
0x7D 0x64 0x36 0x66 0x61 0x65 0x32 0x66 0x35 0x37 0x61 0x6A 0x33 0x66
0x33 0x70 0x38 0x61 0x33 0x50 0x61 0x72 0x36 0x65 0x6D 0x7F".split()])

'ITRACE{b4d_c0d35_h1d1n6_1N_p4ck}'
```

Flag: ITRACE{b4d_c0d35_h1d1n6_1N_p4ck}

module_2 (90 poin)

Soal:

Unlock [module_2](#) | [module_2_dependencies_optional.tar.gz](#)

Hint:

-

Solusi:

Decompile APK yang dikasih dengan apktool atau yang online³, cari main activity, ternyata app dibuat dengan Cordova, berarti bentuknya halaman web.

```
package com.cafelabs.itrace;

import android.os.Bundle;
import org.apache.cordova.CordovaActivity;

public class MainActivity extends CordovaActivity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        loadUrl(this.launchUrl);
    }
}
```

Di folder assets/www, ini halaman utama aplikasinya (index.html), ini script di halaman itu:

³ <http://www.javadecompilers.com/apk>


```

<script type="text/javascript" charset="utf-8">
    register_local();
    document.addEventListener('deviceready',
        function() {
            register_gkey();
            getservertime();
            if(localStorage.isstart=="zero"){
                toast("didn't received registration_id from Google yet");
            } else {

            }
            FCMPlugin.onNotification(
                function(data) {
                    if(data.wasTapped) {
                        alert(data.body);
                    } else {
                        var c={id:genrandom(),html:data.title+
                            "<br />" +data.body}
                        appendto(c);
                        alert("Flag received");
                    }
                },
                function(msg) {
                    //register_gkey();
                },
                function(err) {
                    alert('Error registering onNotification callback: ' +
err);
                }
            );
        }, false
    );
</script>

```

Pertama dia register_local, mengeset localStorage “gkey” dan “isstart”.

```

function register_local(){
    if(!localStorage.gkey) localStorage.setItem("gkey","no");
    if(!localStorage.isstart) localStorage.setItem("isstart","zero");
}

```

Kemudian register_gkey, seperti retrieve token FCM (Firebase Cloud Messaging⁴) dan mengeset localStorage “gkey” dari token yang didapat:

```

function register_gkey(){
    FCMPlugin.getToken(
        function(token) {
            if(!localStorage.firstUse) {
                localStorage.setItem("firstUse",1);
            }
            if(token=="null" || token=="") {
                return register_gkey();
            } else {
                localStorage.gkey=token;
            }
            FCMPlugin.subscribeToTopic("contestant");
        }
    );
}

```

⁴ <https://firebase.google.com/docs/cloud-messaging/>

```

    },
    function(err) {
        alert('error retrieving token: ' + err);
    }
);
}

```

Kemudian `getservertime`, dia request ke `_app_host`
<http://cafelinux.info/sideprojects/itrace-lvl-3/time?gkey=GKEY&ver=APPVER>, kemudian
 mengeset `localStorage` “cookie” dari cookie yang didapat, lalu memanggil `syncapp(0)`

```

function getservertime() {
    var a = {id: "a", html: "Initializing..."};
    appendto(a);
    var obj = {target: _app_host + "time?gkey=" + localStorage.gkey + "&ver=" + _app_version};
    CZAjax(obj,
        function(x) {
            var o = JSON.parse(x);
            var a = {id: 0, html: "Server time is: <b>" + o.servertime + "</b><br />"};
            appendto(a);
            localStorage.setItem("cookie", o.cookie);
            syncapp(0);
        }
    );
}

```

Di `syncapp`, dia request ke
<http://cafelinux.info/sideprojects/itrace-lvl-3/sync?gkey=GKEY&ver=APPVER&cookie=COOKIE>.
 COOKIE adalah cookie yang didapat dari `getservertime` tadi. Setelah dapat dia update cookie di
`localStorage` (`localStorage.cookie=o.cookie`). Kalau “module” yang didapat dari request itu
`== 2` maka stop.

```

function syncapp(s) {
    if(s==0) {_r_module=[];}
    var a = {id: genrandom(), html: "module_" + s + " sending SYN at " + now()};
    appendto(a);
    var b = {id: genrandom(), html: "module_" + s + " waiting response..."};
    appendto(b);
    var ctime = new Date().getTime();
    var obj = {
        target: _app_host + "sync?gkey=" + localStorage.gkey + "&ver=" + _app_version + "&cookie=" + localStorage.cookie,
    };
    CZAjax(obj,
        function(x) {
            var o = JSON.parse(x);
            var c = {id: genrandom(), html: o.errmsg};
            appendto(c);
            localStorage.cookie = o.cookie;
            if(o.module == 2) {
                clog('stahp');
            } else {
                if(!pushtomodule(o.module)) {
                    var a = {id: genrandom(), html: "<span style='color:red'>session module_" + (o.module+1) + " is invalid.</span><br />exit."};

```

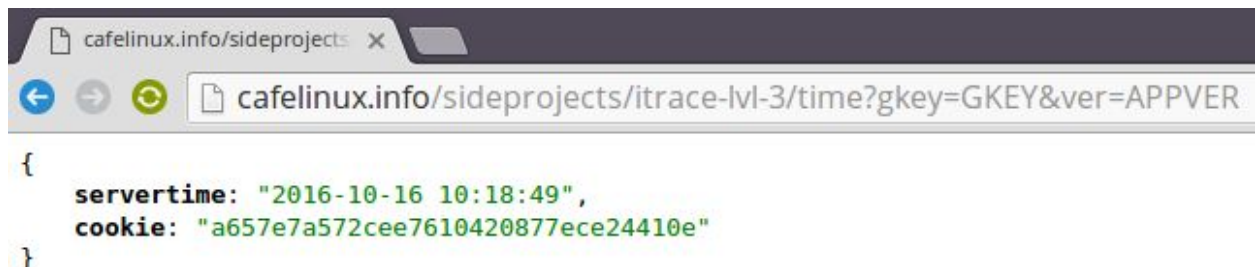
```

        appendto(a);
    } else {
        var a={id:genrandom(),html:"<span
style='color:green'>module_"+o.module+" synced.</span>"};
        appendto(a);
        syncapp(o.module+1);
    }
}
};
}
}

```

Mungkin bisa kita simulasikan request ke /time dan /sync itu lewat browser, dengan “gkey” ngasal saja siapa tahu berhasil. Kalau pakai browser kita nggak dapat gkey karena gkey di-push ke perangkat mobile.

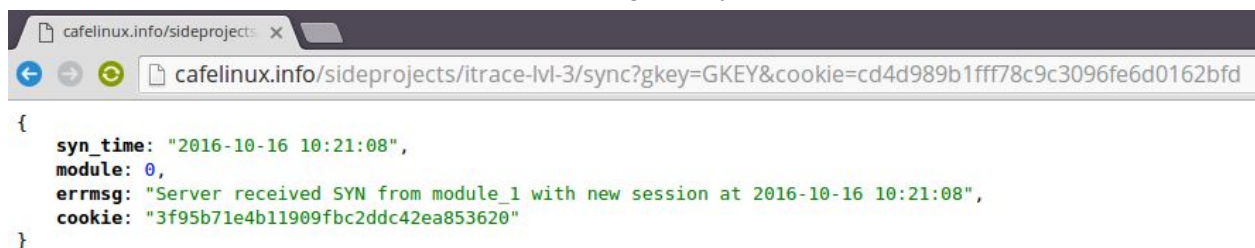
Pertama request ke /time



Dapat cookie, pakai buat request ke /sync seperti di alur programnya



Dapat module 0 dan cookie baru, kita request lagi ke /sync pakai cookie baru




Kok tetap module 0? Harusnya module: 1. Kalau kita ulang terus tetap dapat module 0. Kayaknya ada yang salah. Mungkin cookie-nya harusnya jangan diganti baru? Jadi pakai cookie yang didapat pertama kali saja.

Kita ulang lagi dari awal, request ke /time



```
{
  "servertime": "2016-10-16 10:24:09",
  "cookie": "f42b91d01b8257ac22cce2acd47de31a"
}
```

Dapat cookie, pakai buat request ke /sync



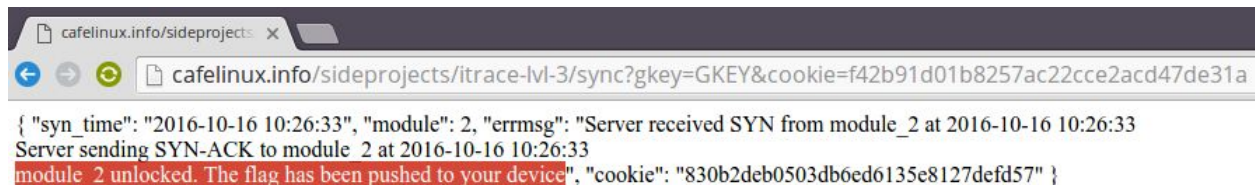
```
{ "syn_time": "2016-10-16 10:25:02", "module": 0, "errmsg": "Server received SYN from module_0 at 2016-10-16 10:25:02\nServer sending SYN-ACK to module_0 at 2016-10-16 10:25:02", "cookie": "521774599cc7f5a8bf16f1d0e396b9cc" }
```

Kita request lagi, tapi kali ini **jangan diganti cookie-nya**. Berarti cukup refresh.



```
{ "syn_time": "2016-10-16 10:25:55", "module": 1, "errmsg": "Server received SYN from module_1 at 2016-10-16 10:25:55\nServer sending SYN-ACK to module_1 at 2016-10-16 10:25:55", "cookie": "2394e3685eba73609441365918465bd3" }
```

Nah sukses dapat module: 1. Refresh lagi untuk dapat module: 2.



```
{ "syn_time": "2016-10-16 10:26:33", "module": 2, "errmsg": "Server received SYN from module_2 at 2016-10-16 10:26:33\nServer sending SYN-ACK to module_2 at 2016-10-16 10:26:33\nmodule_2 unlocked. The flag has been pushed to your device", "cookie": "830b2deb0503db6ed6135e8127defd57" }
```

Dapat module 2, dan flag has been pushed to your device. Tapi karena kita jalanin di browser ya tidak dapat push notificationnya. Berarti sampai sini kita punya dua pilihan:

1. Intercept request dari app ke server pakai Burp untuk mengedit request supaya cookie yang dikirim tetap
2. Edit aplikasinya (kode JS) supaya tidak merubah cookie setiap kali request ke /sync

Kita coba yang kedua, ngedit APK yang dikasih. File APK kan sebenarnya file ZIP, bisa di-extract. Dan karena kodenya JS, bisa langsung diubah. Yang diubah di fungsi syncapp, baris yang mengubah cookie (`localStorage.cookie=o.cookie`) dihapus supaya waktu request ke /sync selanjutnya, cookienya tetap.

Setelah diubah, lalu signing. Ingat file APK kalau diubah, supaya bisa jalan harus di-sign. Untungnya di soal dikasih informasi signing-nya (module_2_dependencies_optional.tar.gz).

```
itrace.config:  
-- if needed for re-signing --
```

```
Pass:1tr4c3lvI3  
alias:itrace-key
```

Kita sign APK yang sudah diubah, dia minta password yang juga sudah diberikan di file itrace.config:

```
$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore  
itrace.keystore module_2.apk itrace-key  
  
Enter Passphrase for keystore: 1tr4c3lvI3
```

Selesai, kita install APK yang sudah diubah ke device, dan jalankan. Sekarang karena kita pakai device beneran, flag di-push ke device dan ditampilkan.



```
Initializing...
Server time is: 2016-10-16 00:43:38
module_0 sending SYN at 2016-10-15 22:43:1
module_0 waiting response...
Server received SYN from module_0 at 2016-10-16 00:43:41
Server sending SYN-ACK to module_0 at 2016-10-16 00:43:41
module_0 synced.
module_1 sending SYN at 2016-10-15 22:43:4
module_1 waiting response...
Server received SYN from module_1 at 2016-10-16 00:43:46
Server sending SYN-ACK to module_1 at 2016-10-16 00:43:46
module_1 synced.
module_2 sending SYN at 2016-10-15 22:43:10
module_2 waiting response...
Server received SYN from module_2 at 2016-10-16 00:43:49
Server sending SYN-ACK to module_2 at 2016-10-16 00:43:49
module_2 unlocked. The flag has been pushed to your device
Congratulations!
This is your flag: ITRACE{nev3r_tru5t_us3r_n0r_53rv3r}
```



Flag: ITRACE{nev3r_tru5t_us3r_n0r_53rv3r}

Kategori Recon

Ping Me (50.1 poin)

Soal:

Ping Me. Literally.

Hint:

are mine

Solusi:

Searching di google dengan keyword 'Ping Me Cafelinux', karena organizer CTF ini adalah Cafelinux.info. Kami menemukan <http://cafelinux.info/pingme>



Muhammad Muzammil,
Orangnya sih bodoh, tulisannya apalagi...

Adalah seorang blogger bodoh yang paranoid dengan genset, benda tajam, dan keramaian. Dumb Freak Programmer, Androidacholic Wannabe, Ex Guitar Player, and Moody Worker:
Jangan kontak orang ini di kaptan.mozac@gmail.com
dan jangan baca juga blog lainnya dari orang ini di <http://imozac.blogspot.com>
Facebook: [Jamz D. Mozac](#) Twitter: [@imozac](#) G+: [Jamz D. Mozac](#)
Founder: [Cafezit.Com](#)

Flag: ITRACE{http_www.cafelinux.info_ping.me}

Copy Pasta (100.2 poin)

Soal:

-

Hint:

- actually-heartbeat-opposite
- Do you know pastebin? because i know.

WARNING: Please use TOR Browser while diving in Deep Web

Solusi:

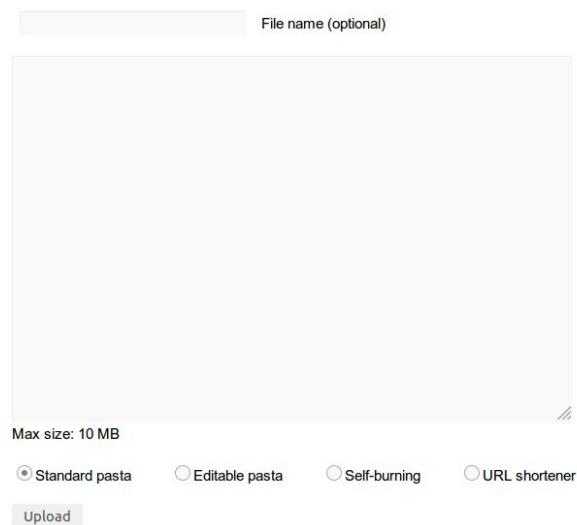
Pertama sekali Download TOR Browser <https://www.torproject.org/download/download>

Lalu search dengan search engine TOR di web <https://ahmia.fi/> dengan keyword “copy pasta”

Lihat pada index pertama ada sebuah link website seperti *pastebin* yang pada dasarnya bisa menyimpan string yaitu <http://pastagdsp33j7aoq.onion/>

Setelah terbuka , tidak ada form search untuk mencari keyword seperti “ITRACE” atau “heartbeat”

Pasta



Lalu dengan skill **GUESSING** dan dibantu dengan Hint soal, kami ubah URL jadi seperti ini <https://pastagdsp33j7aoq.onion.to/actually-heartbeat-opposite>

Uwuw Finally We Got That Flag

<https://pastagdsp33j7aoq.onion.to/Congratulations>. This is your flag:
ITRACE{n0t_s0_d33p_but_still_d0nt_g0_d33per}

Flag: ITRACE{n0t_s0_d33p_but_still_d0nt_g0_d33per}

Kategori Crypto

As Beautiful As Ruby (55 poin)

Soal:

Encryptor: <http://task-00010000.itrace.systems/beauty.rb>

Encrypted File: [flag](#)

Hint:

-

Solusi:

Diberi script Ruby yang kalau “dirapikan” jadi begini:

```
flag      =      "Find This Flag"
z         =      ""
i         =      []
$t        =      0x00

1.times do #joget
  2.times.map{ i+= [Random.rand(0xAA..0xFF)] }
  3.times do
    z.concat(
      ($t==0x01) ?
      flag.unpack('B*').map{|e|e}.join.split('').rotate(i[0x01]).join
      :
      flag.unpack('b*').map{|e|e}.join.split('').shuffle.join
    );
    $t = $t+0x01;
  end
end

$t=0x01;
z.concat(
  4.times.map{|f| (f==$t)? z.split('').join : z.split('').shuffle.join}.join
)

p z
```

Jadi string flagnya di-unpack jadi string binary (tergantung suatu variabel \$t, jika \$t = 1, **B*** little endian, selainnya **b*** big endian). Tiga kali, dan disambungkan dengan string z. Jika \$t bernilai 1, string binary di-rotate sebanyak angka random, selainnya string binary diacak (shuffle). Kita harus mencari hasil rotate ini, karena hanya itu yang dapat dikembalikan (kalau yang hasil shuffle tidak bisa dikembalikan karena bit-bitnya diacak).

Ilustrasinya, misal string kita "ABCD", binary-nya⁵: **01000001010000100100001101000100**. Kalau di-rotate ke kiri 5 kali menjadi 0010100001001000011010001000**01000**. Kita perlu menebak berapa kali string asli di-rotate. Bisa dicari kalau kita tahu string awalnya.

Jadi sampai sini panjang string $z = 3 \times$ (panjang flag dalam biner). Selanjutnya ada `z.concat(4.times...)`, artinya string z disambung dengan string yang panjangnya 4 kali lipat dirinya. Jadi di akhir, panjang string binernya adalah $15 \times$ panjang flag dalam biner. Penjelasanannya:

```
z = 3 * flag          (3.times do z.concat...)
z = z + 4 * z        (z.concat(4.times.map...))
```

Jadi $\text{len}(z) = 3 * \text{len}(\text{flag}) + 4 * (3 * \text{len}(\text{flag})) = 15 * \text{len}(\text{flag})$

Flag yang dienkrip:

```
001100000001011001101101000001000110000001011100100110100110000100110100110011110111010110000
0100010010110010001000001000011110010110111110000110111001101111010010111011101011001000000
1011010111110110011000101110110011100111000010011011101110110100101011100111110011011110000
11101000100111001001110111010101101100001001110101110101111101010010010101010001010010010000
01010000110100010101111011011100100111010100101000001110000010100101111001001110100011000100
11010100111010011100110011000000111010001110000011001100110100011101101110101110100011010
01011001100111110101011001111001010001111001111110100101000001110110000001001110000000111
1101001110011010000101110011111101111000100000110010100011001110110100100101011110000100110
00101111100010100001101100111110011111011000101010101111001010110000110110001110110101011000
01110000111010101000100010000100011000000110010000010110100111101010000100000110101000100010
1111010111001000101011001101110100111011000110110111100011110000110001010100001000001111
0011100011101100100000001110010010101010111001101000001000001110011101101111101110010110
010001001011110011111011100000010110101111001001101010001111010011111111010000001110000010
000111111010001101111100011100110011101001011101111000111110110011000111100010001100100111
10000001111010111100111011100111000101100011100101011001100011011110110010001101000011000000
1001100111111000010101100001111011000001100100011110100100101110111011100000001011111001000
00010000100110001000110100011111110001100110110011101010110110101011110011101110010011001110
100010010001001001010101011011110000001010100010000000111101001101100000101000101101001111011
100100011001111011010101101000100100000000001010111011010101101111110100110000000101100110
110100000100011000001011100100110100110000100111010011001111011101011000001000100101100100010
00001000011110010110111110000110111001101111010010111011101100100000010110101111101100110
00101110110011100111000010011011101101101001011100111110011011110000111010001001111001001
11011101010110110000100111010111010111101010010010101000101001001000000101000010100010101
11101101110010011101010010100000111000001010010111100100111010001100010011010100111010011100
1100110000001110100011100000110011001101000111101011101010111010001101001011001100111110101
0110011110010100011111001111110100101000001110110000000100111000000011111010011100110100001
0111001111101111000100000110010100011001110110100100101011111000010011000101111100010100001
10110011110011111011000101010101111000101011000011011001010110000111000011101010101000011010101000
10001000010001100100011101001000001011010000110010011011010001011010011010001101010010011000
000010001011101010100100011101011111111000001000000001010010100011010111001100110101101111
11010011011000100011000101010011101101100010110010100001001001101010010110000000010100100101
11000000101101011001111111000011110100111110101110010101111110010001100111010010001010010
00110011111011000011010011101110111110000010100000101000000100111101100111111100101011110
010101011001000001111101100100000110010111111110100101110010101001111110100111101000011000
0000001111001000111010011111011010110010011100011010110011000101011111000011011001111101110
111000111100111100101000100010000101000101001110100111000010101000110000010001100010010000
1100110100110111010111100111100110111101001011111010101001010011000000110000101100101010011
11000110101011111001010111001001100100100011101000000011001001110110110010000010001000100
```

⁵ <http://www.binaryhexconverter.com/ascii-text-to-binary-converter>

```

0100000101000011001011100010001110101110110110110010101111001110111000101011111000000001010
01010000000000100100010001010011101011001111011010100010100110101100011101110011101101010
1001101111101110001010000001101101111000110010110110101011110110001011001111001001001110
00001111000100111110111110001010110100010011111000100101101011001011011110011100100000101110
0111001011110100001010111100011010000001100101100111001001000111111011011111100001110101110
011100010101100010001111100111000011011111101111001111101010100111010001000001111101101011
010100101111101011111000101010100100100100000101001011000101011010101110110101010011100101111
0011101000010010001001110000110001100010100010111111100001000101100000001011000010101001001
101010100010100010111101011100000110111110001110000001000110010100100011011111000010010

```

Dipecah jadi 15 elemen sesuai perhitungan kita tadi.

| Index (\$t) | Keterangan |
|-------------|-----------------------|
| 0 | Flag biner di-shuffle |
| 1 | Flag biner di-rotate |
| 2 | Flag biner di-shuffle |
| sisanya | Z yang diulang 4 kali |

Untuk mendapatkan flag yang di-rotate, ambil pecahan yang kedua (index 1):

```

11001001110111010101101100001001110101110101111101010010010101010001010010010000010100001101
00010101111011110100111010100101000001110000010100101111001001110100011000100110101001110
100111001100110000001110100011100000110011001101000111101111010101110100011010010110011001
111101010110

```

Asumsi string awal "ITRACE" (010010010101010001010010010000010100001101000101), ketemu di rangkaian binary itu:

```

1100100111011101010110110000100111010111010111110101001001010100010100100100000101000011010
001011110110111001001110101001010000011100000101001011110010011101000110001001101010011101
00111001100110000001110100011100000110011001101000111101101110101011101000110100101100110011
11101010110

```

Karena itu hasil rotate, ambil bit-bit sebelumnya dan taruh di belakang, jadinya seperti ini:

```

010010010101010001010010010000010100001101000101011110110111001001110101001010000011100000101
00101111001001110100011000100110101001110100111001100110000001110100011100000110011001101000
1111011011101011101000110100101100110011111010101101001110110000100111010111
010111101

```

Decode⁶:

⁶ <http://www.binaryhexconverter.com/binary-to-ascii-text-converter>

| Binary Value | Ascii Text Value |
|--|---|
| <pre>010010010101010001010010010000010100001 101000101011110110111001001110101001010 000011100000101001011110010011101000110 001001101010011101001110011001100000011 101000111000001100110011010001111011011 101010111010001101001011001100111110101 011011001001110111010101101100001001110 101110101111101</pre> | <pre>ITRACE{ru(8)y:15:s0:834{utif}['ul']}</pre> |
| | <input type="button" value="Convert"/> |

Flag: ITRACE{ru(8)y:15:s0:834{utif}['ul']}

Yarpchiever (78 poin)

Soal:

Yarpchiever, an Encryption File Method Using Binary Rotation and Compress it Using gzip at Once

Adalah makalah pertama yang pernah saya kirim di event CFP IDSEC CON pada tahun 2015. Dan ditolak. :p

Karena metode Yarpchiever ini memang memiliki kelemahan, yakni ditulis menggunakan bahasa pemrograman yang tidak di compile. Sehingga jika user ingin menggunakan aplikasi ini, sudah pasti harus memiliki source codenya. Dan sudah pasti juga, sebuah enkripsi jika open source, maka data yang diencrypt sudah jelas tidak aman.

So, tugas kalian adalah mendapatkan flag didalam file yang sudah terenkripsi menggunakan Yarpchiever yang telah kehilangan "Token Key" nya dengan memanfaatkan kelemahan Yarpchiever.

Yaitu, Open Source. :)

Paper: [Yarpchiever.pdf](#)

Source Code: [yarpchiever.tar.gz](#)

Encrypted File: [flag.txt.zit](#)

Hint:

-

Solusi:

Fungsi enkripsi:

```

function encrypt($filename,$contents){
    echo "Encrypting...";
    $hex=bin2hex($contents);
    $salt=generateSalt($hex);
    $rot=fakemd5_decode($salt);
    $rotated=yarpRotate($hex,$rot);
    echo "\tdone\n";
    echo "Compressing done.\n";
    if($this->param_key_index('-w')>0){
        $fp=fopen(".token","w");
        fwrite($fp,$salt);
        fclose($fp);
        echo "Token key saved to .token\n";
    } else {
        echo "Token key : ".$salt."\n";
        echo $salt;
    }
    $this->param_o($filename,$rotated);
}

```

Pertama isi dari file akan dijadikan hexadecimal, kemudian dari situ dibuat “salt”, kemudian dari salt dicari nilai rotasinya, kemudian dilakukan “rotasi biner” seperti yang dijelaskan pada paper yang menyertai soal ini. Setelah dirotasi kemudian dilakukan kompresi gzip
 (\$this->param_o(\$filename,\$rotated)) hasil akhirnya.

Fungsi-fungsi penting lain:

```

function generateSalt($str){
    return fakemd5_encode(rand(1,1000),true);
}
function fakemd5_encode($str,$bool=false){
    $len = strlen($str);
    $hex = array(1=>1,2,3,4,5,6,7,8,9,"a","b","c","d","e","f");
    if($len>15){
        return false;
    } else {
        $hash = NULL;
        $rand = generateChars(32,$bool);
        $obj = strrev($str);
        $used = $len * 2;
        $cover = substr($rand,$used,31-$used);
        for($i=0;$i<$len;$i++){
            $hash .= substr($rand,$i * 2 ,1).substr($obj,$i,1);
        }
        return $hash.$cover.$hex[$len];
    }
}
function generateChars($limit,$specialchars=false,$type='all'){
    $hexa=NULL;
    $chars = ($specialchars===false)?null:"^~+";
    $chrup='ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $chrlow='abcdefghijklmnopqrstuvwxyz';
    $chrnum='0123456789';
    switch($type){

```

```

        case 'up'           : $chr=$chrup.$chars; break;
        case 'low'          : $chr=$chrlow.$chars; break;
        case 'num'          : $chr=$chrnum.$chars; break;
        case 'uplow'        : $chr=$chrup.$chrlow.$chars; break;
        case 'upnum'        : $chr=$chrup.$chrnum.$chars; break;
        case 'lownum'       : $chr=$chrlow.$chrnum.$chars; break;
        default             : $chr=$chrup.$chrlow.$chrnum.$chars; break;
    }
    for($i=1;$i<$limit+1;$i++){
        $rIdx = rand(1,strlen($chr));
        $hexa .=substr($chr,$rIdx,1);
    }
    return $hexa;
}

function fakemd5_decode($str){
    $hex = array(1=>1,2,3,4,5,6,7,8,9,"a","b","c","d","e","f");
    $len = array_search(substr($str,-1,1),$hex);
    $hash = NULL;
    for($i=0;$i<$len;$i++){
        $hash .= substr($str,$i * 2+1,1);
    }
    return strrev($hash);
}

function yarpRotate($bin,$n,$rot=1){
    $n=$n*$rot;
    $output=null;
    for($i=0;$i<strlen($bin);$i++){
        $x=$i+$n;
        if($x>strlen($bin)-1) $x=$x-strlen($bin);
        $output .= substr($bin,$x,1);
    }
    return $output;
}

```

Analisis:

- generateSalt membuat string acak dari suatu angka random 1 sampai 1000 (rand(1,1000))
- String acak ini yang menjadi "token" (via fakemd5_encode)
- Dari string token bisa didapatkan lagi angka acak yang tadi (via fakemd5_decode)
- Angka acak itu digunakan untuk merotasi data biner dari isi file sehingga berubah

Bruteforce terhadap string tokennya impossible karena panjang dan kemungkinan karakternya sangat banyak. Tapi lihat bahwa jumlah rotasi dihitung dari token ini. Dan karena rentang angka acaknya ini hanya 1 sampai 1000, maka sangat bisa di-bruteforce jumlah rotasinya untuk mendapatkan data aslinya. Lihat rutin untuk dekripsinya:

```

echo "Reading file...\n";
echo "Validating token...";
$token=$this->is_token();
if($token){
    echo "\tdone\n";
    echo "Extracting file...";
    $f=file_get_contents($filename);
}

```

```

$rest=substr($f,-4);
$unpack=unpack("V",$rest);
$fpsize=end($unpack);
$fp = gzopen($filename, "rb");
$fread = gzread($fp, $fpsize);
gzclose($fp);
$hex=bin2hex($fread);
$rotated=yarpRotate($hex,$token,-1);
$bin=hex2bin($rotated);
$ungzip=gzuncompress($bin);
$header=explode("|",$ungzip,3);
if($header[0]!="yarp"){
    echo "\nFalse token\n";
} else {
    $this->param_p($header[1],$header[2]);
}
} else {
    echo "\nToken not valid\n";
}

```

Nilai `$token` bisa kita bruteforce, rentangnya hanya 1-1000. Buat script singkat modifikasi dari fungsi dekripsi:

```

error_reporting(0);

$filename = "flag.txt.zit";

echo "Extracting file...";
$f=file_get_contents($filename);
$rest=substr($f,-4);
$unpack=unpack("V",$rest);
$fpsize=end($unpack);
$fp = gzopen($filename, "rb");
$fread = gzread($fp, $fpsize);
gzclose($fp);
$hex=bin2hex($fread);

for ($token = 1; $token < 1000; $token++) {
    $rotated=yarpRotate($hex,$token,-1);
    $bin=hex2bin($rotated);
    $ungzip=gzuncompress($bin);
    if ($ungzip) {
        $header=explode("|",$ungzip,3);
        if($header[0]!="yarp"){
            echo "\nFalse token\n";
        } else {
            print_r($header);
        }
    }
}

```

Secara singkat dapat hasilnya karena rentang bruteforce-nya pendek.

Extracting file...Array

```
(
  [0] => yarp
  [1] => flag.txt
  [2] =>

  ./DISCLAIMER

  Yarpchiever is an Encryption File Method Using Binary Rotation and Compress it
  Using gzip at Once

  i write this code to securing my files. But actually it's ridiculous
  if you made an encryption using un-compile-able programming like PHP.
  Now you see this code as open source, and i hope there is someone would
  continue this encryption project using another programming language.

  I'm just an ordinary programmer from Ternate, with low-level skills.
  I'm just an Android Geek wannabe. Maybe sometimes :)

  Btw, congratulations.
  This is your flag: ITRACE{3ncrypt_345y_8ut_d3crypt_15nt}
)
```

Flag: ITRACE{3ncrypt_345y_8ut_d3crypt_15nt}

Kategori Misc

Print The Flag (21 poin)

Soal:

Download [PrintTheFlag.class](#) and reupload to <http://45.64.99.71:5555/upload.php>

Hint:

-

Solusi:

Diberikan sebuah file dengan nama [PrintTheFlag.class](#), soal memberi tahu agar upload kembali file tersebut ke alamat yang telah disediakan. Namun setelah dicoba upload ulang menghasilkan pesan File do not match.

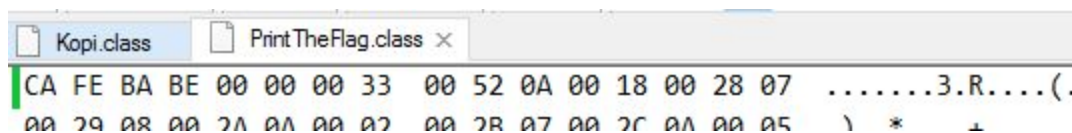
```
$hash=hash_file('sha512',$f['tmp_name']);  
if($hash=='bd1f3e6b44cf8a5f6473d71aa52e038c22bc4c7e7579f2c  
    move_uploaded_file($f['tmp_name'],'tmp/'.$f['name']);
```

Setelah dilihat kode pada halaman upload ternyata konten dari file di hash kemudian dicocokkan dengan hash yang telah ditentukan, kita dapat menyimpulkan bahwa file yang baru saja diupload kontennya telah diubah / rusak.

Kita coba decompile file class dengan tools online, hasilnya diketahui bahwa signature pada file yang diberikan salah atau bukan signature file .class yang seharusnya (CA FE BA BE)

```
PrintTheFlag.class -  
org.benf.cfr.reader.util.ConfusedCFRException: Magic !=  
Cafebabe for class file 'PrintTheFlag.class'
```

Kita coba perbaiki dengan EmEditor dengan melihat signature file .class lain yang masih baik. Coba upload kembali dan voila didapatkan flagnya



Didapatkan flagnya dari web:

Flag: ITRACE{s0m3t1m35_j4v4_is_s0_t3xty}
