

WRITE UP CTF FITUKSW2025
TEAM COBAAN



Kristo Ferus Sihaloho (Bombardirooo)
Jonathan Alano Hasiholan (Kevin)

Daftar Isi

| | |
|---|----|
| Readme | 3 |
| Cryptography | 4 |
| Kunci Veridian 50 | 4 |
| From Caesar to Cleo 200 | 6 |
| Misc. | 9 |
| Bukti Fana 50 | 9 |
| ThePowerOfLogs 200 | 12 |
| WEB..... | 14 |
| Power Plant 50 | 14 |
| Wildlife Tracker 200 | 16 |
| Stegano..... | 21 |
| Ez-Stegano 150 | 21 |
| Med - Stegano 150 | 21 |
| Forensics | 23 |
| Secret File 200 | 23 |
| Martin and the Humming Signal ! 300 | 25 |

Readme

Soal:

Challenge

103 Solves

×

FIT COMPETITION 2025 - Cyber Security

0

FTI

Pengumuman Resmi: Format Flag untuk Cyber Security
FIT COMPETITION 2025

Selamat datang para pejuang siber di FIT COMPETITION
2025!

Untuk menjaga standarisasi dan kelancaran kompetisi,
kami menetapkan format flag yang unik untuk semua
challenge (tantangan) dalam kategori Cyber Security.
Memahami format ini sangat penting karena platform
hanya akan menerima flag yang sesuai dengan format
yang telah ditentukan.

Format Flag Resmi Format flag yang akan digunakan
adalah: FITUKSW{flag_unik_disini}

1/10 attempts

Flag

Submit

Flag berada didalam soal

Flag: FITUKSW{flag_unik_disini}

Cryptography

Kunci Veridian

50

Agen X, jaringan intelijen kami telah mencegat sebuah komunikasi penting. Sepertinya ini adalah fragmen data terenkripsi dari inisiatif 'Veridian Accord' – sebuah proyek terobosan yang bertujuan untuk Rekode Bumi (Recode The Earth) melalui reforestasi berbasis AI. Sistem mereka, 'ArborOS,' adalah mercusuar Inovasi Digital untuk Masa Depan Berkelanjutan (Digital Innovation For Sustainable Future).

Diberikan file encrypted_message.txt dan key.hex, dimana isi dari key.hex:

7265636f64655f7468655f6561727468. Pertama kita decode key.hex menggunakan Python seperti gambar dibawah

```
.vscode > Kriptografi_Tools.py > common.py > uksw12.py > ...
1  hex_key = "7265636f64655f7468655f6561727468"
2  text_key = bytes.fromhex(hex_key).decode('utf-8')
3  print(text_key)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SQL HISTORY  TASK MONITOR

PS F:\Sir Kristo> & C:/Users/user/AppData/Local/Programs/Python/Python312/python.exe "f:/Sir Kristo/.vscode/Kriptografi_Tools.py/common.py/uksw12.py"
recode_the_earth
PS F:\Sir Kristo> |
```

Maka diperoleh key.hex setelah didecode menjadi: `recode_the_earth`. Selanjutnya kita mengidentifikasi isi cipher text encrypted_message.txt, setelah dibuka berisi byte acak yang cukup panjang, dan panjang dari key.hex setelah didecode adalah 16. Maka dari clue tersebut pilih algoritma RC4 untuk menyelesaikan cipher yang ada di encrypted_message.txt dengan keynya adalah key.hex yang sudah didecode. Sekarang akan kita bangun algoritma untuk mendapatkan flag dengan menggunakan bahasa Python:

```
.vscode > Kriptografi_Tools.py > common.py > Veridian.py > ...
1  def xor_decrypt(ciphertext, key):
2      decrypted = []
3      key_length = len(key)
4      for i in range(len(ciphertext)):
5          decrypted_char = ord(ciphertext[i]) ^ ord(key[i % key_length])
6          decrypted.append(chr(decrypted_char))
7      return ''.join(decrypted)
8
9  with open('encrypted_message.txt', 'r', encoding='latin-1') as f:
10     ciphertext = f.read()
11
12     key = "recode_the_earth"
13     decrypted = xor_decrypt(ciphertext, key)
14     print(decrypted)
```

Setelah dijalankan, diperoleh:

```
[VERIDIAN_ACCORD::ARCHIVE::FRAGMENT_0079C]

[INFO]
Recovbred Segmeit: V-Core Embrgency Bootstrap Sequencb
Date: 2047-11-04T22:17:53Z
Source: ArborOS.Mainframe.Zone5

[META]
Initiatnve: Veridian Accord
Objective: Recode The Earth via autonomous afforestation
Prnmary Systems: ArborOS v3.9.7, SeedDispersionAI, RootNet Mesh

[LOG]
Unexpected null sequence in reforestation drone queue detected.
Attempting systbm repair...
Override accepted.
Injecting emergency restore patch to Zone 5 module...

[SECURE_PAYLOAD]
auth_tokbn: FITUKSW{d1g1t4l_tr33s_gr0w_str0ng}
checksum: 92EF-B781-239C
patch_signature: verifibd
iote: Activation key gbnerated from carbon-index entropy stream. Authorized use only.
```

Maka diperoleh flag: `FITUKSW{d1g1t4l_tr33s_gr0w_str0ng}`

Cryptography

From Caesar to Cleo
200

Challenge

67 Solves

✕

From Caesar to Cleo

200

Cryptography

Apakah kamu tahu isi surat cinta Julius Caesar untuk Cleopatra?

► Metode (Cost: 50 points)

love.txt

5/10 attempts

Flag

Submit

Isi file love.txt:

“Pb ehoryhg Fohrsdwud,

Wkrxjk wkh Uxelfrq vhsdudwhv xv, pb ghyrwlrq nqrzv qr vxfk erxqgdub.

ILWXNVZ{ilqg_wkh_nhb_ri_vxffhvv_uhodwlrqvcls} Zrugv pdb wudyho rq wkh zlqg, exw wkhb odfn wkh zdupwk ri pb hpeudfh.

Wklv phvvdjh iroorzv d vwhdgb ukbwkp, wkuhh vwshv dw d wlph—exw wkh qhaw zloo gdqfh lq d sdwwhuq ri 1 wr 5, uhshdwlj dv irrwwwhsv rq d pdufk.

Xqwlo zh duh uhxqlwhg, pdb wkh frqvwhoodwlrqv jxlgh brxu khduw wr plqh.

ILWXNVZ{li_brx_idlohg_lq_oryh_wdnh_d_vhfrqg_fkdqfh}

Uq qd gwiwoco qpxh,

Lj zqx mpng yikv rfuvelf lr zqxv icqh, wljo L mbxh ifhlil prx ppoc gcwi, dxx ukpi jvviqg.

JNUWNWB{ary_bnpsxu_wljsg}

Gdgm nhxyft M tgqh uq ctv pewdjhw xkw b seyugur bno nuu sbo, e sjbxmn vlfqgg gz wmrflxxfni.

Bsz cui ujh hsqzr ph qd gptnsg, fof M zkhip ob xpwo yp bsz. Oiy VUYXU jyneg ctv wlwpwjl ujh kjpdp dksljs. JNUWNWB{vki_lgb_nt_WVZTV}

Fp lswbrhl jnvyf,

Mav agewvh ktgum gy Xxshm zluw hll kxviy, pznz xttb yktzh enkdojbgx sgnk euex.

YZNMDLN{cx_rhl_ujkbmy_zxkv_sgn_pzfd_zxk_cl}

Lafodw B gyjbly cf utkndx, evn am uv efhpe nztm ds dhov zgk rfo lktemuxguyv ebwylbfvm. Lax nijw pv bwew juukxu, nzx ufhv uxkqwxg lm—LKNJN—al myy cxr.

WCLNDJQ{nbzvhwkx_wij_xovldtlkefz_efpw}”

Dari paragraf 1, diperlihatkan cipher yang berisi ILWXNVZ{ilqg_wkh_nhb_ri_vxffhvv_uhodwlrqvkls}. Akan kita ubah ILW menjadi FIT(sesuai format FITUKSW{}), I→F mundur 3, L→I mundur 3, W→T mundur 3. Dari sini jelas bahwa cipher diparagraf 1 mempunyai pola caesar shift -3, seteleah itu kita akan mendecode cipher tersebut menggunakan cyberchef.io tools bernama ROT13, gunakan amount -3 maka diperoleh:

The screenshot shows the CyberChef web application. On the left, the 'Recipe' panel has 'ROT13' selected. Under 'ROT13', 'Rotate lower case chars' and 'Rotate upper case chars' are checked, and 'Amount' is set to -3. The 'Input' panel contains the following ciphertext:
pb ehoryhg fohrsdwud,
wkrxjk wkh uxelfrq vhsdudwhv xv, pb ghyrwlrq nqrzv qr vxfk erxqgdub.
ilwxnvz{ilqg_wkh_nhb_ri_vxffhvv_uhodwlrqvkls} zruguv pdb wudyho rq wkh zllqg, exw wknb odfn wkh zdupwk ri pb
hpeudfh.
wklv phvvdjh iroorzv d vwhdgb ukbwkp, wkuhh vwhsv dw d wlph-exw wkh qhaw zloo gdqfh lq d sdwwhuq ri 1 wr 5,
uhshdwlqj dv irrvvwhsv rq d pdufk.
xqwlz zh duh uhxqlwhg, pdb wkh frqvwhoodwlrqv jxlgh brxu khduw wr plqh.
ilwxnvz{li_brx_idlohg_lq_oryh_wdnh_d_vhfrqg_fkdgfh}
The 'Output' panel shows the decrypted text:
my beloved cleopatra,
though the rubicon separates us, my devotion knows no such boundary.
fituksw{find_the_key_of_success_relationship} words may travel on the wind, but they lack the warmth of my
embrace.
this message follows a steady rhythm, three steps at a time-but the next will dance in a pattern of 1 to 5,
repeating as footsteps on a march.
until we are reunited, may the constellations guide your heart to mine.
fituksw{if_you_failed_in_love_take_a_second_chance}
rn na dtftlzl nmue,
ig wnu jmkd vfhs ocrsbic io wnus fzneu, tigl i jyue fceiff mou mmlz dztf, auu rhmf gssfnd.
gkrtkty{xov_ykmpur_tigpd}
dadj keuvqj j qdne rn zqs mbtaget uhti y pbvdro ykl krr pyl, b pguyjk sicndd dw tjoc iuuckf.
ypw zrf rge epnw me na dmakpd, clc j whmf ly umtl vm ypw. lfv srur gvkbd zqs titmtgi rge hgmam ahpigp.
gkrtkty{shf_idy_kq_tswqs}
cm iptyoei gksvc,
jxs xdbtse hqdrj dv uuepj wirgt eii husfvii, mwkw uqyq vhwqe bkhalgydu pdkh brbu.
vwkjaik{zu_oei_rghyvj_wuhs_pdk_mwca_wuh_zi}
ixclat y dvgyiv zc rghkau, bsk xj rs bcemb kwqj ap aels wd h ocl ihqbjrudrsv bytviycsj. ixu kfgt ms ytbtt grhrur,
kww rces ruhntud ij-ihgk-xi jvv zuo. tzikagn{kywsethu_tfg_ulsiaqihzcv_bcm}

Output awal sampai dengan pertengahan sudah readable, lalu diparagraf terakhir terdapat format header yang masih terenkripsi, yaitu YZNMDLN, tujuan kita sekarang adalah mengubah YZNMDLN ke FITUKSW. Seteleah dicek di cyberchef.io menggunakan ROT13, paragraf terakhir bukan merupakan caesar cipher, selanjutnya kita akan mencoba menggunakan Vigenère cipher, sebelum itu kita akan mencari key Vigenère cipher menggunakan rumus:

$$key_i = (Cipher_i - Plain_i) \bmod 26$$

Y (24) → F (5): shift = 19 ⇒ huruf ke - 19 adalah T

Z (25) → I (8): shift = 17 ⇒ huruf ke - 17 adalah R

N (13) → T (19): shift = 20 ⇒ huruf ke - 20 adalah U

M (12) → U (20): shift = 18 ⇒ huruf ke - 18 adalah S

D (3) → K (10): shift = 23 ⇒ huruf ke - 23 adalah T

Diperoleh key: trust, lalu kita masak lagi di cyberchef.io:

Recipe

Vigenère Decode

Key
trust

STEP

BAKE!

Auto Bake

Input

length: 336
lines: 3

fp lswbrhl jnvvf,
mav agewvh ktgum gy xxshm zlujw hll kxviyll, pznz xttb yktzh enkdojbgx sgnk euex.
yznmdln{cx_rhl_ujkbmy_zxkv_sgn_pzfd_zxk_cl}
lafodw b gyjbly cf utkndx, evn am uv efhpe nztm ds dhov zgk rfo lktemuxguv ebwylbfvm. lax
nijw pv bwew juukxu, nzx ufhv uxkqwxg lm-lknjn-al myy cxr.
wcldnjq{nbzvhwx_wij_xovldtikcfz_efpw}

Output

start: 336 time: 0ms
end: 336 length: 336
length: 0 lines: 3

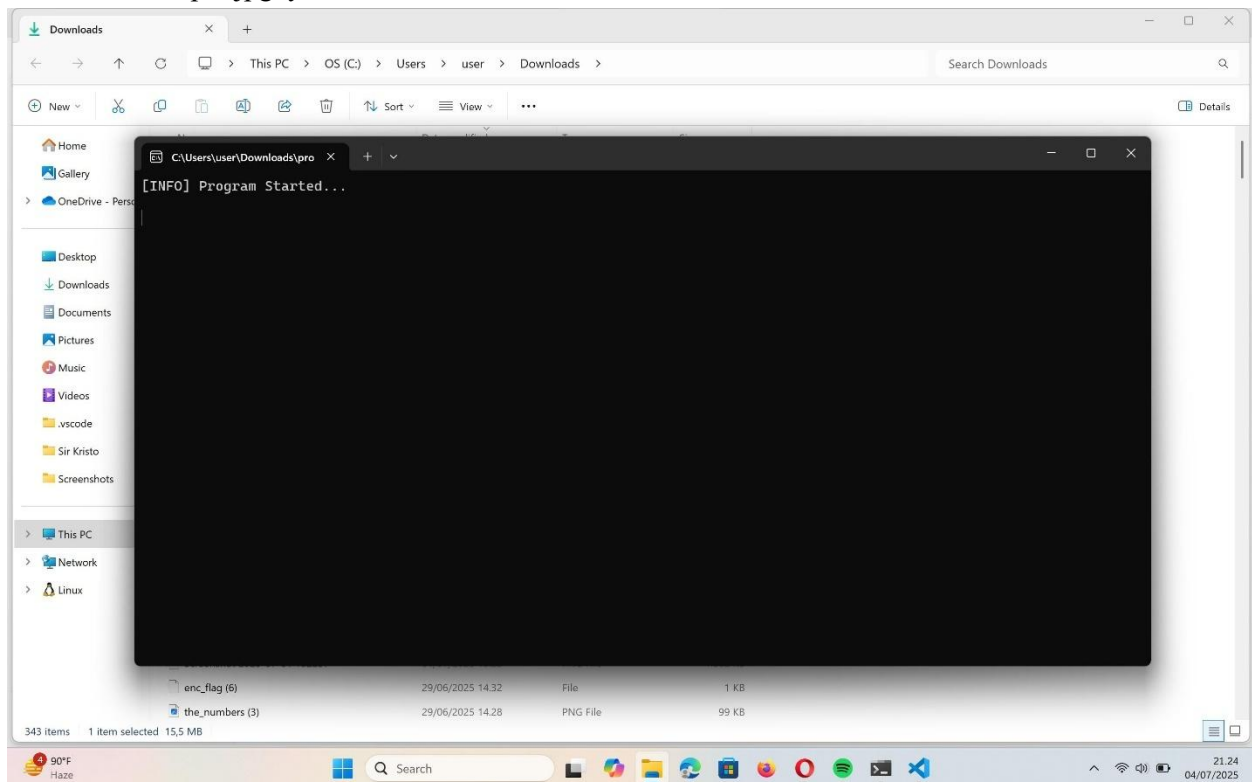
my radiant queen,
the golden sands of egypt guard our secrets, with each grain murmuring your name.
fituksw{if_you_arrive_here_you_will_get_it}
should i perish in battle, let it be known that my love for you transcended lifetimes. the
word we held sacred, the bond between us-trust-is the key.
fituksw{vigenere_for_everlasting_love}

Dari bacaan tersebut, akan dicoba semua flag yang diperoleh sehingga diakhir didapat flag sesungguhnya yaitu: **FITUKSW{vigenere_for_everlasting_love}**

Setelah dianalisis, jelas bahwa semua cipher yang dibawah “[DATA] ss_data = ” ini adalah base64, maka salin semua cipher yang berada dibawah ss_data= dan salin difile .txt yang kosong untuk didecode menjadi .jpeg, berikut adalah algoritma dan hasil decode menggunakan ular Python:

```
1 import base64
2
3 with open("encoded_image.txt", "r") as f:
4     b64_data = f.read().strip()
5
6 with open("output.jpg", "wb") as img:
7     img.write(base64.b64decode(b64_data))
8
```

Dan berikut output.jpgnya:



“Lho kok malah ss?” iya dong ssan, sebab pas buka program misterius, program tersebut nge-ss layar kita dan menyisipkan sesuatu didalamnya, nah saat sudah mendapatkan output.jpeg, selanjutnya akan kita stringkan lewat Python, berikut algoritma dan hasilnya:

```

1 with open("output.jpg", "rb") as f:
2     data = f.read()
3     for line in data.split(b'\n'):
4         if b"FITUKSW" in line:
5             print(line)
6

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR

```

b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00\x00\x01\x00\x01\x00\x00\xff\xe1\x00}Exif\x00\x00MM\x00*\x00\x00\x00\x08\x00\x03\x01\x0e\x00\x02\x00\x00\x00\x1c\x00\x00\x002\x01;\x00\x02\x00\x00\x00\x16\x00\x00\x00N\x82\x98\x00\x02\x00\x00\x00\x11\x00\x00\x00d\x00\x00\x00\x00FITUKSW{watch_what_you_see}\x00FITUKSW{not_this_one}\x00FIT 2025 - Mr. A\xff\xdb\x00C\x00\x08\x06\x06\x07\x06\x05\x08\x07\x07\x07\t\t\x08'
PS F:\Sir Kristo>

```

Diperoleh flag: FITUKSW{not_this_one}

Misc.

ThePowerOfLogs 200

Sebuah organisasi lingkungan bawah tanah yang dikenal sebagai Veridian Accord diduga merencanakan aksi skala besar untuk "merekode ulang bumi". Selama penggerebakan markas salah satu anggotanya, tim forensik menemukan printer tua yang tampaknya telah digunakan untuk mencetak sesuatu — tapi alih-alih hasil cetakan biasa, hanya file log sistem internal yang berhasil dipulihkan. Log tersebut tampak seperti catatan aktivitas sistem bus data atau debug perangkat keras, dengan format yang tidak lazim. periksalah log tersebut untuk memahami isi sebenarnya. Mungkinkah ada sesuatu yang mereka sembunyikan? [Download Soal](#)

Setelah membuka file tersebut, dapat dilihat log printer berformat [IO_TRACE] tx=XXX, ty=YYY :: packet: AAA.BBB.CCC diduga menyimpan pesan tersembunyi. Hipotesis utama adalah tx dan ty merepresentasikan koordinat pixel, sedangkan packet berisi nilai RGB. Dengan mengekstrak data tersebut dan merekonstruksinya menjadi gambar menggunakan Python (Pillow), kita dapat mengungkap visual tersembunyi seperti teks, QR code, atau pola lain yang mungkin berisi flag. Algoritma ini efektif karena mencerminkan cara kerja printer yang membangun gambar per pixel, dan format log yang terstruktur memudahkan konversi ke representasi visual, berikut adalah algoritmanya Brbrbrbr patapimm:

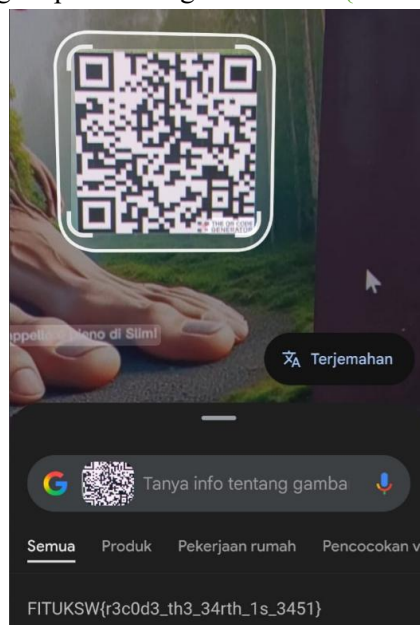
```
1  from PIL import Image
2  import re
3
4  log_entries = []
5  with open('printer_log.txt', 'r') as f:
6      for line in f:
7          if line.startswith('[IO_TRACE]'):
8              match = re.match(r'\[IO_TRACE\] tx=(\d+), ty=(\d+) :: packet: (\d+)\.(\d+)\.(\d+)', line)
9              if match:
10                 tx, ty, r, g, b = map(int, match.groups())
11                 log_entries.append((tx, ty, (r, g, b)))
12  max_tx = max(entry[0] for entry in log_entries) + 1
13  max_ty = max(entry[1] for entry in log_entries) + 1
14  img = Image.new('RGB', (max_tx, max_ty), (255, 255, 255))
15  for tx, ty, color in log_entries:
16      img.putpixel((tx, ty), color)
17  img.save('reconstructed_image.png')
18  img.show()
```

Diperoleh reconstructed_image.png sebagai berikut:



(OMAGAA WE GOT BRBRBR PATAPIM IN FITUKSW2025 BEFORE GTA6????)

Lalu scan Barcode tersebut sehingga diperoleh flag: `FITUKSW{r3c0d3_th3_34rth_1s_3451}`



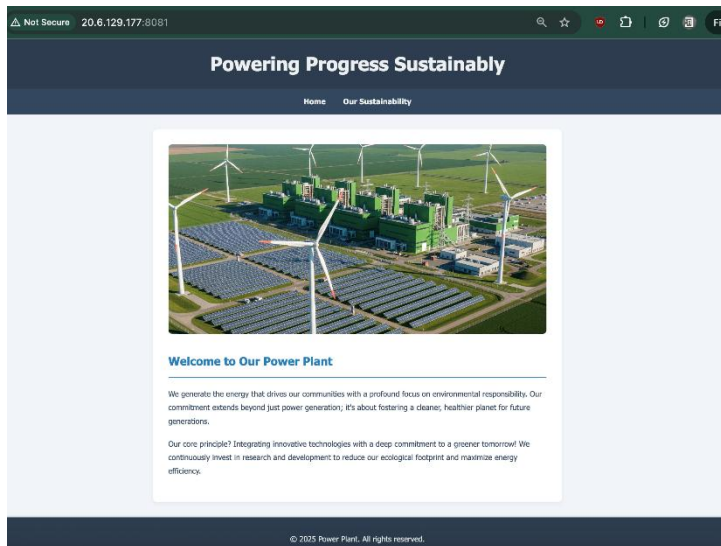
WEB

Power Plant 50

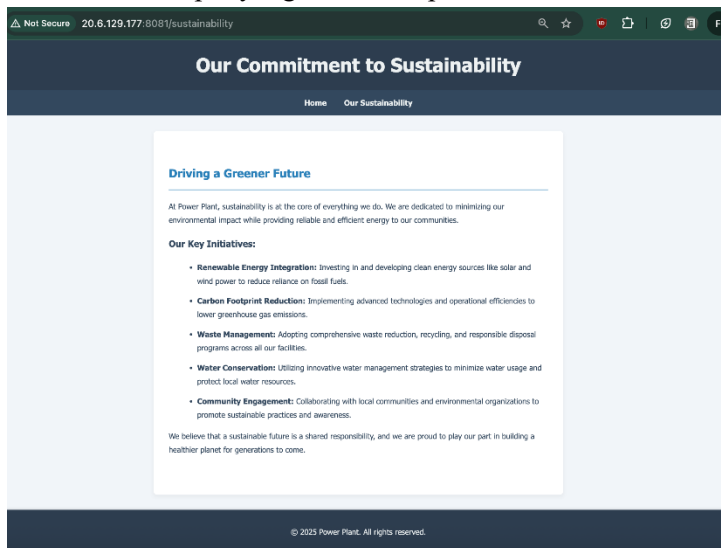
This power plant's website is open for public viewing, but perhaps they've been a little too open with certain configurations.

<http://20.6.129.177:8081/>

Kita coba cek halaman utama



Tidak terlihat input yang bisa di eksploitasi, kita ke halaman “our sustainability”



Websitenya simpel, di source code webnya juga tidak terlihat apa-apa.

Lalu kita coba cek robots.txt <http://20.6.129.177:8081/robots.txt>

tertampil “User-agent: * Disallow: /secret_code.txt”

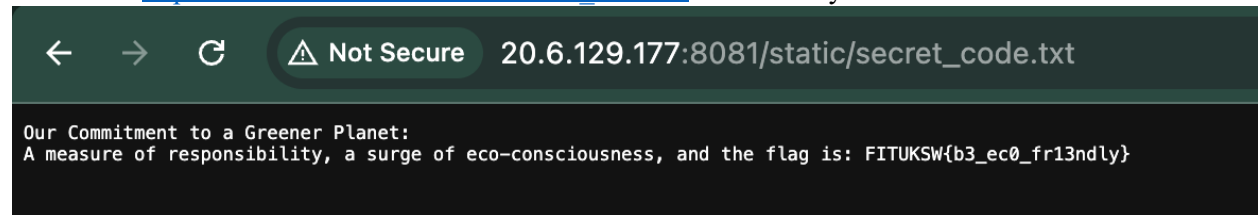
dapat clue, lalu kami langsung mencoba mengakses direktori-direktori yang kemungkinan menyimpan “secret_code.txt”:

http://20.6.129.177:8081/secret_code.txt

http://20.6.129.177:8081/static/secret_code.txt

http://20.6.129.177:8081/static/images/secret_code.txt

Ketemu di http://20.6.129.177:8081/static/secret_code.txt berikut isinya:



Flag: `FITUKSW{b3_ec0_fr13ndly}`

WEB

Wildlife Tracker 200

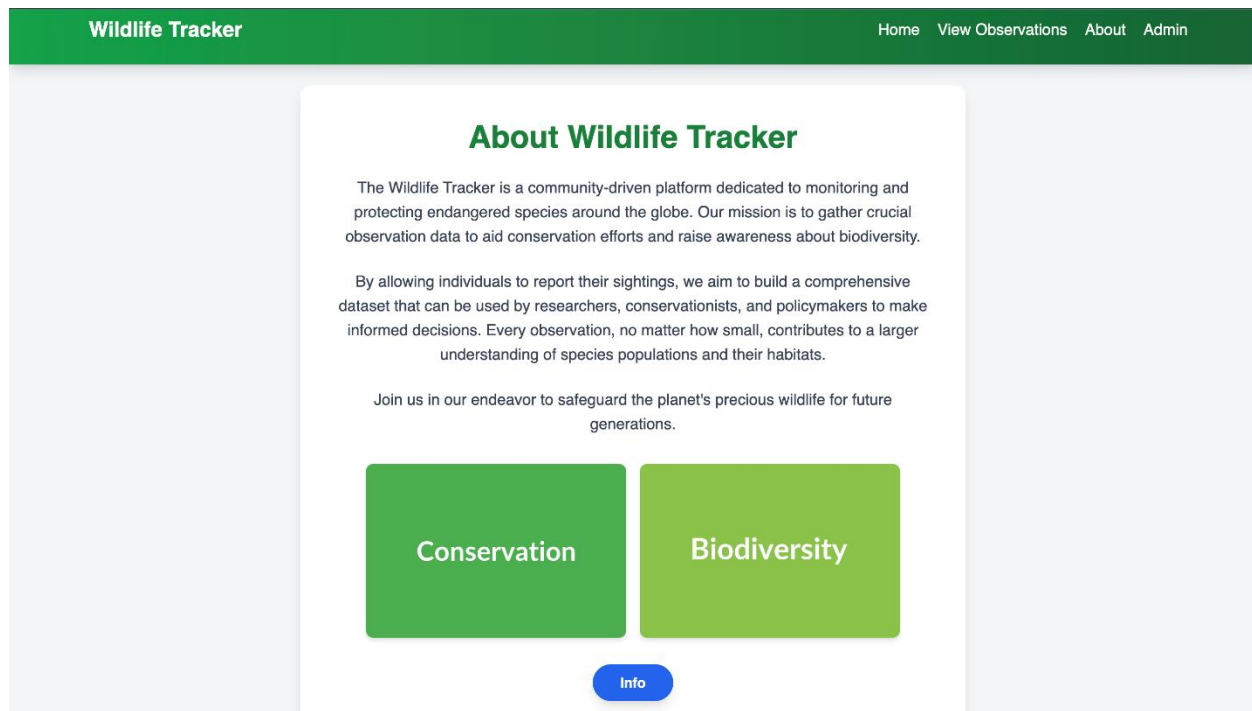
The "Wildlife Tracker" promises to help keep tabs on various species. However, every good system has its blind spots, and this one might be no exception. Can you exploit its nuances and gain unauthorized access to its deeper operations?

<http://134.209.102.23:8082/>

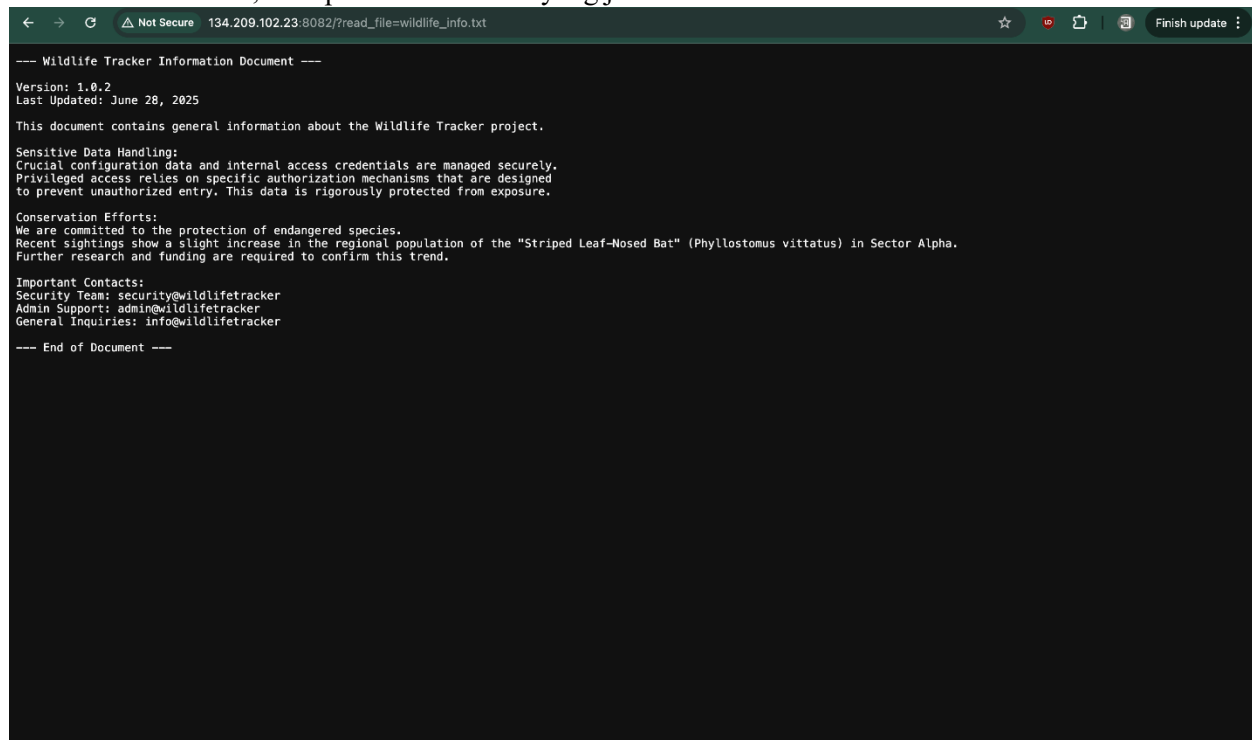
Di halaman utama, terdapat input yang kemungkinan menggunakan template engine, kita coba beberapa server side template injection

The screenshot shows a web browser window with the address bar displaying "Not Secure 134.209.102.23:8082". The page title is "Wildlife Tracker". The navigation bar includes links for "Home", "View Observations", "About", and "Admin". The main content area is titled "Track Endangered Species" and contains the following text: "Share your valuable observations about endangered wildlife and help us protect them." Below this is a section labeled "Your Observation:" with a text input field containing the placeholder text "Describe what you observed (e.g., species, location, behavior)..." and a green "Submit Observation" button. Below the submit button is a section labeled "Your Recent Observation:" with a light green box containing the placeholder text "{{ 7*7 }}". At the bottom of the page, a note states: "Note: Your observation is displayed as submitted. Please ensure content is appropriate."

Ternyata tidak ada yang berhasil, lalu kita coba ke halaman lain

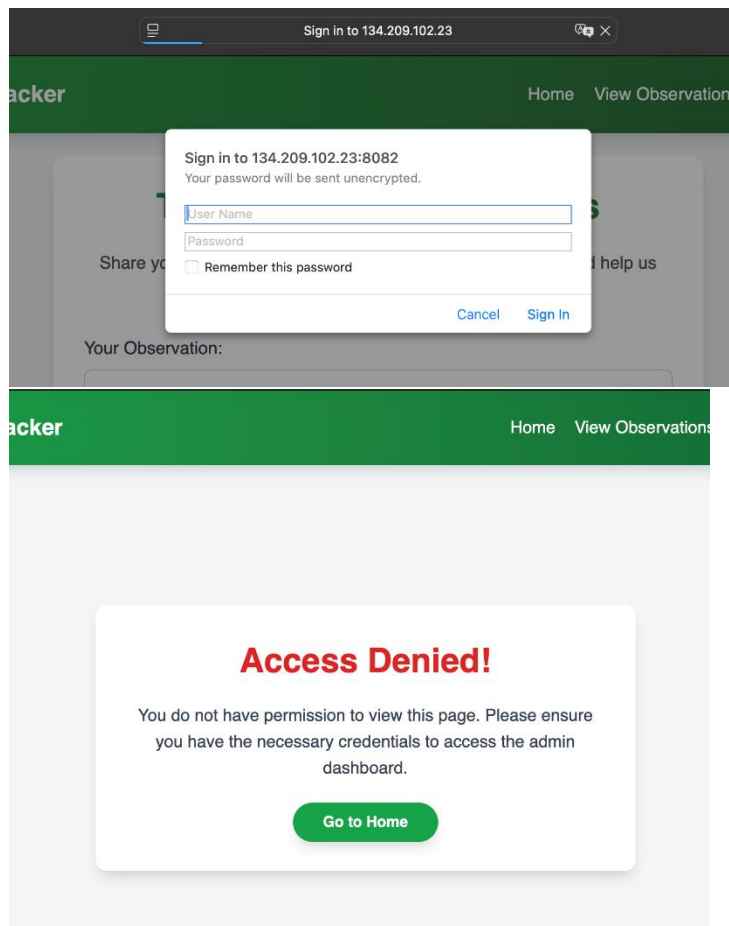


Di halaman “About”, terdapat tombol “Info” yang jika di klik:



muncul halaman ini. Ternyata web ini mengizinkan user untuk membaca isi direktori website dengan `?read_file=`

Dari sini kita bisa mulai exploitasi. Tapi sebelumnya saya cek halaman “Admin”:



Web ini menggunakan http Basic-Auth yang sayangnya, tidak bisa kita exploitasi menggunakan injection. Dari sini, saya mulai mencoba mencari flag atau password menggunakan bug read file yang kita sudah temukan.

1. http://134.209.102.23:8082/?read_file=../../../../../../../../etc/passwd

Biasanya ini direktori menyimpan file password di unix. Sayangnya, website dapat mendeteksi path traversal “Access denied: Path traversal detected!”

2. http://134.209.102.23:8082/?read_file=app.py

Didapat sourcode website mentahannya. Berikut sorotan yang penting:

```
app.config['SECRET_KEY'] = "wildlife-2025-fit-challenge-secret"
token = request.cookies.get('admin_token')
payload = jwt.decode(token, app.config['SECRET_KEY'], algorithms=['HS256'])
if payload['role']=='admin' and payload['authorized']:
    # show flag
```

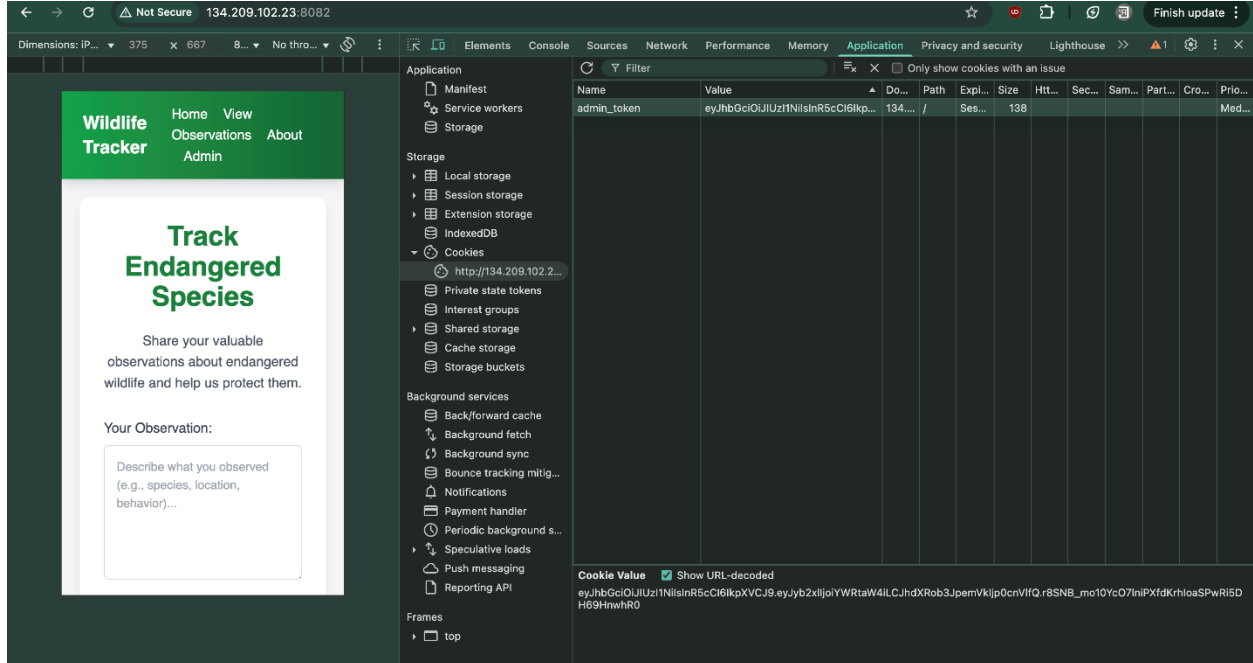
Dari logic kodingan ini, ditemukan bahwa website ini memakai admin cek berbasis JWT. Lalu akan kita buat JWT kita sendiri dengan secret key dari source code app.py

```
import jwt

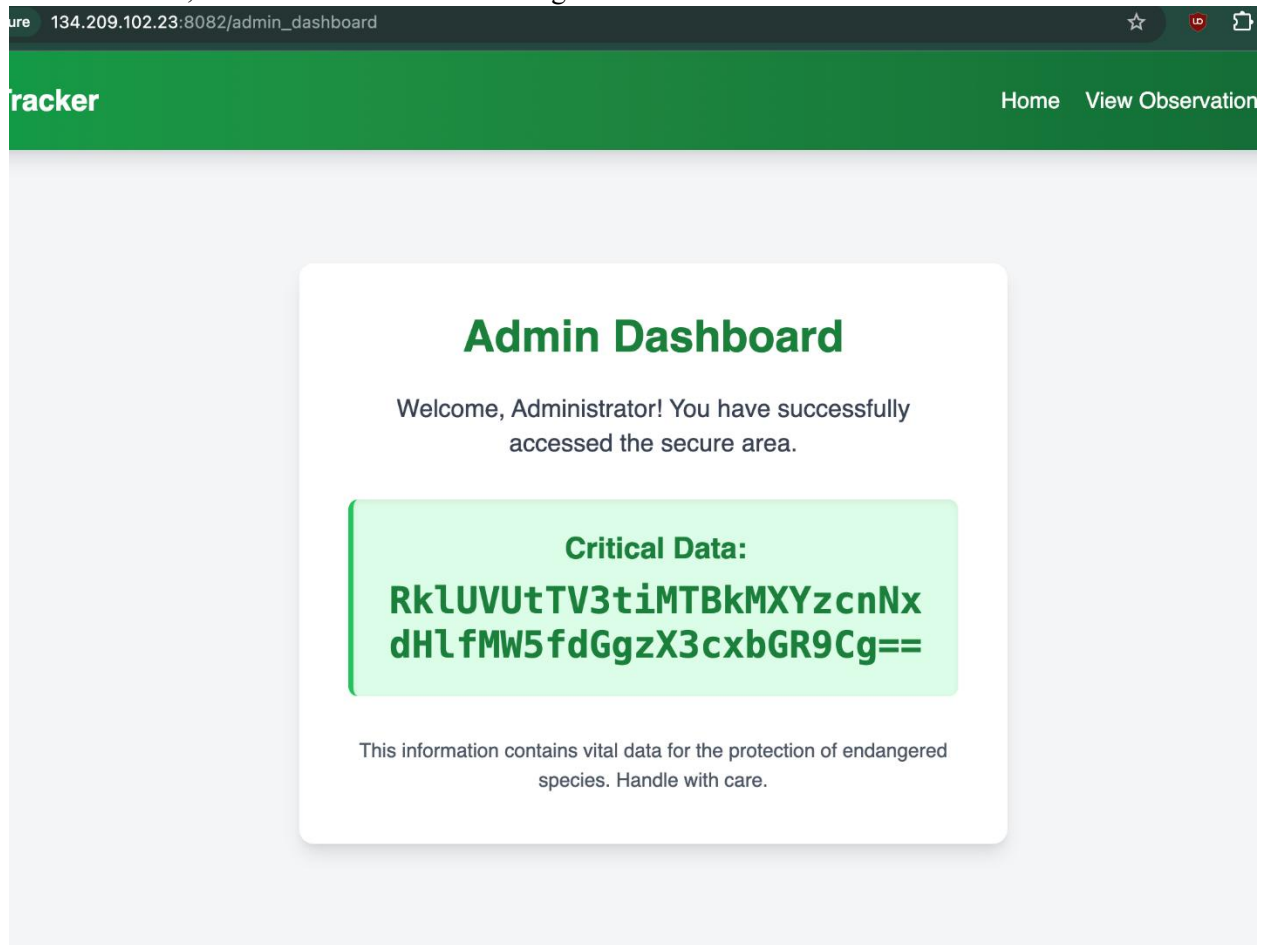
secret = "wildlife-2025-fit-challenge-secret"
payload = {
    "role": "admin",
    "authorized": True
}
token = jwt.encode(payload, secret, algorithm="HS256")
print(token)
```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoieWRTaW4iLCJhdXRob3JpemVkIjp0cnVlQ. r8SNB_mo10Yc07lniPXfdKrhIoaSPwRi5DH69HnwhR0

Lalu kita inject JWT dengan menambahkan cookie



setelah refresh, kita masuk ke dashboard sebagai admin



Didapat

RklUVUftTV3tiMTBkMXYZcnNxdHlfMW5fdGgzX3cxbGR9Cg==

Yang setelah di decode dari Base64, didapat flag:

FITUKSW{b10d1v3rsqty_1n_th3_w1ld}

Stegano

Ez-Stegano

150

Ada sebuah file EASY.jpg dimana file tersebut tersimpan file .txt

Gunakan steghide untuk mengecek apakah ada file tersembunyi, jika diminta password, enter saja:

```
hellyeah04@LAPTOP-NJHI85P5:~$ steghide extract -sf EASY.jpg
Enter passphrase:
wrote extracted data to "secret.txt".
hellyeah04@LAPTOP-NJHI85P5:~$
```

Diperoleh file secret.txt, selanjutnya buka file tersebut:

```
hellyeah04@LAPTOP-NJHI85P5:~$ cat secret.txt
FITUKSW{FT1K4ub3r4ada}
```

Maka diperoleh flag: FITUKSW{FT1K4ub3r4ada}

Stegano

Med - Stegano

150

Ada sebuah file MEDIUM.jpg dimana file tersebut tersimpan file .txt

Gunakan steghide untuk mengecek apakah ada file tersembunyi, jika diminta password, enter saja:

```
hellyeah04@LAPTOP-NJHI85P5:~$ steghide extract -sf MEDIUM.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
hellyeah04@LAPTOP-NJHI85P5:~$
```

Alright gang, now we cooked.

Selanjutnya kita akan menggunakan stegseek dan rockyou.txt untuk mencari passwordnya:

```
hellyeah04@LAPTOP-NJHI85P5:~$ stegseek MEDIUM.jpg /usr/share/wordlists/seclists/Passwords/Common-Credentials/10-million
password-list-top-1000000.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[!] error: could not open the wordlist "/usr/share/wordlists/seclists/Passwords/Common-Credentials/10-million-password-l
ist-top-1000000.txt".
hellyeah04@LAPTOP-NJHI85P5:~$ wget https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt
stegseek MEDIUM.jpg /rockyou.txt
--2025-07-05 19:01:36-- https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/97553311/d4f580f8-6b49-11e7-8f70-
7f460f85ab3a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250705%2Fus-east-1%2Ffs3%2Faws4
_request%2X-Amz-Date=20250705T120136Z&X-Amz-Expires=1800&X-Amz-Signature=f857dd7585bc4ff1ca5d87dea525be9cac09199cfdcde731
913303dc27f218aa&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Drockyou.txt&response-c
ontent-type=application%2Foctet-stream [following]
--2025-07-05 19:01:36-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/97553311/d4f580f8-
6b49-11e7-8f70-7f460f85ab3a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250705%2Fus-eas
t-1%2Ffs3%2Faws4_request%2X-Amz-Date=20250705T120136Z&X-Amz-Expires=1800&X-Amz-Signature=f857dd7585bc4ff1ca5d87dea525be9ca
c09199cfdcde731913303dc27f218aa&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Drockyou
.txt&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109
.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 139921497 (133M) [application/octet-stream]
Saving to: 'rockyou.txt.1'

rockyou.txt.1      58%[>          1   7.11M   1.51MB/s   eta 2m 20s
```

Selanjutnya tunggu hingga selesai menginstall dan memperoleh pass:

```
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek
```

```
[i] Found passphrase: "123"  
[i] Original filename: "secret.txt".  
[i] Extracting to "MEDIUM.jpg.out".  
the file "MEDIUM.jpg.out" does already exist. overwrite ? (y/n)  
y  
hellyeah04@LAPTOP-NJHI85P5:~$  
hellyeah04@LAPTOP-NJHI85P5:~$
```

Maka diperoleh passnya, lalu mulai steghide lagi dan masukkan password serta buka file secret.txt:

```
hellyeah04@LAPTOP-NJHI85P5:~$ steghide extract -sf MEDIUM.jpg  
Enter passphrase:  
wrote extracted data to "secret.txt".  
hellyeah04@LAPTOP-NJHI85P5:~$ cat secret.txt  
FITUKSW{D4r4hb1ruFt1}
```

Diperoleh flag: FITUKSW{D4r4hb1ruFt1}

Forensics

Secret File 200



Tobi, seorang pemain crypto, dia pengusaha dan mempunyai lambo warna ungu.

Suatu hari, dia pengen menghapus file-file yang ngga dibutuhin di PC nya, tapi Tobi ngga sengaja ngehapus file yang berisi passphrase wallet yang berisi 5 BTC.

Bisakah kamu menemukan file itu? [Download Soal](#)

Author : Mas Raya

Diberikan file:

| Name | Last modified | File size |
|--|---------------|-----------|
|  Tobi_Secret_File.E01 | Mar 14, 2025 | 2 MB |
|  Tobi_Secret_File.E01.txt | Mar 14, 2025 | 1 KB |

Kita buka di shell dan coba cek hashnya

```
md5sum Tobi_Secret_File.E01 -> 8d38fe4ed34a595fb77983a6e1a89c7c
```

```
sha1sum Tobi_Secret_File.E01 -> 2bcc374e2021afb18869ab2b74bed0f5d9e082ba
```

Kedua hash ini cocok dengan yang ada di file .txt

lalu kita download tools yang kita perlu yaitu libewf-tools dan sleuthkit.

```
ewfexport -t raw -f image -o 0 -S Tobi.raw Tobi_Secret_File.E01
```

Output:

```
ewfexport 20210414
```

```
Acquiry phase: "image"
```

```
Number of sectors to export: 49152
```

```
Exporting: Tobi.raw
```

```
Export completed.
```

Kita dapat Tobi.raw di folder.

Kita coba cari layout partitionnya

```
mmls Tobi.raw
```

Output:

DOS Partition Table

Offset Sector: 0

Units are in 512-byte sectors

| | Slot | Start | End | Length | Description |
|-----|-------|-------|-------|--------|-------------|
| 00: | ----- | 0 | 49151 | 49152 | Unallocated |

Lalu kita buat inode list dengan “fls -r -o 0 Tobi.raw > fls.txt”

Setelahnya:

```
grep -Ei "(pass|flag|wallet|fituksw)" fls.txt
```

Output:

```
d/d * 46: $Recycle.Bin/S-1-5-21-385157629-4095870897-1986556360-1001/passphrase.txt
```

Recover file yg di hapus

```
icat -o 0 Tobi.raw 46 > passphrase.txt
```

dan di dalam passphrase.txt, flagnya ditemukan yaitu:

```
FITUKSW{nice_step_for_better_forensic_master_on_2025_669534}
```


Forensics

Martin and the Humming Signal !

300

Martin tinggal sendirian di ujung gang, rumahnya penuh barang-barang aneh—dari jam dinding yang berputar mundur sampai radio tua yang selalu menyala, bahkan saat mati lampu.

Suatu malam, terdengar suara berdesis dari radionya. Martin bilang itu “pesan penting” yang dikirimkan entah dari siapa... entah dari mana.

Sebelum menghilang, Martin meninggalkan satu file rekaman yang katanya: “Dengerin baik-baik... mereka cuma bisa bicara lewat cara ini.” [Download](#) Sekarang rekaman itu ada padamu.

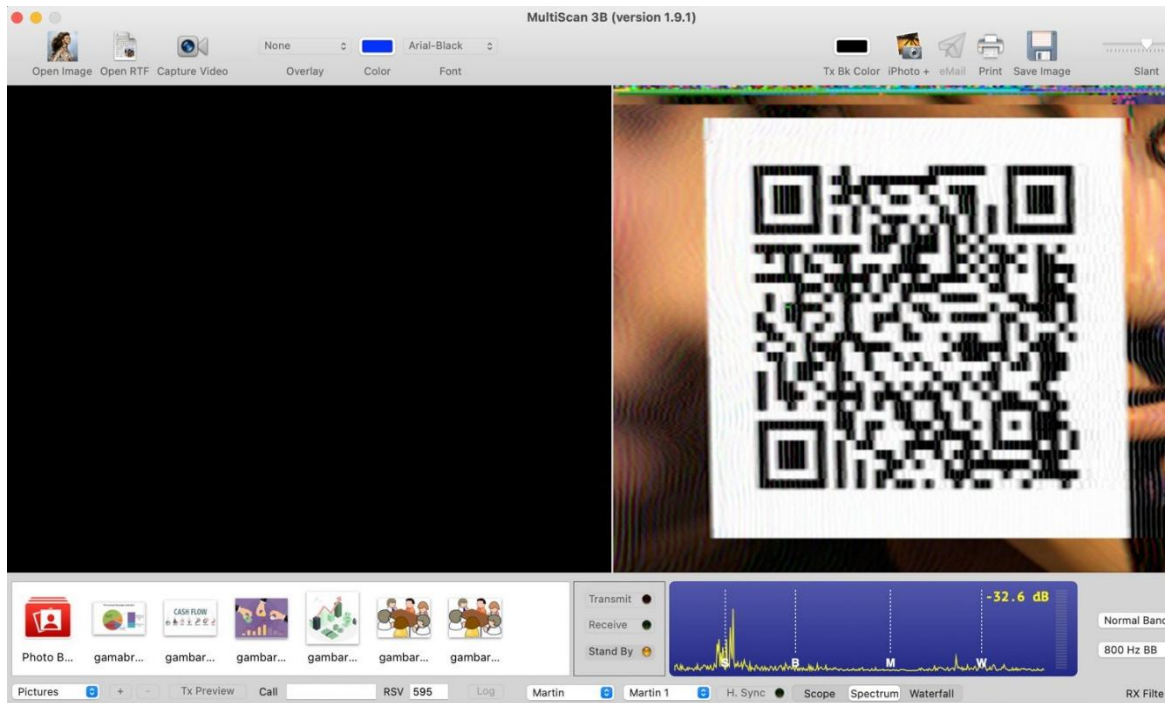
Author : Bebekk

Diberikan file audio hummingsignal.wav

Setelah mengecek tipe file dan steganografi sekilas, kami tidak menemukan hal – hal yang memicu perhatian. Dari tes suara dan melihat spectrogram file, kami melihat similaritas dengan transmisi audio sstv. Hal ini didukung juga dengan clue “Martin” di deskripsi soal. File audio ini berdurasi 1:55 (115 detik) yang cocok dengan durasi mode Martin 1.

| Family | Developer | Name | Color | Time | Lines |
|---------|---------------------------------|------|-----------------------|-------|--------------------------------|
| AVT | Ben Blish-Williams, AA7AS / AEA | 8 | BW or 1 of R, G, or B | 8 s | 128×128 |
| | | 16w | BW or 1 of R, G, or B | 16 s | 256×128 |
| | | 16h | BW or 1 of R, G, or B | 16 s | 128×256 |
| | | 32 | BW or 1 of R, G, or B | 32 s | 256×256 |
| | | 24 | RGB | 24 s | 128×128 |
| | | 48w | RGB | 48 s | 256×128 |
| | | 48h | RGB | 48 s | 128×256 |
| | | 104 | RGB | 96 s | 256×256 |
| Martin | Martin Emmerson - G3OQD | M1 | RGB | 114 s | 240 ¹ |
| | | M2 | RGB | 58 s | 240 ¹ |
| Robot | Robot SSTV | 8 | BW or 1 of R, G or B | 8 s | 120 |
| | | 12 | YUV | 12 s | 128 luma, 32/32 chroma × 120 |
| | | 24 | YUV | 24 s | 128 luma, 64/64 chroma × 120 |
| | | 32 | BW or 1 of R, G or B | 32 s | 256 × 240 |
| | | 36 | YUV | 36 s | 256 luma, 64/64 chroma × 240 |
| | | 72 | YUV | 72 s | 256 luma, 128/128 chroma × 240 |
| Scottie | Eddie Murphy - GM3SBC | S1 | RGB | 110 s | 240 ¹ |
| | | S2 | RGB | 71 s | 240 ¹ |
| | | DX | RGB | 269 s | 320 x 256 |

Selanjutnya, akan kita decode transmisi ini dengan software sstv. Saya gunakan MultiScan 3B. Setelah mengkonfigurasi modenya ke Martin 1, saya nyalakan audionya lewat smartphone saya dan hasil akhirnya sebagai berikut:



Isi QRnya adalah

RkIUUVUtTV3t0aGV5X3NpbmdfaW5fc3RhdGljX2FuZF9kcmVhbV9pbl9ub2lzZX0=

Yang ketika di decode dari Base64, menjadi wujud asli flag:

FITUKSW{they_sing_in_static_and_dream_in_noise}