Write UP CTF Lomba FESTI 2025

Challenge Misc

Cuman nulis ulang dan selesai

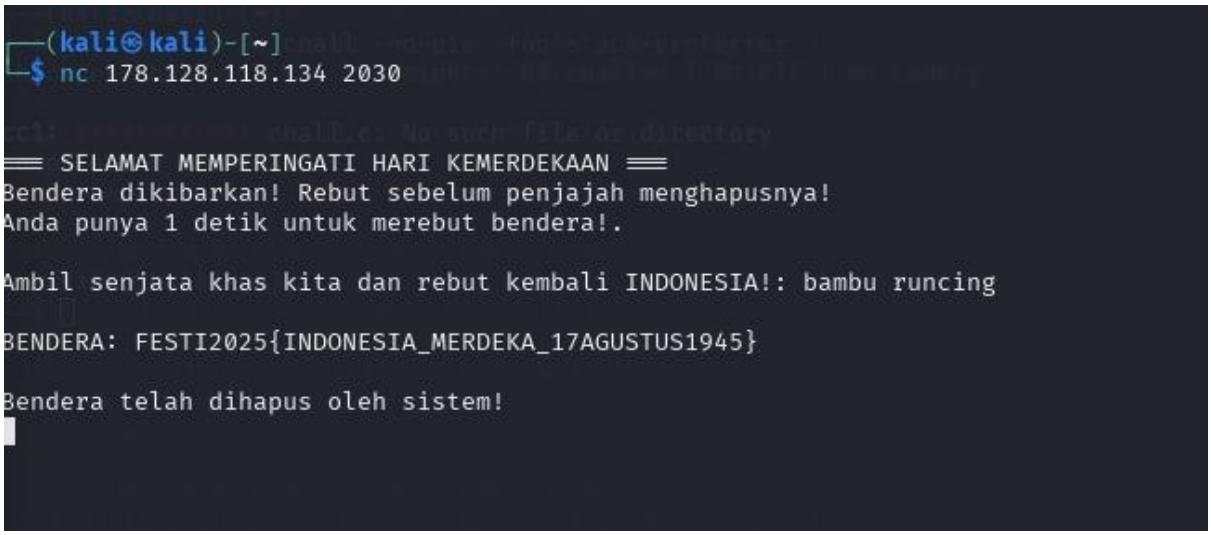
FLAG: FESTI2025{welcome}

Challenge PWN

Merdeka

Disini saya menghubungkan ke server dan lalu input bambu runcing (sebenarnya menebak sih soalnya kan ditanya nya senjata khas indonesia)

FLAG: FESTI2025{INDONESIA_MERDEKA_17AGUSTUS1945}



Challenge Cryptography

1. XOR it Again

Dari XOR.txt itu isinya heksadesimal ciphertext sepanjang 28 byte :

1f 30 30 27 34 17 5a 6b 40 18 0b 12 57 35 38 01 17 12 1e 4e 35 30 06 3c 15 08 4b 17

Challenge ini saya pakai google collab dan begini detailnya:

```
[2]
from pathlib import Path

hex_bytes = Path("XOR.txt").read_text().strip().split()
cipher = bytes.fromhex("".join(hex_bytes))
print("Cipher length:", len(cipher), "bytes")
print("Cipher (hex):", cipher.hex())
```

```
Cipher length: 28 bytes
Cipher (hex): 1f30302734175a6b40180b125735380117121e4e3530063c15084b17
```

```
[3]
known_pt = b"FESTI2025{"
assert len(known_pt) == 10
```

```
[4] key_stream_part = bytes([c ^ p for c, p in zip(cipher, known_pt)])
print("Key bytes (partial):", key_stream_part.hex())
```

```
Key bytes (partial): 597563737d256a597563
```

```
[5]
def candidate_key(cipher_part):
    for L in range(2, 16):
        chunk = key_stream_part[:L]
        if all(key_stream_part[i] == chunk[i % L] for i in range(len(key_stream_par
            return chunk
    return None

key = candidate_key(cipher_part=key_stream_part)
print("Recovered key (hex):", key.hex(), "| length =", len(key))
```

```
Recovered key (hex): 597563737d256a | length = 7
```

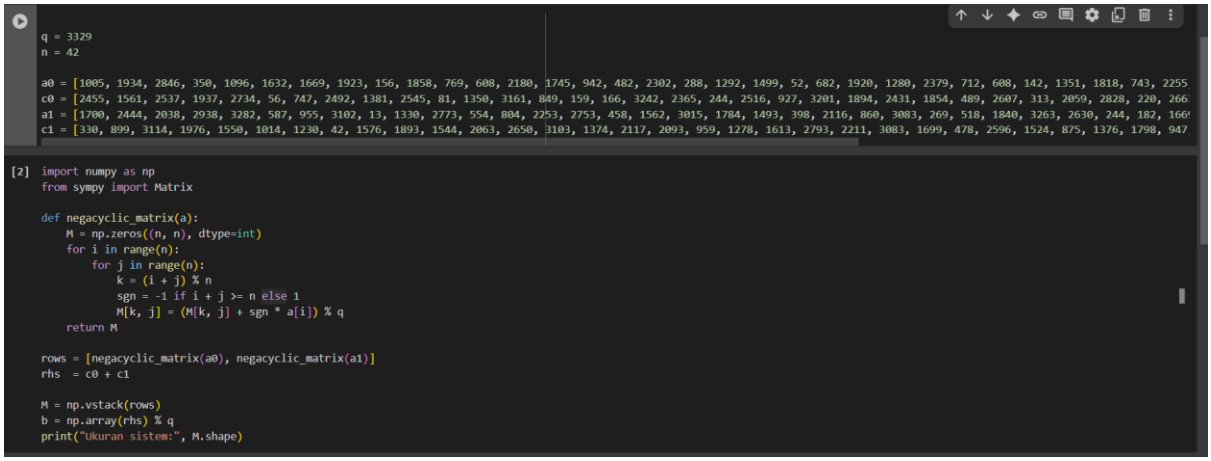
```
[6]
full_key_stream = (key * (len(cipher) // len(key) + 1))[:len(cipher)]
plaintext = bytes([c ^ k for c, k in zip(cipher, full_key_stream)])
print("Plaintext:", plaintext.decode())
```

```
Plaintext: FESTI2025{xor_attack_is_fun}
```

FLAG: FESTI2025{xor_attack_is_fun}

2. Polynomian Trap II

Saya pakai google collab lagi untuk dapat flagnya:



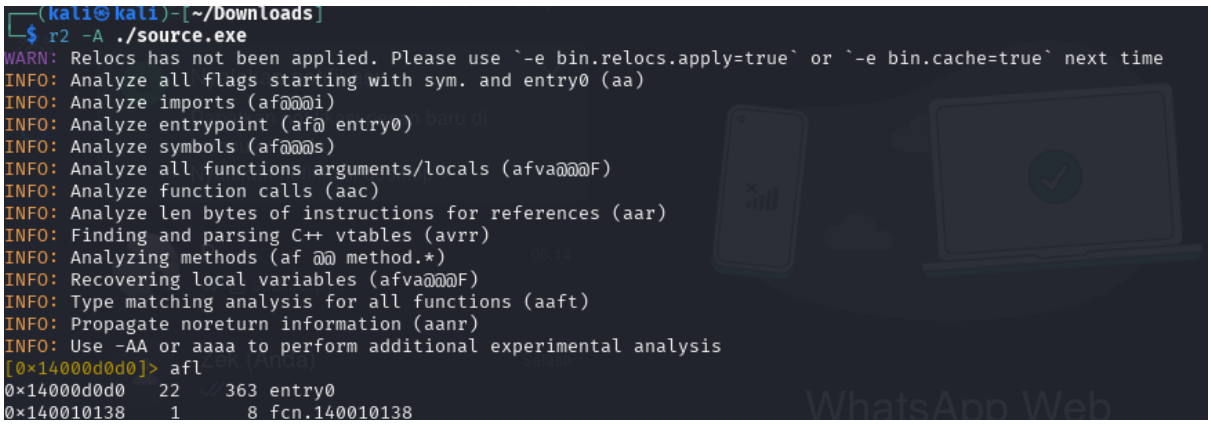
```
q = 3329
n = 42

a0 = [1005, 1934, 2846, 350, 1096, 1632, 1669, 1923, 156, 1858, 769, 608, 2180, 1745, 942, 482, 2302, 288, 1292, 1499, 52, 682, 1920, 1280, 2379, 712, 608, 142, 1351, 1818, 743, 2255
c0 = [2455, 1561, 2537, 1937, 2734, 56, 747, 2492, 1381, 2545, 81, 1350, 3161, 849, 159, 166, 3242, 2365, 244, 2516, 927, 3201, 1894, 2431, 1854, 489, 2607, 313, 2059, 2828, 220, 266
a1 = [1700, 2444, 2038, 2938, 3282, 587, 955, 3102, 13, 1330, 2773, 554, 804, 2253, 2753, 458, 1562, 3015, 1784, 1493, 398, 2116, 860, 3083, 269, 518, 1840, 3263, 2630, 244, 182, 166
c1 = [330, 899, 3114, 1976, 1550, 1014, 1230, 42, 1576, 1893, 1544, 2063, 2650, 3103, 1374, 2117, 2093, 959, 1278, 1613, 2793, 2211, 3083, 1699, 478, 2596, 1524, 875, 1376, 1798, 947

[2] import numpy as np
    from sympy import Matrix

    def negacyclic_matrix(a):
        M = np.zeros((n, n), dtype=int)
        for i in range(n):
            for j in range(n):
                k = (i + j) % n
                sgn = -1 if i + j >= n else 1
                M[k, j] = (M[k, j] + sgn * a[i]) % q
        return M

    rows = [negacyclic_matrix(a0), negacyclic_matrix(a1)]
    rhs  = c0 + c1

    M = np.vstack(rows)
    b = np.array(rhs) % q
    print("Ukuran sistem:", M.shape)
```

FLAG: FESTI2025{Th15_1s_Th3_L393nD4Ry_p0lyHard!}

Challenge Reverse Engineering

Obviously

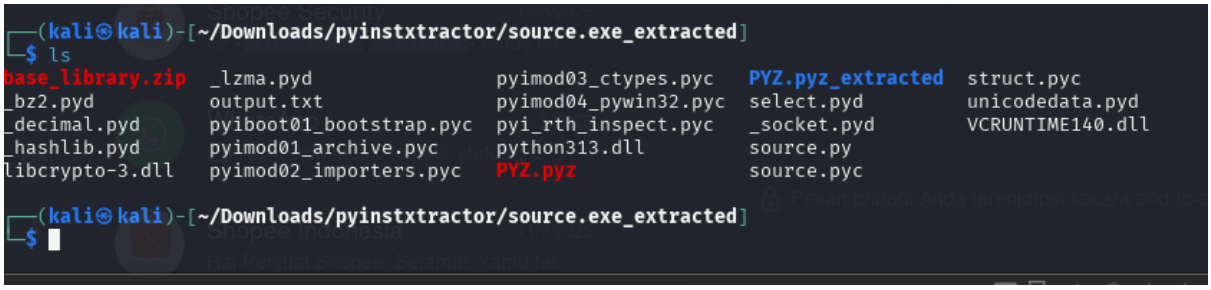
Disini saya pake Radare2 untuk buka source.exe



```
┌──(kali㉿kali)-[~/Downloads]
└─$ r2 -A ./source.exe
WARN: Relocs has not been applied. Please use `-e bin.relocs.apply=true` or `-e bin.cache=true` next time
INFO: Analyze all flags starting with sym. and entry0 (aa)
INFO: Analyze imports (af@@@i)
INFO: Analyze entrypoint (af@ entry0)
INFO: Analyze symbols (af@@@s)
INFO: Analyze all functions arguments/locals (afva@@@F)
INFO: Analyze function calls (aac)
INFO: Analyze len bytes of instructions for references (aar)
INFO: Finding and parsing C++ vtables (avrr)
INFO: Analyzing methods (af @@ method.*)
INFO: Recovering local variables (afva@@@F)
INFO: Type matching analysis for all functions (aaft)
INFO: Propagate noreturn information (aanr)
INFO: Use -AA or aaaa to perform additional experimental analysis
[0×14000d0d0]> afl
0×14000d0d0    22    363 entry0
0×140010138     1      8 fcn.140010138
```

Selanjutnya saya menginstall tool Pyinstxtractor guna mempermudah mencari flagnya (karena sudah stuck kurleb 30 menit)

Nah singkatnya ekstraksi berhasil dan saya langsung ekstraksi manual source.pyc nya dengan skrip python.



```
┌──(kali㉿kali)-[~/Downloads/pyinstxtractor/source.exe_extracted]
└─$ ls
base_library.zip  _lzma.pyd              pyimod03_ctypes.pyc   PYZ.pyz_extracted  struct.pyc
_bz2.pyd          output.txt             pyimod04_pywin32.pyc  select.pyd         unicodedata.pyd
_decimal.pyd      pyiboot01_bootstrap.pyc pyi_rth_inspect.pyc  _socket.pyd        VCRUNTIME140.dll
_hashlib.pyd      pyimod01_archive.pyc   python313.dll         source.py
libcrypto-3.dll   pyimod02_importers.pyc PYZ.pyz               source.pyc

┌──(kali㉿kali)-[~/Downloads/pyinstxtractor/source.exe_extracted]
└─$
```

Skrip Python

```
 GNU nano 8.3                                ekstra.py
import re
with open("source.pyc", "rb") as f:
    data = f.read()

strings = re.findall(rb'[ -~]{6,}', data)
for s in strings:
    if b'FESTI2025' in s:
        print(s)
```

Lalu saya jalankan dan ketemu flagnya:

```
┌──(kali㉿kali)-[~/Downloads/pyinstxtractor/source.exe_extracted]
└─$ python3 ekstra.py
b'z%Debug: FESTI2025{H1de_1n_Pl4iN_51ght}z'
```

Maka flagnya: FESTI2025{H1de_1n_Pl4iN_51ght}