WriteUp Technofair CTF
CTF Demi IU

MBEERRR
ChaO
AnehMan

Cryptography

1. Legend Said

a. Executive Summary

Legend said only DUKUN can solve this

nc 103.152.242.172 9070

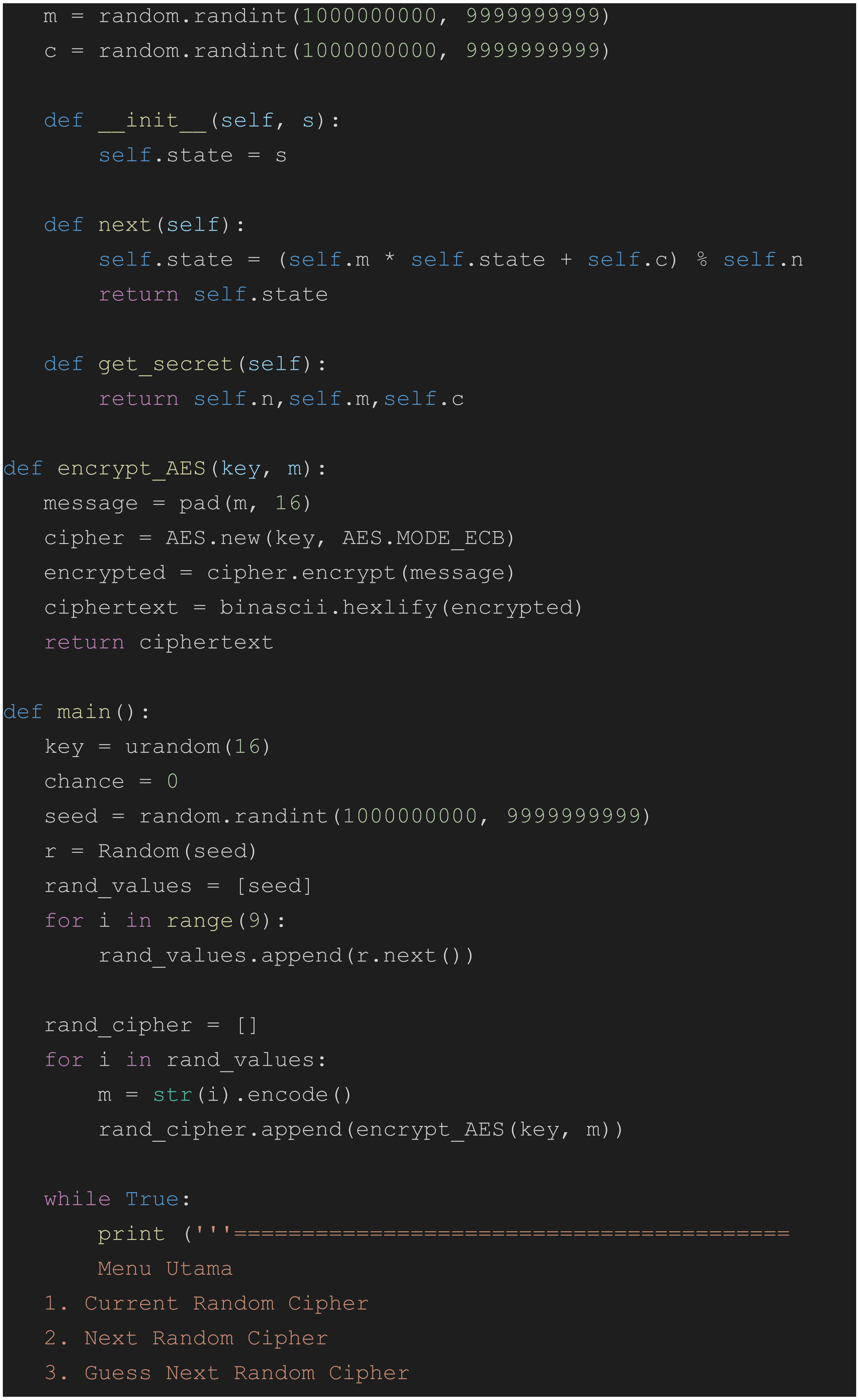
format flag: technofair{}

author: twistbil

b. Technical Report

Diberikan file server.py Berikut penampakannya

```
#!/usr/bin/python3
from secret import flag
from Crypto.Cipher import AES
import sys, binascii
from Crypto.Util.Padding import pad
from os import urandom
import random

class Unbuffered(object):
    def __init__(self, stream):
        self.stream = stream
    def write(self, data):
        self.stream.write(data)
        self.stream.flush()
    def writelines(self, datas):
        self.stream.writelines(datas)
        self.stream.flush()
    def __getattr__(self, attr):
        return getattr(self.stream, attr)

sys.stdout = Unbuffered(sys.stdout)

class Random():
    n = random.randint(1000000000, 9999999999)
```



```
    m = random.randint(1000000000, 9999999999)
    c = random.randint(1000000000, 9999999999)

    def __init__(self, s):
        self.state = s

    def next(self):
        self.state = (self.m * self.state + self.c) % self.n
        return self.state

    def get_secret(self):
        return self.n,self.m,self.c

def encrypt_AES(key, m):
    message = pad(m, 16)
    cipher = AES.new(key, AES.MODE_ECB)
    encrypted = cipher.encrypt(message)
    ciphertext = binascii.hexlify(encrypted)
    return ciphertext

def main():
    key = urandom(16)
    chance = 0
    seed = random.randint(1000000000, 9999999999)
    r = Random(seed)
    rand_values = [seed]
    for i in range(9):
        rand_values.append(r.next())

    rand_cipher = []
    for i in rand_values:
        m = str(i).encode()
        rand_cipher.append(encrypt_AES(key, m))

    while True:
        print ('''=========================================
        Menu Utama
    1. Current Random Cipher
    2. Next Random Cipher
    3. Guess Next Random Cipher
```

```
   4. Encrypt Something
   5. Panggil Dukun
   6. Exit
=========================================''')

        print(r.get_secret())
        pilihan = input("Masukan Pilihan: ")
        print('----------------------------------------')
        if pilihan == '1':
                        print('Current  Random  Ciphertext:  ',
rand_cipher[chance])

        elif pilihan == '2':
            chance += 1
            if chance >= 9:
                print('Auu ah cape :(')
                exit()
            else:
                            print('Next  Random  Ciphertext:  ',
rand_cipher[chance])

        elif pilihan == '3':
            guess = input('Masukkan Prediksi Next Cipher: ')
            guess = guess.encode()
            if guess == rand_cipher[chance+1]:
                print('=========== CONGRATSSS ============')
                print('this is your flag:', flag)
                exit()
            else:
                print('Salah bwangg')

        elif pilihan == '4':
            message = str(input('Plaintext: '))
            cipher = encrypt_AES(key, message.encode())
            print('Ciphertext: ', cipher)

        elif pilihan == '5':
             print('Dukun will help you with the current random
cipher')
            message = input('Pesan untuk dukun: ')
```

```
                                        cipher    =    encrypt_AES(key,
(message+str(rand_values[chance])).encode())
            print('Balasan dari dukun: ', cipher)

        elif pilihan == '6':
            print('Babayy ~~~~')
            exit()

        else:
            print(">:(")
            exit()

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

Jadi kita harus menebak angka selanjutnya yang di-generate oleh class Random, tapi angka-angka sebelumnya di-encrypt dengan AES ECB (key random tiap akses service). Untungnya, pada pilihan panggil dukun, inputan kita di-append dengan hasil random. Jadi kita perlu menginputkan satu karakter sebanyak 15 kali. Nanti akan jadi seperti ini

```
encrypt(“AAAAAAAAAAAAAAAX”,KEY)
```

Karakter “X” kita brute dengan karakter 0-9. Ketika karakter X ditemukan, offset dikurangi 1 (dari 15 jadi 14). Nanti akan jadi seperti ini

```
encrypt(“AAAAAAAAAAAAAA1X”,KEY)
```

Hal ini terus dilakukan sampai seluruh angka bisa tertebak. Cara ini sama seperti challenge Cryptopals set 2 challenge 12. Hanya saja kita lakukan ini sebanyak kurang lebih 5-7 kali.

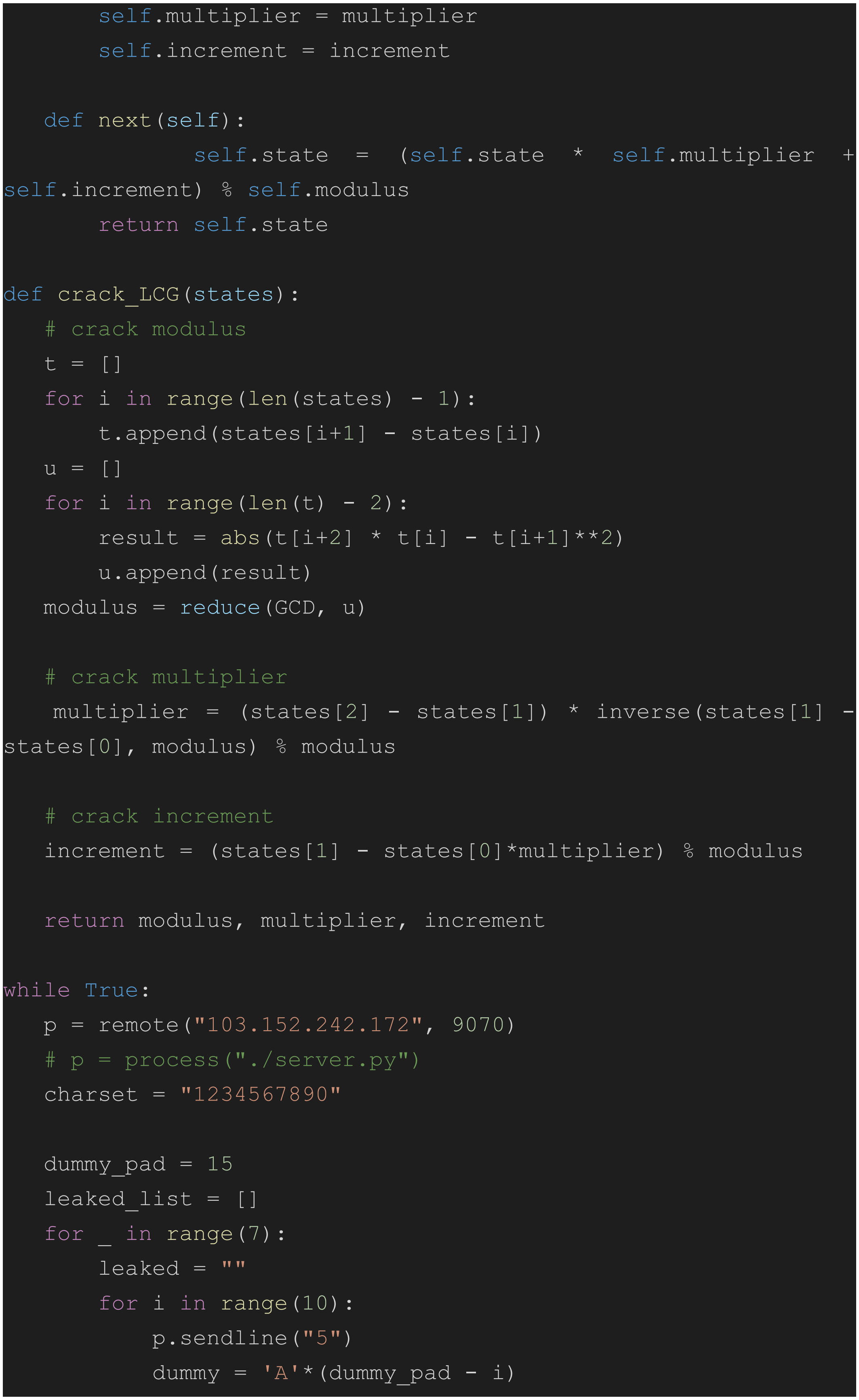
Karena panjang angka selalu sama (10 karakter), kita tidak perlu mengurangi offset sampai 0. Berikut scriptnya

```
p = remote("103.152.242.172", 9070)
# p = process("./server.py")
charset = "1234567890"

dummy_pad = 15
leaked_list = []
```

```
for _ in range(7):
    leaked = ""
    for i in range(10):
        p.sendline("5")
        dummy = 'A'*(dummy_pad - i)
        p.sendline(dummy)
        p.recvuntil("Balasan dari dukun:  b'")
        reference = p.recvline()[:-2]
        reference_block = [reference[i:i+32] for i in range(0,len(reference),32)]
        for c in charset:
            guess = dummy + leaked + c
            p.sendline("5")
            p.sendline(guess)
            p.recvuntil("Balasan dari dukun:  b'")
            result = p.recvline()[:-2]
            result_block = [result[i:i+32] for i in range(0,len(result),32)]
            if reference_block[0] == result_block[0]:
                leaked += c
                break
    leaked_list.append(int(leaked))
    # print(leaked_list)
    if _ != 6:
        p.sendline("2")
```

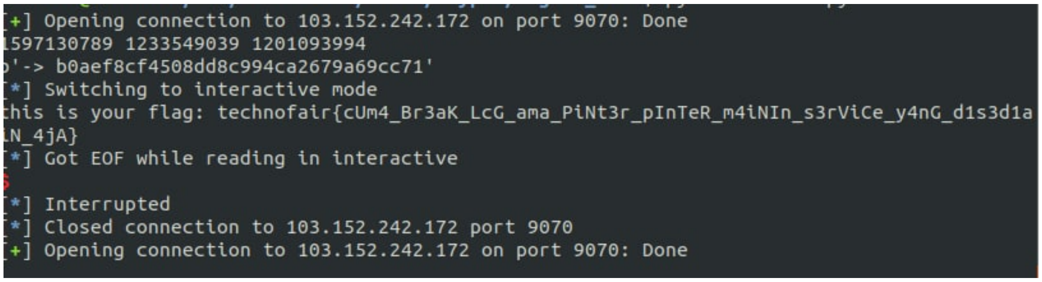
Untuk crack Random, uhh… tbh kami tidak terlalu mengerti cara kerjanya, tapi kami tau itu LCG, lalu mencoba cara ini. Hanya saja, cara tersebut akan tidak berhasil apabila multiplier atau increment lebih besar dari modulus. Jadi kita perlu reconnect apabila tebakan salah. Berikut full scriptnya

```
from pwn import *
from Crypto.Cipher import AES
from Crypto.Util.number import *
from functools import reduce

class LCG:
    def __init__(self, state, modulus, multiplier, increment):
        self.state = state
        self.modulus = modulus
```



```
        self.multiplier = multiplier
        self.increment = increment

    def next(self):
        self.state = (self.state * self.multiplier +
self.increment) % self.modulus
        return self.state

def crack_LCG(states):
    # crack modulus
    t = []
    for i in range(len(states) - 1):
        t.append(states[i+1] - states[i])
    u = []
    for i in range(len(t) - 2):
        result = abs(t[i+2] * t[i] - t[i+1]**2)
        u.append(result)
    modulus = reduce(GCD, u)

    # crack multiplier
    multiplier = (states[2] - states[1]) * inverse(states[1] -
states[0], modulus) % modulus

    # crack increment
    increment = (states[1] - states[0]*multiplier) % modulus

    return modulus, multiplier, increment

while True:
    p = remote("103.152.242.172", 9070)
    # p = process("./server.py")
    charset = "1234567890"

    dummy_pad = 15
    leaked_list = []
    for _ in range(7):
        leaked = ""
        for i in range(10):
            p.sendline("5")
            dummy = 'A'*(dummy_pad - i)
            p.sendline(dummy)
            p.recvuntil("Balasan dari dukun:  b'")
            reference = p.recvline()[:-2]
            reference_block = [reference[i:i+32] for i in range(0,len(reference),32)]
            for c in charset:
                guess = dummy + leaked + c
                p.sendline("5")
                p.sendline(guess)
                p.recvuntil("Balasan dari dukun:  b'")
                result = p.recvline()[:-2]
                result_block = [result[i:i+32] for i in range(0,len(result),32)]
                if reference_block[0] == result_block[0]:
                    leaked += c
                    break
        leaked_list.append(int(leaked))
        # print(leaked_list)
        if _ != 6:
            p.sendline("2")

    n,m,c = crack_LCG(leaked_list)
    print(n,m,c)

    r = LCG(leaked_list[-1], n, m, c)
    plaintext = r.next()
    p.sendline("4")
    p.sendline(str(plaintext))
    p.recvuntil("Ciphertext:  b'")
    result = p.recvline()[:-2]
    print(b"-> " + result)
    p.sendline("3")
    p.sendline(result)
    p.recvuntil("Next Cipher: ")
    flag = p.recvline()
    if b"CONGRATSSS" in flag:
        p.interactive()
    p.close()
```

Hasil:



```
[+] Opening connection to 103.152.242.172 on port 9070: Done
1597130789 1233549039 1201093994
'-> b0aef8cf4508dd8c994ca2679a69cc71'
[*] Switching to interactive mode
this is your flag: technofair{cUm4_Br3aK_LcG_ama_PiNt3r_pInTeR_m4iNIn_s3rViCe_y4nG_d1s3d1a
iN_4jA}
[*] Got EOF while reading in interactive
$

[*] Interrupted
[*] Closed connection to 103.152.242.172 port 9070
[+] Opening connection to 103.152.242.172 on port 9070: Done
```

c. Flag

Flag: technofair{cUm4_Br3aK_LcG_ama_PiNt3r_pInTeR_m4iNIn_s3rViCe_y4nG_d1s3d1aiN_4jA}

2. Sphinx SPARK

a. Executive Summary

tebak-tebakan lagi yukk..

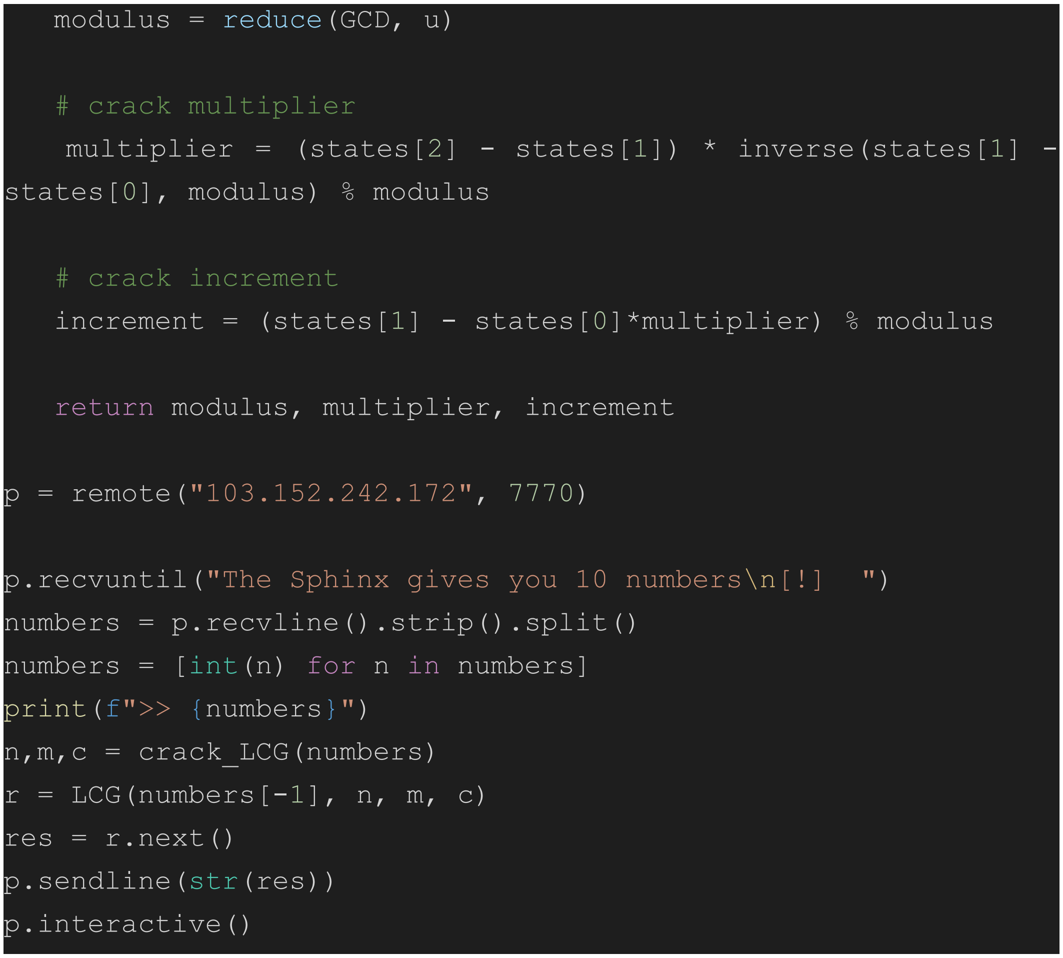
nc 103.152.242.172 7770

author : T-K!

b. Technical Report

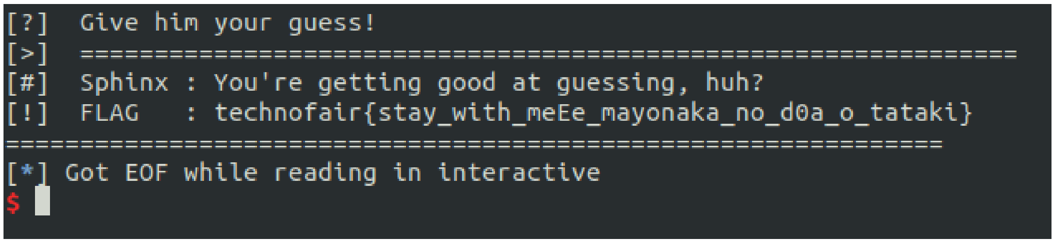
Tidak diberikan file apa-apa. Sphinx memberikan 10 angka, dan kita disuruh menebak angka selanjutnya. Karena tidak ada source, jadi kami menebak kalau ini LCG (lagi). Jadi kami pakai script yang tadi (dengan sedikit perubahan), jalanken, dapet flag. Berikut full scriptnya

```
from pwn import *
from Crypto.Util.number import *
from functools import reduce

class LCG:
   def __init__(self, state, modulus, multiplier, increment):
       self.state = state
       self.modulus = modulus
       self.multiplier = multiplier
       self.increment = increment

   def next(self):
              self.state  =  (self.state  *  self.multiplier  +
self.increment) % self.modulus
       return self.state

def crack_LCG(states):
   # crack modulus
   t = []
   for i in range(len(states) - 1):
       t.append(states[i+1] - states[i])
   u = []
   for i in range(len(t) - 2):
       result = abs(t[i+2] * t[i] - t[i+1]**2)
       u.append(result)
```



```
    modulus = reduce(GCD, u)

    # crack multiplier
     multiplier = (states[2] - states[1]) * inverse(states[1] -
states[0], modulus) % modulus

    # crack increment
    increment = (states[1] - states[0]*multiplier) % modulus

    return modulus, multiplier, increment

p = remote("103.152.242.172", 7770)

p.recvuntil("The Sphinx gives you 10 numbers\n[!]  ")
numbers = p.recvline().strip().split()
numbers = [int(n) for n in numbers]
print(f">> {numbers}")
n,m,c = crack_LCG(numbers)
r = LCG(numbers[-1], n, m, c)
res = r.next()
p.sendline(str(res))
p.interactive()
```

Hasil:



```
[?]  Give him your guess!
[>]  ===============================================================
[#]  Sphinx : You're getting good at guessing, huh?
[!]  FLAG   : technofair{stay_with_meEe_mayonaka_no_d0a_o_tataki}
===============================================================
[*] Got EOF while reading in interactive
$ 
```

c. Flag

Flag: **technofair{stay_with_meEe_mayonaka_no_d0a_o_tataki}**

Forensic

1. Ingatan_MR_2

a. Executive Summary

Sebelum laptop teman saya rusak, dia bilang bahwa dia sempat mengunduh file audio aneh berekstensi .wav saat ia sedang berselancar di internet secara otomatis. Waktu dia putar file tersebut, terdengar suara aneh yang dia sendiri tidak tahu itu apa. Bisakah kamu memecahkan misteri dari file tersebut?

File berikut digunakan untuk chall Ingatan_MR_1 dan Ingatan_MR_2 https://mega.nz/file/h8kmkJLJ#j_WiXc03OwodHw9y0QJ0DSE5LeW_s1GhoOIKGjaYQBc sha256sum 7z: cad18b04ac9e4edc7898e098c7a43ec34498248436f10870deb0652ab734e666 sha256sum raw: ad41d4fa4f4800017183842e34c11d23e85fc4674b97984e5844924da616d8e7

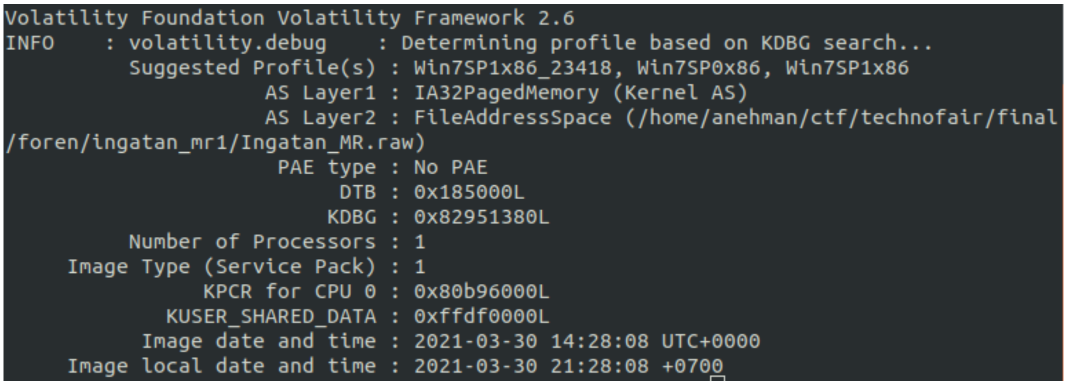
author: MidnightRumble

b. Technical Report

Diberikan file 7z. Extract, duar 1GB. Size besar == memory analysis. Jadi langsung pakai volatility

```
volatility -f Ingatan_MR.raw imageinfo
```

Hasil:

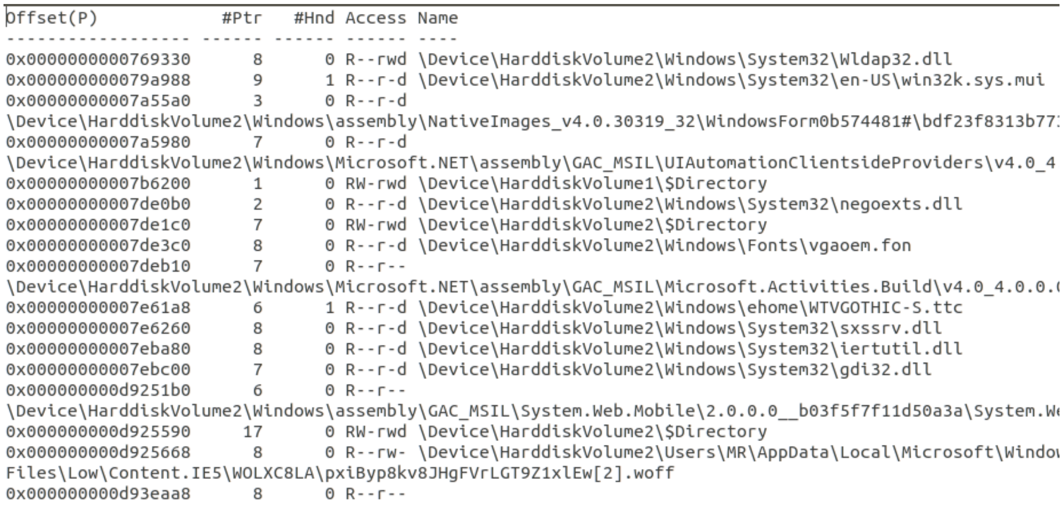


```
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/anehman/ctf/technofair/final
/foren/ingatan_mr1/Ingatan_MR.raw)
                      PAE type : No PAE
                           DTB : 0x185000L
                          KDBG : 0x82951380L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x80b96000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2021-03-30 14:28:08 UTC+0000
     Image local date and time : 2021-03-30 21:28:08 +0700
```

Ok, profile sudah diketahui, sekarang cek isi file apa saja

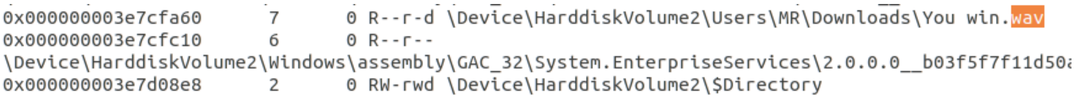
```
volatility -f Ingatan_MR.raw --profile=Win7SP1x86_23418 filescan > fscan
```

Hasil (potongan):



```
Offset(P)            #Ptr   #Hnd Access Name
------------------ ------ ------ ------ ----
0x0000000000769330      8      0 R--rwd \Device\HarddiskVolume2\Windows\System32\Wldap32.dll
0x000000000079a988      9      1 R--r-d \Device\HarddiskVolume2\Windows\System32\en-US\win32k.sys.mui
0x00000000007a55a0      3      0 R--r-d
\Device\HarddiskVolume2\Windows\assembly\NativeImages_v4.0.30319_32\WindowsForm0b574481#\bdf23f8313b77
0x00000000007a5980      7      0 R--r-d
\Device\HarddiskVolume2\Windows\Microsoft.NET\assembly\GAC_MSIL\UIAutomationClientsideProviders\v4.0_4
0x00000000007b6200      1      0 RW-rwd \Device\HarddiskVolume1\$Directory
0x00000000007de0b0      2      0 R--r-d \Device\HarddiskVolume2\Windows\System32\negoexts.dll
0x00000000007de1c0      7      0 RW-rwd \Device\HarddiskVolume2\$Directory
0x00000000007de3c0      8      0 R--r-d \Device\HarddiskVolume2\Windows\Fonts\vgaoem.fon
0x00000000007deb10      7      0 R--r--
\Device\HarddiskVolume2\Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.Activities.Build\v4.0_4.0.0.
0x00000000007e61a8      6      1 R--r-d \Device\HarddiskVolume2\Windows\ehome\WTVGOTHIC-S.ttc
0x00000000007e6260      8      0 R--r-d \Device\HarddiskVolume2\Windows\System32\sxssrv.dll
0x00000000007eba80      8      0 R--r-d \Device\HarddiskVolume2\Windows\System32\iertutil.dll
0x00000000007ebc00      7      0 R--r-d \Device\HarddiskVolume2\Windows\System32\gdi32.dll
0x000000000d9251b0      6      0 R--r--
\Device\HarddiskVolume2\Windows\assembly\GAC_MSIL\System.Web.Mobile\2.0.0.0__b03f5f7f11d50a3a\System.W
0x000000000d925590     17      0 RW-rwd \Device\HarddiskVolume2\$Directory
0x000000000d925668      8      0 R--rw- \Device\HarddiskVolume2\Users\MR\AppData\Local\Microsoft\Windo
Files\Low\Content.IE5\WOLXC8LA\pxiByp8kv8JHgFVrLGT9Z1xlEw[2].woff
0x000000000d93eaa8      8      0 R--r--
```

Sekarang kita mencari file wav. Langsung aja Ctrl+F

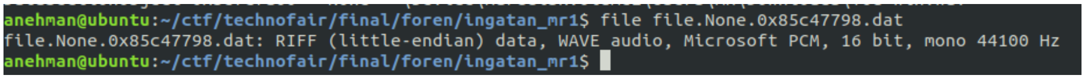


```
0x000000003e7cfa60      7      0 R--r-d \Device\HarddiskVolume2\Users\MR\Downloads\You win.wav
0x000000003e7cfc10      6      0 R--r--
\Device\HarddiskVolume2\Windows\assembly\GAC_32\System.EnterpriseServices\2.0.0.0__b03f5f7f11d50
0x000000003e7d08e8      2      0 RW-rwd \Device\HarddiskVolume2\$Directory
```

Extract filenya

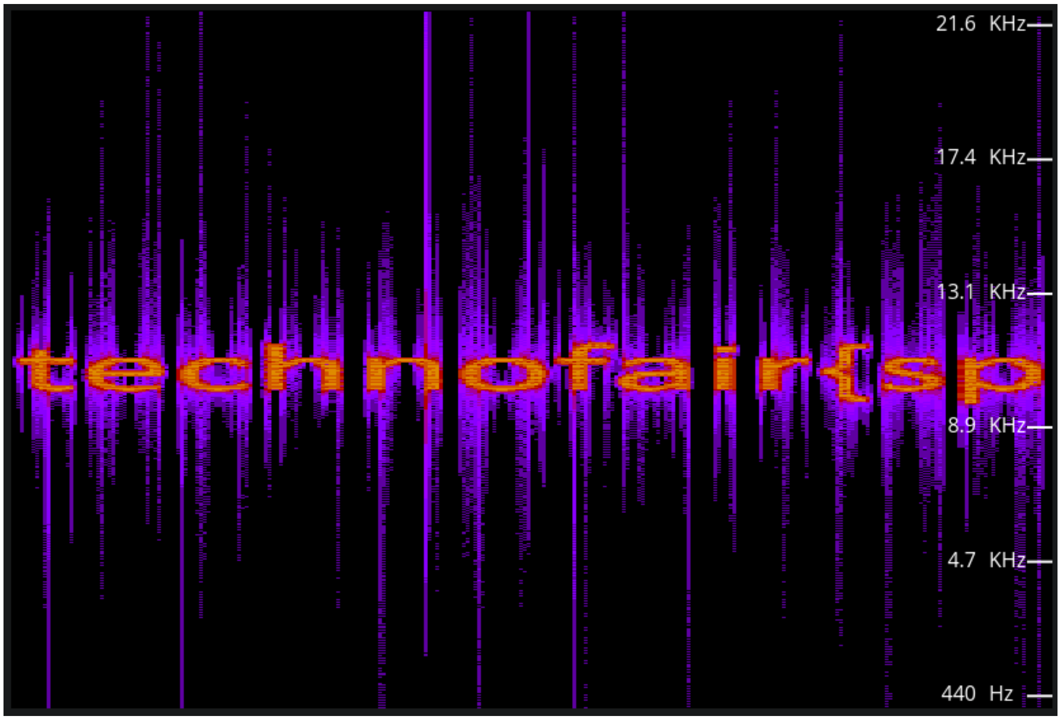
```
volatility -f Ingatan_MR.raw --profile=Win7SP1x86_23418 dumpfiles -D . -Q 0x000000003e7cfa60
```

Hasil:



```
anehman@ubuntu:~/ctf/technofair/final/foren/ingatan_mr1$ file file.None.0x85c47798.dat
file.None.0x85c47798.dat: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 44100 Hz
anehman@ubuntu:~/ctf/technofair/final/foren/ingatan_mr1$ 
```

Ketika kita play, terdengar suara dengan frekuensi tinggi (bikin sakit telinga). Jadi kita cek spektogram dengan onlen tool ini. Berikut hasilnya



Lanjutin terus, dapet deh flagnya

c. Flag

Flag: **technofair{sp3cToGr4m_k3reN_uY}**