Write Up ARA 5.0
blankx

SMK Telkom Malang

Daniel Dhaniswara
Farhan Diwan Ananta
Rafif Dhaifulloh Musyaffa Kumoro

Daftar Isi

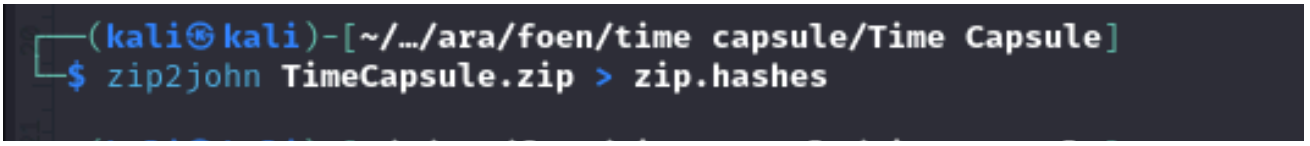
Forensic

Time Capsule

Pada soal ini kita diberikan sebuah file zip dengan nama TimeCapsule.zip yang dimana harus kita pecahkan passwordnya. Kita diberi clue untuk passwordnya. Dari clue tersebut kita bisa buat code untuk generate wordlistnya.

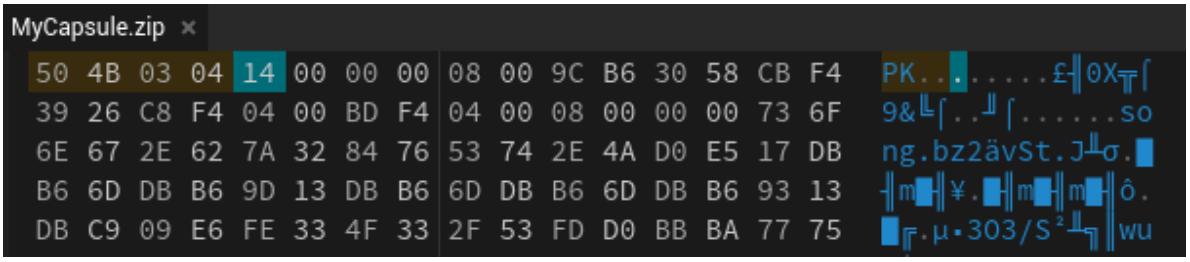
```
 import itertools

# Define the possible elements
months = ['1', '2', '03', '4', '5', '6', '7', '8', '9', '10', '11', '12']
chars = ['kAor1', 's3nKu', 'sTev3', 'Lev1', 'L1Ly']
specials1 = ['*', '#', '!', '%', '&', '+']
nums = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
letters = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V',
'W', 'X', 'Y', 'Z']
specials2 = ['*', '#', '!', '%', '&', '+']

# Generate all possible combinations
combinations = itertools.product(months, chars, specials1, nums, letters, specials2)

# Convert combinations to strings and write to file
with open('wordlist.txt', 'w') as f:
        for combo in combinations:
        password = ''.join(combo)
        f.write(password + '\n')
```

dengan wordlist ini kita bisa brute force dengan John The Ripper

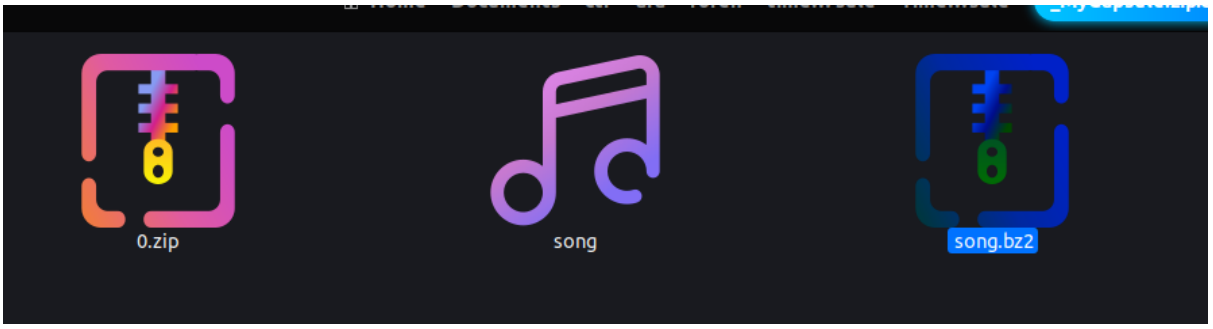



```
(haha㉿haha)-[~/…/ara/foren/time capsule/Time Capsule]
$ /tools/JohnTheRipper/run/john --wordlist=wordlist.txt zip
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 325008 for all loaded hashes
Will run 12 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
5s3nKu%3T+       (TimeCapsule.zip/MyCapsule.zip)
1g 0:00:00:01 DONE (2024-01-28 18:57) 0.6757g/s 149448p/s 149448c/s 149448C/s 5s3nKu*0I*..5Lev1%7V+
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

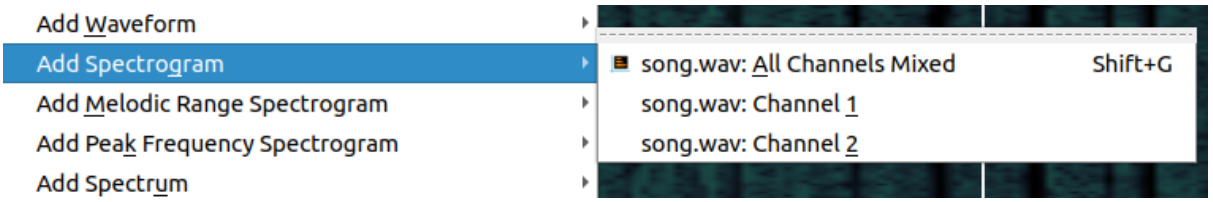
Setelah kita ekstrak kita mendapatkan file yang tidak bisa dibuka. ternyata signature dari file tersebut salah. kita rubah signature file tersebut.



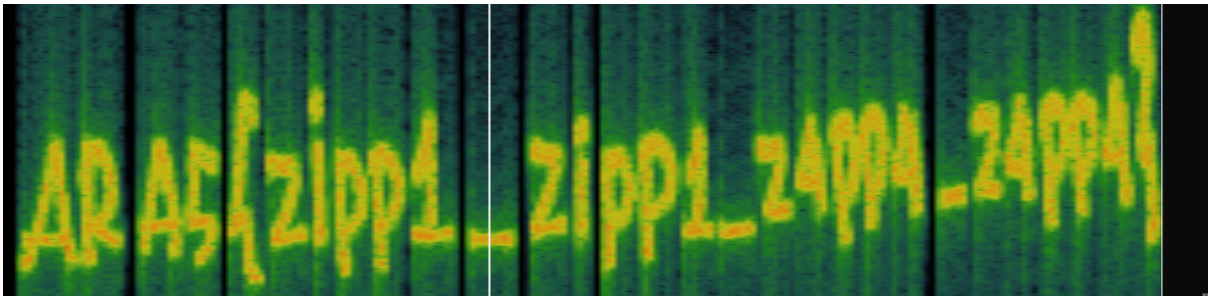
setelah itu kita bisa ekstrak file ini menggunakan binwalk kemudian masuk ke folder output ekstrak file MyCapsule.zip kemudian ekstrak file song.bz2 kita akan mendapat file song. File song ini adalah file wav, maka dari itu kita rename menjadi song.wav



kemudian dengan tools sonic visualiser kita tambahkan layer spectrogram -> song.wav: Channel 1.



dan kita kan mendapat flagnya.



Flag: **ARA5{zipp1_zipp1_z4pp4_z4pp4}**

Sussy Bakaware

Pada soal ini kita diberi sebuah file zip yang berisi file pcap. Kita diminta untuk menganalisa dan menjawab beberapa pertanyaan yang tersedia.

1. What the IP and DNS that host the malware? (ip_domain)
   ip dan domain dari dari malware ini bisa didapat pada bagian awal paket yang tercapture.

   ```
   mimsmehediclub.com A 46.4.205.200
   Len=0 MSS=1460 WS=256 SACK_PERM=1
   ```

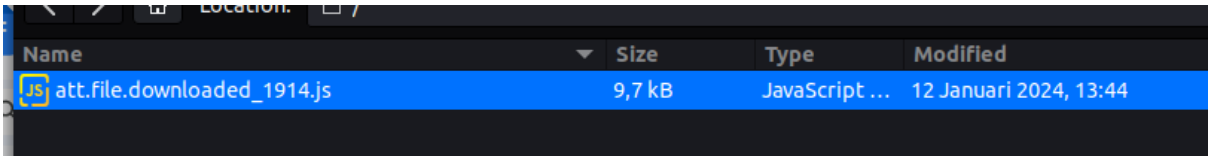
   Jawaban: **46.4.205.200_mimsmehediclub.com**
2. IP Address that has been infected?
   Ip yang terinfeksi bisa dilihat pada yang sama dimana kita melihat ip dan domain untuk jawaban soal pertama pada kolom destination

   ```
   2 0.046622      10.1.12.1           10.1.12.101
   ```

   Jawaban: **10.1.12.101**
3. What is the request token when the malware initiated the connection to the CnC?
   Kita bisa export semua file yang ada pada protocol http kemudian buka salah satu file dengan nama de4846fc29f26952.php yang memiliki form-data dengan name=”token”

   ```
   ------AECFCAAECBGDGDHIEHJE
   Content-Disposition: form-data; name="token"

   f960cc969e79d7b100652712b439978f789705156b5a554db3acca13cb298050efa268fb
   ------AECFCAAECBGDGDHIEHJE
   ```

   Jawaban: **f960cc969e79d7b100652712b439978f789705156b5a554db3acca13cb298050efa268fb**
4. The filename of malware? (xxx.xxxx.redacted_xxxx.ext)
   dari semua file yang kita export di step sebelumnya ada file yang bernama download.php yang ketika di cat bukan menampilkan code php tetapi seperti sebuah file zip, kita coba rename dengan ekstensi zip dan kita mendapati ada file bernama att.file.downloaded_1914.js tetapi ketika kita coba tidak bisa, kta coba jika kita input nama malwarenya dengan ekstensi zip dan ternyata bisa

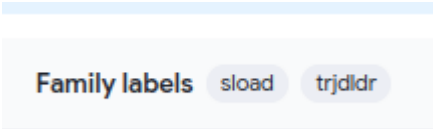
   

   Jawaban: **att.file.downloaded_1914.zip**
5. Arrival or timestamp of malware? (UTC Format, YYYY-MM-DD HH:MM:SS UTC)
   Kita ubah terlebih dahulu format waktu di wireshark kita menjadi UTC Format, YYYY-MM-DD HH:MM:SS. pada paket yang tercapture pada nomor 16 terlihat sepertinya malware terkirim pada waktu itu. dan kita coba input waktu dari paket itu tercapture.

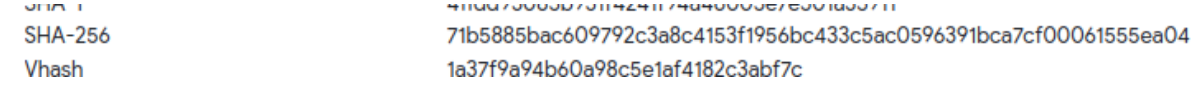
   ```
   15 2024-01-12 20:34:39,534828 10.1.12.101    46.4.205.200    TCP     54 50841 → 80 [ACK] Seq=822 Ack=3
   16 2024-01-12 20:34:43,620559 10.1.12.101    46.4.205.200    HTTP   540 GET /download.php HTTP/1.1
   17 2024-01-12 20:34:43,620738 46.4.205.200   10.1.12.101     TCP     54 80 → 50841 [ACK] Seq=3041 Ack=
   ```

   Jawaban: **2024-01-12 20:34:43 UTC**

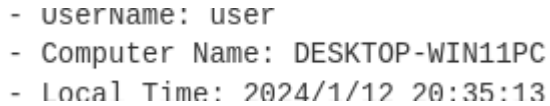
6. Malware family labels? (format: lowercase, fam1_fam2)
   Kita bisa check malware family dan labelnya dengan virus total

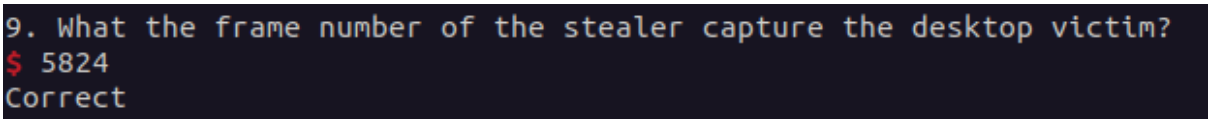
   Jawaban: **calisto_sload**
7. SHA-256 of malware?
   SHA-256 juga bisa kita check dengan virus total pada bagian details

   Jawaban:
   7acaa1011452c0d1a72dd162a8d78e07fbe0cce56276a937eacff119aa39da83
8. What the computer name of victim? (xxxxxxx-xxxxxPC)
   pada file yang sudah kita export sebelumnya terdapat banyak file dengan nama de4846fc29f26952.php yang dimana jika kita export all maka akan diberi nama de4846fc29f26952.php de4846fc29f26952(1).php dan seterusnya. pada de4846fc29f26952(6).php kita menemui form-data dengan name="file_name" dengan value c3lzdGVtX2luZm8udHh0 yang jika di decrypt dengan base64 adalah system_info.txt kemudian kita coba decrypt content filenya dan kita bisa mendapat nama pcnya.

   Jawaban: **DESKTOP-WIN11PC**
9. What the frame number of the stealer capture the desktop victim?
   Pada awalnya kita coba nomor 6038 dimana kita bisa mendapat foto dekstop victimnya tetapi tidak bisa, kemudian kita mencoba 5822 karena pada saat export kita sebelum file dimana kita mendapat foto desktopnya. ternyta masih salah dan kita mencoba untuk masukan nilai 5824 dan ternyata benar.

   Jawaban: **5824**
10. What the function name that has loaded command for the malware
    Disini kita beautify terlebih dahulu source codenya. kemudian kita identifikasi dan kita menemukan function _0x3cef yang membuat perintah PowerShell dan kemudian mengeksekusinya menggunakan Windows Script Host(WSH).
    Jawaban: **_0x3cef**

Flag:
ARA5{1t5_4ll_4b0ut_4tt3nt10n_th3_M4lW4r3_1nv3st1g4t0r_0x69a221}

Cryptography

Ryan's Strange Assignment

Pada soal ini kita diberikan sebuah file ct, dan RSAgeneratornya.

ct

```
Public Key: [ e, N ]
Public Key: [ 114886333760015985036554090542783661670178316083,
65666763392503492856526565702975459212561217488 7 ]

Ciphertext = [388470564545595079878104053981025526531939606859,
453176023391532805708302460105667157725589851094,
388470564545595079878104053981025526531939606859,
758023579890743132932455047454644956725865001 94,
530636545397020801879048076629625949622834349271,
375102954800183654669573725068164483048779280257,
996716606688375639052503768166393567155691356 61,
375102954800183654669573725068164483048779280257,
375102954800183654669573725068164483048779280257,
548590315496515548263582684646962335108239338721,
375102954800183654669573725068164483048779280257,
140887375510816447108962772482031766699016216554,
140212787491282887085498898710330206078088868768,
242179089744385364312781540147541186854680604100,
398044336768077716652000929266760922026198523016,
328163223491055229981745557826815118704798556561,
548590315496515548263582684646962335108239338721,
203670039431684285409927419369078161781353023554,
140887375510816447108962772482031766699016216554,
140212787491282887085498898710330206078088868768,
282461792303566009334287359856182792688545271 52,
352317776039632073723207591355488816387781272693,
548590315496515548263582684646962335108239338721,
245693816302915231385429799263018906306181844928,
328163223491055229981745557826815118704798556561,
284701600970156838561135032032260883397153054123,
443620019394148520237590263606896913967512611950]
```

RSA

```
#! usr/bin/env Python3
import random
import math
import sympy
from sympy import mod_inverse

def is_prime(num):
        if num < 2:
        return False
        for i in range(2, num // 2 + 1):
        if num % i == 0:
        return False
        return True


def generate_prime(min_val, max_val):
        prime = random.randint(min_val, max_val)
        while not is_prime(prime):
        prime = random.randint(min_val, max_val)
        return prime


p, q = generate_prime(100000000000000000000000, 1000000000000000000000000),
generate_prime(100000000000000000000000, 1000000000000000000000000)
while p == q:
        q = generate_prime(100000000000000000000000,
1000000000000000000000000)

n = p*q
phi = (p-1)*(q-1)

e = random.randint(3, phi-1)
while math.gcd(e, phi) != 1:
        e = random.randint(3, phi-1)

d = mod_inverse(e, phi)

print("Public Key: [",e, n,"]")
print("Private Key: [",d, n,"]")
print("N is: ",n)
print("Phi is: ",phi)
print("P is: ",p)
print("Q is: ",q)

plaintext = "ARA{REDACTED}"
encodedtext = [ord(ch) for ch in plaintext]

# (txt ^ e) mod n = cph
ciphertext = [pow(ch, e, n) for ch in encodedtext]
print(ciphertext)


# (cph ^ d) mod n = txt
```

```
encodedtext = [pow(ch, d, n) for ch in ciphertext]
plaintext = "".join(chr(ch) for ch in encodedtext)

print(plaintext)
```

dari N = 65666763392503492856526565702975459212561217 4887
dicari faktor nya dari factordb.com

http://factordb.com/index.php?query=65666763392503492856526565702975459212561217 4887

p = 750654204080680317868433
q = 874793787013089568682039

sehingga untuk mencari decrypted text dengan source code berikut ini

```
from Crypto.Util.number import inverse, long_to_bytes

c = [38847056454559507987810405398102552653193960685 9,
45317602339153280570830246010566715772558985109 4,
38847056454559507987810405398102552653193960685 9,
75802357989074313293245504745464495672586500194,
53063654539702080187904807662962594962283434927 1,
37510295480018365466957372506816448304877928025 7,
99671660668837563905250376816639356715569135661,
37510295480018365466957372506816448304877928025 7,
37510295480018365466957372506816448304877928025 7,
54859031549651554826358268464696233510823933872 1,
37510295480018365466957372506816448304877928025 7,
14088737551081644710896277248203176669901621655 4,
14021278749128288708549889871033020607808886876 8,
24217908974438536431278154014754118685468060410 0,
39804433676807771665200092926676092202619852301 6,
32816322349105522998174555782681511870479855656 1,
54859031549651554826358268464696233510823933872 1,
20367003943168428540992741936907816178135302355 4,
14088737551081644710896277248203176669901621655 4,
14021278749128288708549889871033020607808886876 8,
28246179230356600933428735985618279268854527152,
35231777603963207372320759135548881638778127269 3,
54859031549651554826358268464696233510823933872 1,
24569381630291523138542979926301890630618184492 8,
32816322349105522998174555782681511870479855656 1,
28470160097015683856113503203226088339715305412 3,
44362001939414852023759026360689691396751261195 0]
```

```
n = 656667633925034928565265657029754592125612174887
e = 114886333760015985036554090542783661670178316083
p = 750654204080680317868433
q = 874793787013089568682039

phi = (p-1)*(q-1)
d = inverse(e, phi)
decrypted_text = ""
for ch in c:
        m = pow(ch,d,n)
        decrypted_text += long_to_bytes(m).decode()

print(decrypted_text)
```

ARA5{y4yy_y0u've_f0uNd_me!}

Mandarin Class from wish

```
import random
from random import randint

flag = "???"

encrypted_flag = ""

key = randint(1,500)

for ch in flag:

        e = chr(ord(ch)*key)
        encrypted_flag += e

print(key)
print(encrypted_flag)

# print(key) = ???
# print(encrypted_flag) = "桶顆桶ひ兊帯□□孨囖挀櫞婨懎囖崐陳栴囖涾舟牂"
```

format flag = ARA5{XXXXXXXXX}
diketahui bahwa huruf pertama adalah A
sehingga dari soal chr(ord(ch)*key) = “桶”
huruf A pada karakter pertama nilai ord adalah 65

sehingga untuk mencari
key = ord(encrypted_flag[0])/65
key = ord('桶')/65
setelah tahu nilai key, untuk mencari setiap huruf dari flag dengan menggunakan solver berikut ini
solver.py

```
import math

encrypted_flag="桶顙桶ひ兊带□□孨囖捉穖婾懎囖崵噉栴囖滘舟牂"
key=ord(encrypted_flag[0])/65
decrypted_flag = ""
for ch in encrypted_flag:
    d = chr(math.floor(ord(ch) / key))
    decrypted_flag += d
print(decrypted_flag)
```

flag nya **ARA5{g00d_luck_for_y4}**