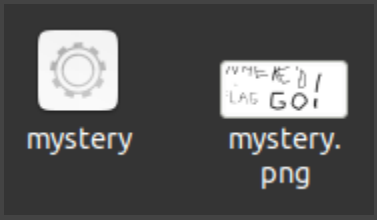
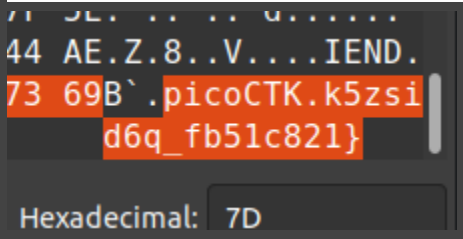
Mark Seliternikov
picoCTF - Investigative Reversing 0 [300 points]

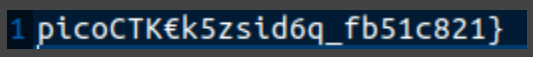


לאתגר זה מקבלים 2 קבצים ונאמר שהדגל חבוי היכנשהו.
באתגר מרומז שיהיה צורך ביכולות reversing.

התחלתי מהתמונה, כי זה משהו שהיה לי די קל לעבוד איתו בתרגילי עבר.

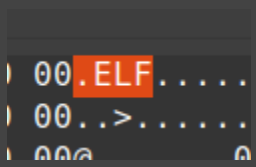
ככה נראית התמונה כאשר פותחים אותה, בפיקסלים של התמונה עצמה לא נראה שיש דגל או משהו שקשור ממש.



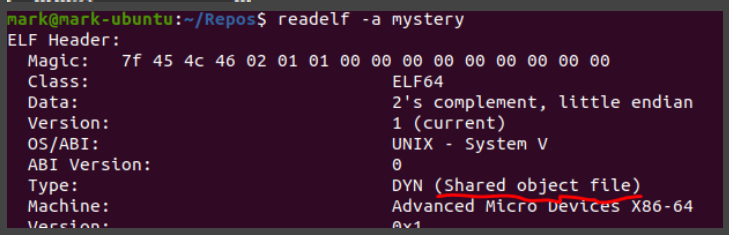


פתחתי את התמונה ב-Hex editor (השתמשתי ב-Ghex)

ונראה שמצאתי משהו שמזכיר דגל, רק שחלק מה-format לא נכון וגם אין "הודעה" בתוכן של הדגל עצמו (משהו ש-pico עשו עד כה).



לאחר מכן פתחתי את הקובץ השני "mystery", גם ב-Ghex.
מצאתי שזה קובץ מסוג ELF, בגלל שלא הכרתי באותו זמן מה זה סוג הקובץ הזה, אז חיפשתי באינטרנט.



מסתבר שזה קובץ מקומפל מ-C, ספציפית קובץ Shared library.
בעצם Libraries ב-C הם כלים שמקלים על העבודה של כותב התוכנה (בעצם זה קוד שנכתב על ידי בן אדם אחר ומשומש על ידי המתכנת)

אפשר ללמוד על הקובץ בעזרת פקודה בלינוקס "readelf".

לא ידעתי איך לקרוא את התוכן המקומפל עצמו של הקובץ, לכן הייתי צריך לחפש איך לעשות זאת.

משם הגעתי למסקנה שאני צריך ללמוד רברסינג.
לאחר למידה בעזרת טוטוריאלים ביוטיוב מצאתי איך לעשות זאת, בעזרת תוכנה הנקראת Ghidra.
התוכנה נכתבה על ידי ה-NSA, ומשום מה האתר חוסם IP ישראלי.

לכן, אם אתם גם רוצים להוריד את התוכנה, תשמשו ב-VPN (אני ממליץ לפתוח אחד בחינם שאתם תנהלו דרך AWS)

לאחר שהשתמשתי ב-Ghidra מצאתי את main.
והתחלתי לעבור על הקוד שנתנה התוכנה.

לפי הקוד נראה שזאת תוכנה שאמורה לרהחביא את הדגל (שמצאנו) בתוך התמונה. (Steganography)

```
local_10 = *(long *)(in_FS_OFFSET + 0x28);
__stream = fopen("flag.txt","r");
__stream_00 = fopen("mystery.png","a");
if (__stream == (FILE *)0x0) {
  puts("No flag found, please make sure this is run on t
}
if (__stream_00 == (FILE *)0x0) {
  puts("mystery.png is missing, please run this on the s
}
```

בשורות קוד כאן ניתן לראות כתיבה של ה-6 תווים הראשונים של הדגל נכתבים לתמונה. (picoCT)

```
puts("at insert");
fputc((int)local_38[0],__stream_00);
fputc((int)local_38[1],__stream_00);
fputc((int)local_38[2],__stream_00);
fputc((int)local_38[3],__stream_00);
fputc((int)local_34,__stream_00);
fputc((int)local_33,__stream_00);
```

כאן ניתן לראות שמהאות ה-7 עד ה-14, העלו את התו ב-5 מקומות

```
local_54 = 6;
while (local_54 < 0xf) {
  fputc((int)(char)(local_38[local_54] + '\x05'),__stream_00);
  local_54 = local_54 + 1;
}
```

כאן ניתן לראות איך כתבו את האות ה-15

```
fputc((int)(char)(local_29 + -3),__stream_00);
```

כאן ניתן לראות שהאותיות האחרונות נכתבו בצורה רגילה ללא שינוי.

```
local_50 = 0x10;
while (local_50 < 0x1a) {
  fputc((int)local_38[local_50],__stream_00);
  local_50 = local_50 + 1;
}
```

כתבתי את התוכנה הבאה בפייתון כדי שתתרגם לי הכל כבר במכה במקום לעשות זאת ידנית (יותר קל).

הבעיה הייתה לי עם הסמל של ה-€.

מסתבר שקוד ה-ASCII שלו משתנה לפי איזה ISO משומש, לכן את הקוד שלו הייתי צריך למצוא ידנית בטבלת ASCII באינטרנט:
https://www.ascii-code.com/

```
# Made by Mark. S.

flag_hidden = "picoCTK€k5zsid6q_fb51c821}"

print(flag_hidden[: 6], end="")

for i in range(6, 0xf):

    c = ord(flag_hidden[i])

    c -= 5

    # The euro character was causing problems for me
    # so I changed it manualy by using an ascii table
    # online at https://www.ascii-code.com/
    if i == 7:

        c = 0x7b

    print(chr(c), end="")

print(chr(ord(flag_hidden[0xf]) + 3), end="")

print(flag_hidden[0x10: 0x1a])
```

לאחר מכן בדקתי את הפלט ב-picoCTF וזה אכן הדגל הנכון :)

```
mark@mark-ubuntu:~/Repos$ python3 solverp1.py
picoCTF{f0und_1t_fb51c821}
mark@mark-ubuntu:~/Repos$
```

<u>ה-main במלואו למי שמעוניין:</u>

```c
void main(void)

{
  FILE *__stream;
  FILE *__stream_00;
  size_t sVar1;
  long in_FS_OFFSET;
  int local_54;
  int local_50;
  char local_38 [4];
  char local_34;
  char local_33;
  char local_29;
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  __stream = fopen("flag.txt","r");
  __stream_00 = fopen("mystery.png","a");
  if (__stream == (FILE *)0x0) {
    puts("No flag found, please make sure this is run on the server");
  }
  if (__stream_00 == (FILE *)0x0) {
    puts("mystery.png is missing, please run this on the server");
  }
  sVar1 = fread(local_38,0x1a,1,__stream);
  if ((int)sVar1 < 1) {
                    /* WARNING: Subroutine does not return */
    exit(0);
  }
  puts("at insert");
  fputc((int)local_38[0],__stream_00);
  fputc((int)local_38[1],__stream_00);
  fputc((int)local_38[2],__stream_00);
  fputc((int)local_38[3],__stream_00);
  fputc((int)local_34,__stream_00);
  fputc((int)local_33,__stream_00);
  local_54 = 6;
  while (local_54 < 0xf) {
    fputc((int)(char)(local_38[local_54] + '\x05'),__stream_00);
    local_54 = local_54 + 1;
  }
  fputc((int)(char)(local_29 + -3),__stream_00);
  local_50 = 0x10;
  while (local_50 < 0x1a) {
    fputc((int)local_38[local_50],__stream_00);
    local_50 = local_50 + 1;
  }
  fclose(__stream_00);
  fclose(__stream);
  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
  }
  return;
}
```