Mark Seliternikov
picoCTF - Vault Door 4 [250 points]

לאתגר זה מקבלים סקריפט של Java שנראה כך.

הקובץ מבקש סיסמה ולאחר מכן בודק את התאימות של הסיסמה.

```java
import java.util.*;

class VaultDoor4 {
    public static void main(String args[]) {
        VaultDoor4 vaultDoor = new VaultDoor4();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter vault password: ");
        String userInput = scanner.next();
    String input = userInput.substring("picoCTF{".length(),userInput.length()-1);
    if (vaultDoor.checkPassword(input)) {
        System.out.println("Access granted.");
    } else {
        System.out.println("Access denied!");
        }
    }

    // I made myself dizzy converting all of these numbers into different bases,
    // so I just *know* that this vault will be impenetrable. This will make Dr.
    // Evil like me better than all of the other minions--especially Minion
    // #5620--I just know it!
    //
    //  .::::.    .::::.
    // :::::::.::::::::
    // ::::::::::::::::
    // `::::::::::::::'
    //   `::::::::::'
    //     `::::::'
    //       `::'
    // -Minion #7781
    public boolean checkPassword(String password) {
        byte[] passBytes = password.getBytes();
        byte[] myBytes = {
            106 , 85  , 53  , 116 , 95  , 52  , 95  , 98  ,
            0x55, 0x6e, 0x43, 0x68, 0x5f, 0x30, 0x66, 0x5f,
            0142, 0131, 0164, 063 , 0163, 0137, 0143, 061 ,
            '9' , '4' , 'f' , '7' , '4' , '5' , '8' , 'e' ,
        };
        for (int i=0; i<32; i++) {
            if (passBytes[i] != myBytes[i]) {
                return false;
            }
        }
        return true;
    }
}
```

לפי השורה הזאת אפשר להבין שהסיסמה היא בעצם הדגל (כמו בשאר אתגרי vault door)
ניתן לראות שהסקריפט מפרק את הסיסמה לבייטים ואז בודק כל בייט אם הוא נכון.

```java
input = scanner.next();
userInput.substring("picoCTF{".length(),userInput.length()-1);
checkPassword(input)) {
```

```java
byte[] passBytes = password.getBytes();
byte[] myBytes = {
    106 , 85  , 53  , 116 , 95  , 52  , 95  , 98  ,
    0x55, 0x6e, 0x43, 0x68, 0x5f, 0x30, 0x66, 0x5f,
    0142, 0131, 0164, 063 , 0163, 0137, 0143, 061 ,
    '9' , '4' , 'f' , '7' , '4' , '5' , '8' , 'e' ,
};
for (int i=0; i<32; i++) {
    if (passBytes[i] != myBytes[i]) {
        return false;
    }
}
return true;
```

במבט ראשון הבחנתי שכל שורה ב-array היא בעצם צורה שונה לקודד ASCII. כלומר, Decimal, Hexadecimal, Octal.
והשורה האחרונה היא פשוט Chars.

החלטתי לפתור בעזרת פייתון.
אז בהתחלה עשיתי בדיקה אם אני בכיוון הנכון.

VaultDoor4.java | solver.py

```python
dec = [106, 85, 53, 116, 95, 52, 95, 98]

for i in dec:

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


print("\n")
```

לאחר בדיקה של הפלט הבנתי שאני אכן בכיוון.

```
mark@mark-ubuntu:~/Desktop$ python3 solver.py
jU5t_4_b
```

בשביל לפתור פשוט העתקתי את כל הארריי לפייתון ותרגמתי ל-ASCII.

ההתאמות שהייתי צריך לעשות הן להוסיף 'o' לאחר ה-0 ב-Oct כדי שפייתון ידע שזה Oct.

בנוסף הייתי צריך להפריד את השורה של ה-chars. מכיוון שלא ניתן לתרגם strig ל-chars בפייתון.

הוספתי גם את ההתחלה והסוף של הדגל כדי שיהיה לי קל יותר פשוט להעתיק את זה מהטרמינל ולהעתיק משם כבר את הפתרון

```
arr = [
106 , 85  , 53  , 116 , 95  , 52  , 95  , 98  ,
0x55, 0x6e, 0x43, 0x68, 0x5f, 0x30, 0x66, 0x5f,
0o142, 0o131, 0o164, 0o63 , 0o163, 0o137, 0o143, 0o61 ,
]

arr_2 = ['9' , '4' , 'f' , '7' , '4' , '5' , '8' , 'e']

print("picoCTF{", end="")

for i in arr:
    print(chr(i), end="")

for i in arr_2:
    print(i, end="")

print("}", end="")

print("\n")
```

ולאחר בדיקה זה באמת היה הדגל וסיימתי את האתגר :)

```
mark@mark-ubuntu:~/Desktop$ python3 solver.py
picoCTF{jU5t_4_bUnCh_0f_bYt3s_c194f7458e}
```