CSAW CTF - Misc: Coinslot

Denne oppgaven er fra en CTF som heter CSAW. Det var en oppgave i kategorien **misc**.

Første som må gjøres er å koble til serveren, prøve seg fram og se hva som skjer. I denne oppgaven fikk jeg dette som output i terminalen:

```
$0.09
$10,000 bills:
```

Jeg kunne se at det bes om én input per pengeenhet. Den første linjen fra serveren er en pengesum, og blir kalt *pengesummen* heretter. Oppgaven her var da å fylle ut pengeenhetene som tilsvarer pengesummen. F.eks:

```
$0.03
$10,000 bills: 0
$5,000 bills: 0
$1,000 bills: 0
$500 bills: 0
$100 bills: 0
$50 bills: 0
$20 bills: 0
$10 bills: 0
$5 bills: 0
$1 bills: 0
half-dollars (50c): 0
quarters (25c): 0
dimes (10c): 0
nickels (5c): 0
pennies (1c): 3
correct!
$0.02
$10,000 bills:
```

På serveren kunne dette bli generert om igjen tusenvis av ganger med en ny pengesum, men det er jo ikke godt å vite før man har flagget! Derfor lagde jeg et python-script for å løse denne oppgaven.

Pythonscriptet

Python har noen veldig gode verktøy som kan anbefales å importere til CTFer, det viktigste å nevne er vel **pwn**. Pwn er brukt ved å importere som øverst i kodesnutten under. Verktøy gjør det mulig å koble til serveren ved å bruke funksjonen remote(). Med denne funksjonen er det blant annet mulig å lese linjer (med **recvline()**) og lese til en gitt string (med **recvuntil()**).

Funksjonene som ble forklart over har jeg brukt til å løse oppgaven. Kodesnutten under viser hvordan jeg har koblet til serveren og jeg har en define som leser inn en ny pengesum, **get_coin()**. Slutten av kodesnutten viser hvordan de første to linjene fra oppgaven mottatt og gjort om for å fungere med float. Disse to delen kunne vært i loopen også.

```
from pwn import *
from math import ceil

#Kobler til server
r = remote("misc.chal.csaw.io", 8000)

#Define som leser in pengesum
def get_coin(line):
        words = line.split()
        coin = words[0][1:]
        coin = round((float(coin) * 100), 2)
        coin = int(coin)
        return coin

#Leser inn første linje: $0.03
data = r.recvline()
coin = get_coin(data)

#Leser inn andre linje: $10,000 bills:
#Setter verdien til å bli: 10,000
data = r.recvuntil("bills:")
words = data.split()
value = words[0][1:]
value = float(value.replace(',',''))*100
value = int(value)
```

```
#Loopen kjører helt til 'break'
while True:

        #True hvis nåværende enhet er større enn pengesummen
        if value > coin:
                r.sendline('0')

        #True hvis nåværende enhet er mindre enn pengesummen.
        #Hvis true: MÅ være noe høyere enn 0 i input her.
        elif value <= coin:
                amount = coin / value
                coin = coin - (amount*value)
                r.sendline(str(amount))

        #True hvis ferdig med en deloppgave
        if coin == 0 and value == 1:
                line = r.recvline()
                line = r.recvline()

                #Flagg eller ny pengesum på vei!
                if "$" in line:
                        coin = get_coin(line)
                else:
                        r.interactive()
                        break

        data = r.recvuntil(":")
        words = data.split()

        #Sjekk om det er bills eller mynter
        if words[0][0] == '$':
                value = words[0][1:]
                value = float(value.replace(',', ''))*100
                value = int(value)
        else:
                value = words[1][1:-3]
                value = int(value)

#Denne er superviktig! MÅ ha på slutten!
r.interactive()
```

De to kodesnuttene ovenfor er min fungerende kode. En ting som er viktig å merke seg, er

funksjonen interactive(). Det er viktig å avslutte koden med det om en bruker remote(). Hvis ikke vil ikke resten av outputen fra serveren blir vist i terminalen. En får altså ikke se flagget(!), som i dette tilfelle var:

```
flag{started-from-the-bottom-now-my-whole-team-fucking-here}
```

Det kunne vært løst på en finere og enklere måte, men på CTFer tenker man som regel på å løse oppgaven fortest mulig fremfor fin kode. Blant annet kan man bruke andre matematiske verktøy for å slippe å ha så mye trøbbel med float-verdier, f.eks. **numpy**.

Verktøyene presentert i denne teksten skal kunne brukes i en av oppgavene i CTFen.
Lykke til!
\(^_^)/