Lazy Game CTF Writeup

This document is a walkthrough on one way to solve the **Lazy Game CTF** on **CTFLearn**.
The objective is to explain how I was able to solve this CTF to my future self.

General Information

- *Difficulty:* Easy
- *Category:* Binary Exploitation
- *Link:* Lazy Game Challenge - CTFlearn - CTF Practice

Introduction

Lazy Game Challenge — **30** points — Easy

I found an interesting game made by some guy named "John_123". It is some betting game. I made some small fixes to the game; see if you can still pwn this and steal $1000000 from me!

To get flag, pwn the server at: `nc thekidofarcrania.com 10001`

What we have to do is connect to the server provided in the description of the challenge
To do this, we can use **Netcat** (nc for short), a useful networking tool that's capable of establishing TCP/UDP connections with a server.

Let's connect to the server:

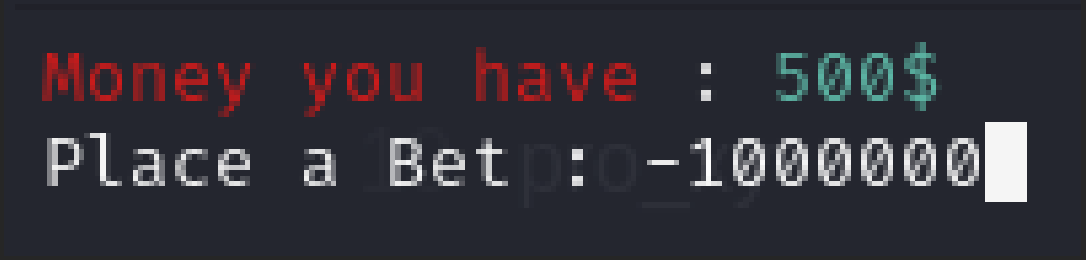
```
┌──(alexandre㉿vbox)-[~]
└─$ nc thekidofarcrania.com 10001
```

We're greeted with the following prompt:

```
Welcome to the Game of Luck !.

Rules of the Game :
(1) You will be Given 500$
(2) Place a Bet
(3) Guess the number what computer thinks of !
(4) computer's number changes every new time !.
(5) You have to guess a number between 1-10
(6) You have only 10 tries !.
(7) If you guess a number > 10, it still counts as a Try !
(8) Put your mind, Win the game !..
(9) If you guess within the number of tries, you win money !
(10) Good Luck !..

theKidOfArcrania:
  I bet you cannot get past $1000000!


Are you ready? Y/N :
```

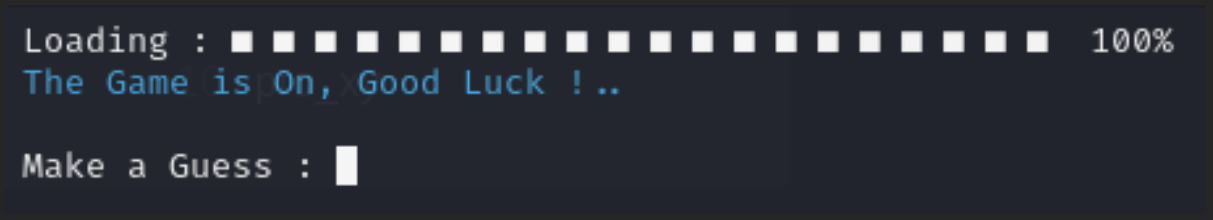
It looks like our goal is to get more than $1000000, starting with $500 and by placing numerous bets Of course via traditional methods, it would require tons of time to get the perfect amount of luck, so we're going to have to exploit the algorithm behind the game.

In order to exploit this game, we simply have to enter a negative number as a bet, and forcefully lose The algorithm doesn't even check if the bet is a negative number, which is why it's vulnerable to exploitation.

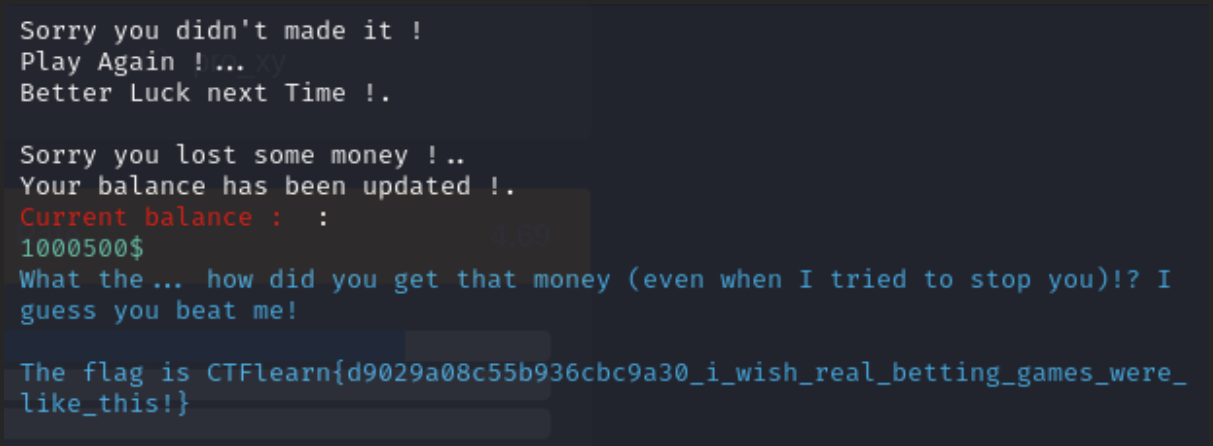


What's happening here is that when you lose, the algorithm subtracts your current amount by the bet you made, but if that bet is negative, it essentially adds its absolute value to your total.

The second step is to forcefully lose the game. Yet again, we're allowed to bet numbers outside of the [1, 10] range, any character as a matter of fact, which shows more negligence from the program.



After doing this, we get the following screen:



We get the following flag:
CTFlearn{d9029a08c55b936cbc9a30_i_wish_real_betting_games_were_like_this!}