EECS 444 Homework 1 Part 2

Yida Liu

September 23, 2019

1 Crack the password

1.1 Zip Password

Here we used brute force method for resolving the password for the `.zip` file. We noticed that the password is only a combination of 1-2 words from the word list, numbers and two special character, and $. The cracking is not as hard as it might seem to be: it took about an hour and 15 minutes in an laptop with 2 cores and 8 Gib memory with power-saver mode on. The password for file `Group A_CYBE_HW1-P2_64bit.zip` is `5@Paris\$`.

The source code can be found on this IPython notebook on Github

1.2 System Password

After we open the program, `CYBE_HW1-P2_64bit.exe`, we were prompted to enter a password for system entry. At the first attempt, we tried inputting a relatively long string for the password. To be specific, we entered 257 "a"s. This does not work out. This suggest that the input buffer is a large one.

The program prints exactly the string we entered. This might be a loop hole. Therefore, we changed our mind by using the string formatting attack with `\%x`. Our assumption is that, if the password-input buffer and the actual-password buffer are closely defined, their relative location on the stack will be close. As sufficient `\%x` is supplied, the program will print the stack values which contains the actual password. Our 2nd attempt worked out. We put 35 `\%x`s and the program output is

```
d8da5940 62f958 70 61 73 73 77 72 6f 73 64 3a 43 53 24 34 39 33 5e 32 30 31 38 2d 30 32 2
    d 30 31 0 0 0 0 0 408ba0 402460 400000 is not the password
```

The suspicious numbers in the middle could be decoded to ASCII string `passwrosd:CS\$493^2018-02-01`. Fortunately, the password to the ticket system is exactly the string after the colon `CS\$493^2018-02-01`, notwithstanding the misspelled "password".

2 Purchase the ticket

On the next screen, we were prompted to purchase a 500-dolloar ticket with $480 balance in the account. Although we cannot buy 1 ticket with the $480, dollars, we utilized the arithmetic overflow of the `unsigned short` data type for storing the subtotal price to purchase -131 tickets and we were able to get the ticket for only $36.

3 Upgrade the seat

Lastly, we were prompted to enter the customer information for the person taking planes. We thought that if the class is stored in some buffer, it is possible that we could use buffer overflow to over-write the

class information. Notice that the d.o.b is right before the class information when printed, after a few trial, we entered `09/08/1996      First` (notice the 6 spaces) as our d.o.b and successfully obtain the following print-out:

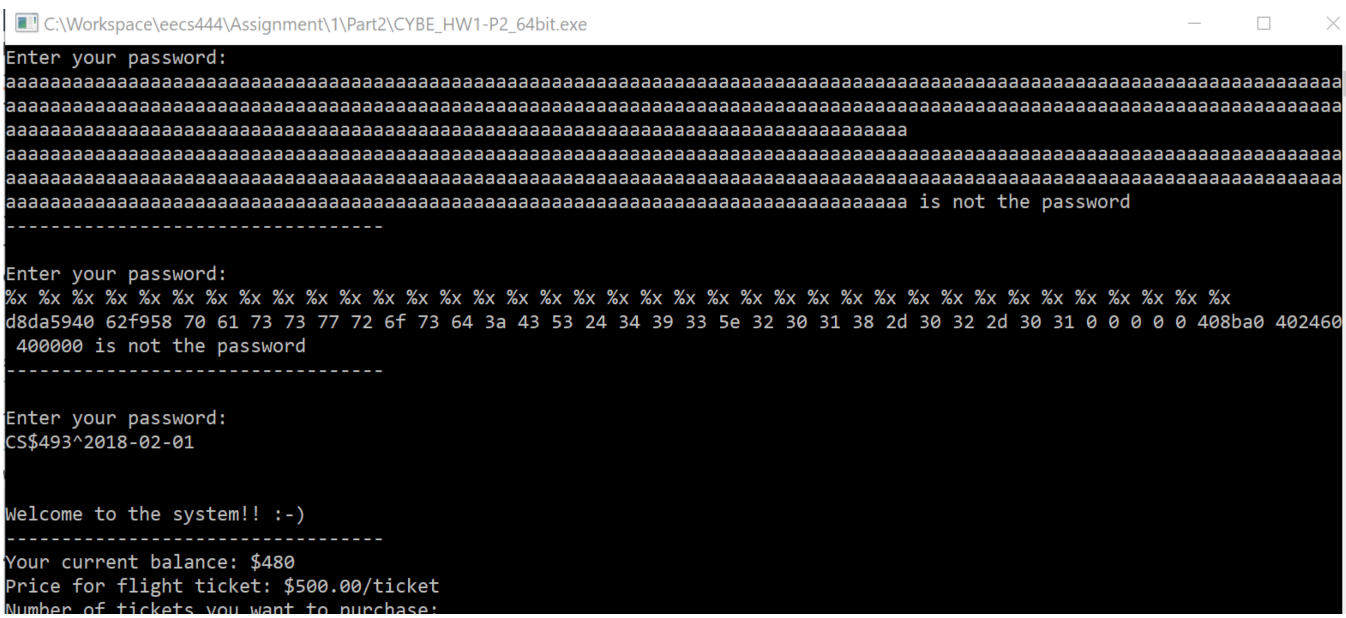
```
--------ticket 5 info-------------
Name:Liu,Yida
Date of birth:09/08/1996
Class:First
----------------------------------
```

, which upgraded the class to first.

A Screenshots

A.1 System Password

The screenshot to the Section 1.2, System Password.



A.2 Purchase the Ticket

The screenshot to the Section 2, Purchase the Ticket.

```
Welcome to the system!! :-)
---------------------------------
Your current balance: $480
Price for flight ticket: $500.00/ticket
Number of tickets you want to purchase:
1

Total cost:500

Sorry, no enough balance! :-(
---------------------------------
Your current balance: $480
Price for flight ticket: $500.00/ticket
Number of tickets you want to purchase:
-60

Total cost:35536

Sorry, no enough balance! :-(
---------------------------------
Your current balance: $480
Price for flight ticket: $500.00/ticket
Number of tickets you want to purchase:
-100

Total cost:15536

Sorry, no enough balance! :-(
---------------------------------
Your current balance: $480
Price for flight ticket: $500.00/ticket
Number of tickets you want to purchase:
-131

Total cost:36

---------------------------------
Congratulations! Your tickets have been successfully purchased! :-)

Your current balance: 444
---------------------------------
Press any key to continue . . .
```

A.3 Upgrade the Seat

The screenshot to the Section 3, Upgrade the seat.