System Programming, Spring 2020
Project: Cracking a Windows Binary

Introduction

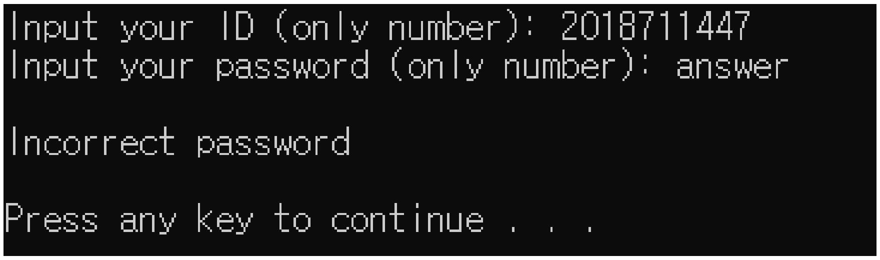
In this project, we will crack a Windows binary which requires correct inputs. Many engineers, even their majors are computer sciences, may not be able to find the correct inputs. However, as we are system programmers, we have something to do with the Windows binary. Let's find the correct inputs by cracking the Windows binary with our knowledge of system programming. This project will help you understanding how Windows programs work

Get the Windows Binary from i-Campus

You can obtain your Windows binary from i-Campus. If you execute it (maybe you have to disable your anti-virus programs), you can see two fields where your inputs are put. One of the input must be your student number. Two problems will be given. The answers of problems differ depending on your student number, which means all the students have different answers. The problems are described as below

Problem 1: Find the password when the ID is your student ID

In this problem, your job is to find the correct password **when the upper field(ID) is your student ID**. As described below figure, put your student ID in the name field, and crack the binary to find the input.

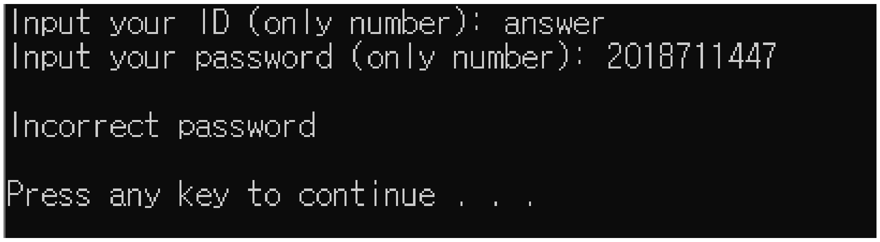


```
Input your ID (only number): 2018711447
Input your password (only number): answer

Incorrect password

Press any key to continue . . .
```

Figure 1: Problem 1 Example



```
Input your ID (only number): answer
Input your password (only number): 2018711447

Incorrect password

Press any key to continue . . .
```

Figure 2: Problem 2 Example

Problem 2: Find the name when the password is your student ID

In this problem, your job is to find the correct ID **when the password is your student ID**. As described upper figure, put your student ID in the password field, and crack the binary to find the input.

Hints

There are many ways of cracking a Windows binary. You can examine it in great detail without ever running the program, and figure out exactly what it does. This is a useful technique, but it not always easy to do. You can also run it under a debugger, watch what it does step by step, and use this information to crack it.

There are many tools which are designed to help you figure out both how programs work in Windows. Here is one example of the tools.

- `x96dbg`

  The open source Windows debugger, this is a GUI debugger tool. You can trace through a program line by line, examine memory and registers, set breakpoints, set memory watch points, and write scripts.

Submission

You must submit your report to i-Campus.

Your report include

- Detail description about your answer

- Screenshot which includes results of your crack

- Progress and unique experience of your work