Steve Penland

Network Security

Homework 2

3/08/08

Reverse Engineering C

# Introduction

One way to explore and modify the way a program works is by a process called reverse engineering. Reverse engineering (RE) is the process of discovering the technological principles of a device, object or system through analysis of its structure, function and operation1. Reverse engineering of software involves viewing the binary output and tracing the code until the desired information is reached. This could mean understanding the execution of the program, finding security holes, or discovering software cracks.

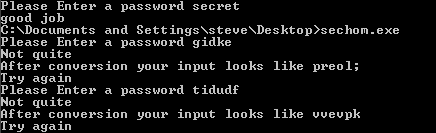
# Objectives

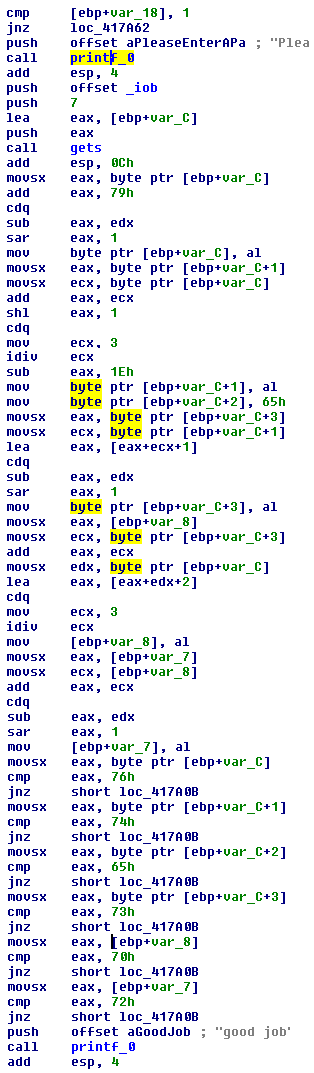
What I intend to do is create a complex ‘C’ program that requires a username and password in order to be run. The username and password will be modified in the code so that it is not apparent to someone looking at the binary data of the program to see it. This will be accomplished by using several mathematical operations as well some data manipulation using inline assembly language. I attend to make the program very difficult to reverse engineer. Also because this is a program that I am writing, I think that will help my reverse engineering skills because I will be able to see how ‘C’ functions are shown in assembly language.

# Initial Research

For this project not too much initial research is necessary. The reverse engineering programs that will be used are IDA Pro and Ollydebug. The program will be written in ‘C’ using Microsoft’s Visual Studio on a Windows XP machine.

# Creating the Program

A program was created using Microsoft Visual Studio 2003. It is a ‘C’ program named ‘sechom’. It is a relatively simple program that asks for a password, modifies the user input, and checks each character against a decimal value to determine if the user input is the correct password. If the password is correct it displays a congratulations message. If the password is incorrect it displays the modified user input and announces that the password is incorrect.

By making the user’s input go through several mathematical calculations before it is checked, it was assumed that this would cause the program to be more difficult to reverse engineer. Each letter of the user’s input it modified by performing a calculation, and in many cases it is actually added to another letter of the input causing multiple characters to be correct before any one character will be correct.

# Reverse Engineering the Program

The program was reverse engineered with IDA Pro Freeware Version 4.9. The first step was to open up the IDA Pro program and import the “sechom.exe” file. Once the file was loaded the next step accomplished was to find the main function (shown in the image on the left). Once the main function was located the problem of reverse engineering the program became merely trivial. “Var\_c” is clearly the user’s input. The mathematical operations that are done to protect the password from being discovered are basically in plain sight.

# Conclusion

Despite my best efforts to make this program difficult to reverse engineer, it was actually easier than I anticipated. As soon as I found main I realized that my method of encoding the data was not as good as I though. Reverse engineering the program is just a matter of performing a few mathematical operations to get the desired output. This assignment helped me to better understand how a program is dissembled. If I had to do this again I would defiantly find a different way to encrypt the user’s input.

# References

1 Reverse Engineering. 3 March 2008. Wikipedia.com, <http://en.wikipedia.org/wiki/Reverse\_engineering>.