Obfuscation

COMP620

Security Through Obscurity

- Obfuscation is the intentional rearranging of source code or machine language to conceal the logic or deter reverse engineering
- If an application is to be run on a client system, the client must have a copy of the executable code
- Executable code can be reverse engineered to determine what the program does

Reverse Engineering

- A compiler translates source code to machine language
- A decompiler translates machine language to source code
- There are several commercial and free programs to decompile most popular languages
- Reverse engineering can be hindered by not saving debugging information

Original Code

```
package decrypt;

public class Single {
    public static void main(String[] args) {
        java.util.Scanner keyboard = new java.util.Scanner(System.in);
        long inText, cipher;
        java.util.Date startTime, endTime;
        System.out.print("Enter the plaintext and the ciphertext>");
        inText = keyboard.nextLong();
        cipher  = keyboard.nextLong();

        startTime = new java.util.Date();

        for (long key = 0; key < Long.MAX_VALUE; key++) {
            if (COMP755.encrypt(inText, key) == cipher) {
                System.out.println("The encryption key is "+key);
                break;
              }
        }

        endTime = new java.util.Date();
        System.out.println("Elapsed time: "+(endTime.getTime()-
   startTime.getTime())
                                          +" milliseconds");
    }
}
```

Reverse Engineered Code

```
package decrypt;

   import decrypt.COMP755;
   import java.util.Date;
   import java.util.Scanner;

   public class Single {

      public static void main(String[] args) {
         Scanner keyboard = new Scanner(System.in);
         System.out.print("Enter the plaintext and the ciphertext>");
         long inText = keyboard.nextLong();
         long cipher = keyboard.nextLong();
         Date startTime = new Date();

         for(long key = 0L; key < Long.MAX_VALUE; ++key) {
            if(COMP755.encrypt(inText, key) == cipher) {
               System.out.println("The encryption key is " + key);
               break;
            }
         }

         Date endTime = new Date();
         System.out.println("Elapsed time: " + (endTime.getTime() -
    startTime.getTime()) + " milliseconds");
      }
   }
```

Example Obfuscation Transforms

Possible source code obfuscation

- Change loops to recursion
- Change variable names
- Replace if – else with ? conditional operator
  - if(A) B else if(C) D else E; becomes A ? B : C ? D : E;
- Remove all indenting and white space

Impact

- Obfuscated code is obviously much more difficult for a human to read, but programs will still execute
- The performance of the program may be significantly impacted
- Obfuscation is *"security through obscurity"* and does not provide true security

Example Need

- Students in COMP375 take an online quiz that is an applet running in their browser
- The applet sends the student's score to the server
- The quiz is timed so doing the scoring in the applet eliminates any network delay problems that might occur with a server oriented quiz
- It is possible for a student to write a program to send good scores to the server