IDA – Exploration of Lossy Compilation

Introduction

The objective of this laboratory assignment is to explore the differences that arise between the original source code and its disassembly from its binary form. You will use IDA to disassemble a sample C program names sample.c.

Upon completion of this laboratory assignment you will be able to describe some of the aspects of the lossy process of assembly and disassembly that can make it challenging to reverse engineer a software program.

Setup

Start the lab with:

```
labtainer ida
```

The resulting virtual terminal is where you will perform the lab. The home directory includes an IDA Pro installation script, run it:

```
./idafree70_linux.run
```

You will use IDA to view the program named "sample" in the home directory. That program was compiled using the mk.sh script. You need not recompile the program.

Instructions

Step 1. Launch the IDA tool, naming the sample program:

```
./idafree-7.0/ida64 sample
```

The first time you run IDA, you will see if click-through license. If the pop-up window is black, simply resize it and the text will appear. Ignore IDA errors regarding missing libraries.

Step 2. From within the IDA desktop, view the disassembled version of sample-program.

Step 3. Compare the disassembled version of sample-program to the original source code shown below (and in the sample.c file).

```
#include <stdio.h>
int main(int argc, char * argv[])
{
    char string[100];
    int c = 0, count[26] = {0};
    printf("Enter a string:\n");
```

```
    gets(string);
    while ( string[c] != '\0' )
    {
       if ( string[c] >= 'a' && string[c] <= 'z' )
          count[string[c]-'a']++;
       c++;
    }
    for ( c = 0 ; c < 26 ; c++ )
    {
       if ( count[c] != 0 )
          printf("%d %d.\n",c+'a',count[c]);
    }
    return 0;
}
```

For each of the differences you detect please document how they map to the following aspects of lossy transformations within the forward development process:

1. Stripping of symbol names and comments
2. Compiler optimizations
3. Generation of assembly language instructions that are unfamiliar to you
4. Transformation of data structures into chunks of bytes

Step 4. Propose at least two modifications to the original source code that could be used to protect that code from analysis by reverse engineers who use IDA.

When you are done, stop the lab with:

```
stoplab
```

Deliverables

Submit a short report (one report per team) in class documenting your team's results from completing steps 4 and 5. In addition, please provide any feedback on how this laboratory assignment could be improved.