Lab2: Crack the Serial

1 Introduction

Reverse Engineering is a process in which software, machines, aircraft, architectural structures, and other products are deconstructed to extract design information from them. Reverse engineering software often involves examining the program binary file without source code or design manual. Reverse engineering is an essential skill in many fields. In this course, reverse engineering is mainly used as an educational tool for learning *how computers work at the low level*.

2 `SerialKey` Program Specification

Each student gets a unique copy of the `SerialKey` program. Each copy has exactly one valid serial key that can be used to unlock the particular copy. The program takes a serial key input from the command line and validates it. The output from the program will clearly tell you if your serial key is either valid or invalid.

Serial key generation algorithm. The serial key generation and validation algorithm is classified information and thus not known to you. However, a close examination of the binary should reveal the validation algorithm and serial key generation algorithm. Both algorithms (validation, generation) use a set of randomly generated seeds (*seed*) and keys (*key*).

Seed and Key. A seed value is a 20 characters long string composed of [a-z], [A-Z] and [0-9]. A key is an integer that is bigger than 17 and smaller than 256 ($17 < k < 256$). These values are unique to each copy of the `SerialKey` program and can be obtained through reverse engineering.

3 Handout Organization and Deliverables

3.1 Lab2 Handout Organization

Lab2 handout is per-individual. Unzip the `lab2-serialkey.tar.gz` file to find your copy of `SerialKey` named `2021xxxxxx.tar.gz`. Unzip the `.tar.gz` as the following to extract the Lab2 handout root directory named `2021xxxxxx`.

```
$ tar -zxvf 2021xxxxxx.tar.gz
```

The unzipped directory will contain the files as shown below:

```
./2021xxxxxx/
├── grade.sh   // test case script for your keygen
├── keygen.c  // use this as a skeleton for your keygen (C)
├── keygen.py // use this as a skeleton for your keygen (Python)
├── SerialKey2021xxxxxx // This is your SerialKey program
└── answers.txt // Answers
```

3.2 Tasks and Deliverables

Task	Description	Points
Task 1	Find out the seed and key	20 / 100 pts
Task 2	Find out your serial	40 / 100 pts
Task 3	Write a keygen	40 / 100 pts

Task 1 & 2 Seed, Key, and Serial. You must write your seed, key , and serial in the `answers.txt` file. The seed, key, and serial must be written in this exact order and separated by a newline ('`\n`'). If you do not have one or more of the answers, leave them blank. That is, regardless of the number of answers you have, the seed must be placed in line 1, key in 2, and serial in 3. Note that we will run an automated grading script, and any mistake in `answers.txt` formatting would result in an incorrect answer.

```
answers.txt:
{Your Seed}
{Your Key}
{Your Serial}
```

Task 3. Writing Keygen. After understanding how keygen is validated and generated by understanding the `SerialKey` program, you can begin writing a keygen that generates a serial

for a given seed and key pair. You can write your keygen in either C or Python and the skeleton files for each language is provided in the lab2 handout (`keygen.{c,py}`). An automatic grader `grade.sh` is provided such that you can test your keygen for correctness against three test cases. Note that we will use randomly generated seed and key for grading, and only passing the given test cases would not guarantee full credit.

```
Usage: ./grade.sh c  (for C keygen)
       ./grade.sh py (for Python keygen)
```

4 Submission

You will submit `2021xxxxxx` directory with your answers and keygen, zipped with `tar` (e.g., `2021xxxxxx.tar.gz`) to *icampus*.

Archiving the handout directory. Be warned that our grading system would not recognize other compression formats such as `.zip` or `.7z`. Hence, we ask you to archive the directory exactly as the follows:

```
$ tar -zcvf 2021xxxxxx.tar.gz ./2021xxxxxx
```

Academic integrity. This is an individual lab, and you are *not allowed* to discuss with your classmates. Your keygen program will be checked for plagiarism using our system. Be warned that it is extremely difficult to fool the plagiarism checker, and it is wise not to copy someone else's work in the first place. Any identified act of academic dishonesty would result in a Lab2 grade of zero.

And Lastly… Have Fun!!!