Reverse engineering assignment

Introduction

In this assignment, you will reverse engineer a binary that validates a username and key that can used for authenticating users and write a keygen("key generator") program that can generate the key for any username.

The binary accepts two command line arguments: a username and a key and prints out if the key is valid for the user or not. As a first step, you should analyze the provided binary and understand how it validates the key. After this, you may optionally write an algorithm or pseudocode that describes the validation process. This is to help you ensure that you have correctly understood the validation routine and also help you write the keygen program. Once you have understood the validation algorithm, it is straightforward to write the keygen program.

In addition to submitting the keygen program, you have to also submit a plain text write-up describing how you reverse engineered the binary and understood the key validation algorithm as well a description of the key validation algorithm. To help with this process, maintain notes when you are reversing the binary. You can either write them down or maintain comments inside the disassembler. You have to upload this writeup along with the keygen program.

Many of the real world keygen programs work similarly - they reverse engineer the key generation or key validation routine and are able to determine how to generate valid keys. However, the key generation/validation routines will not be as simple as the one above.

Note: Please do not use the skill you develop after having completed this assignment to reverse engineering and break any commercial software's key generation algorithm. If caught, you can spend many years in prison for violating the software's license terms, pay a huge fine and maybe even be banned from using computers for the rest of your life. Don't do bad things with the skills you developed.

How we will test your program

We will simply invoke the run.sh file using "sh". Be sure that your run.sh file is compatible with "sh". We will test your submissions on Ubuntu 14.04 and by default, Ubuntu symlinks sh to dash. If you are using a different Linux/Unix based OS, be sure to test with dash. For most simple commands, dash is mostly a drop-in replacement for sh but if you run some fancy commands in run.sh, expect issues. We will pass a username as a command line argument.