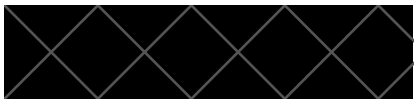
ENEE 457 Project 4



Contents

Overall System Design

The gradebook is represented as a statically sized struct containing arrays of both students and assignments. Each assignment contains a name (with max length 49), an int for max points, a double for weight, and an assignment id. Each student contains a first and last name (max length 49 each) and 2 parallel arrays, one with assignment ids and grades for the corresponding assignment. Because the arrays have fixed capacity regardless of how many entries are there, every array has a corresponding length variable. The gradebook also maintains a counter that represents the next assignment id to use when adding an assignment.

The encryption method is AES-256-GCM, which is an authenticated encryption method. The initial key and IV are generated by reading from /dev/urandom, and the encrypted gradebook file consists of the IV, the tag from the encryption result, and the encrypted gradebook itself. After using gradebookadd, the IV is re-rolled by reading from /dev/urandom again before re-encrypting with the same key. Because this is an authenticated encryption scheme, it is easy to detect when the key is incorrect, or the file has been tampered with.

Attacks and Counters

Buffer Overflows (from fread() and from user strings)

The only opportunities for user input are reading the encrypted gradebook and parsing the command line strings for names. Since the gradebook size is fixed, we can check if the size of the encrypted gradebook file matches the size of the IV, tag, and gradebook combined (gradebookdisplay.c:91 and gradebookadd.c:110). Since the command line argument strings are allocated by the OS, the only check we have to do is a simple strlen() before inserting into the gradebook (gradebook.c:127,183).

Brute Forcing Key

The encryption method is AES-256-GCM, with 2^{256} possible keys. This makes it infeasible to brute force the key and gain unintended access to the gradebook. (gradebookcrypto.c:43)

Tampering with Gradebook File

Because AES-256-GCM has authentication, tampering with the gradebook would cause the authentication to fail (gradebookcrypto.c:73,121). The only way to forge an authentication would be to find the key, which is infeasible by brute forcing.

ROP

I compiled the program with -fstack-protector-all to add stack canaries to all functions, and this will prevent overwriting any return addresses on the stack in potential ROP attacks. (CFLAGS in Makefile)