**COSC 440 – Operating System Security Practical Assignment #3**

**Fall 2021 - Assign: October 12, 2021 Due: October 21, 2021 BCT**

**Instruction – This assessment is an individual work. Student should not discuss nor consult with their lab group or class peers on this assignment.**

**Assignment Name:** Library Interpolation for Security Auditing

­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Purpose:** Library Interpolation is a technique that allows the interception of certain function calls and without requiring the recompilation of the calling program or shared objects. With this technique, a special interposable library is constructed so that at load time, the program links to the interposable library instead of the shared library. For each function in the shared library for which auditing is to be invoked, the interposable library contains a function with the same name. If the desired function is not contained in the interposed library, the loader continues its search in the shared library and links directly with the target function. The interposed module can perform any auditing-related function, such as recording the event, fact of the call, the parameters passed and returned, the return address in the calling program, and so forth. Typically, the interposed module will call the actual shared function so that the application’s behavior is not altered, just instrumented.

LD\_PRELOAD is an optional environmental variable that allows additional, user-specified, ELF shared objects to be loaded before all others. This is called preloading a library. With Preloading, library functions will be used before others of the same name in later libraries. This enables library functions to be intercepted and replaced. As a result, program behavior can be non-invasively modified, i.e. a recompile is not necessary. Thus, in this assignment students will develop a program that intercept and force security auditing of a library function using LD\_PRELOAD.

* Skills: Some of the practical skills students will gain from this assignment include:
  + Using LD\_PRELOAD to enforce security auditing
* Knowledge: This assignment will also help students become familiar with:
  + Application-level security auditing
  + Library Interpolation

**Tasks:**

Your task is to create an interposable library with a new definition for the “opendir” function call which will be used to restrict and audit the use of the ***ls*** command. The new implementation of opendir will intercept the call to access any target directory issued by ls and then makes a decision to whether allow ls to open the directory or not based on the following conditions:

* If the target path is within the /home directory, the new opendir will call original opendir function and passed the parameter (the target directory) as expected.
* If the target path is not within the /home directory, the new function will log this access as an error in the syslog and then deny access to open the target directory.

Below is a sample execution of ls with preloaded library modlib2.so that contains a new implementation of opendir.

Text

Description automatically generated

**Submission:** Submit a 1-2page document containing a fully commented source code for your interposable library, the screenshot of your log file and two sample executions of ls with LD\_PRELOAD taking /home and /etc as parameters respectively.