IC221 Lab: O.S. Security AY22S

**IMPORTANT**! Must be completed on a Hopper Hall Linux lab machine. This lab will not work locally on your VM or on WSL because of the way the permissions and folders are set up.

Files: read-log.c write-log.c Makefile

**Logger**

You will complete, compile, and execute two programs that implement a simple logging utility that is run with set-user-id. It operates as follows:

- Any user can run your write-log program to write to a log file in *your* directory.

- Anyone can read the last n entries in *your* log file using *your* read-log program

- Only you have access to read or write from the log file directly. Others can only access it using the write-log and read-log programs.

To do this, you will use set-user-id on the two programs, so that when another user launches the programs they will run with the privileges of the file owner (you), not the privileges of that user.

The two programs to complete and run are write-log.c and read-log.c. These programs will manipulate a log file that is also located in your home directory. After compilation, you will have two files, as follows:

/home/mids/mXXXXXX/ic221/write-log

/home/mids/mXXXXXX/ic221/read-log

Note: when you are logged in to csmidn, this is the same (for you) as the following locations:

~/ic221/write-log

~/ic221/read-log

**IMPORTANT**! You must use exactly this folder structure.

**Submission**

There is no file submission. Once your programs are working, just leave them in place in your folders. They will be tested directly in that location.

**Write-Log** First complete the write-log.c program, which will log a message to the log file and record which user wrote the message. The log file can also be cleared, which is a protected function. To clear the logfile, the real user must be *your* user id.

Requirements:

* Other users can log to your file by calling your program and giving it a text message. Each message should be written to the log with a newline (\n) at the end.
* Log file entries will include a time stamp and the *real* UID of the message author.
* Only *you* can clear your file (by sending CLEAR as the input message). If another user executes your write-log program with an input of CLEAR, the program should report an error and exit.
* Only *you* can read your log file directly, without having to call read-log (for example, using cat, more, or less). This will be determined by the file permissions on your log.txt file.

**Find Your User ID** Run the command id at a terminal. Look for UID= and make a note of the number.

**Read-Log** The second task is to edit the read-log.c program, which will print out the last n log entries, to match your path. After each re-compilation, the program again must be changed to set-user-id.

**IMPORTANT!** **Location**. your own programs must first be compiled into the following exact locations on midn.cs.usna.edu (your home while you are running from any lab machine, directly or by ssh), where XXXXXX is your alpha code:

/home/mids/mXXXXXX/ic221/write-log

/home/mids/mXXXXXX/ic221/read-log

**IMPORTANT**! **File Permissions**. all the sub-folders, from your home folder up through logger, must be world-readable and world-executable. But don’t make them world-writable, unless you really trust your friends :-).

For example:

ssh m242424@csmidn

cd ~

chmod 755 .

mkdir ic221 (unless it already exists)

chmod 755 ic221

chmod 755 ic221/\*

cd ic221

**IMPORTANT**! **Re-apply set-uid**. Each time you compile your read-log and write-log programs, the setuid will need to be re-applied:

chmod u+s read-log

chmod u+s write-log

To save time, you can also include the above statements in a Makefile, right after the compilation steps.

**Testing**

**IMPORTANT**! You must ssh into csmidn to conduct the tests!

ssh m242424@csmidn

**Indirect Tests** (these will have user m999999 indirectly access the logger programs of m242424)

To test your program, there are two programs that will simulate running your program from a different user id. The input argument, your user id, will tell the tester where to find your program. There is a dummy user, m999999, who has the two tester programs configured for you.

# This has m999999 execute the write-log program of user m242424:

/home/m999999/test-write-log m242424 <message>

# This has m999999 execute the read-log program of user m242424:

/home/m999999/test-read-log m242424 <number>

**Direct Tests** (these will allow mids to access and execute each other's logger programs)

After you have tested using the m999999 test functions, you should have other midshipmen in your class attempt to call your read-log and write-log functions. If they leave inappropriate comments, remind them that their UID is recorded, so the posts are not "anonymous."

To test m242424's logger programs *directly* (assuming you are someone *else*), do this (while still ssh'd into csmidn):

/home/mids/m242424/write-log <message>

/home/mids/m242424/read-log <number>

**Cleanup**

Once you are done, you can close the ssh connection with exit

Sample output:

$ ssh m232323@csmidn

$ ./write-log "I can write to my own log file (as me)"

$ /home/m999999/test-write-log m232323 "others can write to my log, too"

$ ./read-log 2

[Mon Apr 2 13:49:23 2018] (35001) I can write to my own log file (as me)

[Mon Apr 2 13:49:34 2018] (999999) others can write to my log, too

$ /home/m999999/test-read-log m232323 2

[Mon Apr 2 13:49:23 2018] (35001) I can write to my own log file (as me)

[Mon Apr 2 13:49:34 2018] (999999) others can write to my log, too

$ /home/m999999/test-write-log m232323 CLEAR

ERROR: Unauthorized user attempting to clear the log file

$ ./write-log CLEAR

$ ./read-log