CSCE 1102 Spring 2021
Lab 7 – Enabling KASLR

Enabling the Kernel Address Space Layout Randomization

Assigned: Tuesday, March 23rd in Lab

Due: Tuesday, March 30th at at Lab time

Delayed submission with penalty until Thursday, April 1st at 11:55pm.

Goals

The goal of this lab exercise is to achieve the same features of the previous lab exercise with KASLR enabled. This will entail changing the way your kernel module works as you will not be able to use the System.map file to retrieve the address of the System Call Table.

Details

This lab exercise is an individual exercise that you need to carryout on your own. In this exercise you need to achieve the same functionalities that your current kernel module is capable of but with KASLR enabled. With KASLR enabled the kernel code and data segments are essentially loaded at random locations. Of course the System.map is used to adjust addresses within the kernel source code through dynamic random linking and reallocation at runtime, but once the kernel is loaded there is now way to use the System.map to retrieve symbol addresses. This is a security measure that disallows system call hooking and other types of kernel level rootkit attacks. There are some APIs provided by the kernel that allows you to retrieve symbol addresses, which are provided on demand and can be disabled through special configuration for kernel compilation to achieve higher levels of kernel trip-wiring. Luckily you have such APIs available by default.

Now you need to reorganize your code with the minimal changes possible to accommodate KASLR. Your kernel module should be functioning the same in the new KASLR enabled environment.

IMPORTANT Note: Please make sure that you boot up with the kernel grub entry with KASLR enabled.

What to submit

1. All the C code you wrote for the version kernel module.
2. You Makefile.
3. A small readme file explaining how to use your make files to compile the programs.

How to submit:

Compress all your work: source code, report, readme file, and any extra information into a zip archive. You should name your archive in the specific format <Student_ID>_<Name>_Lab7.zip. Finally, upload your code to blackboard.

Grade

This Lab exercise is worth 5 % of the overall course grade. The exercise will be graded on a 100% grade scale, and then will be scaled down to the 5% its worth. The grading of the assignment will be broken down as follows:

1. 10 % for just submitting a meaningful assignment before or on the due date. This 10% does not account for the correctness of your assignment but submitting an empty assignment without code will definitely results in loosing this 10% and consequently the whole grade of this assignment.
2. 80 % for the correctness and the quality of the submitted code and make files.
3. 10 % readme file.

Delays

You have up to 2 working days of delay, after which the assignment will not be accepted and your grade in that case will be ZERO. For every day (of the 2 allowed days), a penalty of 10% will be deducted from the grade. And of course you will lose the 10% mentioned in point 1 above under the “Grade” section.