Operating Systems Homework 1 Report

Andres Ponce, 0616110

2020-10-24

Discussion Questions

1. **What is a kernel function? What is a system call?**

A kernel function allows us to interact directly with the kernel. The kernel allows us to write functions that implement some of the kernel utilities on the system. For example, if we want our program to make a new directory, we can use the provided kernel functions on our operating system of choice. These functions will then execute the required system calls.

A system call requests something directly from the kernel. However, a kernel function might just be a wrapper around the system call that directly executes the responsible code.

2. **what is KASLR? What is it for?**

KASLR stands for **K**ernel **A**ddress **S**pace **L**ayout **R**andomizer. This utility will load the kernel in a random place in memory during boot time. If the kernel code were loaded in the same memory location every time, then we could exploit the location of the kernel functions for some nefarious use. We would just need to know the code structure and then find a way to insert malicious code into that address space. By loading the kernel at a random location in memory, an attack of this sort is made much harder.

We can turn this setting off during boot time by using the `nokaslr` option.

3. **What are GDB's non-stop and all-stop modes?**

In GDB, the all-stop and non-stop modes refer to how the program stops execution. In the former, *all* the currently executing threads stop. This allows us to view the entire state of the program at a certain point. The latter mode refers to only stopping certain threads while allowing other currently executing threads to continue.

One might be more useful for isolating the behavior of a single thread, while the other might be more useful for viewing the entire state of the program at a given point.

4. **Explain what the command `echo g > /proc/sysrq-trigger` does.**

The `/proc/sysrq-trigger` file allows us to issue instructions directly to the kernel. In the Linux file system, the `proc` directory contains information about the currently executing processes. The file `sysrq-trigger` *triggers* something to happen in the kernel. The `g` that we echo into the file is specifically used by kgdb. This is why kgdb regains control after we echo it into this file. Besides this, we can also crash the system or immediately restart the system.