Project Description: Survey of Exploit Mitigations in Modern Operating Systems

Background

So-called exploit mitigations are the last line of defense against cyberattacks that take advantage of memory errors in apps to gain access to a system. These techniques generally cannot stop attacks altogether, but raise the bar for attackers, both in terms of time and expertise needed to exploit a security bug in software. Therefore, exploit mitgations are considered a crucial part of the security of modern OSes.

Aim and purpose

Common techniques such as ASLR (that randomizes the layout of memory in a running process to make attacks harder) mitigate a broad class of exploits, and are implemented in all common OSes. Other mitigations target specific attack techniques that might only be feasible on a single OS. The aim of this project is to perform a technical deep-dive into exploit mitigations on different popular desktop OSes (e.g., Ubuntu Linux, Windows, macOS), and survey how they differ in terms of which mitigations are implemented. As part of the project, you will develop a taxonomy (i.e., categorization) of different mitigations, in terms of, for example, which class of attacks that are mitigated, where/how the mitigation is implemented, if the mitigation is general or OS-specific, and so on.

Prerequisites

A basic course on OS internals (e.g., TDDB68/TDDE68) is necessary. Some basic knowledge of C/C++ (e.g., from the OS course) is helpful to understand how memory vulnerabilities arise. Prior basic knowledge of binary exploits and ROP (e.g., from TDDC90) is a merit, but not strictly necessary. A course on compilers (e.g., TDDB44/TDDE66) could also be helpful, but is not a necessity.