Protecting Commodity OS Kernels from Vulnerable Device Drivers

Shakeel Butt, Chih-Cheng Chang*, Vinod Ganapathy
Rutgers University

Michael M. Swift
University of Wisconsin

Abstract

Device drivers on commodity operating systems execute with kernel privilege and have unfettered access to kernel data structures. Several recent attacks demonstrate that such poor isolation exposes kernel data to exploits against vulnerable device drivers, for example through buffer overruns in packet processing code. Prior architectures to isolate kernel data from driver code either sacrifice performance, execute too much driver code with kernel privilege, or are incompatible with commodity operating systems.

In this paper, we present the design, implementation and evaluation of a novel security architecture that better isolates kernel data from device drivers without sacrificing performance or compatibility. In this architecture, a device driver is partitioned into a small, trusted kernel-mode component and an untrusted user-mode component. The kernel-mode component contains privileged and performance-critical code. It communicates via RPC with the user-mode component which contains the rest of the driver code. A RPC monitor mediates all control and data transfers between the kernel- and user-mode components. In particular, it verifies that all data transfers from the untrusted user-mode component to the kernel-mode component preserve kernel data structure integrity. We also present a runtime technique to automatically infer such integrity specifications. Our experiments with a Linux implementation of this architecture show that it can prevent compromised device drivers from affecting the integrity of kernel data and do so without impacting common-case performance.