WPI

Computer Science Department
MS Thesis Presentation

Implicit Cache Lockdown on ARM: An Accidental Countermeasure to Cache-Timing Attacks

Marc Green
MS Student
WPI – Computer Science

Friday, January 20, 2016
Time: 12:00 p.m. – 1:00 p.m.
Atwater Kent 128

Advisor: Prof. Thomas Eisenbarth
Reader: Prof. Craig Shue

Abstract:

As Moore's law continues to reduce the cost of computation at an exponential rate, embedded computing capabilities spread to ever-expanding application scenarios, such as smartphones, the Internet of Things, and automation, among many others. This trend has naturally caused the underlying technology to evolve and has introduced increasingly complex microarchitectures into embedded processors in attempts to optimize for performance. While other microarchitectures, like those used in personal computers, have been extensively studied, there has been relatively less research done on embedded microarchitectures. This is especially true in terms of their security, which is growing more important as widespread adoption increases.

This thesis explores an undocumented cache behavior found in ARM Cortex processors that we call implicit cache lockdown. While it was presumably implemented for performance reasons, it has a large impact on the recently popular class of cybersecurity attacks that utilize cache-timing side-channels. These attacks leverage the underlying hardware, specifically, the small timing differences between algorithm executions due to CPU caches, to glean sensitive information from a victim process.

Since the affected processors are found in an overwhelming majority of smart phones, this sensitive information can include cryptographic secrets, credit card information, and passwords. As the name implies, implicit cache lockdown limits the ability for an attacker to evict certain data from a CPU's cache. Since this is precisely what known cache-timing attacks rely on, they are rendered ineffective in their current form. This thesis analyzes implicit cache lockdown in great detail, including the methodology we used to discover it, its implications on all existing cache-timing attacks, and how it can be circumvented by an attacker.