17-355 Project Proposal: Detecting and Flagging Self-Timing Code

Cam Wong, Joe Doyle

April 9, 2018

1 Motivation

"Timing attacks" are a broad set of attacks that can be used to covertly transmit information or inspect the internal workings of otherwise-isolated programs. Modern microprocessors have many complex execution strategies for their instruction sets, but they are designed so that every execution path is semantically indistinguishable (at least, in theory). However, although the semantics stay the same, performance characteristics – primarily how much time a sequence of instructions takes to complete – can vary depending on the state of the processor and the particular execution strategy used. When performance-critical resources (such as caches) are shared, processes can influence these resources to signal across VM sandboxes, or they can watch cache effects to try to extract security-critical data, or (in the case of Spectre) they can take advantage of processor flaws to inspect kernel memory.

There is existing work doing dynamic analysis of programs to detect attempts to certain kinds of attacks in progress, and there are analysis tools and techniques to make cryptography software less vulnerable to this kind of attack, but to our knowledge there is no existing framework for analyzing whether a piece of software attempts to invoke a timing attack.

2 Project Description

3 Papers