Biologically Inspired Software Defenses

Submitted by franz on Thu, 04/23/2015 - 6:42pm. Contributor: Michael Franz

Abstract:
Today's software monoculture creates asymmetric threats. An attacker needs to find only one way in, while defenders need to guard a lot of ground. Adversaries can fully debug and perfect their attacks on their own computers, exactly replicating the environment that they will later be targeting. One possible defense is software diversity, which raises the bar to attackers. A diversification engine automatically generates a large number of different versions of the same program, potentially one unique version for every computer. These all behave in exactly the same way from the perspective of the end-user, but they implement their functionality in subtly different ways. As a result, a specific attack will succeed on only a small fraction of targets and a large number of different attack vectors would be needed to take over a significant percentage of them. Because an attacker has no way of knowing a priori which specific attack will succeed on which specific target, this method also very significantly increases the cost of attacks directed at specific targets.

We have built such a diversification engine, which is now available as a prototype. We can diversify large software distributions such as the Firefox and Chromium web browsers or a complete Linux distribution. We will present our overall system architecture and preliminary insights and measurements. We will also discuss some practical issues, such as the problem of reporting errors when every binary is unique.

Finally, we will report on a set of groundbreaking new software diversity techniques that can additionally also defend against side-channel attacks by dynamically and systematically randomizing the control flow of programs. Previous software diversity techniques transform each program trace identically. Our new technique instead transforms programs to make each program trace unique. This approach offers probabilistic protection against both online and off-line side-channel attacks, including timing and cache-based attacks.

In particular, we create a large number of unique program execution paths by automatically generating diversified replicas for parts of an input program. At runtime we then randomly and frequently switch between these replicas. As a consequence, no two executions of the same program are ever alike, even when the same inputs are used. Our method requires no manual effort or hardware changes, has a reasonable performance impact, and reduces side-channel information leakage significantly when applied to known attacks on AES.

Biography:
Michael Franz is the director of the Secure Systems and Software Laboratory at the University of California, Irvine (UCI). He is a Full Professor of Computer Science in UCI's Donald Bren School of Information and Computer Sciences and a Full Professor of Electrical Engineering and Computer Science (by courtesy) in UCI's Henry Samueli School of Engineering. Prof. Franz was an early pioneer in the areas of mobile code and dynamic compilation. He created an early just-in-time compilation system, contributed to the theory and practice of continuous compilation and optimization, and co-invented the trace compilation technology that eventually became the JavaScript engine in Mozilla's Firefox browser. More recently, his work has focused on moving-target cyber defenses with an emphasis on automatically generated software diversity. Franz received a Dr. sc. techn. degree in Computer Science (advisor: Niklaus Wirth) and a Dipl. Informatik-Ing. ETH degree, both from the Swiss Federal Institute of Technology, ETH Zurich.