Confining (Un)Trusted Execution Environments

Michael SCHWARZ

Graz University of Technology (Austria)
michael.schwarz91@gmail.com
https://misc0110.net/web/

Abstract

With Trusted Execution Environments, modern CPUs provide users with the possibility to isolate security-relevant applications from an untrusted system using so-called enclaves. Intel SGX is such a Trusted Execution Environment, which is supported on most Intel CPUs since around 2015. To prevent enclaves from harming the system, they run as unprivileged user code and are not allowed to execute any system calls. In this talk, we present the first practical enclave malware, which allows hiding arbitrary payloads, including zero-days, inside an enclave. This demonstrates that with the current hardware design, enclaves are a double-edged sword. They not only provide a way to protect trusted software, but they also give adversaries a powerful tool to hide malicious software with ready-to-hit exploits. We show that minimal changes to the SGX specification, which consider such scenarios, can fully prevent these attacks. As these changes require small changes in hardware or microcode, they are a long-term solution. As a short-term solution, we present SGXJail, a software mechanism to protect hosts from malicious enclaves. We show that our software confinement is compatible with existing enclaves while preventing a wide range of enclave malware threats in an efficient way.