Exposing confidential WASM code in AMD SEV

***Title work in progress**

Markus Berthilsson and
Christian Gehrmann

Lund Univeristy, Ole Römers väg 3, 223 63 Lund, Sweden
markus.berthilsson@eit.lth.se

Abstract. Cloud computing opens the possibility of companies accessing high computational power without requiring the need to invest large sums of money into the necessary hardware and setup. This introduces a new security risk, however, as data needs to be stored at the cloud service providers (CSP), which demands trust of the customer in the CSP, limiting the potential use-case of the technology.

As a means to protect the customers, vendors like AMD and Intel have both released Trusted Execution Environments (TEE), which aim to add confidentiality to the customer's data through encryption. Intel released SGX, which is an enclave that encrypts the data related to one application. AMD later released Secure Encrypted Virtualization (SEV), which encrypts the entire Virtual Machine (VM) used by the guest. This encryption is supposed to protect the guest user from a malicious hypervisor. However, since the release of both SGX and SEV, side channel attacks have continuously broken this confidentiality in multiple studies. In the paper "On (the Lack of) Code Confidentiality in Trusted Execution Environments", Puddu et al. showed that executing code with Web Assembly (WASM) increased the side channel leakage. This was exploited, and they were able to extract confidential code being executed in SGX through fingerprinting the WASM instructions.

In the paper, they only applied the attack on SGX. We are attempting to perform a similar attack in AMD SEV by utilizing the framework SEV-step. We have been able to extract the side channels, such as the latency of machine code with single-instruction granularity, and are attempting to extract other side channels, such as power consumption and read and write to memory, to perform the attack.