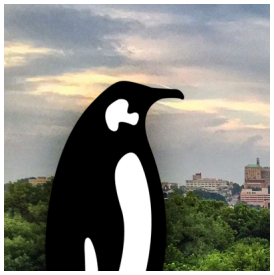

Contribution ID: **77** Type: **not specified**

Supporting guest private memory in Protected KVM on Android

Tuesday, 14 November 2023 12:40 (20 minutes)

Abstract

Please consider as a submission for a small topic.

In this talk we, present the current approach for supporting guest private memory in Protected KVM (pKVM) on Android for Arm64.

Support for confidential computing is rapidly becoming more popular, with hardware-supported solutions such as Intel's TDX, AMD's SEV, and Arm's CCA, and software-based implementations such as pKVM. One of the aspects in common is the ability to create and run a "protected" guest, whereby no other entity in the system, including the host, has access to the guest's data (unless explicitly shared).

The current KVM API presents guests' memory to the host via a host userspace address. Although the host is prevented from accessing the guest memory via that address, such an erroneous access could be fatal to the system and result in a full reset. Moreover, memory for protected guests might demand additional restrictions, such as preventing swapping and migration, which may require host access to the underlying physical pages.

Guest private memory was proposed for Intel TDX as a new API and solution to address these issues. It represents guest memory using a file descriptor, along with a new allocator that imposes restrictions on what can be done with the memory. It removes the need altogether for having any mapping of the guest private memory in the host operating system, whether in userspace or in the kernel.

This talk describes the work to port the (proposed) guest memory interface to pKVM on Android (Arm64). Being a software-based Arm64 solution, pKVM has some differences from the guest memory's original target of TDX. The most relevant difference is that pKVM allows sharing memory in place, since it does not encrypt guest memory. Moreover, in pKVM, host stage-2 faults are not necessarily fatal to the system, as they can be injected back to the host giving it a chance to handle them in some situations.

Audience

Kernel developers with an interest in confidential computing or virtualization.

Benefits to the ecosystem

At the time of this submission, the guest memory work is still in flux, the biggest issue stalling its progress being the API. All Linux (or at least KVM-based) confidential computing proposals would benefit in using the same API, which should be flexible enough to allow for differences in implementation to accommodate the differences between them. Moreover, Android is currently shipping with a GUP-based approach to prevent swapping and migration, with the drawbacks mentioned earlier, and would ideally switch to an upstream-supported interface as soon as possible.

By presenting the pKVM work in this area, we hope it would lead to a better understanding of the issues involved, aid in arriving at a consensus (if it hasn't happened already), and serve as a guide to other confidential computing proposals that haven't started using guest private memory yet.

Primary author: TABBA, Fuad (Google)

Presenter: TABBA, Fuad (Google)

Session Classification: KVM MC

Track Classification: LPC Microconference: KVM MC