An seL4-based Architecture for Layered Attestation

Grant Jurgensen, Adam Petz, Michael Neises, and Perry Alexander

The University of Kansas - ITTC

Background: Remote attestation is the process by which one computer system, the *attester*, sends evidence of its own state and/or identity to an external system, the *appraiser*, in order to obtain the appraiser's trust. To trust said evidence, the appraiser must be confident that the attester's architecture separates potentially malicious actors from critical attestation components.

Overview

- Copland [1] *attestation managers* (AMs) coordinate measurements from various layers in the architecture
- A Linux VM hosts the attestation target and the *User AM*
- seL4 hosts the VM and the *Platform AM*
- The User AM receives Copland attestation protocols which it executes in cooperation with the Platform AM
- The User AM measures applications in the Linux VM
- The Platform AM measures the Linux VM itself

Trusted Architecture

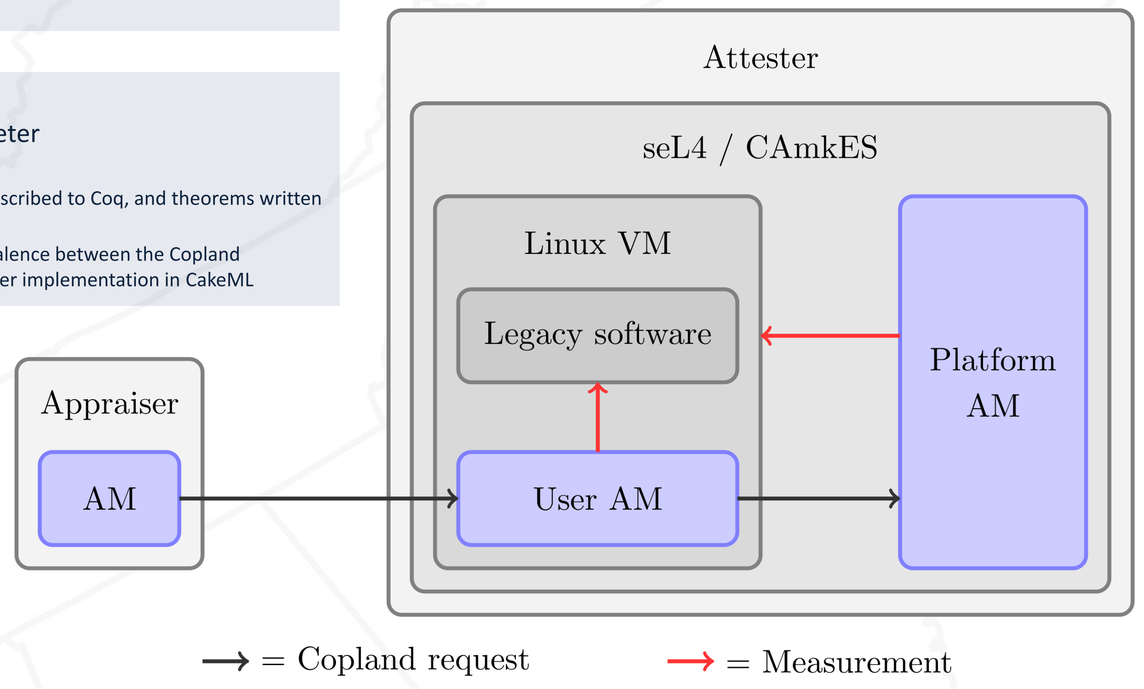
- Trust is established from the bottom up
- The Platform AM is trustworthy if the proper image is booted
  - Component is defined statically, seL4 guarantees memory is private
  - Proper image can be assumed if the system is physically secure (attackers lack physical access). Alternatively, can be established via trusted/secure boot
- The Platform AM extends trust to the User AM via system measurements
  - If the Linux VM is shown to be in a good state, we can trust the User AM's measurements

Trusted Components

- seL4 is a formally verified microkernel, proven to enforce the memory separation of statically defined components [2]
- The AMs are written in
  - CakeML, a near subset of Standard ML, with a verified compiler
  - C, compiled with the verified CompCert compiler
- The AMs use the verified HACL* cryptographic library

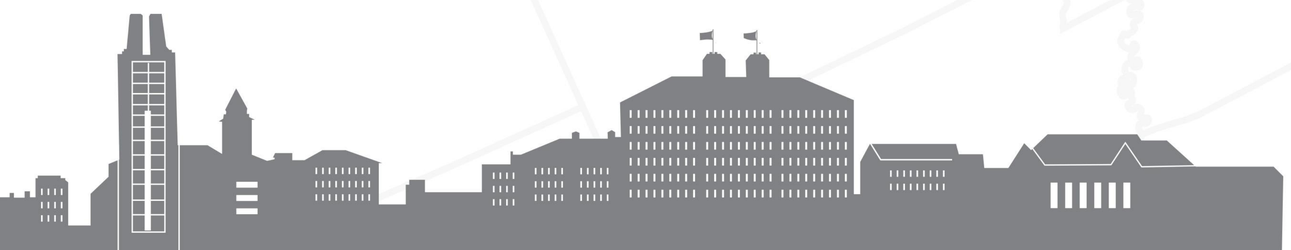
Ongoing Work

- Measurement of the Linux VM
  - We hope to incorporate an existing kernel measurement tool
  - The seL4 CAmkES component architecture needs to be expanded to allow the Linux VMM to share its memory with other components (i.e. the Platform AM)
- Verification of the Copland interpreter
  - Copland semantics defined in Coq
  - The CakeML semantics are being transcribed to Coq, and theorems written to ease the proofs
  - Ultimately, we aim to prove an equivalence between the Copland Semantics and the Copland interpreter implementation in CakeML



[1] A. Petz and P. Alexander, "A Copland Attestation Manager," *Hot Topics in Science of Security (HoTSoS'19)*, Nashville, TN, April 1-3, 2019.

[2] T. Murray *et al.*, "seL4: From General Purpose to a Proof of Information Flow Enforcement," *2013 IEEE Symposium on Security and Privacy*, Berkeley CA, 2013, pp. 415-429.



HOTSOS 2020
7TH ANNUAL
HOT TOPICS *in the* **SCIENCE OF SECURITY**
SEPTEMBER 22-23, 2020 | LAWRENCE, KANSAS