# Measured Launch

A review of the changes coming in Stable-6.

OpenXT Summit

June 7, 2016

Daniel P. Smith

apertus solutions

# Setting the Pace

Presenter:

Daniel P. Smith, Co-Owner of Apertus Solutions

Roadmap:

Basics → Goals → Implementation → Future

Format:

~15 minute talk, feel free to ask clarifying questions

Remainder reserved for discussion questions

apertus solutions

# Some Basics

# What is Measured Launch

- Consists of two main components
  - Intel TXT (Authenticated Code Module - ACM)
  - Tboot (Measured Launch Environment)
- Establishes a Dynamic Root of Trust Measurement (DRTM)
  - Measures
    - TXT launch itself
    - Policy
    - MLE
    - All multiboot modules in the boot config (tboot default)
    - TPM NVRAM (with recent versions of tboot)

# Measured Launch Policy

- There are two types of policies
  - TXT's Launch Control Policy (LCP)
  - Tboot's Verified Launch Policy (VL)
- The LCP is used by ACM to compare against predetermined/known-good measurements of the platform and MLE
- The VL is used by tboot to control what additional components gets measured and/or validates the measurements against predetermined/known-good measurements
  - different policy enforcements (nonfatal, continue, halt)

# How did OpenXT use ML

- Measurement Configuration
  - Does not use a LCP
  - Extends tboot's policy capabilities
  - Relies on tboot's default compiled in VL
  - Early init extends PCR15 with hash of root logical volume
- Measurement Usage
  - Used to seal/unseal LUKS key for the config logical volume

# Motivation & Goals

# What is success

- Three main motivations
  - Reduce maintenance burden
  - Provide consistency
  - Provide flexibility
- Goals
  - Centralize logic into one place and expose an API
  - Make the contained logic a separate OE recipe that an implementer can override in their layer

# Reference Implementation

apertus solutions

# Focus of the Work

- There are three areas that were worked
  - Key Management – the platform relies on different keys that are for specific functions (new*)
  - TPM Management – the platform has some standard ways it interacts with the TPM
  - Measured Launch Management – the platform requires setup, seal, and reseal procedures (new*)

* new just means existing logic was separated into a new OE recipe

# Key Management Interface

- Based on existing key usage patterns
  - Three core types of keys,
    - Recovery, Platform, Encrypted(config)
  - Ancillary keys,
    - Cores, Log, Own, Device
  - Initial API
    - Gen_*, Set_*, *_Unlock, *_Clear
    - seed_entropy, finalize_keys
- Compromised a little consistency for clarity

# TPM Management

- Prior implementation
  - was already fairly well abstracted
  - had a mix of Key and Measured Launch functionality
- Did not have to change much,
  - pulled any Key and ML functionality out
  - added TPM forward seal function

# Measured Launch Management

- Prior implementation only had a configure and success check
- For the initial implementation,
  - pulled the configure and success check away from TPM functions
  - mainly focused on adding the forward seal predictions
  - didn't really focus on refining an API

# Looking Forward

# Key Management

- The first go was a bottoms up, aligning an API to the current implementation
- Another round of refactoring is needed based more on a top down approach that is driven by an architectural view
- Reference implementation is a shell script library. One consideration is to migrate toward a key service. For instance, extend blktap2 to uses the Linux kernel's Trusted/Encrypted Key combination

# TPM Management

- A lot to consider
  - Trousers code base hasn't been touched in 1.5 yrs
  - Intel is working on a TPM 2.0 stack
  - Google ditched TSS in Android and ChromiumOS
    - TPM over DBus (tpm_manager)
  - The introduction of vTPMs
  - Serious discussion needs to occur on where the project wants to go with regard to accessing TPM functionality

# Measured Launch Management

- The initial implementation achieved centralizing the logic, but did not really provide an API
- For forward seal on a TPM1.2, PCR-17 was reused since DOM0 access to TXT heap is blocked by Xen. Xen will need to be modified to allow read-only access
- Need to develop a design that better accommodates both LCP and VL policies as well as allow implementers to override policy generation
- Explore how TPM2.0 Authorization Policies might allow a better mechanism for resealing after update

Discussion Time

Any Questions?

apertus solutions