TrenchBoot Secure Launch Status

Ross Philipson
Daniel Smith
Linux Plumbers Conferece
September 2021

TrenchBoot Secure Launch Status Overview

- The purpose of this discussion it to cover the the effort merge the TrenchBoot Linux kernel solution upstream.
- A brief overview on the current status of the upstreaming effort.
- For more in-depth coverage of the TrenchBoot project, please see the presentations from the Linux Security Summit 2019 and the Linux Plumbers Conference Boot and Security MC 2020:

https://linuxplumbersconf.org/event/4/contributions/522/attachments/394/637/trenchboot_lpc_20190906.final.dk.pdf

https://static.sched.com/hosted_files/lssna19/75/trenchboot_ot_lss_20190815.final.ds.dk.pdf

Secure Launch for Linux

- TrenchBoot Secure Launch for Linux provides for different strategies to build trust in the platform
- The initial implementation being worked is to demonstrate the common First Launch use case
  - This type of launch is commonly referred to as an early launch
  - This is the traditional approach that uses Dynamic Launch to root the target kernel in hardware
- TrenchBoot approach expands the standard approach for adding entry points into Linux kernel
  - Leverages existing UEFI support to handle EBS hand-off
  - On Intel platforms, SEXIT is called on shutdown paths to close access to DRTM PCRs and exit Secure Mode

GRUB

- GRUB is the most common boot loader in deployment thus making it the choice initial boot loader to make capable of being a DCE Preamble for DL
- Summary of changes
  - Expand the Linux loader to support the Secure Launch kernel_info structure
  - Add additional commands to identify Secure Launch and load the DCE module, Intel ACM or AMD Secure Loader
  - Add DL Event relocators, one for Intel SENTER and one for AMD SKINIT
- Both Intel and AMD in process of going up

The kernel_info Patch Set

- Secure Launch requires new information to be passed from the kernel to the boot loader

- The kernel’s setup_header has been the method to convey this info but has a hard limit on the amount of info it can hold and space has been almost out for a very long time

- Collaborated with H. Peter Anvin to develop the kernel_info structure as the way for the boot protocol to be extended regardless of the setup_header limitations

- This work has been accepted upstream and the current Secure Launch feature is based on it.

Early TPM Access

- The initial RFC and early submissions introduced a lightweight TPM access code in the compressed kernel code base.

- This was done to allow measurements to be extended to the TPM PCRs early in boot process.

- The introduction of a second TPM code base was not acceptable to the community and TPM maintainers.

- It was suggested we rewrite the existing mainline TPM driver to allow it to work in the compressed kernel but this approach was not feasible. The TPM driver is tied to closely to the kernel timer framework and none of that is in place in the early compressed kernel.

- We accommodated the objection to the addition of the TPM access code by removing it and deferring the PCR updates until the mainline TPM driver is available and can be used.

As of Today…

- The initial RFC was sent in March of 2020.

- Since then there have been four follow on submissions to LKML. The fourth one was posted at the end of August this year.

- Each submission has addressed any issues brought up by the community.

- We would like to get the capability merged, what next steps could be taken to assist in that goal?

ORACLE