TECTONIC Enterprise

Technical Brief

Distributed Trusted Computing

Josh Wood

Look inside to learn about Distributed Trusted Computing in Tectonic Enterprise, an industry-first set of technologies that cryptographically verify, log, and audit every layer of the modern microservices platform, from nodes to clusters to the containers that isolate and execute business applications. Get an overview of how Tectonic DTC leverages hardware, firmware, and software to form a chain of trust binding the entire enterprise stack to integrity validation of code, configuration, and environment. Find out how to begin testing and exercising DTC in the enterprise today.

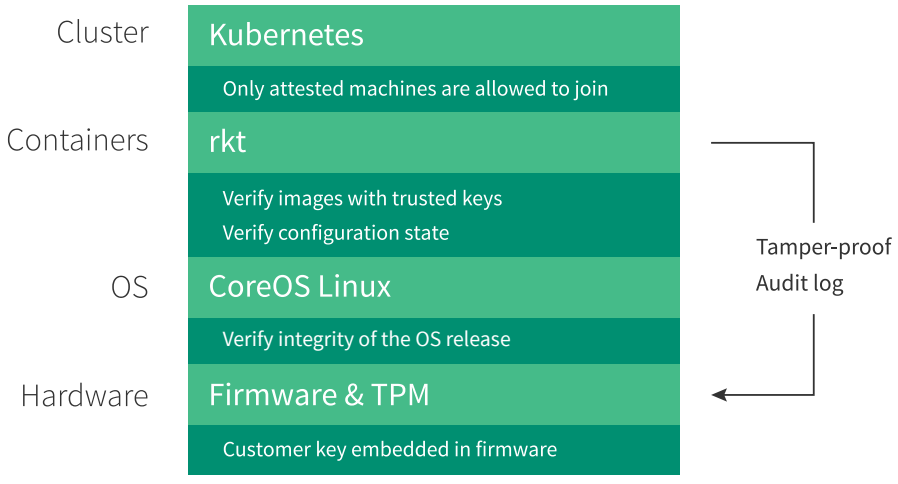
Summary

Tectonic Leads in Security

Tectonic Enterprise with Distributed Trusted Computing (DTC) comprises an industry-first set of capabilities to secure enterprise infrastructure across layers and across machines, from the hardware, to the container, to the distributed application cluster. In this overview, we outline the two architectural components that make Tectonic Enterprise with Distributed Trusted Computing the most trusted and secure place to build, run and manage containers: Secure Boot, which provides a verified system platform, and DTC itself, which harnesses both hardware and software to isolate and cryptographically verify and audit containers executing on clusters of Secure Booted nodes.

We discuss development in the CoreOS bootloader, kernel, and operating system to enable cryptographic verification of boot mechanisms, system images, application containers and their runtime environment in Tectonic clusters. This work constructs a chain of trust rooted at enterprise public keys initialized into system firmware and used to validate signed binaries and hashes of their configuration arguments in order to provably demonstrate deployed systems match specifications exactly, even in potentially hostile environments presenting the threat of low-level system attacks. We show how these facilities ensure sensitive data, such as cluster secrets, and even cluster membership itself, are granted only to validated, trusted nodes.

Components of Trusted Computing



Distributed Trusted Computing Components

DTC Secure Boot

System Image Verification

In DTC Secure Boot, the CoreOS boot process from power-on through user login is modified so that every stage is subjected to cryptographic validation, and can only execute if proven to validate with a public key, held in firmware, and used to sign the component during a controlled initialization. After power is applied, the bootloader, a small executable responsible for reading the operating system kernel from disk and setting up its initial state in memory, is the first item validated. The bootloader stage is executed only if the signature is valid, proving no out of band modification has occurred. Before exiting, the loader validates the next link in the chain, which is usually a second-stage bootloader, or the operating system kernel itself. Once again the preceding stage cryptographically validates the next unit of code to be executed, and a decision is made about whether or not to proceed.

Atomic Operating System

CoreOS is particularly suited to this kind of validation by cryptographic signature, as the system is released in a signed, atomic version representing all software, kernel and user-mode, and stored on a read-only mount point. Since the CoreOS software stack isn't just a loose collection of packages, the collective environment can be verified during the boot process, and only allowed to proceed if its integrity can be mathematically assured.

Making the TPM an Enterprise Asset

Apply SecureBoot Principles to the Application

Most software isn't part of the operating system or the bootloader. While Secure Boot provides a platform of verified integrity, the applications that a business creates and depends on, the software of the most interest to users, could still be vulnerable to replacement, modification, or other attacks. Tectonic Distributed Trusted Computing extends the principles of Secure Boot to the cluster's container execution environment.

Making the TPM an Enterprise Asset

Trusted Platform Module (TPM)

Many recent computer systems include a motherboard component called a Trusted Platform Module (TPM). Specified by the industry Trusted Computing Group (TCG), the TPM is a cryptographic processor with special operating rules designed to integrate encryption keys directly into hardware. The TPM is a general mechanism for the assurance of integrity, and, with proper support in the operating system, can be employed by system owners to verify deployed hardware and software.

Extending the Chain of Trust

With DTC, the TPM becomes an enterprise asset used to extend the chain of trust established by Secure Boot to container execution on the cluster, logging, and audit forensics. At each stage of a system's boot process and container start up, the TPM generates a cryptographically signed, unique hash of the expected code and configuration. Because this verification computes a mathematical representation, a hash, of a given target, the TCG specification refers to the process as "measurement".

Each system's TPM consists of a microprocessor specialized for a few encryption and hashing functions, with access to a small amount of storage in the form of Platform Configuration Registers (PCRs). Thus, the TPM cryptoprocessor can measure requested targets: boot stages, executables, or environments, and store the measurement results in a PCR. The TPM then allows software to read PCR contents according to strict protection rules, and use the measurements stored there to make decisions about the integrity of the target and the next action to take. Under the protection and write semantics enforced by TPM hardware, it is not easily possible even for system-level software to tamper with the measurements stored in TPM registers without detection.

Making the TPM an Enterprise Asset

TPM Utilized by rkt

The rkt container execution environment, part of CoreOS, automatically uses the TPM to perform measurement of containers when running on a system that has performed a Secure Boot to join a Tectonic DTC cluster. The App Container Image (ACI) format used natively by rkt includes a cryptographic signature feature for image verification. The combination of these two facilities brings cryptographic verification to the cluster's container execution layer. A cluster of trusted nodes executing validated, isolated application containers is, by definition, Distributed Trusted Computing.

Containers are Verified Before Execution

Tectonic Enterprise with Distributed Trusted Computing means that no machine can boot without being mathematically validated as exactly the intended deployment, from firmware to operating system; that only verified, known nodes join clusters and gain access to cluster keys, secrets, or other sensitive data, and that the containers that execute applications and services on Tectonic clusters are subject to the same integrity verification and merciless restriction.

Enterprise Deployments

Secure Admission to the Cluster

To this point, we have described the construction of a chain of trust, rooted at an enterprise-owned, initialized key in each system's firmware, extending through the boot process, to the operating system and utilities, with assurance of integrity at the cluster level for each executing application container. This has profound implications for deployment technology and economics.

As briefly mentioned above, there is a potential space for vulnerability at the cluster admission level. DTC Secure Boot advances the current state of this art, even without specific cluster orchestration support, by effectively preventing anything but verified system images from booting and joining clusters. Systems whose integrity cannot be verified are never able to request or obtain cluster data or configuration secrets, never have containers scheduled on them, and never become a threat to the security of a Tectonic cluster. Future work is planned to expand this foundation along with proposed Kubernetes admission control mechanisms to provide higher-level abstractions for acting on node integrity.

Enterprise Deployments

Safely Deploy to Hostile Locations

Cryptographic and hardware-protected system and container image integrity validation proposes an answer to the emerging question of datacenter and provider security. Even if we imagine a completely hostile – but tantalizingly low-cost – colocation facility, deployers don't have to trust their provider. Tampering with rack units' hardware or software to subvert or spy on operations is immediately detected, and stopped, by the Secure Boot process. Even a network attack on a container deployment link is foiled by the same integrity checking, integrated with rkt, the cluster container execution environment. The threat of even very low-level provider espionage is mitigated into a machine replacement issue, rather than a data disclosure or executable penetration.

As the union of these facilities, Tectonic Enterprise with Distributed Trusted Computing builds trusted nodes together into clusters provably deployed exactly as specified, whether on-premises or in the least trustworthy datacenter conditions. Enterprise applications are then executed on trusted, verified Tectonic DTC clusters within isolated containers, each secured and validated by the same mechanisms. The result is a complete distributed system in a known state exactly matching the intended deployment, providing mission-critical services with assurance of trust at each layer of the platform, even from the other side of the corporate firewall.