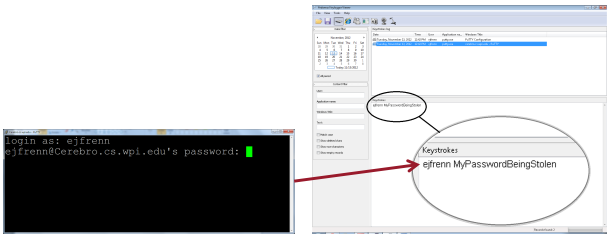
Securing Enterprise Networks using Trusted Thin Terminals

Evan Frenn and Craig A. Shue
Worcester Polytechnic Institute
ejfrenn@cs.wpi.edu, cshue@cs.wpi.edu



Problem Statement

Modern systems require applications to trust the operating system. Generally, these operating systems consist of millions of lines of code, making formal verification infeasible. Security vulnerabilities in the OS allow malware to undermine trust of the entire system. Further complicating this issue, responsibility for client security configurations frequently fall on untrained users.

Design Goals

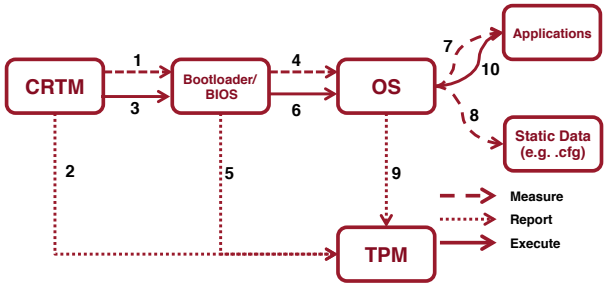
1) Provide a provably secure client, mitigating the risk of hidden malware on the machine.
   - Utilize hardware enforced attestation, allowing the client to prove the software running on its machine. Hardware based attestation provides strong assurance against software based attacks.
   - Minimize the number of trusted components to reduce analysis of the verifying system and reduce the risk of exploitation.
2) Push security responsibilities to skilled professionals.

Trusted Platform Module

The Trusted Platform Module (TPM) is a secure coprocessor that provides the ability to remotely attest to the machine's state via a hash representing the software history of the attested machine since boot. Attestation is performed by extending one of the TPM's Platform Configuration Registers (PCR). This process is generally referred to as reporting. A PCR's value can only be changed by extending its current value hashed with the newly desired value and can only be reset by a system reboot. This feature provides the basis for many prominent attestation techniques that have been developed over the last decade.

Previous Approaches

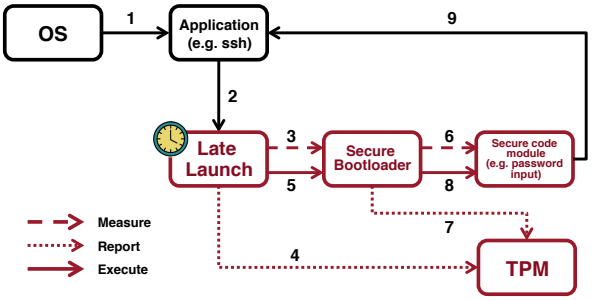
IBM's Integrity Measurement Architecture – The traditional approach extending trusted boot to encompass user space applications. The key concept of the IMA platform is the ability of each software component to measure the subsequent component prior to allowing it's execution. The measurement process consists of extending one of the TPM's PCRs with a hash for the executable data of the succeeding component. This approach requires trust in the Core Root of Trust for Measurement (CRTM), which is the first application to run once the machine is booted.



Static Root of Trust for Measurement

Limitations:
- Measurement semantics – Requires an understanding of the security risks associated with each application.
- Requires modification of every application.
- Does not address runtime integrity of the measured applications.

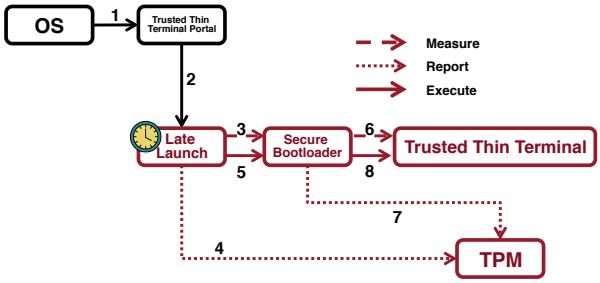
Policy Reduced Integrity Measurement Architecture - Reduces measurement list to only components that influence a target applications execution based on a Clark-Wilson integrity model. Addresses system runtime integrity by requiring sanitization all low integrity input.
Limitations:
- Requires mandatory access control implementation and analysis.
- Reduces the number of modifications to application required, however, this still includes the operating system.



Dynamic Root of Trust for Measurement

Late Launch

With the introduction of the latest TPM specification, the capability to late launch a secure environment became possible. Late launch provides a CPU instruction to invoke a secure bootloader, which measures and loads a specified application into memory. Both the bootloader and application measurements are reported to the TPM for later attestation requests. Once the application has been loaded, it receives full control of processor in a secure state. The addition of processor support for attestation, along with the static structure of the bootloader, allow late launch to be utilized at any point during execution. In comparison to a Static Root of Trust for Measurement, Late Launch allows for a significant reduction in the number of trusted components, as well as providing a static Core Root of Trust for Measurement with greater security assurances.



Late Launching a Trusted Thin Terminal

Our Approach

Our approach consists of providing an application for remote access, while utilizing late launch. Our current implementation is targeted at creating a secure shell environment. We believe this design will accomplish our original goals as follows:

1) The Late launch architecture, specifically TPM-based attestation and single threaded environment, allows for provably secure software. The remotely accessed machine is provided proof the Trusted Thin Terminal is the only application executing on the client machine.
   - In utilizing the TPM, we root the trust of our application on the hardware implementation. This provides strong assurance that physical access to the device is required to covertly compromise the machine.
   - Our implementation requires the remote system to trust only the thin terminal application, as well as the late launch functionality. Notably, our implementation removes the necessity to place trust in the host operating system.
2) Remote access to an enterprise cloud architecture allows client security and maintenance responsibilities to be pushed to IT professionals.

Challenges

- The Late Launch environment lacks both interrupts and device drivers, requiring the functionality they both provide to be handled by our Trusted Thin Terminal. Specifically, this is a significant challenge in regard to our future work of extending the Trusted Thin Terminal to a graphical remote desktop environment.
- The lack of OS functionality in the late launch environment removes the maintenance functionality it performs, specifically in regard to network access.
- The structure of Late Launch requires the use of on a single core of the CPU. This incurs substantial performance delays that must be taken into consideration in developing our Trusted Thin Terminal.

Expected Use Case

We envision an environment in which a client can have full administration over her corporate machine. When working on security sensitive tasks, our Trusted Thin Terminal would allow the user to launch a secure remote access session with similar functionality and user interface to commodity remote access applications. The user would be connected to an enterprise cloud network, with the network receiving high assurance of the remote client's security based on attestation reports received from the TPM. Further, the enterprise network would be given full control over security responsibilities of sensitive data, as it is never required to leave the cloud infrastructure.