Untrustworthy code is frequently run on computers regardless of its impact on system integrity. It is difficult for a user to tell whether an arbitrary program is trustworthy, which further complicates issues.

Ideally, a system should always behave as expected (with high integrity). This requires certain trusted components to be of high integrity from the time they are loaded and beyond.

We use the TPM to perform measurements and reliably report system state. From boot we measure all code loaded into the system as well as information flow and MAC policies. By recording only code loaded into trusted subjects, we reduce the number of measurements. This approach is similar to PRIMA, but we integrate our integrity management architecture directly into SELinux.

Integrity management is integrated into SELinux and utilizes subject types to identify trusted subjects types. When code is loaded under one of these types, a local database is referenced to compare known good hashes to what has been measured. A system can then attest its state by passing its measurement log to a verifying party where it can be check against a similar trust database.

Secure Core
- This system is stripped down to its essential programs, which must all be of high integrity.
- All configuration data must also be traceable back to creation and administration should be extremely limited.

Mixed Environment
- This scenario allows for untrustworthy code to be present in the system so long as the high integrity components are protected.
- Data on the system can evolve and less guarantees can be established for it. Rely upon security policy and filters to manage data integrity.
- This is typical of most computing systems.