Today’s Systems are Weakly Managed

Our computing environment is federated

- Assets managed by different organizations
- Many assets hardly managed at all (home machines)
- There are natural conflicts of interest in security policies
- Assessment of trustworthiness based on observation and shared reputation
Trust has meaning only in a particular context

- A system is not trusted absolutely, but instead it is trusted to operate in a particular way that is dependent upon its intended purpose.
- For a system to be secure, we must consider the different functions, and manage the contexts associated with each function.
- We need basic security functions that can provide separation for each context, allowing a trusted virtual system to be established for each context.
- Finer-grained access control can then be supported within each context.
The Focus of Trusted Computing

Recent work emphasizes mechanism.

- How to provide attestation, isolation, and secure storage.
- Policy is understood in support of the mechanism.

But the mechanism must support policy.

- Policy focus has been limited to applications like DRM and NAC.
- These applications see a system as trusted or not.
- We need to understand how to define and enforce understandable policies that better model real systems that support multiple views on trust.
The Trust View Security Architecture

TVSA Policies are separated into:

- First level provides coarse-grained authorization
  - Basic Policies of separation supported by trusted computing functions of attestation of components, isolation, and secure storage.
  - Almost capability like
  - Based on being in the right virtual system.
- Second level enforces fine-grained policies
  - Supported by the trusted components within a virtual system.
- Precomputed policies for managing virtual systems
  - Say how pieces fit together.
  - Trust, ability, and obligation negotiated in advance.
Rings represent
Precomputed policy
Virtual System
identifiers used to
enforce simple
policies.
Fine grained policies
enforced by the
individual
components
embedded within the
rings to the right.
Once a Virtual System Formed

**Accepted components have access to resources within the virtual system.**

- But they have agreed to limits on what they can do.
How to Allow Flow Across Boundaries

Some components trusted to make fine grained decisions which allow data to flow across VS boundaries.

- Component is in multiple virtual systems.
- Data flows to component, in one VS.
- Data flows out of component in other VS.
- Component decides where data can flow.
May not be allowed to join certain other virtual systems.

- Could require approval by other members
- Might carry a policy that says what other components can join.
- Might allow joins if component is known to provide controls on cross VS information flow.
- These are the pre-computed policies that determine what policies are enforced by the basic control mechanisms.

These limits do not apply to new instances

- They can form new Virtual Systems
- But these components do not gain ability to move information across VS boundaries.
The negotiation phase required assurances that the component could and would enforce those limits. Less trusted components end up encapsulated in components that will provide the enforcement.
What Kinds of Policy Work Best

Can standard policy templates be created that correspond to the intrinsic policies that people expect, corresponding to common business, personal, government, or national security interactions.

Can these “templates” be used to structure virtual systems around particular functions that are commonly used in distributed systems.
Many Applications

**DRM (or EDRM)**

- But protect not just the content owner, also the systems on which the data is accessed.

**Lampson’s Red – Green Network**

- But really a rainbow of color.
- Examples are NAC, secure VPN from external machines.

**SCADA Applications**

- Push 1st class of rules into network infrastructure
- Get performance isolation