Enable Application-level Reference Monitors
- Application control is necessary for its objects
- Reference Monitor implementation is complex
- User-level Reference Monitor support is there
- Can we build tools to support implementation?

Challenges
- SELinux depends on the Xserver to control copies
- Bugs found in LSM design; 3 years to acceptance
- SELinux Policy Server enables authorization checks
- Need to develop an approach to automate placement

Reference Monitor Design and Implementation

Security Architecture Requirements
- Hook placement is crucial.
- Hooks must pose the correct authorization request.

So… what is our contribution?
- A technique for automatic Application-Level Authorization Policy Enforcement (ALPEN).

Mechanisms

Tahoe

Alpen

Results and Challenges

Tahoe Mechanism and Results
- Hook analysis is flow and context-sensitive
- Determines parameter requirements for authorization
- Kernel analysis is uses descriptions of conceptual operations
- Called idioms that describe when code implements an operation

Alpen Challenges over Tahoe
- No hook implementations
- Policy determines conceptual operations
- Identification of conceptual operations in code
- Domain-specific metaphors (e.g., run the conceptual operation)

Ultimate Goals
- Automate Generation of Security Code
- Memory Reuse -- scrub for secrecy
- Key Handling -- ensure secret communication
- Automate Generation of Filtering Interfaces
- Hook analysis can derive legal calls to authorization

Joint work with the University of Wisconsin-Madison