

Aries²⁰⁰² Security

Distributed Security Infrastructure (DSI)

Distributed Security Policy (DSP)

Charles.Lever@ericsson.ca

ARIES Security Team
Open Systems Lab
Montréal – Canada

Outline

- Definition
- Issues
- Potential Areas for Policy
- Design Choices
- Architecture
- “Security Policy Generation through Package Management”

Security Policy: Definition

- Explicit set of rules that govern the (configurable part of the) behavior of a system's security features.

DSP: Issues

- Minimal and flexible (dynamic) administration
- Assess impact on application programmers, packagers, and system administrators
- Efficient distribution mechanisms

DSP: Potential Areas for Policy

- Access Control (mainly)
 - Acceptable accesses of (operations on) objects (resources) by subjects
- Authentication
 - Acceptable authentication mechanisms as a function of location (incoming interface: internal, external, etc.) and requested identity
- Confidentiality and Integrity
 - Integrate IPSec SPD (or leave it separate)?
 - Equivalent policy if something other than IPSec is used.
- Packet filtering rules
- Other services? (Have a look in `/etc` on UNIX!)

DSP: Design Choices (1/3)

- Definition of policy rules (syntax and semantics)
 - Fields
 - Constant rules vs. Template matching rules
 - Inheritance (or similar construct)
- Policy updates
 - API and Distribution
 - Complete vs. incremental (differential) updates
 - Triggering events
 - process creation, shutdown, authenticated connect from client, private resource creation, ...?
 - linked to new Security Contexts creation/assignment
 - templates to create policy rules vs. policy rules that are themselves templates
 - other areas than access control?
 - Policy Composition (processing node local + ... + cluster global)

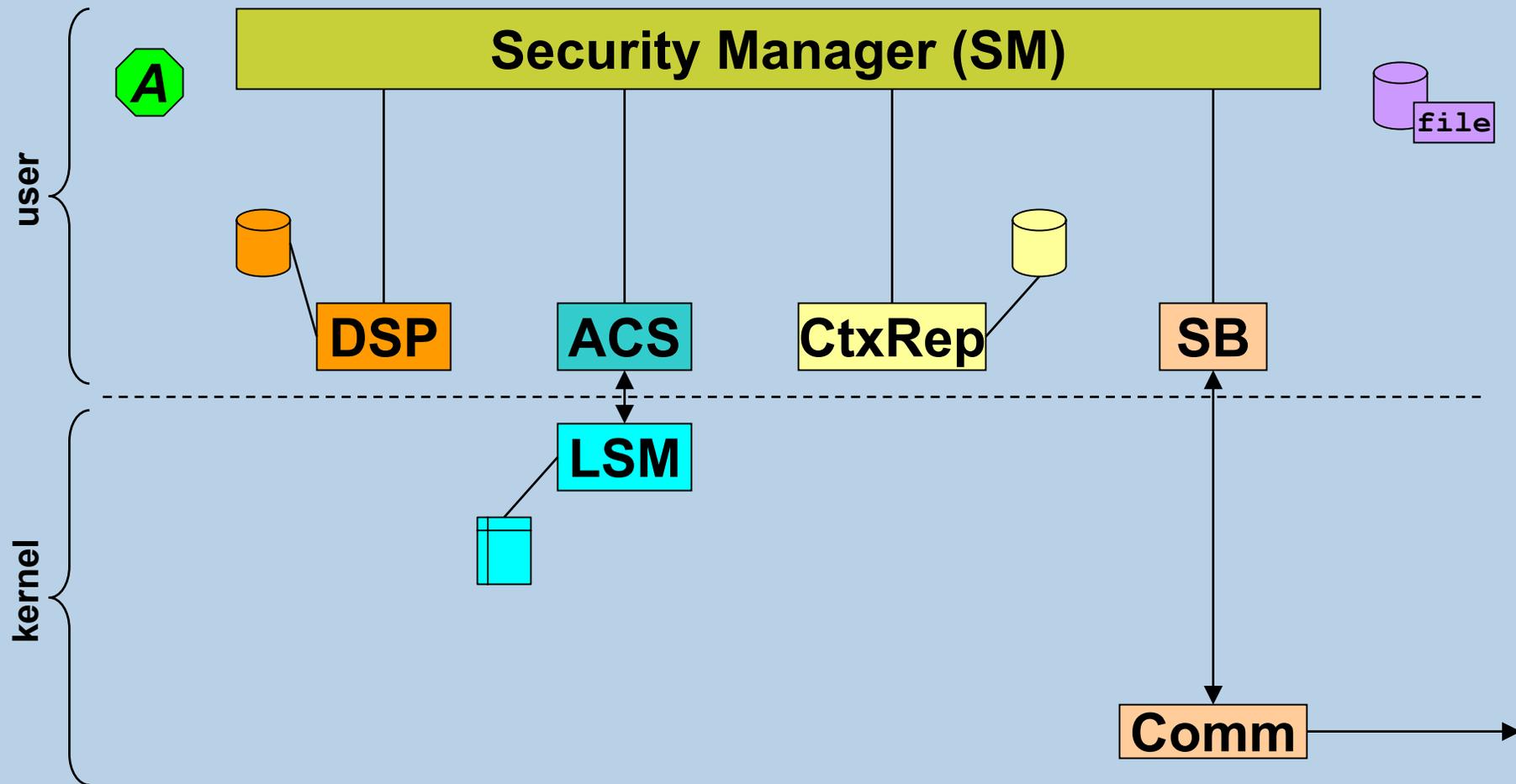
DSP: Design Choices (2/3)

- Definition of an actual policy
 - Manual configuration by the Security Administrator
 - Inferred from software package definitions or other existing source
 - Red Hat's RPM
 - Debian's dpkg
 - Rely on default behaviors as much as possible
 - Consequence of dynamic events (c.f. policy updates)
 - Ideally, the notions of policy and of system state (set of security contexts that are currently defined) should stay separated and the services (e.g., the Access Control Service) should be the ones to bring them together at decision time.
 - Updates to the policy should only come from events such as the installation of new software components.

DSP: Design Choices (3/3)

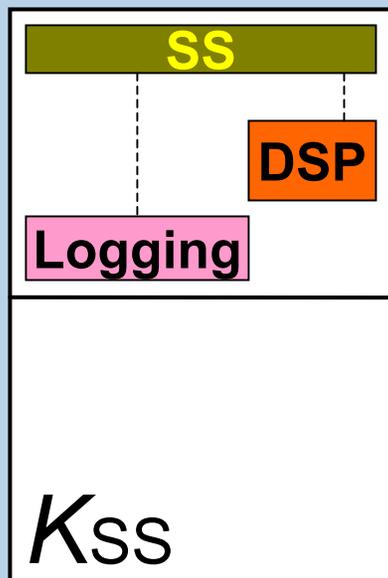
- Architecture
 - Entities
 - Main copy on security server nodes
 - Local copy on processing nodes
 - Support software
 - Interfaces
 - With Access Control Service
 - Various data representations
 - Source files
 - Compiled format?
 - In memory
 - Generation/Transformation chain

DSP: Architecture (1/2)

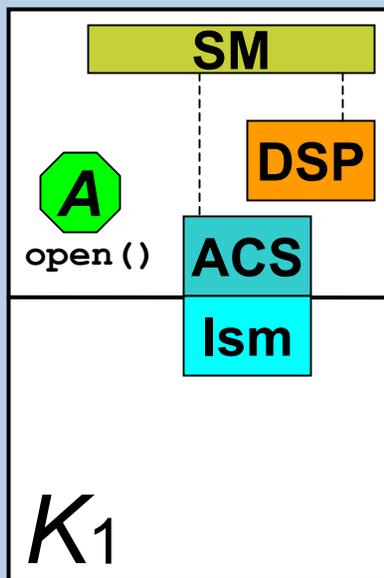


DSP: Architecture (2/2)

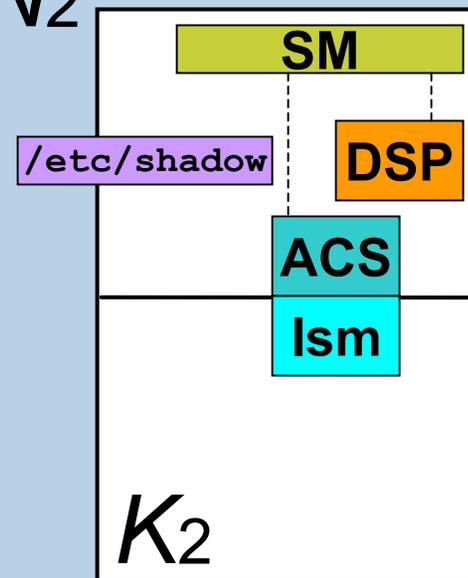
SS



N₁



N₂



DSP, Alarms, ...

Security Broker

“Security Policy Generation through Package Management”

- Based on sound security principles
- Review of
 - package management systems
 - service control schemes
 - file standard conventions
 - existing security framework configuration
- Proposed approach:
 - additions to per-package information
 - additions to site-specific information
 - modifications to existing software (`rpm`, `init`, `service`, `chkconfig`, `telinit`, configuration programs)
- Presented at OLS 2002

In Summary...

- Best approaches for clustering retained for further investigation
 - DSI strategy document
 - DSI paper
- Identifying the needs and currently missing features for HA clusters
 - One masters student (Éric Gingras)
- Build prototype to help understand, compare, and choose among several possible design choices
 - Minimal DSP (enough to test DSI), rest as open source

ERICSSON 