Enforcing flexible access control in a networked policy domain

Joshua Brindle, Karen Vance, Chad Sellers, Tresys Technology

Introduction

- SELinux is now deployed in many distributions
- These deployments control a single system
- Multi-system deployments assume alike policies
- Network capabilities of SELinux now available
- Supporting distributed systems is the next logical step

Requirements

- Analyzability of network policy in its entirety
- Efficient distribution of policy to affected systems
- Use of upstream policies where possible
- Synchronization and atomicity in policy updates

Current State

- The reference policy is used in all modern distributions
- Assumptions are made about the similarity of systems
- A single security server used for each system
- The policy is wholly contained and analyzable
- Labeled networking capabilities exist but are not able to handle disparate policies

Expressing network-wide policies

- Security goals are inherently network-wide
- Data is stored in lots of places all over a network
- Information flow policies should address an entire network
- Analyzability is a must

Expressing network-wide policies

- Explicitly separating types
  - For example, external_webserver_httpd_t
  - Does not scale well
  - Requires unified namespace
  - No automatic policy distribution

Expressing network-wide policies

- Implicitly separating types
  - Using a new operator to separate types (@)
  - Automatically inherit permissions to make policy development easier
  - Harder to subtract permissions from some systems
  - Allows a single policy to be written and distributed to all systems
  - Analyzable using current technologies

Expressing network-wide policies

- Translating type interactions between systems
  - Allows each system to use current policies
  - When one system contacts another a translation occurs
  - Translation allows each system to specify how other systems interact with it independently
  - New analysis tools would be needed
  - Preferred way to solve the problem

Distributing multi-system policies

- No matter which method is used distribution is necessary
- Some distribution is easy
- Implicit separation of types allows automatic distribution
- Explicit separation and translation require intervention
- Atomicity and synchronization are a concern

Conclusions

- A single coherent policy is necessary for analysis of network-wide policy
- The policy must be intelligently partitioned and distributed
- Translation is the only mechanism that allows the use of upstream policy
- Other approaches do not scale well