Overview of NSA Security Enhanced Linux

James Morris

jmorris@namei.org

FOSS.IN/2005

Bangalore, India

What is SELinux?

- Fine-grained Mandatory Access Control (MAC)
- Strongly separated domains
- Provides confinement of:
    - malicious code
    - flawed code
    - user error/intention
- Enforces least privilege
- Protects integrity
- Flexible policy

Discretionary Access Control (DAC)

- Traditional Unix security model
- DAC is inadequate:
  - Decisions only based on user identity and ownership
  - No protection against malicious or flawed software
  - Each user has complete discretion over own objects
  - Only two major categories of users: admin and other
  - Coarse-grained and too much privilege
  - Unbounded privilege escalation

Mandatory Access Control (MAC)

- "Missing link" of security in general OSs
- Primary features:
  - Administratively-set security policy
  - Control over all processes and objects
  - Decisions based on all security-relevant information

MAC Implementation

- Traditionally inflexible, maps poorly to general case
- More than just Multilevel Security (MLS):
  - Enforce integrity, least privilege
- Generalized MAC via Type Enforcement (TE)
- Transparent to applications
- Decompose administrator role
- Policy flexibility

SELinux Framework

- Composition of several security models
  - Role-Based Access Control (RBAC)
  - Type Enforcement (TE)
  - Optional Multilevel Security (MLS)
- Separation of mechanism and policy
- Can support further models as required

Type Enforcement (TE)

- Domains for processes, types for objects
- Control access to objects by domains
- Control interactions between domains
- Control entry into domains
- Bind domains to code

Types

- Attributes assigned to objects and domains
- Things of the same type are security-equivalent
- Examples:
  - httpd_t (apache execution domain)
  - httpd_log_t (apache log files)
  - httpd_config_t (apache configuration files)
  - httpd_sys_content_t (web content)

Type Enforcement Rules

- TE rules defined in policy
- Include object labeling, and access control rules
- Examples:

· allow httpd_t httpd_log_t:file { create ioctl read getattr lock append };

· allow httpd_t httpd_config_t:file { read getattr lock ioctl };

· allow httpd_t httpd_sys_content_t:file { read getattr lock ioctl };

Role Based Access Control (RBAC)

- Roles are attributes assigned to domains, e.g.
  - sysadm_r
  - system_r
  - user_r
- Specifies domains that can be entered by each role
- Specifies roles that are authorized for each user
- Initial domain associated with each user role

Current Status

- Merged into upstream kernel during 2.5 series
- Adopted by several distributions
- Targeted policy for network facing services
- Possibly millions of users
- Basic tools

Current Development

- MLS for production use
- Certifications: LSPP, RBACPP at EAL4+
- Multi-category Security (MCS)
- Reference policy
- Modular policy

Future Developments & Participation

- Better tools
  - Policy development and analysis
- High level policy language
- Distributed security management
  - User management, policy distribution, logging
- Desktop
- Application-level

More Future Developments...

- Networked filesystems
- Hardware security (TPM):
  - Verified boot
  - Integrity verification
  - Remote attestation
  - Trusted path
- Virtualization
- Cryptographic policy

Resources

- NSA SELinux Pages
  http://www.nsa.gov/selinux/
- SELinux for Distributions
  http://selinux.sourceforge.net/
- Mailing Lists (see above)
- IRC: irc.freenode.net #selinux
- SELinux Symposium 2006 (Feb/Mar)
  http://www.selinux-symposium.org/

Acknowledgments

- Much of the source material for these slides was derived from existing NSA presentations:
  - http://www.nsa.gov/selinux/info/docs.cfm