Extending Linux for MLS

George Wilson, Loulwa Salem
IBM Linux Technology Center

Klaus Weidner, atsec

Agenda

- Introduction and History
- LSPP/RBACPP/CAPP/EAL4+ Development Effort
- Current State of Development Effort
- Opportunities for Enhancement
- Conclusion
- Questions

Introduction

- The LSPP/RBACPP/CAPP/EAL4+ Common Criteria evaluations are helping realize of the fruits of SELinux development
- Members of the Open Source community have sponsored CC evaluations to bring to software security that meets federal government standards
- Enablement of hardware and application solutions
- Commoditization of technologies in a supported Open Source OS along with the skills to use it
- Governments and businesses deserve modern and secure software at a reasonable cost

Introduction (cont'd)

- The Common Criteria are a set of multinational security evaluation criteria
- Define seven Evaluation Assurance Levels EAL1-EAL7
- Mutual recognition up to EAL4
- CC define functional and assurance requirements
- Protection Profiles
  - Predefined set of functional and assurance requirements
  - Controlled Access Protection Profile applies to DAC based access
  - Label Security Protection Profile applies to MAC based access
  - New profiles evolving
- Common Criteria certified products required for national

Introduction (cont'd)

- LSPP is the Labeled Security Protection Profile
  - Requires labels, MLS rules, user data import/export controls, and audit
  - Requires minimum EAL3
  - Similar to old TCSEC B1
- RBACPP is the Role Based Access Control Protection Profile
  - Specifies role characteristics, management of roles, hierarchical roles, and self-test
  - Requires minimum EAL2
  - Precedent for combining with LSPP

Introduction (cont'd)

- LSPP/RBACPP/CAPP/EAL4+ is the latest in an progressive series of certifications
- Uses SELinux to fulfill specific security goals
- SELinux is used in this instance to specify well defined security policies for Linux – not simply a backup to contain unsafe applications
- Many have the habit of turning off SELinux to work around restrictions – if using SELinux MLS, disabling SELinux allows everybody to bypass the MLS rules
- New features, not just new restrictions, such as labeled printing, labeled networking, and polyinstantiation are value adds for users

Linux and the Common Criteria

- Until 2003, many people believed that Linux would not be able to be CC certified
- Now, four years later, no other operating system has received more Common Criteria certificates than Linux®
  - Two distributions (Novell SUSE and Red Hat)
  - Two different kernel versions (2.4 and 2.6)
  - Many different hardware platforms
    - IBM® Pentium, XEON, and Opteron systems
    - IBM pSeries®, iSeries™, and zSeries® systems
    - HP Pentium, XEON, and Itanium systems
    - SGI Itanium systems
  - Two certifying agencies (BSI & NIAP)
  - Assurance levels up to EAL4 augmented by ALC_FLR.3

Linux Features Added

- Existing but Modern Features Only – SLES 8 – EAL2+
  - Broken PAM modules fixed
  - Documentation added – many manpages
- LAuS (Linux Audit Subsystem) 2.4 – SLES 8 and RHEL 3 - CAPP/EAL3+
- LAuS 2.6 – SLES 9 - CAPP/EAL4+
- LAF (Lightweight Audit Framework) – RHEL 4 & SLES 10 - CAPP/EAL4+
- Many more features for LSPP/RBACPP/CAPP/EAL4+

LSPP/RBACPP/CAPP/EAL4+

- All development is or will soon be upstream
- Targeted for RHEL 5 GA
- Includes updated packages beyond GA
- Evaluations scheduled to complete this summer
- Mandates use of MLS policy; requires extensive control and audit of data import/export
- Makes use of SELinux MLS, TE, and RBAC features
- First CC evaluation with SELinux included in TOE Security Enforcing Functions

LSPP Enhancements

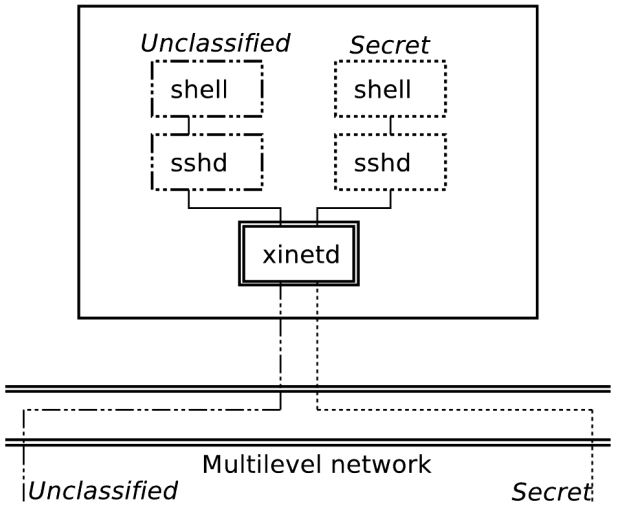
- Base SELinux Enhancements: Augmentation and test of MLS mode, user and role management utilities
- MLS Policy
- Labeled Networking
- MLS-Aware Network Services: racoon SA management; xinetd labeled networking, ssh role/level selection
- Labeled Print
- Polyinstiantiation (Multi-Level Directory) support for MLS
- Label Translation Daemon
- Multi-level Cron
- Labels in Audit Records
- Additional Audit Events & Audit Filtering on Labels

LSPP Community

- Show of hands!
- Model public/private partnership
- A true open source effort
- All development takes place on open mailing lists
- Weekly open telecon
- More than 60+ participants from 14+ different organizations over time
- Code flows through upstream maintainers
- Fedora Rawhide provides daily builds
- Red Hat hosts test packages for features pending maintainer acceptance
- Real users provide feedback during development

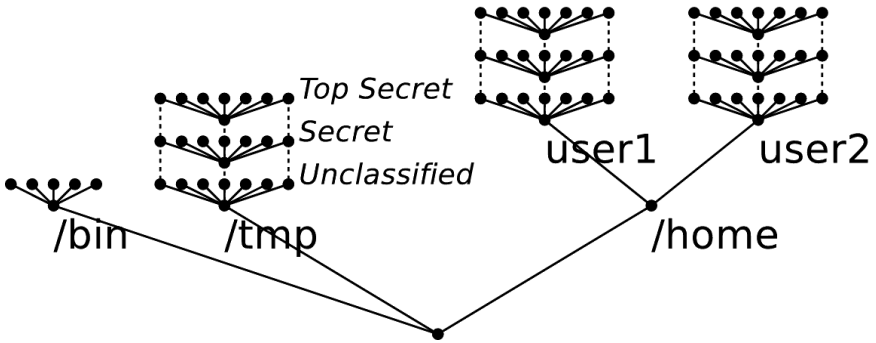
Labeled Networking

- network packets have associated labels
  - CIPSO (per-packet options)
  - Labeled IPsec (stored in corresponding SA)
- sshd launch via xinetd
  - xinetd is label aware
  - launch sshd at level of network connection
  - verify level is permitted for user



VFS Polyinstantiation

- Legacy programs stop working in MLS mode without polyinstantiation
  - they expect to be able to write to /tmp, $HOME
  - MLS restrictions forbid this
- Multiple versions of dirs at single location
  - implemented as PAM module using VFS namespaces and *unshare*(2) syscall



Polyinstantiation issues

- “Where did my home directory go?”
  - The content of $HOME is different when changing level or categories
  - $HOME is empty when first using a level
  - more restrictive than required by LSPP (user can't see files that would be readable, for example the “Unclassified” home dir when logged in at “Secret”)
- “mount/unmount is broken?”
  - admin changes seem to not take effect
  - can unmount “/var/log” in running system
  - use “mount --make-shared” etc.

Modifications to Existing Tools

- Multilevel Cron
  - MLS_LEVEL environment variable (paper still refers to SELINUX_ROLE_TYPE, this was changed recently)
- Audit Enhancements
  - add subject & object labels to records
  - admins need help understanding denials
    - audit2why
- Labeled Print
  - Human readable labels, MLS restrictions

Critical programs not designed for MLS

- Typical SELinux approach:
  - keep programs mostly unmodified
  - add additional restrictions to improve security
- MLS and RBAC are now basic security policies
  - OS is expected to enforce them
  - Applications can violate MLS rules if SELinux permits too much access

Critical programs not designed for MLS (cont'd)

- SELinux type attributes add override privileges
- “Confused Deputy” problem
- Example: sshd is not aware of MLS restrictions
  - Privileges include “mlsfileread” and “mlsfilewrite”
    - would be nice to restrict privileges to code paths that need them (PAM modules)
  - does not have special network privileges required to protect against port forwarding MLS exploits

Critical programs not designed for MLS (cont'd)

- Not obvious where override privileges come from
  - *apol* shows privileges, but these are not easily visible in the refpolicy source for the specific app
- Restrictions on permitted transition limit exposure
  - *consoletype_t* has “mlsfileread” capability, but *user_t* is not permitted to transition to it

Newrole and PTY information flow

- No MLS check between ends of PTY pair
  - adding a check was considered too invasive
- Simple exploit to declassify information
  - use “expect”-style program to run “newrole -l”
  - needs the user's password
- Fix: change “newrole -l”
  - now refuses to change level on insecure terminal devices
  - */etc/selinux/mls/contexts/securetty_types*

Executable type transitions

- *_exec_t* automatic domain transitions change privileges
  - Can be used to add privileges or to remove them
  - similar to SUID programs
- Binaries are protected
  - kernel/glibc “atsecure” mechanism activated when changing types
- Scripts need additional protection
  - kernel doesn't support SUID scripts, but it does permit scripts to do type transitions

Executable type transitions (cont'd)

- Environment contamination
  - dangerous $PYTHONPATH, $PERLLIB
  - use “perl -T”, “python -E”
  - don't use #!/usr/bin/env python
- Time of check/time of use race condition
  - interpreter re-opens script file
  - no fix currently available
- Don't add privileges for scripts
  - ok for removing privileges
  - *rpm_t* (used by *yum* script) has more privileges than *sysadm_t*

User and role management

- MLS policy constraints and overrides
  - easy to define new roles with specific privileges
    - Example: “backupadm” with read but not write override rights
- RBACPP requires “hierarchical roles”
  - defining roles in terms of other roles
- SELinux supports “dominates” operator
  - permitted to associate with union of types of dominated roles
  - does not grant privileges to default type for role

Conclusions

- MLS and TE features of SELinux provided much of the needed functionality
- Modifications needed to make legacy applications work in restrictive MLS environment
  - some features excluded from evaluated config
  - impose restrictions on existing programs
- New opportunities for Linux use
  - MLS for military and government systems
  - RBAC for medical and financial institutions

Conclusions (cont'd)

- Can help improve general security
- Future evaluations can improve on current status
- Beyond LSPP
  - We have considered MLOSPP. But community and commercial viability of specialized features is questionable.
  - EAL4 is likely the highest achievable by a general purpose OS, unless specifically designed
  - Higher levels also possible via separation kernel

Questions?

George C. Wilson
<gcwilson@us.ibm.com>

Klaus Weidner
<klaus@atsec.com>

Loulwa Salem
<loulwa@us.ibm.com>

Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of IBM Corporation or the atsec Corporation.
- All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries: IBM (logo).
- A full list of U.S. trademarks owned by IBM may be found at http://www.ibm.com/legal/copytrade.shtml. Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.