Extending SELinux to Meet LSPP and RBACPP Data Import/Export Requirements

Janak Desai <janak@us.ibm.com>, IBM Corporation
George Wilson <ltcgcw@us.ibm.com>, IBM Corporation
Chad Sellers <csellers@tresys.com>, Tresys Technology

Session Objective

The objective of this session is to highlight proposed extensions to SELinux to meet a subset of the Labeled Security Protection Profile (LSPP) and Role-Based Access Control Protection Profile (RBACPP) requirements for import and export of data and their impact on the usability.

Agenda

- LSPP/RBACPP data import/export requirements and their impact on usability
- Data import/export using terminals
- Polyinstantiated directories
- Multi-context aware cron

LSPP/RBACPP data import/export development items

- Extensions to networking subsystem
- Policy development
- Import/export on interactive terminals
- Polyinstantiated directories
- Multi-context aware cron
- Extensions to print subsystem
- Device allocation subsystem

Agenda

- LSPP/RBACPP data import/export requirements and their impact on usability
- Data import/export using terminals
- Polyinstantiated directories
- Multi-context aware cron

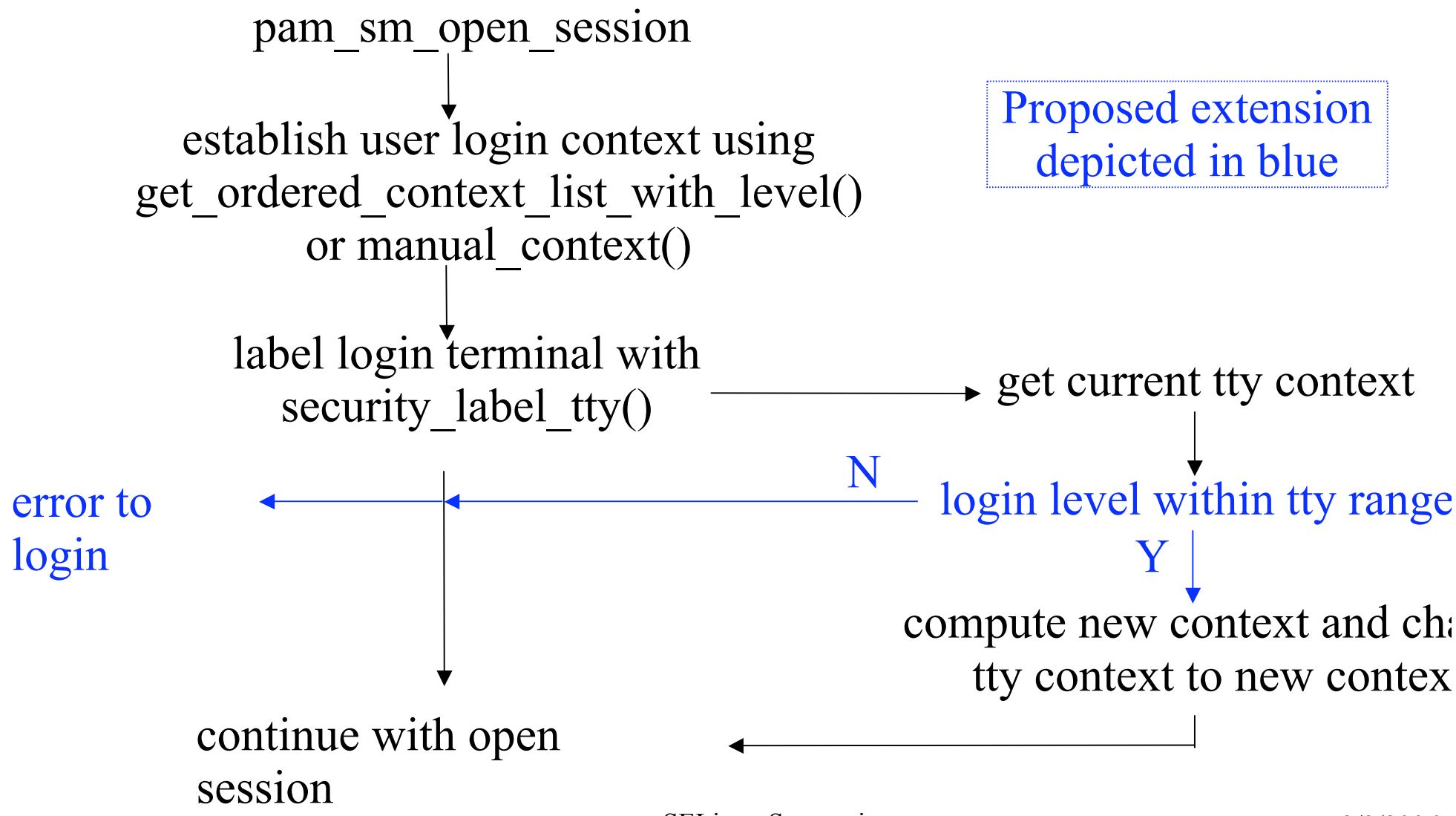
Data import/export requirement for terminals

- **LSPP requirements**
  - Security labels must be placed on i/o device objects.
  - Mandatory Access Control policy must be enforced on I/O devices as objects.
  - Login sensitivity label must fall within the terminal label range.
- **Current SELinux behavior**
  - Unconditionally change the terminal sensitivity label to match the login sensitivity label.
- **Proposed extension**
  - Extend SELinux PAM module to enforce the LSPP requirement.

Extension to SELinux PAM module

- **Description**
  - Obtain the terminal security context and extract the label range from the context.
  - Verify that the user login label falls within the terminal label range. If it doesn't, return an error and prevent the login from succeeding.

SELinux PAM module



Proposed extension depicted in blue

Agenda

- LSPP/RBACPP data import/export requirements and their impact on usability
- Data import/export using terminals
- Polyinstantiated directories
- Multi-context aware cron

Polyinstantiated directories

- The BLP *-property requires that higher level process cannot write to lower level directories.
- Enforcement of the *-property results in diminished usability because of the existence of public directories such as /tmp and /var/tmp.
- Enforcement of the *-property also makes home directories unusable for users that can login at different sensitivity levels.
- Proposed polyinstantiated directories improve the usability diminished by the enforcement of the *-property.

Polyinstantiated directories: description

- A Polyinstantiated directory provides different instances of itself based on the different security contexts of processes accessing it.
- Some trusted operating systems altered their path name translation functions to implement polyinstantiated directories.
- On SELinux, polyinstantiated directories are implemented using the kernel's per-process namespace feature.
- There are 2 main components of the feature:
  - A new system call, unshare, which dissociates a process namespace from the parent namespace
  - A new PAM module, pam_namespace, which sets up the namespace for a new user session

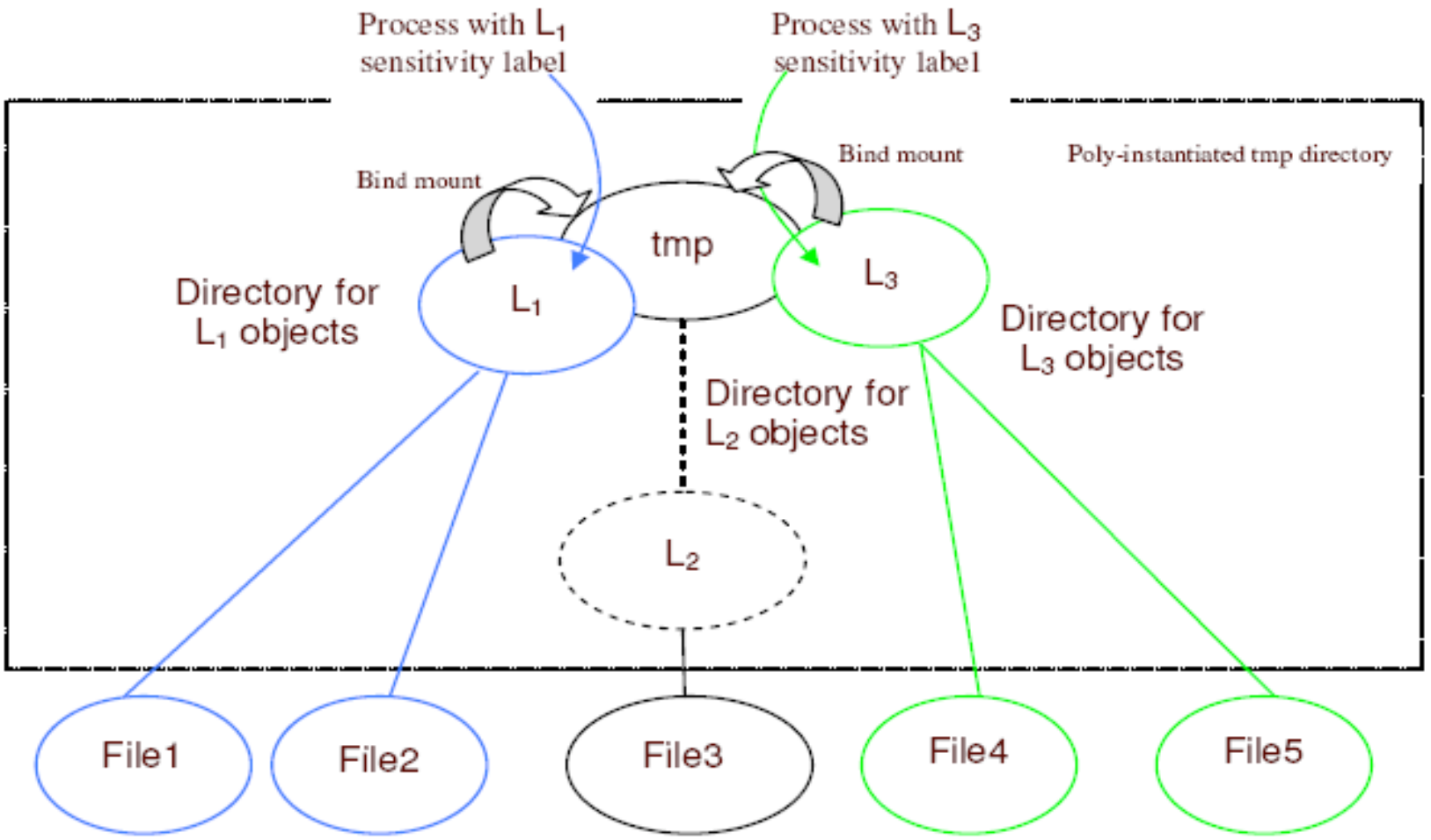
Polyinstantiated directory implementation: unshare

- Unshare allows a process to selectively disassociate parts of its execution context that are being shared with other processes.
- Unshare alters the current process as opposed to clone, which creates a new process with selectively shared parts of the execution context.
- Unshare allows the polyinstantiation mechanism to be localized in a PAM module, without which all session establishing programs such login, sshd, su, newrole and gdm would require modification.

Polyinstantiated directory implementation: pam module

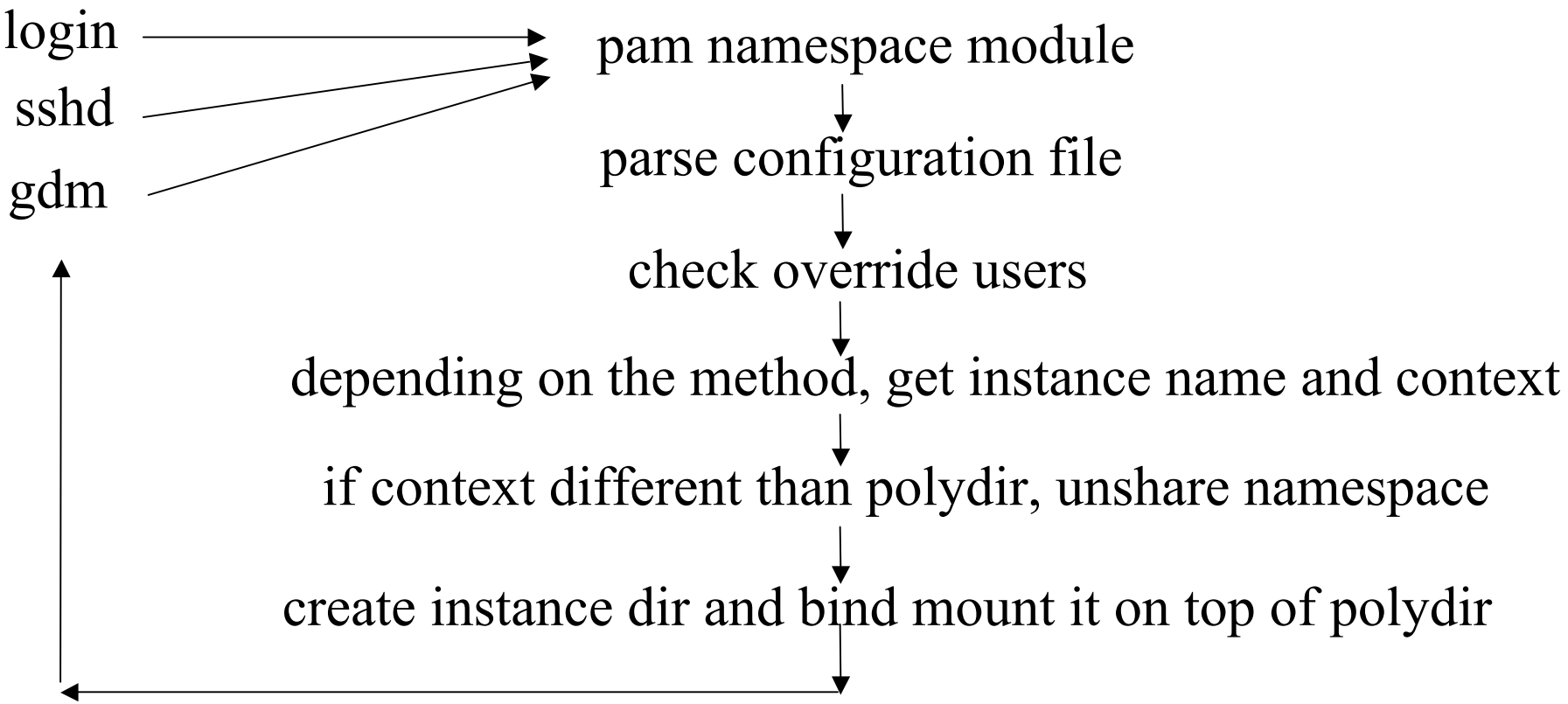
- Uses session management functions to setup customized namespace based on a configuration file.
- In the configuration file, an administrator can designate directories to polyinstantiate, the location of their instance directories and a list of override users for whom the polyinstantiation is not performed.
- After unsharing the namespace from the parent, the appropriate instance directory is located based on the type-member rules of the policy. The Instance directory is bind mounted on the top of the directory that is being polyinstantiated.

Polyinstantiated directory : example



Polyinstantiation process

```
#poly dir    instance dir          poly method    override users
#
/tmp         /tmp/.inst-$USER-     both           root,adm
```



Agenda

- LSPP/RBACPP data import/export requirements and their impact on usability
- Data import/export using terminals
- Polyinstantiated directories
- Multi-context aware cron

Multi-context aware cron

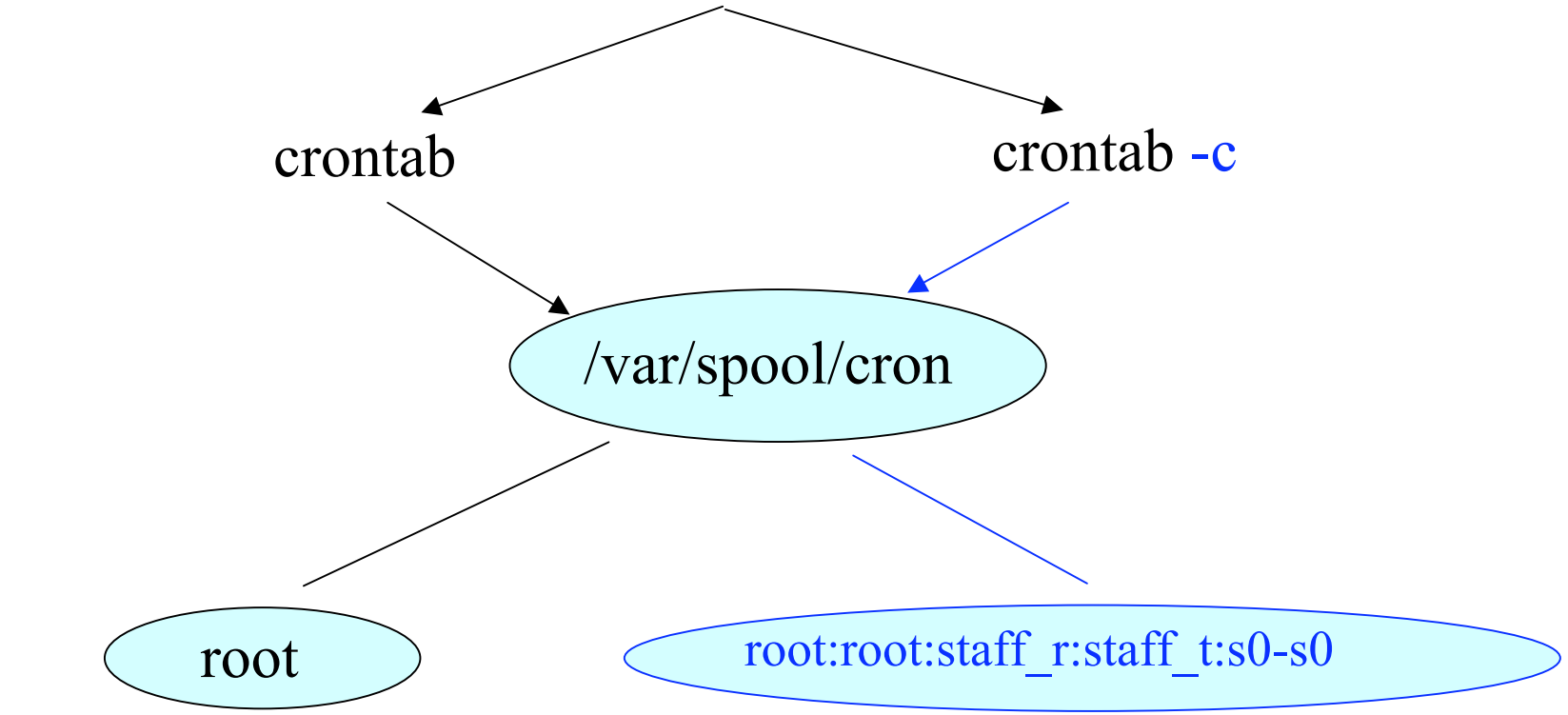
- RBACPP requires that the system associate appropriate user security attributes with subjects acting on behalf of the users.
- In the current SELinux it is possible for users to submit jobs after changing their security context.
- Cron jobs are executed in the derived context controlled by the system security policy.
- With the proposed extension, cron will assume the security context, under which the job was submitted, before processing the job for a user

Multi-context aware cron : implementation

- Instead of creating the cron job filename with just the user name, multi-context aware cron appends the user security context to the user name.
- When processing the job, the cron daemon obtains the security context from the cron job filename.
- If the system security policy allows it, the process context is set to the context obtained from the filename, before processing the cron job.

Multi-context aware cron example

process with uid 0 and context root:staff_r:staff_t:s0-s0



job executed with derived context

job executed with root:staff_r:staff_t:s0-s0 context

Summary

- SELinux provides a fully functional mandatory access control system. However, extensions are needed to satisfy specific LSPP and RBACPP requirements, and to improve the usability diminished from meeting those requirements.
- Work is progressing on these and other extensions in collaboration with SELinux open source community.