Security-Enhanced Linux

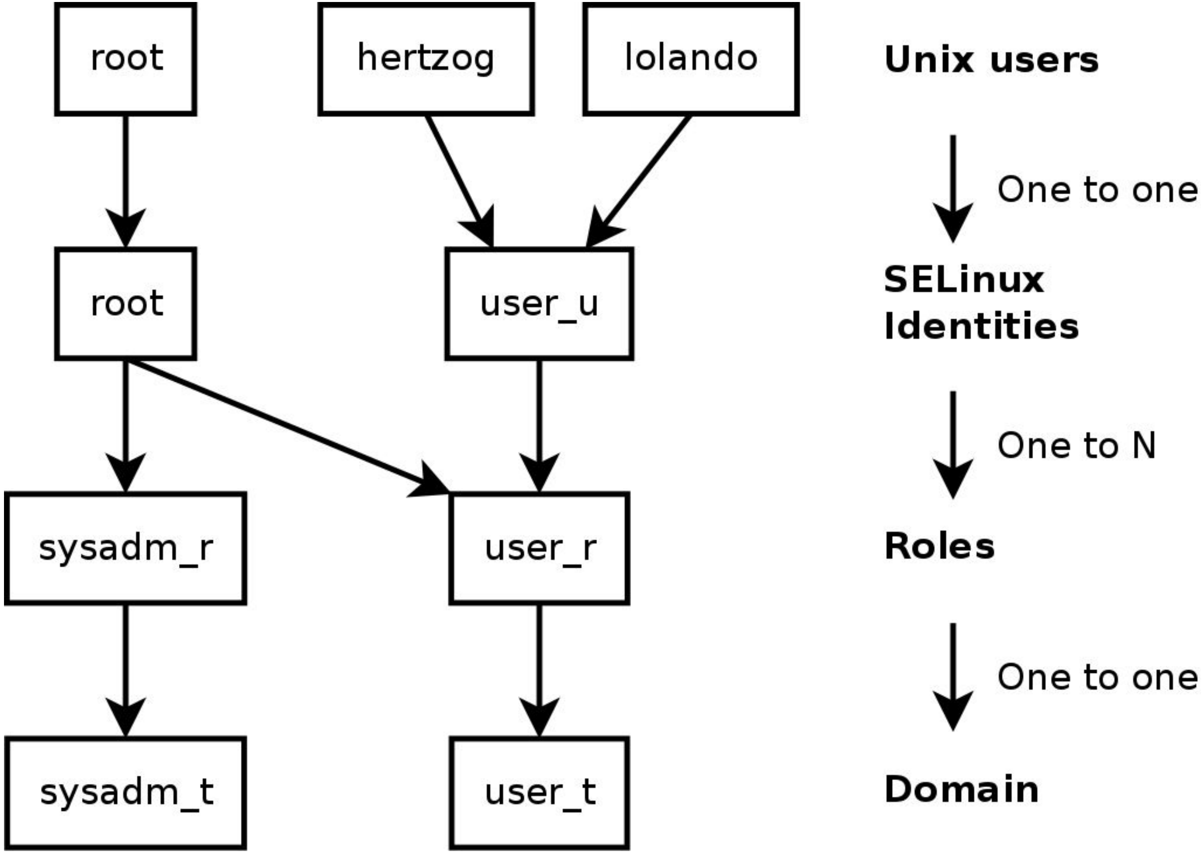
Solomon Rubin & Justin W. Flory

What is it?

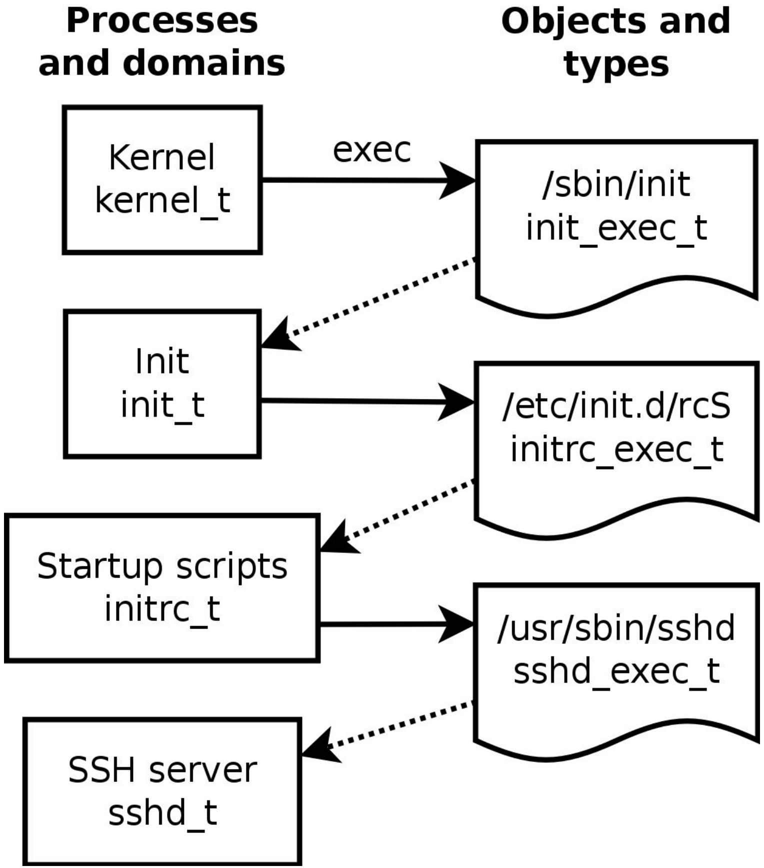
- Powerful Linux kernel module for access control
- Uses system of “contexts” rather than Unix-style permissions
- Allows for higher security even when multiple programs use elevated privileges
- Mostly used to confine daemons so that there is more defined data access control
- Used heavily in Android systems

SELinux Contexts

- A context defines the parameters for SELinux access control
- **User**: An identity that is authorized for a specific set of roles. A user may have more than one role attached. Unlike a Linux User, the SELinux user never changes even if their context changes (i.e. su or sudo).
- **Role**: Similar to a Unix permission group, a role allows access to the file for a specific set of SELinux users.
- **Domain**: Every role has a single domain in which it is allowed to run. Generally it is inherited by the user's domain
- **Type**: Defines a domain for files. Policy rules define how types can access each other. Ie. domain accessing another domain.
- **Level**: Security sensitivity levels



Contexts and Users



Processes and domains

Objects and types

Kernel
kernel_t

exec

/sbin/init
init_exec_t

Init
init_t

/etc/init.d/rcS
initrc_exec_t

Startup scripts
initrc_t

/usr/sbin/sshd
sshd_exec_t

SSH server
sshd_t

Domains, Objects, and Types

Where art thou, SELinux?

- While not exactly new, support for SELinux varies across Linux distributions
  - **Found in**: RHEL/CentOS 4 or later, Fedora Core 2 or later, Ubuntu 8.04 or later, SUSE Enterprise 11 or later
  - **Not found in**: Arch Linux (but in AUR), and surely others
- Heavily found in Android since 4.3
  - Varying **policies** used to contain processes each other and prevent them from accessing other parts of the system that they shouldn’t need access to

SELinux for Mere Mortals! (on your time)

Enforcing vs. Permissive

- **Enforcing**: SELinux policies are strictly enforced. Any SELinux user or domain without the correct contexts for a file will be denied access.
- **Permissive**: Opposite to **enforcing**; SELinux, while it still exists in the kernel, will not enforce the policies set in place.
    - Will log incidents to a log file
    - Useful for debugging

Please, please, do not setenforce=0

- …do not `setenforce=0` as a permanent solution
  - Opens system up to vulnerabilities and other risks
- In the past, documentation was more sparse / harder to find
  - Now, it is more common
- Easy to find a solution for **making it work** with whatever problem you have versus turning it off
  - Turning it off will fail you in a Red Hat certification exam (…probably)
- stopdisablingselinux.com

Quick introduction to SELinux

Video series

Understanding
SELinux - part 1 of 3

RHAT
CERTIFICATION

FREE RHCSA / RHCE
Video Training Videos

Additional readings

- SELinux User and Administrator’s Guide (via Red Hat)
- Official SELinux wiki
- Arch Linux Wiki (as always)
- HowTo guide from CentOS Wiki
- SELinux on Android