A brief overview of SELinux

Ralph Angenendt <ralph@centos.org>

SELinux in CentOS 5

- Short overview of old security model
- What is available?
- Policies, Booleans and Modules
- Tools to interact with SELinux
- Confine a self written webserver with the available toolchain

So what's old?

- rwxr-xr-x is the classical model of giving rights to users (or take them away)
- Simple model which can be easily taught to beginners – thus chmod 777 doesn't have to happen
- KISS
- But ...
- It's too simple for complex environments

rwxrwxrwx

- Problematic in complex setups
  - Kernel 2.6 allows 65535 groups the user can be in
  - But …
  - Using NFS leaves you with 16
- A little guesswork
  - /var/www/html is owned by apache
  - Group content may read and write there
  - Group backup may only read
  - Solve that …

Captain ACL to the rescue

- Modern file systems can store extended attributes
- EAs store metadata in them
- So why not store access control lists?
- Great. Now we can assign more than one user or group to a file or directory
- This helps us to model complex structures
- The problem from last slide is solvable

Enter SELinux

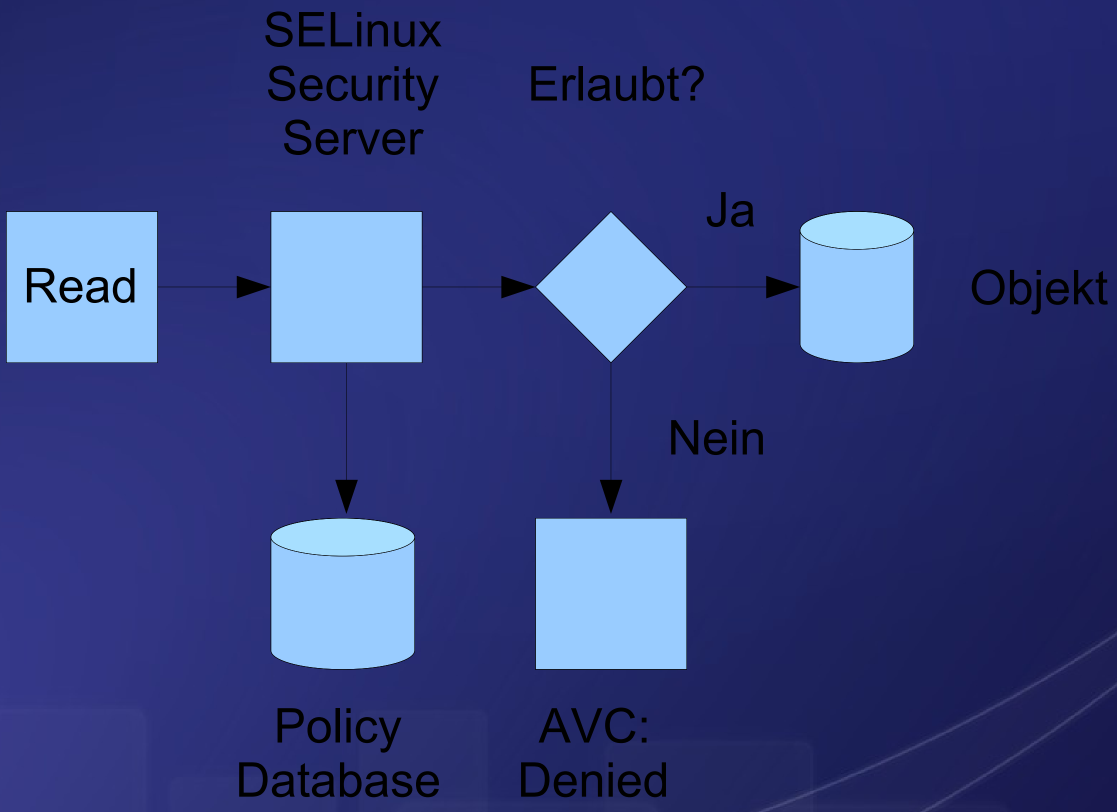
- Rethink who can do what where
- OLD: User controls who may do what to the data (with restrictions)
- NEW: Mandatory access system
  - All things are labeled with a context
  - User has to be able to access that context
  - Otherwise he is not able to change the file
  - compromised process is not able to access files which have access to “other” (rwxrwxrwx)

What's more?

- SELinux also offers you an RBAC system
  - Access rights to objects are given to roles
  - Roles can be modeled after your business model (management, HR, finance, techies)
- And Multi Level Security
  - Modeled after DOD requirements
  - Unclassified → Confidential → Secret → TOPS
  - Objects get classifications, Subjects get Clearance Levels

So how does it work?

- Overview:



So how does it work? (II)

- Three modes of operation
  - Enforcing
  - Permissive
  - Disabled
- Two policy modes (if enforcing or permissive)
  - strict
  - targeted
  - targeted is default

Which tools do we have?

- setenforce and getenforce
- chcon
- restorecon
- semodule
- semanage
- fixfiles
- system-config-selinux
- ls -Z to see security contexts (ps also knows)

Hands on

- Example:
  - httpd is restricted to /var/www/html
  - pages should be served out of /data/
  - “chcon -R --reference=/var/www/html /data” changes the security context of files in /data
  - httpd is able to serve files from there.

Booleans

- Clever way to interact with the policy
- No need to recompile policy
- getsebool -a shows all available booleans
- Example:
  - Users can serve pages out of homedirs
  - Management doesn't want that
  - setsebool -p httpd_enable_homedirs off
  - Voilà

Other Booleans

- allow_execstack
- allow_ftpd_use_cifs
- httpd_ssi_exec
- samba_share_nfs
- httpd_can_network_connect_db

SELinux modules

- Insert new modules into policy
- Without recompiling policy
- Use audit2allow to write new policy modules
- Reads avc:denied messages
- semodule manages modules (loads, unloads, updates)
- Example: vsftpd should be able to read httpd_syscontent_t directories

audit2allow

- setenforce= 0, run vsftpd, collect avc: denied

```
grep vsftpd /var/log/audit/audit.log | audit2allow -m local
module local 1.0;

require {
        type ftpd_t;
        type httpd_sys_content_t;
        class dir { read search getattr };
        class file { read getattr };
}

#============= ftpd_t ==============
allow ftpd_t httpd_sys_content_t:dir { read search getattr };
allow ftpd_t httpd_sys_content_t:file { read getattr };
```

And Now!

- Demotime!
- Questions!
- Answers!
- Thank you very much!