Do not turn it off: SELinux

Open Source Days 2012, Copenhagen

Robert Scheck

fedora™

Robert Scheck

Fedora Package Maintainer and Provenpackager

Fedora Ambassador and Ambassador Mentor

Part of Fedora Websites and Translation teams

Open Source Contributor and Software Developer

Mail: robert@fedoraproject.org
Web: http://fedoraproject.org/wiki/RobertScheck

What is SELinux?

- Security-Enhanced Linux
- Implementation of FLASK concept (Flux Advanced Security Kernel)
- Access control on resources in the meaning of Mandatory Access Control (MAC)
- Mostly developed by NSA and Red Hat
- Licensed under GNU General Public License

fedora

Linux Access Control

- Linux access control involves
  - kernel controlling
  - processes (running programs) and access to
  - resources (files, directories, sockets, ...)
- For example:
  - web server process can read web files,
  - but not `/etc/shadow`
- How are these decisions made?

Standard Access Control

- Processes and files have security properties
  - process: user/group (real and effective)
  - resources: user/group and access bits
    - read, write and execute for user, group and other
- Policy is hard-coded in the kernel
- Example: Can Firefox read my private SSH key?
  - `robert 3127  1 5 10:00 ?  00:00:29 firefox`
  - `-rw------- 1 robert users 993 Feb  6  2005 id_rsa`

Standard Security Problems

- Access is based on users' access
- Example: Firefox can read SSH keys
  - generally has no reason to read them, but
  - if compromised can – potentially disastrous
- Fundamental problem:
  - Security properties are not specific enough
  - Kernel can not distinguish applications from users

Standard Security Problems

- Processes can change security properties
- Example: Mail files readable only only by me
  - Evolution can make them world readable
- Fundamental problem:
  - User definable access control, also called Discretionary Access Control (DAC)
  - Processes can adapt or ignore security policy

Standard Security Problems

- Only two privilege levels: user and root
- Example: Apache privilege escalation
  - Apache bug allows obtaining root shell
  - Entire Linux system is compromised
- Fundamental problem:
  - Simplistic security policy
  - No way to enforce least-privilege

Solution: SELinux

- SELinux adds additional access control
  - new security properties on processes/resources
  - flexible security policy that can be changed
- Kernel and application based enforcement
- Designed to address security problems
  - mandatory (Mandatory Access Control, “MAC”), least-privilege and fine-grained
  - no all powerful root
- Transparent to applications

SELinux Access Control

- SELinux has 3 forms of access control
  - Type Enforcement (TE), primary mechanism
  - Role-Based Access Control (RBAC)
  - Multi-Level Security (MLS)
- Configurable via policy language
  - central configuration files control all access
  - several policies (targeted, mls, minimum)
- All access is denied by default

SELinux Security Properties

- Processes and files have a security context
  - `robert_u:staff_r:firefox_t:s0`
  - `robert_u:object_r:user_home_t:s0`
  - Benutzer:Rolle:Typ:Level
- The key field is the type
  - used to implement Type Enforcement
- Other fields used for RBAC and MLS
  - more on these later

SELinux Security Properties

- Several utilities modified for SELinux
- The “Z” option usually used to view contexts
- Examples:
  - `ps auxZ` (view contexts of processes)
  - `ls -laZ` (view contexts of files and directories)
- Output examples of “ls -Z”:
  - `----------. system_u:object_r:shadow_t:s0    /etc/shadow`
  - `-rwxr-xr-x. system_u:object_r:udev_exec_t:s0 /sbin/udevd`

Introduction: Type Enforcement

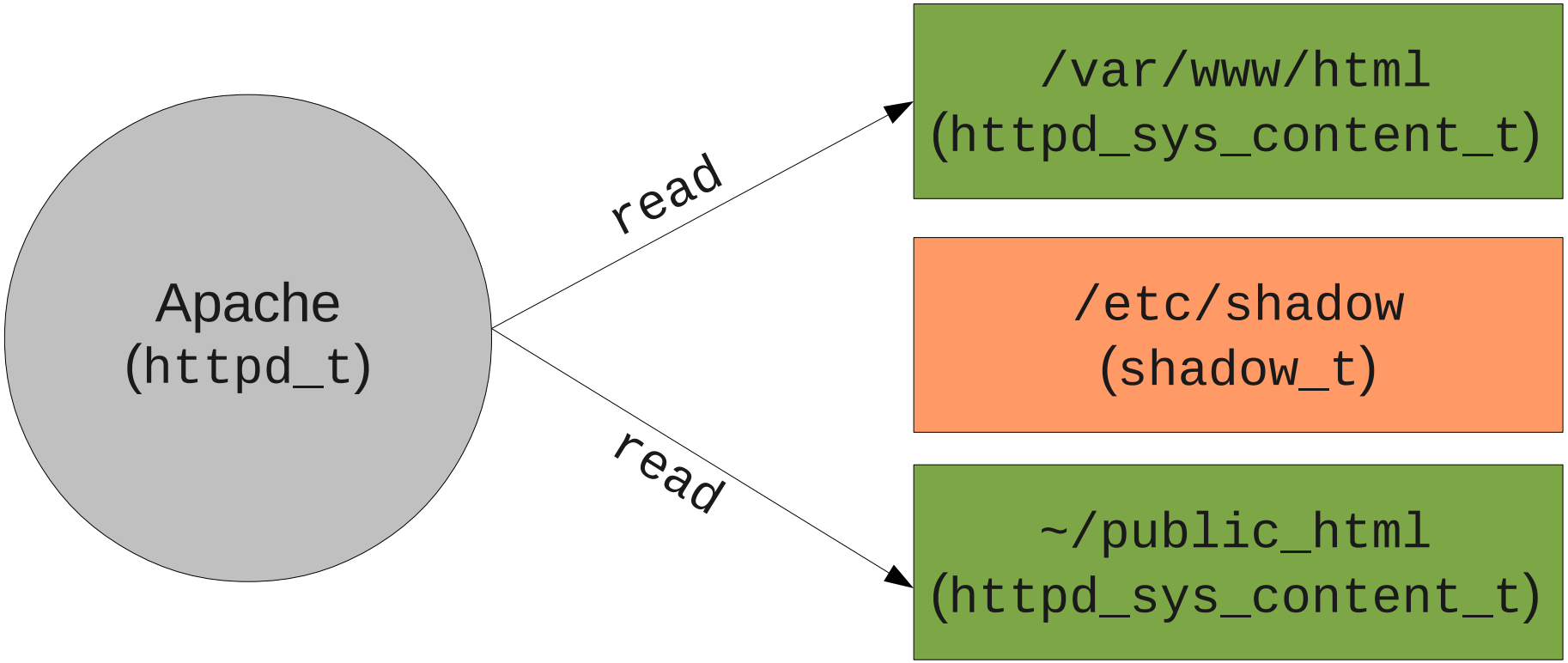
- Based on a single security property: type
  - applied to all processes and resources
  - represents all security relevant information
- Types are assigned to processes & resources
  - Apache processes → `httpd_t`
  - /var/www/html/index.html → `httpd_sys_content_t`
- Access is allowed between types,
  - e.g. `httpd_t` can read `httpd_sys_content_t`

Introduction: Object Classes

- Object classes specify the details of access
- Resources are divided into classes
  - e.g. `file`, `lnk_file`, `dir`, `socket`, `process`
- Each class has permissions,
  - e.g. for `file`: `read`, `write`, `execute`, `getattr`
- Full access in Type Enforcement:
  - `allow httpd_t httpd_sys_content_t:file read;`

Overview: Type Enforcement

```
allow httpd_t httpd_sys_content_t:file read;
```



Concept: Type Enforcement

- Access is allowed exclusively by type
  - many processes and resources have same type
    - simplifies policy by grouping
  - policies with same type have same access
    - same for resources (files)
- Process types are also called “domains”
  - sometimes applied to resources, e.g. sockets
- Different resources can have same type

Assigning Initial Types

- Files and directories:
  - configuration file specifies default context
    - so-called „file contexts“ (`*.fc`)
    - regular expressions, `/usr/(.*/)?bin(/.*)? → bin_t`
  - Inherited from parent directory at runtime
- Applications can explicitly set context
  - `chcon` – utility to set contexts (→ `chown`)
  - `passwd` – maintains context on `/etc/shadow`

Assigning Process Types

- Process types are
  - (default) inherited from parent process
  - set by policy (type transition rule)
  - set by application (e.g. `login`)
- Examples:
  - `bash` (`user_t`) → `ls` (`user_t`)
  - `init` (`init_t`) → httpd init script (`initrc_t`) → `httpd` (`httpd_t`)
  - `login` (`login_t`) → `bash` (`user_t`)

Type Transition Rules

- Type Transition rules set process types using:
  - parent process type and executable file type
  - similar to setuid()
- Example: starting name server
  - Policy rule:
    - `domain_auto_trans(initrc_t, named_exec_t, named_t)`
  - Parent process (`initrc_t`)
  - Executable file type (`named_exec_t`)
  - Result: `named_t`

Type Transition Notes

- Primary reasons for setting process type
  - ensures applications run in correct domain
  - does not require application modification
- Must be allowed by policy
  - e.g. Apache can not start processes in `init_t`
  - prevents applications from gaining privilege
- Binds specific executable to a domain
  - e.g. only `/usr/bin/passwd` can run in `passwd_t`

User Field in Security Context

- `robert_u:staff_r:firefox_t:s0`
- Not necessarily the same as the Linux user
- Often ends in "_u": `system_u`, `user_u`
- Not currently used in the "targeted" policy
- Files and directories:
  - user inherited from process
  - system processes create files with the file context `system_u`

Role Field in Security Context

- `robert_u:staff_r:firefox_t:s0`
- Used for Role-Based Access Control (RBAC)
  - role further restricts available type transitions
  - together with Type Enforcement (`user_r`/`user_t`)
- Usually ends with “_r”
- Resources get by default `object_r`
- Used in “mls” policy
  - `user_r`, `staff_r`, `secadmin_r`

MCS Level Field Details

- `robert_u:staff_r:firefox_t:s0`
- Used for multi-level security, short: MLS
  (or for multi categories security, short: MCS)
- Often hidden in “targeted” policy
- Identifies one level or range
  - single level: `s0`
  - range: `s0-s15:c0.c1023`
- Usually translated with labels
  - `s15:c0.c1023` → “SystemHigh”

SELinux Security Benefits

- Types capture important security information:
  - access is based on user *and* application function
  - transitions capture process call chains
- Processes run with least-privilege
  - only what is allowed for the type
  - e.g. `httpd_t` can only read web pages
- Privilege escalation tightly controlled
  - a compromise of Apache limited by policy

The “mls” Policy

- Policy with Bell-LaPadula support
  - model: Confidential information shall not be passed to non-confidential persons (thus: no read-up and no write-down)
- Intended for server only operating systems
  - no X-window support
  - limited to particular packages/services
- Certification of Red Hat Enterprise Linux in 2007 (with IBM) against LSPP, RBACPP & CAPP on EAL 4+

The “targeted” Policy

- Processes are by default unconfined
  - only “targeted” processes are confined
- Unconfined domains
  - by default user processes run in `unconfined_t`
  - system processes run in `initrc_t`
  - unconfined processes have same access as they would have without SELinux running
- Daemons with policy have a transition from `unconfined_t` to e.g. `httpd_t` (limited access)

Configuration Files

- SELinux configuration in `/etc/selinux`

```
-rw-r--r--. 1 root root  458 Aug 26  2010  config
-rw-r--r--. 1 root root 2271 Jul 22  2010  semanage.conf
drwxr-xr-x. 5 root root 4096 Jun  7 01:53 mls
drwxr-xr-x. 5 root root 4096 Jun  7 01:53 targeted
```

- `/etc/selinux/config` – policy and mode

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Configuration Files

- contexts: Default contexts for the system
- modules: Modules to build the policy
- policy: Compiled SELinux policy
- setrans.conf: MLS/MCS translations
- seusers: Mapping Linux-/SELinux users

```
$ ls -l /etc/selinux/targeted/
drwxr-xr-x. 4 root root 4096 Jun  7 01:53 contexts
drwxr-xr-x. 3 root root 4096 Jun  7 01:53 modules
drwxr-xr-x. 2 root root 4096 Jun  7 01:53 policy
-rw-r--r--. 1 root root  607 May 27 15:44 setrans.conf
-rw-r--r--. 1 root root  176 Jun  7 01:53 seusers
$
```

Kernel Boot Parameters

- Kernel parameters override settings in `/etc/selinux/config`
- `selinux=0`
  - boots the kernel with SELinux turned off
  - all files will no longer get created with file context
  - later SELinux usage requires a relabeling
- `enforcing=0`
  - boots the kernel in “permissive” mode
  - may not give same error messages as in “enforced”

“man pages” for “targeted”

```
httpd_selinux(8)       httpd Selinux Policy documentation      httpd_selinux(8)

NAME
       httpd_selinux - Security Enhanced Linux Policy for the httpd daemon

DESCRIPTION
       Security-Enhanced  Linux secures the httpd server via flexible mandatory
       access control.

FILE_CONTEXTS
       SELinux requires files to have an extended attribute to define the  file
       type.   Policy  governs the access daemons have to these files.  SELinux
       httpd policy is very flexible allowing users to setup their web services
       in as secure a method as possible.

       The following file contexts types are defined for httpd:

       httpd_sys_content_t
       - Set files with httpd_sys_content_t if you want httpd_sys_script_exec_t
       scripts and the daemon to read the file,  and  disallow  other  non  sys
       scripts from access.

       httpd_sys_script_exec_t
       - Set cgi scripts with httpd_sys_script_exec_t to allow them to run with
```

Modified System Utilities

- “Z” is the answer for SELinux
  - `ls -Z`
  - `id -Z`
  - `ps auxZ`
  - `lsof -Z`
  - `netstat -Z`
  - `find / -context=`

Modified System Utilities

- `cp`
  - inherits context from parent directory or sets the context based on the system standard
  - option “-a” keeps the source (original) context
- `mv`
  - keeps the source (original) context
- `install`
  - sets security context based on system defaults
- Exceptions via `restorecond`

SELinux Packages & Utilities

- libselinux is the default SELinux library
- libselinux-utils
  - `getenforce`: tells enforcing/permissive/disabled
  - `setenforce 0/1`: sets permissive/enforcing
  - `selinuxenabled`: SELinux status for scripting
  - `matchpathcon`: tells default context
  - `avcstat`: displays SELinux AVC statistics
- libselinux-python and libselinux-ruby
  - API bindings to libselinux

Policycoreutils

- genhomedircon, fixfiles, setfiles, chcat, restorecon, restorecond
- audit2allow, audit2why
  - show/understand SELinux AVC messages
- secon
  - see context of files and programs
- semodule, semodule_deps, semodule_link, semodule_expand, semodule_package
  - management of modules

Understand SELinux Messages

- Access Vector Cache (AVC)
  - `/var/log/messages` (without auditd)
  - `/var/log/audit/audit.log` (with auditd)

```
type=AVC msg=audit(1140184056.443:78): avc: denied { use } for ↩
  pid=2185 comm="mingetty" name="ptmx" dev=tmpfs ino=699 ↩
  scontext=system_u:system_r:getty_t:s0 ↩
  tcontext=system_u:system_r:kernel_t:s0 tclass=fd

type=AVC msg=audit(1166017682.366:876): avc: denied { getattr } for ↩
  pid=23768 comm="httpd" name="index.html" dev=dm0 ino=7996439 ↩
  scontext=user_u:system_r:httpd_t:s0 ↩
  tcontext=user_u:object_r:user_home_t:s0 tclass=file
```

Understand SELinux Messages

- AVC messages can get created for a variety of reasons
  - a mislabeled file (wrong context)
  - a process running under wrong context
  - a bug in the SELinux policy
    - basically an application goes down a code path that was never tested by the policy writer and gets unexpected AVC
  - an intruder

Understand SELinux Messages

- audit2allow
  - tool that generates policy “allow” rules from logs of denied operations
  - `audit2allow -i /var/log/audit/audit.log`
    - `allow httpd_t user_home_t:file getattr;`
- audit2why
  - translates SELinux audit messages into a description of why the access was denied
  - not very helpful to novice users, mostly used by policy developers

Analyzing AVC Messages

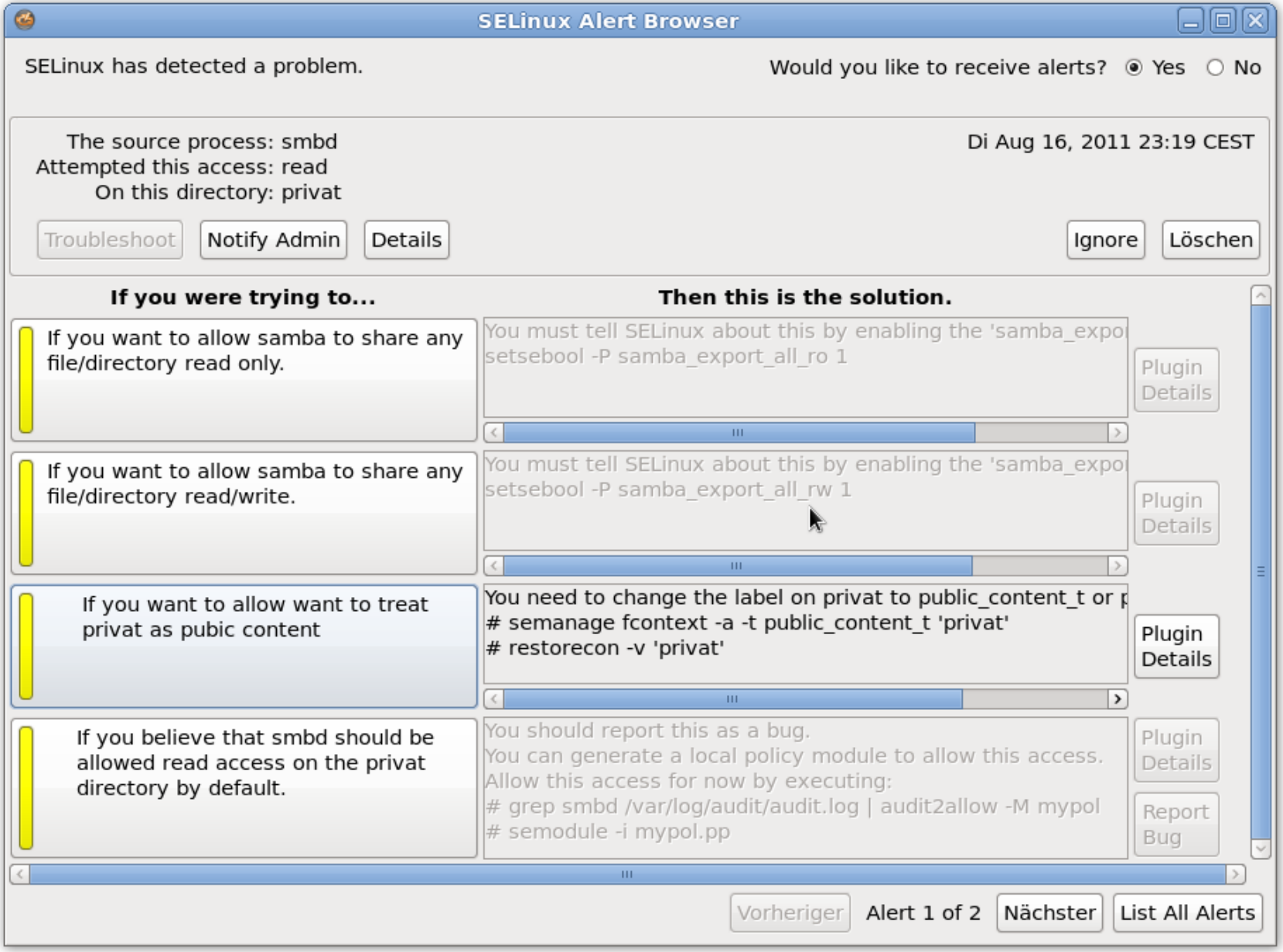
- AVC messages referring to files with `*:file_t`
  - major labeling problem, all files require labels
  - file was created when running `selinux=0`
  - perform relabeling of the file system
    - `touch /.autorelabel; reboot`
  - new disk? `restorecon -R -v /<mnt>`
- AVC messages containing `default_t`
  - probably a labeling problem
  - relabel with `chcon` or see above

Analyzing AVC Messages

- Many similar messages about the same file
  - usually indicates a labeling problem
  - example:
    - create file `/home/robert/resolv.conf`
    - `mv /home/robert/resolv.conf /etc/`
    - `ls -lZ /etc/resolv.conf`
    - confined domains will report errors when accessing `user_home_t`
    - `restorecon /etc/resolv.conf`

SELinux Troubleshoot Tool

- setroubleshoot
  - service listens to audit daemon for AVC messages
  - then processes plugin database for known issues
    - `/usr/share/setroubleshoot/plugins/`
  - displays knowledge base how to handle/solve
  - `sealert` can launch browser or analyze log files
  - configuration for e-mail notification possible
    - `/etc/setroubleshoot/setroubleshoot.conf`


SELinux Alert Browser
SELinux has detected a problem.
Would you like to receive alerts? Yes No
The source process: smbd
Attempted this access: read
On this directory: privat
Di Aug 16, 2011 23:19 CEST
Troubleshoot
Notify Admin
Details
Ignore
Löschen
If you were trying to...
Then this is the solution.
If you want to allow samba to share any file/directory read only.
You must tell SELinux about this by enabling the 'samba_expo
setsebool -P samba_export_all_ro 1
Plugin Details
If you want to allow samba to share any file/directory read/write.
You must tell SELinux about this by enabling the 'samba_expo
setsebool -P samba_export_all_rw 1
Plugin Details
If you want to allow want to treat privat as pubic content
You need to change the label on privat to public_content_t or p
semanage fcontext -a -t public_content_t 'privat'
restorecon -v 'privat'
Plugin Details
If you believe that smbd should be allowed read access on the privat directory by default.
You should report this as a bug.
You can generate a local policy module to allow this access.
Allow this access for now by executing:
grep smbd /var/log/audit/audit.log | audit2allow -M mypol
semodule -i mypol.pp
Plugin Details
Report Bug
Vorheriger
Alert 1 of 2
Nächster
List All Alerts

fedora

Missing AVC Messages

- Applications fail with no AVC messages
  - try to use `setenforce 0` – does it work?
- `dontaudit` rules avoid AVC messages
- Fedora 14+ and Red Hat Enterprise Linux 6
  - `semodule -DB  # --disable_dontaudit --build`
- Red Hat Enterprise Linux 5
  - `semodule -b /usr/share/selinux/targeted/enableaudit.pp`
  - `semodule -b /usr/share/selinux/targeted/base.pp`

Managing File Labeling

- chcon
  - fundamental utility used to change file contexts
    - `chcon -R -t httpd_sys_script_rw_t \ /var/www/myapp/data`
    - `chcon -t httpd_sys_script_t \ /var/www/cgi-bin/myapp`
  - modeled after chmod command
  - customizable types: no relabeling
    - `/etc/selinux/targeted/contexts/customizable_types`
- `touch /.autorelabel; reboot`
  - complete relabeling

Managing File Labeling

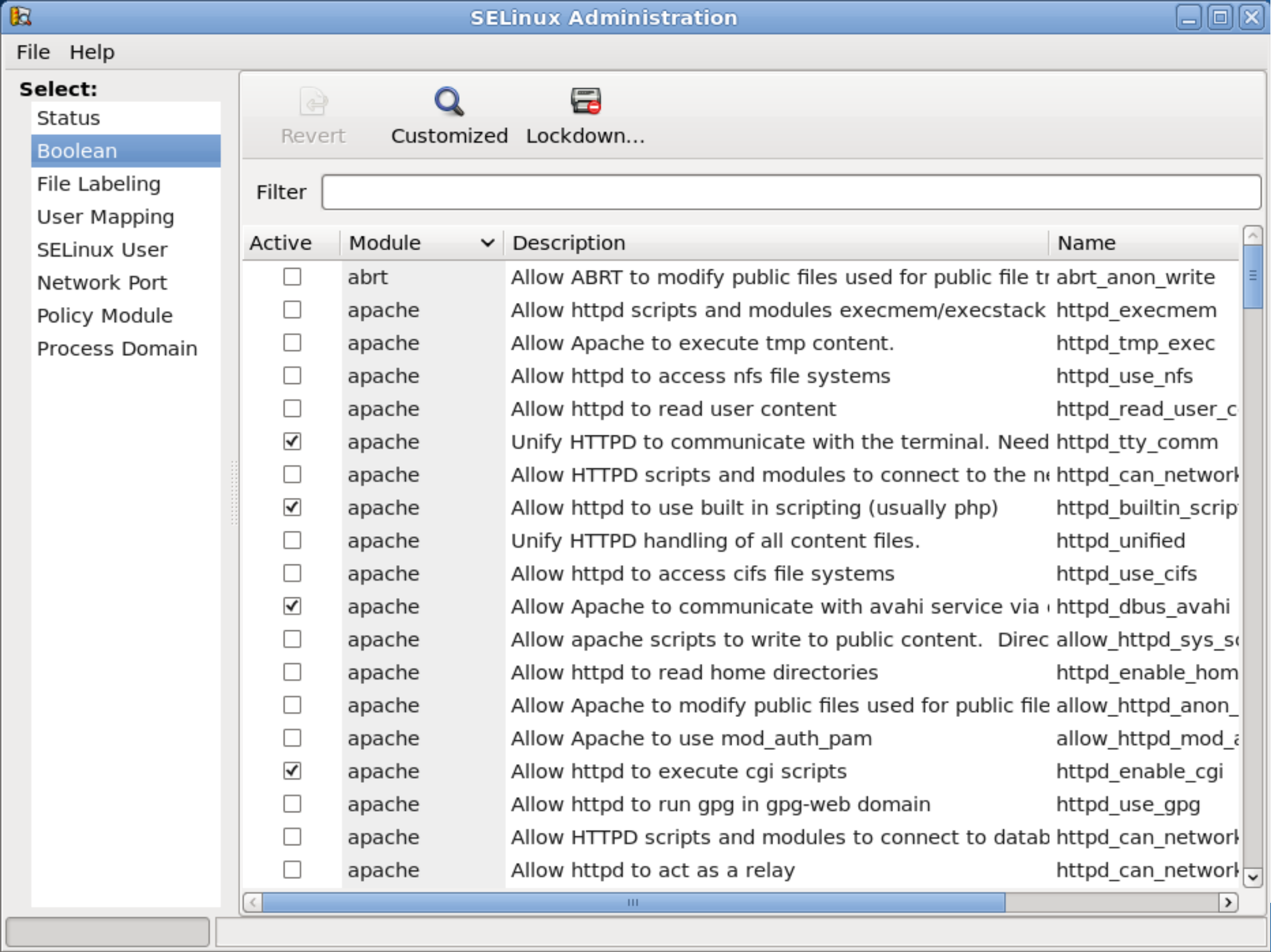
- `restorecon`
  - sets a file back to the default context
  - works on directory/file level
- `setfiles`
  - for system initialization, on file system level
  - expects `file_contexts` file to be specified
- `fixfiles`
  - script wrapper around `setfiles` and `restorecon`
  - RPM name as argument for relabeling of files in package

Managing File Labeling

- `matchpathcon`
  - shows the standard context of resources
- `semanage`
  - show/modify standard context of resources
  - uses regular expressions for path specifications
  - lots of other functions
- `system-config-selinux`
  - graphical frontend for various CLI utilities
  - approx. `semanage` functionality

SELinux Booleans

- Booleans are if/else statements in policy
- Configure policy without editing policy
- `getsebool`
  - `getsebool -a`
- `setsebool`
  - `setsebool -P -allow=[1|0]`
- `system-config-selinux`
- Turns on/off sections of policy
  - `setsebool -P virt_use_usb 1`

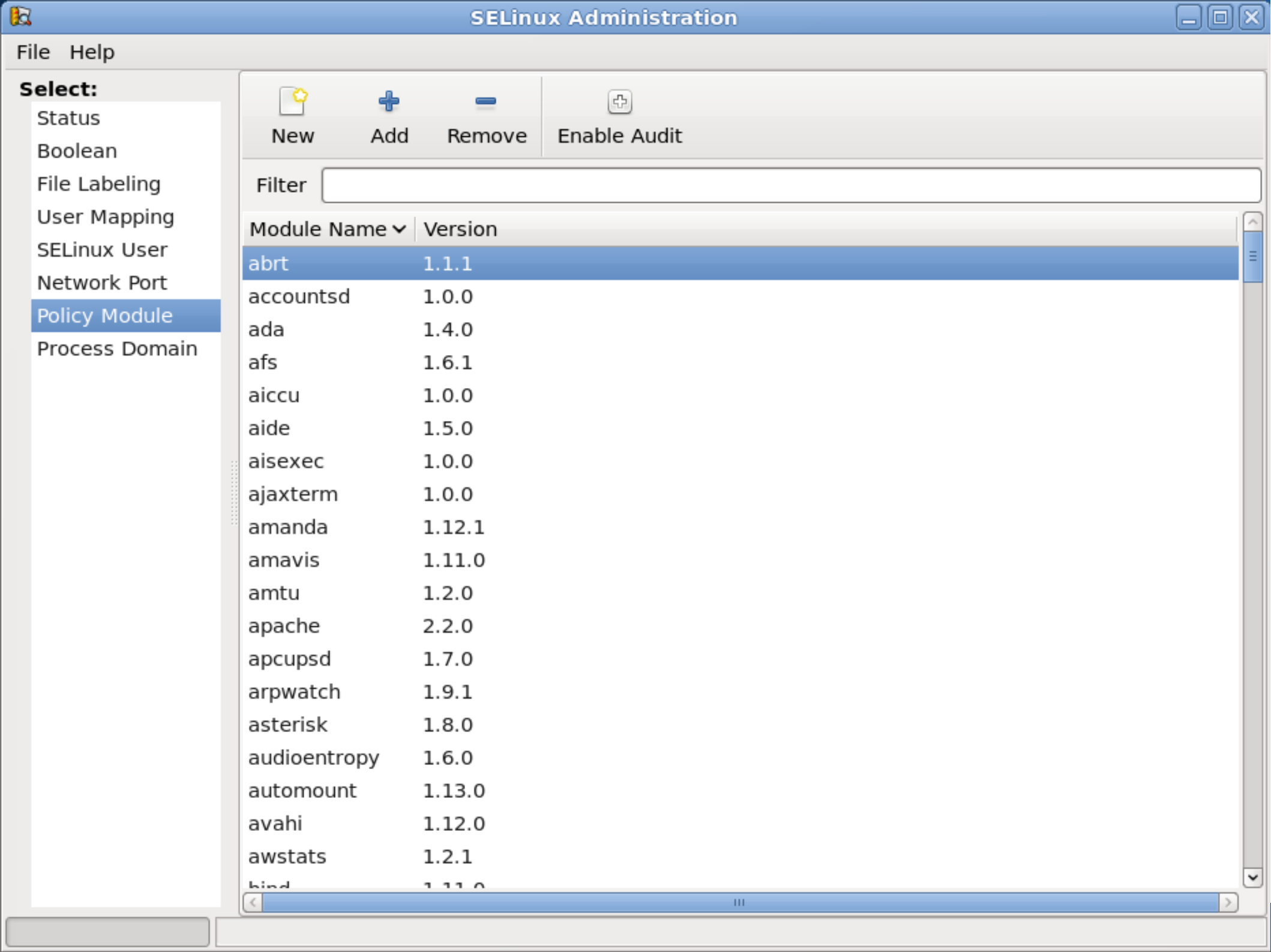

SELinux Administration
File
Help
Select:
Status
Boolean
File Labeling
User Mapping
SELinux User
Network Port
Policy Module
Process Domain
Revert
Customized
Lockdown...
Filter
Active
Module
Description
Name
abrt
Allow ABRT to modify public files used for public file tr
abrt_anon_write
apache
Allow httpd scripts and modules execmem/execstack
httpd_execmem
apache
Allow Apache to execute tmp content.
httpd_tmp_exec
apache
Allow httpd to access nfs file systems
httpd_use_nfs
apache
Allow httpd to read user content
httpd_read_user_c
apache
Unify HTTPD to communicate with the terminal. Need
httpd_tty_comm
apache
Allow HTTPD scripts and modules to connect to the ne
httpd_can_networ
apache
Allow httpd to use built in scripting (usually php)
httpd_builtin_scrip
apache
Unify HTTPD handling of all content files.
httpd_unified
apache
Allow httpd to access cifs file systems
httpd_use_cifs
apache
Allow Apache to communicate with avahi service via
httpd_dbus_avahi
apache
Allow apache scripts to write to public content. Direc
allow_httpd_sys_s
apache
Allow httpd to read home directories
httpd_enable_hom
apache
Allow Apache to modify public files used for public file
allow_httpd_anon_
apache
Allow Apache to use mod_auth_pam
allow_httpd_mod_a
apache
Allow httpd to execute cgi scripts
httpd_enable_cgi
apache
Allow httpd to run gpg in gpg-web domain
httpd_use_gpg
apache
Allow HTTPD scripts and modules to connect to datab
httpd_can_networ
apache
Allow httpd to act as a relay
httpd_can_networ

SELinux Modules

- Modular Policy
  - concept of modules since Fedora Core 5
- `semodule` command:
  - copies the “policy package” (`*.pp`) in the directory `/etc/selinux/targeted/modules/active/modules`
  - compiles all installed `*.pp` files into new policy file `/etc/selinux/targeted/policy/policy.24`
  - creates the new `file_context` file and also `file_context.homedirs`
  - loads new policy

SELinux Modules

- `semodule` command:
  - `semodule -l`
    - lists all SELinux modules currently loaded
  - `semodule -i /usr/share/selinux/targeted/gpg.pp`
  - `semodule -i mymodule.pp`
    - loads (installs) a “policy package”
  - `semodule -r mymodule`
    - unloads (removes) a “policy package”


SELinux Administration
File
Help
Select:
Status
Boolean
File Labeling
User Mapping
SELinux User
Network Port
Policy Module
Process Domain
New
Add
Remove
Enable Audit
Filter
Module Name
Version
abrt 1.1.1
accountsd 1.0.0
ada 1.4.0
afs 1.6.1
aiccu 1.0.0
aide 1.5.0
aisexec 1.0.0
ajaxterm 1.0.0
amanda 1.12.1
amavis 1.11.0
amtu 1.2.0
apache 2.2.0
apcupsd 1.7.0
arpwatch 1.9.1
asterisk 1.8.0
audioentropy 1.6.0
automount 1.13.0
avahi 1.12.0
awstats 1.2.1

Generating Policy Modules

- Policy modules consist out of three files
  - Type Enforcement file (`*.te`)
    - contains `allow` rules and interface calls associated with the confined domain
  - File Context file (`*.fc`)
    - contains all resource labels of the module
  - Interface File (`*.if`)
    - contains all interfaces used by other domains to interact with this confined domain
    - `DOMAIN_domtrans`, `DOMAIN_read_config`

Policies with audit2allow

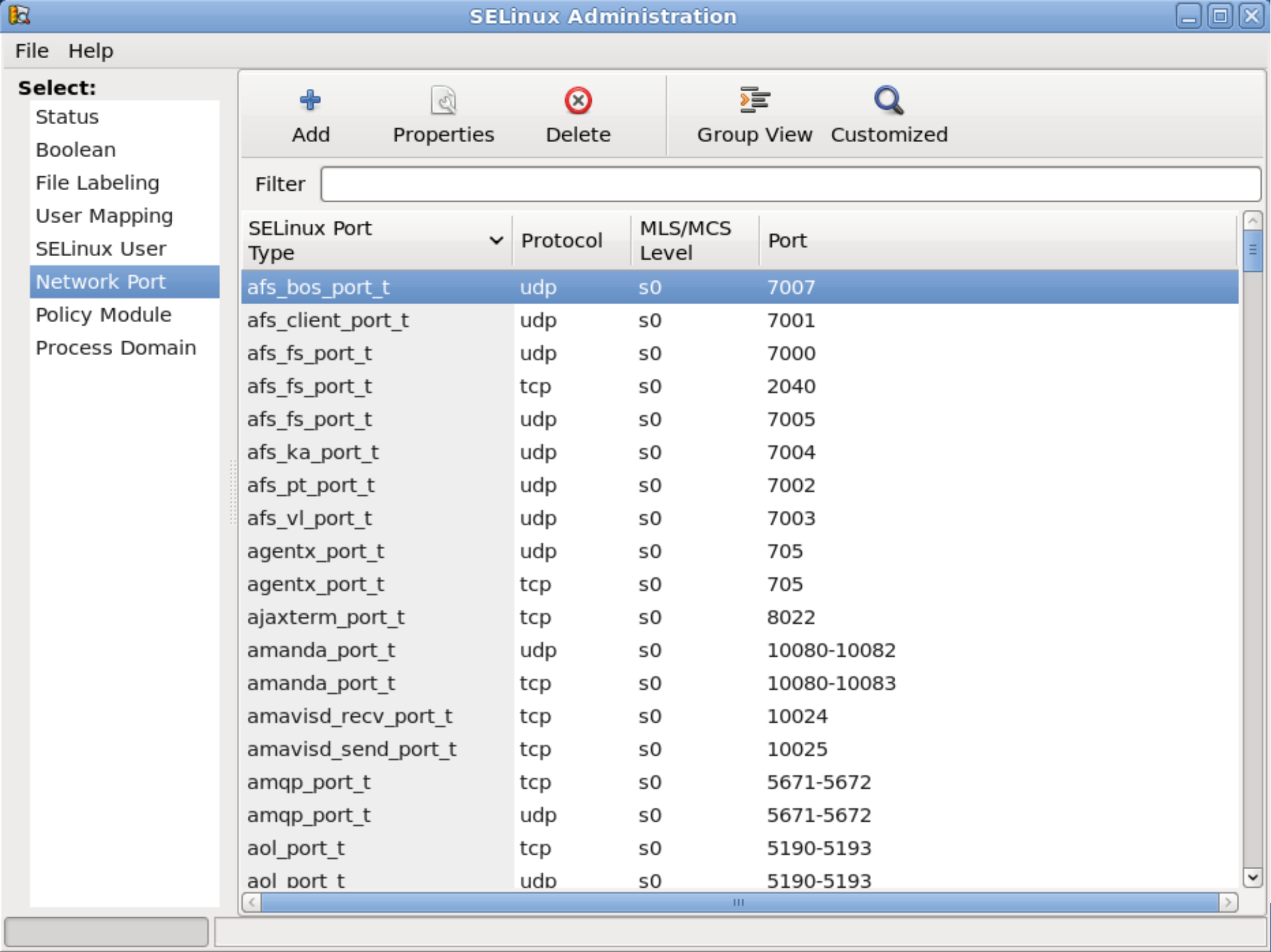
- Making small customizations to policy
- `grep httpd_t /var/log/audit/audit.log \ | audit2allow -M mypolicy`
  - generates a `*.te` file and compiles it into a `*.pp` binary file
- `semodule -i mypolicy.pp`

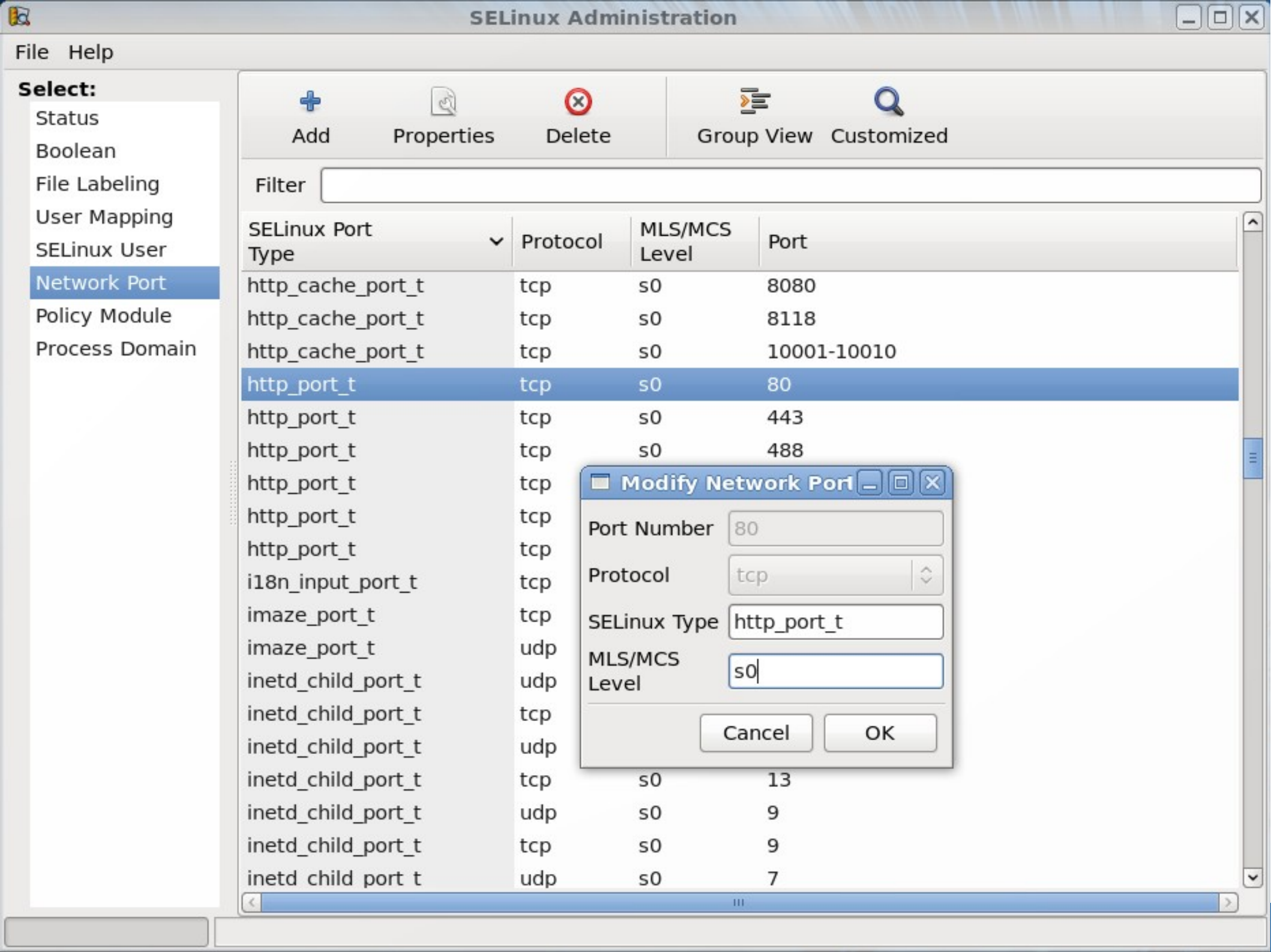
Managing SELinux Systems

- `semanage` framework since Fedora Core 5
- Avoids many own policies/modules
- Example:
  - without `semanage` framework:
    - allowing Apache to listen on port 81
    - required policy sources and tools
  - with `semanage` framework:
    - `semanage port -a -t http_port_t -P tcp 81`

semanage Commands

- SELinux users
  - `semanage user -l`
  - `semanage user -a guest_u`
- Linux to SELinux user mapping
  - `semanage login -a -s guest_u robert`
- File context
  - `semanage fcontext -a -t \ httpd_bugzilla_script_exec_t \ '/usr/share/bugzilla/cgi(/.*)?'`


SELinux Administration
File
Help
Select:
Status
Boolean
File Labeling
User Mapping
SELinux User
Network Port
Policy Module
Process Domain
Add
Properties
Delete
Group View
Customized
Filter
SELinux Port Type
Protocol
MLS/MCS Level
Port
afs_bos_port_t udp s0 7007
afs_client_port_t udp s0 7001
afs_fs_port_t udp s0 7000
afs_fs_port_t tcp s0 2040
afs_fs_port_t udp s0 7005
afs_ka_port_t udp s0 7004
afs_pt_port_t udp s0 7002
afs_vl_port_t udp s0 7003
agentx_port_t udp s0 705
agentx_port_t tcp s0 705
ajaxterm_port_t tcp s0 8022
amanda_port_t udp s0 10080-10082
amanda_port_t tcp s0 10080-10083
amavisd_recv_port_t tcp s0 10024
amavisd_send_port_t tcp s0 10025
amqp_port_t tcp s0 5671-5672
amqp_port_t udp s0 5671-5672
aol_port_t tcp s0 5190-5193
aol_port_t udp s0 5190-5193


SELinux Administration
File Help
Select:
Status
Boolean
File Labeling
User Mapping
SELinux User
Network Port
Policy Module
Process Domain
Add
Properties
Delete
Group View
Customized
Filter
SELinux Port Type
Protocol
MLS/MCS Level
Port
http_cache_port_t tcp s0 8080
http_cache_port_t tcp s0 8118
http_cache_port_t tcp s0 10001-10010
http_port_t tcp s0 80
http_port_t tcp s0 443
http_port_t tcp s0 488
http_port_t tcp
http_port_t tcp
http_port_t tcp
i18n_input_port_t tcp
imaze_port_t tcp
imaze_port_t udp
inetd_child_port_t udp
inetd_child_port_t tcp
inetd_child_port_t udp
inetd_child_port_t tcp s0 13
inetd_child_port_t udp s0 9
inetd_child_port_t tcp s0 9
inetd_child_port_t udp s0 7
Modify Network Port
Port Number 80
Protocol tcp
SELinux Type http_port_t
MLS/MCS Level s0
Cancel
OK

Auditing

- Audit system receives SELinux events
- No auditd process running
  - AVCs in `/var/log/messages` and `dmesg`
- With auditd process running
  - AVCs in `/var/log/audit/audit.log`
- Full auditing requires kernel parameter
  - `audit=1`

LSPP, CAPP & RBAC: EAL 4+

- Labeled Security Protection Profile (LSPP)
  - protection profile with MLS/MCS and MAC (→ B1)
- Controlled Access Protection Profile (CAPP)
  - protection profile with users/authentication (→ C1)
- Role-Based Access Control (RBACPP)
  - protection profile with role-based access control
- Evaluation Assurance Level (EAL 4+)
  - level of tests and documentation
  - methodically developed, tested and reviewed

aureport

- Generate summary reports of audit logs
  - `-a` – report about AVC messages
  - `-i` – interpret numeric fields for human consumption
  - `-ts` “start time” `-te` “end time”
    - `aureport -a -ts 1:00:00`
- `--success` / `--failed` – default is both
- `--summary` – totals of events

ausearch

- Search audit daemon logs
  - `-m avc` – event type, e.g. AVC messages
  - `-ts` – start time of search
  - `-x` – executable file
    - `ausearch -m avc -ts 1:00:00 -x named`

Conclusion

- SELinux
  - just use it
  - please do not turn it off
  - really protects against intrusion
  - NSA grade security for free

fedora

Further Resources

- Information
  - http://www.nsa.gov/research/selinux
  - http://docs.fedoraproject.org/en-US/ ↩ Fedora/13/html/Security-Enhanced_Linux/
  - http://fedoraproject.org/wiki/SELinux
- Mailing lists
  - selinux@tycho.nsa.gov
  - selinux@lists.fedoraproject.org

Questions?

fedora™


Thank you!