# SELinux Updates

**Thorsten Scherf**
**Senior Consultant**

**Red Hat Global Professional Services**

**01.12.2011 - Berlin / Germany**

redhat

# Agenda

SELinux review

What happened to strict policy

Policy customization and development

Booleans

Sandbox

Kiosk system

sVirt

SELinux and networking

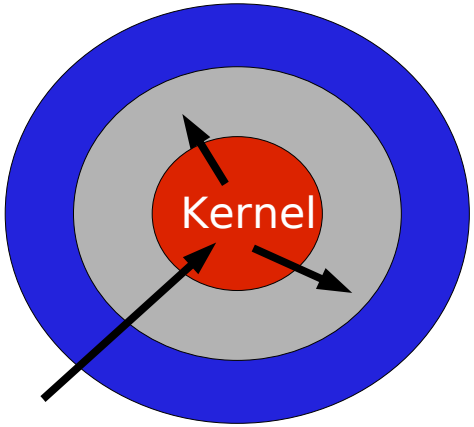
Memory protection

# What is wrong with UNIX security?

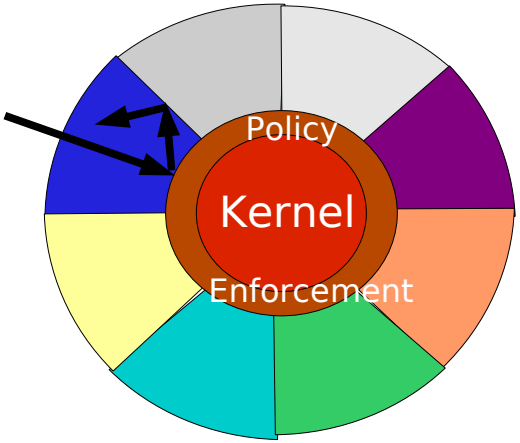
- Programs have full control over the access given to files they create (Discretionary Access Control - DAC)
- Therefore no protection against malicious software, “social engineering” and bugs in privileged software which may result in the software granting inappropriate access to files (eg, creating a mode 777 file in /tmp)
- No protection against 0-Day exploits
- Isolation in cloud environments

# What is SELinux?

- SELinux uses MAC (Mandatory Access Control)
- User/Programs has limited  privilege
- Security policy set by administrator and enforced by the System
- Several machines running root as guest account

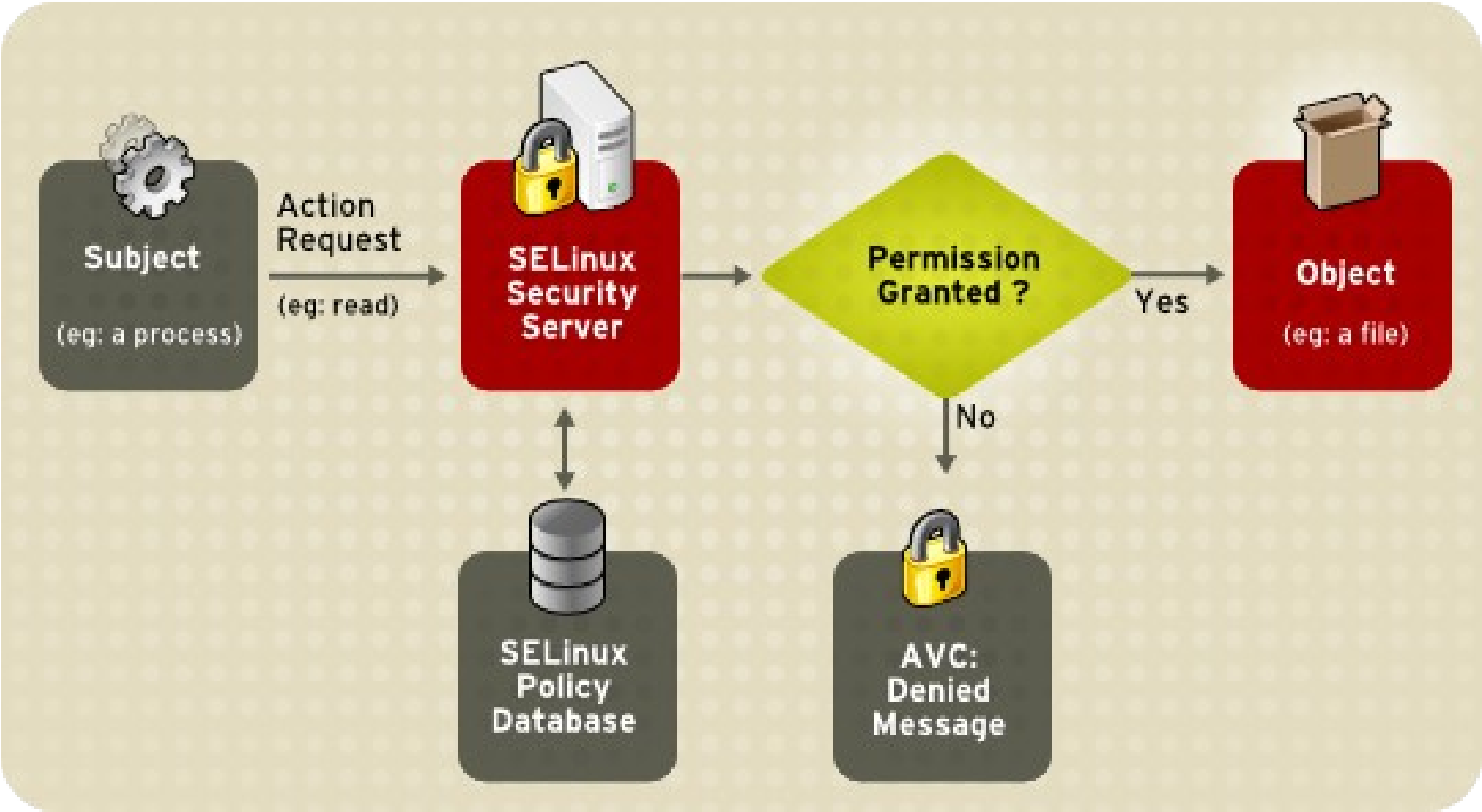



Classical UID based Access Control
Once a security exploit gains access to privileged system components the entire system is compromised




Domain-Type based Access Control
Kernel policy defines application rights, firewalling applications from compromising the entire system

# Architecture



# SELinux Key Components
# Security Context

- Basic labels used in SELinux
  - system_u:object_r:httpd_exec_t
  - system_u:system_r:httpd_t
- All subjects/objects have an associated security context
- Called a domain when used on a process
- Called a file_context when associated with a file
  - File Context are stored as extended attributes with the inode on the file system
  - On some file systems the kernel that do not support extended attributes the kernel provides the file context.

# What happened to strict policy

- Targeted policy
  - Main focus is to protect processes
  - Confined and unconfined processes
  - httpd_t / initrc_t / unconfined_t
- Strict policy
  - Used to confine users
  - guest_t / user_t / staff_t
- Targeted and strict policy have been merged with Fedora 9
- Now we have targeted, mls and minimum (f10) policy available
- Modules instead of monolithic policy

# Some more infos about current policy

- Size of targeted policy

  **# du -h /etc/selinux/targeted/policy/policy.24**
  6.2M /etc/selinux/targeted/policy/policy.24

- Size of minimum policy

  **# du -h /etc/selinux/minimum/policy/policy.24 (only base module)**
  856K /etc/selinux/minimum/policy/policy.24

- Number of confined processes

  **# seinfo -t | grep exec_t|wc -l**
  699

- Number of available booleans

  **# seinfo |grep -i booleans**
  Booleans: 179

# Example:
# Confined process

**# echo "Hello World..." > /var/www/html/foo**
**# ls -lZ /var/www/html/foo**
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/foo

**# wget -nv -O - http://localhost/foo**
Hello World...

**# chcon -t admin_home_t /var/www/html/foo**
# wget -nv -O - http://localhost/foo
2011-11-28 20:03:59 ERROR 403: Forbidden.

**# ausearch -m avc -ts today 20:00:00|grep httpd**
type=AVC msg=audit(1322507039.173:408): avc: denied { getattr } for pid=8818 comm="httpd" path="/var/www/html/foo" dev=sda3 ino=9344 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=system_u:object_r:admin_home_t:s0

# Example:
# Confined user

**# id -Z**
user_u:user_r:user_t:s0

**# ping -c1 www.redhat.de**
PING www.redhat.de (209.132.183.88) 56(84) bytes of data.

**# semanage login -m -s guest_u thscherf**

**# id -Z**
guest_u:guest_r:guest_t:s0

**# ping -c1 www.redhat.de**
ping: icmp open socket: Permission denied

# How does this work?

- Or - How do we transition from an unconfined to a confined process?

**# id -Z**
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

**# ls -lZ /etc/init.d/httpd**
-rwxr-xr-x. root root system_u:object_r:httpd_initrc_exec_t:s0 /etc/init.d/httpd

**# sesearch -T -s unconfined_t -t initrc_exec_t**
type_transition unconfined_t httpd_initrc_exec_t : process initrc_t;

**# sesearch -T -s initrc_t -t httpd_exec_t**
type_transition initrc_t httpd_exec_t : process httpd_t;

**# ps -efZ | grep httpd**
unconfined_u:system_r:httpd_t:s0 root root 00:00:00 /usr/sbin/httpd

# Unconfined and permissive domains

- Making a domain unconfined (there is no disable_trans anymore)

**# cat myweb.te:**

```
policy_module(myweb, 1.0)
gen_requires(`
        type httpd_t;
')
unconfined_domain(httpd_t)
```

**# make -f /usr/share/selinux/devel/Makefile**
**# semodule -i mypam.pp**

- Creating a permissive domain

**# semanage permissive -a httpd_t**

# Policy customization

- Modular instead of monolithic
- No need to access policy source anymore...
- ...semanage can do the job for you

```
# semanage port -l|grep http
http_port_t                    tcp      80, 443, 488, 8008, 8009, 8443

# semanage port -a -t http_port_t -p tcp 8888

# semanage port -l|grep http
http_port_t                    tcp      8888, 80, 443, 488, 8008, 8009, 8443

# semanage fcontext --add --type 'public_content_rw_t '/www(/.*)?'

# restorecon -Rv /www
```

# Policy customization II

- Check audit.log for AVC deny messages
  - Raw log messages
- Setroubleshoot daemon
  - Easy to read log messages
- Disable don't audit rules

  **# semodule -DB**

- Use restorecon over chcon whenever possible

  **# restorecon -v /var/www/html/foo**

- If you have a new file-type, use semanage to add it to the policy and use restorecon afterwards

# Policy customization III

- Create a local module to fix problems with the policy
  - Use audit2allow to create a policy file
  - Make yourself familiar with interfaces
  - Use interfaces

```
# cat /var/log/audit/audit.log | audit2allow -R -m local > local.te
# cat local.te
policy_module(local, 1.0)
        gen_require(‘
                type myapp_t;
                type etc_t;
        );
files_read_etc_files(myapp_t)
```

- Review local.te and customize as desired

# Graphical log analyzer

SELinux Alert Browser

SELinux has detected a problem.

Would you like to receive alerts? Yes No

The source process: /usr/sbin/httpd
Attempted this access: getattr
On this file: /var/www/html/foo

Mon Nov 28, 2011 20:00 CET

Troubleshoot | Notify Admin | Details | Ignore | Delete

| If you were trying to... | Then this is the solution. | |
|---|---|---|
| If you want to fix the label. /var/www/html/foo default label should be httpd_sys_content_t. | you can run restorecon.<br># /sbin/restorecon -v /var/www/html/foo | Plugin Details<br>Restore Context |
| If you believe that httpd should be allowed getattr access on the foo file by default. | You should report this as a bug.<br>You can generate a local policy module to allow this access.<br>Allow this access for now by executing:<br># grep httpd /var/log/audit/audit.log \| audit2allow -M mypol<br># semodule -i mypol.pp | Plugin Details<br>Report Bug |

Previous | Alert 13 of 50 | Next | List All Alerts

# Booleans

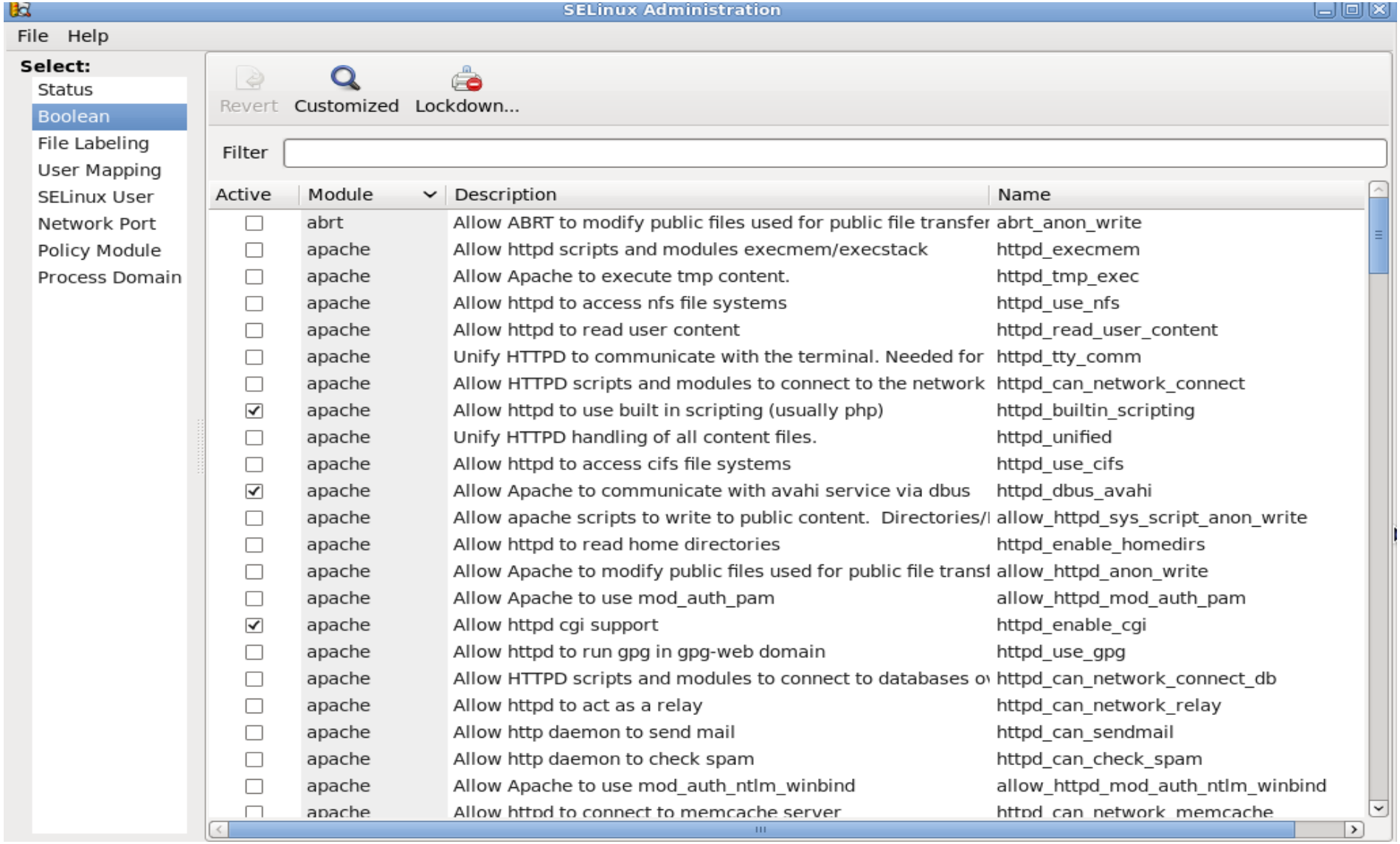
- Can also be used to change policy “on the fly”
- Don't trust your users?
- Simply put them into user_r role and deny content execution in /home and /tmp

**# getsebool -a |grep user_exec**
allow_user_exec_content --> on

**# setsebool -P allow_user_exec_content off**
**# ./virus**

**./virus: Permission denied**

# Graphical management tool



# Policy development

- Easy to build new modules
- Again, no need to access existing policy source anymore...
- ...just create a new module for your own application

**# /usr/bin/sepolgen -t 3 /usr/bin/foo foo**

Created the following files in ./ :

foo.te: Type Enforcement file

foo.if: Interface file

foo.fc: File Contexts file

foo.sh: Setup Script

# Policy development II

- There is also a graphical tool available
  - **selinux-polgengui**

# SELinux X Sandbox

- Run general applications in a locked down environment
  - Less privileged then other processes run by the user
  - Creates new temporary /home and /tmp
  - Block networking
- Easy to use on random applications
  - No need to create special policy
- Xephyr X-Server and Matchbox Window Manager
- Default type: sandbox_x_client_t -> no network
- More types available, like sandbox_web_t

**# sandbox -X -t sandbox_web_t epiphany**

# SELinux Kiosk System

- Locked down GNOME Desktop system
- Uses the xguest RPM package
- No network except Firefox
- Customization is easy:
  - Use proper interfaces to allow additional access
  - corenet_tcp_connect_smtp_port(xguest_t)
- Can be installed via Fedora Kiosk Spin or kickstart file:
  - http://people.fedoraproject.org/~dwalsh/SELinux/kiosk/kiosk.{iso,ks}

# sVirt – Securing your virtual machines

- KVM processes have a uniq security label (svirt_t:c1,c2)
- Isolate virtual guests using SELinux security policy
- MCS Categories are used to define access control on objects
- Integrated into libvirt tools (virt-install, virtmanager)

# SELinux and Networking

- Policy Based packet filtering
  - Netfilter framework “tags” IP packets with security context
  - SELinux policy is used for access control
  - Example: http_server_packet_t (port 443) is only readable by httpd_t but not from sshd_t
- IPSec based Labeling
  - Implements access control between local and remote processes
  - Needs IPSec
  - Security Policy Database (SPD) contains SELinux label for established  Security Associations (SA)

# mod-selinux

- Apache module, just like mod-security
- Enables multiple Apache instances to run with different security context, based on user logins
- Requires user authentication before access is granted
- Flat files or databases can be used for user<->context mapping

# Memory protection

- The following error sounds familiar to you?
  - **error while loading shared libraries: /usr/lib/libfoo.so.42: cannot restore segment prot after reloc: Permission denied**
- Selinux does memory checks also on unconfined processes
- Bad libraries try bad things like text relocations – SELinux prevents this
- Best to file a bug report against the software
- If you need to workaround the problem

```
# /usr/sbin/semanage fcontext -a -t textrel_shlib_t '/usr/lib/libfoo.so.42'
# restorecon -v /usr/lib/libfoo.so.42
```

redhat

**This is the end.**

**Thanks for listening.**

**Still questions?**

**tscherf@redhat.com**