

SELinux Introduction

...

Jason Zaman

FOSSASIA 2017 March 17th - 19th

blog.perfinion.com

Overview

1. Who am I?
2. What is SELinux?
3. DAC vs MAC
4. Type Enforcement
5. Labels
6. Sometimes SELinux denies badness
7. Fixing problems with booleans
8. Example policy
9. Fixing problems with new file contexts

\$ whoami

Jason “perfinion” Zaman

Gentoo Linux Developer

SELinux and Hardened Gentoo projects

jason@perfinion.com GPG keyid: 0x7EF137EC935B0EAF

Blog: <http://blog.perfinion.com/> Twitter: @perfinion Github: github.com/perfinion

What is SELinux?

Security

Offering security services to the operating system

Enhanced

Extends existing security subsystems, does not substitute them!

Linux

Works on all Linux systems as it is part of the mainline Linux kernel

What is SELinux?

SELinux *controls access* between applications and resources.

By using a *mandatory* security policy, SELinux enforces the security goals of the system regardless of whether applications misbehave or users act carelessly.

SELinux is capable of enforcing a wide range of security goals, from simply sandboxing applications to locking down network facing daemons and restricting users to only the resources they need to work.

SELinux History

- Created by the United States National Security Agency (NSA) as set of patches to the Linux kernel using Linux Security Modules (LSM)
- Released by the NSA under the GNU General Public License (GPL) in 2000
- Adopted by the upstream Linux kernel in 2003
- Used in many distros:
 - Hardened Gentoo
 - Redhat Enterprise Linux (4+) / CentOS
 - Fedora (2+)
 - Debian (optional)
 - Android (4.3+, full enforcing 5.0+)

DAC vs MAC - Discretionary Access Control

The owner can decide (has the *discretion*) how a resource is shared

A directory can be made world-writable by its owner, and from that point onward everything on the system can write to the directory

```
$ chmod -R 777 $HOME
```

```
$ chmod 777 /etc/shadow
```

Examples of DAC: POSIX user/group, permissions, ACLs

root user is omnipotent

DAC vs MAC - Mandatory Access Control

Access to a resource is *mandated* by a fixed policy loaded on the system by the administrator

Cannot be worked around by malicious users or programs

Even if you change DAC settings on resources, MAC can prevent access

DAC vs MAC - Linux Security Modules

Linux Security Modules (LSM) is a framework in the kernel which adds hooks for access control checks

Used by all the MAC frameworks in the kernel:

- SELinux
- AppArmor
- Smack
- TOMOYO
- Yama

/etc/selinux/config

```
# This file controls the state of SELinux on the system on boot.
```

```
# 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 four values:
```

```
#   targeted - Only targeted network daemons are protected.
```

```
#   strict   - Full SELinux protection.
```

```
#   mls      - Full SELinux protection with Multi-Level Security
```

```
#   mcs      - Full SELinux protection with Multi-Category Security
```

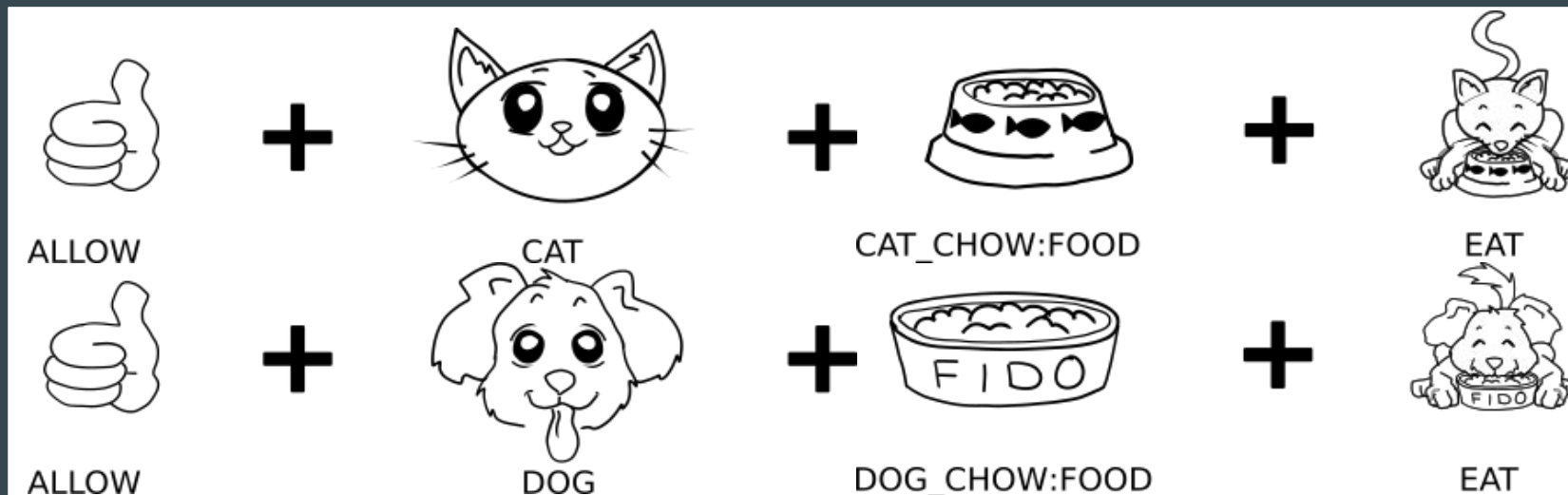
```
#             (mls, but only one sensitivity level)
```

```
SELINUXTYPE=strict
```

SELinux policy Type Enforcement rules

```
allow <source> <target>:<class> <permissions>;
```

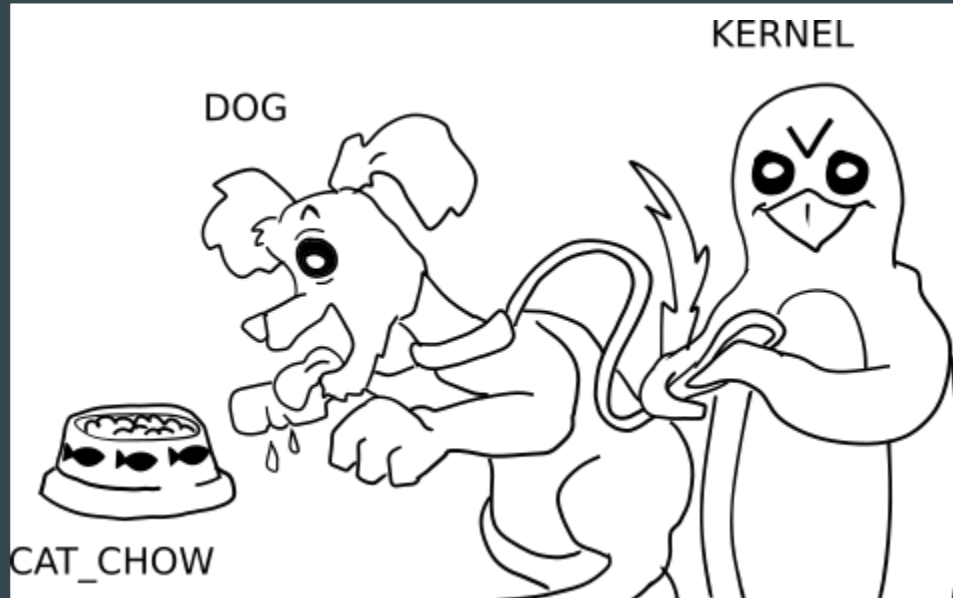
Allow source to do permissions to class belonging to target



SELinux Type Enforcement

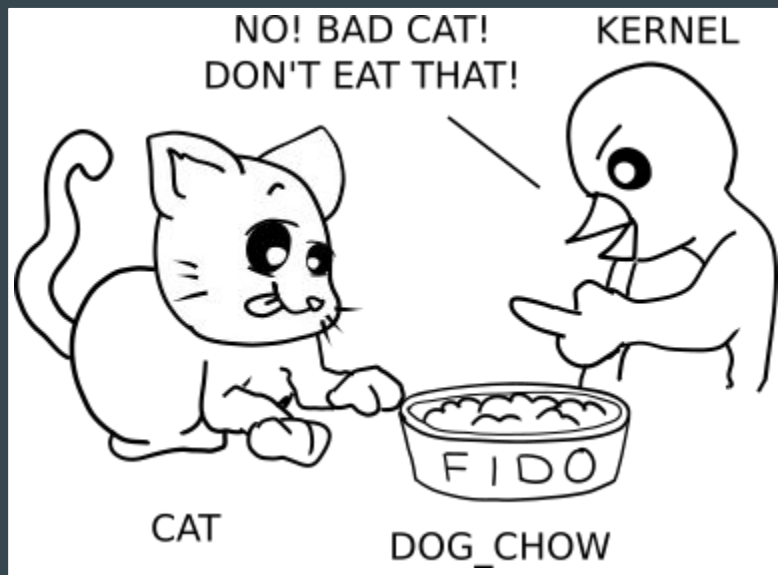
By default, everything is denied

Dogs can't eat cat food



SELinux Type Enforcement

Cats also can't eat dog food



SELinux labels

- SELinux is a labelling system
- Everything has a label
- Labels are also called contexts
- An SELinux policy uses the labels of things to decide to allow or deny
- Format of a label: `<user>:<role>:<type>[:<sensitivity>]`

SELinux label examples

Everything in SELinux has a label. Some examples:

<code>system_u:system_r:httpd_t</code>	system user, system role, httpd type
<code>user_u:user_r:user_t</code>	user user, user role, user type
<code>staff_u:sysadm_r:sysadm_t</code>	staff user, sysadmin role, sysadmin type
<code>system_u:object_r:bin_t</code>	stuff in /bin
<code>system_u:object_r:httpd_sys_content_t</code>	httpd system content

Labels on files

Most tools use `-Z` for SELinux, `ls -Z` will show labels

```
# ls -lZ /
drwxr-xr-x.  2 root root system_u:object_r:bin_t          210 Mar 14 18:22 bin/
drwxr-xr-x. 18 root root system_u:object_r:device_t      4640 Mar 10 15:12 dev/
drwxr-xr-x. 121 root root system_u:object_r:etc_t          235 Mar 14 18:30 etc/
drwxr-xr-x.  4 root root system_u:object_r:home_root_t      5 Sep  2  2014 home/
dr-xr-xr-x. 12 root root system_u:object_r:sysfs_t          0 Mar 10 15:10 sys/
drwxrwxrwt. 17 root root system_u:object_r:tmp_t          880 Mar 15 12:33 tmp/
drwxr-xr-x. 15 root root system_u:object_r:usr_t           17 Feb 19  2016 usr/
drwxr-xr-x.  9 root root system_u:object_r:var_t           12 Feb 10 10:25 var/
```


Labels on processes

ps Z will show labels on processes

```
# ps auxZ
```

LABEL	USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
system_u:system_r:init_t	root	1	0.0	0.0	13008	808	?	Ss	Mar10	0:06	init [3]
system_u:system_r:xserver_t	root	5128	1.7	0.6	393464	77720	tty7	Ssl+	Mar10	120:27	/usr/bin/X :0
staff_u:staff_r:staff_t	jason	6269	0.0	0.0	254484	7024	?	Sl	Mar10	2:36	xfce4-session
staff_u:staff_r:chromium_t	jason	9317	0.0	0.5	904808	72592	?	Sl	08:49	0:00	chrome

Id -Z will show your user's current domain

```
$ id -Z
```

```
staff_u:staff_r:staff_t
```

```
# id -Z
```

```
staff_u:sysadm_r:sysadm_t
```

SELinux denied something ... sometimes that's good

AVC denials go to audit logs or dmesg

Check the audit logs and let's see what we can find

```
# ausearch -m avc -ts recent
```

```
avc: denied { read } for pid=5445 comm="cat" name="shadow" dev="zfs" ino=1683602  
scontext=staff_u:staff_r:staff_t tcontext=system_u:object_r:shadow_t tclass=file
```




¿Why is something reading /etc/shadow?!

You may have been hacked...



Using SELinux booleans to fix things

Some parts of the policy can be enabled or disabled by the admin

1. Enabled userdir in apache config
2. 403 Forbidden 
3. `# chmod +x ~jason/ ~jason/public_html/`
4. Still get 403 Forbidden 
5. Check audit logs ...
6. `# semanage boolean --modify --on httpd_enable_homedirs`
7. It works! 

Example SELinux policy - file contexts

File contexts are regexes and match the most specific matches

```
/usr/bin/apache          --          system_u:object_r:httpd_exec_t

/etc/httpd(/.*)?        system_u:object_r:httpd_config_t
/etc/apache(2)?(/.*)?  system_u:object_r:httpd_config_t

/var/log/apache(2)?(/.*)? system_u:object_r:httpd_log_t

/var/www(/.*)?          system_u:object_r:httpd_sys_content_t

HOME_DIR/((www)|(web)|(public_html))(/.+)? system_u:object_r:httpd_user_content_t
```

Basic rules for a web server


```
# read the config files
allow httpd_t httpd_config_t:dir { getattr search read open };
allow httpd_t httpd_config_t:file { getattr read open };

# read all the htdocs, macro expands to all the perms
allow httpd_t httpd_sys_content_t:dir list_dir_perms;
allow httpd_t httpd_sys_content_t:file read_file_perms;
allow httpd_t httpd_sys_content_t:lnk_file read_lnk_file_perms;

# write to the logs, macro expands to all the rules
manage_files_pattern(httpd_t, httpd_log_t, httpd_log_t)


# bind to port 80
allow httpd_t self:tcp_socket { accept listen };
allow httpd_t httpd_port_t:tcp_socket name_bind;
```

Troubleshooting when the contexts are wrong

1. You decide to move Apache's website from `/var/www/` to `/website/` 
2. Everything breaks
3.

```
# ls -ldZ /var/www/ /website/  
system_u:object_r:httpd_sys_content_t /var/www/  
system_u:object_r:default_t /website/
```
4.

```
# semanage fcontext --add --type httpd_sys_content_t '/website(/.*)?'
```
5.

```
# restorecon -rv /website  
restorecon: Relabeled /website from system_u:object_r:default_t to  
system_u:object_r:httpd_sys_content_t
```
6. Yay! 

setenforce 1

Learning more

Slides will be on my blog: <http://blog.perfinion.com/>

SELinux coloring book: <http://blog.linuxgrrl.com/2014/04/16/the-selinux-coloring-book/>

SELinux project wiki: <https://selinuxproject.org/>

Gentoo SELinux wiki: <https://wiki.gentoo.org/wiki/SELinux>

Fedora SELinux wiki: <https://fedoraproject.org/wiki/SELinux>

SELinux Notebook:

http://freecomputerbooks.com/books/The_SELinux_Notebook-4th_Edition.pdf