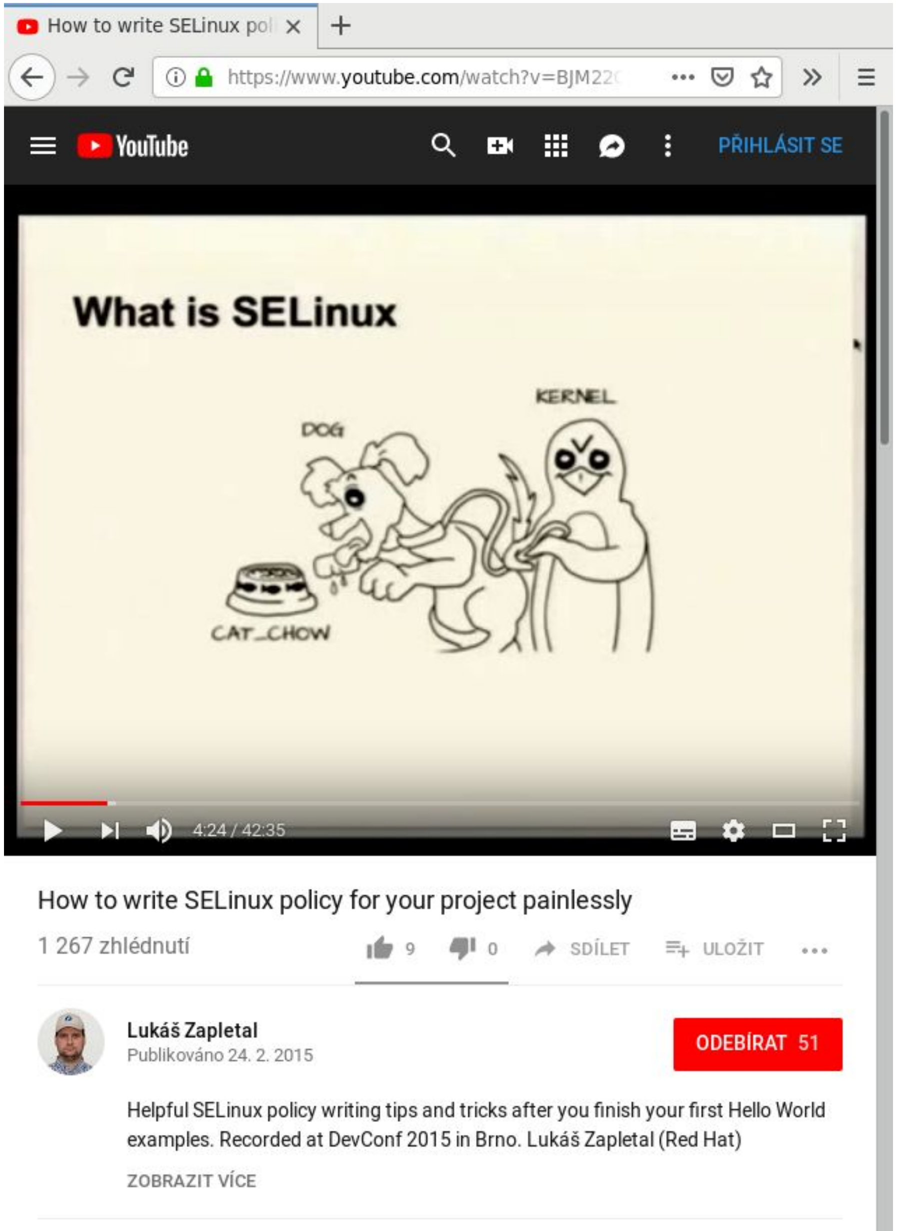
SELinux policy for your app

How to write SELinux policy for your project painlessly

Lukáš Zapletal

@lzap



How to write SELinux policy for your project painlessly

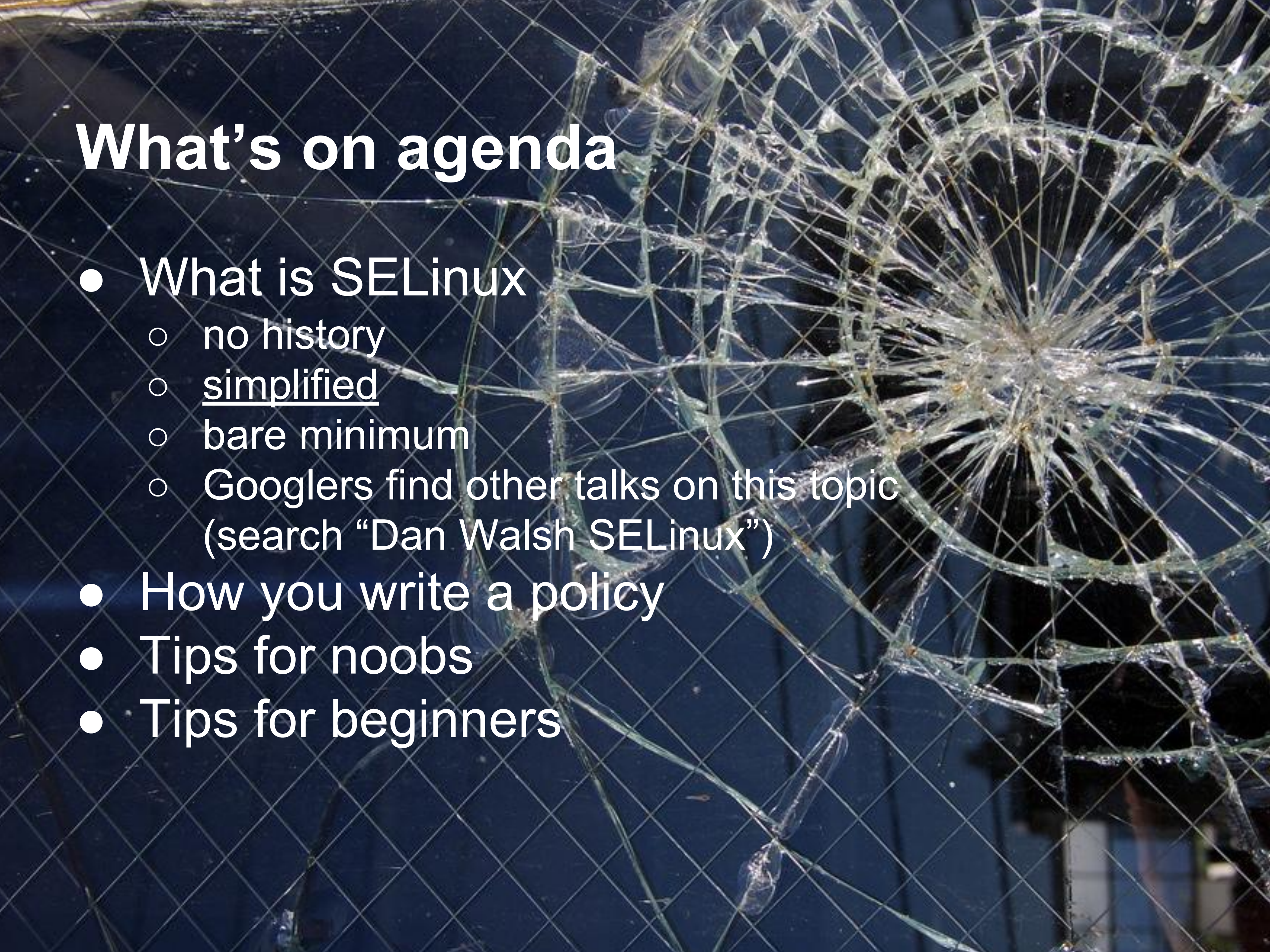
1 267 zhlédnutí

Lukáš Zapletal
Publikováno 24. 2. 2015

Helpful SELinux policy writing tips and tricks after you finish your first Hello World examples. Recorded at DevConf 2015 in Brno. Lukáš Zapletal (Red Hat)

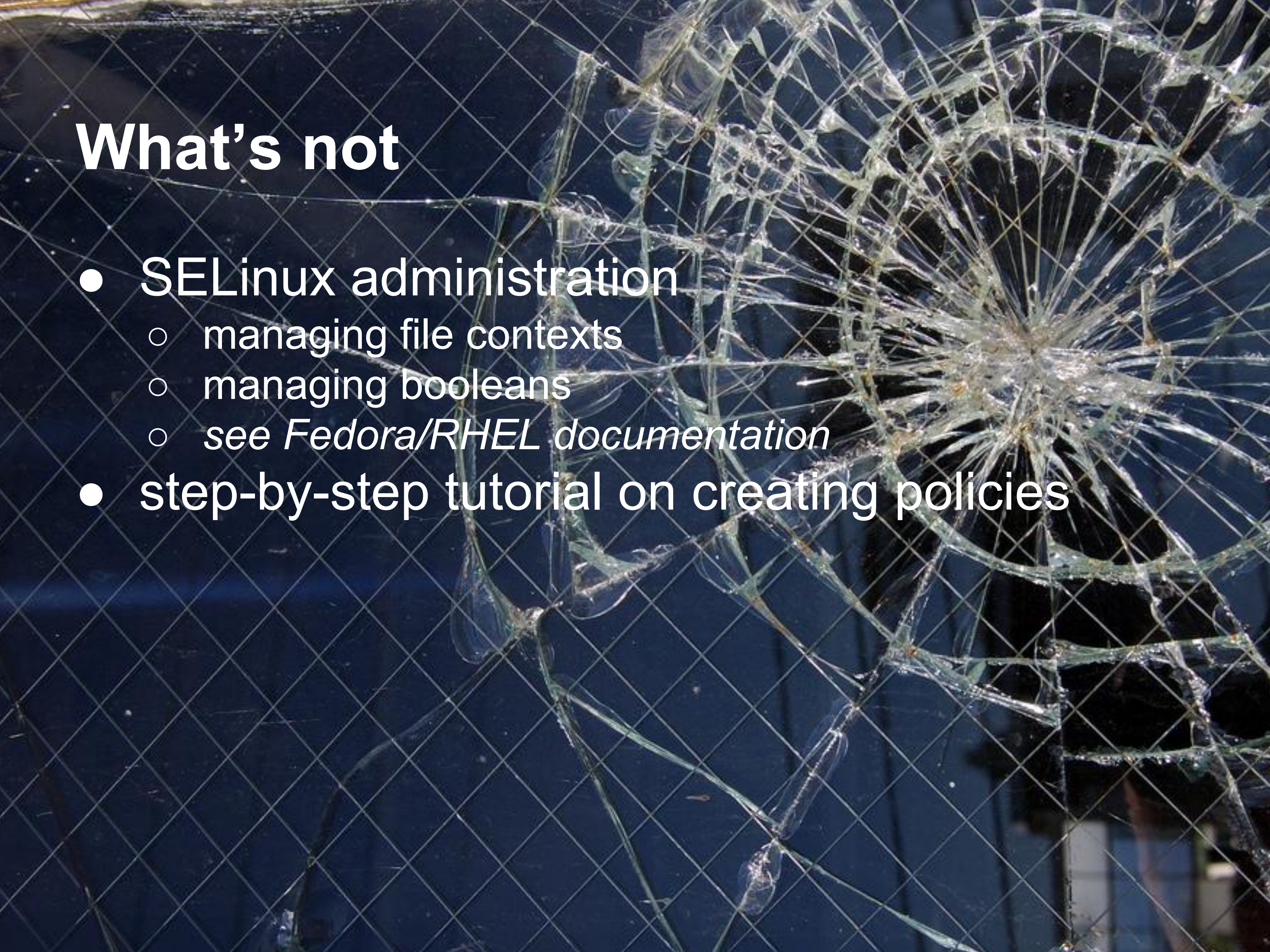
Purpose of this talk

Writing your policy is easy,
you can do it!



What’s on agenda

- What is SELinux
  - no history
  - simplified
  - bare minimum
  - Googlers find other talks on this topic (search “Dan Walsh SELinux”)
- How you write a policy
- Tips for noobs
- Tips for beginners



What's not

- SELinux administration
  - managing file contexts
  - managing booleans
  - *see Fedora/RHEL documentation*
- step-by-step tutorial on creating policies

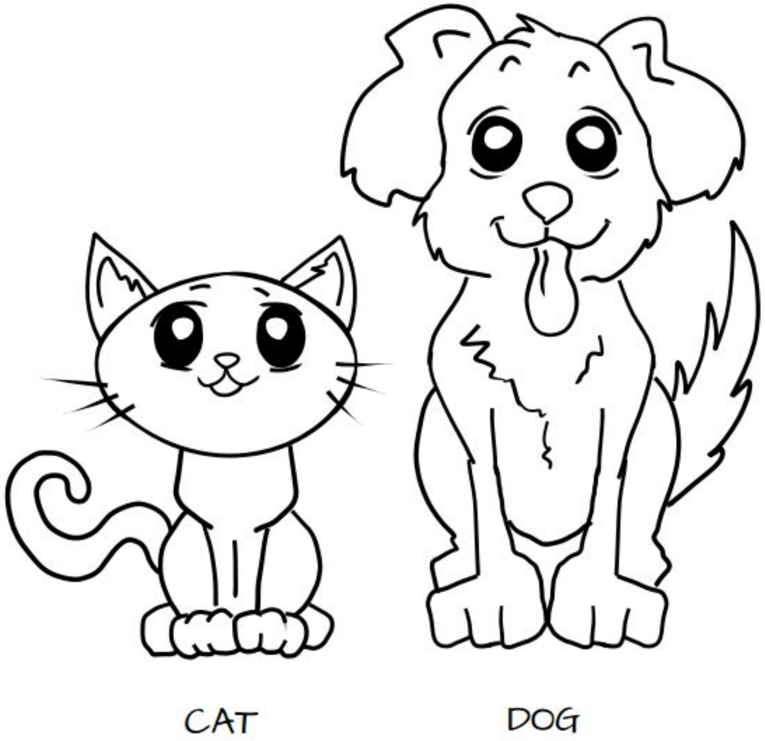
What is SELinux

Linux kernel **module** that **enforces** mandatory access-control **policies**.

What is SELinux

SELinux makes sure that **subject** (process) does **follow** granular **set of rules**.

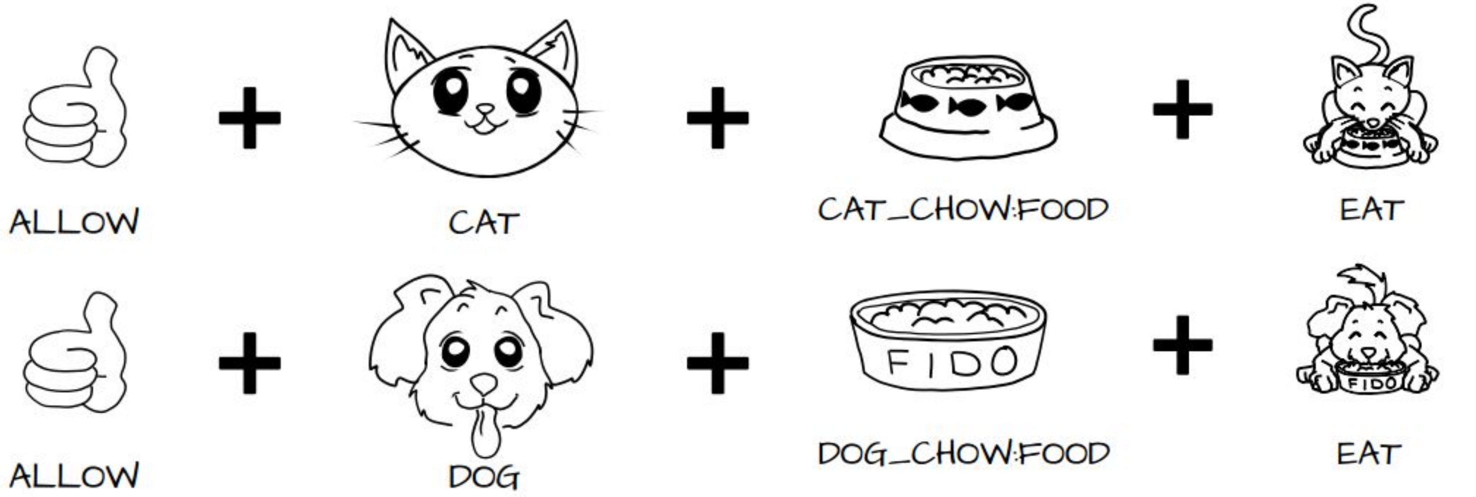
What is SELinux



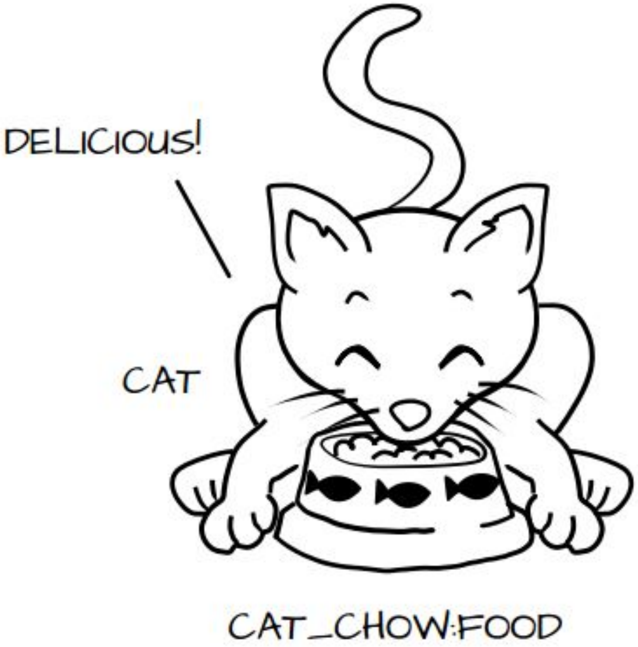
What is SELinux

FIDO
CAT_CHOW
DOG_CHOW

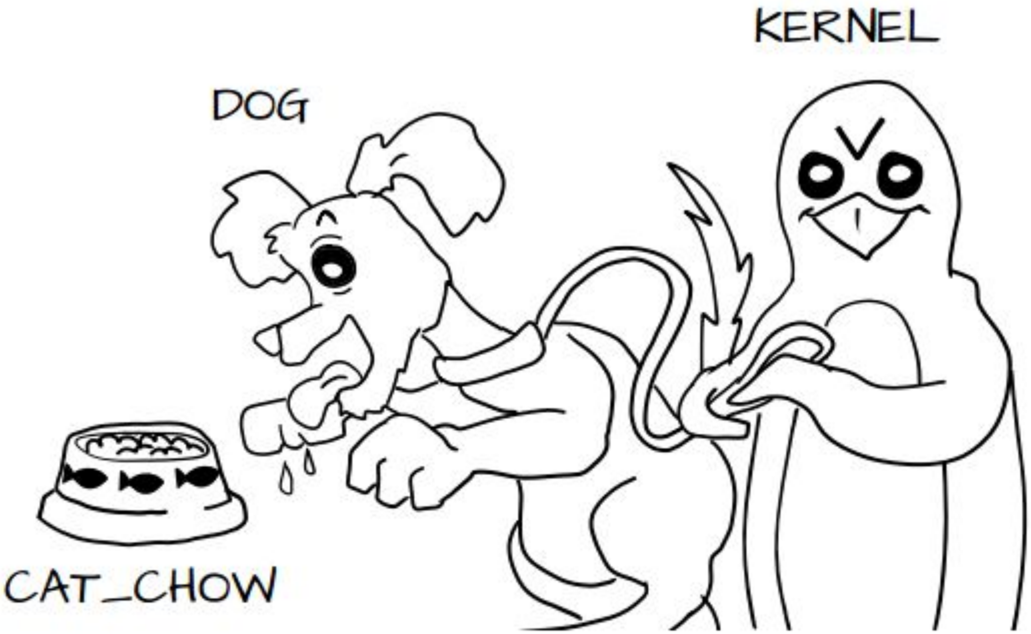
What is SELinux



What is SELinux



What is SELinux



What is SELinux

MCS (Multi Category Security)

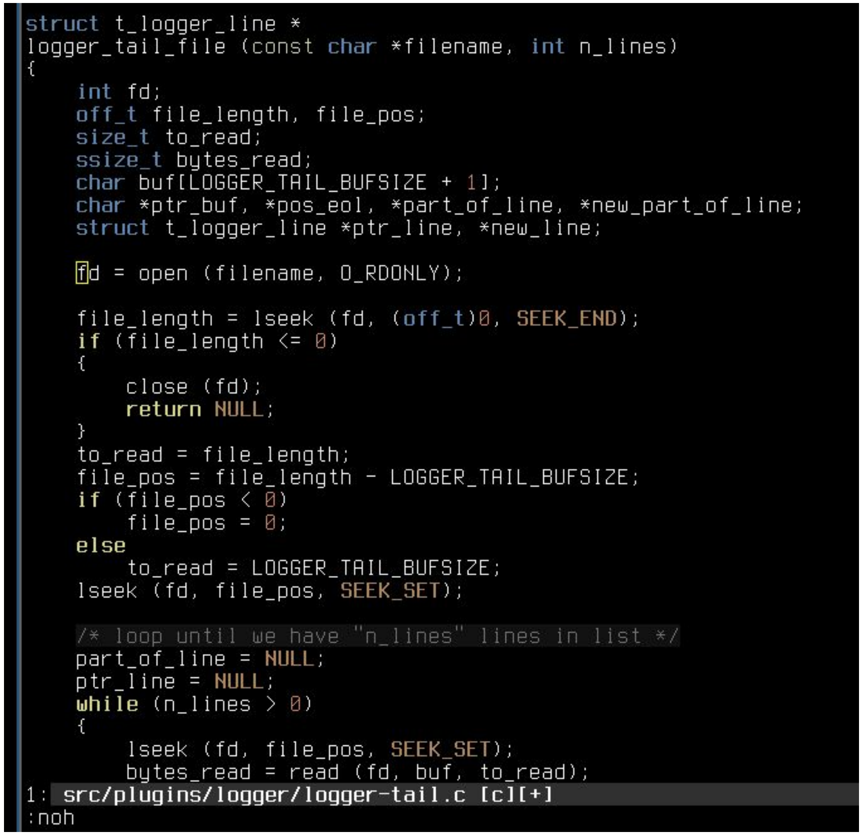
MLS (Multi Level Security)

What can SELinux do for you

- increases security
  - prevents from attacks
(sql injection vs shellshock)
  - restricts investigations
after successful
attacks (open remote port)
  - warns during attacks
(denials)

What can SELinux do for you

- find software bugs
  - unchecked file open return values
  - leaked descriptors
- workarounds proprietary behavior



```
struct t_logger_line *
logger_tail_file (const char *filename, int n_lines)
{
    int fd;
    off_t file_length, file_pos;
    size_t to_read;
    ssize_t bytes_read;
    char buf[LOGGER_TAIL_BUFSIZE + 1];
    char *ptr_buf, *pos_eol, *part_of_line, *new_part_of_line;
    struct t_logger_line *ptr_line, *new_line;

    fd = open (filename, O_RDONLY);

    file_length = lseek (fd, (off_t)0, SEEK_END);
    if (file_length <= 0)
    {
        close (fd);
        return NULL;
    }
    to_read = file_length;
    file_pos = file_length - LOGGER_TAIL_BUFSIZE;
    if (file_pos < 0)
        file_pos = 0;
    else
        to_read = LOGGER_TAIL_BUFSIZE;
    lseek (fd, file_pos, SEEK_SET);

    /* loop until we have "n_lines" lines in list */
    part_of_line = NULL;
    ptr_line = NULL;
    while (n_lines > 0)
    {
        lseek (fd, file_pos, SEEK_SET);
        bytes_read = read (fd, buf, to_read);
1: src/plugins/logger/logger-tail.c [c][+]
:noh
```

SELinux policy in Fedora

SELinux policy in Fedora

```
$ rpm -qa selinux-policy*
selinux-policy-3.12.1-196.fc20.noarch
selinux-policy-targeted-3.12.1-196.fc20.noarch
selinux-policy-devel-3.12.1-196.fc20.noarch
```

```
$ rpm -ql selinux-policy-targeted
…
/etc/selinux/targeted/contexts/files/file_contexts
…
/etc/selinux/targeted/modules/active/modules/abrt.pp
/etc/selinux/targeted/modules/active/modules/apache.pp
…
```

SELinux policy in Fedora

```
$ rpm -ql selinux-policy-devel
…
/usr/share/man/man8/sshd_selinux.8.gz
…
/usr/share/selinux/devel/html/telnetd.html
…
/usr/share/selinux/devel/Makefile
/usr/share/selinux/devel/include/Makefile
…
/usr/share/selinux/devel/include/contrib/postfix.if
/usr/share/selinux/devel/include/kernel/corecommands.if
/usr/share/selinux/devel/include/system/iptables.if
…
/usr/share/selinux/devel/include/support/ipc_patterns.spt
…
```

SELinux custom policy - hello world

- mypolicy.te (type enforcement)
- mypolicy.if (interfaces and docs)
- mypolicy.fc (file contexts)

```
# touch mypolicy.{te,if,fc}
# echo “policy_module(mypolicy, 0.1)” > *te
# make -f /usr/share/selinux/devel/Makefile
# semodule -i mypolicy.pp
# semodule -l | grep mypolicy
mypolicy 0.1
```

SELinux custom policy - makefile

Default makefile targets

- all (compile, generate docs, load)
- load/reload
- refresh (reload all policies)
- clean

Important variables:

- NAME (targeted, minimum, mls)
- TYPE (standard, mls, mcs)
- QUIET (set to “n” for verbose output)

This m4 preprocessor

Example SELinux policy

myapp.te:

```
policy_module(myapp, 1.0.0)

# Declarations

type myapp_t;
type myapp_exec_t;
domain_type(myapp_t)
domain_entry_file(myapp_t, myapp_exec_t)

type myapp_log_t;
logging_log_file(myapp_log_t)

type myapp_tmp_t;
files_tmp_file(myapp_tmp_t)

# Myapp local policy

allow myapp_t myapp_log_t:file {  read_file_perms append_file_perms };

allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t, myapp_tmp_t, file)
```

Example SELinux policy

myapp.if:

```
interface(`myapp_domtrans',`
  gen_require(`
    type myapp_t, myapp_exec_t;
  ')

  domtrans_pattern($1, myapp_exec_t, myapp_t)
')

interface(`myapp_read_log',`
  gen_require(`
    type myapp_log_t;
  ')

  logging_search_logs($1)
  allow $1 myapp_log_t:file  read_file_perms;
')
```

Example SELinux policy

myapp.fc:

```
/usr/sbin/myapp   --  gen_context(system_u:object_r:myapp_exec_t,s0)
/var/log/myapp    -d  gen_context(system_u:object_r:myapp_log_t,s0)
```

Important interface files

- application.if
- corenetwork.if
- files.if
- miscfiles.if
- devices.if
- terminal.if

- apache.if
- abrt.if

Important support files

- file_patterns.spt
- misc_macros.spt
- misc_patterns.spt
- loadable_module.spt

```
$ find /usr/share/selinux/devel/include -name \*.if | wc
-l
474
$ find /usr/share/selinux/devel/include -name \*.spt | wc
-l
8
```

This m4 preprocessor

```
/usr/bin/checkmodule:  loading policy configuration from tmp/foreman.tmp
foreman.te":238:ERROR 'syntax error' at token 'xxx_pattern'  on line 10522:
    xxx_pattern(passenger_t, httpd_tmp_t, httpd_tmp_t)
```

This m4 preprocessor

```
#line 238

#line 238
                } # end require
#line 238

#line 238

#line 238
        if (httpd_run_foreman) {
#line 238

#line 238
    manasge_dirs_pattern(passenger_t, httpd_tmp_t, httpd_tmp_t)
#line 238

#line 238
        allow passenger_t httpd_tmp_t:dir { open read getattr lock search ioctl add_name remove_name write };
#line 238
        allow passenger_t httpd_tmp_t:file { create open getattr setattr read write rename link unlink ioctl lock };
#line 238

#line 238

#line 238
        allow passenger_t httpd_tmp_t:dir { open read getattr lock search ioctl add_name remove_name write };
#line 238
        allow passenger_t httpd_tmp_t:sock_file { create open getattr setattr rename link unlink ioctl lock append };
#line 238
```

When to semicolon with m4

```
allow blah_t blahblah_t:file { read };
```

vs

```
myapp_read_blahblah_files(blah_t)
```

Interface naming

```
# from files.if
interface(`files_read_usr_files',`
   gen_require(`
      type usr_t;
   ')

   allow $1 usr_t:dir list_dir_perms;
   read_files_pattern($1, usr_t, usr_t)
   read_lnk_files_pattern($1, usr_t, usr_t)
')
```

Interface naming

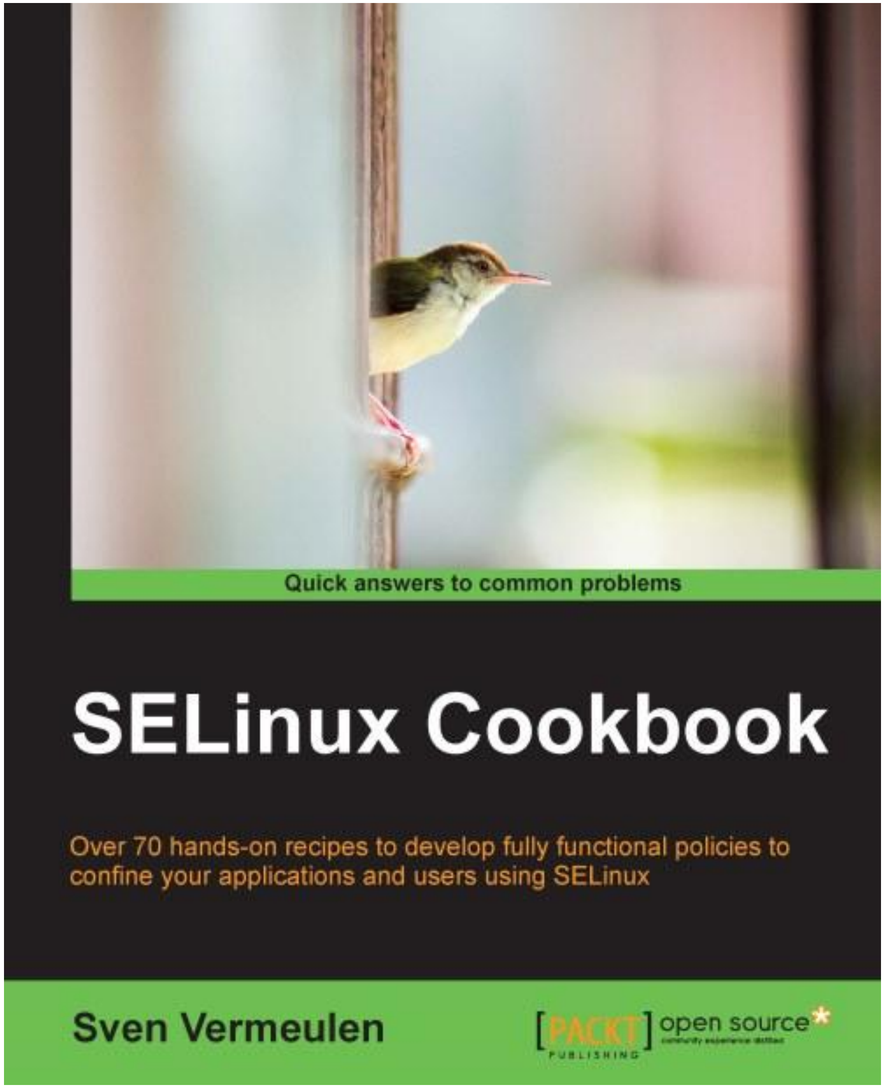
```
# from obj_perm_sets.spt
define(`list_dir_perms',
`{ getattr search open read lock ioctl }')
```

Interface naming

```
# from file_patterns.spt
define(`read_files_pattern',`
   allow $1 $2:dir search_dir_perms;
   allow $1 $3:file read_file_perms;
')

define(`read_lnk_files_pattern',`
   allow $1 $2:dir search_dir_perms;
   allow $1 $3:lnk_file read_lnk_file_perms;
')
```

Searching for interface definitions

- code examples
- free download
- functions.sh
    - seshowif
    - sefindif
    - seshowdef
    - sefinddef



Quick answers to common problems

SELinux Cookbook

Over 70 hands-on recipes to develop fully functional policies to confine your applications and users using SELinux

Sven Vermeulen

Searching for interface definitions

```
$ seshowif logging_log_file
interface(`logging_log_file',`
        gen_require(`
                attribute logfile;
        ')

        files_type($1)
        files_associate_tmp($1)
        fs_associate_tmpfs($1)
        typeattribute $1 logfile;
')
```

```
$ seshowdef search_dir_perms
define(`search_dir_perms',`{ getattr search open }')
```

Searching for interface definitions

```
$ sefindif logging_log_file
…
contrib/pki.if: template(`pki_apache_template',`
contrib/pki.if:         logging_log_file($1_log_t)
contrib/pki.if:         logging_log_filetrans($1_t, $1_log_t, { file dir } )
contrib/razor.if: template(`razor_common_domain_template',`
contrib/razor.if:       logging_log_filetrans($1_t, razor_log_t, file)
contrib/sendmail.if: interface(`sendmail_create_log',`
contrib/sendmail.if:    logging_log_filetrans($1, sendmail_log_t, file)
contrib/tomcat.if: template(`tomcat_domain_template',`
contrib/tomcat.if:      logging_log_file($1_log_t)
contrib/tomcat.if:      logging_log_filetrans($1_t, $1_log_t, { dir file })
kernel/files.if: interface(`files_stub_tmp',`
kernel/files.if: ##             <li>logging_log_file()</li>
system/authlogin.if: interface(`auth_log_filetrans_login_records',`
system/authlogin.if:    logging_log_filetrans($1, wtmp_t, file)
system/logging.if:
system/logging.if: ##           <li>logging_log_filetrans()</li>
system/logging.if: ##   logging_log_file(mylogfile_t)
system/logging.if: ##   logging_log_filetrans(mydomain_t, mylogfile_t, file)
system/logging.if: interface(`logging_log_file',`
…
```

How to navigate through with ctags

```
#!/bin/bash

if [ $? == 0 ]; then
    if [ -d /usr/share/selinux/devel ]; then
        ctags -e --langdef=te --langmap=te:..te.if.spt \
            --regex-te='/^type[ \t]+(\w+)(,|;)/\1/t,type/' \
            --regex-te='/^typealias[ \t]+\w+[ \t+]+alias[ \t]+(\w+);/\1/t,type/' \
            --regex-te='/^attribute[ \t]+(\w+);/\1/a,attribute/' \
            --regex-te='/^[ \t]*define\(`(\w+)/\1/d,define/' \
            --regex-te='/^[ \t]*interface\(`(\w+)/\1/i,interface/' \
            --regex-te='/^[ \t]*bool[ \t]+(\w+)/\1/b,bool/'\
            /usr/share/selinux/devel/include/*/*.if \
            /usr/share/selinux/devel/include/support/*.spt *.te
    else
        echo "You need to install selinux-policy-devel package"
        exit 1
    fi
else
    echo "You need to install ctags package"
    exit 1
fi
```

You lucky Vim user!

https://github.com/lzap/vim-selinux

Anatomy of SELinux denial

```
# grep AVC /var/log/audit/audit.log
type=AVC msg=audit(1413987601.193:1489): avc: denied {  name_bind } for
pid=12828 comm="ruby" src= 1251 scontext=system_u:system_r: passenger_t:s0
tcontext=system_u:object_r: unreserved_port_t:s0 tclass=udp_socket
```

```
# ausearch -m AVC
--
type=AVC msg=audit(1413987601.193:1489): avc: denied { name_bind } for
pid=12828 comm="ruby" src=1251 scontext=system_u:system_r:passenger_t:s0
tcontext=system_u:object_r:unreserved_port_t:s0 tclass=udp_socket
type=SYSCALL msg=audit(1413987601.193:1489): arch=x86_64 syscall=bind
success=no exit=EACCES a0=b a1= 7f5438524080 a2=10 a3=0 items=0 ppid=1
pid=12828 auid=4294967295 uid=997 gid=995 euid=997 suid=997 fsuid=997
egid=995 sgid=995 fsgid=995 tty=(none) ses=4294967295 comm=ruby
exe=/opt/rh/ruby193/root/usr/bin/ruby
subj=system_u:system_r:passenger_t:s0 key=(null)
--
```

The audit2allow thing

```
# audit2allow -al
allow passenger_t unreserved_port_t:udp_socket name_bind;
```

```
# audit2allow -Ral
corenet_udp_bind_generic_port(passenger_t)
```

```
# audit2allow -R
<paste> Ctrl+D
```

```
# audit2allow -RalM quickfix
******************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i quickfix.pp
```

The audit2allow abuse

permissive + audit2allow =

The audit2allow abuse

- file contexts
- domain transitions
- software bugs are hidden
- not following the least privilege principle

SELinux policy artifacts

- the policy itself
- the process
  - design issues
  - misconfigurations
  - bugs

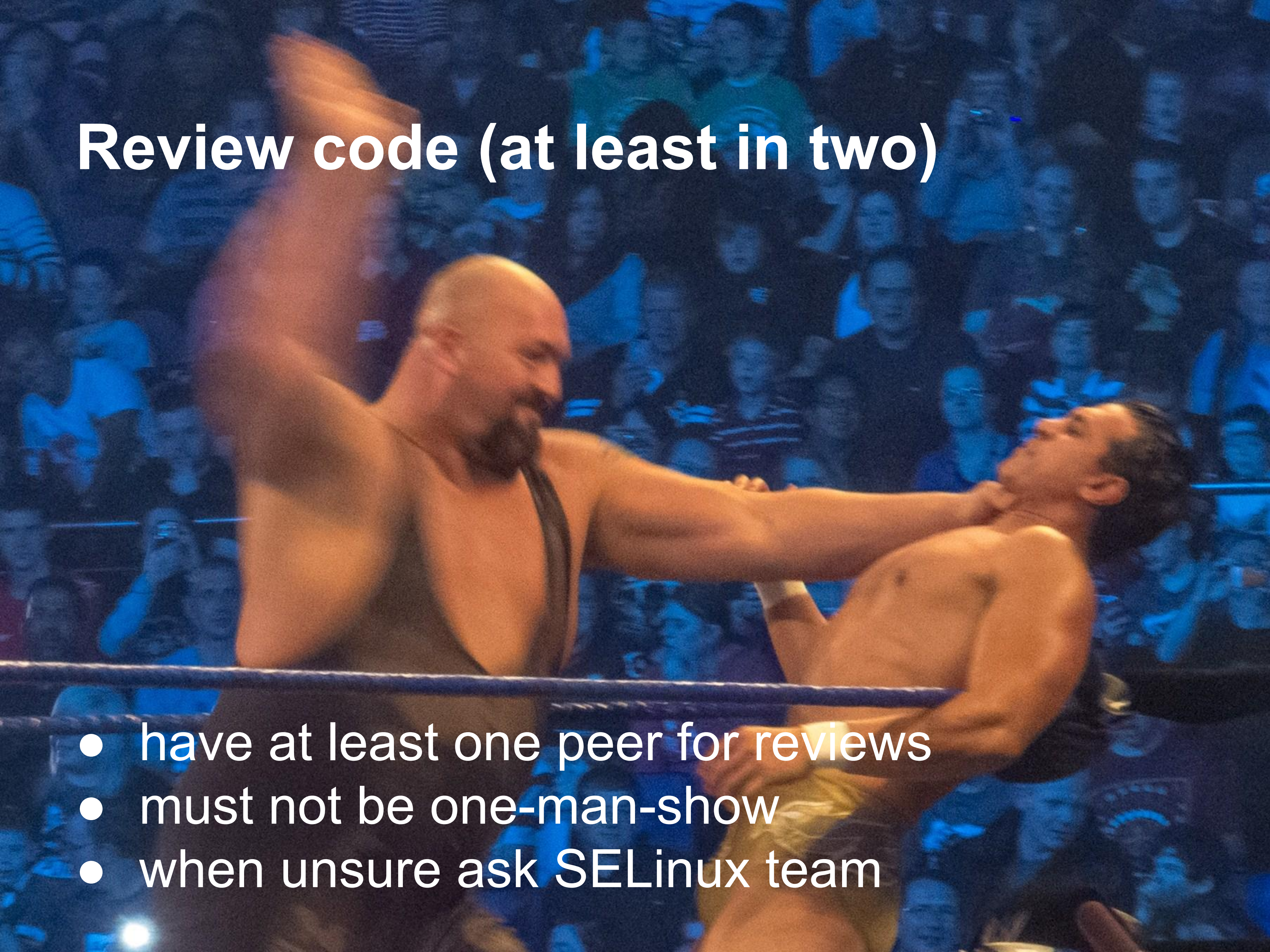
Take small steps

- Modify
- Compile
- Load
- Commit
- Repeat

One commit one issue (w/ denial)

```
commit 2a8011b2d211a043868c1bf3cff3d0dd084575eb
Refs: [docker-port-8989]
Author:     Lukas Zapletal <lzap+git@redhat.com>
AuthorDate: Fri Jan 16 10:34:44 2015 +0100
Commit:     Lukas Zapletal <lzap+git@redhat.com>
CommitDate: Fri Jan 16 10:34:44 2015 +0100

    Fixes #8989 - Add docker_port_t port and boolean

    Boolean passenger_can_connect_docker allows connections to newly created
    docker_port_t which is not yet defined in RHEL7/Fedora. This can be used when
    users starts Docker on TCP (defaults to UNIX sockets). Ports were reserved
    by IANA 2015-01-09: http (2375), https (2376).

    Denial:

    type=AVC msg=audit(1421352630.245:15331): avc:  denied  { name_connect } for
    pid=4803 comm="ruby" dest=2375 scontext=unconfined_u:system_r:passenger_t:s0
    tcontext=system_u:object_r:port_t:s0 tclass=tcp_socket
```

Review code (at least in two)



- have at least one peer for reviews
- must not be one-man-show
- when unsure ask SELinux team

You will not be famous

One more thing

How to file a SELinux bug

PROCESSES

```
ps axuZ
```

FILES

```
restorecon -rvn /
```

DENIALS

```
ausearch -m AVC
```

Q&A

Image credits - thanks

http://en.wikipedia.org/wiki/Joke_chess_problem#cite_note-1 (V. Ropke, Skakbladet 1942)

https://www.flickr.com/photos/x1brett/4600461689/

https://www.flickr.com/photos/nesster/3168425434/

https://openclipart.org/detail/4735/police-car-alarm-by-toplus

https://www.flickr.com/photos/caitlinator/3708011885/

http://aerokay.deviantart.com/art/The-Who-Poster-236014991

http://commons.wikimedia.org/wiki/File:PEO_M4_Carbine_RAS.jpg

http://commons.wikimedia.org/wiki/File:Horror_Images_Revolt_of_the_Zombies.jpg

http://commons.wikimedia.org/wiki/File:Blue_alarm_clock_(1).jpg

http://en.wikipedia.org/wiki/Big_Show

http://commons.wikimedia.org/wiki/File:INF_inspection.JPEG

http://pixabay.com/id/editor-teks-vim-perangkat-lunak-27620/

http://en.wikipedia.org/wiki/Smoking_in_Albania

http://commons.wikimedia.org/wiki/File:Bank-Security-Guard-Sleeping.jpeg

http://commons.wikimedia.org/wiki/File:Question_mark_(3534516458).jpg

https://openclipart.org/detail/182513/hazard-x-gold-by-Magirly-182513

http://pixabay.com/es/electricidad-flash-rayo-peligro-98819/

http://en.wikipedia.org/wiki/User:JustinTime55/sandbox/Apollo_11

http://en.wikipedia.org/wiki/Automotive_design