



ОНЛАЙН-ОБРАЗОВАНИЕ

# Когда не хватает прав и песочниц

## Selinux

Александр Румянцев



# Спецслужбы бывают полезны

**Selinux (Security Enhanced Linux) - система принудительного контроля доступа (Mandatory Access Control)**

**Изначально подсистема selinux разрабатывалось, как ни странно, АНБ.  
В ядре появилась в 1998 году.**

**Задачи, решаемые selinux:**

- **доступ к ресурсам (файл, сокет, системный вызов)**
- **логирование использования ресурсов**

**В это же время в SUSE появилась аналогичная система - AppArmor, включена в ядро уже в 2.6.36 в 2009 году**



# За что не любят selinux

Нелюбовь к selinux имеет те же принципы, что и нелюбовь к systemd, т.е. никакой рациональной идеи нет.

- Развивался очень долго, политики очень большие (и, соответственно, сложные)
- Каждый ресурс должен быть описан и сопоставлен с сервисом (что должны делать разработчики, но на деле легло на плечи дистрибьютеров)
  - до недавнего времени политики содержали большое количество ошибок
- При установке по-умолчанию отсутствуют инструменты для простого решения проблем

В данный момент накладные расходы на работу selinux минимальны, так что отключать selinux "для скорости" тоже бессмысленно

Вывод: отключение selinux равноценно установке прав 777 на каталог, вместо того, чтобы разбираться с правами и группами.



# Делаем selinux удобным

Нам понадобятся два пакета:

- **setools-console**
  - **sesearch**
  - **seinfo**
  - **findcon**
- **polycoreutils-python**
  - **audit2allow**
  - **audit2why**



# Основные понятия SELinux

**Домен** — список действий, которые может выполнять процесс. Обычно в качестве домена определяется минимально-возможный набор действий, при помощи которых процесс способен функционировать. Таким образом, если процесс дискредитирован, злоумышленнику не удастся нанести большого вреда.

**Роль** — список доменов, которые могут быть применены. Если какого-то домена нет в списке доменов какой-то роли, то действия из этого домена не могут быть применены.

**Тип** — набор действий, которые допустимы по отношению к объекту. Тип отличается от домена тем, что он может применяться к пайпам, каталогам и файлам, в то время как домен применяется к процессам.

**Контекст безопасности** — все атрибуты SELinux — роли, типы и домены.



# Варианты управления доступом в SELinux

**Type Enforcement (TE):** Группировка и сопоставление по типам (группам правил) - основной режим работы

**Role-Based Access Control (RBAC):** разграничение доступа по ролям

**Multi-Level Security (MLS) и Multi-Category Security(MCS):** многоуровневые комплексные политики

Контекст описывается четырьмя строками через двоеточие:  
`user:role:type:mls`

Из них нам важен, по-сути, только `type` (строковые константы с суффиксом `_t`), т.к. только Type Enforcement используется по-умолчанию (и его достаточно даже для систем, сертифицированных PCI DSS)

Четвертый компонент, бывает, скрыт



# Z means Selinux

Большинство утилит имеет параметр -Z, для работы с selinux contexts

```
# ps -eZ
```

```
system_u:system_r:lvm_t:s0      2272 ?          00:00:00 lvmetad
system_u:system_r:syslogd_t:s0  2345 ?          00:00:00 rsyslogd
system_u:system_r:auditd_t:s0   2371 ?          00:00:00 auditd
system_u:system_r:chronyd_t:s0  2469 ?          00:00:00 chronyd
system_u:system_r:crond_t:s0-s0:c0.c1023 2520 ? 00:00:00 crond
system_u:system_r:policykit_t:s0 23255 ?        00:00:00 polkitd
```

```
# ls -Z
```

```
drwxr-xr-x. root root system_u:object_r:var_t:s0      adm
drwxr-xr-x. root root system_u:object_r:var_t:s0      cache
drwxr-xr-x. root root system_u:object_r:kdump_crash_t:s0 crash
drwxr-xr-x. root root system_u:object_r:system_db_t:s0 db
drwxr-xr-x. root root system_u:object_r:var_t:s0      empty
```





# Контекст процесса

Как и всё в Unix - наследуется от родителя

Для того, что бы процесс получил нужный контекст, существует процесс transition

```
# ls -Z /usr/sbin/httpd
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/
sbin/httpd

# sesearch -s httpd_t -t httpd_exec_t -c file -p execute -Ad
Found 1 semantic av rules:
    allow httpd_t httpd_exec_t : file { ioctl read getattr lock
execute execute_no_trans entrypoint open } ;
```



# Selinux режимы работы

**/etc/selinux/config:**

```
# 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.
```

**Узнать, в каком режиме - `sestatus`**

**Для отладки можно отключить запрет:  
`setenforce 0`**



# chcon, restorecon, autorelabel

**chcon** меняет контекст для файлов... до первой okazji, потому что в некоторых случаях автоматически запускается процесс **restorecon**

**Правильно задавать контекст ветвей файловой системы нужно через semanage**

```
# semanage fcontext -a -t httpd_sys_content_t "/html(/.*)?"
```

**после чего уже выполнить**

```
# restorecon /html
```

**Ну, впрочем, вы уже сталкивались с этим в ДЗ по systemd ;-)**

**Посмотреть текущие правила**

```
# sesearch -A -s httpd_t
```



# Как я перестал бояться и полюбил бомбу^W selinux\*

```
# audit2why < /var/log/audit/audit.log
# audit2allow -M httpd_add --debug < /var/log/audit/audit.log
# semodule -i httpd_add.pp
# vi httpd_add.te
# semodule -r httpd_add
# checkmodule -M -m -o httpd_add.mod httpd_add.te
# semodule_package -o httpd_add.pp -m httpd_add.mod
# semodule -i httpd_add.pp
```



# Как я перестал бояться и полюбил бомбу^W selinux\_

## Makefile

```
SRCS := $(wildcard *.te)
PPS := $(patsubst %.te,%.pp,$(SRCS))

all: $(PPS)

%.mod: %.te
    checkmodule -M -m -o $@ $<

%.pp: %.mod
    semodule_package -o $@ -m $<
```



# Параметризованные политики

Разные правила для разных случаев были бы не так удобны, но есть параметры политик, определяемые переменными с булевым типом.

Параметры управляются утилитами:

- `getsebool`
- `setsebool`

`audit2why` знает умеет искать причину и в этих параметрах





**Спасибо  
за внимание!**

**Вопросы?**