



# Онлайн-образование



**Не забыть включить запись!**







# Меня хорошо видно && слышно?

Ставьте ☐+, если все хорошо  
Напишите в чат, если есть проблемы



# Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack #канал группы или #general



Вопросы вижу в чате, могу ответить не сразу





SELinux

Викирюк Павел

Системный инженер



# Маршрут вебинара

SELinux - что это?



Основные термины и понятия



Как работать с SELinux?

# Цели занятия | После занятия вы сможете

- 1 Понять, что такое системы принудительного контроля доступа
- 2 Узнать SELinux поближе и познакомиться с инструментами для работы с ним
- 3 Перестать бояться SELinux и научиться управлять его политиками

# Смысл | Зачем вам это уметь

1 Чтобы понимать, как работает система SELinux и на что она влияет

2 Чтобы отлаживать процесс работы сервисов и приложений с учетом взаимодействия с SELinux

3 Чтобы усилить меры безопасности вашей инфраструктуры





# История создания SELinux



# История создания SELinux

**SELinux** (англ. *Security Enhanced Linux*) — система принудительного (мандатного) контроля доступа (Mandatory Access Control)


- разрабатывалась в National Security Agency (NSA, Агентство национальной Безопасности, АНБ)
- в ядре впервые появилась в 1998 году
- в 2003 году появился фреймворк LSM для реализации подключаемых модулей безопасности и SELinux вошел в состав ядра 2.6.x
- аналогичная система - AppArmor появилась только в ядре 2.6.36 в 2009 году в SUSE





# SELinux: назначение





# Вопрос к аудитории: Что такое системы MAC (Mandatory Access Control)?



# SELinux: назначение

## Основные особенности применения:

- гибкое ограничение прав пользователей и процессов на уровне ядра
- работа совместно с **DAC** (discretionary access control или матричное управление доступом)
- снижение риска, возникающего вследствие допущенных ошибок в коде, приложении или конфигурации
- ограничение потенциально опасных или скомпрометированных процессов в правах
- протоколирование действий в системе





# SELinux: мифы и реальность







**Вопрос к аудитории:**

**За что не любят SELinux?**



# SELinux: мифы и реальность

## Факты о SELinux:

- очень долгое развитие проекта
- большие и сложные политики
- каждый ресурс должен быть описан и сопоставлен с сервисом
- большое количество ошибок (см. предыдущий пункт)
- отсутствие инструментов для работы с политиками “из коробки”
- минимальные накладные расходы на работу системы



# SELinux: мифы и реальность

## **Вывод:**

Отключение SELinux равноценно установке прав 777 на каталог, вместо того, чтобы разбираться с правами и группами



# Маршрут вебинара

SELinux - что это?



Основные термины и понятия



Как работать с SELinux?





# SELinux: термины и понятия





# SELinux: термины и понятия

**Субъект** - пользователь или процесс, то есть то, что выполняет действия в системе

**Объект** - то над чем выполняются действия, то есть файлы, порты, сокеты и прочее



# SELinux: термины и понятия

## Механизмы мандатного управления доступом:

**MLS** (Multi-Level Security, многоуровневая система безопасности) и **MCS** (Multi-Category Security, мультикатегорийная система безопасности)

- MLS базируется на формальной модели **модели Белла-Лападулы**
- все субъекты и объекты имеют свой уровень допуска
- субъект с определенным уровнем допуска имеет право читать и создавать (писать/обновлять) объекты с тем же уровнем допуска
- кроме того, он имеет право читать менее секретные объекты и создавать объекты с более высоким уровнем
- субъект никогда не сможет создавать объекты с уровнем допуска ниже, чем он сам имеет, а также прочесть объект более высокого уровня допуска.

**В оригинале:** «write up, read down» и «no write down, no read up»



# SELinux: термины и понятия

## Механизмы мандатного управления доступом:

**MLS** (Multi-Level Security, многоуровневая система безопасности) и **MCS** (Multi-Category Security, мультикатегорийная система безопасности)

## Особенности:

- вертикальный уровень контроля
- использование меток
- ограничение для субъектов доступа к объектам с ограниченным допуском на основе меток
- сложное проектирование системы безопасности
- применение в определенных сферах согласно их требованиям



# SELinux: термины и понятия

## Механизмы мандатного управления доступом:

**RBAC** (Roles Based Access Control, управление доступом на основе ролей)

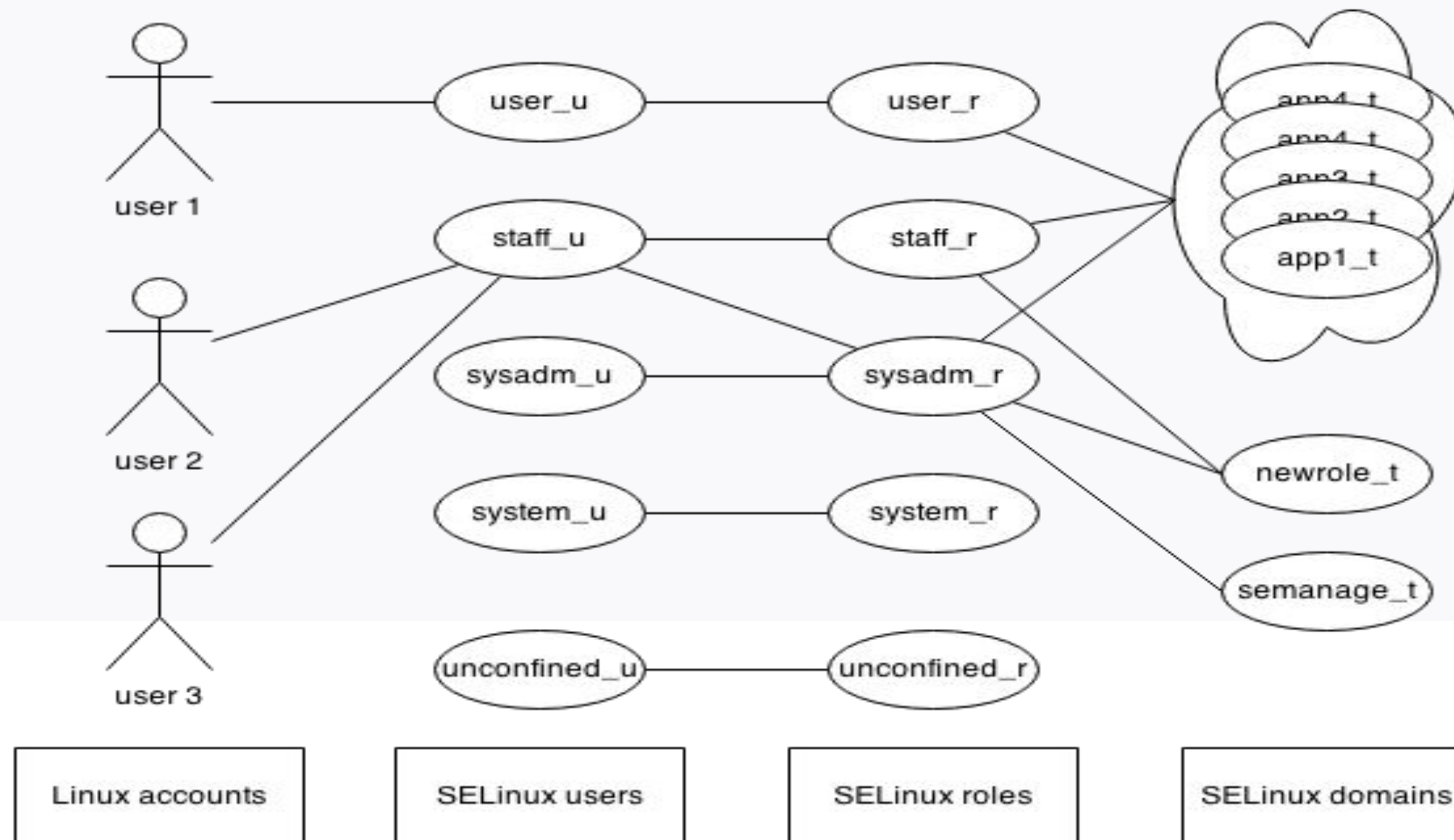
- контроль доступа к объектам файловой системы через роли, созданные на основании требований бизнеса или других критериев
- роли могут быть разных типов и уровней доступа к объектам



# SELinux: термины и понятия

## Механизмы мандатного управления доступом:

**RBAC** (Roles Based Access Control, управление доступом на основе ролей)





# SELinux: термины и понятия

## Механизмы мандатного управления доступом:

**TE** (Type Enforcement, принудительная типизация доступа)

**Контекст безопасности (context)** - по сути та же метка, выглядит как строка переменной длины и хранится в расширенных атрибутах файловой системы. Объединяет в себе **роли, типы и домены**

**Домен (domain)** - список действий, которые может выполнять процесс по отношению к различным объектам

**Тип (type)** - атрибут объекта, который определяет, кто может получить к нему доступ

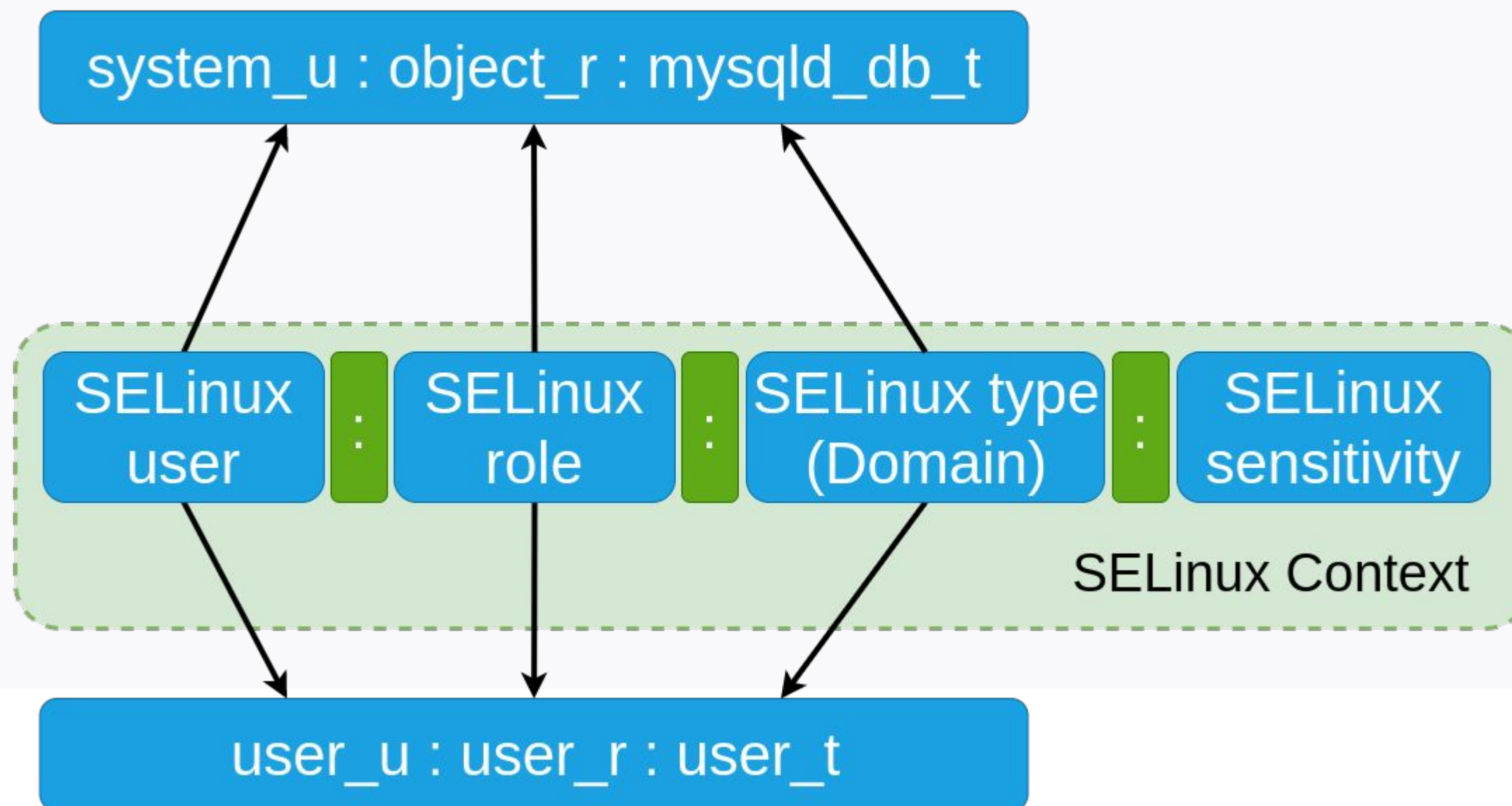
**Роль** - атрибут, который определяет, в какие домены может входить пользователь, то есть какие домены пользователь имеет право запускать



# SELinux: термины и понятия

**Механизмы мандатного управления доступом:**

**TE** (Type Enforcement, принудительная типизация доступа)

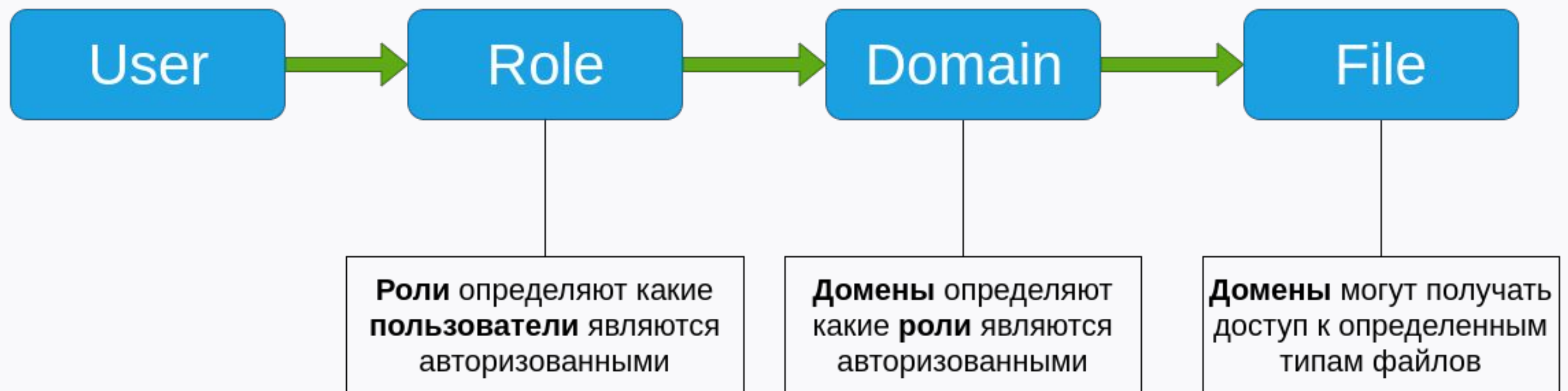




# SELinux: термины и понятия

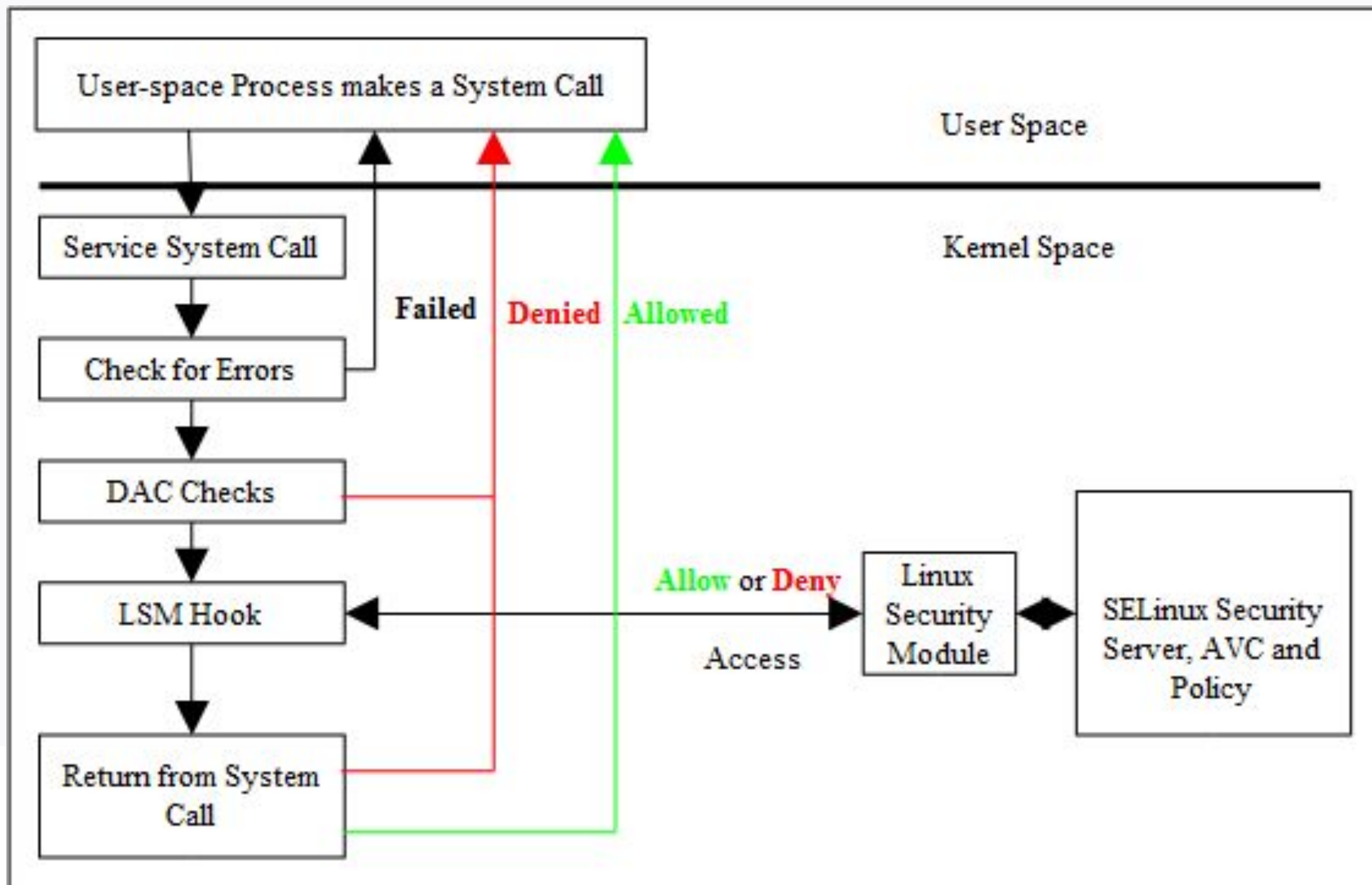
**Механизмы мандатного управления доступом:**

**TE** (Type Enforcement, принудительная типизация доступа)



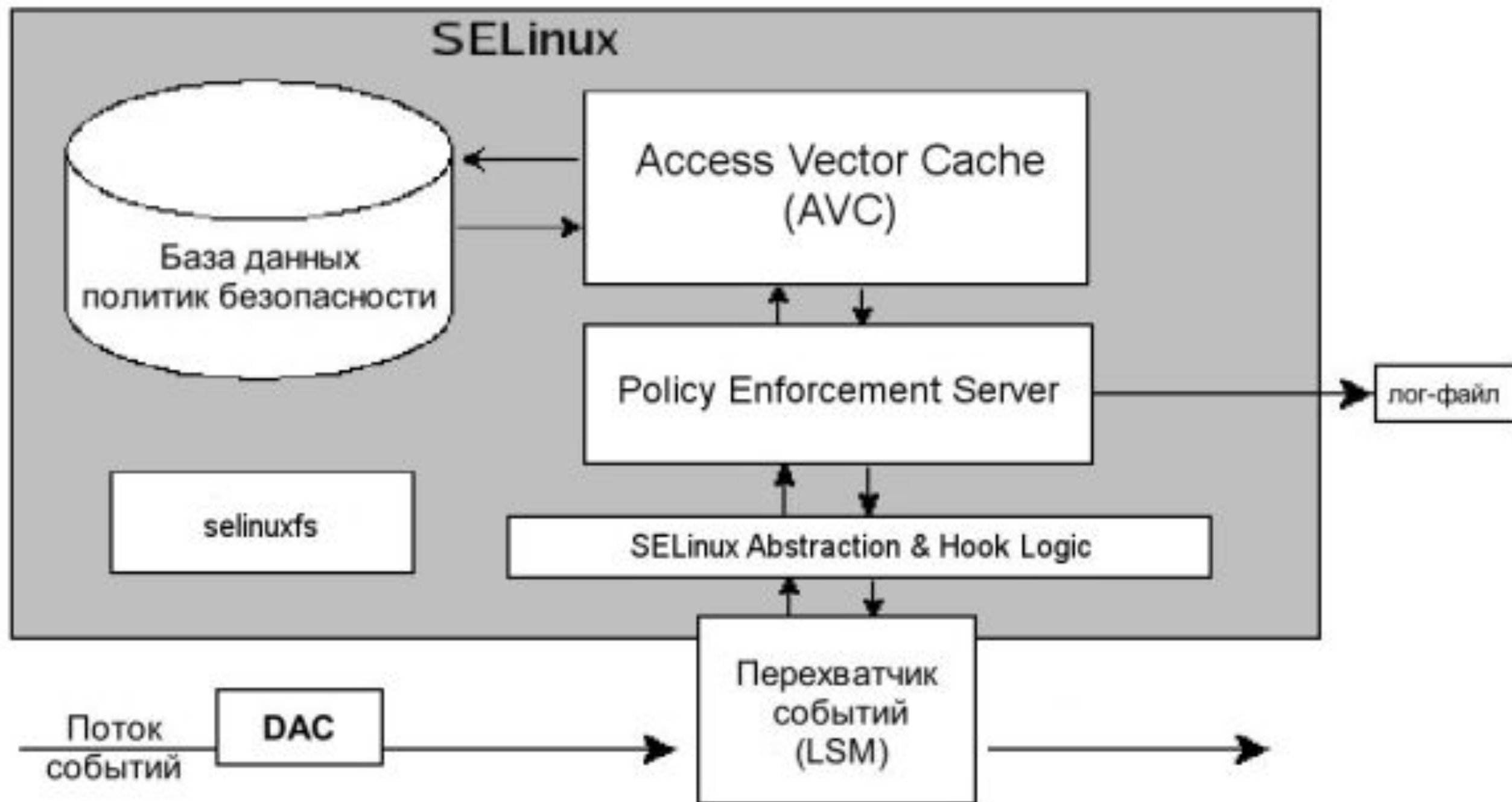


# SELinux: термины и понятия



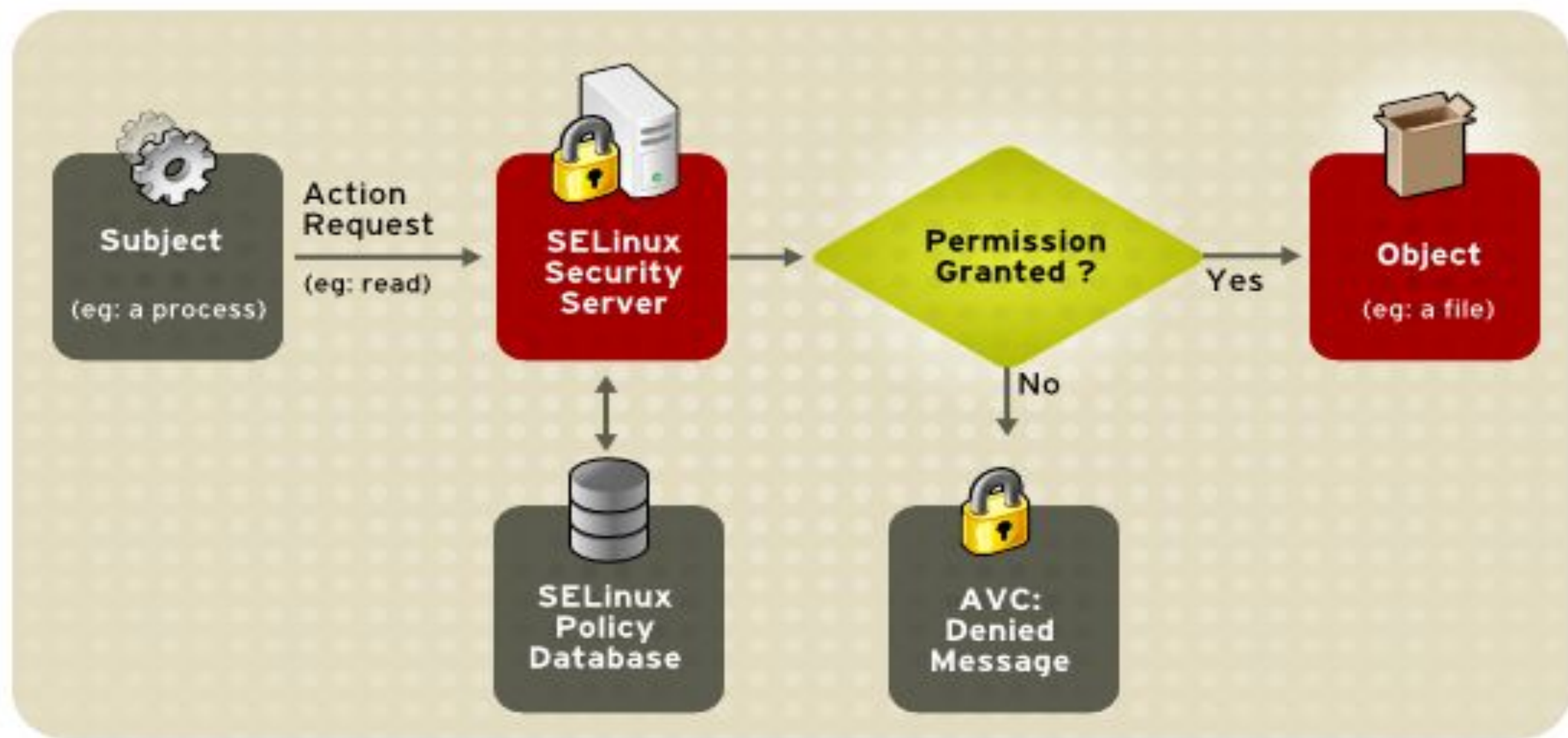


# SELinux: термины и понятия





# SELinux: термины и понятия







**Ваши вопросы?**



# Маршрут вебинара

SELinux - что это?



Основные термины и понятия



Как работать с SELinux?





# Как работать с SELinux?





# Основные инструменты



# SELinux: основные инструменты

## **Пакет setools-console:**

- sestatus
- seinfo
- findcon
- getsebool
- setsebool

## **Пакет polycoreutils-python:**

- audit2allow
- audit2why

## **Пакет polycoreutils-newrole:**

- newrole



# SELinux: основные инструменты

Смотрим контекст безопасности каталога пользователя

```
ls --context /root/  
-rw----- . root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg  
-rw----- . root root system_u:object_r:admin_home_t:s0 original-ks.cfg
```

**system\_u:object\_r:admin\_home\_t:s0**

**system\_u** - информация о пользователе

**object\_r** - роль пользователя

**admin\_home\_t** - тип или домен

**s0** - уровень MLS



# SELinux: основные инструменты

Смотрим информацию о правах пользователей

```
semanage login -l
```

Login Name	SELinux User	MLS/MCS Range	Service
__default__	unconfined_u	s0-s0:c0.c1023	*
root	unconfined_u	s0-s0:c0.c1023	*
system_u	system_u	s0-s0:c0.c1023	*



# SELinux: основные инструменты

Смотрим контекст безопасности объекта

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



# SELinux: основные инструменты

Смотрим контекст безопасности процесса

```
ps -Z 6798
LABEL                PID TTY   STAT TIME COMMAND
system_u:system_r:httpd_t:s0 6798 ?     Ss   0:00 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf
```



# SELinux: основные инструменты

Ищем правила преобразования, которые соответствуют этим типам

```
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 map execute execute_no_trans  
entrypoint open } ;
```



# SELinux: основные инструменты

Пример поиска правил для типа httpd\_t

```
sesearch -A -s httpd_t | grep 'allow httpd_t'
```

```
allow httpd_t zarafa_var_lib_t : dir { ioctl read write create getattr setattr lock unlink link rename  
add_name remove_name reparent search rmdir open } ;
```

```
allow httpd_t antivirus_t : process transition ;
```

```
allow httpd_t dirsrvadmin_unconfined_script_t : process { transition sigchld sigkill sigstop  
signull signal } ;
```

```
allow httpd_t httpd_unconfined_script_t : process { transition sigchld sigkill sigstop signull  
signal } ;
```

```
allow httpd_t httpd_tmpfs_t : fifo_file { ioctl read write create getattr setattr lock append unlink  
link rename open } ;
```

```
allow httpd_t jetty_log_t : lnk_file { ioctl read write create getattr setattr lock relabelfrom  
relabelto append unlink link rename } ;
```



# SELinux: основные инструменты

## Особенности работы:

### Как происходит наследование типов в SELinux?

Примерно также, как и все прочие права в Linux - например в случае создания файлов в каталоге, известном определенному контексту, файл наследует тип этого каталога.

**Context transition** (переход контекста) может быть инициирован политикой, такими инструментами, как `runcon`, или с помощью SELinux API

Переход **Process context (domain)** может происходить при наличии трех условий:

- целевой контекст файла является исполняемым для исходного домена
- целевой контекст файла помечен как точка входа для целевого домена
- исходный домен разрешен для перехода в целевой домен



# SELinux: основные инструменты

## Особенности работы:

### А если мне нужно запустить самосборное приложение?

Если нужно запустить несговорчивое или сомнительное приложение - запускать его надо из каталога **/opt**, в нем SELinux не работает

### А где же лежат все эти контексты и как они выглядят?

Контексты лежат вот по этому пути:  
**/etc/selinux/targeted/contexts/files**

и выглядят как обычные текстовые конфиги





# SELinux: режимы работы



# SELinux: основные инструменты

Конфигурация SELinux:

```
/etc/selinux/config
```

Узнать в каком режиме SELinux сейчас:

```
sestatus или getenforce
```

Отключить SELinux:

```
setenforce 0
```

Включить SELinux:

```
setenforce 1
```





# SELinux: приемы работы





# SELinux: работа с контекстом





# SELinux: приемы работы

**Меняем тип в контексте каталога:**

```
chcon -R -t type /home/user
```

**Проверяем контекст каталога:**

```
ls -Z /home/user
```

**Восстанавливаем контекст каталога:**

```
restorecon -v /home/user
```





# SELinux: audit2why vs audit2allow



# SELinux: приемы работы

## 1. Очищаем audit.log:

```
echo > /var/log/auditd/audit.log
```

## 2. Включаем в SELinux режим permissive:

```
setenforce 0
```

## 3. Запускаем приложение и получаем ошибки в audit.log

## 4. Смотрим ошибки и рекомендации в audit.log

```
audit2why < /var/log/audit/audit.log
```

## 5. Формируем модуль с правилами для SELinux из данных лога

```
audit2allow -M httpd_add --debug < /var/log/audit/audit.log
```

## 6. Загружаем модуль

```
semodule -i httpd_add.pp
```





# SELinux: параметризованные ПОЛИТИКИ





# SELinux: приемы работы

## Параметризованные политики SELinux

- представляют из себя политики, которые описаны переменные с булевым типом (on/off)
- управляются утилитами: **getsebool** и **setsebool**



# SELinux: приемы работы

## Параметризованные политики SELinux

### Пример работы:

### Просмотр политик в отношении сервиса samba:

```
getsebool -a | grep samba
```

### Меняем значение выбранной политики:

```
setsebool -P samba_share_fusefs on
```





**Ваши вопросы?**



# Рефлексия

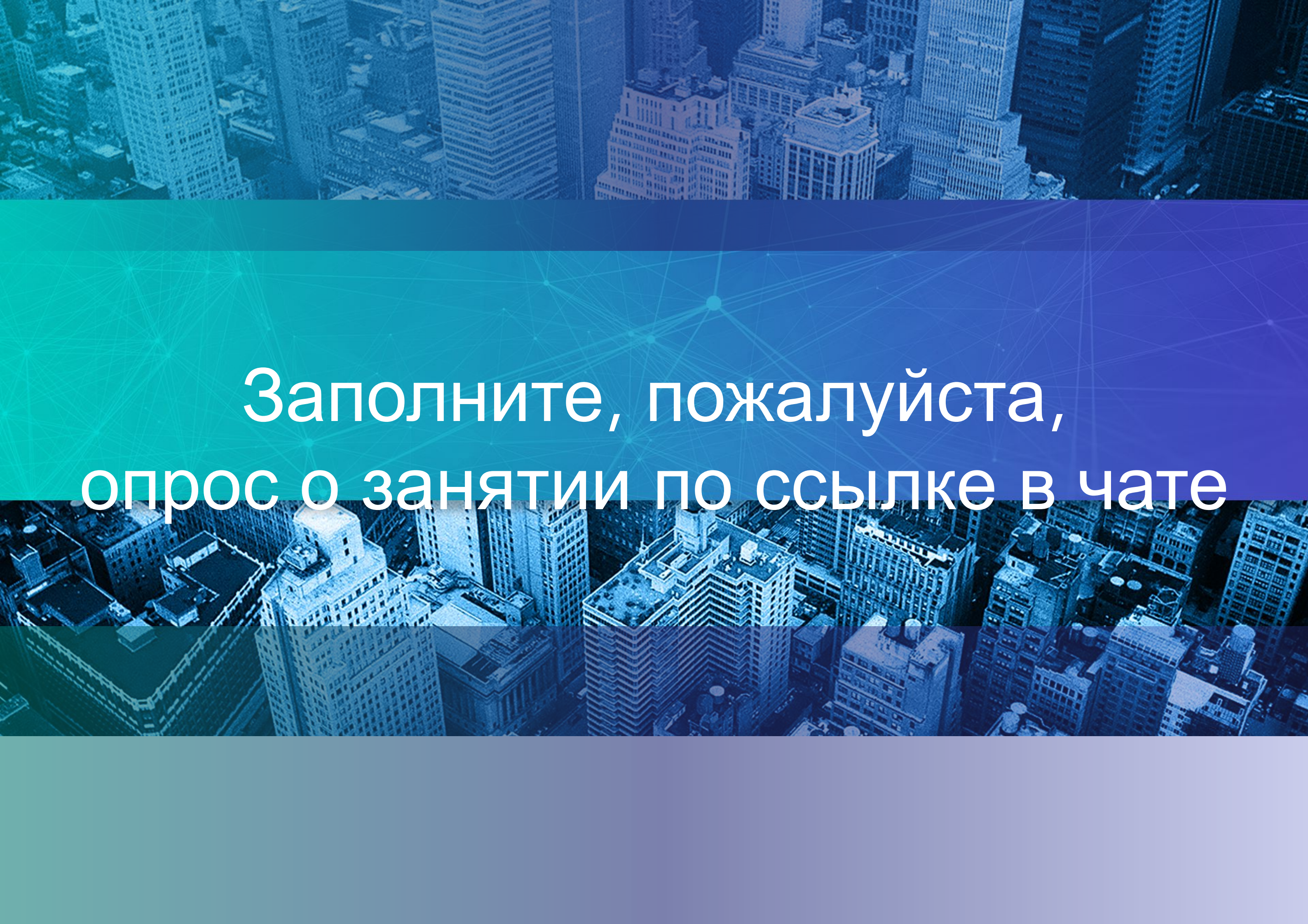


Назовите 3 момента, которые вам запомнились в процессе занятия



Что вы будете применять в работе из сегодняшнего вебинара?





Заполните, пожалуйста,  
опрос о занятии по ссылке в чате





Спасибо за внимание!  
Приходите на следующие вебинары

Викирюк Павел

Системный инженер