

Повышение привилегий доменных пользователей с помощью правил SUDO

ALD Pro

Exported on 08/11/2023

Table of Contents

| | | |
|----------|--|-----------|
| 1 | Что такое правила SUDO | 4 |
| 2 | Механизм работы правил SUDO | 7 |
| 3 | Механизм получения правил SUDO из LDAP | 9 |
| 4 | Настройка правил SUDO в домене..... | 11 |
| 4.1 | Через портал управления ALD Pro | 11 |
| 4.1.1 | Создание команды..... | 11 |
| 4.1.2 | Создание правила..... | 11 |
| 4.2 | Через web-интерфейс FreeIPA | 15 |
| 4.2.1 | Создание команды..... | 15 |
| 4.2.2 | Создание правила..... | 17 |
| 4.3 | Через терминал | 18 |
| 4.3.1 | Создание команды..... | 18 |
| 4.3.2 | Создание правила..... | 19 |
| 5 | Отладка правил SUDO | 21 |
| 5.1 | Список правил пользователя | 21 |
| 5.2 | Журнал отладки sudo..... | 21 |
| 5.3 | Журнал отладки SSSD | 22 |
| 6 | Лучшие практики | 25 |
| 6.1 | Работа с локальными настройками sudo..... | 25 |
| 6.2 | Будьте осторожны с символами подстановки..... | 25 |
| 6.3 | Не разрешайте использование редактора vi..... | 26 |
| 6.4 | Использовать группы пользователей, оставлять комментарии | 27 |
| 6.5 | Принцип предоставления минимальных прав..... | 27 |

Авторы: Федоров Руслан, Анатолий Лысов

Для установки программного обеспечения и выполнения других задач администрирования пользователю нужны привилегии суперпользователя. Сотрудники могут использовать учетную запись root, но из соображений безопасности более корректным считается работать из-под обычной учетной записи и повышать привилегии только при выполнении отдельных команд. Еще более востребован указанный подход, когда часть административных прав нужно делегировать обычным пользователям, например, чтобы разрешить им перезапуск служб или установку приложений.

В ОС Windows повышение привилегий реализуется с помощью команды «Запуск от имени администратора», которая вызывает утилиту runas.exe с требуемыми параметрами. На компьютерах под управлением Linux аналогичного результата можно добиться с помощью утилиты sudo (Substitute User and do, подменить пользователя и выполнить), которая имеет богатые настройки и позволяет журналировать неудачные аутентификации.

Например, вызовом следующей команды обычный пользователь Иван Кузнецов может установить приложение htop, если ему разрешено запускать утилиту apt через sudo:

```
ivan.kuznetsov@dc-1:~$ sudo apt install htop
```

ВНИМАНИЕ!

Кроме sudo повышать привилегии возможно также с помощью команд su/runuser и битов SUID/GUID, но эти способы не являются предметом рассмотрения данной инструкции.

1 Что такое правила SUDO

Правила SUDO позволяют определенным пользователям на конкретных хостах выполнять отдельные команды с повышенными привилегиями, создавая, таким образом, дополнительный слой авторизации, также как и в случае правил HBAC. Отличие между этими видами правил заключается в том, что правила HBAC проверяются на уровне PAM-стека, а правила SUDO непосредственно утилитой sudo.

Для возможности использования утилиты sudo пользователю в первую очередь нужны права на обращение к этому приложению на уровне HBAC-правил, так как при вызове утилиты sudo сначала создается PAM-контекст, а уже потом утилита приступает к проверке правил. Если таковых прав у него не будет, то до проверки правил SUDO дело не дойдет.

Локальные настройки утилиты sudo находятся в файле /etc/sudoers, который назван так потому, что пользователей, кому разрешено повышать привилегии с помощью утилиты sudo, называют **sudo enabled users** или кратко sudoers. В файле могут быть строки трех типов: параметры по умолчанию, псевдонимы (алиасы, именованные списки или проще переменные) и сами правила. Синтаксис правил представлен на рисунке 1.

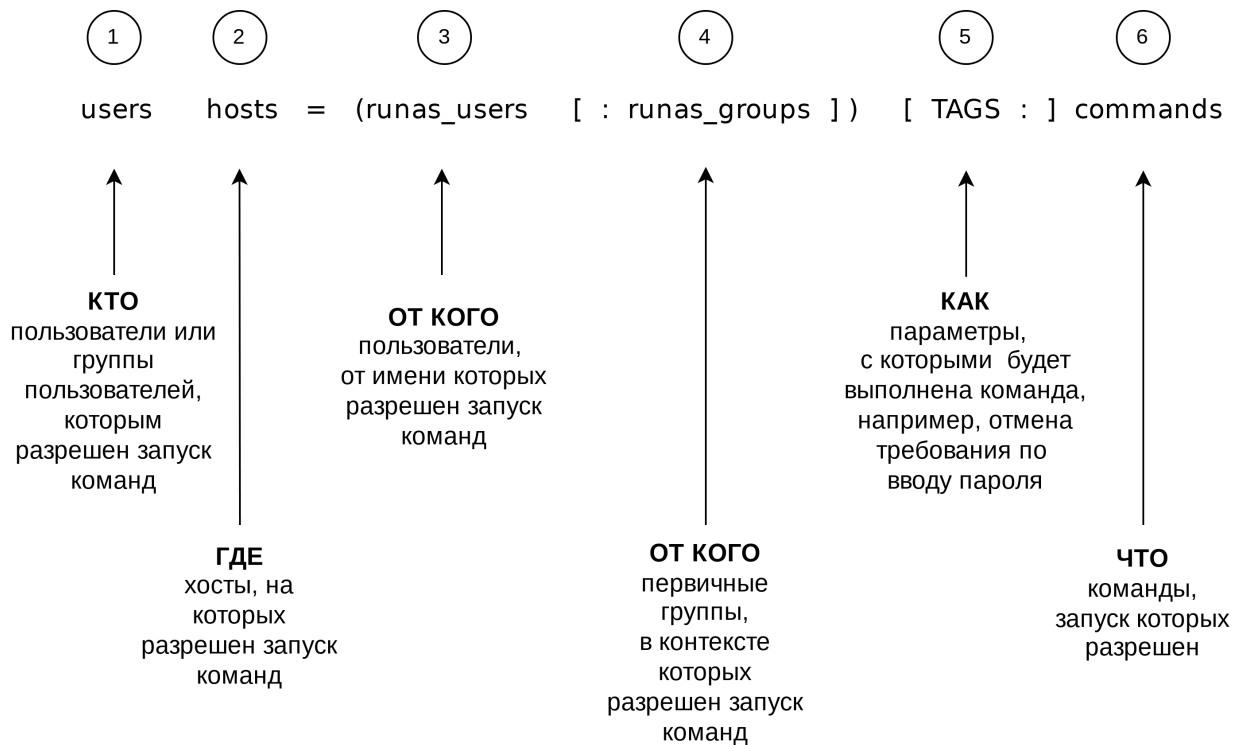


Рис. 1. Синтаксис правил SUDO

Правила могут быть как разрешающими, так и запрещающими, но по умолчанию считается, что прав на выполнение команд через sudo ни у кого нет. Для первых четырех компонентов правил следует определить область действия одним из двух способов:

- Любой субъект (ALL) — правило будет распространяться на все субъекты данного вида.
- Указанные субъекты — правило будет распространяться только на указанный перечень субъектов данного вида. Если требуется задать несколько значений, элементы списка должны быть разделены символом запятой. Для упрощения работы с большими списками синтаксис файла позволяет задавать именованные списки, или так называемые алиасы. Для удобства настройки синтаксис именованных списков позволяет исключать из них отдельные значения.

Давайте рассмотрим внимательнее каждый компонент правила:

- Пользователи и группы пользователей**, которым разрешен запуск команд в рамках данного правила. Перед именем группы следует указывать символ процента.

```
localuser    ALL=(ALL:ALL)          NOPASSWD:    /usr/bin/netstat
%localgroup  pc1=(root:root)        PASSWD:      /usr/bin/systemctl restart sssd
%localgroup  192.168.45.12=(root:root)  PASSWD:      /usr/bin/systemctl restart sssd
```

- Хосты**, на которых разрешен запуск команд. Это может быть имя компьютера или его IP адрес. Параметр полезен, если один и тот же файл копируется на несколько хостов.

```
localuser    ALL=(ALL:ALL)          NOPASSWD:    /usr/bin/netstat
%localgroup  pc1=(root:root)        PASSWD:      /usr/bin/systemctl restart sssd
%localgroup  192.168.45.12=(root:root)  PASSWD:      /usr/bin/systemctl restart sssd
```

- Пользователи, от имени которых разрешен запуск команд.** При выполнении команды через sudo по умолчанию предполагается, что команда запускается от имени root, но можно указать имя пользователя в явном виде с помощью ключа -u, и данный компонент правила позволяет ограничить перечень допустимых значений.

```
localuser    ALL=(ALL:ALL)          NOPASSWD:    /usr/bin/netstat
%localgroup  pc1=(root:root)        PASSWD:      /usr/bin/systemctl restart sssd
%localgroup  192.168.45.12=(root:root)  PASSWD:      /usr/bin/systemctl restart sssd
```

- Первичные группы, в контексте которых разрешен запуск команд.** По умолчанию используется первичная группа пользователя, от имени которого выполняется команда, но группу можно указать явно с помощью ключа -g. Данное значение проявляет себя, когда выполняются команды, которые создают новые файлы и папки, например, touch или mkdir. Этот параметр не является обязательным.

```
localuser    ALL=(ALL:ALL)          NOPASSWD:    /usr/bin/netstat
%localgroup  pc1=(root:root)        PASSWD:      /usr/bin/systemctl restart sssd
%localgroup  192.168.45.12=(root:root)  PASSWD:      /usr/bin/systemctl restart sssd
```

- Параметры, с которыми будет выполнена команда**, позволяют изменить поведение утилиты sudo, например, можно отключить запрос пароля с помощью параметра NOPASSWD. Этот параметр не является обязательным. Полный перечень доступных значений: EXEC, NOEXEC, FOLLOW, NOFOLLOW, LOG_INPUT, NOLOG_INPUT, LOG_OUTPUT, NOLOG_OUTPUT, MAIL, NOMAIL, PASSWD, NOPASSWD, SETENV и NOSETENV. О значении параметров можно посмотреть в справке man sudoers.

```
localuser    ALL=(ALL:ALL)          NOPASSWD:    /usr/bin/netstat
%localgroup  pc1=(root:root)        PASSWD:      /usr/bin/systemctl restart sssd
%localgroup  192.168.45.12=(root:root)  PASSWD:      /usr/bin/systemctl restart sssd
```

- Команды**, которые разрешено запускать в рамках этого правила. Это должен быть полный путь к исполняемому файлу, в конце строки можно указать допустимые параметры вызова.

```
localuser ALL=(ALL:ALL) NOPASSWD: /usr/bin/netstat
%localgroup pc1=(root:root) NOPASSWD: /usr/bin/systemctl restart sssd
%localgroup 192.168.45.12=(root:root) NOPASSWD: /usr/bin/systemctl restart sssd
```

Полный путь к исполняемым файлам можно узнать с помощью команды `which`, которую лучше запускать на целевых хостах, где предполагается запускать эти команды:

```
# which systemctl
/usr/bin/systemctl
```

В правилах SUDO можно использовать не только конкретные значения, но и шаблоны. В Astra Linux до версии 1.7.4 включительно используется проверенная версия `sudo` 1.8.x, в которой доступны только шаблоны в стиле `shell`, обработка которых выполняется через функции `glob` и `fnmatch`. С версии `sudo` 1.9.10 появится возможность использовать полноценные регулярные выражения.

В шаблонах можно использовать следующие символы подстановки (wildcards), или как их еще называют метасимволы:

- «?» – соответствует одному любому символу
- «*» – соответствует любому количеству любых символов, в т.ч. пустой строке. С использованием символа * следует быть крайне осторожным, подробнее смотри в разделе «5 Лучших практики».
- «\» – позволяет экранировать спецсимволы, т.е. отключить их управляющую функцию. Используется, когда нужен знак вопроса, звездочки, двоеточия и др.
- «"""» – соответствует пустой строке, если пустая строка указана в качестве единственного параметра команды, то эта команда может быть выполнена только без параметров.
- [...] – соответствует одному символу из указанного диапазона, например:
 - [abc] соответствует символу a, b или c;
 - [a-z] соответствует строчному символу латинского алфавита;
 - [[:lower:]] соответствует строчному символу латинского алфавита, но диапазон задан с помощью именованного класса символов (named character classes), полный перечень которых включает `alnum`, `alpha`, `blank`, `cntrl`, `digit`, `graph`, `lower`, `print`, `punct`, `space`, `upper`, `xdigit`.
- [!...] – восклицательный знак в начале диапазона позволяет инвертировать набор символов, т.е. шаблон соответствует любому символу, который не входит в указанный диапазон.

2 Механизм работы правил SUDO

Как уже было сказано, настройки утилиты sudo определяются содержимым файла `/etc/sudoers`. В этом файле находится также инструкция `#includedir /etc/sudoers.d/`, которая включает содержимое дополнительных файлов из указанной директории. Инструкция начинается с символа решетки #, как обычный комментарий, но комментарием не является, что может ввести в заблуждение. Сделано это так из соображений обратной совместимости, т.к. инструкции `include` и `includedir` были добавлены значительно позже, в 2004 и 2017 годах соответственно. С версии 1.9.1 появится возможность использовать символ @ собачки вместо решетки, что уменьшит путаницу.

В начале файла вы найдете предупреждение о том, что редактировать правила sudo напрямую не рекомендуется, и нужно воспользоваться утилитой `visudo`. Указанная утилита откроет файл в nano и обеспечит проверку синтаксиса перед сохранением изменений. Если вам ближе `vi` или `mcedit`, вы можете заменить редактор по умолчанию командой `sudo update-alternatives --config editor`.

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults          env_reset
Defaults          mail_badpass
Defaults          secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin"

# Host alias specification
# User alias specification
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d

%astra-admin    ALL=(ALL:ALL) ALL
```

Рис. 2. Содержимое файла `/etc/sudoers`

После предупреждения задано несколько параметров по умолчанию:

- Параметр **env_reset** позволяет ограничить набор переменных из среды окружения пользователя, которые будут доступны запускаемой утилите. Это важно из соображений безопасности, поскольку эти переменные могут влиять на поведение утилит, запускаемых с привилегиями супрепользователя.

- Параметр **mail_badpass** предписывает системе отправлять уведомления о неудачных попытках ввода пароля при выполнении команды `sudo`. Предполагается доставка в локальный почтовый ящик `/var/mail/root` через `exim`. На ALSE 1.7 с уровнем доступа Смоленск `exim` не заработает без дополнительных настроек системы мандатного контроля.
- Параметр **secure_path** позволяет задать список каталогов, в которых будет выполняться поиск запускаемых через `sudo` утилит, когда не указан полный путь к файлу. Это исключает запуск вредоносных приложений с повышенными привилегиями.

Давайте рассмотрим правила, представленные в файле `/etc/sudoers` сразу после установки операционной системы:

- Правило «`root ALL=(ALL:ALL) ALL`» означает, что пользователь `root` может на любом хосте от имени любого пользователя и в контексте любой первичной группы выполнить любую команду.
- Правило «`%sudo ALL=(ALL:ALL) ALL`» означает тоже самое для группы `sudo`.
- Правило «`%astra-admin ALL=(ALL:ALL) ALL`» означает тоже самое для группы `astra-admin`. Обратите внимание на тот факт, что после продвижения сервера доменный пользователь `admin` автоматически вносится в список участников локальной группы `astra-admin`, за счет чего получает право на выполнение команд от имени суперпользователя.

3 Механизм получения правил SUDO из LDAP

Правила SUDO можно хранить не только в локальных файлах, но и централизованно. Любой сервер каталогов можно сделать поставщиком правил SUDO, если расширить схему должным образом и назначить его источником правил. Список источников утилита sudo получает через библиотеку службы имен (Name Service Switch, NSS), настройки которой находятся в файле `/etc/nsswitch.conf`. В операционных системах Linux через этот механизм настраиваются источники для получения информации о пользователях, группах, DNS-записях и многом другом. Основные вызовы NSS реализованы в библиотеке `libc`, а та уже, в свою очередь, выполняет обращение к необходимым бэкендам, см. Рисунок 3.

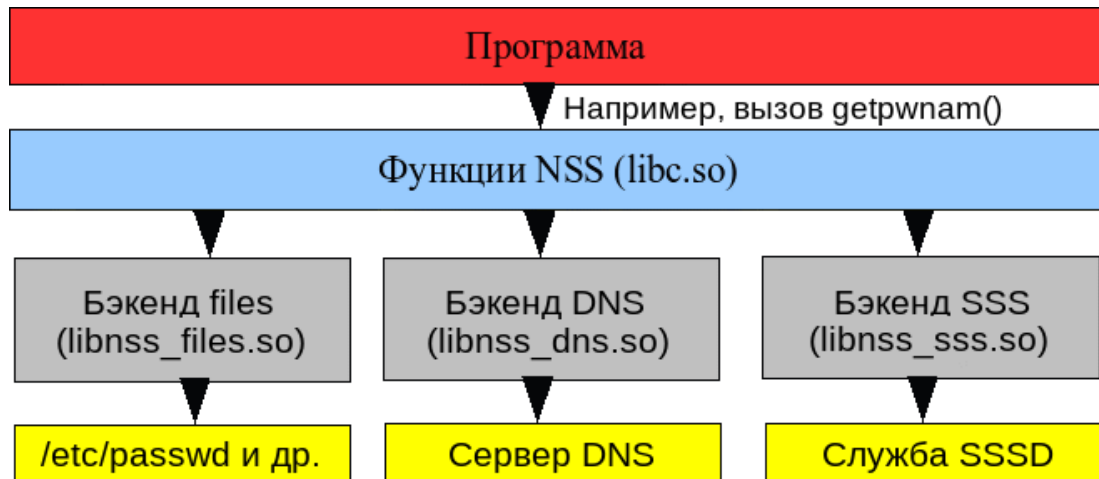


Рис. 3.

Архитектура диспетчера службы имен
(Name Service Switch, NSS)

После установки `freeipa-client` в файле `/etc/nsswitch` можно найти строку с настройкой базы данных `sudoers`. По умолчанию правила сначала берутся из локального файла, а затем через модуль `sss`, который отвечает за взаимодействие с LDAP-каталогом через службу SSSD.

```
$ cat /etc/nsswitch.conf
...
sudoers: files sss
...
```

Поддержка каталогов, появилась в `sudo` с выходом модуля `ldap` для `nss` в 2004 году. Источником правил для модуля служили записи из DN «ou=**sudoers**,dc=имя,dc=домена,dc=организации». Модуль использовал примитивную схему хранения данных, которая повторяла синтаксис локального файла `sudoers`, игнорируя доступную в каталоге нормализацию данных, например, в части пользователей, групп и хостов. Поэтому при реализации поддержки правил SUDO разработчики FreeIPA создали новую схему, лишенную указанных недостатков. Информация о правилах во FreeIPA хранится в DN «cn=**sudorules**,cn=sudo,dc=имя,dc=домена,dc=организации», и модуль `sss` через службу SSSD берет данные напрямую из этой ветки каталога.

Для обеспечения совместимости со старым модулем `ldap`, который все еще используют UNIX клиенты, например FreeBSD, служба каталога FreeIPA с помощью плагина `Compat` автоматически конвертирует настройки правил в старый формат, см. рисунок 3. Например, если для правила в `cn=sudorules`

установить `ipaEnabledFlag=FALSE`, то соответствующая запись в `ou=sudoers` будет автоматически удалена, но стоит вернуть атрибуту значение `TRUE` и запись будет автоматически воссоздана.

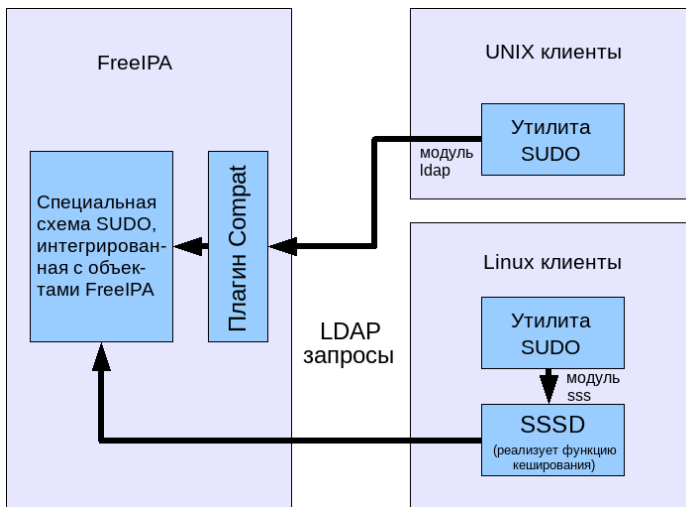


Рис. 4. Источник правил SUDO в зависимости от реализации клиентской части

Служба SSSD реализует дополнительно функцию кеширования, что дает пользователям возможность повышать свои привилегии, даже если они находятся вне домена. По умолчанию время кеширования составляет 5400 секунд, и для немедленного применения правил на клиентской машине необходимо выполнить очистку кеша следующими командами:

```
sudo systemctl stop sssd
sudo rm /var/lib/sss/db/*
sudo systemctl start sssd
```

Или воспользоваться инструментом `ssstcl`, входящим в пакет `sssd-tools`:

```
sudo ssstcl cache-remove
```

Еще один важный момент. В силу особенностей `FreelPA` в правилах `sudo` не получится использовать следующие группы:

- группу пользователей **`ipausers`**, т.к. у нее нет POSIX идентификатора и поэтому на целевых хостах служба SSSD не отображает участие пользователей в этой группе
- группу хостов **`ipaservers`**, т.к. у нее нет класса **`memberofentry`** и соответствующих зависимостей.

Самый простой способ обойти указанные проблемы – это создать вспомогательные группы `sudo-ipausers/sudo-ipaservers` и сделать группы `ipausers/ipaservers` их участниками. В этом случае вы сможете использовать вспомогательные группы в правилах SUDO без ограничений.

4 Настройка правил SUDO в домене

4.1 Через портал управления ALD Pro

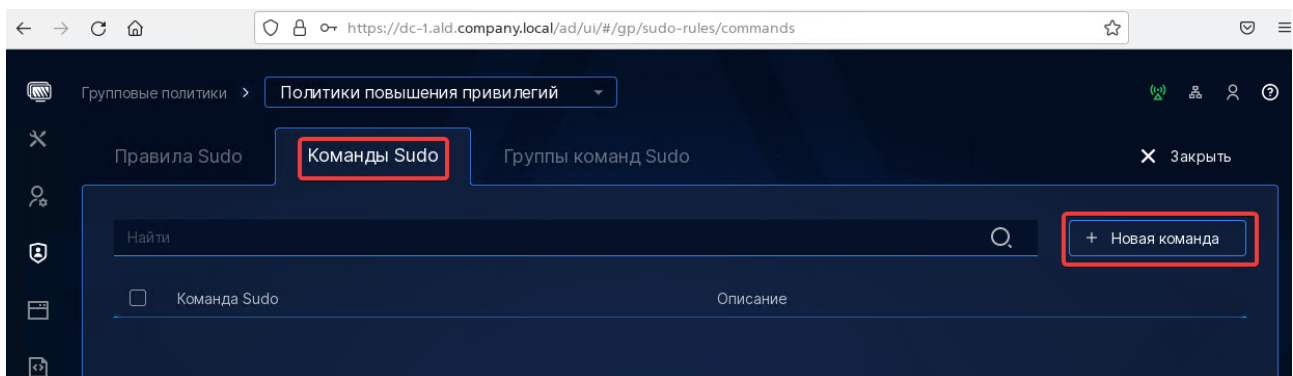
4.1.1 Создание команды

Учитывая, что пользователи и хосты уже есть в домене, настройку правил следует начать с создания команды. Сделать это можно через портал управления ALD Pro, веб-интерфейс FreeIPA или из командной строки. Единственно, интерфейс ALD Pro до версии 1.4 не позволяет использовать пробелы в названии команд, поэтому для создания команд с параметрами вам потребуется воспользоваться интерфейсом FreeIPA или командной строкой.

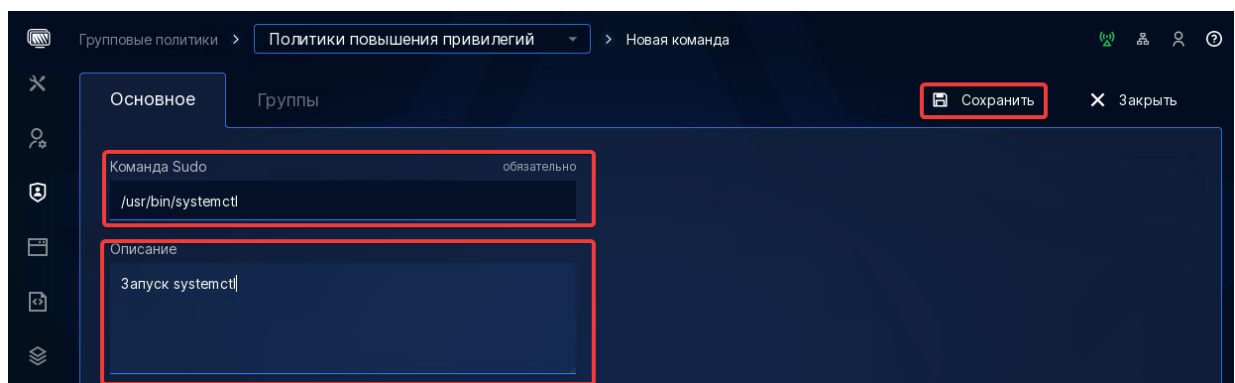
1. Откройте страницу «Групповые политики > Политики повышения привилегий > Команды Sudo»
2. Нажмите кнопку «+ Новая команда»

ВАЖНО!

В имени команды можно использовать `a-z`, `A-Z`, `0-9`, `-_./~` и пробелы (кроме первого и последнего).



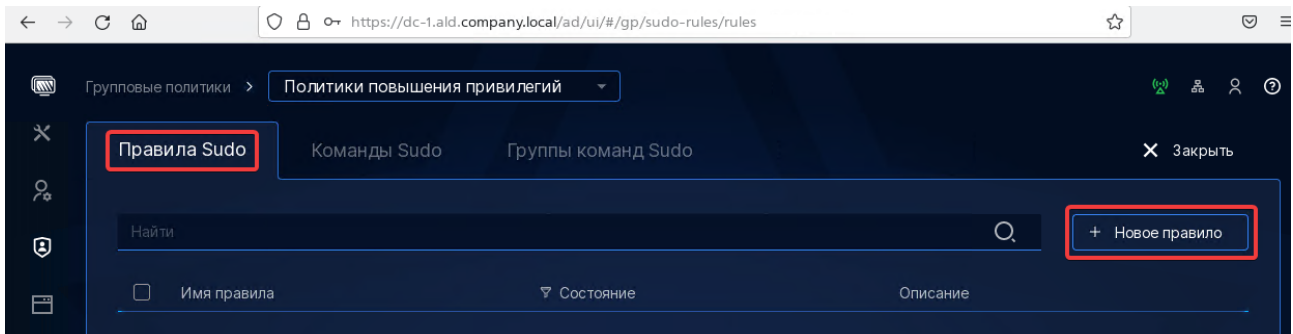
3. Введите команду `/usr/bin/systemctl`, описание (опционально) и нажмите «Сохранить»



4.1.2 Создание правила

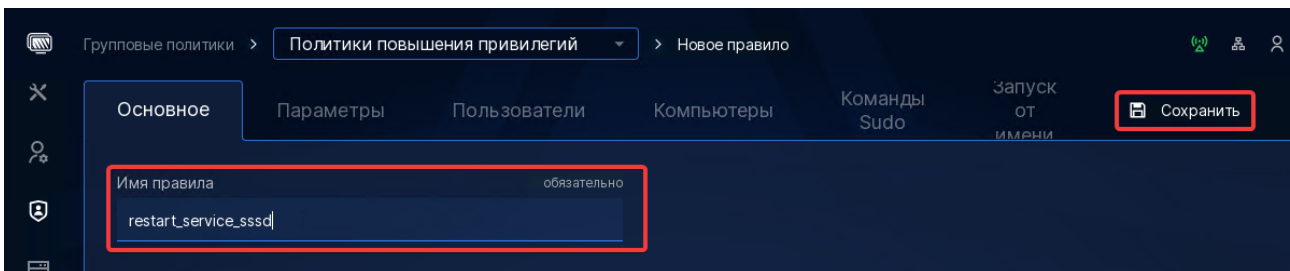
1. Откройте страницу «Групповые политики > Политики повышения привилегий > Правила Sudo»

2. Нажмите кнопку «+ Новое правило»

**ВАЖНО!**

В имени правила можно использовать `a-z`, `A-Z`, `0-9`, `-`, `_`

3. На странице «Новое правило» введите имя правила и нажмите кнопку «Сохранить». Имя не должно содержать заглавные буквы, пробелы и символы кириллицы.



4. На странице правила задайте следующие параметры:

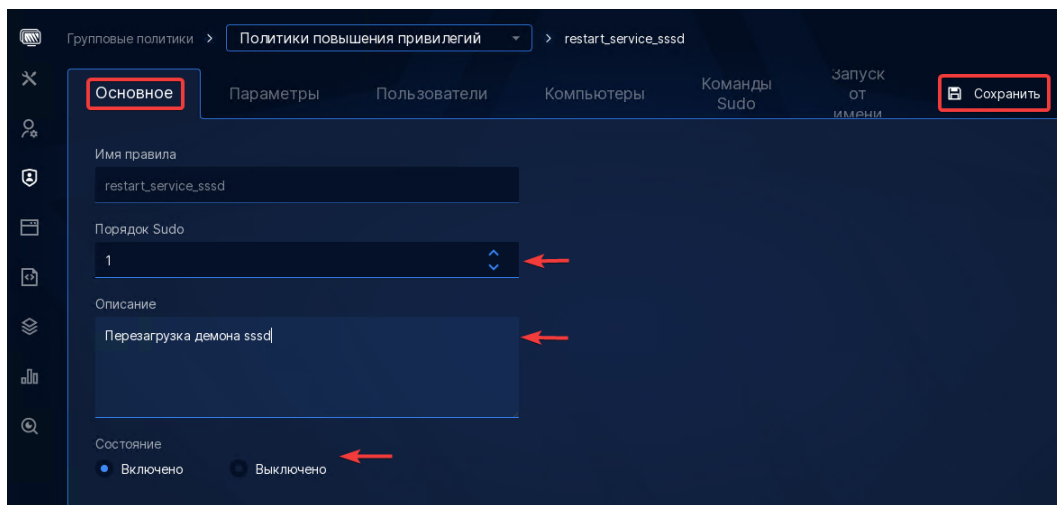
1. Раздел «Основные»

- Порядок sudo (необязательный параметр) – целое число, которое определяет очередность выполнения правил. Чем больше значение, тем позже обрабатывается правило, а значит оно может переопределить те правила, которые стоят перед ним.

Если список команд содержит несколько значений, они обрабатываются в указанном порядке. Если у правила одновременно заданы и разрешающие и запрещающие команды, то сначала обрабатываются разрешающие.

- Описание – необязательный комментарий к правилу
- Состояние – переключатель определяет, включено правило или нет

Обязательно сохраните изменения до перехода к следующей вкладке, иначе изменения будут утеряны.



б. Раздел «Параметры»

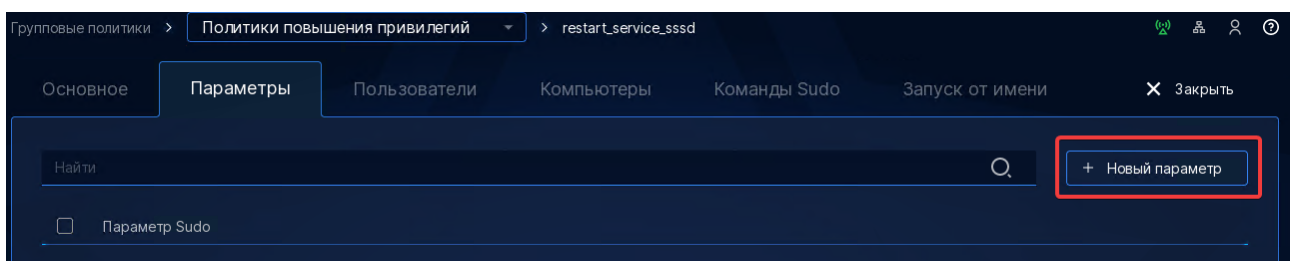
С помощью параметров можно изменить поведение утилиты для ее тонкой настройки. Ниже приведено несколько наиболее востребованных параметров:

- `authenticate` – с помощью этого флага можно обязать пользователей вводить пароль при выполнении команды через `sudo`. Параметр включен по умолчанию, и для отмены требования по вводу пароля его следует отключить, для чего нужно поставить восклицательный знак перед названием параметра «`!authenticate`»
- `passwd_tries` – задает количество попыток ввода пароля, прежде чем `sudo` завершит работу и зарегистрирует ошибку. Задается в виде переменной, по умолчанию `passwd_tries=3`
- `timestamp_timeout` – задает количество минут, которое должно пройти перед тем, как `sudo` повторно запросит пароль. Если установить таймаут равным 0, то утилита будет запрашивать пароль всегда, если установить отрицательное значение, таймаут будет отключен и введенный ранее пароль будет храниться бессрочно. По умолчанию таймаут составляет 15 минут.

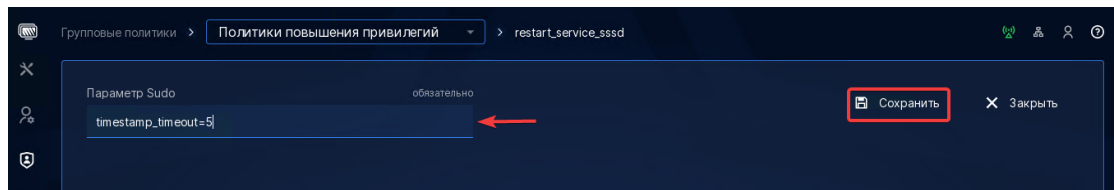
Информацию по остальным параметрам можно найти в `man sudoers`. Значения по умолчанию, с которыми утилита `sudo` была скомпилирована, можно узнать, вызвав команду `sudo -V` под суперпользователем, например «`sudo sudo -V`»

Для создания параметра:

1. Нажмите кнопку «+ Новый параметр»

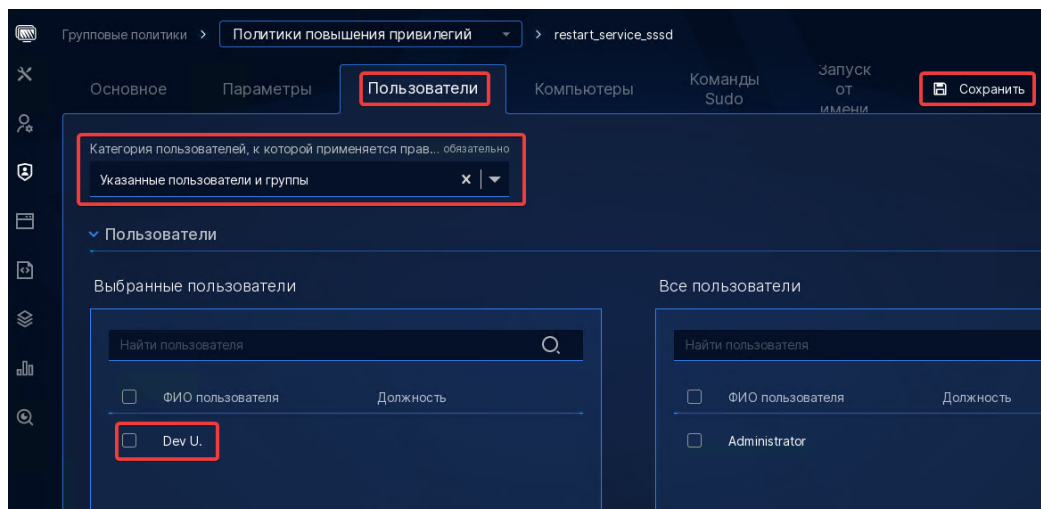


2. Введите параметр, например «`timestamp_timeout=5`», и нажмите кнопку «Сохранить»



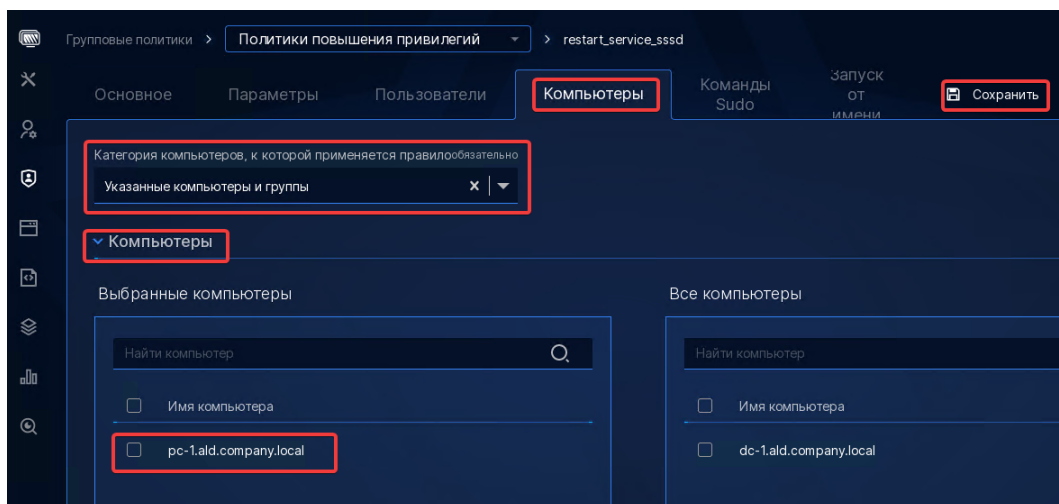
с. Раздел «Пользователи»

На этой вкладке можно задать список пользователей и их групп, которым в соответствии с этим правилом будет разрешено вызывать команды через sudo. Укажите пользователя и нажмите кнопку «Сохранить».



d. Раздел «Компьютеры»

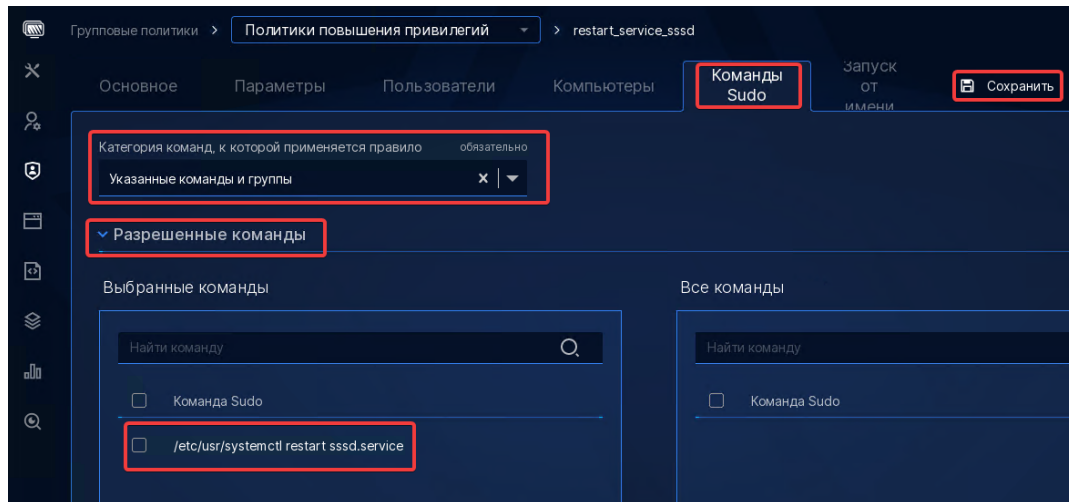
На этой вкладке можно задать список компьютеров или их групп, на которых в соответствии с этим правилом будет разрешено вызывать команды через sudo. Укажите компьютер и нажмите кнопку «Сохранить».



е. Раздел «Команды Sudo»

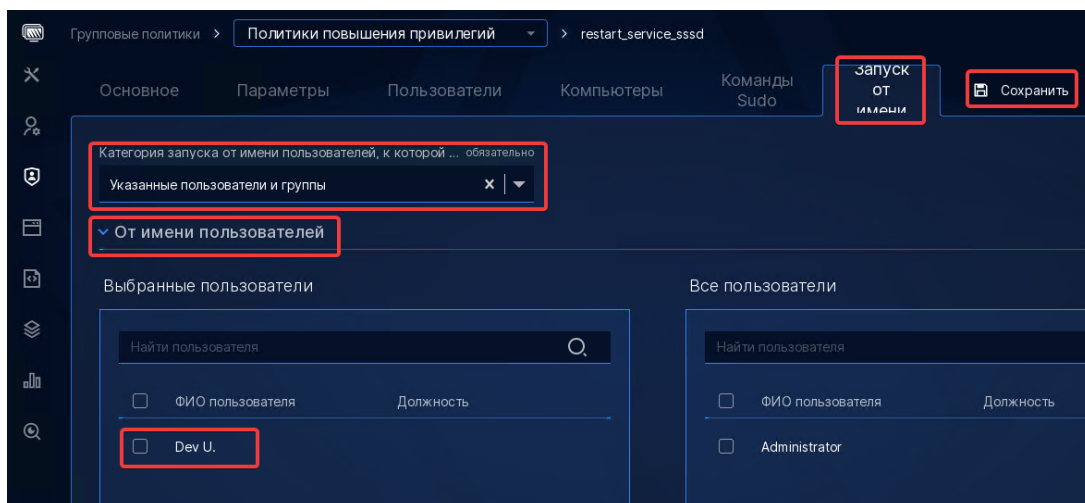
На этой вкладке можно задать список команд, которые разрешено/запрещено будет выполнять. Выберите команды и нажмите кнопку «Сохранить». Напоминаем, что сначала

применяются разрешающие команды, затем запрещающие, поэтому у запрещающих будет выше приоритет.



f. Раздел «Запуск от имени»

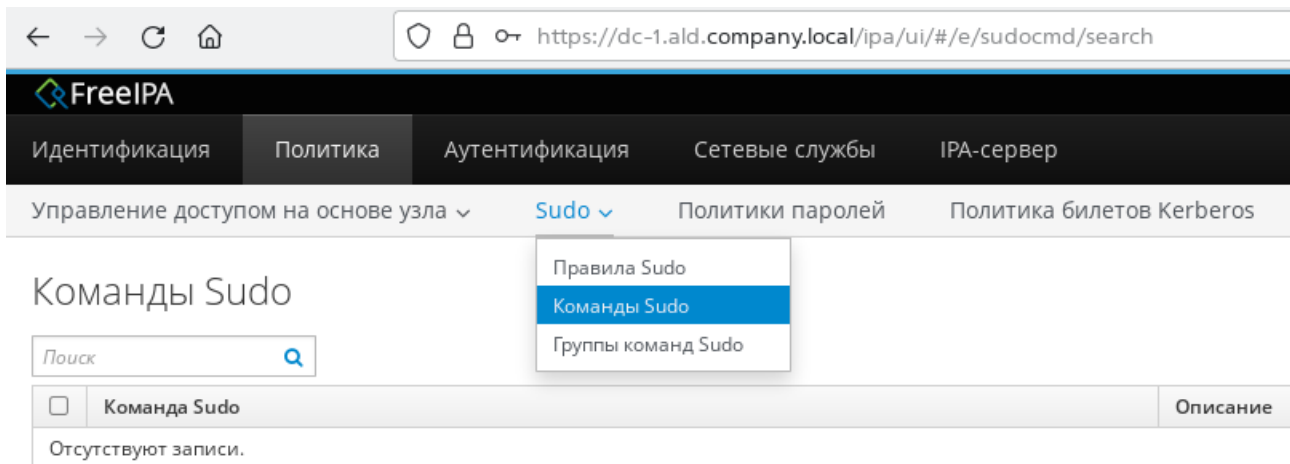
По умолчанию команды sudo запускаются от имени суперпользователя root в контексте его первичной группы, но это поведение можно изменить с помощью ключей -u и -g. На этой вкладке можно определить список пользователей и групп, от имени которых пользователь сможет действовать. Для того, чтобы разрешить действовать от суперпользователя следует оставить эту вкладку незаполненной.



4.2 Через web-интерфейс FreeIPA

4.2.1 Создание команды

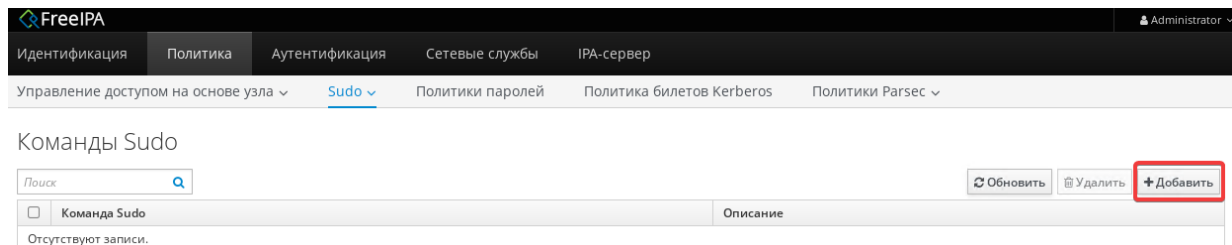
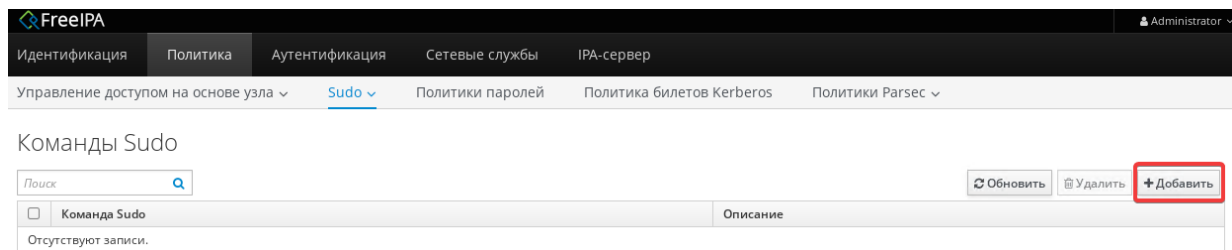
1. Откройте страницу «Политика > Sudo > Команды Sudo»



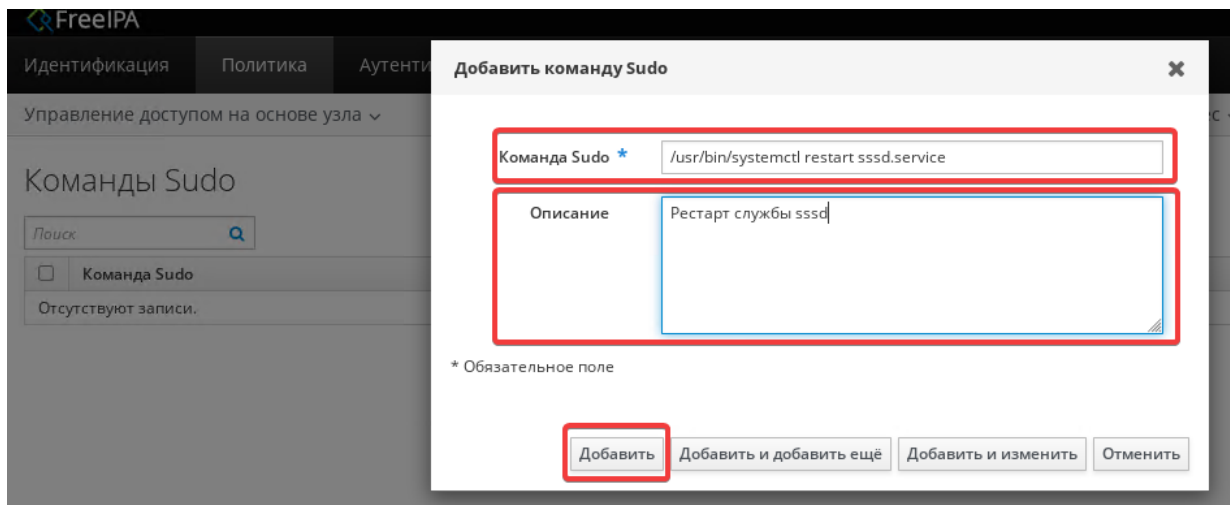
2. Нажмите кнопку «+ Добавить»

ВАЖНО!

В имени команды можно использовать `a-z`, `A-Z`, `0-9`, `-_./~` и пробелы (кроме первого и последнего).

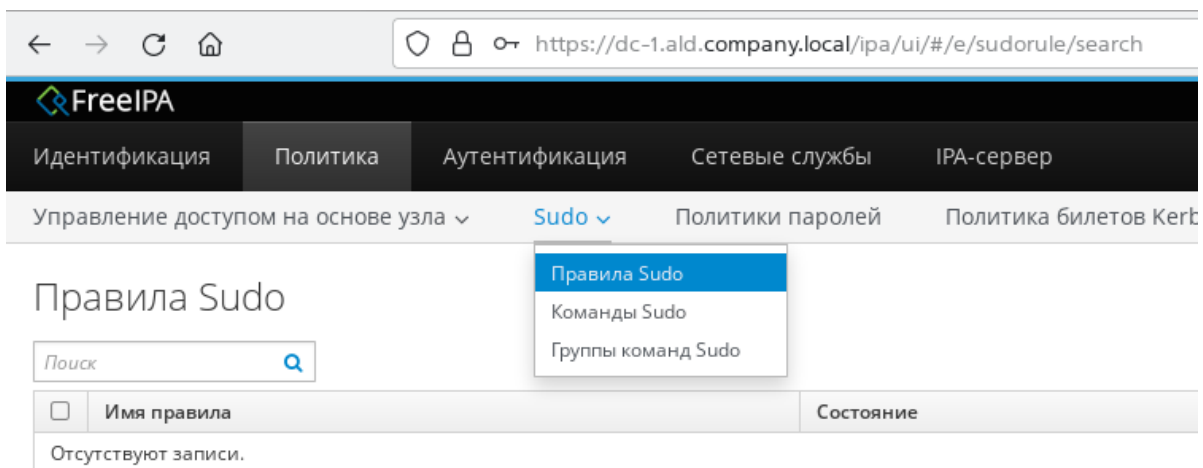


3. В открывшемся окне введите команду «`/usr/bin/systemctl restart sssd.service`» и её описание, нажмите кнопку «Добавить».



4.2.2 Создание правила

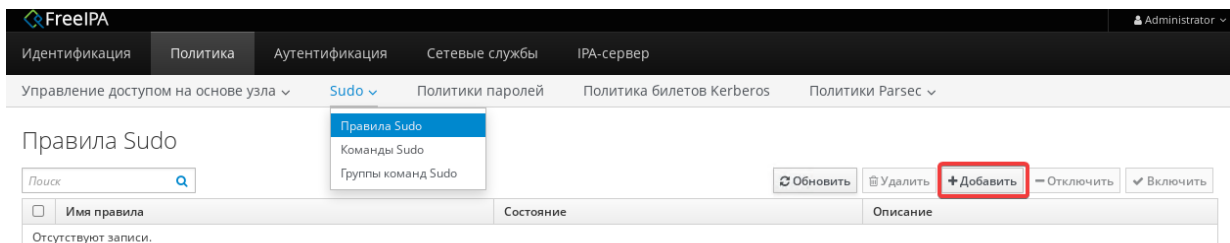
1. Откройте страницу «Веб-портал FreeIPA» Политика > Sudo > Правила Sudo»



2. Нажмите кнопку «+ Добавить»

ВАЖНО!

В имени правила можно использовать `a-z`, `A-Z`, `0-9`, `-`, `_`.



3. В открывшемся окне введите имя правила Sudo (обязательно) и нажмите кнопку «Добавить».

4. Укажите необходимые параметры

4.3 Через терминал

4.3.1 Создание команды

Создайте команду SUDO через терминал с помощью команды `sudocmd-add`:

```
ipa sudocmd-add '/etc/usr/systemctl restart sssd.service' --desc='sssd_daemon_restart'
```

где:

- sudocmd-add – название команды, с помощью которой можно создать в системе новую команду sudo
 - "/etc/usr/systemctl restart sssd.service" – полный путь к утилите и разрешенные параметры вызова
 - desc – ключ, который позволяет задать описание команды "sssd_daemon_restart"

ВАЖНО!

В имени команды можно использовать `a-z`, `A-Z`, `0-9`, `-_./~` и пробелы (кроме первого и последнего).

ПРИМЕЧАНИЕ

Возможно вы удивились почему мы назвали sssd демоном. Дело в том, что в мире Linux программы, которые запускаются самой системой и работают в фоновом режиме без прямого взаимодействия с пользователем принято называть не службами, как в Windows, а демонами (англ. daemon). Термин был впервые введен в обращение в далеком 1963 году разработчиками Массачусетского технологического института (MIT), которые посчитали, что демон будет подходящим названием для фонового процесса, который неустанно работал над выполнением системных задач. Причем, они использовали устаревшее написание слова daemon вместо более современного demon, но именно в таком виде термин и прижился.

4.3.2 Создание правила

1. Создайте правило SUDO через терминал с помощью команды sudorule-add:

```
ipa sudorule-add 'sssd_daemon_restart'
```

где:

- sudorule-add – команда, с помощью которой можно создать новое правило sudo
 - "sssd_daemon_restart" – имя нового правила.

ВАЖНО!

В имени правила можно использовать `a-z`, `A-Z`, `0-9`, `-_`

2. Добавьте пользователя в правило:

```
ipa sudorule-add-user 'sssd_daemon_restart' --users crashtest
```

3. Добавьте в правило целевые хосты:

```
ipa sudorule-add-host 'sssd_daemon_restart' --hosts client2
```

4. Добавьте команду в правило:

```
ipa sudorule-add-allow-command 'sssd_daemon_restart' --sudocmds="/usr/bin/systemctl  
restart sssd.service"
```

5. Проверьте результат:

```
ipa sudorule-show 'sssd_daemon_restart'
```

5 Отладка правил SUDO

5.1 Список правил пользователя

Результирующий набор правил SUDO для конкретного пользователя можно узнать вызовом на целевой машине команды `sudo` с ключами `-l` и `-U`:

```
root@dc-1:~# sudo -l -U admin
Matching Defaults entries for admin on dc-1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/
sbin\:/usr/bin\:/sbin\:/bin,
    secure_path=/usr/lib/parsec/bin\:/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/
usr/bin\:/sbin\:/bin

User admin may run the following commands on dc-1:
    (ALL : ALL) ALL
    (root) ALL
```

5.2 Журнал отладки sudo

Для включения журналирования требуется создать файл `/etc/sudo.conf` со следующим содержимым:

```
Debug sudo /var/log/sudo_debug.log all@debug
Debug sudoers.so /var/log/sudo_debug.log all@debug
```

На контроллере домена редактирование этого файла заблокировано подсистемой мандатного контроля, см. `sudo astra-mic-control status`.

В файле `sudo_debug.log` будет представлена информация о пользователе и среде окружения в момент запуска команды `sudo`:

```
sudo[22259] settings: debug_flags=all@debug
sudo[22259] settings: run_shell=true
sudo[22259] settings: progname=sudo
sudo[22259] settings: network_addrs=192.0.2.1/255.255.255.0 fe80::250:56ff:feb9:7d6/
ffff:ffff:ffff:ffff::
sudo[22259] user_info: user=user_name
sudo[22259] user_info: pid=22259
sudo[22259] user_info: ppid=22172
sudo[22259] user_info: pgid=22259
sudo[22259] user_info: tcpgid=22259
sudo[22259] user_info: sid=22172
sudo[22259] user_info: uid=10000
sudo[22259] user_info: euid=0
sudo[22259] user_info: gid=554801393
sudo[22259] user_info: egid=554801393
```

```

sudo[22259] user_info: groups=498,6004,6005,7001,106501,554800513,554801107,554801108,5
54801393,554801503,554802131,554802244,554807670
sudo[22259] user_info: cwd=/
sudo[22259] user_info: tty=/dev/pts/1
sudo[22259] user_info: host=client
sudo[22259] user_info: lines=31
sudo[22259] user_info: cols=237

```

С помощью этой информации можно получить ответы на ряд вопросов.

Какой источник информации использовался для извлечения правил SUDO

```

sudo[22259] <- sudo_parseIn @ ./fileops.c:178 := sudoers: files sss

```

Со следующей строки включается в работу плагин SSSD.

```

udo[22259] <- sudo_sss_open @ ./sssd.c:305 := 0

```

Как много правил было получено от службы SSSD.

```

sudo[22259] Received 3 rule(s)

```

Подшли эти правила или нет.

```

sudo[22259] sssd/ldap sudoHost 'ALL' ... MATCH!
sudo[22259] <- user_in_group @ ./pwutil.c:1010 := false

```

5.3 Журнал отладки SSSD

Чтобы включить отладку SSSD в файле /etc/sss/sss.conf в секциях domain и sudo нужно установить параметр уровня отладки debug_level на значение 0x3ff0, что соответствует восьмому уровню, который содержит достаточно информации для решения большинства проблем и включает флаги SSSDBG_FATAL_FAILURE, SSSDBG_CRIT_FAILURE, SSSDBG_OP_FAILURE, SSSDBG_MINOR_FAILURE, SSSDBG_CONF_SETTINGS, SSSDBG_FUNC_DATA, SSSDBG_TRACE_FUNC, SSSDBG_TRACE_LIBS, SSSDBG_TRACE_INTERNAL.

```

[domain/domain_name]
debug_level = 0x3ff0
...
[sudo]
debug_level = 0x3ff0

```

После внесения изменений для того, чтобы настройки вступили в силу, нужно выполнить перезапуск службы.

```
# systemctl restart sssd
```

При использовании утилиты sudo будет создан файл журнала /var/log/sss/sudo_domain_name.log с помощью которого можно будет получить информацию по ряду вопросов.

Как много правил было получено от службы SSSD.

```
[sdap_sudo_refresh_load_done] (0x0400): Received 4-rules rules
```

Какие правила служба SSSD загрузила с сервера.

```
[sssdb[LDAP.PB]] [sysdb_save_sudorule] (0x0400): Adding sudo rule demo-name
```

Находились ли подошедшие правила в кеше.

```
[sdap_sudo_refresh_load_done] (0x0400): Sudoers is successfully stored in cache
```

Какой фильтр был использован для загрузки правил с сервера.

```
[sdap_get_generic_ext_step] (0x0400): calling ldap_search_ext with
[(&(objectClass=sudoRole)(|(! (sudoHost=*)) (sudoHost=ALL) (sudoHost=client.example.com)
(sudoHost=client) (sudoHost=192.0.2.1) (sudoHost=192.0.2.0/24) (sudoHost=2620:52:0:224e:
21a:4aff:fe23:1394) (sudoHost=2620:52:0:224e::/64) (sudoHost=fe80::21a:4aff:fe23:1394)
(sudoHost=fe80::/64) (sudoHost=+*) (| (sudoHost=*\*) (sudoHost=*\?) (sudoHost=*\2A*)
(sudoHost=*\[*\*]))))] [dc=example,dc=com]
```

Используйте этот фильтр, чтобы выполнить поиск в базе LDAP каталога напрямую:

```
# ldapsearch -x -D "cn=Directory Manager" -W -H ldap://server.example.com -b
dc=example,dc=com '(&(objectClass=sudoRole)...)'
```

Собеседник (Responder) службы SSSD регистрирует свои события в файле журнала /var/log/sss/sudo.log, с помощью которого можно ответить на следующие вопросы.

Как много правил было получено от службы SSSD.

```
[sssdb[sudo]] [sudsrv_get_sudorules_from_cache] (0x0400): Returning 4-rules rules for
[user@idm.example.com]
```

Какой фильтр был применен при поиске кеша SSSD.

```
[sudsrv_get_sudorules_query_cache] (0x0200): Searching sysdb with
[(&(objectClass=sudoRule)(| (sudoUser=ALL) (sudoUser=user) (sudoUser=#10001)
(sudoUser=%group-1) (sudoUser=%user) (sudoUser=+*)))]
```

Для поиска извлечения правил из кеша используйте команду `ldbsearch` из состава пакета `ldb-tools`:

```
# ldbsearch -H /var/lib/sss/db/cache_domain_name.ldb -b cn=sysdb  
'(&(objectClass=sudoRule)...)'
```


6 Лучшие практики

6.1 Работа с локальными настройками sudo

Предполагается, что в файле `sudoers` должны быть заданы правила для локальных пользователей, а в LDAP-каталоге, соответственно, для управления привилегиями доменных пользователей, но использование двух источников одновременно может привести к очень неприятной коллизии.

Администратор с ограниченными правами, которому были делегированы полномочия только на управление объектами отдельного структурного подразделения, может создать в этом подразделении группу с именем `sudo` или `astra-admin`, включить себя в состав одной из этих групп и получить привилегии суперпользователя на всех компьютерах в домене, включая сервера, так как в файле `/etc/sudoers` на этих машинах по умолчанию содержатся соответствующие правила.

Чтобы избежать указанной проблемы воспользуйтесь одним из следующих способов:

1. Заранее создайте на портале управления ALD Pro группы с именами `astra-admin` и `sudo`, чтобы администраторы с ограниченными правами не смогли создать такие группы в вверенных им подразделениях. Список зарезервированных имен можно расширить с учетом того, какие операционные системы используются в домене и какие на них настройки в файле `sudoers`.
2. Удалите источник `files` для базы `sudoers` из файла `/etc/nsswitch`. В этом случае при вызове утилиты `sudo` настройки из локального файла учитываться не будут, и пользователи групп `sudo` и `astra-admin` потеряют возможность повышать свои привилегии.

Для повышения безопасности по умолчанию с версии 2.0.0 слова `sudo` и `astra-admin` будут включены в список системных имен.

6.2 Будьте осторожны с символами подстановки

Символ «звездочки» в правилах SUDO следует использовать крайне осторожно, так как ошибки в его использовании могут привести к предоставлению несанкционированного доступа.

Допустим, администратору нужно было предоставить сотруднику доступ на чтение журналов `messages` и он создал следующее правило:

```
localuser      ALL=(ALL:ALL) NOPASSWD :      /usr/bin/cat /var/log/messages*
```

На первый взгляд все правильно, и пользователь сможет получить доступ к журналам:

```
ocaluser@astra:~$ cat /var/log/messages
localuser@astra:~$ cat /var/log/messages.1
```

Но утилита `cat` называется так от слова **concatenate** (сцеплять), и на самом деле она позволяет объединять в один поток содержимое сразу нескольких файлов, поэтому никто не помешает пользователю добавить к журналу `messages` содержимое файла `shadow`, чтобы увидеть пароли:

```
localuser@astra:~$ cat /var/log/messages /etc/shadow
...
```

```
localadmin:$gost12512hash$JQmInL3jM2ni7vs/$qVDRaInXXpPDgQW1/
e26C7bAvRaMrwizV924KN4YYXDgPnYDlWqvpETfk29S9q7LKlxZe07qA/.0cC02XG3U/:19296:0:99999:7
:::
localuser:$gost12512hash$0EbYsS/
b0DT9ux.t$0CY2yXvZTdZ3L03cCfD7KI61DQiu0Z6bHEvzn3YXWZLj0.vcNU6pQQEz/
hhWXHmuVCQbMHFWtL.YmTdoctUZq.:19518:0:99999:7:::
...
```

Конкретно в этом случае для предотвращения нежелательного поведения утилиты sudo в шаблон следует добавить еще одну команду, которая будет запрещать вызов команды cat с пробелами в параметре:

```
localuser      ALL=(ALL:ALL) NOPASSWD :      /usr/bin/cat /var/log/messages*, !/
usr/bin/cat /var/log/messages* *
```

6.3 Не разрешайте использование редактора vi

Работая в приложении vi, пользователь может не только редактировать текст, но и запускать команды оболочки, что дает значительные преимущества. Например, если в процессе редактирования файла конфигурации потребуется ввести точный путь к какому-то сертификату, пользователь сможет выполнить команду :shell, чтобы провалиться в оболочку и стандартными командами cd и ls уточнить необходимую информацию, а затем командой exit вернуться к редактированию файла.

Вместе с тем, такая реализация утилиты делает крайне опасным использование этого редактора вместе с правилами SUDO. Допустим, администратору нужно было предоставить сотруднику право на редактирование файла ldap.conf и он создал следующее правило:

```
localuser      ALL=(ALL:ALL) NOPASSWD :      /usr/bin/vi /etc/ldap/ldap.conf
```

На первый взгляд все правильно, и пользователь сможет получить право редактировать файл от имени суперпользователя. Но при этом ему ничто не мешает запустить из редактора оболочку и прочитать содержимое файла shadow

```
localuser@dc-1:~$ sudo vi /etc/ldap/ldap.conf
...
# File modified by ipa-client-install
# We do not want to break your existing configuration, hence:
#   URI, BASE, TLS_CACERT and SASL_MECH
:shell
...
root@dc-1:/home/localuser# cat /etc/shadow
...
localadmin:$gost12512hash$JQmInL3jM2ni7vs/$qVDRaInXXpPDgQW1/
e26C7bAvRaMrwizV924KN4YYXDgPnYDlWqvpETfk29S9q7LKlxZe07qA/.0cC02XG3U/:19296:0:99999:7
:::
localuser:$gost12512hash$0EbYsS/
b0DT9ux.t$0CY2yXvZTdZ3L03cCfD7KI61DQiu0Z6bHEvzn3YXWZLj0.vcNU6pQQEz/
hhWXHmuVCQbMHFWtL.YmTdoctUZq.:19518:0:99999:7:::
```

6.4 Использовать группы пользователей, оставлять комментарии

Довольно простой рекомендацией является отказ от назначения прав на конкретных пользователей – используйте вместо этого группы. В этом случае и список правил будет короче, и за списками участников групп обычно удастся лучше следить.

6.5 Принцип предоставления минимальных прав

При настройке правил SUDO следует предоставлять доступ только к тем командам, которые необходимы сотрудникам для выполнения должностных обязанностей. Чтобы избежать наличие излишних привилегий у пользователей крайне важно регулярно проводить аудит и отзывать те разрешения, которые более не требуются.