

# Инструкция по обеспечению безопасной работы в домене ALD Pro: правила HBAC

ALD Pro

Exported on 08/11/2023

## Table of Contents

1 Что такое HBAC-правила .....	4
2 Механизм работы HBAC-правил .....	5
3 Доступ для администраторов ко всем компьютерам в домене, ограничение правила allow_all .....	9
4 Доступ для сотрудников на рабочие станции, создание правила allow_computers .....	11
5 Гранулированный доступ к отдельным службам и отладка правил .....	15
6 Проверка HBAC-правил командой hbactest .....	21
7 Лучшие практики: ограничение доступа локальным пользователям.....	22
8 Лучшие практики: создание HBAC-правил для структурных подразделений .....	23

## Ограничение доступа к компьютерам в домене с помощью HBAC-правил

Исполнитель [Руслан Федоров ЕХТ](#)<sup>1</sup>

v 1.1

см. [\[проект\] Инструкция по обеспечению безопасной работы в домене ALD Pro](#)<sup>2</sup>

Когда сотруднику выдают доменную учетную запись, у него появляется возможность зайти с ее помощью на любой хост в домене, включая сервера, что при неправильной настройке зон ответственности при администрировании к объектам файловой системы создает угрозу несанкционированного доступа к информации, хранящейся на этих компьютерах.

Для повышения безопасности работы в домене администраторам следует ограничить доступ к компьютерам, что можно сделать с помощью правил управления доступом к хостам (host-based access control, HBAC).

---

<sup>1</sup> <https://life.astralinux.ru/display/~ruslan.fedorov>

<sup>2</sup> <https://life.astralinux.ru/pages/viewpage.action?pageId=176032084>

## 1 Что такое НВАС-правила

Правила НВАС создают дополнительный слой авторизации, позволяя разрешить определенным **пользователям** использовать указанные **службы** на конкретных **хостах**.

Правила являются разрешающими, для каждого из трех субъектов безопасности следует определить область одним из двух способов:

- Любой субъект — правило будет распространяться на все субъекты и группы субъектов данного вида.
- Указанные субъекты — правило будет распространяться только на указанный перечень субъектов и групп субъектов данного вида.

Что такое «пользователи» и «хосты» обычно вопросов не вызывает, но вот понятие «служб» требует отдельного пояснения.

## 2 Механизм работы NBAC-правил

Службами в контексте NBAC являются любые приложения, которые используют PAM-стек для авторизации пользователей, при этом не важно, являются ли эти приложения обычными исполняемыми файлами или работают в фоне.

Стек подключаемых модулей аутентификации (Pluggable Authentication Modules, PAM) - это система библиотек, обеспечивающая унифицированный программный интерфейс (Application Programming Interface, API) для абстрагирования приложений (таких как [login](https://linux.die.net/man/1/login)<sup>3</sup> и [s<sup>4</sup>udo](https://linux.die.net/man/1/su)) от выполнения стандартных задач аутентификации, причем, настройки аутентификации администраторы могут задавать для каждого приложения индивидуально с помощью файлов из директории `/etc/pam.d/*`

Библиотека PAM делит задачи аутентификации на четыре независимые группы управления, которые отвечают за различные аспекты пользовательских запросов:

- **account** – группа модулей, которые отвечают за проверку аккаунта, не истек ли пароль, разрешен ли пользователю доступ к запрашиваемому сервису.
- **authentication** – группа модулей, которые отвечают за получение учетных данных пользователя и выполнение аутентификации. Чаще всего они реализуют какой-то диалог с пользователем для получения данных, но также возможны и способы аутентификации с использованием аппаратных ключей, биометрических устройств и пользовательских сертификатов.
- **password** – группа модулей, которые отвечают за проверку пароля на соответствие требованиям безопасности по длине, стойкость к перебору, наличию часто запрещенных слов.
- **session** – группа модулей, которые отвечают за выполнение задач до и после предоставления услуги, например, запись событий в журналы, монтирование домашнего каталога.

За работу NBAC-правил отвечает модуль [pam\\_sss.so](http://pam.sss.so)<sup>5</sup>, механизм отражен на рисунке 1.

---

<sup>3</sup> <https://linux.die.net/man/1/login>

<sup>4</sup> <https://linux.die.net/man/1/su>

<sup>5</sup> <http://pam.sss.so>

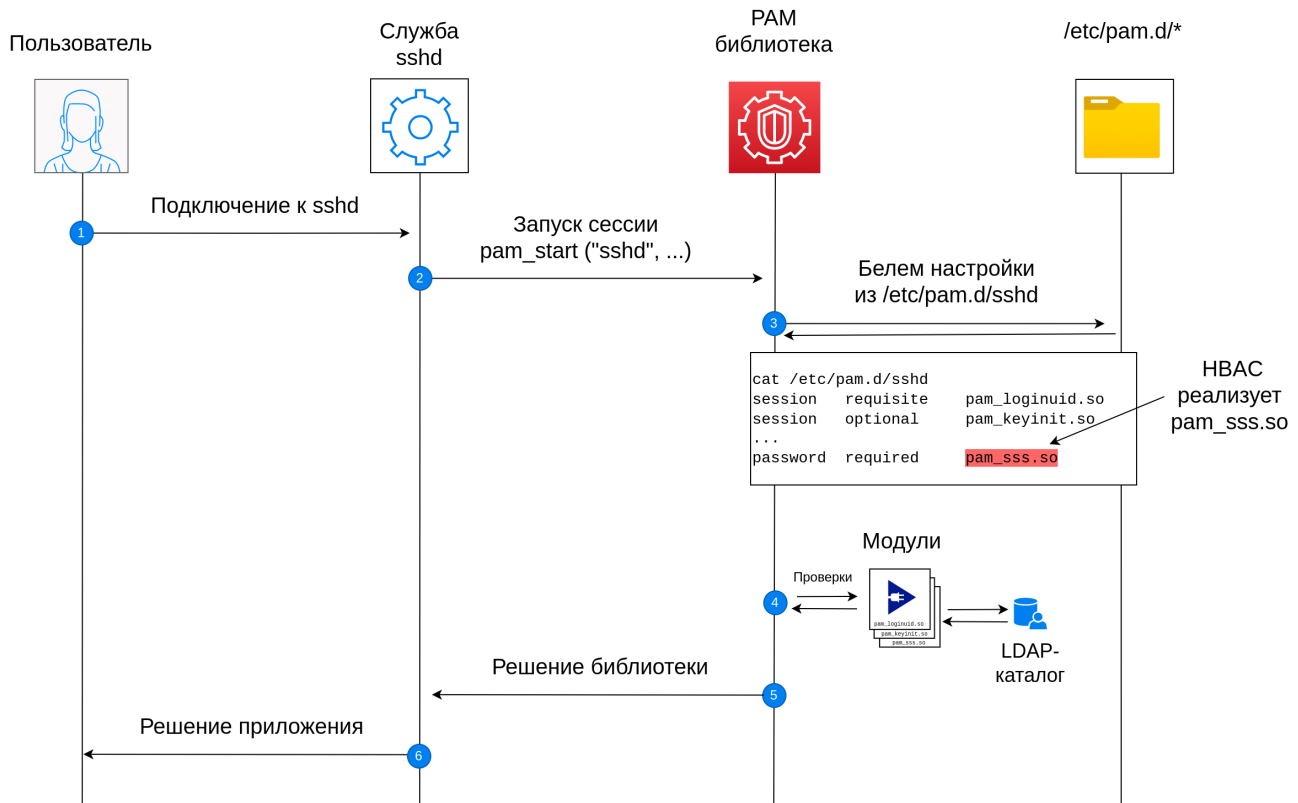


Рис. 1. Механизм работы HBAC-правил

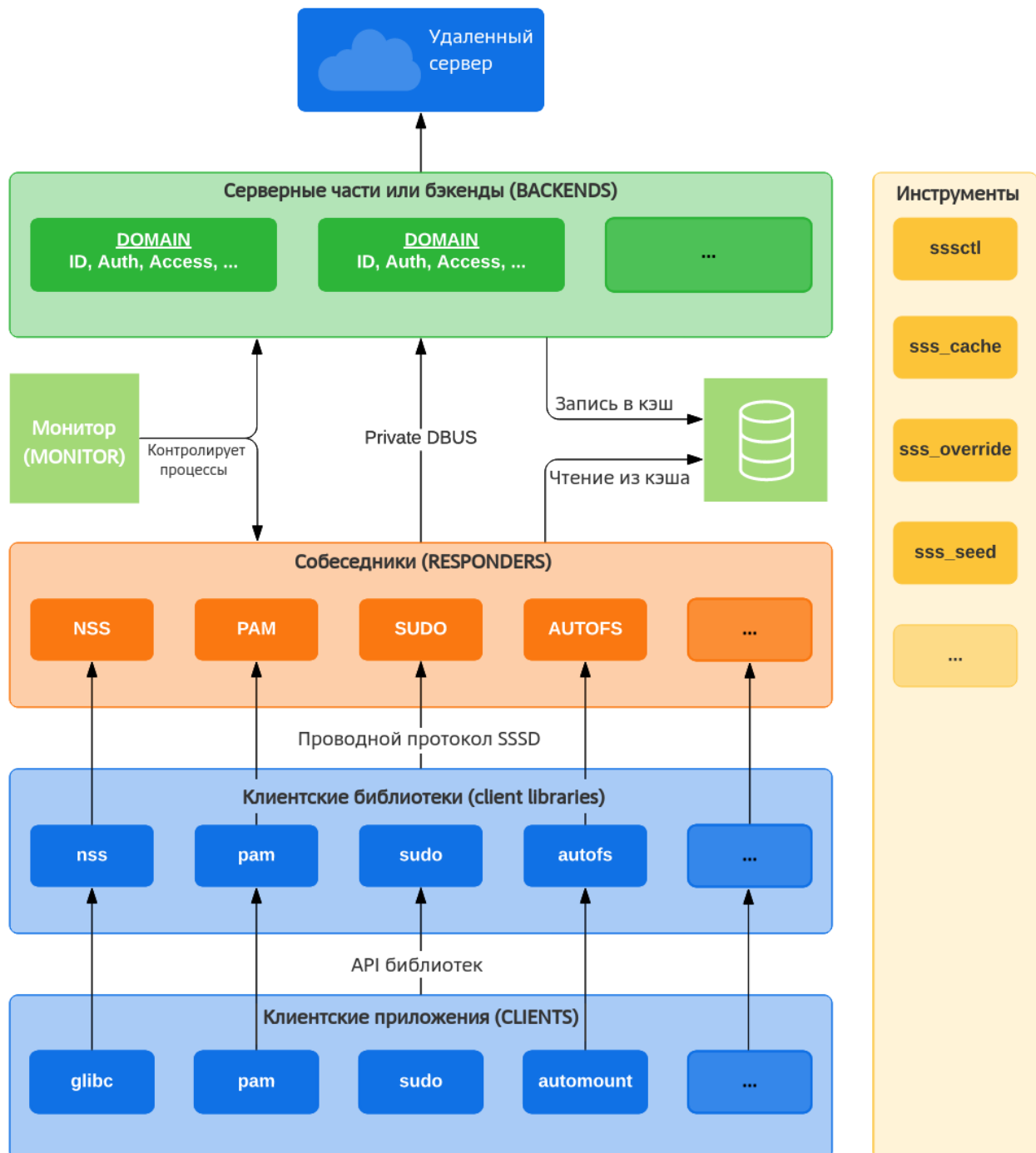
При подключении пользователя по SSH (1) служба обращается к PAM библиотеке, чтобы создать контекст безопасности для выполнения команд от его имени (2). При вызове функции `pam_start` служба передает библиотеке свой идентификатор (PAM service name), который представляет из себя обычную строку, например `sshd`. Идентификатор службы обычно совпадает с именем исполняемого файла, но это не обязательно. Некоторые приложения могут использовать несколько идентификаторов, например, утилита `sudo` использует дополнительный идентификатор «`sudo-i`», а модуль `mod_authnz_pam` веб-сервера Apache2 так вообще позволяет проверять доступ к каждому разделу сайта с помощью отдельного идентификатора. Обратите также внимание, что в разных дистрибутивах Linux одни и те же службы могут использовать разные идентификаторы, например, `ssh` и `sshd`.

Значение идентификатора службы определяет имя файла, откуда библиотека PAM будет брать настройки стека модулей (3), для службы `sshd` настройки будут браться из файла `/etc/pam.d/sshd`. В конфигурационных файлах PAM-стека перечисляются необходимые модули и параметры их вызова. Для того, чтобы упростить управление PAM-стеком, конфигурационные файлы допускают использование инструкций `@include`, которые позволяют включить содержимое других файлов.

Получив необходимые настройки, библиотека PAM выполняет проверки, используя указанные модули (4), причем, за работу HBAC-правил отвечает модуль `pam_sss.so`<sup>6</sup>. Итоговый ответ библиотека передает приложению (5), которое, в свою очередь, использует эту информацию для организации взаимодействия с пользователем (6).

<sup>6</sup> [http://pam\\_sss.so](http://pam_sss.so)

Модуль `pam_sss.so`<sup>7</sup> является клиентской библиотекой службы `sss`, основная задача которой заключается в получении информации от службы каталога и локальном кешировании данных для ускорения обработки запросов. Учитывая необходимость кеширования, архитектура службы предусматривает наличие серверной части (backend), собеседника (responder) и локального кеша между ними, см. рисунок 2.



<sup>7</sup> [http://pam\\_sss.so](http://pam_sss.so)

## Рисунок 2. Архитектура службы SSSD

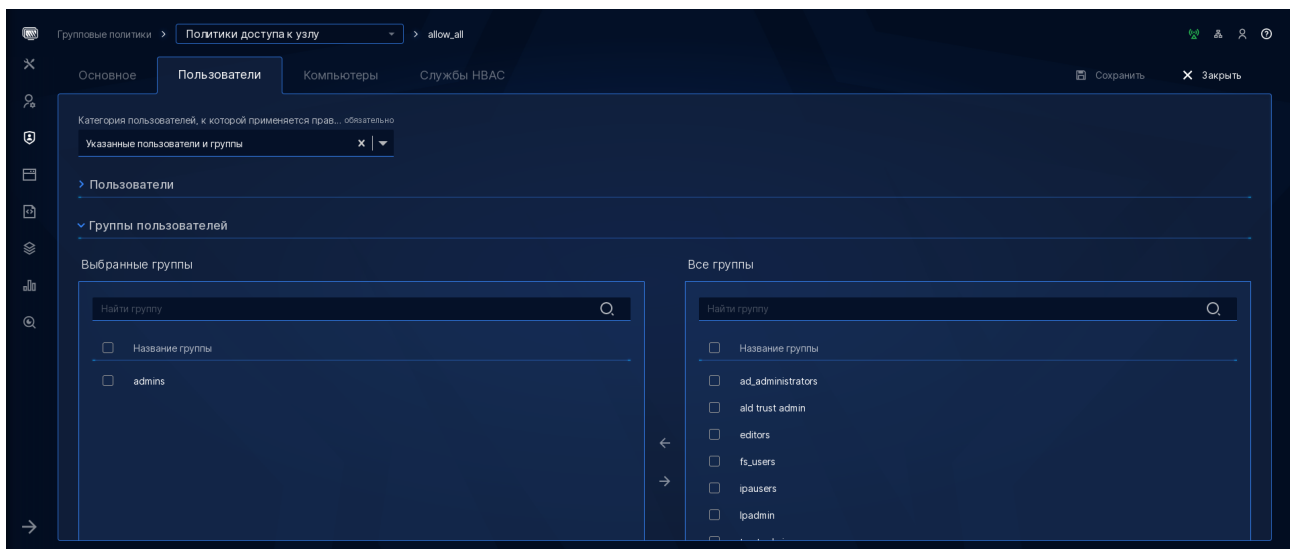
Для вступления в силу изменений в настройках HBAC-правил вы можете воспользоваться на целевой машине командами «`sss_cache -E`» или «`sssctl cache-remove`», чтобы очистить локальный кеш службы sssd. Утилиты входят в состав пакета sssd-tools, который нужно устанавливать дополнительно.



### 3 Доступ для администраторов ко всем компьютерам в домене, ограничение правила allow\_all

Правила HBAC являются «разрешающими», т. е. «по умолчанию» доступ к службам на доменных компьютерах запрещен, и его нужно открывать с помощью правил. Однако, при развертывании домена автоматически создается правило allow\_all, которое разрешает доступ «всех ко всему», поэтому для управления авторизацией на уровне HBAC вам нужно сначала ограничить область применения этого правила, например, только группой администраторов.

Внести указанные настройки можно через веб-портал на странице «Групповые политики > Политики доступа к узлу > allow\_all > Пользователи»



или из командной строки:

```
# ipa hbacrule-mod allow_all --usercat=''
# ipa hbacrule-mod allow_all --desc='Разрешает администраторам доступ к любому хосту в домене'
# ipa hbacrule-add-user allow_all --group admins
# ipa hbacrule-show allow_all
```

где

- hbacrule-mod — команда, с помощью которой можно модифицировать настройки существующей группы
  - allow\_all — имя правила, которые мы хотим модифицировать
  - usercat — ключ, который позволяет изменить категорию для области применения в части пользователей. Может принимать два возможных значения — 'all' и пустая строка ''

- `hbacrule-add-user` — команда, с помощью которой можно расширить область применения правила в части пользователей.
  - `allow_all` — имя правила, которое мы хотим модифицировать
  - `group` — ключ, который позволяет добавить группу пользователей в область применения HBAC-правила
  - `admins` — имя группы, которая будет добавлена в область применения правила
- `hbacrule-show` — команда, с помощью которой можно получить информацию о существующем HBAC-правиле
  - `allow_all` — имя правила, по которому мы хотим получить информацию

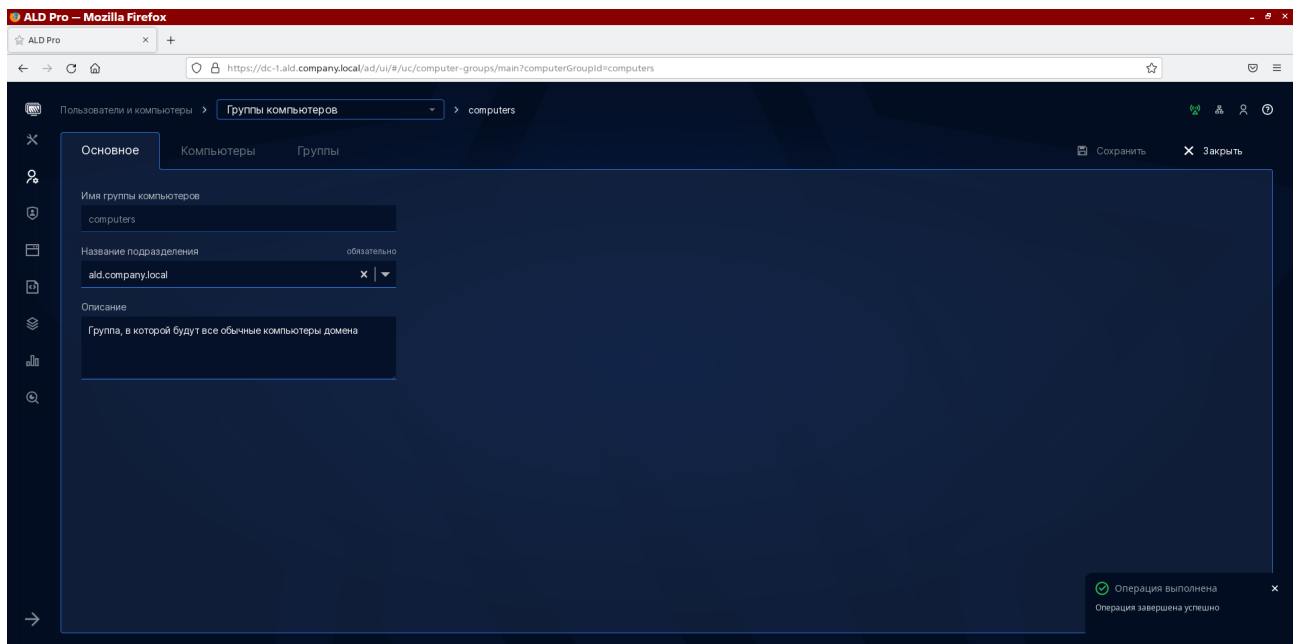
**Примечание.** Если из-за неправильной настройки HBAC-правил доступ к хостам будет все же заблокирован, вы сможете подключиться к portalу управления с любого другого компьютера, который не находится в домене, чтобы исправить ошибку. Доступ к portalу управления не регулируется через механизм HBAC.

## 4 Доступ для сотрудников на рабочие станции, создание правила allow\_computers

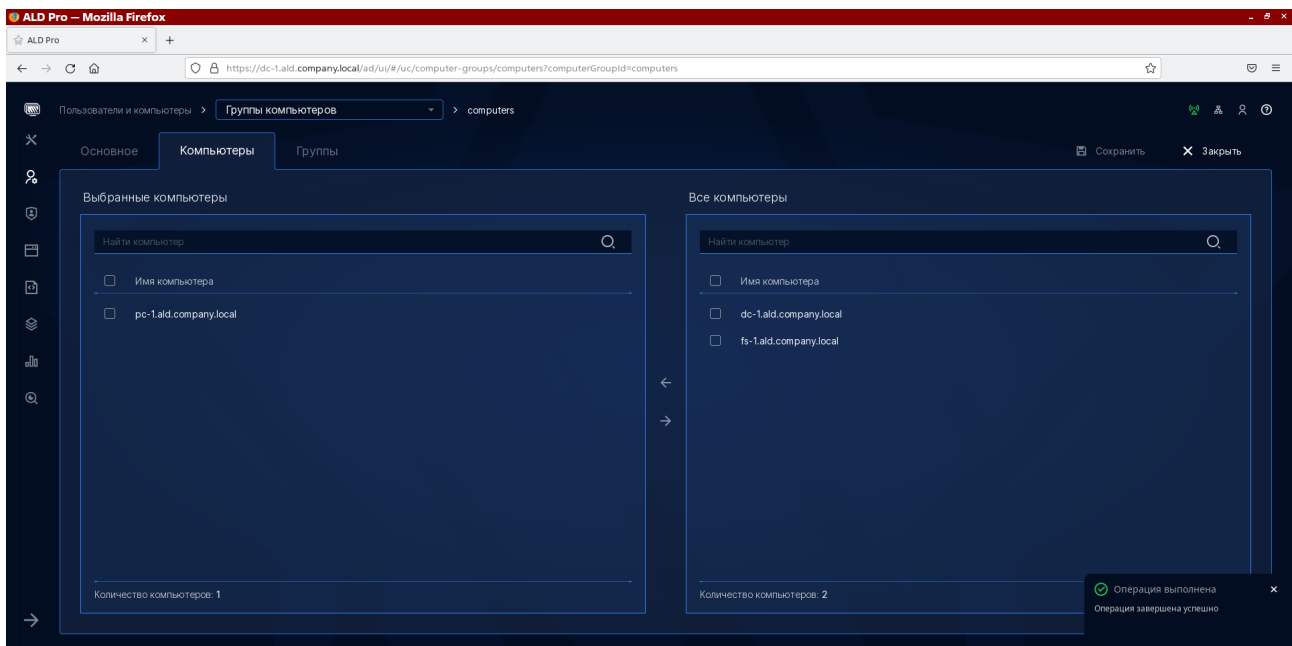
Если ограничить область действия правила allow\_all группой администраторов, то для остальных сотрудников компании нужно будет создать правило allow\_computers, которое предоставит им право входа на обычные компьютеры в домене.

Создадим это правило с использованием веб-интерфейса:

1. Создайте группу хостов **computers**. Откройте страницу «Пользователи и компьютеры > Группы компьютеров» и нажмите кнопку «Новая группа». Введите имя группы, ее описание и нажмите кнопку «Сохранить».

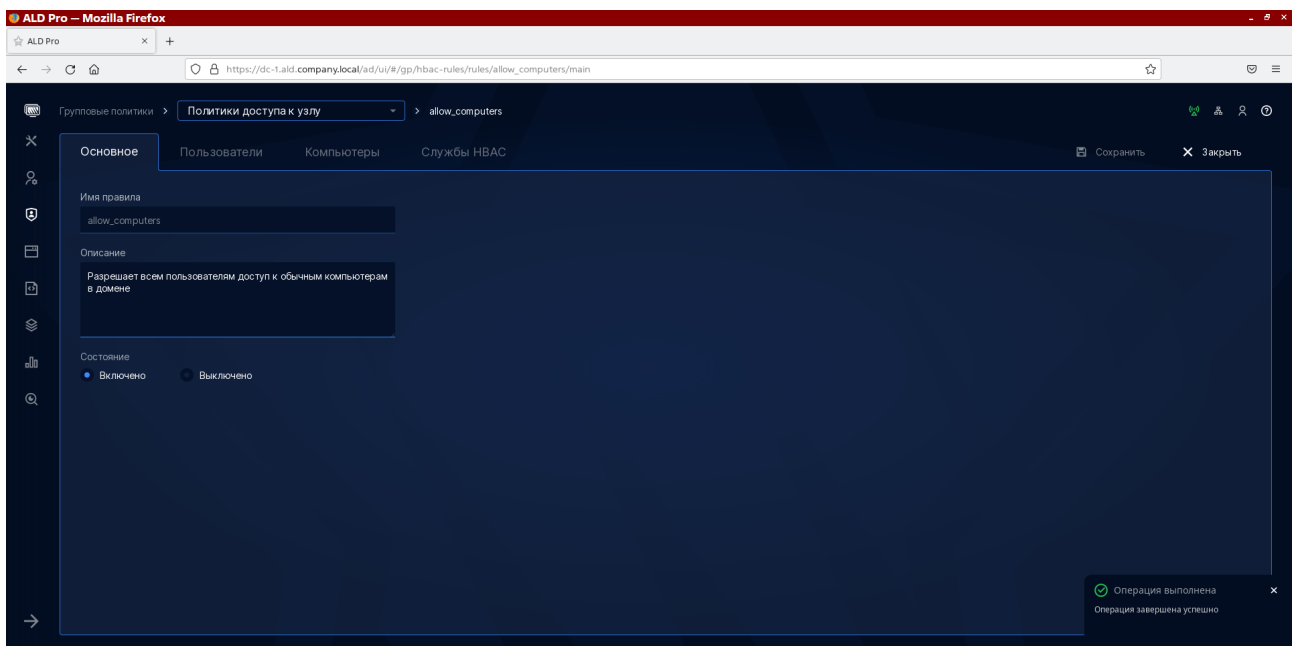


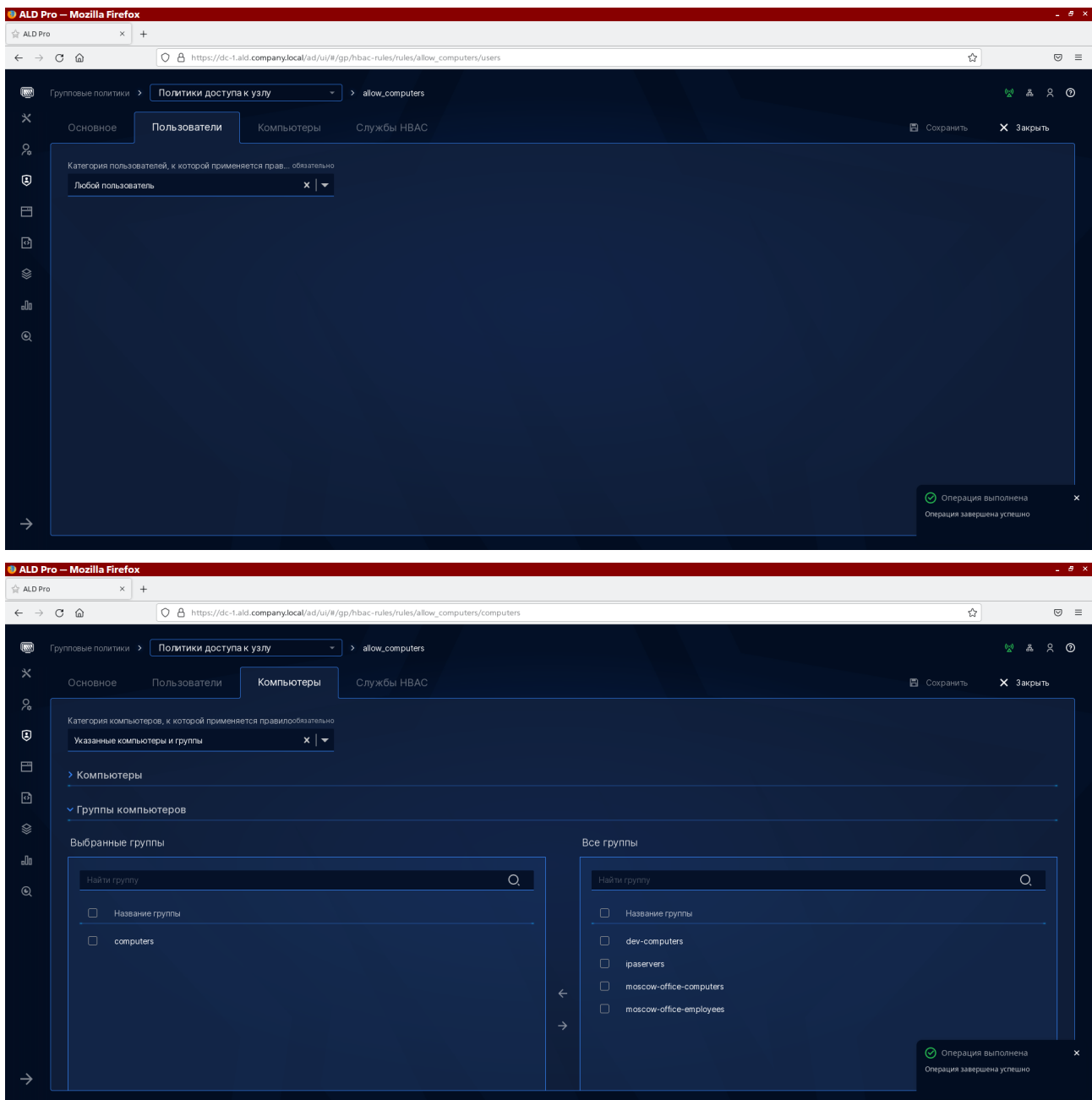
2. На вкладке «Компьютеры» внесите рабочие станции в список участников группы.

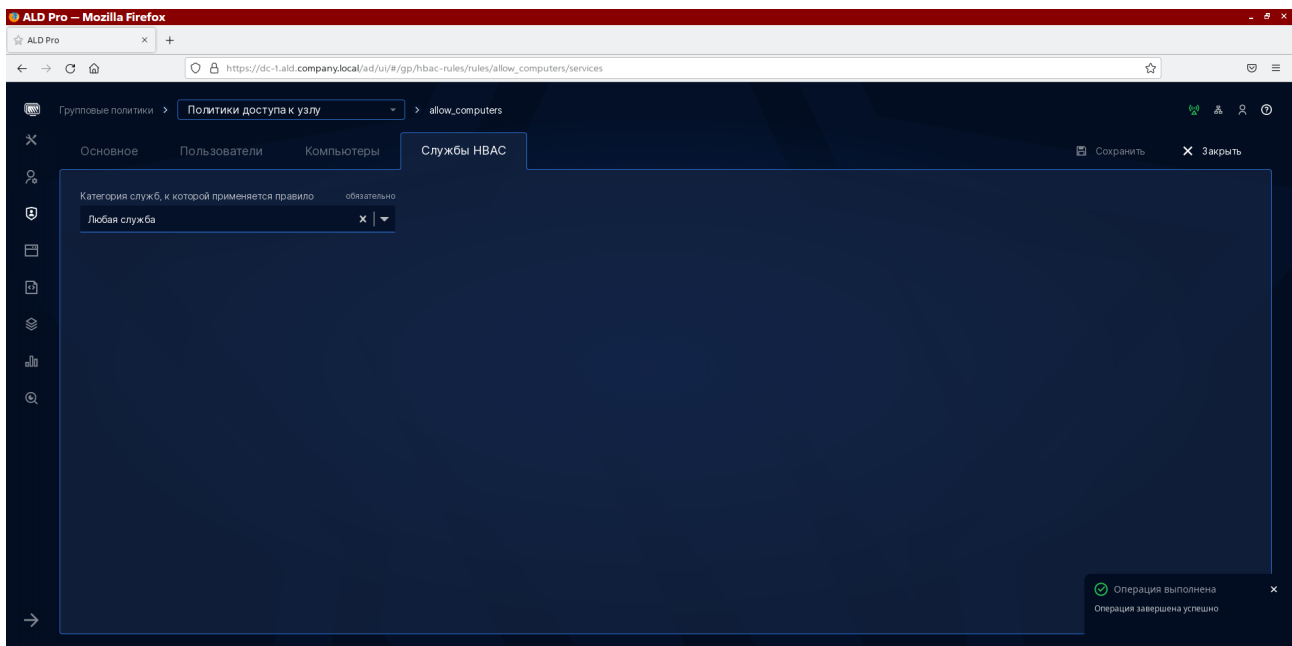


3. Создайте HBAC-правило. Откройте страницу «Групповые политики > Политики доступа к узлу > Правила HBAC» и нажмите кнопку «Новое правило». Введите имя правила **allow\_computers** и сохраните его. Для созданного правила определите следующую область действия:

- пользователи — любой пользователь
- хосты — указанные компьютеры и группы, группа computers
- службы — любая служба







Сделаем тоже самое из командной строки:

```
ipa hostgroup-add computers
ipa hostgroup-mod computers --desc='Группа, в которой будут все обычные компьютеры домена'
ipa hostgroup-add-member computers --hosts pc-1

ipa hbacrule-add allow_computers
ipa hbacrule-mod allow_computers --desc='Разрешает всем пользователям доступ к обычным компьютерам в домене'
ipa hbacrule-mod allow_computers --usercat=all
ipa hbacrule-mod allow_computers --servicecat=all
ipa hbacrule-add-host allow_computers --hostgroup computers
```

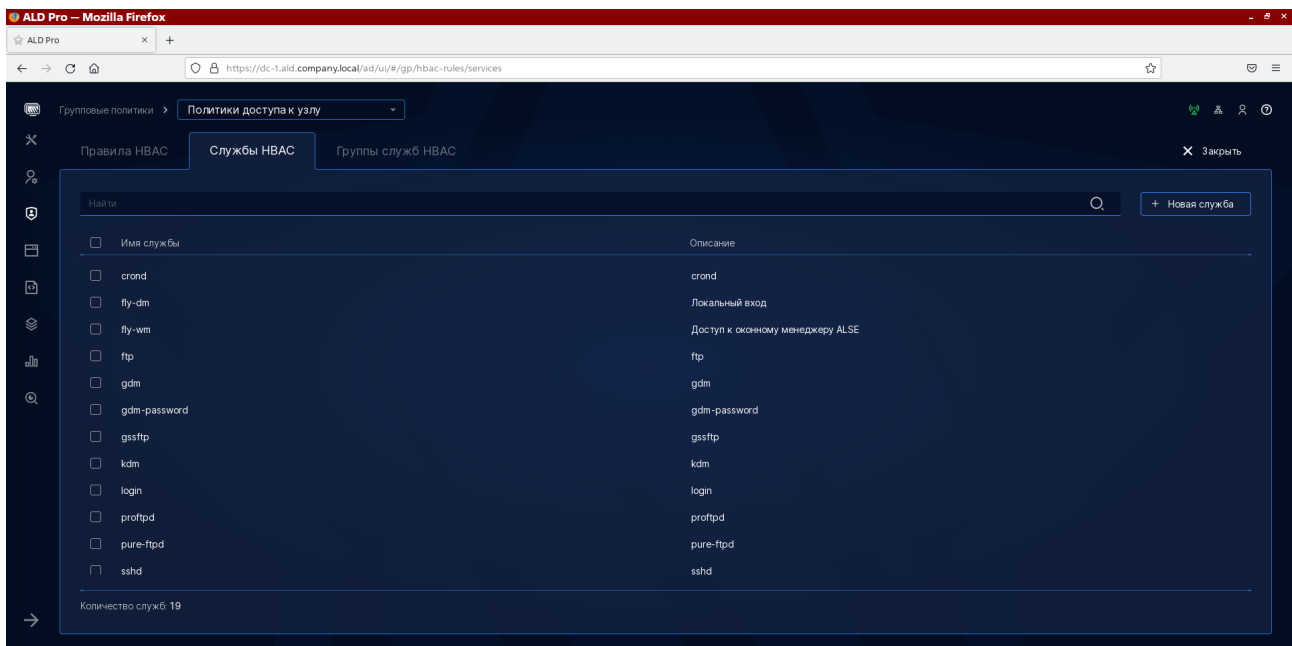
## 5 Гранулированный доступ к отдельным службам и отладка правил

Для тонкой настройки HBAC вам потребуется тщательно анализировать типовые сценарии работы пользователей в части используемых служб. Например, для подключения по RDP потребуются fly-wm и xrdp-sesman, см. таблицу 1.

**Таблица 1.** Службы, необходимые для типовых сценариев работы

Сценарий работы пользователя	К каким идентификаторам служб следует предоставить доступ	Комментарий
Локальный вход в операционную систему	fly-dm, fly-wm	fly-dm - разрешает вход, fly-wm - нужен, чтобы можно было разблокировать экран
Удаленный доступ к менеджеру окон по протоколу RDP	xrdp-sesman, fly-wm	xrdp-sesman - разрешает вход по rdp fly-wm - нужен, чтобы можно было разблокировать экран
Удаленное администрирование по SSH	sshd, sudo	sshd - разрешает подключение по ssh sudo - разрешает повышать привилегии в соответствии с правилами SUDO

После установки системы в домене уже есть список наиболее распространенных служб, но какие-то службы вам все равно потребуется добавить вручную. Сделать это можно будет через веб-интерфейс на странице «Групповые политики > Политики доступа к узлу > Службы HBAC». Давайте создадим «fly-dm», «fly-wm» и «xrdp-sesman».



То же самое можно сделать из командной строки:

```
# ipa hbacsvc-add 'fly-dm'
# ipa hbacsvc-mod 'fly-dm' --desc='Локальный вход ALSE'
# ipa hbacsvc-add 'fly-wm'
# ipa hbacsvc-mod 'fly-wm' --desc='Доступ к оконному менеджеру ALSE'
# ipa hbacsvc-add 'xrdp-sesman'
# ipa hbacsvc-mod 'xrdp-sesman' --desc='Доступ по RDP'
```

Чтобы понять, к какой службе требуется предоставить доступ, выполните необходимое действие на целевой системе и посмотрите, какие сообщения об ошибках появятся в журнале авторизации auth.log:

```
# tail -f varlog/auth.log
...
Mar 15 15:25:22 client 4 sshd[30424]: pam_sss(sshd:account): Access denied for user
ivanov: 6 (Permission denied)
...
```

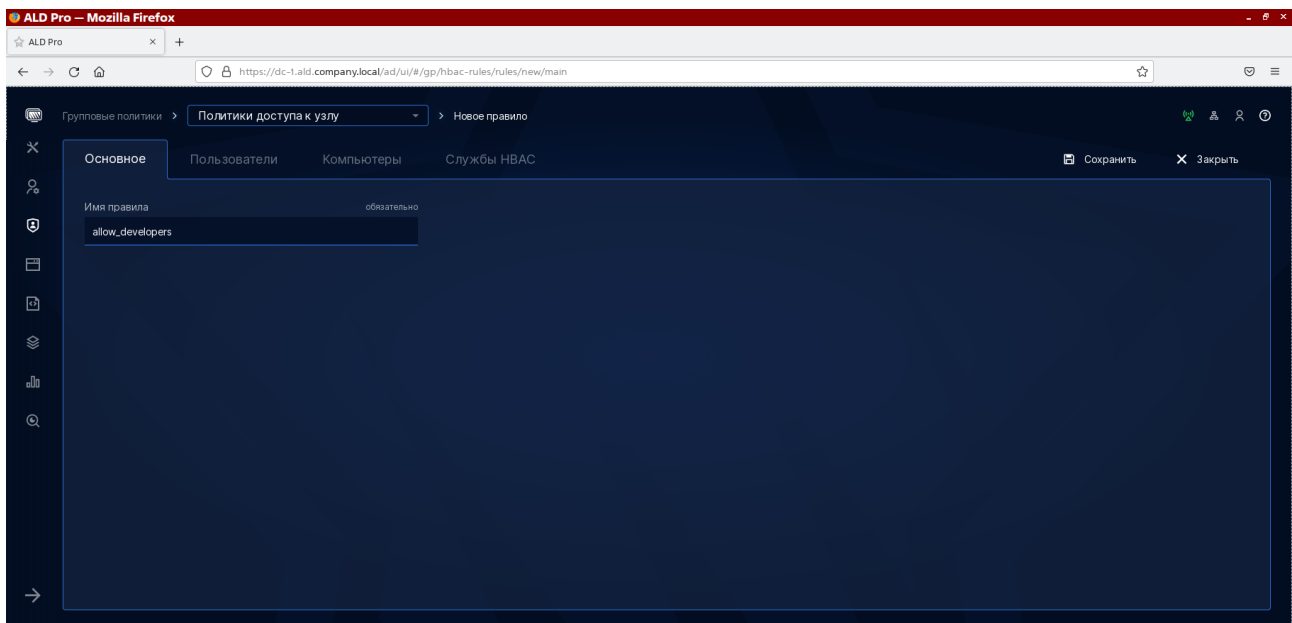


Допустим, нам нужно предоставить возможность разработчикам из группы **dev-users** на своих компьютерах из группы **dev-computers** только входить в операционную систему, а далее уже расширять возможности при администрировании до root с помощью утилиты **su**. Создать соответствующее HBAC-правило можно как через портал управления ALD Pro, так и из командной строки.

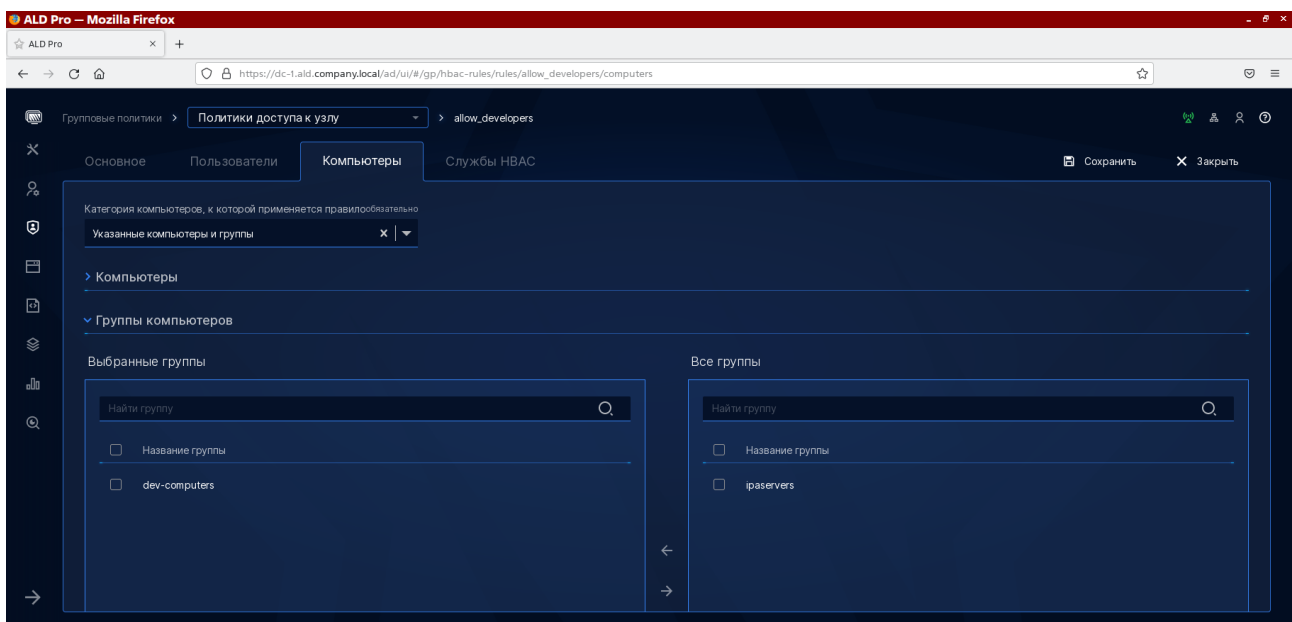
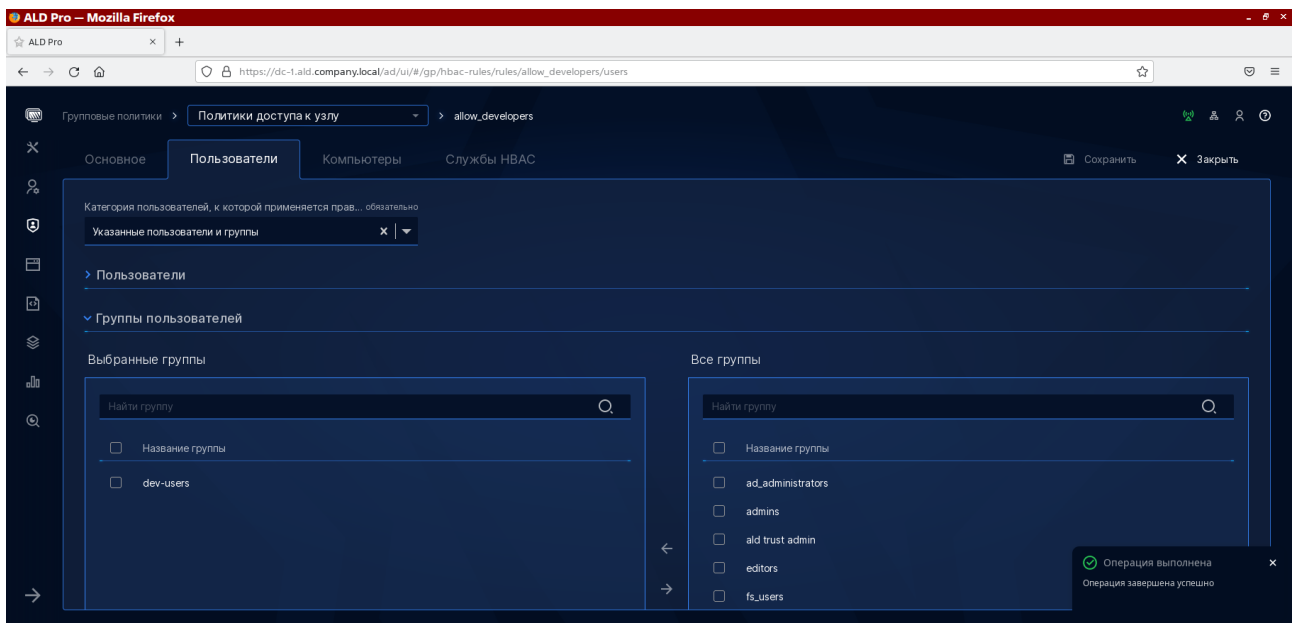
С использованием веб-интерфейса делается это следующим образом:

1. Создайте HBAC-правило. Для этого откройте страницу «Групповые политики > Политики доступа к узлу > Правила HBAC» и нажмите кнопку «Новое правило».

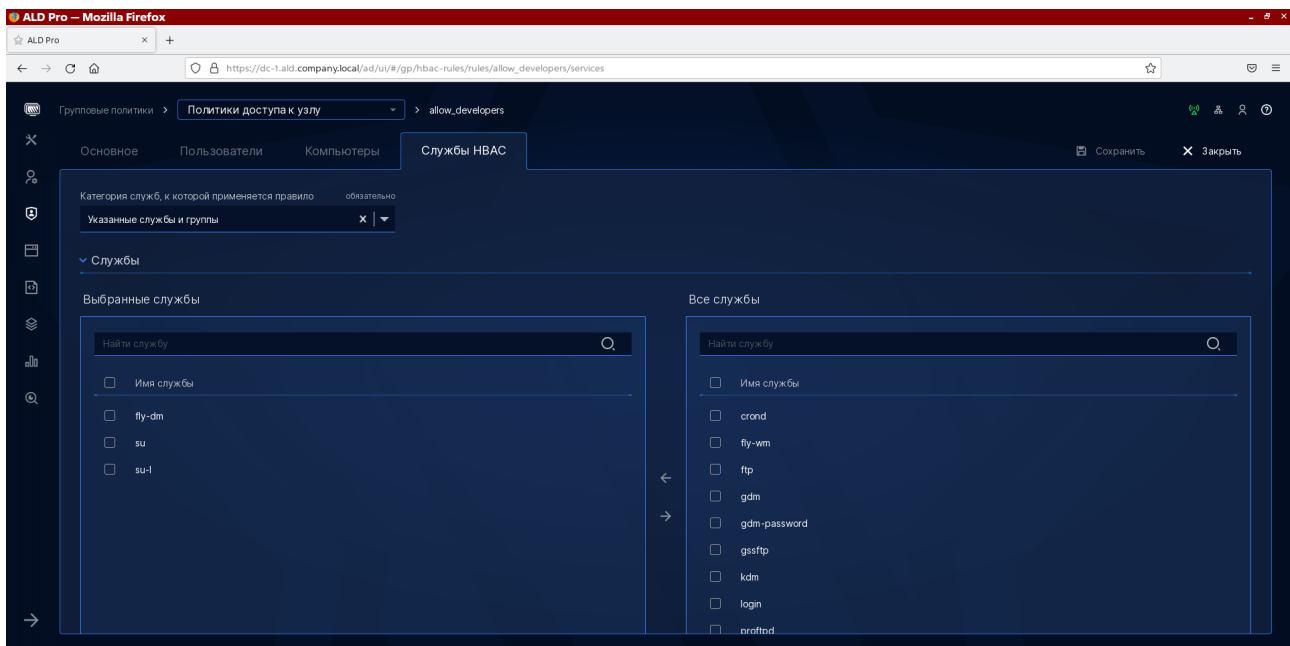
Введите имя правила **allow\_developers** и нажмите кнопку «Сохранить». Пока вы не сохраните правило, остальные вкладки с настройками будут недоступны.



2. Настройте область применения правила в части пользователей, выберите категорию «Указанные пользователи и группы» и добавьте группу **dev-users**. В части компьютеров добавьте группу **dev-computers**. Не забудьте нажать кнопку «Сохранить» в правом верхнем углу прежде чем переходить к следующей вкладке.



3. Настройте область применения правила в части служб, добавьте «fly-dm», «su» и «su-l» (используется утилитой su при вызове с параметром «-», «-l» или «--login»).



Из командной строки такое правило можно создать следующим образом:

```
# kinit admin
# ipa hbacrule-add allow_developers
# ipa hbacrule-mod allow_developers --desc='Доступ для разработчиков'
# ipa hbacrule-add-user allow_developers --groups dev-users
# ipa hbacrule-add-host allow_developers --hostgroups dev-computers
# ipa hbacrule-add-service allow_developers --hbacsvcs fly-dm
# ipa hbacrule-add-service allow_developers --hbacsvcs su
# ipa hbacrule-add-service allow_developers --hbacsvcs su-l
```

где

- kinit admin — аутентификация в системе под учетной записью admin
- hbacrule-add — команда для создания HBAC-правила
- hbacrule-mod — команда для модификация правила, ключ desc позволяет задать описание
- команды hbacrule-add-user, hbacrule-add-host и hbacrule-add-service позволяют определить область применения правила
- ключ groups — позволяет указать группу пользователей
- ключ hostgroups — позволяет указать группу хостов
- ключ hbacsvcs — позволяет указать идентификатор РАМ службы

Следует напомнить, что сразу после создания правила оно может заработать на целевой машине не сразу из-за кеширования sssd.

Теперь, чтобы пользователь смог воспользоваться утилитой su на личном компьютере, вам нужно передать ему пароль от учетной записи root.

Скорее всего, у пользователя root нет пароля и в файле /etc/shadow- в том месте, где должен быть указан хэш пароля вы увидите восклицательный знак. Откройте терминал на целевой системе и установите пользователю root пароль следующим образом:

```
admin@pc-1:~$ sudo -i
root@pc-1:~# passwd root
Новый пароль : *****
Повторите ввод нового пароля : *****
passwd: пароль успешно обновлён
```

Теперь пользователь может проверить, что у него есть доступ к компьютеру через графику и он может с помощью команды su запустить bash от имени root.

## 6 Проверка HBAC-правил командой hbactest

Когда у вас в домене два-три правила, их довольно легко проверить напрямую подключаясь к целевым хостам под тестовыми учетными записями, но в реальной инфраструктуре потребуются управлять десятками правил, и упростить их отладку поможет команда `ipa hbactest`. Команду можно выполнить на контроллере домена и для любого сочетания пользователь-хост-сервис получить ответ, есть ли в домене правило, которое соответствует этим критериям.

Выполним проверку, сможет ли пользователь `ivanov` воспользоваться службой `sshd` при подключении к хосту `client4` по протоколу `ssh`:

```
ipa hbactest --user=ivanov --host=client4 --service=sshd
-----
Доступ предоставлен: False
-----
Несоответствующие правила: 1
Несоответствующие правила: allow_systemd-user
```

Решение `False` означает, что пользователю будет отказано в доступе.

Выполним проверку конкретного правила `allow_developers` с помощью ключа `--rules`, сможет ли пользователь `ivanov` выполнить вход на компьютер `client4`, для чего ему нужна служба `fly-dm`

```
ipa hbactest --user=ivanov --host=client4 --service=fly-dm --rules allow_developers
-----
Доступ предоставлен: True
-----
Соответствующие правила: allow_developers
```

Решение `True` означает, что пользователю будет предоставлен доступ в соответствии с правилом `allow_developers`, значит пользователь `ivanov` входит в группу пользователей `dev-users`, а хост `client4` в группу хостов `dev-computers`.

## 7 Лучшие практики: ограничение доступа локальным пользователям

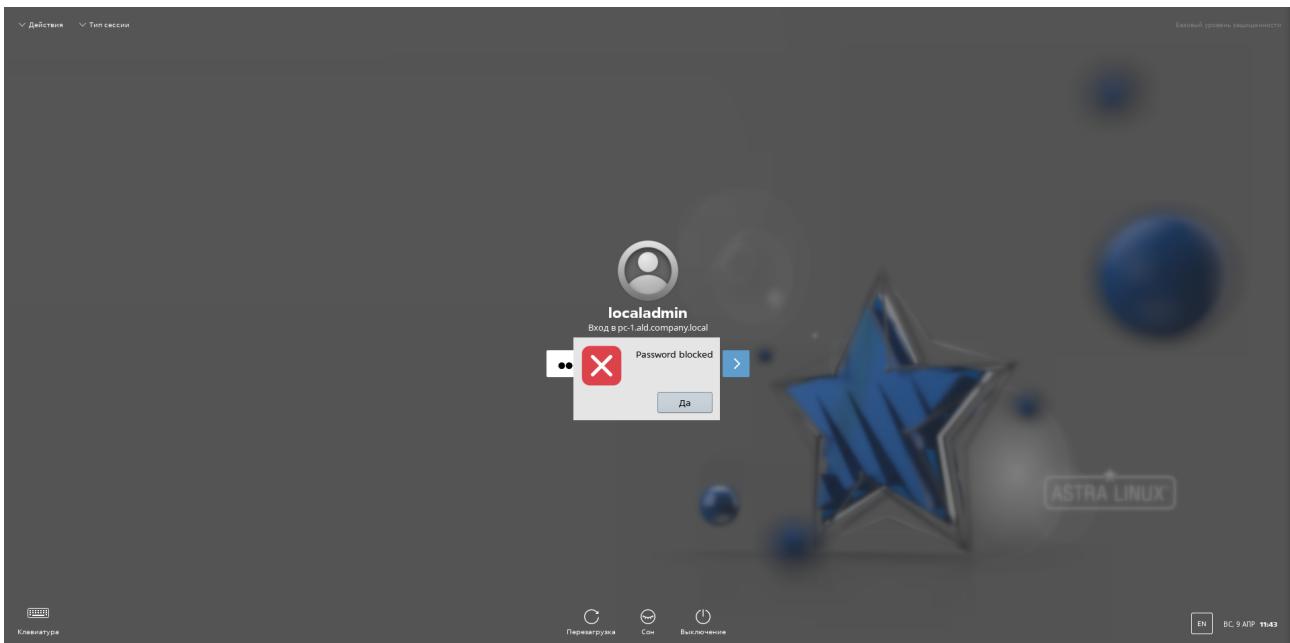
Вопрос управления локальными учетными записями на компьютерах домена является одним из важнейших аспектов безопасности, который требует повышенного внимания со стороны системных администраторов.

Есть разные подходы к организации управления учетными записями локальных администраторов в домене, например, вы можете их полностью отключить, организовать управление через скрипты групповых политик или даже разработать стороннее решение, реализующее функции, аналогичные программному продукту LAPS от Microsoft (Local admin password solution).

В рамках данной статьи рассмотрим самый простой из них — это полное блокирование локальных учетных записей. Для того, чтобы заблокировать локальную учетную запись localadmin после ввода компьютера в домен вам достаточно будет выполнить команду:

```
# passwd -l localadmin
```

Теперь вы можете убедиться, что войти в систему с помощью этой учетной записи и убедиться, что доступ будет запрещен.



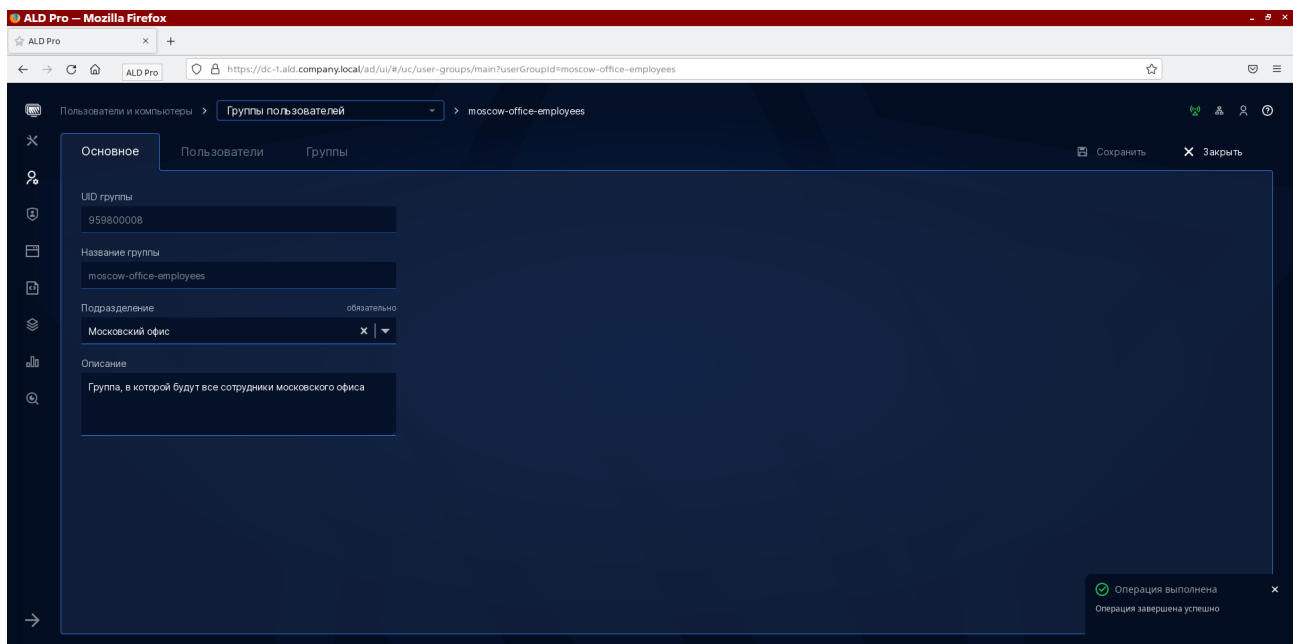
## 8 Лучшие практики: создание НВАС-правил для структурных подразделений

Область применения НВАС-правил можно задать с помощью групп пользователей и групп хостов, но в некоторых случаях может быть удобнее использовать для этого структурные подразделения. В этом случае вы можете воспользоваться вспомогательными группами и правилами автоучастия (automember).

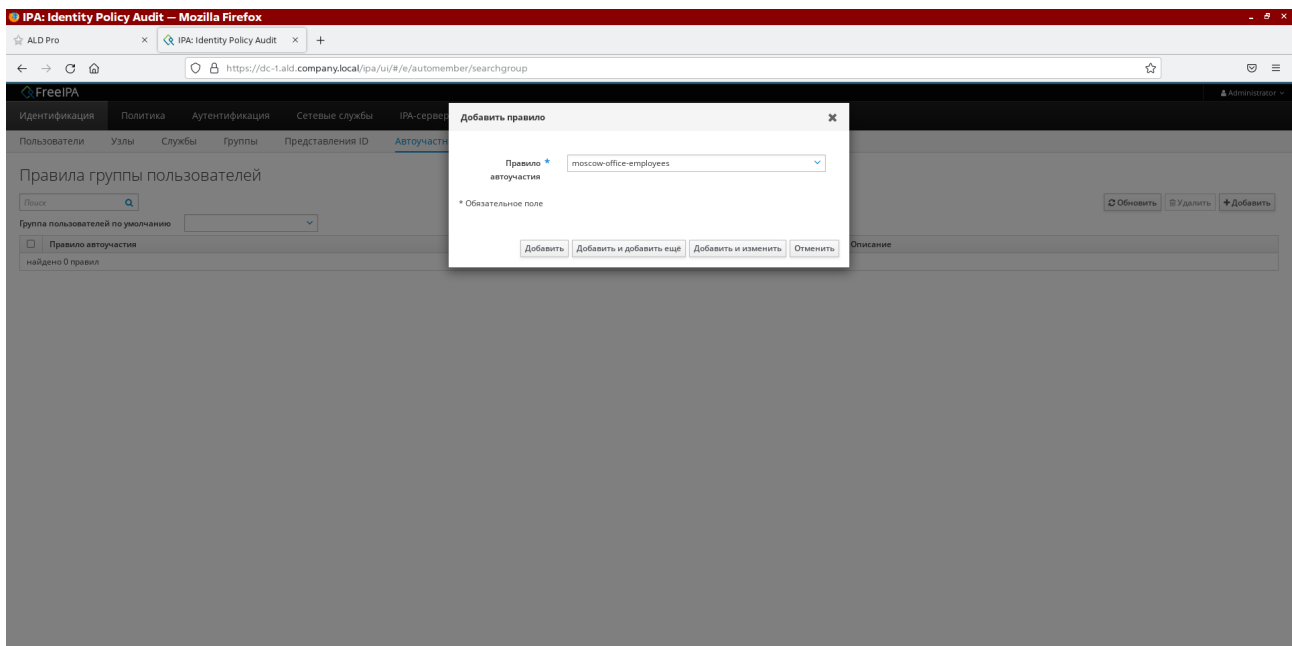
Привязка объектов к структурным подразделениям в ALD Pro осуществляется с помощью атрибута `rbtdap`, в котором хранится ссылка на целевое подразделение в формате полного уникального имени записи (Distinguished name, DN). Например, если у вас в корне домена есть структурное подразделение «Московский офис», то значение атрибута `rbtdap` всех его дочерних объектов будет содержать «`ou=Московский офис,ou=ald.company.local,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=local`». Если вам потребуется ограничить выборку только прямыми наследниками, вы можете поставить символ подстановки «`^`» в начало строки, что позволит вам исключить объекты из подразделений, расположенных ниже по иерархии организационной структуры.

Создадим группу пользователей и правило автоучастия из веб-интерфейса:

1. Создайте группу пользователей, для этого на странице «Пользователи и компьютеры > Группы пользователей» нажмите кнопку «Новая группа», введите название группы «`moscow-office-employees`», выберите подразделение «Московский офис» и нажмите кнопку «Сохранить».



2. Создайте правило автоучастия, для этого потребуется воспользоваться интерфейсом FreeIPA. Откройте страницу «Идентификация > Автоучастник > Правила группы пользователей», нажмите кнопку «Добавить». В диалоговом окне добавления правила выберите группу «`moscow-office-employees`» из списка и нажмите кнопку «Добавить и изменить».



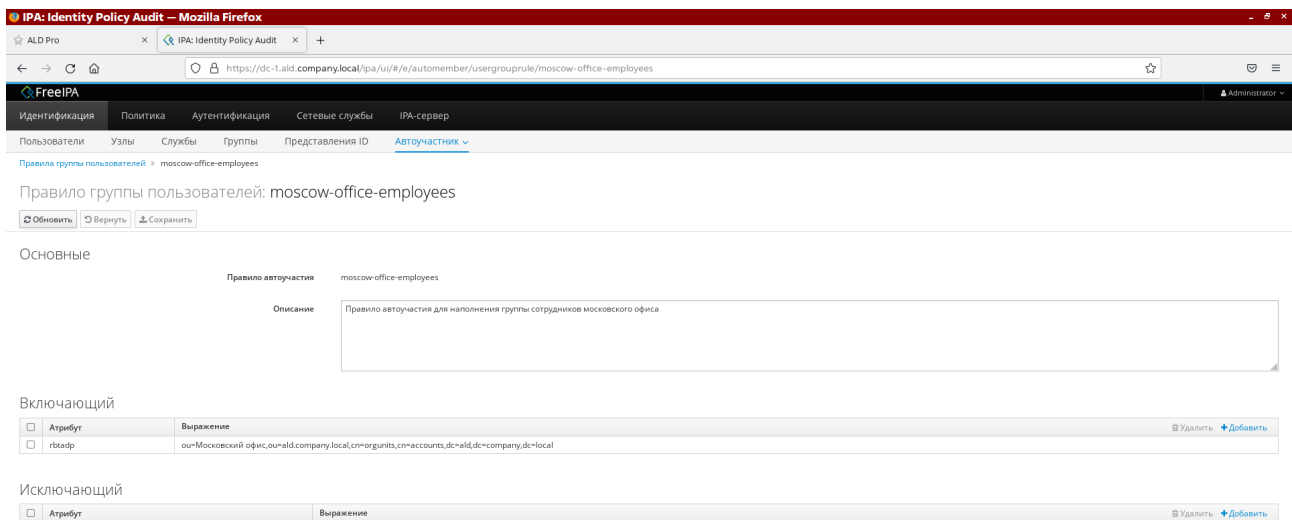
3. На странице редактирования правила добавьте включающий критерий отбора записей по атрибуту `rbtadp` и значению «`ou=Московский офис,ou=ald.company.local,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=local`».

Пересчет правил автоучастия происходит не мгновенно, но вы можете ускорить применение этих правил с помощью команды `automember-rebuild`:

```
ipa automember-rebuild --type=group
```

Если пользователи так и не появятся в списке участников группы, проверьте корректность фильтра. Обор записей выполняется по полному вхождению, поэтому никаких лишних пробелов в середине строки быть не должно.





То же самое вы можете сделать их командной строки:

1. Создайте группу пользователей moscow-office-employees в подразделении «Московский офис»:

```
ipa group-add moscow-office-employees
ipa group-mod moscow-office-employees setattr="rbtadp=ou=Московский
офис,ou=ald.company.local, cn=orgunits,cn=accounts,dc=ald,dc=company,dc=local"
ipa group-mod moscow-office-employees --desc='Группа, в которой будут все сотрудники
московского офиса'
```

2. Создайте правило автоучастия и определите критерий автоучастия:

```
ipa automember-add moscow-office-employees --type=group
ipa automember-mod moscow-office-employees --type=group --desc='Правило автоучастия для
наполнения группы сотрудников московского офиса'
ipa automember-add-condition moscow-office-employees --type=group --key=rbtadp --
inclusive-regex='ou=Московский офис,ou=ald.company.local,
cn=orgunits,cn=accounts,dc=ald,dc=company,dc=local'
```

3. Принудительно обновим состав автоучастников (запускать команду нужно только 1 раз, в дальнейшем правила будут срабатывать автоматически при изменении подразделения объекта):

```
ipa automember-rebuild --type=group
```