ALD Pro

РУКОВОДСТВО АДМИНИСТРАТОРА

РДЦП.10101-01 95 01
Версия: 2.4.0

Содержание

Общие сведения

1.1. Требования к квалификации администратора

Инструкция предназначена для администратора, обладающего знаниями и опытом в следующих областях:

- Администрирование Linux основной AL-1702
- Администрирование Linux расширенный курс AL-1703
- Администрирование компьютерных сетей (материалы курса AL-1704)

Желательно, чтобы администратор обладал базовыми навыками администрирования LDAP-каталога и использования протокола аутентификации **Kerberos**.

1.2. Функциональное назначение

«ALD Pro» (Astra Linux Directory Pro) - это набор сетевых служб сервера Astra Linux для организации централизованного управления ИТ-инфраструктурой. Система построена на хорошо известных компонентах с открытым исходным кодом, использующих стандартные протоколы обмена информацией: FreeIPA, 389 Directory Server, MIT Kerberos, Bind9, ISC DHCP, NTP, SSSD, CUPS, Samba, Zabbix, Syslog-NG, RabbitMQ, SaltStack.

В основе лежит служба каталога, которая обеспечивает централизованное хранение данных о пользователях, хостах, сервисных учетных записях и обеспечивает доступ к этой информации по открытым протоколам. Управление системой можно осуществлять из веб-интерфейса, командной строки или через REST API.

Система «ALD Pro» расширяет возможности по администрированию серверов и рабочих станций следующими функциями:

- централизованное управление учетными записями пользователей и компьютеров, организационной структурой предприятия, конфигурациями доменных компьютеров с помощью применения групповых политик, включая установку и удаление программного обеспечения;

- синхронизация времени на всех компьютерах предприятия;
- автоматизированная установка операционных систем по локальной сети;
- управление печатью и общим доступом к файлам в компьютерной сети предприятия;
- мониторинг серверной группировки и журналирование наиболее важных событий.

1.3. Технологический стек

Система ALD Pro использует клиент-серверную архитектуру и состоит из следующих компонентов:

- Серверные подсистемы устанавливаются на выделенный или виртуальный сервер:
  - Контролер Домена (Портал Управления, FreeIPA, 389 Directory Server, MIT Kerberos, Bind9, Samba, Chrony, SSSD, SaltStack, RabbitMQ)
  - Подсистема Печати (CUPS)
  - Подсистема Общий Доступ к Файлам (Samba)
  - Подсистема Мониторинг (Zabbix)
  - Подсистема Журналирования (Сбора Логов) (Syslog-NG)
  - Подсистема Репозитория ПО (Reprepro)
  - Подсистема Установка ОС по Сети (TFTP + PXE)
  - Подсистема Динамической Настройки Узла (ISC DHCP)
- Клиентская часть — реализована в виде агентов, которые устанавливаются на все компьютеры домена:
  - aldpro-client — клиентская часть ALD Pro, через которую устанавливаются и настраиваются остальные клиентские приложения и сервисы;
  - astra-freeipa-client — утилита от Astra Linux для внесения дополнительных настроек в конфигурацию sssd и других служб в соответствии с требованиями операционной системы;
  - freeipa-client — клиентская часть подсистемы FreeIPA, настраивает службу SSSD;
  - krb5-user — поддержка работы с Kerberos, утилиты kinit и др.;
  - sssd — набор служб для работы машины в домене;
  - sssd-ldap — LDAP бэкенд службы SSSD;

- ldap-utils — утилиты ldapsearch и др.;
- freeipa-admintools — набор утилит администрирования ipa CLI;
- aldpro-salt-minion — клиентская часть групповых политик;
- zabbix-agent — клиентская часть подсистемы мониторинга;
- syslog-ng — клиентская часть подсистемы журналирования;

Для обеспечения базовой функциональности ALD Pro необходимо установить один контроллер домена, однако, рекомендуется устанавливать не менее двух контроллеров для обеспечения отказоустойчивости и возможности обслуживать систему без прерывания доступа к сервисам аутентификации и авторизации. Минимальную конфигурацию оборудования для развёртывания подсистем, см. табл. 1.

Компонент	CPU Ядер	RAM	SSD	Кол-во
Контроллер Домена до 3 тыс. пользователей	4	8 ГБ	>= 50 ГБ	2 шт.
Дополнительные ресурсы на 10 тыс. пользователей	10	0.5 ГБ	0.5 ГБ	
Подсистема Мониторинга	2	2 ГБ	>= 30 ГБ	1 шт
Подсистема «Динамическая Настройка Узла» (DHCP)	2	2 ГБ	>= 30 ГБ	1 шт
Подсистема «Репозитории ПО»	2	2 ГБ	>= 100 ГБ	1 шт
Подсистема «Установка ОС по сети» (PXE)	2	2 ГБ	>= 30 ГБ	1 шт
Подсистема «Служба Печати»	2	2 ГБ	>= 30 ГБ	1 шт
Подсистема «Общий Доступ к Файлам»	2	2 ГБ	>= 30 ГБ	1 шт

Таблица 1 — Минимальные характеристики для установки подсистем ALD Pro

Внимание: Не рекомендуется устанавливать в одной операционной системе сразу несколько подсистем ALD Pro, т.к. это может привести к некорректной работе продукта и нежелательной конкуренции за ресурсы CPU и RAM.

1.4. Планирование ресурсов

От работы доменных служб зависит надежность работы всей ИТ инфраструктуры, поэтому при планировании домена администраторы должны действовать проактивно и применять различные способы вертикального и горизонтального масштабирования, чтобы исключить проблемы недостаточной производительности. Ниже представлены рекомендации, которые помогут в решении указанной задачи.

1.4.1. Вертикальное масштабирование

Вертикальным масштабированием называют повышение производительности системы за счет повышения производительности отдельного узла путем выделения серверу дополнительных вычислительных ресурсов — оперативной памяти, потоков центрального процессора и т.п.

1.4.1.1. Объем памяти

В работе службы каталога преобладают операции чтения, поэтому в производительности контроллера решающую роль играет достаточный объем оперативной памяти, чтобы контроллер мог обрабатывать запросы без обращения к медленным дискам.

Файлы каталога расположены в папке */var/lib/dirsrv*, после развертывания первого контроллера размер базы составляет не менее 45 Мб и увеличивается по мере создания объектов, в среднем по 50 КБ на каждую дополнительную учетную запись пользователя. Таким образом, объем памяти диска рассчитывается: для нужд самой операционной системы контроллеру следует выделить не менее 50 ГБ, а под хранение каталога для упрощения расчетов выделяется по 1 ГБ на каждые 20 тысяч объектов. Расчетное значение должно составлять не более 40% от доступного пространства.

$$HDD,\ GB = 50 + N\ objects/20\,000$$

Внимание: Производительность файловой подсистемы влияет на скорость записи, поэтому рекомендуется использовать SSD или RAID с высокой производительностью чтения/записи. Скорость произвольного чтения у современных твердотельных дисков составляет порядка 50-100 тысяч операций в секунду (IOPS).

Ещё одним вариантом увеличения производительности LDAP может быть использование разных устройств для корня файловой системы (/) и каталога с БД (*/var/lib/dirsrv/*). Смотрите руководство администратора ОС, раздел монтирования дисков.

Для загрузки каталога требуется больше оперативной памяти, т.к. для ускорения операций поиска служба **ns-slapd** индексирует данные каталога. Пустая база данных занимает в памяти не менее 65 Мб, и это значение растет по мере увеличения числа объектов, на каждую дополнительную учетную запись в среднем добавляется 50 КБ. Таким образом, минимально необходимый объем оперативной памяти можно рассчитать математически: для нужд самой операционной системы контроллеру следует выделить порядка 2-3 ГБ, а для работы с каталогом в целях упрощения расчетов возьмем по 1ГБ на каждые 20 тысяч объектов:

$$RAM,\ GB = 3 + N\ objects/20\ 000$$

Ниже приведено несколько примеров:

- Для работы с каталогом, в котором содержится 10 000 пользователей и 100 групп, контроллеру нужно выделить порядка 4 ГБ ОЗУ:

$$RAM,\ GB = 3 + 10\ 100\ objects/20\ 000 \approx 4\ GB\ RAM$$

- Для работы с каталогом, в котором 100 000 пользователей и 30 000 групп, контроллеру нужно выделить до 10 ГБ ОЗУ:

$$RAM,\ GB = 3 + 130\ 000\ objects/20\ 000 \approx 10\ GB\ RAM$$

Примечание: При создании учетных записей контроллер домена потребляет в два раза больше оперативной памяти, т.е. примерно 100 КБ на каждую новую запись, которая высвобождается после перезапуска службы **dirsrv**. Эта особенность работы не проявляется в сценарии репликации, но ее следует учитывать при планировании начальной миграции.

Несмотря на возможность работы с ограниченным объёмом оперативной памяти (см. пример расчёта выше) на контроллере домена, это может вызвать высокую интенсивность чтения данных с диска, что приведёт к общему замедлению системы и сервиса **LDAP**

(**ns-slapd**) в частности. Для обеспечения надёжной работы контроллера домена и сопутствующих сервисов, а также для исключения увеличения времени репликации из-за дополнительных операций чтения данных с диска, рекомендуется на контроллерах домена не уменьшать объём оперативной памяти ниже 8 ГБ.

Примечание: Рекомендуемый минимальный размер оперативной памяти одного контроллера домена — 8 ГБ.

1.4.1.2. Количество потоков центрального процессора

Если требуемый объем оперативной памяти зависит от размера базы данных, то количество ресурсов центрального процессора определяется тем, сколько пользователей должен обслуживать конкретный контроллер. Например, в базе может быть 100 тысяч пользователей, но нагрузка по их обслуживанию может распределяться между 20 репликами, тогда на каждый контроллер будет приходиться не более 5 тысяч обслуживаемых пользователей.

Приложения имеют разный сценарий взаимодействия с каталогом, что влияет на его нагрузку. Таким образом, у каждой организации получается свой уникальный профиль использования ресурсов этой службы. Поэтому администраторам следует рассчитать данный показатель для своей организации самостоятельно, достигая использования ЦПУ на уровне 40% от максимума в пиковые периоды, а отправной точкой может быть выделение одного потока для работы операционной системы и еще по одному потоку на каждую тысячу пользователей.

$$CPUs = (1 + N\ users/1\,000)$$

Ниже приведено несколько примеров:

- Для обслуживания 2-3 тысяч пользователей контроллеру домена нужно выделить не менее 4 потоков.

$$CPUs = (1 + 3\,000/1\,000) = 4$$

Примечание: Компания **RedHat** в своей документации рекомендует использовать контроллеры домена именно такой производительности, но эта рекомендация не подойдет

крупным организациям, в штате которых работает несколько сотен тысяч сотрудников, т.к. контроллер, обслуживающий 10 тысяч сотрудников, будет в три раза эффективнее использовать оперативную память в пересчете на активного пользователя, чем контроллер, обслуживающий 3 тысячи сотрудников.

- Для обслуживания 10 тысяч пользователей контроллеру домена нужно выделить 11 потоков.

$$CPUs = (1 + 1\ 0000/1\ 000) = 11$$

Следует отметить, что данные оценки сильно зависят от сценария использования службы каталога, поэтому на такое количество пользователей рекомендуется выделять порядка 8-12 потоков.

- Если под контроллер домена выделить физический сервер с двумя процессорами Xeon Silver (48 потоков), то теоретически он сможет обслуживать 47 тысяч пользователей

$$CPUs = (1 + 47\ 000/1\ 000) = 48$$

При планировании таких контроллеров следует принимать во внимание ограничения по пропускной способности сети. Для обслуживания такого числа пользователей потребуется использование сетевого интерфейса со скоростью передачи данных не менее 1 Гбит/с.

При выборе процессора следует учитывать, что LDAP-запросы в большей степени зависят от скорости дисковой подсистемы, а Kerberos-аутентификация использует сложные криптографические алгоритмы, поэтому в большей степени зависит от вычислительной производительности контроллера. Увеличивать вычислительные мощности контроллера можно как за счет добавления дополнительных ядер, так и за счет увеличения их частоты, при этом следует учитывать, что, если нагрузка на ядра не превышает 40%, то дополнительный прирост производительности можно получить только за счет увеличения частоты ядер.

- Не рекомендуется использовать процессоры с базовой тактовой частотой ниже 1,8 ГГц. Это может негативно сказаться на общей скорости работы контроллера домена, как части информационной системы.

Примечание: При изменении количества потоков процессора (замена процессора или изменение настроек виртуальной машины) необходимо обновить переменную

DAEMON_ARGS в файле */etc/default/krb5-kdc* и перезапустить службу **krb5-kdc**.

1.4.1.3. Пропускная способность сети

При планировании домена следует учитывать два основных вида трафика:

- трафик клиентских запросов (**Kerberos** и **LDAP** запросы к каталогу)
- трафик репликации (по стандартному **LDAPS** протоколу)

1.4.1.4. Клиентский трафик

В процессе обслуживания клиентов контроллеры домена, как правило, получают небольшие входящие запросы на предоставление относительно больших объемов данных, поэтому на контроллерах обычно преобладает исходящий трафик.

Любые оценки являются субъективными, однако, в качестве общих рекомендаций следует исходить из того, что при количестве пользователей до 5 000 будет достаточно интерфейса с пропускной способностью в 100 Мбит/с, а если предполагается обслуживать больше пользователей, нужно установить на контроллере интерфейс с пропускной способностью в 1 Гбит/с. В результате оптимизации нужно добиться, чтобы показатель использования сети не выходил за пределы 40% от максимума в пиковые периоды.

Учитывая то, что внешние каналы связи редко достигают гигабитных скоростей и активно используются пользователями для выхода в интернет, в каждом офисе рекомендуется создавать отдельный сайт и размещать в нем один-два контроллера для локальной обработки клиентских запросов. Такой подход к организации домена позволит разгрузить VPN туннели между офисами и использовать эти каналы связи только для репликации.

1.4.1.5. Трафик репликации

В ходе репликации контроллеры обмениваются только новыми и измененными данными, поэтому объем трафика оказывается небольшим, и его довольно просто прогнозировать, например:

- при добавлении 1000 пользователей трафик репликации окажется в два раза больше, чем при добавлении 500 пользователей;

- если устанавливается новый контроллер в удаленном офисе с подключением в 5 Мбит/с, то ожидается, что на репликацию базы данных размером 1,5 ГБ потребуется не менее часа.

Внимание: Крайне важно исключить долгую работу контроллера в автономном режиме, т.к. после этого может потребоваться повторная инициализация контроллера с загрузкой всего каталога.

1.4.1.6. Сервис Глобального каталога

Глобальный каталог представляет из себя ещё одну базу данных (экземпляр **dirsrv**), который содержит частичную копию (часть атрибутов) некоторых объектов службы каталога. Он необходим для корректной работы доверительных отношений и устанавливается на контроллере домена, выделенном под него. На контроллер домена, где будет установлен глобальный каталог, необходимо выделить дополнительно 15-20% дискового пространства и такой же объем ОЗУ. Дополнительным потреблением процессорного времени можно пренебречь, т.к. чтение объектов из памяти достаточно малозатратная операция для ЦПУ.

1.4.1.7. Модуль синхронизации

Основные ресурсы, потребляемые при работе модуля синхронизации — это дисковое пространство, размер которого будет прямо пропорционален количеству синхронизируемых объектов. Дисковое пространство потребляется БД **PostgreSQL**, где хранятся метаданные всех синхронизированных объектов. Для корректной работы модуля синхронизации необходимо выделить дополнительно 15-20% дискового пространства от расчётного.

С настройками по умолчанию **PostgreSQL** хранит кеш объектов в оперативной памяти. Учитывая, что около 80% обрабатываемых запросов являются запросами на чтение, будет достаточно выделить дополнительно 10-15% от расчетного объёма ОЗУ.

1.4.1.8. Расчет отдельных технических пакетов

> **Внимание:** В разделе описаны рекомендации только для части используемых технологий. Остальные рекомендации находятся в разработке. Если возникли сложности с работой **RabbitMQ**, рекомендации по расчету ресурсов для этих технических пакетов представлены ниже.

1.4.1.9. Планирование ресурсов RabbitMQ

Начиная с версии ALD Pro 2.3.0, для **RabbitMQ** отключена кластеризация между контроллерами домена, поэтому брокер сообщений обеспечивает только транзакционность Портала управления ALD Pro и его накладными расходами можно пренебречь, отнеся их к базовым потребностям операционной системы.

1.4.1.10. Планирование ресурсов Salt

Начиная с версии ALD Pro 2.4.0, для управления хостами используется только автономная служба **Salt Minion**, которой не требуется **Salt Master** на сервере, поэтому на контроллерах домена дополнительные ресурсы для работы системы конфигурирования выделять не требуется.

1.4.2. Горизонтальное масштабирование

Горизонтальным масштабированием называют повышение производительности системы за счет увеличения количества вычислительных узлов, обслуживающих клиентов, без изменения их производительности. Например, если предполагается использовать контроллеры с 8-12 потоками ЦПУ, которые могут обслуживать по 10 тысяч пользователей, то в организации со штатом в 30 тысяч сотрудников потребуется не менее 3 контроллеров такой производительности:

$$N\ controllers = 30\,000\ users / 10\,000\ per\ controller = 3\ controllers.$$

Синхронизация данных между контроллерами обеспечивается процессом репликации, во **FreeIPA** она происходит в мультимастер-режиме, т.е. изменения можно вносить на любом контроллере. Балансировка нагрузки и отказоустойчивость обеспечивается через DNS —

клиентам выдается до трех DNS-серверов, каждый из которых может принадлежать определенному сайту и выдавать SRV-записи в соответствии принадлежностью служб к соответствующим сайтам.

1.4.2.1. Топология соглашений о репликации

В качестве примера обычно приводят топологию из четырех сайтов, см. *Топология из четырех сайтов*, в каждом из которых по четыре контроллера, в результате чего полученная схема вводит администраторов в заблуждение, что в сайте может быть не более четырех контроллеров.

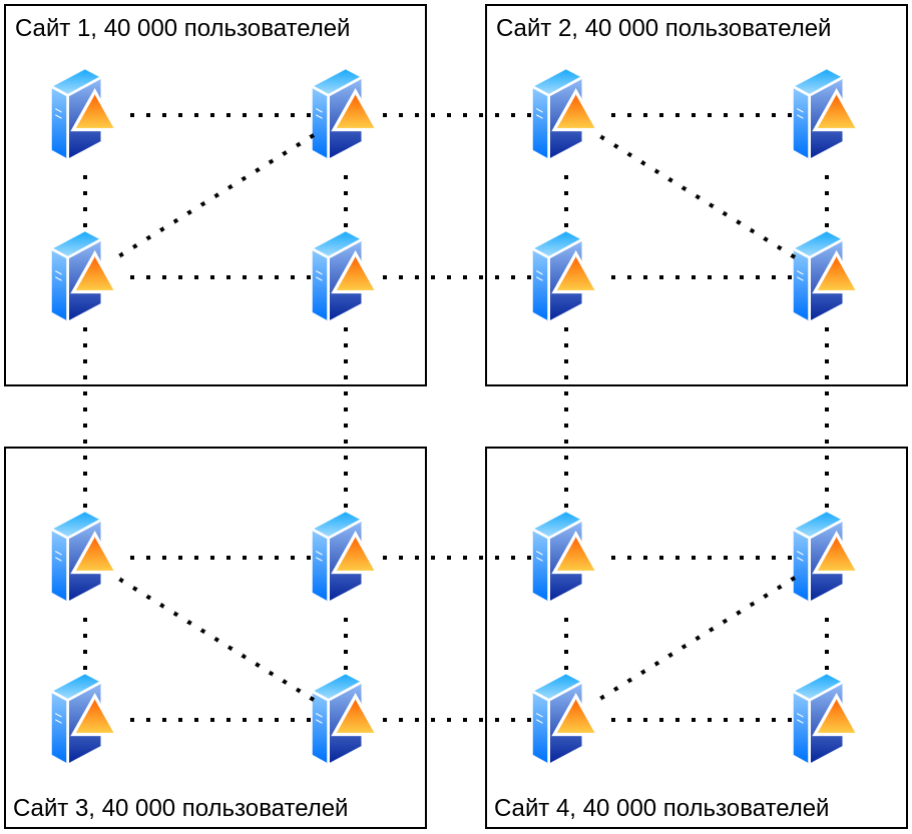


Рисунок 1.1 – Топология из четырех сайтов

В реальной жизни редко бывает, чтобы численность сотрудников в каждой локации была одинаковой, поэтому количество контроллеров может быть, как больше, так и меньше. Ниже приведен пример с тремя сайтами численностью 20, 30 и 80 тысяч человек, см. *Топология из трех сайтов*.

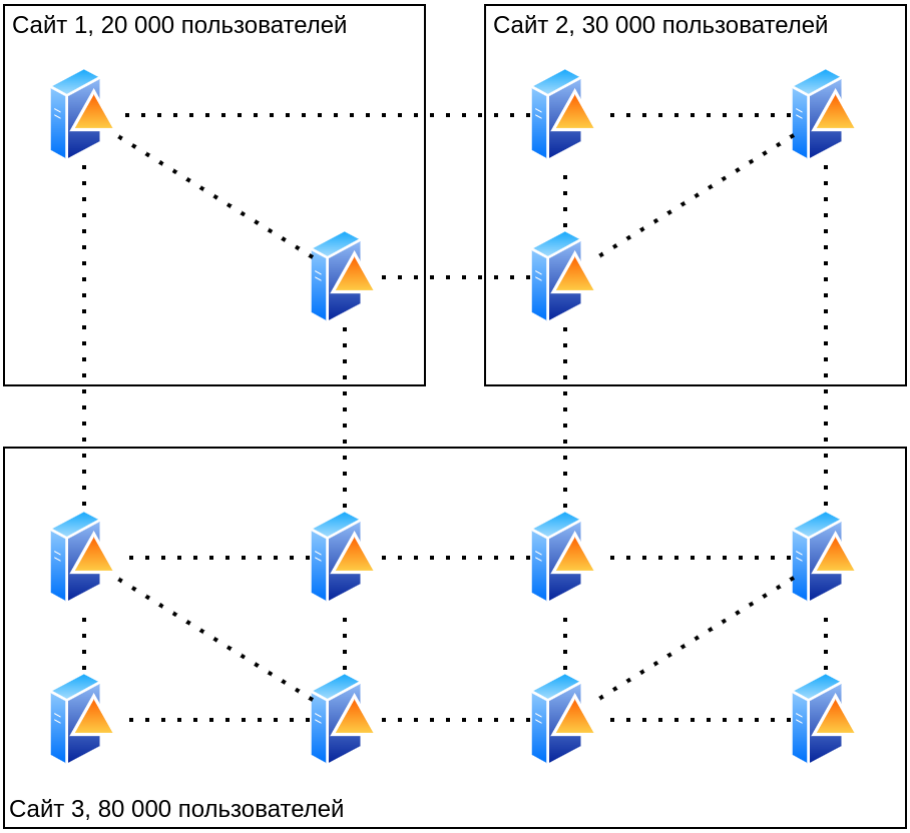


Рисунок 1.2 – Топология из трех сайтов

1.4.2.2. Рекомендации по построению топологии

При планировании топологии следует учитывать несколько рекомендаций:

- Для обеспечения надежности на уровне **0,99** следует установить в каждом сайте не менее двух контроллеров домена.

Если предположить, что сервер работает с надежностью в 90%, тогда при добавлении еще одного такого сервера получается, что в 90% случаев от оставшихся 10% система будет оставаться все так же работоспособной. Итоговая надежность повысится до **0.9 + 0.1 * 0.9 = 0.99**

Максимальное количество контроллеров на сайт не лимитируется, но в настройках рабочих станций можно задать только три DNS-сервера, поэтому стандартными средствами можно обеспечить надежность до **0,999**, а дополнительные контроллеры будут использоваться только для распределения нагрузки.

- Для обеспечения надежности репликации контроллеров домена на уровне 0,99 у каждого из них должно быть не менее 2 соглашений о репликации с другими контроллерами домена. Для обеспечения надежности межсайтовой репликации на

том же уровне у каждого сайта должно быть не менее 2 соглашений о репликации с контроллерами из других сайтов.

- Не рекомендуется создавать более четырех соглашений о репликации на одном контроллере домена, т.к. при недоступности каких-то из серверов алгоритм репликации будет останавливаться на каждом из них, ожидая ответа, что будет приводить к значительному снижению скорости репликации. По этой же причине рекомендуется удалять из домена контроллеры, которые более не используются, а число временно выключенных серверов сводить к минимуму.
- При планировании топологии нужно стремиться к тому, чтобы между любыми двумя контроллерами в домене было не более 5-10 промежуточных узлов, но при этом количество контроллеров, у которых сразу более 4 соглашений о репликации, нужно стараться сводить к минимуму. Такая оптимизация необходима для уменьшения количества прыжков (hops), за которые изменения могут быть распространены в домене.

При этом вне зависимости от топологии домена работу ИТ-службы нужно организовать таким образом, чтобы изменения в домен вносились как можно ближе к потребителю. Например, если потребуется сбросить пароль сотруднику из Новосибирска, то эти изменения должен внести местный администратор с контроллера из своего сегмента сети. В этом случае такие изменения будут максимально быстро реплицированы на все контроллеры, которые обслуживают сотрудника непосредственно, а когда изменения станут доступны на серверах московского региона — это уже не принципиально.

- Для максимальной производительности репликации под эту функцию можно выделить отдельные контроллеры и снять с них обслуживание обычных пользователей путем переключения в режим скрытой реплики (hidden replica).

1.4.3. Результаты нагрузочного тестирования

В части горизонтального масштабирования:

- В один домен можно ввести >400 контроллеров, и это не нарушит работу механизма репликации. Не наблюдается технических ограничений для создания доменов на 2-3 тысячи контроллеров, а с выходом функции доверительных отношений между доменами ALD Pro в версии 2.3.0 большую ИТ-инфраструктуру можно разбить на несколько доменов и вопрос горизонтального масштабирования снимается полностью.
- У контроллера домена может быть >150 соглашений о репликации, и это не нарушит

работу механизма репликации. Значение не является техническим ограничением, но дальнейшие тесты не имеют практического смысла.

При создании большого количества соглашений о репликации следует учитывать, что при недоступности напарника по соглашению контроллер некоторое время будет ожидать от него ответа, что создаст небольшую задержку. Если у контроллера будет много неработающих соглашений, то задержка окажется ощутимой, поэтому рекомендуется не создавать более четырех соглашений, если только это не обоснованно с позиции оптимизации топологии.

В части вертикального масштабирования:

- Контроллер может держать нагрузку >10 тысяч рабочих станций. При увеличении количества ядер производительность возрастает линейно с коэффициентом 0.9 в широком диапазоне (проверяли до 48 ядер), т.е. увеличение числа ядер в 2 раза повышает производительность на 80%.

Использование технологии гиперпоточности добавляет 50-75%, но ее эффективность снижается, если нагрузка на ядра становится выше 80%.

В дополнение к сказанному текущая архитектура групповых политик ALD Pro переносит всю нагрузку по суммированию параметров на рабочие станции, поэтому обслуживание тысяч хостов не создает ощутимой нагрузки на сервер.

- В домене можно создать >30 млн. объектов, и это не нарушит работу контроллеров. Увеличение количества объектов приведет к увеличению размера базы данных и объема потребляемой оперативной памяти. В больших инфраструктурах настоятельно рекомендуется отключать плагин **Schema Compatibility**, т.к. построение виртуального дерева объектов каталога создает существенную задержку в запуске служб (на базе в 30 млн. объектов задержка составляла чуть более 30 минут).

Пример производительности в пересчете на один контроллер домена, полученные на каталоге размером 30 млн. объектов (более 100 ГБ на диске). Характеристики каждого контроллера 16 ЦПУ, 128 ГБ ОЗУ. (рис.1.3-1.5)

- LDAP Search - 5 040 чтений в секунду
- LDAP Bind - 126 аутентификаций в секунду
- LDAP Modify - 126 изменений в секунду
- Kerberos TGT - 180 аутентификаций в секунду

Рисунок 1.3 – Результаты нагрузочного тестирования

Рисунок 1.4 – Результаты нагрузочного тестирования

Рисунок 1.5 – Результаты нагрузочного тестирования

1.4.4. Практический кейс по планированию ресурсов

Исходные требования:

- Объектов в каталоге: 1 млн. (650 тыс. пользователей, 350 тыс. групп).
- Обслуживаемых пользователей: 2 тыс.

Расчеты:

Для размещения файлов операционной системы и каталога на жестком диске потребуется не менее 60 ГБ HDD:

$$HDD = 50 + 1\ 000\ 000/20\ 000 \approx 100\ GB$$

Для того, чтобы контроллер домена смог разместить каталог с 1 млн. объектов в оперативной памяти, ему потребуется не менее 100 ГБ ОЗУ:

$$RAM = 3 + 1\ 000\ 000/20\ 000 \approx 66\ GB$$

Если к контроллеру будут обращаться 2 тысячи пользователей, то ему потребуется не менее 3 потоков:

$$CPUs = 1 + 2\ 000/1\ 000 = 3$$

Конкретно к этому серверу будет обращаться 2 тыс. пользователей, поэтому достаточно одной виртуальной машины. Но если нужно будет обеспечить единовременную работу всех 600 тысяч пользователей, то потребуется не менее 300 таких контроллеров и более целесообразным будет увеличить производительности контроллеров до 10 тысяч пользователей (8-12 потоков ЦПУ), тогда количество контроллеров можно будет сократить до 60 единиц.

Следует учесть, что объекты в каталоге создаются в среднем со скоростью один объект в секунду и это значение растет по мере увеличения количества объектов в каталоге, поэтому на создание 1 млн. объектов уйдет не менее 11 дней. Ускорить процесс можно увеличением числа потоков, но зависимость будет нелинейной, т.к. потоки будут конкурировать за общие ресурсы.

Дополнительного ускорения процедуры можно достичь путем создания пользователей на нескольких контроллерах одновременно, т.к. при репликации вычислительных ресурсов требуется меньше, чем при создании объектов.

Для тестирования остальных подсистем в соответствии с типовой схемой, см. *Типовая схема инфраструктуры*, нужно дополнительно 8 виртуальных машин: 2 ЦПУ, 4 ГБ ОЗУ, 100 ГБ HDD (DHCP, установка ОС по сети, сервер печати, сервер общего доступа к файлам, сервер аудита, сервер мониторинга, сервер репозиториев, клиентская машина).

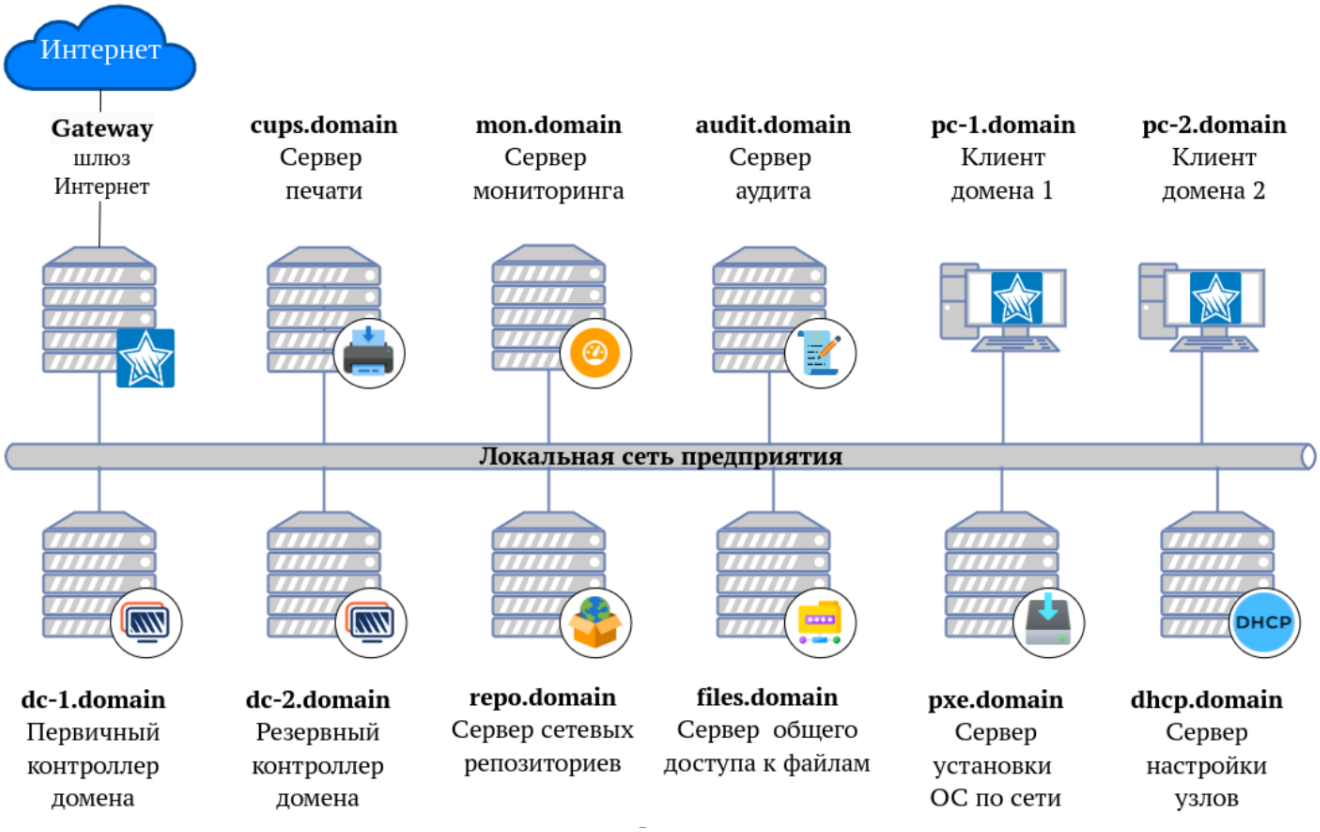


Рисунок 1.6 – Типовая схема инфраструктуры

Для упрощения предварительного расчёта ресурсов можно воспользоваться двумя таблицами:

На один КД	RAM	HDD
Базовые требования	3	50
Количество объектов		
3 000	8,0	50,2
5 000	8,0	50,3
10 000	8,0	50,5
50 000	8,0	52,5
100 000	8,0	55,0
150 000	10,5	57,5
200 000	13,0	60,0
250 000	15,5	62,5
300 000	18,0	65,0
400 000	23,0	70,0
500 000	28,0	75,0
700 000	38,0	85,0
800 000	43,0	90,0
1 000 000	53,0	100,0
1 500 000	78,0	125,0
2 000 000	103,0	150,0

Таблица 2 — Расчёт дискового пространства и количество оперативной памяти на один контроллер домена

На домен, сайт	CPU	Кол-во КД
Базовые требования	1	
коэффициент	1000	32
Активные объекты		
3 000	8,0	1,0
5 000	8,0	1,0
10 000	11,0	1,0
50 000	51,0	2,0
100 000	101,0	4,0
150 000	151,0	5,0
200 000	201,0	7,0
250 000	251,0	8,0
300 000	301,0	10,0
400 000	401,0	13,0
500 000	501,0	16,0
700 000	701,0	22,0
800 000	801,0	26,0
1 000 000	1001,0	32,0
1 500 000	1501,0	47,0
2 000 000	2001,0	63,0

Таблица 3 — Расчёт необходимого количества процессорных ядер и количества контроллеров домена для обработки запросов от активных пользователей

1.5. Рекомендации по именованию домена

В документации и обучающих курсах применяются домены **contoso.com** или **ad.example.test**, которые используют для обучающих целей, но не приемлемы для развёртывания продуктивных сред. Поэтому рекомендуется придерживаться следующих правил:

- Лучший вариант - это использовать свой домен третьего уровня. Например, если организация приобрела домен **mycompany.ru** для своего Web-сайта и почты, то для **ALD Pro** можно использовать доменное имя третьего уровня **ald.mycompany.ru**.

Если в группе компаний планируется несколько юридических лиц, то каждому юр. лицу можно создать домен четвертого уровня. Таким образом будет гарантироваться отсутствие конфликтов с публичными DNS-именами.

- Если публичный домен не используется, то можно задействовать зоны **.lan** или **.internal**, например, **ald.company.lan**. Во многих инструкциях упоминается возможность использования зоны **.local** в качестве частного домена верхнего уровня для внутренних нужд предприятия, но в этом случае возникают конфликты Multicast DNS (RFC6762) и невозможность использования протокола **zeroconf**, который реализован в Linux службой **Avahi**, поэтому данная зона не рекомендуется.

Важно: Предложенные зоны **.lan** и **.internal** не зарегистрированы в глобальном списке Top-Level Domains, но всегда будет оставаться вероятность, что их ведут в эксплуатацию в будущем. Соответственно, следует использовать зоны **.lan**, **.internal** и **.local** учитывая эти риски.

Выбор префикса для домена:

- Следует выбирать префикс, который вряд ли устареет, т.е. избегать имен, таких как линейка продуктов или операционная система, которые могут измениться в будущем. Рекомендуется использовать универсальные имена, такие как **corp**, **ent** или **ald**.
- Следует выбирать префикс, содержащий только стандартные символы Интернета **a–z**, **0–9** и (**-**), но не полностью числовые.
- Следует выбирать префикс длиной не более 15 символов, потому что первый компонент доменного имени будет использован в качестве NetBIOS-имени.
- При выборе префикса следует учитывать, что два домена не смогут работать в доверительных отношениях, если у них будут совпадать NetBIOS-имена и имена контроллеров домена, например, если текущий домен AD DS имеет имя **corp.mycompany.com**, то не стоит для домена **ALD Pro** использовать имя **corp.mycompany.ru**. Иначе не получится создать между этими доменами доверительные отношения.

Важно: В стандарте доменных имен (RFC1034) разрешается использовать заглавные буквы латинского алфавита **A-Z**, но служба каталога **FreeIPA** налагает дополнительные требования и обязует использовать только нижний регистр символов.

Если в команде ввода в домен указать имя хоста заглавными буквами, то процедура

завершиться ошибкой. Если поменять имя хоста на заглавные (например, **FS-01.ald.company.lan**) после ввода в домен, то появится ошибка при установке подсистем на такие хосты.

Перед созданием имени ознакомьтесь с 2.1.1.3. *Настройка имени хоста*

На предприятии принято называть сервера и персональные компьютеры по заданной конвенции наименований, например, конвенция для серверов **<location>-<type>-<number>.<domain>** разрешает такой вариант наименования **msk-dc-1.ald.company.lan**. Во-первых, это нужно для стандартизации и общего понимания архитектуры системы, а во-вторых позволит качественно управлять узлами с помощью автоматизации. Например, имя узла **dc-1** с легкостью разделяется на части по разделителю «**-**» с помощью команды **cut**:

```
echo "dc-1.ald.company.lan" | cut -d"." -f 1 | cut -d"-" -f 1
```

Результат выполнения покажет, что выбранный узел является контроллером, а не мониторингом:

```
dc
```

Развертывание подсистем

2.1. Развертывание первого контроллера домена

2.1.1. Подготовка окружения сервера к установке контроллера домена

2.1.1.1. Требования к ОС

Последняя версия доменных служб ALD Pro 2.4.0 для установки серверной части поддерживает совместимость с ОС Astra Linux версии с 1.7.5 до 1.7.6 UU1 включительно, а клиентской - до 1.8.1. Однако, система ALD Pro крайне чувствительна к настройкам параметров Операционной Системы. Поэтому установку домена и реплики ALD Pro необходимо выполнять на чистой ОС, на которой ранее не были установлены другие сервисы.

На Контроллерах Домена и подсистемах должна быть установлена Операционная Система с уровнем защищенности «Максимальный» («Смоленск»). Для рабочих станций можно использовать любой уровень защищённости. Для проверки версии и уровня защищенности используйте команды:

```
cat /etc/astra/build_version
sudo astra-modeswitch getname
```

При первом выполнении команды `sudo` система потребует ввести пароль, а после успешной аутентификации внесет необходимую информацию в кэш и будет хранить ее следующие 15 минут в соответствии со значением параметра `timestamp_timeout` в файле */etc/sudoers*.

Чтобы расширить ограничение в 15 минут можно запустить новую сессию от имени суперпользователя командой `sudo -i` и команды можно будет запускать без `sudo`.

```
sudo -i
```

Результатом вывода в терминале будет имя `root` и символ #:

```
[sudo] пароль для localadmin:****
root@astra:~#
```

Выйти из привилегированной сессии можно будет командой `exit`.

```
exit
```

Результат вывода в терминале о том, что сессия вернулась в обычный режим будет `$`:

```
Выход
localadmin@astra:~$
```

2.1.1.2. Настройка сетевого интерфейса

Перед развертыванием доменной инфраструктуры важно проектировать топологию локальной сети. Важно отметить, что на контроллерах домена и других серверах группировки настоятельно рекомендуется использование статических адресов.

Если операционная система была установлена с графическим интерфейсом **fly-wm**, то в качестве менеджера сетевых подключений устанавливается демон **NetworkManager**. Если нет необходимости использовать его для настройки сетевого интерфейса, то отключить его можно следующими командами:

```
sudo systemctl stop NetworkManager
sudo systemctl disable NetworkManager
sudo systemctl mask NetworkManager
sudo systemctl status NetworkManager
```

В последней строке в результате должна отобразиться информация об маскировке службы **NetworkManager**:

```
NetworkManager.service
Loaded: masked (Reason: Unit NetworkManager.service is masked.)
Active: inactive (dead) since Tue 2023-09-19 18:16:15 MSK; 3min 35s ago
Main PID: 493 (code=exited, status=0/SUCCESS)
...
```

В Astra Linux Special Edition для оперативных обновлений версии ниже 1.7.4 требуется выполнять следующие команды:

```
sudo systemctl stop network-manager
sudo systemctl disable network-manager
sudo systemctl mask network-manager
sudo systemctl status network-manager
```

После отключения NetworkManager сетевые настройки необходимо задавать в файлах *interfaces* и *resolv.conf*. Файл */etc/network/interfaces* используется командами `ifup`/`ifdown` для конфигурирования сетевых интерфейсов.

Служба каталога тесно интегрирована со службой разрешения имен, поэтому контроллер домена выступает еще и в качестве DNS-сервера. Адреса DNS-серверов через DHCP или вручную распространяются по всей сети, поэтому на контроллере домена настоятельно рекомендуют устанавливать статический адрес. Отредактировать файл конфигурации сети можно командой:

```
sudo nano /etc/network/interfaces
```

Пример настройки сети контроллера на статический IP-адрес:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 10.0.1.11
   netmask 255.255.255.0
   gateway 10.0.1.1
```

Комментарии по использованным параметрам конфигурации:

- `auto eth0` — строка, начинающаяся со слова `auto`, указывает интерфейс, который будет подниматься автоматически при вызове команды `ifup -a`. Посмотреть список доступных интерфейсов можно командой `ip a`, первый сетевой интерфейс обычно имеет идентификатор `eth0`. Выберите нужный сетевой интерфейс из этого списка.
- `iface eth0 inet static` — строка со словом `iface`, начинает группу строк, отвечающих за настройку указанного интерфейса. Следующее слово `inet`/`inet6` указывает, какой протокол будет использоваться — **IPv4** или **IPv6** соответственно. Следующее слово *static/dhcp* указывает способ назначения настроек — вручную, или динамически.

- `address`, `netmask`, `gateway` — задают IP адрес, маску и шлюз по умолчанию для интерфейса, указанного в предшествующей ей строке `iface`, если для него выбран способ назначения настроек `static`.

В некоторых инструкциях можно встретить указание в файле *interfaces* таких параметров, как `dns-nameservers` и `dns-search`, но они имеют силу только в том случае, если в системе работает служба **resolvconf**, которая переносит эти настройки соответствующим образом в файл */etc/resolv.conf*, но в Astra Linux по умолчанию она не устанавливается. Для получения развернутой информации о допустимом синтаксисе файла *interfaces* можно выполнить команду `man interfaces`.

Чтобы применить новые настройки, следует перезапустить службу **Networking**, может так же потребоваться очистить старое соединение командой `flush`:

```
sudo ip addr flush dev eth0
sudo systemctl restart networking
```

Для возможности обращения к публичным репозиториям следует настроить функцию разрешения имен. Файл */etc/resolv.conf* определяет настройки для процедур разрешения имен из библиотеки **glibc**, которая используется в сетевых утилитах **ping**, **dig** и т.д. В этом файле рекомендуется указать бесплатный сервер разрешения имен от Яндекс. Отредактировать его можно командой:

```
sudo nano /etc/resolv.conf
```

Вставить в этот файл следующее содержимое:

```
nameserver 77.88.8.8
```

После установки этих настроек проверить подключение к репозиториям Astra Linux:

```
ping -c 4 dl.astralinux.ru
```

Если продукт необходимо установить в закрытом периметре без доступа в Интернет, это можно сделать с использованием установочного диска или путем установки из локального репозитория (подробнее см. *Подключение репозиториев*).

2.1.1.3. Настройка имени хоста

Имя хоста должно быть задано в формате полного имени **FQDN** (от англ. **Fully Qualified Domain Name**), например, **dc-1.ald.company.lan**, где **ald.company.lan** — это название домена предприятия. Такой подход принят в **Red Hat**, и он наследуется при использовании **FreeIPA**. При данном подходе вызов команды `hostname` без параметров будет выдавать FQDN-хоста.

> **Внимание:** При вводе имени контроллера домена допускается использование заглавных и строчных букв латинского алфавита (**A-Z**, **a-z**), цифр (**0-9**), точки (**.**) и дефиса (**-**). Данное правило касается имен хоста клиентов домена.

Если устанавливать имя хоста напрямую в файле */etc/hostname*, то изменения вступят в силу только после перезагрузки, поэтому в Astra Linux рекомендуется это делать с помощью утилиты **hostnamectl**:

```
sudo hostnamectl set-hostname dc-1.ald.company.lan
```

Для того чтобы имя контроллера всегда могло быть преобразовано в IP-адрес, даже при недоступности сервиса DNS, в файл */etc/hosts* требуется добавить строку, соответствующую его FQDN. В эту строку рекомендуется вписать не только полное, но короткое имя хоста, первым по списку обязательно должно идти полное имя хоста, т.к. первое имя считается каноническим и будет возвращаться командой `hostname -f`, что требуется для корректной работы скриптов автоматизации. Указывать имена необходимо в нижнем регистре.

Еще крайне важно удалить строку, связывающую имя хоста с адресом `localhost`, т.к. в соответствии с настройками */etc/gai.conf* эти адреса имеют выше приоритет, а нам крайне важно, чтобы имя хоста разрешалось в локальный адрес, потому что некоторые службы могут прослушивать порты только на этом адресе.

Изменить содержимое файла можно командой:

```
sudo nano /etc/hosts
```

Примерное содержимое файла */etc/hosts* на **dc-1**:

```
127.0.0.1 localhost.localdomain localhost
#127.0.1.1 dc-1 - закомментировать или удалить строку с адресом локальной
↪петли
10.0.1.11 dc-1.ald.company.lan dc-1
```

Где `10.0.1.11` — статический IP-адрес контроллера домена, который выбран на шаге настройки сетевого интерфейса.

2.1.1.4. Подключение репозиториев

Файлы программ Linux объединяются в пакеты и распространяются через специальные хранилища, называемые репозиториями. Основным файлом для хранения списка доступных репозиториев является */etc/apt/sources.list*, дополнительные списки могут храниться в файлах **.list* в директории */etc/apt/sources.list.d/*. Отредактировать список можно командой:

```
sudo nano /etc/apt/sources.list
```

- **через интернет-репозиторий**

Для установки на сервере под управлением Astra Linux Special Edition 1.7.6 программного продукта ALD Pro версии 2.4.0 из официальных интернет-репозиториев РусБИТех-Астра, содержание этого файла должно быть следующим:

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-main 1.7_
↪x86-64 main non-free contrib
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-update 1.
↪7_x86-64 main contrib non-free
```

Кроме того, содержание файла может быть следующим:

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base 1.7_
↪x86-64 main non-free contrib
```

По умолчанию Astra Linux предлагает использовать репозитории `stable`, которые соответствуют последней версии операционной системы, но для работы с ALD Pro требуется переключить репозитории на `frozen`, чтобы гарантировать полную совместимость пакетов. Информация о поддержке очередных обновлений и возможности обновления операционной системы публикуется в **release notes**.

- **с помощью установочного диска**

Для установки на сервере под управлением Astra Linux Special Edition 1.7.6 программного продукта ALD Pro версии 2.4.0 с установочного диска, необходимо примонтировать диск, используя команду:

```
sudo mount /dev/sr0 /media/cdrom
```

Создать отдельный список репозиториев */etc/apt/sources.list.d/aldpro.list* командой:

```
sudo nano /etc/apt/sources.list.d/aldpro.list
```

Добавить следующую строку в файл:

- при установке через интернет-репозиторий:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0/ 1.7_x86-64 main base
```

- при использовании установочного диска

```
deb file:///media/cdrom 1.7_x86-64 main base
```

В source-листах менеджера **apt** каждая строка указывает путь к репозиторию в формате:

```
deb <путь_корневому_каталогу_репозитория> <код_дистрибутива> <компонент1>
↪<компонент2> <компонент3>
```

Где:

- deb — указывает на то, что репозиторий соответствует репозиторию бинарных файлов с предварительно скомпилированными пакетами. Для репозиториев с исходными кодами используют `deb-src`.
- `uri` — задает uri-адрес репозитория, у интернет-репозиториев адрес начинается с `http(s)://`, адреса локальных репозиториев начинаются с `file:/`. При добавлении репозитория с диска командой `apt-cdrom add` в файле появится строка `cdrom:[]/`.
- `код дистрибутива` — дополняет uri, уточняя необходимый релиз продукта. В одном репозитории могут находиться пакеты сразу для нескольких релизов.
- `компонент` — это группа пакетов, объединенная по условиям использования:
  - `non-free` — группа содержит пакеты, которые не соответствуют принципам

свободного ПО, имеют патенты или другие юридические ограничения;

- `contrib` — группа содержит пакеты, которые сами по себе соответствуют принципам свободного ПО, но зависят от пакетов из группы `non-free` (т.е. не могут без них работать);
- `main` — группа содержит пакеты свободного ПО, которые не зависят от пакетов из групп `contrib` и `non-free`.

После изменения состава репозиториев следует обновить индекс доступных пакетов с помощью команды:

```
sudo apt update
```

Данная команда получает список пакетов и зависимостей, но не обновляет само программное обеспечение.

2.1.1.5. Настройка приоритетов

Пакетному менеджеру **APT** может быть доступно сразу несколько версий одного и того же приложения из разных репозиториев, поэтому он выбирает наиболее подходящего кандидата для установки в соответствии с приоритетами пакетов.

По умолчанию для всех пакетов, находящихся в репозиториях, приоритет `P=500`. Переопределить приоритет по умолчанию можно с помощью конфигурационных файлов в директории */etc/apt/preferences.d/*. В системе Astra Linux уже есть один такой конфигурационный файл, устанавливающий приоритет `900` для пакетов релиза 1.7_x86-64.

```
sudo cat /etc/apt/preferences.d/smolensk
```

Результат выполнения:

```
Package: *
Pin: release n=1.7_x86-64
Pin-Priority: 900
```

Это правило позволяет избежать установки и обновления пакетов из сторонних репозиториев, если компанией РусБИТех-Астра для операционной системы под релиз 1.7_x86-64 была разработана специальная версия.

2.1.1.6. Обновление программных пакетов

Необходимо проверить наличие доступных для обновления пакетов и выполнить обновление пакетов в случае их наличия:

```
sudo apt update
sudo apt list --upgradable
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confnew
```

> **Внимание:** Использовать команду `apt upgrade` категорически запрещается, т.к. она не удаляет устаревшие пакеты, даже если это требуется для обновления приложений, что может нарушить работу операционной системы. Поэтому вместо нее следует использовать команду `dist-upgrade`, причем, для продукта ALD Pro ее необходимо вызывать с опцией `--force-confnew`, чтобы пакетный менеджер мог переопределить содержимое конфигурационных файлов, даже если они были изменены из скриптов установки или вручную. Похожий результат можно получить с помощью утилиты **astra-update**, но она не подходит, так как использует иную стратегию обновления конфигурационных файлов, вызывая `dist-upgrade` с опцией `--force-confold`.

2.1.2. Установка контроллера домена

Примечание: В данном разделе описано **несколько** вариантов установки Контроллера Домена: с модулями Глобального Каталога и Синхронизации, а также без них. Просьба **внимательно** ознакомиться со всеми вариантами, прежде чем запускать скрипт установки и продвижения.

Теперь система готова к установке ALD Pro.

Если дополнительные модули синхронизации и глобального каталога не требуются, для установки системы необходимо выполнить команду:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-mp
```

Комментарии по использованным инструкциям и параметрам:

- `DEBIAN_FRONTEND` — переменная окружения, которая позволяет изменить режим взаимодействия с пользователем при установке пакетов менеджером APT. Многие приложения на стадии установки уточняют необходимые настройки для последующей работы, что станет помехой для автоматического развертывания. Переключение менеджера пакетов в режим `noninteractive` позволяет избежать уведомлений от **Kerberos**, **OpenDNSSec** и **PAM**.
- `-y` — параметр позволяет автоматически ответить «Да» на все возможные вопросы в ходе установки.
- `-q` — параметр позволяет скрыть сообщения о прогрессе установки, делая журнал более читаемым.
- `aldpro-mp` - инсталляционный пакет портала управления (management portal), который через зависимости позволяет установить все остальные пакеты продукта ALD Pro.

Если планируется использовать двусторонние доверительные отношения с лесом доменов Microsoft Active Directory, то нужно дополнительно установить модуль глобального каталога. В этом случае будет работать поиск объектов из домена ALD Pro в стандартных приложениях MS Windows.

При возникновении ошибок, информация о них записывается в журнал пакетного менеджера в файле */var/log/apt/term.log*.

В предыдущих версиях ALD Pro перед установкой первого контроллера домена требовалось перезагрузить сервер. Начиная с версии 2.0.0 перезагрузка не требуется, скрипт установки автоматически выполняет перезапуск всех необходимых служб, таких как **aldpro-salt-minion**.

В предыдущих версиях ALD Pro перед установкой первого контроллера домена требовалось вручную задать содержимое файла */etc/resolv.conf*, выполнив следующие шаги:

- Обновить файл */etc/resolv.conf*, так как на прошлом шаге уже установился DNS-сервер BIND, поэтому указывается **127.0.0.1** и домен **ald.company.lan** командой:

```
sudo nano /etc/resolv.conf
```

- Добавить следующее содержимое:

```
nameserver 127.0.0.1
search ald.company.lan
```

Где:

- `nameserver` указывает, что разрешение имен должно выполняться через локальную службу ISC Bind, которая обрабатывает запросы, поступающие на 53-й порт сервера;
- `search` задает DNS-суффикс, используемый при разрешении имен, поэтому при обращении к хосту по имени `pc-1` будет предпринята также попытка преобразования имени `pc-1.ald.company.lan`.

В последних релизах продукта данные настройки выполняются автоматически скриптом установки.

Все готово для продвижения сервера. Необходимо выполнить установку запуском **aldpro-server-install**, утилита является неинтерактивной, все параметры обязательны, включая пароль, иначе установка не будет выполнена успешно. Так как в команде требуется указать пароль в открытом виде, перед ее вызовом рекомендуется отключить запись истории команд.

Примечание: Если пароль попал в журнал команд, то можно удалить последнюю команду из истории с помощью команды `history -d $(history 1)` или напрямую отредактировать файл */root/.bash_history*

Отключить историю, чтобы пароль не был в нее записан, можно командой:

```
set +o history
```

Запуск скрипта продвижения домена:

```
sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'AstraLinux_176' --
↪ip 10.0.1.11 --no-reboot
```

Комментарии по использованным ключам:

- `-d` (domain) — имя домена.
- `-n` (name) — имя сервера.
- `-p` (password) — пароль администратора домена.
- `--ip` — ip адрес контроллера домена. Адрес требуется указывать явно, если на контроллере домена активно несколько сетевых интерфейсов.
- `--no-reboot` — отменяет перезагрузку после завершения процедуры настройки.

Выполнение скрипта занимает некоторое время, поэтому мы рекомендуем выполнить перезагрузку вручную после ознакомления с журналом.

- Описание параметров скрипта можно получить с помощью ключа `-h`.

> **Внимание:** В параметре `-n` нужно передать короткое имя сервера без указания домена, т.е. первую часть от полного FQDN-имени, которое выдает команда `hostname -f`.

Пароль должен быть не менее 8 символов. Для использования специальных символов в пароле, например, знака доллара, заключите пароль в одинарные кавычки, например: **„pa_s$w0rd“**.

При необходимости инициализации модуля синхронизации и/или глобального каталога добавляются ключи `--setup_syncer` и/или `--setup_gc` соответственно.

Пример:

```
sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'AstraLinux_176' --
↪ip 10.0.1.11 --no-reboot --setup_syncer --setup_gc
```

Теперь можно включить ведение истории команд:

```
set -o history
```

Необходимо проверить командой

```
cat /etc/resolv.conf
```

В обновленном скрипте установки этот файл настраивается на службу **bind** и поиск имен в домене **ald.company.lan**. Если файл *resolv.conf* по прежнему имеет ссылку на **77.88.8.8**, то необходимо заменить его содержимое на:

```
search ald.company.lan
nameserver 127.0.0.1
```

Контроллер домена установлен. Для применения изменений необходимо перезагрузить сервер:

```
sudo reboot
```

Для проверки статуса сервисов ALD Pro, их запуска, остановки и перезагрузки используется утилита **aldproctl** (функции аналогичны функциям утилиты **ipactl**).

```
sudo aldproctl start/stop/restart/status
```

Для действий с конкретным сервисом ALD Pro требуется задать имя сервиса:

```
sudo aldproctl start/stop/restart/status –service=<имя_сервиса>
```

Например:

```
sudo aldproctl status –service=syncer
```

После загрузки сервера необходимо войти в систему, используя доменную учетную запись администратора:

- login: admin
- password: **** (пароль администратора домена из строки продвижения сервера)

Внимание: Окно для входа может отобразиться раньше, чем станут доступны доменные службы, поэтому вход доменной учетной записью может стать доступен не сразу. Пока не станет доступна **LDAP-служба**, в списке источников учетных данных вместо имени домена будет отображаться «Ожидание ответа домена…». Пока не станет доступна **KDC-служба**, нельзя успешно пройти аутентификацию на сервере новым пользователем, у которого еще не сохранены учетные данные в кэше **SSSD-службы**.

Поэтому, если не получилось войти на сервер под доменной учетной записью сразу, необходимо подождать пару минут и попробовать еще раз. Если доступ так и не появился, необходимо войти на сервер локальной учетной записью и проверить журналы */var/log/auth.log* и */var/log/messages*

Чтобы проверить доступность доменных служб можно войти локальным пользователем и воспользоваться утилитой **ipactl**:

```
sudo ipactl status
```

Результат выполнения в консоли:

```
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

2.1.2.1. Установка модуля синхронизации и глобального каталога

Глобальный каталог (далее **ГК**) в **ALD Pro** - это отдельно поднятый экземпляр ldap-сервера **ALD Pro** имеющий схему схожую с **Active Directory** (далее **MS AD**). **ГК** служит для обеспечения **полной** поддержки двусторонних доверительных отношений с **MS AD**, обеспечивает поддержку поиска объектов (пользователей и групп) через стандартные утилиты **MS AD**. **ГК** состоит из следующих компонентов:

- Экземпляр **389-ds** (dirsrv@GLOBAL-CATALOG) слушающий порт **3268** (стандартный порт **ГК** для **MS AD**).
- Модуль синхронизации (**ipa-gcsyncd**) служба, обеспечивающая синхронизацию атрибутов безопасности из обычного экземпляра **ldap ALD Pro** в **ГК**. Своего рода адаптер между двумя разными схемами **ldap**.
- Модуль проверки целостности **ГК** (**aldpro-gc-inspector**) - служба для отслеживания конфликтов при синхронизации.

Важным условием для успешной синхронизации объектов, является наличие у них атрибута **ipaNTSecurityIdentifier** который конвертируется в **ObjectSid** в **ГК**. Для **MS AD** это идентификатор безопасности, который она добавить в свои списки доступа **MS AD**.

Работа назначения прав доступа к ресурсу **MS AD** доверенному объекту (пользователю или группе **ALD Pro**) происходит следующим образом:

1. Администратор через оснастку **MS AD** выбирает в параметрах поиска домен **ALD Pro** и выполняет поиск. В этот момент происходит подключение по **kerberos** к **ldap ГК** по порту **3268**. Если действия выполняются от учетной записи **MS AD**, корректная привязка не происходит, т.к. данной учетной записи нет в **ldap ГК**. Поэтому появляется запрос на ввод учетных данных домена **ALD Pro**. Необходимо ввести данные, как при **kinit** (например, admin@EXAMPLE.COM). Если обращение к **ГК** происходит от доверенной учетной записи <ДОМЕН>$$, то система безопасности **ГК** выполняет корректное сопоставление и запрос на ввод учетных данный домена **ALD Pro** не возникнет.

2. После успешного запроса, появиться список объектов **ALD Pro** с отображением их `uid` или `cn` в зависимости от типа объекта (`uid` - для пользователя, `cn` - для группы).

3. После выбора объекта и подтверждении его выбора, система безопасности **MS AD** выполняет запрос к контроллеру домена **ALD Pro** через механизмы **samba** с указанием в запросе **objectSID** выбранного ранее объекта.

4. На стороне контроллера домена **ALD Pro** модули **samba** принимают запрос обрабатывают его и делают внутренний запрос в **ldap** (**порт 389**) **ALD Pro** через специальный адаптер - библиотеку **ipasam** (в некоторых реализациях, для дополнительного функционала доверительных отношений между доменами **ALD Pro** используется **aldprosam**).

5. После успешной обработки запроса, механизмы **samba** отправляют подтверждение о существовании объекта.

6. После получения ответа от сервера **ALD PRo**, модули безопасности **MS AD** добавляют идентификатор безопасности в свои списки доступа.

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -q -y aldpro-mp aldpro-gc
```

Если планируется использовать расширенные функции интеграции с доменом **Microsoft Active Directory**, то нужно выполнить установку модуля синхронизации, который обеспечивает поддержание целостности в двух доменах **ald** и **adds**. В этом случае в двух доменах будет один и тот же состав пользователей и групп, синхронизироваться будет необходимый список атрибутов, включая пароли. Это позволит пользователям получать доступ к любым информационным системам по логину и паролю вне зависимости от того, через контроллер какого домена информационная система проверяет аутентичность пользователей.

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -q -y aldpro-mp aldpro-
↪syncer
```

Возможно применение ключей установки модулей в команде установки контроллера домена:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -q -y aldpro-mp aldpro-
↪gc aldpro-syncer
```

2.1.2.2. Установка модуля “Доверие между доменами ALD Pro”

Если планируется устанавливать доверительные отношения между доменами ALD Pro, необходимо установить модуль “Доверие между доменами ALD Pro”.

Технические требования для установки доверительных отношений между доменами ALD Pro:

1. Установку пакетов и выполнение инструкций по настройке модуля доверия для КД необходимо выполнять на обоих КД, между которыми настраивается доверие;
2. Установку необходимо выполнять от имени учетной записи администратора системы с высоким уровнем целостности;
3. На обоих Контроллерах Домена (КД) должно быть установлено обновление ALD Pro 2.4.0.

Внимание: При установке модуля «Доверие между доменами ALD Pro» (далее — модуль) версии обновляемых компонентов SSSD будут снижены до 2.4.0-1astra.se8-aldpro1, что соответствует версиям в составе срочного оперативного обновления 1.7.3.UU.2 операционной системы специального назначения «Astra Linux Special Edition» (очередное обновление 1.7).

Порядок установки модуля на **Контроллерах Домена** ALD Pro:

1. Подключить репозиторий, содержащий необходимые для функционирования модуля обновления:
    - Примонтировать ISO-образ *ALDPro-2.4.0-trust.iso*, выполнив команду:

```
sudo mount /opt/ALDPro-2.4.0-trust.iso /media/cdrom
```

- Создать файл */etc/apt/sources.list.d/trust.list* со следующим содержимым:

```
deb file:///media/cdrom 2.4.0 aldpro
```

2. Добавить в файл */etc/apt/preferences.d/aldpro* следующее содержимое:

```
Package: *
Pin: release n=2.4.0
Pin-Priority: 1000
```

3. Выполнить команды:

```
sudo apt update
sudo apt dist-upgrade
sudo apt install libipa-aldpro
```

4. Добавить в файл */usr/share/aldpro/updates/92-aldpro-extdom-extop-plugin.update*:

```
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
only:changetype: add
only:objectclass: top
only:objectclass: nsSlapdPlugin
only:objectclass: extensibleObject
only:cn: ipa_extdom_extop
only:nsslapd-pluginpath: libaldpro_extdom_extop
only:nsslapd-plugininitfunc: ipa_extdom_init
only:nsslapd-plugintype: extendedop
only:nsslapd-pluginenabled: on
only:nsslapd-pluginid: ipa_extdom_extop
only:nsslapd-pluginversion: 1.0
only:nsslapd-pluginvendor: AstraLinux
only:nsslapd-plugindescription: Support resolving IDs in trusted domains to
 ↪names and back
only:nsslapd-plugin-depends-on-type: database
only:nsslapd-basedn: $SUFFIX
```

5. **Последовательно** выполнить команды, каждую **после завершения** предыдущей:

```
sudo ipa-ldap-updater /usr/share/aldpro/updates/92-aldpro-extdom-extop-plugin.
↪update

sudo net conf setparm global "passdb backend" "aldprosam:ldapi://%2fvar%2frun
↪%2fslapd-$(hostname -d | sed 's/\./-/g; s/.*/\U&/').socket"

sudo ipactl restart
```

Для автоматизации процесса установки модуля «Доверие между доменами ALD Pro» на **клиентах** домена ALD Pro рекомендуется создать репозиторий из ISO-образа *ALDPro-2.4.0-trust.iso* и с помощью задания автоматизации произвести установку модуля.

Описание процесса создания репозитория из ISO-образа приведено в *Создание репозиториев из ISO образа*. Описание порядка создания и запуска заданий автоматизации приведено в Справочном центре ALD Pro.

Порядок установки модуля на **Клиентах** ALD Pro:

> **Внимание:** Следующие действия актуальны при использовании ОС Astra Linux версией ниже 1.7.6. Если на клиентах установлена ОС Astra Linux версии 1.7.6, см. *Особенности установки при использовании Astra Linux версии 1.7.6 и выше*.

1. Подключить репозиторий, содержащий необходимые для функционирования модуля обновления:

   - Примонтировать ISO-образ *ALDPro-2.4.0-trust.iso*, выполнив команду:

   ```
   sudo mount /opt/ALDPro-2.4.0-trust.iso /media/cdrom
   ```

   - Создать файл */etc/apt/sources.list.d/trust.list* со следующим содержимым:

   ```
   deb file:///media/cdrom 2.4.0 aldpro
   ```

2. Добавить в файл */etc/apt/preferences.d/aldpro* следующее содержимое:

```
Package: *
Pin: release n=2.4.0
Pin-Priority: 1000
```

3. Выполнить команды:

```
sudo apt update
sudo apt dist-upgrade
```

Особенности установки при использовании Astra Linux версии 1.7.6 и выше

При использовании Astra Linux версией 1.7.6 и выше, установка дополнительных пакетов не требуется, так как реализация доверительных отношений ALD Pro встроена в пакеты системы.

Чтобы настроить модуль **Доверие между доменами ALD Pro**, необходимо:

1. На контроллере домена выполниться следующие действия:

```
sudo apt update
sudo apt install libipa-aldpro
```

2. Добавить в файл `/usr/share/aldpro/updates/92-aldpro-extdom-extop-plugin.update` следующую информацию:

```
plugin.update
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
only:changetype: add
only:objectclass: top
only:objectclass: nsSlapdPlugin
only:objectclass: extensibleObject
only:cn: ipa_extdom_extop
only:nsslapd-pluginpath: libaldpro_extdom_extop
only:nsslapd-plugininitfunc: ipa_extdom_init
only:nsslapd-plugintype: extendedop
only:nsslapd-pluginenabled: on
only:nsslapd-pluginid: ipa_extdom_extop
only:nsslapd-pluginversion: 1.0
only:nsslapd-pluginvendor: AstraLinux
only:nsslapd-plugindescription: Support resolving IDs in trusted
domains to names and back
only:nsslapd-plugin-depends-on-type: database
only:nsslapd-basedn: $SUFFIX
```

3. Выполнить команды

```
sudo ipa-ldap-updater /usr/share/aldpro/updates/92-aldpro-extdom-extop-plugin.
↪update
sudo net conf setparm global "passdb backend" "aldprosam:ldapi://%2fvar%2frun
↪%2fslapd-$(hostname -d | sed's/\./-/g; s/.*/\U&/').socket"
sudo aldproctl restart
```

Перечень обновляемых пакетов:

- libipa-hbac-dev
- libipa-hbac0
- libnss-sss
- libpam-sss
- libsss-certmap-dev
- libsss-certmap0
- libsss-idmap-dev
- libsss-idmap0
- libsss-nss-idmap-dev
- libsss-nss-idmap0
- libsss-simpleifp-dev
- libsss-simpleifp0
- libsss-sudo
- libwbclient-sssd
- libwbclient-sssd-dev
- python3-libipa-hbac
- python3-libsss-nss-idmap
- python3-sss
- sssd
- sssd-ad
- sssd-ad-common
- sssd-common
- sssd-dbus
- sssd-ipa

- sssd-kcm
- sssd-krb5
- sssd-krb5-common
- sssd-ldap
- sssd-proxy
- sssd-tools

2.1.3. Проверка работы портала на контроллере

Для доступа на Портал управления необходимо открыть на контроллере домена браузер Mozilla Firefox, адрес портала будет установлен страницей по умолчанию: https://dc-1.ald.company.lan (подробнее о политике, заданной для браузера Firefox, см. раздел *Безопасный обмен данными с применением SSL/TLS*).

Пример содержимого файла */usr/lib/firefox/distribution/policies.json*:

```
{
   "policies": {
         "BlockAboutAddons": true,
         "BlockAboutConfig": true,
         "Authentication": {
         "SPNEGO": ["ald.company.lan"]
      },
         "Certificates": {
         "ImportEnterpriseRoots": true,
         "Install": ["/etc/ipa/ca.crt"]
      },
         "Homepage": {
         "URL": "https://dc-1.ald.company.lan/",
         "Locked": true,
         "StartPage": "homepage-locked"
      }
   }
}
```

Если зайти на портал управления с контроллера домена и кликнуть по ссылке «Вход с Kerberos», то будет предпринята попытка прозрачной аутентификации по **Kerberos**, но для этого в связке ключей пользователя уже должен быть TGT-билет, наличие которого можно

проверить из окна терминала командой **klist**. С помощью этой же утилиты можно проверить, что аутентификация на портале прошла именно по **Kerberos** - в связке ключей появится сервисный билет на доступ к службе **HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN**. При повторном входе на портал, если cookie не удаляются и срок жизни предыдущей пользовательской сессии еще не истек, доступ к порталу будет предоставлен без повторной аутентификации - в этом случае сервисный билет на доступ к **HTTP-службе** в связке ключей не появится.

Если зайти на портал управления с любого другого компьютера, потребуется согласиться с риском использования самоподписанного сертификата, и **Kerberos** аутентификация работать не будет, потребуется ввести логин и пароль вручную. В релизах до 2.0.0 вместо окна входа использовалось всплывающее окно простой аутентификации (basic auth). При входе с Windows-компьютера логин нужно будет вводить полностью с доменной частью в формате **admin@ald.company.lan**.

В случае выхода из строя первого Контроллера Домена выдача билетов также может прекратиться. Для предотвращения такой ситуации необходимо выполнить действия, описаные в *Решение проблемы с запросами на получение билетов kerberos* - удалить или сделать не работающую ссылку для *winbind_krb5_locator.so*:

```
sudo ln -s /dev/null /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/winbind_
↪krb5_locator.so
```

Затем выполнить `ipactl restart`.

2.1.4. Проверка работы портала на другом компьютере домена

Если необходимо настроить доменную аутентификацию на портале управления с другого компьютера в домене, потребуется в настройках браузера включить домен в список доверенных для Kerberos-аутентификации и установить корневой сертификат домена. Подробнее в разделе «*Безопасный обмен данными с применением SSL/TLS*».

2.1.5. Отключение DNSSEC и настройка глобального перенаправления

Служба **bind9** по умолчанию использует механизм DNSSEC для проверки ответов, но его лучше отключить, т.к. технология все еще не получила широкого распространения, и ошибки в настройках зон могут приводить к невозможности разрешения имен. Для этого в

файле */etc/bind/ipa-options-ext.conf* для параметра `dnssec-validation` рекомендуется задать значение «**no**».

Еще одной особенностью настроек **bind9** по умолчанию является запрет на обработку рекурсивных DNS-запросов от клиентов, находящихся за пределами той же подсети, в которой находится сам DNS-сервер. Сделано это для предотвращения DDoS-атак с DNS-усилением, но эта защита не актуальна для контроллеров домена, которые работают в закрытом периметре, поэтому в файле *ipa-options-ext.conf* рекомендуется задать также значение «**any**» для параметров `allow-recursion` и `allow-query-cache` или определить в файле */etc/bind/ipa-ext.conf* список доверенных сетей.

Изменим настройки **bind** командой:

```
sudo nano /etc/bind/ipa-options-ext.conf
```

Необходимо внести указанные изменения в файл:

```
allow-recursion { any; };
allow-query-cache { any; };
dnssec-validation no;
```

Для применения изменений перезапустить DNS-службу на контроллере домена:

```
sudo systemctl restart bind9-pkcs11.service
```

Если до установки пакетов ALD Pro перенаправление DNS-запросов на **localhost** (**127.0.0.1**) привело бы к отказу в работе механизма разрешения имен, то сейчас этого не произойдет, т.к. в системе работает сервис **bind9**, который выполняет функцию рекурсивного разрешителя имен. **Bind9** сам находит запрашиваемые DNS-записи, последовательно обращаясь ко всем DNS-серверам, обслуживающим зону, начиная с корневой (см. файлы */etc/bind/named.conf.default-zones* и */usr/share/dns/root.hints*).

Добавить настройку глобального перенаправления на вкладке **Роли и службы сайта > Служба разрешения имен > Глобальная конфигурация DNS**, см. *Глобальная конфигурация DNS*. Рекомендуется установить адрес публичного DNS, например, от Яндекс 77.88.8.8, с политикой перенаправления «Только перенаправлять» или «Сначала перенаправлять». Затем нажать кнопку «Сохранить» в правом верхнем углу.

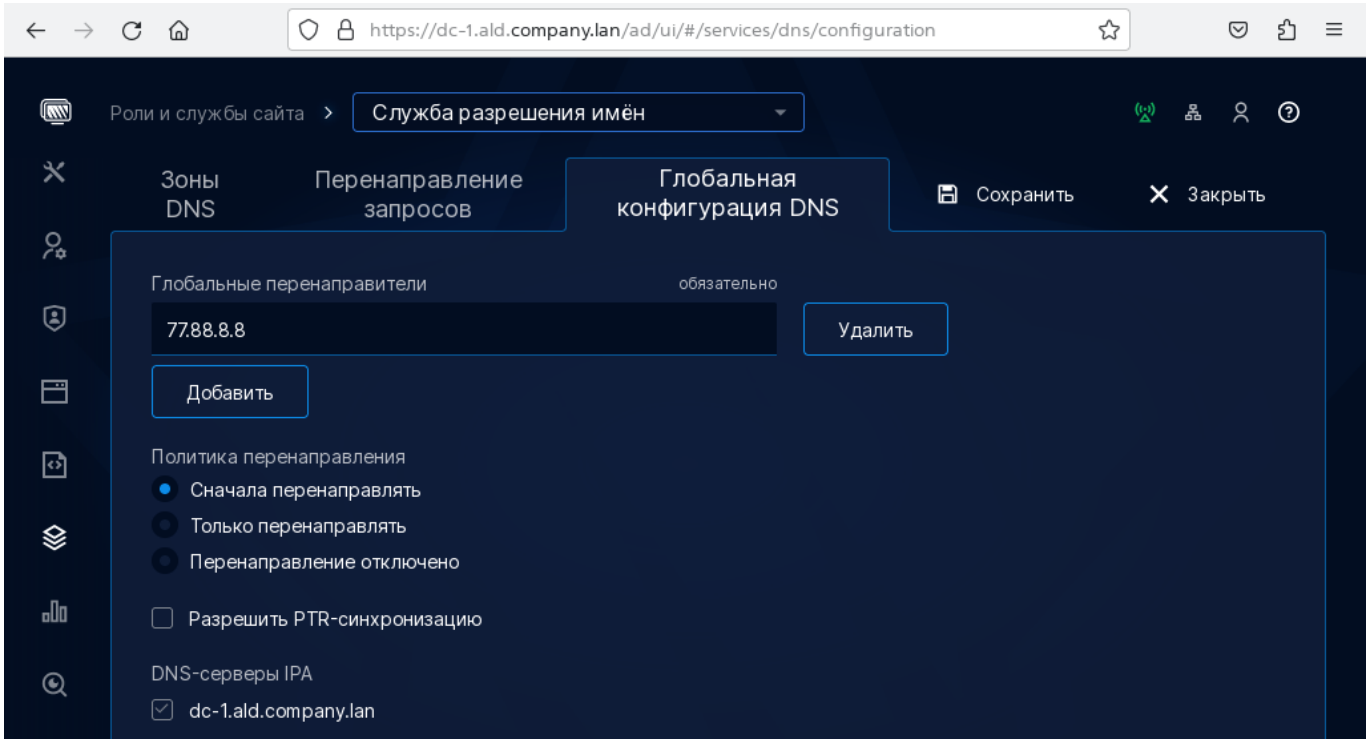


Рисунок 2.1 – Глобальная конфигурация DNS

Проверить настройки DNS-службы можно из командной строки от привелегированного пользователя с высоким уровнем целостности:

```
sudo ipa dnsconfig-show
```

Результат выполнения проверки настроек DNS:

```
[sudo] пароль для admin:
Глобальные перенаправители: 77.88.8.8
Политика перенаправления: first
Разрешить PTR-синхронизацию: FALSE
DNS-серверы IPA: dc-1.ald.company.lan
```

В некоторых инструкциях для проверки DNS предлагают использовать утилиту **dig** с ключом `+trace`, но в этом случае **dig** вместо того, чтобы обратиться к внешнему DNS-серверу, станет выполнять рекурсивные запросы, начиная с зоны верхнего уровня. Поэтому, если нужно увидеть подтверждение, что при разрешении имен запросы пошли к внешнему DNS-серверу, запустите в отдельном окне **tcpdump** для прослушивания пакетов, отправляемых на 53 порт:

```
sudo apt install tcpdump
sudo tcpdump port 53
```

2.2. Ввод компьютера в домен

2.2.1. Необходимые привилегии для ввода компьютера в домен с помощью портала управления ALD Pro

2.2.1.1. Предварительная настройка

Для выполнения предварительной настройки ввода компьютеров в домен с помощью Портала Управления ALD Pro необходимо:

- Создать и настроить сервер **DHCP** для установки ОС по сети (см. **Справочный Центр - Роли службы сайта - Служба динамической настройки узла**).
- Создать и настроить сервер репозиториев ПО, а также репозиторий ПО, откуда будет проходить установка ОС на компьютер (см. **Справочный центр - Установка и обновление ПО - Репозитории ПО**).
- Создать и настроить сервер установки ОС по сети, добавить шаблон компьютера и профиль загрузки (см. **Справочный центр - Автоматизация - Установка ОС по сети**).

Для выполнения настройки вышеуказанных разделов необходимо:

- создать роль,
- привязать ее к корню домена,
- установить признак **Включая дочерние подразделения**,
- сохранить эту роль,
- добавить необходимый набор привилегий для управления всеми разделами портала управления,
- назначить роль на администратора, который будет выполнять предварительную настройку. Данный администратор будет обладать всеми необходимыми привилегиями для выполнения настройки и ввода компьютера в домен.

Привилегии, необходимые для работы с подразделом портала Роли службы сайта - Служба динамической настройки узла

Для создания и настройки нового сервера службы динамической настройки узла (**DHCP**), который будет использоваться в процессе ввода компьютера в домен, необходимы

следующие привилегии:

1. `DHCP Servers - Create` (привязка ко всем сайтам домена), для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

    1.1 `Computers - Read`

    1.2 `DHCP - Read`

    1.3 `DNS Zones - Read`

    1.4 `IPA Servers - Read`

    1.5 `Organization units - Read`

    1.6 `Sites - Read`

2. `DHCP Configuration - Modify` (привязка ко всем сайтам домена), для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

    2.1 `Computers - Read`

    2.2 `DHCP - Read`

    2.3 `DNS Zones - Read`

    2.4 `IPA Servers - Read`

    2.5 `Organization units - Read`

    2.6 `Sites - Read`

Привилегии, необходимые для работы с подразделом портала Установка и обновление ПО - Репозитории ПО

Для создания и настройки нового сервера репозиториев ПО, а также нового репозитория на этом сервере, который будет использоваться в процессе ввода компьютера в домен, необходимы следующие привилегии:

1. `Repository Servers - Create` (привязка ко всем сайтам домена), для корректной работы данной привилегии в роль также необходимо добавить следующие связанные

привилегии:

1.1 Computers - Read

1.2 DNS Zones - Read

1.3 IPA Servers - Read

1.4 Organization units - Read

1.5 Repository Servers - Read

1.6 Sites - Read

2. Repositories - Add, для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

2.1 Computers - Read`

2.2 DNS Zones - Read

2.3 IPA Servers - Read

2.4 Organization units - Read

2.5 Repositories - Read

2.6 Repository Servers - Read

2.7 Sites - Read

3. Repository Versions - Manage (привязка ко всем сайтам домена), опционально при необходимости, для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

3.1 Computers - Read

3.2 DNS Zones - Read

3.3 IPA Servers - Read

3.4 Organization units - Read

3.5 Repositories - Read

3.6 Repository Servers - Read

3.7 Sites - Read

Привилегии, необходимые для работы с подразделом портала Автоматизация - Установка ОС по сети

Для создания и настройки нового сервера установки ОС по сети, который будет использоваться в процессе ввода компьютера в домен, а также для добавления компьютеров и создания новых профилей загрузки, необходимы следующие привилегии:

1. Installation Server Computers - Manage, для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

    1.1 Computers - Delete

    1.2 Computers - Read

    1.3 DNS Zones - Read

    1.4 DHCP - Read

    1.5 Domain Info - Read

    1.6 IPA Servers - Read

    1.7 Installation Server Profiles - Manage

    1.8 Installation Servers - Read

    1.9 Organization units - Read

    1.10 Sites - Read

2. Installation Servers - Create (привязка ко всем сайтам домена), для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

    2.1 Computers - Read

    2.2 DNS Zones - Read

    2.3 DHCP - Read

    2.4 Domain Info - Read

2.5 `IPA Servers - Read`

2.6 `Installation Servers - Read`

2.7 `Organization units - Read`

2.8 `Sites - Read`

3. `Installation Servers - Modify` (при необходимости не только создавать, но и изменять имеющиеся сервера установки ОС), (привязка ко всем сайтам домена), для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

    3.1 `Computers - Read`

    3.2 `DNS Zones - Read`

    3.3 `DHCP - Read`

    3.4 `Domain Info - Read`

    3.5 `IPA Servers - Read`

    3.6 `Installation Servers - Read`

    3.7 `Organization units - Read`

    3.8 `Sites - Read`

Полный набор привилегий, необходимых для настройки возможности ввода компьютера в домен с помощью портала управления ALD Pro

Таким образом, для настройки возможности ввода компьютера в домен необходимо создать роль с приведенным ниже набором привилегий и назначить эту роль на того администратора, который будет выполнять настройку:

1. `DHCP Servers - Create` (привязка ко всем сайтам домена)
2. `DHCP Configuration - Modify` (привязка ко всем сайтам домена)
3. `Repository Servers - Create` (привязка ко всем сайтам домена)
4. `Repositories - Add`

5. Repository Versions - Manage (привязка ко всем сайтам домена)
6. Installation Server Computers - Manage
7. Installation Servers - Create (привязка ко всем сайтам домена)
8. Computers - Delete (связанная)
9. Computers - Read (связанная)
10. DHCP - Read (связанная)
11. DNS Zones - Read (связанная)
12. IPA Servers - Read (связанная)
13. Domain Info - Read (связанная)
14. Organization units - Read (связанная)
15. Sites - Read (связанная)
16. Installation Server Profiles - Manage (связанная)
17. Installation Servers - Read (связанная)
18. Repositories - Read (связанная)
19. Repository Servers - Read (связанная)

Настройка возможности ввода компьютеров в домен с помощью ролей для подразделов портала управления

Для обеспечения предварительной настройки возможности ввода компьютера в домен также можно воспользоваться поставляемыми при начальной установке ролями для подразделов Портала Управления ALD Pro.

Для этого администратору необходимо делегировать следующие роли:

1. ALDPRO - DHCP Service Administrators
2. ALDPRO - Software Repositories Administrators
3. ALDPRO - Install OS Service Administrators

Однако необходимо помнить, что при этом администратор будет обладать полными правами на эти подразделы (то есть добавляется возможность удалять и модифицировать все данные во всех подразделах портала, на которые предоставляется доступ, а не только возможностью создания новых данных).

Привилегии, необходимые для ввода компьютера в домен, при условии, что все настройки сделаны предварительно

В том случае, если все необходимые настройки для ввода компьютера в домен были предварительно проведены одним администратором, то набор привилегий, необходимых администратору исключительно для ввода компьютера в домен без возможности модификации данных в подразделах портала, не связанных с установкой ОС по сети, может быть сокращен до следующего набора привилегий:

1. `Installation Server Computers - Manage`, для корректной работы данной привилегии в роль также необходимо добавить следующие связанные привилегии:

    1.1 `Computers - Delete`

    1.2 `Computers - Read`

    1.3 `DNS Zones - Read`

    1.4 `DHCP - Read`

    1.5 `Domain Info - Read`

    1.6 `IPA Servers - Read`

    1.7 `Installation Server Profiles - Manage`

    1.8 `Installation Servers - Read`

    1.9 `Organization units - Read`

    1.10 `Sites - Read`

2.2.2. Настройка сети на клиентских компьютерах

На пользовательских компьютерах настройка сети выполняется через стандартную службу **NetworkManager**. В реальной инфраструктуре для настройки пользовательских

компьютеров используется DHCP, но в рамках данной инструкции с целью упрощения компьютеру будет назначен статический адрес.

На вкладке «Параметры IPv4» установите следующие значения, см. *Настройки сети с помощью графической утилиты NetworkManager*:

- Метод: Вручную
- Адрес: 10.0.1.51
- Маска: 255.255.255.0
- Шлюз: 10.0.1.1 шлюз маршрутизатора или gateway
- Серверы DNS: 10.0.1.11 (адрес dc-1)
- Поисковый домен: ald.company.lan

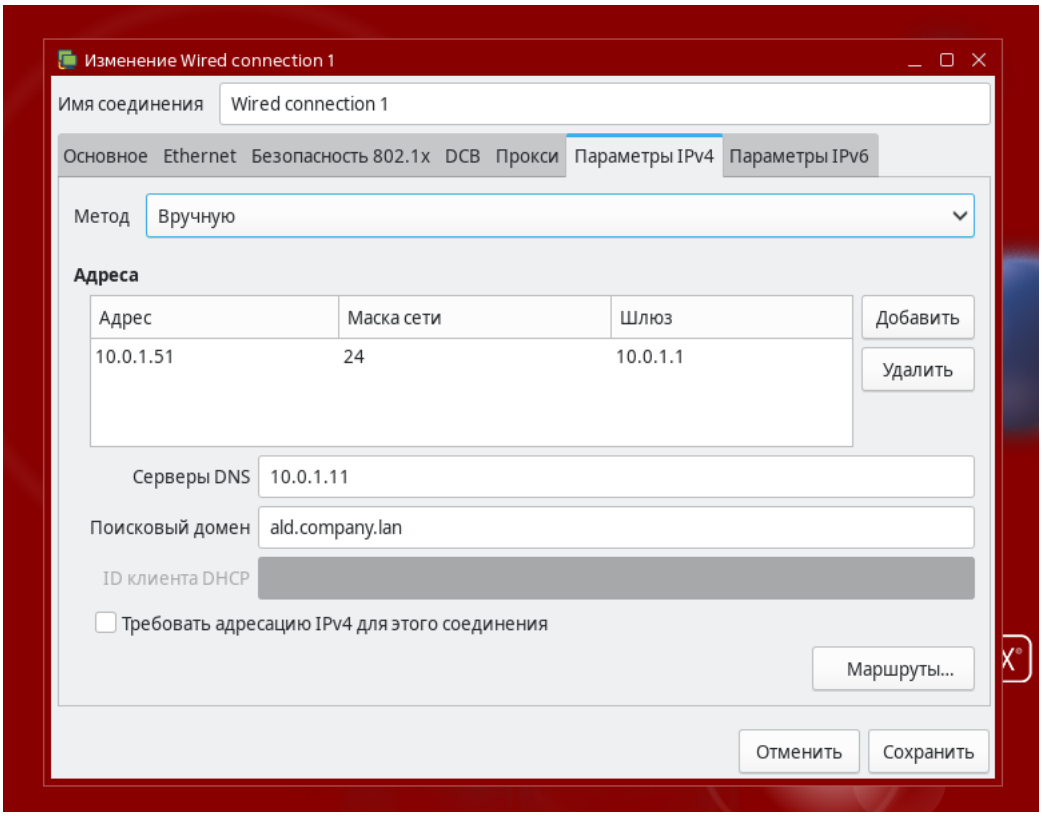


Рисунок 2.2 – Настройки сети с помощью графической утилиты NetworkManager

Для установки пакетов компьютеру нужно иметь доступ к репозиториям, расположенным в сети Интернет по адресу https://dl.astralinux.ru. Можете проверить примененный IP, а также доступность сайта и хостов сети командами:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
```

(продолжение на следующей странице)

```
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

2.2.3. Настройка доступных репозиториев

Для установки клиентской части ALD Pro версии 2.4.0 на ALSE 1.7.6 и выше из официальных интернет-репозиториев РусБИТех-Астра содержание файла */etc/apt/soureces.list* должно быть таким же, как при установке серверной части:

```
sudo nano /etc/apt/sources.list
```

Файл *sources.list* с таким содержимым:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-main/ 1.
↪7_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-update/
↪ 1.7_x86-64 main contrib non-free
```

Кроме того, содержимое может иметь такой вид:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/ 1.
↪7_x86-64 main contrib non-free
```

А также создать отдельный список для */etc/apt/sources.list.d/aldpro.list*:

```
sudo nano /etc/apt/sources.list.d/aldpro.list
```

Вставить в этот файл содержимое:

```
deb https://download.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main
↪base
```

Обновить индекс и проверить, нет ли пакетов, доступных для обновления. Обновить систему, если таковые будут обнаружены, командами:

```
sudo apt update
sudo apt list --upgradable
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confnew
```

2.2.4. Настройка доступных репозиториев для 1.8.1

Для установки клиентской части ALD Pro версии 2.4.0 на ALSE 1.8.1 и выше из официальных интернет-репозиториев РусБИТех-Астра содержание файла */etc/apt/sources.list* должно быть таким же, как при установке серверной части:

```
sudo nano /etc/apt/sources.list
```

Файл sources.list с таким содержимым:

```
deb https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.1/repository-main/ 1.
↪8_x86-64 main contrib non-free
```

Кроме того, содержимое может быть таким:

```
deb https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.1/repository-main/ 1.
↪8_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.1/repository-
↪extended/ 1.8_x86-64 main contrib non-free
```

А также создать отдельный список для */etc/apt/sources.list.d/aldpro.list*:

```
sudo nano /etc/apt/sources.list.d/aldpro.list
```

Вставить в этот файл содержимое:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.8_x86-64 main base
```

Обновить индекс и проверить, нет ли пакетов, доступных для обновления. Обновить систему, если таковые будут обнаружены, командами:

```
sudo apt update
sudo apt list --upgradable
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confnew
```

2.2.5. Установка пакетов на клиентский компьютер

Теперь система готова к установке клиентской части ALD Pro, для этого необходимо выполнить команду:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Комментарии к использованным ключам можно найти в разделе инструкции по установке пакетов на контроллере домена.

Если перезагружать пользовательский компьютер сейчас, то в сообщениях ядра можно будет увидеть ошибки запуска **SSSD** и зависящих от нее служб (журнал загрузки можно найти в файле **/var/log/boot.log**). Это происходит по причине того, что служба еще не настроена соответствующим образом (журнал службы **sssd** можно найти в файле **/var/log/sssd/sssd.log**).

При установке клиента в системе устанавливается более 130 зависимостей, отдельного внимания из которых заслуживают **freeipa**, **sssd**, **krb5**, **ldap-utils**, **chrony**, **salt**, **zabbix-agent**, **syslog-ng** и **td-agent**

2.2.6. Выполнить ввод компьютера в домен

Для успешного ввода компьютера в домен требуется несколько условий:

- у компьютера должно быть задано уникальное имя, которое еще не используется в домене.
- в качестве DNS-сервера должен быть указан IP-адрес контроллера домена.
- установлен пакет клиентского программного обеспечения **aldpro-client**.

Например, имя компьютера **pc-1**. Для начала необходимо проверить уникальность имени в домене **ald.company.lan** можно командой `nslookup`:

```
nslookup pc-1
```

С помощью данной команды проверяется, что хост с указанным именем не найден на DNS-сервере. Данная команда проверит не только имя **pc-1**, но и **pc-1.ald.company.lan**, т.к. в настройках **NetworkManager** на предыдущем шаге указан DNS-суффикс **ald.company.lan**. Вместо фиксированного имени компьютера **pc-1** можно использовать переменную **$pc_name**, значение которой можно сгенерировать случайным образом:

```
pc_name="pc-$(expr $RANDOM$(date +%s) | md5sum | head -c 11)"
echo $pc_name
nslookup $pc_name
```

Все готово для ввода компьютера в домен. Выполнить ввод **pc-1** можно с помощью команд:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.
↪lan --account admin --password 'AstraLinux_176' --host pc-1 --gui --force
set -o history
```

Комментарии по использованным ключам **aldpro-client-installer**:

- `--domain` — имя домена, которое вы выбрали на основе третьего уровня приобретённого домена, например, `ald.company.lan`
- `--account` — логин администратора домена
- `--password` — пароль администратора домена
- `--host` — имя компьютера в нижнем регистре
- `--gui` — использовать интерактивный режим
- `--force` — продолжить ввод компьютера в домен, даже если в домене для его имени уже есть учетная запись. Требуется в тех случаях, когда администратор переустанавливает операционную систему и хочет ввести компьютер в домен с тем же именем.

Описание параметров скрипта можно получить с помощью ключа `-h`. А также доступны короткие псевдонимы, например, в место `--domain` можно указать `-c`, но эти сокращения могут быть менее запоминающимися.

Для сокращения потребления памяти и увеличения скорости запуска службы каталога можно отключить плагин **Schema Compatibility** командой:

```
sudo ipa-compat-manage disable
```

Если запустить утилиту **aldpro-client-installer** без параметров, то появится окно для ввода необходимых параметров из графики, см. *Графическая утилита для ввода компьютера в домен*.

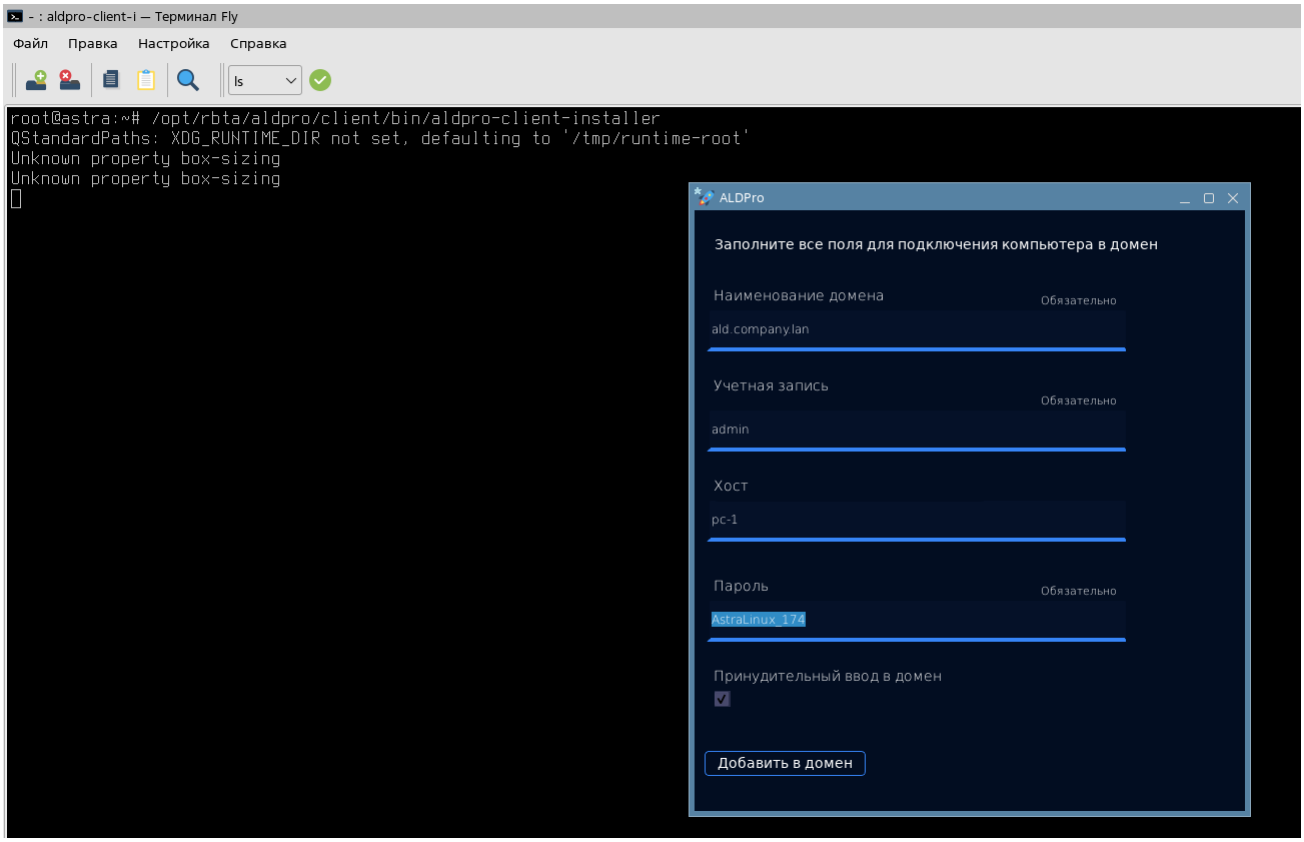


Рисунок 2.3 – Графическая утилита для ввода компьютера в домен

В обоих случаях вызывается **ipa-getkeytab**, которая устанавливает Kerberos-пароль для учетной записи компьютера, используя привилегированную учетную запись. В небольших компаниях для этого используют учетную запись **admin** или включают персональную учетную запись сотрудника в группу **admins**, но это не подходит для крупных организаций с большим штатом ИТ специалистов, где эту задачу поручают сотрудникам, не имеющим отношения к администрированию серверной группировки. В этом случае рекомендуется делегировать сотрудникам только те права, которые действительно необходимы для выполнения поставленной задачи, что можно сделать через настройку прав доступа службы каталога.

Права доступа в службе каталога **FreeIPA** назначаются с помощью трехуровневой модели безопасности, которая включает в себя **Роли** (**Roles**), **Привилегии** (**Privileges**) и **Разрешения** (**Permissions**). На базовом уровне разрешениям соответствуют инструкции управления доступом **389 Directory Server** (**Access Control Instructions, ACI**), с помощью которых можно предоставить доступ на чтение, запись, поиск и другие действия применительно ко всему каталогу, его ветке или конкретной записи. Разрешения группируются в привилегии, привилегии группируются в роли, а роли уже назначаются пользователям.

В домене ALD Pro уже есть роль **Enrollment Administrator**, которая включает привилегию

Host Enrollment, объединяющую почти все необходимые разрешения за исключением права на создание хостов **System: Add Hosts**, поэтому самым простым способом решения задачи делегирования является расширение списка разрешений и назначение роли **Enrollment Administrator** соответствующему пользователю.

```
ipa privilege-add-permission 'Host Enrollment' --permissions='System: Add
↪Hosts'
ipa role-add-member 'Enrollment Administrator' --users=enrolladmin
```

При необходимости можно создать и новую роль **New Host Enrollment**:

```
ipa privilege-add 'New Host Enrollment' --desc='Ввод новых хостов в домен'

ipa privilege-add-permission 'New Host Enrollment' \
--permissions='System: Add Hosts' \
--permissions='System: Add krbPrincipalName to a Host' \
--permissions='System: Enroll a Host' \
--permissions='System: Manage Host Certificates' \
--permissions='System: Manage Host Enrollment Password' \
--permissions='System: Manage Host Keytab' \
--permissions='System: Manage Host Principals'

ipa role-add 'New Host Enrollment Administrator' \
--desc='Участники роли имеют привилегии, необходимые для регистрации новых
↪компьютеров в домене'

ipa role-add-privilege 'New Host Enrollment Administrator' \
--privileges='New Host Enrollment'

ipa role-add-member 'New Host Enrollment Administrator' --users=enrolladmin
```

После создания новой роли продвижения можно назначить её через веб-интерфейс на странице **Управление доменом > Роли и права доступа > Роли в системе**, см. *Редактор новой роли для продвижения*.

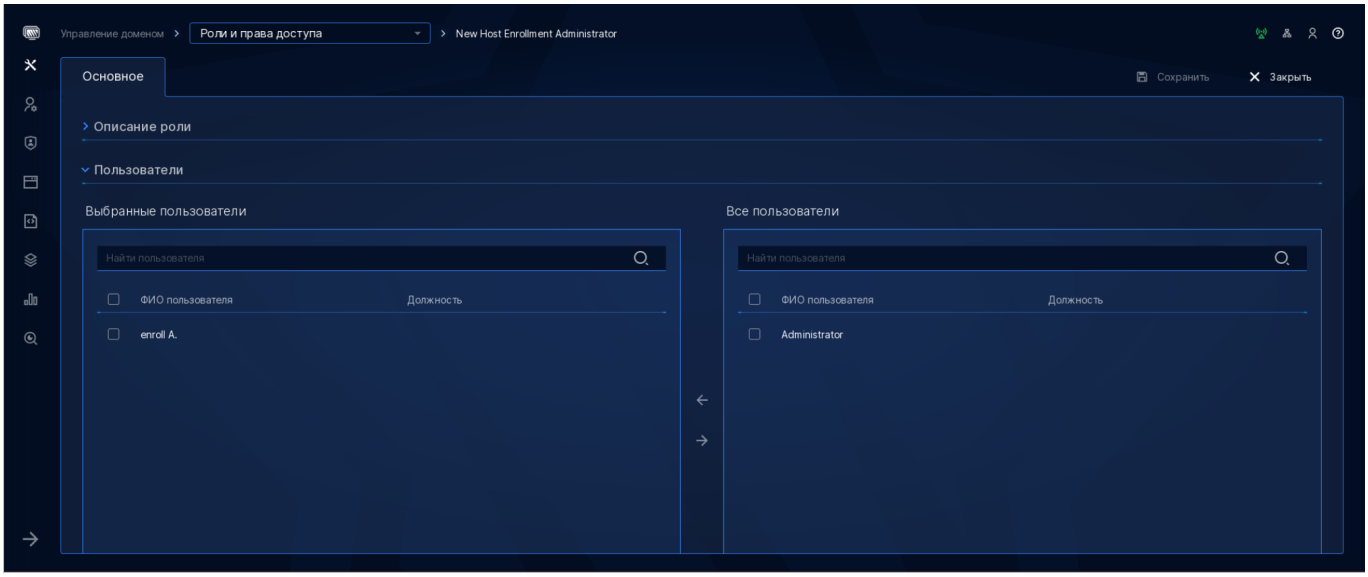


Рисунок 2.4 – Редактор новой роли для продвижения

После назначения роли **New Host Enrollment Administrator** пользователь сможет вводить машины в домен, не получая при этом полных административных прав.

Ранее нужно было устанавливать имя узла до его ввода в домен, но в последних редакциях установщика этого не требуется, скрипт сам изменит `hostname` в системе:

```
exec bash
echo $HOSTNAME
```

Результат выполнения проверки:

```
pc-1.ald.company.lan
```

Для применения всех настроек необходимо выполнить перезагрузку компьютера:

```
sudo reboot
```

После перезагрузки необходимо войти в систему, используя доменную учетную запись администратора с паролем из строки продвижения сервера:

- login: admin
- password: ****** (пароль администратора домена)

Для первого входа в систему доменной учетной записью требуется доступ к контроллеру домена. В дальнейшем аутентификация пользователя возможна через кэш **sssd** службы.

2.2.7. Присоединение к домену с помощью одноразового пароля

Чтобы при вводе компьютеров домен уменьшить риск разглашения привилегированных учетных данных, в службе каталога **FreeIPA** предусмотрена возможность установки Kerberos-пароля с помощью одноразового LDAP-пароля компьютера. Для создания учетной записи и установки одноразового пароля привилегии администратора домена потребуются, но все эти действия можно выполнить на контроллере домена заранее, в том числе с использованием инструментов автоматизации, а на стороне рабочей станции в дальнейшем сотруднику потребуются привилегии только локального администратора.

Одноразовые пароли в настоящее время можно использовать только с дефолтным клиентом **FreeIPA**, но полная поддержка этого сценария в клиентской части ALD Pro планируется уже в ближайших релизах, т.к. этот способ обладает несколькими преимуществами:

- Учетная запись хоста может быть создана сразу в правильном организационном подразделении, чтобы на него распространялось действие необходимых групповых политик.
- Учетную запись хоста можно сразу включить во все группы, чтобы на него распространялись необходимые правила HBAC и SUDO.
- Для ввода машины в домен на стороне рабочей станции не нужно будет использовать пароль привилегированной учетной записи.

Продемонстрируем работу данного сценария на примере утилиты **ipa-client-install**. Сначала нужно создать учетную запись компьютера с помощью команды `ipa host-add`, используя ключ `--random` для генерации одноразового пароля:

```
kinit admin
ipa host-add pc-2.ald.company.lan --random --ip-address=10.0.1.52 --force
```

Результат выполнения:

```
--------------------------------------
Добавлен узел "pc-2.ald.company.lan"
--------------------------------------
Имя узла: pc-2.ald.company.lan
Случайный пароль: 2Hy0bL8abgQ2IOddYwl3TOc
Link to department:
ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
Пароль: True
Таблица ключей: False
Managed by: pc-2.ald.company.lan
```

Где:

- `pc-2.ald.company.lan` – FQDN имя компьютера в нижнем регистре
- `--random` – ключ, указывающий на требование сгенерировать случайный одноразовый пароль
- `--ip-address` – адрес компьютера для создания DNS записи в домене
- `--force` – ключ, позволяющий принудительно установить значение имени узла, даже если такое имя уже присутствует в DNS

На стороне компьютера останется назначить хосту правильное имя и выполнить команду **ipa-client-install**, используя одноразовый пароль, полученный на предыдущем шаге:

```
sudo hostnamectl set-hostname pc-1.ald.company.lan
sudo ipa-client-install --mkhomedir --password='2Hy0bL8abgQ2IOddYwl3TOc' -U
```

2.2.8. Повторное присоединение к домену, используя keytab-файл хоста

Если администратору нужно переустановить операционную систему на доменном компьютере, то утилита **ipa-client-install** позволяет для авторизации повторного ввода машины в домен с тем же именем воспользоваться предыдущим *krb5.keytab*-файлом.

При этом она не просто скопирует файл, а сбросит хосту пароль и запишет новые ключи, поэтому версия ключей изменится:

```
sudo klist -k /home/localadmin/krb5.keytab
```

Результат выполнения:

```
Keytab name: FILE:/home/localadmin/krb5.keytab
KVNO Principal
----------------------------------------------------------------------------
 ↪
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
  1 host/pc-1.ald.company.lan@ALD.COMPANY.LAN
  1 host/pc-1.ald.company.lan@ALD.COMPANY.LAN
```

Необходимо установить имя хоста и клиент с keytab-файлом:

```
sudo hostnamectl set-hostname pc-1.ald.company.lan
sudo ipa-client-install -U -k /home/localadmin/krb5.keytab
```

Результат выполнения:

```
...
The ipa-client-install command was successful
root@pc-1:~# klist -k /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
----------------------------------------------------------------------------
 ↪
2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN
2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN
```

Для применения всех настроек необходимо выполнить перезагрузку компьютера:

```
sudo reboot
```

2.2.9. Проверка работы синхронизации времени

Вопрос синхронизации времени требует отдельного рассмотрения, так как для работы протокола проверки подлинности Kerberos необходимо, чтобы время на клиенте и на сервере расходилось не более, чем на 5 минут.

По умолчанию в Astra Linux синхронизация времени отключена, но в некоторых виртуальных средах машины берут время из хостовой операционной системы во время загрузки после полного выключения, поэтому отсутствие синхронизации времени можно заметить только при работе с горячими снимками, которые были сделаны во время работы операционной системы.

При установке ALD Pro (как клиентской, так и серверной части) в системе появляется служба **chrony**, содержание конфигурационного файла которой автоматически

редактируется через механизм групповых политик в соответствии с текущими настройками домена «Роли и службы сайта \ Служба синхронизации времени». Пользовательские компьютеры синхронизируют время с контроллером, а контроллер берет его у публичных серверов, см. *Настройки синхронизации даты и времени*.

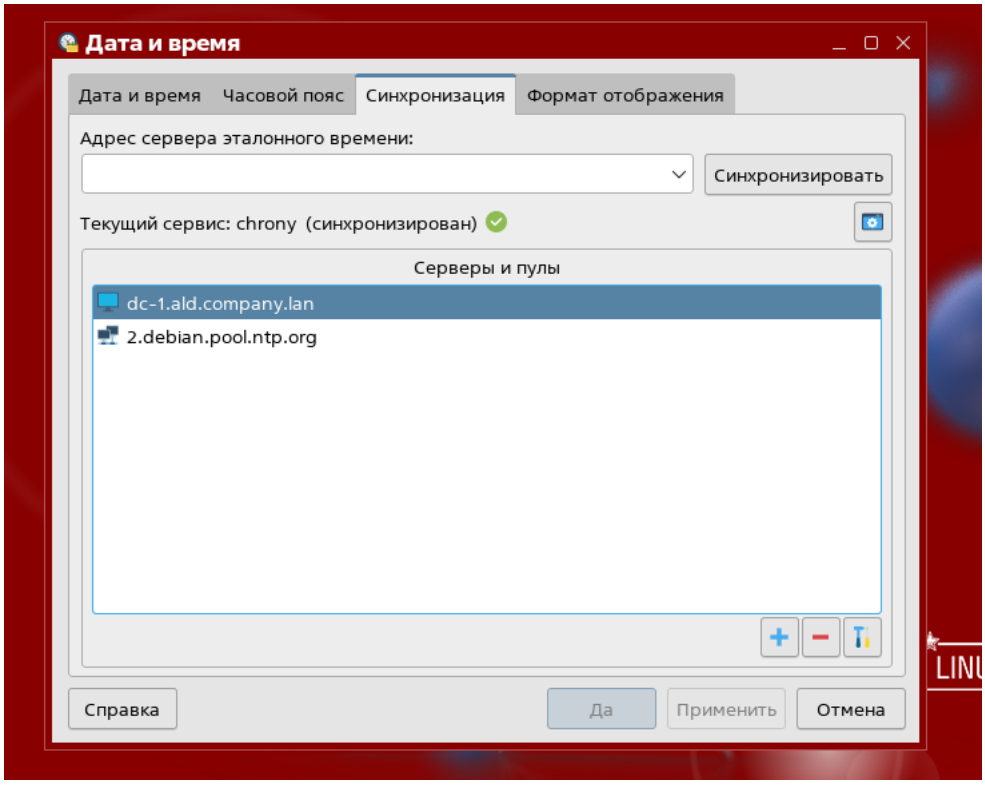


Рисунок 2.5 – Настройки синхронизации даты и времени

Текущие настройки службы синхронизации времени на хосте можно посмотреть в файле *chrony.conf*:

```
cat /etc/chrony/chrony.conf
```

В настройках chrony, которые использует ALD Pro, указан параметр **rtcsync**, включающий режим, в котором системное время периодически копируется в RTC. Параметр **rtcsync** так же необходим для того, чтобы служба chrony при синхронизации времени сбрасывала флаг STA_UNSYNC, иначе в приложении «Дата и время» на **dc-1** будет оставаться предупреждение об отсутствии синхронизации.

Принудительно обновить содержание конфигурационного файла через механизм групповых политик можно перезапуском службы **aldpro-salt-minion**:

```
systemctl restart aldpro-salt-minion.service
```

Принудительно запустить синхронизацию времени можно перезапуском службы:

```
sudo systemctl restart chrony
```

Текущее состояние синхронизации можно узнать в приложении «Дата и Время» или командой **timedatectl**:

```
sudo timedatectl
```

Для взаимодействия со службой **chronyd** во время ее работы предназначен интерфейс командной строки **chronyc**. Чтобы определить, с какими серверами служба устанавливает соединение, можно выполнить команду sources:

```
sudo chronyc sources -v
```

Результат выполнения команды, см. *Список серверов с какими серверами служба chronyd устанавливает соединение*, где символом звездочки «*****» отмечен сервер, время которого установлено в системе.

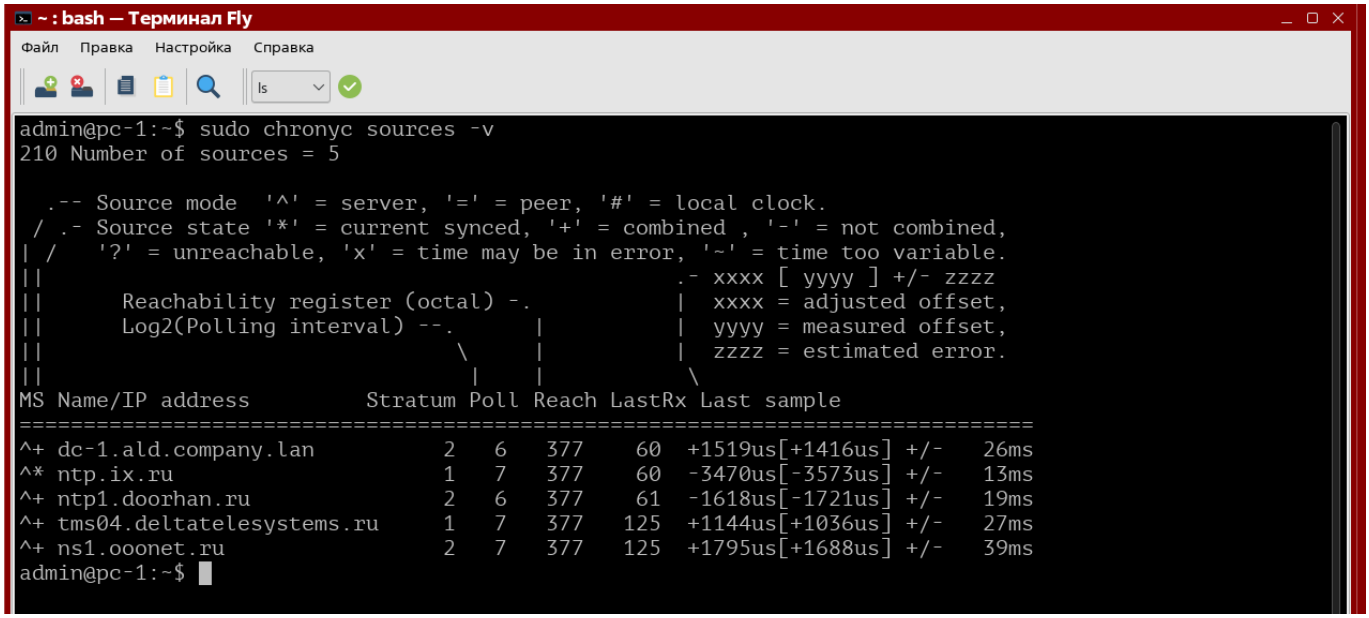


Рисунок 2.6 – Список серверов с какими серверами служба chronyd устанавливает соединение

В настройках **chrony**, которые использует ALD Pro, указан параметр **makestep**, поэтому при выполнении синхронизации компьютер сразу устанавливает требуемое значение. Если отсутствует параметр **makestep**, то служба будет крайне медленно «подтягивать» время к требуемом значению (по несколько секунд в минуту). Форсировать переход к целевому значению в этом случае можно вызовом команды `makestep` через `chronyc`:

```
sudo chronyc makestep
```

Результат команды:

```
200 OK
```

Если требуется проверить работу NTP-сервера, можно воспользоваться командой `ntpdate` с ключом `-q` (**query only**, отправить только запрос без изменения времени). Крайне полезными являются также ключи `-v` и `-d`, включающие подробный вывод (**verbose**) и отладку (**debugging**) соответственно.

```
sudo ntpdate -qvd dc-1.ald.company.lan
```

После синхронизации времени, указанная выше команда **timedatectl** может показать расхождение между системным временем Astra Linux Special Edition (**Universal time**) и значением времени в BIOS (**RTC time, real time clock**), так как запись в BIOS происходит только при выключении компьютера. Записать текущее время системы в BIOS можно утилитой **hwclock** с параметром **systohc**:

```
sudo hwclock --systohc
```

При значительном изменении времени, ранее выданные билеты Kerberos могут оказаться недействительными, поэтому может потребоваться повторно пройти аутентификацию в домене командой **kinit**:

```
kinit
```

Информацию о выданных билетах можно увидеть командой **klist**:

```
klist
```

Результат выполнения:

```
Ticket cache: KEYRING:persistent:287600000:krb_ccache_dsnwzSY
Default principal: admin@ALD.COMPANY.LAN

Valid starting    Expires    Service principal
23.08.2023 11:59:21 24.08.2023 11:59:15
krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

2.3. Вывод компьютера из домена

Вывод компьютера из домена предполагает приведение его операционной системы в исходное состояние и удаление всех записей о хосте из каталога. Это может потребоваться, например, в случаях:

- Администратор допустил опечатку в названии имени хоста;
- Оборудование устарело и нужно его вывести из эксплуатации.

2.3.1. Действия на рабочей станции

В мире Linux очень высока степень повторного использования программного кода, поэтому для правильной работы приложения ему могут требоваться десятки и сотни вспомогательных пакетов.

Для упрощения работы с такими зависимостями предназначены специальные служебные программы, которые называются пакетными менеджерами, но даже они не могут справиться с полным удалением сложного системного программного обеспечения.

Поэтому для приведения системы в исходное состояние правильнее всего восстановить виртуальную машину из резервной копии или выполнить повторную установку операционной системы, однако, в данном разделе представлено несколько рекомендаций, с помощью которых можно максимально приблизиться к желаемому результату без столь кардинальных действий.

На рабочей станции сначала следует сделать откат настройки клиента **FreeIPA**, чтобы привести содержание общих конфигурационных файлов, включая настройки PAM стека, в исходное состояние. Для этого нужно войти в систему учетной записью локального администратора, например, **localadmin** и выполнить команду **astra-freeipa-client** с ключом `-U`:

```
sudo astra-freeipa-client -U
```

Результат выполнения команды:

```
Unenrolling client from IPA server
...
Client uninstall complete.
```

Внимание: Если выполнить команду `sudo astra-freeipa-client -U` из сессии доменного администратора, например, **admin**, то потеряется доступ к командам `sudo` и система HBAC, не найдя учетных записей и сервера с правилами, запретит работать пользователю **admin@ALD.COMPANY.LAN** на этой машине. Поэтому необходимо, чтобы:

- учетная запись локального администратора была активной на момент вывода компьютера из домена;
- был выполнен вход в сессию под учетной записью локального администратора.

При выполнении команды в системе происходят следующие действия:

- В службе каталога снимается отметка о том, что хост зарегистрирован в домене;
- Ключи хоста удаляются из файла */etc/krb5.keytab*;
- Содержимое конфигурационных файлов */etc/krb5.conf*, */etc/ldap/ldap.conf* и др. приводится в исходное состояние;
- К имени файла конфигурации */etc/sssd/sssd.conf* добавляется суффикс *.deleted*.

Далее следует удалить пакеты приложений, указав явно **aldpro**, **freeipa**, **sssd** и **krb5**. Все другие пакеты будут удалены через зависимости.

```
sudo apt purge 'aldpro*' 'freeipa*' 'sssd*' 'krb5*'
sudo apt autoremove --purge
```

Внимание: Перед удалением данных необходимо убедиться, что пользователи сохранили свои важные файлы. Процедура восстановления очень сложная и порой не все данные удается восстановить. А также необходимо сохранить папку локального администратора, который был создан при установке системы.

Установить сетевые настройки с учетом нового сценария использования рабочей станции.

Проверить список профилей в домашней папке:

```
cd /home/ && ls
```

В результате отображаются профили нескольких пользователей, один из которых

локальный администратор, например, **localadmin**, который был указан при установке:

```
admin localadmin ivani ppetr
```

Теперь нужно удалить домашние директории доменных пользователей и оставшиеся в системе артефакты. Файлы в этих директориях создаются в ходе выполнения скриптов, поэтому пакетный менеджер о них ничего не знает и не может взять на себя ответственность за их удаление.

```
sudo rm -rf /home/admin
```

Также можно использовать **bash** скрипт для рекурсивного удаления каталогов, где переменная `$adm` содержит имя локального администратора:

```
adm=localadmin
ls -1a | grep -v "^$adm\|\.$" | while read line; do echo "removing ./$line";
sudo rm -rf "./$line"; done
```

В результате выполнения скрипта будут удалены все каталоги пользователей, кроме **localadmin**:

```
removing ./admin
removing ./ivani
removing ./ppetr
```

Также необходимо удалить каталоги клиентских приложений подсистем ALD Pro:

```
sudo rm -rf /var/lib/sss/pubconf/krb5.include.d/
sudo rm -rf /etc/krb5.conf.d/
sudo rm -rf /var/lib/sss/
sudo rm -rf /etc/sssd/
sudo rm -rf /opt/rbta/aldpro/
sudo rm -rf /etc/syslog-ng/aldpro/
sudo rm -rf /srv/aldpro-salt/minion.d/
sudo rm -rf /srv/aldpro-salt/config/
sudo rm -rf /var/lib/certmonger/
```

Для удаления корневого сертификата домена из файла */etc/ssl/certs/ca-certificates.crt* следует вызвать утилиту **update-ca-certificates**:

```
sudo update-ca-certificates
```

В завершение действий над выводимым компьютером необходимо вернуть компьютеру исходное имя хоста, например, **astra**, и перезагрузить его:

```
sudo hostnamectl set-hostname astra
sudo reboot
```

2.3.2. Действия на контроллере домена

На сервере **dc-1** нужно удалить DNS-записи хоста, а затем и сам хост.

```
kinit admin
ipa dnsrecord-del ald.company.lan. pc-1 --del-all
```

Результат удаления записей DNS:

```
---------------------
Удалена запись "pc-1"
---------------------
```

Удалить выводимый компьютер из базы системы **FreeIPA**:

```
ipa host-del pc-1.ald.company.lan
```

Результат удаления компьютера:

```
----------------------------------
Удалён узел "pc-1.ald.company.lan"
----------------------------------
```

В завершение на каждом контроллере нужно выполнить команду для удаления **Salt** ключа компьютера, чтобы исключить попытки обращения со стороны **Salt Master** к этой рабочей станции при обработке широковещательных запросов:

```
sudo salt-key -y -d pc-1.ald.company.lan
```

Результат удаления ключей **pc-1.ald.company.lan**:

```
[sudo] пароль для admin:
The following keys are going to be deleted:
Accepted Keys:
pc-1.ald.company.lan
[INFO ] Rotating AES key
Key for minion pc-1.ald.company.lan deleted.
```

Удалить ключ с резервного контроллера домена **dc-2**:

```
ssh dc-2
sudo salt-key -y -d pc-1.ald.company.lan
```

Результат удаления ключей **pc-1.ald.company.lan** на **dc-2**:

```
[sudo] пароль для admin:
The following keys are going to be deleted:
Accepted Keys:
pc-1.ald.company.lan
[INFO ] Rotating AES key
Key for minion pc-1.ald.company.lan deleted.
```

2.4. Механизм репликации FreeIPA

Репликация — это процесс, в ходе которого содержимое каталога синхронизируется между серверами, за счет чего достигается целостность данных или так называемая конвергентность. В службе каталога **FreeIPA** используется протокол «DS 5.0 multi-supplier incremental replication protocol», в котором учитываются отдельные положения (**RFC3384**).

Каждая Реплика FreeIPA является Мастером, т.е. доступна на запись, и с ее помощью можно вносить изменения в каталог. Разрешение конфликтов основано на использовании меток времени, поэтому для корректной работы механизма крайне важно, чтобы время было синхронизировано между всеми серверами и ни в коем случае не уходило вперед/назад более, чем на сутки, иначе для устранения проблемы искажения времени (too much time skew) может потребовать восстановления контроллеров из резервной копии.

2.4.1. Соглашения о репликации

На низком уровне репликация проходит по модели **Ведущий-Ведомый** (**Master-Slave**), в рамках которой Сервер, передающий изменения, называется **Поставщиком**, а Сервер, принимающий изменения, называется **Потребителем**. По итогам репликации на **Потребителе** формируется полная копия данных каталога, поэтому его также называют **Репликой**.

Для работы механизма репликации требуется заранее определить топологию домена через создание **Сегментов** и **Соглашений о репликации**. **Сегмент топологии** (**Topology segment**) определяет связь между двумя узлами и является информацией, которая реплицируется в домене между всеми контроллерами, а **Соглашения о репликации** хранятся только на **Поставщике** и содержат настройки для подключения к **Потребителю**. Таким образом, на каждый двусторонний **Сегмент** приходится по два **Соглашения**.

Для управления топологией на портале управления нужно перейти к странице **Управление доменом > Сайты и службы > Соглашения о репликации**, см. *Соглашения о репликации контроллера домена dc-1 с резервным контроллером dc-2*.

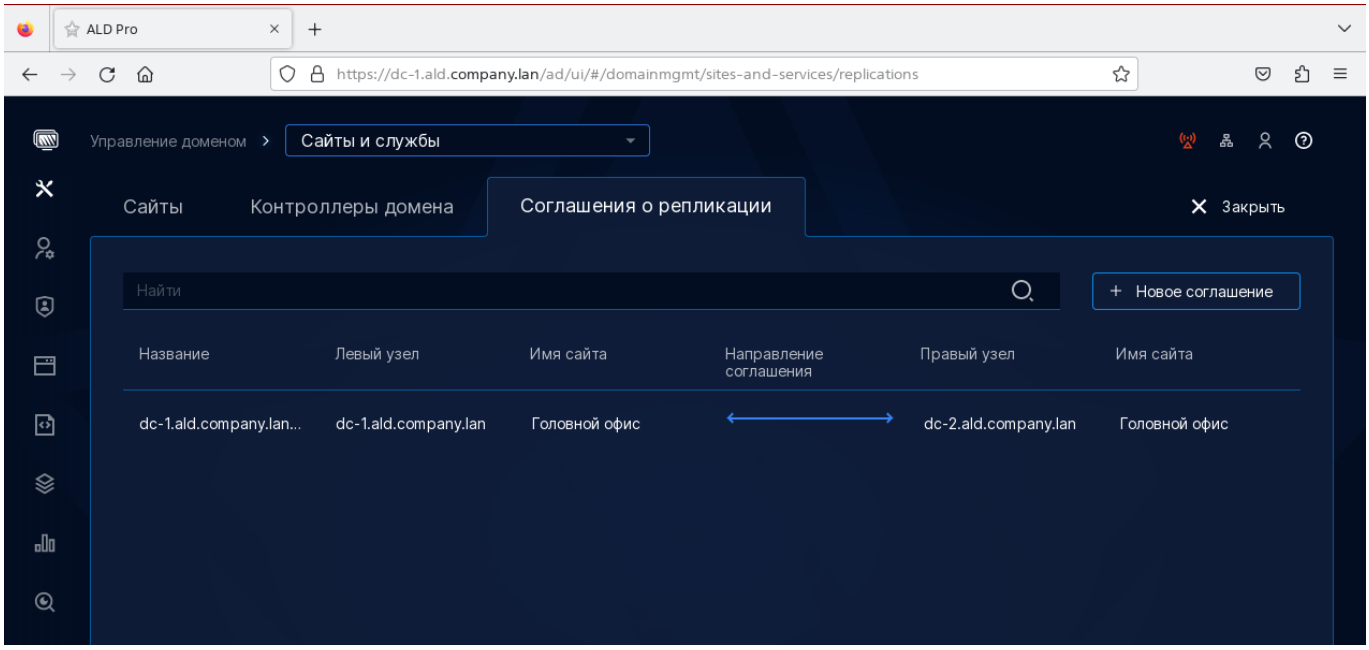


Рисунок 2.7 – Соглашения о репликации контроллера домена **dc-1** с резервным контроллером **dc-2**

В интерфейсе сущности названы **Соглашениями**, но, по сути, это **Сегменты топологии**. Соответствующие **Соглашения о репликации** будут созданы на каждом из контроллеров домена автоматически, как только ими будет получена информация о соответствующем **Сегменте топологии**, за что отвечает плагин топологии (IPA Topology Configuration).

2.4.2. Порядок решения конфликтов репликации

Репликация всегда инициируется **Поставщиком**, а не **Потребителем** (т.е. работает по модели **Push**). Алгоритм основан на использовании журнала изменений, который хранится в отдельной базе данных, для ALSE до версии 1.7.4 см. файл «/var/lib/dirsrv/slapd-ALD-COMPANY-LOCAL/cldb/*.db», для ALSE 1.7.4 и выше см. файл */var/lib/dirsrv/slapd-ALD-COMPANY-LOCAL/db/userRoot/replication_changelog.db*.

Открыть файл можно командой **dbscan -f**:

```
dbscan -f
/var/lib/dirsrv/slapd-ALD-COMPANY-LOCAL/db/userRoot/replication_change
log.db
```

В сеансе репликации **Поставщик** связывается с **Потребителем**, запрашивает у того сводную информацию о состоянии его репликации (так называемую таблицу RUV, Replica Update Vector), чтобы определить, есть ли у него для **Потребителя** более свежие данные, которые можно было бы передать, и запускает передачу данных. В случае изменения одного и того же атрибута на двух разных контроллерах домена, при репликации приоритет будет выше у последнего изменения по времени.

В случае, когда на нескольких контроллерах одновременно создаются записи с одинаковыми DN, автоматический механизм разрешения конфликтов оставит только ту запись, которая была создана раньше других по метке времени. Остальные записи будут переименованы и скрыты. К имени записи добавляется атрибут `nsuniqueid`, например, дубликат учетной записи пользователя **ivan.kuznetsov** будет называться `nsuniqueid=7341f021-20b331ee-a3ef96ae-a91d3705+uid=ivan.kuznetsov, cn=accounts,dc=ald,dc=company,dc=lan`.

Дополнительно таким записям назначается атрибут `nsds5ReplConflict`, чтобы их было легко найти в каталоге:

```
ldapsearch -H ldap://localhost:389 -x -D "cn=Directory Manager" -W -b
↪"cn=users,cn=accounts,dc=ald,dc=company,dc=local" -s one -a always -z 1000
↪"(|(objectClass=subentry)(objectClass=ldapSubentry))" "hasSubordinates"
↪"objectClass"
```

Если создать пользователя на одном сервере, а на другом сервере удалить структурное подразделение, в котором этот пользователь был создан, то на уровне репликации LDAP конфликт не произойдёт, т.к. отношение пользователя к структурному подразделению в

ALD Pro реализуется через атрибут `rbtadp`, а не через связь родительской и дочерней записи. Пользователь не будет отражаться в каком-либо структурном подразделении. Но будет присутствовать в списке всех пользователей.

2.5. Установка резервного контроллера домена

Установка резервного контроллера домена выполняется в два шага:

- Сначала сервер вводится в домен как обычная рабочая станция.
- Затем на портале управления серверу назначается роль контроллера домена, и установка происходит автоматически благодаря системе автоматизации.

2.5.1. Подготовка к установке резервного контроллера

Необходимо подготовить резервный сервер **dc-2.ald.company.lan**, который должен быть с такими же системными параметрами, как первый контроллер домена. Они должны быть равны по объему оперативной памяти и количеству ядер, т.к. реплика будет содержать в базе данных полную копию объектов каталога и должна принять на себя всю нагрузку при отключении основного контроллера.

Как и при установке первого контроллера **dc-1**, нужно убедиться, что на сервере **dc-2** установлена ОС Astra Linux Special Edition 1.7.6 с максимальным уровнем защищенности:

```
cat /etc/astra/build_version
sudo astra-modeswitch getname
```

Сетевые настройки резервного контроллера **dc-2** немного отличаются от первого контроллера, потому что разрешение имен на **dc-2** до продвижения должно выполняться через DNS службу основного контроллера **dc-1**.

Отключить **NetworkManager** на резервном контроллере командами:

```
sudo systemctl stop NetworkManager
sudo systemctl disable NetworkManager
sudo systemctl mask NetworkManager
sudo systemctl status NetworkManager
```

В AstraLinux Special Edition до версии 1.7.4 служба называлась **network-manager**, поэтому нужно было выполнять следующие команды:

```
sudo systemctl stop network-manager
sudo systemctl disable network-manager
sudo systemctl mask network-manager
sudo systemctl status network-manager
```

Дополнительную информацию см. раздел :ref:`network_interface_settings`.

Задать статический адрес и другие параметры сетевому интерфейсу:

```
sudo nano /etc/network/interfaces
```

Пример настройки сети контроллера для **dc-2**:

```
auto lo
iface lo inet loopback

auto eth0
    iface eth0 inet static
    address 10.0.1.12
    netmask 255.255.255.0
    gateway 10.0.1.1
```

Где:

- `10.0.1.12` - это статический IP-адрес резервного контроллера домена **dc-2**

Содержимое файла */etc/resolv.conf* должно быть:

```
nameserver 10.0.1.11
search ald.company.lan
```

Установить запрет на редактирование файла */etc/resolv.conf* с помощью команды:

```
sudo chattr +i /etc/resolv.conf
```

Под успешной установкой резервного контроллера подразумевается наличие статуса «**Успешно**» у соответствующего задания в подразделе **Автоматизация > Задания автоматизации**, а также отсутствие ошибок в отчете о выполнении задания и логе ошибки исполнения в самом задании. По завершении успешной установки резервного

контроллера необходимо снять запрет на редактирование файла */etc/resolv.conf* командой:

```
sudo chattr -i /etc/resolv.conf
```

А также привести файл */etc/resolv.conf* к следующему виду:

```
nameserver 127.0.0.1
search ald.company.lan
```

В случае, если реплика добавляется в новую подсеть, то администратору необходимо вручную добавить обратную зону через портал управления ALD Pro (см. **Руководство пользователя - Роли и службы сайта - Служба разрешения имен - Перенаправление запросов - добавление зоны**) или с помощью следующей команды:

```
ipa dnszone-add --name-from-ip=192.0.2.0/24
```

Автоматическое добавление обратной зоны выключено, как потенциально небезопасный механизм.

Еще крайне важно удалить строку, связывающую имя хоста с адресом **localhost**, т.к. в соответствии с настройками */etc/gai.conf* эти адреса имеют выше приоритет, а крайне важно, чтобы имя хоста разрешалось в локальный адрес, потому что некоторые службы могут прослушивать порты только на этом адресе.

Изменить содержимое файла можно командой:

```
sudo nano /etc/hosts
```

Примерное содержимое файла */etc/hosts* на **dc-2**:

```
127.0.0.1 localhost.localdomain localhost
#127.0.1.1 dc-2 - закомментировать или удалить строку с адресом локальной
 ↪петли
10.0.1.12 dc-2.ald.company.lan dc-2
```

Где

- `10.0.1.12` — статический IP-адрес контроллера домена, который выбран на шаге настройки сетевого интерфейса.

Проверить доступность контроллера домена **dc-1** и сервера репозиториев **dl.astralinux.ru**:

```
ping -c 4 dc-1
ping -c 4 dl.astralinux.ru
```

Проверить, что подключены все репозитории для Astra Linux 1.7.6 и выше и ALD Pro 2.4.0 по тому же принципу как в разделе **Ввод компьютера в домен**:

```
sudo nano /etc/apt/sources.list
```

Содержание этого файла должно быть следующим:

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base 1.7_
 ↪x86-64 main non-free contrib
```

Кроме того, содержание файла может быть следующим:

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-main 1.7_
 ↪x86-64 main non-free contrib
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-update 1.
 ↪7_x86-64 main contrib non-free
```

Дополнительно требуется создать отдельный список для */etc/apt/sources.list.d/aldpro.list* командой:

```
sudo nano /etc/apt/sources.list.d/aldpro.list
```

Добавить в этот файл содержимое:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0/ 1.7_x86-64 main base
```

Обновить списки пакетов и программное обеспечение:

```
sudo apt update
sudo apt list --upgradable
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confnew
```

После обновления выполнить перезагрузку:

```
sudo reboot
```

Установить программное обеспечение клиента ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

После установки пакета **aldpro-client** проверить настройки *resolv.conf* командой:

```
sudo cat /etc/resolv.conf
```

Файл должен иметь следующее содержимое:

```
nameserver 127.0.0.1
nameserver 10.0.1.11
search ald.company.lan
```

Ввести **dc-2** в домен ALD Pro **ald.company.lan** командами:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain
ald.company.lan --account admin --password 'AstraLinux_176' --host dc-2 --
 ↪gui --force
set -o history
```

Для применения новых параметров требуется перезагрузить **dc-2**:

```
sudo reboot
```

2.5.2. Установка и продвижение резервного контроллера через портал управления

Используемая учетная запись **admin** должна обладать достаточными правами и входить в группу **ald trust admin** для установки компонентов, необходимых для использования доверительных отношений с Microsoft AD DS. До версии 2.2.0 пользователя **admin** требовалось добавлять в эту группу вручную.

Добавить **admin** в группу **ald trust admin** командой на контроллере домена **dc-1**:

```
ipa group-add-member 'ald trust admin' --user admin
```

Продвижение резервного Контроллера Домена осуществляется с Портала Управления на странице **Управление доменом > Сайты и службы > Контроллеры домена**. Нажать

кнопку [Новый контроллер домена], выбрать из списка сервер **dc-2.ald.company.lan** и остальные параметры, как показано на *Добавление нового контроллера домена* Задать пароль для администратора ALD Pro.

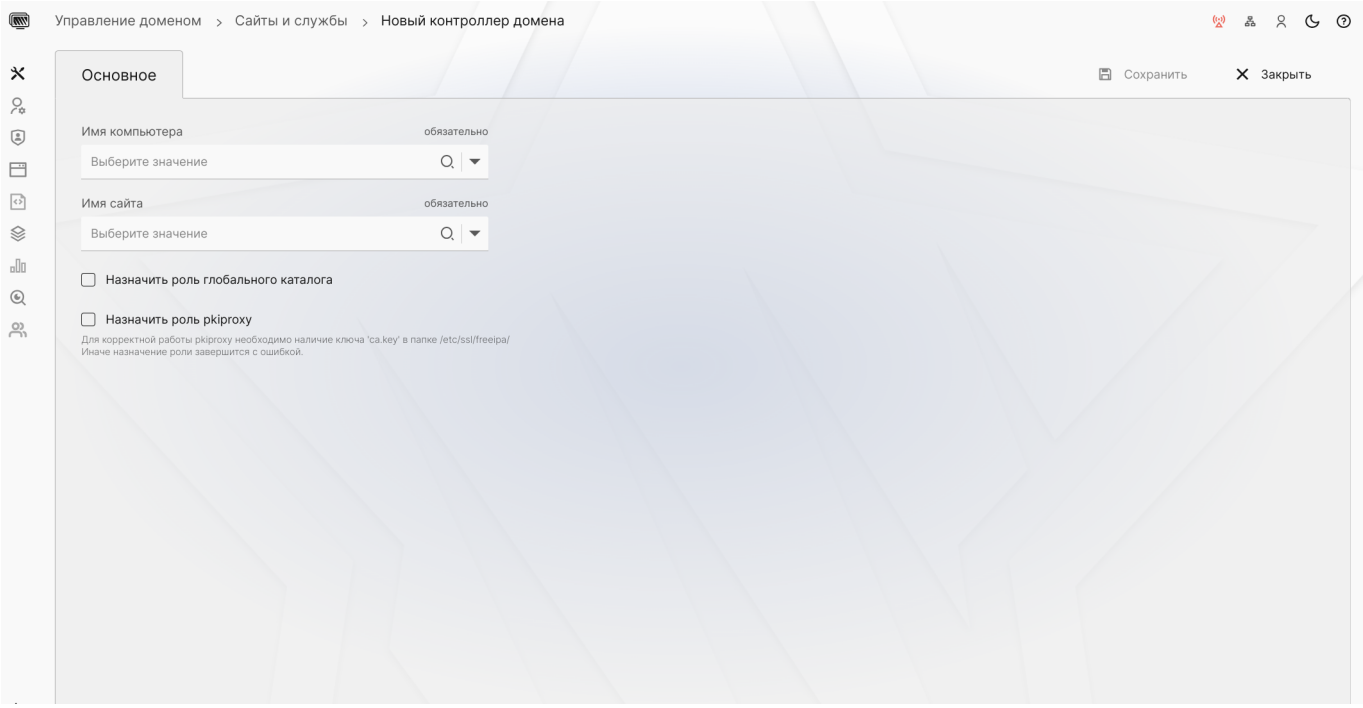


Рисунок 2.8 – Добавление нового контроллера домена

Результат выполнения задания по установке резервного контроллера можно посмотреть в карточке резервного контроллера домена.

Дополнительную информацию о процессе установки можно получить на **dc-2** в журналах */var/log/apt/term.log* и */var/log/aldpro-salt/minion* или воспользоваться утилитой **journalctl** с параметром `-f`, которая позволяет получить информацию из **systemd**:

```
sudo journalctl -f
```

После установки резервного контроллера домена начнется репликация данных, которая займет некоторое время.

По завершении успешной установки резервного контроллера необходимо снять запрет на редактирование файла */etc/resolv.conf* командой:

```
sudo chattr -i /etc/resolv.conf
```

А также привести файл */etc/resolv.conf* к следующему виду:

```
nameserver 127.0.0.1
search ald.company.lan
```

Под успешной установкой резервного контроллера подразумевается наличие статуса «**Успешно**» у соответствующего задания в подразделе **Автоматизация > Задания автоматизации**, а также отсутствие ошибок в отчете о выполнении задания и логе ошибки исполнения в соответствующем задании.

2.5.3. Проверка статуса репликации

2.5.3.1. Проверка статуса репликации утилитой dsconf

Команда, показывающая информацию по соглашениям о репликации текущего сервера:

```
sudo dsconf -j ALD-COMPANY-LAN replication status --suffix dc=ald,dc=company,
↪dc=lan
```

Результат вывода в формате json:

```
{
    "type": "list", "items": [{
        "agmt-name": ["meTodc-2.ald.company.lan"],
        "replica": ["dc-2.ald.company.lan:389"],
        "replica-enabled": ["on"],
        "update-in-progress": ["FALSE"],
        "last-update-start": ["20230824143651Z"],
        "last-update-end": ["20230824143651Z"],
        "number-changes-sent": ["4:25/723 5:1/0 "],
        "number-changes-skipped": ["unavailable"],
        "last-update-status": ["Error (0) Replica acquired successfully:
↪Incremental update succeeded"],
        "last-init-start": ["20230824120948Z"],
        "last-init-end": ["20230824120954Z"],
        "last-init-status": ["Error (0) Total update succeeded"],
        "reap-active": ["0"],
        "replication-status": ["Not in Synchronization: supplier
↪(64e76843000000040000) consumer (Unavailable) State (green) Reason (error
↪(0) replica acquired successfully: incremental update succeeded)"],
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
        "replication-lag-time": ["unavailable"]
    }]
}
```

Приведенную информацию можно представить в формате сводной таблицы, но потребуется установить пакет **jq** и использовать комплексную команду форматирования:

```
apt install jq -y; (printf 'SUFFIX \tAGREEMENT \tSTATE \tTIME-SINCE \tLDAP-
↪STATUS \tREPL-STATUS \n'; dsconf -j ALD-COMPANY-LAN replication list | jq '.
↪items[]' -r | xargs -P8 -i -- dsconf -j ALD-COMPANY-LAN repl-agmt list --
↪suffix={} | jq '.items[].attrs | (.nsds5replicalastupdatestatusjson[0] |
↪fromjson) as $status | [.nsds5replicaroot[0], .cn[0], $status.state,
↪$status.date, $status.ldap_rc_text, $status.repl_rc_text] | @tsv' -r |
↪sort ) | column -s$'\t' -t
```

Результат вывода команды, см. *Информация о соглашении в табличном виде*.

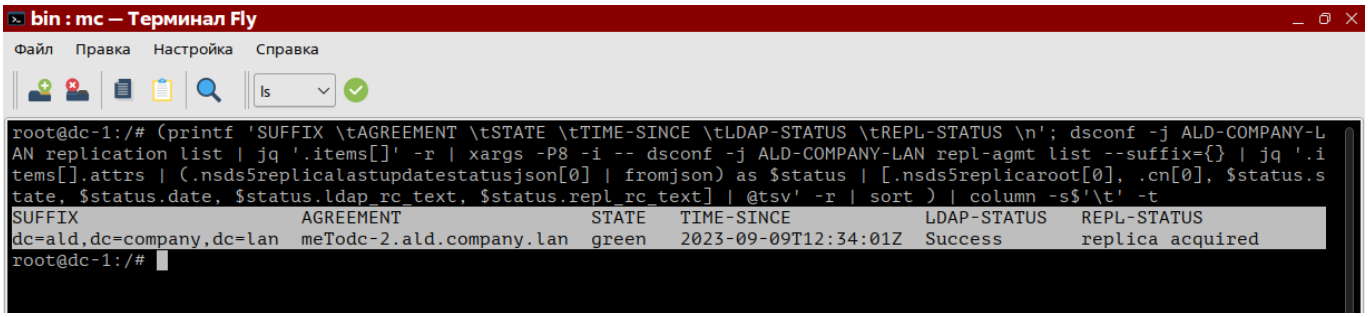


Рисунок 2.9 – Информация о соглашении в табличном виде

Есть также возможность с помощью команды `dsconf replication monitor` собрать информацию сразу со всех контроллеров домена, но в состоянии репликации не будет указан агрегированный статус, что затрудняет анализ:

```
set +o history
dsconf -D 'cn=Directory Manager' -w 'AstraLinux_176' ldap://localhost:389
replication monitor
set -o history
```

Учетные данные для выполнения этой команды нужно будет вводить вручную для каждого сервера или задать их в файле *~/.dsrc*:

```
[repl-monitor-connections]
connection1 = dc-1.ald.company.lan:389:cn=Directory Manager:*
connection2 = dc-2.ald.company.lan:389:cn=Directory Manager:[~/pwd.txt]
connection3 = dc-3.ald.company.lan:389:cn=Directory Manager:S3cret
```

Где:

- для **dc-1** будет запрошен пароль;
- для **dc-2** пароль будет взят из файла *pwd.txt*;
- для **dc-3** в качестве пароля будет использована строка **S2cret**.

2.5.3.2. Проверка репликации с помощью скриптов проекта checkipaconsistency

> **Внимание:** В данном разделе представлена информация о том, как можно проверить целостность домена с использованием скриптов **checkipaconsistency** из публичного **pip** репозитория, но установка модулей из открытых источников без дополнительной проверки кода в продуктивных средах категорически запрещается.

Для успешной установки потребуется выполнить вход в сессию **root** пользователя:

```
sudo su
```

Ввести пароль от **admin** и войти в сессию пользователя **root**, так как пользователь **admin** находится в группе **astra-admin**:

```
root@dc-1:/home/admin#
```

Теперь выполнить установку пакеты **pip**, чтобы поставить пакет **checkipaconsistency**:

```
apt install libsasl2-dev python-dev libldap2-dev libssl-dev python-pip
```

Проверить работоспособность **pip**, проверив версию и установив утилиту **cipa**

```
pip --version && pip install --user checkipaconsistency
```

Добавить ссылку на запуск по сокращенному имени **cipa**:

```
ln -s /root/.local/bin/cipa /usr/bin/cipa
```

Проверить реплику можно командой из-под пользователя **root**, отключив историю:

```
set +o history
cipa -d ald.company.lan -W AstraLinux_176
set -o history
```

Результат выполнения утилиты **cipa**, см. *Результат выполнения checkipaconsistency*

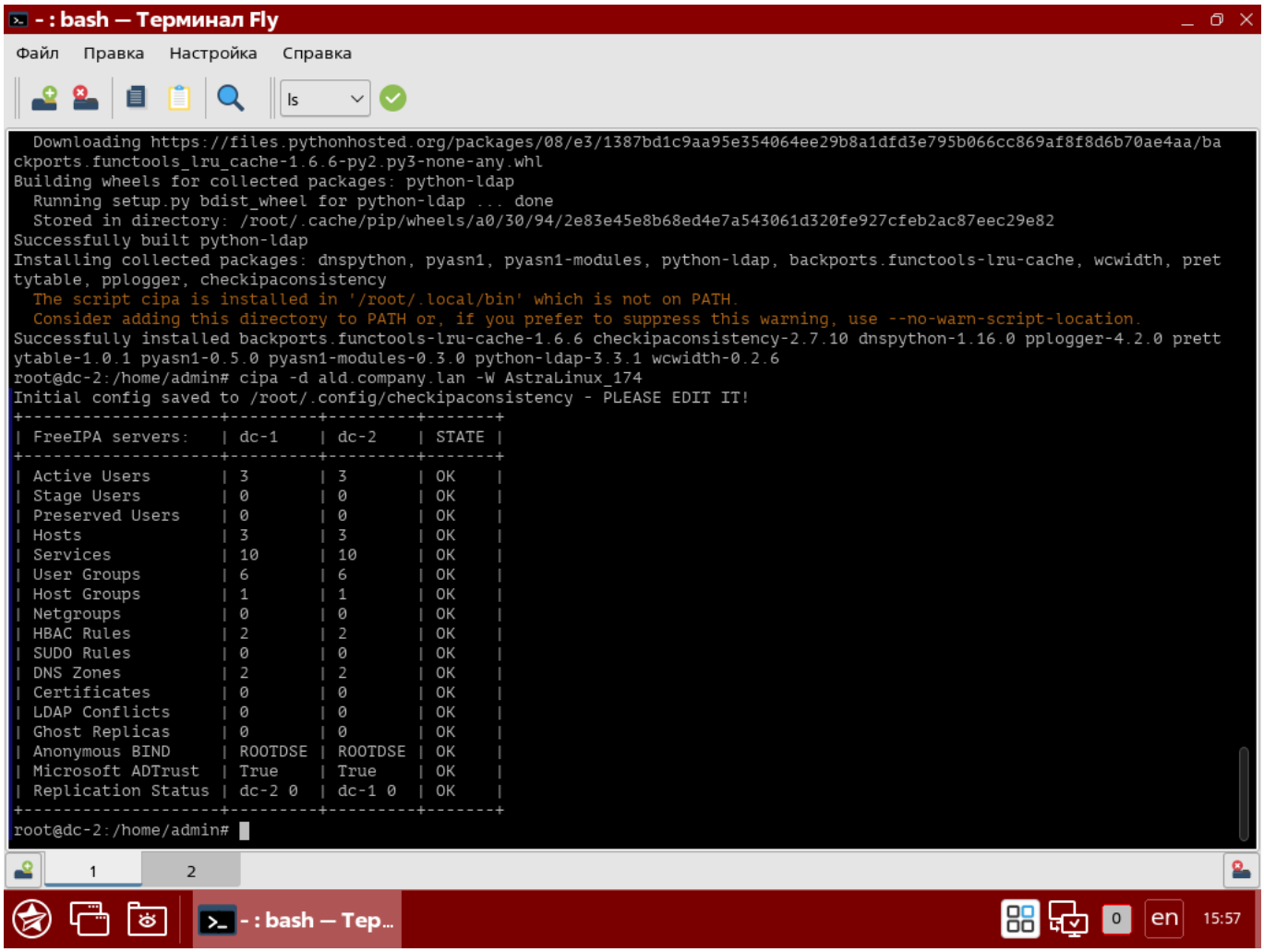


Рисунок 2.10 – Результат выполнения checkipaconsistency

2.6. Удаление первого контроллера домена

Удаление первого контроллера домена возможно только при наличии доступа к нему.

2.6.1. Подготовка к удалению

До начала процедуры удаления первого контроллера домена необходимо произвести следующие действия:

1. Выполнить резервное копирование всех контроллеров домена одним из следующих способов:
    1. создание **снапшотов**,
    2. согласно данной инструкции, при условии, что контроллеры не являются виртуальными машинами.
2. Изменить сервер авторизации для всех серверов мониторинга
    1. Произвести авторизацию по адресу https://<адрес сервера мониторинга>/zabbix. Логин и пароль для входа можно узнать с любого контроллера домена при помощи команды

```
sudo cat /opt/rbta/ad/mgmtportal/api/core/.env | grep ZABBIX
```

    2. слева нажать: **Administration** - **Authentication**
    3. выбрать вкладку **LDAP settings**
    4. в поле **LDAP host** ввести адрес любого контроллера, кроме первого (**который будет удален**)
    5. нажать кнопку «**Update**»
3. Удалить первый контроллер домена из всех серверов мониторинга
    1. произвести авторизацию на сервере мониторинга как в пункте №2
    2. слева нажать: **Configuration** > **Hosts**
    3. отметить галкой нужный сервер и нажать кнопку «**Delete**»
4. Адреса контроллеров в файле */etc/resolv.conf*
    1. На каждом компьютере в домене в файле */etc/resolv.conf* необходимо изменить адрес удаляемого сервера на адрес любого другого.

Пример:

Если адрес первого контроллера домена 10.2.0.10, а адрес второго контроллера домена 10.2.0.11, то содержимое файла должно быть следующим:

```
domain <имя домена>
nameserver 10.2.0.11
```

2. Данное действие можно выполнить при помощи задания автоматизации или групповой политикой. Пример кода:

```
replace_string:
file.replace:
    - name: /etc/resolv.conf
    - pattern: 10.2.0.10
    - repl: 10.2.0.11
```

5. Убедиться, что между удаляемым контроллером домена и другим контроллером домена установлены **двусторонние соглашения о репликации**. Сделать это можно двумя способами:

    1. На портале управления в разделе **Управление доменом** > **Сайты и службы** > **Соглашения о репликации** должны быть **двусторонние стрелки** между контроллерами домена

    2. В консоли любого контроллера домена командой (вместо **dc=aldpro,dc=test** необходимо подставить имя своего домена)

```
ldapsearch -D "cn=directory manager" -W -b "cn=topology,
↪cn=ipa,cn=etc,dc=aldpro,dc=test"
```

    В выводе команды должно быть

```
ipaReplTopoSegmentDirection: both
```

    для всех контроллеров домена.

    3. Если это не так, необходимо восстановить репликации (раздел *Восстановление Контроллера Домена*). В противном случае, удаление контроллера домена будет невозможно.

6. Механизм обработки ролей (активация роли) и очистки каталога от неактуальных сервисных данных запускается на первом контроллере. В

случае необходимости отключить первый КД или перенести задачу такой обработки на другой КД, необходимо отключить службы:

```
systemctl disable --now aldpro-preserved-users-sweep.timer
↪aldpro-roles-sweeper.timer aldpro-process-pending-roles.
↪timer aldpro-check-in-restore.timer
```

и включить службы на новом выбранном для такой задачи КД:

```
systemctl enable --now aldpro-preserved-users-sweep.timer
↪aldpro-roles-sweeper.timer aldpro-process-pending-roles.
↪timer aldpro-check-in-restore.timer
```

Для проверки служб:

```
systemctl status aldpro-preserved-users-sweep.timer aldpro-
↪roles-sweeper.timer aldpro-process-pending-roles.timer
↪aldpro-check-in-restore.timer
```

Если службы отключены, в консоль будет выведено:

```
Active: inactive (dead)
Trigger: n/a
```

Если службы включены, в консоль будет выведено:

```
Active: active (waiting) since Sat 2024-11-09 08:38:44 MSK;
↪8min ago
Trigger: Sat 2024-11-09 09:38:44 MSK; 51min left
```

2.6.2. Удаление контроллера домена

1. Произвести авторизацию по **ssh** в консоль удаляемого сервера и выполнить команду

```
sudo astra-freeipa-server -U
```

В выводе команды должно быть сообщение **The ipa-server-install command was successful**. Если произошла ошибка, требуется вернуться к пункту №5 раздела «**Подготовка к удалению**»

2. Авторизоваться на портале управления по адресу https://<адрес второго контроллера домена>
3. Зайти в раздел **Роли и службы сайта** - **Служба разрешения имён**
4. В списке «**Имя зоны**» кликнуть на имя вашего домена
5. В строке поиска набрать имя удаленного контроллера домена
6. Галочкой выделить все появившиеся записи и нажать кнопку «**Удалить**»
7. Пункты 5-6 повторить до полного удаления всех записей, относящихся к удаленному контроллеру домена
8. Произвести авторизацию в интерфейсе **FreeIPA** по адресу https://<адрес второго контроллера домена>/ipa/ui
9. Сверху нажать кнопку **IPA-сервер**, далее «**Топология**»
10. Слева выбрать **IPA-серверы**
11. Кликнуть на имя удаленного контроллера домена и нажать появившуюся кнопку «**Удалить сервер**»
12. Проверить успешность удаления можно на портале управления в разделе **Управление доменом** > **Сайты и службы** > **Контроллеры домена**. Удаленного контроллера домена не должно быть в списке.
13. Очистка кластера **RabbitMQ**. Для очистки нужно зайти под **admin** пользователем на контроллер домена и выполнить следующую команду:

```
sudo rabbitmqctl forget_cluster_node rabbit@dc01
```

14. Если после удаления контроллера домена подсистемы невозможно развернуть, необходимо проверить содержимое файла */etc/resolv.conf* на компьютере, на котором будет развернута подсистема - в файле не должно быть адреса удаленного контроллера, а должны быть адрес(а) других контроллеров домена.

2.7. Удаление резервного контроллера домена, отключенного от сети

Удаление контроллера из домена при недоступном сервере включает в себя следующие этапы:

1. Удаление DNS-записей,
2. Удаление записей из **SALT**,
3. Удаление следов контроллера домена,
4. Удаление записей из **LDAP**.

2.7.1. Удаление DNS-записи

Для удаления DNS-записи необходимо войти на портал управления под учетной записью администратора (**admin**).

Перейти в раздел **Роли и службы сайта** > **Служба разрешения имен**.

С помощью поиска найти имя выводимого (удаляемого) компьютера.

Выделить все записи в отфильтрованном списке, установив флаг выбора всех записей, и нажать **Удалить**.

Примечание: Необходимо убедиться, что все выбранные записи относятся к выводимому компьютеру.

Повторить поиск по имени выводимого компьютера, чтобы убедиться, что не осталось записей. При необходимости - повторить удаление.

2.7.2. Удаление записи из SALT

Для удаления записи из **SALT** необходимо зайти под привилегированным пользователем с высоким уровнем целостности на один из работающих контроллеров домена и выполнить команду:

```
sudo salt-key -y -d <fqdn_выводимого_компьютера>
```

Пример:

```
sudo salt-key -y -d dc-2.ald.company.lan
```

2.7.3. Удаление следов контроллера домена

Перейти в раздел **Пользователи и компьютеры** > **Компьютеры**, не остался ли выводимый компьютер в списке.

Если остался, то его необходимо удалить. Для этого необходимо зайти под привилегированным пользователем **admin** с высоким уровнем целостности на один из работающих контроллеров домена и выполнить следующие команды:

Авторизоваться под учетной записью администратора:

```
sudo kinit admin
```

Выполнить команду:

```
sudo ldapdelete <fqdn-до-записи-с-компьютером-в-LDAP>
```

где имеет вид `fqdn=,cn=computers,cn=accounts,`

Пример:

```
sudo ldapdelete fqdn=dc-2.ald.company.lan,cn=computers,cn=accounts,dc=ald,
↪dc=company,dc=lan
```

На контроллерах, которые были недоступны в момент удаления реплики, может остаться вектор удаленной реплики.

Для удаление вектора необходимо зайти под привилегированным пользователем с высоким уровнем целостности на один из работающих КД и выполнить следующую команду:

```
sudo ipa-replica-manage clean-dangling-ruv
```

Примечание: Требуется ввести пароль от Directory Manager.

Следы от удаляемого контроллера домена останутся также в кластере **RabbitMQ**.

Для их очистки нужно зайти под привилегированным пользователем **admin** с высоким уровнем целостности на один из работающих контроллеров домена и выполнить следующую команду:

```
sudo rabbitmqctl forget_cluster_node rabbit@<deleting_dc_name>
```

Пример:

```
sudo rabbitmqctl forget_cluster_node rabbit@dc-2
```

Где: `dc-2` - относительное имя удаляемого контроллера домена, которое можно увидеть в списке кластера.

2.7.4. Удаление записи из LDAP

При удалении реплики удаляются все записи из LDAP, кроме `cn=host_name,cn=masters,cn=ipa,cn=etc`. При этом именно данная запись служит триггером для отображения соглашений о репликации.

После удаления реплики необходимо проверить и снести запись `cn=host_name,cn=masters,cn=ipa,cn=etc` со всеми ее вложенными записями:

```
sudo ldapdelete cn=host_name,cn=masters,cn=ipa,cn=etc
```

2.8. Установка подсистемы сетевого репозитория

Установка сервера подсистемы сетевого репозитория выполняется в два шага:

1. Сначала сервер вводится в домен как обычная рабочая станция.
2. Затем на портале управления серверу назначается роль **Сервера репозиториев ПО**, установка и настройка происходит автоматически благодаря системе автоматизации.

2.8.1. Установка сервера репозиториев

Подготовить сервер репозиториев для установки и обновления программного обеспечения по сети с учетом минимальных требований.

На основании раздела ввода в домен настроить сеть:

- IP адрес — 10.0.1.13
- Маска подсети — 255.255.255.0;
- Шлюз — 10.0.1.1;
- Сервер DNS — 10.0.1.11.
- Поисковый домен: ald.company.lan

Проверить доступность репозиториев **dl.astralinux.ru** и ответ от контроллера домена **dc-1**:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

Настроить репозитории и обновить программное обеспечение см. *Настройка доступных репозиториев*.

Установить клиентский пакет программ ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Ввести сервер репозиториев **repo** в домен **dc-1**:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.
↪lan --account admin --password 'AstraLinux_176' --host repo --gui --force
set -o history
```

Перезагрузить сервер **repo**, чтобы настройки вступили в силу:

```
sudo reboot
```

В портале ALD Pro в разделе **Установка и обновление ПО > Репозитории ПО** на вкладке

Серверы репозиториев ПО развернуть сервер репозиториев нажав на кнопку [**Новый сервер репозиториев ПО**], привязав его к сайту «Головной офис», см. *Добавление сервера репозиториев*.

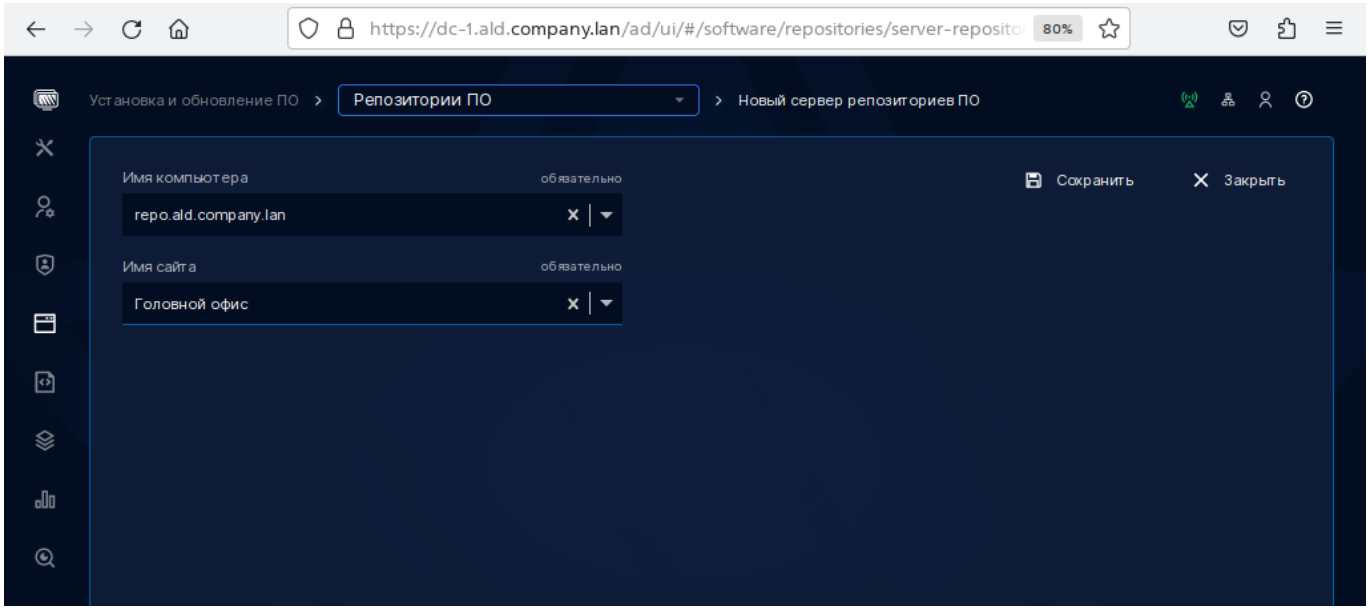


Рисунок 2.11 – Добавление сервера репозиториев

В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

Проверить выполнение скрипта автоматизации можно в карточке подсистемы.

2.8.2. Создание репозиториев из ISO образа

Для установки программного обеспечения через групповые политики требуется репозиторий, созданный с использованием официального ISO-образа. Это позволит устанавливать и обновлять программное обеспечение без обращения к сети Интернет.

Необходимо скачать образ-ISO с базовым дистрибутивом на сайте в личном кабинете, перейдя по ссылке https://lk-new.astralinux.ru/, или вставить диск в привод, скопировав ISO-образ в текущую директорию командой dd:

```
dd if=/dev/sr0 of=al174main.iso bs=100M status=progress
```

Создать новый репозиторий для корпоративной сети, перейдя в раздел «Установка и

обновление ПО — Репозитории ПО». На вкладке «Репозитории ПО» нажать на кнопку «Новый репозиторий».

В открывшемся окне «**Основное**» назвать новый репозиторий «astra-linux-base» и указать абсолютный путь */astralinux17base*, см. *Новый репозиторий с базового образа Astra Linux 1.7*. Необходимо обратить внимание, что путь начинается с символа «/». После заполнения нажать на кнопку «**Сохранить**».

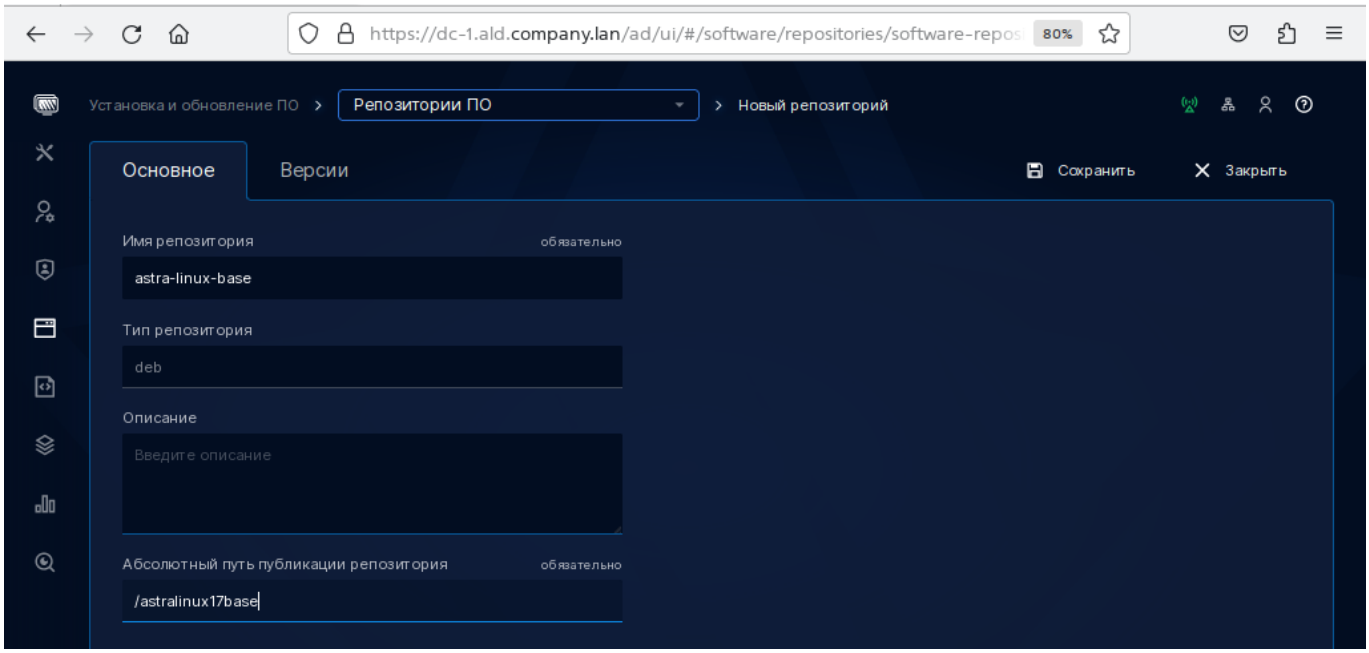


Рисунок 2.12 – Новый репозиторий с базового образа Astra Linux 1.7

После сохранения станут доступны дополнительные вкладки для настройки репозитория. Перейти на вкладку «**Версии**» и создать новую запись, см. *Вкладка Версии нового репозитория astra-linux-base*.

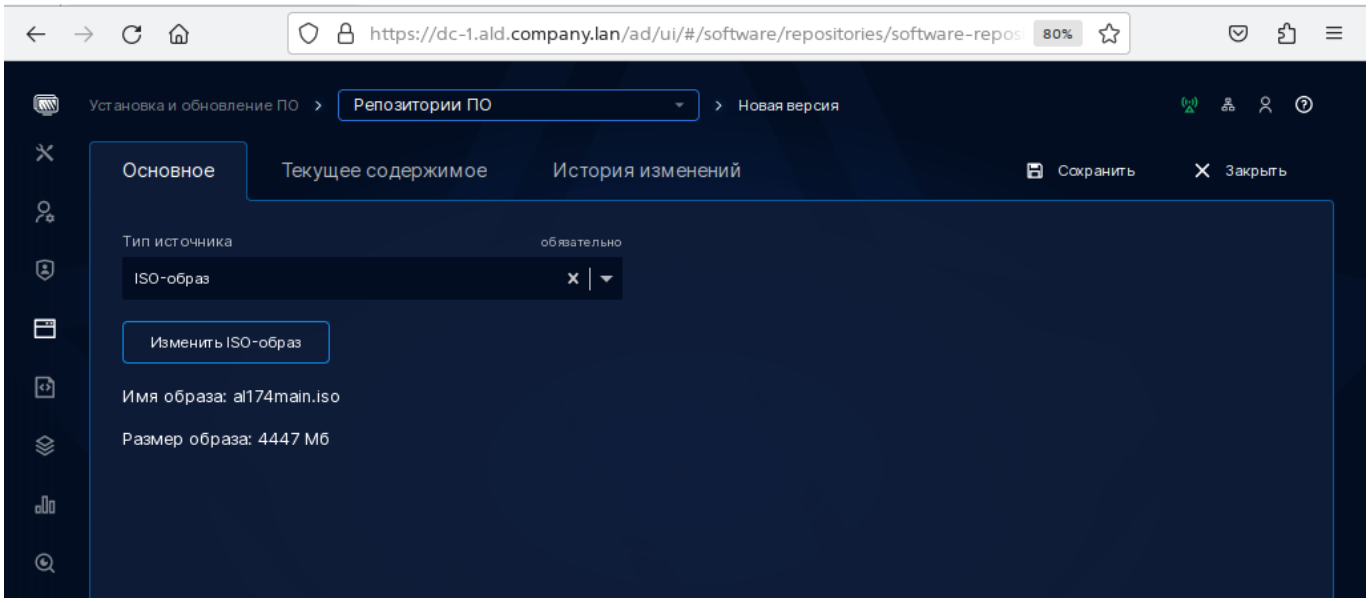


Рисунок 2.13 – Вкладка Версии нового репозитория astra-linux-base

Необходимо создать Версию из ISO-образа, для этого из списка выбрать загруженный образ. Чтобы начался процесс загрузки образа на сервер репозиториев, нажать на кнопку «**Сохранить**».

После загрузки файла на сервер, он будет обрабатываться некоторое время. Проверить статус обработки можно на странице с версией, см. рис. *Обработка ISO образа на вкладке «Версия»*.

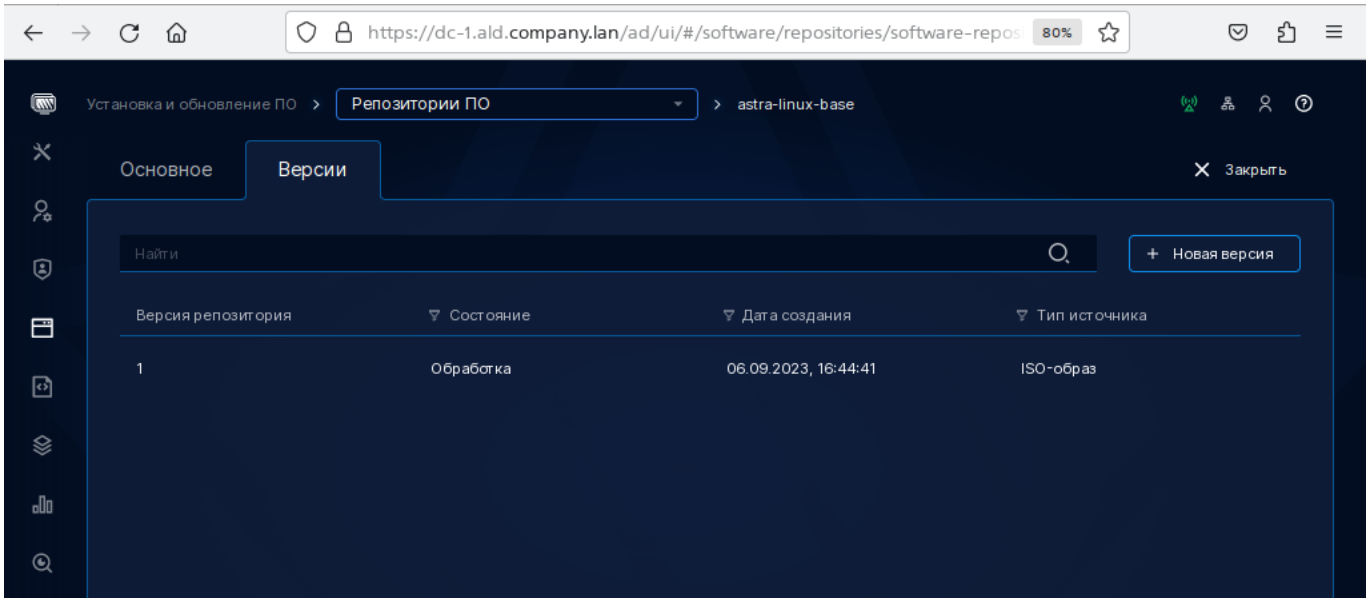


Рисунок 2.14 – Обработка ISO образа на вкладке «Версия»

После обработки статус поменяется на Опубликована, см. *Обработанный ISO образ Версии 1 со статусом «Опубликована»*, после чего его можно использовать в файле

/etc/apt/sources.list.

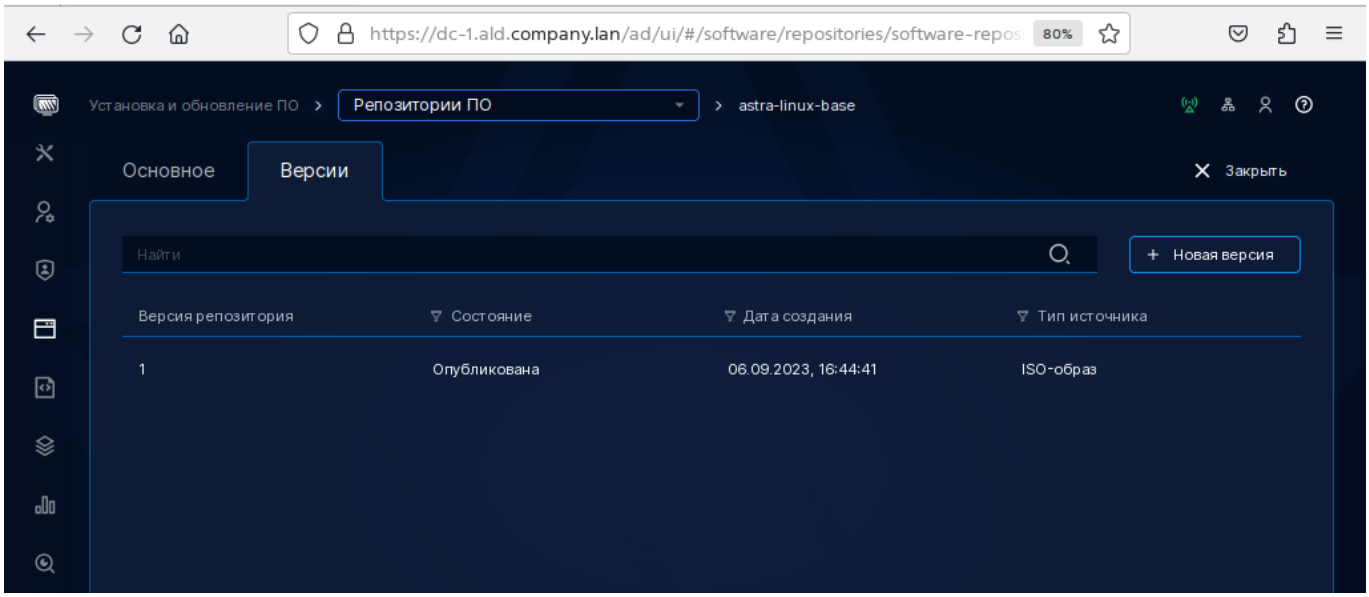


Рисунок 2.15 – Обработанный ISO образ Версии 1 со статусом «Опубликована»

Открыть файл *sources.list* на **pc-1** для настройки базовой версии Astra Linux:

```
ssh pc-1
sudo nano /etc/apt/sources.list
```

Вставить добавленный репозиторий в *source.list*, а адреса **dl.astralinux.ru** закомментировать через символ #:

```
#deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/
↪1.7_x86-64 main contrib non-free
#deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-
↪extended/ 1.7_x86-64 main contrib non-free
deb [trusted=yes] https://repo.ald.company.lan/repos/astralinux17base/ 1.7_
↪x86-64 main contrib non-free
```

Где:

- `[trusted=yes]` — обозначает, что есть доверие к этому репозиторию, т.е можно устанавливать пакеты без цифровых подписей.
- `https://repo.ald.company.lan/repos/astralinux17base/` — полный путь до репозитория, который можно посмотреть во вкладке «Основное» выбранного репозитория. Порт 443 можно не указывать, потому что он используется для протокола https.

В версии 2.1.0 была особенность, что подсистема репозитория получала сертификат на короткое имя по названию компьютера, например, **repo** вместо **repo.ald.company.lan**, поэтому для использования **https** в этом случае требовалось указывать адрес в формате `deb [trusted=yes] https://repo/repos/astralinux17base/ ...`.

- `1.7_x86-64` — кодовое имя дистрибутива, которое можно посмотреть в описании версии.
- `main`, `contrib` и `non-free` — компоненты дистрибутива, которое можно посмотреть в описании версии.

Сохранить и проверить, что установка пакетов будет с адреса нового репозитория в корпоративной сети:

```
sudo apt update
```

Результат выполнения обновления кэша:

```
Игн:1 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64
 ↪InRelease
Пол:2 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64 Release
 ↪[5 766 B]
Пол:3 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64 Release.
 ↪gpg [833 B]
Пол:4 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64/main
 ↪amd64 Packages [1 310 kB]
Пол:5 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64/contrib
 ↪amd64 Packages [2 155 B]
Пол:6 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64/non-
 ↪free amd64 Packages [55,8 kB]
Сущ:7 https://dl.astralinux.ru/aldpro/stable/repository-main 2.4.0 InRelease
Сущ:8 https://dl.astralinux.ru/aldpro/stable/repository-extended generic
 ↪InRelease
Получено 1 374 kB за 0с (3 214 kB/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Все пакеты имеют последние версии.
```

Установить **htop** для проверки, что пакет установится из сервера репозитория **astralinux17base**:

```
sudo apt install htop -y
```

Результат выполнения установки **htop**:

```
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Предлагаемые пакеты:
strace
Следующие НОВЫЕ пакеты будут установлены:
htop
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления
отмечено 0 пакетов, и 0 пакетов не обновлено.
...
Необходимо скачать 89,9 kB архивов.
После данной операции объём занятого дискового пространства
возрастёт на 213 kB.
Пол:1 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64/main
amd64 htop amd64 2.4.0-1 [89,9 kB]
Получено 89,9 kB за 0с (6 817 kB/s)
Выбор ранее не выбранного пакета htop.
(Чтение базы данных ... на данный момент установлено 176626 файлов и
 ↪каталогов.)
Подготовка к распаковке .../htop_2.4.0-1_amd64.deb ...
Распаковывается htop (2.4.0-1) ...
Настраивается пакет htop (2.4.0-1) ...
Обрабатываются триггеры для man-db (2.8.5-2) ...
Обрабатываются триггеры для desktop-file-utils (0.26-1astra1) ...
Обрабатываются триггеры для mime-support (3.62) ...
Обрабатываются триггеры для xserver-xorg-core (2:1.20.14-
1ubuntu1astra.se31) ...
update exec ids due to /usr/bin changed
```

В результате видно, что пакет установился с корпоративного репозитория **repo** с адреса: https://repo.ald.company.lan/repos/astralinux17base

2.8.3. Создание репозитория из загруженных пакетов deb

В качестве примера создается репозиторий с названием **yandex-browser-corp**, в который загрузится корпоративный браузер Яндекс. Для этого потребуется скачать deb пакет для Astra Linux. Его по умолчанию можно загрузить командой:

```
cd ~/Загрузки
wget https://download.yandex.ru/browser/astra-os/yandex-browser.deb
mv yandex-browser.deb "yandex-browser-$(dpkg -f yandex-browser.deb version).
↪deb"
```

В рабочей директории появится файл с последней версией, например, **yandex-browser-23.7.1.1219-1.deb**. Его нужно загрузить в репозиторий, также есть возможность настроить пакет в личном кабинете https://browser.yandex.ru/corp, см. *Подготовка сборки для Linux из личного кабинета*.

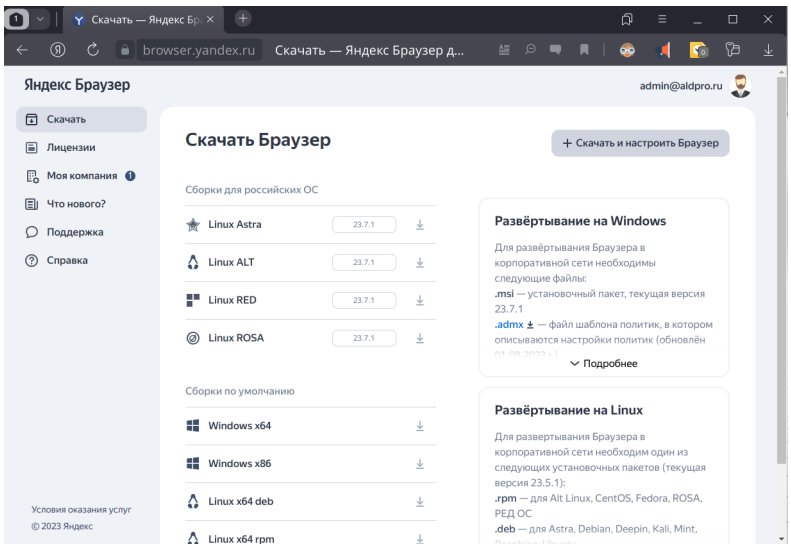


Рисунок 2.16 – Подготовка сборки для Linux из личного кабинета

На портале ALD Pro создать новый репозиторий с названием «**yandex-browser-corp**» и указать абсолютный путь */yandexbrowser*, см. *Новый репозиторий для пакета deb на примере корпоративного браузера Яндекс*. Примените изменения нажатием кнопки «**Сохранить**».

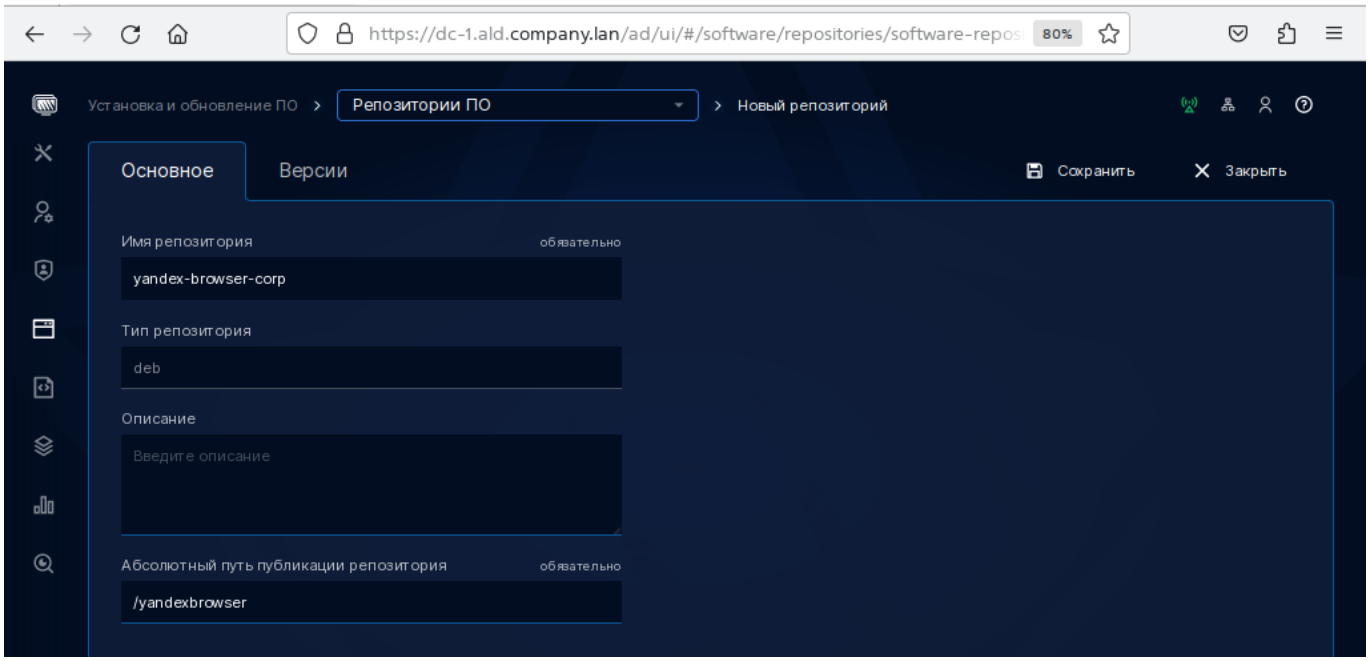


Рисунок 2.17 – Новый репозиторий для пакета deb на примере корпоративного браузера Яндекс

Создать новую версию deb репозитория в **yandex-browser-corp** во вкладке Версии, см. *Описание версии deb пакета для браузера Яндекс*.

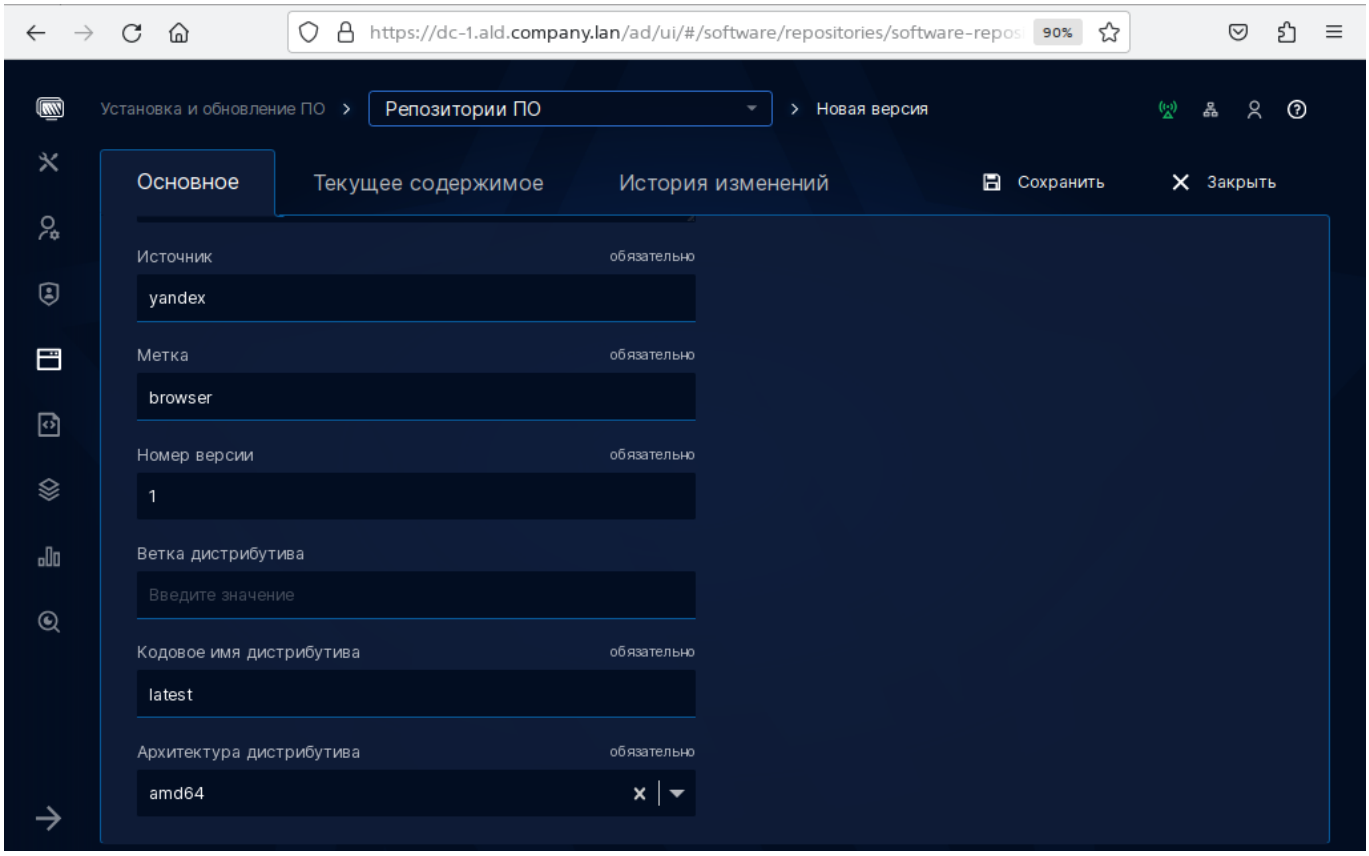


Рисунок 2.18 – Описание версии deb пакета для браузера Яндекс

Заполните поля значениями, см. табл. 4, и нажмите на кнопку «**Сохранить**», чтобы активировать вкладку «**Текущее содержимое**».

Поле	Значение	Описание
Источник	yandex	Справочное поле для описания источника пакета
Метка	browser	Справочное поле для описания вида программного пакета
Номер версии	1	Целое число для версионирования репозитория. При обновлении версии репозитория нужно использовать следующий порядковый номер
Кодовое имя дистрибутива	latest	Используется для возможности размещения в одном репозитории нескольких дистрибутивов, но репозиторий «ALD Pro» позволяет разместить только один дистрибутив, поэтому не имеет практического значения,рекомендуется использовать значение latest
Архитектура дистрибутива	amd64	Используется для возможности размещения в одном репозитории пакетов для разных архитектур. Репозиторий «ALD Pro» позволяет разместить только одну версию пакетов, поэтому не имеет практического значения, можно использовать значение all
Компоненты дистрибутива	main	Используется для возможности распределения пакетов по категориям. В репозитории «ALD Pro» можно разместить пакеты только одной категории main, поле не редактируется.

Таблица 4 — Назначение полей при создании репозитория из deb пакета

Добавить в версию №1 файл *yandex-browser-23.7.4.981-1.deb* на вкладке «**Текущее содержимое**», см. *Вкладка «Текущее содержимое» версии №1 deb репозитория yandex-browser-corp*.

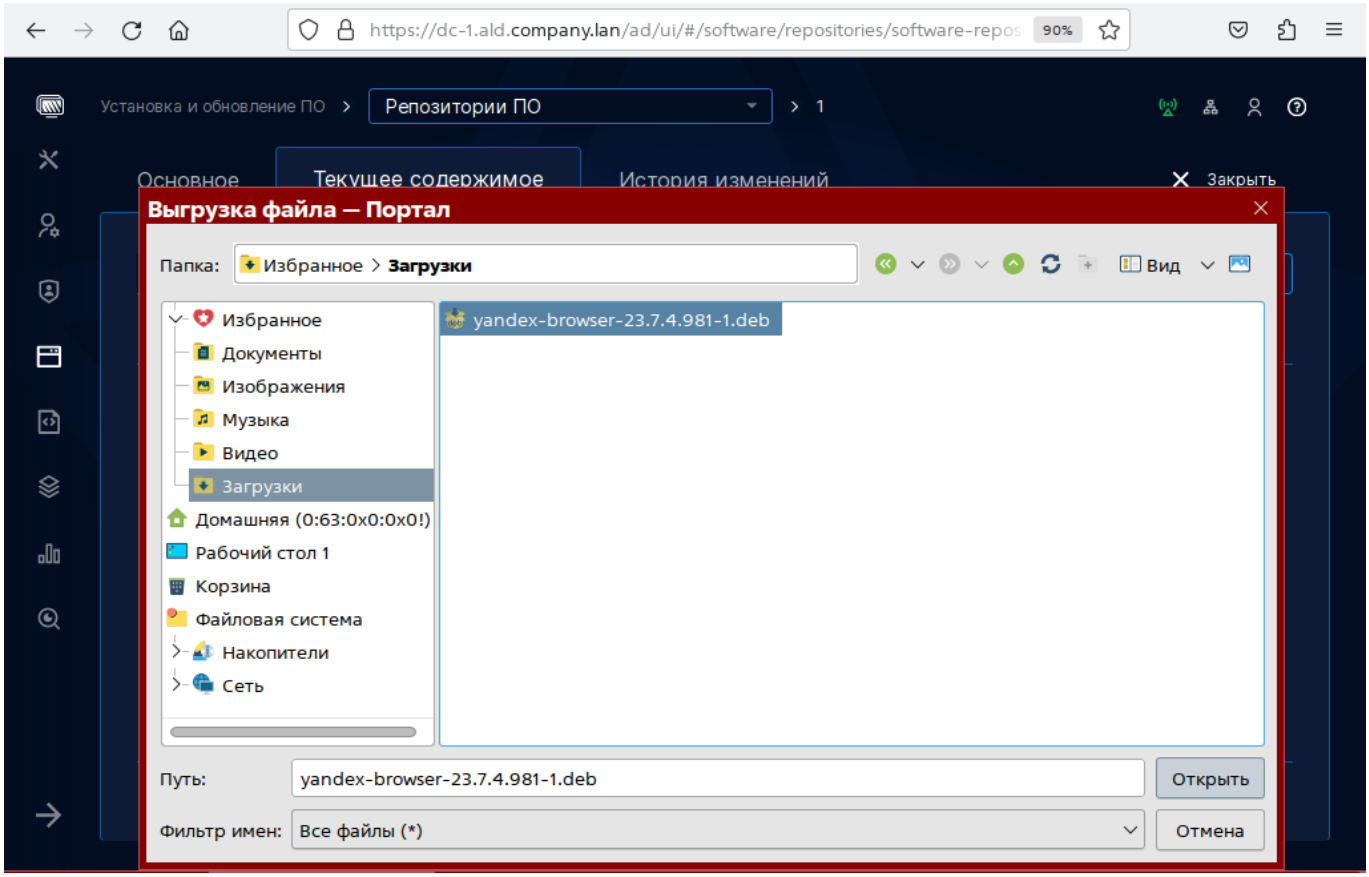


Рисунок 2.19 – Вкладка «Текущее содержимое» версии №1 deb репозитория yandex-browser-corp

Выбрать загруженный файл из директории и дождаться обработки его добавления в список пакетов. После обработки файлов требуется нажать на кнопку «**Опубликовать**», см. *Публикация версии №1 deb-репозитория yandex-browser-corp*.

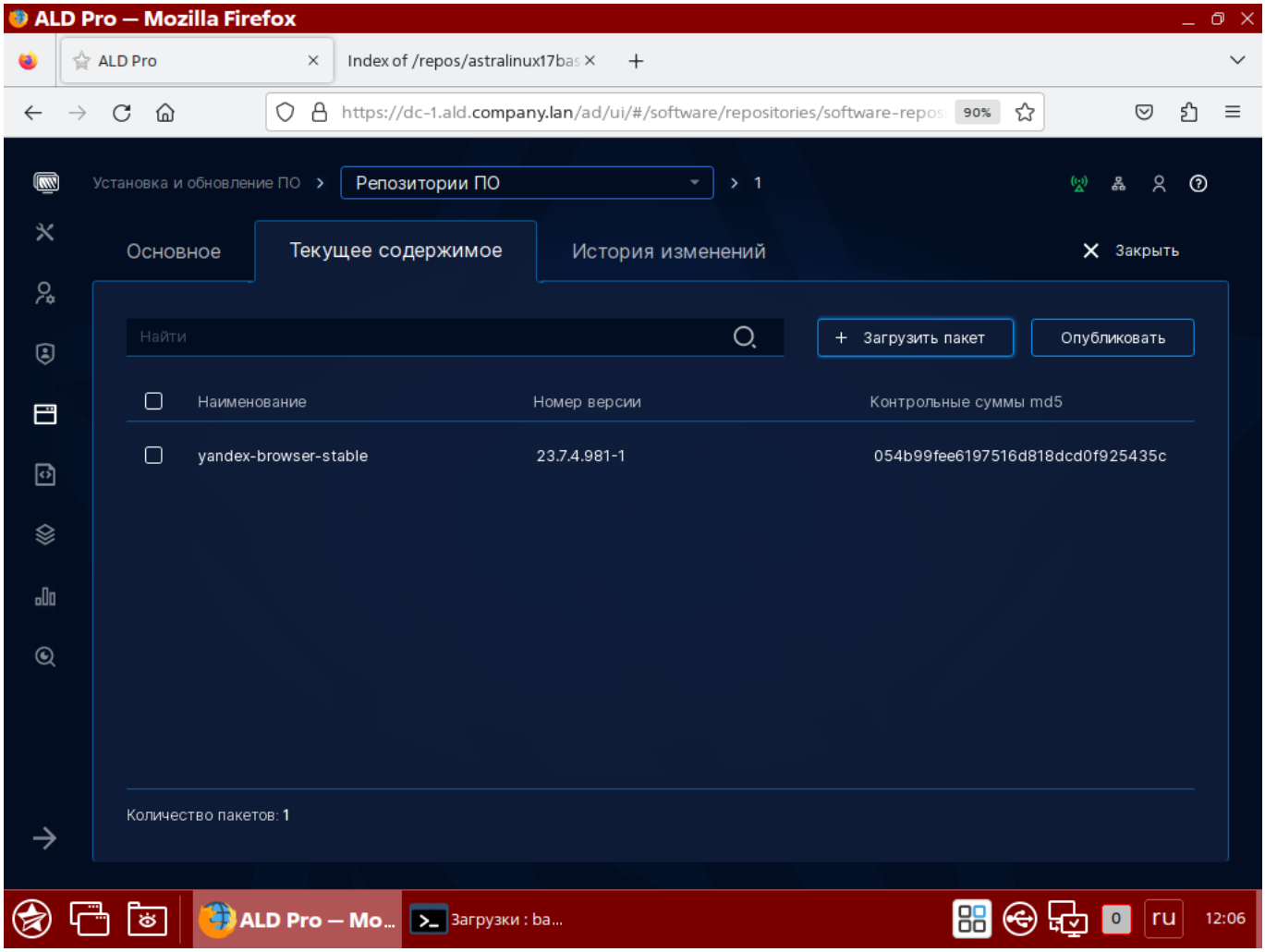


Рисунок 2.20 – Публикация версии №1 deb-репозитория yandex-browser-corp

---

Важно: В текущей реализации подсистемы репозиториев действие «Опубликовать» нельзя отменить, поэтому для обновления приложения нужно будет создать новую версию репозитория.

2.8.4. Установка браузера из deb репозитория

На клиентском компьютере настроить *sources.list*, добавив репозиторий с браузером Яндекс:

```
admin@pc-1:~$ sudo nano /etc/apt/sources.list
```

Добавить второй репозиторий в файл *sources.list*:

```
deb [trusted=yes] https://repo.ald.company.lan/repos/astralinux17base/ 1.7_
↪x86-64 main contrib non-free
deb [trusted=yes] https://repo.ald.company.lan/repos/yandexbrowser/ latest
↪main
```

Внимание: Базовый дистрибутив 1.7_x86-64 main тоже подключен, потому что могут потребоваться дополнительные пакеты и библиотеки.

Обновить новые списки в кэше **apt**:

```
sudo apt update
```

В результате добавлен репозиторий https://repo.ald.company.lan/repos/yandexbrowser latest:

```
Игн:1 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64
↪InRelease
Игн:2 https://repo.ald.company.lan/repos/yandexbrowser latest InRelease
Сущ:3 https://repo.ald.company.lan/repos/astralinux17base 1.7_x86-64 Release
Пол:4 https://repo.ald.company.lan/repos/yandexbrowser latest Release [865 B]
Игн:5 https://repo.ald.company.lan/repos/yandexbrowser latest Release.gpg
Пол:6 https://repo.ald.company.lan/repos/yandexbrowser latest/main amd64
↪Packages [854 B]
Сущ:8 https://dl.astralinux.ru/aldpro/stable/repository-main 2.4.0 InRelease
Сущ:9 https://dl.astralinux.ru/aldpro/stable/repository-extended generic
↪InRelease
Получено 1 719 B за 0c (4 396 B/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Все пакеты имеют последние версии.
```

Установить браузер из нового репозитория командой:

```
sudo apt install yandex-browser-stable -y
```

После установки, браузер появится в меню Пуск - Сеть, см. *Браузер был установлен из корпоративного репозитория*.

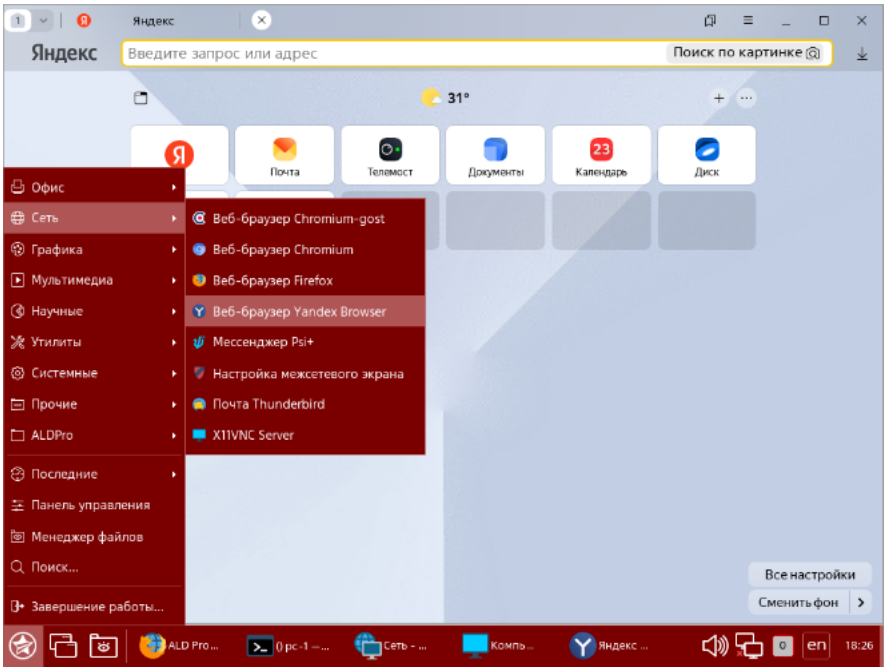


Рисунок 2.21 – Браузер был установлен из корпоративного репозитория

Настроим возможность Kerberos-аутентификации для браузера Яндекс:

```
#создать категорию так как по умолчанию её не существует
sudo mkdir -p /etc/opt/yandex/browser/policies/managed/
sudo nano /etc/opt/yandex/browser/policies/managed/managed_policies.json
```

Вставить в файл *managed_policies.json* следующее содержимое:

```
{
    "AuthServerAllowlist": "*.ald.company.lan",
    "AuthNegotiateDelegateAllowlist": "*.ald.company.lan",
    "HomepageLocation": "https://dc-1.ald.company.lan/"
}
```

Примечание: Также системный администратор может управлять браузером Яндекс через политики. Более подробно с политиками можете ознакомиться в справке браузера: https://yandex.ru/support/browser-corporate/policy/list.html. Посмотреть список примененных политик можно перейдя по адресу browser://policy/

2.9. Установка подсистемы общего доступа к файлам

Установка сервера подсистемы общего доступа к файлам выполняется в два шага:

1. Сначала сервер вводится в домен как обычная рабочая станция с именем **files**.
2. Затем на портале управления серверу назначается роль **Сервер общего доступа к файлам**, установка и настройка происходит автоматически благодаря системе автоматизации.

2.9.1. Установка сервера общего доступа к файлам

Подготовить сервер общего доступа к файлам (минимальные характеристики см. табл. 1 пункт подсистема общего доступа к файлам).

На основании раздела ввода в домен настроить сеть с параметрами:

- IP адрес — 10.0.1.14
- Маска подсети — 255.255.255.0;
- Шлюз — 10.0.1.1;
- Сервер DNS — 10.0.1.11.
- Поисковый домен: ald.company.lan

Проверить доступность репозиториев **dl.astralinux.ru** и ответ от контроллера домена **dc-1**:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

Настроить репозитории и обновить программное обеспечение см. *Настройка доступных репозиториев*.

Установить клиентский пакет программ ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Ввести сервер общего доступа к файлам **files** в домен **dc-1**:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain
ald.company.lan --account admin --password 'AstraLinux_176' --host files --
↪gui --force
set -o history
```

Перезагрузить сервер **files**, чтобы настройки вступили в силу:

```
sudo reboot
```

В портале ALD Pro в разделе **Роли и службы сайта > Общий доступ к файлам** на вкладке **Перечень серверов** развернуть сервер общего доступа, нажав на кнопку [**Новый сервер**]. В окне нового сервера выбрать **files.ald.company.lan**, привязав его к сайту «**Головной офис**», см. *Добавление сервера общего доступа к к файлам*. Чтобы применить изменения, нажмите на кнопку «**Сохранить**».

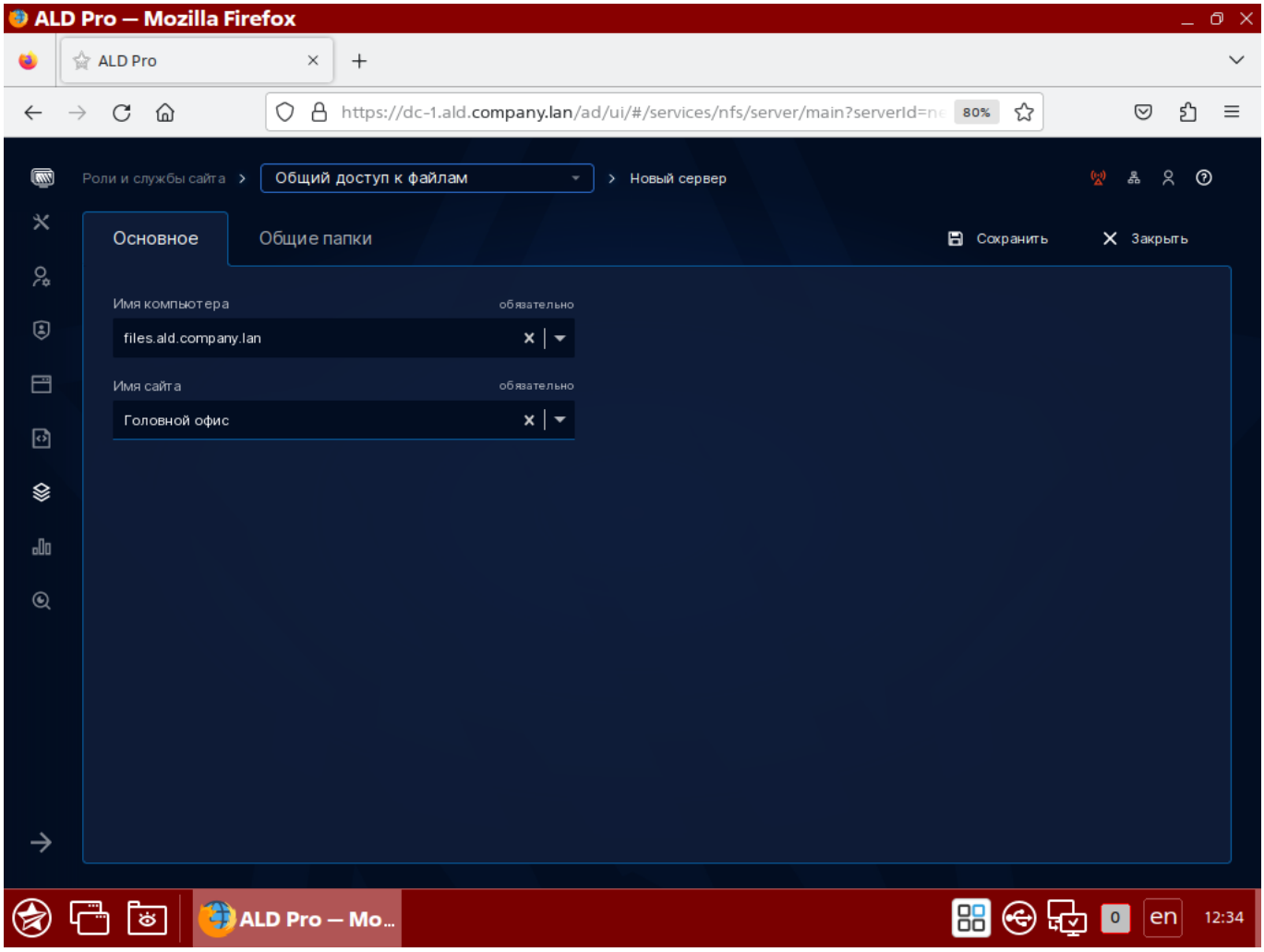


Рисунок 2.22 – Добавление сервера общего доступа к к файлам

В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

Проверить выполнение скрипта автоматизации можно в карточке подсистемы.

2.9.2. Добавление общего ресурса и назначение прав доступа

После установки подсистемы Общего доступа к файлам, на сервере **files** создается общая папка *shared*, в которую можно зайти доменным пользователям после получения билета Kerbeos по адресу «smb://files.ald.company.lan/shared», введя его в поле адресной строки приложения «**Менеджер файлов**» **fly-fm**, см. *Общая папка shared на сервере files.ald.company.lan*.

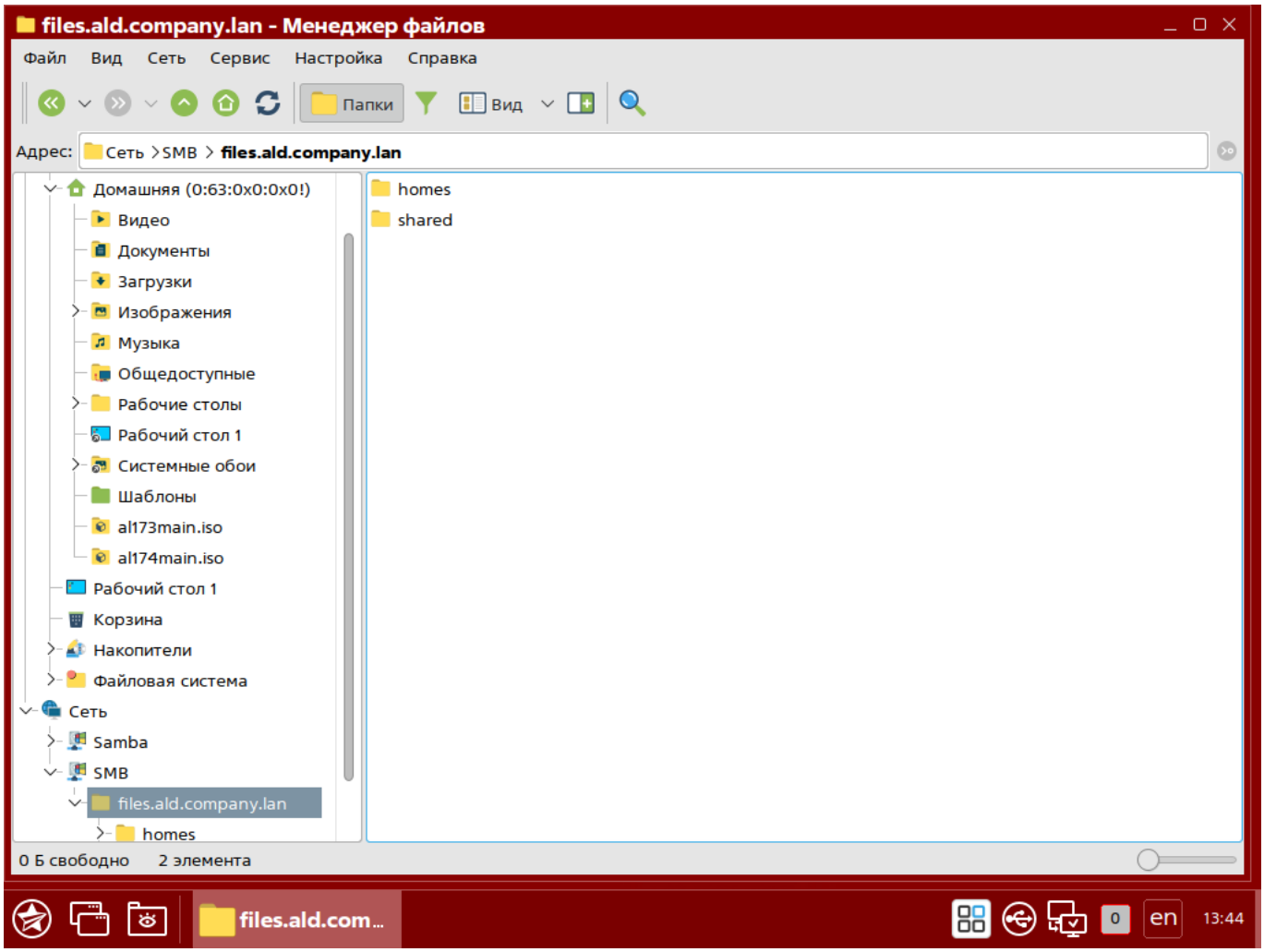


Рисунок 2.23 – Общая папка shared на сервере files.ald.company.lan

2.9.3. Конфигурация

2.9.3.1. Изменить настройки сетевых папок

- Изменить шаблон конфигурации для новой папки, включить «map archive = no» и явно задавать все необходимые параметры, даже если их значения не заданы
- Уровень «Полный доступ» переименовать в «Полный административный доступ» и дополнительно включать пользователей/группы с этим уровнем доступа в список «write list», чтобы пользователи не теряли возможность редактирования файлов, если им через другую группу назначен уровень «Чтение»
- Добавить уровень «Изменение и назначение прав», включать пользователей/группы с этим уровнем в список «write list», но не включать в «admin users».
- Добавить уровень «Доступ запрещен», включать пользователей/группы с этим уровнем в список «invalid users» и НЕ включать в список «valid users»

2.9.3.2. Права доступа на новую папку

Следующие параметры должны быть обязательными для общих папок, настраиваемых через интерфейс ALD Pro: «path», «map archive», «browseable», «writable», «valid users», «invalid users», «read list», «write list» и «admin users»:

```
[test_share]
path = /opt/samba_shares/test_share
map archive = no
browseable = yes
writable = yes
valid users = @NOBODY
invalid users =
read list =
write list =
admin users =
```

Для того, чтобы при создании новых файлов владельцу не устанавливался флаг Execute, нужно отключить сопоставление DOS атрибута Archive с помощью параметра «map archive = no».

Для того, чтобы все пользователи не получали доступ к общему ресурсу, пока список «valid users» не определен, в него нужно вписывать какую-то служебную группу, например,

NOBODY. В настоящий момент мы, по сути, так и делаем, используя в списке «valid users» несуществующую группу «admin», но ее название выглядит как ошибка, потому что группы «admin» не существует, есть группа «admins»:

```
[test_share]
path = /opt/samba_shares/test_share
browseable = yes
valid users = @admin
```

2.9.3.3. Уровень «Чтение»

Этот уровень аналогичен уровню «Allow Read» файлового сервера Windows. Доступ на чтение работает корректно: если группе viewers назначен доступ на чтение, то включаем ее в списки «valid users» и «read list», а группу NOBODY в этом случае можно убрать:

```
[test_share]
path = /opt/samba_shares/test_share
browseable = yes
writable = yes
valid users = @viewers
invalid users =
read list = @viewers
write list =
admin users =
```

Пользователь видит папку, но доступа на создание, редактирования и удаления нет.

2.9.3.4. Изменить уровень «Полный доступ»

Этот уровень не имеет аналога на файловом сервере Windows, т.к. пользователи и группы включаются в список «admin users», который позволяет им действовать на файловом сервере от имени root в обход списков доступа файловой системы.

Превращать этот уровень в настоящий «Полный доступ», как он работает в Windows будет неправильно из соображений обратной совместимости. Правильнее будет переименовать его в «Root доступ» или «Полный административный доступ» и оставить таким пользователям/группам участие в списке «admin users».

Однако, кроме участи в списках «valid users» и «admin users», такие субъекты нужно

обязательно включать в список «write list», иначе они будут терять возможность создавать и редактировать файлы, когда им через какую-то другую группу назначат уровень «Чтение».

```
[test_share]
path = /opt/samba_shares/test_share
browseable = yes
writable = yes
valid users = @viewers @admins
invalid users =
read list = @viewers
write list = @admins
admin users = @admins
```

Пользователь видит папку, доступы на создание, редактирования и удаления есть.

2.9.3.5. Новый уровень «Изменение и назначение прав»

Этот уровень будет аналогичен уровню «Allow Full Control» файлового сервера Windows, который предполагает, что у пользователя есть не только права на запись, но и права на изменение прав доступа к объектам. Его можно назвать «Полный доступ в соответствии с ACL файловой системы».

Если группе editors назначен «Полный доступ в соответствии с ACL файловой системы», то включаем ее в списки «valid users» и «write list». Ни в коем случае не добавляем ее в список «admin users», чтобы пользователь не мог обходить списки доступа файловой системы:

```
[test_share]
path = /opt/samba_shares/test_share
browseable = yes
writable = yes
valid users = @viewers @admins @editors
invalid users =
read list = @viewers
write list = @admins @editors
admin users = @admins
```

Сценарий тестирования: 1. Назначаем на шару роль editor пользователю или группе 2. Форсируем конфигурацию на smb сервере 3. Под учетной записью с правами на sudo

выполняем на smb сервере sudo -i (это не обязательно пользователь из п1) 4. Создаем внутри шары папку mydir 5. Назначаем ей acl для пользователя/группы из п1

- Пример для пользователя my_user

```
setfacl -m u:my_user:rwx /path/to/your/folder
```

- Пример для группы my_group

```
setfacl -m g:my_group:rwx /path/to/your/folder
```

6. Авторизуемся под my_user на клиенте и подключаем самба шару
7. Так как на этапе 5 мы разрешили чтение и запись со стороны файловой системы, а на этапе 1 разрешили запись (на самом деле чтение и запись), то внутри каталога, созданного в п4 пользователь сможет создавать и изменять файлы. При этом, если пропустить пункт 5, то создавать файлы он не сможет

Если создается директория в папке общего доступа, и если в ней создается файл под системной УЗ, даже при выданных acl правах у пользователя не будет прав на редактирование файла, но удаление файла будет доступно. Связанно с тем, что при выдаче acl выдаются права на запись именно на директорию, и этого достаточно, чтоб удалить файл, созданный системной УЗ.

2.9.3.6. Новый уровень «Доступ запрещен»

Этот уровень будет аналогичен уровням «Deny Read», «Deny Change» и «Deny Full Control» файлового сервера Windows. Если группе guests назначен уровень «Доступ запрещен», то она должна быть включена в список «invalid users»:

```
[test_share]
path = /opt/samba_shares/test_share
browseable = yes
writable = yes
valid users = @viewers @admins @editors
invalid users = @guests
read list = @viewers
write list = @admins @editors
admin users = @admins
```

Пользователь не видит папку, соответственно доступов на создание, редактирования и удаления нет.

2.9.3.7. Дополнительные параметры

Добавить в свойства сетевой папки параметры «Тип виртуальной файловой системы», «Права доступа для новых файлов и папок» и «Наследовать права доступа»

Дополнительный параметр «Тип виртуальной файловой системы» В настройки общей сетевой папки нужно добавить переключатель «Тип виртуальной файловой системы» со следующими значениями:

POSIX ACL — использование стандартных механизмов POSIX ACL без подключения дополнительных модулей виртуальной файловой системы, что обеспечивает полную поддержку Linux и базовую совместимость с Windows. При выборе этого типа виртуальной файловой системы параметр vfs objects не должен содержать модулей ACL:

```
[test_share]
...
vfs objects =
```

ACL XATTR — подключение модуля acl_xattr, который обеспечивает хранение Windows ACL, используя базовые механизмы POSIX ACL, что обеспечивает максимально возможную совместимость моделей безопасности Linux и Windows. При выборе этого типа виртуальной файловой системы параметр vfs objects должен содержать модуль acl_xattr:

```
[test_share]
...
vfs objects = acl_xattr
```

NFS4 ACLs — подключение модуля nfs4acl_xattr, который хранит списки доступа NFS4 в виде бинарных объектов в расширенных атрибутах файлов (EAs/xattrs), что обеспечивает полную поддержку Windows без поддержки Linux. При выборе этого типа виртуальной файловой системы параметр vfs objects должен содержать модуль nfs4acl_xattr:

```
[test_share]
...
vfs objects = nfs4acl_xattr
```

2.9.3.8. Дополнительный параметр «Права доступа для новых файлов и папок»

При создании новых файлов/папок через SMB подключение им назначаются права доступа «rw-r–r–» и «rwxr-xr-x» соответственно, что подходит для личных папок, но не подходит для совместной работы, поэтому в настройки сетевой папки нужно добавить переключатель «Права доступа на файлы/папки» со следующими значениями:

- Права на файлы — устанавливает параметр «force create mode», по умолчанию «0777»
- Права на папки — устанавливает параметр «force directory mode», по умолчанию «0777»

2.9.3.9. Дополнительный флажок «Наследовать права доступа»

В ряде случаев намного удобнее будет выставлять не фиксированные права доступа, а управлять ими через механизм наследования, поэтому в настройки сетевой папки нужно добавить флажок «Наследовать права доступа». Если этот флажок включен, то значения параметров «force create mode» и «force directory mode» игнорируются, поэтому соответствующие поля можно делать неактивными. Параметры можно оставить в настройках сетевой папки в файле smb.conf, но правильно будет исключать, чтобы не вводить пользователей в замешательство.

При установке флажка «Наследовать права доступа» в конфигурационном файле smb.conf должны быть заданы следующие параметры:

```
[test_share]
...
inherit permissions = yes
inherit acls = yes
```

2.9.3.10. Управление папками

Если общую папку удалить, то локальная папка остается.

path:

Описание: Указывает на директорию на сервере Samba, которая будет доступна по сети. Если изменить этот параметр, сетевая папка будет указывать на другую локальную папку. Если по пути локальной папки нет, то она создастся Пример: path =

/home/samba/shared

browseable:

Описание: Определяет, будет ли эта папка доступна для просмотра в браузерах сетевых ресурсов (например, в проводнике Windows). Значения: *yes*: папка будет доступна для просмотра. no: папка будет скрыта. Пример: browseable = yes

access based share enum (ABE):

Описание: Позволяет создавать правила доступа к папке на основе пользователей и групп. Например, можно разрешить только чтение для определенной группы, а другим - запись. Значения: yes: ABE включен. *no*: ABE отключен. Пример: access based share enum = yes

vfs objects:

Описание: Список модулей, которые расширяют функциональность Samba. Например, модуль *streams_xattr* позволяет работать с расширенными атрибутами файлов. Пример: vfs objects = streams_xattr

force directory mode:

Описание: Устанавливает режим доступа (права) для всех новых подкаталогов, создаваемых в данной папке. Значения: Число, представляющее собой режим доступа в восьмеричной системе счисления. Пример: force directory mode = 0755 (чтение, запись для владельца, чтение для группы, чтение для других)

force create mode:

Описание: Устанавливает режим доступа для всех новых файлов, создаваемых в данной папке. Значения: Число, представляющее собой режим доступа в восьмеричной системе счисления. Пример: force create mode = 0644 (чтение, запись для владельца, чтение для группы, чтение для других)

inherit permissions:

Описание: Указывает, будут ли права доступа к файлам и папкам наследуемы от родительской папки. Значения: *yes*: права наследуются. no: права не наследуются. Пример: inherit permissions = yes

inherit acls:

Описание: Указывает, будут ли ACL (списки управления доступом) наследуемы от родительской папки. Значения: yes: ACL наследуются. *no*: ACL не наследуются. Пример: inherit acls = yes

2.9.3.11. Новые права доступа

Доступ запрещен invalid Пример для такого параметра в конфигурации шары

```
invalid users = user1
```

Полный доступ fullcontrol Пример для такого параметра в конфигурации шары

```
valid users = user1
admin users = user1
```

Изменение и назначение прав editor Пример для такого параметра в конфигурации шары

```
valid users = user1
write list = user1
```

Чтение viewer Пример для такого параметра в конфигурации шары

```
valid users = user1
read list = user1
```

Если для шары не задано ни одного из значений выше, то

```
valid users = @NOBODY
```

Данные для конфигурации из ldap попадают в файл:
/etc/aldpro-salt/stack/manage/samba.yml

Конфигурация самбы лежит в: /etc/samba/smb.conf

При отсутствии атрибута не должны вписываться в конфиг. те, если для шары не заданы access based share enum, то в конфиге должно быть не

```
access based share enum =
```

а данный атрибут должен быть не задан

Для изменения префикса создаем ямл файл в директории /etc/aldpro-salt/stack/manage/ с содержимым

```
aldpro_deploy_data:
  config:
    smb_path: /path/to/file
```

если это сделать, то все шары будут иметь в параметры path smb_path+rbtaSubsystemFSLocalPath. Если smb_path не существует, то /opt/samba_shares + rbtaSubsystemFSLocalPath.

2.9.3.12. Аутентификация

passdb backend больше не используется. теперь конфигурация самбы генерируется по шаблону:

```
[global]
workgroup = POOL-10
realm = POOL-10.ALDPRO-TEAM.ASTRALINUX.RU
dedicated keytab file = FILE:/etc/samba/samba.keytab
kerberos method = dedicated keytab
idmap config POOL-10 : 1457400000-1457599999
idmap config POOL-10 : backend = sss
idmap config * : range = 0 - 0
log file = /var/log/samba/log.%m
include = registry
include = /etc/samba/share.conf
```

```
[homes]
browsable = yes
writable = yes
create mask = 0600
directory mask = 0700
valid users = %S
read only = No
guest ok = no
```

где: POOL-10 - это значение до первой точке в имени домена в верхнем регистре. 1457400000-1457599999 - это диапазон id. Он считается по формуле из атрибутов записи cn=POOL-10.ALDPRO-TEAM.ASTRALINUX.RU_id_range,cn=ranges,cn=etc,dc=pool-10,dc=aldpro-team,dc=astralinux,dc=ru: первое число - ipaBaseID второе - ipaBaseID + ipaIDRangeSize - 1 include = registry - активирует работу регистра. позволяет

переопределять в нем конфигурации include = /etc/samba/share.conf - подключает конфигурационный файл с шарами. теперь конфигурации шар будут писаться сюда

2.9.4. Команды для форсирования

2.9.4.1. Установка подсистемы

Для форсированной установки на машины с подсистемой необходимо сначала сформировать переменные окружения:

```
aldpro-salt-call aldpro_subsystems.build_deploy_pillar -c /srv/aldpro-salt/
↪config/
```

в результате будет создан файл */etc/aldpro-salt/stack/deploy/subsystem.yml* *aldpro_deploy_data*:

```
action:install
  location: hq
  service_name: smb
  site: Головной офис
  state_created: 20240621131248Z
  state_updated: 20240621131248Z
  target_host: os01.pool-07.aldpro-team.astralinux.ru
```

Далее запустить код установки подсистемы:

```
aldpro-salt-call state.orchestrate orch.subsystems -c /srv/aldpro-salt/config/
↪
```

Применение этих параметров выполняется раз в 30 минут. Ручной запуск:

```
aldpro-salt-call state.apply roles.smb -c /srv/aldpro-salt/config/
```

2.9.4.2. Команды

Реализовать команду форсирования установки/обновления КД/подсистемы на хосте:

`aldpro-roles --iud` - Собирает pillar, переменных окружения из ldap, запускает

установку/обновление/удаление;

`aldpro-roles --iud --action [install, update, remove]` - Собирает pillar, переменных окружения из ldap, заменяя action на указанный, запускает установку/обновление/удаление, согласно указанному action. Пример: `aldpro-roles --iud --action update`;

`aldpro-roles --subsystem_settings smb` - Собирает pillar переменных окружения из ldap, применяет параметры для указанной подсистемы. Пример: `aldpro-iud --subsystem_settings cups`;

Внимание: Запрещено одновременно использовать аргументы по установке подсистемы и получению параметров как указано ниже:

```
aldpro-iud --iud --action update --subsystem_settings dhcp
```

2.10. Установка подсистемы сервера печати

Подготовить сервер печати в соответствии с минимальными требованиями см. табл. 1 подсистема службы печати.

На основании раздела ввода в домен настроить сеть с такими сетевыми параметрами:

- IP адрес — 10.0.1.15
- Маска подсети — 255.255.255.0;
- Шлюз — 10.0.1.1;
- Сервер DNS — 10.0.1.11.
- Поисковый домен: ald.company.lan

Проверить доступность репозиториев **dl.astralinux.ru** и ответ от контроллера домена **dc-1**:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

Настроить репозитории и обновить программное обеспечение см. *Настройка доступных репозиториев*.

Установить клиентский пакет программ ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Ввести сервер печати **cups** в домен **dc-1**:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.
↪lan --account admin --password 'AstraLinux_176' --host cups --gui --force
set -o history
```

Перезагрузить сервер **cups**, чтобы настройки вступили в силу:

```
sudo reboot
```

В портале ALD Pro в разделе **Роли и службы сайта > Служба печати** на вкладке **Серверы печати** развернуть сервер общего доступа **cups**, нажав на кнопку [**Новый сервер печати**], привязать его к сайту «**Головной офис**» см. *Добавление сервера печати*. Нажмите кнопку «Сохранить», чтобы применить изменения.

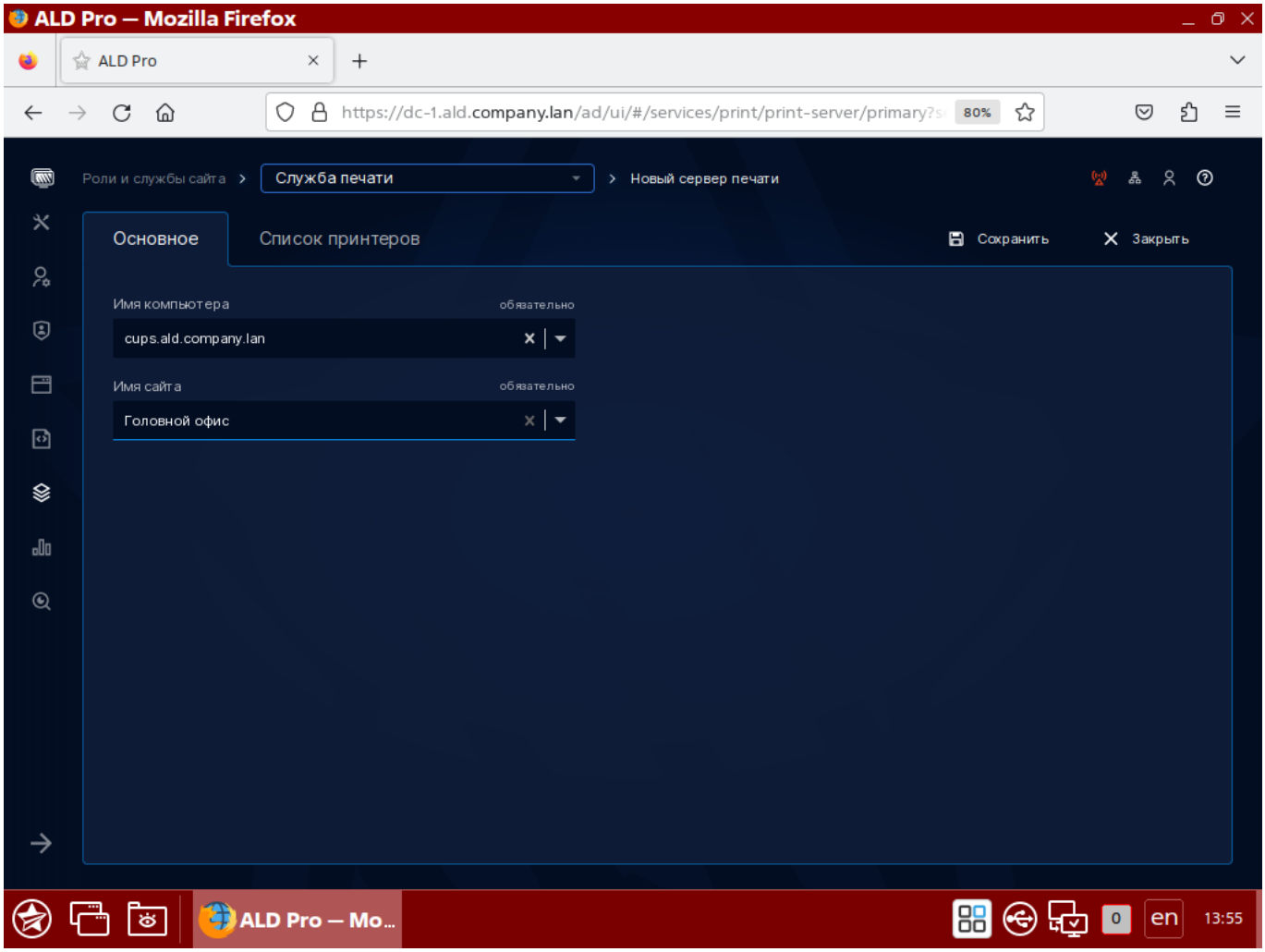


Рисунок 2.24 – Добавление сервера печати

В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

Проверить выполнение скрипта автоматизации можно также в разделе **Автоматизация — Задания автоматизации** на вкладке **Журнал заданий**.

2.10.1. Команды для форсирования

2.10.1.1. Установка подсистемы

Для форсированной установки на машины с подсистемой необходимо сначала сформировать переменные окружения:

```
aldpro-salt-call aldpro_subsystems.build_deploy_pillar -c /srv/aldpro-salt/
↪config/
```

в результате будет создан файл */etc/aldpro-salt/stack/deploy/subsystem.yml*
aldpro_deploy_data:

```
action:install
  location: hq
  service_name: smb
  site: Головной офис
  state_created: 20240621131248Z
  state_updated: 20240621131248Z
  target_host: os01.pool-07.aldpro-team.astralinux.ru
```

Далее запустить код установки подсистемы:

```
aldpro-salt-call state.orchestrate orch.subsystems -c /srv/aldpro-salt/config/
↪
```

Применение этих параметров выполняется раз в 30 минут. Ручной запуск:

```
aldpro-salt-call state.apply roles.smb -c /srv/aldpro-salt/config/
```

2.10.1.2. Команды

Реализовать команду форсирования установки/обновления КД/подсистемы на хосте:

`aldpro-roles --iud` - Собирает pillar, переменных окружения из ldap, запускает установку/обновление/удаление;

`aldpro-roles --iud --action [install, update, remove]` - Собирает pillar, переменных окружения из ldap, заменяя action на указанный, запускает установку/обновление/удаление, согласно указанному action. Пример: `aldpro-roles --iud --action update`;

`aldpro-roles --subsystem_settings smb` - Собирает pillar переменных окружения из ldap, применяет параметры для указанной подсистемы. Пример: `aldpro-iud --subsystem_settings cups`;

Внимание: Запрещено одновременно использовать аргументы по установке подсистемы и получению параметров как указано ниже:

```
aldpro-iud --iud --action update --subsystem_settings dhcp
```

2.11. Установка подсистемы «Динамическая настройка узла» DHCP

Подготовить сервер динамической настройки узлов в соответствии с минимальными требованиями, см. табл. 1.

На основании раздела ввода в домен настроить сеть с такими сетевыми параметрами:

- IP адрес — 10.0.1.16
- Маска подсети — 255.255.255.0;
- Шлюз — 10.0.1.1;
- Сервер DNS — 10.0.1.11;
- Поисковый домен: ald.company.lan.

Проверить доступность репозиториев **dl.astralinux.ru** и ответ от контроллера домена **dc-1**:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

Настроить репозитории и обновить программное обеспечение см. *Настройка доступных репозиториев*.

Установить клиентский пакет программ ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Ввести сервер **dhcp** в домен **dc-1**:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.
↪lan --account admin --password 'AstraLinux_176' --host dhcp --gui --force
set -o history
```

Перезагрузить сервер **dhcp**, чтобы настройки вступили в силу:

```
sudo reboot
```

В портале ALD Pro в разделе **Роли и службы сайта > Служба динамической настройки узла** на вкладке **Перечень серверов** развернуть сервер **DHCP**, нажав на кнопку [**Новый сервер**].

Выбрать сервер **dhcp.ald.company.lan** и привязать его к сайту «**Головной офис**» см. *Добавление сервера DHCP*. Нажмите кнопку «**Сохранить**», чтобы применить изменения.

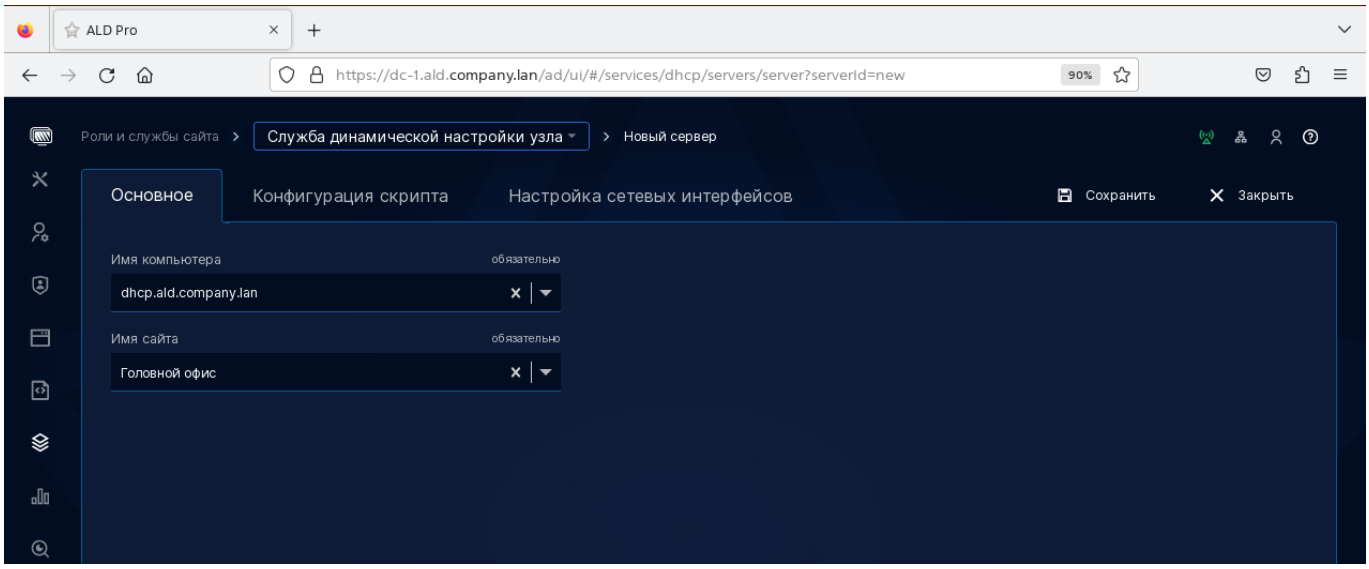


Рисунок 2.25 – Добавление сервера DHCP

В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

Проверить выполнение скрипта автоматизации можно также в разделе **Автоматизация — Задания автоматизации** на вкладке **Журнал заданий**.

2.11.1. Команды для форсирования

2.11.1.1. Установка подсистемы

Для форсированной установки на машины с подсистемой необходимо сначала сформировать переменные окружения:

```
aldpro-salt-call aldpro_subsystems.build_deploy_pillar -c /srv/aldpro-salt/
↪config/
```

в результате будет создан файл */etc/aldpro-salt/stack/deploy/subsystem.yml* *aldpro_deploy_data*:

```
action:install
  location: hq
  service_name: smb
  site: Головной офис
  state_created: 202406211312487
  state_updated: 202406211312487
  target_host: os01.pool-07.aldpro-team.astralinux.ru
```

Далее запустить код установки подсистемы:

```
aldpro-salt-call state.orchestrate orch.subsystems -c /srv/aldpro-salt/config/
↪
```

Применение этих параметров выполняется раз в 30 минут. Ручной запуск:

```
aldpro-salt-call state.apply roles.smb -c /srv/aldpro-salt/config/
```

2.11.1.2. Команды

Реализовать команду форсирования установки/обновления КД/подсистемы на хосте:

`aldpro-roles --iud` - Собирает pillar, переменных окружения из ldap, запускает установку/обновление/удаление;

`aldpro-roles --iud --action [install, update, remove]` - Собирает pillar, переменных окружения из ldap, заменяя action на указанный, запускает

установку/обновление/удаление, согласно указанному action. Пример: `aldpro-roles --iud --action update`;

`aldpro-roles --subsystem_settings smb` - Собирает pillar переменных окружения из ldap, применяет параметры для указанной подсистемы. Пример: `aldpro-iud --subsystem_settings cups`;

> **Внимание:** Запрещено одновременно использовать аргументы по установке подсистемы и получению параметров как указано ниже:
>
> ```
> aldpro-iud --iud --action update --subsystem_settings dhcp
> ```

2.12. Установка подсистемы «Установка ОС по сети» TFTP + PXE

Подготовить сервер подсистемы установки ОС по сети в соответствии с минимальными требованиями, см. табл. 1.

На основании раздела ввода в домен настроить сеть с такими сетевыми параметрами:

- IP адрес — 10.0.1.17
- Маска подсети — 255.255.255.0;
- Шлюз — 10.0.1.1;
- Сервер DNS — 10.0.1.11;
- Поисковый домен: ald.company.lan.

Проверить доступность репозиториев **dl.astralinux.ru** и ответ от контроллера домена **dc-1**:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

Настроить репозитории и обновить программное обеспечение см. *Настройка доступных репозиториев*.

Установить клиентский пакет программ ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Ввести сервер установки ОС по сети **pxe** в домен **dc-1**:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.
↪lan --account admin --password 'AstraLinux_176' --host pxe --gui --force
set -o history
```

Перезагрузить сервер **pxe**, чтобы настройки вступили в силу:

```
sudo reboot
```

В портале ALD Pro в разделе **Автоматизация > Установка ОС по сети** на вкладке **Серверы установки ОС** развернуть сервер **pxe**, нажав на кнопку [**Новый сервер**].

Выбрать сервер **pxe.ald.company.lan**, привязав его к сайту «**Головной офис**». А также выбрать DHCP-сервер **dhcp.ald.company.lan**, который будет настраивать параметры новым сетевым устройствам. см. *Добавление нового сервера установки ОС по сети PXE*. Нажмите кнопку «**Сохранить**», чтобы применить изменения.

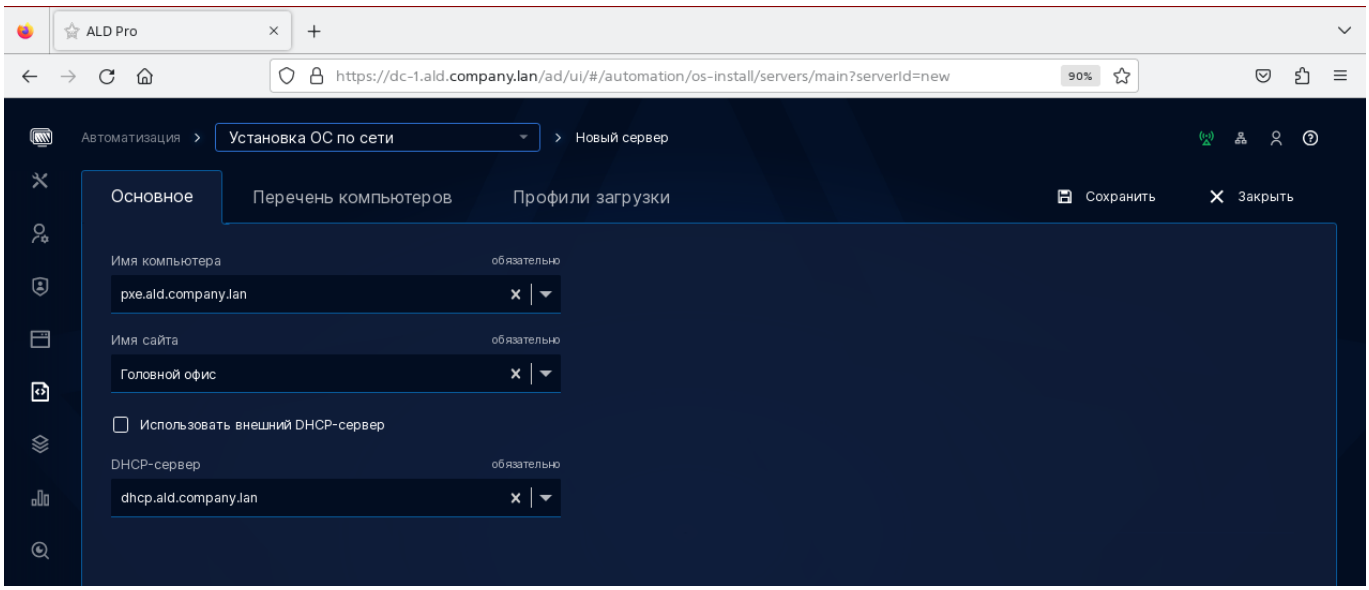


Рисунок 2.26 – Добавление нового сервера установки ОС по сети PXE

В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

Проверить выполнение скрипта автоматизации можно также в разделе **Автоматизация — Задания автоматизации** на вкладке **Журнал заданий**.

2.12.1. Команды для форсирования

2.12.1.1. Установка подсистемы

Для форсированной установки на машины с подсистемой необходимо сначала сформировать переменные окружения:

```
aldpro-salt-call aldpro_subsystems.build_deploy_pillar -c /srv/aldpro-salt/
↪config/
```

в результате будет создан файл */etc/aldpro-salt/stack/deploy/subsystem.yml* *aldpro_deploy_data*:

```
action:install
  location: hq
  service_name: smb
  site: Головной офис
  state_created: 20240621131248Z
  state_updated: 20240621131248Z
  target_host: os01.pool-07.aldpro-team.astralinux.ru
```

Далее запустить код установки подсистемы:

```
aldpro-salt-call state.orchestrate orch.subsystems -c /srv/aldpro-salt/config/
↪
```

Применение этих параметров выполняется раз в 30 минут. Ручной запуск:

```
aldpro-salt-call state.apply roles.smb -c /srv/aldpro-salt/config/
```

2.12.1.2. Команды

Реализовать команду форсирования установки/обновления КД/подсистемы на хосте:

`aldpro-roles --iud` - Собирает pillar, переменных окружения из ldap, запускает установку/обновление/удаление;

`aldpro-roles --iud --action [install, update, remove]` - Собирает pillar, переменных окружения из ldap, заменяя action на указанный, запускает установку/обновление/удаление, согласно указанному action. Пример: `aldpro-roles --iud --action update`;

`aldpro-roles --subsystem_settings smb` - Собирает pillar переменных окружения из ldap, применяет параметры для указанной подсистемы. Пример: `aldpro-iud --subsystem_settings cups`;

Внимание: Запрещено одновременно использовать аргументы по установке подсистемы и получению параметров как указано ниже:

```
aldpro-iud --iud --action update --subsystem_settings dhcp
```

2.13. Установка подсистемы мониторинга

Примечание: Подсистема мониторинга домена, в версиях ALD Pro предшествующих 2.4.0, не установится на ALSE 1.7.6.

Подготовить сервер подсистемы мониторинга в соответствии с минимальными требованиями, см. табл. 1.

На основании раздела ввода в домен настроить сеть с такими сетевыми параметрами:

- IP адрес — 10.0.1.18;
- Маска подсети — 255.255.255.0;
- Шлюз — 10.0.1.1;
- Сервер DNS — 10.0.1.11;

- Поисковый домен: ald.company.lan.

Проверить доступность репозиториев **dl.astralinux.ru** и ответ от контроллера домена **dc-1**:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

Настроить репозитории и обновить программное обеспечение см. *Настройка доступных репозиториев*.

Установить клиентский пакет программ ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Ввести сервер мониторинга **mon** в домен **dc-1**:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.
↪lan --account admin --password 'AstraLinux_176' --host mon --gui --force
set -o history
```

Перезагрузить сервер **mon**, чтобы настройки вступили в силу:

```
sudo reboot
```

На портале ALD Pro в разделе **Мониторинг > Журнал событий мониторинга** на вкладке **Серверы мониторинга** развернуть сервер мониторинга **mon**, нажав на кнопку [**Развернуть сервер мониторинга**].

В окне нового сервера выбрать **mon.ald.company.lan**, привязав его к сайту «**Головной офис**», см. *Добавление нового сервера мониторинга*. Нажмите кнопку «**Сохранить**», чтобы применить изменения.

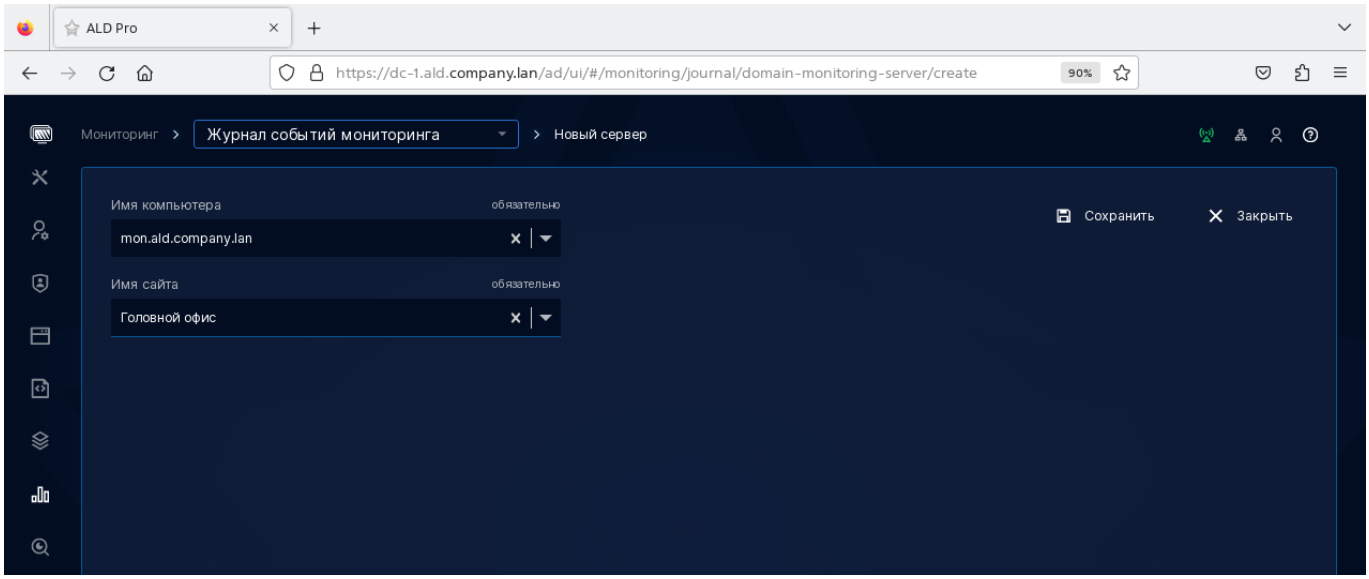


Рисунок 2.27 – Добавление нового сервера мониторинга

В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

Проверить выполнение скрипта автоматизации можно также в разделе **Автоматизация — Задания автоматизации** на вкладке **Журнал заданий**.

2.13.1. Команды для форсирования

2.13.1.1. Установка подсистемы

Для форсированной установки на машины с подсистемой необходимо сначала сформировать переменные окружения:

```
aldpro-salt-call aldpro_subsystems.build_deploy_pillar -c /srv/aldpro-salt/
↪config/
```

в результате будет создан файл */etc/aldpro-salt/stack/deploy/subsystem.yml* *aldpro_deploy_data*:

```
action:install
  location: hq
```

(продолжение на следующей странице)

```
  service_name: smb
  site: Головной офис
  state_created: 20240621131248Z
  state_updated: 20240621131248Z
  target_host: os01.pool-07.aldpro-team.astralinux.ru
```

Далее запустить код установки подсистемы:

```
aldpro-salt-call state.orchestrate orch.subsystems -c /srv/aldpro-salt/config/
↪
```

Применение этих параметров выполняется раз в 30 минут. Ручной запуск:

```
aldpro-salt-call state.apply roles.smb -c /srv/aldpro-salt/config/
```

2.13.1.2. Команды

Реализовать команду форсирования установки/обновления КД/подсистемы на хосте:

`aldpro-roles --iud` - Собирает pillar, переменных окружения из ldap, запускает установку/обновление/удаление;

`aldpro-roles --iud --action [install, update, remove]` - Собирает pillar, переменных окружения из ldap, заменяя action на указанный, запускает установку/обновление/удаление, согласно указанному action. Пример: `aldpro-roles --iud --action update`;

`aldpro-roles --subsystem_settings smb` - Собирает pillar переменных окружения из ldap, применяет параметры для указанной подсистемы. Пример: `aldpro-iud --subsystem_settings cups`;

Внимание: Запрещено одновременно использовать аргументы по установке подсистемы и получению параметров как указано ниже:

```
aldpro-iud --iud --action update --subsystem_settings dhcp
```

2.14. Установка подсистемы журналирования

Подготовить сервер подсистемы мониторинга в соответствии с минимальными требованиями, см. табл. 1.

На основании раздела ввода в домен настроить сеть с такими сетевыми параметрами:

- IP адрес — 10.0.1.19;
- Маска подсети — 255.255.255.0;
- Шлюз — 10.0.1.1;
- Сервер DNS — 10.0.1.11;
- Поисковый домен: ald.company.lan.

Проверить доступность репозиториев **dl.astralinux.ru** и ответ от контроллера домена **dc-1**:

```
ip a
ping -c 4 77.88.8.8
ping -c 4 dl.astralinux.ru
ping -c 4 dc-1.ald.company.lan
ping -c 4 dc-1
```

Настроить репозитории и обновить программное обеспечение см. *Настройка доступных репозиториев*.

Установить клиентский пакет программ ALD Pro:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

Ввести сервер журналирования **audit** в домен **dc-1**:

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.
↪lan --account admin --password 'AstraLinux_176' --host audit --gui --force
set -o history
```

Перезагрузить сервер **audit**, чтобы настройки вступили в силу:

```
sudo reboot
```

В портале ALD Pro в разделе **Журнал событий > Серверы журнала событий** на вкладке

Серверы журнала событий развернуть сервер журналирования **audit**, нажав на кнопку [**Развернуть сервер журнала событий**].

В окне нового сервера выбрать **audit.ald.company.lan**, привязав его к сайту «**Головной офис**», см. *Добавление нового сервера журналирования*. Нажмите кнопку «**Сохранить**», чтобы применить изменения.

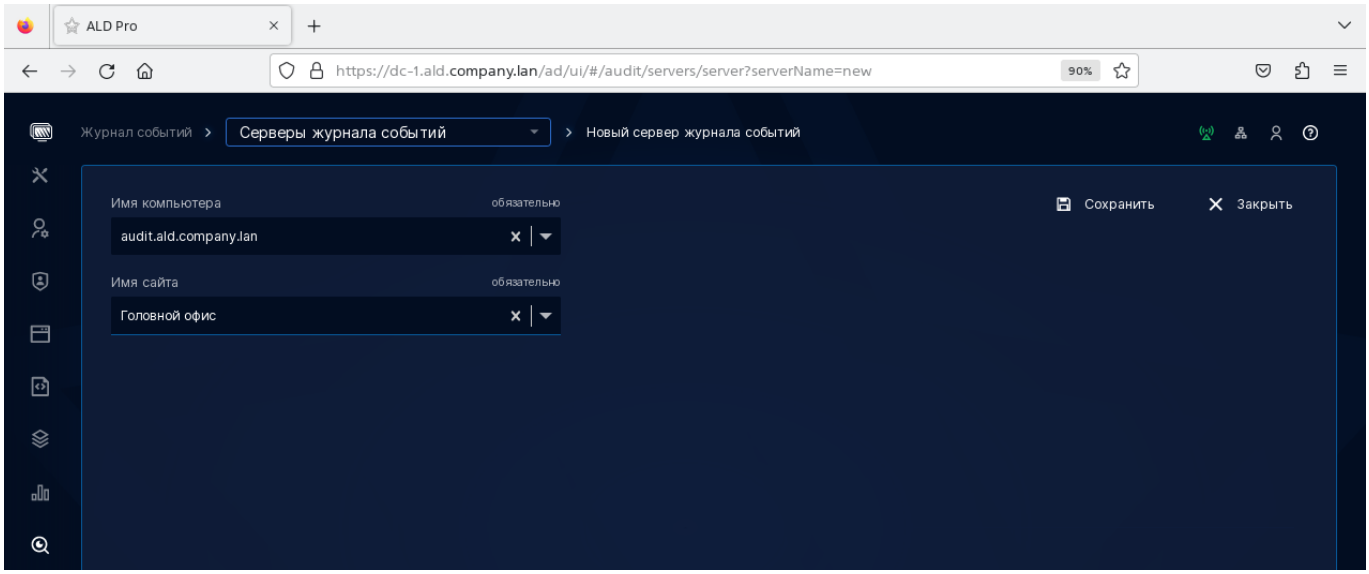


Рисунок 2.28 – Добавление нового сервера журналирования

В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

Проверить выполнение скрипта автоматизации можно также в разделе **Автоматизация — Задания автоматизации** на вкладке **Журнал заданий**.

2.14.1. Команды для форсирования

2.14.1.1. Установка подсистемы

Для форсированной установки на машины с подсистемой необходимо сначала сформировать переменные окружения:

```
aldpro-salt-call aldpro_subsystems.build_deploy_pillar -c /srv/aldpro-salt/
↪config/
```

в результате будет создан файл */etc/aldpro-salt/stack/deploy/subsystem.yml* *aldpro_deploy_data*:

```
action:install
  location: hq
  service_name: smb
  site: Головной офис
  state_created: 202406211131248Z
  state_updated: 202406211131248Z
  target_host: os01.pool-07.aldpro-team.astralinux.ru
```

Далее запустить код установки подсистемы:

```
aldpro-salt-call state.orchestrate orch.subsystems -c /srv/aldpro-salt/config/
↪
```

Применение этих параметров выполняется раз в 30 минут. Ручной запуск:

```
aldpro-salt-call state.apply roles.smb -c /srv/aldpro-salt/config/
```

2.14.1.2. Команды

Реализовать команду форсирования установки/обновления КД/подсистемы на хосте:

`aldpro-roles --iud` - Собирает pillar, переменных окружения из ldap, запускает установку/обновление/удаление;

`aldpro-roles --iud --action [install, update, remove]` - Собирает pillar, переменных окружения из ldap, заменяя action на указанный, запускает установку/обновление/удаление, согласно указанному action. Пример: `aldpro-roles --iud --action update`;

`aldpro-roles --subsystem_settings smb` - Собирает pillar переменных окружения из ldap, применяет параметры для указанной подсистемы. Пример: `aldpro-iud --subsystem_settings cups`;

> **Внимание:** Запрещено одновременно использовать аргументы по установке подсистемы и получению параметров как указано ниже:
> `aldpro-iud --iud --action update --subsystem_settings dhcp`

2.15. Проверка работы сервисов ALD Pro

Для проверки работоспособности сервисов ALD Pro добавлена утилита **ALDProCTL**, которая расширяет функционал **ipactl**, проверяя также сервисы ALD Pro.

Для выполнения команд с сервисами используется `systemctl`.

2.15.1. Конфигурация

Файл конфигурации имеет три обязательных блока: **FreeIPA**, **ALD Pro** и **Other**. Для начала работы необходимо скопировать файл *aldproctl.env* и внести перечень сервисов для проверки по примеру ниже:

```
[FreeIPA]
Directory Service
krb5kdc
kadmin
named
httpd
ipa-custodia
smb
winbind
ipa-otpd
ipa-dnskeysyncd
[ALD Pro]
GLOBAL-CATALOG
GCSync
aldpro-mp-services
aldpro-canclient
ad-salt-canrunner
syncer
syncer.timer
[Other]
celery
celerybeat
```

Примечание: Название сервисов в конфигурационном файле должны полностью соответствовать примеру. Ниже приведен список сервисов, доступных к проверке:

- Directory Service
- krb5kdc
- kadmin
- named
- httpd
- ipa-custodia
- smb
- winbind
- ipa-otpd
- ipa-dnskeysyncd
- dirsrv@GLOBAL-CATALOG
- ipa-gcsync
- celery
- celerybeat
- aldpro-mp-service
- aldpro-canclient
- ad-salt-canrunner
- syncer
- syncer.timer

2.15.2. Перечень команд

Команда	Описание
`sudo aldproctl -h`	Отображает справочное сообщение
`sudo man aldproctl`	Отображает развернутое справочное сообщение с описанием работы утилиты
`sudo aldproctl --version`	Отображает используемую версию ALD Pro
`sudo aldproctl start|stop|restart|status --timeout=TIMEOUT`	Устанавливает время ожидания для выполнения операции. По умолчанию 60 секунд
`sudo aldproctl start`	Запускает сервисы, указанные в конфигурационном файле
`sudo aldproctl status`	Отображает статус сервисов, указанные в конфигурационном файле
`sudo aldproctl stop`	Останавливает работу сервисов, указанные в конфигурационном файле
`sudo aldproctl restart`	Перезапускает сервисы, указанные в конфигурационном файле
`sudo aldproctl start|stop|restart|status --service=service_name`	Флаг service выполняет операцию start|stop|restart|status для service_name (название сервиса)
`sudo aldproctl start|stop|restart|status --ipa-check-only`	Флаг -i, –ipa-check-only - выполняет операцию start|stop|restart|status для сервисов блока FreeIPA
`sudo aldproctl start|stop|restart|status --aldpro-check-only`	Флаг -a, –aldpro-check-only - выполняет операцию start|stop|restart|status для сервисов блока ALD Pro
`sudo aldproctl start|stop|restart|status --other-check-only`	Флаг -o, –other-check-only - выполняет операцию start|stop|restart|status для сервисов блока Other

Таблица 5 — Перечень команд

2.15.3. Варианты отображения статуса сервиса

- ЗАПУЩЕН - команда start сработала без ошибок и сервис запустился

- ОСТАНОВЛЕН - команда stop сработала без ошибок и сервис остановлен
- ПЕРЕЗАПУЩЕН - команда restart сработала без ошибок и сервис перезапущен
- НЕ НАЙДЕН - сервис с таким именем не найден
- НЕАКТИВЕН (МЕРТВ) - сервис неактивен
- ОШИБКА - ошибка при вызове команды или работе сервиса
- ПРЕВЫШЕНО ВРЕМЯ ОЖИДАНИЯ - превышено время выполнения операции

Обновление подсистем

3.1. Обновление подсистем ALD Pro до новой версии

Примечание: Список обновлений, исправное взаимодействие которых гарантированно, представлен в разделе *Матрица совместимости ПК ALD Pro*.

Обновления продукта являются кумулятивными, т.е. возможна установка сразу актуальной версии, без последовательной установки обновлений. Наиболее полное тестирование продукта выполняется для стендов обновленных с предыдущей минорной версии продукта. Описанные действия выполняют обновление всех составляющих домена: контроллера домена, подсистем и клиентской части ALD Pro до версии 2.4.0.

3.1.1. Подготовка к обновлению ALD Pro

Перед обновлением ALD Pro важно учитывать:

- Обновление продукта ALD Pro необходимо выполнять на контроллере домена от имени учетной записи администратора системы с высоким уровнем целостности.
- Обновления следует начинать с **первого** Контроллера Домена. Соответственно, все команды должны вводиться в его консоли (терминале).
- В версии 2.2.0 был полностью обновлен механизм работы групповых политик, политик ПО и мониторинга. Для их корректной работы необходимо обновить ALD Pro на всех компьютерах в домене.
- Начиная с версии 1.3.0 в ALD Pro реализовано разграничение доступа к функциям системы. При обновлении системы до версии 2.3.0 администратору должна быть назначена роль «**ALDPRO — Main Administrator**» (пользователю **admin** роль назначается автоматически), также учетную запись администратора необходимо добавить в группу **ald trust admin**. Остальным пользователям (администраторам) системы соответствующие роли при необходимости нужно назначать в ручном режиме. Подробная информация о работе ролевого доступа находится в Справочном Центре Портала Управления в подразделе «Роли и права доступа» — «Роли в

системе».

- Программное обеспечение ОС Astra Linux Special Edition должно совпадать на контроллерах домена, между которыми настроена репликация. Одновременная установка обновлений ОС и ALD Pro может привести к неработоспособности контроллеров домена, серверов подсистем и клиентов.

Примечание: В случае обновления с ALD Pro **ранее** 2.3.0 на ALSE **ранее** 1.7.5, перед обновлением ОС следует сначала обновить ALD Pro до актуальной версии. Обновлять Глобальный Каталог нужно после обновления Операционной Системы. При возникновении ошибок обновления Глобального Каталога см. *Исправление ошибки Глобального Каталога*.

Если требуется установка оперативного обновления 1.7.6 UU1, она должна быть выполнена перед обновлением ALD Pro. Для установки обновления ОС на всех контроллерах домена необходимо выполнить следующие действия:

1. Подготовить окружение сервера, убедившись, что файл */etc/apt/sources.list* содержит следующие строки, при необходимости — добавить, если имеются другие записи, то закомментировать их или удалить (для корректного копирования команд рекомендуется использовать):

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base 1.7_
↪x86-64 main non-free contrib
```

2. Обновить пакеты ОС, выполнив в терминале команды:

```
sudo apt update
sudo apt install astra-update -y && sudo astra-update -A -r -T
```

3. Для корректной работы функций репликации на контроллере домена необходимо импортировать новые конфигурации службы каталога, выполнив в терминале команды:

```
sudo ipa-server-upgrade
sudo ipactl restart
```

3.1.2. Обновление

Для установки обновления ALD Pro на **первом** Контроллере Домена необходимо подключить репозиторий **aldpro**, выполнив в терминале команды:

```
sudo nano /etc/apt/sources.list.d/aldpro.list
```

Полностью заменить содержимое файла *aldpro.list*:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0/ 1.7_x86-64 main base
```

Обновить индекс пакетов, выполнив в терминале команду:

```
sudo apt update
```

Обновить пакеты продукта ALD Pro командой:

```
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confnew
```

В процессе выполнения обновления, при появлении сообщения с подтверждением изменения файла настройки пакета, необходимо выбрать «Установить версию, предлагаемую сопровождающим пакета», введя в командной строке «**Y**».

Перезагрузить контроллер, выполнив в терминале команду:

```
sudo reboot
```

После перезагрузки первого контроллера домена и проверки статуса прохождения аутентификации, для завершения обновления первого контроллера домена и автоматического обновления остальных контроллеров домена, подсистем и клиентов в терминале выполнить команды:

```
set +o history
sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'AstraLinux_176' --
↪ip 10.0.1.11 --update
set -o history
```

Где:

- -d (domain) — имя домена
- -n (name) — имя сервера

- `-p` (password) — пароль администратора домена
- `--ip` — ip адрес контроллера домена. Адрес требуется указывать явно, если на контроллере домена активно несколько сетевых интерфейсов
- `--update` — ключ запуска процесса обновления

Описание параметров скрипта можно получить с помощью ключа `-h`.

При необходимости инициализации модуля Синхронизации и/или Глобального Каталога добавляются ключи `--setup_syncer` и/или `--setup_gc` соответственно.

После обновления первого КД на Портале Управления обновится информация о подсистемах (рис. рис. 3.1):

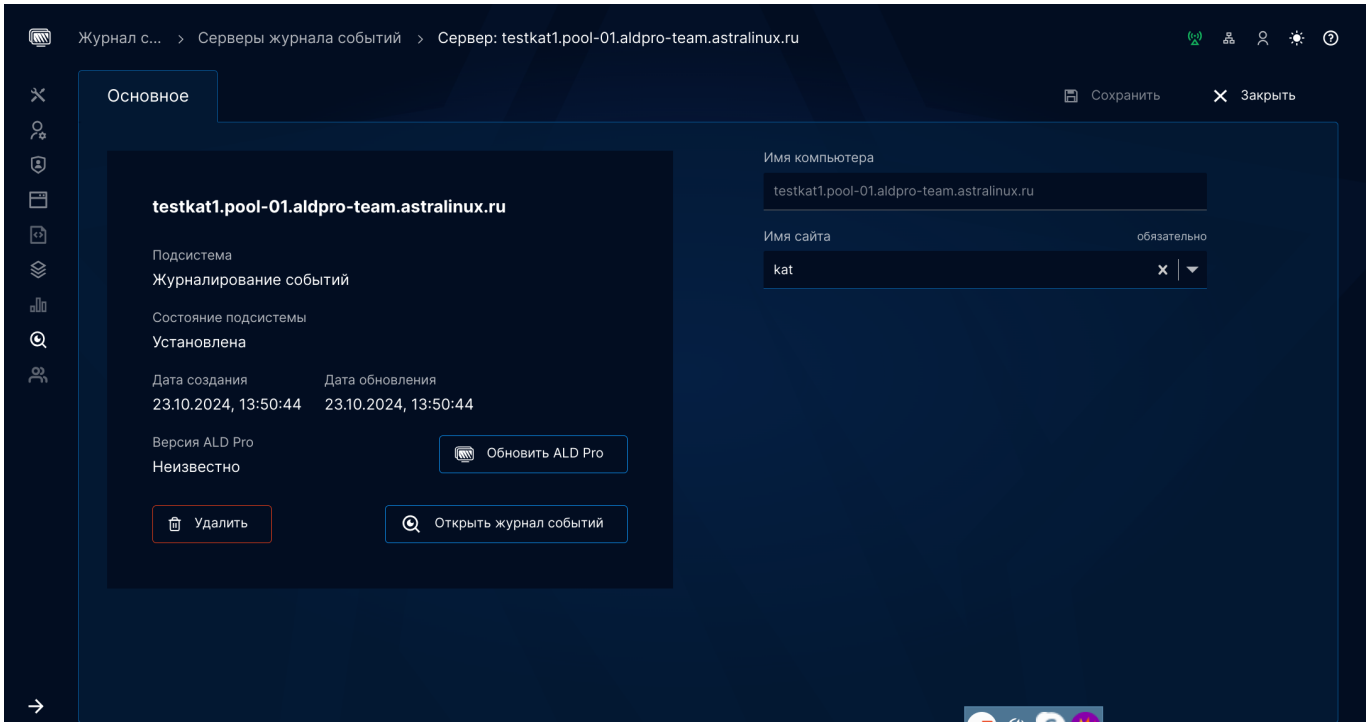


Рисунок 3.1 – Карточка необновленной подсистемы в обновленном домене

3.1.3. Установка и обновление Глобального Каталога и Модуля Синхронизации

Если требуется использовать двустороннее доверительное отношения с Microsoft AD, необходимо установить модули синхронизации и/или глобального каталога. Для этого необходимо добавить ключ `--setup_syncer` и/или `--setup_gc` соответственно, в зависимости от сценария работы с Active Directory.

Модуль синхронизации всегда устанавливается на первый контроллер домен. При

установке модуля синхронизации контроллер домена **ALD Pro** добавляется в карточку **Контроллеры домена ALD** автоматически.

При установке модуля синхронизации создается учетная запись имеющая вид `syncer/dc01.<имя домена>@<ИМЯ ДОМЕНА>`. Данная учетная запись является сервисной и используется для работы модуля синхронизации. Пароль данной учетной записи является бессрочным, что делает работу модуля синхронизации более стабильной и устойчивой.

При обновлении контроллера домена ALD Pro до версии 2.4.0, с ранее установленным модулем синхронизации, процесс миграции Базы Данных может занять продолжительное время. Если БД содержит 30000 пользователей и более, то рекомендуется сначала обновить ALD Pro, затем выбрав отдельный интервал профилактики обновить модуль синхронизации, согласно *Инструкция по обновлению ALD Pro до версии 2.4.0 с ранее установленным модулем синхронизации*.

Внимание: При обновлении до 2.4.0 учетная запись текущего подключения изменится на сервисную.

Важно: Для установки глобального каталога необходима ОС Astra Linux очередного обновления 1.7 с установленным оперативным обновлением не ранее 1.7.4

Команды установки модулей синхронизации и глобального каталога на контроллере домена:

```
set +o history
sudo apt update && sudo apt install aldpro-gc aldpro-syncer
sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'AstraLinux_176' --
↪ip 10.0.1.11 --update --setup_syncer --setup_gc
set -o history
```

Команда установки контроллера домена с модулем синхронизации:

```
set +o history
sudo apt update && sudo apt install aldpro-syncer
sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'AstraLinux_176' --
↪ip 10.0.1.11 --update --setup_syncer
```

(продолжение на следующей странице)

```
set -o history
```

Команда установки контроллера домена с глобальным каталогом:

```
set +o history
sudo apt update && sudo apt install aldpro-gc
sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'AstraLinux_176' --
↪ip 10.0.1.11 --update --setup_gc
set -o history
```

3.2. Обновление подсистем ALD Pro через портал управления

В системе с версии 2.2.0 внедрен функционал централизованного обновления **aldpro-client** на компьютерах домена через **Портал Управления**. Функционал реализован в виде Политик обновления ALD Pro и функционирует аналогично **Групповым Политикам** по pull-модели. **Standalone-миньон** - автономный клиент на компьютерах домена для запуска заданий и локальных задач без подключений к мастеру - отвечает за формирование *source.list* и обновление **aldpro-client** при появлении новых версий **ALD Pro** в репозиториях, повышая гибкость и автономность процесса обновления на компьютерах клиентов.

Управление политикой обновления осуществляется через интерфейс **Портала Управления** в разделе **Установка и обновление ПО** – **Политики обновления ALD Pro**:

Для создания новой политики обновления ALD Pro необходимо нажать на кнопку [**+ Новая политика обновления**]. Будет выполнен переход на карточку новой политики.

На карточке на вкладке **Основное** задать имя политики обновления ALD Pro. Также можно указать описание политики. Указать статус политики: [**Включена**] и нажать кнопку [**Сохранить**] (рис. 3.2).

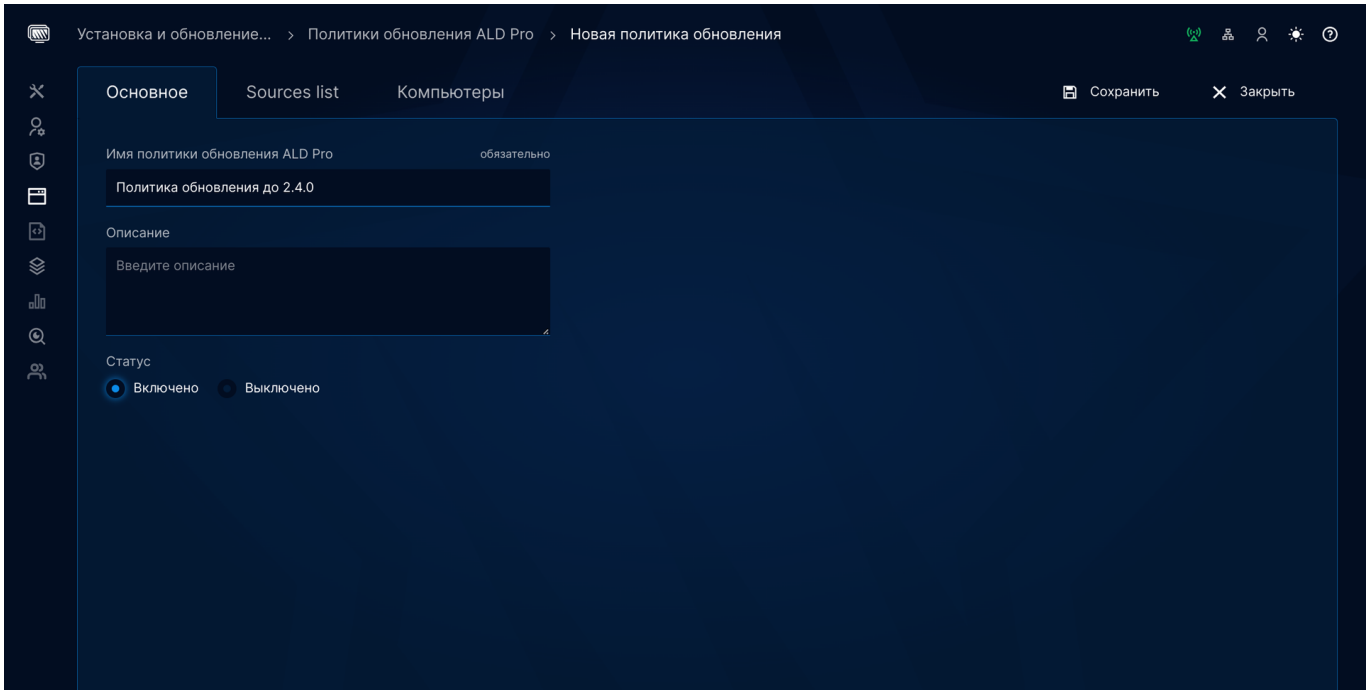


Рисунок 3.2 – Создание Групповой Политики обновления подсистем. Вкладка «Основное»

На вкладке **Source list** нажать кнопку [**Редактировать**] и ввести:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0/ 1.7_x86-64 main base
```

Затем оставить внизу комментарий (обязательно) и нажать кнопку [**Сохранить**].

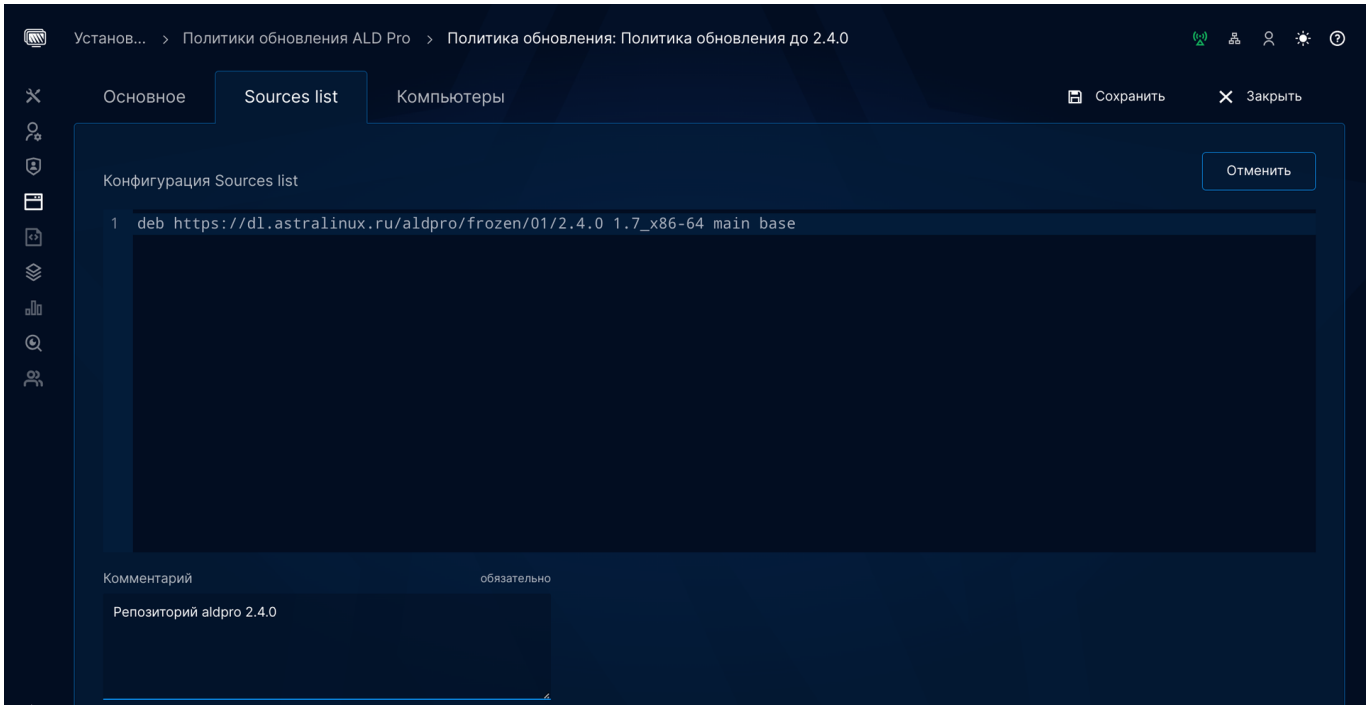


Рисунок 3.3 – Создание Групповой Политики обновления подсистем. Вкладка «Source list»

На вкладке **Компьютеры** указать все компьютеры или группы компьютеров, требующих обновления:

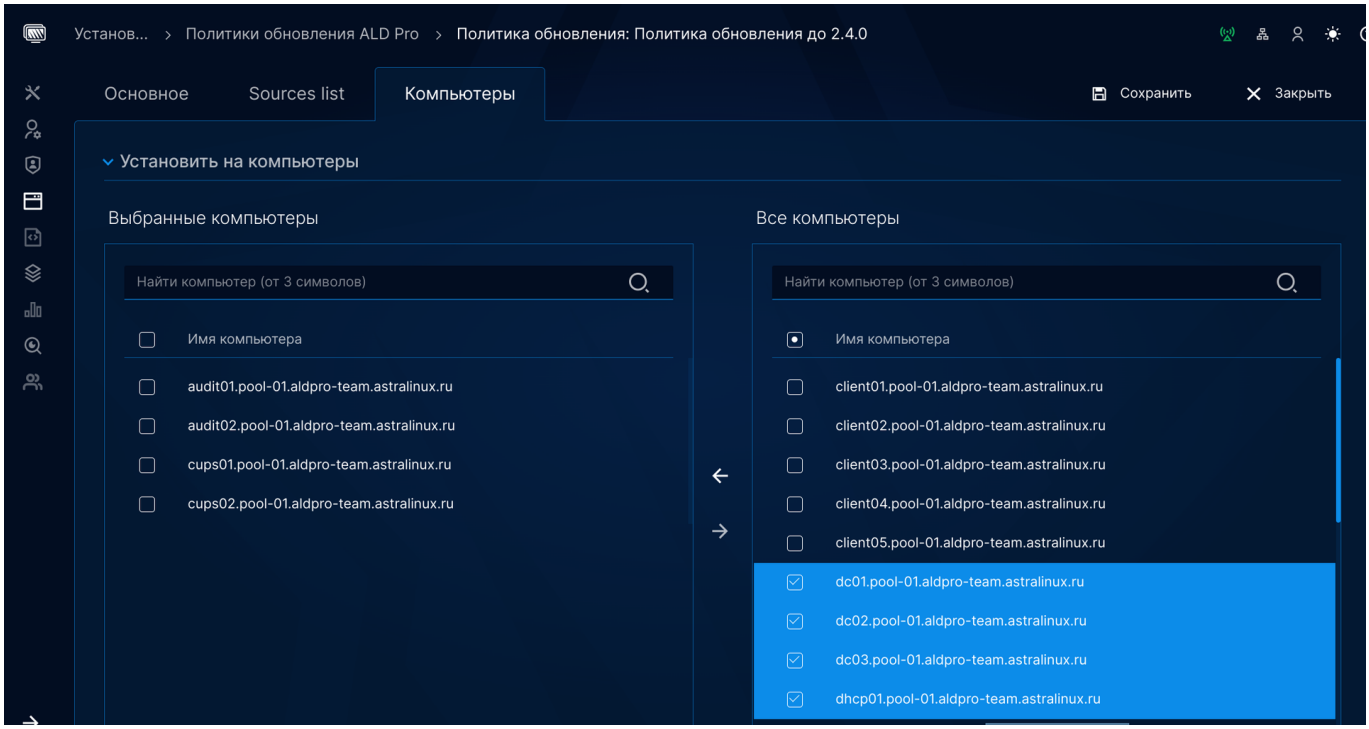


Рисунок 3.4 – Создание Групповой Политики обновления подсистем. Вкладка «Компьютеры»

Политка обновления выполняется раз в сутки по таймеру.

Начиная с версии 2.3.0 после создания ГП и перед обновлением клиента, необходимо **на компьютере подсистемы Общего Доступа к Файлам** выполнить следующую команду:

```
kdestroy -A
```

Для проверки успешного применения политик обновления необходимо выполнить следующие команды (при условии, что в */etc/apt/sources.list* указан репозиторий с версией выше, чем установлена на компьютере):

```
# Проверка версии aldpro-client
apt-cache policy aldpro-client
# Проверка версий зависимостей
apt-cache policy astra-freeipa-client salt-minion zabbix-agent python3 aldpro-
 ↪client-rdm syslog-ng initramfs-tools aldpro-common python3-distro python3-
 ↪yaml aldpro-policy-manager
```

Форсированное применение политики:

```
sudo salt-call aldpro_update_policy.build_and_run_updatepolicy -c /srv/salt/
↪standalone/config
```

`Verbose=True` указывает на необходимость вывода подробной информации, а `force=True` обозначает, что нужно очистить кэши перед началом работы команды. По умолчанию оба параметра имеют значение `False`.

Если политика была успешно применена, в выводе будут строки, подобные следующим:

```
aldpro-client:
    new:
        2.4.0
    old:
        2.3.0-8
```

Для принудительного обновления пакетов ALD Pro в терминале клиентов:

```
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confnew
```

Так же необходимо в версии ALD Pro 2.4.0 выполнить принудительно через терминалы клиентов скачивание и установку последней версии **Policy Manager**:

```
sudo aldpro-gpupdate --pm
```

Для повышения производительности домена с помощью Групповой Политики можно отключить связь с **salt master-minion** в разделе **Групповые Политики** на вкладке **Параметры компьютеров** в папке **Система** (рис. 3.5):

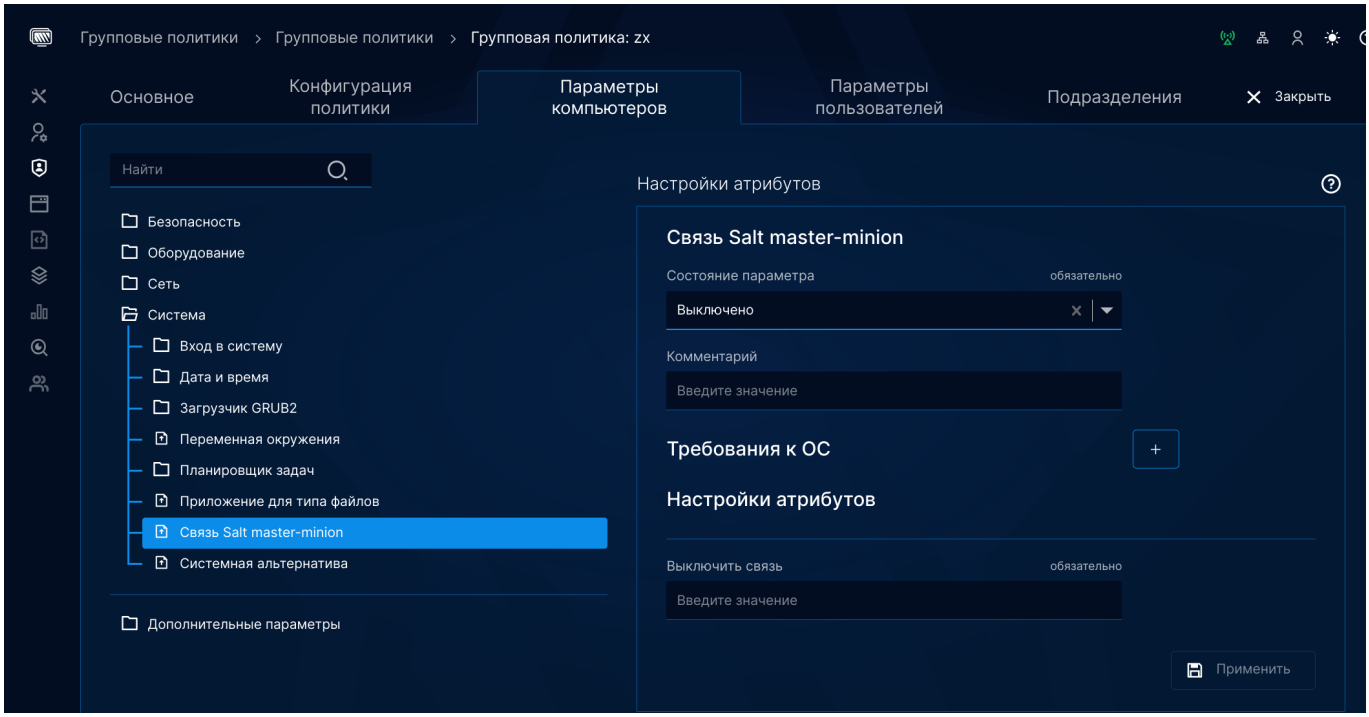


Рисунок 3.5 – Перевести состояние параметра в режим «Выключено»

Принудительное применение ГП:

```
sudo aldpro-gpupdate --gp
```

В связи с тем, что в рамках 2.4.0 был реализован функционал централизованного обновления **Policy Manager** на клиентских компьютерах, для применения ГП необходимо дождаться загрузки актуального **Policy Manager** на клиент, что занимает максимум 80 минут. Ожидание применения ГП на клиенте также составляет максимум 80 минут. Таким образом, максимальное время применения ГП - 160 минут.

Обновление подсистем запускается в карточке подсистемы из списка, находящегося в **Управление доменом** - **Общая информация** - **Состав системы** (рис. 3.5):

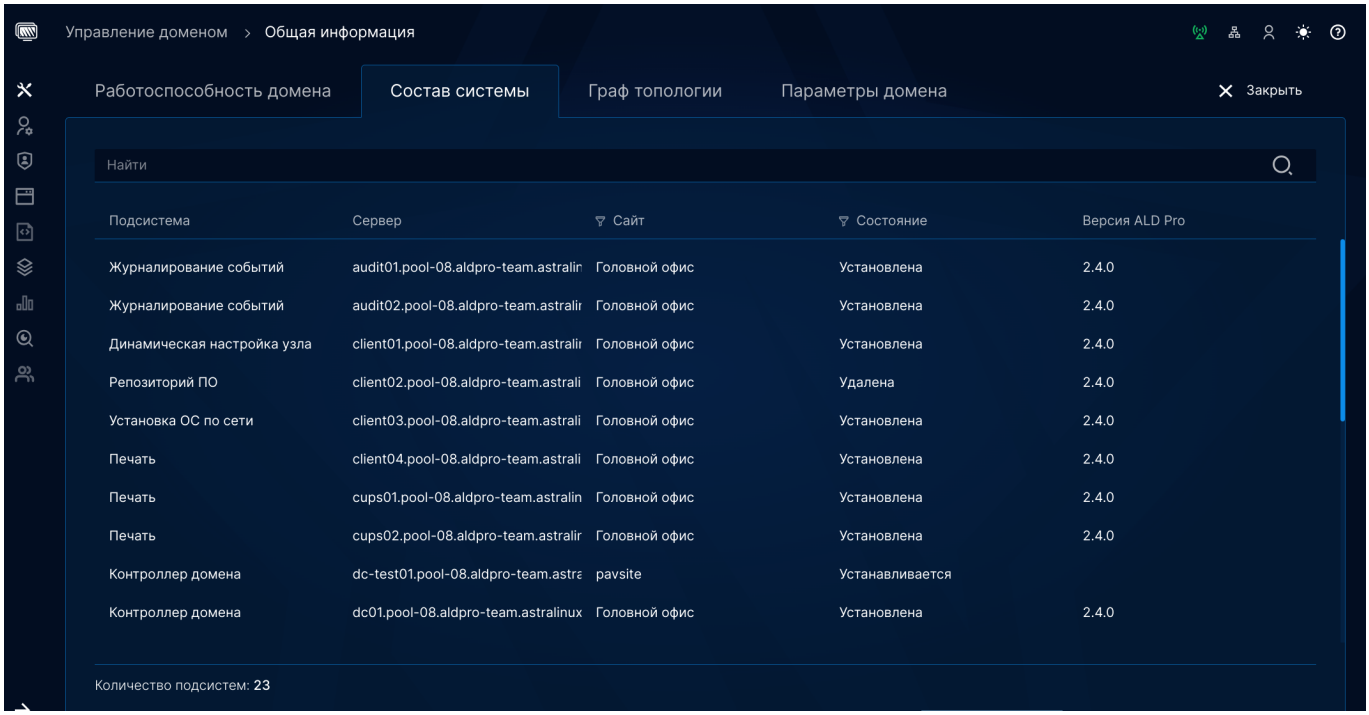


Рисунок 3.6 – Список подсистем

Обновление запускается кнопкой [**Обновить ALD Pro**] (рис. 3.6):

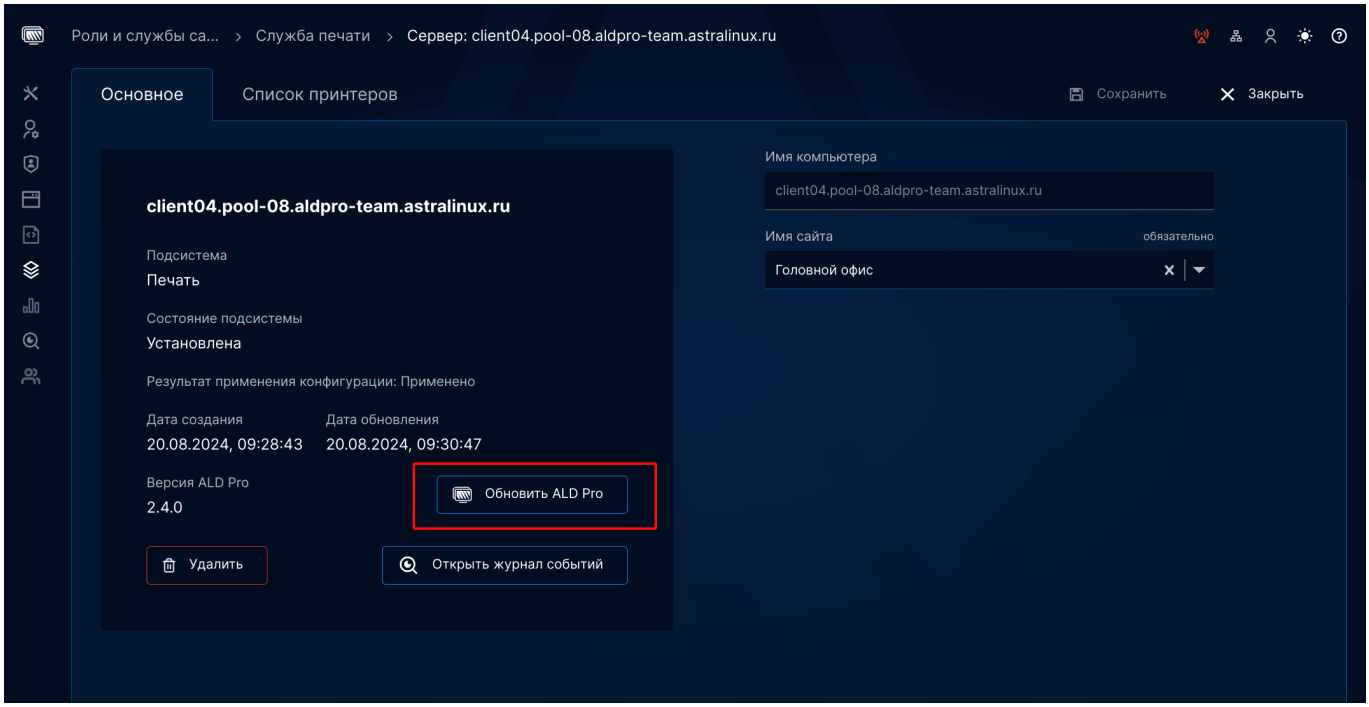


Рисунок 3.7 – Карточка подсистемы

На клиенте выполнить форсированное применение обновления:

```
aldpro-roles --iud
```

В случае успешного обновления ALD Pro на компьютере подсистемы карточка подсистемы будет выглядеть так (рис. 3.7):

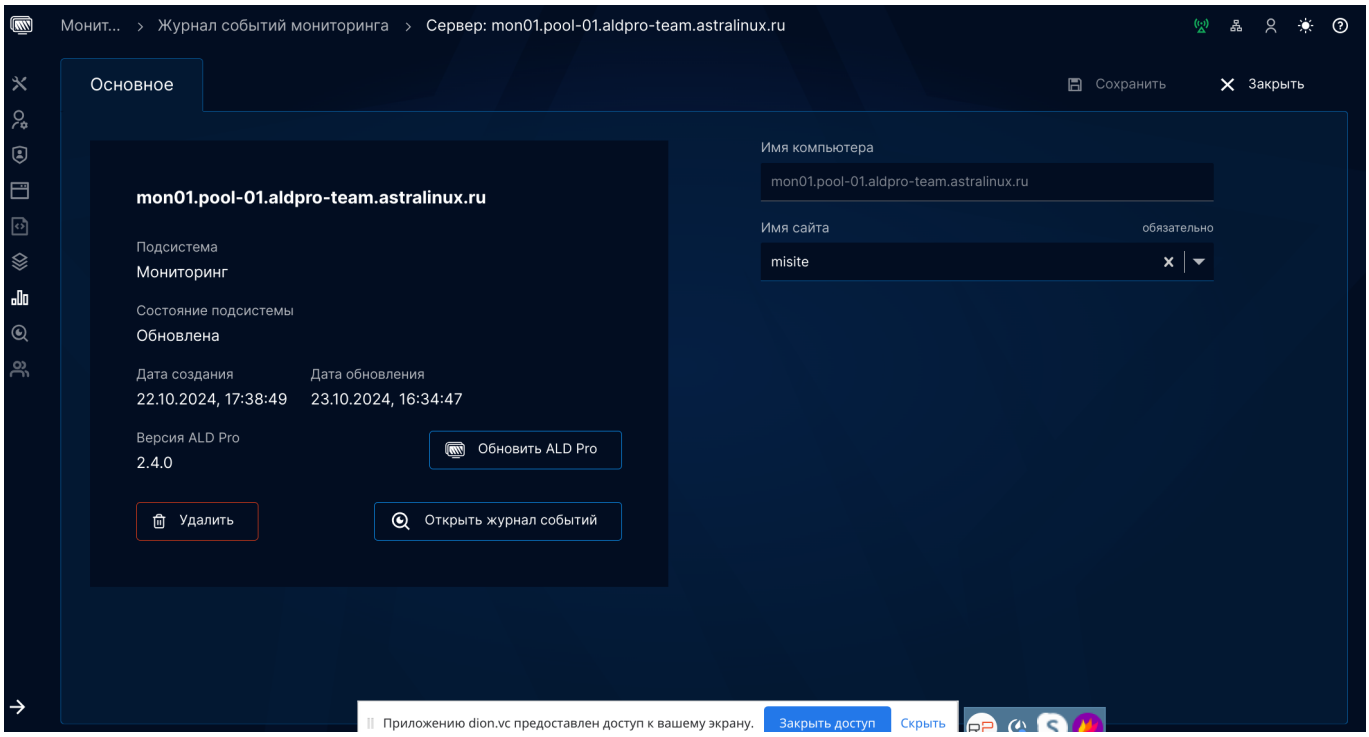


Рисунок 3.8 – Карточка подсистемы после успешного обновления

Если в процессе обновления что-то пошло не так, в карточке подсистемы будет отображаться ошибка (рис. 3.8):

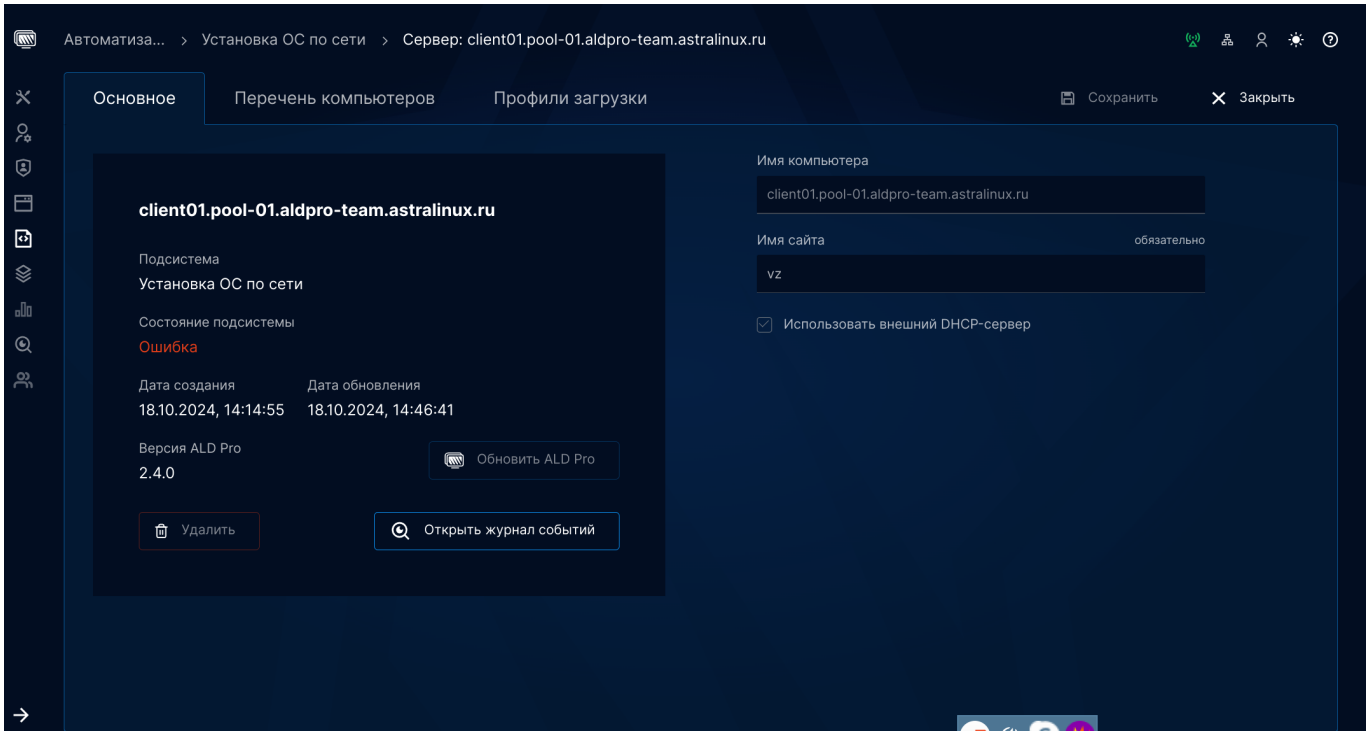


Рисунок 3.9 – Ошибка в карточке подсистемы

3.3. Обновление службы мониторинга Zabbix

Примечание: Подсистема мониторинга домена, в версиях ALD Pro предшествующих 2.4.0, не установится на ALSE 1.7.6. После обновления ОС до версии 1.7.6 необходимо перезапустить **Zabbix** командой `sudo systemctl restart zabbix-server`.

Для обновления **Zabbix** с 5.0.7 до 6.0.7 следует создать резервные копии конфигурационных файлов **Zabbix**, последовательно выполнив команды на сервере мониторинга:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
cp /etc/apache2/conf-enabled/zabbix-frontend-php.conf /opt/zabbix-backup/
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

Установить обновленные пакеты **Zabbix**:

```
sudo apt install --only-upgrade zabbix-server-pgsql zabbix-frontend-php
 ↪zabbix-agent
```

Для обновления базы данных **Zabbix** следует остановить службу севера командой:

```
sudo systemctl stop zabbix-server
```

В конфигурационный файл `/etc/zabbix/zabbix_server.conf` добавить параметр

```
AllowUnsupportedDBVersions=1;
```

Если требуется сохранить текущие таблицы с историей, то переименовать их, выполнив команду:

```
zcat /usr/share/zabbix-server-pgsql/history_pk_prepare.sql.gz | psql -h
 ↪localhost zabbix
```

Проверить в `/etc/zabbix/zabbix_server.conf` логин и пароль для подключения к базе данных **Zabbix** (параметры **DBUser** и **DBPassword**);

Запустить службу сервера **Zabbix** командой:

```
sudo systemctl start zabbix-server
```

Будет выполнено обновление базы данных **Zabbix**. Проверить статус обновления возможно командой:

```
sudo cat /var/log/zabbix-server/zabbix_server.log | grep database
```

Пример вывода команды:

```
82263:20230627:235018.484 current database version
(mandatory/optional):
05000000/05000002
82263:20230627:235018.484 starting automatic database upgrade
82263:20230627:235018.487 completed 0% of database upgrade
...
82263:20230627:235023.532 completed 97% of database upgrade
82263:20230627:235023.535 completed 98% of database upgrade
82263:20230627:235023.537 completed 99% of database upgrade
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
82263:20230627:235023.542 completed 100% of database upgrade
82263:20230627:235023.542 database upgrade fully completed
```

Для восстановления таблиц с историей мониторинга сначала следует остановить службу севера **Zabbix**:

```
sudo systemctl stop zabbix-server
```

Подключиться к серверу **PostgreSQL**:

```
psql -U zabbix -h localhost
```

Перезаписать историю из сохраненных таблиц в новые таблицы, выполнив запросы:

```
INSERT INTO history SELECT * FROM history_old ON CONFLICT
(itemid,clock,ns) DO NOTHING;
INSERT INTO history_uint SELECT * FROM history_uint_old ON CONFLICT
(itemid,clock,ns) DO NOTHING;
INSERT INTO history_str SELECT * FROM history_str_old ON CONFLICT
(itemid,clock,ns) DO NOTHING;
INSERT INTO history_log SELECT * FROM history_log_old ON CONFLICT
(itemid,clock,ns) DO NOTHING;
INSERT INTO history_text SELECT * FROM history_text_old ON CONFLICT
(itemid,clock,ns) DO NOTHING;
```

Удалить переименованные таблицы, выполнив запросы:

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

Запустить службу сервера **Zabbix**:

```
sudo systemctl start zabbix-server
```

3.4. Централизованное обновление менеджера политик программного обеспечения

Начиная с версии **ALD Pro** 2.4.0 внедрена возможность централизованного автообновления менеджера политик (**Policy Manager**) на клиентских компьютерах без участия администратора. Таким образом, процесс управления обновлениями и поддержание актуальности **ALD Pro** стали гораздо проще.

Менеджер политик отвечает за работу и применение групповых политик, политик программного обеспечения, правил сбора событий, развертывание подсистем, установку репликации контроллеров домена, заданий автоматизации.

Для корректной работы рекомендуется обновить все клиентские компьютеры до 2.4.0. При обновлении **ALD Pro** до 2.4.0 на компьютерах появится задание, которое централизованно обновляет менеджер политик.

В дальнейшем при обновлении **ALD Pro** на контроллере домена до последней версии, менеджер политик будет самостоятельно централизовано обновляться на клиентах, если в новой версии **ALD Pro** в него были внесены изменения.

Данные изменения гарантируют использование последней актуальной версии **ALD Pro** на доменных компьютерах без участия администраторов.

Актуальная версия менеджера политик обновляется на первом контроллере домена вместе с обновлением **ALD Pro**.

На остальных контроллерах домена информация об актуальной версии и содержание **Policy Manager** обновляется при репликации.

На всех клиентских компьютерах домена выполняется задание, которое гарантировано проверяет наличие обновления каждые 80 минут. Пакет менеджера политик обновляется при наличии изменений.

Если на отдельной клиентской машине нужно обновить **Policy Manager** вне расписания, необходимо использовать команду:

```
sudo aldpro-gpupdate --pm
```

или

```
sudo aldpro-gpupdate --policy-manager
```

Известные проблемы

4.1. Исправление односторонних отношений Left-Right

Перед созданием соглашения о репликации в LDAP-каталоге сначала создается такая сущность, как сегмент. Как только **Topology** плагин контроллера видит относящийся к нему сегмент, для которого не создано соглашение, он создает такое соглашение. Сначала создается соглашение в одном направлении, затем в обратном, и при наличии двух взаимонаправленных сегментов они удаляются и создается третий сегмент, у которого направление **Both**.

Если в домене наблюдается однонаправленное соглашение о репликации, т.е. стрелочка только в одну сторону, и не становится двунаправленной, это означает, что второй сегмент не был создан и для устранения проблемы его нужно создать вручную.

Найти однонаправленные **Left-Right** сегменты и записать их в файл.

```
set +o history
ldapsearch -Q -xLLL -D "cn=directory manager" -w "AstraLinux_176" -b
↪"cn=domain,cn=topology,cn=ipa,cn=etc,dc=ald,dc=company,dc=lan"
↪"(ipaReplTopoSegmentDirection=left-right)" > direction.ldif
set -o history
```

Создать **python** скрипт *parcing_topology.py* командой, а также этот файл можно скачать в личном кабинете:

```
nano parcing_topology.py
```

Сохранить файл с таким содержимым:

```
with open('direction.ldif') as f:
    lines = f.readlines()
ldif_change=""
text_block=""
for line in lines:
    text_block=f"{text_block}{line}"
    if "ipaReplTopoSegmentLeftNode" in line:
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
        left_node=line.split(":")[1].strip()
    if "ipaReplTopoSegmentRightNode" in line:
        right_node=line.split(":")[1].strip()
    if not line.strip():
        text_block=text_block.replace(f"{left_node}-to-{right_node}",f"{right_
↪node}-to-{left_node}")
        text_block=text_block.replace(f"ipaReplTopoSegmentLeftNode: {left_
↪node}",f"ipaReplTopoSegmentLeftNode: {right_node}")
        text_block=text_block.replace(f"ipaReplTopoSegmentRightNode: {right_
↪node}",f"ipaReplTopoSegmentRightNode: {left_node}")
        ldif_change+=f"{text_block}"
        text_block=""
with open('add_direction.ldif','w') as f:
    f.write(ldif_change)
```

Запустить следующий скрипт в той же папке и убедиться, что скрипт отработал корректно.

```
python3 parcing_topology.py
```

Теперь внести изменения из LDIF-файла в каталог от пользователя **admin**:

```
ldapadd -f add_direction.ldif
```

Проверить, что в системе не осталось **Left-Right** сегментов.

```
set +o history
ldapsearch -xLLL -D "cn=Directory Manager" -w "AstraLinux_176" -b "cn=domain,
↪cn=topology,cn=ipa,cn=etc,dc=ald,dc=company,dc=lan"
↪"(ipaReplTopoSegmentDirection=left-right)"
set -o history
```

Перезагрузить сервис **ipa** или последовательно все серверы, где были проблемы с односторонними соглашениями, для того, чтобы плагин **Topology** создал недостающие соглашения.

```
sudo ipactl restart
```

Соглашение создается, но не всегда запускается репликация после рестарта реплик. Как решение можно использовать команду:

```
ipa-replica-manage force-sync --from
```

4.2. Вход в систему занимает слишком много времени

Если время выполнения команды `id $USER` занимает слишком много времени, то в первую очередь необходимо выполнить команду `id -G $USER`. Если команда отработает мгновенно, то требуется добавить следующие значения в */etc/sssd/sssd.conf* в секцию `[domain/ald.company.lan]`:

```
[domain/ald.company.lan]
...
ignore_group_members = true
subdomain_inherit = ignore_group_members
...
```

Для применения настроек нужно перезапустить службу **sssd**:

```
sudo systemctl restart sssd
```

Обычно наиболее трудоемкой операцией является загрузка групп, включая их участников. На начальном этапе важно знать, членом каких групп является пользователь (`id aduser@ad_domain`), а не какие участники входят в конкретные группы (`getent group adgroup@ad_domain`). При установке, для параметра `ignore_group_members`, значения **True** - все группы отображаются как пустые. Таким образом загружается только информация о самих объектах группы, а не об их членах, что обеспечивает значительное повышение производительности LDAP запросов. Важно обратить внимание, что идентификатор `aduser@ad_domain` все равно вернет все правильные группы.

Параметр `ignore_group_members` может быть указан в разделе домена в *sssd.conf* как явно, так и одним из значений параметра `subdomain_inherit`. В этом случае этот параметр применится и к родительскому домену ALD Pro, и к доверенным поддоменам MS AD.

4.3. Решение проблемы с запросами на получение билетов kerberos

При попытке получить билет kerberos с реплик, все запросы уходят на первый Контроллер Домена вместо того КД, с которого делается запрос. Данное поведение может привести к невозможности работы портала при выходе из строя первого контроллера домена.

Для решения проблемы необходимо удалить или сделать не работающую ссылку для *winbind_krb5_locator.so*

```
sudo ln -s /dev/null /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/winbind_
↪krb5_locator.so
```

Затем выполнить `ipactl restart`.

4.4. Команды getent passwd и getent group не отображают пользователей и групп

Функция **Enumeration** отключена. Для получения дополнительной информации обратитесь в службу технической поддержки.

4.5. Изменений на сервере довольно долго не видно на клиенте

Служба **SSSD** кэширует идентификационную информацию в течение некоторого времени. Можно принудительно обновить кэш при следующем поиске с помощью команды `sss_cache -E`, предварительно установив пакет с нужной утилитой:

```
sudo apt install sssd-tools
```

Важно обратить внимание, что при входе в систему обновленная информация всегда перечитывается с сервера.

4.6. Как включить аутентификацию LDAP через незащищенное соединение

Это запрещено по соображениям безопасности, служба **SSSD** требует использовать TLS или LDAPS для аутентификации по LDAP.

4.7. В журналах нет ни одного сообщения от pam_sss

Скорее всего, стек **PAM** настроен неправильно. Нужно убедиться, что используется аутентификация, которая поддерживает **PAM**, поскольку некоторые методы аутентификации, такие как открытые ключи **SSH**, обрабатываются непосредственно в **SSHD** и вообще не используют стек **PAM**.

4.8. Я могу переключиться на доменного пользователя командой su из-под root, но не под обычным пользователем, SSH тоже не работает

Если есть сложность с переключением на другого пользователя командой `su` из-под **root**, это означает обычно, что полностью обходится аутентификация **SSSD**, используя модуль `pam_rootok.so`. Вероятно, неверно настроена служба **SSSD**. Необходимо войти в систему как локальный пользователь и продолжить отладку под этим профилем.

4.9. Я получаю сообщение Access denied for user $user: 6 (Permission denied)

Аутентификация прошла успешно, но пользователю было отказано в доступе к клиентской машине. Можно временно отключить управление доступом, задав `access_provider=permit`, но важно вернуть значение по умолчанию после выяснения основной причины.

Если отключение контроля доступа не помогает, возможно, учетная запись заблокирована

на стороне сервера. Нужно проверить журналы домена **SSSD**, чтобы узнать больше.

4.10. Исправление ошибки Глобального Каталога

При выполнении команды `ipactl start` и `ipactl restart` может возникать ошибка:

```
ipactl start
Existing service file detected!
Assuming stale, cleaning and proceeding
Starting Directory Service
Starting krb5kdc Service
Starting kadmin Service
Starting named Service
Starting httpd Service
Starting ipa-custodia Service
Starting smb Service
Starting winbind Service
Starting ipa-otpd Service
Starting ipa-dnskeysyncd Service
Starting globalcatalog Service
Failed to start globalcatalog Service
Shutting down
```

Для исправления ошибки необходимо:

Запустить сервисы **ipa** на контроллерах домена:

```
ipactl start --ignore-service-failure
```

Обновить **Global Catalog** на контроллерах домена:

```
aldpro-gc-install
```

На все вопросы ответить «**yes**».

Перезапустить сервисы **ipa** на контроллерах домена:

```
ipactl restart
```

Отладка подсистем ALD Pro

Для эффективного управления доменом на предприятии системный администратор должен четко понимать, какие службы обеспечивают работу компьютера в домене и как выполнять отладку их работы, если возникнут какие бы то ни было внештатные ситуации. Настоящая инструкция ставит целью ответить на все указанные вопросы с максимальной степенью детализации.

5.1. Архитектура SSSD

За работу компьютера в домене отвечает служба **SSSD** (System Security Services Daemon), установка которой происходит вместе с пакетом **aldpro-client**, а настройка выполняется при вводе компьютера в домен с помощью утилиты **aldpro-client-installer**. Для управления службой используется приложение **systemctl**, которое позволяет запускать и останавливать службу, а также просматривать ее текущее состояние:

```
sudo systemctl stop sssd
sudo systemctl start sssd
sudo systemctl restart sssd
sudo systemctl status sssd
```

Результат выполнения команды:

```
sssd.service - System Security Services Daemon
  Loaded: loaded (/lib/systemd/system/sssd.service; enabled; vendor preset:
↪enabled)
  Active: active (running) since Mon 2023-07-31 22:38:35 MSK; 5s ago
Main PID: 3740 (sssd)
   Tasks: 4 (limit: 4597)
  Memory: 10.1M
  CGroup: /system.slice/sssd.service
          ├─3740 /usr/sbin/sssd -i --logger=files
          ├─3741 /usr/lib/x86_64-linux-gnu/sssd/sssd_be --domain ald.company.
↪lan --uid 0 --gid 0 --logger=files
          ├─3742 /usr/lib/x86_64-linux-gnu/sssd/sssd_ifp --uid 0 --gid 0 --
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
↪logger=files
         └─3743 /usr/lib/x86_64-linux-gnu/sssd/sssd_pac --uid 0 --gid 0 --
↪logger=files
```

Основная задача службы **SSSD** заключается в предоставлении доступа к информации удаленной службы каталога и механизмам централизованной аутентификации. В качестве поставщика данных для **SSSD** может выступать **FreeIPA**, **Active Directory**, **Kerberos** домен и даже простой LDAP каталог. Продукт ALD Pro построен на базе службы каталога **FreeIPA**, поэтому в конфигурационных файлах */etc/sssd/sssd.conf* на доменных компьютерах будет указан поставщик `ipa`, например, `id_provider = ipa`.

При входе доменного пользователя в систему у него появляется домашняя директория, но служба **SSSD** не создает его как локального пользователя. Для ускорения работы системы и возможности обслуживания пользователя в автономном режиме служба **SSSD** помещает информацию, полученную от поставщика данных, в локальный кэш.

Идентификационные данные пользователей всегда кэшируются, так же как и информация о доменных службах. Это означает, что **SSSD** всегда проверяет запросы на аутентификацию. Поддержание кэша в актуальном состоянии является сложной задачей, поэтому **SSSD** состоит из множества компонентов, которые взаимодействуют друг с другом с помощью различных методов межпроцессного взаимодействия, архитектура службы представлена см. *Архитектура службы SSSD*.

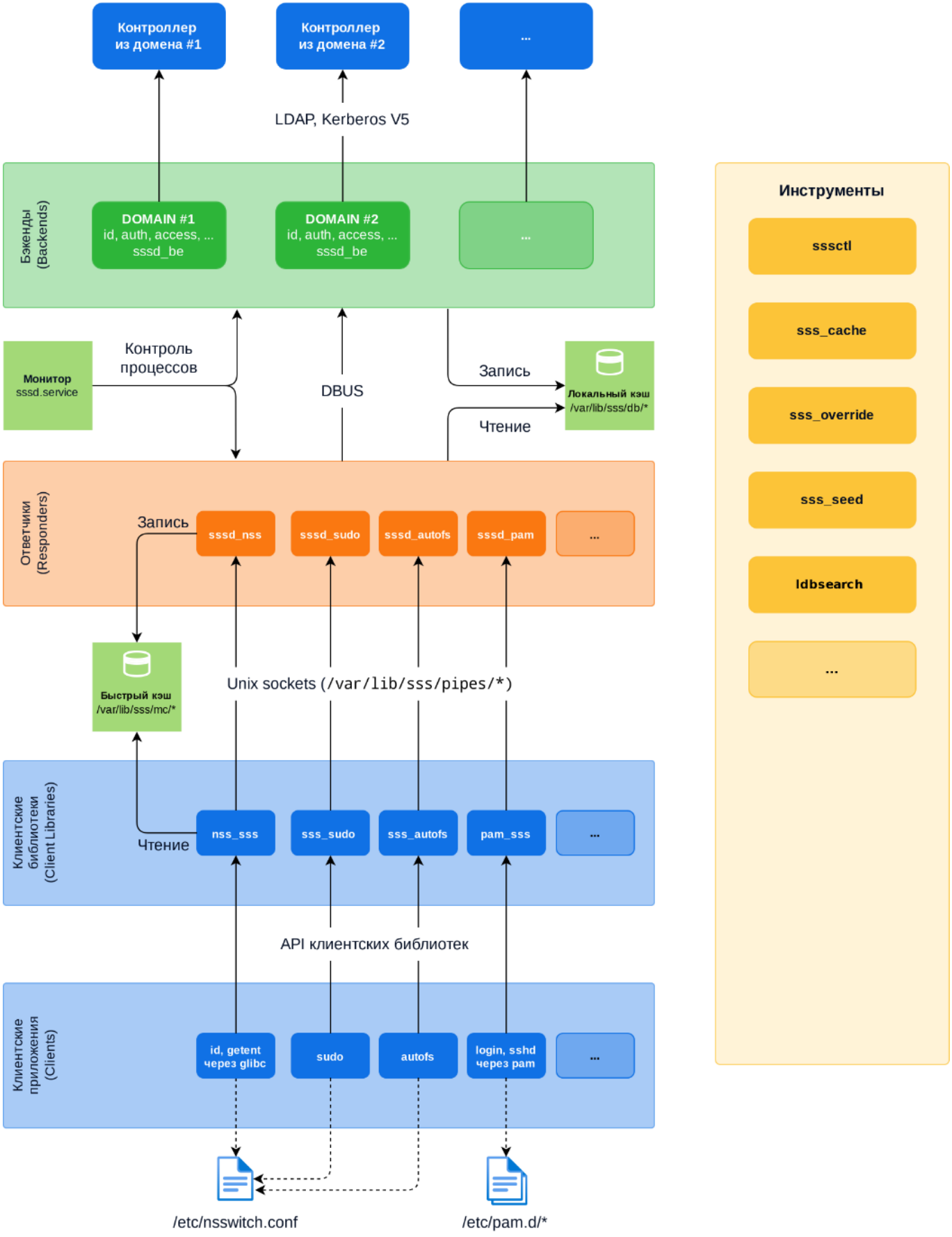


Рисунок 5.1 – Архитектура службы SSSD

5.1.1. Монитор (Monitor)

Службу **sssd.service** называют **Монитором** или **Наставником**, так как она порождает процессы всех других компонентов и управляет ими через систему межпроцессного взаимодействия **SBus** (**SSSD D-Bus**). Чтобы понять диаграмму процесса процесса загрузки службы см. *Диаграмма загрузки службы SSSD*.

- В момент запуска службы **Монитор** анализирует файл конфигурации **sssd.conf** (1) и загружает его в кэш */var/lib/sss/db/config.ldb* (2). Далее **Монитор**, используя системные вызовы, порождает процесс **Бэкенда** (3), который загружает информацию из **config** (4, 5) и регистрирует себя в **Мониторе** (6).
- На следующем шаге **Монитор** порождает **Ответчиков NSS** и **PAM** (7), которые считывают информацию из **config** (8, 9) и регистрируются в **Мониторе** (10) и **Бэкенде** (11).

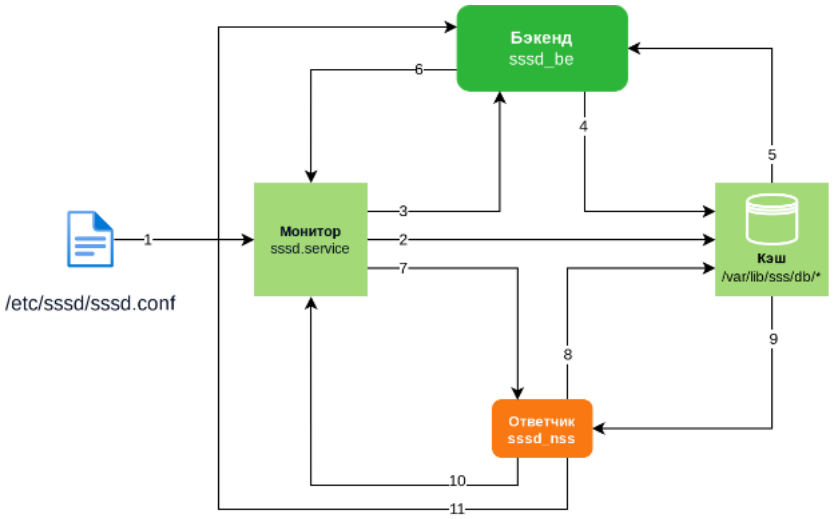


Рисунок 5.2 – Диаграмма загрузки службы SSSD

Помимо внутреннего взаимодействия через **SBus**, служба **SSSD** использует UNIX-сигналы, например, в тех случаях, когда процесс завис и не отвечает через **SBus**. Обработчики сигналов обычно интегрированы с главным циклом событий **tevent** с использованием вызова `tevent_add_signal`:

- ``SIGTERM — сигнал на завершение работы процесса.
- `SIGKILL` — сигнал для принудительного завершения работы процесса, который направляется монитором, если процесс не завершается после получения **SIGTERM**.
- `SIGUSR1` — сигнал на переключение **Бэкенда** в автономный режим. Если сигнал будет направлен Монитору, то он перешлет его всем дочерним процессам **sssd_be**.
- `SIGUSR2` — сигнал на переключение **Бэкенда** в режим онлайн. Если сигнал будет

направлен Монитору, то он перешлет его всем дочерним процессам **sssd_be**.

- `SIGHUP` — может быть передан **Монитору**, после чего тот начнет новый лог-файл и вызовет метод сброса журналов управляемым процессам. Управляемые процессы также начинают новые лог-файлы. Кроме того, процессы **sssd_be** перечитывают *resolv.conf*, а процесс **sssd_nss** выполнит очистку быстрого кэша.

5.1.2. Серверные части или Бэкенды (backends)

Бэкенд (Backend, BE) включает в себя **Поставщики данных** (Data Provider, DP) – это процесс приложения */usr/lib/x86_64-linux-gnu/sssd/sssd_be*, который запускается **Монитором** для взаимодействия с доменом в соответствии с настройками секции [domain/<имя_домена>] конфигурационного файла *sssd.conf*.

```
sudo ps -aux | grep sssd_be
```

Результат выполнения:

```
root    5885 0.0 0.6 118804 26532 ?  S 21:43 0:00 /usr/lib/x86_64-linux-gnu/
↪sssd/sssd_be --domain ald.company.lan --uid 0 --gid 0 --logger=files
```

Описание **бэкендов** в конфигурационном файле можно просмотреть командой:

```
sudo cat /etc/sssd/sssd.conf
```

Результат выполнения:

```
...
[domain/ald.company.lan]
id_provider = ipa
ipa_server = _srv_, dc-1.ald.company.lan
ipa_domain = ald.company.lan
ipa_hostname = pc-1.ald.company.lan
auth_provider = ipa
chpass_provider = ipa
access_provider = ipa
cache_credentials = True
ldap_tls_cacert = /etc/ipa/ca.crt
krb5_store_password_if_offline = True
...
```

Бэкенд не является монолитом и в зависимости от настроек подключает один и более **Поставщиков** данных, каждый из которых представляет из себя совместно используемую библиотеку (Shared Object, SO). Библиотеки **SSSD** находятся в каталоге */usr/lib/x86_64-linux-gnu/sssd/*, и за работу компьютера в домене ALD Pro (FreeIPA), например, отвечает **Поставщик ipa**, которому соответствует библиотека *libsss_ipa.so*. Если же ввести Linux компьютер напрямую в домен Active Directory, то будет использоваться *libsss_ad.so*.

Каждый **Поставщик** данных может оказывать **Бэкенду SSSD** следующие услуги:

- идентификация (`id_provider`) — предоставление информации о пользователях, служебных учетных записях, группах и т.д.;
- аутентификация (`auth_provider`) — проверка аутентичности пользователей и служб;
- авторизация (`access_provider`) — проверка прав доступа на запуск служб на уровне HBAC;
- поставщик sudo (`sudo_provider`) — предоставление информации о правилах SUDO на повышение привилегий;
- и другие.

Разные модули могут использовать разные настройки, например, для **Поставщика ipa** требуется задать параметр **ipa_server**, который используется для определения контроллера домена **FreeIPA**, с которым требуется поддерживать соединение. По умолчанию в конфигурационном файле *sssd.conf* для домена явно задается только несколько провайдеров (`id`, `auth`, `chpass`, `access`), а для других провайдеров подразумевается, что их имя совпадает с именем поставщика идентификационных данных (`id_provider`).

Для удобства, из соображений сокращения размера конфигурационного файла, служба **SSSD** руководствуется следующими правилами:

- Если параметр `ipa_domain` не указан, то используется FQDN имя домена из секции [`domain/<fqdn домена>`].
- Если параметр `ipa_server` не указан, то сервер определяется автоматически через DNS, как будто параметр `ipa_server` равен `_srv_`.
- Если провайдер не указан, то используется значения `id_provider` (например, **ipa**). Исключением является `access_provider`, если его значение не указано, то по умолчанию параметр принимает значение `permit`, что означает «всем пользователям разрешен доступ». Для управления доступом параметр

`access_provider` должен быть задан явно.

Таким образом, стандартный файл конфигурации */etc/sssd/sssd.conf* может быть упрощен до следующего вида:

```
...
[domain/ald.company.lan]
id_provider = ipa
access_provider = ipa
cache_credentials = True
ldap_tls_cacert = /etc/ipa/ca.crt
krb5_store_password_if_offline = True
...
```

5.1.3. Ответчики (responders)

Ответчик служит посредником между пользовательским приложением, таким как **id** или **getent**, и кэшем **SSSD**. Когда приложение запрашивает данные, например, ищет пользователя по имени, **Ответчик** выполняет поиск в локальном кэше и, если данные не найдены или устарели, обращается к **Бэкенду**, чтобы тот совершил запрос к северу службы каталога. Клиентские библиотеки взаимодействуют с **Ответчиком** через **Unix socket** (например, */var/lib/sss/pipes/nss*), а внутри ядра **SSSD** между **Ответчиками**, **Бэкендами** и **Монитором** взаимодействие осуществляется через **DBus**.

Каждый **Ответчик** работает в отдельном процессе и выполняет строго определенные задачи, например:

- Ответчик **NSS** (`sssd_nss`) обеспечивает данными диспетчер службы имен (Name Service Switch, NSS).
- Ответчик **PAM** (`sssd_pam`) предоставляет данные для работы подключаемых модулей аутентификации (Pluggable Authentication Modules, PAM).
- Ответчик **SUDO** (`sssd_sudo`) обеспечивает правилами утилиту `sudo`.
- И так далее.

5.1.4. Клиентские библиотеки и приложения

Для взаимодействия с **Ответчиком** приложению нужно обращаться к соответствующему **unix сокету** по специфичному сетевому протоколу, поэтому для упрощения разработки клиентских приложений были созданы вспомогательные совместно используемые библиотеки, такие как *libnss_sss.so.2*, *pam_sss.so* и др., которые можно найти в каталоге */usr/lib/x86_64-linux-gnu*.

Например, если в конфигурационном файле *nsswitch.conf* для базы **sudoers** источником данных будет указан **sss**, то утилита **sudo** для взаимодействия с **Ответчиком sssd_sudo** воспользуется функциями из библиотеки *libsss_sudo.so*. Аналогичным образом работает **autofs**.

Однако, есть и такие приложения, которые могут даже не знать о существовании **SSSD**, если используют более универсальные функции из библиотек **pam** и **glibc**, которые взаимодействуют с клиентскими библиотеками **SSSD** от имени приложения. Например, утилиты **id**, **getent**, **ls** вызывают стандартные функции **NSS**, которые извлекают информацию из нескольких баз данных имен (например, `passwd`, `group`, `service`, `netgroup` и т.д.). Рассмотрим поток данных, создаваемый клиентским приложением **id**, выполняющим **NSS** запрос через службу **SSSD**, см. *Диаграмма потока данных при вызове команды id*:

- Клиентское приложение */usr/bin/id* вызывает функцию `getpwnam` из библиотеки **glibc** для получения информации о пользователе (1).
- Библиотека `glibc` загружает конфигурационный файл *nsswitch.conf* (2, 3) и в соответствии с его настройками загружает совместно используемую библиотеку *libnss_sss.so.2* и вызывает ее функцию `_nss_sss_getpwnam_r` (4)
- Для получения данных от **Ответчика** */usr/lib/x86_64-linux-gnu/sssd/sssd_nss* библиотека *libnss_sss.so.2* отправляет запрос через **unix сокет** */var/lib/sss/pipes/nss* по специализированному протоколу обмена данными (5).
- Ответчик **NSS** проверяет наличие данных в кэше (6, 7), и в случае отсутствия данных или истекшего срока их действия отправляет запрос `getAccountInfo` к **Бэкенду**, запрашивая необходимые данные (8).
- **Бэкенд** из-под учетной записи хоста */etc/krb5.ketyab* направляет LDAP-запрос для получения данных от службы каталога (9). Если запрашиваемый пользователь принадлежит тому же домену, то это будет прямой запрос на поиск информации, в противном случае **SSSD** выполнит расширенную LDAP операцию.

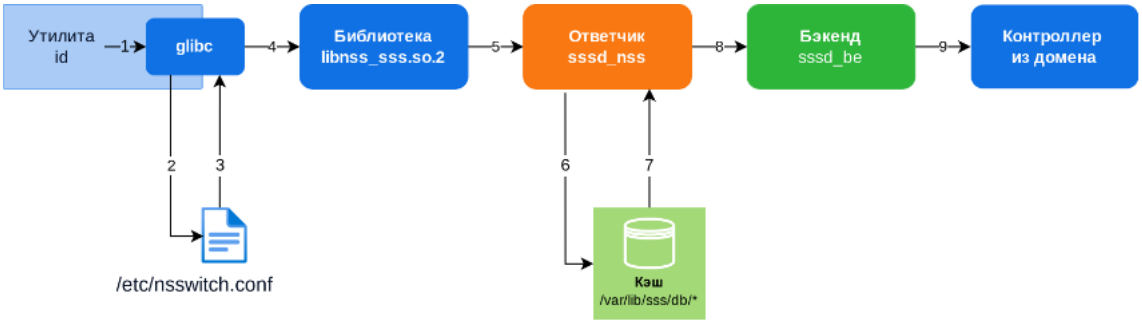


Рисунок 5.3 – Диаграмма потока данных при вызове команды id

Библиотека **PAM** работает аналогичным образом с небольшими отличиями. Например, если служба **SSSD** находится онлайн, то при входе пользователя в систему она обязательно предпримет попытку повторной аутентификации и получения информации об участии пользователя в группах, даже если срок действия кэша еще не истек, а все команды выполняются в контексте одной транзакции **PAM**. В качестве примеров клиентских приложений, использующих **PAM**, можно привести **login**, **su** и **sshd**.

Не смотря на то, что клиентские приложения и библиотеки не являются синонимами, в рабочей документации и программном коде службы **SSSD** под словосочетанием **SSSD Client** (суффикс **sss_cli**) часто подразумеваются именно клиентские библиотеки, а не сами приложения. Например, идентификатором **sss_cli** обозначается сетевой протокол, с помощью которого клиентские библиотеки общаются с **Ответчиками** службы **SSSD**.

5.1.5. Механизм кэширования службы SSSD

Служба **SSSD** в своей работе активно использует кэш, что позволяет исключить повторные обращения к серверу за получением одних и тех же данных, когда в этом нет необходимости, и обеспечивает возможность работы в автономном режиме без подключения к серверу. Система кэширования довольно сложна и включает в себя несколько взаимодополняющих механизмов.

5.1.5.1. Локальный кэш (local cache, cache)

Локальный кэш – это база данных, в которую **Бэкенды** записывают результаты, получаемые из службы каталога, чтобы **Ответчики** могли считывать их напрямую. При помещении объектов в кэш им назначается срок жизни, и до тех пор, пока он не истечет, все последующие запросы к этим данным обрабатываются без обращения к серверу.

Устаревшие данные могут извлекаться из кэша только в том случае, если служба находится в автономном режиме.

Файлы локального кэша находятся на диске в каталоге */var/lib/sss/db/*, для работы с этим кэшем служба **sssd** использует встроенную библиотеку баз данных облегченного доступа (light weight embedded database library, **ldb**) — ту же библиотеку, с помощью которой реализована база данных службы каталога **Samba AD**. Если установить пакет **ldb-tools**, то станет доступна утилита **ldbsearch**, с помощью которой можно заглянуть внутрь **ldb**-файлов. Структура этой базы данных подобна LDAP-каталогу, а параметры `ldbsearch` во многом повторяют параметры утилиты **ldapseach**:

```
sudo rm -rf /var/lib/sss/db/*; systemctl restart sssd ldbsearch -H /var/lib/
↪sss/db/cache_ald.company.lan.ldb /
-b name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb uidNumber
asq: Unable to register control with rootdse!
# returned 0 records
# 0 entries
# 0 referrals
```

Для проверки кэша запрашивается информация командой `id`:

```
sudo id admin
```

Результат выполнения:

```
uid=959800000(admin) gid=959800000(admins) группы=959800000(admins),
↪1001(astra-admin),113(lpadmin)
```

Отображается, что сохранилось в локальном кэше командой:

```
sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb /
-b name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb uidNumber
```

Результат выполнения:

```
asq: Unable to register control with rootdse!
# record 1
dn: name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb
uidNumber: 959800000
# returned 1 records
# 1 entries
```

(продолжение на следующей странице)

```
# 0 referrals
```

Но на низком уровне **ldb** использует библиотеку **Trivial DataBase**, поэтому сырые данные в формате ключ-значение можно извлекать с помощью утилит из пакета **tdb-tools**:

```
root@dc-1:~# tdbtool /var/lib/sss/db/config.ldb keys
```

Результат выполнения:

```
key 18 bytes: DN=@INDEX:CN:SSSD
key 21 bytes: DN=CN=SSSD,CN=CONFIG
key 18 bytes: DN=@INDEX:CN:SUDO
key 17 bytes: DN=@INDEX:CN:PAC
key 13 bytes: DN=CN=CONFIG
...
```

В папке */var/lib/sss/db/* находятся следующие **ldb**-файлы локального кэша:

- *config.ldb* – кэш для хранения настроек службы, загружаемых из конфигурационного файла /etc/sssd/sssd.conf
- *sssd.ldb* – база данных локального домена (**local provider**), который позволял использовать возможности вложенных групп без централизованной службы каталога. Для администрирования использовались команды `sss_useradd`, `sss_groupadd` и др., поддержка функции прекращена с версии **SSSD** 2.0.0.
- *cache_ald.company.lan.ldb* – кэш для хранения критически важной информации, получаемой из домена. Сохранение таких данных на диск требуется выполнять сразу при получении новых данных.
- *timestamps_ald.company.lan.ldb* – кэш для хранения некритичной и часто обновляемой информации. Сохранение данных на диск выполняется в фоновом режиме по усмотрению операционной системы. Кроме файлов локального кэша в этой же папке находятся файлы кэша учетных данных Kerberos, которые можно просматривать с помощью утилиты **klist** с ключом `-c`.
- *ccache_ALD.COMPANY.LAN* – кэш для хранения Kerberos-билетов службы **SSSD**, с помощью которых она выполняет LDAP-запросы к службе каталога.
- *fast_ccache_ALD.COMPANY.LAN* – кэш для хранения TGT-билета, с помощью сессионного ключа которого которого обеспечивается дополнительное шифрование запросов по технологии **FAST Armoring**.

Удаление всех файлов из директории **db** является самым простым, но в тоже время и самым полным способом очистки локального кэша службы **sssd**:

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

То же самое можно сделать с помощью команды `cache-remove` утилиты **sssctl** из состава пакета **sssd-tools**:

```
sudo sssctl cache-remove
```

Результат выполнения:

```
SSSD must not be running. Stop SSSD now? (yes/no) [yes] yes
Creating backup of local data...
Removing cache files...
SSSD needs to be running. Start SSSD now? (yes/no) [yes] yes
```

Однако, для отладки **SSSD** на нагруженном сервере рекомендуют использовать специализированную утилиту **sss_cache** из того же пакета **sssd-tools**, с помощью которой можно удалять не весь кэш, а отметить недействительными только отдельные записи, например, по конкретному пользователю, чтобы они были повторно извлечены из каталога при следующем обращении:

```
sudo sss_cache -u admin
```

5.1.5.2. Быстрый кэш (in-memory cache, memcache)

Все процессы **SSSD** (**Монитор**, **Бэкенд**, **Ответчик**), являются однопоточными приложениями и обрабатывают запросы со скоростью одного ядра, поэтому **Ответчик** мог бы стать узким местом и ограничить возможности вертикального масштабирования. В тоже время, из соображений безопасности клиентским библиотекам нельзя давать прямой доступ к локальному кэшу, т.к. в нем содержатся в том числе конфиденциальные данные, например, хэши паролей. Для устранения противоречия был создан дополнительный быстрый кэш, который доступен клиентским библиотекам напрямую, но содержит при этом только общую информацию о пользователях и группах.

Быстрый кэш представляет из себя хэш-таблицы, которые находятся в каталоге

/var/lib/sss/mc/ и отображаются на память (Memory-mapped). Ответчик **NSS** записывает информацию в быстрый кэш и до тех пор, пока эти данные остаются актуальными, клиентской библиотеке не требуется связываться с **SSSD** для их извлечения. Если же запись будет отсутствовать в кэше или ее срок жизни истек, запрос будет перенаправлен **Ответчику NSS**, который извлечет данные из локального кэша или обратится к контроллеру домена через **Бэкенд**.

По умолчанию время хранения записей в быстром кэше составляет 300 секунд и может быть изменено с помощью параметра `memcache_timeout` в файле *sssd.conf*. Для отключения быстрого кэша нужно установить `memcache_timeout=0`, а для очистки, как и в случае локального кэша, просто удалить файлы и перезапустить службу:

```
sudo systemctl stop sssd
sudo rm -rf /var/lib/sss/mc/*
sudo systemctl start sssd
```

Команда `cache-remove` утилиты **sssctl** не удаляет файлы быстрого кэша, но `sss_cache`, как и в случае локального кэша, позволяет отметить недействительными отдельные записи, например, по конкретному пользователю, чтобы они были повторно извлечены из каталога при следующем обращении.

```
sudo sss_cache –u admin
```

5.1.5.3. Негативный или безрезультатный кэш (negative cache, ncache)

Для уменьшения нагрузки на сервер при обращении к несуществующим объектам каталога, внутри **Ответчика NSS** и ряда других компонентов реализован так называемый негативный или безрезультатный кэш. Этот кэш находится в оперативной памяти и обновляется каждые 15 секунд, изменить эту настройку можно с помощью параметра `entry_negative_timeout` в конфигурационном файле *sssd.conf* Для полной очистки негативного кэша достаточно просто перезапустить службу командой:

```
sudo systemctl restart sssd
```

5.1.5.4. Алгоритм использования кеша (cache lookup)

Служба **SSSD** кэширует пользователей, группы, правила HBAC/SUDO, SSH-ключи, карты монтирования дисков и другую информацию, но вне зависимости от типов объектов поиск

в кэше выполняется очень похожим образом, см. *Упрощенный алгоритм поиска*.

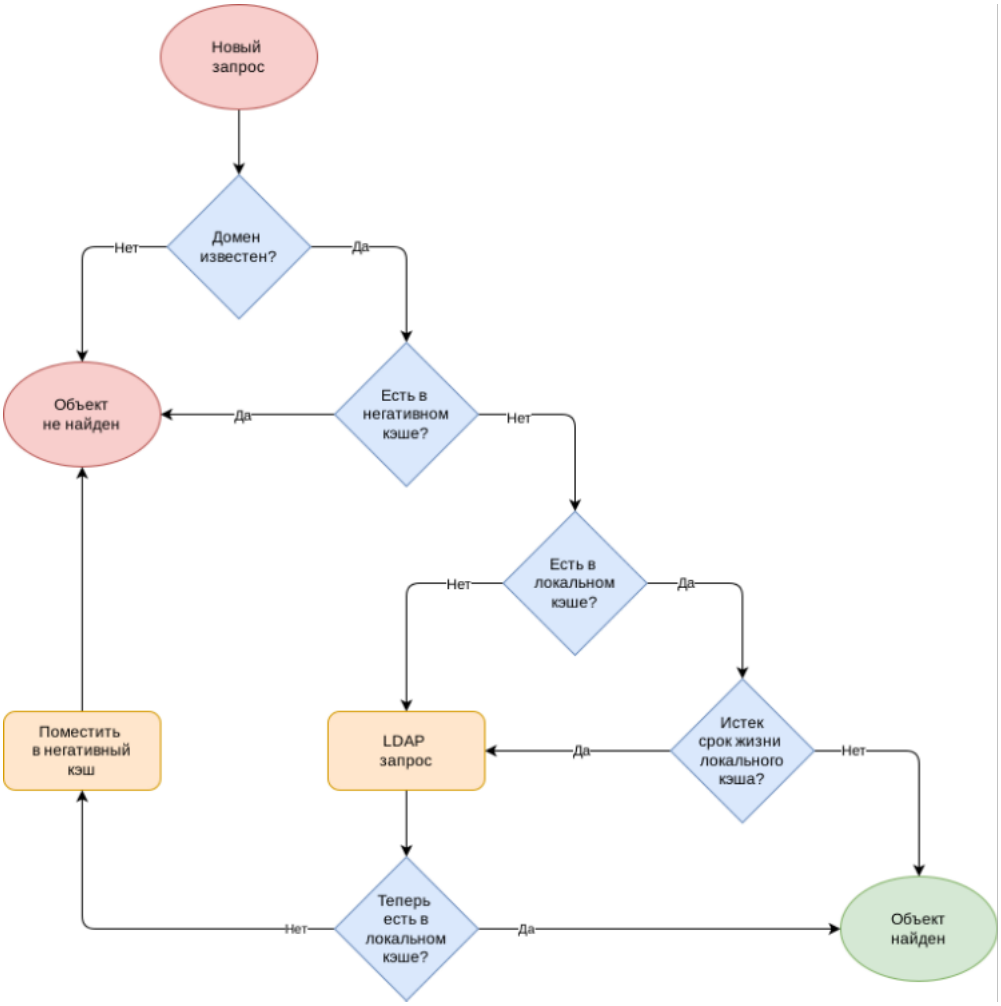


Рисунок 5.4 – Упрощенный алгоритм поиска

В действительности алгоритм немного сложнее и выполняется чуть больше проверок и действий, но приведенной схемы достаточно для получения общих представлений.

Рассмотрим поток данных при поиске информации с использованием кэша, см. *Поток данных при поиске информации о пользователе с использованием информации из кэша*:

- Клиентское приложение вызывает функцию **getpwnam** из библиотеки **glibc**, которая в соответствии настройками из *sswitch.conf* загружает Клиентскую библиотеку *libnss_sss.so.2* и вызывает ее функции, подробнее см. в разделе «Клиентские приложения и библиотеки».
- Клиентская библиотека проверяет, есть ли информация о запрашиваемом объекте в быстром кэше (**memcache**), и только после этого обращается к **Ответчику sssd_nss**.
- **Ответчик** проверяет, есть ли информация о запрашиваемом объекте в локальном кэше, и только после этого обращается к **Бэкенду sssd_be**.

- **Бэкенд** выполняет запрос к LDAP-каталогу, сохраняет информацию в локальный кэш и сообщает **Ответчику** о готовности.
- **Ответчик** повторно обращается к локальному кэшу, извлекает оттуда необходимую информацию, записывает ее в быстрый кеш и передает Клиентской библиотеке в качестве ответа.
- Клиентская библиотека возвращает ответ в функцию **glibc**, на чем поиск завершается.

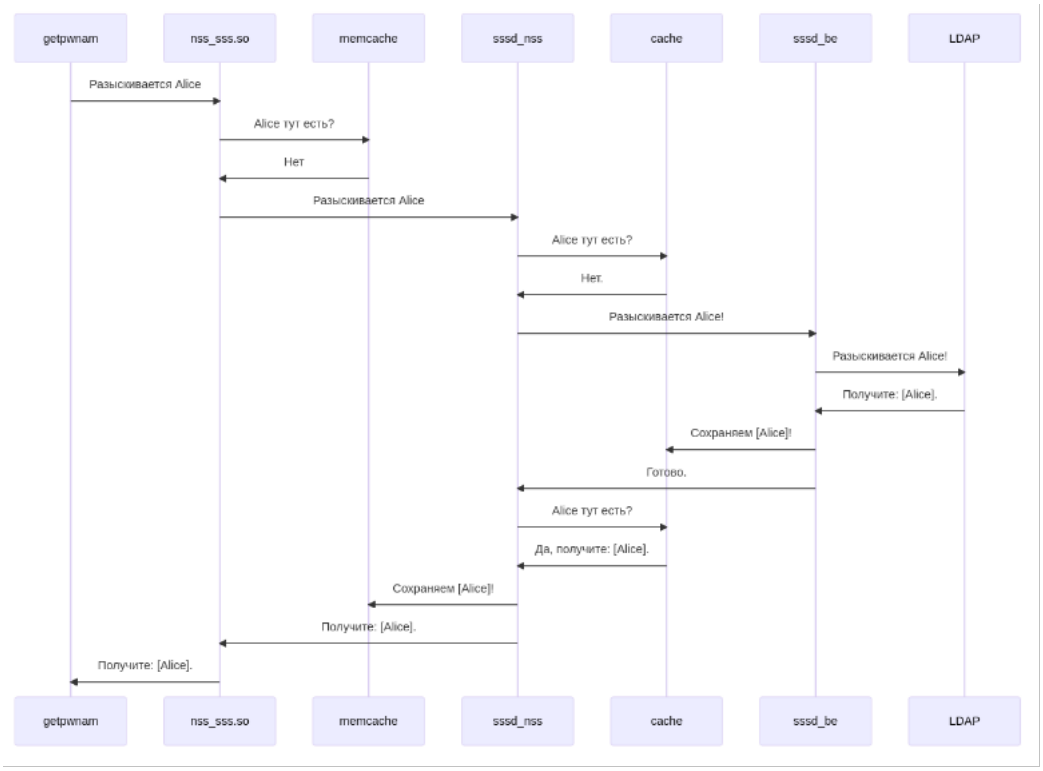


Рисунок 5.5 – Поток данных при поиске информации о пользователе с использованием информации из кэша

5.1.6. Инструменты администрирования

Одним из наиболее полезных приложений для диагностики работы службы **SSSD** является утилита **sssctl** из пакета **sssd-tools**, которая взаимодействует с компонентами **SSSD** напрямую через **SBus**. С ее помощью можно узнать, например, какой контроллер в данный момент обслуживает запросы компьютера:

```
sudo sssctl domain-list
```

Результат команды:

```
ald.company.lan
```

Узнать статус домена можно командой `domain-status` утилиты **sssctl**:

```
sudo sssctl domain-status ald.company.lan
```

Результат выполнения:

```
Online status: Online


Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-1.ald.company.lan
- dc-2.ald.company.lan
- dc-3.ald.company.lan
```

С помощью команды `access-report` можно посмотреть HBAC правила, определенные в домене:

```
sudo sssctl access-report ald.company.lan
```

Результат выполнения:

```
2 rules cached

Rule name: allow_all
    User category: all
    Service category: all

Rule name: allow_systemd-user
    User category: all
    Member services: systemd-user
```

С помощью команды `debug-level` можно быстро изменить степень детализации журналирования, которая будет оставаться в силе до перезапуска службы, потому что файл */var/lib/sss/db/config.ldb* пересоздается при запуске службы.

Проверить текущий уровень отладки можно командой:

```
sudo ldbsearch -H /var/lib/sss/db/config.ldb | grep debug | head -n 1
```

Результат выполнения:

```
debug_level: 0x1f7f0
```

Применить новый уровень отладки можно командой:

```
sudo sssctl debug-level 6
```

Проверить новое значение в конфигурации можно командой:

```
sudo ldbsearch -H /var/lib/sss/db/config.ldb | grep debug | head -n 1
```

Результат выполнения:

```
debug_level: 0x07f0
```

5.2. Учетная запись компьютера в домене

У каждого компьютера в домене есть своя собственная учетная запись, с помощью которой он выполняет авторизованные LDAP-запросы к каталогу и проверяет аутентичность пользователей по протоколу **Kerberos V5**. Учетные записи компьютеров хранятся в контейнере `cn=computers,cn=accounts,dc=ald,dc=company,dc=lan`, имя LDAP-записи включает значение атрибута **fqdn**, например, `fqdn=pc-1.ald.company.lan`, а для аутентификации по протоколу Kerberos используется имя принципала в формате `host/pc-1.ald.company.lan@ALD.COMPANY.LAN`.

Билеты службы **SSSD** кэшируются в файле */var/lib/sss/db/ccache_ALD.COMPANY.LAN*, прочитать который можно с помощью утилиты **klist**, используя параметр `-c`:

```
sudo klist -c /var/lib/sss/db/ccache_ALD.COMPANY.LAN
```

Результат выполнения:

```
Ticket cache: FILE:/var/lib/sss/db/ccache_ALD.COMPANY.LAN
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
Default principal: host/dc-1.ald.company.lan@ALD.COMPANY.LAN

Valid starting    Expires    Service principal
10.10.2023 14:40:23 11.10.2023 14:40:23     krbtgt/ALD.COMPANY.LAN@ALD.
 ↪COMPANY.LAN
10.10.2023 14:40:23 11.10.2023 14:40:23     ldap/dc-1.ald.company.lan@ALD.
 ↪COMPANY.LAN
```

Пароль учетной записи хоста хранится в файле */etc/krb5.keytab*, поэтому можно быстро проходить аутентификацию и выполнять запросы от имени хоста. Проверка валидности keytab-файла является одной из рекомендаций по устранению неисправностей, если в журналах **SSSD** появится информация о том, что служба не смогла пройти аутентификацию в домене. Такие ситуации могут возникать, например, если компьютер был удален из домена.

Содержимое keytab-файла можно посмотреть с помощью утилиты **klist**, используя параметр `-k`. Если добавить параметр `-e`, то можно будет увидеть, что для одной версии пароля в файле содержится сразу два ключа, полученные с помощью разных алгоритмов хэширования:

```
sudo klist -ke /etc/krb5.keytab
```

Результат выполнения:

```
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------
 ↪-
   1 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   1 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

Для того, чтобы пройти аутентификацию от имени хоста, достаточно использовать команду **kinit** с параметром `-k`, но если нужно задать путь к keytab-файлу в явном виде, то можно это сделать, используя дополнительный параметр `-t`. Если выполнить команду `ldapsearch`, то можно будет увидеть наличие сервисного билета для аутентификации в службе **LDAP**:

```
sudo kinit -kt /etc/krb5.keytab
ldapsearch > /dev/null
```

(продолжение на следующей странице)

```
klist
```

Результат выполнения

```
Ticket cache: KEYRING:persistent:1421600000:krb_ccache_zgnOUA8
Default principal: host/dc-1.ald.company.lan@ALD.COMPANY.LAN
Valid starting Expires Service principal
10.10.2023 14:43:06 11.10.2023 14:43:06     ldap/dc-1.ald.company.lan@ALD.
 ↪COMPANY.LAN
10.10.2023 14:43:06 11.10.2023 14:43:06     krbtgt/ALD.COMPANY.LAN@ALD.
 ↪COMPANY.LAN
```

Для хэширования паролей используются устойчивые к взлому алгоритмы, и перед этим к паролю добавляется основное имя пользователя в качестве соли для того, чтобы исключить возможность перебора по таблицам хэшей, но даже такие меры не защищают от всех видов атак, поэтому пароли хостов рекомендуется время от времени менять, например, компания Microsoft предлагает делать это не реже одного раза в 30 дней.

Чтобы установить хосту новый пароль и выгрузить его в keytab-файл, можно воспользоваться утилитой **ipa-getkeytab**. Важно обратить внимание, что после смены пароля команда **klist** в колонке **KVNO** показывает новую версию ключей.

```
sudo rm -rf /etc/krb5.keytab
sudo ipa-getkeytab -p host/pc-1.ald.company.lan -k /etc/krb5.keytab -s dc-1.
 ↪ald.company.lan -D uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan -
 ↪W -P
```

Результат выполнения:

```
Введите пароль LDAP: **********
Новый пароль учётной записи: **********
Проверка пароля учётной записи: **********
Таблица ключей успешно получена и сохранена в: /etc/krb5.keytab
```

Проверить новый keytab-файл можно командой:

```
sudo klist -ke /etc/krb5.keytab
```

Результат выполнения:

```
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
↪-
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

Поясним параметры вызова утилиты **ipa-getkeytab**:

- `-p` — имя учетной записи, для которой меняется пароль
- `-k` — имя keytab-файла, в который добавляются ключи
- `-s` — адрес контроллера домена, через который следует выполнить запрос
- `-D` — имя привилегированной учетной записи, от которой вносятся изменения в каталог.
- `-W` — ключ, указывающий на то, что пароль привилегированной учетной записи будет предоставлен в интерактивном режиме. Для того, чтобы задать пароль прямо в командной строке, используйте ключ `-w`, но делать это крайне не рекомендуется, т.к. значение этого параметра будет видно в выводе команды `ps` и попадет в историю команд.
- `-P` — ключ, указывающий на то, что пароль хоста будет введен вручную. Если этот параметр не указывать, пароль будет сгенерирован автоматически.

Если параметр `-D` не будет указан, то утилита **ipa-getkeytab** воспользуется учетными данными текущего Kerberos-пользователя из связки ключей. Это позволяет сменить пароль для компьютера, используя учетные данные самого хоста из keytab-файла:

```
sudo kinit -kt /etc/krb5.keytab
sudo ipa-getkeytab -p host/pc-1.ald.company.lan -k /etc/krb5.keytab
sudo klist -ke /etc/krb5.keytab
```

Результат выполнения:

```
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
↪-
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
  3 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
  3 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

5.3. Способы аутентификации в домене

Служба каталога ALD Pro построена на базе **FreeIPA** и поддерживает три вида аутентификации: **LDAP Bind**, **Kerberos** и частично **NTLM**.

5.3.1. Аутентификация по протоколу LDAP

Простая аутентификация или привязка (**bind**) является самым распространенным, но при этом не самым безопасным способом аутентификации, т.к. пароль передается на сервер открытым текстом.

Поэтому во избежание разглашения учетных данных при выполнении авторизованных LDAP-запросов с использованием простой аутентификации рекомендуется всегда использовать шифрованный протокол **LDAPS** или включать **StartTLS**:

```
ldapsearch -H ldaps://dc-1.ald.company.lan -D 'uid=admin,cn=users,cn=accounts,
↪dc=ald,dc=company,dc=lan' -W
```

А также командой с ключом `-ZZ`:

```
ldapsearch -ZZ -H ldap://dc-1.ald.company.lan -D 'uid=admin,cn=users,
↪cn=accounts,dc=ald,dc=company,dc=lan' -W
```

Для лучшего понимания угрозы приведен пример, как делать категорически запрещается:

```
ldapsearch -H ldap://dc-1.ald.company.lan -D 'uid=admin,cn=users,cn=accounts,
↪dc=ald,dc=company,dc=lan' -W
```

Из программы **Wireshark** можно увидеть, см. *Пароль пользователя открытым текстом в запросе bindRequest*, что злоумышленник, у которого будет доступ к просмотру сетевых пакетов, может легко извлечь пароль «**AstraLinux_172**» из запроса, если канал связи не будет зашифрован:

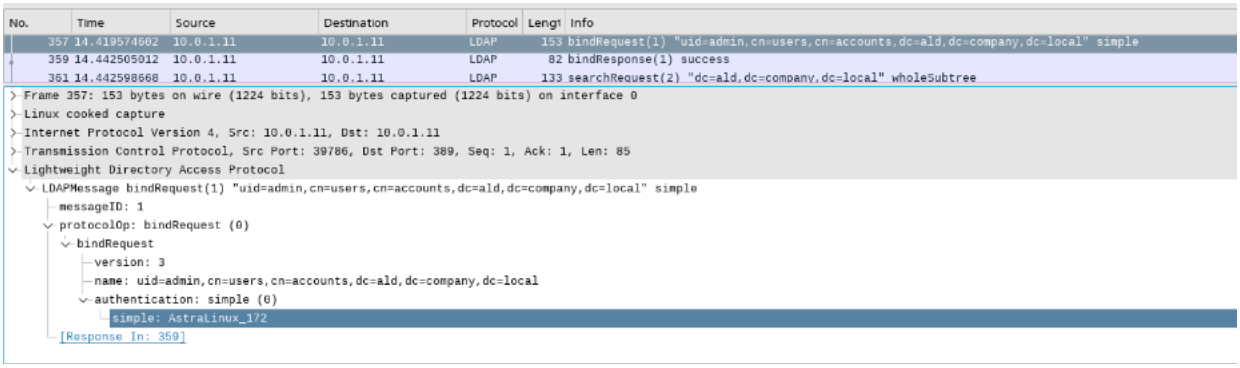


Рисунок 5.6 – Пароль пользователя открытым текстом в запросе bindRequest

Для проведения простой аутентификации LDAP-каталог хранит хэш пароля в атрибуте `userPassword`. Алгоритм проверки аутентичности работает следующим образом:

- Пользователь передает на сервер свое имя и пароль в открытом виде.
- Сервер извлекает из каталога хэш пароля для указанного пользователя.
- Пароль, полученный в запросе на аутентификацию, хэшируется тем же алгоритмом и сравнивается с значением, полученным из каталога. Если значения совпали, аутентификация считается успешной.

Хэширование паролей выполняется автоматически при изменении пароля, по умолчанию используется устойчивый к взлому алгоритм PBKDF2_SHA256:

```
ldapsearch -H ldaps://dc-1.ald.company.lan -o ldif-wrap=no -D 'cn=Directory
↪Manager' -W -b 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' |
↪grep 'userPassword::' | cut -d" " -f2 | base64 -d
```

Где:

- опция `-o ldif-wrap=no` - выводит текст без переноса в одну строку, если она не помещается в 80 символов, что полезно использовать в **bash** командах и скриптах;
- командой `grep` мы делаем поиск строки с полем `userPassword::` и два символа «**:**» означает что строка закодирована в кодировке **base64**;
- командой `cut` отрезаем по символу пробел значение пароля во второй колонке;
- командой `base64 -d` значение из **base64** декодируем в текст.

Результат выполнения, декодированный из **base64** пароль пользователя admin:

```
{PBKDF2_SHA256}AAAIAPiA62CPFChuE0ZU908GI9DjhCb3fXPR4mWPH9/0molxZd7/
↪nWEGglROvy6WCXqri61+Wi76wyVSop6igG2JQR2RyHSI7Fxarm7Ur/2Fri70hw4kBl09Ih...
```

Если включить режим миграции пользователей, то при создании новых учетных записей в атрибут `userPassword` можно будет записать хэш пароля, при этом поддерживается целый ряд алгоритмов: CRYPT, CRYPT-MD5, CRYPT-SHA256, CRYPT-SHA512, MD5, SHA, SHA256, SHA384, SHA512, SMD5, SSHA, SSHA256, SSHA384, SSHA512.

```
sudo ipa config-mod --enable-migration=true
sudo ipa user-add userName --setattr userPassword='{TYPE}HASH'
```

Для завершения миграции пользователю потребуется один раз выполнить простую LDAP-аутентификацию, чтобы сервер смог сгенерировать недостающие ключи. Для этого на сервере существует даже специальная страница миграции, доступная по адресу https://dc-1.ald.company.lan/ipa/migration/, см. *Страница LDAP-аутентификации для завершения миграции пользователей*.

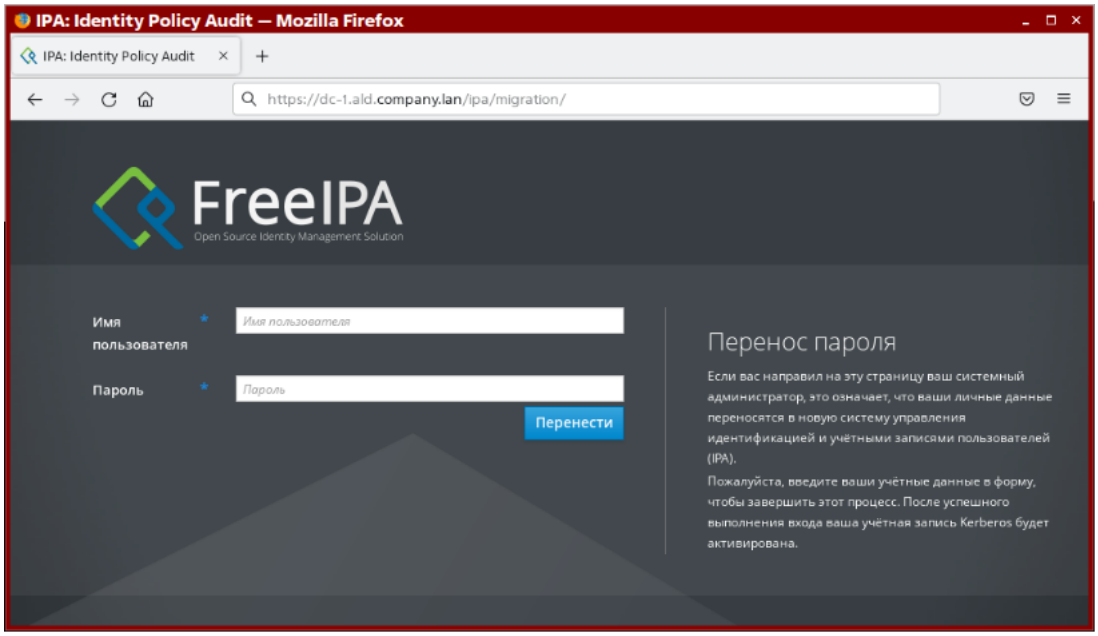


Рисунок 5.7 – Страница LDAP-аутентификации для завершения миграции пользователей

5.3.2. Аутентификация по протоколу Kerberos

Основным протоколом аутентификации в домене является Kerberos V5. Три участника процедуры аутентификации:

- **Клиент** (**Client**) — субъект, желающий получить доступ к ресурсу.
- **Сервер приложения** (Application Server, **AP**) — служба, к ресурсу которой **клиент** хочет получить доступ.
- **Центр распределения ключей** (Key Distribution Center, **KDC**) — доверенная сторона,

отвечающая за аутентификацию (Authentication Service, **AS**) пользователей и выпуск билетов для доступа к сетевым службам в домене (Ticket Granting Server, **TGS**).

Протокол аутентификации был разработан для эксплуатации в незащищенных компьютерных сетях, когда сетевые пакеты могут быть подслушаны и изменены злоумышленником. Рассмотрим процесс аутентификации в упрощенном виде, см. *Начало аутентификации пользователя Алисы*, опуская несущественные детали, такие как фаза предварительной аутентификации, использование nonce и др.

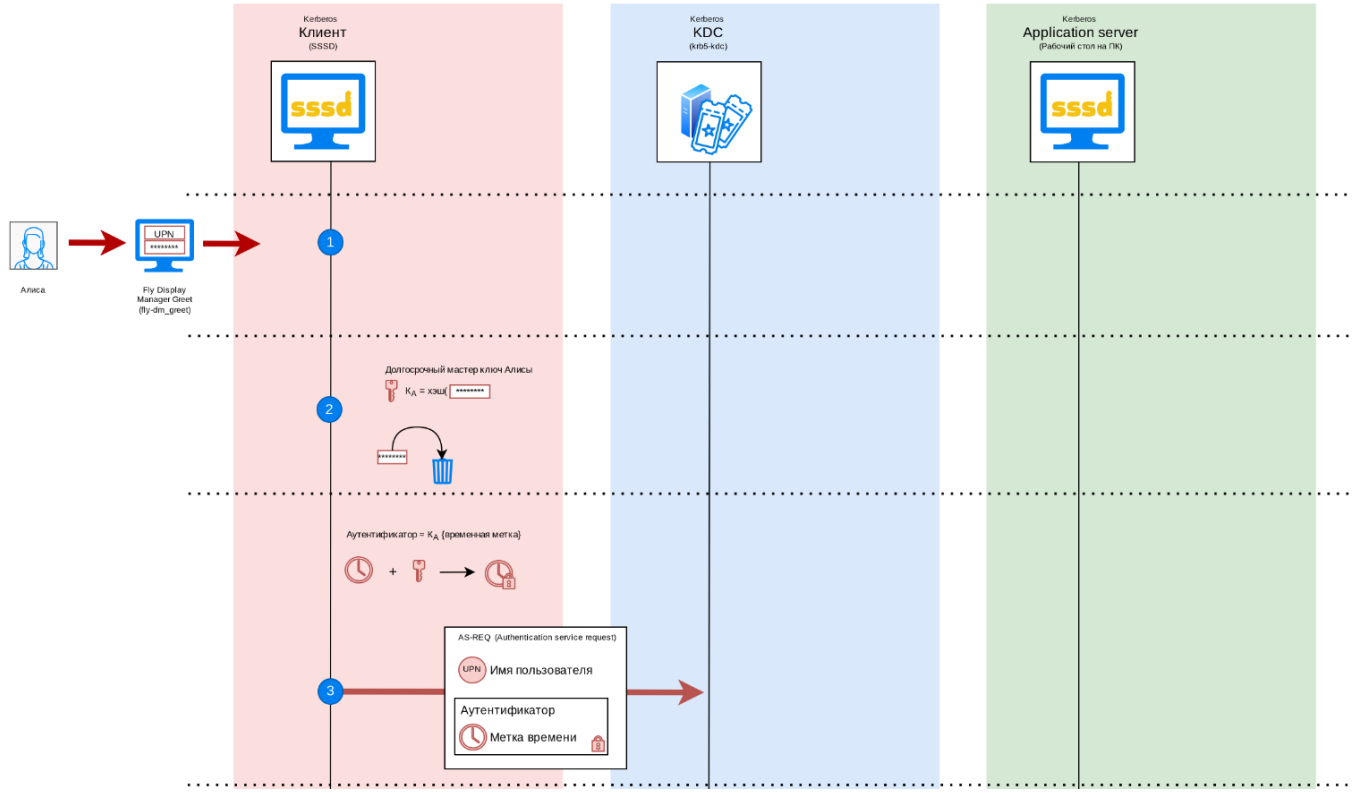


Рисунок 5.8 – Начало аутентификации пользователя Алисы

Шаг (1) Пользователь Алиса через приложение графического входа (Fly Display Manager Greet) передает стеку модулей аутентификации (Pluggable Authentication Modules, **PAM**) логин и пароль в открытом виде. За аутентификацию в домене отвечает модуль */usr/lib/x86_64-linux-gnu/security/pam_sss.so*, см. файл */etc/pam.d/common-auth*:

```
cat /etc/pam.d/common-auth
```

Результат выполнения:

```
...
auth [success=1 default=ignore] pam_sss.so use_first_pass
...
```

Шаг (2) Библиотека **libpam** загружает Клиентскую библиотеку *pam_sss.so*, через функции которой обращается к **Ответчику** */usr/lib/x86_64-linux-gnu/sssd/sssd_pam*. **Ответчик** обращается к **Бэкенду**, который порождает процесс **krb5_child** для аутентификации в домене по протоколу **Kerberos V5**.

Керберос клиент рассчитывает долгосрочный мастер-ключ Алисы (UPN long-term key или Master key) и может удалить из памяти компьютера пароль в открытом виде для повышения устойчивости системы к взлому. Хэш является соленым, к паролю добавляется основное имя принципала, включающее логин пользователя и реалм домена. В соответствии с политикой Kerberos для хэширования может использоваться алгоритм **AES128_CTS_HMAC_SHA1_96** или **AES256_CTS_HMAC_SHA1_96**.

Шаг (3) Клиент отправляет запрос службе аутентификации Центра распределения ключей (Key Distribution Center, **KDC**). В четвертой версии протокола на этом шаге никакие учетные данные не требовались, т.к. ответ будет зашифрован долгосрочным ключом пользователя, и способность расшифровать ответ является подтверждением того, что пользователь является тем, за кого себя выдает. Но с пятой версии протокола сервер отклонит запрос и потребует предварительную аутентификацию. В качестве аутентификатора выступает метка времени, зашифрованная симметричным алгоритмом с помощью долгосрочного ключа клиента. Такая проверка существенно затрудняет кибератаки, так как лишает злоумышленника возможности запросить **TGT**-билет на любого пользователя в системе.

Шаг (4) KDC расшифровывает аутентификатор, используя хэш пароля Алисы из LDAP-каталога, см. *Обработка запроса на стороне KDC, создание и отправка TGT-билета*. Ключи для аутентификации по протоколу Kerberos хранятся в атрибуте **krbPrincipalKey**, который представляет из себя бинарный объект, зашифрованный мастер-ключом **KDC** (**krbMKey**). Если процедура расшифровки завершилась успешно и полученная временная метка расходится с временем сервера не более, чем на 5 минут, то считается, что предварительная аутентификация пройдена успешно. По этой причине для корректной работы протокола важна синхронизация времени между всеми участниками.

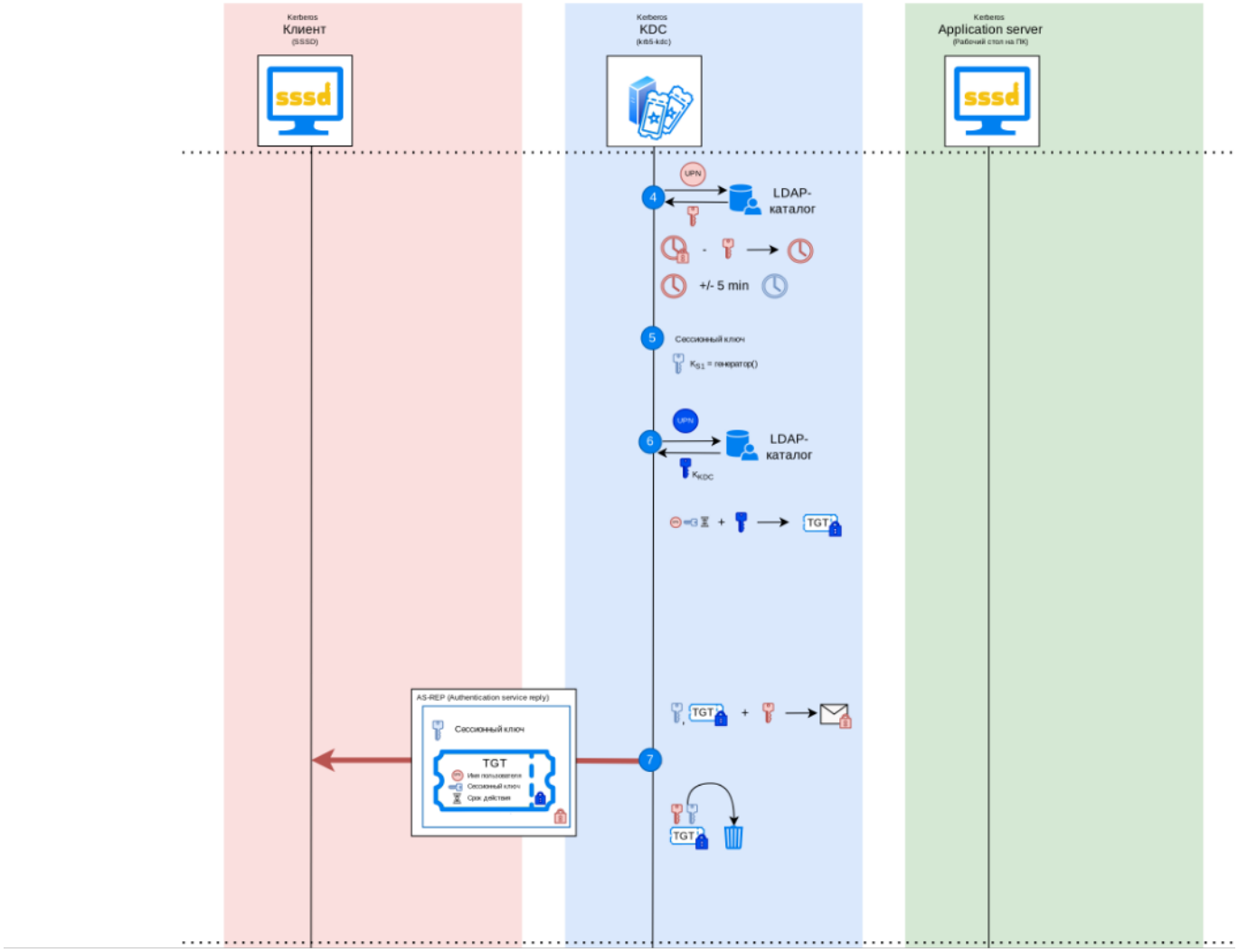


Рисунок 5.9 – Обработка запроса на стороне KDC, создание и отправка TGT-билета

Шаг (5) Для повышения безопасности системы **KDC** генерирует временный сессионный ключ (**S1**) для последующей передачи его **клиенту**, чтобы использовать в дальнейшем для шифрования сообщений между **клиентом** и **KDC** вместо хэша пароля пользователя.

Шаг (6) Несмотря на то, что сессионный ключ был сгенерирован сервером, в домене может быть несколько контроллеров, и **Клиент** вправе обратиться с последующим запросом к любому из них. Учитывая случайный характер сессионного ключа, другой контроллер домена не сможет рассчитать его математически, поэтому сессионный ключ следует передать ему в явном виде. По протоколу Kerberos ответственность за передачу сессионного ключа возлагается на **клиента**, для чего ему выдается зашифрованный билет на выдачу билетов (Ticket-granting ticket, **TGT**), который он должен предъявлять в **KDC** при последующих обращениях.

Внутри **TGT**-билета содержится имя пользователя, сессионный ключ и информация по сроку действия билета. Билет зашифрован симметричным алгоритмом с помощью долгосрочного ключа **KDC** (хэш пароля служебной учетной записи **KRBTGT**, Key

Distribution Center Service Account), поэтому расшифровать его можно только на контроллере, и подделать билет на стороне **Клиента** невозможно.

Шаг (7) Сессионный ключ и билет шифруются симметричным алгоритмом с помощью долгосрочного ключа клиента, поэтому только **клиент** сможет расшифровать сообщение, подтверждая этим фактом, что является тем, за кого себя выдает. Данная проверка аутентичности считается основной.

Долгосрочный ключ клиента, сессионный ключ и билет могут быть удалены из памяти сервера за ненадобностью для повышения безопасности системы.

Шаг (8) Клиент расшифровывает сессионный ключ и **TGT** билет своим долгосрочным ключом, см. *Запрос от клиента на получение сервисного билета с помощью TGT-билета*. Возможность использования этих данных в последующих запросах означает, что **Клиент** является тем, за кого себя выдает. Если результат расшифровки окажется недействительным, значит ответ получен от подставного Центра распределения ключей. Долгосрочный ключ может быть удален из памяти компьютера за ненадобностью для повышения безопасности системы.

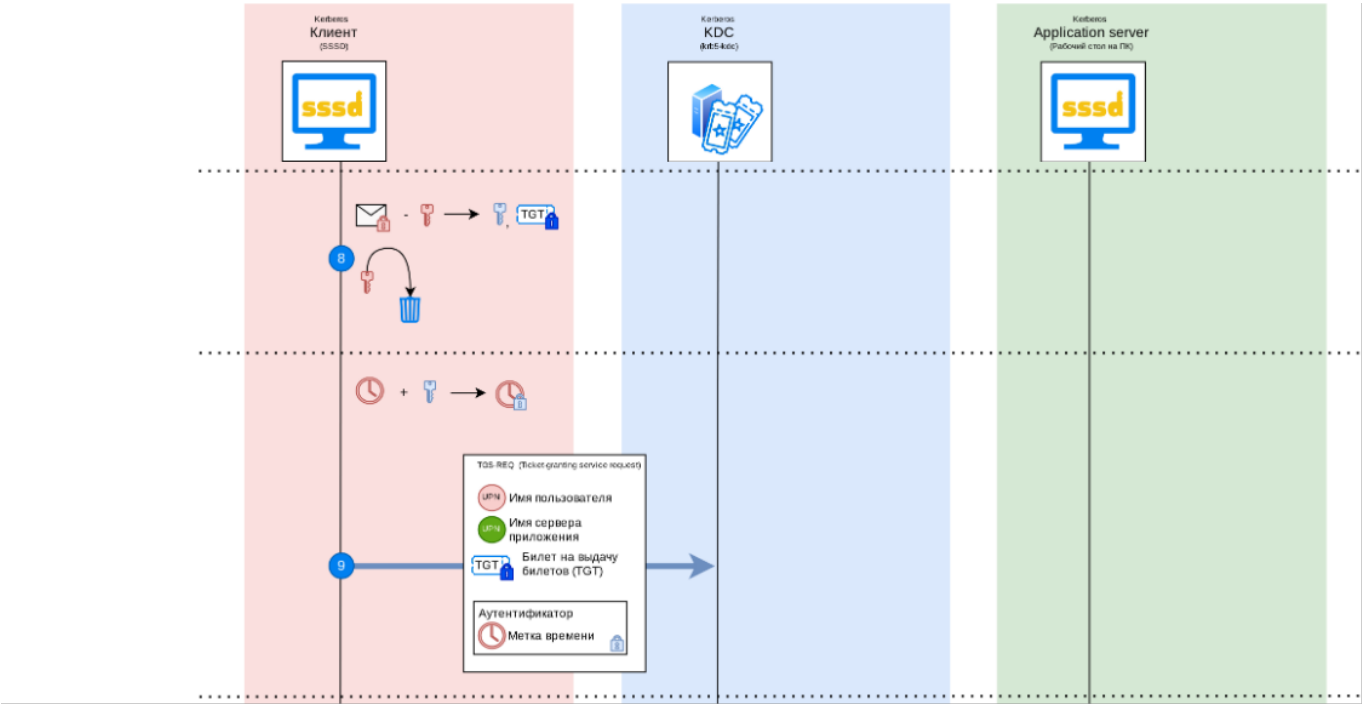


Рисунок 5.10 – Запрос от клиента на получение сервисного билета с помощью TGT-билета

Шаг (9) Процесс **krb5_child** сохраняет **TGT** билет в связке ключей Linux и передает результаты по цепочке **Бэкенд** — **Ответчик** — **Клиентская библиотека**, после чего **Клиентская библиотека** отправляет запрос на получение сервисного билета на доступ к серверу приложения, в котором содержится имя пользователя (**user principal name**), имя сервера приложения (**service principal name**), билет на выдачу билетов (**TGT**) и

аутентификатор. В качестве аутентификатора выступает метка времени, зашифрованная симметричным алгоритмом с помощью сессионного ключа **S1**. В качестве имени сервиса при входе в компьютер выступает **host/pc-fqdn**.

На этом шаге **TGT** билет закодирован только долгосрочным ключом **KDC**, но этого достаточно, чтобы злоумышленник, перехвативший сообщение, не смог подделать запросы, т.к. он не располагает сессионным ключом **S1**.

Шаг (10) KDC расшифровывает информацию из **TGT** билета, используя долгосрочный ключ **KDC** из LDAP-каталога, после чего ему становится доступна следующая информация: имя пользователя, сессионный ключ **S1** и срок действия билета, см. *Обработка запроса на сессионный ключ и выдача сессионного ключа по запросу клиента*.

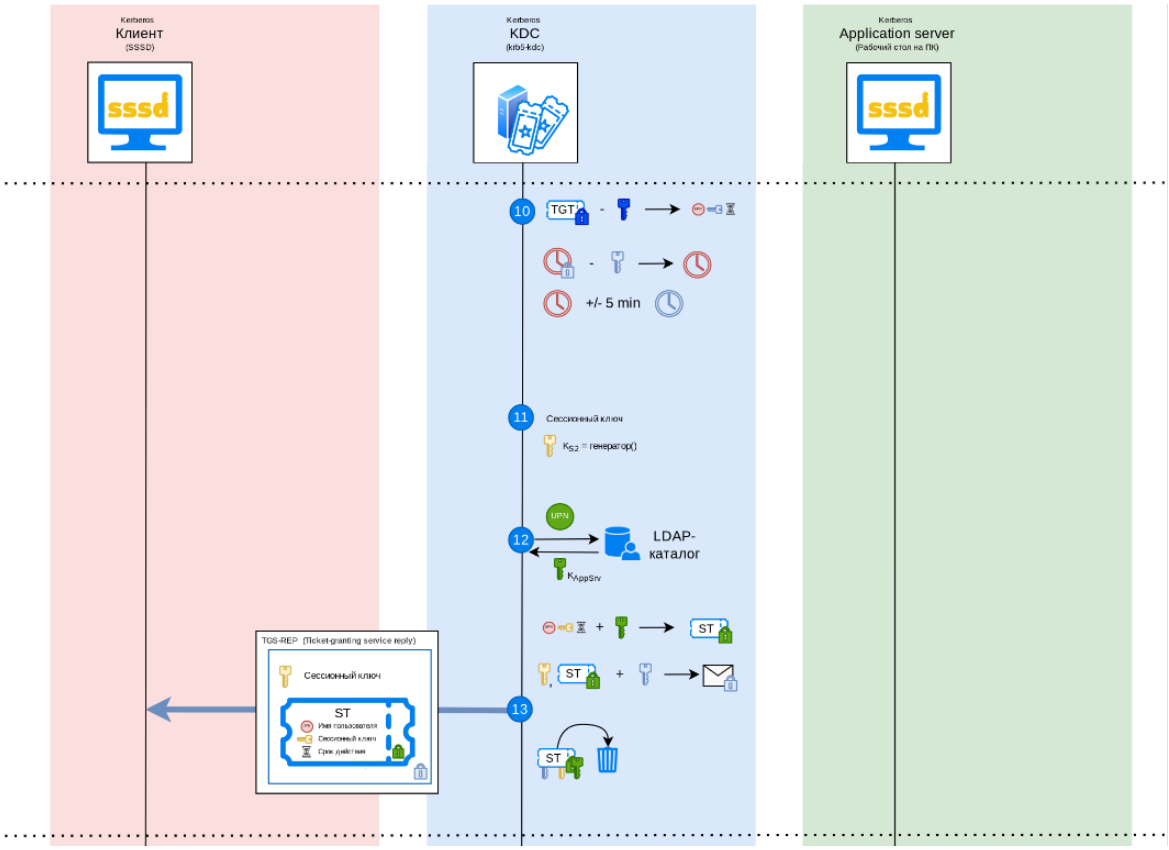


Рисунок 5.11 – Обработка запроса на сессионный ключ и выдача сессионного ключа по запросу клиента

Сервер расшифровывает аутентификатор, используя сессионный ключ из **TGT** билета, и, если полученное значение расходится с временем сервера не более, чем на 5 минут, то считается, что аутентификация пройдена успешно.

Шаг (11) Для повышения безопасности протокола **KDC** генерирует новый сессионный ключ (**S2**) для последующей передачи его клиенту, чтобы использовать в дальнейшем для шифрования сообщений между клиентом и сервером приложения вместо сессионного

ключа **S1**.

Шаг (12) Ключ **S2** был сгенерирован сервером **KDC** и его следует передать Серверу приложения. По протоколу Kerberos ответственность за передачу ключа возлагается на клиента, для чего ему выдается зашифрованный сервисный билет (Service ticket, **ST**), который он должен предъявлять серверу приложения.

Внутри **ST**-билета содержится имя пользователя, сессионный ключ и информация по сроку действия билета. Билет зашифрован симметричным алгоритмом с помощью долгосрочного ключа Сервера приложения, поэтому расшифровать его сможет только сервер приложения, и подделать билет на стороне **Клиента** невозможно.

У приложения может быть отдельная учетная запись, но может использоваться и учетная запись компьютера, на котором это приложение работает. В нашем случае это пароль компьютера, в который мы хотим войти.

Шаг (13) После передачи сервисного билета **Клиенту** информация о ключах больше не требуется и может быть удалена для повышения безопасности системы.

Шаг (14) Клиент расшифровывает сессионный ключ **S2** и сервисный билет **ST** известным ему сессионным ключом **S1**. Возможность использования этих данных в последующих запросах означает, что **Клиент** является тем, за кого себя выдает. Если результат расшифровки окажется недействительным, значит ответ получен от подставного Центра распределения ключей.

Шаг (15) Клиент отправляет серверу приложения запрос на аутентификацию, в котором содержится имя пользователя, сервисный билет (**ST**) и аутентификатор. В качестве аутентификатора выступает метка времени, зашифрованная симметричным алгоритмом с помощью сессионного ключа **S2**.

На этом шаге **ST** билет закодирован только долгосрочным ключом сервера приложения, но этого достаточно, чтобы злоумышленник, перехвативший сообщение, не смог подделать запросы.

Шаг (16) Сервер приложения, в роли которого выступает служба **SSSD** на пользовательском компьютере, извлекает хэш пароля из файла */etc/krb5.keytab* для расшифровки запроса. Используя этот долгосрочный ключ, служба **SSSD** может расшифровать информацию из сервисного билета (**ST**), после чего ей становится доступна следующая информация: имя пользователя, сессионный ключ **S2** и срок действия билета. **SSSD** расшифровывает аутентификатор, используя сессионный ключ **S2** из сервисного билета, и, если полученное значение расходится с временем компьютера не более, чем на 5 минут, то считается, что аутентификация пройдена успешно (см. *Запроса*

клиента к службе с помощью сервисного ключа).

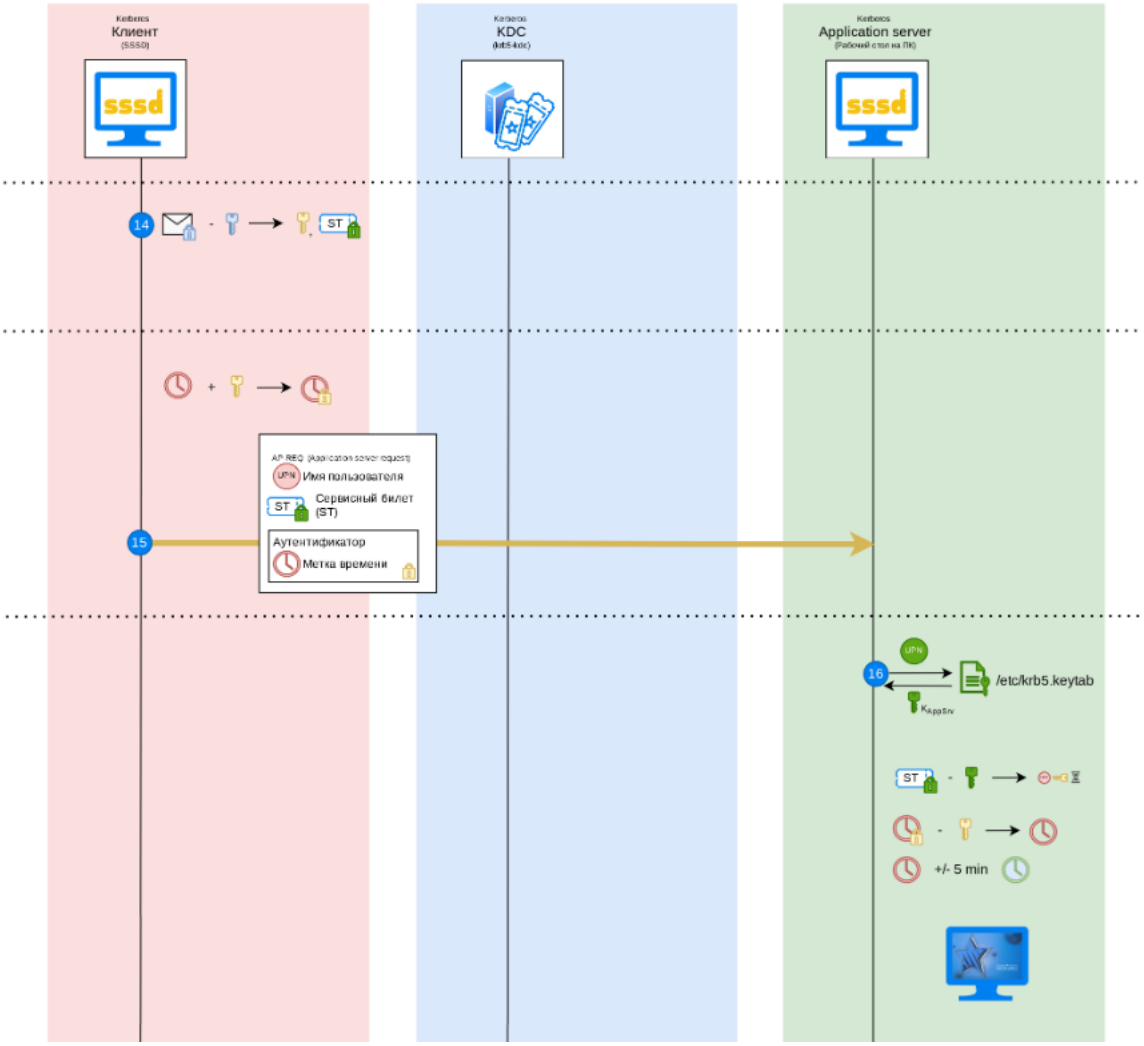


Рисунок 5.12 – Запроса клиента к службе с помощью сервисного ключа

Получив подтверждение, что вход в компьютер хочет выполнить действительно Алиса, приложение **Display Manager** запускает рабочий стол (Fly Windows Manager, **fly-wm**) от имени этого пользователя.

Сервисный билет на доступ к хосту более не требуется, поэтому он не сохраняется в связке ключей, и вы не увидите его в выводе команды **klist**, но на контроллере в журнале */var/log/auth.log* вы можете найти подтверждение, что такой билет запрашивался и был предоставлен рабочей станции.

5.3.3. Аутентификация по протоколу NTLM

Служба каталога **FreeIPA** тесно интегрирована с файловым сервером **Samba**, в котором реализована поддержка **NTLM**-протокола для возможности аутентификации

пользователей по логину/паролю, например, если подключение выполняется не по доменному имени сервера, а по его IP адресу. Для этого у пользователей в каталоге есть атрибут `ipaNTHash`, который обновляется плагином **ipa_pwd_extop** каждый раз при смене пароля.

Рассмотрим поток данных при аутентификации по протоколу **NTLM** в упрощенном виде, см. *Поток данных при NTLM-аутентификации*:

- **Клиент** обращается к **Сервису** с запросом на аутентификацию (1) и получает в ответ случайное число длиной 8 байт, которое ему нужно подписать, используя свой пароль, для подтверждения аутентичности (2).
- **Клиент** подписывает сообщение и передает его **Сервису** (3), который пересылает подпись вместе с исходным сообщением на **Контроллер Домена** (4).
- **Контроллер** проверяет подпись и возвращает **PAC** сертификат, если проверка аутентичности пройдена успешно (5).

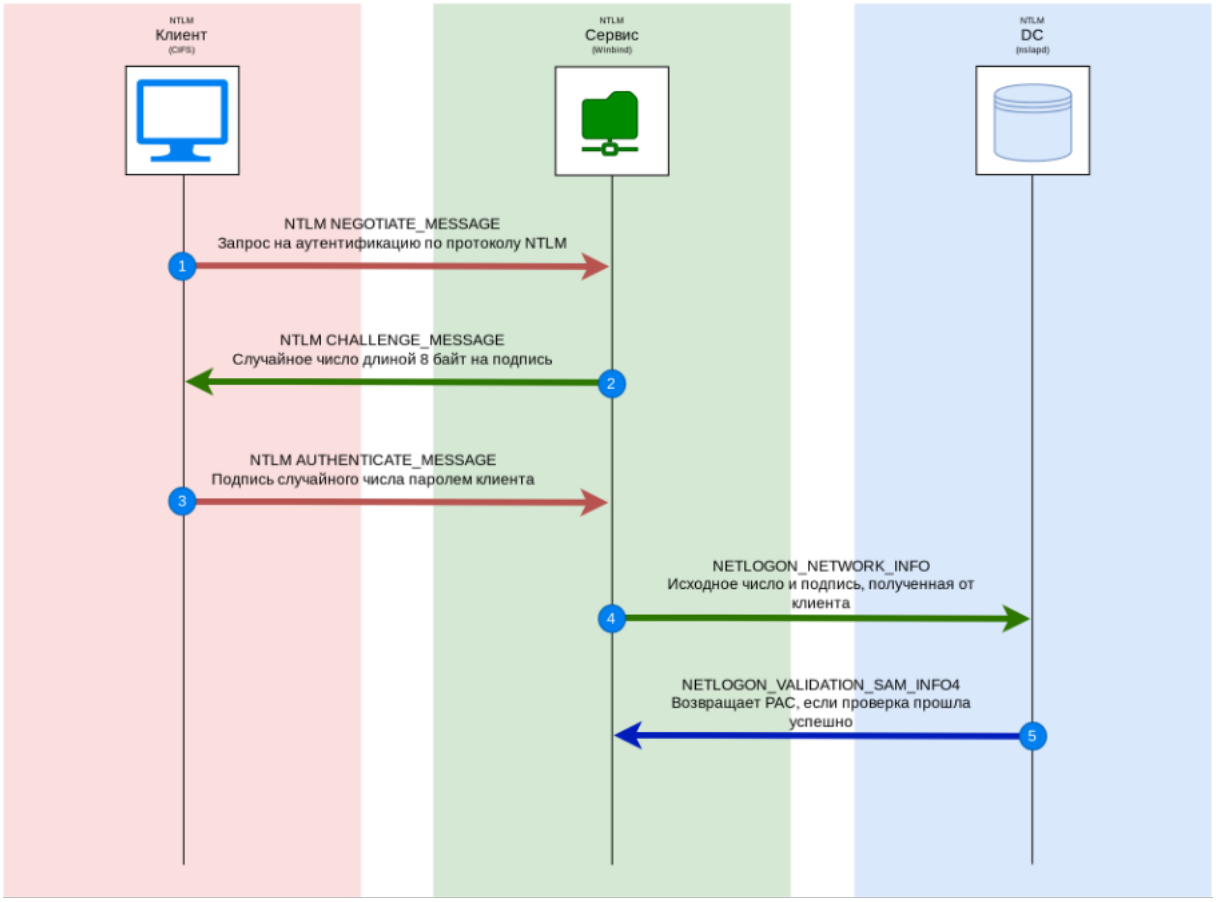


Рисунок 5.13 – Поток данных при NTLM-аутентификации

Интеграция **Samba** реализует представленную схему не полностью: служба **Winbind**, которая отвечает за **NTLM** аутентификацию на стороне **Samba**, не передает контроллеру домена информацию для проверки аутентичности пользователя, а выполняет ее

самостоятельно, получая значение атрибута `ipaNTHash` из каталога напрямую через модуль **ipasam**. Доступ к этому атрибуту предоставляется учетной записи файлового сервера через назначение роли **CIFS server**.

Указанный способ интеграции допустим, если **Samba** устанавливается на контроллер домена, но создает дополнительную точку уязвимости, если **Samba** используется отдельно, как сервер общего доступа к файлам. Учитывая вышеизложенное, ИТ-директорам нужно либо отключать возможность **NTLM**-аутентификации в домене, либо делегировать управление файловыми серверами тем же администраторам, кто управляет контроллерами домена.

5.4. Безопасный обмен данными с применением SSL/TLS

Доменный компьютер взаимодействует с подсистемами ALD Pro по разным протоколам, некоторые из которых используют шифрование:

- Доступ к каталогу по протоколам **LDAP+StartTLS** (**TCP/389**) и **LDAPS** (**TCP/636**) со стороны службы **SSSD** и с помощью инструментов пакета **ldap-utils**.
- Доступ к веб-интерфейсам контроллера домена по протоколу **HTTPS** (**TCP/443**) с помощью браузера и к **REST API** с помощью утилиты автоматизации **ipa**.
- Доступ к удаленному рабочему столу доменного компьютера с портала управления ALD Pro по протоколу **VNC** через **HTTPS** (**TCP/6080** - **TCP/608x**)
- Аутентификация с использованием расширения **PKINIT** по протоколу **Kerberos** (**TCP/88**) использует **X.509** сертификаты для взаимной аутентификации центра распределения ключей и клиента, но не является примером **SSL**-протокола.

Сертификаты применяются также для доступа к **PostgreSQL** (**TCP/5432**), взаимодействия со службой печати по протоколу **IPP** (**TCP/631**), а до версии 2.0.0 использовалось **REST API** системы конфигурирования **SaltStack** (**TCP/8000**), но все эти примеры не представляют интереса в виду локального характера сетевого взаимодействия.

5.4.1. Механизм защиты данных по протоколу SSL

Протокол **SSL** (Secure Sockets Layer, уровень защищенных сокетов) в его текущей реализации **TLS** (Transport Layer Security, безопасность транспортного уровня) обеспечивает быстрый и безопасный обмен данными между клиентом и сервером за счет

сочетания симметричных и асимметричных алгоритмов шифрования: медленные асимметричные алгоритмы используются только на фазе рукопожатия для того, чтобы договориться о параметрах шифрования и безопасно обменяться сессионным ключом, а непосредственно обмен данными происходит уже с применением быстрых симметричных алгоритмов.

Разница между этими алгоритмами заключается в том, что симметричные алгоритмы позволяют расшифровать данные тем же ключом, с помощью которого они были зашифрованы, просто выполняя те же действия в обратном порядке, а в асимметричных алгоритмах используется пара взаимосвязанных ключей, поэтому данные, зашифрованные одним из ключей, могут быть расшифрованы только вторым ключом из той же пары.

Суть работы асимметричных алгоритмов можно продемонстрировать на примере из жизни, см. *Иллюстрация идеи асимметричного шифрования на примере использования сейфа с самозакрывающимся механизмом*.

Представьте, что для установления безопасного канала связи нужно обменяться с второй стороной общим кодовым словом. Для этого через службу доставки пересылается ей открытый сейф с самозакрывающимся механизмом, оставляя ключ у себя. Вторая сторона должна будет поместить в этот сейф записку с кодовым словом, защелкнуть замок и переслать сейф обратно. Не взламывая замок, открыть этот сейф можно будет только на вашей стороне, т.к. ключ есть только у вас.

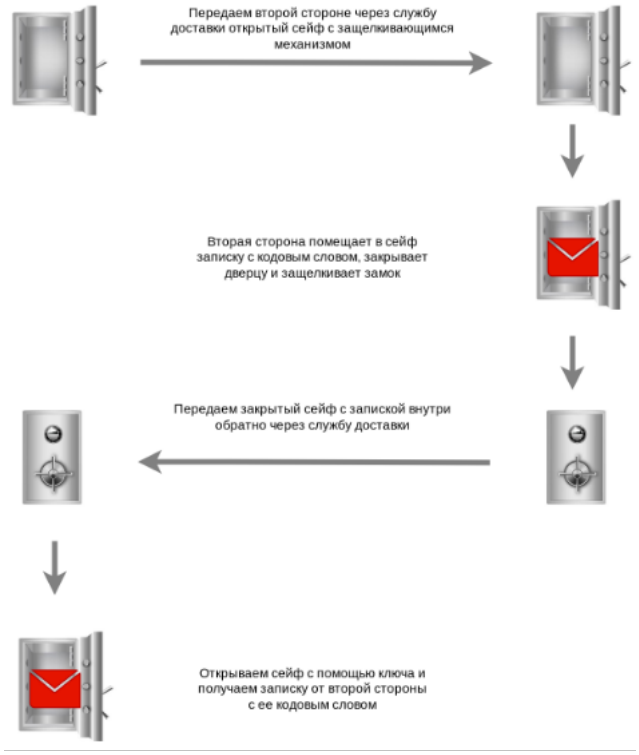


Рисунок 5.14 – Иллюстрация идеи асимметричного шифрования на примере использования сейфа с самозакрывающимся механизмом

Изначально протокол **SSL** разрабатывался для **HTTP**, но его универсальный **TCP**-подобный интерфейс позволил создать расширения и других транспортных протоколов прикладного уровня, например, **LDAPS**, **FTPS**, **SMTPS**, **IMAPS** и др. В качестве примера рассматривается открытие веб-страницы по **HTTPS**, см. *Установление безопасного HTTPS-соединения*.

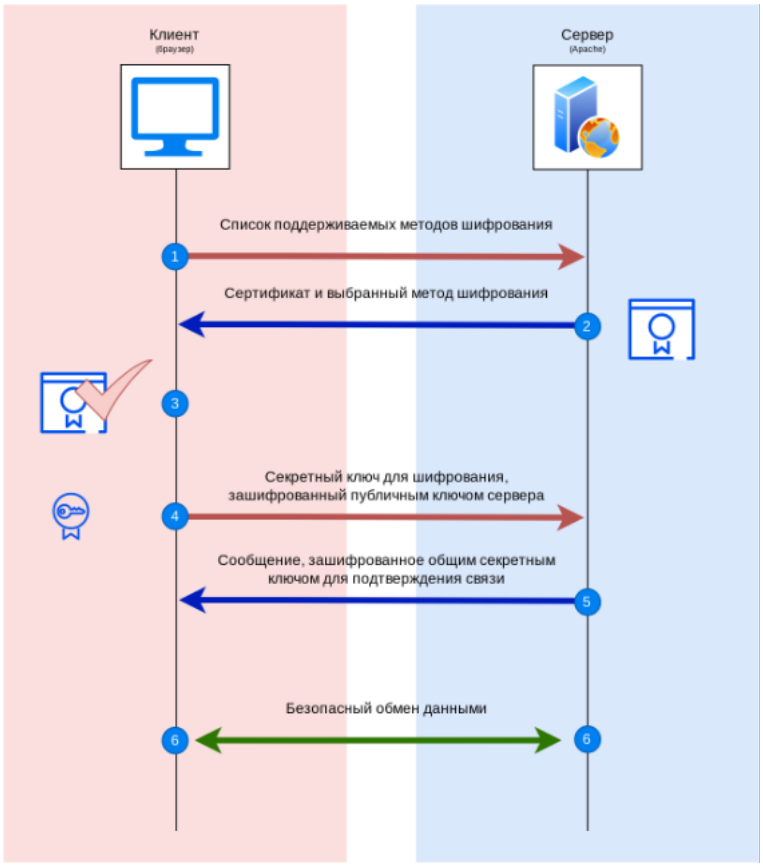


Рисунок 5.15 – Установление безопасного HTTPS-соединения

Шаг (1) Когда пользователь переходит по **https**-ссылке, браузер выполняет подключение к **веб-серверу** на **TCP/443** порт и передает список поддерживаемых методов шифрования для установления безопасного соединения.

Шаг (2) Веб-сервер выбирает метод шифрования и передает **клиенту** свой **SSL**-сертификат, который содержит публичный ключ и цифровую подпись для этого ключа. Вернее, он передает не один сертификат, а всю цепочку до корневого сертификата включительно.

Шаг (3) Клиент проверяет валидность цифровой подписи по своей базе доверенных корневых сертификатов, используя алгоритмы асимметричного шифрования. Также **клиент** проверяет подлинность **сервера**, сопоставляя имя, по которому обращается с именем указанным внутри сертификата. Если подпись не пройдет проверку, то пользователь получит сообщение о небезопасности соединения ввиду возможности атак посредника (Man in the middle, **MITM**)

Шаг (4) Клиент генерирует случайный ключ и шифрует его асимметричным алгоритмом публичным ключом **сервера** для безопасной передачи данных по сети.

Шаг (5) Сервер передает **клиенту** сообщение, зашифрованное общим ключом,

подтверждающее возможность начала безопасного обмена данными.

Шаг (6) Клиент и **сервер** выполняют обмен данными по защищенному каналу связи.

5.4.2. Доступ к каталогу по протоколам LDAP+StartTLS и LDAPS

Служба **SSSD** получает данные из каталога по протоколу **LDAP** (**TCP/389**) с использованием расширения **StartTLS**, которое позволяет инициировать шифрованный обмен данными внутри уже установленного **TCP**-соединения вместо открытия нового соединения на отдельном порту.

Администратор может проверить, что порт **TCP/389** действительно защищен шифрованием, и получить цепочку сертификатов с помощью утилиты **openssl**. Первый сертификат выписан на конкретный FQDN контроллера домена, а второй соответствует корневому сертификату домена.

```
echo Q | openssl s_client -starttls ldap -showcerts dc-1.ald.company.lan:389
```

Результат выполнения:

```
CONNECTED(00000003)
depth=1 CN = CA Signing Certificate
verify return:1
depth=0 CN = dc-1.ald.company.lan
verify return:1
---
Certificate chain
0 s:CN = dc-1.ald.company.lan
i:CN = CA Signing Certificate
1 s:CN = CA Signing Certificate
i:CN = CA Signing Certificate
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDxjCCAq6gAwIBAgIUZ4JathCvRrV6v+Aoi4kUNScViBswDQYJKoZIhvcN
AQEL
BQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0y
MjExMTYx
NzM4MjVaFw0zMjExMTMxNzM4MjVaMCExHzAdBgNVBAMMFmRjLTEuY
```

(продолжение на следующей странице)

```
WxkLmNvbXBh
...
-----BEGIN CERTIFICATE-----
MIIDIzCCAgugAwIBAgIUULQd6OzgJ/ik7wJvV0gd6kUIFKAwDQYJKoZIhvcN
AQEL
BQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0y
MjExMTYx
NzM4MjVaFw00MjExMTExNzM4MjVaMCExHzAdBgNVBAMMFkNBIFNpZ
25pbmcgQ2Vy
...
```

Доверие к этой цепочке проверяется с помощью корневого сертификата из файла */etc/ipa/ca.crt*, на который указывает параметр `ldap_tls_cacert` из секции **domain** конфигурационного файла */etc/sssd/sssd.conf*. Данный сертификат загружается на компьютер автоматически при вводе компьютера в домен. Содержимое этого файла соответствует последнему сертификату из цепочки, полученной от сервера.

Открыть конфигурацию *sssd.conf* для проверки сертификата:

```
sudo cat /etc/sssd/sssd.conf | grep ldap_tls_cacert
```

Результат отображает файл */etc/ipa/ca.crt*:

```
ldap_tls_cacert = /etc/ipa/ca.crt
```

Открыть файл сертификата командой:

```
sudo cat /etc/ipa/ca.crt
```

В результате отображается сертификат **X509** формате **Base64String**:

```
-----BEGIN CERTIFICATE-----
MIIDIzCCAgugAwIBAgIUULQd6OzgJ/ik7wJvV0gd6kUIFKAwDQYJKoZIhvcN
AQEL
BQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0y
MjExMTYx
NzM4MjVaFw00MjExMTExNzM4MjVaMCExHzAdBgNVBAMMFkNBIFNpZ
25pbmcgQ2Vy
```

Проверить содержимое сертификата можно с помощью все той же утилитой **openssl**. Как

можно заметить, по умолчанию корневой сертификат выписан на 10 лет, после чего его нужно будет продлевать.

```
sudo openssl x509 -in /etc/ipa/ca.crt -text -noout
```

Результат выполнения:

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
50:b4:1d:e8:ec:e0:27:f8:a4:ef:02:6f:57:48:1d:ea:45:08:14:a0
Signature Algorithm: sha256WithRSAEncryption
Issuer: CN = CA Signing Certificate
Validity
Not Before: Nov 16 17:38:25 2023 GMT
Not After : Nov 11 17:38:25 2043 GMT
...
```

Для упрощения отладки работы компьютеров в домене на них автоматически устанавливается и настраивается пакет **ldap-utils**, в состав которого входят такие утилиты как **ldapsearch**, **ldapadd** и др. Безопасность соединения проверяется с использованием корневого сертификата из файла */etc/ssl/certs/ca-certificates.crt*, на который указывает параметр `TLS_CACERT` из конфигурационного файла */etc/ldap/ldap.conf*.

```
sudo cat /etc/ldap/ldap.conf
```

Результат выполнения:

```
...
TLS_CACERT
/etc/ssl/certs/ca-certificates.crt
URI ldaps://dc-1.ald.company.lan
BASE dc=ald,dc=company,dc=lan
SASL_MECH GSSAPI
...
```

Файл *ca-certificates.crt* является связкой сертификатов, в которой содержится более 100 корневых сертификатов удостоверяющих центров, которым доверяет операционная система Astra Linux. Корневой сертификат домена, который ранее отображался в файле */etc/ipa/ca.crt*, обычно находится в самом конце этого файла. Чтобы просмотреть

информацию по всем сертификатам из файла, можно использовать следующую команду:

```
openssl crl2pkcs7 -nocrl -certfile /etc/ssl/certs/ca-certificates.crt |
↪openssl pkcs7 -print_certs -text -noout | less
```

5.4.3. Доступ к веб-интерфейсам и REST API контроллера домена по протоколу HTTPS

Портал управления ALD Pro и веб-интерфейс **FreeIPA** работают только по безопасному протоколу **HTTPS**, и, если обратиться к контроллеру с помощью утилиты **openssl**, то можно заметить, что веб-сервер **Apache** использует тот же самый сертификат, что и **LDAP-каталог**.

```
echo Q | openssl s_client -showcerts dc-1.ald.company.lan:443
```

Результат выполнения для сайта **dc-1.ald.company.lan**:

```
-----BEGIN CERTIFICATE-----
MIIDxjCCAq6gAwIBAgIUZ4JathCvRrV6v+Aoi4kUNScViBswDQYJKoZIhvcN
AQEL
BQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0y
MjExMTYx
NzM4MjVaFw0zMjExMTMxNzM4MjVaMCExHzAdBgNVBAMMFmRjLTEuY
WxkLmNvbXBh
...
```

Проверить сертификат для службы **389 directory server**:

```
echo Q | openssl s_client -showcerts dc-1.ald.company.lan:636
```

Результат для службы **LDAP-каталога** на **636** порту:

```
...
-----BEGIN CERTIFICATE-----
MIIDxjCCAq6gAwIBAgIUZ4JathCvRrV6v+Aoi4kUNScViBswDQYJKoZIhvcN
AQEL
BQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0y
MjExMTYx
NzM4MjVaFw0zMjExMTMxNzM4MjVaMCExHzAdBgNVBAMMFmRjLTEuY
```

(продолжение на следующей странице)

```
WxkLmNvbXBh
...
```

Для того, чтобы браузер доменного компьютера открывал эти ресурсы без предупреждений, корневой сертификат ALD Pro должен быть добавлен в хранилище доверенных корневых сертификатов. На контроллерах домена для браузера Firefox это настроено через корпоративную политику, описанную в файле */usr/lib/firefox/distribution/policies.json*. С помощью этой же политики можно разрешить прозрачную Kerberos-аутентификацию для ресурсов из корпоративного домена, для этого нужно в секции **Authentication** задать параметр **SPNEGO** (Simple and Protected GSS-API Negotiation Mechanism, простой и защищенный механизм согласования **GSS-API**).

Проверить файл командой:

```
cat /usr/lib/firefox/distribution/policies.json
```

```
{
    "policies": {
        "BlockAboutAddons": true,
        "BlockAboutConfig": true,
        "Authentication": {
            "SPNEGO": ["ald.company.lan"]
        },
        "Certificates": {
            "ImportEnterpriseRoots": true,
            "Install": ["/etc/ipa/ca.crt"]
        },
        "Homepage": {
            "URL": "https://dc-1.ald.company.lan/",
            "Locked": true,
            "StartPage": "homepage-locked"
        }
    }
}
```

Для автоматической настройки рабочих станций можно задействовать механизм групповых политик ALD Pro:

1. Создать простой дополнительный параметр компьютера «**Политика браузера**

Firefox» с уникальным идентификатором **firefox_policy**.

2. Создать внутри этого параметра следующие атрибуты:

   - «Блокировать доступ к странице addons», идентификатор **block_about_addons**
   - «Блокировать доступ к странице config», идентификатор **block_about_config**
   - «Список доменов для Kerberos аутентификации», идентификатор **authentication_spnego**
   - «Список доверенных сертификатов», идентификатор **trusted_certificates**
   - «Адрес домашней страницы», идентификатор **homepage_url**

3. Задать текст скрипта

```
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':firefox_policy' )
↪%}
{% if gpo %}
firefox_policy:
file.managed:
    - name: /usr/lib/firefox/distribution/policies.json
    - user: root
    - group: root
    - mode: 644
    - contents:
    - '{'
    - '        "policies": {'
{% if gpo['block_about_addons'] %}
    - '                "BlockAboutAddons": {{ gpo['block_about_addons'] }},'
{% endif %}{% if gpo['block_about_config'] %}
    - '                "BlockAboutConfig": {{ gpo['block_about_config'] }},'
{% endif %}{% if gpo['authentication_spnego'] %}
    - '                "Authentication": {'
    - '                        "SPNEGO": [{{ gpo['authentication_spnego'] }}
↪]'
    - '                },'
{% endif %}{% if gpo['trusted_certificates'] %}
    - '                "Certificates": {'
    - '                        "ImportEnterpriseRoots": true,'
    - '                        "Install": [{{ gpo['trusted_certificates'] }}]
↪'
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
    - '                },'
{% endif %}{% if gpo['homepage_url'] %}
    - '                "Homepage": {'
    - '                        "URL": "{{ gpo['homepage_url'] }}",'
    - '                        "Locked": true,'
    - '                        "StartPage": "homepage-locked"'
    - '                }'
{% endif %}
    - '        }'
    - '}'
{% endif %}
```

5.4.4. Доступ к удаленному рабочему столу по протоколу VNC через HTTPS

Для оказания пользователям услуги технической поддержки администратор может подключиться к удаленному рабочему столу компьютера, для этого пользователю нужно запустить приложение «Удаленный рабочий стол» и сообщить администратору случайный пароль из окна приложения. Подключение к сессии выполняется со страницы компьютера на портале управления, функция реализована на базе программного кода из проекта noVNC.

Открывая приложение удаленного рабочего стола, пользователь фактически запускает на своем компьютере упрощенный веб-сервер **websockify**, который перенаправляет запросы на порт **TCP/5900**, где их обрабатывает **x11vnc**-сервер. Веб-серверу порт назначается динамически, начиная с **TCP/6080** в зависимости от того, сколько сессий удаленного подключения было инициировано на рабочей станции. Для безопасного обмена данными **websockify** использует закрытый ключ и **SSL**-сертификат из файла */opt/rbta/aldpro/rd-gui/noVNC/novnc.pem*. Можно проверить, что это именно тот сертификат с помощью утилиты **openssl**.

Вывести сертификат для сравнения на экран:

```
cat /opt/rbta/aldpro/rd-gui/noVNC/novnc.pem
```

Результат выполнения:

```
-----BEGIN CERTIFICATE-----
MIIDazCCAlOgAwIBAgIUbKtAWP6659L+jsNqaemnAQVksoswDQYJKoZIhvc
NAQEL
BQAwRTELMAkGA1UEBhMCQVUxEzARBgNVBAgMClNvbWUtU3RhdGUxI
TAfBgNVBAoM
GEludGVybmV0IFdpZGdpdHMgUHR5IEx0ZDAeFw0yMjA2MTQxNzAxMDF
aFw0yMzA2
...
```

Сверить **SSL**-сертификат на порту **6080** утилитой **openssl**:

```
echo Q | openssl s_client -connect pc-1.ald.company.lan:6080
```

В результате выполнения команды отображается тот же сертификат:

```
-----BEGIN CERTIFICATE-----
MIIDazCCAlOgAwIBAgIUbKtAWP6659L+jsNqaemnAQVksoswDQYJKoZIhvc
NAQEL
BQAwRTELMAkGA1UEBhMCQVUxEzARBgNVBAgMClNvbWUtU3RhdGUxI
TAfBgNVBAoM
GEludGVybmV0IFdpZGdpdHMgUHR5IEx0ZDAeFw0yMjA2MTQxNzAxMDF
aFw0yMzA2
...
```

Подробности об этом сертификате можно посмотреть с помощью той же утилиты:

```
openssl x509 -in /opt/rbta/aldpro/rd-gui/noVNC/novnc.pem -text -noout
```

Результат выполнения команды это описание сертификата в удобочитаемом формате:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            6c:ab:40:58:fe:ba:e7:d2:fe:8e:c3:6a:69:e9:a7:01:05:64:b2:8b
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
...
```

Когда администратор подключается к **клиенту** с **веб-портала**, он обращается к URL вида

wss://dc-1.ald.company.lan/aldpro/rdproxy/pc-1.ald.company.lna/6080/websockify, поэтому **веб-сервер контроллера домена** проксирует запросы на указанный порт доменного компьютера. В конфигурационном файле */etc/apache2/conf-aldpro/aldpro-mp-rdproxy.conf* для параметра **SSLProxyVerify** определено значение none, поэтому принимаются любые самоподписанные сертификаты.

5.5. Работа механизмов автоматического обнаружения сервисов LDAP и KDC

5.5.1. Автоматическое обнаружение при вводе хоста в домен утилитой ipa-client-install

Ввод машины в домен осуществляется с помощью утилиты **aldpro-client-installer**, которая по цепочке вызывает **astra-freeipa-client** и **ipa-client-install**, см. табл. 6.

aldpro-client-installer >	astra-freeipa-client >	ipa-client-install
получает:`--domain ald.company.lan--account admin--password 'AstraLinux_176'--host pc-1--gui--force`	получает:`-d ald.company.lan-u admin-p AstraLinux_172-y--pa ald.company.lan"--force-join`	получает:`'domain_name': None,'principal': 'admin',` пароль не логируется, `'host_name': 'pc-1.ald.company.lan','force': False,'force_join': True, 'mkhomedir': True,` и др.
вызывает:`/usr/bin/astra-freeipa-client-d "{domain_name}"-u "{login}"-p "{password}"-y--par "--hostname=pc-1.ald.company.lan --force-join"`	устанавливает:`hostnamectl set-hostname $compname.${domain}`и далее вызывает:`ipa-client-install $username-w $userpass $par`	

Таблица 6 — Передача параметров при вводе машины в домен

Механизм автоматического обнаружения сервисов предусмотрен только в скрипте **ipa-client-install** и работает для определения следующих параметров:

- **Realm/DNS Domain**: `ALD.COMPANY.LAN/ald.company.lan` - область Керберос и FQDN домена, отличаются обычно только регистром символов. Можно сделать так, чтобы эти значения были разными, но такие настройки сложны и не имеют значительных преимуществ, поэтому не рекомендуются.
- **IPA Server**: `dc-1.ald.company.lan` - FQDN имя контроллера, через который должен быть выполнен ввод машины в домен. Обычно не указывается явно и определяется автоматически.
- **Client hostname**: `pc-1.ald.company.lan` - FQDN имя, которое должно быть у компьютера после ввода в домен, DNS-зона должна совпадать с FQDN домена. Если значения не будут совпадать, то работа хоста будет нарушена, но разработчики целенаправленно не сделали такую проверку, т.к. это дает гибкость в редких сценариях развертывания.

Если домен не указан явно через параметр `--domain`, то скрипт будет пытаться получить это значение из следующих источников:

- из параметра `--host`;
- из текущего имени компьютера `hostname` (используется имя, а не FQDN, т.е. это то значение, которое возвращает команда `hostname`, а не `hostname -f`);
- из файла *resolv.conf*, параметры `search` и `domain`.

Проверка имени домена выполняется путем извлечения **SRV**-записей для протокола **LDAP** по **TCP**. Для каждого FQDN скрипт проверяет все компоненты домена вверх по иерархии, поэтому, например, если в переменной `hostname` было значение «**pc-1.ald.company.lan**», то скрипт установки проверит следующие **SRV**-записи:

1. `_ldap._tcp.ald.company.lan`
2. `_ldap._tcp.company.lan`
3. `_ldap._tcp.lan`

Важно обратить внимание, что утилита **astra-freeipa-client** нарушает логику передачи параметров, т.к. вызывает утилиту **ipa-client-install** с значением `{'domain_name': None}`, но перед этим она устанавливает в системе имя хоста `pc-1.ald.company.lan`, поэтому ввод машины в домен отрабатывает корректно.

Сервер **IPA** можно указать явно через параметр `server`, но в том случае параметр `domain`

также должен быть задан в явном виде. Если параметр `server` не будет указан, то сервер будет найден через поиск **SRV**-записей `_ldap._tcp.DOMAIN`.

При использовании опции `server` в файлы *sssd.conf* и *krb5.conf* будет внесен фиксированный список серверов. Параметр можно указывать несколько раз подряд, чтобы задать для переменной `ipa_server` список значений.

5.5.2. Автоматическое обнаружение сервисов в SSSD

Для обеспечения надежной работы компьютера в домене в службе **SSSD** реализован механизм автоматического обнаружения контроллеров домена. В момент загрузки служба **SSSD** берет параметр `domains` и обращается к соответствующей секции файла */etc/sssd/sssd.conf* для настройки **бэкенда**:

```
[sssd]
domains = ald.company.lan

[domain/ald.company.lan]
id_provider = ipa
ipa_server = _srv_, dc-1.ald.company.lan
ipa_domain = ald.company.lan
ipa_hostname = pc-1.ald.company.lan
auth_provider = ipa
chpass_provider = ipa
access_provider = ipa
cache_credentials = True
ldap_tls_cacert = /etc/ipa/ca.crt
krb5_store_password_if_offline = True
```

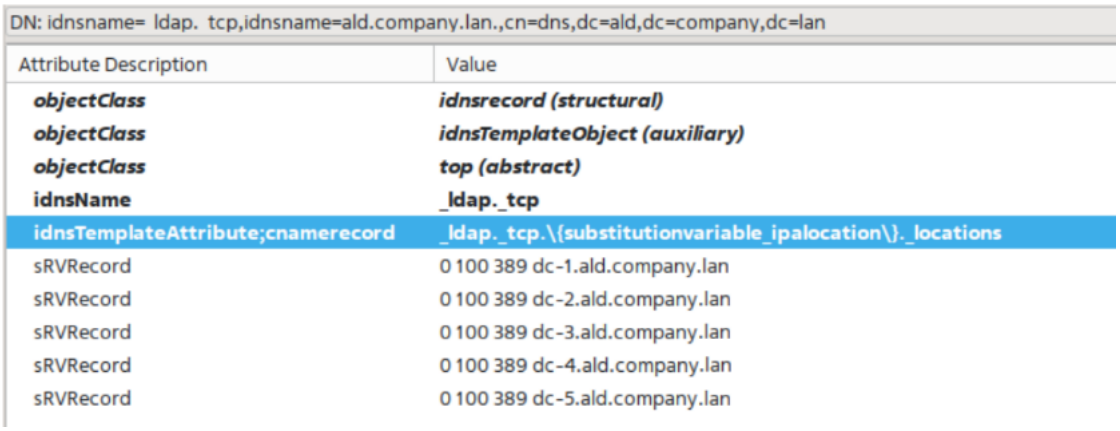
Ключевые параметры **бэкенда**:

- `id_provider` — указывает, что в качестве поставщика идентификационных данных будет выступать сервер **FreeIPA** **ipa_server** — список FQDN имен и IP адресов контроллеров домена в порядке, в каком нужно к ним подключаться. Может использоваться также служебное имя «**_srv_**», соответствующее механизму автоматического обнаружения (**service discovery**) через поиск **SRV**-записей `_ldap._tcp.DOMAIN`, как указано в стандарте (**RFC2782**).
- `ipa_backup_server` — с помощью этого параметра можно указать список резервных серверов, которые будут выбираться только в случае недоступности

основных. Этот параметр не предполагает возможности использования механизма автообнаружения.

- `ipa_domain` — указывает имя домена **IPA**. Это необязательный параметр, если он не указан, то используется доменное имя из названия конфигурации.
- `ipa_hostname` — требуется устанавливать на машинах, где имя хоста не соответствует полному имени компьютера, которое используется в домене **IPA** для идентификации этого хоста.

Записи **SRV**, создаваемые **FreeIPA** для своих сервисов, существенно отличаются от тех, которые можно создать вручную из интерфейса. Этим записям в каталоге назначен класс `idnsTemplateObject` и задано дополнительное свойство `idnsTemplateAttribute;cnamerecord`, которое определяет шаблон подстановки для поддержки технологии сайтов (**локаций**), см. *Шаблон записи для SRV-записей в DNS*.


DN: idnsname= ldap. tcp,idnsname=ald.company.lan.,cn=dns,dc=ald,dc=company,dc=lan

Attribute Description	Value
objectClass	***idnsrecord (structural)***
objectClass	***idnsTemplateObject (auxiliary)***
objectClass	***top (abstract)***
idnsName	**_ldap._tcp**
idnsTemplateAttribute;cnamerecord	**_ldap._tcp.\(substitutionvariable_ipalocation\)._locations**
sRVRecord	0 100 389 dc-1.ald.company.lan
sRVRecord	0 100 389 dc-2.ald.company.lan
sRVRecord	0 100 389 dc-3.ald.company.lan
sRVRecord	0 100 389 dc-4.ald.company.lan
sRVRecord	0 100 389 dc-5.ald.company.lan

Рисунок 5.16 – Шаблон записи для SRV-записей в DNS

Для привязки серверов к локациям у дочерних записей **DN** `cn=servers,cn=dns,dc=ald,dc=company,dc=lan` задано значение атрибута `idnsSubstitutionVariable;ipalocation`, соответствующее имени сайта, в котором этот сервер находится.

Например, в домене из 5 контроллеров при обращении к DNS серверу 10.0.1.11 (**dc-1.ald.company.lan**) **Bind** выполняет подстановку и возвращает результат для `_ldap._tcp.hq_locations.ald.company.lan`, что соответствует сайту **hq** (head quarters, головной офис), см. *Атрибут idnsSubstitutionVariable для контроллера домена dc-1*. В список будут включены все контроллеры домена, но для серверов из сайта **hq** приоритет будет равен 0, а у всех остальных 50, поэтому служба **SSSD** в первую очередь будет обращаться к контроллерам из «своего» сайта, к которому компьютер косвенно привязан через DNS-сервер, который обслуживает его запросы командой:

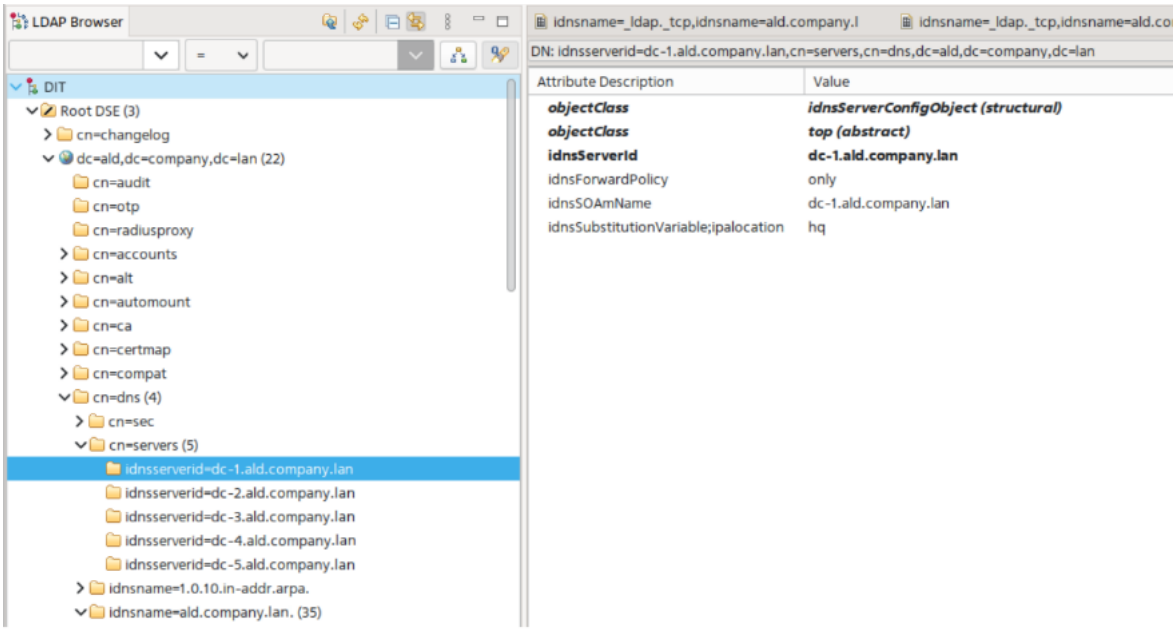


Рисунок 5.17 – Атрибут idnsSubstitutionVariable для контроллера домена dc-1

```
nslookup -q=SRV _ldap._tcp.ald.company.lan
```

Результат выполнения:

```
Server:     127.0.0.1
Address:    127.0.0.1#53

_ldap._tcp.ald.company.lan canonical name =
_ldap._tcp.hq._locations.ald.company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 50 100 389 dc-4.ald.
↪company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 0 100 389 dc-1.ald.company.
↪lan.
_ldap._tcp.hq._locations.ald.company.lan service = 0 100 389 dc-2.ald.company.
↪lan.
_ldap._tcp.hq._locations.ald.company.lan service = 50 100 389 dc-3.ald.
↪company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 50 100 389 dc-5.ald.
↪company.lan.
```

После того, как список серверов определен, **SSSD** начинает искать доступный контроллер. Сначала **SSSD** выполняет поиск по серверам с приоритетом 0 из того же сайта, и только в случае неудачи переключается на остальные. Если компьютер работает в домене ALD Pro (**FreeIPA**), то проверка выполняется по одному серверу, а если провайдером выступает AD,

то сервера проверяются группами по 5 штук, но доступность контроллера в любом случае определяется возможностью совершения анонимных LDAP-запросов, например, выдержка из журнала */var/log/dirsrv/access.log*:

```
SRCH base="" scope=0 filter="(objectClass=*)" attrs="* altServer
↪namingContexts supportedControl supportedExtension supportedFeature"
```

Если служба **SSSD** сможет получить **TGT/TGS** билеты для доступа к **LDAP**, используя */etc/krb5.keytab* файл хоста, то она выполнит еще несколько авторизованных запросов для получения дополнительной информации о диапазонах идентификаторов, участии хоста в группах и др. настройках. Это можно увидеть также из файла журнала */var/log/dirsrv/access.log*:

```
...
SRCH base="fqdn=pc-1.ald.company.lan,cn=computers,cn=accounts,dc=ald,
↪dc=company,dc=lan" scope=0 filter="(objectClass=*)" attrs="objectClass cn
↪memberOf ipaUniqueID"
...
```

Адрес сервера, с которым происходил обмен, будет сохранен как активный сервер, а FQDN остальных контроллеров, полученных через **SRV**-запрос, служба запишет, как обнаруженные сервера (Discovered IPA servers).

Информацию о результатах автоматического обнаружения, можно узнать с помощью утилиты **sssctl** из состава пакета **sssd-tools**. Данный инструмент администрирования взаимодействует со службой **SSSD** напрямую через **SBus**.

```
sudo sssctl domain-list
```

Результат выполнения список доменов:

```
ald.company.lan
```

Вывести статус по имени домена:

```
sudo sssctl domain-status ald.company.lan
```

Результат выполнения команды это список найденных контроллеров в домене:

```
Online status: Online
```

(продолжение на следующей странице)

```
Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-1.ald.company.lan
- dc-3.ald.company.lan
- dc-2.ald.company.lan
- dc-4.ald.company.lan
- dc-5.ald.company.lan
- dc-1.ald.company.lan
```

5.5.3. Автоматическое обнаружение сервисов в библиотеке libkrb5

5.5.3.1. На обычном компьютере в домене

Для администрирования компьютеров в домене системные администраторы часто используют утилиты из пакета **krb5-user**, такие как **kinit**, **klist** и др., которые обращаются к функциям библиотеки *libkrb5*.

```
sudo ldd $(which kinit) | grep libkrb5
```

Результат выполнения

```
libkrb5.so.3 => /lib/x86_64-linux-gnu/libkrb5.so.3 (0x00007fa558914000)
libkrb5support.so.0 => /lib/x86_64-linux-gnu/libkrb5support.so.0
↪(0x00007fa5588cb000)
```

Настройки этой библиотеки задаются в файле */etc/krb5.conf*, но при установке **sssd** в папку плагинов **libkrb5** копируется файл *sssd_krb5_locator_plugin.so*, поэтому, если служба **sssd** включена, то при выполнении **kinit** учитываются настройки из */etc/sssd/sssd.conf*.

```
sudo dpkg -S /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/sssd_krb5_locator_
↪plugin.so
```

Результат выполнения:

```
sssd-common: /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/sssd_krb5_locator_
↪plugin.so
```

На доменном компьютере в файле *sssd.conf* в секции `[domain/ald.company.lan]` в параметре `ipa_server` первым значением по умолчанию идет «**_srv_**», поэтому служба **sssd** выполняет поиск сервера **kdc** через обращение к **SRV**-записям `_kerberos._tcp.ald.company.lan`.

```
cat /etc/sssd/sssd.conf
```

Параметр `ipa_server` в конфигурационном *файле sssd.conf*:

```
[domain/ald.company.lan]
...
ipa_server = _srv_, dc-1.ald.company.lan
...
```

После автоматического обнаружения сервера провайдер аутентификации (auth-provider) службы **sssd** записывает IP адрес контроллера и FQDN-имена еще двух запасных серверов в файл *kdcinfo* и функции Kerberos будут использовать эти данные для подключения.

```
sudo sssctl domain-status ald.company.lan
```

Результат выполнения:

```
Online status: Online

Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-5.ald.company.lan
- dc-1.ald.company.lan
- dc-4.ald.company.lan
- dc-3.ald.company.lan
- dc-2.ald.company.lan
- dc-1.ald.company.lan
```

Проверить какой сервер используется в файле *kdcinfo* можно командой:

```
sudo cat /var/lib/sss/pubconf/kdcinfo.ALD.COMPANY.LAN
```

где `ALD.COMPANY.LAN` - домен предприятия

Результат выполнения:

```
10.0.1.12
dc-1.ald.company.lan
dc-3.ald.company.lan
```

Например, при выполнении **kinit** запрос сначала будет направляться к серверу с IP адресом 10.0.1.11, в случае его недоступности к серверу **dc-2.ald.company.lan**, далее к **dc-3.ald.company.lan** и только в случае недоступности всех трех серверов запрос завершится ошибкой.

Если в параметре `ipa_server` убрать значение «_**srv**_» и оставить только FQDN контроллера, через который этот хост был введен в домен, то после перезапуска **sssd** все Kerberos запросы пойдут через только указанный сервер.

Если службу **sssd** выключить, то контроллер будет определяться штатными механизмами библиотеки *libkrb5*. По умолчанию в файле */etc/krb5.conf*, в секции `[libdefaults]` установлен параметр `dns_lookup_kdc = true`, поэтому поиск сервера будет выполняться похожим образом через обращение к **SRV**-записям, но отличие будет состоять в том, что *libkrb5* не поддерживает кэширования, поэтому автоматическое обнаружение будет срабатывать каждый раз при обращении к функциям библиотеки.

```
sudo cat /etc/krb5.conf | grep dns_lookup_kdc
```

Результат выполнения команды:

```
dns_lookup_kdc = true
```

Файл конфигурации *krb5.conf* позволяет отключать автоматическое обнаружение, тогда с помощью параметра `kdc` можно будет указать конкретный сервер:

```
sudo cat /etc/krb5.conf
```

Настройка параметра `kdc` на конкретный сервер **dc-1**:

```
[libdefaults]
...
dns_lookup_kdc = false
...
[realms]
 ALD.COMPANY.LAN = {
  kdc = dc-1.ald.company.lan
  ...
 }
```

5.5.3.2. На контроллере домена

На контроллерах домена **kinit** ведет себя иначе, чем на доменных компьютерах, т.к. кроме службы **sssd** устанавливается еще и служба **winbind**, у которой есть собственный плагин **winbind_krb5_locator.so**.

```
ls -l /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/
```

Результат выполнения на **dc-1**:

```
-rw-r--r-- 1 root root 19248 ноя 22 2021 sssd_krb5_locator_plugin.so
-rw-r--r-- 1 root root 14840 июн 6 2022 winbind_krb5_locator.so
```

Посмотреть какому пакету принадлежит файл библиотеки можно командой:

```
sudo dpkg -S /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/winbind_krb5_
↪locator.so
```

Результат выполнения:

```
winbind: /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/winbind_krb5_locator.
↪so
```

Активный контроллер домена, предлагаемый службой **winbind** можно узнать через команду `dsgetdcname` утилиты **wbinfo**:

```
wbinfo --dsgetdcname ALD.COMPANY.LAN
```

Результат выполнения команды:

```
dc-1.ald.company.lan
\\10.0.1.11
1
f35731b8-81fe-46c3-ac8f-feb6deb55889
ald.company.lan
ald.company.lan
0xe00003fd
Default-First-Site-Name
Default-First-Site-Name
```

Включить отладку **Kerberos** для проверки работы можно включив переменную KRB5_TRACE:

```
set +o history
env KRB5_TRACE=/dev/stdout kinit admin <<< AstraLinux_176
set -o history
```

Результат выполнения трассировки:

```
[3126] 1696423198.013624: Getting initial credentials for admin@ALD.COMPANY.
↪LAN
[3126] 1696423198.013626: Sending unauthenticated request
[3126] 1696423198.013627: Sending request (191 bytes) to ALD.COMPANY.LAN
[3126] 1696423198.013628: Initiating TCP connection to stream 10.0.1.11:88
[3126] 1696423198.013629: Sending TCP request to stream 10.0.1.11:88
[3126] 1696423198.013630: Received answer (513 bytes) from stream 10.0.1.
↪11:88
[3126] 1696423198.013631: Terminating TCP connection to stream 10.0.1.11:88
[3126] 1696423198.013632: Response was from master KDC
[3126] 1696423198.013633: Received error from KDC: -1765328359/Additional pre-
↪authentication required
[3126] 1696423198.013636: Preauthenticating using KDC method data
[3126] 1696423198.013637: Processing preauth types: PA-PK-AS-REQ (16), PA-FX-
↪FAST (136), PA-ETYPE-INFO2 (19), PA-PKINIT-KX (147), PA-SPAKE (151), PA-ENC-
↪TIMESTAMP (2), PA_AS_FRESHNESS (150), PA-FX-COOKIE (133)
[3126] 1696423198.013638: Selected etype info: etype aes256-cts, salt "T1!
↪dGW5C](`}^?2R", params ""
[3126] 1696423198.013639: Received cookie: MIT1\x00\x00\x00\x01uw\x951\xbb\
↪x06ew\x88]G\xa7\x87J\xf3\x02\xcb\xaf\x97<:\x0d\xa1\xb0\x96\xb4,\xbc\xd2\xb6\
↪x0c\xce\xa4"?\xa6\xe8\xf1\xa5\xf4\xe6K\x01\x00\xe2\x02Z6\x18#M\xf8\xd4w\x17\
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
↪xcd\xc5S\xe8\x13;\xa6\x01t2\xc6M\xdf\xddHIs{#\xc9\x15\xe2\x9d\x1f\x9c\xd20.\
↪x00 gG\x9d\xfb\x0c\x80\x04'\x9f\x8bZJt-S'\x06\x97\xd8\x92;\xa5^~Ep*\x81T\
↪xff\x7f\xd0\xb7\x9b\xf6Y\xf4\xdb\xd9,\xbb\x1a\x93\xd8\xca
[3126] 1696423198.013640: PKINIT client has no configured identity; giving up
[3126] 1696423198.013641: Preauth module pkinit (147) (info) returned: 0/
↪Success
[3126] 1696423198.013642: PKINIT client received freshness token from KDC
[3126] 1696423198.013643: Preauth module pkinit (150) (info) returned: 0/
↪Success
[3126] 1696423198.013644: PKINIT client has no configured identity; giving up
[3126] 1696423198.013645: Preauth module pkinit (16) (real) returned: 22/
↪Недопустимый аргумент
[3126] 1696423198.013646: SPAKE challenge received with group 1, pubkey
↪2F4F0F1B95AF467E3A718CB0B881B2C73FA1D9D132FC15DCD0CE504BC58ED756
Password for admin@ALD.COMPANY.LAN:
[3126] 1696423198.013647: SPAKE key generated with pubkey
↪EFFC0DE0B82CEEBD2B715C87EED0EA086C5F8E9E79570593A4C302125DE253D8
[3126] 1696423198.013648: SPAKE algorithm result:
↪5C447C5BB317672F202E15DC5B54C2DB2288D87D7AE12F75A9FFB6F825A0D930
[3126] 1696423198.013649: SPAKE final transcript hash:
↪F2250C79047924FEAD1416E2B75948CCF3BABD80E9B821CA03C47A74D1D69985
[3126] 1696423198.013650: Sending SPAKE response
[3126] 1696423198.013651: Preauth module spake (151) (real) returned: 0/
↪Success
[3126] 1696423198.013652: Produced preauth for next request: PA-FX-COOKIE
↪(133), PA-SPAKE (151)
[3126] 1696423198.013653: Sending request (450 bytes) to ALD.COMPANY.LAN
[3126] 1696423198.013654: Initiating TCP connection to stream 10.0.1.11:88
[3126] 1696423198.013655: Sending TCP request to stream 10.0.1.11:88
[3126] 1696423198.013656: Received answer (1575 bytes) from stream 10.0.1.
↪11:88
[3126] 1696423198.013657: Terminating TCP connection to stream 10.0.1.11:88
[3126] 1696423198.013658: Response was from master KDC
[3126] 1696423198.013659: AS key determined by preauth: aes256-cts/297C
[3126] 1696423198.013660: Decrypted AS reply; session key is: aes256-cts/D405
[3126] 1696423198.013661: FAST negotiation: available
[3126] 1696423198.013662: Initializing KEYRING:persistent:1421600000:krb_
↪ccache_zlQRxEn with default princ admin@ALD.COMPANY.LAN
[3126] 1696423198.013663: Storing admin@ALD.COMPANY.LAN -> krbtgt/ALD.COMPANY.
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
↪LAN@ALD.COMPANY.LAN in KEYRING:persistent:1421600000:krb_ccache_zlQRxEn
[3126] 1696423198.013664: Storing config in KEYRING:persistent:1421600000:krb_
↪ccache_zlQRxEn for krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN: fast_avail: yes
[3126] 1696423198.013665: Storing admin@ALD.COMPANY.LAN -> krb5_ccache_conf_
↪data/fast_avail/krbtgt\/ALD.COMPANY.LAN\@ALD.COMPANY.LAN@X-CACHECONF: in
↪KEYRING:persistent:1421600000:krb_ccache_zlQRxEn
[3126] 1696423198.013666: Storing config in KEYRING:persistent:1421600000:krb_
↪ccache_zlQRxEn for krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN: pa_type: 151
[3126] 1696423198.013667: Storing admin@ALD.COMPANY.LAN -> krb5_ccache_conf_
↪data/pa_type/krbtgt\/ALD.COMPANY.LAN\@ALD.COMPANY.LAN@X-CACHECONF: in
↪KEYRING:persistent:1421600000:krb_ccache_zlQRxEn
```

В поставке **Samba** идет также утилита **net**, которая тоже выдает информацию о **KDC**, но ее результаты расходятся с тем, какой сервер фактически используют утилиты **kinit**:

```
sudo net ads info
```

Результат выполнения:

```
Failed to get server's current time!
LDAP server: 10.0.1.11
LDAP server name: dc-1.ald.company.lan
Realm: ALD.COMPANY.LAN
Bind Path: dc=ALD,dc=COMPANY,dc=LAN
LDAP port: 389
Server time: Чт, 01 янв 1970 03:00:00 MSK
KDC server: 10.0.1.11
Server time offset: 0
Last machine account password change: Чт, 01 янв 1970 03:00:00 MSK
```

5.5.3.3. Автоматическое обнаружение сервисов в клиенте IPA

Для автоматизации управления доменом системные администраторы используют утилиту **ipa**, которая обращается к одному из контроллеров домена через **REST API**, используя прозрачную Kerberos аутентификацию. После ее использования в связке ключей можно увидеть появление сервисного билета на доступ к HTTP конкретного контроллера.

```
klist
```

В результате в списке только **krbtgt** билет:

```
Ticket cache: KEYRING:persistent:1421600000:krb_ccache_zlQRxEn
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
04.10.2023 15:52:52  05.10.2023 15:52:47  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.
 ↪LAN
```

Получить информацию о пользователе можно через команду `ipa`:

```
ipa user-show admin
```

Результат выполнения просмотра пользователя:

```
Имя учётной записи пользователя: admin
Фамилия: Administrator
Домашний каталог: /home/admin
Оболочка входа: /bin/bash
Псевдоним учётной записи: admin@ALD.COMPANY.LAN, root@ALD.COMPANY.LAN
UID: 1421600000
ID группы: 1421600000
Учётная запись отключена: False
Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,
 ↪dc=company,dc=lan
Link to head department: ald.company.lan
Пароль: True
Участник групп: admins, lpadmin, trust admins, ald trust admin
Роли: ALDPRO - Main Administrator
Доступные ключи Kerberos: True
```

Повторно проверить список билетов можно командой `klist`:

```
Ticket cache: KEYRING:persistent:1421600000:krb_ccache_zlQRxEn
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
04.10.2023 15:54:55  05.10.2023 15:52:47  HTTP/dc-1.ald.company.lan@ALD.
```

(продолжение на следующей странице)

```
 ↪COMPANY.LAN
04.10.2023 15:52:52  05.10.2023 15:52:47  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.
 ↪LAN
```

Утилита **ipa** определяет сервер для подключения в следующем порядке:

- Сначала используется значение директивы **xmlrpc_uri** из файла */etc/ipa/default.conf*
- Если сервер, указанный в *default.conf* окажется недоступен, утилита **ipa** через запрос **SRV**-записей `_ldap._tcp.DOMAIN` получит список всех доступных серверов и будет перебирать их случайным образом.

На контроллерах домена в файле */etc/ipa/default.conf* указан также параметр `ldap_uri`, который определяет адрес для подключения **бэкенда REST API** к LDAP-каталогу. По умолчанию предполагается, что **бэкенд** подключается только к той службе каталога, которая работает на том же сервере, поэтому подключение выполняется через UNIX сокет по **ldapi** (LDAP Over Interprocess Communication facilities, IPC).

```
cat /etc/ipa/default.conf | grep ldap_uri
```

Результат выполнения:

```
ldap_uri = ldapi://%2Frun%2Fslapd-ALD-COMPANY-LAN.socket
```

5.5.3.4. Автоматическое обнаружение сервисов клиентом LDAP

Для работы компьютера в домене утилиты **OpenLDAP** (`ldapsearch`, `ldapmodify` и др.) не требуются, но пакет **ldap-utils** устанавливается и настраивается автоматически для упрощения отладки возникающих проблем. За настройку LDAP-клиента отвечает скрипт *ipaclient/install/client.py*, который устанавливает в файле */etc/ldap/ldap.conf* следующие параметры по умолчанию:

- `TLS_CACERT` — определяет путь к файлу с цепочкой **SSL** сертификатов для проверки безопасности подключения по LDAPS и при использовании команды `StartTLS`.
- `URI` — определяет протокол подключения **LDAPS** и путь к LDAP-серверу по умолчанию. Значение в файле соответствует адресу контроллера домена, через который был выполнен ввод машины в домен. Это значение не обновляется динамически, поэтому при недоступности данного сервера утилиты перестанут работать без указания другого сервера явно через параметр `-H`.

- `BASE` — определяет DN-записи, которые будут использоваться в качестве базы поиска по умолчанию.
- `SASL_MECH` — определяет механизм аутентификации, по умолчанию используется **GSSAPI**, т.е. Kerberos аутентификация через связку ключей Linux.

Для обеспечения отказоустойчивости скриптов автоматизации можно использовать, например, результат автоматического обнаружения сервисов из файла *kdcinfo*:

```
ldapsearch -H ldaps://$(tail -n 1 /var/lib/sss/pubconf/kdcinfo.ALD.COMPANY.
↪LAN) -W -D
```

5.5.3.5. Автоматическое обнаружение сервисов ALD Pro

Автоматическое определение сервисов для подключения агентов **ALDPro-Salt-Minion** и **Zabbix** выполняет скрипт */opt/rbta/aldpro/client/bin/aldpro-service-discovery.py*.

Агент **Salt-Minion** отвечает за работу групповых политик и задания автоматизации. Групповые политики работают по модели **PULL** и для работы этого механизма миньону достаточно подключиться к одному контроллеру. Задания автоматизации работают по модели **PUSH**, поэтому для возможности отправки задания автоматизации с любого контроллера **миньон** должен быть подключен ко всем контроллерам сразу. Поэтому в файле */etc/salt/minion.d/masters.conf* должны быть перечислены все контроллеры домена:

```
cat /etc/salt/minion.d/masters.conf
```

Результат выполнения:

```
master:
- dc-1.ald.company.lan
- dc-2.ald.company.lan
retry_dns: 0
retry_dns_count: 5
```

Для определения списка серверов скрипт *aldpro-service-discovery.py* запрашивает **SRV**-записи `_ldap._tcp.DOMAIN`, в качестве FQDN домена используя значение параметра `domain` секции `global` из файла */etc/ipa/default.conf*. В предыдущих реализациях в список вносились только контроллеры из того же сайта, с версии 1.4.0 скрипт вносит в начало списка контроллеры из того же сайта, а в конец добавляет все остальные.

Агент **Zabbix** отвечает за передачу данных на сервер мониторинга. Он может работать как в активном, так и в пассивном режимах. Настройки подключения к серверу находятся в файле */etc/zabbix/zabbix_agentd.conf.d/00-servers.conf*. Для определения сервера мониторинга скрипт *aldpro-service-discovery.py* выполняет запрос к **REST API** */discover/api/ds/monitoring/servers*.

Агент **Syslog-NG** отвечает за передачу событий в подсистему журналирования. Он не предусматривает поддержки автоматического обнаружения сервисов и конфигурируется через **SaltStack**, за что отвечает скрипт *syslogng_install.sls* из папки */etc/salt/states/aldpro/automations/subsystems/projects/*.

5.6. Работа компьютера в автономном режиме

Для обеспечения комфортной работы пользователей в условиях нестабильного подключения к локальной сети по Wi-Fi или через VPN служба **SSSD** может предоставлять часть сервисов автономно, без подключения к серверу. Переход в автономный режим происходит автоматически в следующих случаях:

- пропало подключение к сети;
- нет возможности преобразовать FQDN сервера в IP адрес;
- не удается подключиться к серверу.

Работая автономно, Монитор службы **SSSD** предпринимает попытки возврата в режим онлайн по следующим триггерам:

- получено уведомление об изменении настроек сети, например, о подключении кабеля или изменении таблицы маршрутизации (см. `man netlink`);
- получено уведомление об изменении конфигурационного файла */etc/resolv.conf* (см. `man inotify`);
- по расписанию каждые 30 секунд;

В ходе отладки администратор может переключать режимы вручную, отправляя процессу **SSSD** сигналы SIGUSR1 (перейти в **offline**) и SIGUSR2 (перейти в **online**), но в большинстве случаев достаточно включить/выключить всю службу целиком.

Включение **Offline** режима можно командой:

```
sudo kill -SIGUSR1 $(pidof sssd) && sudo sssctl domain-status ald.company.
↪lan | grep status
```

Результат выполнения:

```
Online status: Offline
```

Включение **Online** режима можно командой:

```
sudo kill -SIGUSR2 $(pidof sssd) && sudo sssctl domain-status ald.company.
↪lan | grep status
```

Результат выполнения:

```
Online status: Online
```

5.6.1. Автономный вход по кэшу пароля

Когда пользователь в первый раз входит в компьютер, его аутентификация возможна только через контроллер домена, но при получении успешного статуса **PAM_SUCCESS** служба **SSSD** сохранит **SHA512** хэш в локальной базе, что гарантирует возможность автономной аутентификации этого пользователя даже в условиях отсутствия связи с сервером. Извлечь хэш пользовательского пароля из ldb-файла и проверить его валидность можно с помощью утилит **ldbsearch** из пакета **ldb-tools** и **openssl**.

Получить кэшированный пароль пользователя **admin**:

```
sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb -b name=admin@ald.
↪company.lan,cn=users,cn=ald.company.lan,cn=sysdb cachedPassword
```

В результате отображается атрибут `cachedPassword`. Важно обратить внимание, что значение перенесло на другую строку, о чем говорит строка начинающаяся с символа пробел.

```
asq: Unable to register control with rootdse!
record 1
dn: name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb
cachedPassword:
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
$6$o.zJdi5M.yDDUi5j$5YHQgKLFhRzt78sSx5R8AXqLYTo9fgkgzGflHI1SYG
rshTqIGPwEv7G1WUvgd8vW3LoCSYPt88KV7SeN2N6A.0
...
```

Проверить генерацию хеша пароля можно с помощью утилиты **openssl**, где: `o.zJdi5M.yDDUi5j` — соль, которая получается из кэша пароля между символами `$`:

```
set +o history
openssl passwd -6 -salt 'o.zJdi5M.yDDUi5j' 'AstraLinux_176'
set -o history
```

Результат выполнения совпадает с кэшированными данными:

```
$6$o.zJdi5M.yDDUi5j
↪$5YHQgKLFhRzt78sSx5R8AXqLYTo9fgkgzGflHI1SYGrshTqIGPwEv7G1WUvgd8vW3LoCSYPt88KV7SeN2M
↪0
```

Функция хранения кэшей включена по умолчанию, за это отвечает параметр `cache_credentials` из файла *sssd.conf*. Срок хранения паролей не ограничен, так как параметр `offline_credentials_expiration` не задан.

Для первого входа в систему компьютер должен иметь доступ к контроллеру домена, что легко обеспечить из офиса, подключив устройство кабелем к локальной сети, но трудно выполнимо, если устройство нужно передать сотруднику для удаленной работы через службу доставки. В этом случае может оказаться, что офисная сеть будет ему недоступна до тех пор, пока он не войдет в систему и не подключится через VPN. Для решения этой проблемы в составе **sssd-tools** есть утилита **sss_seed**, которая позволяет имитировать первый вход, принудительно устанавливая ldb-кэш пароля.

Установить кэш пароля для тестовой учетной записи **olegovo**:

```
set +o history
echo 'AstraLinux_176' > /tmp/ssd-pwd.txt
sudo sss_seed --domain ALD.COMPANY.LAN --username olegovo --password-file /
↪tmp/ssd-pwd.txt
sudo rm /tmp/ssd-pwd.txt
set -o history
```

Результат выполнения команды:

```
Temporary password added to cache entry for olegovo@ald.company.lan
```

Проверить результат кэша командой:

```
sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb -b
↪name=olegovo@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb
↪cachedPassword
```

Результат выполнения команды:

```
...
dn: name=olegovo@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb
↪cachedPassword:
$6$0Xa5CIrB1WRICTmM$eJ0bVItqEQeSfx6/B3.bodGHdn//uMuAYk.9fCnvcL.7iGdka/
↪Uhrn5TrV4HxKznikXeXc82LQcT/d1CYvWig.
...
```

Имя пользователя должно быть известно в домене, иначе команда не сможет получить его идентификатор и другие сведения. Если на момент выполнения команды пользователь уже входил в это устройство, то утилита переопределит кэш его пароля.

5.6.2. Автоматическая Kerberos аутентификация при переходе в онлайн режим

Хэш **SHA512** отлично подходит для долгосрочного хранения секретов на диске, но с его помощью не получится пройти Kerberos-аутентификацию, поэтому при входе в компьютер в автономном режиме служба **SSSD** помещает пароль в связку ключей ядра Linux, чтобы иметь возможность выписать **TGT**-билет, когда связь с сервером будет восстановлена. За возможность автоматической Kerberos-аутентификации пользователя при переключении в онлайн режим отвечает параметр **krb5_store_password_if_offline**, который по умолчанию равен **True**.

Если потребуется, извлечь пароль из **Keyring**, можно с помощью утилиты **keyctl**, но из-за настроек прав доступа выполнять эти команды нужно будет в контексте процесса службы **SSSD**. Подключиться к действующему процессу можно с помощью отладчика **gdb** (GNU Debugger), для работы которого нужно будет отключить блокировку функции **ptrace**. Продемонстрируем данную технику, для этого сначала установим отладчик и отключим блокировку **ptrace**.

```
sudo apt install gdb
sudo astra-ptrace-lock disable
sudo reboot
```

После перезагрузки выполняется вход в систему тем же пользователем и смотрится pid службы **sssd** для подключения к процессу **Монитора** и номер текущего окна терминала для перенаправления стандартного вывода. Сетевая связанность компьютера **pc-1** с **контроллером домена** в этот момент должна быть заблокирована, т.к. после получения **TGT**-билета пароль больше не требуется и будет удален из связки ключей.

Найти идентификатор процесса можно командой:

```
pidof sssd
```

Результат выполнения потребуется для подключения к процессу **sssd**:

```
487
```

Вычисляется устройство текущего терминала для вывода:

```
tty
```

Результат вывода:

```
/dev/pts/0
```

Теперь осталось только подключиться к процессу, чтобы извлечь содержимое ключа.

```
sudo gdb -p 487
```

В результате подключиться к процессу через отладчик **gdb**:

```
GNU gdb (AstraLinuxSE 8.2.1-2) 8.2.1
...
(gdb) _
```

Для выхода из отладчика ввести команду q.

Выполнить команду gdb для получения связки ключей:

```
call system("keyctl show > /dev/pts/0")
```

Результат выполнения, где важен идентификатор ключа 835659210:

```
Session Keyring
 84684334 --alswrv      0     0  keyring: _ses
835659210 --alswrv      0     0   \_ user: admin@ald.company.lan
...
```

Выполнить запрос данных по идентификатору 835659210 на терминал */dev/pts/0*:

```
call system("keyctl print 835659210 > /dev/pts/0")
```

Результат выполнения:

```
[Detaching after fork from child process 7834]
AstraLinux_172
$2 = 0
```

Технология автоматической Kerberos аутентификации при переходе в онлайн режим является безопасной, потому что:

- пароль хранится в оперативной памяти и не сохраняется в системе при выключении компьютера;
- пароль хранится в системе ограниченное время и автоматически удаляется при появлении связи с контроллером домена;
- извлечь пароль может только пользователь, обладающий привилегиями суперпользователя.

5.7. Динамическое обновление DNS-записей в домене

Компьютеры в домене взаимодействуют между собой по DNS-именам, поэтому в каталоге важно иметь актуальную информацию по используемым IP адресам. С серверами обычно не возникает проблем, т.к. им назначаются статические адреса, а вот интерфейсы рабочих станций настраиваются динамически, поэтому администраторам нужно обеспечить такое же динамическое обновление этой информации в каталоге.

Обновление DNS-записей возможно как по инициативе DHCP-сервера (Remote Name

Daemon Control, **RNDC**), так и по инициативе рабочей станции (**GSS-TSIG**). У каждого из подходов есть свои сильные и слабые стороны, но продуктовой команде ALD Pro второй способ видится наиболее простым и безопасным, так как в этом случае запросы на обновление DNS-записей авторизуются с помощью Kerberos-аутентификации.

Для настройки динамического обновления DNS-записей в конфигурационном файле */etc/sssd/sssd.conf* на стороне рабочей станции нужно добавить следующие параметры:

```
[domain/ald.company.lan]
...
dyndns_update = true
dyndns_refresh_interval = 43200
dyndns_update_ptr = true
dyndns_ttl = 3600
...
```

После внесения указанных изменений можно перезапустить службу **sssd** и проверить обновление записей в каталоге при изменении IP-адреса на хосте.

5.8. Отладка работы службы SSSD

Служба **SSSD** предоставляет две основные функции: получение информации о пользователях и их аутентификацию. Каждая из этих функций использует свои программные интерфейсы, поэтому должна рассматриваться отдельно. Однако успешная аутентификация возможна, только если информация о пользователе получена успешно, поэтому, если что-то не работает, то в первую очередь нужно убедиться, что возможно получить хотя бы информацию о пользователе с помощью команд `getent` и `id`:

```
getent passwd $USER
```

Результат выполнения:

```
admin:*:1421600000:1421600000:Administrator:/home/admin:/bin/bash
```

Выполнить команду `id`:

```
id $USER
```

Результат выполнения:

```
uid=959800000(admin) gid=959800000(admins) группы=959800000(admins),
↪1001(astra-admin),113(lpadmin)
```

5.8.1. Журналы отладки SSSD

Служба **SSSD** состоит из множества компонентов и для настройки каждого из них в конфигурационном файле предназначена отдельная секция (например, за настройки монитора отвечает секция `[sssd]`, а **Бэкенд** настраивается в секции `[domain/...]`), поэтому для включения отладки конкретного компонента в его секции следует задать параметр `debug_level=N`, где **N** - это число от 1 до 10.

Уровни отладки 1-3 регистрируют сбои, а уровни 8-10 обеспечивают излишнюю детализацию, которая затрудняет быстрый анализ журналов, поэтому для решения большинства проблем начать лучше с шестого уровня. Открыть конфигурацию командой:

```
sudo cat /etc/sssd/sssd.conf
```

Результат выполнения:

```
[domain/ald.company.lan]
debug_level = 10
id_provider = ipa
...
[sssd]
debug_level = 10
services = ifp
domains = ald.company.lan
[nss]
debug_level = 10
homedir_substring = /home
...
```

Уровень отладки можно менять не только через конфигурационный файл, но и «на лету» без перезагрузки службы **SSSD**, отправляя ей команды через **SBus** с помощью утилиты **sssctl** из пакета **sssd-tools**. Очистить логи, чтобы удобно было смотреть отладочную информацию:

```
sudo sssctl logs-remove
```

Изменить уровень отладки командой и просмотреть в лог файле изменение `grep` „Debug level changed“:

```
sudo sssctl debug-level 6
sudo cat /var/log/sssd/sssd.log | grep 'Debug level changed'
```

Результат выполнения отображает начало нужного события:

```
(2023-08-20 22:17:00): [sssd] [server_common_rotate_logs] (0x0010): Debug
↪level changed to 0x07f0
```

Повысить уровень логирования до 8:

```
sssctl debug-level 8
cat /var/log/sssd/sssd.log | grep 'Debug level changed'
```

В результате можно увидеть два события начало и конец. Это нужно для удобного просмотра лога, чтобы знать где искать интервал отладочной информации от и до:

```
...[sssd] [server_common_rotate_logs] (0x0010): Debug level changed to 0x07f0
...[sssd] [server_common_rotate_logs] (0x0010): Debug level changed to 0x37f0
```

Выбрать все строки между кодами смены уровня отладки **0x07f0** и **0x37f0** можно с помощью `regex` запроса:

```
cat /var/log/sssd/sssd.log | grep -zoP '(?<=0x07f0)(?s).*(?=0x37f0)'
```

В операционной системе Astra Linux журналы отладки хранятся в папке */var/log/sssd*, по одному файлу журнала на каждый компонент. **Ответчики** пишут в файлы с именами *SSSD_$service*, например, собеседник **NSS** пишет в файл */var/log/sssd/sssd_nss.log*. Сообщения от **Бэкенда** пишутся в файл с именем *sssd_$domainname.log*. Есть еще журналы вспомогательных процессов, таких как *ldap_child.log u krb5_child.log*.

Прежде чем погружаться в изучение журналов и файлов конфигурации, желательно внимательно ознакомиться с описанием архитектуры службы **SSSD**.

Во многих случаях полезно воспроизвести ошибку после очистки кэша, чтобы исключить его влияние, но важно помнить, что в локальной базе кэшируются также и учетные данные,

поэтому не стоит удалять кэш, если работа идет в автономном режиме. А также важно обратить внимание, что приведенные примеры сообщений отладки могут меняться для последующих релизов.

5.8.2. Общие рекомендации

Если команды `getent` или `id` не выводят вообще никакой информации о пользователе или группе, то нужно проверить:

- Работу службы разрешения имен командой `ping dc-1` и, указаны ли в */etc/resolv.conf* правильные DNS-сервера.
- Работает ли служба с помощью команды `systemctl status sssd`.
- Конфигурацию */etc/nsswitch.conf*, что модуль **sss** указан для следующих баз данных: **passwd**, **group**, **shadow**, **services**, **netgroup**, **sudoers**, **automount**.
- Поведение команды `id` на контроллере домена ALD Pro.
- Доходит ли запрос до **Ответчика**. Для этого в разделе `[nss]` установите `debug_level = 6` и перезапустите службу **sssd**.

Приведем пример успешного запроса информации по пользователю.

```
[sssd[nss]] [get_client_cred] (0x4000): Client creds: euid[10327]
↪egid[10327] pid[18144].
[sssd[nss]] [setup_client_idle_timer] (0x4000): Idle timer re-set for client
↪[0x13c9960][22]
[sssd[nss]] [accept_fd_handler] (0x0400): Client connected!
[sssd[nss]] [sss_cmd_get_version] (0x0200): Received client version [1].
[sssd[nss]] [sss_cmd_get_version] (0x0200): Offered version [1].
[sssd[nss]] [nss_getby_name] (0x0400): Input name: admin
```

Если команда достигает **Ответчика NSS**, передается ли она **Поставщику** данных (Data Provider). Неудачный запрос может выглядеть так:

```
[sssd[nss]] [cache_req_search_dp] (0x0400): CR #3: Looking up [admin@ipa.
↪test] in data provider
[sssd[nss]] [sss_dp_issue_request] (0x0400): Issuing request for
↪[0x41e51c:1:admin@ipa.test@ipa.test]
[sssd[nss]] [sss_dp_get_account_msg] (0x0400): Creating request for [ipa.
↪test][0x1][BE_REQ_USER][name=admin@ipa.test:-]
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
[sssd[nss]] [sss_dp_internal_get_send] (0x0400): Entering request
↪[0x41e51c:1:admin@ipa.test@ipa.test]
[sssd[nss]] [sss_dp_req_destructor] (0x0400): Deleting request:
↪[0x41e51c:1:admin@win.trust.test@win.trust.test]
[sssd[nss]] [sss_dp_get_reply] (0x0010): The Data Provider returned an error
↪[org.freedesktop.sssd.Error.DataProvider.Offline]
[sssd[nss]] [cache_req_common_dp_recv] (0x0040): CR #3: Data Provider Error:
↪3, 5, Failed to get reply from Data Provider
[sssd[nss]] [cache_req_common_dp_recv] (0x0400): CR #3: Due to an error we
↪will return cached data
```

Успешно обработанный запрос напротив будет выглядеть так:

```
[sssd[nss]] [cache_req_search_dp] (0x0400): CR #3: Looking up [admin@ipa.
↪test] in data provider
[sssd[nss]] [sss_dp_issue_request] (0x0400): Issuing request for
↪[0x41e51c:1:admin@ipa.test@ipa.test]
[sssd[nss]] [sss_dp_get_account_msg] (0x0400): Creating request for [ipa.
↪test][0x1][BE_REQ_USER][name=admin@ipa.test:-]
[sssd[nss]] [sss_dp_get_reply] (0x1000): Got reply from Data Provider - DP
↪error code: 0 errno: 0 error message: Success
[sssd[nss]] [cache_req_search_cache] (0x0400): CR #3: Looking up [admin@ipa.
↪test] in cache
```

Если запрос к **Поставщику** данных успешно завершен, но по-прежнему не отображаются результаты, следует переходить к журналам **Бэкенда**.

5.8.3. Отладка Бэкенда

Бэкенд выполняет несколько различных операций, поэтому бывает трудно сразу понять, в чем заключается проблема. На самом высоком уровне **Бэкенд** выполняет следующие шаги в указанном порядке:

1. **Бэкенд** получает запрос от клиентской библиотеки и решает, к какому серверу следует подключиться для его обработки. Этот шаг может включать в себя поиск по **SRV**-записям.

2. **Бэкенд** устанавливает соединение с сервером и проходит аутентификацию, используя учетные данные из keytab-файла хоста, если это требуется.

3. Как только соединение установлено, **Бэкенд** отправляет запрос на поиск информации, поэтому вы должны увидеть в запросе критерии фильтрации, базовую запись и запрашиваемые атрибуты.

4. После того, как поиск завершится, соответствующие записи будут сохранены в кэше. Код состояния возвращается **собеседнику**. Установите `debug_level=6` в разделе `[domain/]`, перезапустите службу и выполните еще раз неудачный поиск информации о пользователе. При отладке работы **Бэкенда** в первую очередь убедитесь, что **Бэкенд** находится в режиме **онлайн**, сделать это можно с помощью утилиты **sssctl**.

```
sudo sssctl domain-status ald.company.lan
```

Результат выполнения:

```
Online status: Online

Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-1.ald.company.lan
```

Проверить, можно ли установить соединение с теми же параметрами безопасности, которые использует **SSSD**:

```
kinit -k && klist
```

Результат выполнения:

```
Ticket cache: KEYRING:persistent:959800000:krb_ccache_pMxi5Bu
Default principal: host/dc-1.ald.company.lan@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
20.08.2023 23:15:25  21.08.2023 23:15:25  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.
 ↪LAN
```

Проверить запрос в **LDAP**, где Ключи `-H` и `-ZZ` делают запрос ближе к тому, как служба **SSSD** взаимодействует с каталогом:

```
ldapsearch -ZZ -H ldap://dc-1.ald.company.lan -b uid=admin,cn=users,
↪cn=accounts,dc=ald,dc=company,dc=lan cn
```

Результат выполнения:

```
# admin, users, accounts, ald.company.lan
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
cn: Administrator
```

Для отладки **GSSAPI** аутентификации команду `kinit` можно использовать в сочетании с переменной окружения `KRB5_TRACE`. Если установлен для **Бэкенда** десятый уровень, то информация о трассировке Kerberos аутентификации будет отражаться также в журнале *ldap_child.log*.

```
set +o history
env KRB5_TRACE=/dev/stdout kinit <<< AstraLinux_176
set -o history
```

Результат выполнения:

```
[23001] 1696515238.426608: Getting initial credentials for admin@ALD.COMPANY.
↪LAN
[23001] 1696515238.426610: Sending unauthenticated request
[23001] 1696515238.426611: Sending request (191 bytes) to ALD.COMPANY.LAN
[23001] 1696515238.426612: Initiating TCP connection to stream 10.0.1.11:88
[23001] 1696515238.426613: Sending TCP request to stream 10.0.1.11:88
[23001] 1696515238.426614: Received answer (514 bytes) from stream 10.0.1.
↪11:88
[23001] 1696515238.426615: Terminating TCP connection to stream 10.0.1.11:88
[23001] 1696515238.426616: Response was from master KDC
...
[23001] 1696515238.426651: Storing admin@ALD.COMPANY.LAN -> krb5_ccache_conf_
↪data/pa_type/krbtgt\/ALD.COMPANY.LAN\@ALD.COMPANY.LAN@X-CACHECONF: in
↪KEYRING:persistent:1421600000:krb_ccache_kFlzpn6
```

Необходимо проверить, присутствуют ли на сервере все атрибуты, необходимые для службы **SSSD**, правильно ли указана база поиска, особенно для доверенных поддоменов. По возможности, выполнить такой же поиск вручную утилитой **ldapsearch**.

5.8.4. Устранение неисправностей аутентификации, изменения пароля и контроля доступа

Если информация о пользователе может быть успешно получена, но аутентификация не проходит, то в первую очередь нужно смотреть журнал */var/log/auth.log* на предмет сообщений от **pam_sss**. Аутентификация начинается в **PAM**-стеке на фазе **auth** и выполняется с помощью провайдера аутентификации (auth_provider) службы **SSSD**. Однако, не все запросы на аутентификацию проходят через **SSSD**, например, аутентификация по **SSH** происходит непосредственно в **SSHD**, а **SSSD** взаимодействует с **PAM** стеком только на фазе **account**. Контроль доступа начинается в **PAM** стеке на фазе **account** и выполняется с помощью провайдера доступа (access_provider) службы **SSSD**.

Смена пароля начинается на стороне **PAM** стека на фазе **password** и выполняется с помощью провайдера смены пароля (chpass_provider) службы **SSSD**.

Аутентификация через **PAM** стек происходит по следующему шаблону:

1. Приложение с поддержкой **PAM** аутентификации начинает диалог. В зависимости от настроек **PAM**-стека в диалог может быть вовлечен модуль **pam_sss**. Для отладки процесса аутентификации, сначала нужно проверить журнал */var/log/auth.log* на предмет того, есть ли вообще обращения к **pam_sss**. Если нет упоминания **pam_sss**, скорее всего, **PAM** стек настроен неправильно. Если обращения к **pam_sss есть**, нужно включить отладку для **Ответчика PAM**.

2. **Ответчик PAM** службы **SSSD** получает запрос на аутентификацию и в большинстве случаев пересылает его **Бэкенду**. Важно обратить внимание, что в отличие от запросов на идентификацию, запросы на аутентификацию и контроль доступа обычно не кэшируется и всегда завершаются обращением к **серверу**. В некоторых случаях это может приводить к задержкам, но это достаточно важно, потому что дополнительные группы в GNU/Linux устанавливаются только в момент входа в систему.

Журналы **собеседника PAM** должны показывать запросы, которые пришли от **PAM**-стека и были перенаправлены на **Бэкенд**. Если отображается запрос на аутентификацию к **собеседнику PAM**, но получается ошибка от **Бэкенда**, важно проверить журналы **Бэкенда**.

Пример ошибки может выглядеть как:

```
[sssd[pam]] [sss_dp_issue_request] (0x0400): Issuing request for
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
↪[0x411d44:3:admin@ipa.example.com]
[sssd[pam]] [sss_dp_get_account_msg] (0x0400): Creating request for [ipa.
↪example.com][3][1][name=admin]
[sssd[pam]] [sss_dp_internal_get_send] (0x0400): Entering request
↪[0x411d44:3:admin@ipa.example.com]
[sssd[pam]] [sss_dp_get_reply] (0x1000): Got reply from Data Provider - DP
↪error code: 1 errno: 11 error message: Offline
[sssd[pam]] [pam_check_user_dp_callback] (0x0040): Unable to get information
↪from Data Provider Error: 1, 11, Offline
```

3. **Бэкенд** обрабатывает запрос. Это может включать эквивалент выполнения команды `kinit` в процессе **krb5_child**, аутентификацию по **LDAP** или проверку по списку контроля доступа. После того, как запрос к **Бэкенду** завершится, результат отправляется обратно **собеседнику PAM**. Важно посмотреть также файл *krb5_child.log*, если установить уровень отладки `debug_level = 10`, то в этот файл будет включаться также низкоуровневая информация для трассировки работы протокола Kerberos. Также можно имитировать аутентификацию вручную с помощью утилиты **kinit**.

4. **Ответчик PAM** получает результат и пересылает его обратно в модуль **pam_sss**. Сообщения об ошибке или статусе отображается в */var/log/auth.log*.

5.9. Расположение значимых файлов

В качестве сводной информации по значимым компонентам продукта ALD Pro в табл. 7 представлен список файлов для отладки системы, в которых хранятся конфигурации компонентов и журналы работы этих компонентов.

Компонент системы	Конфигурация	Файл журнала
SSSD	/etc/sssd/sssd.conf	/var/log/sssd/
PAM	/etc/pam.d/system-auth	/var/log/sssd/sssd_pam.log
NSS	/etc/nsswitch.conf	/var/log/sssd/sssd_nss.log
krb5_child	/etc/sssd/sssd.conf	/var/log/sssd/krb5_child.log
ldap_child	/etc/sssd/sssd.conf	/var/log/sssd/ldap_child.log
Data Provider (Backend)	/etc/sssd/sssd.conf	/var/log/sssd/sssd_.log
DNS	/etc/resolv.conf	journalctl -u bind9-pkcs11.service
Samba	/etc/samba/smb.conf	/var/log/samba/*.log
Winbind	/etc/security/pam_winbind.conf	/var/log/samba/log.*
Kerberos	/etc/krb5.conf	/var/log/sssd/krb5_child.log
salt-master	/etc/salt/	/var/log/salt/master
aldpro-salt-minion	/etc/aldpro-salt/	/var/log/aldpro-salt/minion

Таблица 7 — Расположение значимых файлов

Полезные инструкции

6.1. Инструкция по созданию дополнительных параметров групповых политик

6.1.1. Термины и определения

Групповые политики позволяют снизить стоимость управления ИТ-инфраструктурой за счет автоматического применения одинаковых настроек на больших группах компьютеров, имеющих общее целевое назначение.

Каждая политика (в терминах MS объект групповой политики) представляет из себя именованный набор параметров, в соответствии с которыми производится автоматическая настройка операционной системы и прикладного программного обеспечения. Для настройки групповой политики нужно выполнить следующие действия:

- создать новую групповую политику
- добавить конфигурационный параметр в политику (включить его)
- установить желаемые значения атрибутов для параметра
- назначить политику на подразделение, внутри которого есть целевые компьютеры или пользователи
- подождать от 30 минут до 80 минут или изменить таймер для получения и применения политики

Администраторам доступны сотни параметров «из коробки», которые соответствуют настройкам панели управления ОС Astra Linux. Но для решения частных задач конкретного бизнеса может потребоваться разработка дополнительных параметров силами команды внедрения, и в ALD Pro для этого есть все необходимые инструменты.

6.1.2. Как работают групповые политики

Для получения данных групповых политик и их применения используется pull-модель. Инициатором работы является миньон SaltStack (standalone minion), установленный на

каждом компьютере домена, который по протоколу LDAP к серверу службы каталогов получает актуальные данные о групповой политике.

- Миньон Standalone — это рабочие станции или серверы, на котором работает демон-миньон Salt, обладающий широкой функциональностью, работающий автономно от мастера. Используется для запуска заданий в системе без подключения к мастеру.

Важно: Получение данных групповых политик и применение политик с помощью standalone миньона реализовано с версии ALD Pro 2.2.0, ранее для этих задач использовался стандартный миньон SaltStack, который получал данные от мастера.

6.1.3. Как создать дополнительный параметр

Таблица 3. Привилегии для работы с дополнительными параметрами ГП

Привилегия	Описание	На что распространяется
Computer Group Policy Additional Parameters Catalog - Read	Привилегия позволяет: Видеть список параметров компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров); Видеть список папок компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров); Видеть карточки параметров компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Параметр компьютеров>); Видеть карточки папок параметров компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Папка параметров компьютеров>)	На весь домен
User Group Policy Additional Parameters Catalog - Read	Привилегия позволяет: Видеть список параметров пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры пользователей); Видеть карточки параметров пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры пользователей/<Параметр пользователей>)	На весь домен
Computer Group Policy Additional Parameters - Manage	Привилегия позволяет: Создавать новые папки параметров компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров); Изменять папки параметров компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Папка параметров компьютеров>); Удалять папки параметров компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Папка параметров компьютеров>); Создавать новые параметры компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров); Изменять параметры компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Параметр компьютеров>); Удалять параметры компьютеров (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Параметр компьютеров>)	На весь домен
User Group Policy Additional Parameters - Manage	Привилегия позволяет: Создавать новые папки параметров пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры пользователей); Изменять папки параметров пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры пользователей/<Папка параметров пользователей>); Удалять папки параметров пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры пользователей/<Папка параметров пользователей>); Создавать новые параметры пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры пользователей); Изменять параметры пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Параметр пользователей>); Удалять параметры пользователей (Управление доменом/Доп. параметры групповых политик/вкладка Параметры компьютеров/<Параметр пользователей>)	На весь домен

Параметр групповой политики (в терминах MS шаблон групповой политики) подобен классу из теории объектно-ориентированного программирования. Он определяет модель для создания объектов конфигурирования, описывая их свойства (атрибуты) и методы для работы с этими данными (скрипт). Продолжая аналогию из ООП, «объектом» в данном случае будет являться экземпляр параметра, добавленный в конкретную групповую

политику.

Параметр может быть «простым» или «составным»:

- В случае **простого** параметра свойства представляют из себя плоский список атрибутов. Например, для настройки межсетевого экрана может потребоваться задать уровень детализации журнала, и эта настройка подразумевает одно конкретное значение, поэтому она может быть реализована атрибутом «простого» параметра.
- **Составные** параметры позволяют задавать массив списка атрибутов, где в каждом списке представлен однотипный набор данных. Если продолжать пример с межсетевым экраном, то для настройки фильтрации может потребоваться разрешить входящие ssh-соединения, запретить исходящие smtp и т.п., поэтому такие настройки нужно реализовывать атрибутами «составного» параметра.

Параметр может быть «параметром компьютера» или «параметром пользователя». «Параметры компьютеров» отвечают за настройку оборудования и служб вне зависимости от того, какой пользователь работает с системой. «Параметры пользователей» отвечают за настройку рабочей среды пользователя вне зависимости от того, на каком компьютере он работает. В обоих случаях скрипты выполняются с возможностями по администрированию root-пользователя.

Чтобы создать новый параметр групповой политики нужно перейти на страницу **Управление доменом - Доп. параметры групповых политик** и выбрать одну из вкладок:

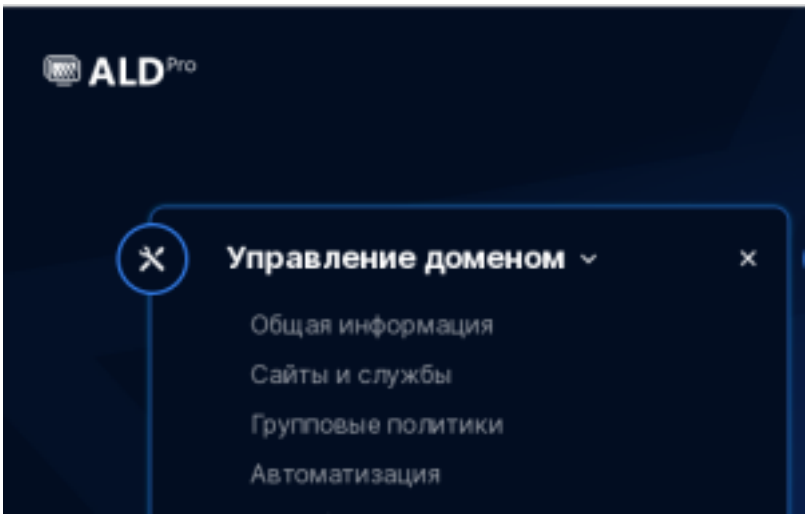


Рисунок 6.1 – Как создать дополнительный параметр 1

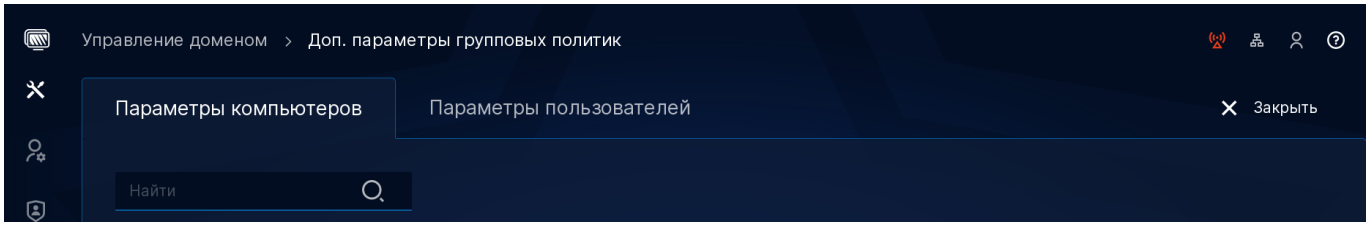


Рисунок 6.2 – Как создать дополнительный параметр 2

На каждой вкладке по умолчанию будет представлен корневой каталог «Каталог дополнительных параметров», внутри которого вы можете создавать свои пользовательские параметры. Если параметров станет слишком много, вы сможете группировать их по подразделам.

Выберите раздел «Каталог дополнительных параметров», нажмите кнопку «+ Новый параметр» и заполните карточку следующими значениями:

Раздел «Основные»

- Название параметра: «Диспетчер задач htop». Это название, которое будет отображаться в каталоге при редактировании групповых политик на портале управления
- Уникальный идентификатор: «software_htop». Это название, которое будет использоваться в качестве имени `\*.sls` файла на диске и в скриптах SaltStack. Формируется по следующей формуле: `rbta_ldap_custom_gp_host_` + введенные пользователем данные «software_htop». Для пользователей `rbta_ldap_custom_gp_user_`.
- Тип каталога: «Параметр компьютерной групповой политики». Определяет, относится ли данный параметр к настройкам компьютера или пользователя. Параметр не доступен для редактирования, его значение определяется автоматически в зависимости от того, с какой страницы вы перешли к созданию дополнительного параметра.
- Тип параметра: «простой». Определяет вид параметра. **Простой** - простой список атрибутов, **Составной** - массив списка атрибутов.
- Папка параметра: «Дополнительные параметры». Указывает раздел каталога, в котором будет доступен данный параметр.
- Назначение параметра: «Управление установкой и настройкой диспетчера задач htop». Текстовое описание, позволяющее указать наиболее важную информацию о работе параметра для удобства последующего использования. Данный текст отображается при редактировании групповой политики по нажатию кнопки вопроса.

Здесь же можно внести данные о требованиях к ОС (см. *Планирование ресурсов*).

- Разделы **Атрибуты параметра** и **Конфигурация скрипта** станут доступны при редактировании сохраненного параметра.

6.1.3.1. Особенности редактирования дополнительных параметров ГП

Для редактирования дополнительного параметра групповых политик нужно перейти на страницу **Управление доменом Доп. параметры групповых политик Параметры компьютеров или Параметры пользователей “Имя параметра”** и внести нужные изменения. После сохранить.

Чтобы эти параметры синхронизировались с групповыми политиками и применились на компьютерах и пользователях нужно перейти на карточку этих политик **Групповые политики Групповые политики “Имя политики”**, внести изменения и сохранить. Изменения в групповой политике приведут к повышению ее версии. Только после этого изменения дополнительных параметров ГП будут применены на компьютерах и для пользователей. Или принудительно применить политику (подробнее читайте в *Отладка*).

Аналогичные действия необходимо применить если дополнительный параметр групповой политики был удален.

6.1.4. Как настроить атрибуты дополнительного параметра

Атрибуты — это свойства, которые можно задавать при добавлении параметра к групповой политике. Например, если мы создадим параметр для управления диспетчером задач `htop` на рабочих станциях, то нам понадобится атрибут «Должен быть установлен», чтобы в зависимости от его значения устанавливать или удалять указанное программное обеспечение.

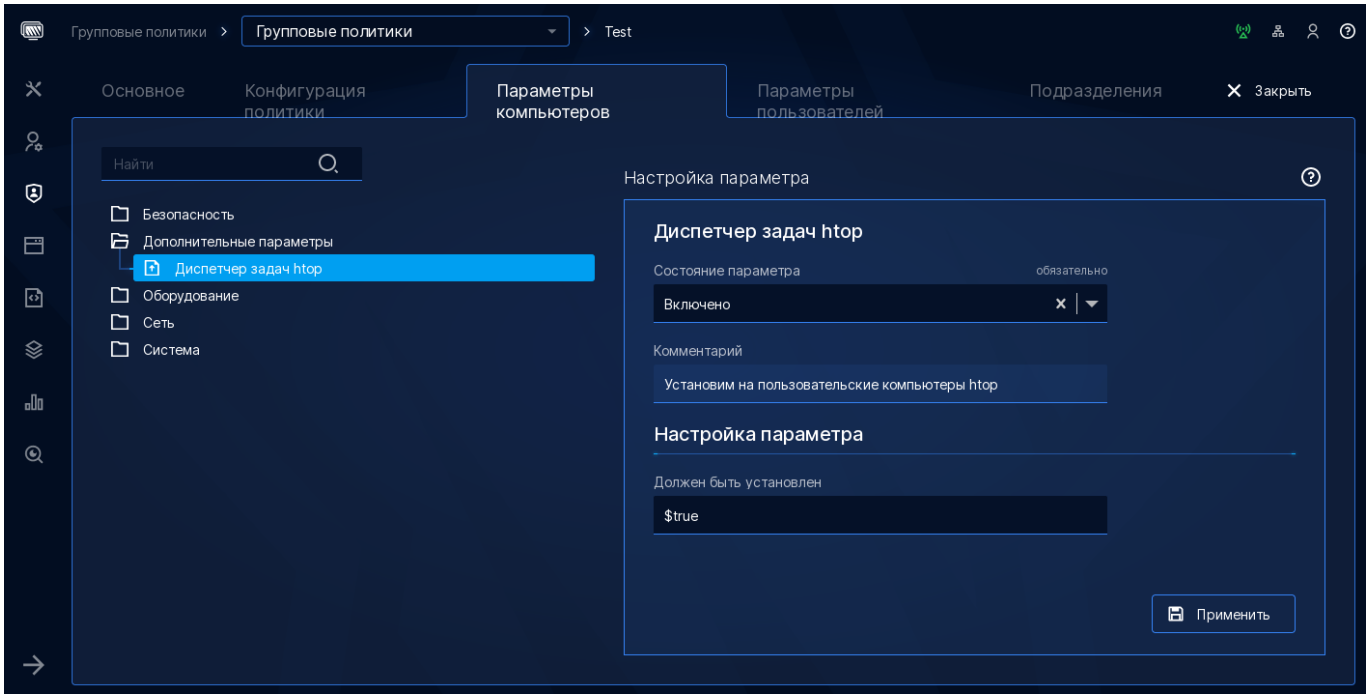


Рисунок 6.3 – Как настроить атрибуты дополнительного параметра

Атрибуты представляют из себя обычные текстовые строки, значение которых может быть интерпретировано в скриптах в том числе как числа, логические значения ($true и $false), списки и т.п. Добавьте атрибут:

- Название атрибута: «Должен быть установлен»

Название, под которым будут видеть этот атрибут на карточке параметра.

- Уникальный идентификатор: «must_be_installed»

Идентификатор атрибута, который будет использоваться как имя переменной в SaltStack скриптах.

- Описание: «атрибут используется как логическая переменная true / false для ветвления шаблона»

Краткие комментарии для инженера, которые помогут упростить поддержку этого атрибута в будущем, когда потребуется внести какие-либо изменения. Данная информация недоступна при настройке групповой политики, поэтому, если вы хотите дать подсказку администратору о возможных значениях этого атрибута, внесите эту информацию в описание параметра.

- Обязательный: `true`

Пользователь должен указать значение атрибута

- Уникальный: `true`

Применяется для составных параметров, у которых возможны конфликты. Для разрешения конфликтов для каждого параметра нужно определить уникальный атрибут. В рамках одного ГПО нельзя будет создать массив списка атрибутов с одинаковыми значениями уникального параметра.

6.1.5. Скрипт дополнительного параметра

Скрипты SaltStack похожи на PHP, в них текст перемежается императивными инструкциями языка программирования, по результату выполнения которых получается итоговый документ. За императивную часть отвечают инструкции шаблонизатора Jinja2, а декларативная часть задается по правилам SaltStack в YAML-подобном синтаксисе.

6.1.5.1. Императивные инструкции шаблонизатора Jinja

Инструкции языка Jinja внедряются с помощью фигурных скобок:

Jinja	Аналог PHP	Комментарий
`{% … %}`	`<? … ?>`	Синтаксис для вставки инструкции языка шаблонизатора Jinja
`{% print(…) %}`	`<? echo (…) ?>`	Пример инструкции для вывода в документ по месту вызова значения выражения
`{{ … }}`	`<?= … ?>`	Краткий синтаксис для вывода в документ по месту вызова значения выражения
`{### … #}`	`<?/* … */?>`	Синтаксис для вставки комментариев средствами языка Jinja2

Для повышения читабельности кода управляющие структуры обычно оформляют отступами, но yaml-документы крайне чувствительны к пробелам, поэтому данный инструмент форматирования оказывается недоступен без применения специальных операторов подавления отступов. Чтобы убрать лишние пробелы и пустые строки вам нужно всего лишь поставить знак дефиса слева, справа или с обоих концов управляющего блока:

Синтаксис	Функция
`{%-`	Удаляет пробелы и пустые строки слева
`-%}`	Удаляет перенос текущей строки и переносит ее на предыдущую
`{%-` и `-%}`	Удаляет пробелы и пустые строки слева, в т.ч. перенос текущей строки, поэтому текст окажется в конце предыдущей строки

> **Внимание:** При сохранении скрипта может появится предупреждение: «Ошибка конфигурации скрипта. Игнорировать ошибку и сохранить изменения?». Встроенный механизм проверяет соответствие указного скрипта формальным правилам, поэтому если есть уверенность, что скрипт корректный необходимо выбирать «Да» для сохранения.

Выполнение команд

Создайте файл `hello.sls` с простейшим Jinja-скриптом, чтобы проверить, как это работает, и выполните его с помощью команды `aldpro-salt-call` с указанными параметрами:

```
echo '{% do salt.log.info("Hello world!") %}' > hello.sls
aldpro-salt-call --local --file-root=. state.sls hello
```

Несколько комментариев:

- Оператор «do» вызывает метод `salt.log.info` с параметром „Hello world!“
- Команда `aldpro-salt-call` отвечает за локальное исполнение скриптов на миньонах.
- Параметр `local` исключает обращение к мастеру для получения настроек.
- Параметр `file-root` со значением «.» устанавливает в качестве рабочего каталога текущую директорию.
- Параметр `state.sls` указывает на то, что следует использовать метод sls модуля `salt.modules.state`, который применяет состояния, описанные в одном и более файлов на диске из рабочей директории.
- Параметр `hello` задает имя файла `hello.sls`, которое должно быть указано без

расширения файла.

Еще один пример. С помощью метода run модуля cmd можно выполнить на удаленном хосте любую команду bash:

```
echo '{% do salt.cmd.run("apt-get install htop") %}' > installhtop.sls
aldpro-salt-call --local --file-root=. state.sls installhtop
```

Полный перечень всех модулей SaltStack можно найти на странице документации https://docs.saltproject.io/en/latest/ref/modules/all/index.html.

Работа с переменными

Переменные на языке Jinja задаются оператором `set`. Вам доступен широкий перечень типов данных: булевые, числовые, текстовые, списки, кортежи, словари:

```
{% set boolean_val = True %}
{% set int_val = 777 %}
{% set string_val = 'hello' %}
{% set list_val = ['h', 'e', 'l', 'l', 'o'] %}
{% set tuple_val = ('h', 'e', 'l', 'l', 'o') %}
{% set dict_val = { b: True, i: 777, s: 'hello') %}
```

При работе с числовыми переменными доступны обычные математические операторы:

```
{% set a = 5 %}
{% set b = 10 %}
{% set a = a + b %}
{% set b = a - b %}
{% set a = a - b %}
```

Конкатенация строк возможна как специальным оператором «~», который предварительно конвертирует все значения в строки, так и обычным оператором сложения «+»:

```
{% set a = 5 %}
{% set b = 10 %}
{% set c = a ~ b %}
{% set d = 'World' %}
{% set e = 'Hello, ' + d %}
```

Для выполнения более сложных преобразований над переменными доступно множество функций, которые можно применять как методы через точку или в стиле конвейеров bash через символ вертикальной черты, за что их так же называют фильтрами:

```
{% set a = 'Hello'.upper() %}
{% set b = 'world' | upper %}
```

Полный перечень встроенных фильтров Jinja можно найти в документации https://jinja.palletsprojects.com/en/2.11.x/templates/#list-of-builtin-filters.

Ветвления и циклы

Если выполнение набора команд должно зависеть от некоторого условия, то ветвление можно задать с помощью операторов if/elseif/else/endif:

```
{% if ram >= 2048 %}
...
{% elseif ram <= 4096 %}
...
{% else %}
...
{% endif %}
```

Циклы for являются аналогом конструкций типа foreach других языков программирования и предназначены для выполнения заданного набора инструкций применительно к каждому элементу из списка:

```
{% set users = ['ivanov', 'petrov', 'kuznetsov'] %}
{% for user in users %}
{% do salt.log.info(user.upper()) %}
{% endfor %}
```

Если нужно обработать данные словаря, используется метод items(), чтобы получить список его элементов:

```
{% set users = {'u1':'ivanov', 'u2':'petrov', 'u3':'kuznetsov'} %}
{% for id, user in users.items() %}
{% do salt.log.info(user.upper()) %}
{% endfor %}
```

Так как циклы Jinja работают только со списками, то для эмуляции обычных циклов со счетчиком вам нужно воспользоваться функцией range([min], max). Если передать этой функции один параметр, то будет сгенерирован список с указанным количеством элементов, нумерация которых будет начинаться с нуля. Если передать два числа, то нумерация элементов будет в указанном диапазоне:

```
{% do salt.log.info('Обратный отсчет') %}
{% for i in range(0, 10) %}
{% do salt.log.info(10 - i) %}
{% endfor %}
```

Информация о целевой системе (grains)

При написании скриптов Jinja можно опираться на информацию о целевом хосте, на котором этот скрипт будет применен. Данная информация доступна в словаре grains, тут находится информация об операционной системе, памяти, дисках, настройках сетевых интерфейсов и многом другом. Например, используя ключ nodename можно получить имя хоста:

```
{% set node = salt['grains.get']('nodename') %}
```

Актуальное содержание словаря `grains` для конкретного хоста можно посмотреть с миньона утилитой `aldpro-salt-call`:

```
aldpro-salt-call grains.items
local:
     ----------
     astra_version:
     ----------
    distr:
        orel
    version:
        1.7.2
    ...
```

Более подробную информацию о модулях grains можно найти в документации по проекту: https://docs.saltproject.io/en/latest/ref/modules/all/salt.modules.grains.html

Передача информации с мастера (pillar)

При настройке групповых политик администраторы задают значения атрибутов, которые доменным компьютерам передаются через механизм пилларов (`salt pillar`, соляной столб). Актуальное содержание словаря `pillar` можно посмотреть с миньона утилитой `aldpro-salt-call`.

Для компьютеров:

```
aldpro-salt-call pillar.get aldpro-hosts:pc-1.ald.company.lan
```

Для пользователей:

```
aldpro-salt-call pillar.get aldpro-users:<логин пользователя>
```

Содержание словаря определяется тем, какие политики назначены конкретному компьютеру (условный аналог GPResult в MS AD). Данные пиллара доступны только тем миньонам, для кого они предназначены, поэтому через атрибуты можно передавать в том числе конфиденциальную информацию, например пароли служебных учетных записей, сертификаты.

Если же нужно проверить, была ли уже применена конкретная политика на хосте, можно применить состояние с параметром test=True. Если по результатам выполнения команды появится сообщение “No changes needed to be made”, значит система уже получила pillar с заданной групповой политикой:

```
aldpro-salt-call state.apply rbta_ldap_env_vars_h test=True
...
         ID: /etc/bash.bashrc
    Function: file.blockreplace
      Result: True
     Comment: No changes needed to be made
...
```

Более подробную информацию о модулях pillar можно найти в документации по проекту:https://docs.saltproject.io/en/latest/topics/tutorials/pillar.html.

Создание файлов на стороне миньона

Есть методы для создания файлов `touch`, переименования rename, удаления `clean`. Подробное описание возможностей доступно на странице:

- https://docs.saltproject.io/en/latest/ref/states/all/salt.states.file.html
- https://www.8host.com/blog/osnovy-saltstack-bazovye-terminy-i-ponyatiya/
- https://serverspace.ru/support/help/ispolzovanie-platformy-salt/

6.1.5.2. Декларативные описания SaltStack

Декларативная часть скрипта описывает в YAML-формате желаемую конфигурацию компьютера. В структуре документа указано, какие методы нужно вызывать для конфигурирования системы, и с какими параметрами. В качестве примера создадим файл `software.sls` со скриптом, который описывает состояние, после применения которого в системе должен стать доступен диспетчер задач `htop`:

```
cat software.sls
software_htop: ### Имя состояния
··pkg.installed: ### Вызов метода installed модуля pkg
·····pkgs: ### Аргументы метода installed
·······-·htop ### Значение аргумента pkgs
```

Можно выполнить скрипт с помощью команды aldpro-salt-call:

```
aldpro-salt-call --local --file-root=. state.sls software
```

Подробное рассмотрение параметров команды:

- `local` — означает локальную обработку без обращения к мастеру
- `file-root=.` — устанавливает рабочую директорию, в которой будет выполняться поиск *.sls файлов
- `state.sls` — указывает, что следует использовать модуль sls, который отвечает за применение состояний из файлов на диске
- `software` — указывает имя *.sls файла без расширения

Язык `YAML`, также как и `JSON`, является языком разметки текста для сериализации данных. Каждая строка задает пару ключ-значение, между которыми стоит знак двоеточия. Ключи

могут состоять из одного и более слов, причем, заключать их в кавычки необязательно. В качестве значений поддерживаются как скалярные типы данных (`int, float, boolean, string`), так и вложенные словари, что позволяет создавать древовидные структуры данных. Второй и последующий уровни дерева должны обозначаться отступами с помощью двух и более пробелов, использовать символы табуляции вместо пробелов недопустимо. Если требуется прокомментировать какой-то фрагмент документа, то слева от комментария нужно поставить символ решетки.

На верхнем уровне структуры данных должен быть указан ключ с именем состояния, например, `software_htop`. Имя выбирается произвольно, в одном документе может быть описано несколько состояний, но их имена не должны повторяться.

На второй строке указано, что требуется вызвать метод `installed` модуля `salt.states.pkg`. Данный модуль является модулем состояния, т. е. его методы приводят систему к требуемому состоянию, описывая желаемый конечный результат, а не то, как к нему прийти. Есть также модули исполнения, которые выполняют в системе конкретные действия, ранее был приведен пример с использованием метода run модуля cmd для выполнения `bash` команды.

На третьей и четвертой строке методу передается аргумент `pkgs` со значением `htop`. Важно обратить внимание, что аргумент и его значение являются элементами списков, поэтому в начале этих строк стоит символ дефиса.

6.1.6. Требования к скриптам и наследование параметров

С 2.2.0 архитектура ALD Pro в части работы групповых политик изменилась, скрипты групповых политик получают только те компьютеры и пользователи, которым назначена политика. В связи с этим в скрипте может не быть проверки на назначение параметра конкретному компьютеру или пользователю.

Если на один компьютер или пользователя назначено несколько политик, использующих один и тот же параметр, то значения простых параметров будут переопределяться, а значения составных параметров суммироваться (см. «Наследования и суммирование групповых политик»).

Миньон получает политики, назначенные на компьютер или авторизованного на компьютере пользователя. ALD Pro анализирует назначенные параметры, снизу вверх по дереву подразделений, оставляя только те которые соответствуют требованиям к операционной системе. Из оставшихся в результате операций суммирования и

наследования формирует единый словарь `pillar`. Скрипты групповых политик отрабатываются миньоном только один раз с итоговым содержанием словаря `pillar`, вне зависимости от того, сколько раз соответствующий параметр был использован в объектах групповых политик, назначенных на этот компьютер или пользователя.

6.1.7. Отладка

Скрипт дополнительного параметра сохраняется в сервере службы каталогов (`{корень} → cn=etc → cn=aldpro → cn=vcsStorage → cn=grouppolicy → cn={Идентификатор ГП}`) и при получении по протоколу LDAP кешируется на миньоне в папках вместе с остальными групповыми политиками:

- Параметры компьютеров: `/srv/aldpro-salt/roots/states/policies/host-policies/`
- Параметры пользователей: `/srv/aldpro-salt/roots/states/policies/user-policies/`

В соответствии с расписанием из файла `/srv/aldpro-salt/roots/_modules/utils.py` скрипты выполняются при запуске службы `aldpro-salt-minion` и по расписанию каждые 30 минут + от 5 до 50 минут случайного времени. Другими словами политики гарантировано применятся в течение 80 минут. Посмотреть текущее значение таймера можно командой:

```
aldpro-salt-call schedule.show_next_fire_time build_gp_pillars
```

При необходимости можно локально изменить время стационарного таймера (по умолчанию 30 минут):

1. В файле на клиентском компьютере: `/srv/aldpro-salt/config/minion.d/standalone_scheduler.con` изменить значение атрибута `seconds/minutes` в блоке интересующего задания. Для групповых политик это `build_gp_pillars`.

2. В файле `/srv/aldpro-salt/_modules/utils.py` заменить `_25_MINUTES = dt.timedelta(minutes=25)`. Выставлять значения таймера менее 10 минут нежелательно, это может привести к повышению нагрузки и на контроллер домена и на доменный компьютер.

3. Перезапускаем миньон командой:

```
systemctl restart aldpro-salt-minion.service
```

Другой способ применения групповых политик: можно на миньоне вручную ввести команду (аналог `gpupdate /force`):

```
aldpro-salt-call state.apply gpupdate.gp
```

Можно в конце команды добавить `pillar='{“force”: True}'` - принудительное удаление связанного с политикой кэша. `pillar='{“verbose”: True}'` нужен для получения логов выполнения заданий применения. Использовать рекомендуется осторожно, поскольку вызывает повышенную нагрузку на контроллер домена.

Применить отдельно политики пользователя можно командой:

```
aldpro-salt-call state.apply policies.user-policies  pillar="{'user':
↪'username'}"
```

Для работы с политиками пользователей нужно явно передавать имя пользователя, чьи настройки вы хотите применить `pillar=“{'user': 'username'}”`.

6.1.7.1. Версионность

Каждая политика ГП имеет версию, которая автоматически изменяется, если наступило одно из следующих событий:

1. Внесены любые изменения в политики ГП или связанный с ней дополнительный параметр ГП. В этом случае дата изменений - это дата, когда пользователь внес изменения. А автор - это ФИО пользователя, внесшего изменения.

2. Произошло обновление версии ALD Pro. В этом случае дата изменений - это дата обновления системы. В этом случае «автором» изменений будет указано Системное обновление.

6.2. Инструкция по резервному копированию подсистем ALD Pro

6.2.1. Резервное копирование (далее - бэкапирование)

- Раздел состоит из перечня скриптов резервного копирования каждой подсистемы (сервера);
- Требования к последовательности запуска скриптов - отсутствуют;
- Допустимо выборочное резервное копирование подсистем (серверов).

6.2.1.1. Резервное копирование Контроллера домена

Для выполнения резервного копирования Контроллера Домена необходимо перейти на рабочую станцию, которая выполняет роль Контроллера Домена и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Создание резервной копии FreeIPA
ipa-backup
# Архивирование PK FreeIPA
tar -zcvf $BACKUP_PATH/ipa.tar.gz /var/lib/ipa/backup
# Очистка промежуточного бэкапа FreeIPA
rm -rf /var/lib/ipa/backup/*
# Остановка затрагиваемых бэкапом сервисов
systemctl stop apache2 celery celerybeat rabbitmq-server postgresql aldpro-
↪salt-minion
# Архивирование БД PostgreSQL
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
# Архивирование RabbitMQ
tar -zcvf $BACKUP_PATH/rabbitmq.tar.gz /var/lib/rabbitmq/mnesia/
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# Архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# Архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
↪log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
# Архивирование директории rbta
tar -zcvf $BACKUP_PATH/rbta.tar.gz /opt/rbta/
# Запуск затрагиваемых бэкапом сервисов
systemctl start apache2 celery celerybeat rabbitmq-server postgresql aldpro-
↪salt-minion
```

Реплики - это равноправные серверы, не требующие отдельных условий для резервного копирования. Для их копирования используются команды, аналогичные командам для резервного копирования копирования Контроллера домена

6.2.1.2. Резервное копирование подсистемы журналирования событий

Для выполнения резервного копирования подсистемы журналирования событий необходимо перейти на рабочую станцию, которая выполняет функцию сервера журналирования событий и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
↪log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
# Архивирование директории ipa-client
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
```

6.2.1.3. Резервное копирование подсистемы печати

Для выполнения резервного копирования подсистемы печати необходимо перейти на рабочую станцию, которая выполняет функцию сервера печати и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Остановка затрагиваемых бэкапом сервисов
systemctl stop cups aldpro-salt-minion
# архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
↪log/
# архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
# Запуск затрагиваемых бэкапом сервисов
systemctl start cups aldpro-salt-minion
```

6.2.1.4. Резервное копирование подсистемы DHCP

Для выполнения резервного копирования подсистемы DHCP необходимо перейти на рабочую станцию, которая выполняет функцию сервера DHCP и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# Архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
↪log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
```

6.2.1.5. Резервное копирование подсистемы мониторинга

Для выполнения резервного копирования подсистемы мониторинга необходимо перейти на рабочую станцию, которая выполняет функцию сервера мониторинга и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Остановка затрагиваемых бэкапом сервисов
systemctl stop apache2 zabbix-agent zabbix-server postgresql aldpro-salt-
↪minion
# Архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# Архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
↪log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# Архивирование zabbix
tar -zcvf $BACKUP_PATH/zabbix.tar.gz /usr/share/zabbix/
# Архивирование БД PostgreSQL
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
# Запуск затрагиваемых бэкапом сервисов
systemctl start apache2 zabbix-agent zabbix-server postgresql aldpro-salt-
↪minion
```

6.2.1.6. Резервное копирование подсистемы установки ОС по сети

Для выполнения резервного копирования подсистемы установки ОС по сети необходимо перейти на рабочую станцию, которая выполняет функцию сервера установки ОС по сети и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Остановка затрагиваемых бэкапом сервисов
systemctl stop apache2 postgresql aldpro-salt-minion
# Архивирование PostgreSQL
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
# Архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
↪log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
# Архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# Архивирование директории tftp
tar -zcvf $BACKUP_PATH/tftp.tar.gz /var/www/tftp/
# Запуск затрагиваемых бэкапом сервисов
systemctl start apache2 postgresql aldpro-salt-minion
```

6.2.1.7. Резервное копирование подсистемы репозиториев ПО

Для выполнения резервного копирования подсистемы репозиториев ПО необходимо перейти на рабочую станцию, которая выполняет функцию сервера репозиториев ПО по сети и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Остановка затрагиваемых бэкапом сервисов
systemctl stop apache2 postgresql rabbitmq-server aldpro-salt-minion
# Архивирование БД PostgreSQL
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
# Архивирование данных брокера очередей RabbitMQ
tar -zcvf $BACKUP_PATH/rabbitmq.tar.gz /var/lib/rabbitmq/mnesia/
# Архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
 ↪log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
# Архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# Архивирование директории repo/storage
tar -zcvf $BACKUP_PATH/storage.tar.gz /opt/rbta/aldpro/repo/storage/
# Запуск бэкапируемых сервисов
systemctl start apache2 postgresql rabbitmq-server aldpro-salt-minion
```

6.2.1.8. Резервное копирование подсистемы общего доступа

Для выполнения резервного копирования подсистемы общего доступа необходимо перейти на рабочую станцию, которая выполняет функцию сервера общего доступа по сети и выполнить скрипт:

```
#!/usr/bin/env bash
# Название директории для резервной копии формируется
# из текущей даты и находится в папке /tmp/backup
now=$(date +"%d-%m-%Y")
BACKUP_PATH=/tmp/backup/$now
# Создание временной директории для резервных копий
mkdir -p $BACKUP_PATH
# Ограничение прав доступа к каталогу с резервными копиями
chown root:root $BACKUP_PATH && chmod 700 $BACKUP_PATH
# Остановка затрагиваемых бэкапом сервисов
systemctl stop aldpro-salt-minion
# Архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/
↪log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
# Архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# Архивирование директории /opt/samba_shares/
tar -zcvf $BACKUP_PATH/samba.tar.gz /opt/samba_shares/
# Запуск затрагиваемых бэкапом сервисов
systemctl start aldpro-salt-minion
```

6.2.2. Раздел 2. Восстановление

- Раздел состоит из перечня скриптов восстановления каждой подсистемы (сервера);
- Требования к последовательности запуска скриптов восстановления - отсутствуют;
- Допустимо выборочное восстановление подсистем (серверов).

6.2.2.1. Восстановление Контроллера Домена

Для восстановления Контроллера Домена необходимо перейти на рабочую станцию, которая выполняет функцию Контроллера домена, далее перейти в директорию, в которой хранятся архивы резервного копирования: `cd /tmp/backup` (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования):

```
#!/bin/bash
# Перейти в директорию с резервными копиями
cd /tmp/backup/
# Разархивирование РК IPA
tar -C "/" -xvf ipa.tar.gz
# Восстановление IPA из РК
ipa-restore /var/lib/ipa/backup/ipa-full-YOUR_BACKUP_DATE
# Остановка затрагиваемых восстановлением сервисов
systemctl stop apache2 celery celerybeat rabbitmq-server postgresql aldpro-
↪salt-minion
# Восстановление БД Postgresql
tar -C "/" -xvf postgresql.tar.gz
# Восстановление RabbitMQ
tar -C "/" -xvf rabbitmq.tar.gz
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Восстановление директории etc
tar -C "/" -xvf etc.tar.gz
# Восстановление директории rbta
tar -C "/" -xvf rbta.tar.gz
# Восстановление ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Перезагрузка
reboot
```

6.2.2.2. Восстановление подсистемы журналирования событий

Для восстановления подсистемы журналирования необходимо перейти на рабочую станцию, которая выполняет функцию сервера журналирования событий, далее перейти в директорию, в которой хранятся архивы резервного копирования: cd /tmp/backup (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования):

```
#!/bin/bash
# Переход в директорию с резервными копиями
cd /tmp/backup/
# Остановка затрагиваемых восстановлением сервисов
systemctl stop syslog-ng aldpro-salt-minion
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Восстановление директории etc
tar -C "/" -xvf etc.tar.gz
# Восстановление ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Перезагрузка
reboot
```

6.2.2.3. Восстановление подсистемы печати

Для восстановления подсистемы печати необходимо перейти на рабочую станцию, которая выполняет функцию сервера печати, далее перейти в директорию, в которой хранятся архивы резервного копирования: cd /tmp/backup (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования):

```
#!/bin/bash
# Переход в директорию с резервными копиями
cd /tmp/backup/
# Остановка затрагиваемых бэкапом сервисов
systemctl stop cups aldpro-salt-minion
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Восстановление директории etc
tar -C "/" -xvf etc.tar.gz
# Восстановление ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Перезагрузка
reboot
```

Внимание: При восстановлении указанным способом информация в LDAP и на сервере может отличаться. Чтобы избежать этого, после восстановления системы необходимо выполнить команду: `aldpro-roles --subsystem_settings cups`

6.2.2.4. Восстановление подсистемы DHCP

Для восстановления подсистемы DHCP необходимо перейти на рабочую станцию, которая выполняет функцию сервера DHCP, далее перейти в директорию, в которой хранятся архивы резервного копирования: `cd /tmp/backup` (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования) :

```
#!/bin/bash
# Перейти в директорию с резервными копиями
cd /tmp/backup/
# Остановка затрагиваемых восстановлением сервисов
systemctl stop isc-dhcp-server aldpro-salt-minion
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Восстановление директории  etc
tar -C "/" -xvf etc.tar.gz
# Восстановление  ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Перезагрузка
reboot
```

Внимание: При восстановлении указанным способом информация в LDAP и на сервере может отличаться. Чтобы избежать этого, после восстановления системы необходимо выполнить команду: `aldpro-roles --subsystem_settings dhcp`

6.2.2.5. Восстановление подсистемы мониторинга

Для восстановления подсистемы мониторинга необходимо перейти на рабочую станцию, которая выполняет функцию сервера мониторинга, далее перейти в директорию, в которой хранятся архивы резервного копирования: `cd /tmp/backup` (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования):

Пароль для восстановления базы данных - `postgres`

```
#!/bin/bash
# Перейти в директорию с резервными копиями
cd /tmp/backup/
# Остановка затрагиваемых бэкапом сервисов
systemctl stop apache2 zabbix-agent zabbix-server postgresql aldpro-salt-
↪minion
# Восстановление БД PostgreSQL
tar -C "/" -xvf postgresql.tar.gz
# Восстановление ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Архивирование директории etc
tar -C "/" -xvf etc.tar.gz
# Восстановление директории zabbix
tar -C "/" -xvf zabbix.tar.gz
# Перезагрузка
reboot
```

6.2.2.6. Восстановление подсистемы установки ОС по сети

Для восстановления подсистемы установки ОС по сети необходимо перейти на рабочую станцию, которая выполняет функцию сервера установки ОС по сети, далее перейти в директорию, в которой хранятся архивы резервного копирования: `cd /tmp/backup` (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования):

```
#!/bin/bash
# Переход в директорию с резервными копиями
cd /tmp/backup
# Остановка затрагиваемых восстановлением сервисов
systemctl stop apache2 tftpd-hpa postgresql aldpro-salt-minion
# Восстановление БД PostgreSQL
tar -C "/" -xvf postgresql.tar.gz
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Восстановление директории etc
tar -C "/" -xvf etc.tar.gz
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# Восстановление ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Восстановление директории tftp
tar -C "/" -xvf tftp.tar.gz
# Перезагрузка
reboot
```

Внимание: При восстановлении указанным способом информация в LDAP и на сервере может отличаться. Чтобы избежать этого, после восстановления системы необходимо выполнить команду: `aldpro-roles --subsystem_settings os`

6.2.2.7. Восстановление подсистемы репозиториев ПО

Для восстановления подсистемы репозиториев ПО необходимо перейти на рабочую станцию, которая выполняет функцию сервера репозиториев ПО, далее перейти в директорию, в которой хранятся архивы резервного копирования: `cd /tmp/backup` (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования):

```
#!/bin/bash
# Переход в директорию с резервными копиями
cd /tmp/backup
# Остановка затрагиваемых восстановлением сервисов
systemctl stop apache2 postgresql rabbitmq-server aldpro-salt-minion
# Восстановление БД PostgreSQL
tar -C "/" -xvf postgresql.tar.gz
# Восстановление RabbitMQ
tar -C "/" -xvf rabbitmq.tar.gz
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Восстановление директории etc
tar -C "/" -xvf etc.tar.gz
# Восстановление ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Восстановление директории с репозиториями
tar -C "/" -xvf storage.tar.gz
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# Перезагрузка
reboot
```

6.2.2.8. Восстановление подсистемы общего доступа к файлам

Для восстановления подсистемы общего доступа к файлам необходимо перейти на рабочую станцию, которая выполняет функцию сервера общего доступа к файлам, далее перейти в директорию, в которой хранятся архивы резервного копирования: `cd /tmp/backup` (директория может иметь другое наименование в зависимости от настроек пути сохранения файлов резервного копирования):

```
#!/bin/bash
# Переход в директорию с резервными копиями
cd /tmp/backup
# Остановка затрагиваемых восстановлением сервисов
systemctl stop smbd nmbd aldpro-salt-minion
# Восстановление логов
tar -C "/" -xvf log.tar.gz
# Архивирование директории etc
tar -C "/" -xvf etc.tar.gz
# Восстановление ipa-client
tar -C "/" -xvf ipa-client.tar.gz
# Восстановление общих директорий samba
tar -C "/" -xvf samba.tar.gz
# Перезагрузка
reboot
```

Внимание: При восстановлении указанным способом информация в LDAP и на сервере может отличаться. Чтобы избежать этого, после восстановления системы необходимо выполнить команду:

```
aldpro-roles --subsystem_settings smb
```

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

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).

6.3.1. Что такое HBAC-правила

Правила HBAC 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.

6.3.2. Механизм работы HBAC-правил

Службами в контексте HBAC 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 и

sudo) 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 HBAC-правил отвечает модуль `pam_sss.so`:

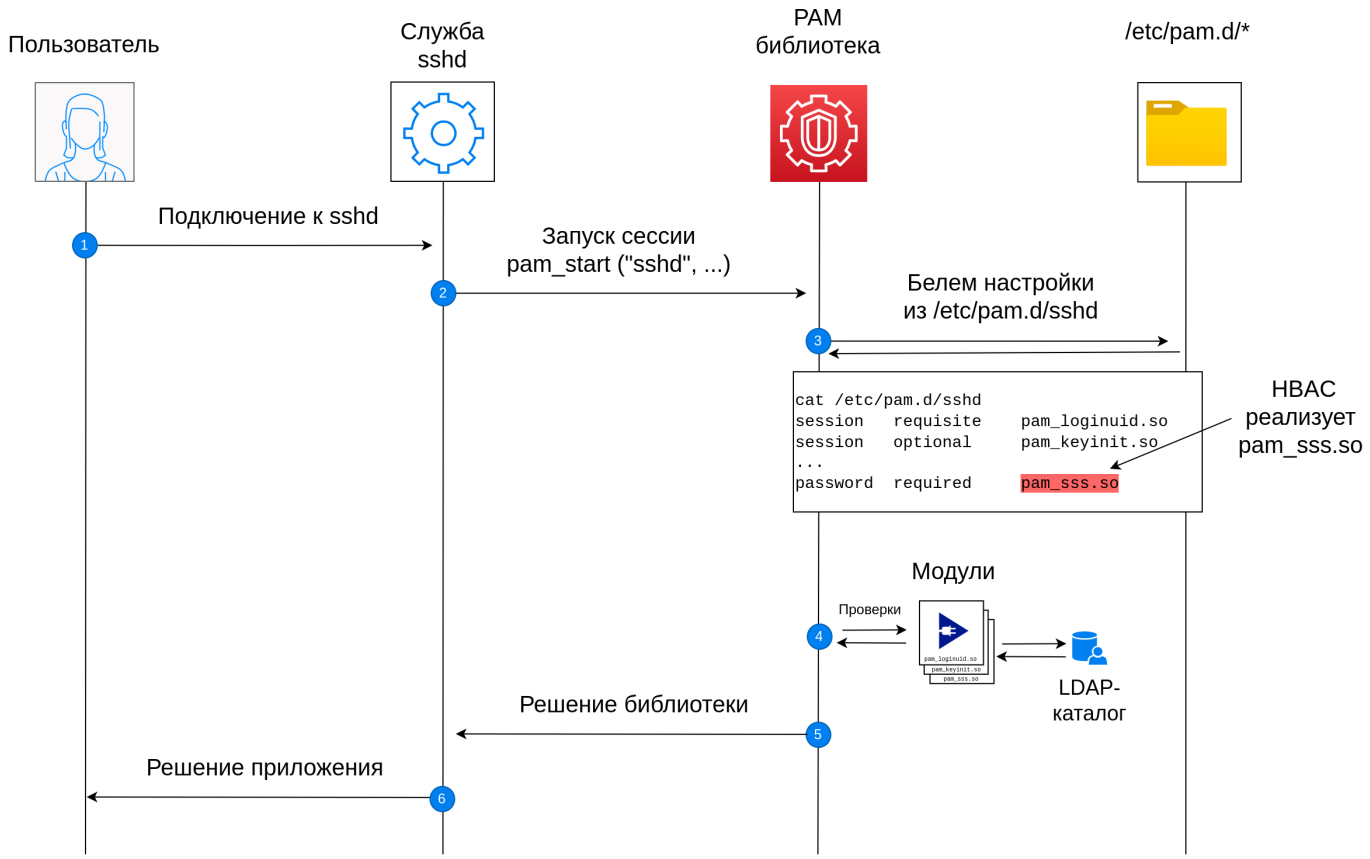


Рисунок 6.4 – Механизм работы HBAC-правил 1

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 упросить 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. Итоговый ответ библиотека передает приложению (5), которое, в свою очередь, использует эту информацию для организации взаимодействия с пользователем (6).

Модуль pam_sss.so является клиентской библиотекой службы sssd, 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.

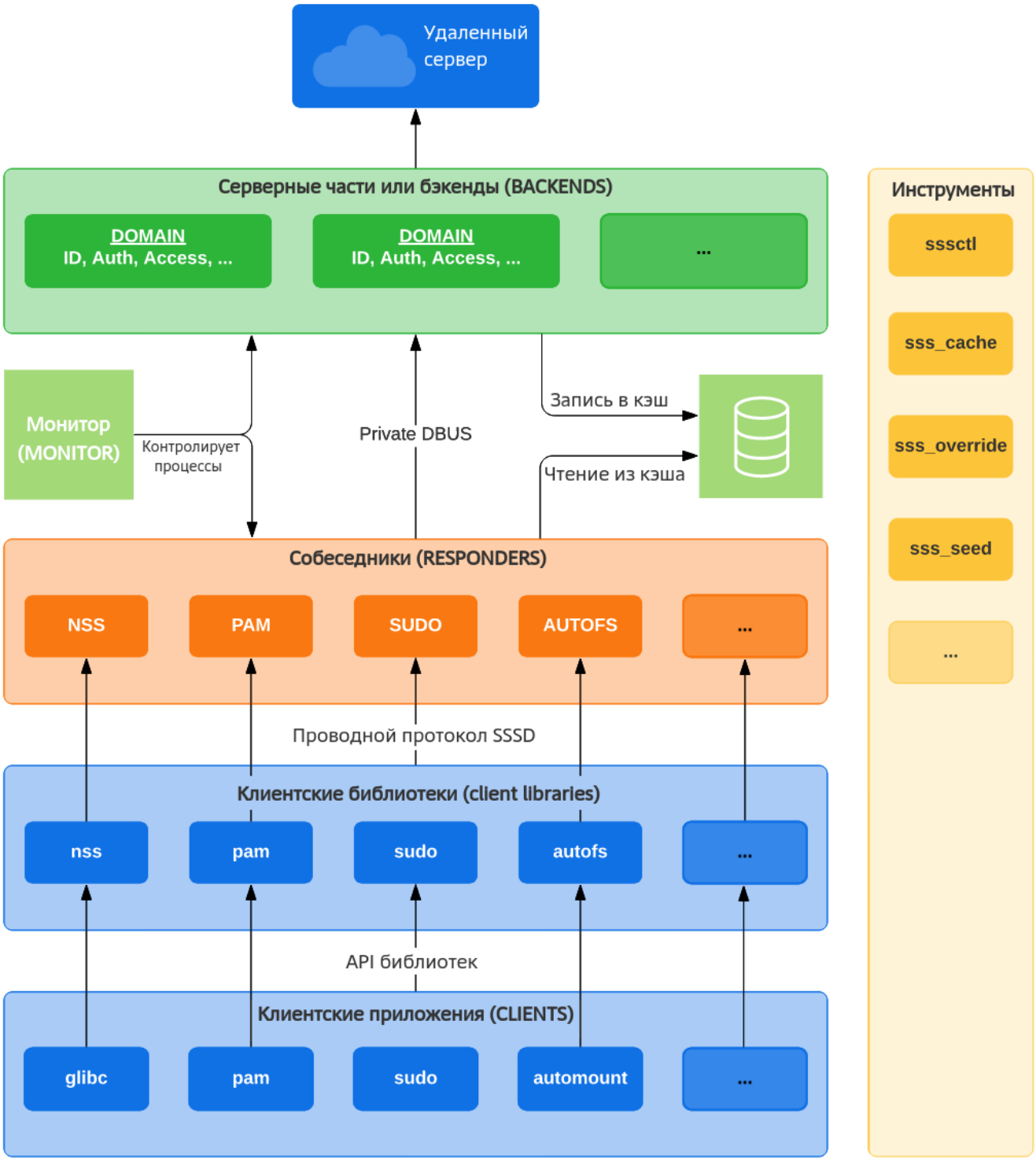


Рисунок 6.5 – Архитектура службы SSSD

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.

6.3.3. 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»

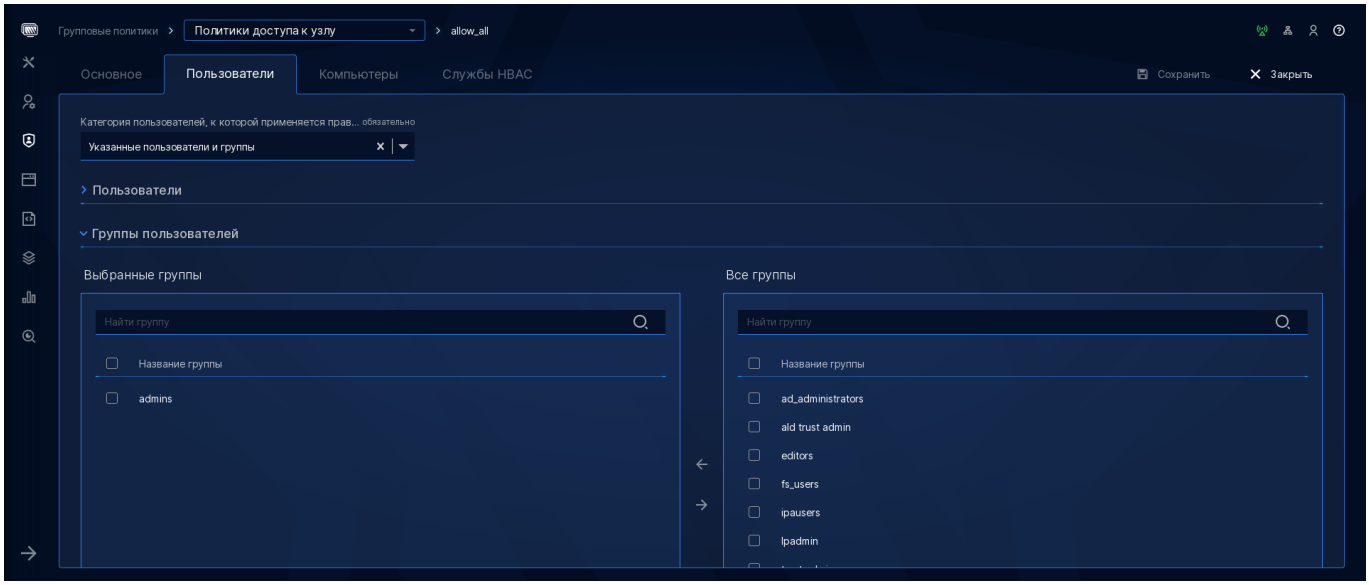


Рисунок 6.6 – Группа компьютера

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 порталу управления с любого другого компьютера, который не находится в домене, чтобы исправить ошибку. Доступ к порталу управления не регулируется через механизм HBAC.

6.3.4. 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».

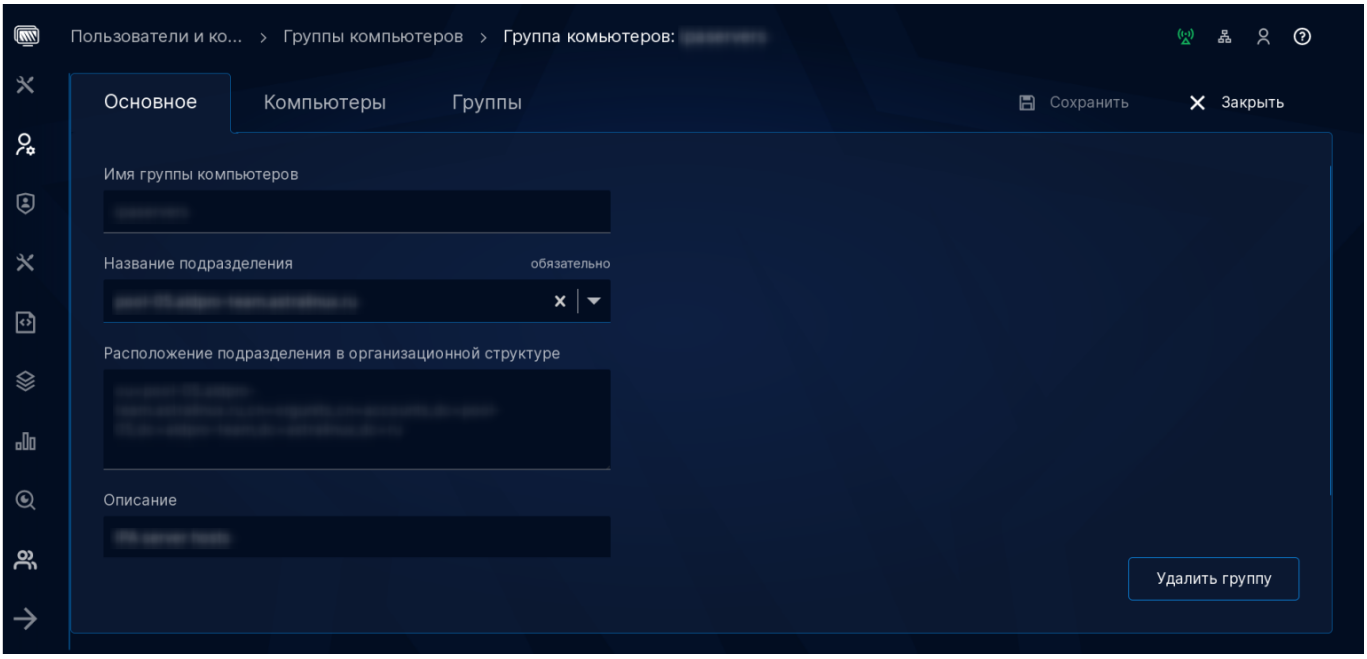


Рисунок 6.7 – Компьютеры

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

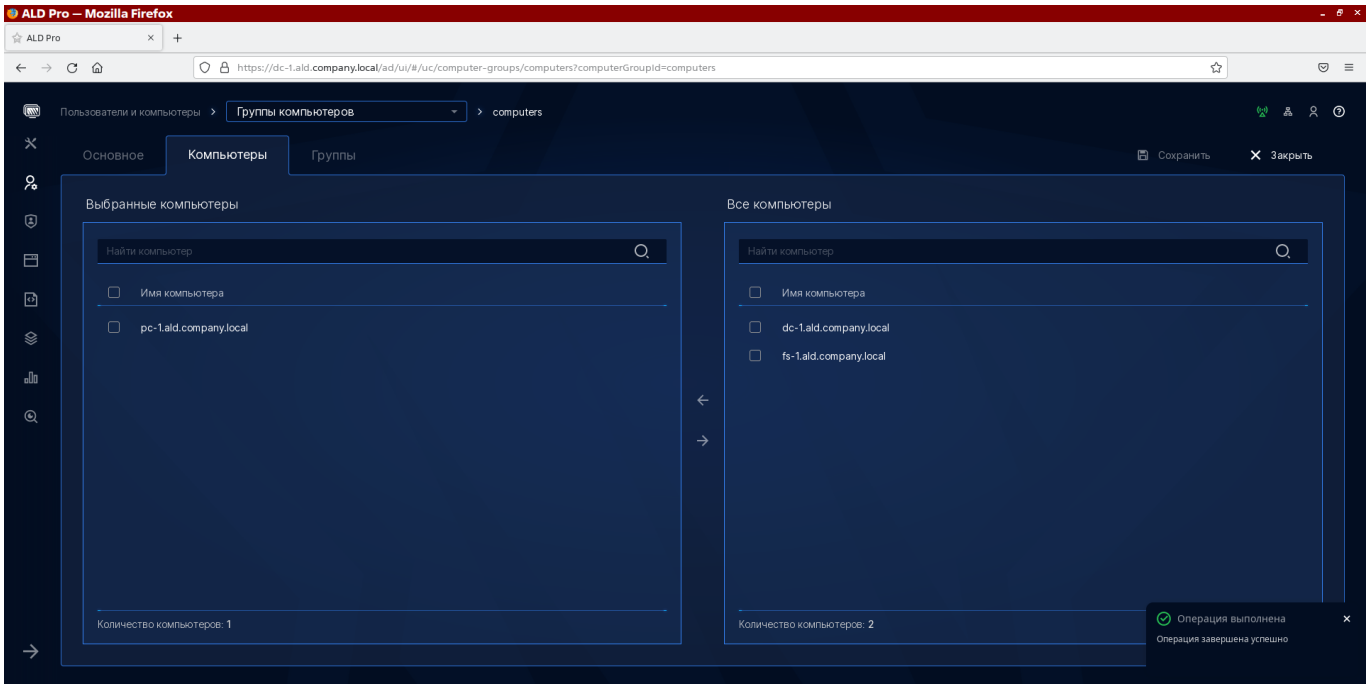


Рисунок 6.8 – Политика доступа к узлу 1

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

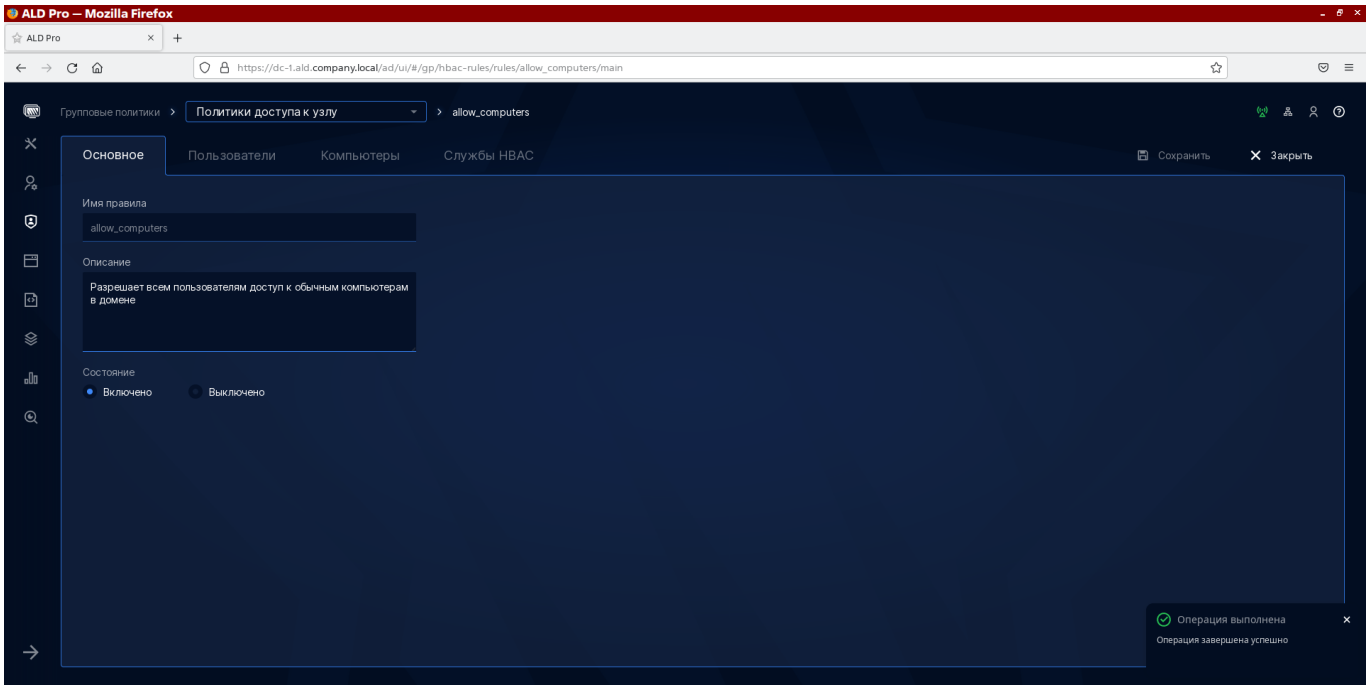


Рисунок 6.9 – Политика доступа к узлу 2

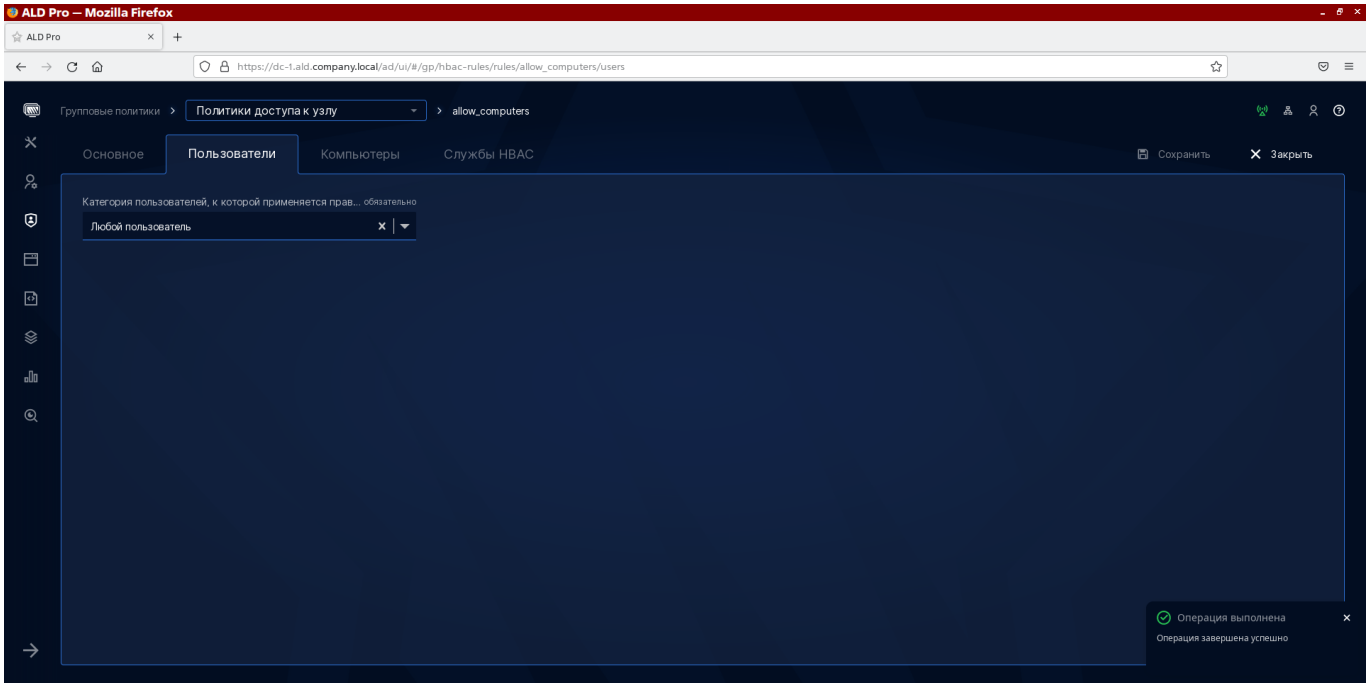


Рисунок 6.10 – Политика доступа к узлу 3

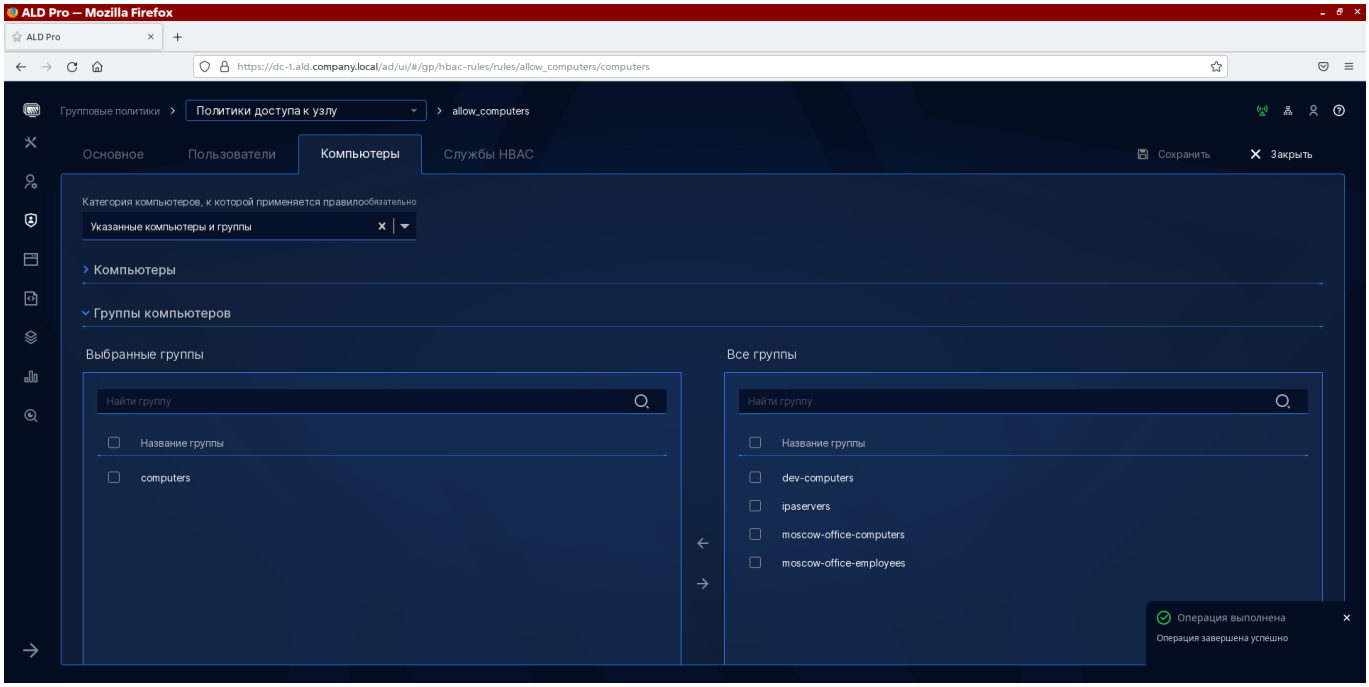


Рисунок 6.11 – Политика доступа к узлу 4

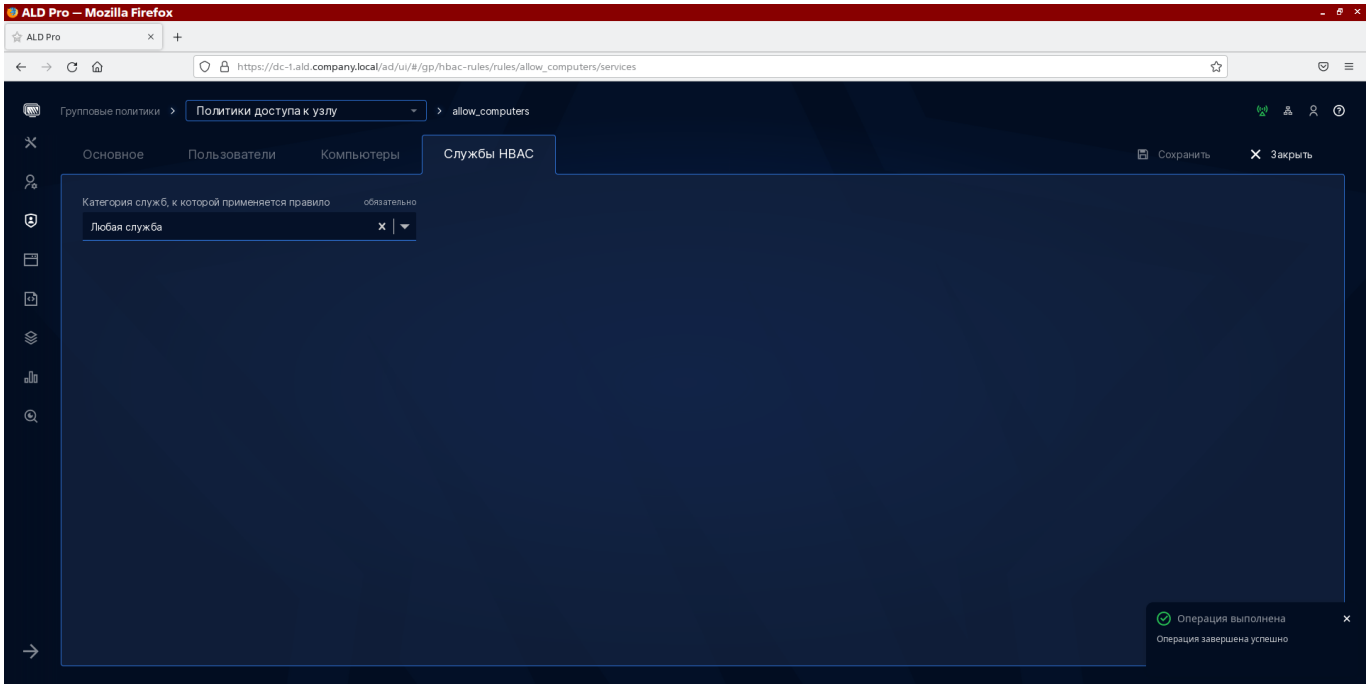


Рисунок 6.12 – Политика доступа к узлу 5

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
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
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
```

6.3.5. 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 RPD	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

Внимание: При переходе с ранних версий ALD Pro до 2.4.0 в системе будет создана группа служб HBAC fly куда будут входить: fly-dm, fly-dm-greeter, fly-dm-np, fly-wm. Если группа fly уже существовала в нее будут добавлены перечисленные службы.

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

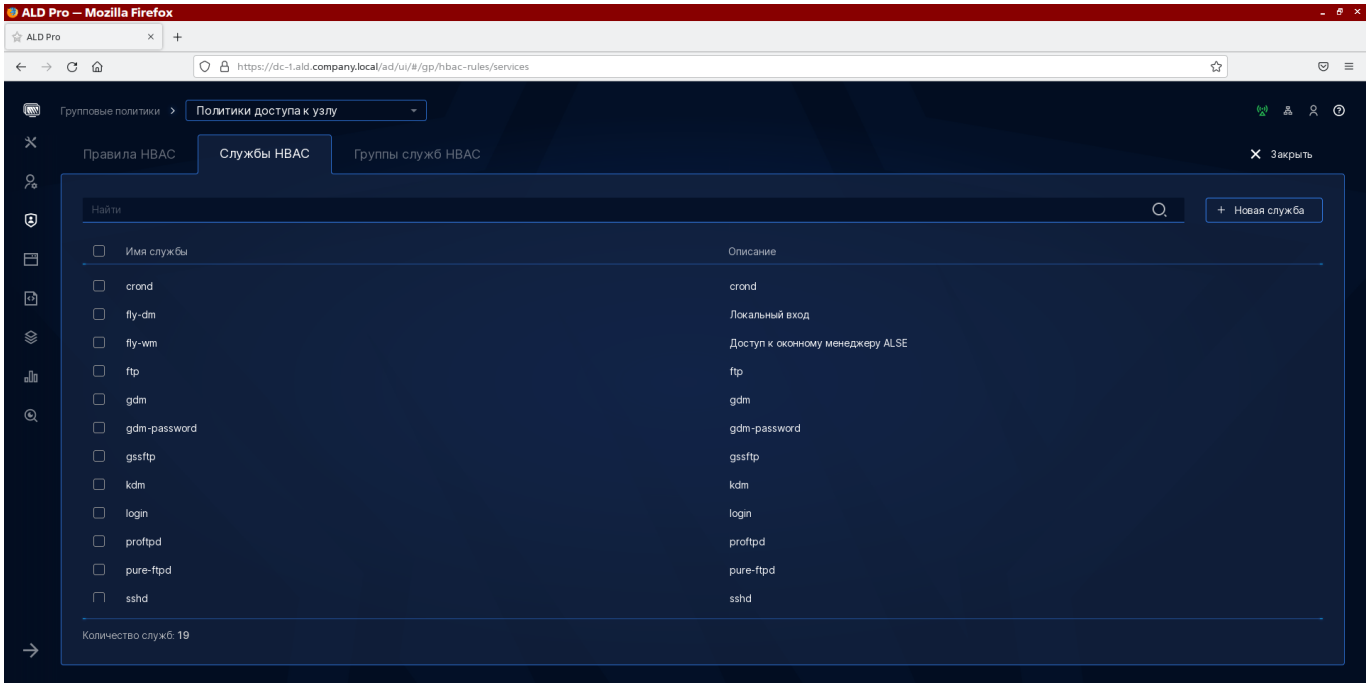


Рисунок 6.13 – Политика доступа к узлу 6

То же самое можно сделать из командной строки:

```
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.

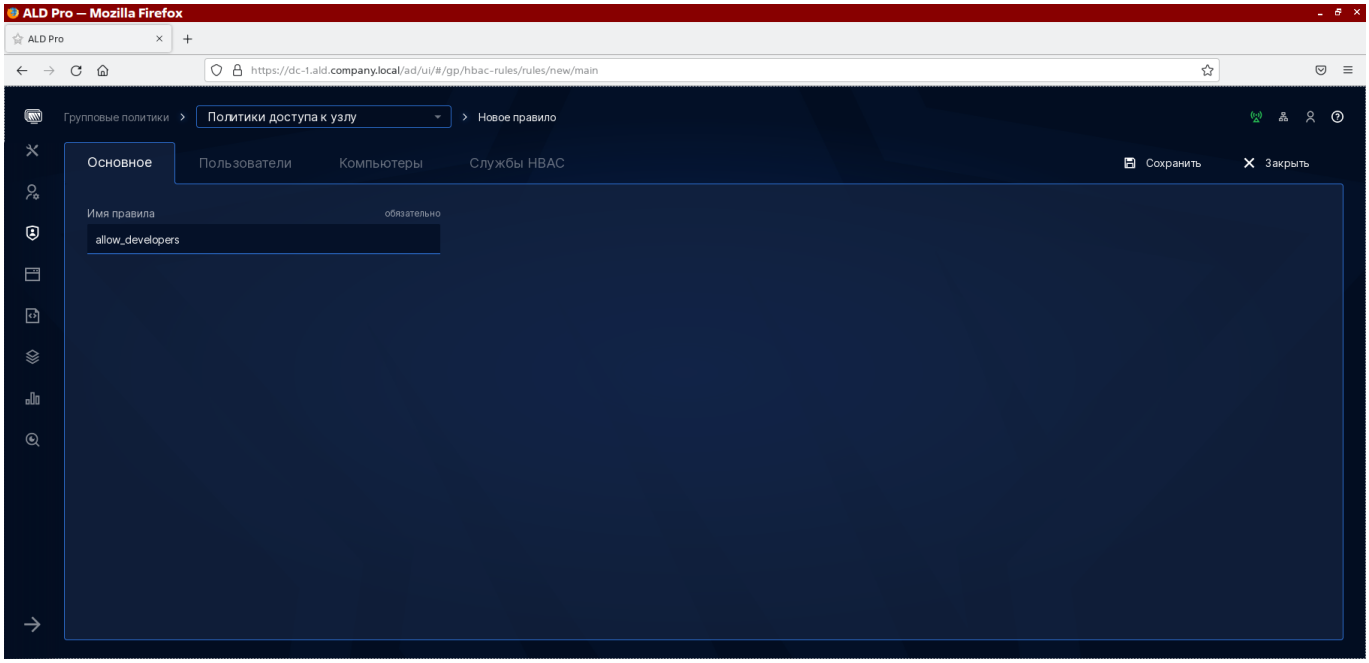


Рисунок 6.14 – Политика доступа к узлу 7

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.

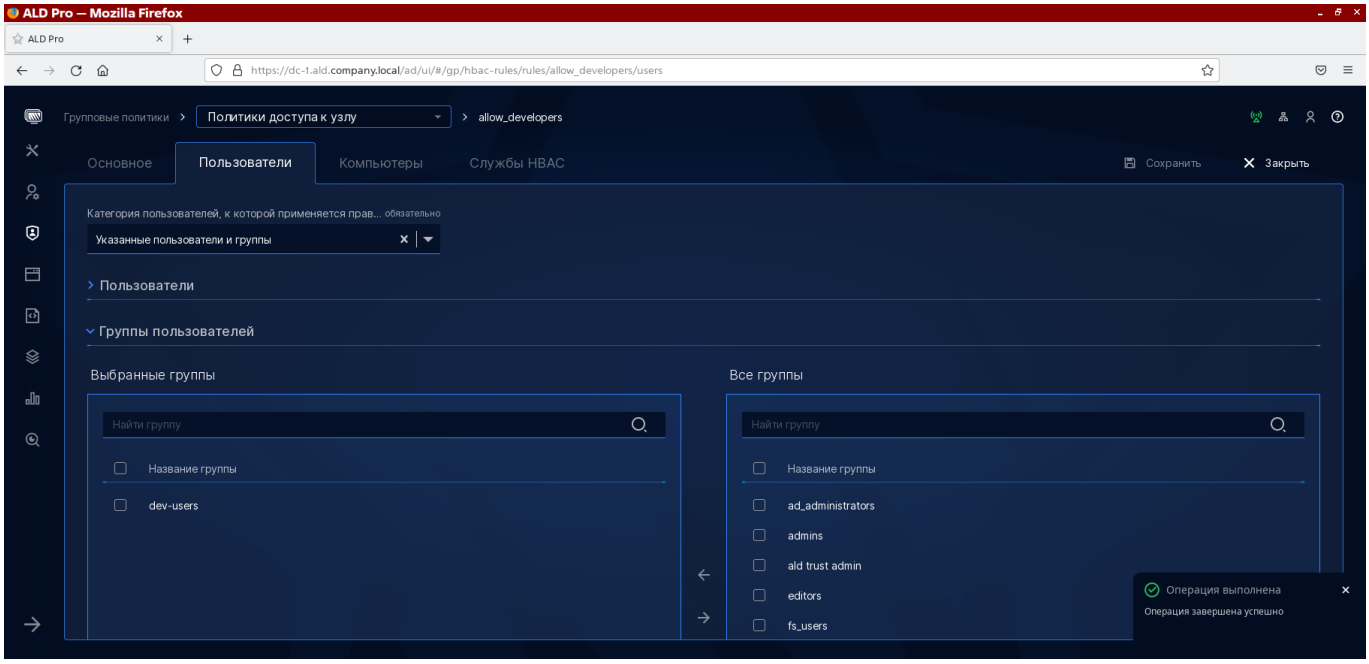


Рисунок 6.15 – Политика доступа к узлу 8

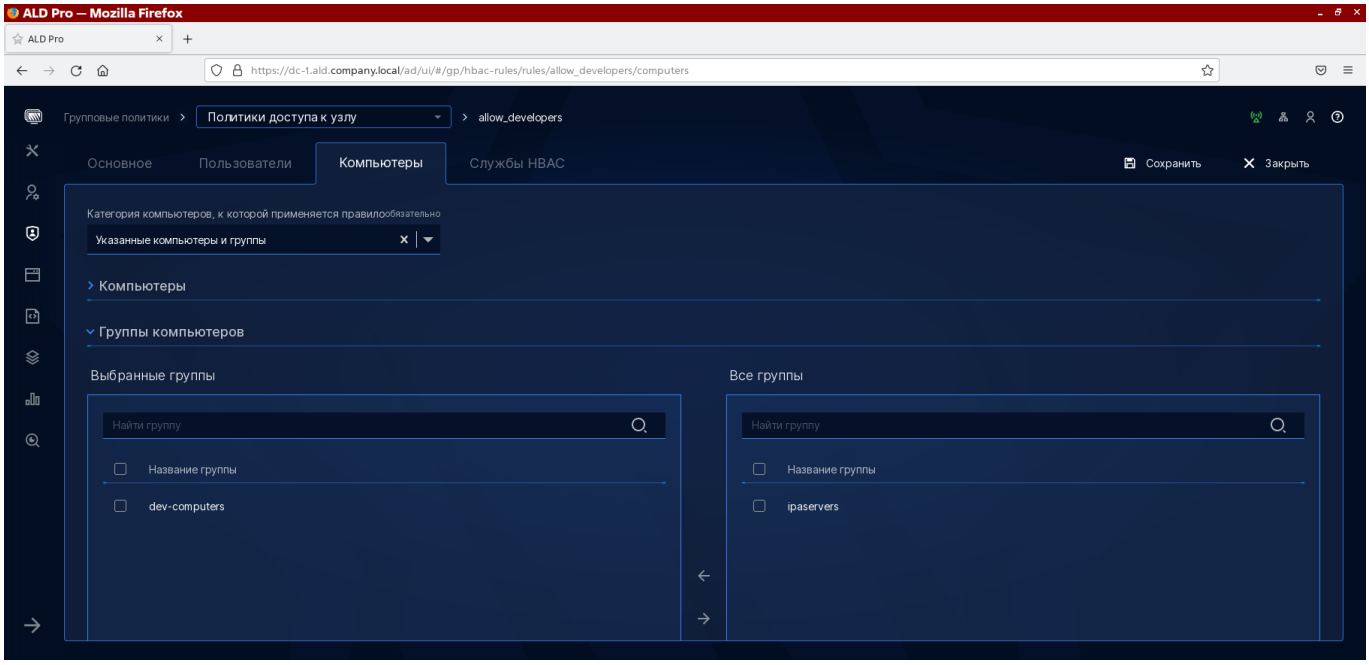


Рисунок 6.16 – Политика доступа к узлу 9

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»).

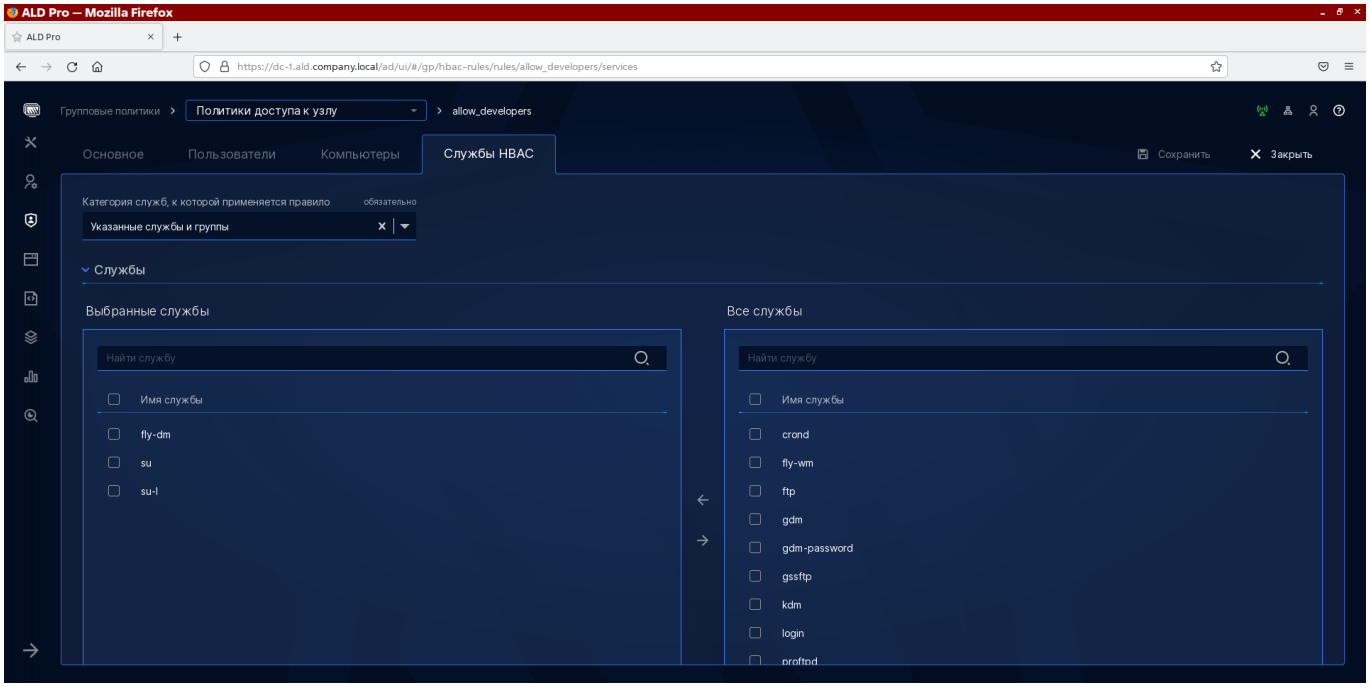


Рисунок 6.17 – Политика доступа к узлу 10

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 PAM 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.

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 потребуется 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 службе 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.

6.3.6. 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.

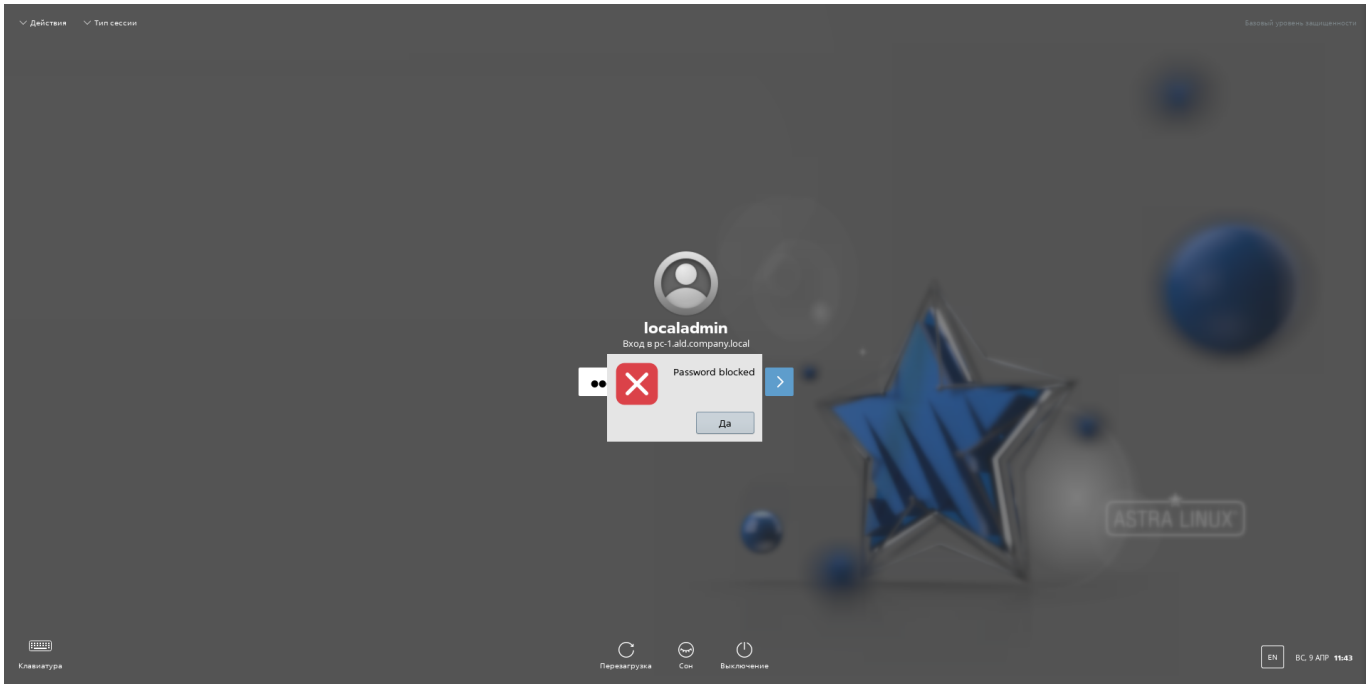


Рисунок 6.18 – Группа пользователей

6.3.7. Лучшие практики: создание HBAC-правил для структурных подразделений

Область применения HBAC-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 rbtadp, 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 rbtadp всех его дочерних объектов будет содержать `ou=Московский офис,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan`. 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».

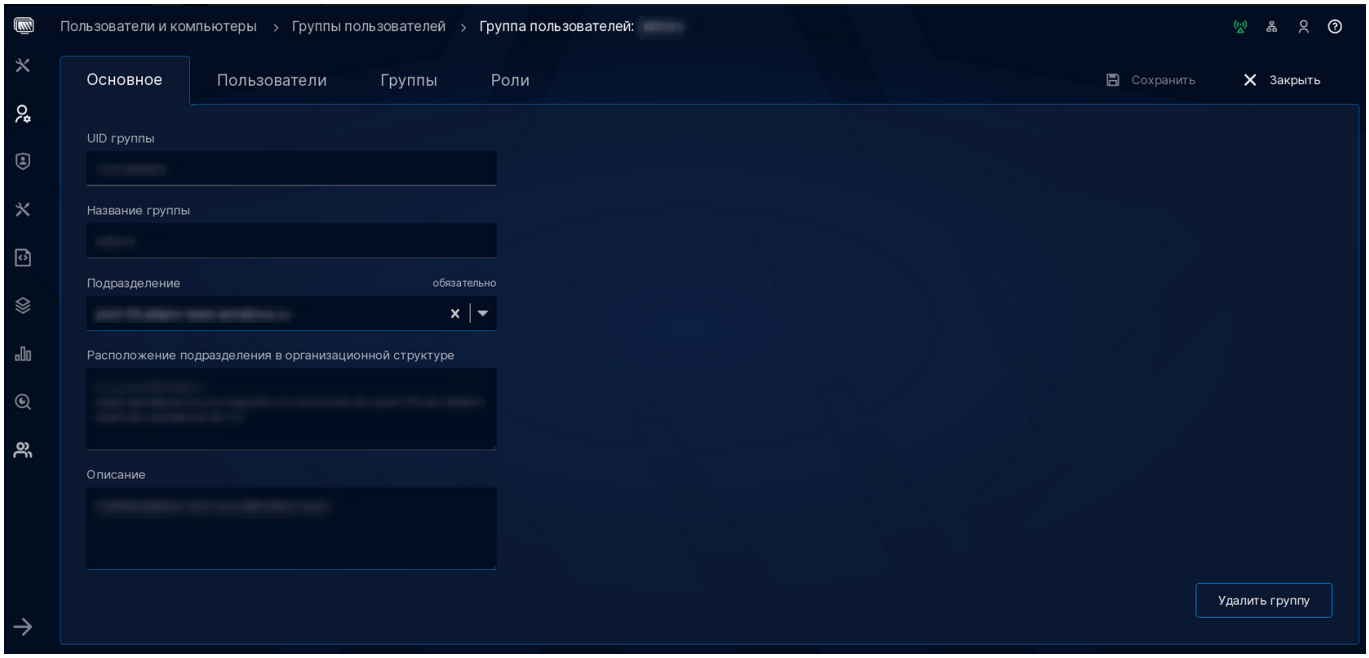


Рисунок 6.19 – Добавление правила

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».

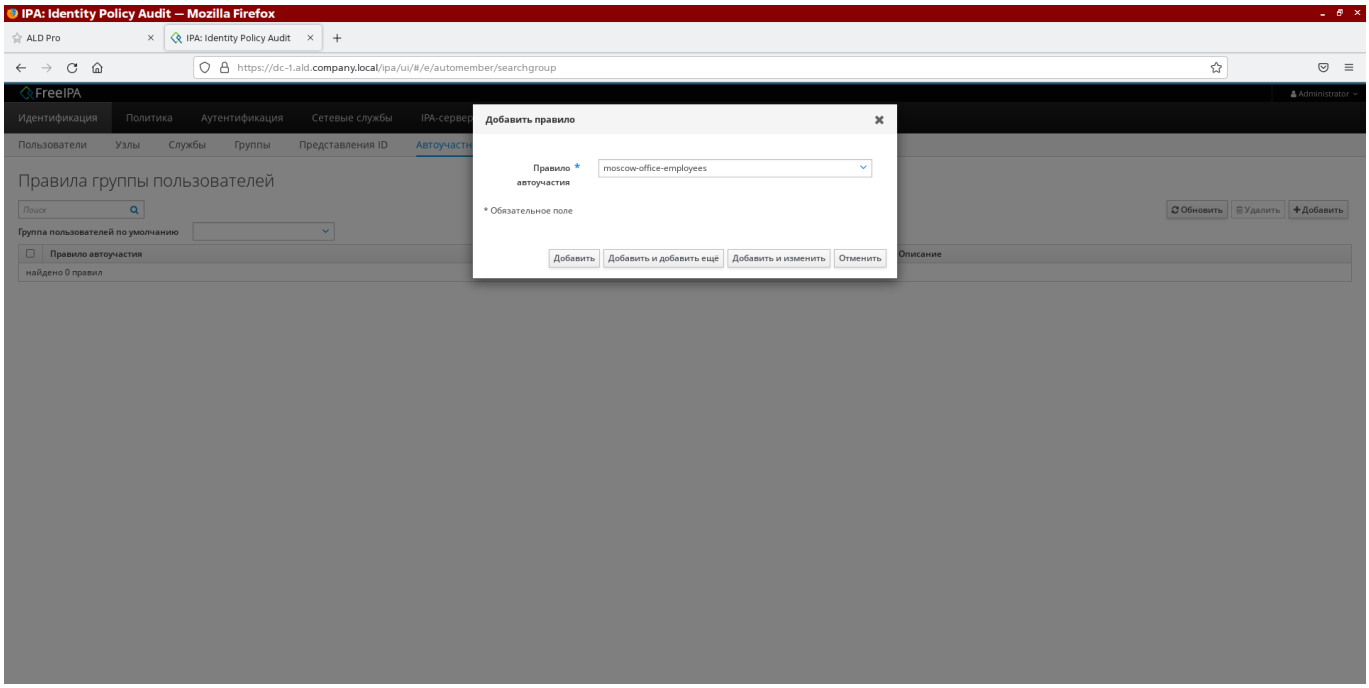


Рисунок 6.20 – Правило группы пользователей

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

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. Обзор записей выполняется по полному вхождению, поэтому никаких лишних пробелов в середине строки быть не должно.

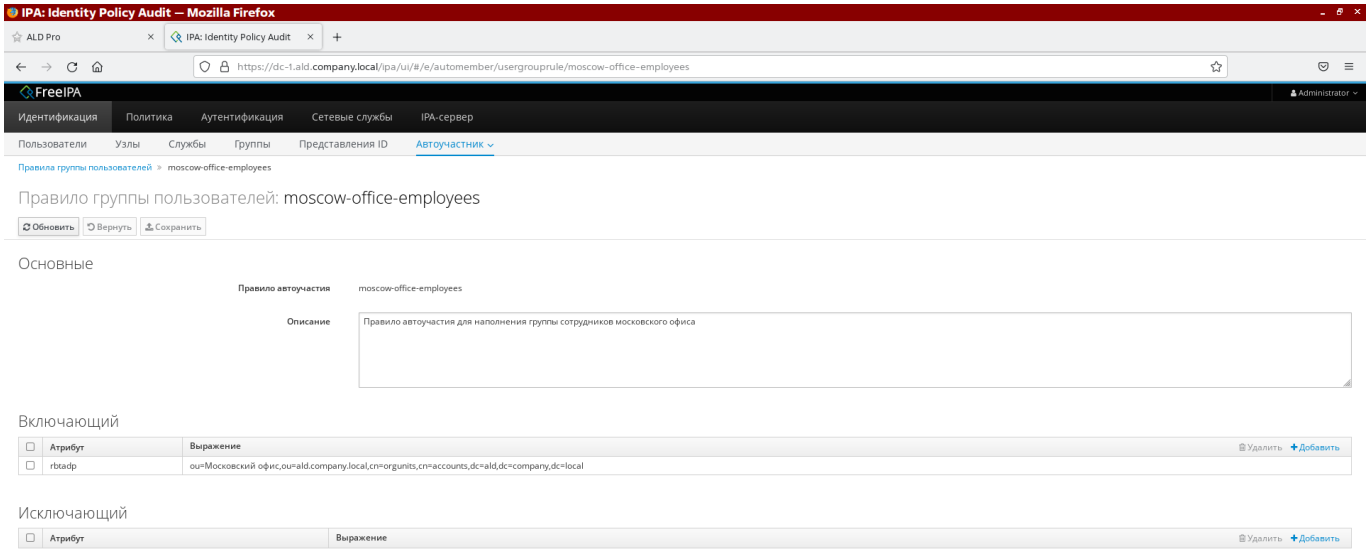


Рисунок 6.21 – Правило группы пользователей

Тоже 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.lan, cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan"
ipa group-mod moscow-office-employees --desc='Группа, в которой будут все
 ↪сотрудники московского офиса'
```

2. Создайте правило автоучастия и определите критерий автоучастие:

```
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.lan,
↪cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan'
```

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
```

6.4. Автоматизация задач администрирования через LDAP запросы

6.4.1. Введение

Автоматизация позволяет ускорить выполнение массовых операций администрирования, например, если вам нужно завести тысячу пользователей, вы можете написать скрипт, чтобы импортировать учетные записи из CSV-файла, автоматически распределить пользователей по подразделениям и включить их в соответствующие группы. Это не только сэкономит рабочее время, но и позволит исключить ошибки, связанные с человеческим фактором.

Из настоящего документа вы узнаете об устройстве LDAP-каталога и о том, как управлять доменом с помощью прямых запросов к каталогу. Мы также рассмотрим несколько полезных запросов для решения реальных задач администрирования.

6.4.2. Технология LDAP

Служба каталога ALD Pro построена на базе 389 Directory Server, который реализует функции каталога и поддерживает протокол LDAP v3. Облегченный протокол доступа к данным каталога (Lightweight Directory Access Protocol, LDAP) является упрощенной

модификацией более строгих стандартов для построения службы распределенного каталога сети X.500.

Каталог LDAP является специализированной нереляционной базой данных, файлы которой вы найдете в папке `/var/lib/dirsrv/slapd-ald-company-lan/db`. Информация каталога представлена в виде древовидной структуры, которую также называют Directory Information Tree или сокращенно DIT

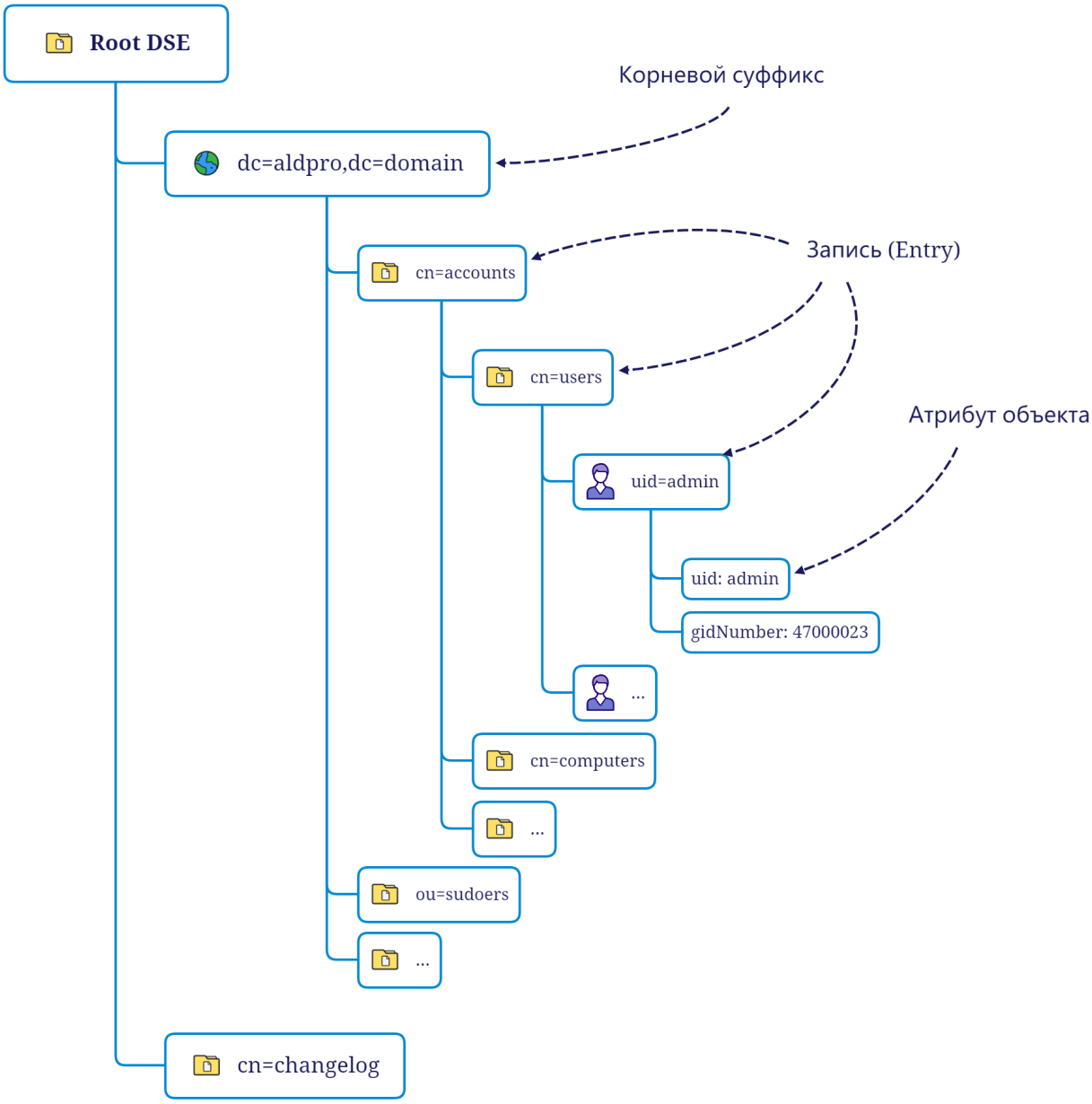


Рисунок 6.22 – Структура каталога Directory Information Tree

Корень каталога называется Root DSE, где DSE означает Directory system agent Specific Entry, то есть специализированная запись агента системы директорий. Эта запись не имеет

родителя, и она описана в файле `/etc/dirsrv/slapd-ALD-COMPANY-LAN/dse.ldif`, где slapd-ALD-COMPANY-LAN директория сервиса `slapd` с именем домена.

Корень Root DSE является родителем для записей **«dc=ald,dc=company,dc=lan и «cn=changelog»**, которые называют так же базовыми записями, корневыми суффиксами или контекстами именования (base entry, root suffix, naming context). В контексте **«dc=ald,dc=company,dc=lan** хранятся все объекты каталога, а в **«cn=changelog»** – журнал изменений для работы плагина «Retro Changelog Plugin». Все контексты, определенные в каталоге, указаны в операционном атрибуте **namingContexts** записи **Root DSE**, однако спецификация LDAP позволяет также использовать служебные контексты, например, в записи **«cn=config»** представлены настройки каталога, а через запись **«cn=monitor»** можно получить доступ к информации о состоянии сервера в режиме реального времени.

Существуют разные подходы к наименованию корневых суффиксов, в ALD Pro (FreeIPA) используются правила спецификации RFC-2247, поэтому для домена ald.company.lan имя корневого суффикса будет **«dc=ald,dc=company,dc=lan**, где **dc** – компонент имени домена, сокращение от англ. Domain Component. Правила наименования необходимы для возможности преобразования DNS-имени в имя LDAP-записи и наоборот.

От корневого суффикса **«dc=ald,dc=company,dc=lan** ответвляются дочерние записи (Entry), которые, в свою очередь, могут быть контейнерами для других записей, за счет чего и образуется древовидная структура. Таким образом записи каталога можно сравнить с директориями файловой системы.

У каждой записи каталога есть имя, которое должно быть уникальным в пределах родительского контейнера, поэтому оно называется относительно уникальным именем Relative Distinguished Name или кратко RDN. Учетные записи пользователей, например, имеют имена **«uid=admin», «uid=ivan.kuznetsov»** и т.д. Особенность имен объектов в LDAP заключается в том, что они хранят не конкретные значения, а только ссылки на хранимые атрибуты записей, которые используются для идентификации объектов. Например, для идентификации учетных записей пользователей используют атрибут **uid**, для учетных записей компьютеров **fqdn** (fully qualified domain name, полное доменное имя хоста), а в именах контейнеров обычно присутствует cn (common name, общее имя).

В приведенном примере RDN учетной записи доменного администратора **«uid=admin»** состоит из названия атрибута **«uid»**, после которого идет символ присвоения «=» и далее значение атрибута «admin». Вся эта запись вместе называется Определением Значения Атрибута от англ. Attribute Value Assertion (AVA). Обычно имена записей задаются значением одного атрибута, но могут использоваться и несколько, тогда в имени RDN эти определения AVA будут объединяться знаком «+», например: **«cn=Ivan+l=Moscow»**.

Каталог 389 Directory Server поддерживает такой способ именования записей, но в ALD Pro (FreeIPA) он не используется.

Для идентификации объекта в пределах всего каталога используют уникальное имя Distinguished Name или сокращенно DN. Уникальное имя представляет из себя цепочку RDN, которые записывают через запятую слева направо, начиная с целевой записи и до корневого суффикса вверх по иерархии, см. Рис 2. Если привести аналогию с объектами файловой системы, то RDN будет соответствовать имени объекта директории или файла, а DN – полному имени объекта файловой системы, которое включает путь к родительской объекту и имени объекта. И также, как на одном диске не может быть двух директорий с одинаковыми полными именами, в каталоге LDAP не может быть двух записей с одинаковыми DN. Описание формата DN можно найти RFC 4514.

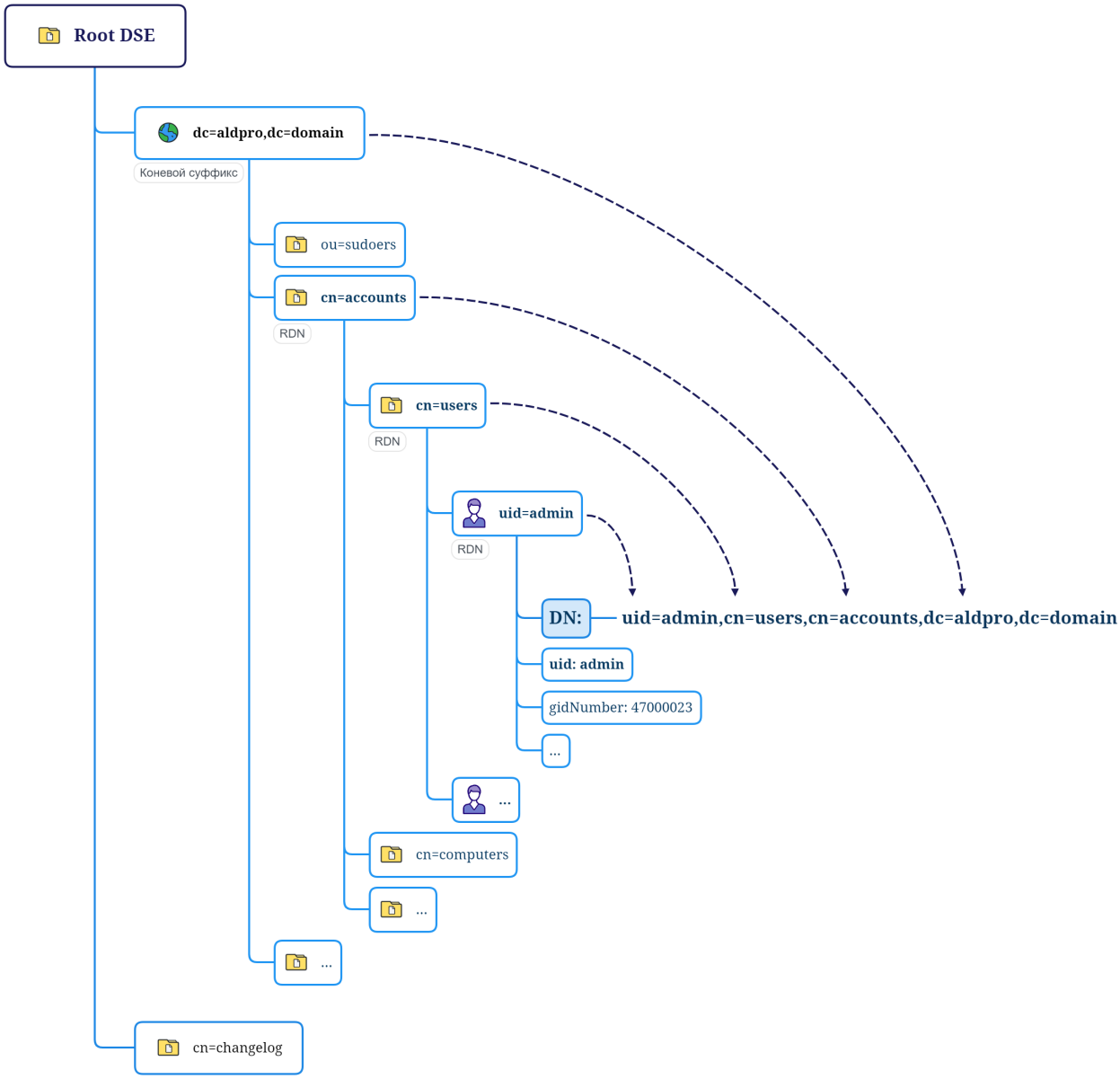


Рисунок 6.23 – Пример формирования уникального имени записи DN

Запись каталога может хранить не только дочерние записи, но и набор атрибутов, описывающих ее свойства. Например, для учетной записи пользователя это могут быть ФИО, должность, номер телефона и т.д.

Данные в LDAP-каталоге строго структурированы и все атрибуты должны быть определены в схеме данных заранее. Перечень доступных для конкретного объекта атрибутов задается списком назначенных ему классов, см. атрибут objectClass. Например, учетные записи пользователей могут содержать атрибут gidNumber по той причине, что им назначен класс объектов posixAccount, см. Рис 3.

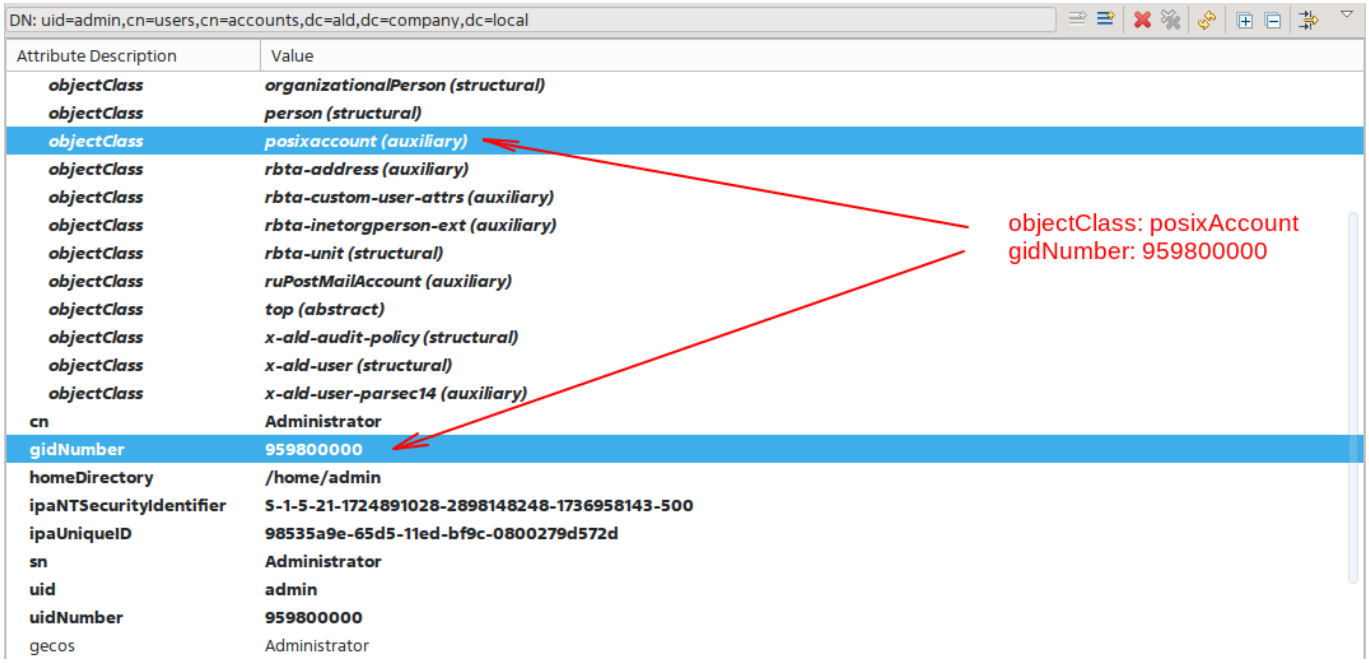


Рисунок 6.24 – Возможность указать gidNumber определяется наличием класса объектов posixAccount

Всем объектам каталога назначен как минимум один класс **top**, т.к. в нем описан атрибут **objectClass**, с помощью которого работает механизм назначения классов. Но практической пользы от объектов с одним классом не много, поэтому обычно в каталоге у объектов два и более классов.

В качестве примера класса рассмотрим класс объектов **posixGroup**, который определен в схеме следующим образом:

```
( 1.3.6.1.1.1.2.2 NAME 'posixGroup' DESC 'Standard LDAP objectclass' SUP top
↪STRUCTURAL MUST ( cn $ gidNumber ) MAY ( userPassword $ memberUid $
↪description ) X-ORIGIN 'RFC 2307' )
```

где:

- **1.3.6.1.1.1.2.2** – это идентификатор объекта (object id, OID). Глобальные идентификаторы присваиваются международными организациями (IANA, ISO, ITU-T, ANSI, BSI), а для расширения схемы в прикладных системах используется пространство номеров с префиксом «1.3.6.1.4.1.X», где X – это внутренний номер организации. Например, объекты РусБИТех-Астра имеют префикс «1.3.6.1.4.1.52616.*»
- **NAME „ „** – инструкция NAME задает имя класса
- **DESC „ „** – инструкция DESC задает описание класса

- **SUP „ „** – инструкция SUP указывает родительский класс. В приведенном примере наследование идет от класса «top», поэтому объектам будет доступен его атрибут objectClass.
- **STRUCTURAL** – инструкция указывает, что класс будет относиться к виду структурных. Существуют также абстрактные (ABSTRACT) и вспомогательные (AUXILIARY) классы, но в контексте автоматизации различия между видами классов не принципиальны.
- **MUST и MAY** – инструкции, которые позволяют задать списки обязательных и дополнительных атрибутов. Полный перечень атрибутов, доступных объекту, расширяется атрибутами, которые наследуются от всех родительских классов.
- **X-ORIGIN „ „** – инструкция, которая позволяет задать комментарий с ссылкой на документацию, из которой можно почерпнуть дополнительную информацию об этом классе. В приведенном примере информацию следует искать в документе RFC 2307.

Обратите внимание на то, что один и тот же атрибут может быть использован в нескольких классах, поэтому пользователям можно задать значение **gidNumber**, т.к. им назначен класс **posixAccount**, и, в тоже время, он может быть задан у групп, т.к. им назначен класс **posixGroup**, см. Рис. 4.

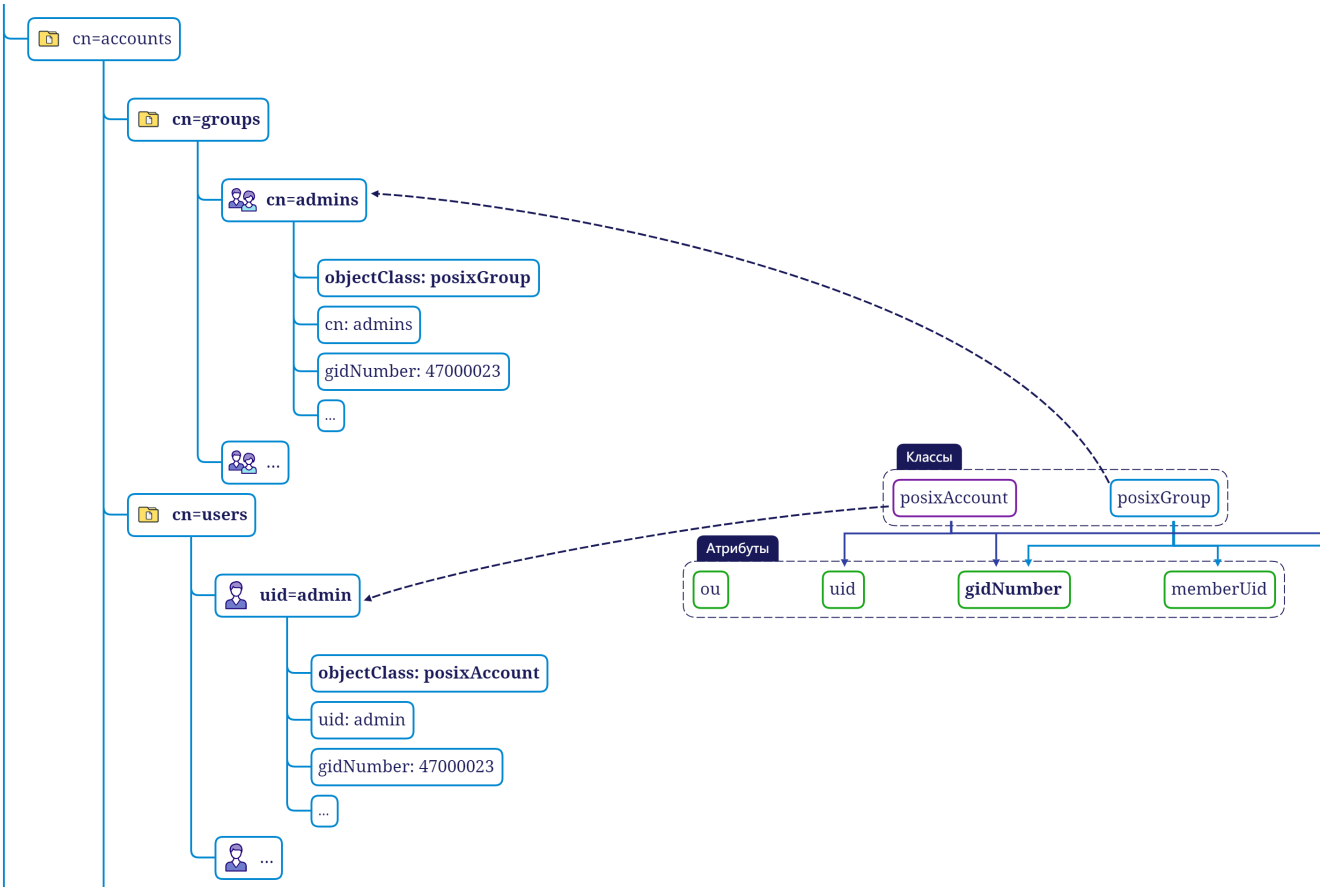


Рисунок 6.25 – Использование атрибута gidNumber классами posixAccount и posixGroup

Теперь рассмотрим описание атрибута **gidNumber** подробнее:

```
( 1.3.6.1.1.1.1.1 NAME 'gidNumber' DESC 'Standard LDAP attribute type'
↪SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE X-ORIGIN 'RFC 2307' )
```

где:

- **1.3.6.1.1.1.1.1** – это уникальный идентификатор атрибута.
- **NAME „ „** – инструкция NAME задает атрибута
- **DESC „ „** – инструкция DESC задает описание атрибута
- **SYNTAX „ „** – инструкция задает тип хранимого в атрибуте значения. В приведенном примере 1.3.6.1.4.1.1466.115.121.1.27 соответствует целым числам. Описание всех типов данных можно найти в RFC4517.
- **SINGLE-VALUE** – инструкция указывает, что атрибут может хранит только одно значение. Если этой инструкции не будет, то объектам можно будет присваивать несколько значений этого атрибута.
- **X-ORIGIN „ „** – инструкция, которая позволяет задать комментарий с ссылкой на документацию.

Описание схемы данных хранится в файлах на диске в директориях:

- `/usr/share/dirsrv/schema/`
- `/etc/dirsrv/schema/`
- `/usr/share/dirsrv/updates/`
- `/etc/dirsrv/slapd-ALD-COMPANY-LAN/`

при обращении к каталогу по LDAP информацию можно получить из операционного DIT «cn=schema».

6.4.3. Взаимодействие с каталогом через LDAP-протокол

6.4.3.1. Графическое приложение для работы с LDAP-каталогом (Apache Directory Studio)

Для работы с LDAP-каталогом из графического интерфейса – просмотра структуры каталога, редактирования записей, импорта и экспорта данных – можно воспользоваться таким бесплатным инструментом как Apache Directory Studio, см. Рис 5. Загрузить

приложение можно с официального сайта directory.apache.org/studio/, для работы потребуется java runtime.

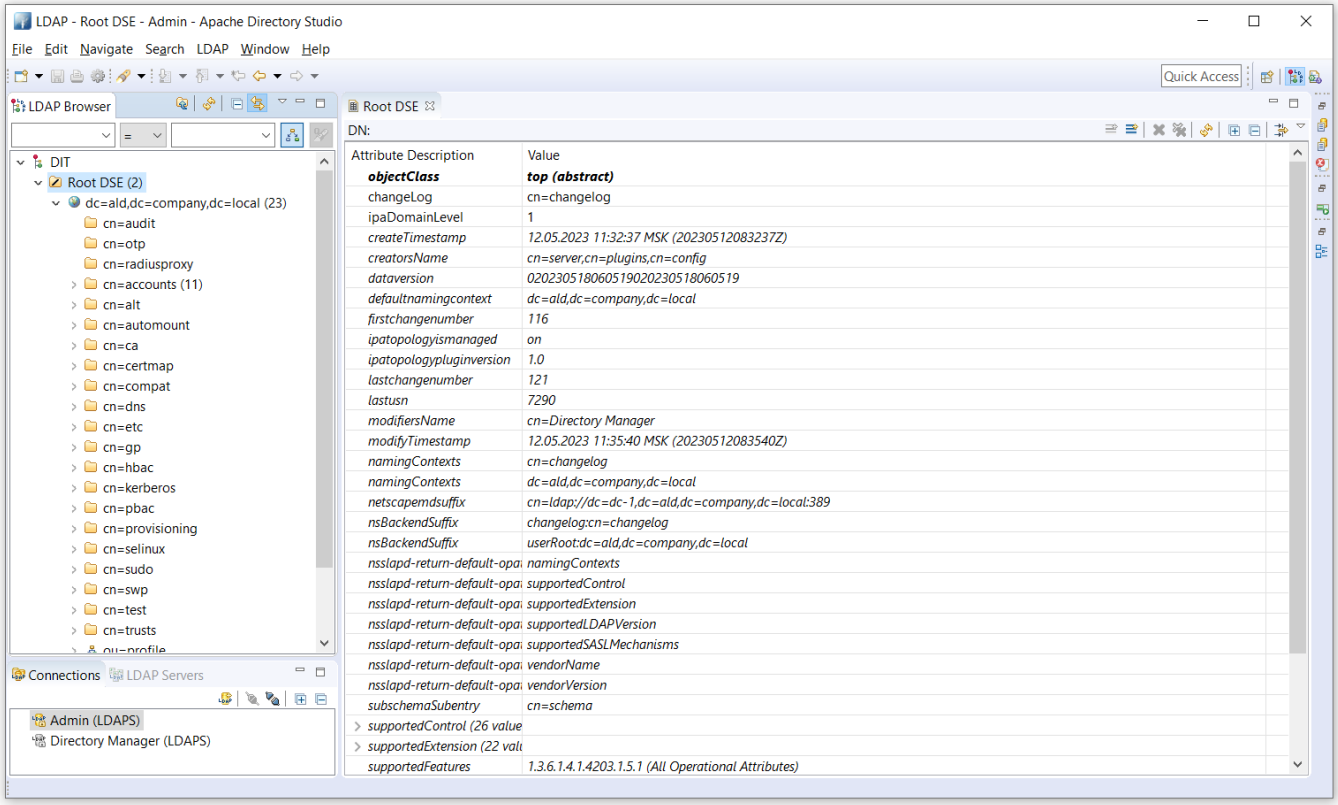


Рисунок 6.26 – Apache Directory Studio

Настройка соединения

Чтобы подключиться к LDAP каталогу нужно создать новое соединение через меню «LDAP > New connection». Откроется окно для создания нового подключения см. Рис 6.

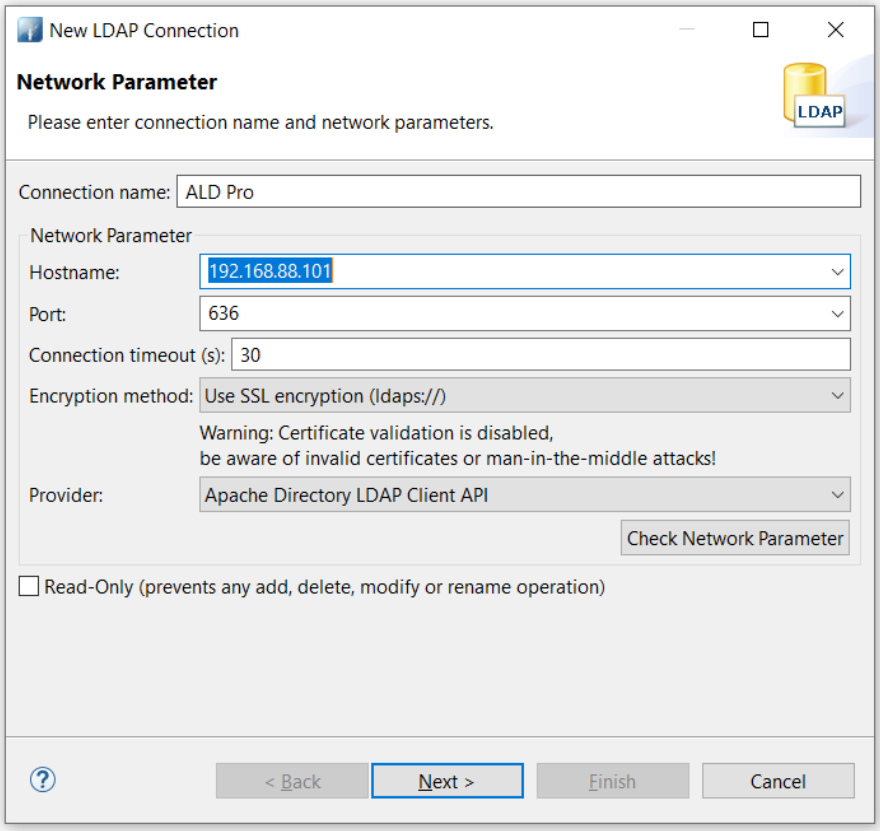


Рисунок 6.27 – Настройка сети для нового LDAP-подключения

Примечание: По умолчанию Apache Directory Studio предлагает создать незащищенное подключение на порт 389, что допустимо только при обращении к каталогу по localhost, т.е. приложение должно быть установлено непосредственно на контроллер домена, т.к. при подключении пароль будет передаваться в открытом виде. Если вы подключаетесь к каталогу с другого компьютера в сети, обязательно используйте порт 636 и метод шифрования SSL, чтобы перехват пароля был невозможен.

Аутентификация по паролю называется связыванием (Bind), вы можете подключиться к каталогу как супер пользователь `cn=Directory Manager` или доменным администратором ALD Pro `uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan`, см. Рис 7. В зависимости от учетной записи у вас будут разные права на доступ к записям и атрибутам.

Примечание: Сразу после установки системы пароли этих учетных записей совпадают. Чтобы сбросить пароль Directory Manager вам потребуется вручную менять хэш,

записанный в файле dse.ldif, в строке, начинающейся с nsslapd-rootpw.

Перед закрытием окна проверьте корректность подключение нажатием кнопки «Check Authentication».

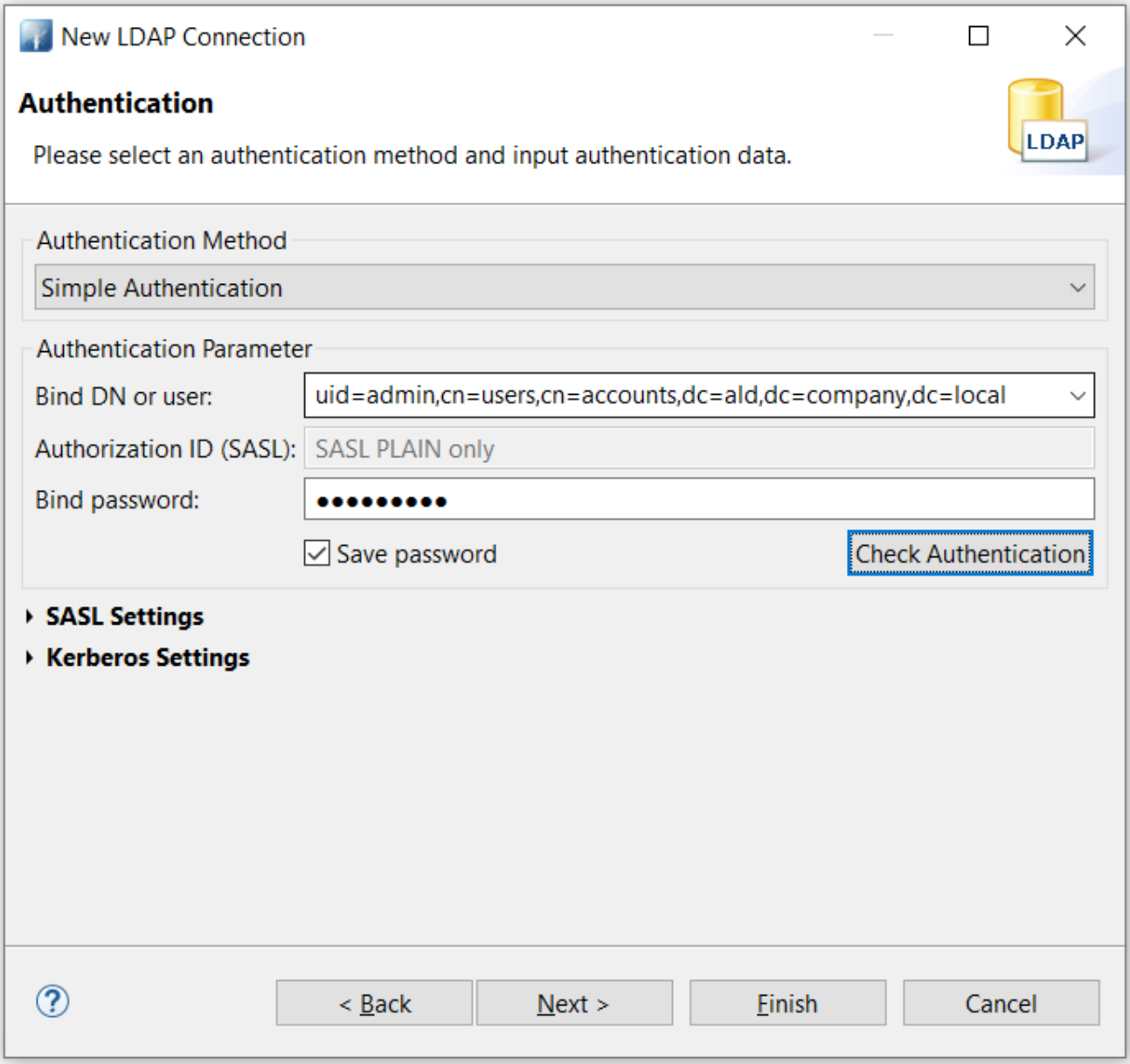


Рисунок 6.28 – Настройка аутентификации для нового LDAP-подключения

После добавления подключения новый сервер появится в списке Connections и вы сможете подключиться к серверу двойным кликом по этой записи.

Просмотр и экспорт объектов каталога

Вы можете просмотреть записи в дереве каталога, выбрав нужный RDN из окна LDAP browse. Например, нажав на пункт `cn=accounts` из списка слева, основные атрибуты отобразятся в центральном окне с именем `cn=accounts,dc=ald,dc=company,dc=lan`.

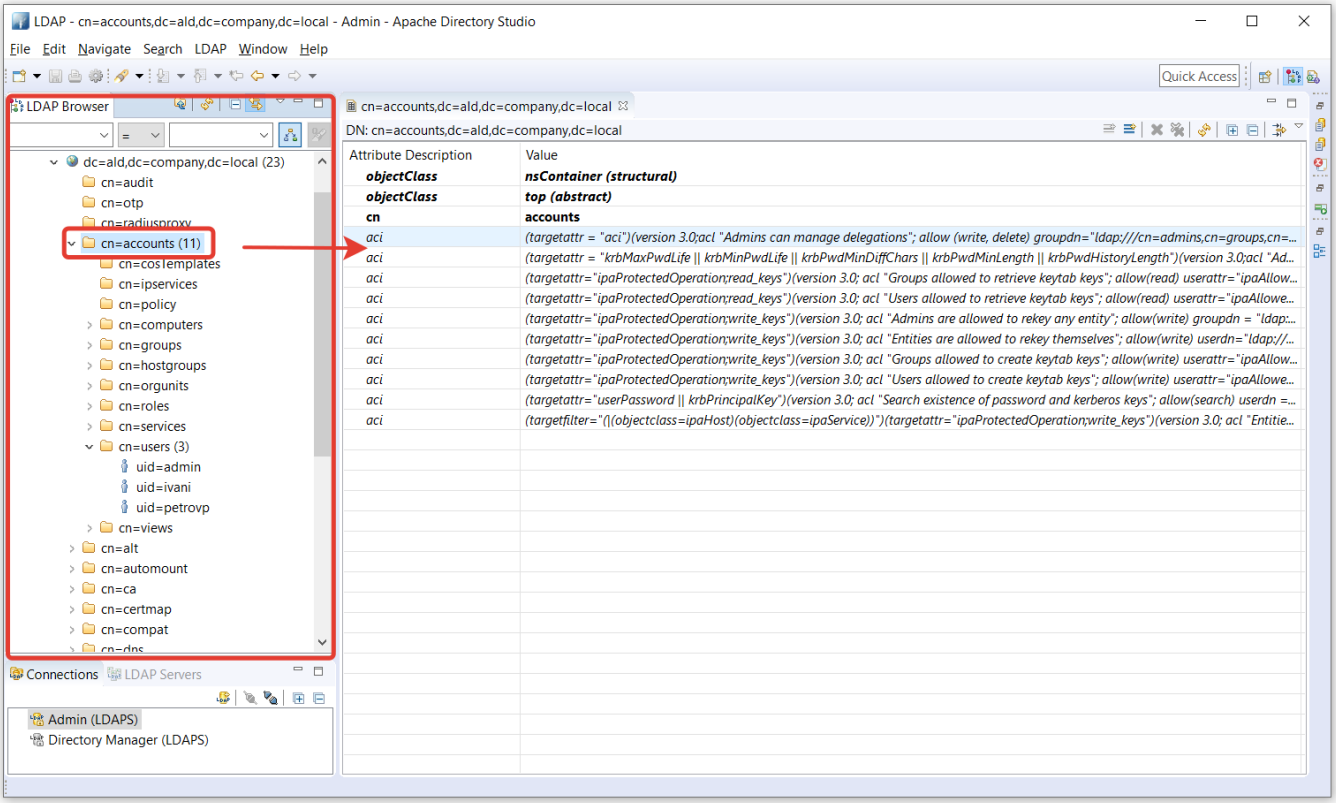


Рисунок 6.29 – Окно LDAP browse

А также можно открыть любой DN из меню «Navigate > Go to DN», например `cn=config`, но для его просмотра нужна учетная запись `cn=Directory Manager`. Рассмотрим некоторые полезные DN, см Таблице 1.

Запись DN	Описание
cn=accounts,dc=ald,dc=company,dc=lan	Контейнер который содержит дочерние записи контейнеров учетных записей, компьютеров, групп и д.р.
cn=computers,cn=accounts,…	Содержит список компьютеры в домене
cn=dns,…	Содержит информацию о записях DNS
cn=groups,cn=accounts,…	Содержит список групп пользователей
cn=hostgroups, cn=accounts,…	Содержит группы компьютеров домена, например ipaservers
cn=orgunits,cn=accounts,…	Содержит список подразделений, которые отображаются у других записей в атрибуте rbtadp, например у пользователя или компьютера
cn=users,cn=accounts,…	Содержит список пользователей домена

Экспортировать объекты можно, нажав контекстное меню по требуемой записи, например, `cn=users` в дереве см. Рис 9, а затем Export и нужный формат рассмотрим CSV.

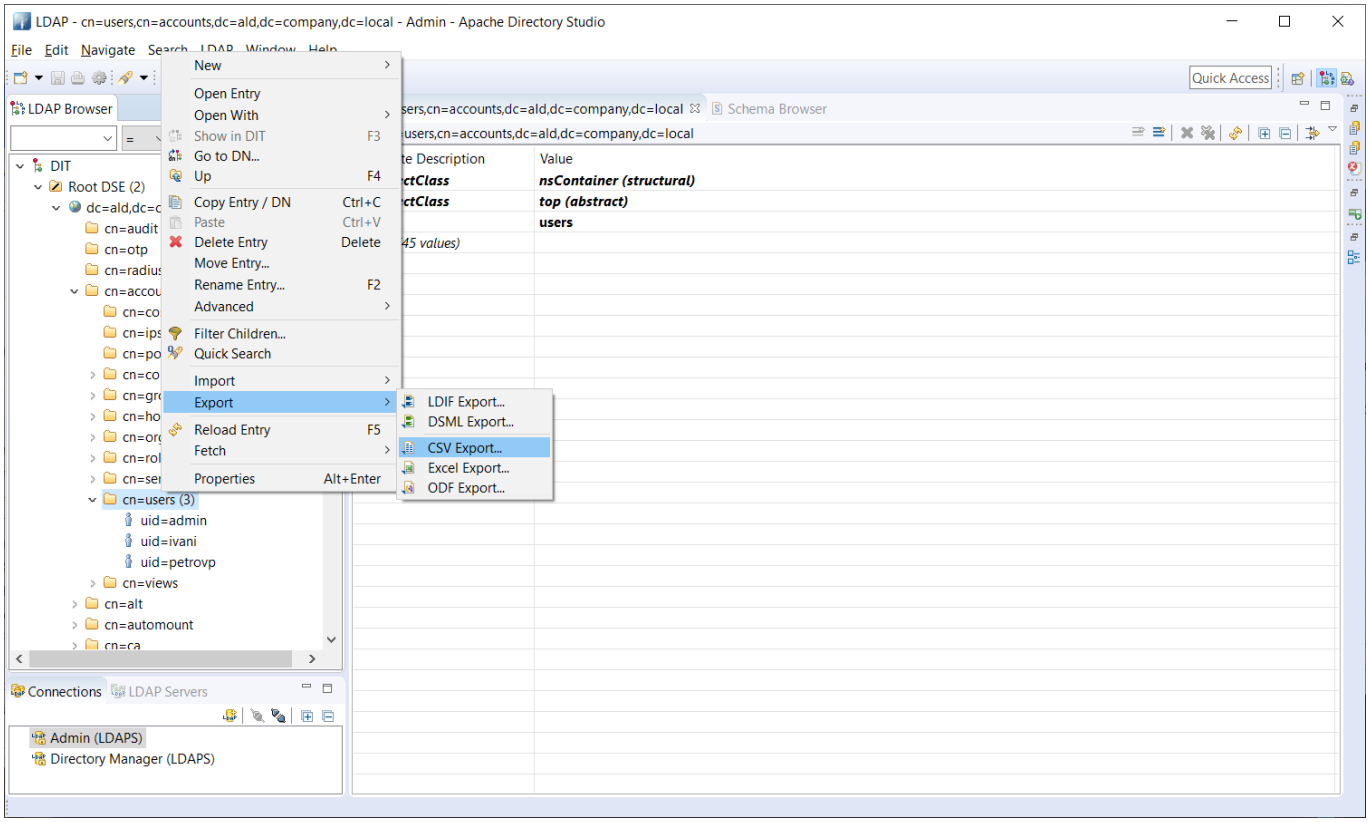


Рисунок 6.30 – Окно LDAP browse

После выбора формата CSV мы видим диалог настроек параметров экспорта данных, см Рис. 8. В окне параметров настройте нужные фильтры и список требуемых атрибутов для вывода.

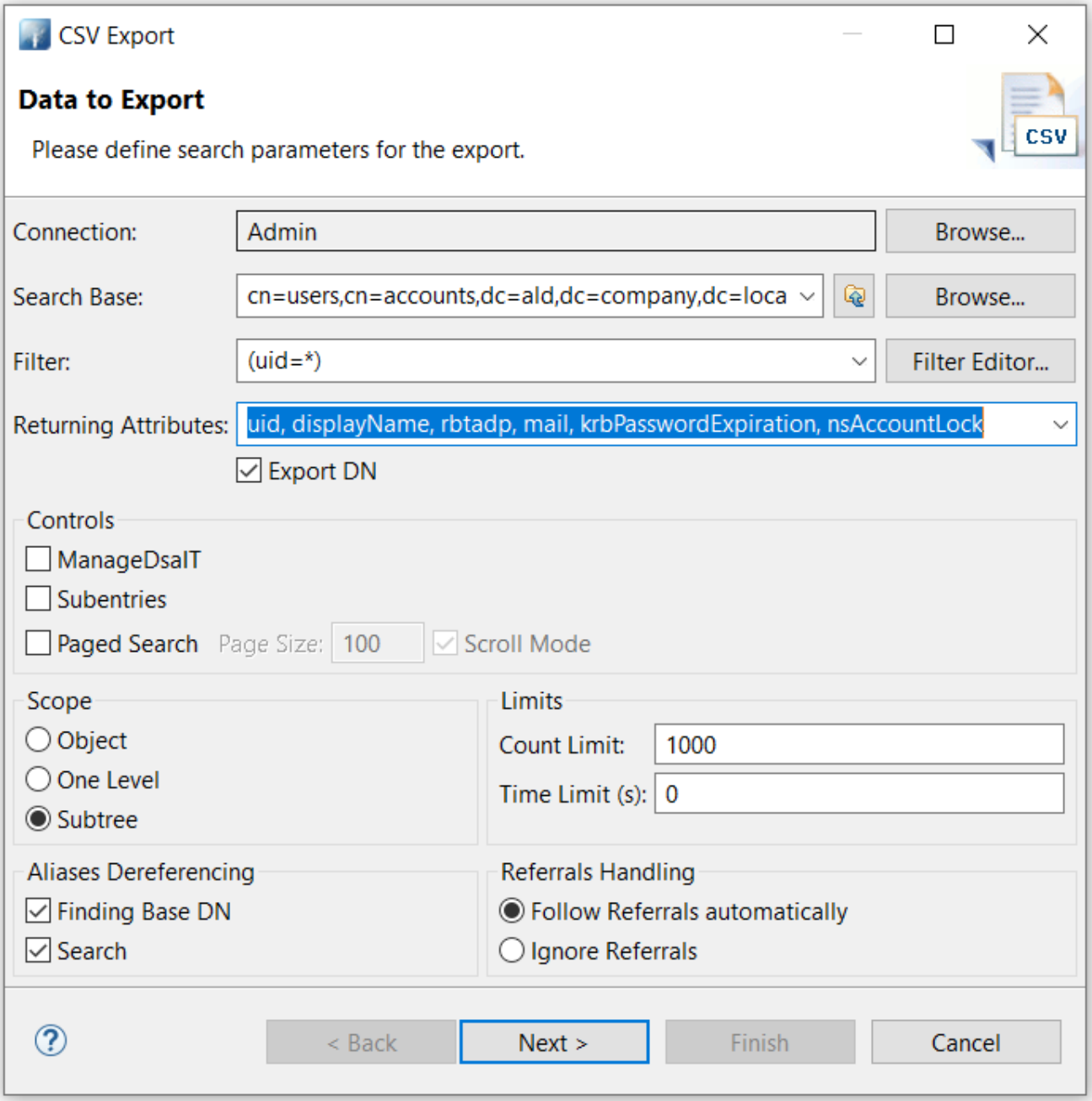


Рисунок 6.31 – Окно параметров CSV Export

Следующим шагом укажите имя файла, в который вы хотите записать CSV, см. Рис 11, а затем нажмите Finish.

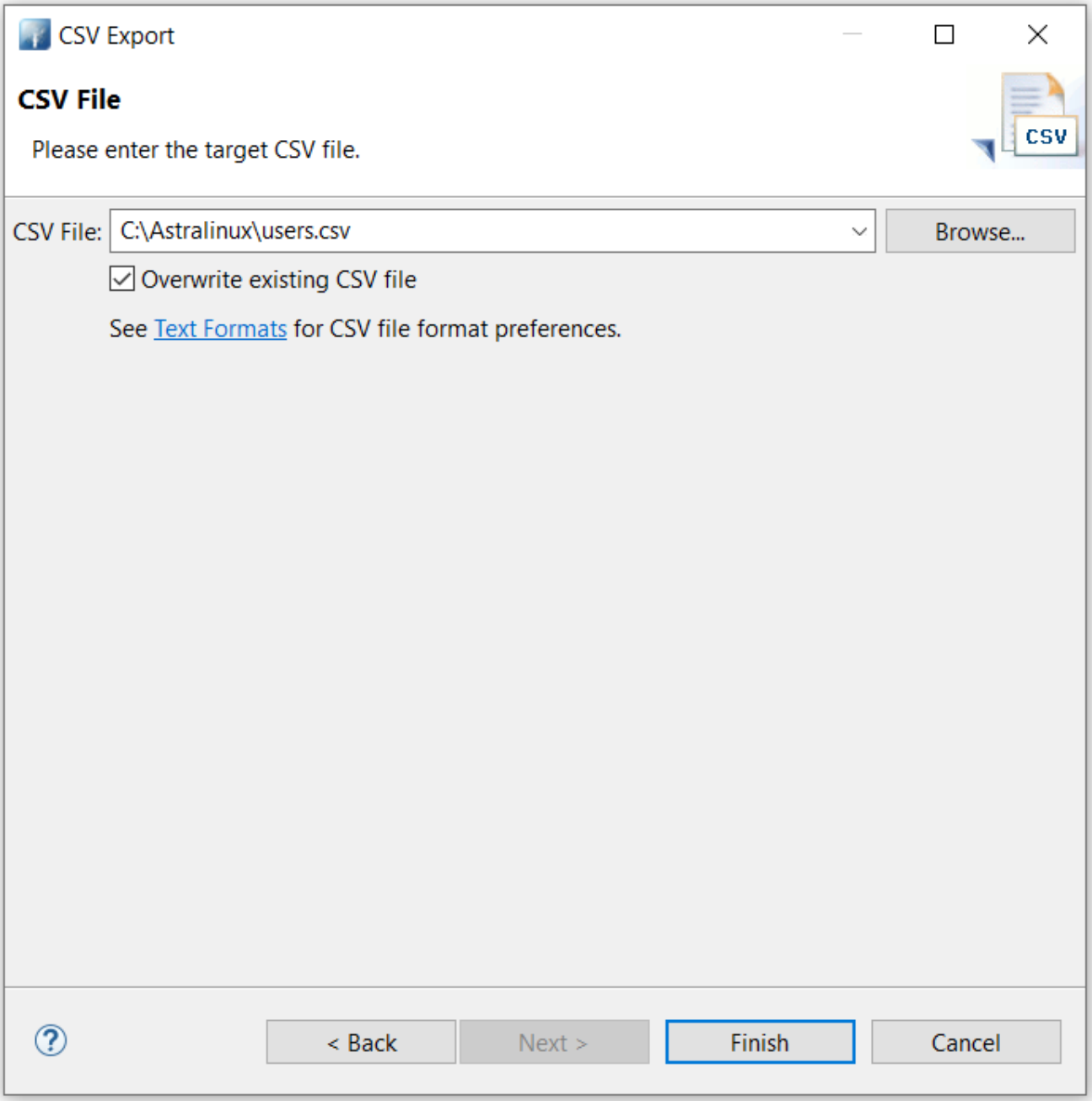


Рисунок 6.32 – Окно CSV Export выбор имени файла

Если нажать на ссылку Text Formats см. Рис. 12, в которой описаны настройки генерации текстовых данных, таких как разделитель, обрамление данных кавычками, разделитель строк и др.

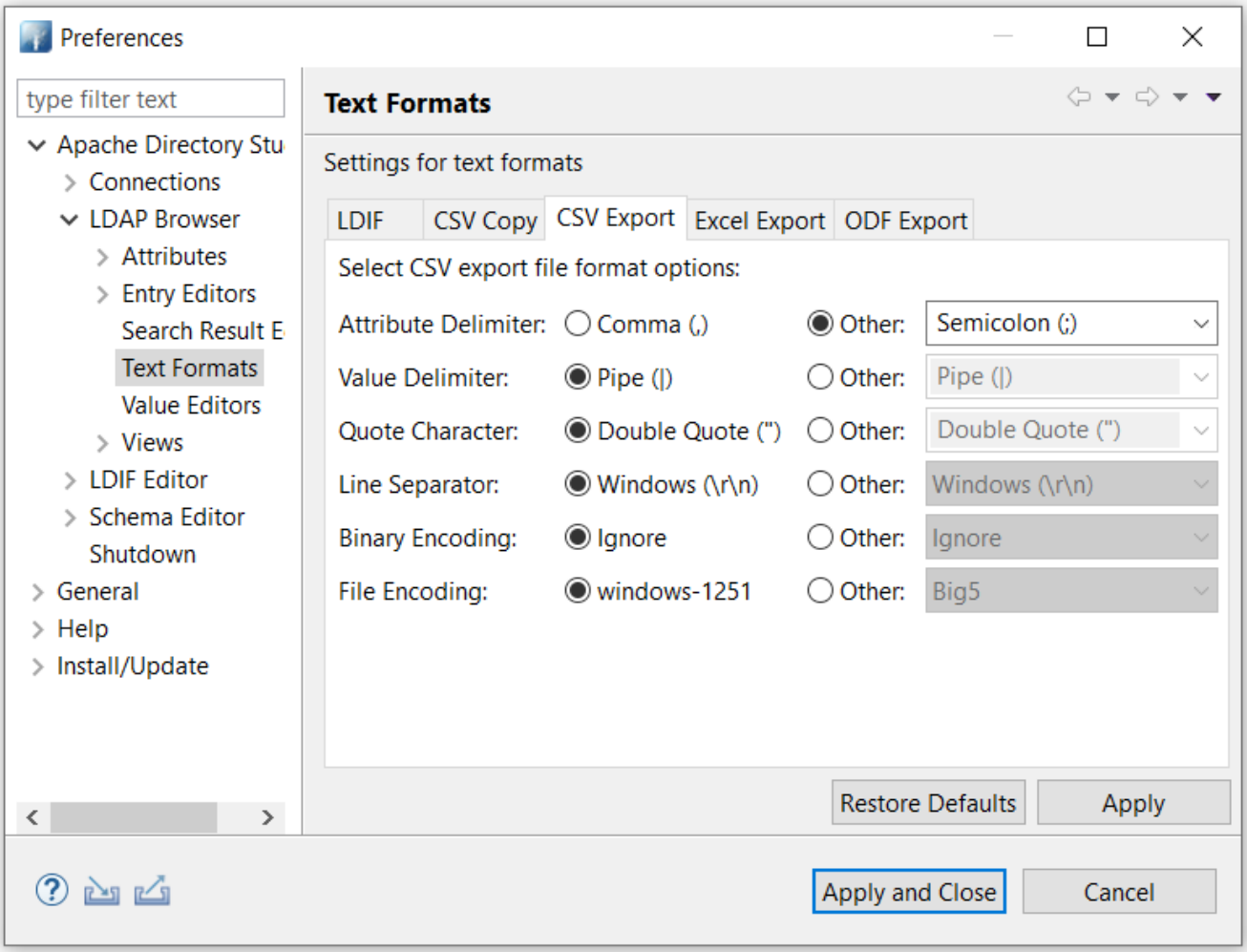


Рисунок 6.33 – Настройки вывода текстового форматирования

Откроем результат выполнения экспорта данных в программе LibreOffice Calc см. Рис. 13.

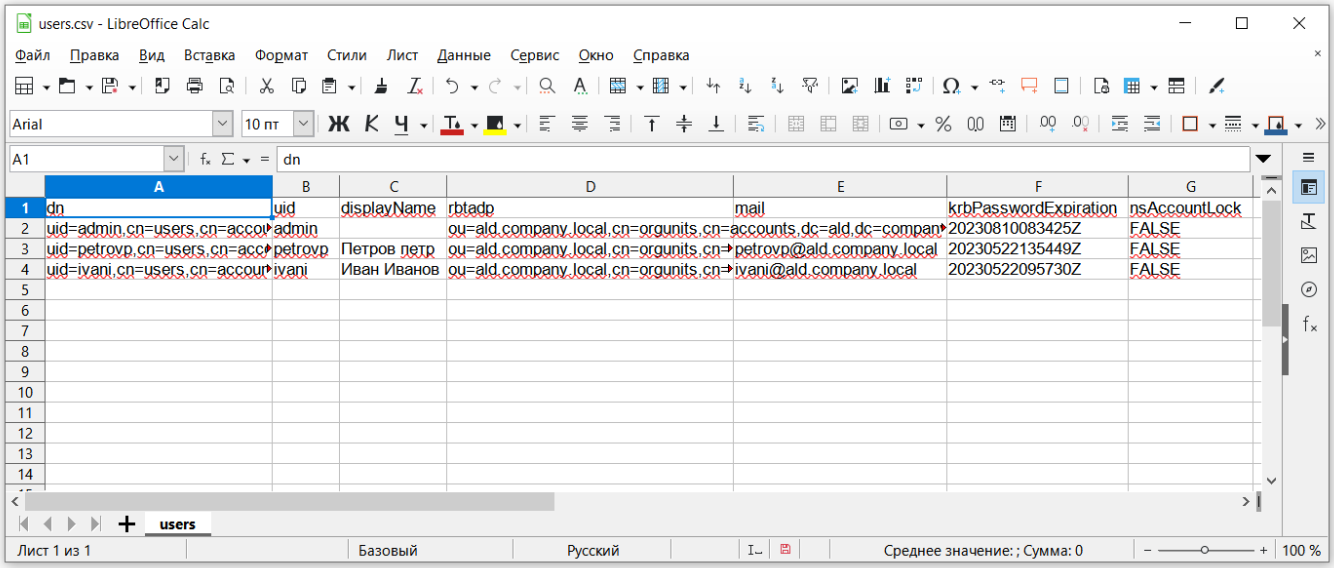


Рисунок 6.34 – Результат выполнения экспорта данных

Вы можете использовать этот файл для автоматизации как входную информацию или подготовить любой отчет по объектам из каталога LDAP.

Просмотр схемы каталога

Для просмотра схемы каталога выберите Root DSE в дереве LDAP Browser и выполните команду «LDAP > Open Schema Browser» см. Рис 14.

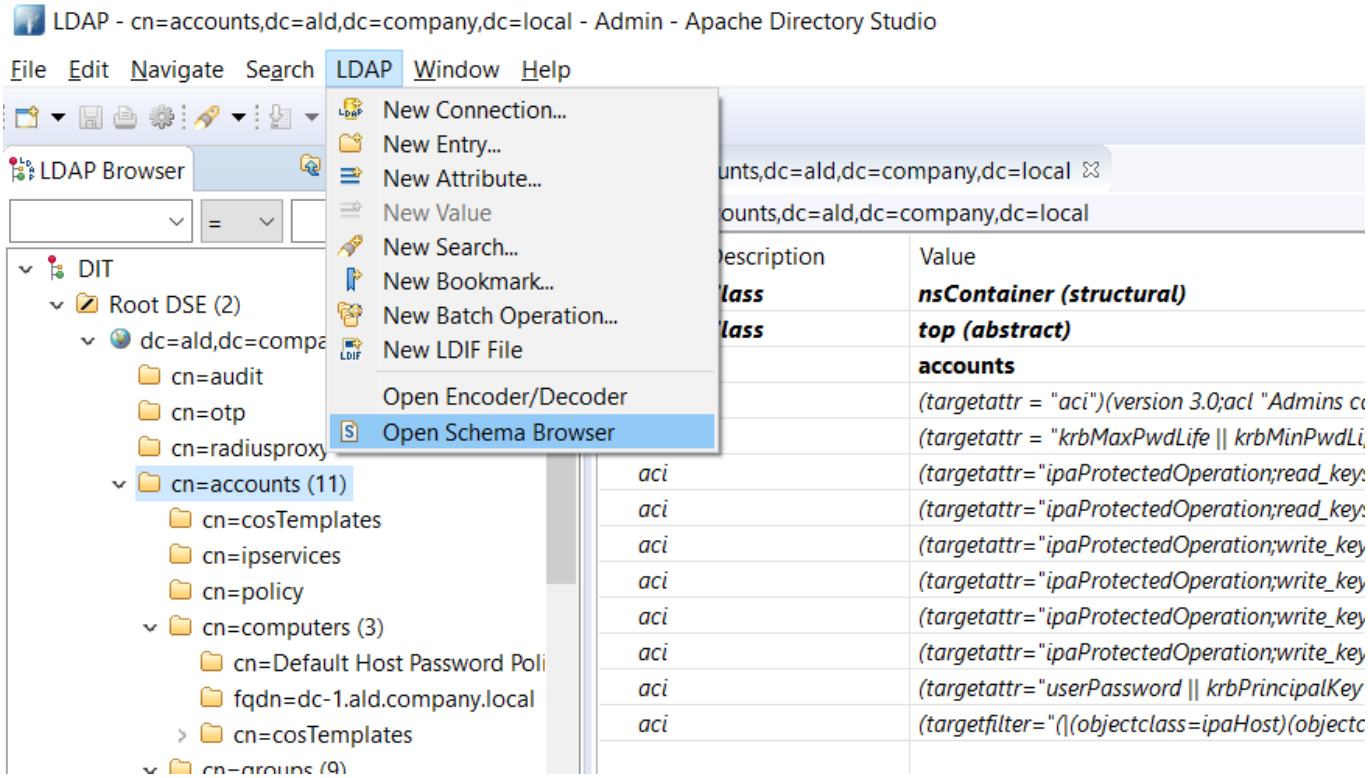


Рисунок 6.35 – Меню Open Schema Browser

По умолчанию в Schema Browser открывается страница для просмотра классов объектов (Object Classes). Вы можете делать поиск и просматривать детали (Details). Например, введите в фильтр «posix» и вы найдете обсуждаемые ранее классы posixAcccount и posixGroup, см. Рис 15.

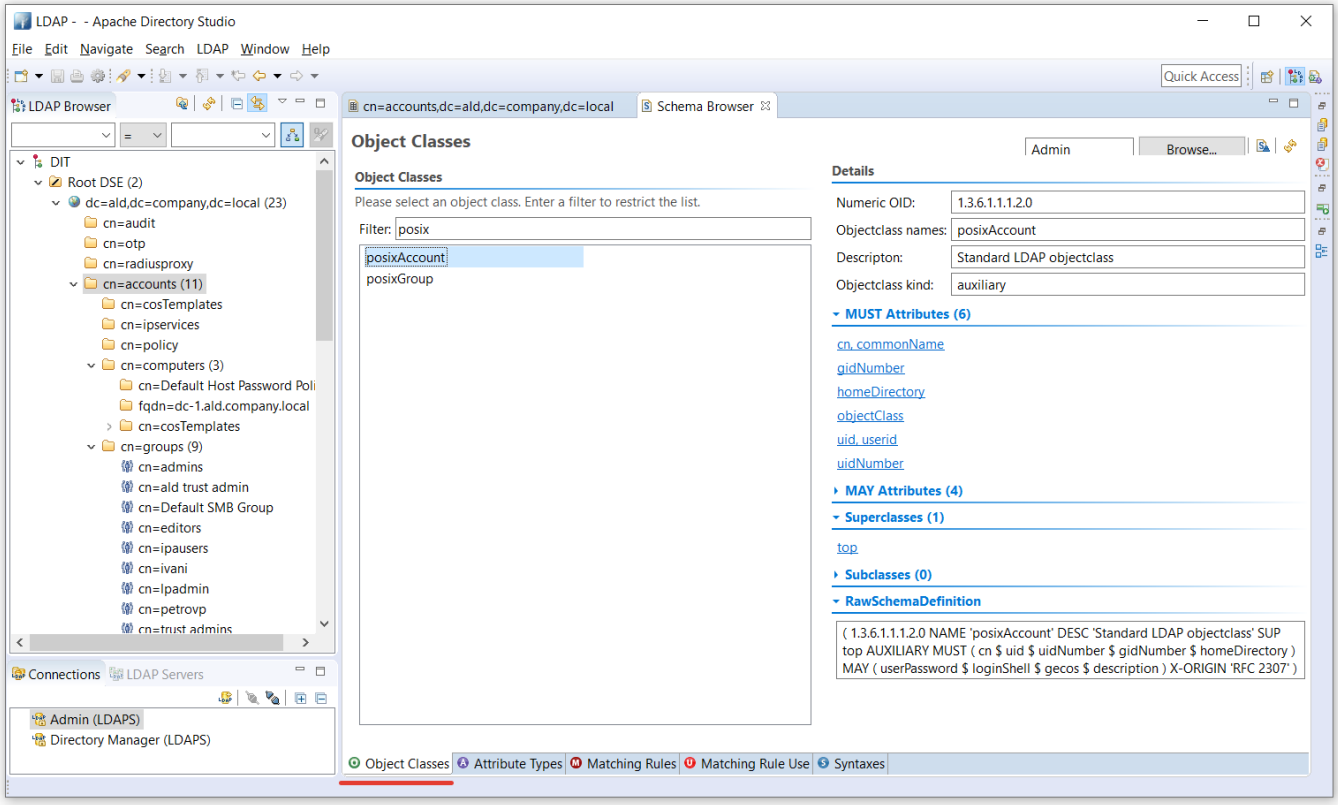


Рисунок 6.36 – Schema Browser и вкладка Object Classes

Для перехода к другим группам объектов используйте ярлычки в нижней части вкладки. Для просмотра атрибутов откройте страницу Attribute Types. В этом окне можно искать по списку всех атрибутов и просматривать детали выбранного атрибута. Например, введите в поле фильтра gid и вы увидите несколько атрибутов, в именах которых содержится эта подстрока, см. Рис 16.

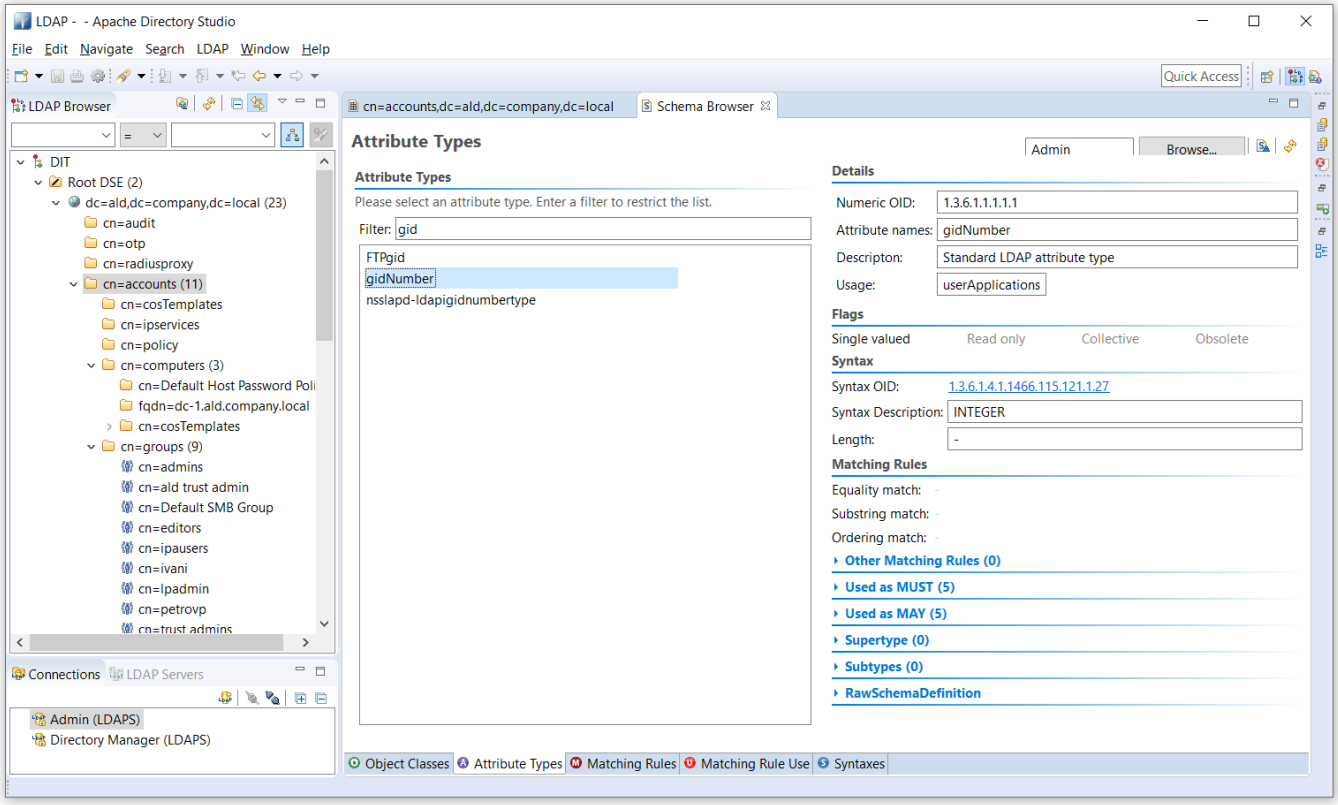


Рисунок 6.37 – Schema Browser и вкладка Attribute Types

6.4.3.2. Утилиты для работы с LDAP-каталогом

Для работы с LDAP-каталогом из командной строки используются утилиты пакета ldap-utils:

- ldapwhoami - выполняет подключение к каталогу и возвращает имя текущего пользователя;
- ldapsearch - выполняет поиск по каталогу с указанными параметрами;
- ldapadd - позволяет создать новый объект в каталоге;
- ldapdelete - позволяет удалить объект из каталога;
- ldapmodify - позволяет изменить атрибуты существующего объекта в каталоге;
- ldapcompare - позволяет сравнить текущее значение атрибута конкретной записи с желаемым значением и получить результат в формате TRUE/FALSE;
- ldapexop - позволяет выполнять расширенные операции, добавленные разработчиками конкретного LDAP-сервера. Расширенные операции подобны хранимым процедурам SQL, и позволяют расширять возможности взаимодействия с сервером без внесения изменений в LDAP-протокол;
- ldappasswd - позволяет сбросить пароль для учетной записи;

- ldapmodrdn - позволяет переименовывать существующие объекты каталога.

Подключение к LDAP

Подключение к LDAP-каталогу происходит в рамках вызова каждой утилиты. Вы можете задавать параметры подключения в явном виде, или полагаться на настройки по умолчанию, которые описаны в файле `/etc/ldap/ldap.conf`. При вводе компьютера в домен указанный файл настраивается автоматически, подключение будет выполняться к одному из контроллеров домена по протоколу LDAPS с использованием Kerberos-билета из связки ключей.

Проверим наш доступ в каталог командой ldapwhoami:

```
ldapwhoami
```

Результат выполнения:

```
SASL/GSSAPI authentication started
ldap_sasl_interactive_bind_s: Local error (-2)
        additional info: SASL(-1): generic failure: GSSAPI Error:
↪Unspecified GSS failure.  Minor code may provide more information (No
↪Kerberos credentials available (default cache: KEYRING:persistent:1000))
```

Данный результат говорит, что мы не имеем учетных записей в кэше Kerberos. Пробуем выполнить вход kinit admin и проверим повторно ldapwhoami:

```
kinit admin
ldapwhoami
```

Результат выполнения:

```
Password for admin@ALDPRO.DOMAIN:
SASL/GSSAPI authentication started
SASL username: admin@ALDPRO.DOMAIN
SASL SSF: 256
SASL data security layer installed.
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
```

Теперь у нас получилось выполнить запрос к LDAP серверу, т.к. у нас был валидны TGT Kerberos билет и система смогла с его помощью успешно получить сервисный билет на

доступ к LDAP серверу. Проверить текущие билеты в кэше можно командой `klist`:

```
klist
```

Результат выполнения:

```
Ticket cache: KEYRING:persistent:1000:1000
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
22.05.2023 09:58:25  23.05.2023 09:58:01  ldap/dc-1.ald.company.lan@ALD.
 ↪COMPANY.LAN
22.05.2023 09:58:08  23.05.2023 09:58:01  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.
 ↪LAN
```

Как видите, в кеше есть билет ldap: ldap/dc-1.ald.company.lan@ALD.COMPANY.LAN, поэтому мы можем использовать утилиты из пакета ldap-utils.

При подключении к серверу мы можем переопределять любой из параметров по умолчанию, задавая их в явном виде:

```
ldapsearch -H dc-1.ald.company.lan -ZZ -x -W -D "uid=admin,cn=users,
 ↪cn=accounts,dc=ald,dc=company,dc=lan" -b "cn=users,cn=accounts,dc=ald,
 ↪dc=company,dc=lan" '(uid=*)' uid givenName sn
```

где:

Параметр -H: для указания адреса LDAP-сервера, например dc-1.ald.company.lan. Параметр позволяет указать нешифрованный протокол ldap или шифрованный ldaps, например, «ldap://dc-1.ald.company.lan». Вы можете выполнять подключение к LDAP каталогу также через unix-сокет, если приложение выполняется на том же сервере, тогда нужно указать «ldapi://%2fvar%2frun%2fslapd-ALD-COMPANY-LAN.socket»

Параметр -ZZ: это параметр включения шифрованного соединения, где первая Z означает отправку серверу запроса STARTTLS. Если сервер не поддерживает TLS, соединение продолжится, и оно не будет шифроваться. Вторая Z устанавливает требование использовать только шифрованное соединение, если сервер не поддерживает TLS, соединение прервется. Рекомендуется использовать двойной ключ -ZZ для установления шифрованного соединения, если по каким-либо причинам 636 порт недоступен, а также требование использовать безопасный протокол TLS на 636 порту можно задать с помощью ключа -H, указав порт явно ldaps://dc-1.ald.company.lan;

Параметр -x: указывает на необходимость выполнить простую аутентификацию по логину/паролю;

Параметр -D: задает Bind DN пользователя для аутентификации, например, `uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan` или супер пользователя, `cn=Directory Manager`;

Параметр -W: указывает, что пароль должен быть предоставлен в интерактивном режиме (а если нужно передать пароль в параметре, то это можно сделать с помощью ключа -w)

Используя параметры -D и -W, вы можете подключиться к нужному серверу по IP или имени сервера и файл конфигурации задействован не будет.

Поиск по каталогу ldapsearch

Для поиска информации вы можете использовать утилиту **ldapsearch**, которая извлекает из каталога набор записей с указанными атрибутами в соответствии с заданными критериями фильтрации и заданного списка атрибутов. Синтаксис команды в общем виде выглядит следующим образом:

```
ldapsearch [options] [filter [attributes…]]
```

где:

- options - параметры вызова, в т.ч. параметры подключения. Для простоты мы будем использовать параметры подключения по умолчанию;
- filter - служит для точного указания критериев поиска;
- attributes - указывает, какие атрибуты необходимо запросить.

Для примера напишем запрос, который будет выдавать список всех пользователей домена из `cn=users`, которая в свою очередь является потомком записи `cn=accounts`, и все они находятся в пространстве корневого суффикса `dc=ald,dc=company,dc=lan`. Таким образом полным уникальным именем записи (Distinguished name, DN), в которой хранятся пользователи, является `cn=users,cn=accounts,dc=ald,dc=company,dc=lan`.

Выполним поиск по каталогу с помощью команды **ldapsearch**:

```
ldapsearch -Q -s sub -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
↪'(uid=*)' uid givenName sn
```

где:

- Параметр **-Q** – это тихий режим SASL, не выводится информация о SASL подключении, который полезен при автоматизации для очитки данных от технической информации.

- **Параметр -s – это область поиска (scope). Может принимать следующие значения:**
  - one - поиск идет по дочерним записям на один уровень ниже в иерархии
  - sub - поиск идет по всем дочерним записям на всю глубину иерархии, параметр по умолчанию
  - children - то же, что и sub, но ограничивает поиск только дочерними записями
  - base - ограничивает поиск по текущей записи, заданной параметром -b. Если задан one, поиск идет по дочерним записям на один уровень ниже в иерархии. Если задан sub, то поиск идет по всем дочерним записям на всю глубину иерархии, начиная с записи, заданной параметром -b, при этом включая саму базовую запись. children - то же, что и sub, но ограничивает поиск только дочерними записями, не включая базовую запись.

- Параметр **-b** – это базовая запись (base), которая будет использоваться в качестве начальной точки для поиска по дереву.
- Параметр **“(uid=*)“** – это фильтр, в котором мы ищем все записи, имеющие атрибут uid. Фильтры и составные фильтры мы рассмотрим далее.
- Параметры **uid givenName sn** – это атрибуты, которые нужно вывести в результат. Если их не указывать, то будут отображены все атрибуты найденных записей.

Результат поиска по каталогу:

```
### extended LDIF
#
### LDAPv3
### base <cn=users,cn=accounts,dc=ald,dc=company,dc=lan> with scope subtree
### filter: (uid=*)
### requesting: uid givenName sn
#

### admin, users, accounts, ald.company.lan
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: admin
sn: Administrator
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
### petrovp, users, accounts, ald.company.lan
dn: uid=petrovp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: petrovp
givenName:: 0J/RkdGC0YA=
sn:: 0J/QtdGC0YDQvtCy

### ivani, users, accounts, ald.company.lan
dn: uid=ivani,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: ivani
givenName:: 0JjQstCw0L0=
sn:: 0JjQstCw0L3QvtCy

### search result
search: 4
result: 0 Success

### numResponses: 4
### numEntries: 3
```

Результат выводится в поток **stdout** в текстовом формате LDIF и его можно использовать по конвейеру pipeline или перенаправить в файл для дальнейшей обработки. Подробнее о формате LDIF мы поговорим в разделе 2.2.3.

Существуют операционные атрибуты (operational attributes), которые встроены в LDAP сервер и управляются им самим, например, **entrydn, entryid, parentid или nsAccountLock**. Большинство операционных атрибутов для пользователей доступны только для чтения. Чтобы их увидеть необходимо указать **“+”** в конце команды, однако не все операционные атрибуты доступны для просмотра таким образом:

```
ldapsearch -Q -LLL -s base -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan
↪" "+"
```

Результат выполнения:

```
dn: cn=users,cn=accounts,dc=ald,dc=company,dc=lan
creatorsName: cn=Directory Manager
modifiersName: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
createTimestamp: 20230324160645Z
```

(продолжение на следующей странице)

```
modifyTimestamp: 20230324161206Z
nsUniqueId: df91d884-ca5d11ed-b5f0ee72-e6acffe1
parentid: 2
entryid: 3
entryusn: 6055
numSubordinates: 6
```

Например, мы видим, что запись **“cn=users”** имеет операционный атрибут **numSubordinates**, который показывает число дочерних записей.

Фильтры запроса

Рассмотрим, как сделать поиск ldapsearch более гибким с помощь фильтров, задав условие для отбора нужной нам информации. Синтаксис фильтра выглядит следующим образом:

(attribute operator value)

В Таблице 2 приведены операторы (operator), используемые в фильтрах:

Опе-ратор	Комментарий
>=	Больше или равно. Возвращает результат, если атрибут больше или равен какому-то значению, например (uidNumber>=180003)
<=	Меньше или равно. Возвращает результат, если атрибут меньше или равен какому-то значения, например: (uidNumber<=180003)
=*	Наличие. Возвращает результат, если атрибут содержит одно или более значений, например: (cn=*)
=	Равенство. Возвращает результат, если атрибут равен некоторому значению, например: (sn=petrov)
~=	Примерное равенство. Возвращает результат, если атрибут содержит приблизительно схожее значение, например: (cn~=petrof)
=string*	Включает в себя. Возвращает значение, если атрибут содержит определенную подстроку. Где знак «*» *означает ноль или более символов, например: (cn=pet*ov)

Например, выведем список пользователей с фамилией petrov. Для этого нам потребуется указать фильтр по атрибуту sn (surname):

```
ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
↪"(sn=Петров)" cn
```

Результат выполнения:

```
dn: uid=ppetrov,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
cn: Petr Petrov
```

В результате запрос отфильтрован по атрибуту sn. Таким образом мы можем подготовить любые данные для обработки.

Составные фильтры запроса

Для большей гибкости вы можете использовать несколько фильтров, объединяя их с помощью логических операторов. Синтаксис составного фильтра выглядит следующим образом:

```
(Boolean-operator(filter)(filter)(filter)...)
```

В Таблице 3 приведены логические операторы (Boolean-operator), используемые в составных фильтрах:

&	Логическое И. Фильтр возвращает те записи, которые удовлетворяют всем указанным условиям, например: (&(uid=user)(uidNumber=180003))
|	Логическое ИЛИ. Фильтр возвращает те записи, которые удовлетворяют одному из указанных условий, например:
!	Логическое НЕ. Фильтр возвращает те записи, которые не удовлетворяют указанному условию, например: (!(uid=admin))

Например, найдем пользователей из группы admins, которые два и более месяца не меняли пароль:

```
ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
↪"(&(memberof=cn=admins*)(krbLastPwdChange>=$(date +"%Y%m%d000000Z" -d "-60
↪days")))" cn krbLastPwdChange
```

Результат выполнения:

```
dn: uid=admin,cn=users,cn=accounts,
cn: Administrator
krbLastPwdChange: 20230324160900Z
```

В результате запроса, мы видим список администраторов, у которых пароль изменялся за последние 60 дней, обратите внимание на то, что можно добавлять подкоманды через конструкцию: $(date +“%Y%m%d000000Z” -d “-60 days”) между двойными кавычками фильтра. Так мы вычислим дату и подставим ее в фильтр в нужном формате.

Получение схемы объектных классов

Объектные классы описаны в операционном атрибуте objectClasses из специальной записи схемы `cn=schema`. Посмотреть список всех классов можно командой:

```
ldapsearch -Q -LLL -o ldif-wrap=no -s base -b "cn=schema" objectClasses
```

Результат выполнения:

```
dn: cn=schema
objectClasses: ( 2.5.6.0 NAME 'top' ABSTRACT MUST objectClass X-ORIGIN 'RFC
 ↪45
 12' )
objectClasses: ( 2.5.6.1 NAME 'alias' SUP top STRUCTURAL MUST
 ↪aliasedObjectNam
 e X-ORIGIN 'RFC 4512' )
objectClasses: ( 2.5.20.1 NAME 'subschema' AUXILIARY MAY ( dITStructureRules
 ↪$
  nameForms $ dITContentRules $ objectClasses $ attributeTypes $
 ↪matchingRules
  $ matchingRuleUse ) X-ORIGIN 'RFC 4512' )
objectClasses: ( 1.3.6.1.4.1.1466.101.120.111 NAME 'extensibleObject' SUP top
 AUXILIARY X-ORIGIN 'RFC 4512' )
objectClasses: ( 2.5.6.11 NAME 'applicationProcess' SUP top STRUCTURAL MUST
 ↪cn
  MAY ( seeAlso $ ou $ l $ description ) X-ORIGIN 'RFC 4519' )
...
```

В результате выведется список всех классов, в котором можно искать через команду `grep`:

```
ldapsearch -Q -LLL -o ldif-wrap=no -s base -b "cn=schema" objectClasses |
 ↪grep --color posix
```

Результат выполнения:

```
objectClasses: ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' DESC 'Standard LDAP
 ↪objectclass' SUP top AUXILIARY MUST ( cn $ uid $ uidNumber $ gidNumber $
 ↪homeDirectory ) MAY ( userPassword $ loginShell $ gecos $ description ) X-
 ↪ORIGIN 'RFC 2307' )
objectClasses: ( 1.3.6.1.1.1.2.2 NAME 'posixGroup' DESC 'Standard LDAP
 ↪objectclass' SUP top STRUCTURAL MUST ( cn $ gidNumber ) MAY ( userPassword
 ↪$ memberUid $ description ) X-ORIGIN 'RFC 2307' )
objectClasses: ( 2.16.840.1.113730.3.2.326 NAME 'dynamicGroup' DESC 'Group
 ↪containing internal dynamically-generated members' SUP posixGroup
 ↪AUXILIARY MAY dsOnlyMemberUid X-ORIGIN 'Red Hat Directory Server' )
```

Получение схемы атрибутов

Также, как и объектные классы, атрибуты описаны через операционный атрибут «attributeTypes» корневой записи схемы `cn=schema`. Посмотреть список всех атрибутов можно командой:

```
ldapsearch -Q -LLL -o ldif-wrap=no -s base -b "cn=schema" attributeTypes
```

Результат выполнения:

```
dn: cn=schema
attributeTypes: ( 2.16.840.1.113730.3.1.582 NAME 'nsDS5ReplicaCredentials'
 ↪DES
 C 'Netscape defined attribute type' SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 ↪SINGL
 E-VALUE X-ORIGIN 'Netscape Directory Server' )
attributeTypes: ( 2.16.840.1.113730.3.8.22.1.2 NAME 'ipaCertMapMapRule' DESC
 ↪'
 Certificate Mapping Rule' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 ↪X
 -ORIGIN 'IPA v4.5' )
attributeTypes: ( 1.3.6.1.4.1.15953.9.1.1 NAME 'sudoUser' DESC 'User(s) who
 ↪ma
 y  run sudo' EQUALITY caseExactIA5Match SUBSTR caseExactIA5SubstringsMatch
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
 ↪SY
 NTAX 1.3.6.1.4.1.1466.115.121.1.26 X-ORIGIN 'SUDO' )
...
```

Результат команды также можно перенаправить в файл командой перенаправления «>»:

```
ldapsearch -Q -LLL -o ldif-wrap=no -s base -b "cn=schema" attributeTypes >
 ↪attrs
```

Формат данных LDIF

Как упоминалось ранее, утилита ldapsearch возвращает результаты в формате LDIF, LDAP Data Interchange Format – это формат представления записей службы каталогов или их изменений в текстовой форме. Записи каталога или их изменения представляются набором LDIF-записей, по одной на каждую запись каталога или изменение. LDIF был разработан в начале 90-х годов и был доработан для использования с LDAPv3. Описание формата опубликовано в RFC 4525. Рассмотрим пример структуры LDIF:

dn: уникальное имя

имя атрибута: значение атрибута

имя атрибута:: base64 значение атрибута

dn: уникальное имя

имя атрибута: значение атрибута

имя атрибута:< значение атрибута url

Записи каталога представляются группами строк, разделенных пустой строкой, при этом каждая строка в группе представляет отдельное значение атрибута записи. Первая строка в группе должна представлять уникальное имя записи DN. Значение атрибута записывается в кодировке ASCII и отделяется от его имени символом «:». Значения, не подходящие под эту кодировку, записываются в кодировке base64 и отделяются от имени атрибута символами «::». Данный формат можно использовать для хранения данных,

добавления и модификации записей в каталоге. Рассмотрим пример LDIF файла добавления записи в каталог нового подразделения add_dp.ldif с таким содержанием:

```
dn: ou=marketing,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
 ↪dc=lan
objectClass: rbta-org-unit
objectClass: top
ou: marketing
displayName: Маркетинг
```

Примечание: В конце LDIF стоит пустая строка, которая определяет разделение между записями. Даже если запись одна нужно ставить разделитель из пустой строки.

При модификации используется директива changetype, которая может принимать значения: add, modify, replace, delete и moddn. Более подробно мы рассмотрим в примере **ldapmodify**.

Добавление записи в каталог через ldapadd

Добавление в каталог информации можно утилитами ldapadd и ldapmodify, где ldapadd это символическая ссылка на команду ldapmodify, поэтому можно использовать ldapmodify -a. Давайте добавим новое подразделение через описанный выше LDIF файл add_dp.ldif командой перенаправления <:

```
Ldapadd -Q < add_dp.ldif
```

Результат выполнения:

```
adding new entry "ou=marketing,cn=orgunits,cn=accounts,dc=ald,dc=company,
 ↪dc=lan"
```

В результате выводится сообщение adding new entry и DN новой записи.

Второй способ добавления – это работа с утилитой в интерактивном режиме для этого запускаем команду `ldapadd`:

```
ldapadd -Q
```

Далее напечатаем или вставим из буфера обмена текст записи в формате LDIF, в конце которого должна быть пустая строка:

```
dn: ou=develop,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
 ↪dc=lan
objectClass: rbta-org-unit
objectClass: top
ou: develop
displayName: Разработка ПО
```

Как мы видим курсор ждет ввода символов или сигналов, поэтому оправим терминалу сигнал EOF командой `Ctrl+`.

```
Ctrl + <d>
```

Результат выполнения:

```
objectClass: rbta-org-unit
objectClass: top
ou: Develop
displayName: Разработка ПО
adding new entry "ou=Develop,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan
 ↪"
```

Программа сообщит об успешном добавлении «adding new entry». Кстати, для выхода без изменений можно отправить другой сигнал SIGINT Ctrl +. Использовать интерактивный режим для скриптов не рекомендуется, потому что может произойти событие ожидания ввода и скрипт зависнет, ожидая ввода пользователя. Поэтому в написании скриптов для команд используют параметр -y для подтверждения всех запросов. Далее мы будем использовать первый подход с перенаправлением.

Модификация записей в каталоге через ldapmodify

Как мы уже говорили есть директива **changetype**, она должна следовать сразу после **dn**. Это нужно для того, чтобы утилита понимала, которую запись мы изменяем. Если в LDIF отсутствует директива **changetype**, то по умолчанию подразумевается **«changetype: add»** добавление записи в каталог.

Директивы changetype: modify и add:

Данная директива LDIF позволяет добавить атрибут к записи, но есть атрибуты, которые могут быть только в единичном числе, например, атрибут **displayName**, то при добавлении второго атрибута возникнет ошибка «ldap_add: Already exists (68)”. В качестве примера добавим к нашему пользователю admin новую локацию атрибут «L» файл **add_loc.ldif**:

```
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
changetype: modify
add: l
l: Казань
```

Обратите внимание на пустую строку - это разделитель между записями и ее нужно добавлять даже мы оперируем с одной записью. А затем запустим команду модификации **ldapmodify**:

```
ldapmodify -Q < add_loc.ldif
```

Или можно передать по конвейеру

```
cat add_loc.ldif  | ldapmodify -Q
```

Результат выполнения:

```
modifying entry "uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
```

Мы видим успешное изменение записи

Директивы changetype: modify и replace:

Переместим пользователя в другой департамент изменив ему атрибут rbtadb. Создадим файл `changedp.ldif`:

```
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
changetype: modify
replace: rbtadp
rbtadp: ou=marketing,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,
 ↪dc=company,dc=lan
```

А затем запустим команду модификации `ldapmodify`:

```
ldapmodify -Q < changedp.ldif
```

Результат выполнения:

```
modifying entry "uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
```

Мы видим успешное изменение записи, которое можно проверить в Apache Directory Studio.

Директивы changetype: modify и delete:

Если нам необходимо изменить несколько атрибутов у записи, то можно несколько операций разделить с новой строки символом «-». Пример изменений нескольких атрибутов удаляет атрибут «l: Казань» и добавляет «l: Москва». Создадим файл change_two_attrs.ldif:

```
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
changetype: modify
delete: l
l: Казань
-
add: l
l: Москва
```

Запустим команду ldapmodify, указав файл

```
change_two_attrs.ldif через параметр -f:
ldapmodify -Q -f change_two_attrs.ldif
```

Результат выполнения:

```
modifying entry "uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
```

Мы видим успешное изменение записи.

Директивы changetype: modrdn и newrdn:

Директива modrdn изменяет RDN записи, другими словами переименовывает запись. Например, переименуем подразделение Разработки ПО `ou=develop` на `ou=arch` архитекторов через файл `rename_rdn.ldif`, в котором директива `deleteoldrdn: 1` удалит старый DN:

```
dn: ou=develop,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
 ↪dc=lan
changetype: modrdn
newrdn: ou=arch
deleteoldrdn: 1

dn: ou=arch,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
 ↪dc=lan
changetype: modify
replace: displayName
displayName: Архитекторы
```

Запустим команду ldapmodify, перенаправив файл

```
rename_rdn.ldif:
ldapmodify -Q < rename_rdn.ldif
```

Результат выполнения:

```
modifying rdn of entry "ou=develop,ou=ald.company.lan,cn=orgunits,cn=accounts,
 ↪dc=ald,dc=company,dc=lan"

modifying entry "ou=arch,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,
 ↪dc=company,dc=lan"
```

Мы видим успешное переименование RDN и изменение атрибута displayName для нового подразделения.

Важно: При использовании директивы `modrdn` есть угроза возникновения ошибки в виде некорректного количества записей (`total`).

Директива changetype: delete

Если нам необходимо удалить запись из каталога, то используем директиву changetype: delete, однако, у записи не должно быть ни одной дочерней записи. Например, удалим группу «ou=arch» через файл `del_ou.ldif`:

```
dn: ou=arch,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
```

(продолжение на следующей странице)

```
 ↪dc=lan
changetype: delete
```

Проверим результат, запустив команду:

```
ldapmodify -Q < del_ou.ldif
```

Результат выполнения:

```
deleting entry "ou=arch,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,
 ↪dc=company,dc=lan"
```

Мы видим успешное удаление записи, а если запись будет не найдена, то выведется сообщение об ошибке:

```
ldap_delete: No such object (32)
        matched DN: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,
 ↪dc=company,dc=lan
```

Работа с кириллицей и Unicode

Если атрибуты содержат значения не в ASCII кодировке, то утилита ldapsearch при выводе значений представляет их в base64 кодировке. В нашем примере, при создании пользователя ppetrov атрибуту displayName было присвоено значение “Петров П.П.”. Посмотрим на вывод команды ldapsearch:

```
ldapsearch -Q -LLL -s base -b "uid=ppetrov,cn=users,cn=accounts,dc=ald,
 ↪dc=company,dc=lan" displayname
```

Результат выполнения:

```
dn: uid=petrov.pp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
displayname:: 0J/QtdGC0YDQvtCyINCfLtCfLg==
```

Поскольку ldapsearch понимает только ASCII символы, она представила значение displayName в неудобной для чтения base64 строке. Мы сможем увидеть исходное значение, только сделав обратное преобразование:

```
echo '0J/QtdGC0YDQvtCyINCfLtCfLg==' | base64 -d
```

Результат выполнения:

```
Петров П.П.
```

Для того, чтобы получить результат сразу в удобном для автоматизации виде, можно использовать команду:

```
ldapsearch -Q -LLL -s base -b "uid=ppetrov,cn=users,cn=accounts,dc=ald,
 ↪dc=company,dc=lan" displayname  | perl -MMIME::Base64 -Mutf8 -pe 's/^([-a-
 ↪zA-Z0-9;]+):(:\s+)(\S+)$/$1.$2.&decode_ba    se64($3)/e'
```

Результат выполнения:

```
dn: uid=petrov.pp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
displayName: Петров П.П.
```

Отметим, что не требуется производить какие-то дополнительные действия в фильтрах при поиске по строкам, не содержащим ASCII символы:

```
ldapsearch -Q -LLL -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
 ↪"(displayName=Петров П.П.)" cn
```

Результат выполнения:

```
dn: uid=petrov.pp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
cn: Petr Petrov
```

6.4.4. Примеры автоматизации

В предыдущих разделах объясняется, как можно управлять каталогом через LDAP-протокол, это своего рода «кирпичики», из которых можно «строить дома», т.е. решать сложные задачи автоматизации. Давайте рассмотрим примеры на языках bash и python, а также поработаем с форматами CSV, LDIF, JSON.

6.4.4.1. Работа с объектами из командной строки bash

В операционных системах Microsoft Windows для решения задач автоматизации раньше использовали командную оболочку и интерпретатор командной строки CMD. На замену ему пришел более мощный объектно-ориентированный PowerShell, который предоставляет удобные инструменты для работы с датами, хеш-таблицами, словарями, CSV и JSON объектами, а также имеет множество дополнительных модулей, например, для взаимодействия с каталогом Active Directory, объектной моделью офисных приложений и пр.

В мире Linux есть множество оболочек и интерпретаторов, например, sh, bash, zsh и др., каждый из которых предлагает свой вариант синтаксиса и может вызывать утилиты, доступные в операционной системе, например, ls, ping, sed, cat, cut и др. Но все они в какой-то степени являются эквивалентом CMD, т.к. работают только с текстовыми переменными и не поддерживают объекты.

Обрабатывать текстовый поток, получаемый от других приложений, средствами командной строки довольно сложно, поэтому bash целесообразно дополнить возможностями языка программирования Python, который в мире Linux можно считать эквивалентом PowerShell, по крайней в части удобства работы с объектами. Мы покажем вам примеры скриптов Python для конвертации LDIF в JSON/CSV, и как с этими данными можно дальше работать с помощью утилит jq/cut.

Конвертер ldif2csv и генерация отчетов

Обрабатывать LDIF довольно сложно, т.к. данные представлены в блоках строк, а нужные атрибуты могут отсутствовать. Вот пример Python скрипта, который позволяет конвертировать поток LDIF в CSV, для его использования создайте файл ldif2csv.py и скопируйте туда следующее содержимое:

```
#!/usr/bin/python3
import sys
import base64
import re

if sys.version_info[0] < 3:
  raise Exception("Use Python 3: python3 ldif2csv.py")

### read from sdtin
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
data = sys.stdin.readlines()
header_string = ""
atrcheck = ""
entry = ""
dic_entries = {}
dic_entry = {}
current_dn = ""
val_base64 = False
max_size_cell = 32000  ### limit chars in excel cell
### main loop for parse headers and collect dict
for line in data:
  ln = line.replace("\n", "").replace("\r", "")
  if len(ln) == 0:
    if not current_dn == "":
      dic_entries[current_dn] = dic_entry
      dic_entry = {}
      entry = ""
  else:
    if ln.startswith("version"):
      #dic_entries["version"] = ln.split(": ")[1]
      continue
    elif ln.lstrip()[0] == "#":
      continue
    elif ln[0] == " ":
      ### if line wrapped line starts with " " then add line to last attr
      dic_entry[atrcheck] += ln.lstrip()
      if val_base64:
        val_to_decode = dic_entry[atrcheck]
        try:
          dic_entry[atrcheck] = base64.b64decode(val_to_decode).decode(
            'utf-8').strip()
        except:
          dic_entry[atrcheck] = val_to_decode
      continue
    entry += ln
    attribute = []
    attribute_name = ""
    attribute_value = ""
    if ln.find(":: ") > 0:
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
      val_base64 = True
      attribute = ln.split(":: ")
      try:
        attribute_value = base64.b64decode(
          attribute[1]).decode('utf-8').strip()
      except:
        attribute_value = attribute[1]
    elif ln.find(":< ") > 0:
      val_base64 = False
      attribute = ln.split(":< ")
      attribute_value = attribute[1]
    else:
      val_base64 = False
      attribute = ln.split(": ")
      try:
        attribute_value = re.sub(r"^.*?: ", "", ln)
      except:
        attribute_value = ""

    atrcheck = attribute[0].replace(":", "")
    ### get attribute and check if attribute exist

    if dic_entry.get(atrcheck):
      new_len = len(dic_entry[atrcheck]) + len("|" + str(attribute_value))
      if new_len < max_size_cell:
        dic_entry[atrcheck] = str(
          dic_entry[atrcheck]) + "|" + str(attribute_value)
    else:
      dic_entry[atrcheck] = attribute_value
    if atrcheck == "dn":
      current_dn = attribute[1]
  if header_string.find(atrcheck) < 0:
    if header_string == "":
      header_string += atrcheck
    else:
      header_string += ";" + atrcheck
### add row if row not empty
if entry != "":
  dic_entries[current_dn] = dic_entry
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
  dic_entry = {}
  entry = ""

### print utf-8-BOM and headers
print('\ufeff' + "\"" + header_string.replace(";", "\";\"") + "\"")
### print data
for d in dic_entries:
  od = dic_entries[d]
  csv_row = ""  ### row csv value
  split_char = ""  ### spliter for values
  for column in header_string.split(";"):
    if len(csv_row) > 0:
      split_char = ";"  ### need split because csv row have chars
    find_column = od.get(column)
    if find_column and find_column.strip():
      csv_row += split_char + "\"" + od[column].replace("\"", "\"\"") + "\""
    else:
      csv_row += split_char + "\"\""
  print(csv_row)
```

Рассмотрим пример конвертации данных из ldapsearch через конвейер:

```
ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
 ↪"(uid=*)" uid displayName sn mail rbtadb | python3 ldif2csv.py
```

Результат выполнения:

```
"dn";"uid";"sn";"displayName";"mail"
"uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan";"admin";
 ↪"Administrator";"";""
"uid=petrovp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan";"petrovp";"Петров
 ↪";"Петров петр";"petrovp@ald.company.lan"
"uid=ivani,cn=users,cn=accounts,dc=ald,dc=company,dc=lan";"ivani";"Иванов";
 ↪"Иван Иванов";"ivani@ald.company.lan"
"uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan";"petrovss";
 ↪"Сидоров";"Петр С.";"petrov.ss@ald.company.lan"
```

Вы можете сохранить этот вывод в файл users.csv простым перенаправлением:

```
ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
↪"(uid=*)" uid displayName sn mail rbtadb | python3 ldif2csv.py > users.csv
```

На рисунке 17 можно увидеть, как будет выглядеть содержимое файла, если его открыть в LibreOffice Calc.

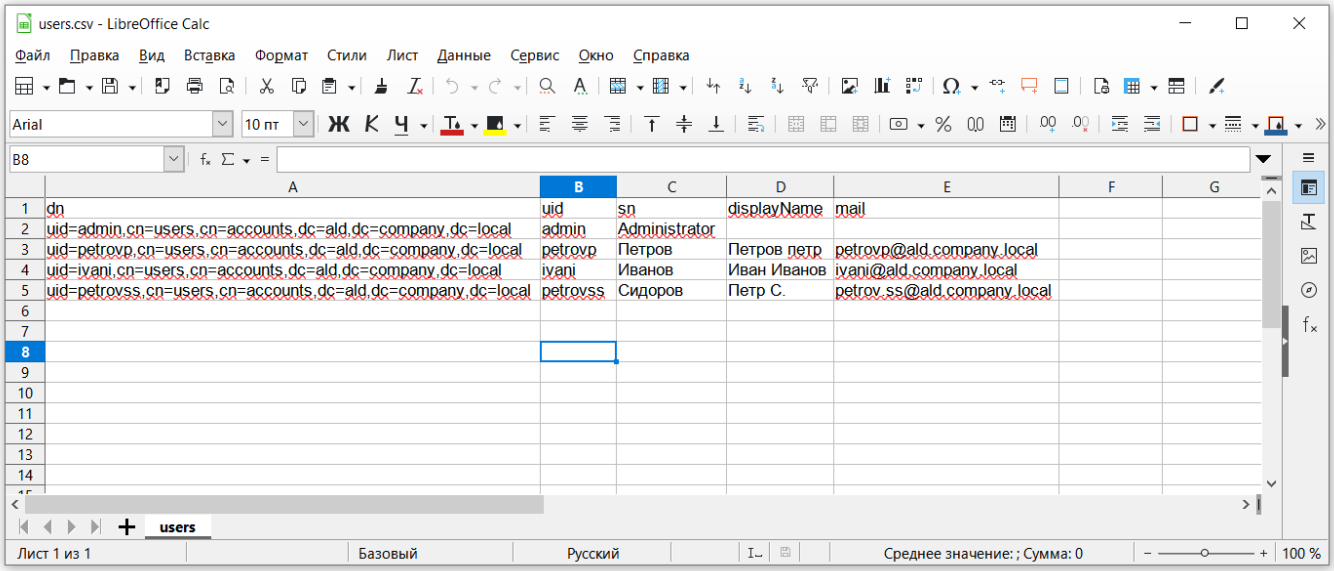


Рисунок 6.38 – Файл, экспортированный с помощью ldif2csv.py, открытый в LibreOffice Calc

Далее вы можете использовать полученный CSV-файл следующим образом:

```
#!/bin/bash
csv_file=$1
cat $csv_file | while read line
do
  if [ ! -z "${line}" ]; then
    if [[ "${line:0:1}" == "\"" ]]; then
     column1=$(echo ${line} | cut -d ";" -f 1 | sed "s/^\"//g" | sed "s/\"$//
↪g")
     column2=$(echo ${line} | cut -d ";" -f 2 | sed "s/^\"//g" | sed "s/\"$//
↪g")
     column3=$(echo ${line} | cut -d ";" -f 3 | sed "s/^\"//g" | sed "s/\"$//
↪g")
     echo -e "$column1\t$column2\t$column3"
     ### можете добавить свои действия по обработке данных из файла
    fi
  fi
done
```

Если сохранить приведенный скрипт в файл `parse_csv.sh`, то ему можно передать имя csv-файла для обработки, как параметр. Не забудьте назначить скрипту права на выполнение:

```
chmod +x ./parse_csv.sh
./parse_csv.sh users.csv
```

Результат выполнения:

```
uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan       admin  
 ↪Administrator
uid=petrovp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan     petrovp Петров
uid=ivani,cn=users,cn=accounts,dc=ald,dc=company,dc=lan       ivani   Иванов
uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan    petrovss      
 ↪ Сидоров
```

Как мы видим, выводится несколько колонок с разделителем табуляции **t**.

Конвертер ldif2json и работа с JSONPath

Для работы со структурированными данными можно также использовать формат JSON, и в Linux есть очень удобный процессор JSON, который называется jq. Вот пример Python скрипта, который позволяет конвертировать поток LDIF в JSON, для его использования создайте файл `ldif2json.py` и скопируйте туда следующее содержимое:

```
#!/usr/bin/python3
import sys
import base64
import json
import re
### parse bool and int to json
def parce_value(value):
  try:
    return int(value)
  except:
    if value == "FALSE" or value == "FALSE": return bool(value)
    else: return value

if sys.version_info[0] < 3:
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
  raise Exception("Use Python 3: python3 ldif2json.py")

data = sys.stdin.readlines()
header_string = ""
atrcheck = ""
entry = ""
dic_entries = {}
dic_entry = {}
current_dn = ""
val_base64 = False
### main loop for parse headers and collect dict
for line in data:
  ln = line.replace("\n", "").replace("\r", "")
  if len(ln) == 0:
    if not current_dn == "":
      dic_entries[current_dn] = dic_entry
      dic_entry = {}
      entry = ""
  else:
    if ln.startswith("version"):
      dic_entries["version"] = ln.split(": ")[1]
      continue
    elif ln.lstrip()[0] == "#":
      continue

    ### if line wrapped line starts with " " then add line to last attr
    elif ln[0] == " ":
      dic_entry[atrcheck] += ln.lstrip()
      if val_base64:
        val_to_decode = dic_entry[atrcheck]
        try:
          dic_entry[atrcheck] = base64.b64decode(val_to_decode).decode(
            'utf-8').strip()
        except:
          dic_entry[atrcheck] = val_to_decode
      continue
    entry += ln
    attribute = []
    attribute_name = ""
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
    attribute_value = ""
    if ln.find(":: ") > 0:
      val_base64 = True
      attribute = ln.split(":: ")
      try:
        attribute_value = base64.b64decode(
          attribute[1]).decode('utf-8').strip()
      except:
        attribute_value = attribute[1]
    elif ln.find(":< ") > 0:
      val_base64 = False
      attribute = ln.split(":< ")
      attribute_value = attribute[1]
    else:
      val_base64 = False
      attribute = ln.split(": ")
      try:
        attribute_value = re.sub(r"^.*?: ", "", ln)
      except:
        attribute_value = ""
    atrcheck = attribute[0].replace(":", "")
    ### get attribute and check if attribute exist

    if dic_entry.get(atrcheck):
      dic_entry[atrcheck] = str(
        dic_entry[atrcheck]) + "|" + str(attribute_value)
    else:
      dic_entry[atrcheck] = parce_value(attribute_value)
    if atrcheck == "dn":
      current_dn = attribute[1]
  if header_string.find(atrcheck) < 0:
    if header_string == "":
      header_string += atrcheck
    else:
      header_string += ";" + atrcheck
### add row if row not empty
if entry != "":
  dic_entries[current_dn] = dic_entry
  dic_entry = {}
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
  entry = ""

### print stdout json from dict
print(json.dumps(dic_entries, ensure_ascii=False))
```

Рассмотрим пример конвертации данных из ldapsearch через конвейер

```
ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
↪"(uid=*)" uid displayName sn mail rbtadb | python3 ldif2json.py
```

Результат выполнения:

```
{
  "uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan": {
    "dn": "uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan",
    "uid": "admin",
    "sn": "Administrator"
  },
  "uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan": {
    "dn": "uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan",
    "uid": "petrovss",
    "displayName": "Петр С.",
    "sn": "Сидоров",
    "mail": "petrov.ss@ald.company.lan"
  },
}
```

Вы можете сохранить этот вывод в файл `users.json` простым перенаправлением и далее с помощью утилиты jq из одноименного пакета извлекать любые данные с помощью запросов JSONPath. Например, узнаем значение атрибута **mail** для пользователя **petrovss**:

```
ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
↪"(uid=*)" uid displayName sn mail rbtadb | python3 ldif2json.py > users.
↪json && jq -r '."uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
↪".mail' users.json
```

где:

- параметр **-r** – означает, что выводить нужно сырые данные без кавычек;

- **«.”uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan.mail»**
  - это текст запроса JSONPath

Результат выполнения:

```
petrov.ss@ald.company.lan
```

Давайте обработаем полученные JSON данные в цикле, создайте файл json_cycle.sh со следующим содержимым:

```
#!/bin/bash
echo -e "uid\tdisplayName\tsn\tmail"
ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
 ↪"(uid=*)" uid displayName sn mail rbtadb | python3 ldif2json.py > users.
 ↪json
cat users.json | jq -r 'delpaths([paths | select(length > 1)])' | jq -r
 ↪'[paths | join(".")]'| jq -r 'join("\n")' | while read key
do
 uid=$(jq -r ".\"$key\".uid" users.json)
 mail=$(jq -r ".\"$key\".mail" users.json)
 displayName=$(jq -r ".\"$key\".displayName" users.json)
 sn=$(jq -r ".\"$key\".sn" users.json)
 echo -e "$uid\t$displayName\t$sn\t$mail"
 ### можете добавить свои действия по обработке данных из файла
done
```

Результат выполнения:

```
uid     displayName     sn      mail
admin   null    Administrator   null
petrovp Петров петр     Петров  petrovp@ald.company.lan
ivani   Иван Иванов     Иванов  ivani@ald.company.lan
petrovss        Петр С. Сидоров petrov.ss@ald.company.lan
```

6.4.4.2. Добавление пользователей

Добавим нового пользователя petrov.ss с помощью *ldapadd*. В этом примере при добавлении пользователя objectClass класс с именем ipantuserattrs не добавляется, чтобы атрибут ipaNTSecurityIdentifier сгенерировался автоматически. Рассмотрим файл

add_user.ldif:

```
dn: uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
objectClass: inetuser
objectClass: posixaccount
objectClass: krbprincipalaux
objectClass: krbticketpolicyaux
objectClass: ipaobject
objectClass: ipasshuser
objectClass: x-ald-user
objectClass: x-ald-user-parsec14
objectClass: x-ald-audit-policy
objectClass: ruPostMailAccount
objectClass: rbtaCustomUserAttrs
objectClass: rbtaUserMeta
objectClass: rbta-unit
objectClass: rbta-address
objectClass: rbta-inetorgperson-ext
objectClass: ipaSshGroupOfPubKeys
objectClass: mepOriginEntry
givenName: Петр
sn: Сидоров
uid: petrovss
cn: petrov.ss
uidNumber: -1
gidNumber: -1
displayName: Петр С.
initials: P.C.
gecos: Тел. +71234567890
rbtamiddlename: Сидоров
rbtadp: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan
loginShell: /bin/bash
homeDirectory: /home/petrov.ss
mail: petrov.ss@ald.company.lan
x-ald-user-mac: 0:0x0:0:0x0
krbCanonicalName: petrov.ss@ALD.COMPANY.LAN
krbPrincipalName: petrov.ss@ALD.COMPANY.LAN
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
userPassword: somepassword
```

Примечание: Атрибуты uidNumber и gidNumber нужно устанавливать равными -1, в этом случае DNA-плагин автоматически сгенерирует значения идентификаторов при добавлении пользователя. Пароль следует передавать открытым текстом. Он будет хеширован алгоритмом PBKDF2_SHA256 автоматически. Записать пользователю, сгенерированный где-то в другом месте hash пароля возможно только при создании пользователя, если сервер переведен в режим миграции.

Выполним команду по добавлению нового пользователя:

```
ldapadd -Q < add_user.ldif
```

Результат выполнения:

```
adding new entry "uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
```

В результате мы увидим успешное выполнение. После входа пользователю нужно будет задать новый пароль, так как пароль назначенный таким способом является временным.

Теперь добавим пользователей в цикле, используя данные из CSV файла. Создадим в `cycleadd.csv` с разделителем **«;»** в любой программе:

```
givenName;sn;uid
Александр;Петров;alexp
Михаил;Иванов;mikhaili
Артём;Сидоров;artems
```

Создадим скрипт `add_from_csv.sh` по добавлению пользователей из CSV файла:

```
#!/bin/bash
csv_file=$1
delim=$2
function template()
{
 psw=$(cat /dev/urandom| tr -dc '0-9a-zA-Z!@#$%^&*_+-' | head -c 14;echo;)
 uid=$1 #установить uid из первого параметра
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
 givenName=$2 #установить Имя из второго
 sn=$3 #установить Фамилию из третьего
cat <<EOF
dn: uid=$uid,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
objectClass: inetuser
objectClass: posixaccount
objectClass: krbprincipalaux
objectClass: krbticketpolicyaux
objectClass: ipaobject
objectClass: ipasshuser
objectClass: x-ald-user
objectClass: x-ald-user-parsec14
objectClass: x-ald-audit-policy
objectClass: ruPostMailAccount
objectClass: rbtaCustomUserAttrs
objectClass: rbtaUserMeta
objectClass: rbta-unit
objectClass: rbta-address
objectClass: rbta-inetorgperson-ext
objectClass: ipaSshGroupOfPubKeys
objectClass: mepOriginEntry
givenName: $givenName
sn: $sn
uid: $uid
cn: $uid
uidNumber: -1
gidNumber: -1
displayName: $givenName ${sn:0:1}.
initials: ${givenName:0:1}. ${sn:0:1}.
rbtamiddlename: $sn
rbtadp: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan
loginShell: /bin/bash
homeDirectory: /home/$uid
mail: $uid@ald.company.lan
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
x-ald-user-mac: 0:0x0:0:0x0
krbCanonicalName: $uid@ALD.COMPANY.LAN
krbPrincipalName: $uid@ALD.COMPANY.LAN
userPassword: $psw

EOF
}
echo Добавление пользователей из $csv_file
truncate -s 0 cycleadd.ldif
chmod 500 ./cycleadd.ldif
while read line; do let c++;
if [ $c -gt 1 ]; then
 p_givenName=$(echo ${line} | cut -d $delim -f 1 | sed "s/^\"//g" | sed "s/\"
 ↪$//g" | sed "s/^\'//g" | sed "s/\'$//g")
 p_sn=$(echo ${line} | cut -d $delim -f 2 | sed "s/^\"//g" | sed "s/\"$//g"
 ↪| sed "s/^\'//g" | sed "s/\'$//g")
 p_uid=$(echo ${line} | cut -d $delim -f 3 | sed "s/^\"//g" | sed "s/\"$//g"
 ↪| sed "s/^\'//g" | sed "s/\'$//g")
 template $p_uid $p_givenName $p_sn >> cycleadd.ldif
fi
done < $csv_file
### добавим пользователей одним пакетом
ldapadd -Q -c < cycleadd.ldif
```

Как мы видим, есть функция template, в которой описан шаблон одной записи. Этот шаблон вы можете настроить по своему усмотрению, прописав свои корневые суффиксы и реалмы.

Назначим права запуска скрипту `add_from_csv.sh` и запустим его:

```
chmod +x ./add_from_csv.sh && ./add_from_csv.sh cycleadd.csv ';'
```

Результат выполнения:

```
Добавление пользователей из cycleadd.csv
adding new entry "uid=alexp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
adding new entry "uid=mikhaili,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
adding new entry "uid=artems,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
```

Скрипт `add_from_csv.sh` создал временный LDIF файл `cycleadd.ldif` и передал

команде ldapadd. После чего мы можем создать новый файл `new_users.csv` уже после добавления пользователей, потому что там уже сохранены новые пароли пользователей:

```
cat cycleadd.ldif | python3 ldif2csv.py > new_users.csv
```

Файл new_users.csv можно скачать, чтоб потом отправить коллегам пароли и учетные данные для входа, см Рис 18.

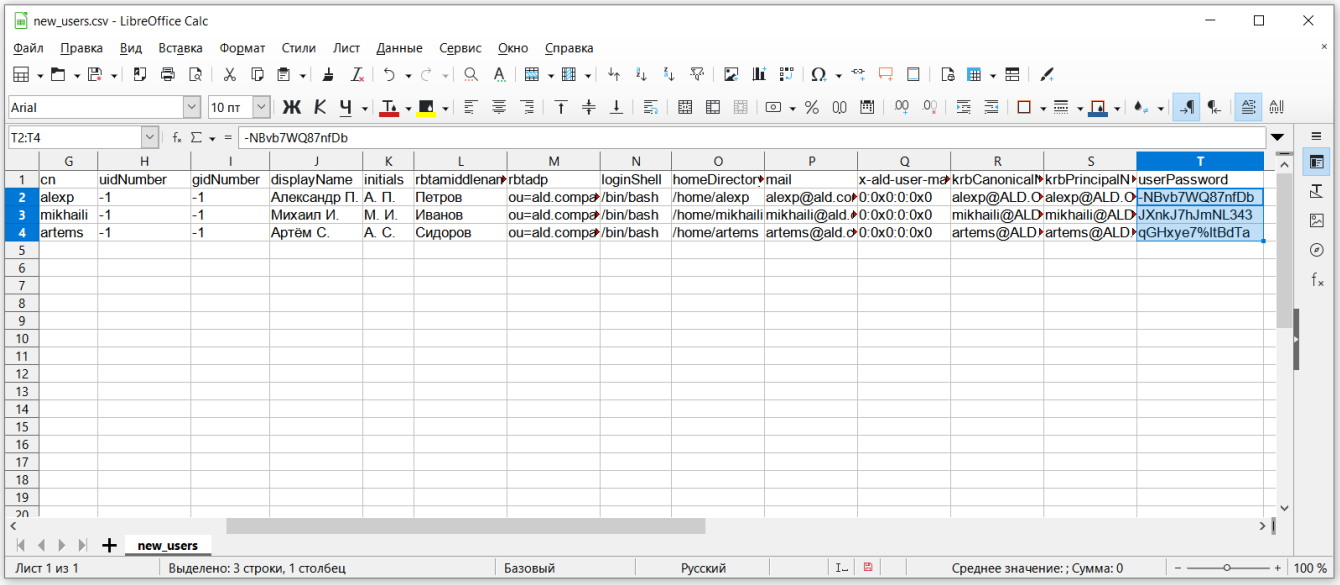


Рисунок 6.39 – Файл с новыми паролями для добавленных в цикле пользователей

6.4.4.3. Смена пароля пользователей

В случае взлома системы нам нужно изменить пароли всех пользователей. Для этого напишем скрипт на языке bash `change_all_pass.sh`:

```
#!/bin/sh
read -p "Вы хотите создать новые пароли всем пользователям (Yes|no)? " yn
if [[ $yn =~ "Yes" ]]; then
  echo '"dn";"password"' > new_passwords.csv
  echo "" > tmp_new_pass
  ldapsearch -Q -LLL -s one -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan
↪" "(!(uid=admin))" dn | while read line
  do
  if [ ! -z "${line}" ]; then
    psw=$(cat /dev/urandom| tr -dc '0-9a-zA-Z!@#$%^&*_+-' | head -c 14;echo;)
    echo "\"$line\";\"$psw\"" >> new_passwords.csv
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
    echo -e "${line}\nchangetype: modify\nreplace: userPassword\
↪nuserPassword: $psw\n" >> tmp_new_pass
  fi
  done
  ldapmodify -Q < tmp_new_pass
fi
```

Запустим скрипт:

```
chmod +x change_all_pass.sh && ./change_all_pass.sh
```

Результат выполнения:

```
Вы хотите создать новые пароли всем пользователям (Yes|no)? Yes
modifying entry "uid=petrovp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
modifying entry "uid=ivani,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
modifying entry "uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
```

Подготовим CSV файл `newpass.csv` с паролями из временного tmp_new_pass LDIF через конвертер `ldif2CSV.py`:

```
cat tmp_new_pass  | python3 ldif2csv.py > newpass.csv
```

Результат выполнения:

```
"dn";"changetype";"replace";"userPassword"
"uid=petrovp,cn=users,cn=accounts,dc=ald,dc=company,dc=lan";"modify";
↪"userPassword";"l@q%yTU1hf_EP7"
"uid=ivani,cn=users,cn=accounts,dc=ald,dc=company,dc=lan";"modify";
↪"userPassword";"nM^ZER_z_sULpE"
"uid=petrovss,cn=users,cn=accounts,dc=ald,dc=company,dc=lan";"modify";
↪"userPassword";"oxK*HzeFmGJw5w"
```

А также можно открыть файл newpass.csv в редакторе электронных таблиц, см. Рис 19.

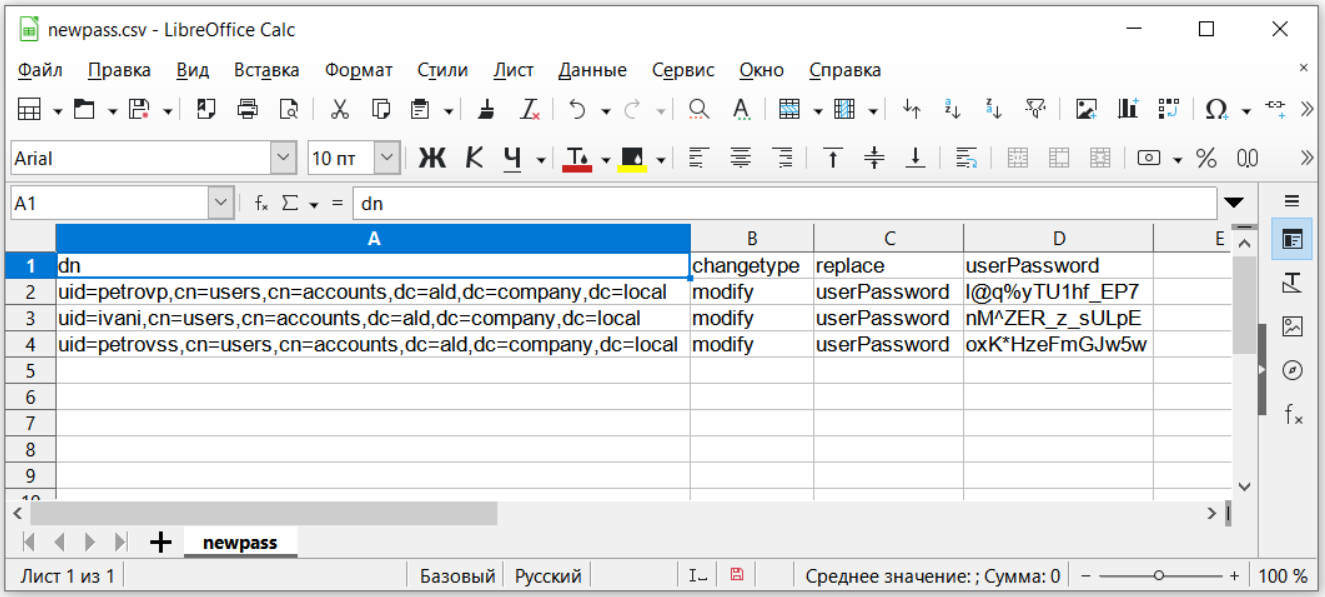


Рисунок 6.40 – Пакетное изменение паролей пользователей

6.4.4.4. Проверка просроченных паролей

Проверим просроченные пароли пользователей с помощью скрипта `check_expired.sh`:

```
#!/bin/bash
### 1. Получить список пользователей из DN cn=users,cn=accounts,dc=ald,
 ↪dc=company,dc=lan
### 2. Отфильтровать список с полями uid, displayname, mail,
 ↪krbPasswordExpiration
### 3. Оставить пользователей, у которых дата просрочки пароля
 ↪(krbPasswordExpiration) от даты сегодня до даты сегодня + 7 дней
ldapsearch -Q -LLL -o ldif-wrap=no -b "cn=users,cn=accounts,dc=ald,dc=company,
 ↪dc=lan" "(&(krbPasswordExpiration>=$(date +"%Y%m%d000000Z
 ↪"))(krbPasswordExpiration<=$(date +"%Y%m%d000000Z" -d "+7 days")))" uid
 ↪displayname mail krbPasswordExpiration > expired_ldif
while read line
do ### 4. Цикл пользователь из пользователей
if [ -z "${line}" ]; then
 if [ ! -z $user_mail ]; then
  ###   4.1 Если почта у пользователя есть
  timeExp=$(date -d "${date_expire:0:4}-${date_expire:4:2}-${date_expire:6:2}
 ↪ ${date_expire:8:2}:${date_expire:10:2}:${date_expire:12:2}Z" +"%s")
  timeNow=$(date +"%s")
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
  seconds=$(( timeExp - timeNow ))
  min=$(( seconds / 60 ))
  hours=$(( min / 60 ))
  days=$(( hours / 24 ))
  ago=$(echo "через $daysдн.")
  if [ $timeNow -gt $timeExp ]; then
   ago="просрочен!!"
  fi
  dateprint=$(date -d "@$timeExp")
  echo "$user_name ($user_mail), пароль просрочится ${dateprint} $ago"
  ### Тут вы можете добавить свою обработку пользователя.
else
  ###   4.3 Иначе добавить пользователя в лог
  echo "$(date) [expire.sh] uid $user_uid mail is epmty" >> "expired.log"
 fi
 ### очистим переменные для следующей записи
 user_uid=""
 user_name=""
 user_mail=""
 date_expire=""
else
  attr=$(echo $line | cut -d ":" -f 1)
  attvalue=$(echo $line | cut -d ":" -f 2)
  if [ -z "$attvalue" ]; then
    attvalue=$(echo $line | cut -d " " -f 2 | base64 -d)
  fi
  ### проверка атрибута и присвоение переменной значения
  case $attr in
    uid) user_uid=$(echo $attvalue | xargs echo -n);;
    displayname) user_name=$(echo $attvalue | xargs echo -n) ;;
    mail) user_mail=$(echo $attvalue | xargs echo -n);;
    krbPasswordExpiration) date_expire=$(echo $attvalue | xargs echo -n);;
  esac
fi
done < expired_ldif
```

Поставим атрибут выполнения и запустим скрипт:

```
chmod +x ./check_expired.sh && ./check_expired.sh
```

Результат выполнения:

```
Петров петр (petrovp@ald.company.lan), пароль просрочится Вт мая 30 12:18:21
 ↪MSK 2023 через 2дн.
Александр П. (alexp@ald.company.lan), пароль просрочится Сб мая 27 12:46:42
 ↪MSK 2023 просрочен!!
Михаил И. (mikhaili@ald.company.lan), пароль просрочится Сб мая 27 12:46:42
 ↪MSK 2023 просрочен!!
Артём С. (artems@ald.company.lan), пароль просрочится Сб мая 27 12:46:42 MSK
 ↪2023 просрочен!!
```

Мы обработали даты просрочки пароля, указав просроченные пароли и дни до их завершения. Заметьте, что в данном примере мы использовали только bash и вывод ldapsearch. Вы также можете добавить в скрипт свою обработку просроченных учетных записей. Например, отправку сообщения в корпоративный мессенджер или на электронную почту.

6.4.5. Заключение

LDAP разрабатывался с целью хранения любой информации об объектах на предприятии, таких как пользователи, компьютеры, серверы, подразделения и др. Информация удобно расположена в виде иерархического дерева, которое строго типизировано объектными классами. База данных ориентирована на чтение, где быстро и легко можно получить любую информацию по объектам.

На сегодняшний день LDAP-каталог – это стандарт. Множество продуктов имеют встроенные интеграции с службой каталога, а также существует большое количество библиотек для разных языков для доступа к LDAP серверу, например: python-ldap для python, ldaptive для java. В некоторые языки программирования уже встроена работа LDAP, например, в языки PHP и C#. А также у продукта ALD Pro есть REST API, через него можно управлять каталогом, используя простые Web запросы.

Мы рассмотрели автоматизацию через LDAP запросы, используя только некоторые инструменты командной строки. Научились получать данные по запросу, а также добавлять, изменять и удалять данные. Комбинируя разные примеры, написали несколько скриптов bash и python для решения задач администрирования. Надеемся, что данные примеры помогут Вам написать свои решения для автоматизации.

6.5. Повышение привилегий доменных пользователей с помощью правил SUDO

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
```

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

6.5.1. 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.

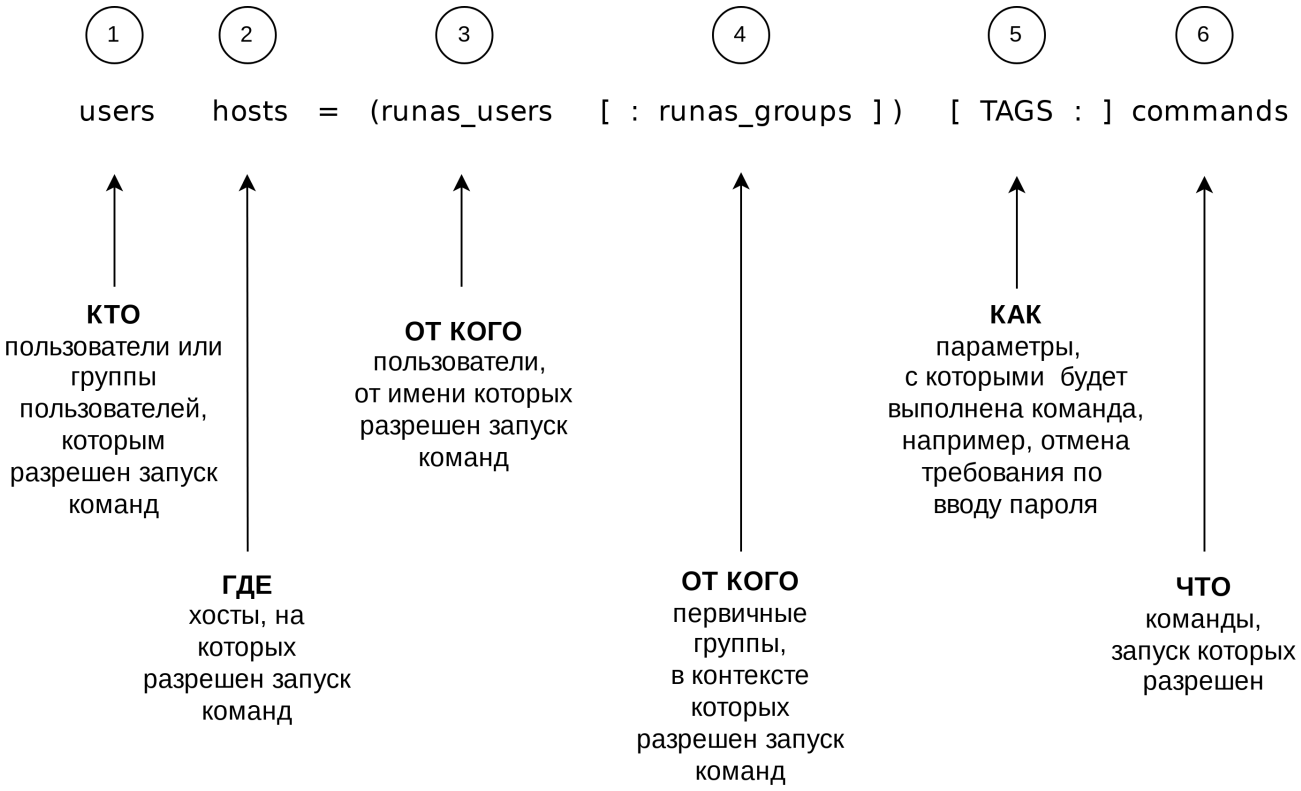


Рисунок 6.41 – Синтаксис правил SUDO

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.

Компоненты правила:

1. **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
```

2. **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
```

3. **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
```

4. **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
```

5. ** 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
```

6. **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.

6.5.2. Механизм работы правил SUDO

Настройки утилиты sudo определяются содержимым файла `/etc/sudoer``s`. 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 есть 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. Заменить редактор по умолчанию можно 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
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
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:

#includedir /etc/sudoers.d

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

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.

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.

6.5.3. 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

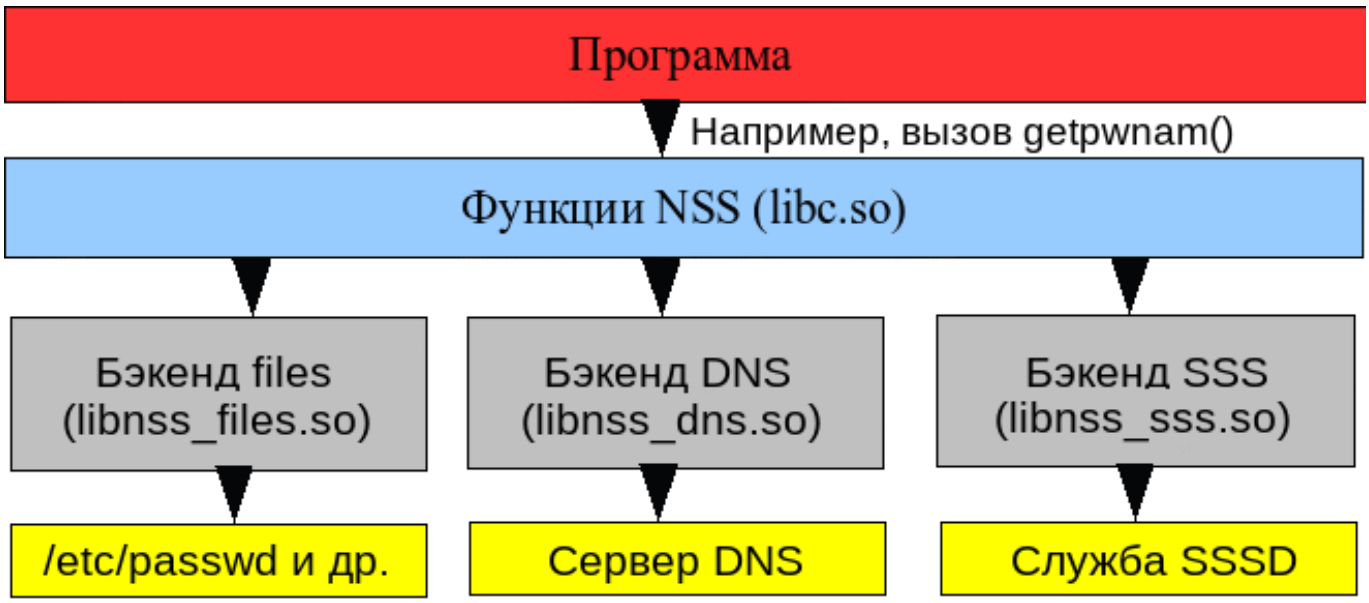


Рисунок 6.42 – Архитектура диспетчера службы имен (Name Service Switch, NSS)

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,{basedn}=имя,{basedn}=домена,{basedn}=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,{basedn}=имя,{basedn}=домена,{basedn}=организации`, и модуль sss через службу SSSD берет данные напрямую из этой ветки каталога.

Для обеспечения совместимости со старым модулем ldap, 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.

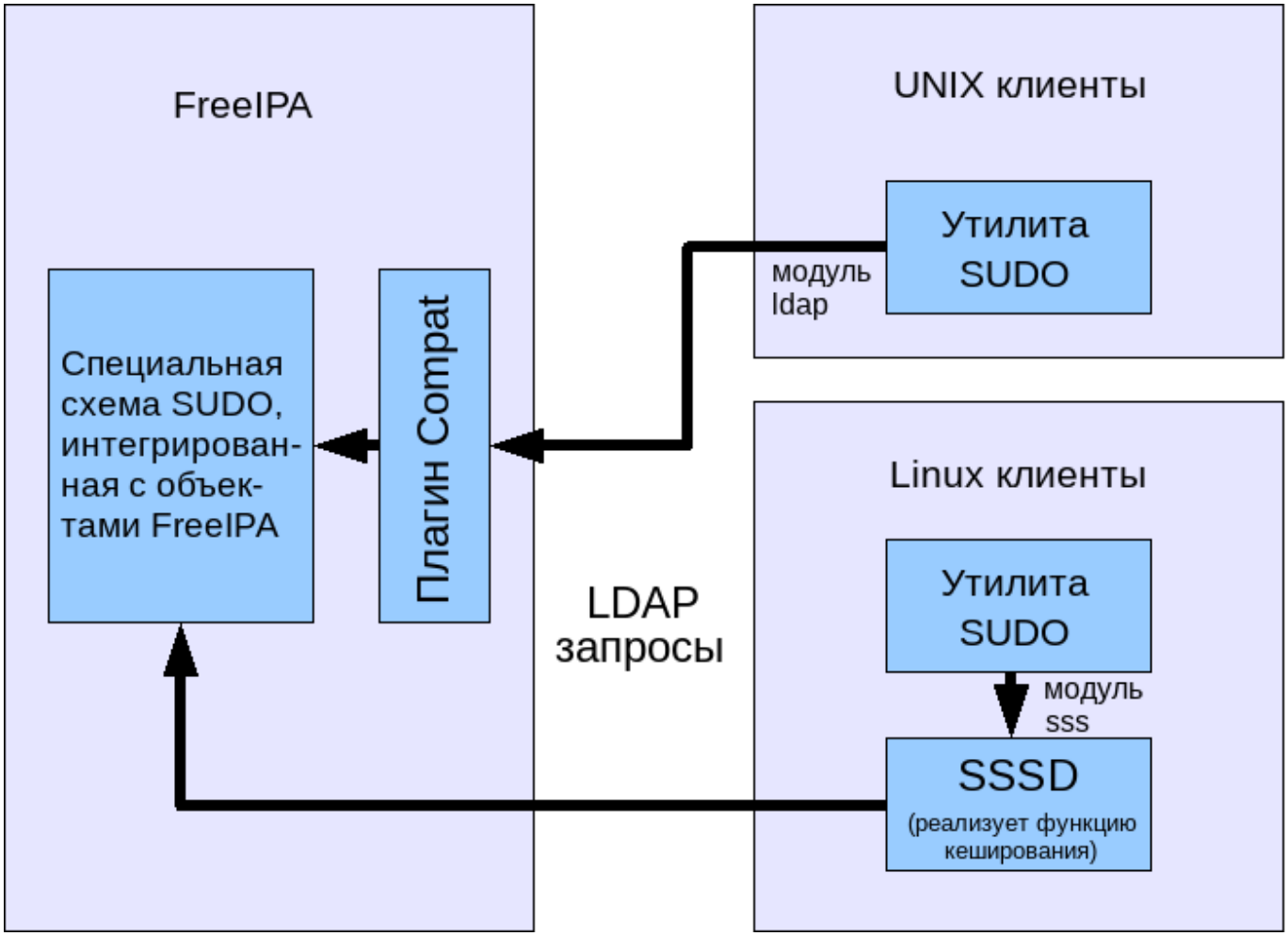


Рисунок 6.43 – Источник правил SUDO в зависимости от реализации клиентской части

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 `sssctl`, входящим в пакет sssd-tools:

```
sudo sssctl cache-remove
```

Еще один важный момент. В силу особенностей FreeIPA 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 mepmanagedentry 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 можно использовать вспомогательные группы в правилах SUDO без ограничений.

6.5.4. Настройка правил SUDO в домене

6.5.4.1. Через портал управления ALD Pro

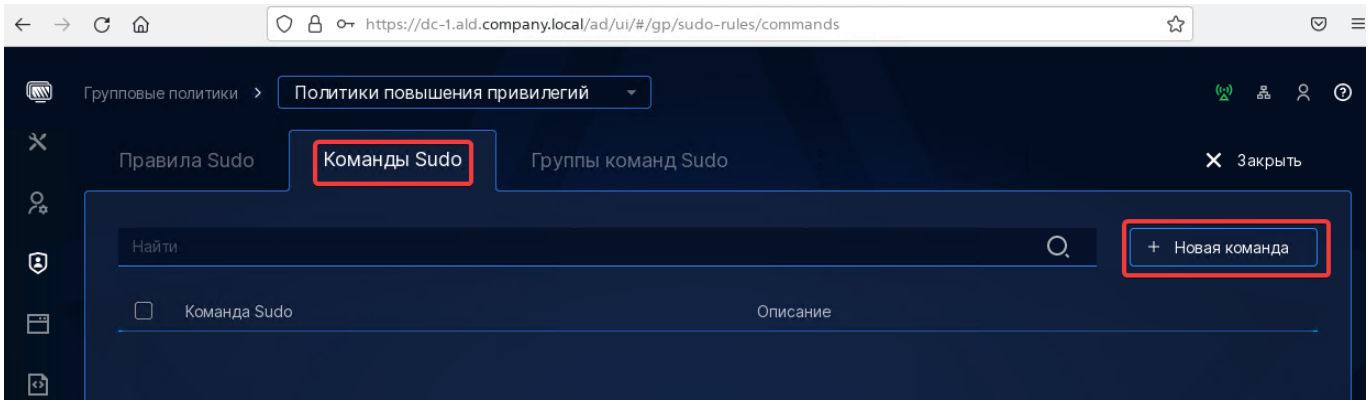
Создание команды

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 2.0.0 не позволяет использовать пробелы в названии команд, поэтому для создания команд с параметрами потребуется воспользоваться интерфейсом FreeIPA или командной строкой.

1. Открыть страницу «Групповые политики > Политики повышения привилегий >

Команды Sudo»

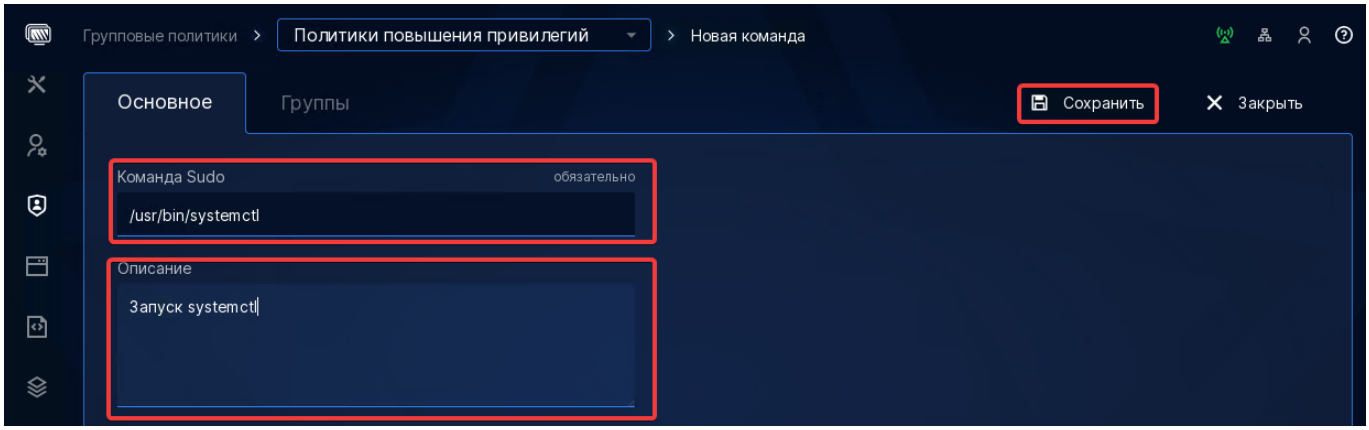
2. Нажать кнопку «+ Новая команда»



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

---

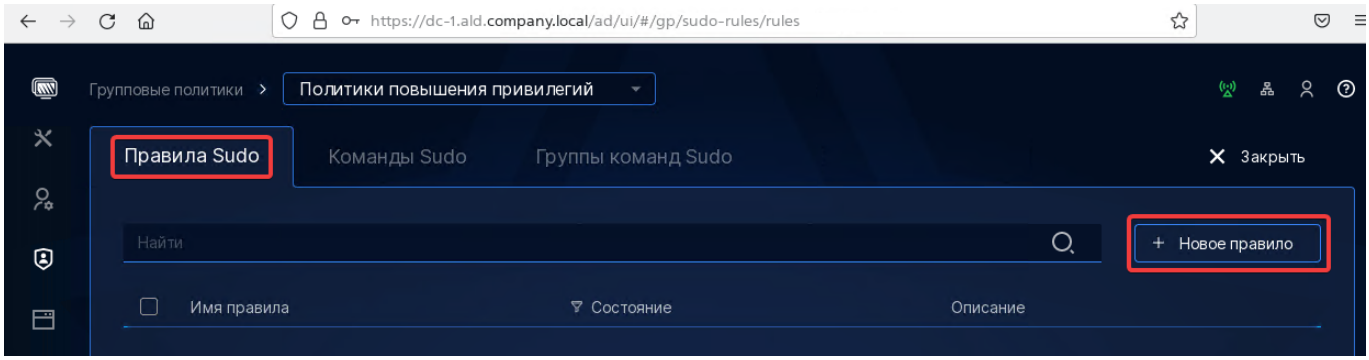
3. Ввести команду /usr/bin/systemctl, описание (опционально) и нажать «Сохранить»



Создание правила

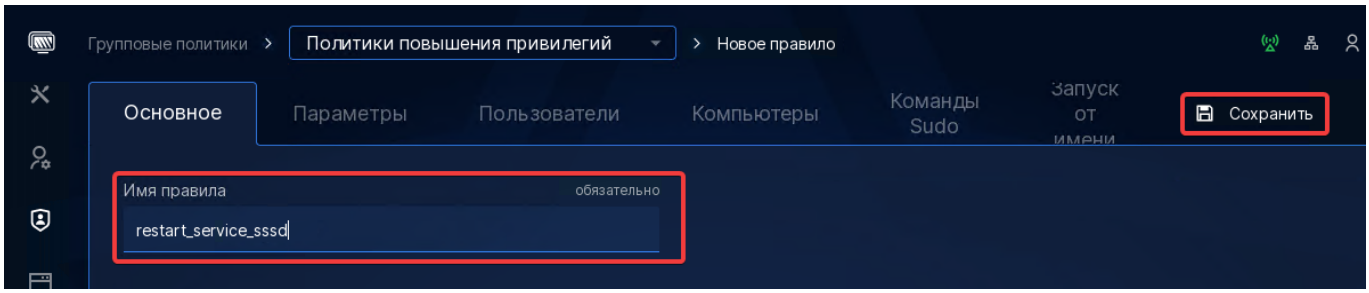
1. Открыть страницу «Групповые политики > Политики повышения привилегий > Правила Sudo»

2. Нажать кнопку «+ Новое правило»



3. На странице «Новое правило» ввести имя правила и нажать кнопку «Сохранить».

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

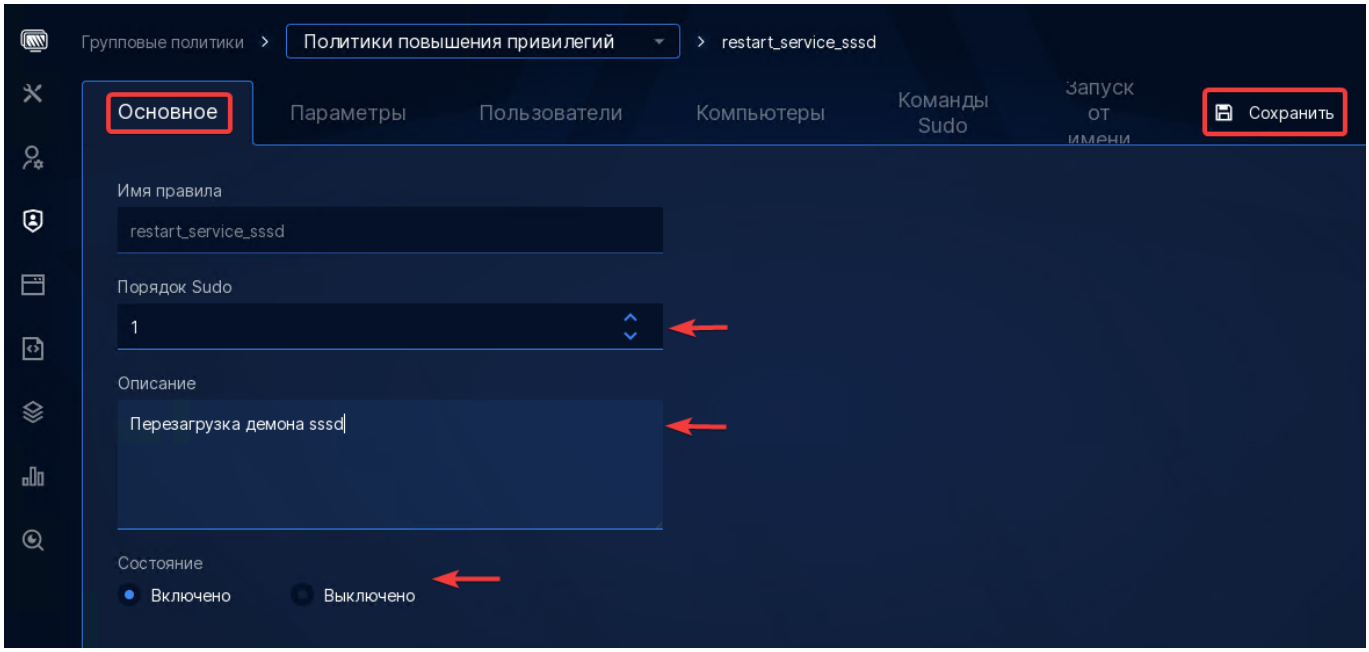


4. На странице правила задать следующие параметры:

    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 сохранить изменения до перехода к следующей вкладке, иначе изменения будут утеряны.



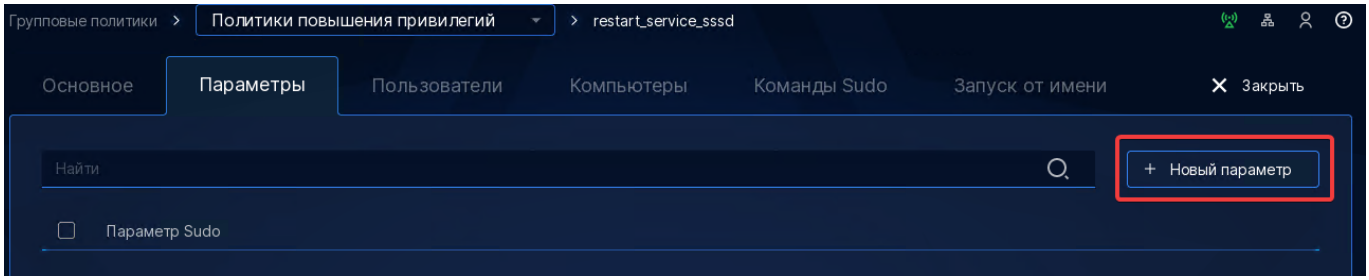
2. Раздел «Параметры». С помощью параметров можно изменить поведение утилиты для ее тонкой настройки. Ниже приведено несколько наиболее востребованных параметров:

    - 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 с ключом -V под суперпользователем, например `sudo sudo -V`

Для создания параметра:

Нажать кнопку «+ Новый параметр»



Ввести параметр, например «timestamp_timeout=5», и нажать кнопку «Сохранить»

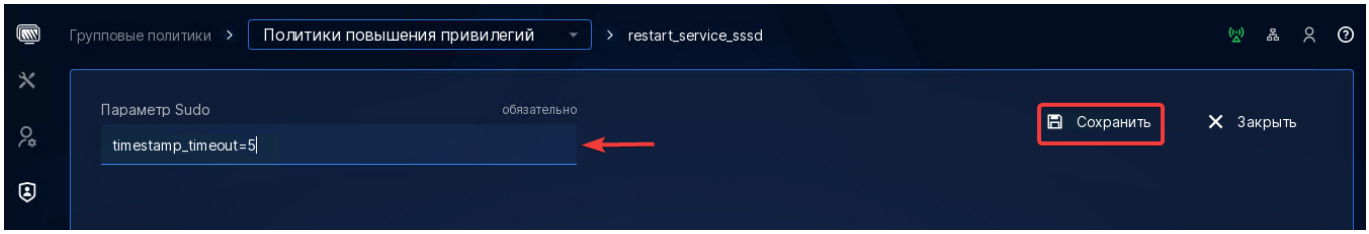


Рисунок 6.44 – Сохранить 1

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

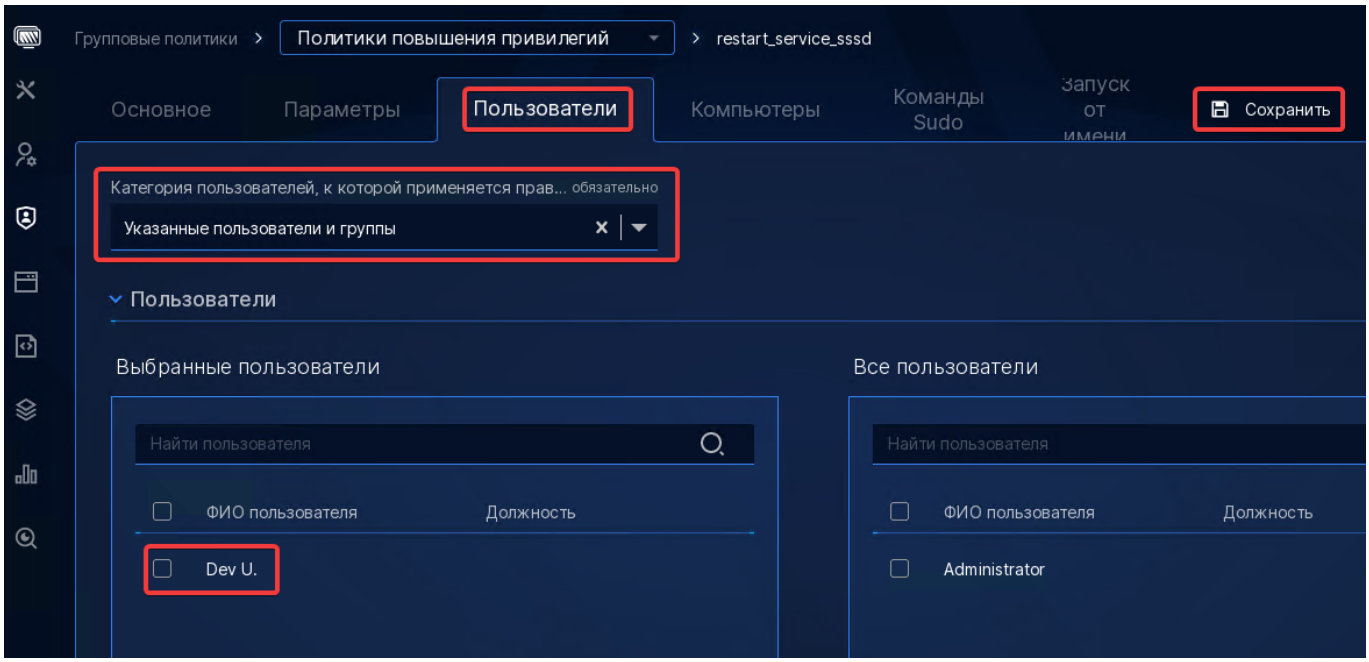


Рисунок 6.45 – Сохранить 2

4. 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. Указать компьютер и нажать кнопку «Сохранить».

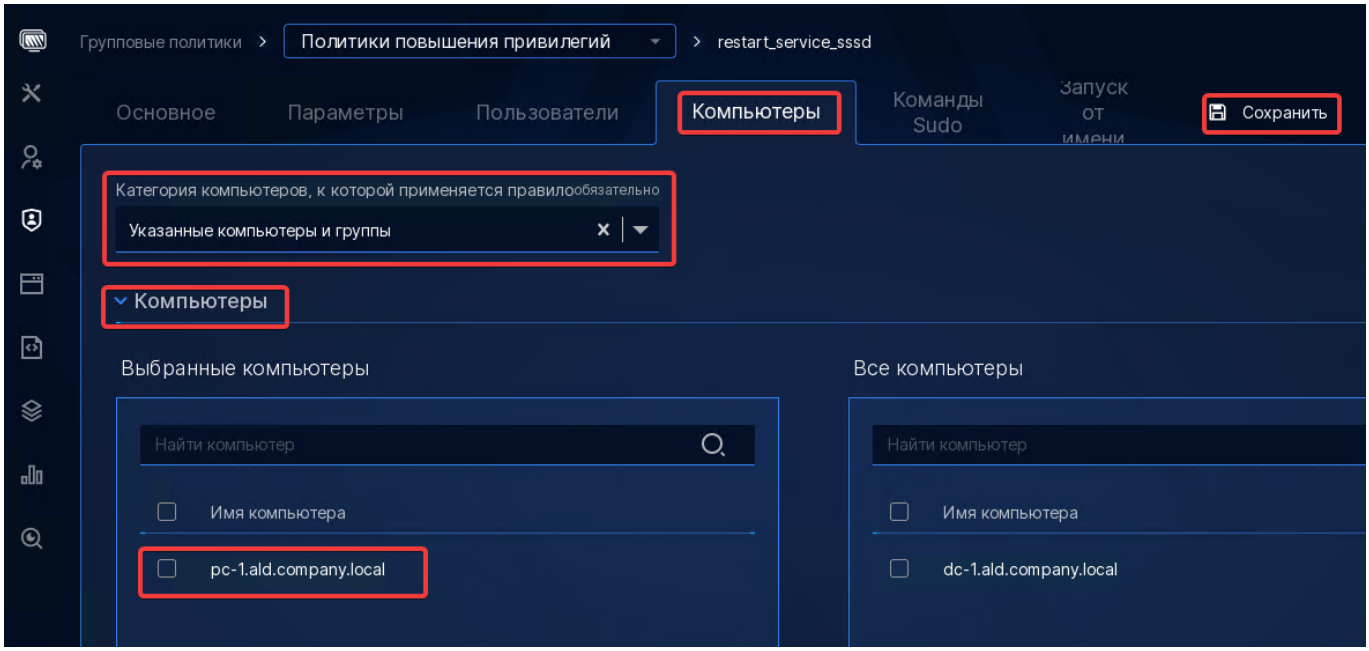


Рисунок 6.46 – Сохранить 3

5. Раздел «Команды Sudo». На этой вкладке можно задать список команд, которые разрешено/запрещено будет выполнять. Выбрать команды и нажать кнопку «Сохранить». Сначала применяются разрешающие команды, затем запрещающие, поэтому у запрещающих будет выше приоритет.

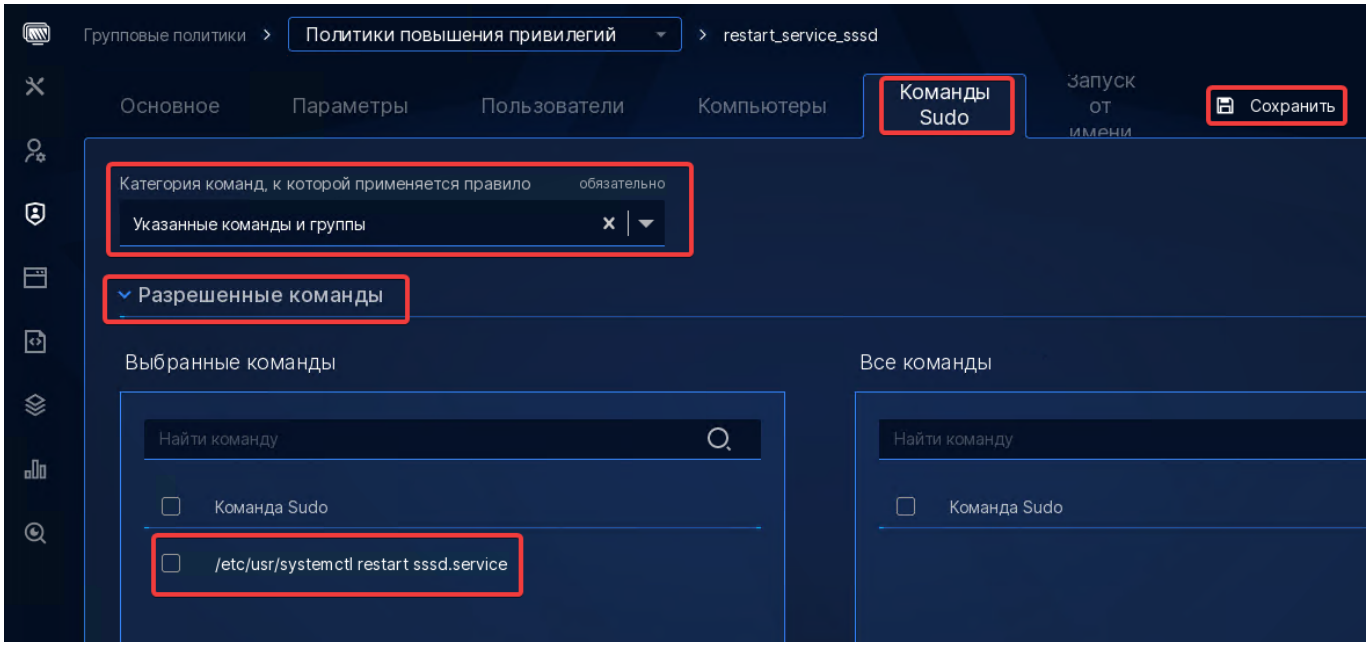
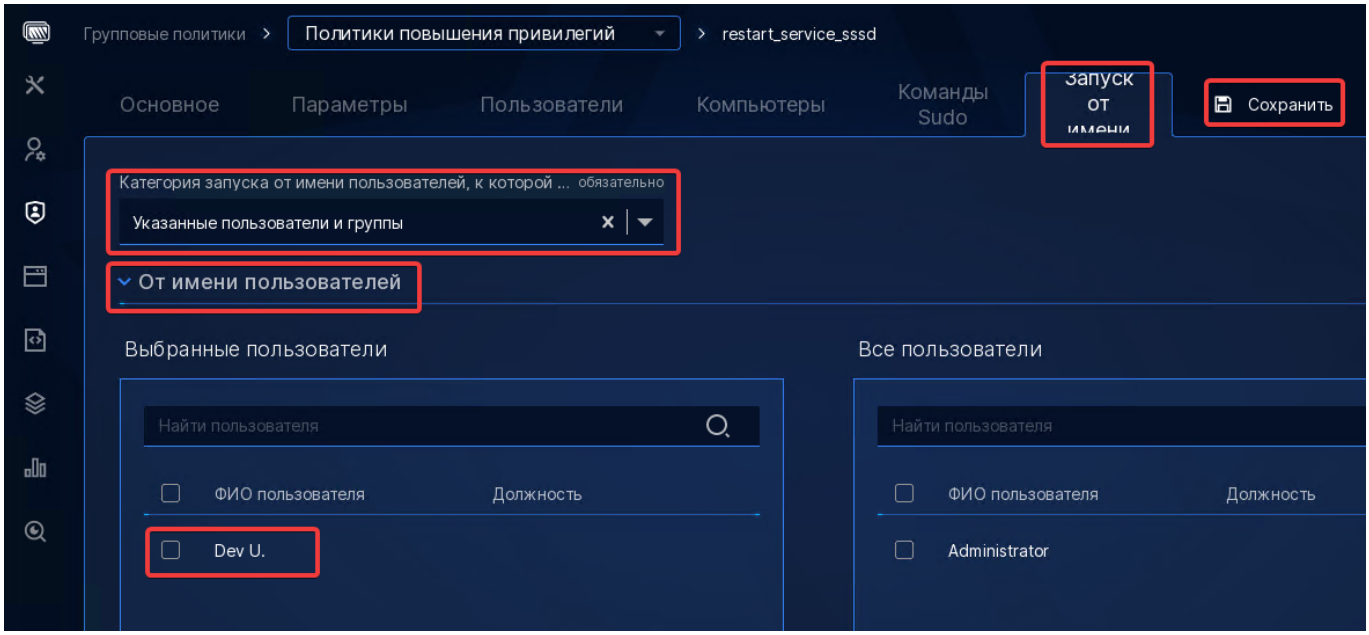


Рисунок 6.47 – Команды sudo

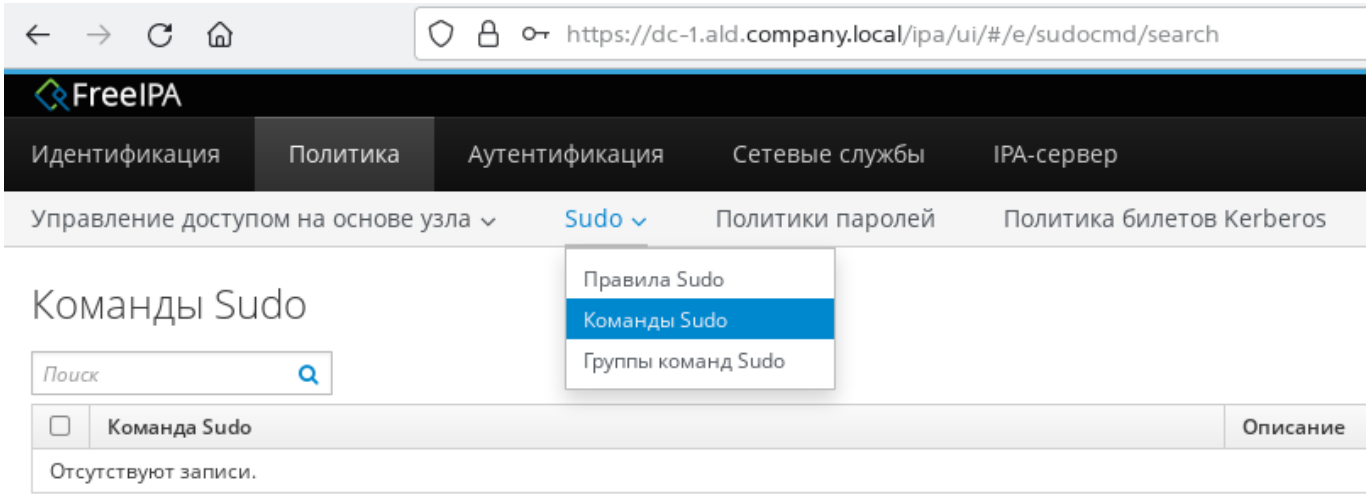
6. Раздел «Запуск от имени» По умолчанию команды sudo запускаются от имени суперпользователя root в контексте его первичной группы, но это поведение можно изменить с помощью ключей -u и -g. На этой вкладке можно определить список пользователей и групп, от имени которых пользователь сможет действовать.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



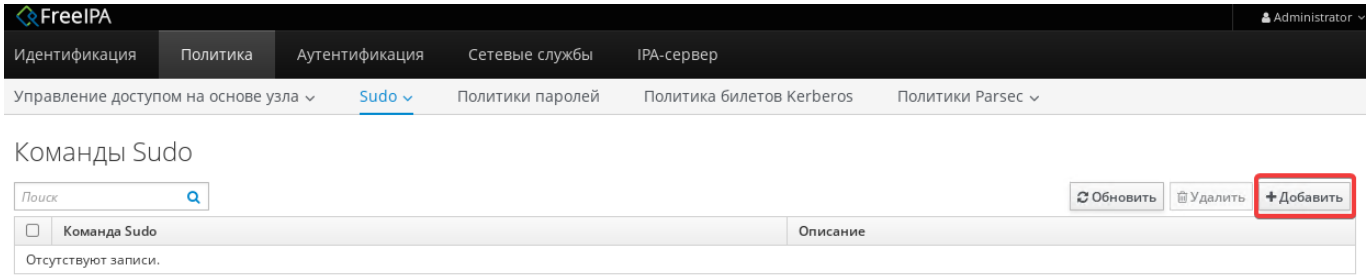
6.5.4.2. Через web-интерфейс FreeIPA

Создание команды

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

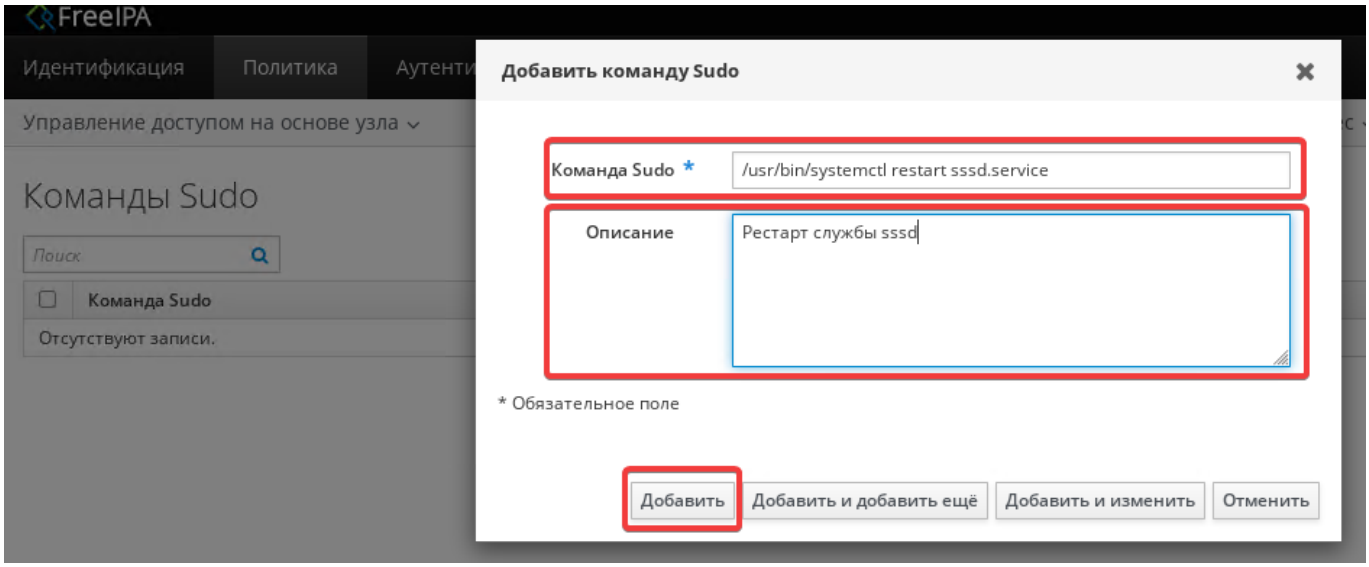


2. Нажать кнопку «+ Добавить»



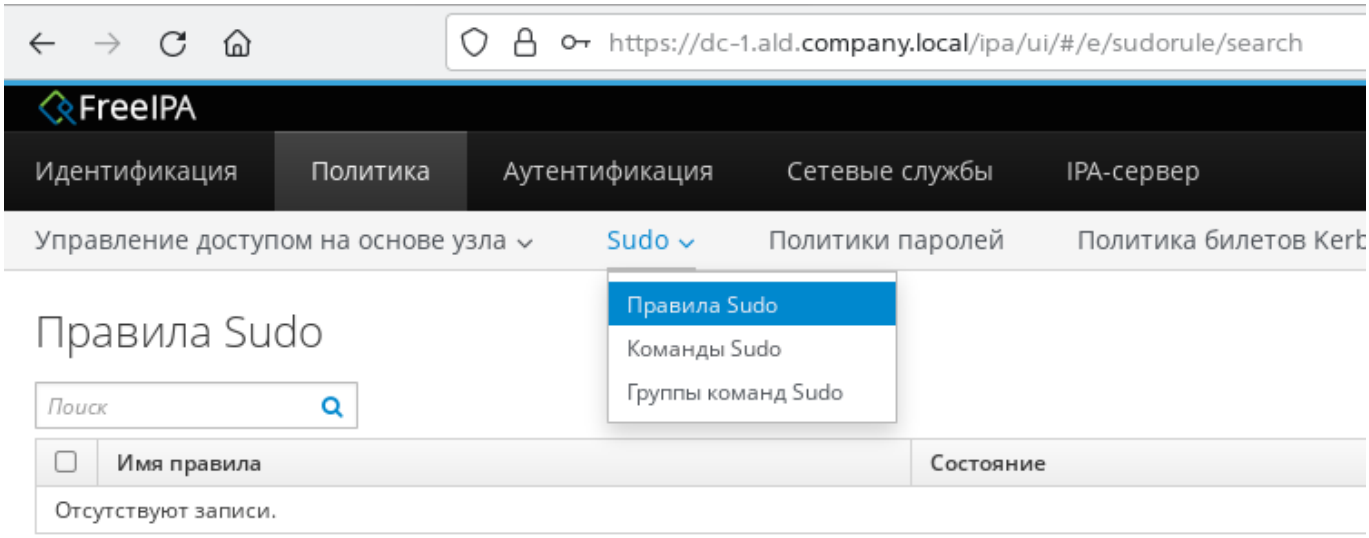
3. В открывшемся окне ввести команду `/usr/bin/systemctl restart sssd.service` и её описание, нажать кнопку «Добавить».

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



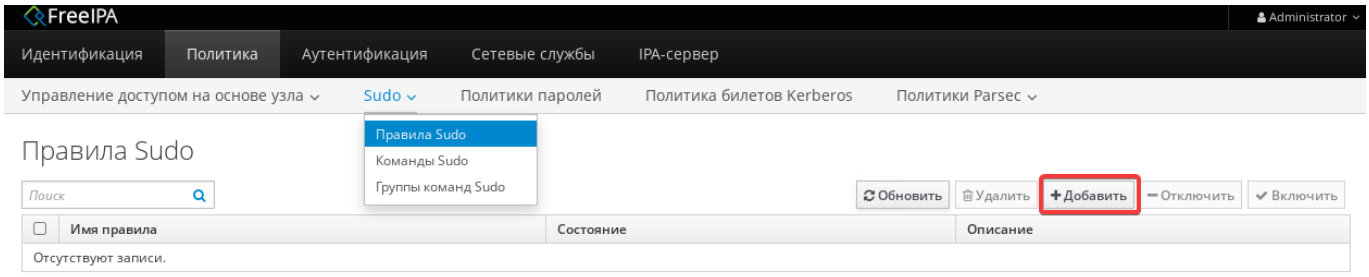
Создание правила

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

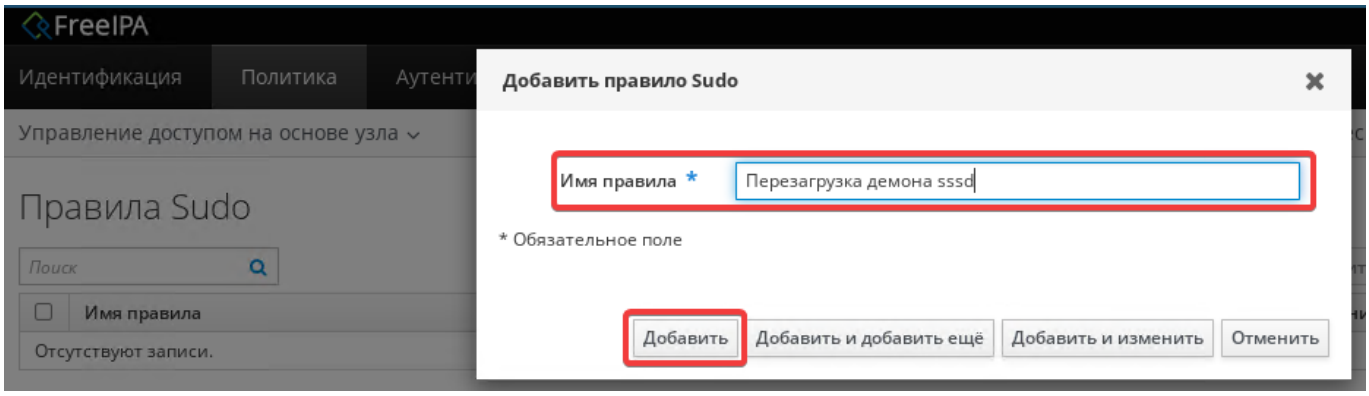


2. Нажать кнопку «+ Добавить»

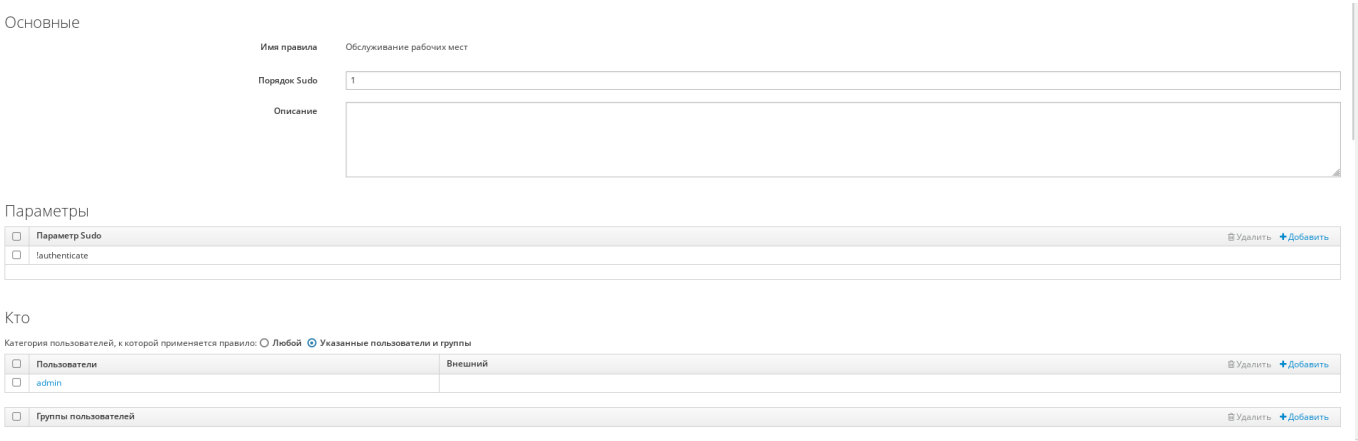
Важно: В имени правила можно использовать `a-z`, `A-Z`, `0-9`, `-_./~`.

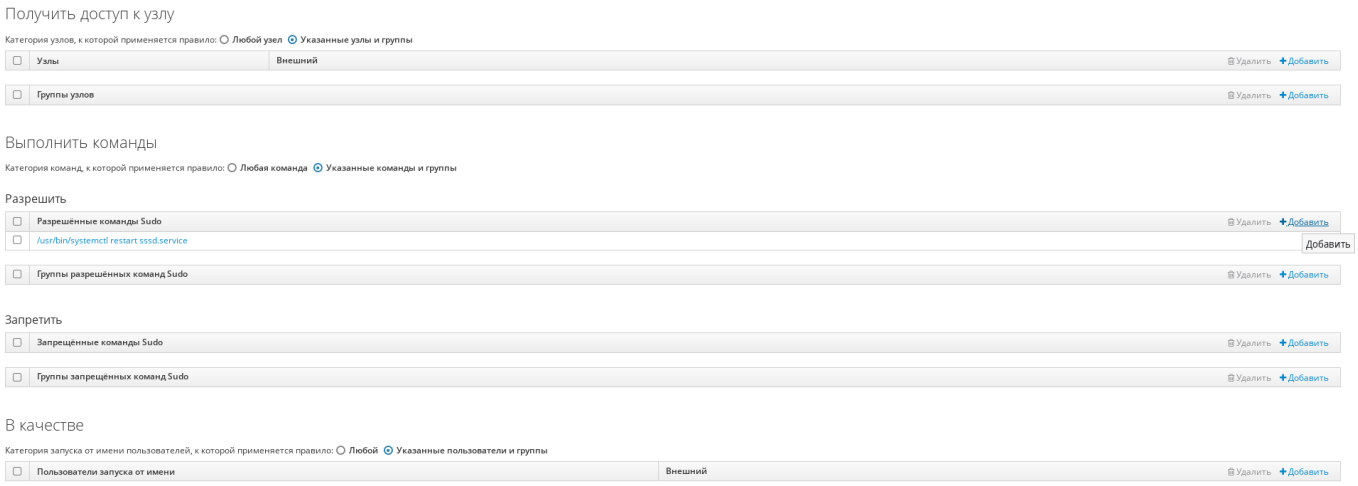


3. В открывшемся окне ввести имя правила Sudo (обязательно) и нажать кнопку «Добавить».



4. Указать необходимые параметры





6.5.4.3. Через терминал

Создание команды

Создать 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

---

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

- `/etc/usr/systemctl restart sssd.service` - полный путь к утилите и разрешенные параметры вызова
- desc – ключ, который позволяет задать описание команды “Рестарт службы sssd”

Создание правила

1. Создать 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.

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

---

2. Добавить пользователя в правило:

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

3. Добавить в правило целевые хосты:

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

4. Добавить команду в правило:

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

5. Проверить результат:

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

6.5.5. Отладка правил SUDO

6.5.5.1. 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\:/
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
 ↪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
```

6.5.5.2. 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,
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
↪554801108,554801393,554801503,554802131,554802244,554807670
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_parseln @ ./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)
```

Подошли эти правила или нет.

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

6.5.5.3. Журнал отладки SSSD

Чтобы включить отладку SSSD в файле `/etc/sssd/sssd.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
```

(продолжение на следующей странице)

```
...
[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/sssd/sssd_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.

```
[sssd[be[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/sssd/sssd_sudo.log`, с помощью которого можно ответить на следующие вопросы.

Как много правил было получено от службы SSSD.

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

Какой фильтр был применен при поиске кеша SSSD.

```
[sudosrv_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)...)'
```

6.5.6. Лучшие практики

6.5.6.1. 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 можно воспользоваться одним из следующих способов:

1. Заранее создать 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. Удалить 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.

6.5.6.2. Особенности использования символов 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:

```
localuser@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/$qVDReAiNXXpPDgQW1/
↪e26C7bAvRaMrwizV924KN4YYXDgPnYDlWqvpETfk29S9q7LKlxZeO7qA/.OcCO2XG3U/
↪:19296:0:99999:7:::
localuser:$gost12512hash$OEbYsS/bODT9ux.t
↪$0CY2yXvZTdZ3LO3cCfD7KI61DQiuOZ6bHEvzn3YXWZLjO.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* *
```

6.5.6.3. Запрет на 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 помешает 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/$qVDReAiNXXpPDgQW1/
↪e26C7bAvRaMrwizV924KN4YYXDgPnYDlWqvpETfk29S9q7LKlxZeO7qA/.OcCO2XG3U/
↪:19296:0:99999:7:::
localuser:$gost12512hash$OEbYsS/bODT9ux.t
↪$0CY2yXvZTdZ3LO3cCfD7KI61DQiuOZ6bHEvzn3YXWZLjO.vcNU6pQQEz/hhWXHmuVCQbMHFWtl.
↪YmTdoctUZq.:19518:0:99999:7:::
```

6.5.6.4. Использование группы пользователей, комментирование

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 удается лучше следить.

6.5.6.5. 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.

6.6. Инструкция по обеспечению безопасной работы в домене ALD Pro: политики паролей

Пароли являются самым простым, но при этом не самым безопасным способом аутентификации, поэтому в работе с паролями пользователи должны придерживаться

определенных правил и политики паролей помогают гарантировать, что эти правила соблюдаются.

6.6.1. Пароли пользователей в домене

Пароль представляет из себя набор символов, который известен только самому пользователю и проверяющей стороне, поэтому, если пользователь может предъявить доказательство того, что пароль ему известен, это является подтверждением аутентичности пользователя, что он именно тот, за кого себя выдает.

В открытом виде пароли не хранят, в базу данных записывают хэши, и так как в домене ALD Pro (FreeIPA) используется сразу несколько разных механизмов аутентификации, у пользователей есть несколько хэшей:

- userPassword хранит PBKDF2_SHA256 хэш, который используется для обычной LDAP аутентификации, так называемой привязки (Bind). Во избежание перехвата пароля этот способ аутентификации рекомендуют использовать только с шифрованием траффика (LDAPS или LDAP+StartTLS).
- krbPrincipalKey хранит AES хэши, которые используются для аутентификации по протоколу Kerberos V5. Это наиболее рекомендуемый способ аутентификации с использованием паролей, т.к. он обеспечивает наибольший уровень безопасности.
- ipaNTHash хранит MD4 хэш, который используется для NTLM аутентификации. Этот механизм аутентификации необходим для интеграции с MS AD, простой аутентификации на файловом сервере при обращении к нему по IP адресу и интеграции с некоторыми другими внешними системами.

Для изменения пароля новое значение следует записать открытым текстом в атрибут userPassword, сервер автоматически сгенерирует все необходимые ключи и запишет в базу уже хешированные значения. В силу такой особенности работы сервера записывать в каталог уже хешированные значения запрещено. Обойти это ограничение можно только при создании новых пользователей, если сервер будет переведен в режим миграции.

6.6.2. Что такое политики паролей

Пароли, к сожалению, являются не самым безопасным механизмом аутентификации, так как их можно подобрать или перехватить, поэтому в работе с паролями необходимо следовать определенным правилам, или так называемым политикам, которые повышают

уровень безопасности учетных записей в домене: пароли нужно периодически обновлять, использовать следует достаточно длинные комбинации, состоящие из разных категорий символов, и т.п.

Чем более строгие требования задает политика паролей, тем сложнее злоумышленнику подобрать пароль и воспользоваться результатами успешной атаки. Но, вместе с тем, и пользователям сложнее работать в таком домене, поэтому для разных групп пользователей следует устанавливать разные требования, обеспечивающий компромисс между удобством и безопасностью.

6.6.3. Механизм работы политик паролей

Механизм политик паролей в домене ALD Pro (FreeIPA) очень гибкий: для каждой группы пользователей можно создать свою собственную политику паролей. Список политик с их приоритетами хранится в контейнере с DN `cn=cosTemplates,cn=accounts,{base_dn}=ald,{base_dn}=company,{base_dn}=lan`, параметры политик вынесены отдельно в `cn=kerberos,{base_dn}=ald,{base_dn}=company,{base_dn}=lan`. Связь между записями осуществляется через значение атрибута krbPwdPolicyReference. При удалении группы пользователей все связанные с ней записи политики паролей удаляются автоматически.

Учитывая, что пользователь может входить сразу в несколько групп, алгоритм проверки выглядит следующим образом:

- Из «cn=cosTemplates, …» отбираются политики, под действие которых попадает текущий пользователь в соответствии с его участием в группах. Параметры политики берутся из «cn=kerberos, …» по ссылке из атрибута krbPwdPolicyReference.
- Если на пользователя не распространяется действие ни одной политики, ему будет назначена глобальная политика по умолчанию (global_policy).
- Если некоторый пользователь попадает под действие сразу нескольких политик, то выбирается одна из них, у которой будет наименьшее значение по приоритету, параметры политик не суммируются, см. таблицу 1.

Таблица 1. Выбор политики в зависимости от приоритета

Параметр	Политика для группы A (приоритет 0)	Политика для группы B (приоритет 1)	Результат (используются параметры для группы A)
Максимальный срок действия	60 дней	90 дней	60 дней
Минимальная длина	10 символов	0 (без ограничений)	10 символов

Проверки паролей ограничены возможностями MIT Kerberos, поэтому они поддерживают тот же самый набор параметров, см. таблицу 2

Таблица 2. Параметры политик паролей

Параметр политики	Значение глобальной политики по умолчанию
Максимальный срок действия задает период в количестве дней, в течение которого система не будет требовать смены пароля	krbMaxPwdLife = 90 Пароль активен 90 дней, после чего пользователю будет предложено сменить его
Минимальный срок действия задает период в часах, в течение которого система будет запрещать повторную смену пароля	krbMinPwdLife = 1 После смены пароля, пользователь должен подождать 1 час перед повторной сменой
Размер журнала определяет количество предыдущих паролей, которые нельзя использовать повторно	krbPwdHistoryLength = 0 Запрет на повторное использование паролей не налагается
Классы символов – этот параметр указывает, сколько разных классов символов должно быть использовано в пароле. Все возможные символы подразделяются на следующие пять классов: цифры, буквы нижнего регистра, буквы верхнего регистра, символы UTF-8. Все остальные символы, не вошедшие ни в одну из предыдущих групп, например, ! « ### $ % и т.д. Использование одного и того же символа более двух раз подряд уменьшает количество классов на один, например, у пароля «Secret11pwd» будет три класса (большие буквы + маленькие буквы + цифры), а у пароля «Secret111pwd» их станет два (минус штраф за повторы символа «1»). Если повторяющиеся символы окажутся в конце пароля, то последний из них не будет учитываться, поэтому на пароль «Secretpwd111» штраф налагаться не будет.	krbPwdMinDiffChars = 0 Значение по умолчанию – 0. Это говорит об отсутствии каких либо требований к сложности пароля
Минимальная длина задает минимально допустимое количество символов в пароле	krbPwdMinLength = 8 Пользователь не может использовать пароль короче 8 символов
Максимальное количество ошибок определяет, сколько раз пользователь может неправильно ввести пароль, прежде чем его аккаунт будет временно заблокирован. Блокировка выполняется только на текущем контроллере, на другие сервера эта информация не передается.	krbPwdMaxFailure = 6 Пользователь будет заблокирован после 7 неверно введенных паролей подряд
Интервал сброса ошибок задает период в секундах, по истечении которого счетчик неудачных попыток входа будет сброшен	krbPwdFailureCountInterval = 60 Если после 6 неудачных попыток введения пароля подряд пользователь подождет 1 минуту, у него будет еще 6 попыток до временной блокировки учетной записи
Длительность блокировки задает период в секундах, в течение которого пользователь не сможет выполнить аутентификацию в домене. Блокировка накладывается после превышения количества разрешенных неудачных попыток входа. Блокировка выполняется только на текущем контроллере, на другие сервера эта информация не передается.	krbPwdLockoutDuration = 600 Заблокированный пользователь не сможет выполнить вход в систему в течение 10 минут

6.6.4. Создание политики паролей

6.6.4.1. Через портал управления

Откройте страницу «Групповые политики > Политики паролей» и нажмите кнопку «+ Новая политика паролей». Заполните поля «Наименование группы пользователей», «Приоритет» и нажмите кнопку «Сохранить».

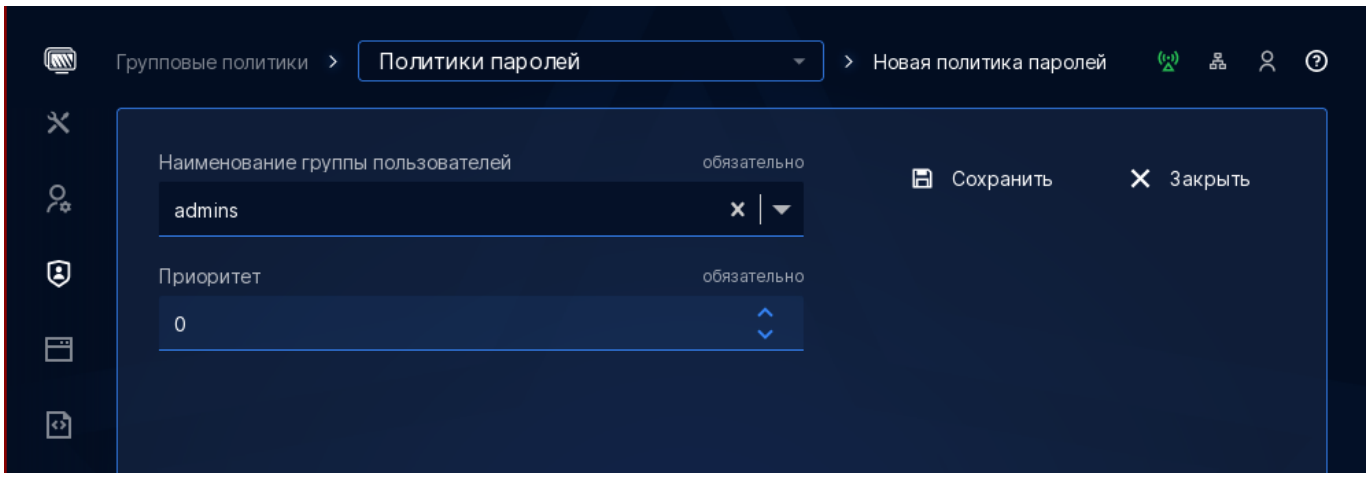


Рисунок 6.48 – Создание политика паролей.

Далее вам станет доступна страница управления политикой, где вы можете задать необходимые настройки.

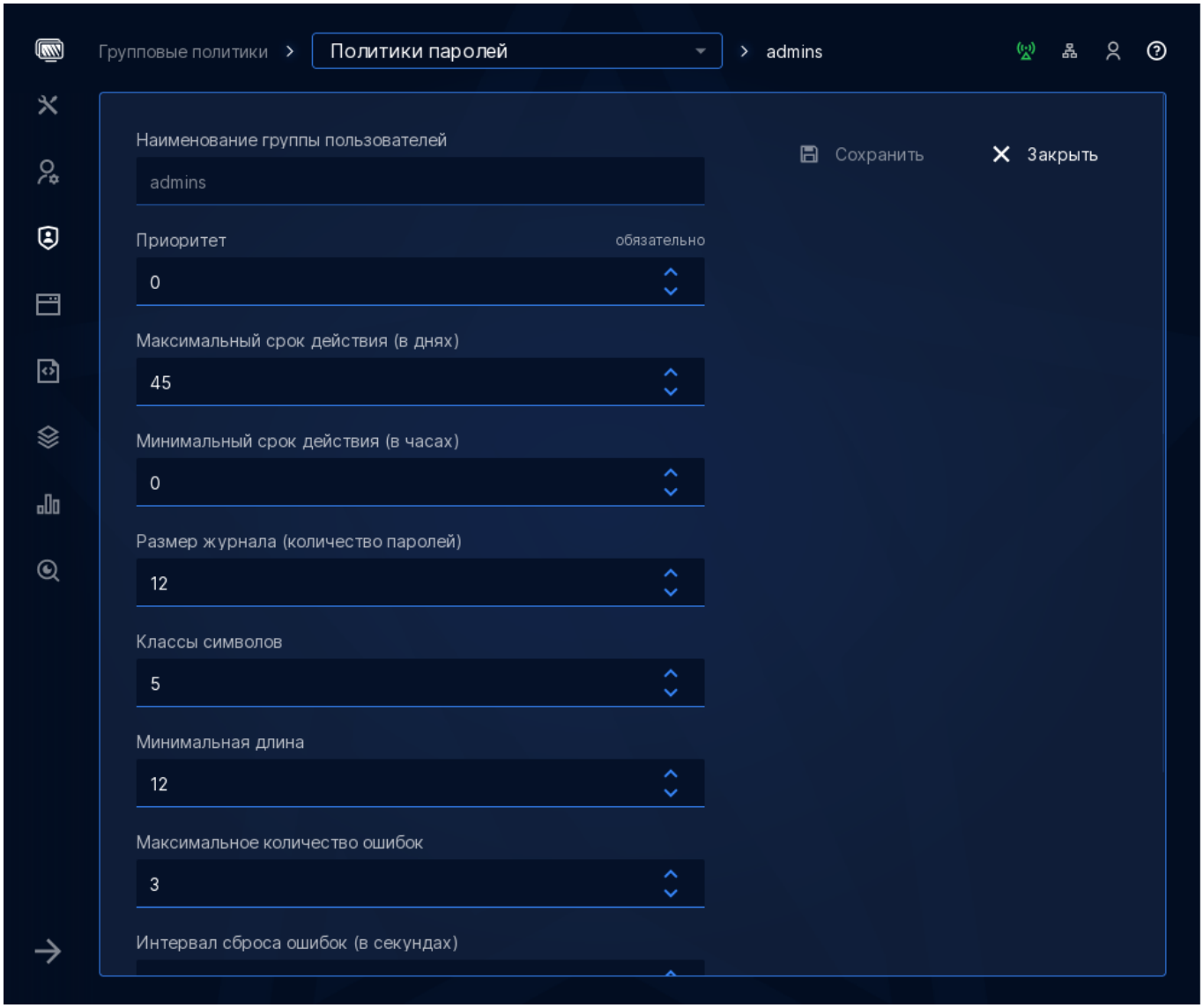


Рисунок 6.49 – Настройка политики паролей

6.6.4.2. Из командной строки

Для создания политики паролей воспользуйтесь командой pwpolicy-add

```
$ ipa pwpolicy-add admins --priority=0 --maxlife=45 --minlife=0 --history=12 \
↪
--minclasses=5 --minlength=12 --maxfail=3 --failinterval=120 --
↪lockouttime=1200
  Группа: admins
  Максимальный срок действия (в днях): 45
  Минимальный срок действия (в часах): 0
  Размер журнала : 12
  Классы символов: 5
```

(продолжение на следующей странице)

```
  Минимальная длина: 12
  Приоритет: 0
  Максимальное количество ошибок: 3
  Интервал сброса ошибок: 120
  Длительность блокировки: 1200
```

где,

–maxlife= - Максимальный срок действия в днях;

–minlife= - Минимальный срок действия в часах;

–history= - Размер журнала;

–minclasses= - Классы символов;

–minlength= - Минимальная длина;

–priority= - Приоритет политики;

–maxfail= - Максимальное количество ошибок;

–failinterval= - Интервал сброса ошибок в секундах;

–lockouttime= - Длительность блокировки в секундах.

Чтобы изменить параметры уже существующей политики, воспользуйтесь командой `ipa pwpolicy-mod`:

```
$ ipa pwpolicy-mod admins --maxlife=30
  Группа: admins
  Максимальный срок действия (в днях): 30
  Минимальный срок действия (в часах): 0
  Размер журнала : 12
  Классы символов: 5
  Минимальная длина: 12
  Приоритет: 0
  Максимальное количество ошибок: 3
  Интервал сброса ошибок: 120
  Длительность блокировки: 1200
```

Следует учитывать, что срок действия пароля проверяется не по значению maxlife в

политике, а по значению атрибута krbPasswordExpiration, которое устанавливается пользователю при изменении пароля, поэтому изменение параметра в политике сразу ни на что не повлияет. Чтобы принудительно изменить пользователю значение атрибута krbPasswordExpiration вы можете воспользоваться командой `user-mod`:

```
$ ipa user-mod admin --password-expiration 20230528010101Z
----------------------------
Изменён пользователь "admin"
----------------------------
  Имя учётной записи пользователя: admin
  Фамилия: Administrator
  Домашний каталог: /home/admin
  Оболочка входа: /bin/bash
  Псевдоним учётной записи: admin@ALD.COMPANY.LAN, root@ALD.COMPANY.LAN
  Окончание действия пароля пользователя: 20230528010101Z
  UID: 959800000
  ID группы: 959800000
  Учётная запись отключена: False
  Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,{base_dn}
↪=ald,{base_dn}=company,{base_dn}=lan
  Пароль: True
  Участник групп: trust admins, lpadmin, admins
  Роли: ALDPRO - Main Administrator
  Доступные ключи Kerberos: True
```

Срок действия пароля задается в формате временной метки, где:

- 2023 – год
- 05 – месяц
- 28 – день месяца;
- 010101 – часы, минуты, секунды
- Z – часовой пояс. Точность до секунд не имеет большого значения, поэтому обычно используют время по нулевому (Zero) меридиану.

Проверить текущее значение можно командой user-show

```
$ ipa user-show admin --raw --all | grep krbPasswordExpiration
  krbPasswordExpiration: 20230528010101Z
```

6.7. Доверительные отношения

6.7.1. Инструкция по работе двусторонних доверительных отношений между MS AD и ALD Pro

6.7.1.1. Введение

Для удобства администрирования в организации может быть несколько доменов.

Например, домен MS AD и домен ALD Pro. В домене MS AD будут находится компьютеры с операционной системой Windows. Домен ALD Pro будут содержать компьютеры с AstraLinux.

Для корректной работы необходима возможность гибридной работы пользователей сразу в двух доменах с помощью механизма доверительных отношений.

Ранее в ALD Pro уже были реализованы доверительные отношения MS AD –> ALD Pro. Но для полноценной работы, необходимы двусторонние доверительные отношения.

Популярный кейс:

Ранее организация работала в домене MS AD, и в рамках миграции инфраструктур был развернут домен ALD Pro (ald.company.lan). Для беспрерывной работы предполагается постепенный перевод сервисов и пользователей на работу в новом домене. В течение некоторого времени необходимо обеспечивать гибридную работу пользователей сразу в двух доменах с помощью механизма доверительных отношений.

6.7.1.2. Как работали доверительные отношения MS AD и ALD Pro ранее

До внедрения функционала глобального каталога и двусторонних доверительных отношений, было реализовано в полном объеме только одностороннее доверие. Домен ALD Pro полностью доверял домену под управлением MS AD, следовательно пользователи MS AD имели возможность доступа на клиентские компьютеры и сетевые ресурсы в домене ALD Pro.

Проблема доверия MS AD с ALD Pro

Доступ на клиентские машины в домене MS AD осуществляется только с машинами под ОС Windows. Доступ через SSSD-client осуществляется только путем ввода машины в два домена, что возможно и без доверия.

При разграничении доступа к ресурсам не было возможности сопоставить пользователя ALD Pro с атрибутом, и осуществлять поиск пользователей.

6.7.1.3. Как работают теперь

Технические требования

1. Версии Windows Server работающие с глобальным каталогом ALD Pro
    - Windows Server 2008 R2
    - Windows Server 2012
    - Windows Server 2012 R2
    - Windows Server 2016
    - Windows Server 2019
2. ALD Pro должен быть версии 2.1.0 и выше, с установленным глобальным каталогом.
3. Версия Astra Linux не ниже 1.7.4.

Глобальный каталог

Для работы с глобальным каталогом, его необходимо установить согласно разделу *Установка и обновление Глобального Каталога и Модуля Синхронизации*.

6.7.1.4. Настройка двусторонних доверительных отношений

Настройка и проверка перенаправления DNS ALD Pro

Для работы доверительных отношений с компьютеров из домена ALD.company.lan должны разрешаться имена компьютеров из домена WIN.company.lan и наоборот, нужно сделать

взаимное перенаправление DNS-зон.

Настройка ALD Pro

Добавление зоны перенаправления можно сделать из графического интерфейса «Роли и службы сайта Служба разрешения имен Перенаправление запросов».

1. Имя зоны = имя домена MS AD
2. Глобальные перенаправители = IP-адрес контроллера домена MS AD, с которым устанавливаются доверительные отношения
3. Остальные поля и параметры оставить без изменений

После на контроллере домена необходимо выполнить команды:

```
sudo net conf setparm global "restrict anonymous" "0"
sudo aldproctl restart -i
```

Настройка MS AD

Настройка контроллера домена MS AD осуществляется согласно официальным инструкциям к MS AD. (Пуск -> Оснастка “DNS”):

1. Контекстное меню к “Серверы условной пересылки” -> Создать сервер условной пересылки.
2. В поле “DNS-домен” ввести имя домена ALD Pro.
3. Добавить IP-адрес контроллера домена ALD Pro в блоке “IP-адреса основных серверов:”

Установка двусторонних доверительных отношений между доменами

Создать двусторонние доверительных отношений осуществляется на портале управление ALD Pro на вкладке **Управление доменом → Интеграция с доменом → Доверительные отношения** (см. Справочный центр на портале или Руководство пользователя).

После успешного создания доверительных отношений с доменом MS AD на контроллере домена ALD Pro необходимо выполнить следующие команды:

```
rm /var/lib/sss/db/* && systemctl restart sssd
net conf setparm global "restrict anonymous" "2"
sudo aldproctl restart -i
```

Выполнение данных команд предоставляет УЗ Администратора доступ к атрибуту krbPrincipalKey у сервисного принципала cifs, что необходимо для успешного создания двусторонних доверительных отношений.

Отключение FAST аутентификации

Для доступа к ресурсам windows с аутентификацией по kerberos (IIIS, cifs, принтеры с общим доступом по smb) необходимо отключить у клиентских компьютеров FAST аутентификацию, так как windows её не поддерживает. Для этого на всех клиентах ALD Pro, где предполагается доступ, необходимо настроить глобальную политику **Групповые политики → Параметры компьютеров → Безопасность → FAST аутентификация**. Выставить параметр never.

Важно: Установка этого параметра понижает безопасность.

При первом применении может понадобиться перезагрузка сервиса sssd.

Проверка двусторонних доверительных отношений

Проверить двусторонние доверительные отношения можно следующими способами:

1. Авторизация на компьютере домена MS AD;
2. Авторизация на компьютере домена ALD Pro;
3. Создание общей папки.

Авторизация на компьютере домена MS AD

Необходимо авторизоваться на компьютере домена MS AD, используя учетную запись пользователя ALD Pro. Логин должен быть указан полностью, включая имя домена:

```
<имя_пользователя ALD Pro>@<имя_домена>
```

В результате пользователь под учетной записью ALD Pro авторизуется на компьютере домена MS AD.

Авторизация на компьютере домена ALD Pro

Необходимо выбрать нужный домен MS AD и авторизоваться на компьютере домена ALD Pro, используя учетную запись пользователя MS AD. В поле **Имя пользователя** необходимо указать имя пользователя, без имени домена.

В результате пользователь под учетной записью MS AD авторизуется на компьютере домена ALD Pro.

Создание общей папки

а) Создание общей папки на компьютере домена MS AD

Необходимо создать папку **C:Common** и добавить в нее хотя бы один файл. Открыть доступ к созданной папке из контекстного меню **Sharing -> Share -> Network access**. Настройки оставить по умолчанию.

В окне **Common PropertiesSharingAdvanced SettingsPermissions** будет отображена информация, что по умолчанию на уровне SMB все пользователи, включая пользователей ALD Pro, имеют полные права.

Но доступ к файлам регулируется также на уровне NTFS разрешений, которые настраиваются на вкладке **Common PropertiesSecurity**. Можно предоставить доступ к общей папке всем аутентифицированным пользователям.

В результате пользователи ALD Pro могут редактировать файлы, находящиеся в папке **Common**.

б) Создание общей папки на компьютере домена ALD Pro

Необходимо создать новое сетевое место, которое соответствует папке **Common** в файловом менеджере. Для этого выбрать **Сеть -> Создать сетевое место**. Указать **Название** и **Адрес** в формате:

```
smb://<полное наименование контроллера домена MS AD, на котором создана папка>
↪/<Наименование папки из MS AD>
```

В результате пользователи MS AD могут редактировать файлы, находящиеся в папке **Common**.

6.7.1.5. Заключение

Благодаря внедрению ряда решений, включая глобальный каталог ALD Pro, есть возможность настраивать двусторонние доверительные отношения. В отличие от MS AD, в ALD Pro направления доверия MS AD → ALD Pro и ALD Pro → MS AD реализованы разными механизмами.

Двусторонние доверительные отношения, предоставляют возможность общаться доменам ALD Pro и MS AD, решая ряд важных задач:

- Авторизация пользователей доверенных доменах на рабочих станциях;
- Доступ к сетевым ресурсам пользователей доверенного домена (веб сервер, файловый сервер, базы данных и т.д.);
- Разграничение доступа к ресурсам доверенных доменов.

6.7.2. Инструкция по присоединению системы Windows к FreeIPA Realm без Active Directory

Большинство системных администраторов имеют опыт использования компьютеров под управлением ОС Windows в домене Active Directory, но в качестве источника идентификационной информации компьютеры Windows могут использовать и область Kerberos от ALD Pro (FreeIPA). Способ, которым можно ввести Windows в домен FreeIPA был известен давно, но не получил широкого распространения, т. к. содержал существенный недостаток — внутри операционной системы пользователь действовал от имени локальной учетной записи, которую нужно было создавать заранее. В настоящей инструкции описан способ, который позволяет обеспечить вход в операционную систему именно доменной учетной записью, сохраняя ее SID, участие в группах и Kerberos билеты, что открывает новые возможности по решению задач гибридного развертывания и миграции.

6.7.2.1. Что такое Active Directory?

Active Directory - это база данных специального назначения со службами, которые позволяют пользователям подключаться к привязанным к ней сетевым ресурсам. В этой базе данных хранится важная информация о вашей среде, такая как пользователь и компьютеры, которым разрешено устанавливать подключения. Поскольку Active Directory является продуктом Microsoft, он часто используется в среде Windows.Он обеспечивает эти функциональные возможности путем хранения пользовательских, групповых, хостовых и любых других данных, необходимых для управления безопасностью сетевых компьютеров. Что делает этот инструмент более совершенным, так это его простой и понятный в использовании веб-интерфейс, а также командная строка для администрирования.

6.7.2.2. Что такое FreeIPA?

FreeIPA - это бесплатное интегрированное решение для управления информацией о безопасности с открытым исходным кодом, спонсируемое RedHat. Он сочетает в себе MIT Kerberos, Dogtag (систему сертификатов), NTP, DNS и сервер каталогов 389. Основная цель состоит в том, чтобы обеспечить функциональность, аналогичную Active Directory. Его можно использовать для обеспечения централизованной аутентификации, авторизации и получения информации об учетной записи.

FreeIPA не является повторной реализацией Microsoft Active Directory и может работать независимо. Основное различие между ними заключается в том, что FreeIPA ориентирована на Linux и другие системы, соответствующие стандартам POSIX, в то время как Active Directory является инструментом Windows.

ALD Pro(FreeIPA) может быть интегрирован для работы с Active Directory путем установления доверия между двумя службами. Но в этом руководстве настраивается система Windows на использование области FreeIPA для аутентификации пользователей без Active Directory.

Чтобы достичь этого, необходимо выполнить приведенные ниже шаги.

6.7.2.3. Требования к настройке

В этом руководстве будут настроены две системы со статическими IP-адресами и именами хостов:

TASK	HOSTNAME	IP_ADDRESS
FreeIPA server(ALD Pro)	ald01.ald.dom	192.168.88.210
Windows client	Win10test.ald.dom	192.168.88.76

6.7.2.4. Порядок присоединения

Шаг 1. Установите имя хоста в Windows

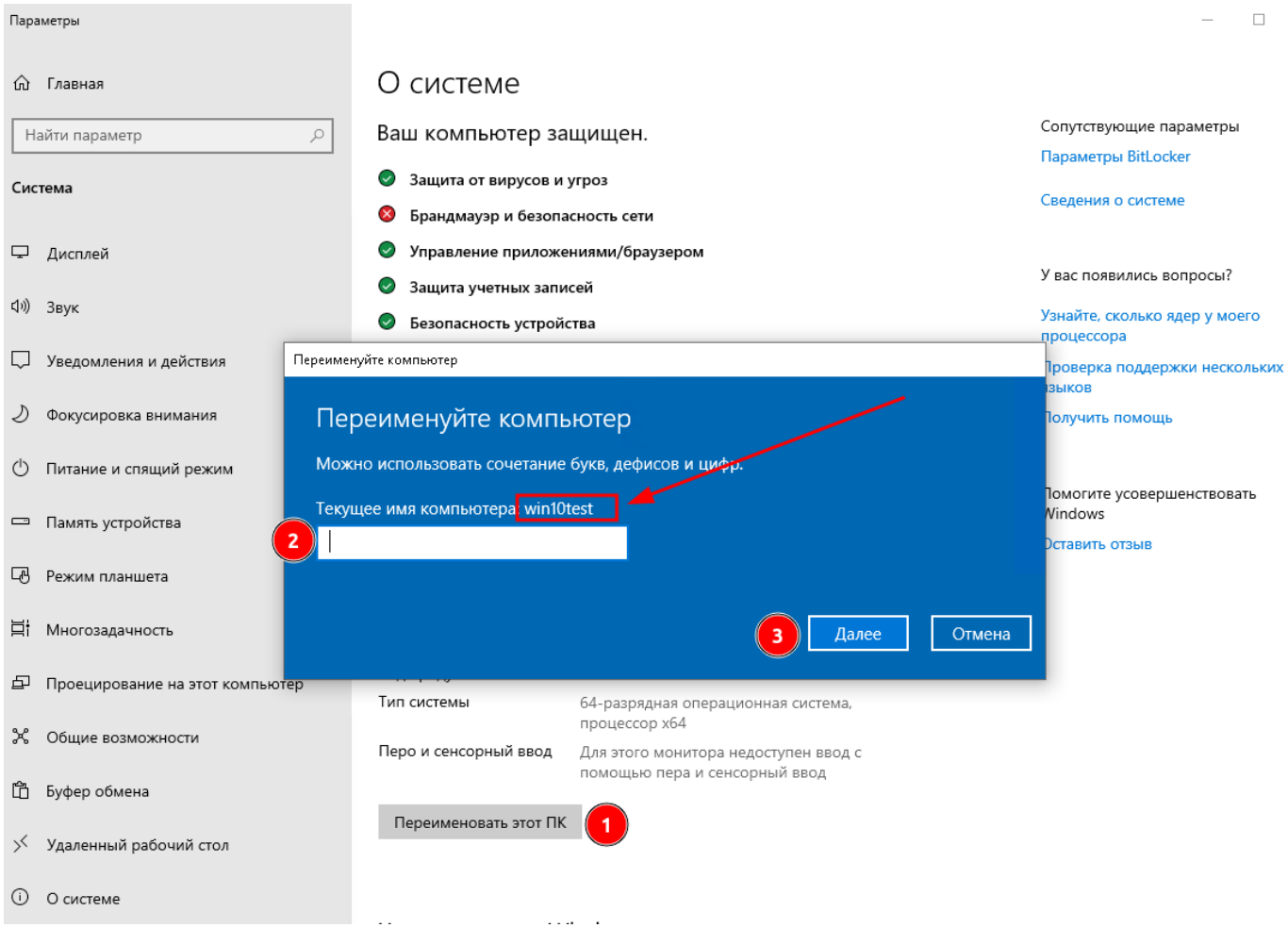


Рисунок 6.50 – Установка имени хоста

Шаг 2. Настроить сеть на Windows машине, в данном примере устанавливается статический IP адрес.

Нажать Win+R и запустить ncpa.cpl для настройки сети Сначала настраивается DNS таким образом, чтобы он мог разрешать имя ALD Pro(IPA)-сервера.

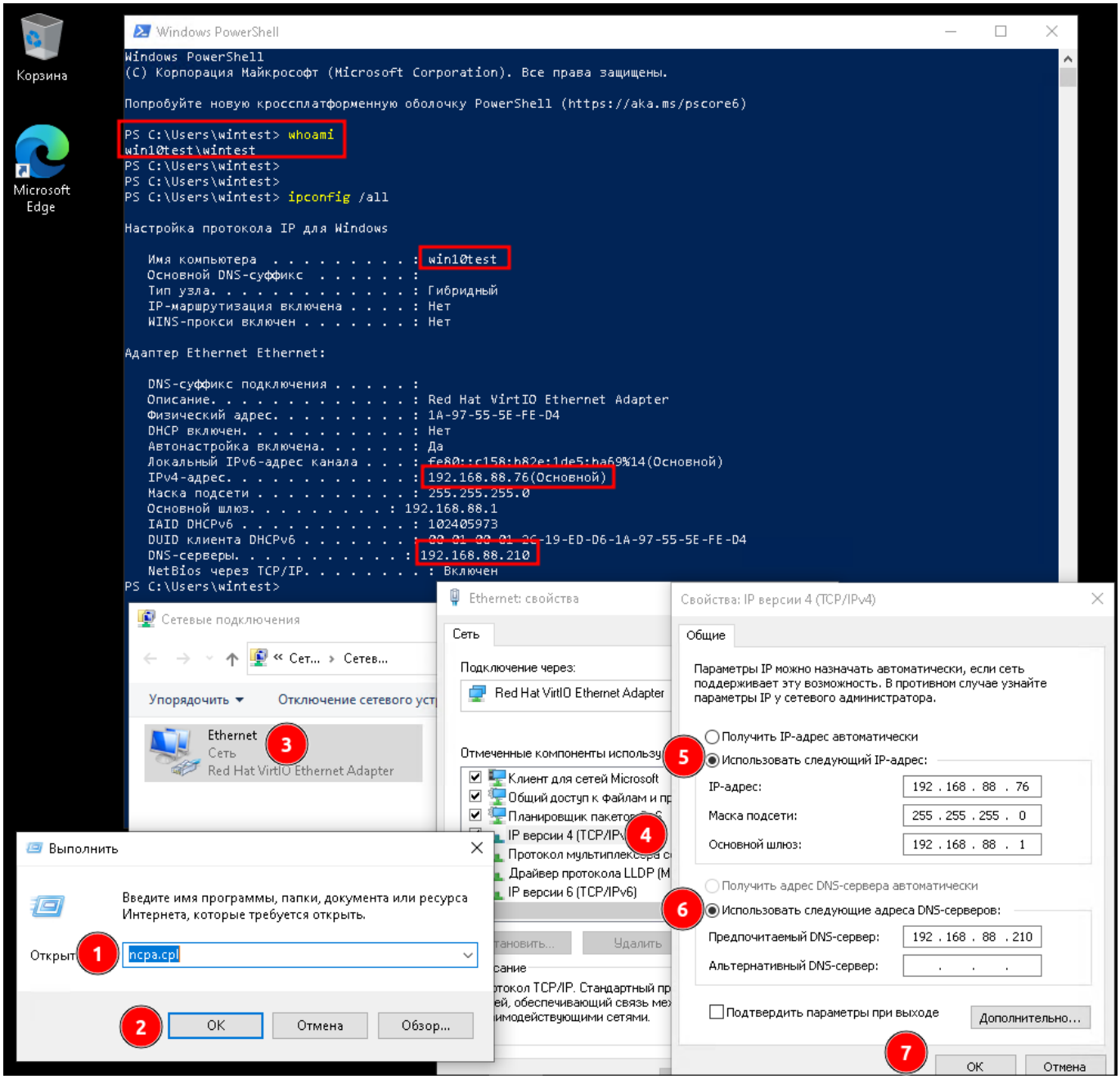


Рисунок 6.51 – Настройка сети на Windows машине 1

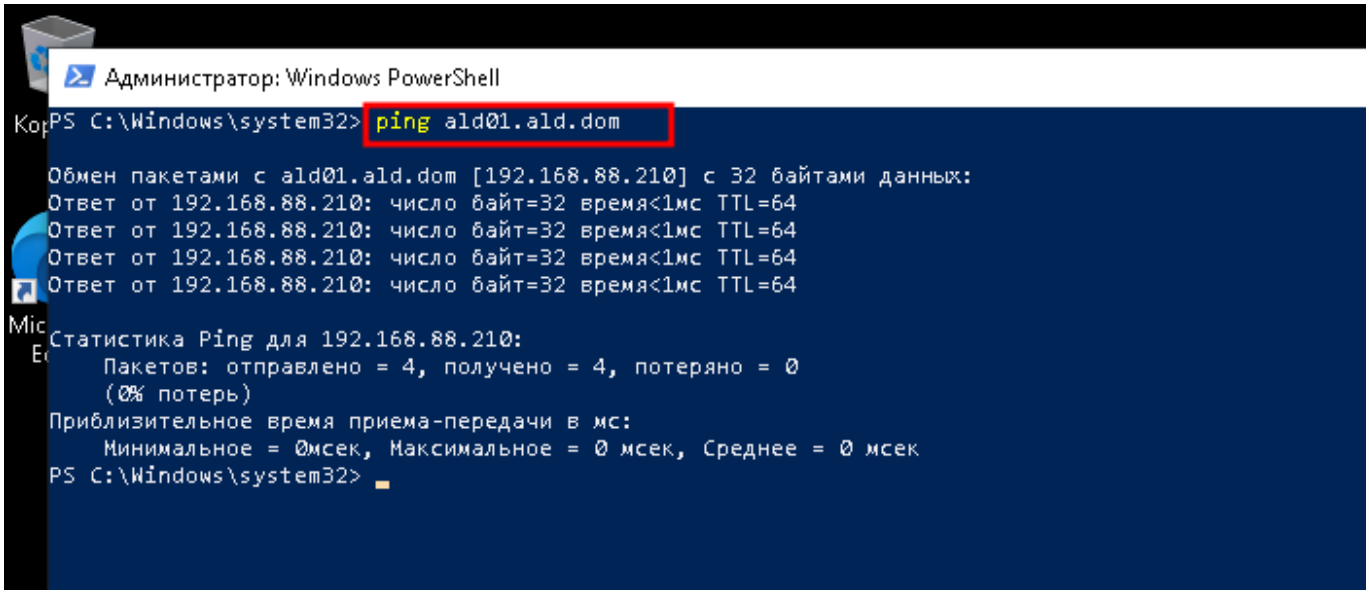


Рисунок 6.52 – Настройка сети на Windows машине 2

Шаг 3. Настройки даты/времени

На стороне ALD Pro(FreeIPA) необходимо убедиться, что время не расходится с внешним NTP сервером

```
# chronyc tracking
root@ald01:~# chronyc tracking
Reference ID    : BCE109A7 (188.225.9.167)
Stratum         : 3
Ref time (UTC)  : Tue Jun 13 10:08:13 2023
System time     : 0.000034270 seconds slow of NTP time
Last offset     : -0.000079430 seconds
RMS offset      : 0.000206107 seconds
Frequency       : 9.831 ppm slow
Residual freq   : -0.001 ppm
Skew            : 0.061 ppm
Root delay      : 0.017628554 seconds
Root dispersion : 0.003091000 seconds
Update interval : 1040.0 seconds
Leap status     : Normal
root@ald01:~#
root@ald01:~#
root@ald01:~# timedatectl
               Local time: Tue 2023-06-13 13:14:55 MSK
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
           Universal time: Tue 2023-06-13 10:14:55 UTC
                 RTC time: Tue 2023-06-13 10:14:55
                Time zone: Europe/Moscow (MSK, +0300)
System clock synchronized: yes
              NTP service: inactive
          RTC in local TZ: no
```

На стороне Windows настраивается синхронизация времени с контролером домена в роли NTP сервера. Можно указать несколько значений(FQDN или IP адрес) через запятую, что не обеспечивает возможность автообнаружения контролеров

```
w32tm /config /reliable:yes /syncfromflags:manual /manualpeerlist:"<NTPServer>
 ↪" /update
```

```
net start w32time

w32tm /config /reliable:yes /syncfromflags:manual /manualpeerlist:"ald01.ald.
 ↪dom" /update
net stop w32time
net start w32time
w32tm /resync
w32tm /monitor /computers:"ald01.ald.dom"
w32tm /stripchart /computer:ald01.ald.dom
```

Результат проверки времени

```
PS C:\Users\Administrator> w32tm /stripchart /computer:ald01.ald.dom
Tracking ald01.ald.dom [192.168.88.210:123].
The current time is 4/13/2023 1:40:55 PM.
13:40:55, d:+00.0091883s o:-00.0110417s  [                         *      
 ↪                    ]
13:40:57, d:+00.0093593s o:-00.0111335s  [                         *      
 ↪                    ]
13:40:59, d:+00.0099441s o:-00.0112867s  [                         *      
 ↪                    ]
13:41:01, d:+00.0100102s o:-00.0113176s  [                         *      
 ↪                    ]
```

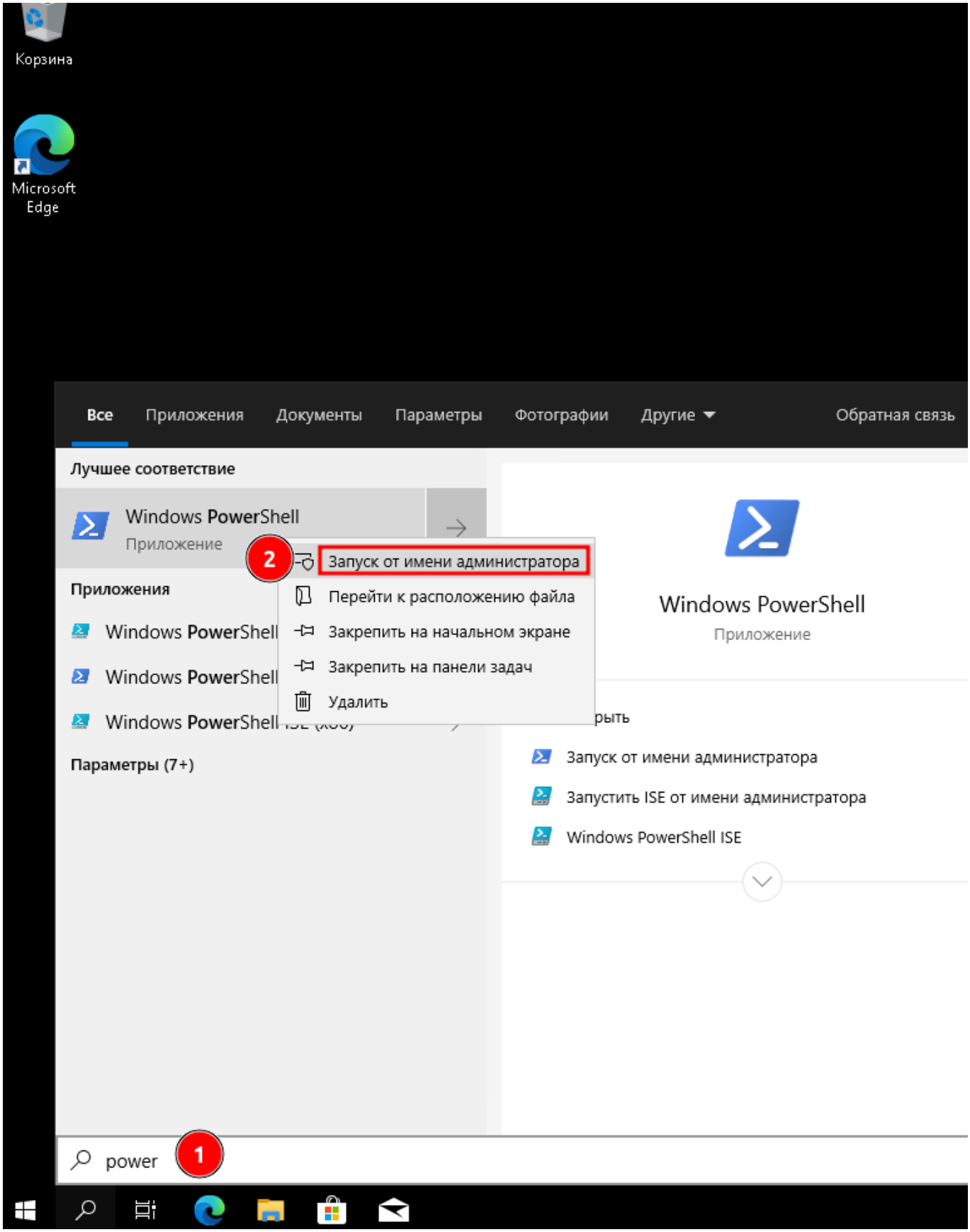


Рисунок 6.53 – Настройки даты/времени 1

В данном примере указан IP адрес

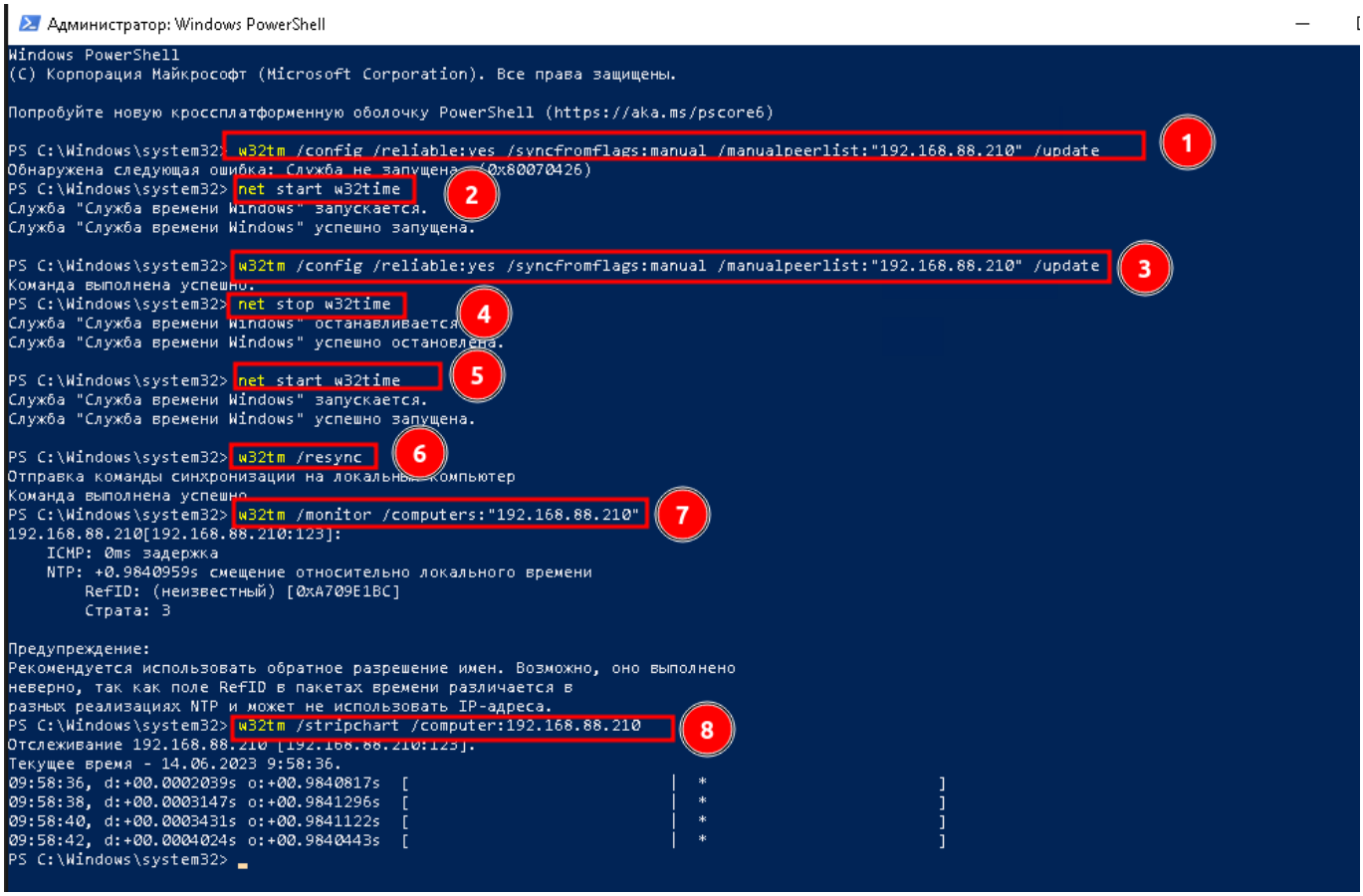


Рисунок 6.54 – Настройки даты/времени 2

Шаг 4. Создание нового узла в консоли FreeIPA

В веб-консоли перейти на вкладку хосты и нажать кнопку Добавить.

Указать имя хоста и IP-адрес клиента Windows, как показано на рисунке.

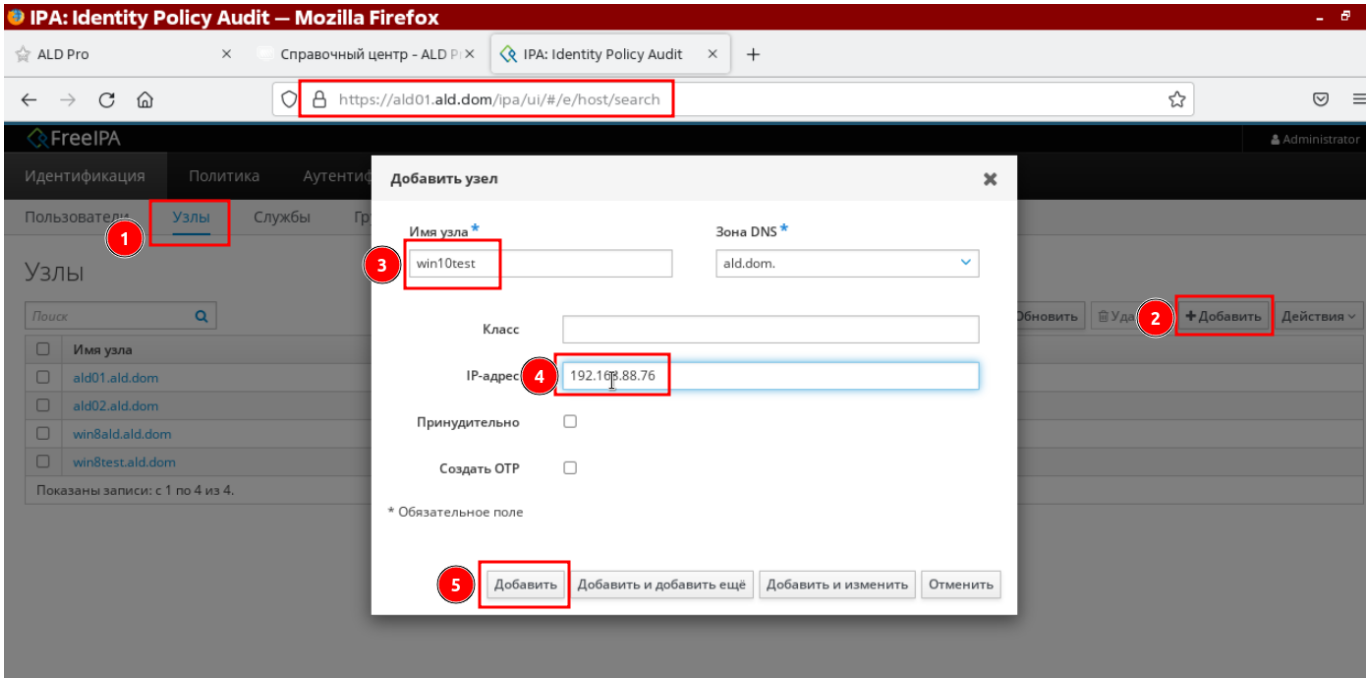


Рисунок 6.55 – Создание нового узла в консоли FreeIPA 1

Клиент был добавлен, но еще не зарегистрирован.

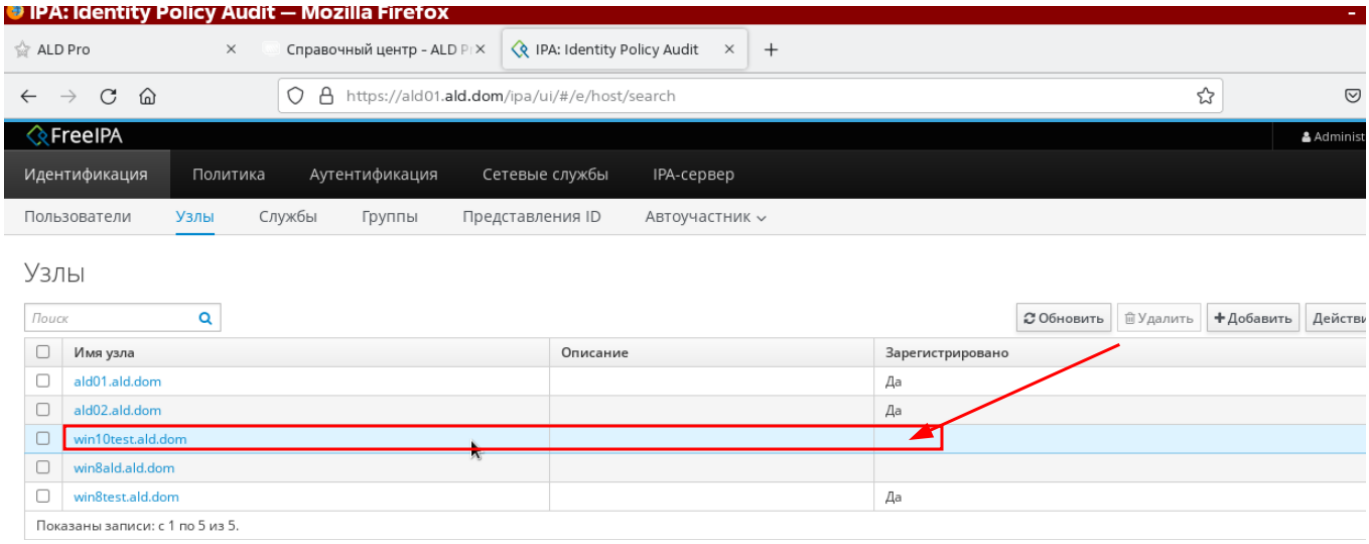


Рисунок 6.56 – Создание нового узла в консоли FreeIPA 2

Шаг 5. Создание keytab

Таблица ключей(keytab) - это файл, содержащий пары участников Kerberos и зашифрованные ключи (которые являются производными от пароля Kerberos).

Аутентификация(Authentication) - это процесс проверки личности зарегистрированного пользователя или процесса перед предоставлением доступа к защищенным сетям и

системам.

В командной строке FreeIPA добавить клиента.

Сначала генерируется билет(Выполните аутентификацию) с помощью команды:

```
root@ald01:~#kinit admin
root@ald01:~#kinit admin
Password for admin@ALD.DOM:
root@ald01:~#
root@ald01:~#
root@ald01:~# klist
Ticket cache: KEYRING:persistent:0:krb_ccache_sN0WuVv
Default principal: admin@ALD.DOM

Valid starting     Expires            Service principal
06/13/23 13:36:28  06/14/23 13:36:24  krbtgt/ALD.DOM@ALD.DOM
```

Для получения поддерживаемых типов шифрования выполнить команду:

```
root@ald01:~# ipa-getkeytab --permitted-enctypes
Failed to load translations
Supported encryption types:
AES-256 CTS mode with 96-bit SHA-1 HMAC
AES-128 CTS mode with 96-bit SHA-1 HMAC
AES-256 CTS mode with 192-bit SHA-384 HMAC
AES-128 CTS mode with 128-bit SHA-256 HMAC
Triple DES cbc mode with HMAC/sha1
ArcFour with HMAC/md5
Camellia-128 CTS mode with CMAC
Camellia-256 CTS mode with CMAC
```

Добавить участника, используя команду с приведенным ниже синтаксисом. Необходимо включить типы шифрования:

```
root@ald01:~#ipa-getkeytab -s ald01.ald.dom -p host/win10test.ald.dom@ALD.
↪DOM -e aes256-cts,aes128-cts,aes256-sha2,aes128-sha2,camellia256-cts-cmac,
↪camellia128-cts-cmac -k /etc/krb5.keytab.windows -P
```

В приведенной выше команде:

-s указывает сервер FreeIPA

-e определяет шифрование

-k путь до существующего или нового keytab файла

-p указывает нового участника, который будет добавлен

-P устанавливает пароль (**не забудьте, так как он будет использоваться при настройке клиента позже**)

```
root@ald01:~# kinit admin
Password for admin@ALD.DOM:
root@ald01:~#
root@ald01:~# ipa-getkeytab -s ald01.ald.dom -p host/win10test.ald.dom@ALD.
 ↪DOM -e aes256-cts,aes128-cts,aes256-sha2,aes128-sha2,camellia256-cts-cmac,
 ↪camellia128-cts-cmac -k /etc/krb5.keytab.windows -P
Failed to load translations
New Principal Password:
Verify Principal Password:
Keytab successfully retrieved and stored in: /etc/krb5.keytab.windows
```

Проверка, был ли добавлен ключ:

```
root@ald01:~# klist -k /etc/krb5.keytab.windows
Keytab name: FILE:/etc/krb5.keytab.windows
KVNO Principal
---- --------------------------------------------------------------------------
 ↪-
   1 host/win8test.ald.dom@ALD.DOM
   1 host/win8test.ald.dom@ALD.DOM
   1 host/win8test.ald.dom@ALD.DOM
   1 host/win8test.ald.dom@ALD.DOM
   1 host/win8test.ald.dom@ALD.DOM
   1 host/win8test.ald.dom@ALD.DOM
   1 host/win10test.ald.dom@ALD.DOM
   1 host/win10test.ald.dom@ALD.DOM
   1 host/win10test.ald.dom@ALD.DOM
   1 host/win10test.ald.dom@ALD.DOM
   1 host/win10test.ald.dom@ALD.DOM
   1 host/win10test.ald.dom@ALD.DOM
```

Вернуться к веб-интерфейсу FreeIPA, клиент должен быть зарегистрирован.

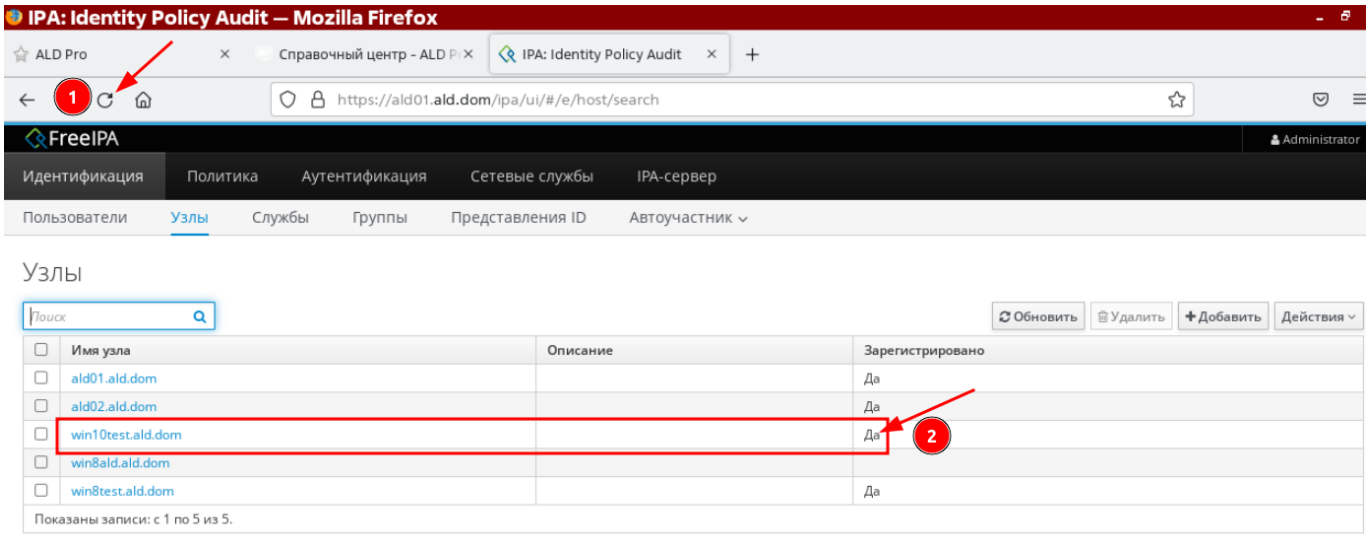


Рисунок 6.57 – Создание keytab 1

Шаг 6. Создание пользователя в FreeIPA

Чтобы иметь возможность использовать FreeIPA для аутентификации в Windows, необходимо создать пользователя в FreeIPA. На FreeIPA странице перейти на вкладку Пользователи и добавить пользователя, как показано на рисунке:

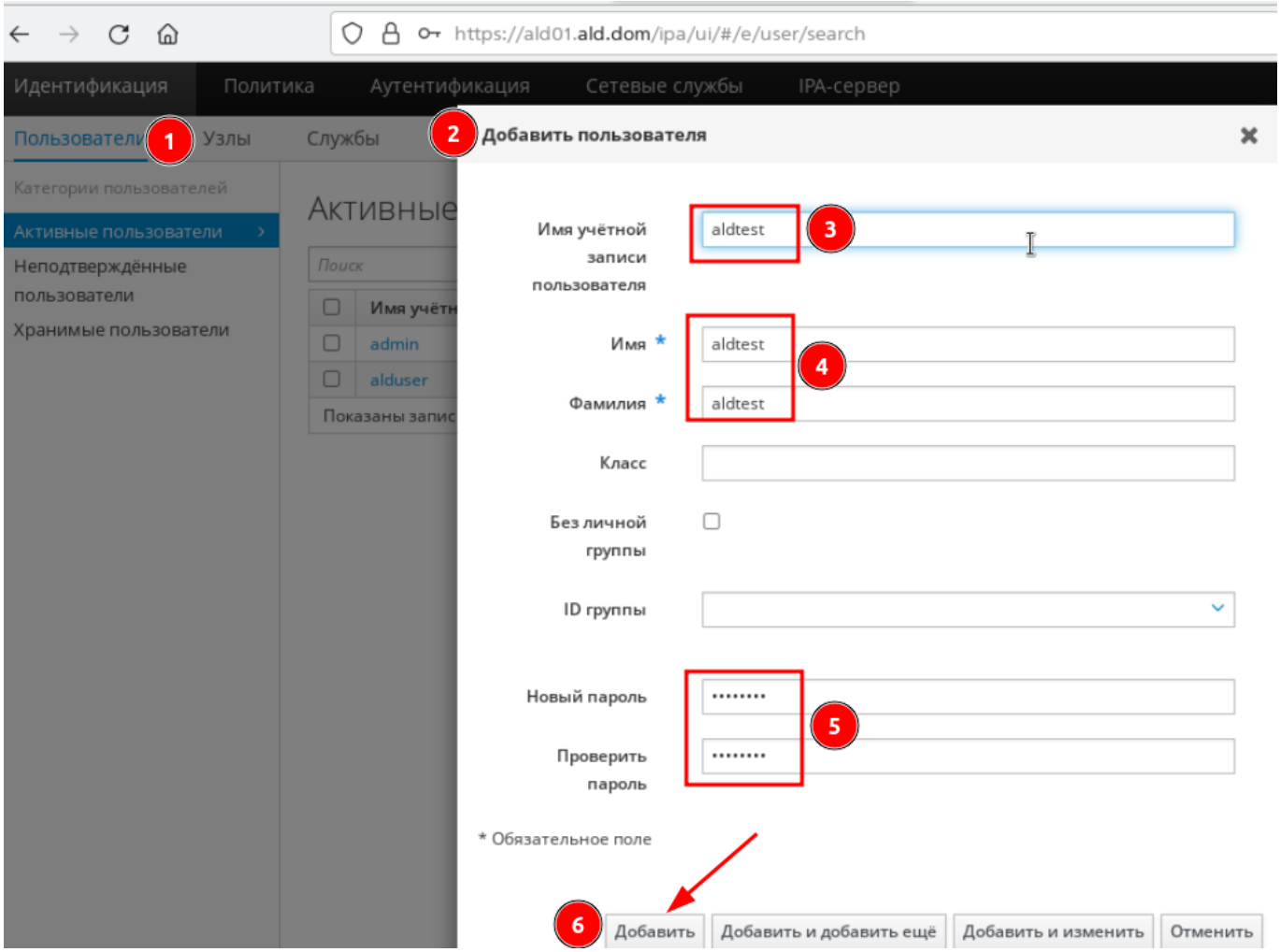


Рисунок 6.58 – Создание пользователя в FreeIPA

Указанный пароль при создании пользователя необходимо сменить при первом входе в систему.

Добавленный пользователь появится, как показано на рисунке.

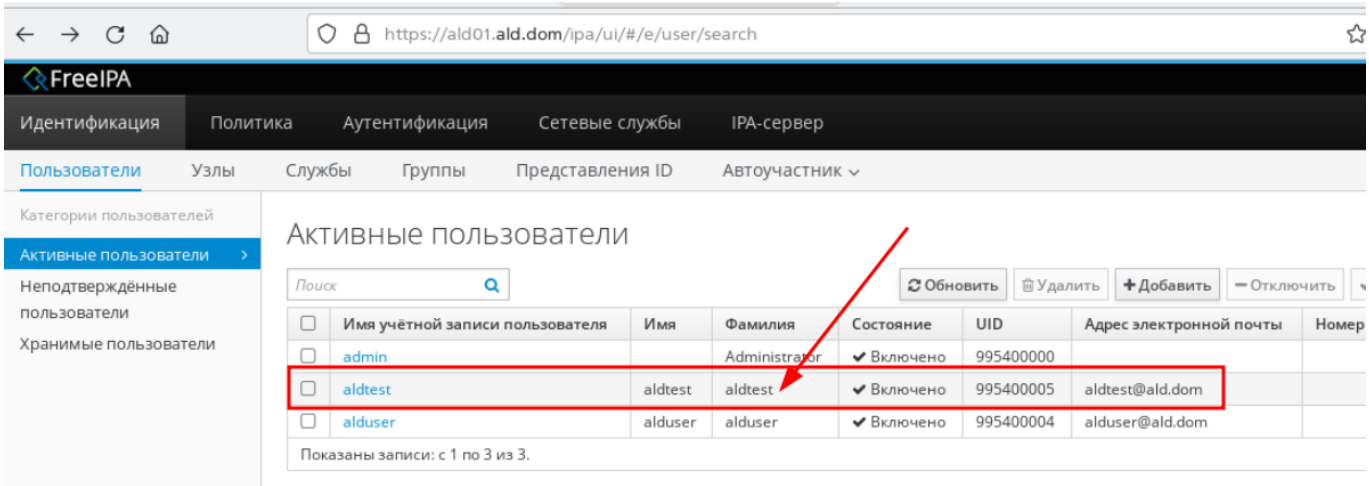


Рисунок 6.59 – Добавленный пользователь

Шаг 7. Настройка Windows системы на использование FreeIPA

```
ksetup /setdomain [REALM NAME]
ksetup /addkdc [REALM NAME] [kdc DNS name]
ksetup /addkpasswd [REALM NAME] [kdc DNS name]
ksetup /setcomputerpassword [MACHINE_PASSWORD] #(пароль из шага 5 при
 ↪генерировании keytab)
# ksetup /mapuser * *
```

`ksetup /setdomain` - Задает доменное имя для всех операций Kerberos.

https://learn.microsoft.com/ru-ru/windows-server/administration/windows-commands/ksetup-domain

`ksetup /addkdc` - Добавляет адрес центра распространения ключей (KDC) для заданной области Kerberos.

https://learn.microsoft.com/ru-ru/windows-server/administration/windows-commands/ksetup-addkdc

`ksetup addkpasswd` - Добавляет адрес сервера kerberos password (kpasswd) для области.

https://learn.microsoft.com/ru-ru/windows-server/administration/windows-commands/ksetup-addkpasswd

`ksetup /setcomputerpassword` - Задает пароль для локального компьютера. Эта команда влияет только на учетную запись компьютера и требует перезагрузки, чтобы изменение пароля вступило в силу.

Примечание: Пароль учетной записи компьютера не отображается в реестре или в качестве выходных данных команды ksetup.

---

https://learn.microsoft.com/ru-ru/windows-server/administration/windows-commands/ksetup-setcomputerpassword

В инструкциях встречается указание, что необходимо выполнить команду ksetup /mapuser, чтобы сопоставить все учетные записи в области ALD.DOM Kerberos с любой существующей учетной записью с тем же именем на этом компьютере.

Благодаря использованию доменных учётных записей пользователей, нет необходимости

сопоставлять им локальные “учётки”.

https://learn.microsoft.com/ru-ru/windows-server/administration/windows-commands/ksetup-mapuser

Выполнить поочередно следующие команды (Не перезагружать компьютер):

```
ksetup /setdomain ALD.DOM
ksetup /addkdc ALD.DOM ald01.ald.dom
ksetup /addkpasswd ALD.DOM ald01.ald.dom
ksetup /setcomputerpassword !QAZ1qaz (укажите свой пароль)
```

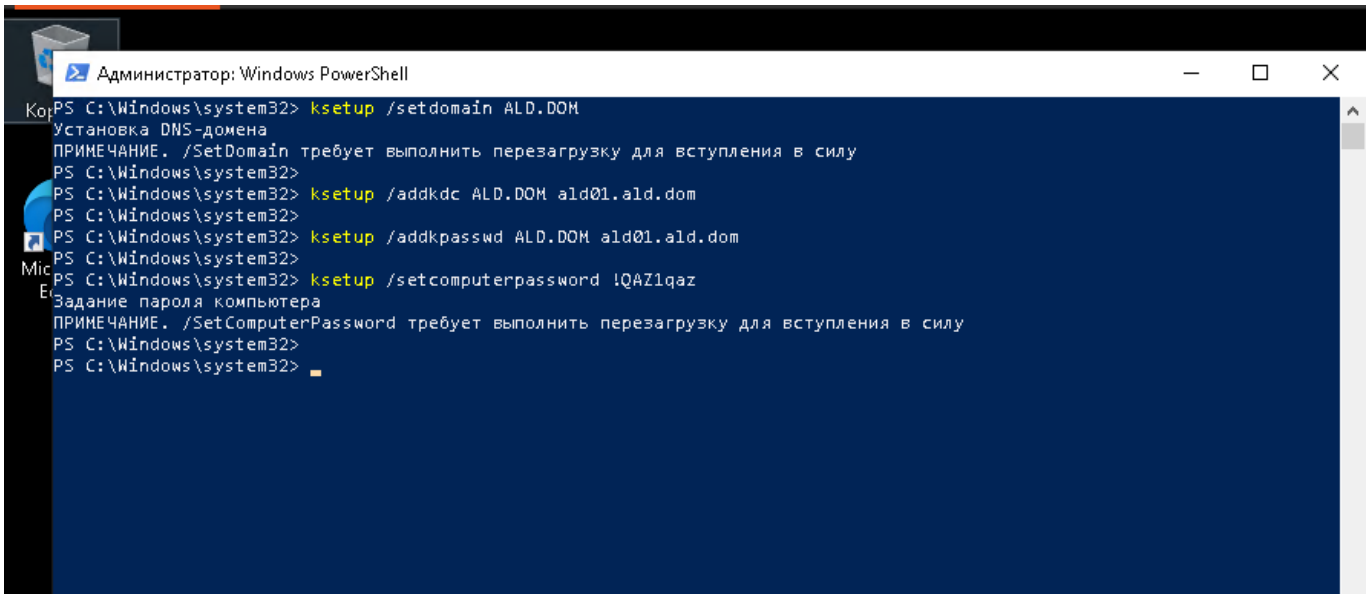


Рисунок 6.60 – Результат выполнения команд

Теперь запустить **gpedit.msc**, нажав клавишу Windows + R.

Конфигурация Windows > Параметры безопасности, Локальные политики, Параметры Безопасности > Сетевая безопасность: настройка типов шифрования, разрешенных Kerberos

Windows Settings > Security Settings > Local Policies > Security Options > Network Security: Configure encryption types allowed for Kerberos

Указать следующие типы шифрования:

RC4_HMAC_MD5,AES128_HMAC_SHA1,AES256_HMAC_SHA1,Будущие типы шифрования

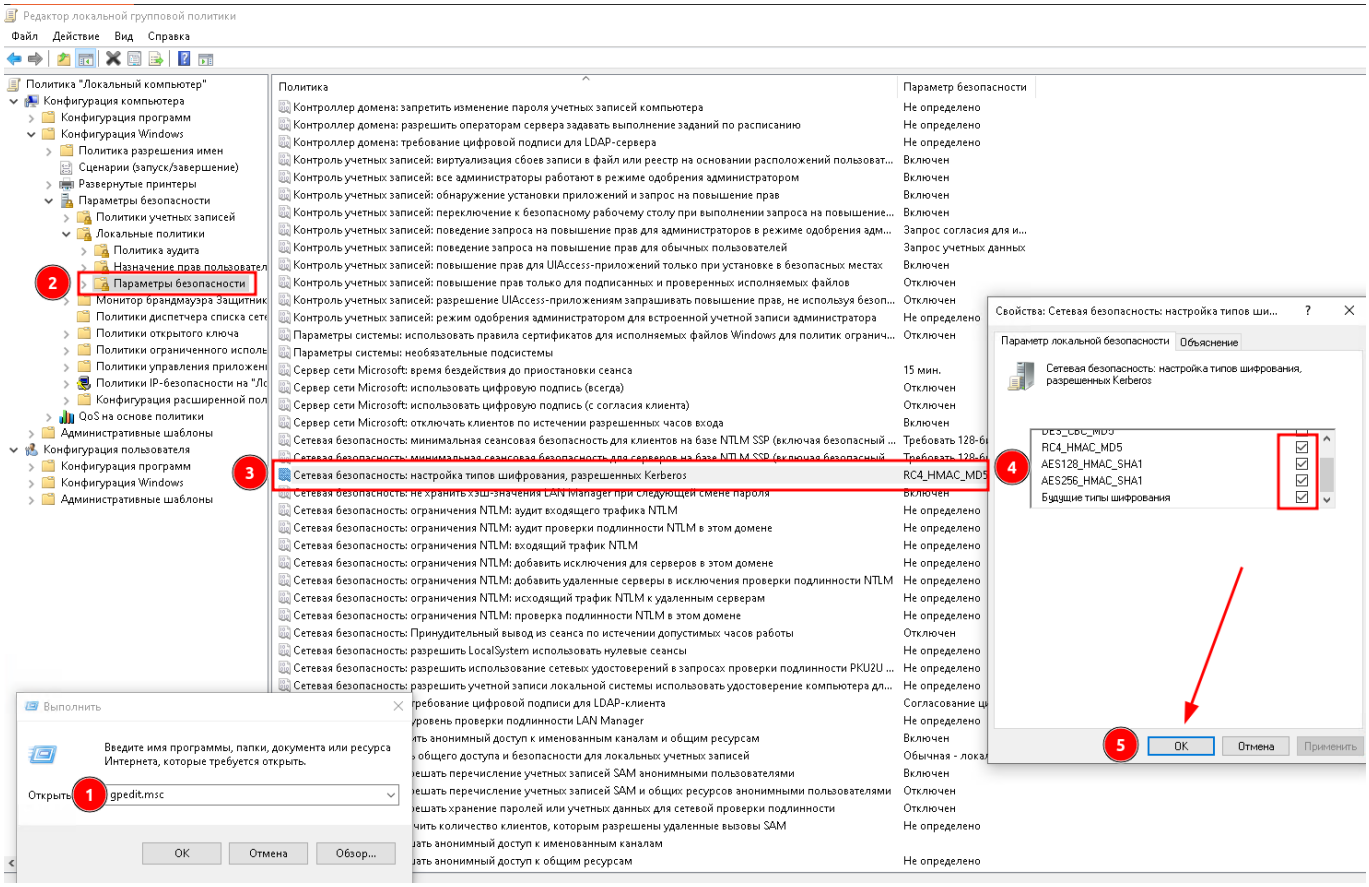


Рисунок 6.61 – Типы шифрования

Применить, нажав кнопку OK, и перезагрузить систему.

Шаг 8. Вход в Windows с помощью пользователей FreeIPA

Имя Пользователя@REALM aldtest@ALD.DOM

Когда система перезагрузится, войти в систему с помощью пользователя FreeIPA, как показано на рисунке:

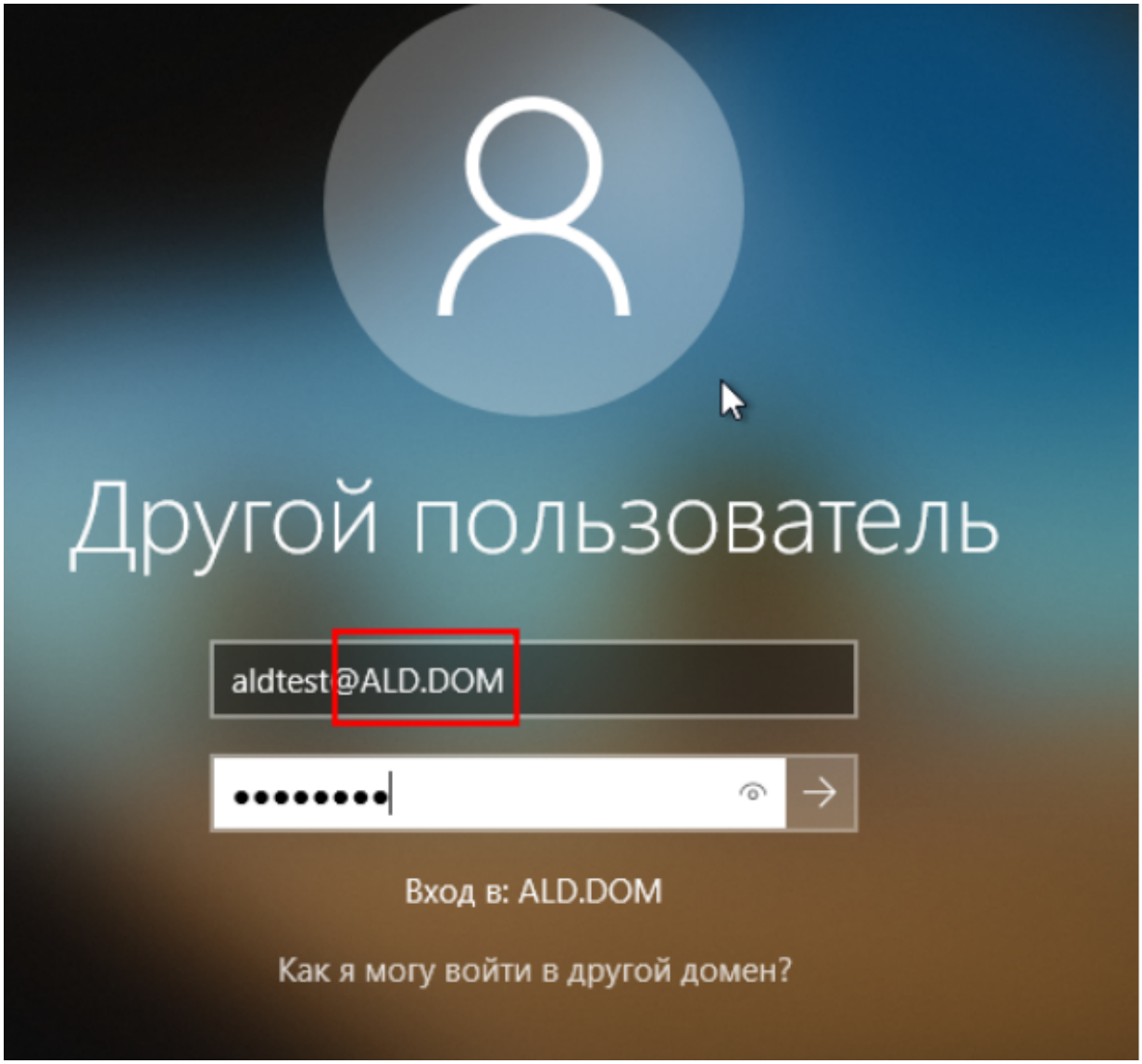


Рисунок 6.62 – Вход в систему

Будет предложено изменить пароль для пользователя IPA.

Шаг 9. Проверки

Выполнить следующие команды в PowerShell или CMD:

```
whoami
klist
```

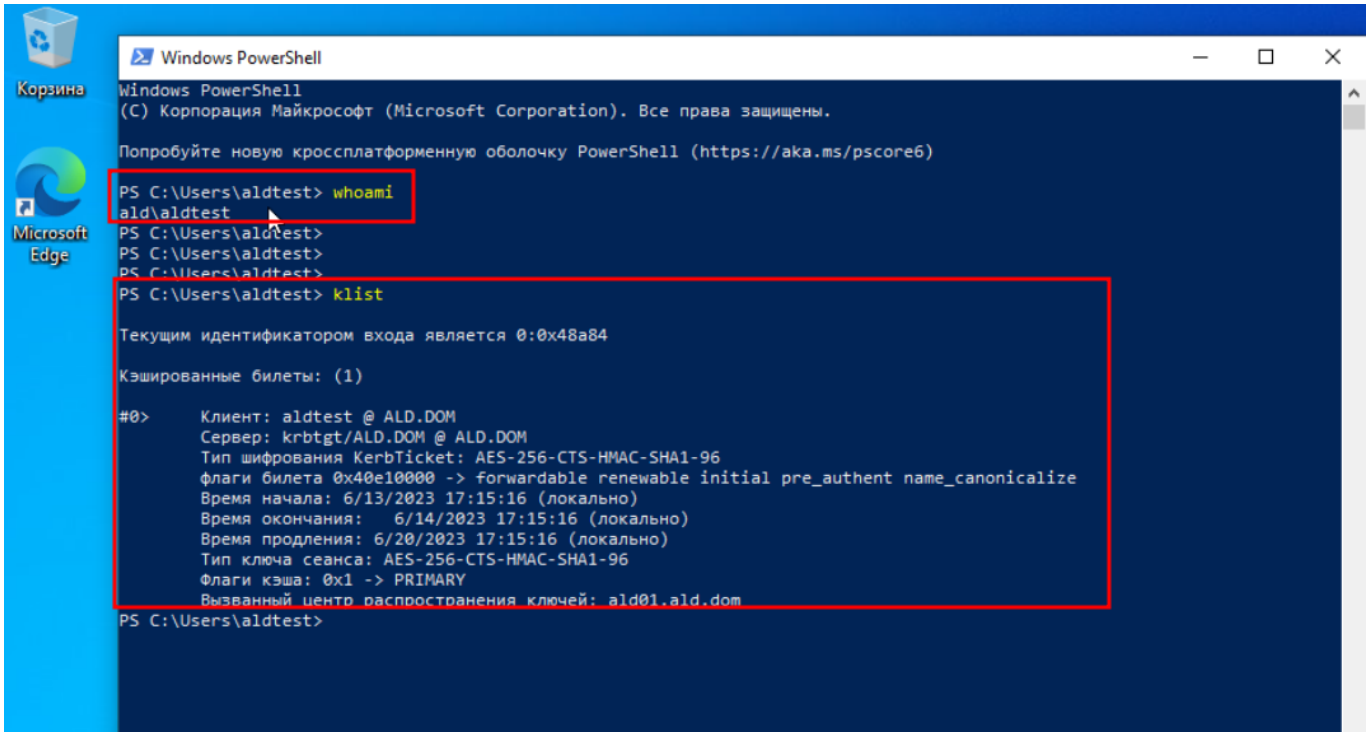


Рисунок 6.63 – Проверки 1

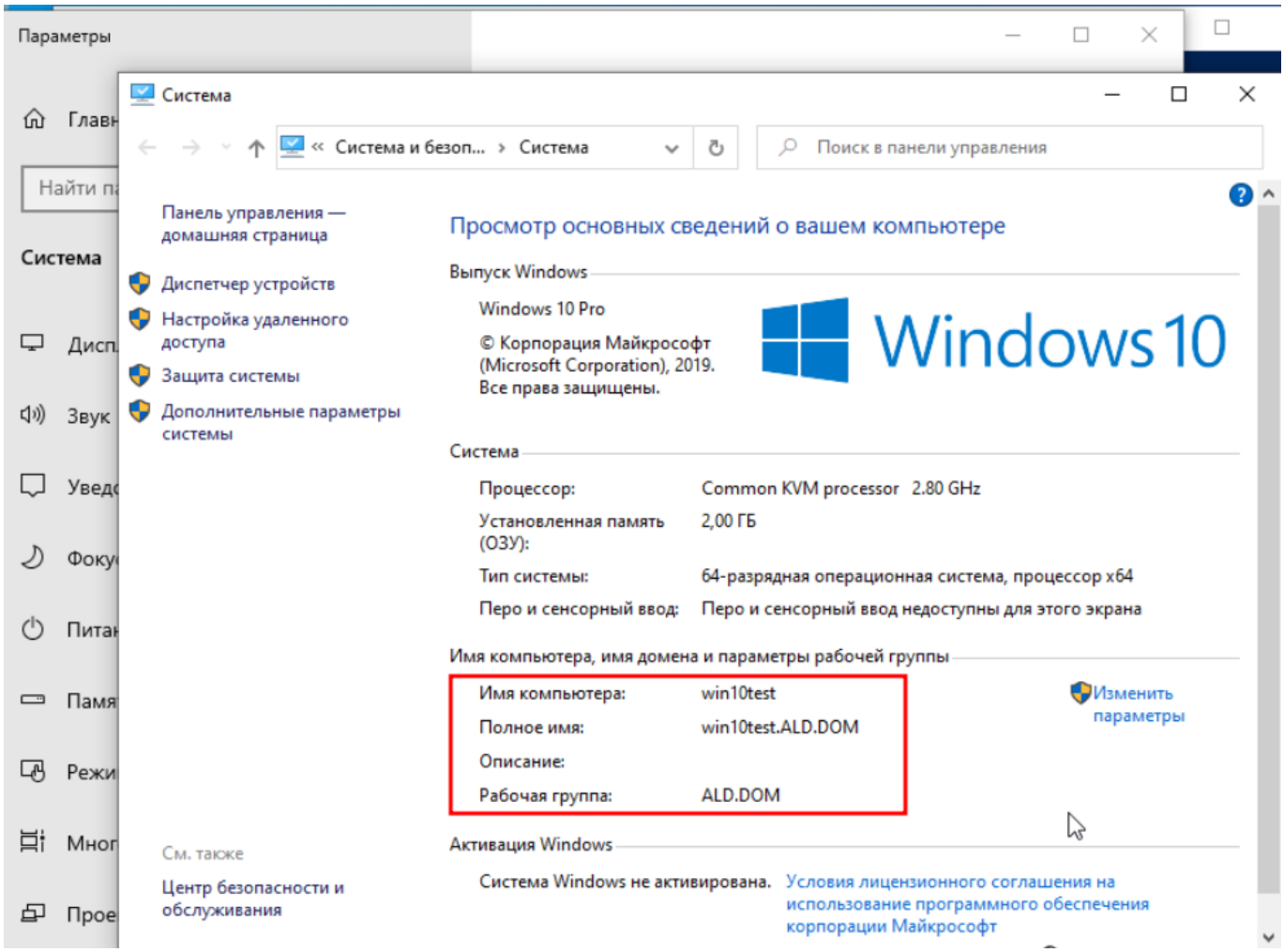


Рисунок 6.64 – Проверки 2

На Ald Pro сервере создать общую тестовую папку и попробовать зайти:

```
root@ald01:~# nano /etc/samba/smb.conf
[testshare]
path = /srv/testshare
browseable = yes
valid users = aldtest
admin users = aldtest
writable = yes
```

```
root@ald01:~# systemctl reload smbd.service
```

После успешного входа klist будет содержать дополнительный билет:

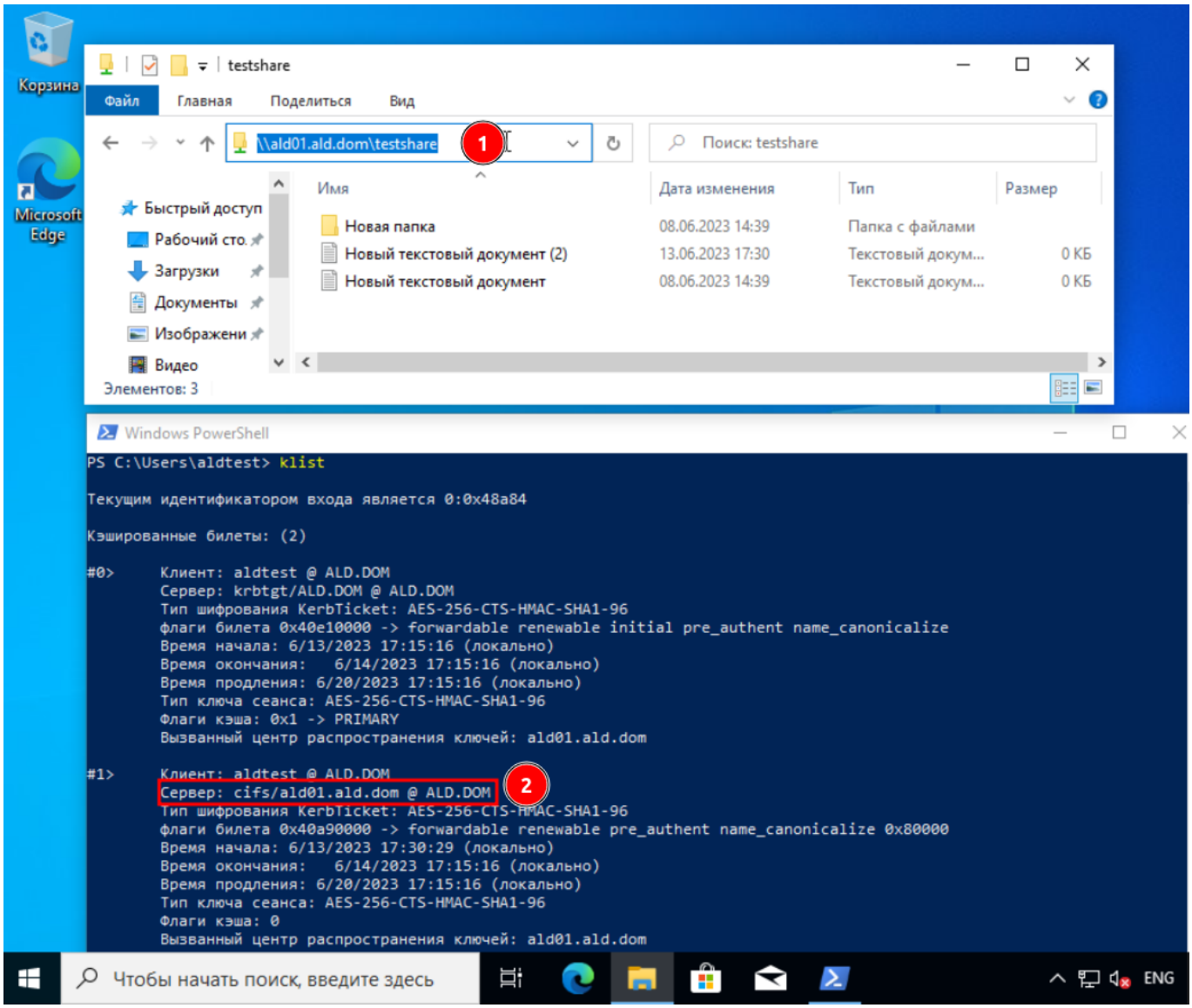


Рисунок 6.65 – Билет Kerberos

6.7.2.5. Добавление доменных(ALD Pro) пользователей или групп в локальные группы Windows

Предварительные требования

Скачать пакет WMF 5.1 для той операционной системы и архитектуры, в которой будет производиться установка.

Операционная система	Предварительные требования	Ссылка на пакеты
Windows Server 2012 R2		Win8.1AndW2K12R2-KB3191564-x64.msu
Windows Server 2012		W2K12-KB3191565-x64.msu
Windows Server 2008 R2	.NET Framework 4.5.2	Win7AndW2K8R2-KB3191566-x64.ZIP
Windows 8.1		x64:Win8.1AndW2K12R2-KB3191564-x64.msu, x86:Win8.1-KB3191564-x86.msu
Windows 7 с пакетом обновления 1 (SP1)	.NET Framework 4.5.2	x64:Win7AndW2K8R2-KB3191566-x64.ZIP, x86:Win7-KB3191566-x86.ZIP

https://learn.microsoft.com/ru-ru/powershell/scripting/windows-powershell/wmf/setup/install-configure?view=powershell-7.3#download-and-install-the-wmf-51-package

Получение SID пользователя

Используя wbinfo

```
root@ald01:~# wbinfo -n "ald\admin"
S-1-5-21-109148531-2531787706-4107538291-500 SID_USER (1)
```

Используя ipa command

```
root@ald01:~# ipa user-show admin --all | grep ipantsecurityidentifier
  ipantsecurityidentifier: S-1-5-21-109148531-2531787706-4107538291-500
```

Получение SID группы

Используя ipa command

```
root@ald01:~# ipa group-show group_high --all | grep ipantsecurityidentifier
  ipantsecurityidentifier: S-1-5-21-109148531-2531787706-4107538291-1007
root@ald01:~#
root@ald01:~# ipa group-show group_high --all
  dn: cn=group_high,cn=groups,cn=accounts,dc=ald,dc=dom
  Имя группы: group_high
  ID группы: 995400007
  Группы-участники: group_low
  Пользователи с непрямым участием: alduser, aldtest
  ipantsecurityidentifier: S-1-5-21-109148531-2531787706-4107538291-1007
  ipauniqueid: d1410fd4-0a8c-11ee-9d12-422c2492509f
  objectclass: top, groupofnames, nestedgroup, ipausergroup, ipaobject, x-ald-
↪audit-policy, rbta-unit, posixgroup, ipantgroupattrs
```

Добавление пользователя или группу по SID в локальную группу (запуск PowerShell из под администратора)

```
Windows PowerShell (C) Корпорация Майкрософт (Microsoft Corporation). Все
↪права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/
↪pscore6)

PS C:\Windows\system32> Add-LocalGroupMember -Group 'Пользователи удаленного
↪рабочего стола' -Member 'S-1-5-21-109148531-2531787706-4107538291-500'
```

Добавление разрешений на папки доменным пользователям ALD Pro

Наиболее популярные разрешения:

r = чтение

rx = Чтение, Выполнение, Список содержимого папки

rxm = Чтение, Выполнение, Список содержимого папки, Запись, Изменение

f = Полный доступ

(OI) = Для этой папки и её файлов

(CI) = Для этой папки и её подпапок

Таким образом, чтобы дать обычные права на чтение и запись на папку, используются разрешения (OI)(CI)rxm. То есть результирующая команда будет выглядеть так:

PowerShell или CMD

```
ICACLS "C:\Temp" /grant "*S-1-5-21-109148531-2531787706-4107538291-
↪500:(OI)(CI)rxm"
```

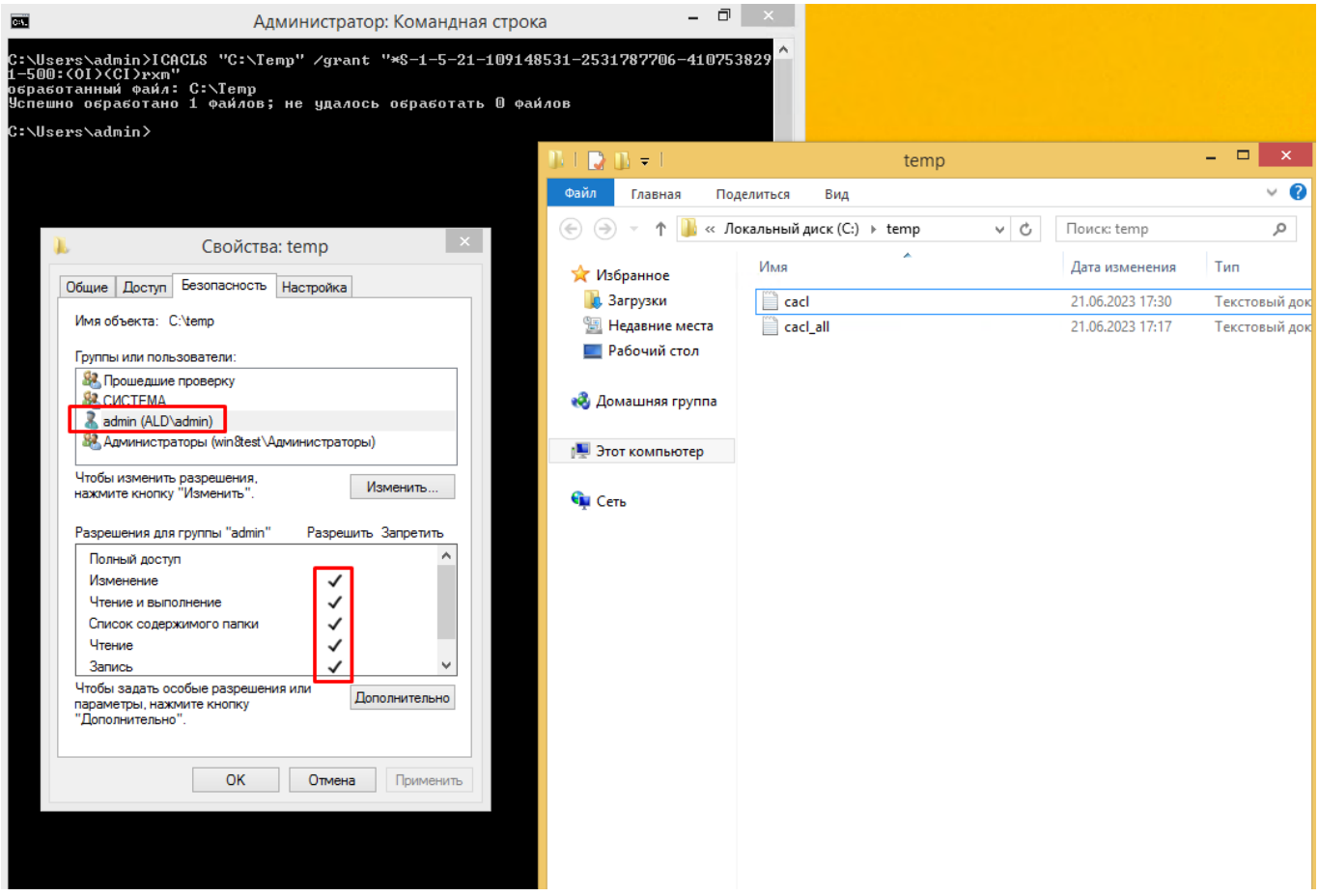


Рисунок 6.66 – Результат работы команд

Более детальная информация может быть найдена на сайте Microsoft:

https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/icacls

Добавление разрешений на папки доменным группам ALD Pro

К сожалению в текущей реализации не поддерживается прямое добавление доменных групп в дискретные списки доступа (DACL).

Для того, чтобы обойти это ограничение:

1. Создать соответствующую доменной локальную группу.
2. В локальную группу добавить SID доменной группы (см. Добавление доменных(ALD Pro) пользователей или групп в локальные группы Windows).
3. Добавить в безопасность папки или файла локальную группу, которую создали в п. 1.

```
Alddom_fs_c_temp_r

Alddom_fs_c_temp_rw
```

6.8. Модуль Синхронизации

6.8.1. Инструкция по обновлению ALD Pro до версии 2.4.0 с ранее установленным модулем синхронизации

6.8.1.1. Предварительный этап

Делаем резервную копию БД модуля в отдельную директорию (выбрать в соответствии с инфраструктурой):

```
mkdir /srv/pg_backup && chown postgres:postgres /srv/pg_backup/
sudo -i -u postgres
$ pg_dump -d syncer -F d -f /srv/pg_backup/syncer_$(date +%Y%m%d-%H%M)
$ du -sh /srv/pg_backup/syncer_20240913-0938/
520K /srv/pg_backup/syncer_20240913-0938/
```

Проверяем, что модуль синхронизации установлен и запрещаем обновлять его пакеты:

```
apt list --installed | grep -E 'aldpro(.*)sync'
aldpro-mp-ui-syncer/1.7_x86-64,now 2.3.0-22 amd64 [установлен, автоматически]
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
aldpro-syncer/1.7_x86-64,now 2.3.0-31 amd64 [установлен]
apt-mark hold aldpro-syncer aldpro-mp-ui-syncer
```

6.8.1.2. Обновление ALD Pro до новой версии

Обновляем source.list и пакеты на КД с установленным модулем обновления.

```
astra-update -A -r -T
aldpro-server-install --update
```

6.8.1.3. Обновление модуля синхронизации

После основного этапа обновления ALD Pro и проверки работоспособности можно переходить к обновлению модуля синхронизации.

Снимаем удержание с пакетов:

```
apt-mark unhold aldpro-syncer aldpro-mp-ui-syncer
```

Обновляем пакеты и завершаем обновление ALD Pro:

```
astra-update -A -r -T
aldpro-server-install --update
```

После завершения обновления, необходимо перезагрузить контроллер домена.

```
sudo reboot
```

Обновление завершено.

6.8.2. Инструкция по дополнительной настройке модуля синхронизации

В виду предотвращения конфликтных ситуаций при миграции и синхронизации данных из MS AD используется синхронная модель поведения синхронизации данных. В качестве

ведущего сервера обработки информации выбран главный сервер в кластере серверов ALD Pro.

6.8.2.1. Исходные настройки

Предполагается, что на момент настройки модуля синхронизации у нас имеются:

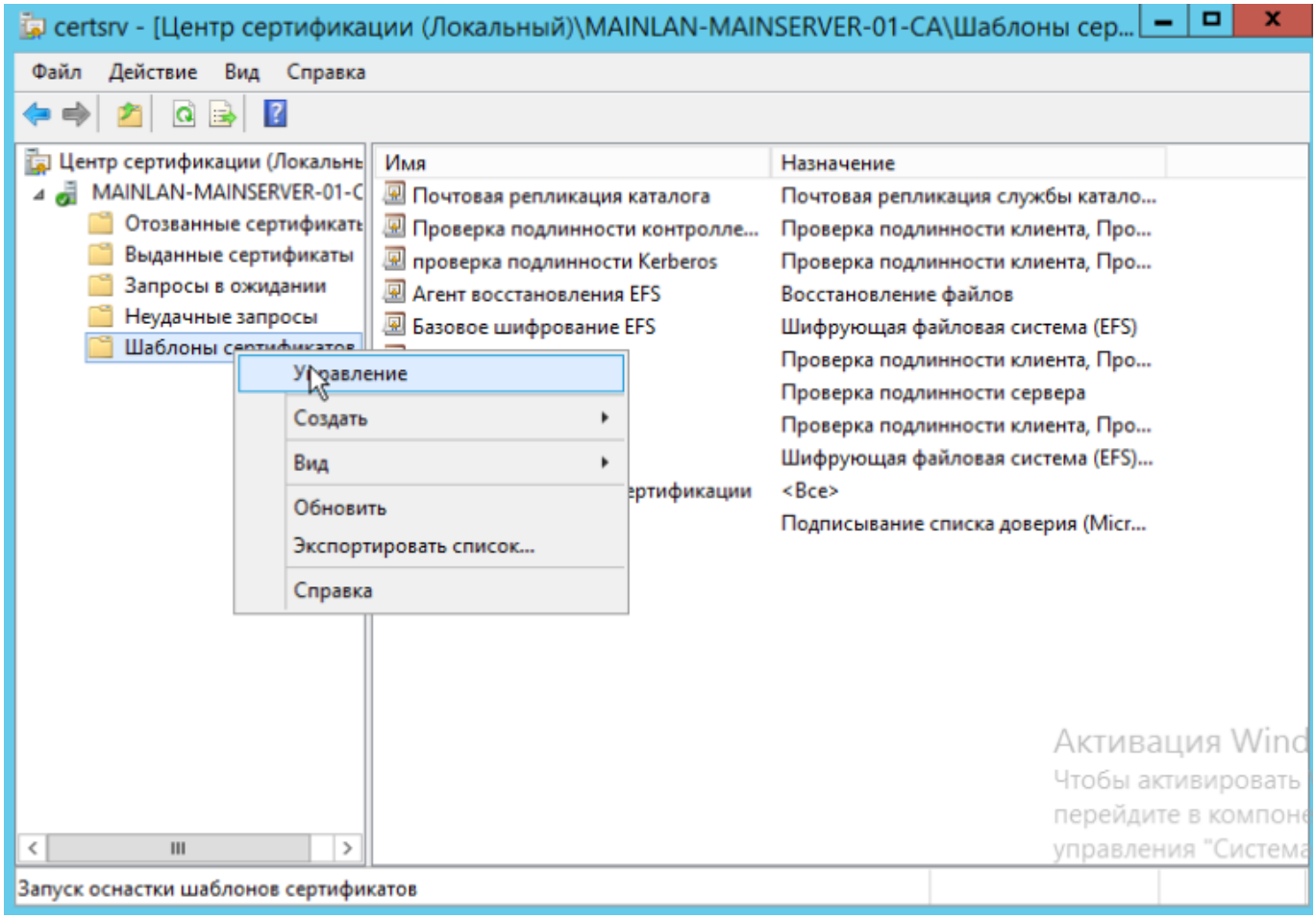
- Настроенный контроллер домена MS AD;
- Настроенный контроллер домена ALD Pro;
- Для сетей, в которых находятся контроллеры домена Microsoft AD и ALD Pro, настроено перенаправление DNS-зон;
- В домене MS AD поднят центр сертификации и выдан сертификат для доступа по LDAPs (см. *Выгрузка сертификатов для контроллера домена MS AD*);
- Для учетной записи AD, под которой идет подключение модуля синхронизации к контроллерам домена AD, необходимо выдать права на контейнер “Deleted Object” через powershell:

```
// для выдачи прав на контейнер запускаем powershell от имени администратора
dsacls "CN=Deleted Objects,DC=winad,DC=lan" /takeownership
dsacls "CN=Deleted Objects,DC=winad,DC=lan" /g winad\aldagent:LCRP
```

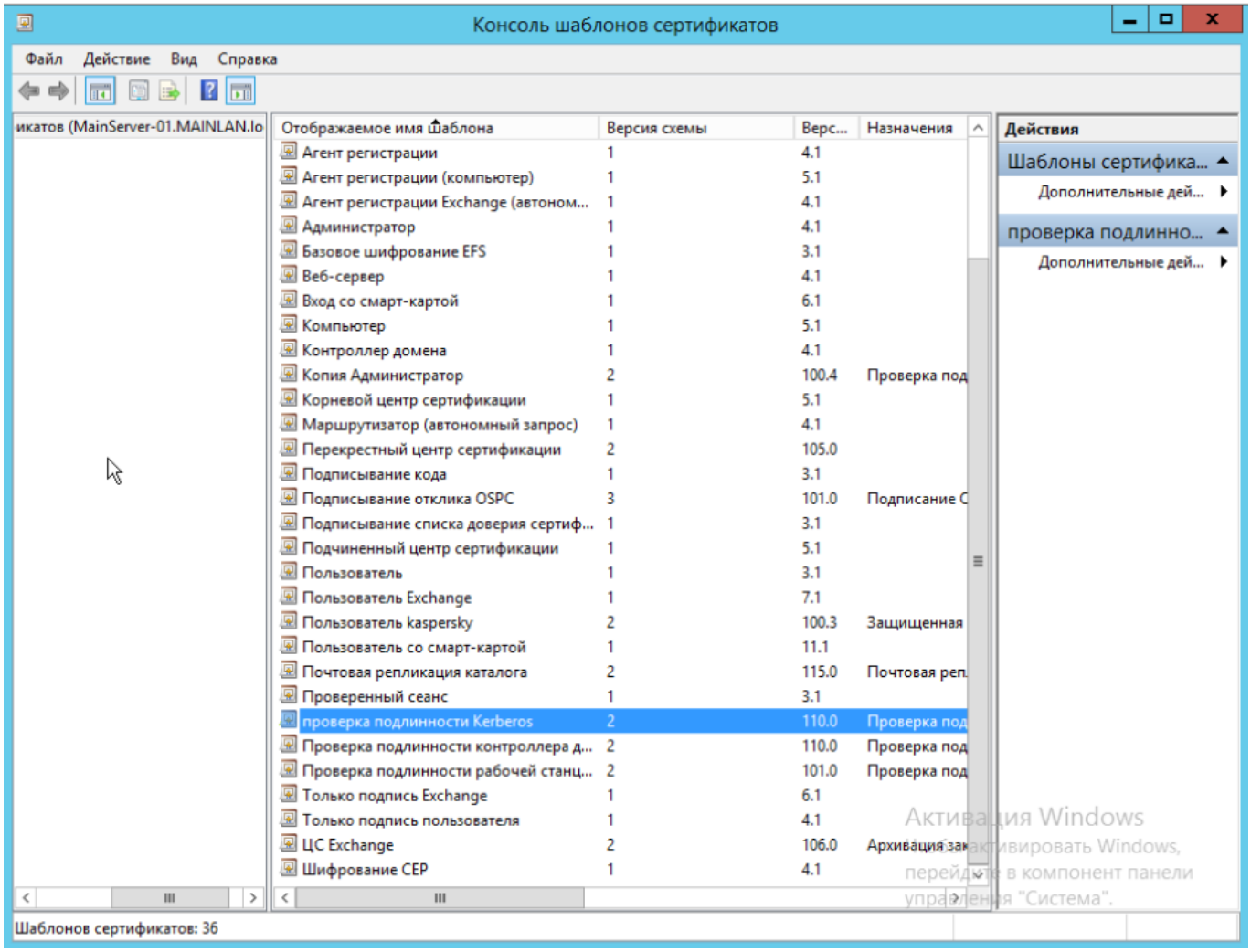
Выгрузка сертификатов для контроллера домена MS AD

Для выгрузки сертификатов необходимо, чтобы в домене был настроен **Центр Сертификации**.

Запустить **Центр Сертификации -> Шаблоны Сертификатов -> Управление** на сервере с ролью **Центр Сертификации**:

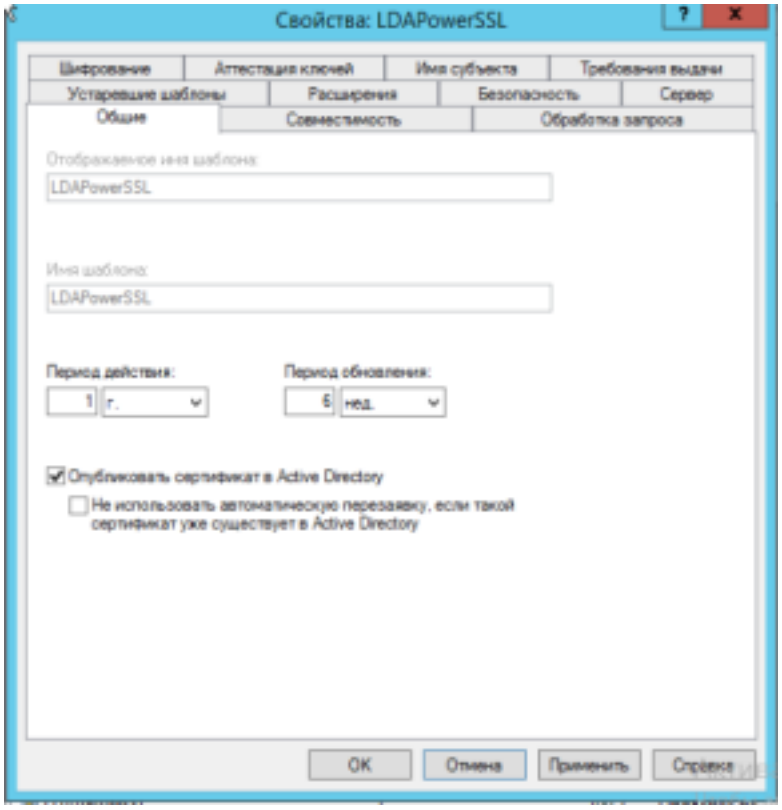


Создать копию шаблона **Проверка подлинности Kerberos**, выбрав пункт **Скопировать шаблон** контекстного меню:



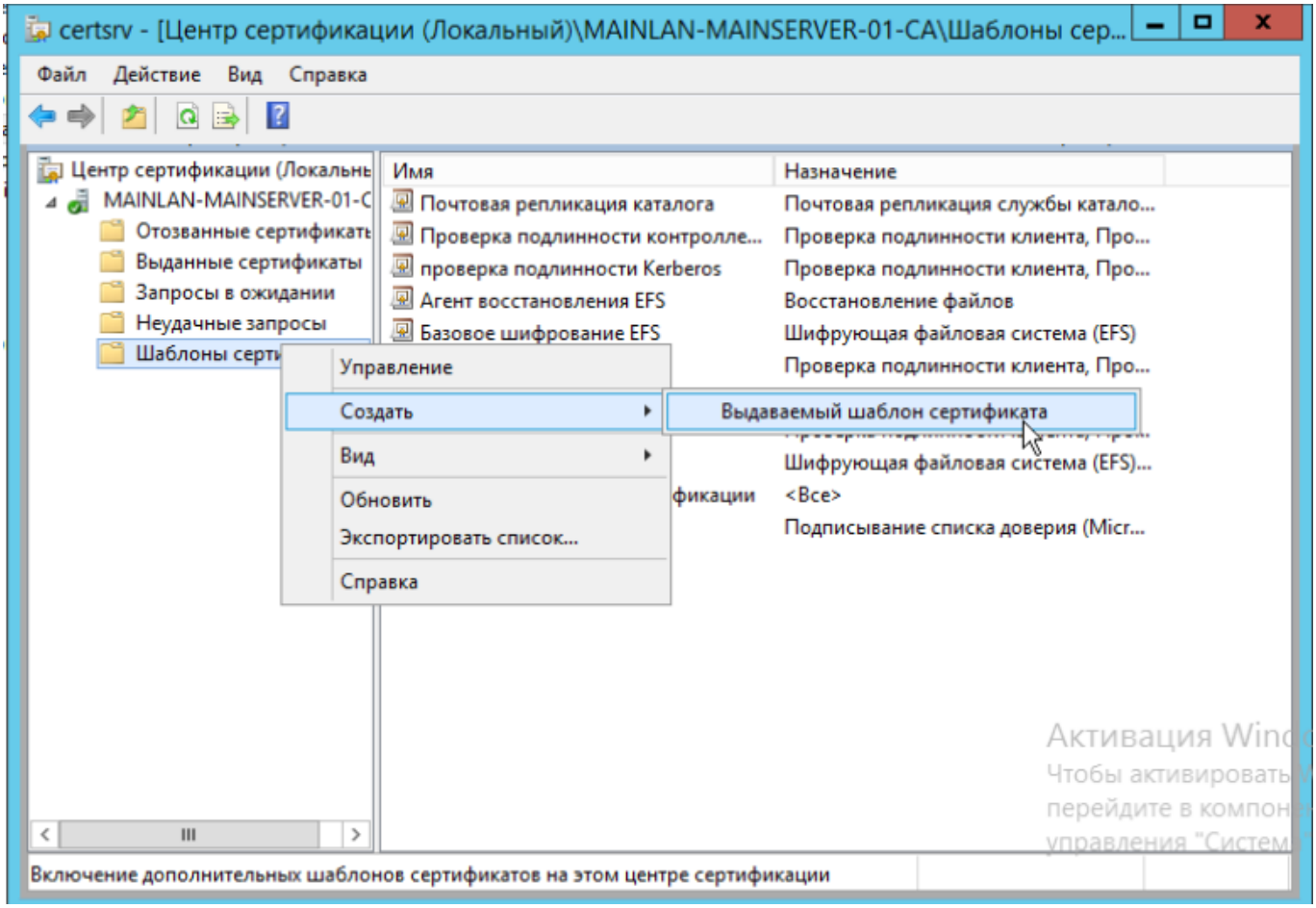
Настроить сертификат на вкладке **Общие**:

- имя сертификата **LDAPoverSSL**;
- период действия сертификата.

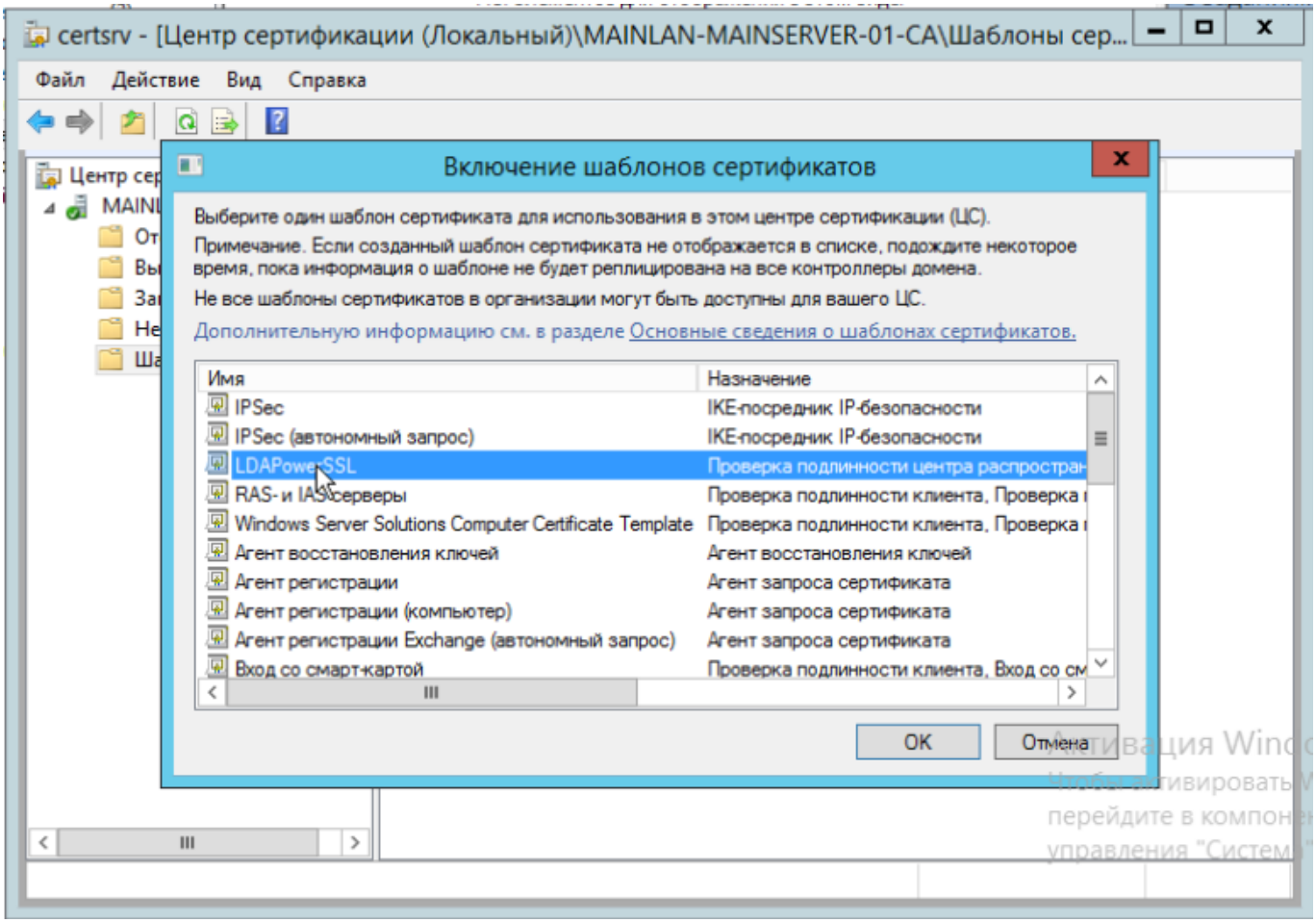


Опубликовать сертификат в MS AD.

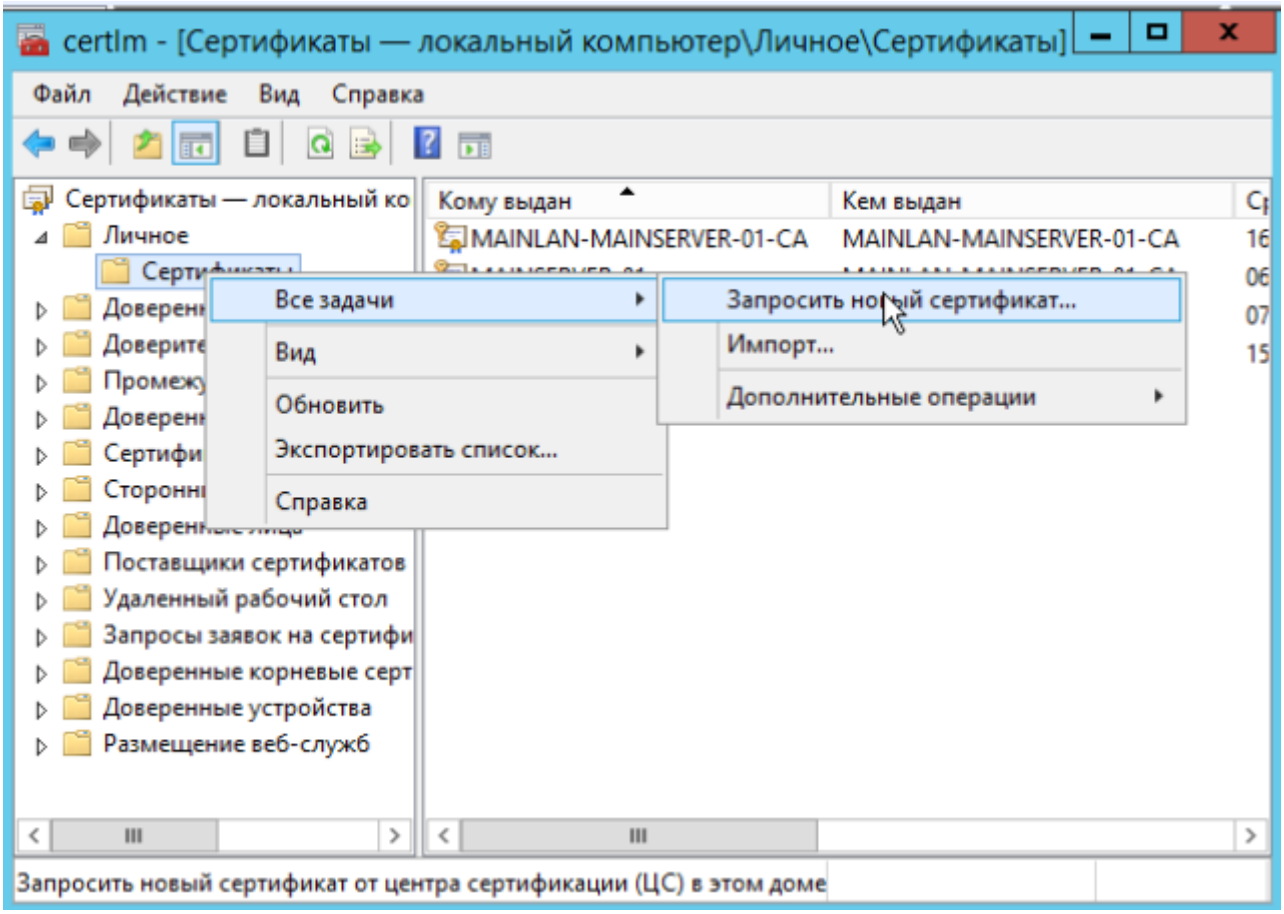
Опубликовать новый тип сертификата. Для этого в в контекстном меню раздела **Шаблоны сертификатов** выбрать **Создать -> Выдаваемый шаблон сертификата**.



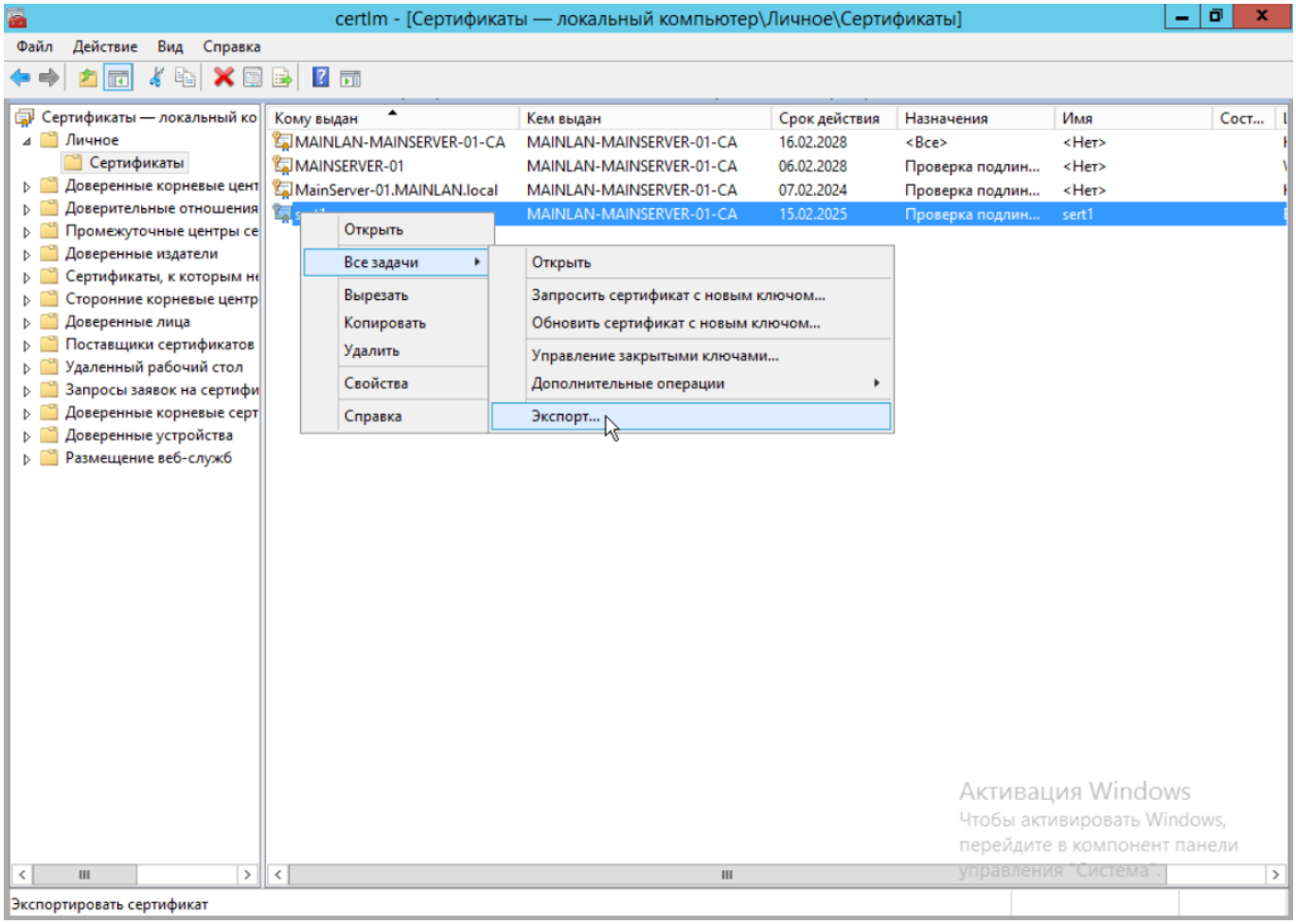
В списке доступных шаблонов выбрать созданный ранее:



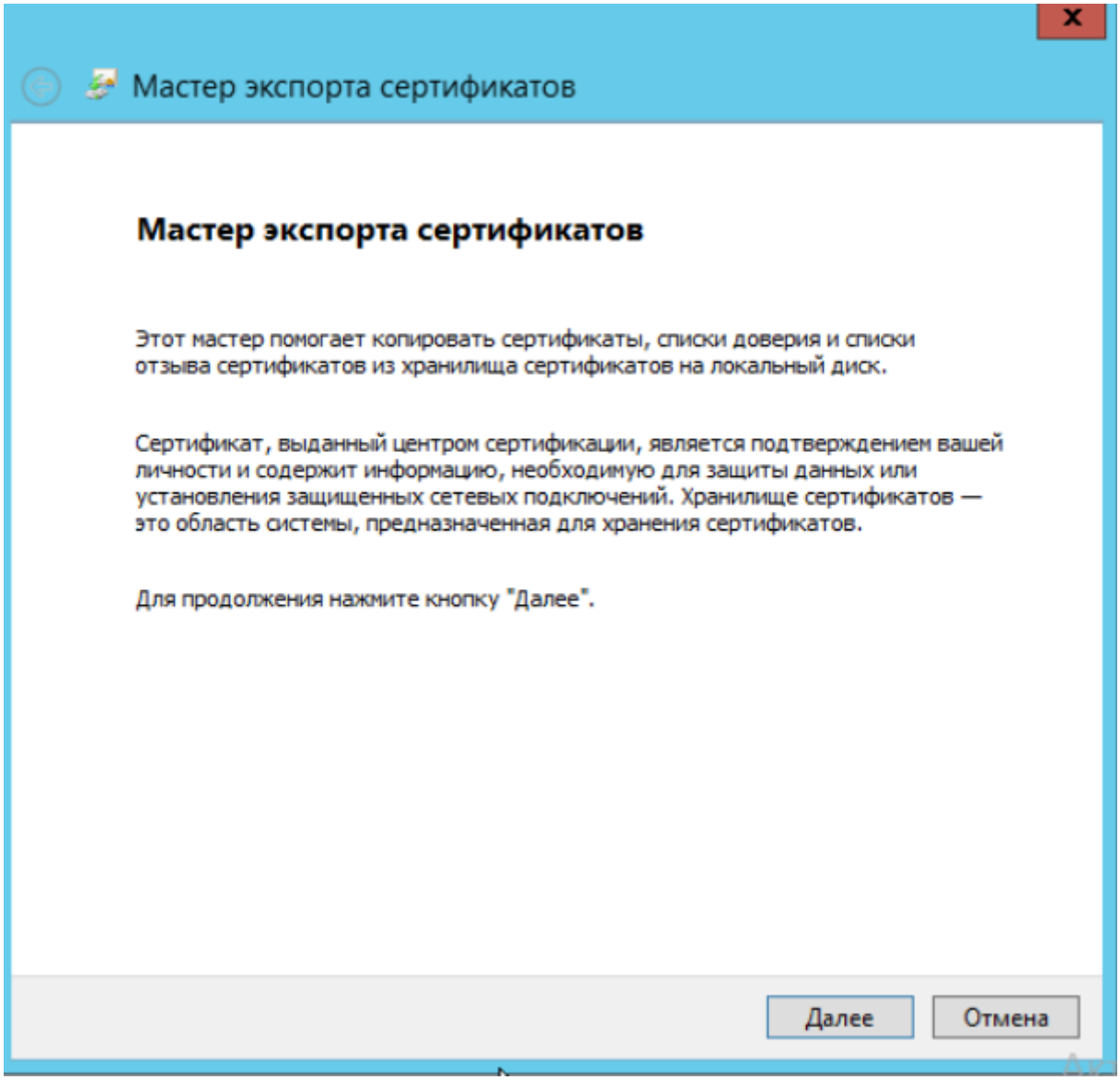
На контроллере домена, который будет задействован в процессе синхронизации данных со стороны MS AD, открыть оснастку **Управление сертификатами** компьютера. В дереве сертификатов перейти в папку **Личное** и в контекстном меню выбрать **Все задачи -> Запросить новый сертификат**:

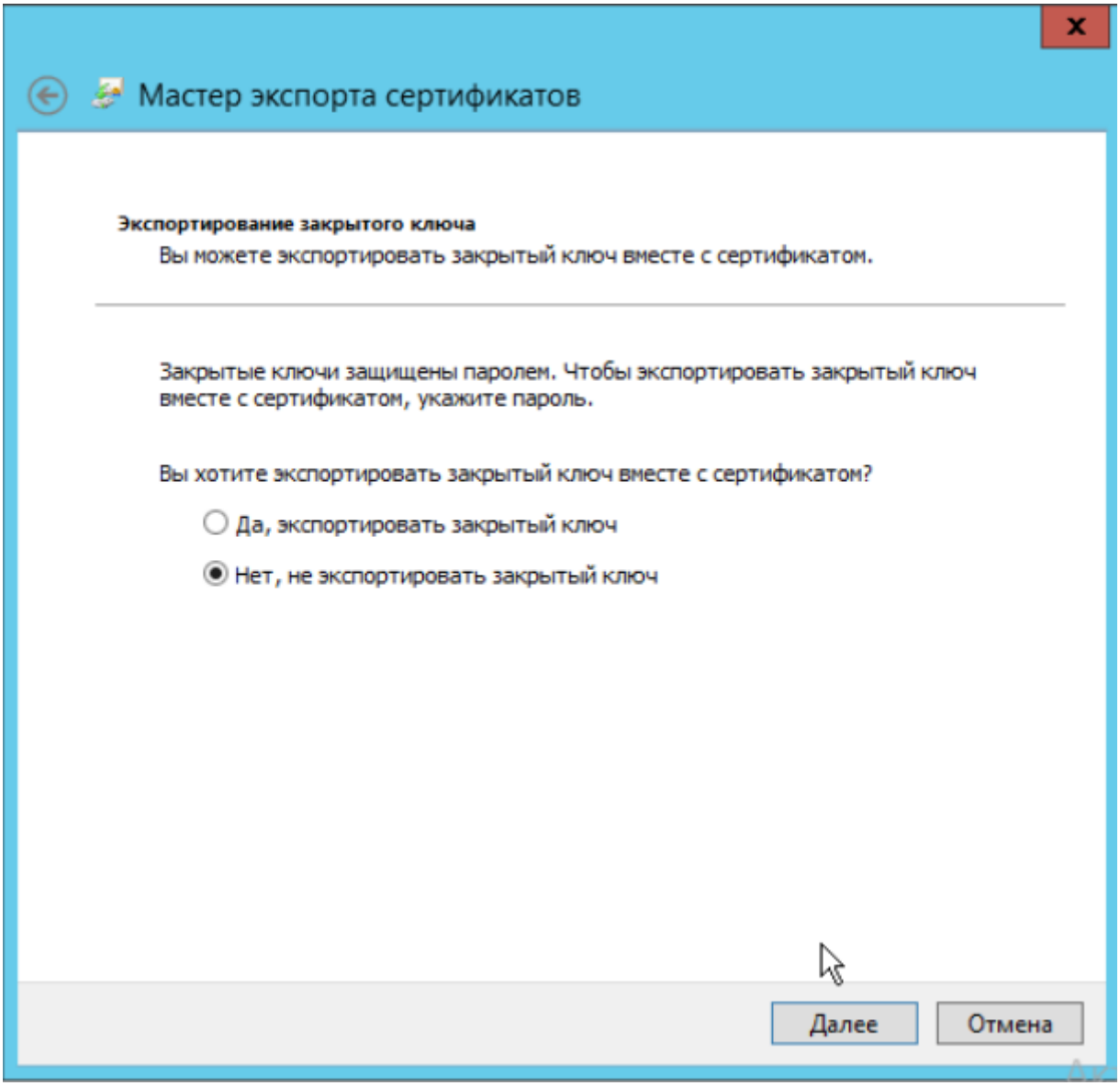


В списке доступных сертификатов выбрать созданный ранее, выпустить сертификат (кнопка **Выпустить сертификат**) и экспортировать (контекстное меню **Все задачи -> Экспорт**).

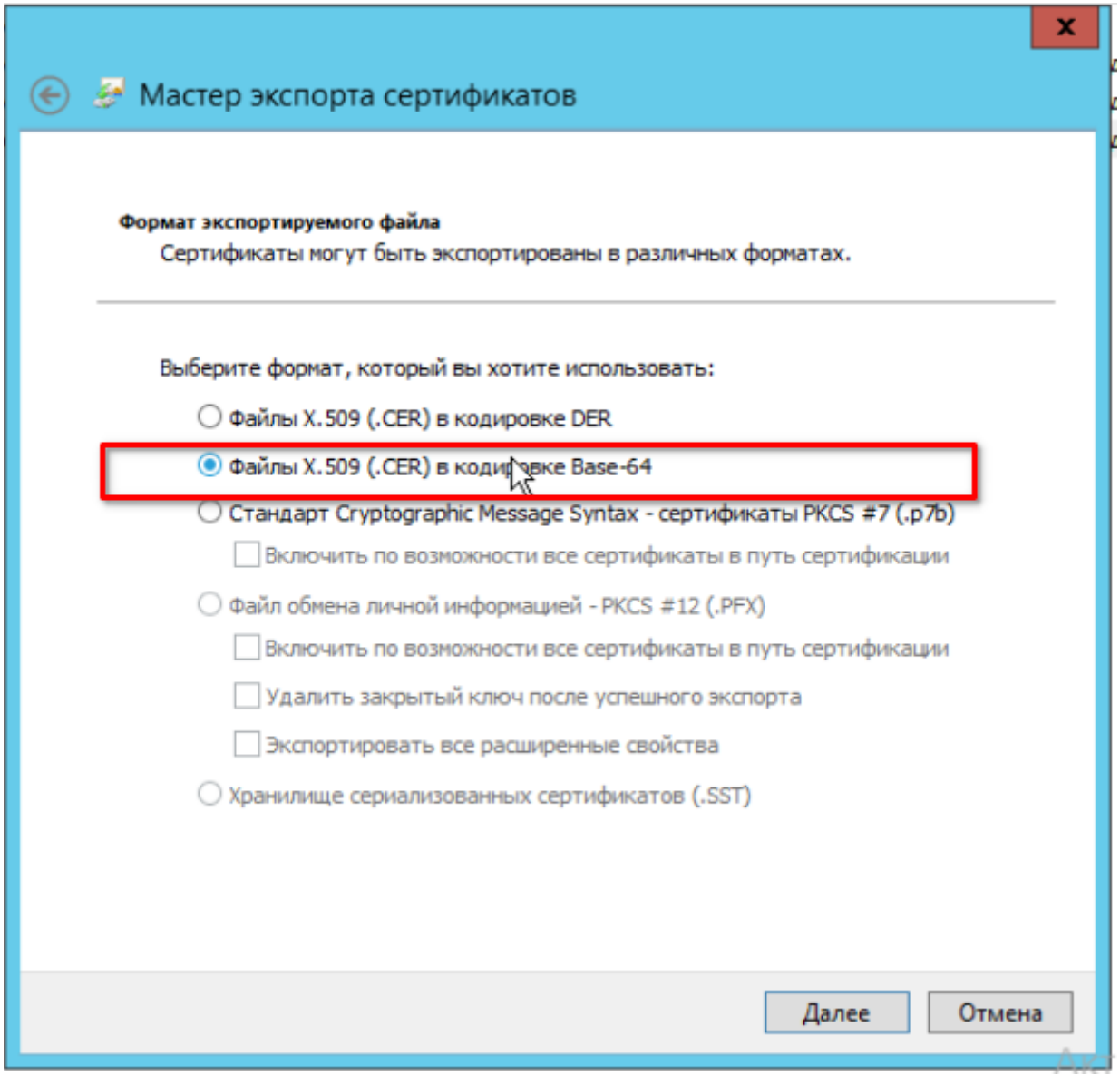


Откроется мастер экспорта сертификатов:

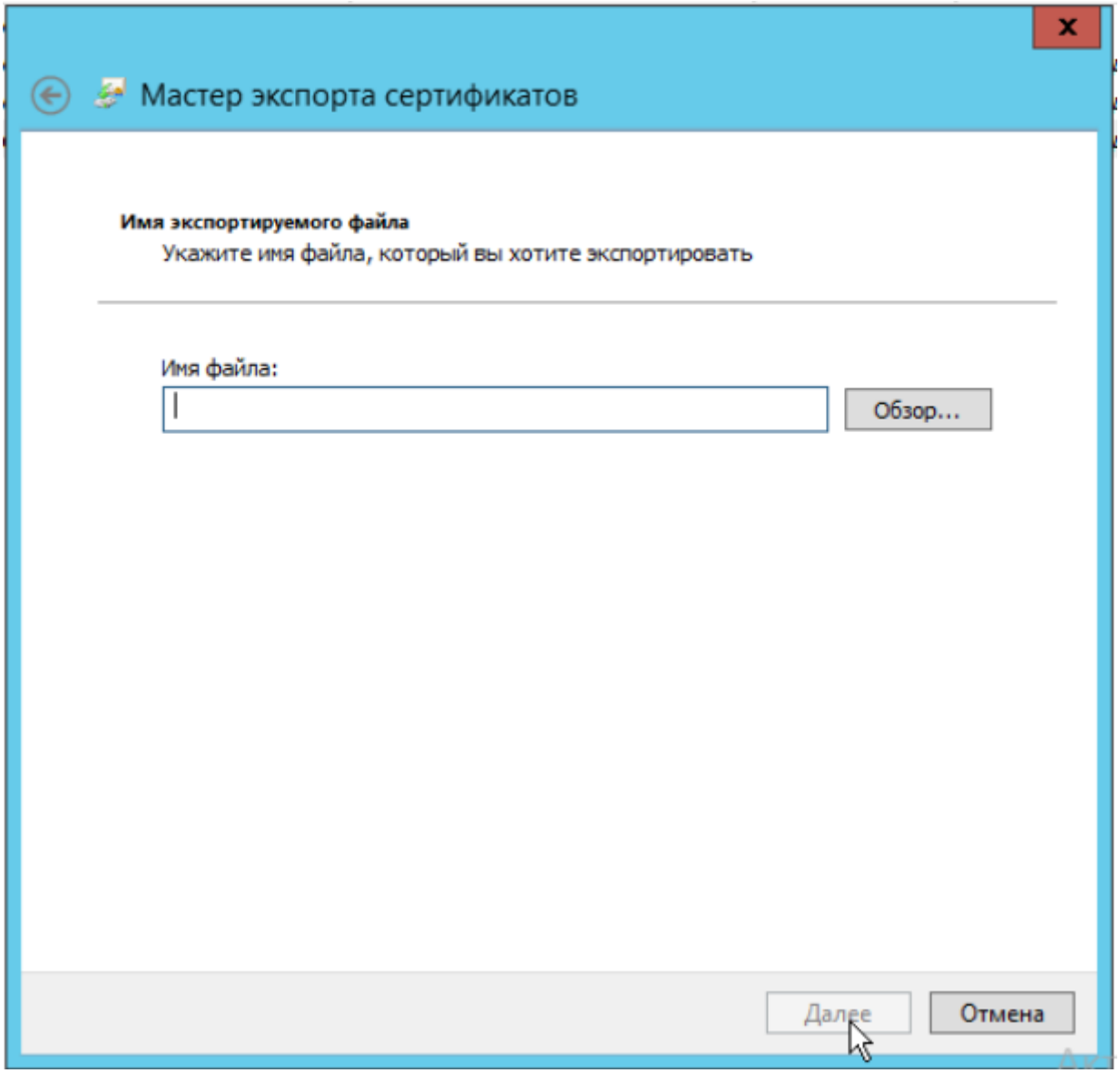

Мастер экспорта сертификатов
Мастер экспорта сертификатов
Этот мастер помогает копировать сертификаты, списки доверия и списки отзыва сертификатов из хранилища сертификатов на локальный диск.
Сертификат, выданный центром сертификации, является подтверждением вашей личности и содержит информацию, необходимую для защиты данных или установления защищенных сетевых подключений. Хранилище сертификатов — это область системы, предназначенная для хранения сертификатов.
Для продолжения нажмите кнопку "Далее".
Далее
Отмена



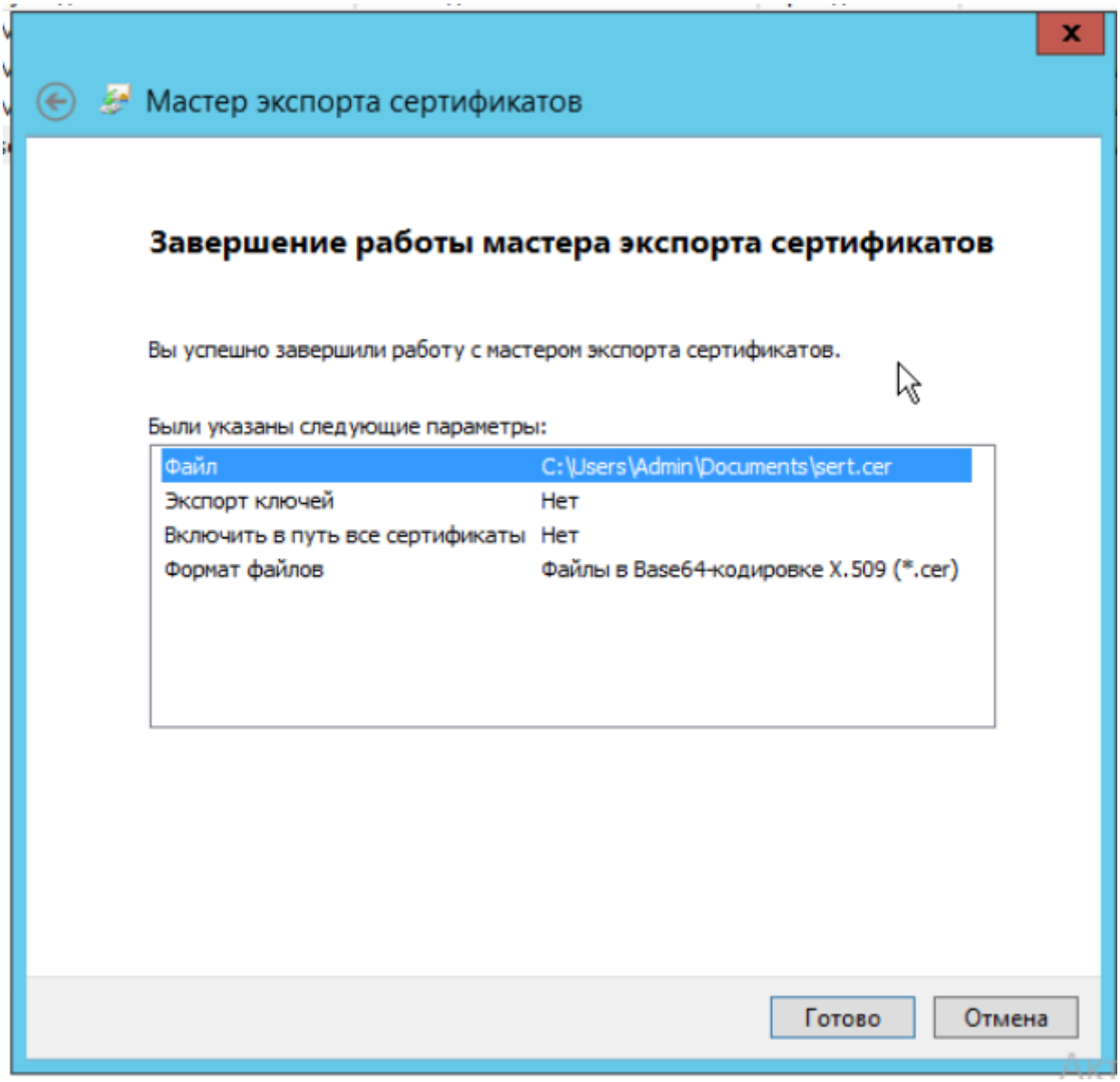
Выбрать кодировку base64 для файла сертификата:



Задать имя файла:



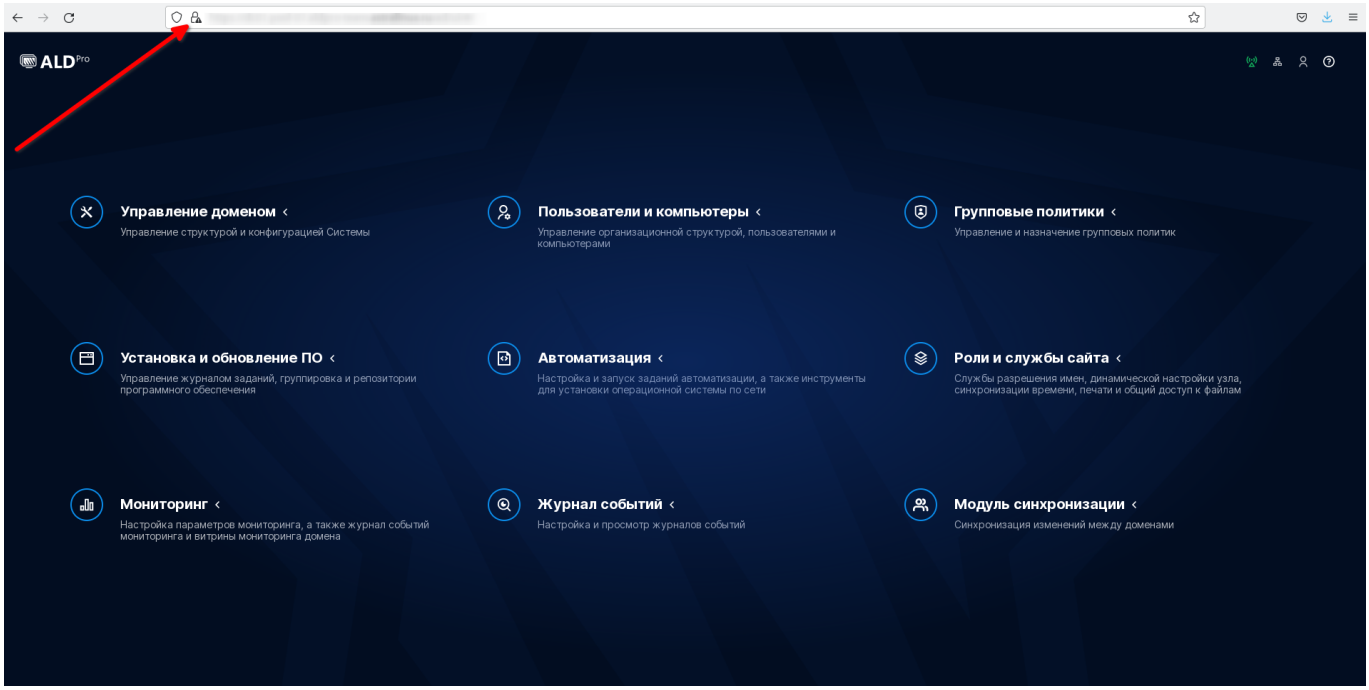
После нажатия кнопки **Готово** будет выгружен файл сертификата <имя_сертификата.cer>.



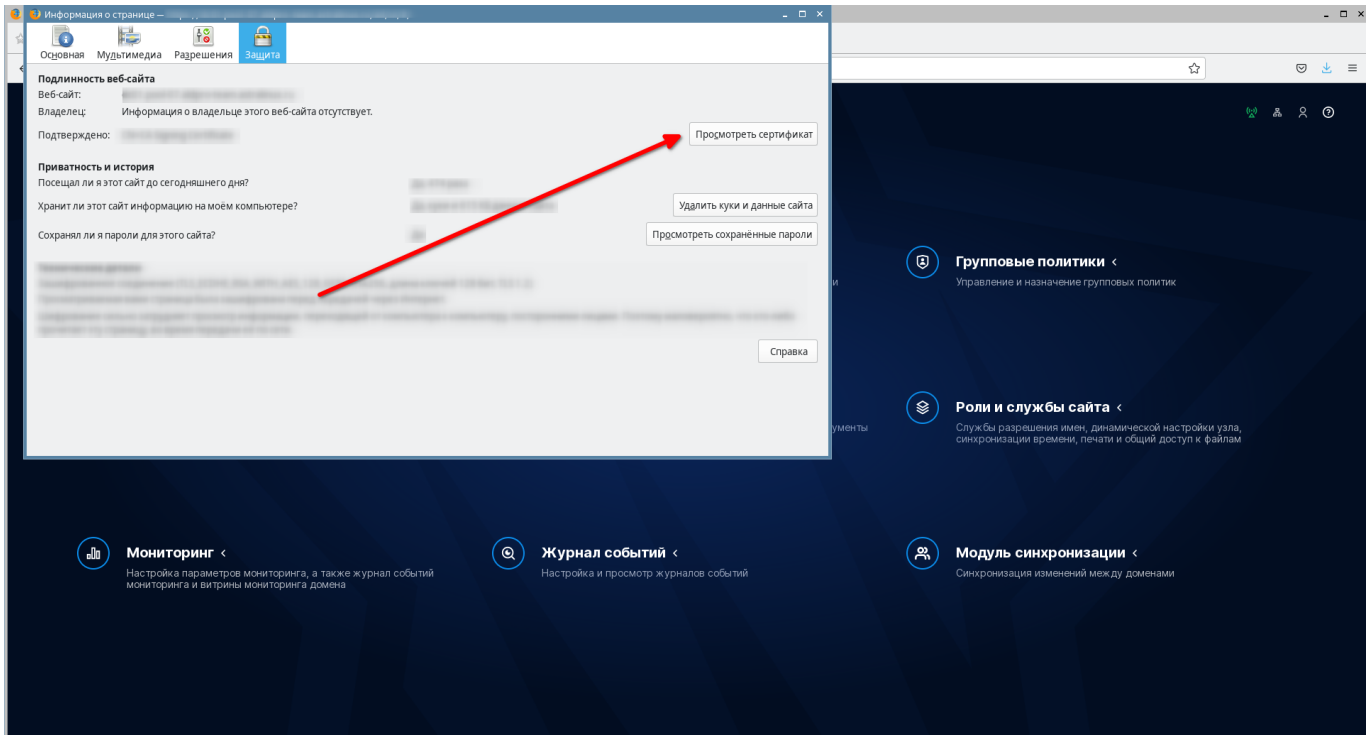
Необходимо изменить формат файла сертификата на *.pem, переименовав файл в <имя_сертификата.pem>.

Выгрузка сертификатов для контроллера домена ALD Pro

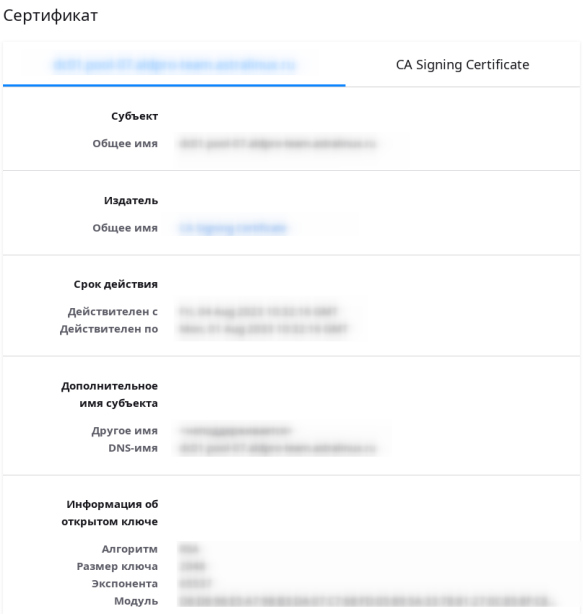
Для выгрузки сертификата необходимо зайти на портал управления ALD Pro, нажать на кнопку просмотра информации о сайте и выбрать **Незащищенное соединение -> Подробнее**:



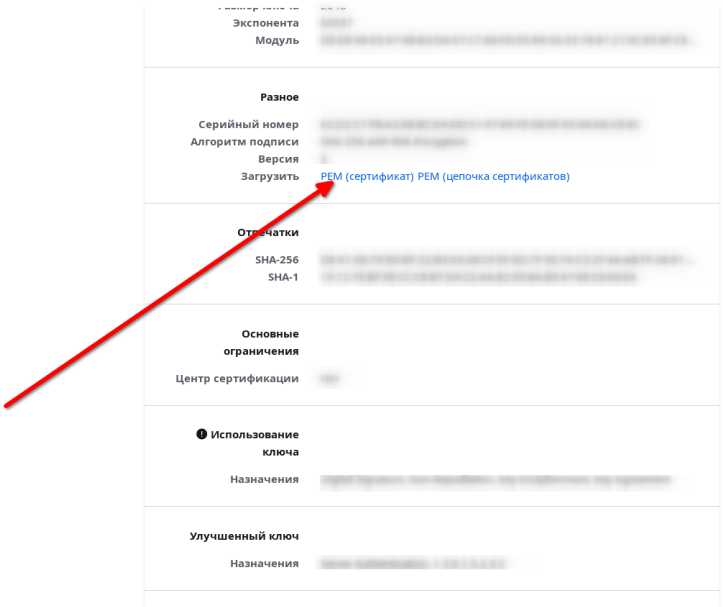
Выбрать **Просмотреть сертификат**:



Откроется файл сертификата:



Открыть ссылку **PEM (сертификат)**:



Сертификат сайта сохранен.

6.8.2.2. Настройка синхронизации паролей MS AD → ALD

Развертывание

Для синхронизации паролей пользователей из домена Windows в домен ALD Pro на контроллере домена Windows необходимо настроить групповую политику фильтра

паролей.

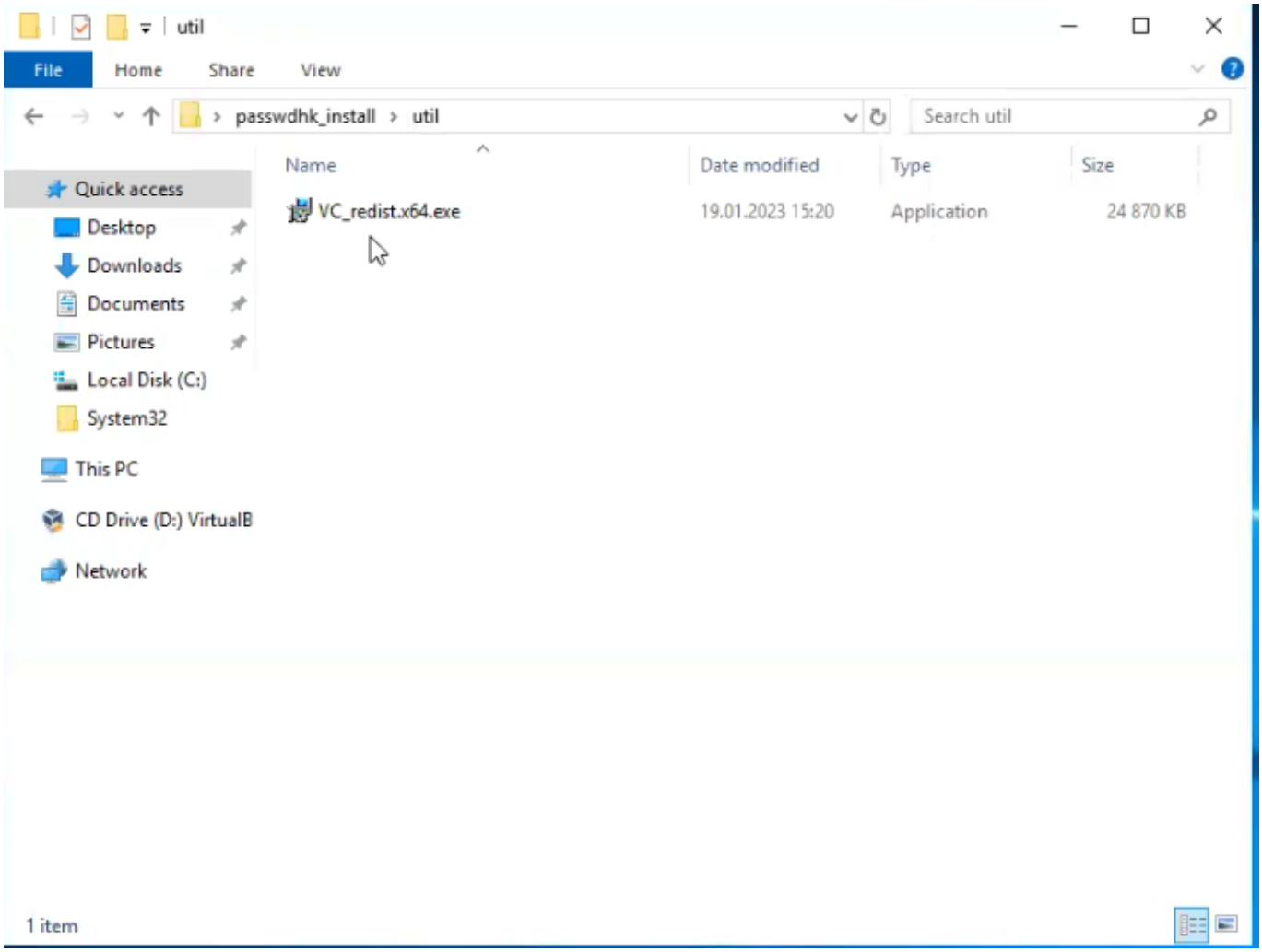
Для настройки потребуется файл **passwdhk_install.zip**, который можно скачать в личном кабинете пользователя Astra Linux https://lk-new.astralinux.ru

---

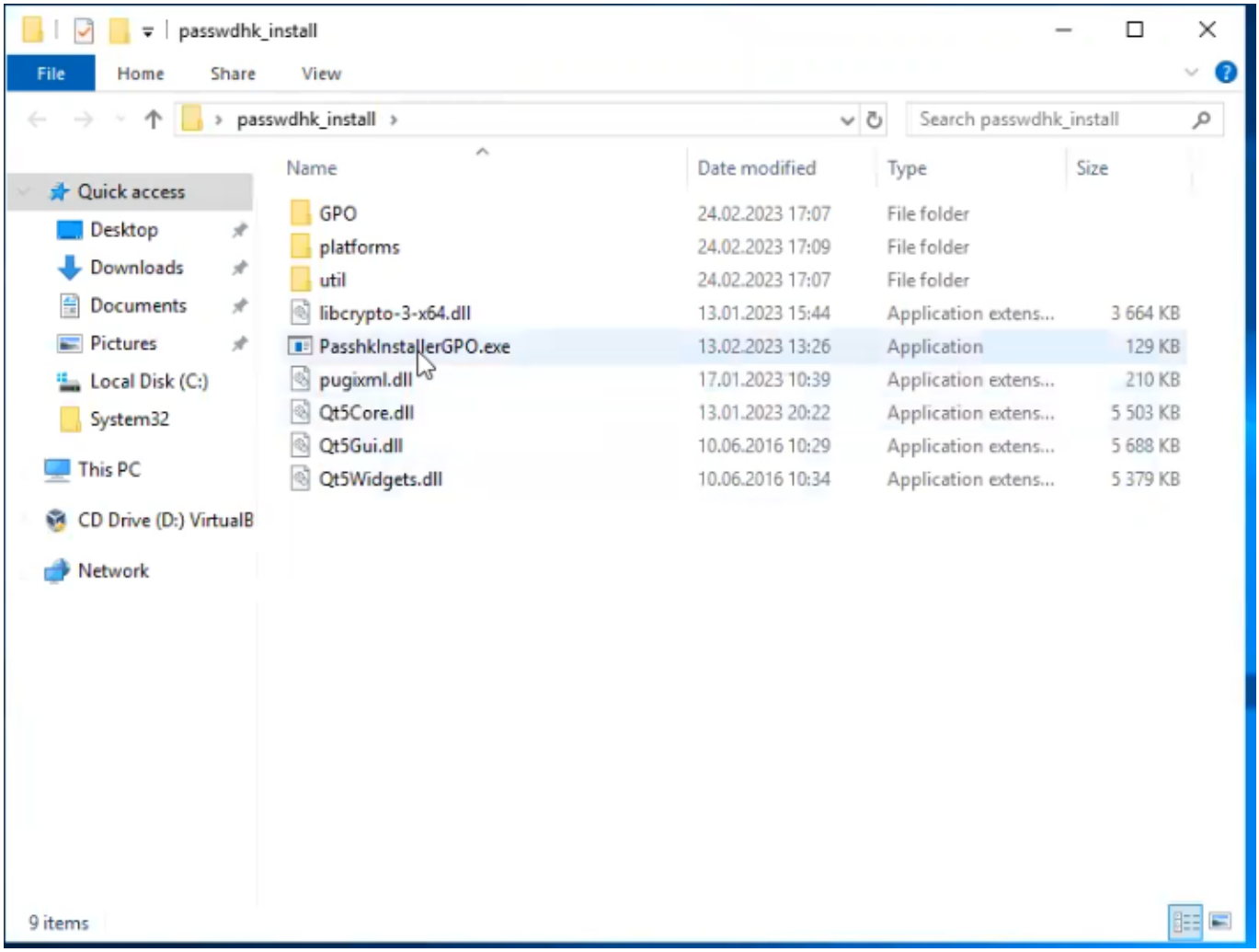
Важно: Синхронизация пароля пользователя будет успешна только для тех учетных записей MS AD, для которых пароль задавался после создания Групповой Политики passwdhk. Для синхронизации старых УЗ требуется либо создавать пользователей после настройки passwdhk, либо менять всем целевым УЗ пароль после настройки passwdhk.

---

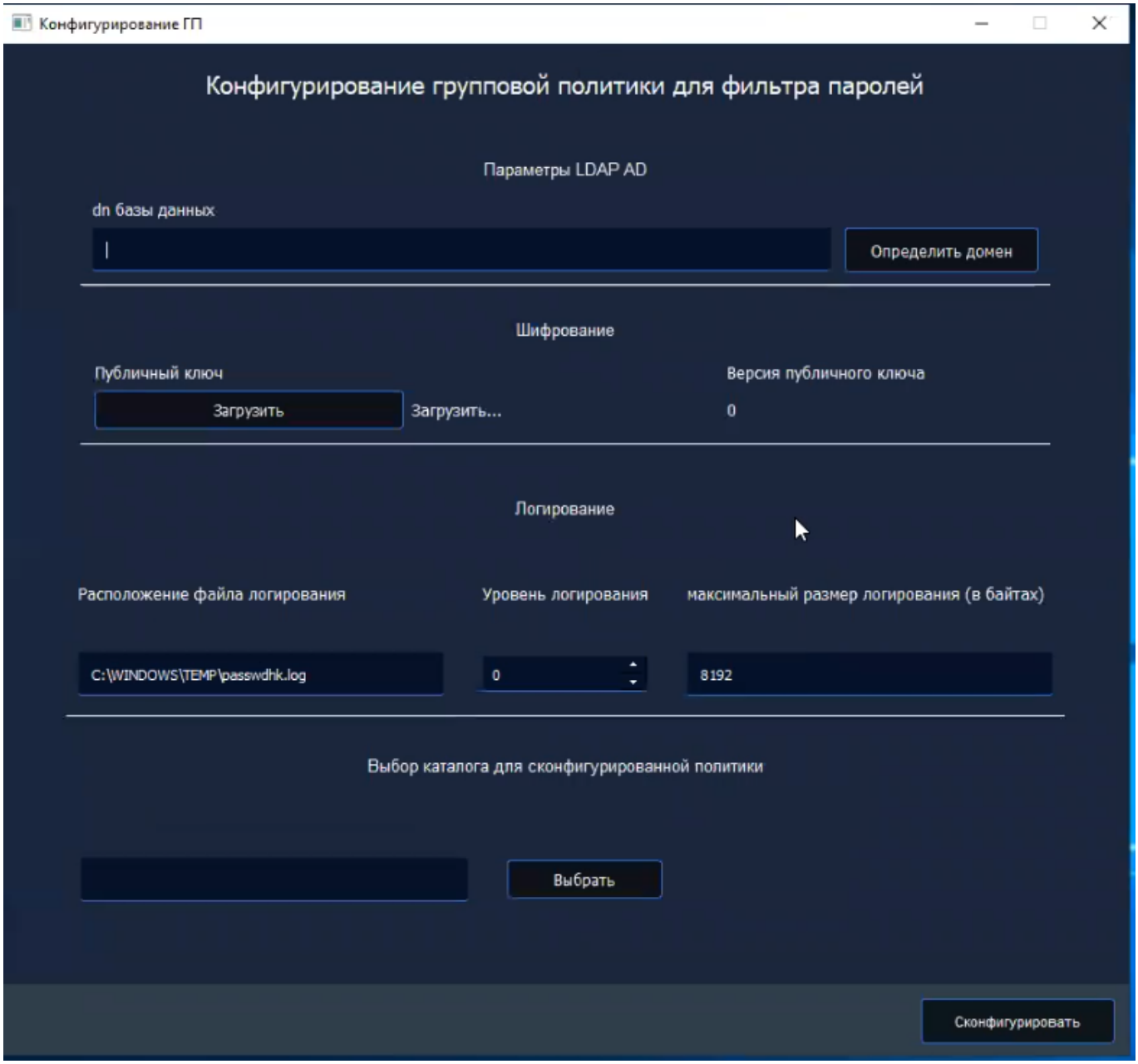
1. Для установки visual c++ sdk необходимо на всех контроллерах домена выполнить `passwdhk_install/util/VC_redistx.64.exe`



2. На любом контроллере домена для настройки групповой политики фильтра паролей необходимо выполнить passwdhk_install/PasskhIntallerGPO.exe



3. Интерфейс приложения



Кнопка **Определить домен**: определяет домен, в котором развернут контроллер домена MS AD;

Публичный ключ → **Загрузить**: загружаем файл открытого ключа, скачанный при развертывании модуля синхронизации в ALD Pro **Модуль синхронизации -> Настройки -> Синхронизация паролей -> Кнопка “Получить ключ”** (подробнее см. *Получение открытого ключа*);

Версия публичного ключа: подтягивается из файла открытого ключа;

Расположение файла логирования: необходимо указать директорию и файл типа .log, куда будут записываться журналы;

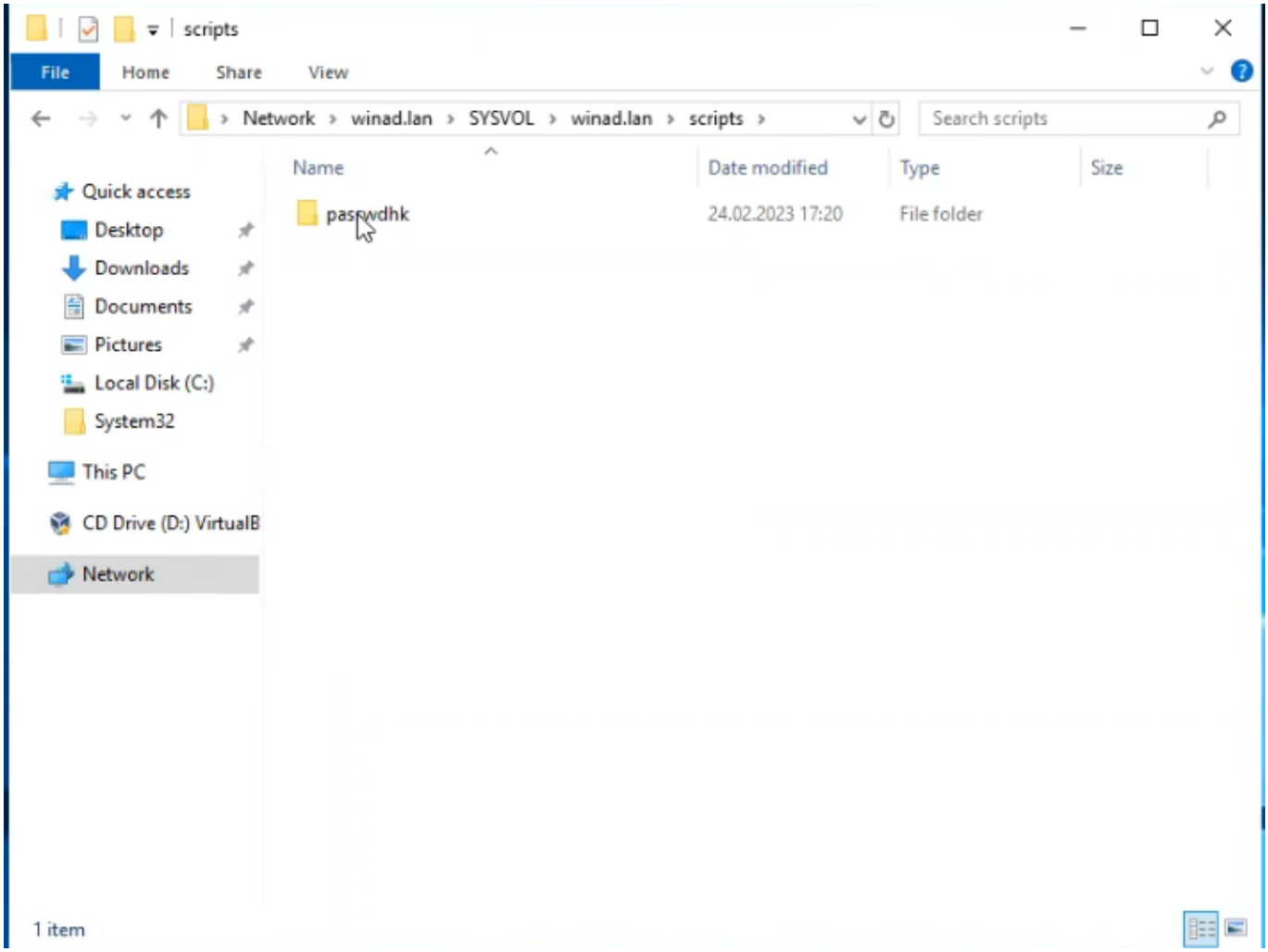
Уровень логирования: указываем один из 4х уровней логирования (подробнее ниже);

Максимальный размер логирования (в байтах): максимальный размер файла логирования (по достижению максимального размера файлом логирования, он архивируется);

Выбор каталога для сконфигурированной политики: указываем директорию для хранения сконфигурированной политики **исключительно** символами латиницы.

Конфигурация групповой политики производится только после заполнения всех полей в приложении и нажатия на кнопку **Сконфигурировать**.

4. В расположение `\winad\sysvol\winad\scripts` необходимо скопировать папку passwdhk из каталога для сконфигурированной политики, указанного на предыдущем шаге:



5. Следующим шагом необходимо запустить оснастку “Управление групповой политикой”.

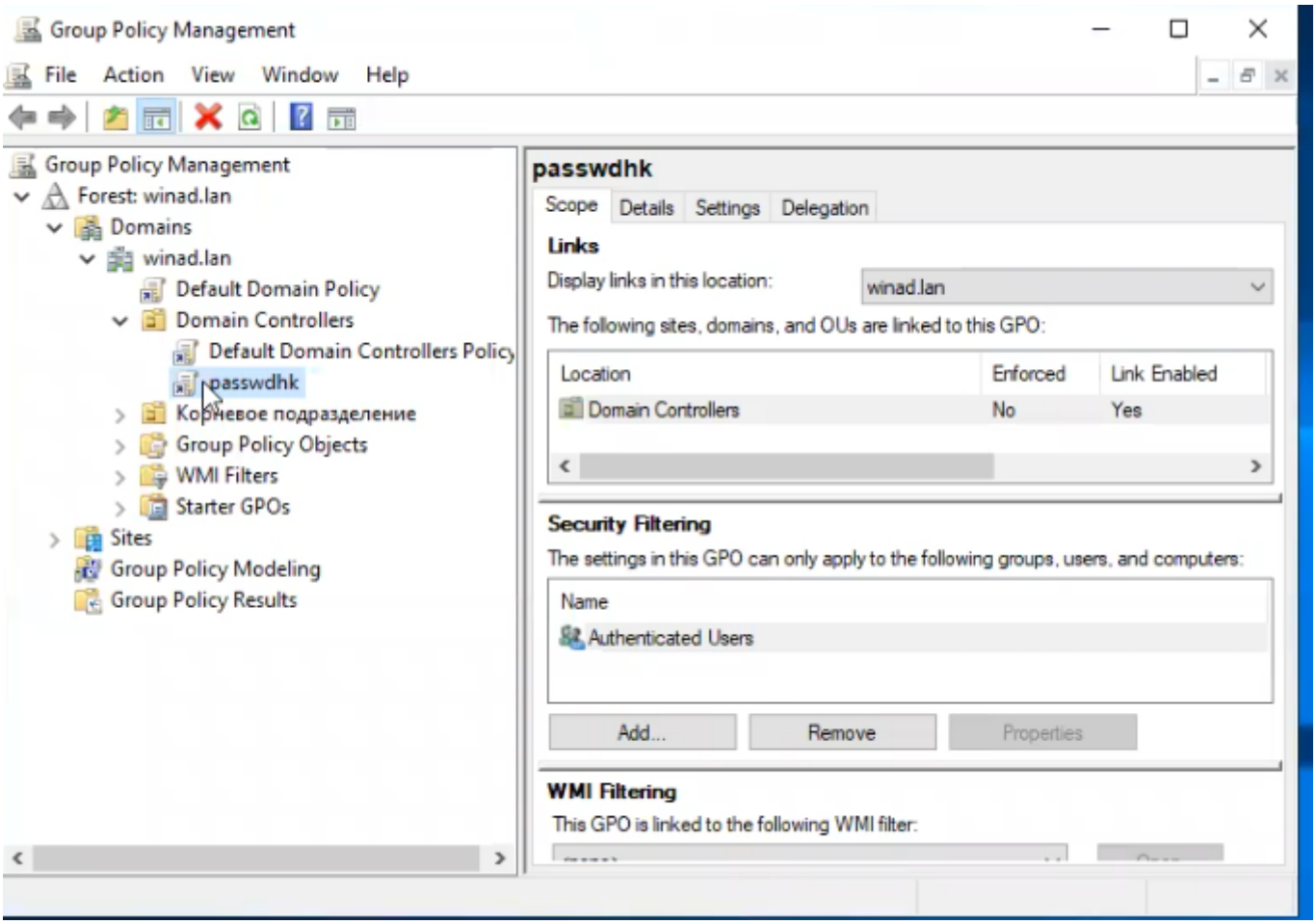
6. В папке “Объекты групповой политики” необходимо создать новую пустую групповую

политику с названием passwdhk.

7. Для созданной политики passwdhk выполнить импорт настроек.

8. На шаге с выбором расположения импорта настроек необходимо указать каталог Policies из каталога, указанного для сконфигурированной политики.

9. Групповую политику необходимо переместить в расположение “Контроллеры домена”



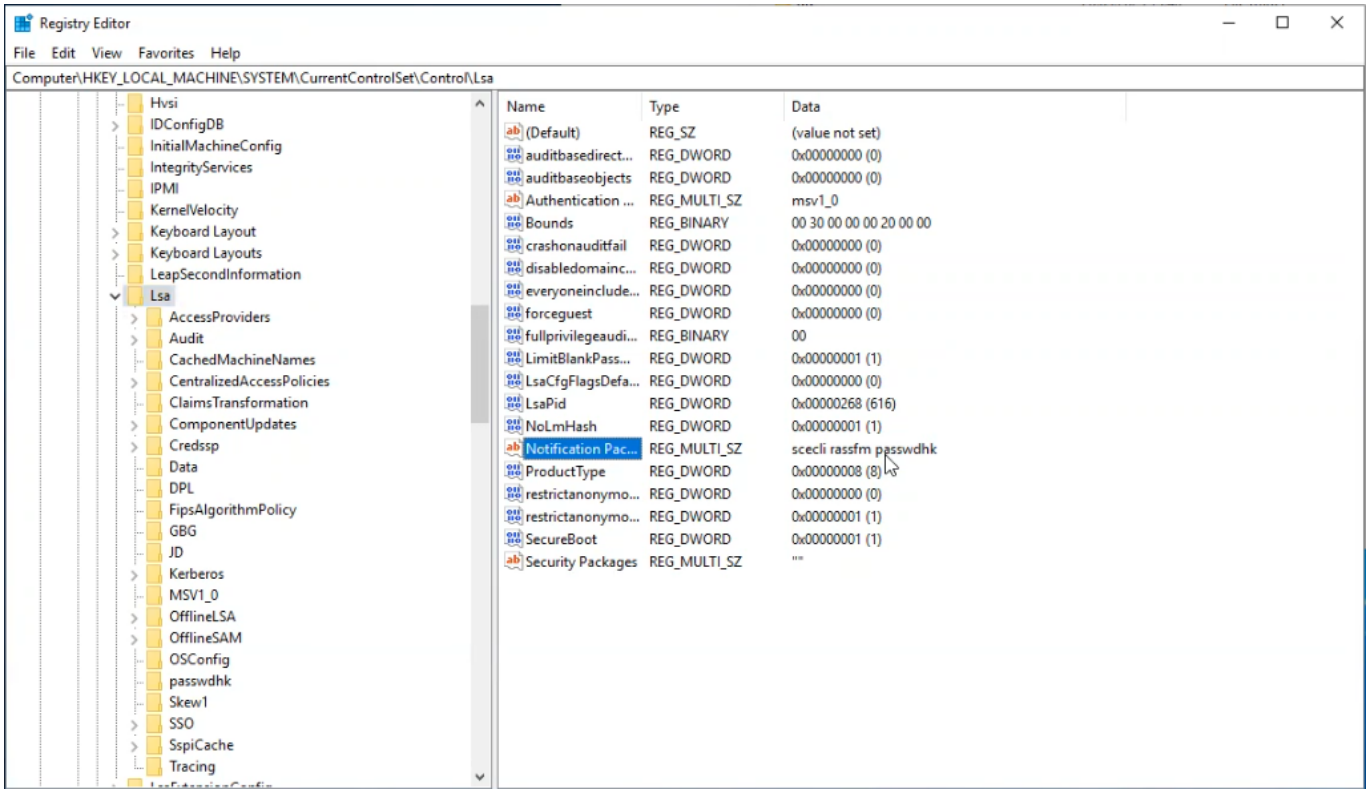
10. Для применения групповой политики в консоли powershell необходимо выполнить команду:

```
gpupdate /force
```

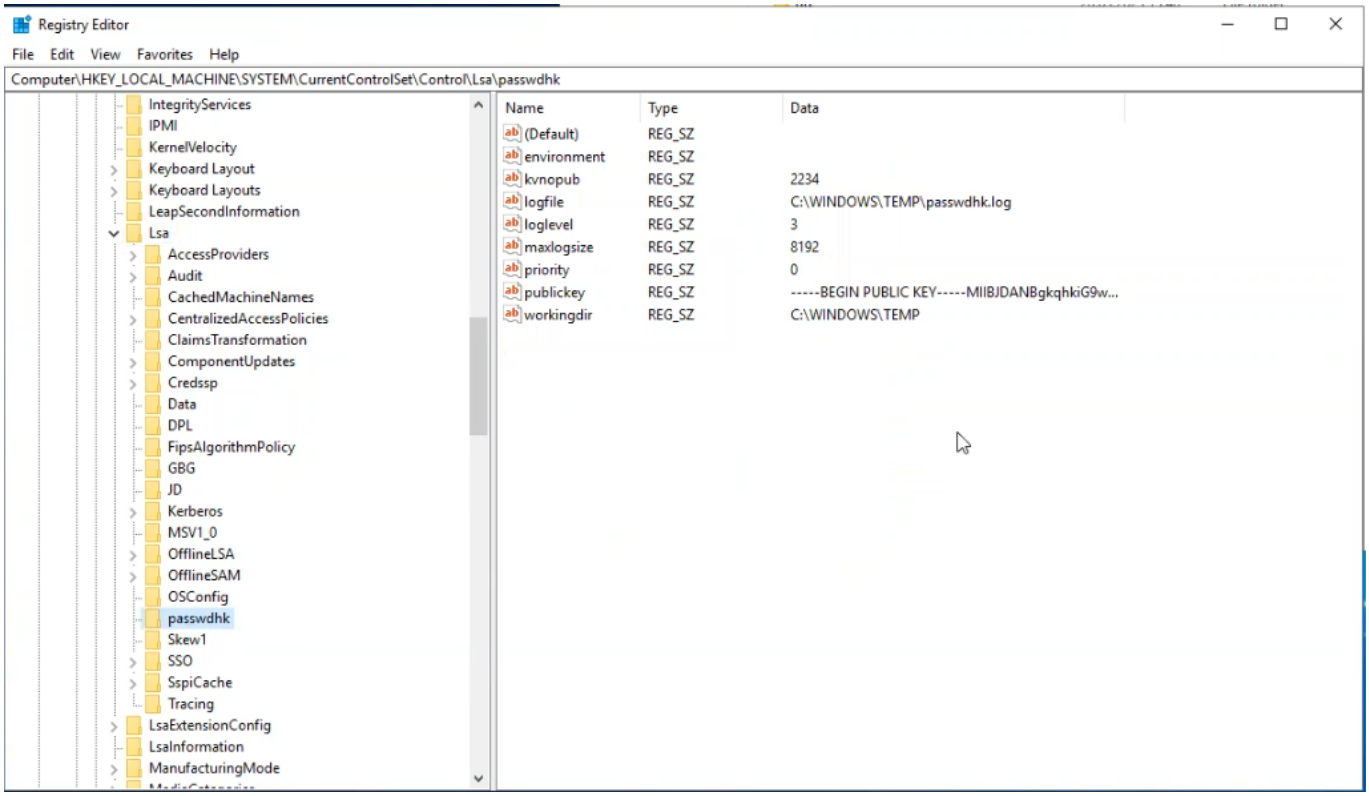
11. Выполнить поочередную перезагрузку контроллеров домена.

12. Проверьте, что в реестре создались необходимые записи.

Первая:



Вторая:



Важно: Файл passwdhk.dll не будет работать при включённом механизме Windows LSA

Protection.

Для проверки работы механизма Windows LSA Protection необходимо:

1. Открыть оснастку **Редактор реестра** (или ввести в поиске `regedit`)
2. В колонке слева открыть вкладки HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Control
3. Выбрать вкладку LSA
4. В окне имен регистров справа выбрать регистр «RunAsPPL»
5. Если регистр «RunAsPPL» принимает значение dword:00000001 или dword:00000002, значит на контроллере домена включен механизм Windows LSA Protection

---

Важно: Решение об отключении механизма Windows LSA Protection должно приниматься администраторами домена

---

1. Открыть оснастку **Редактор реестра** (или ввести в поиске `regedit`)
2. В колонке слева открыть вкладки HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Control
3. Выбрать вкладку LSA
4. В окне имен регистров справа выбрать регистр «RunAsPPL»
5. Для отключения задайте для раздела реестра значение RunAsPPL=dword:00000000 или удалите DWORD.
6. Перегрузите компьютер

Инструкция по замене файла passwdhk.dll

Этот раздел инструкции актуален для ситуации, когда модуль синхронизации был развернут в версиях ALD Pro старше 2.3.0. Замена файла passwdhk.dll решает проблемы зависания консоли MMC при смене пароля пользователя MS AD.

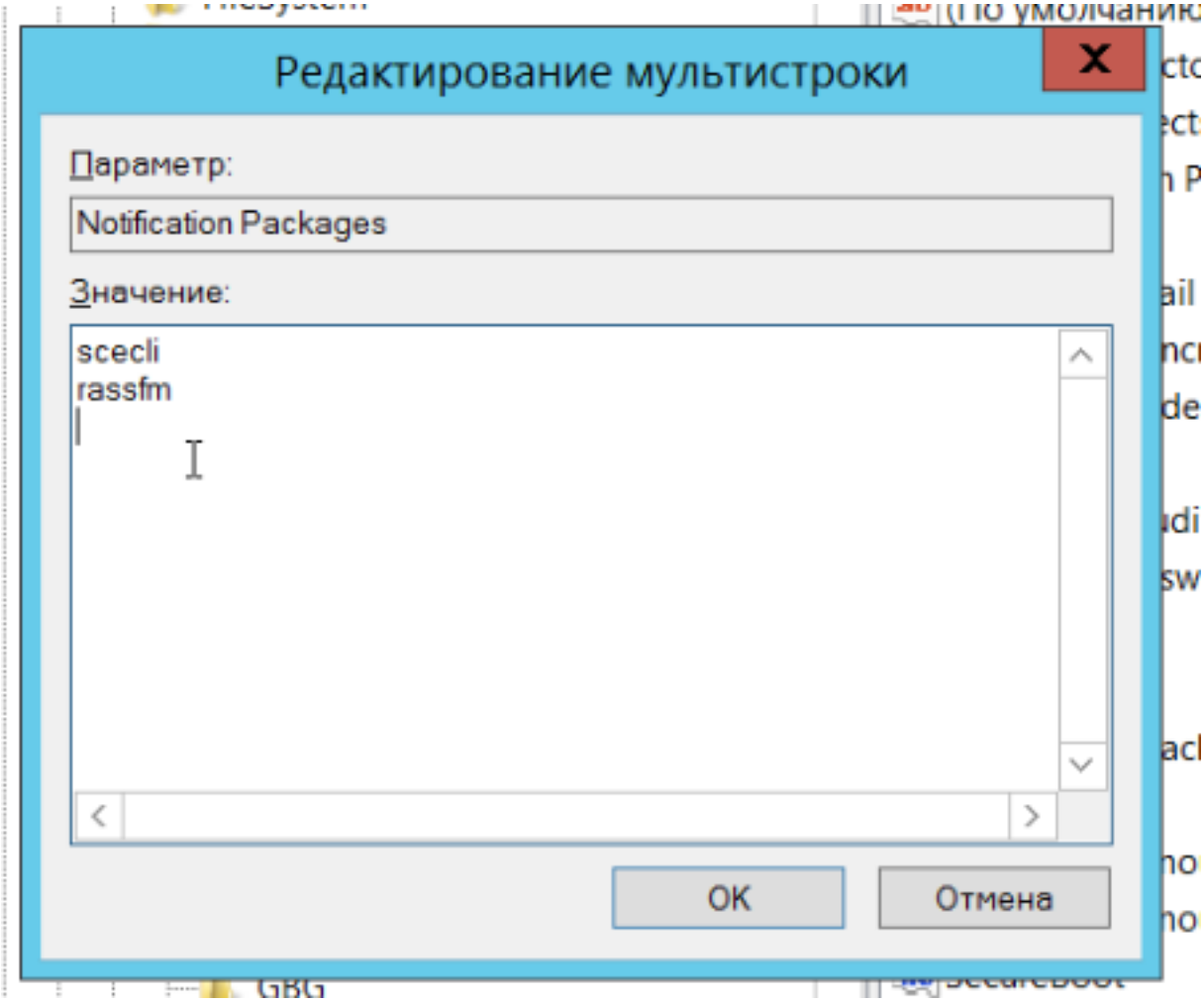
Для настройки потребуется обновленный файл passwdhk_install.zip, который можно

скачать в личном кабинете пользователя Astra Linux

Деактивация текущего passwdhk

1. Открыть оснастку “Редактор реестра” (или ввести в поиске “regedit”)
2. Далее в колонке слева открыть вкладки HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Control
3. Затем найти вкладку Lsa и выбрать её
4. В окне справа будут выведены имена регистров. Нужно найти и кликнуть по регистру “Notification Packages”
5. Далее откроется окно “Редактирование мультистроки”. В нем нужно найти строку passwdhk и удалить её.

Важно: После удаления строки passwdhk проверьте, чтобы самая нижняя строка была пустой - на ней должен мигать курсор.



6. Затем нажать Ок и закрыть Редактор реестра.
7. Перезагрузить машину

Занесение и активация нового passwdhk

1. Распаковка passwdhk
    - Открыть расположение `C:\Windows\System32`. Найти файл `passwdhk.dll` и удалить его.
    - Распаковать архив, найти в нем файл `passwdhk.dll` по расположению (относительно корневой папки проекта) `./GPO/scripts/passwdhk/`
    - Скопировать этот файл в расположение `C:\Windows\System32`
2. Активация `passwdhk`
    - Далее открыть оснастку “Редактор реестра” (или ввести в поиске `regedit`)
    - В колонке слева открыть вкладки HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Control
    - Затем найти вкладку Lsa и выбрать её
    - В окне справа будут выведены имена регистров. Нужно найти и кликнуть по регистру “Notification Packages”
    - Далее откроется окно “Редактирование мультистроки”. В самой нижней строке (она должна быть пустой) ввести passwdhk. После этой строки нужно сделать еще одну пустую.
    - Затем нажать Ок и закрыть Редактор реестра.
    - Перезагрузить машину

Уровни логирования

0 - отсутствие логирования

1 - в лог файл записываются только сообщения об ошибках

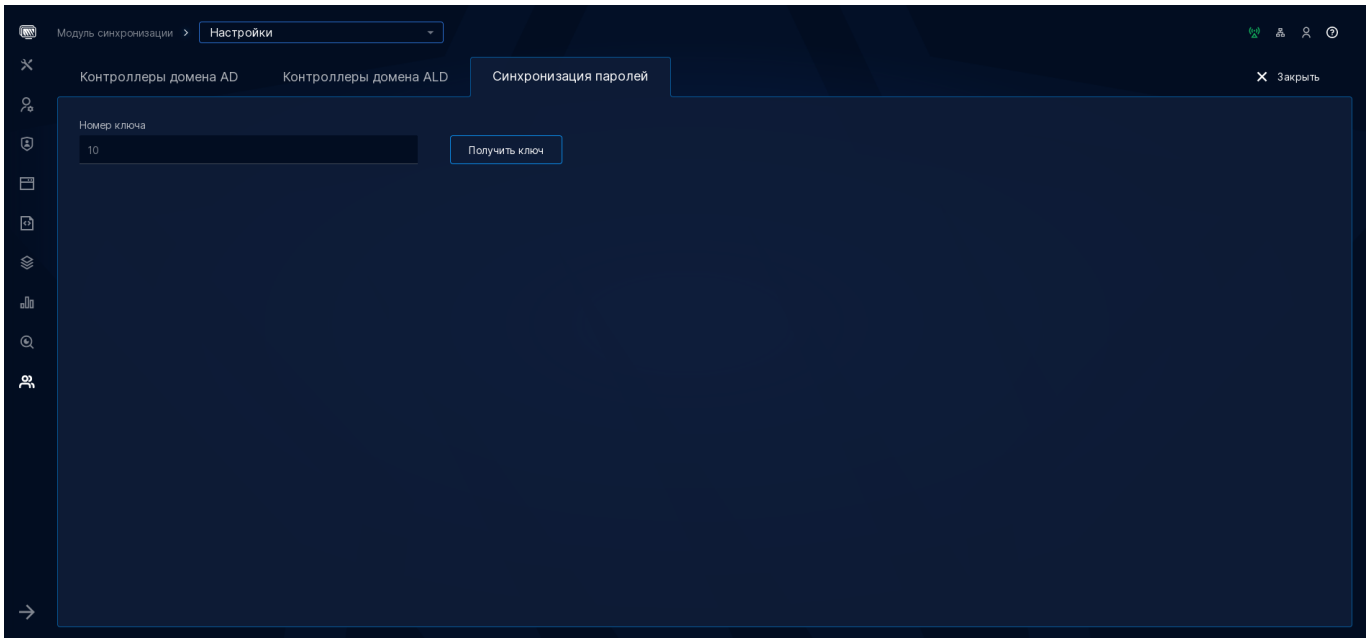
2 - в лог файл записываются сообщения об ошибках и информационные сообщения

3 - в лог файл записываются сообщения об ошибках, информационные сообщения + информация для отладки: все записи о процессе шифрования, какой ключ используется и т.д.

6.8.2.3. Синхронизация паролей ALD Pro → AD

Получение открытого ключа

В интерфейсе ALD Pro открыть **Модуль синхронизации** → **Конфигурации** → **Синхронизация паролей**



По нажатию кнопки **Получить ключ** скачивается файл открытого ключа и актуализируется порядковый номер ключа.

Синхронизация паролей

Скачанный файл открытого ключа необходимо сохранить в любую директорию, например, `/tmp/`. Следующим шагом файл открытого ключа необходимо скопировать в расположение `/opt/rbta/aldpro/syncer/`:

```
sudo cp /tmp/public.gpg  /opt/rbta/aldpro/syncer/public.gpg
```

Перезагружаем контроллер домена:

```
sudo ipactl restart
```

Примечание:

- При изменения пароля синхронизированного пользователя через интерфейс FreeIpa, синхронизация пароля от ALD Pro к AD не произойдет. Для изменения пароля с последующей успешной синхронизацией необходимо использовать интерфейс ПК «ALD Pro»;
- Процесс изменения пароля пользователя через личный кабинет с последующей синхронизацией от ALD Pro к AD проходит успешно только при использовании личного кабинета контроллера домена, на котором был развернут модуль синхронизации;
- При сбросе пароля пользователя через интерфейс ПК «ALD Pro» временный пароль не синхронизируется. После сброса пароля пользователя необходимо залогиниться в домене, изменить пароль с временного на постоянный. При данных условиях произойдет синхронизация пароля от ALD Pro к AD.

Пароли пользователей MS AD, установленные до настройки модуля синхронизации, синхронизированы не будут.

Синхронизация паролей пользователей после настройки модуля синхронизации и сопоставлений подразделений происходит по логике:

- синхронизирован пользователь в домен ALD Pro -> изменен пароль пользователя в домене MS AD -> синхронизирован пароль пользователя в домен ALD Pro
- создан пользователь в домене MS AD -> синхронизирован пользователь в домен ALD Pro

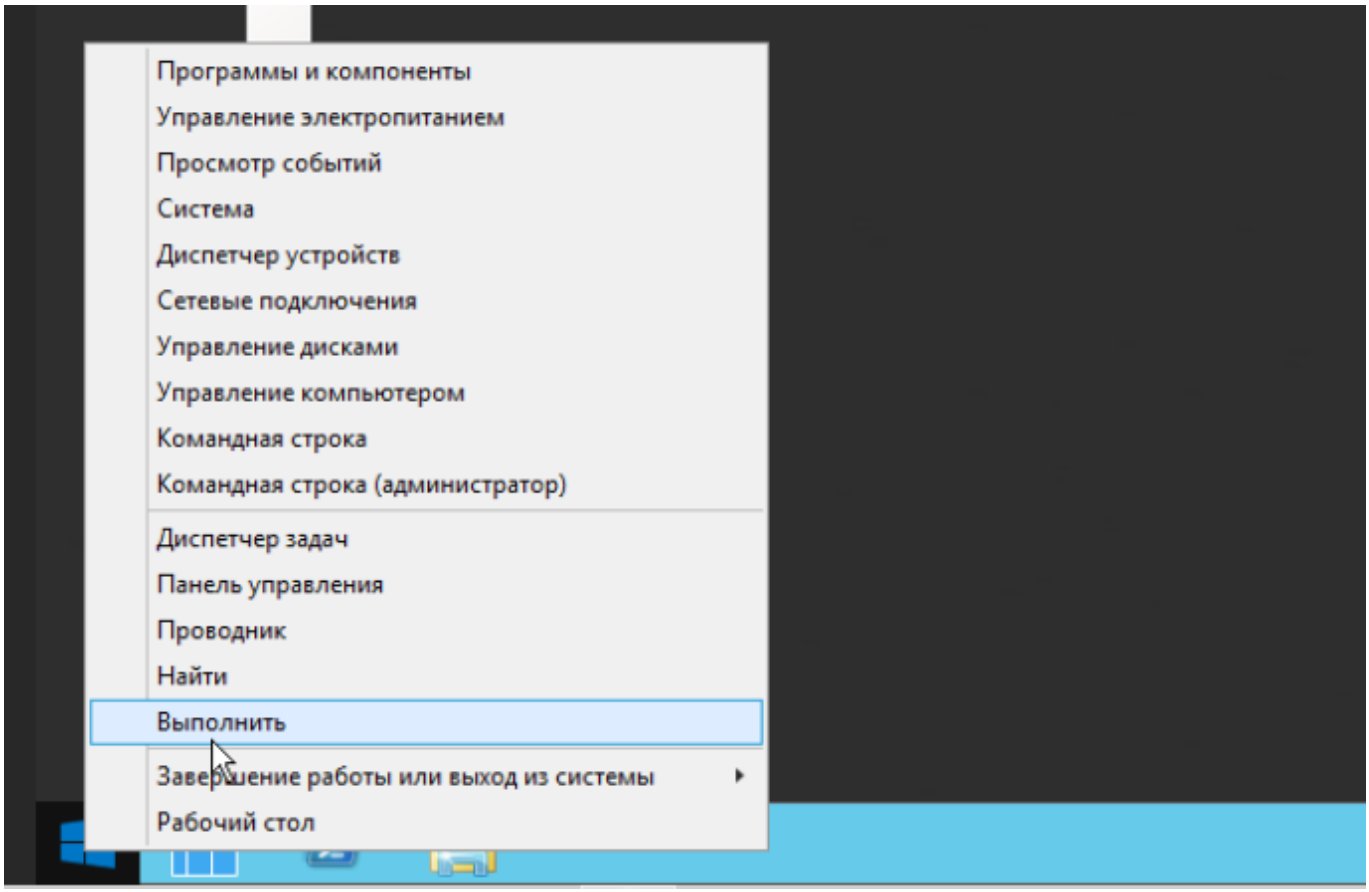
6.8.2.4. Включение TLS на Windows Server 2008R2

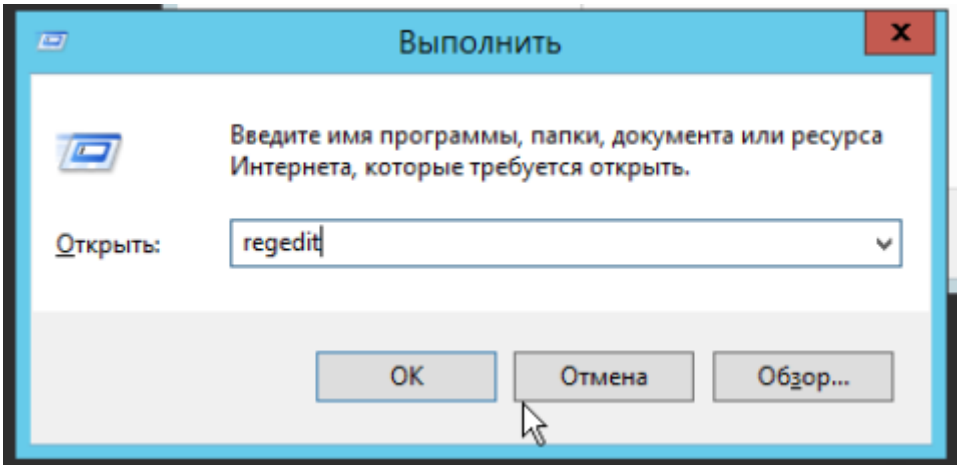
Важно: Настройка выполняется, если на контроллере домена отключено TLS. Данная инструкция применима к контроллерам домена с ОС WS 2008R2. Если версия ОС отличается, необходимо обратиться к официальной документации Microsoft для выбора подходящей инструкции.

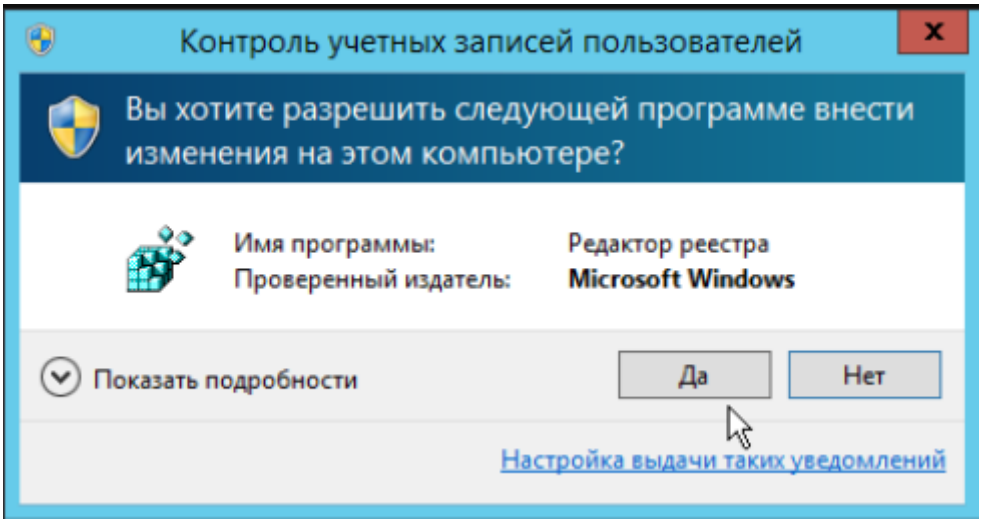
Алгоритм включения TLS

1. Для включения TLS необходимо обновить реестр на сервере контроллера домена.

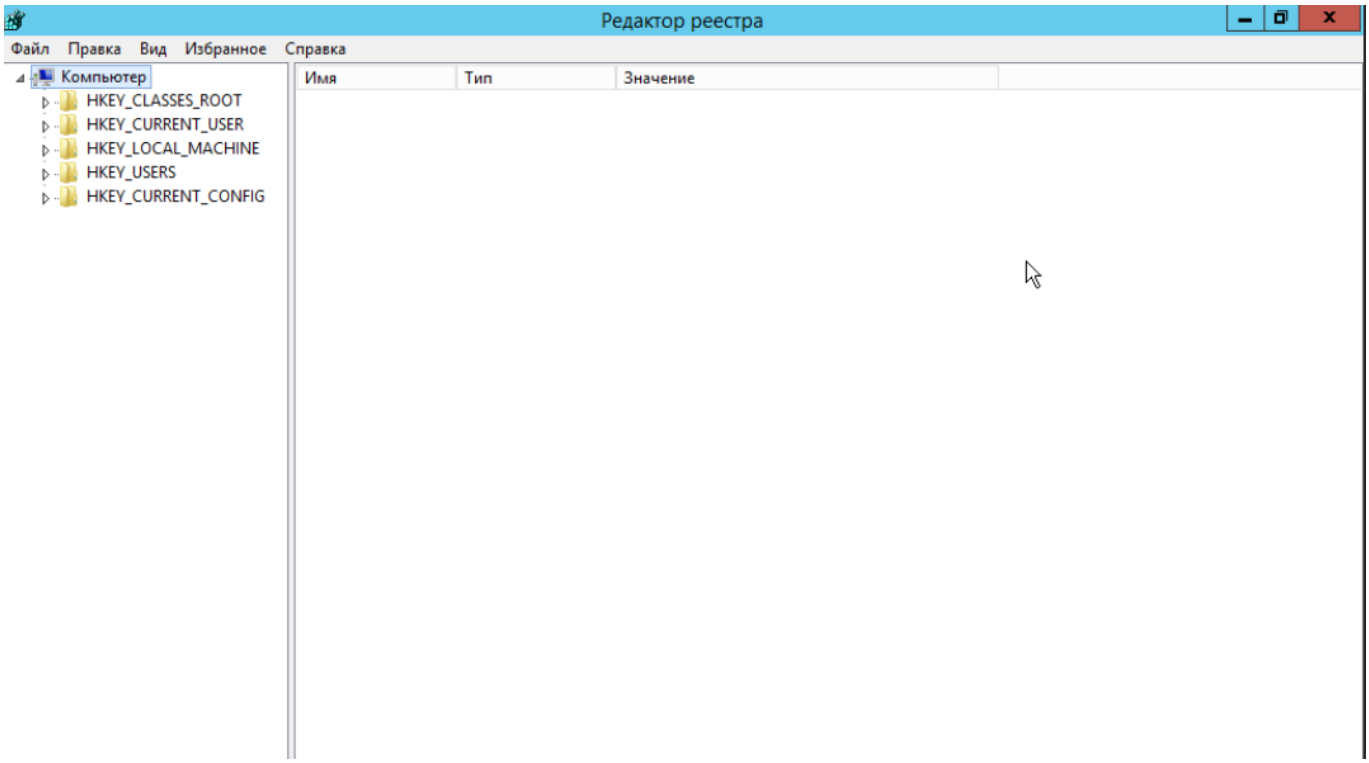
- Необходимо открыть реестр: выполнить “regedit” в окне запуска.





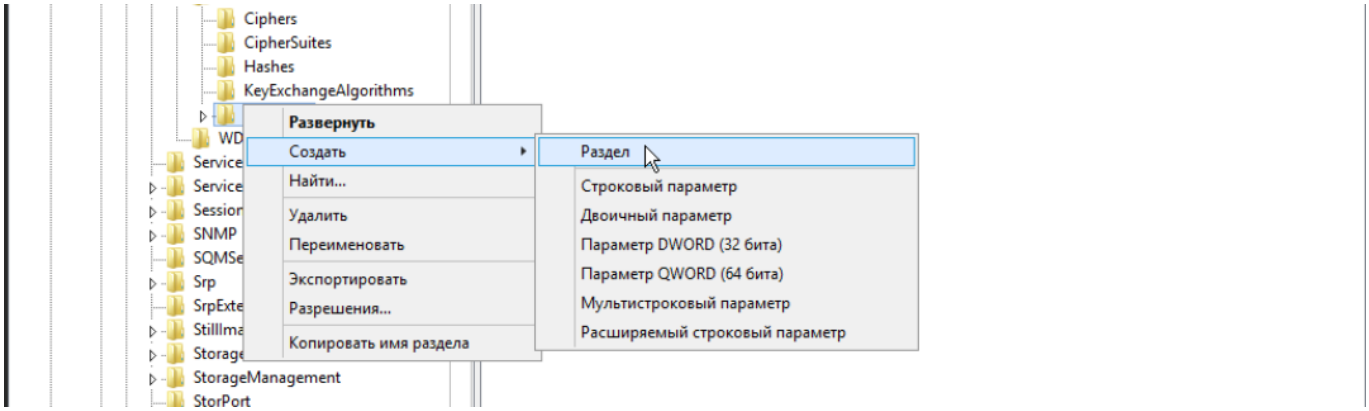


Откроется окно редактора реестра:



- Необходимо перейти к расположению и добавить разделы TLS 1.1 и TLS
- в разделе Protocol:

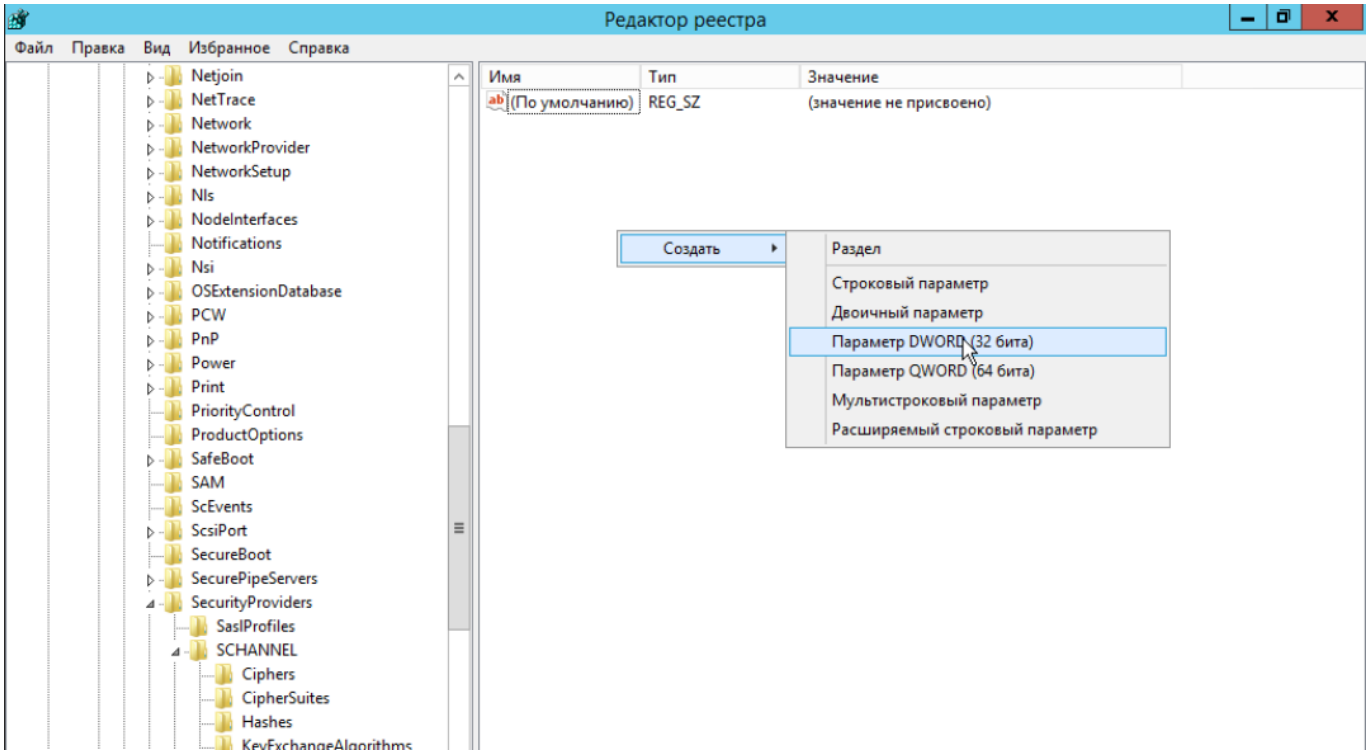
```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\
SCHANNEL\Protocol
```

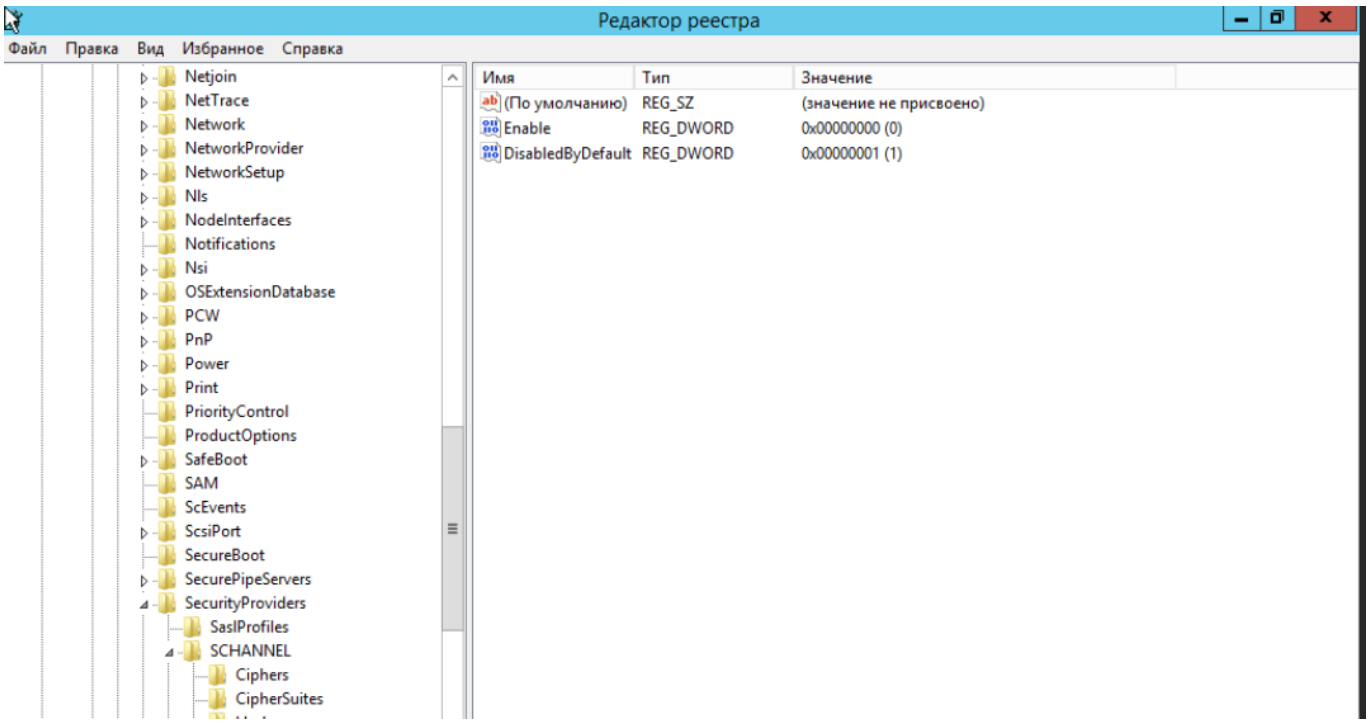


Создаются еще два раздела Client и Server под обоими ключами TLS.

- Теперь необходимо создать значения DWORD в разделе Server и Client со следующими значениями:

```
DisabledByDefault [Значение = 0]
Enabled [Значение = 1]
```





2. Необходимо отключить старых версий TLS и SSL. Для этого:

    1. Открыть реестр на сервере контроллера домена: выполнить “regedit” в окне запуска (аналогично шагу 1)

    2. Перейти к расположению:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\
SCHANNEL\Protocols
```

3. Изменить значения DWORD в разделе Server и Client в разделе ключи TLS 1.0, SSL 3.0 и более старых версий SSL.

```
DisabledByDefault [Значение = 0]
Enabled [Значение = 0]
```

3. После выполнения всех вышеперечисленных настроек необходимо перезагрузить сервер контроллера домена. Данная настройка выполняется на всех контроллерах домена.

6.8.2.5. Особенности настройки при миграции большого количества объектов

По умолчанию 389ds выполняет обновления ссылочной целостности сразу после операции удаления или переименования. В зависимости от количества операций это может повлиять на производительность. Чтобы уменьшить влияние на производительность, есть возможность увеличить время между обновлениями, для этого необходимо задать интервал обновления в секундах. По умолчанию интервал обновления равен 0.

Чтобы показать имя 389ds-сервера:

```
sudo dsctl -l
```

Скопируйте в буфер имя вашего сервера (рядом с ним вы увидите `slapd-GLOBAL-CATALOG`). Чтобы отобразить текущий интервал обновления, запустите (заменив <имя LDAP-сервера> на имя нужного вам LDAP-сервера):

```
sudo dsconf <имя LDAP-сервера> plugin referential-integrity show | grep
↪referint-update-delay
```

Задать новое значение интервала обновления:

```
sudo dsconf <имя LDAP-сервера> plugin referential-integrity set --update-
↪delay <новое_значение>
```

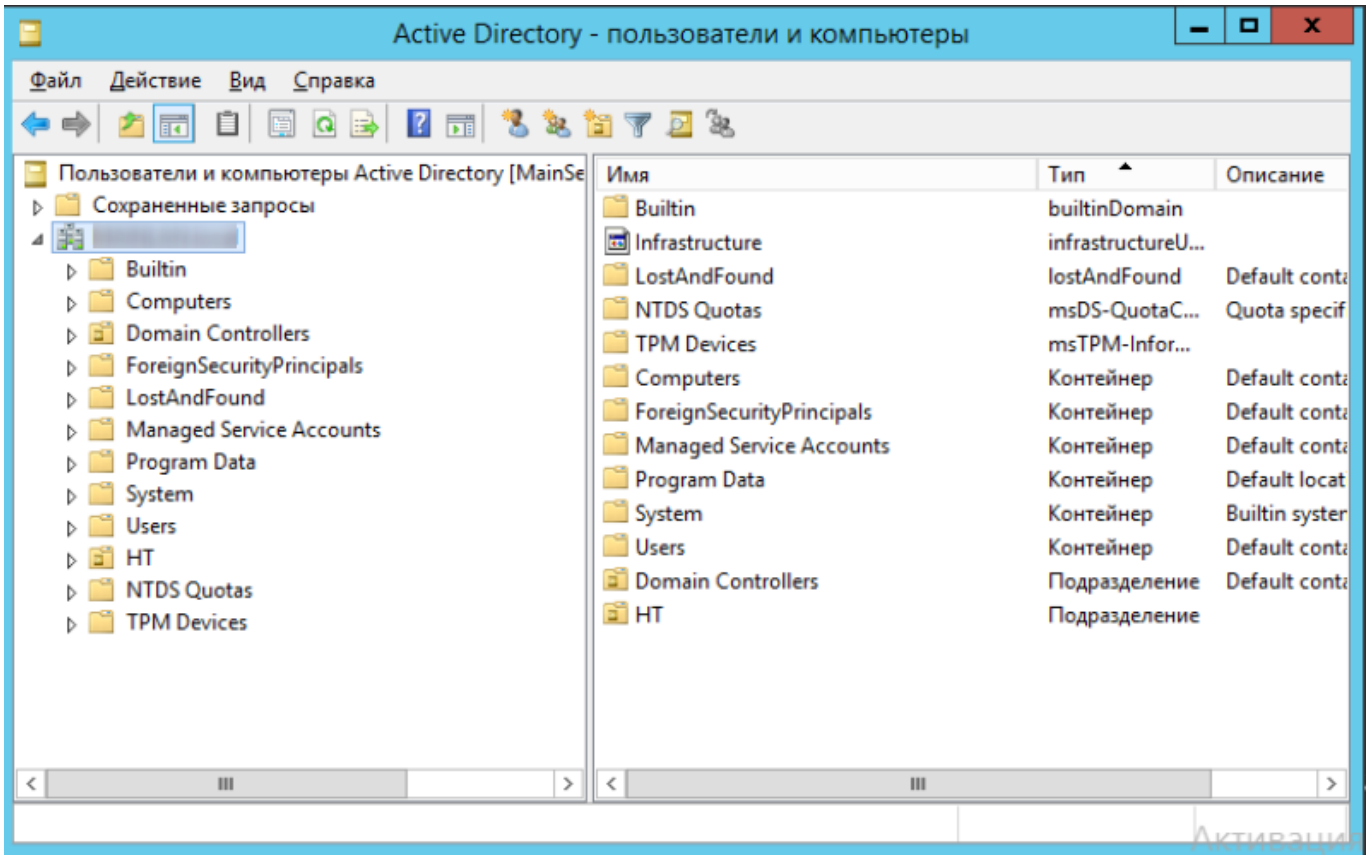
Перезапустите инстанс:

```
sudo dsctl <имя LDAP-сервера> restart
```

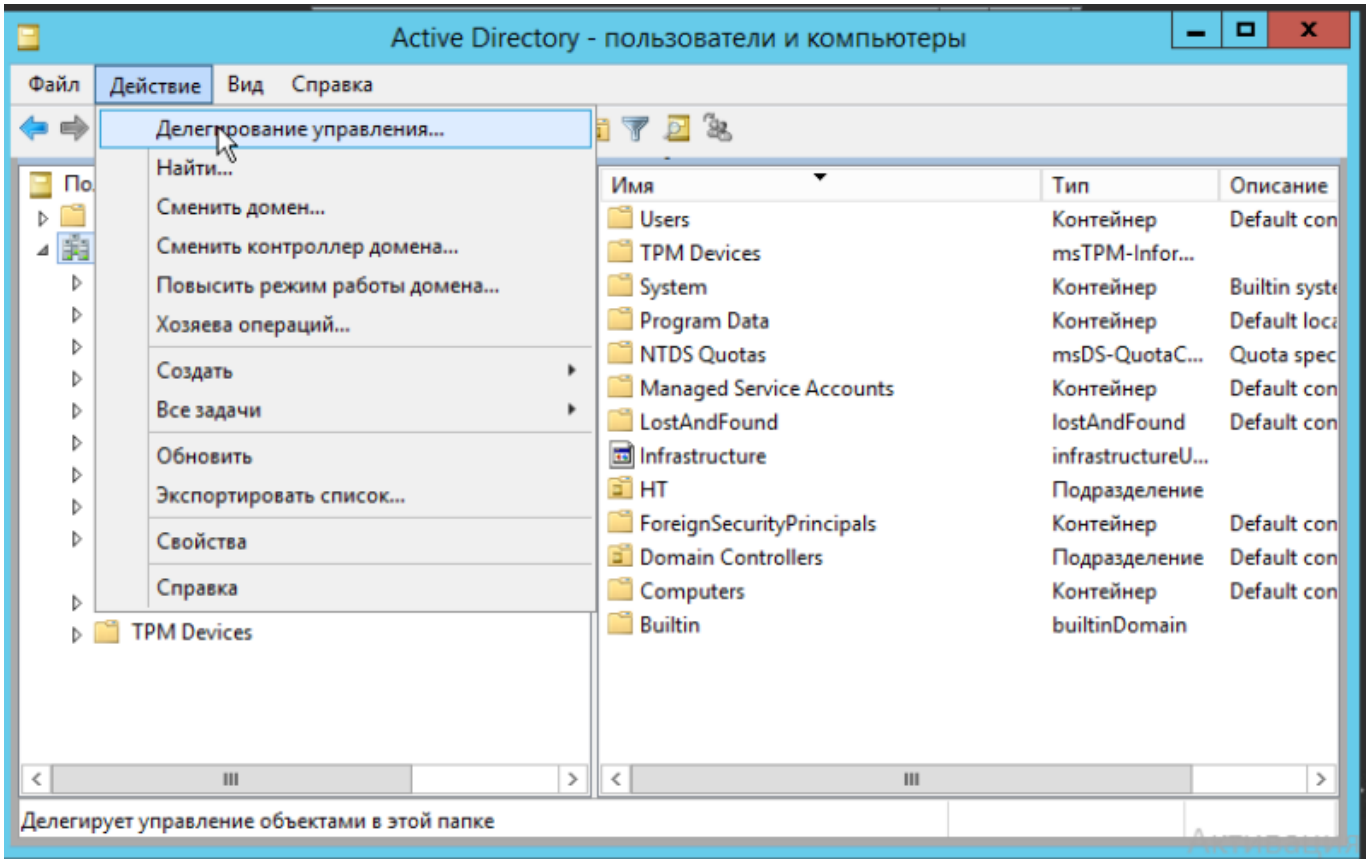
6.8.3. Инструкция по выдаче ограниченных прав на управление паролями пользователей в домене MS AD

Для работы модуля синхронизации учетная запись домена MS AD, под которой устанавливаются отношения синхронизации, должна обладать правами на управление паролями пользователей, которые, по умолчанию, имеются у администраторов домена. В случае невозможности или нежелательности работы модуля синхронизации под учетной записью администратора домена допускается выдача ограниченных прав на учетную запись. Для этого:

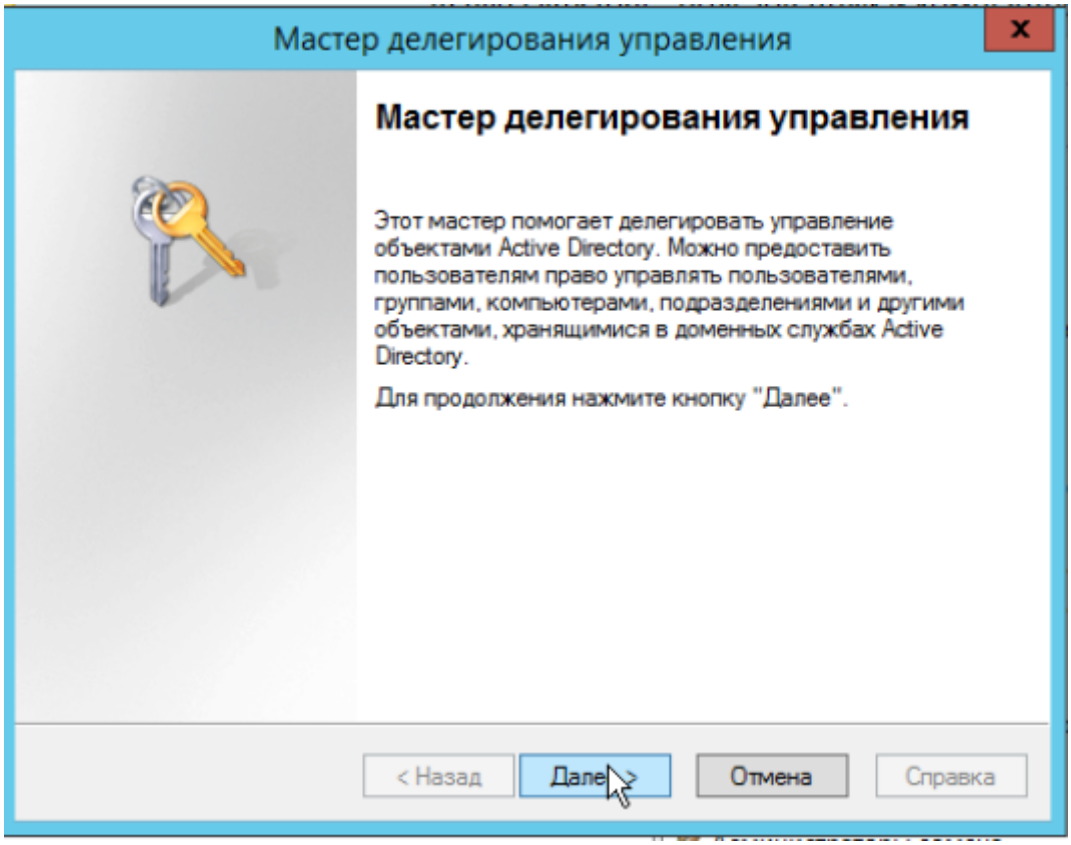
1. На контролере домена MS AD необходимо запустить оснастку “Active Directory - пользователи и компьютеры”

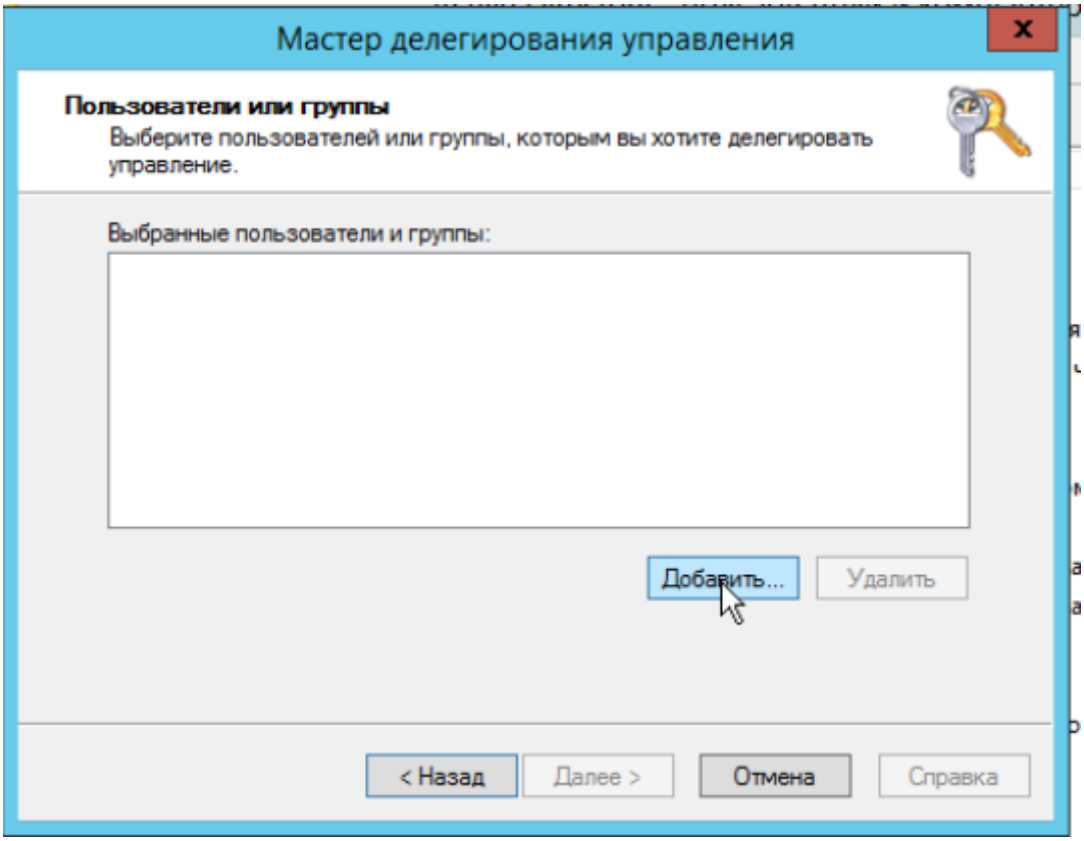


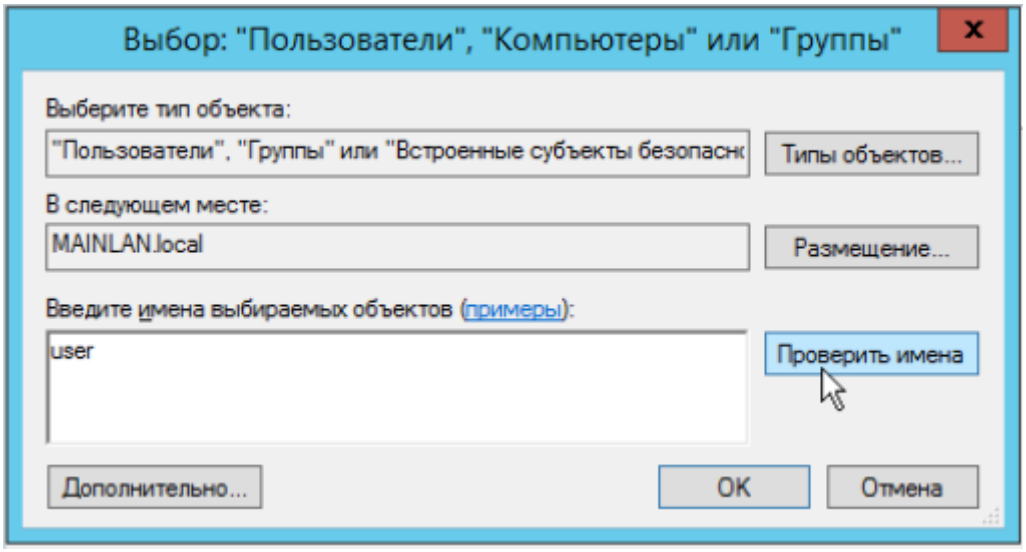
2. Выбрать OU корень домена и открыть “Делегирование управления”

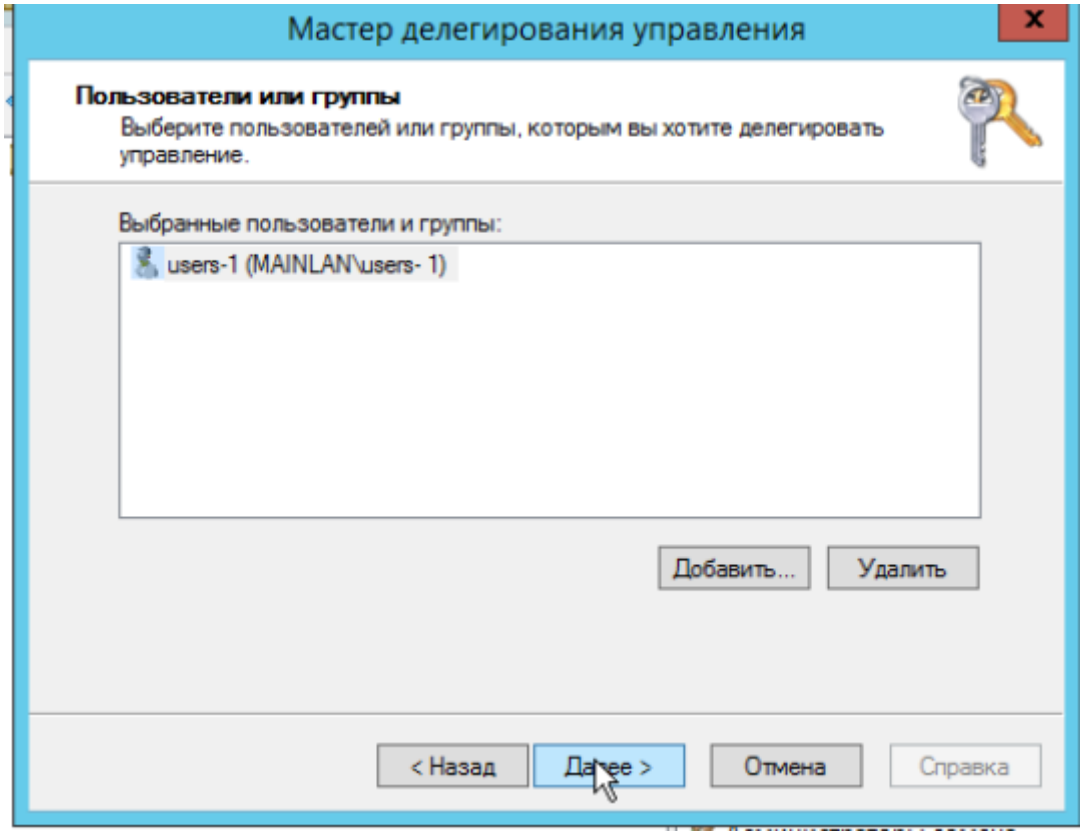


3. В открывшемся окне “Мастер делегирования управления”, после нажатия кнопки **[Далее]** необходимо указать учетную запись пользователя, которой выдаются права

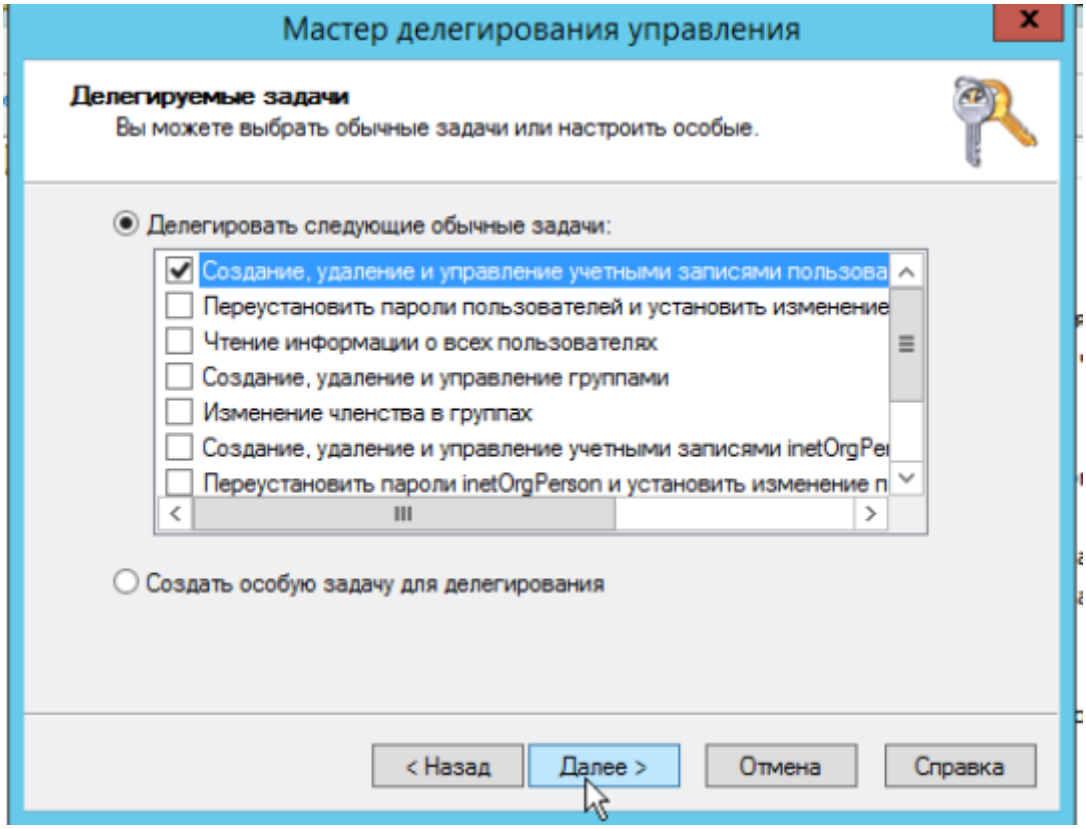



Мастер делегирования управления
Пользователи или группы
Выберите пользователей или группы, которым вы хотите делегировать управление.
Выбранные пользователи и группы:
Добавить...
Удалить
< Назад
Далее >
Отмена
Справка

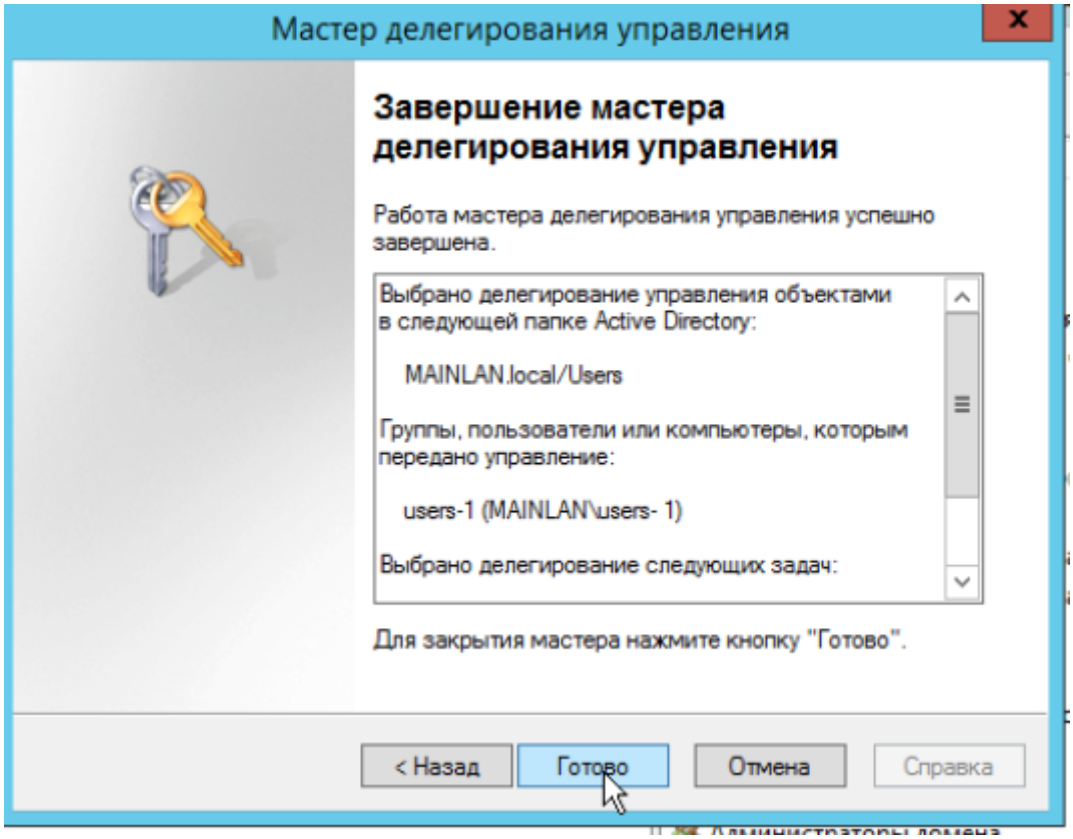

Выбор: "Пользователи", "Компьютеры" или "Группы"
Выберите тип объекта:
"Пользователи", "Группы" или "Встроенные субъекты безопасн
Типы объектов...
В следующем месте:
MAINLAN.local
Размещение...
Введите имена выбираемых объектов (примеры):
user
Проверить имена
Дополнительно...
OK
Отмена



4. Выбрать права для делегирования: Создание, удаление и управление учетными записями пользователей



5. Нажать кнопку [Готово]



Права выданы.

Примечание: В зависимости от версии домена MS AD набор делегируемых задач может отличаться.

6.9. Суммирование политик ПО

6.9.1. Термины и определения

Термин	Определение
SaltStack	Система управления конфигурацией машин и инструмент дистанционного выполнения скриптов, который позволяет администраторам запускать команды на разных машинах.
Salt-master	Центральный демон Salt, которой может отдавать команды слушающим миньонам (Salt-minion).
Salt minion	Сервер, на котором запущен демон Salt minion, который может прослушивать команды от мастера (Salt-master) и выполнять запрошенные задачи.
Standalone minion	Автономно от мастера запущенный клиент на компьютере домена. Выполняет задачи запуска заданий без подключений к мастеру и запуска локальных задач.
Pillar	Хранилище ключ-значение для пользовательских данных, которые будут доступны миньону. Часто используется для хранения и распространения конфиденциальных данных миньонам.
Приоритет политики	Позволяет управлять очередностью применения политик.
Сервер службы каталогов	Сервер, хранящий и передающий важную конфиденциальную информацию, связанную с пользователями, паролями и учетными записями компьютеров.
LDAP-протокол	Протокол прикладного уровня для доступа к службе каталогов.
Фильтр политики ПО	Выборка определенных компьютеров и/или групп компьютеров.
Портал управления	Графический web-интерфейс, предоставляющий привилегированному пользователю единую точку доступа к данным, управления инфраструктурой и объектами домена.
Политика ПО	Набор правил, позволяющих устанавливать, настраивать и обновлять ПО на компьютерах

6.9.2. Назначение политик ПО на подразделение

Для работы с политиками ПО у пользователя должны быть соответствующие права. Для 2.2.0 это ALDPRO - Main Administrator и ALDPRO - IT Specialist. С версии 2.4.0 это может быть набор привилегий (подробнее см. Роли и права доступа). Перед назначением политики ПО на подразделения, подготовьте программное обеспечение (см. Каталог ПО) и создайте политику ПО (см. Создание политики ПО).

Для назначения политик ПО (подробнее см. Подразделения) перейти: **Установка и обновление ПО → Политики ПО → {Имя политики} → Вкладка “Подразделения”**. На вкладке выбрать **+ Добавить подразделение**, на которое будет назначена политика ПО. Или сделать аналогичные действия в карточке подразделения: **Пользователи и компьютеры → Организационная структура → {Имя подразделения} → Вкладка Политики ПО**. На вкладке выбрать **+ Назначить политику ПО**.

Есть 2 способа назначить политику ПО:

- На подразделение. Для этого выберите любое подразделение. В этом случае политика ПО применяется ко всем компьютерам подразделения.
- На корневое подразделение с указанием компьютера или группы компьютеров. Для этого в выпадающем списке выберите корневое подразделение. Вам станут доступны блоки выбора компьютеров и групп компьютеров. В блоке “Все компьютеры” и/или “Все группы” укажите соответствующие компьютеры и/или группы компьютеров для применения политики ПО. Данная функция предусмотрена исключительно для корневого подразделения.

На вкладке **Установка и обновление ПО → Политики ПО → {Имя политики} → Вкладка “Подразделения”** приведен список подразделений, к которым применяется данная политика ПО, с указанием приоритета политики ПО (см. Политики ПО).

6.9.2.1. Версионность

Каждая политика ПО имеет версию, которая автоматически изменяется, если наступило одно из следующих событий:

1. Внесены любые изменения в политику ПО. В этом случае дата изменений - это дата, когда пользователь внес изменения. А автор - это ФИО пользователя, внесшего изменения.

2. Произошло обновление системы. В этом случае дата изменений - это дата обновления системы. А автора изменений - **Системное обновление**.

3. Внесены изменения в ПО, которое есть в составе политики. В этом случае дата изменений - дата внесения изменений в конкретном ПО (изменения в каталоге ПО). А автор - это ФИО пользователя, внесшего изменения в ПО.

6.9.3. Описание принципов суммирования политики ПО

6.9.3.1. Область действия политики ПО

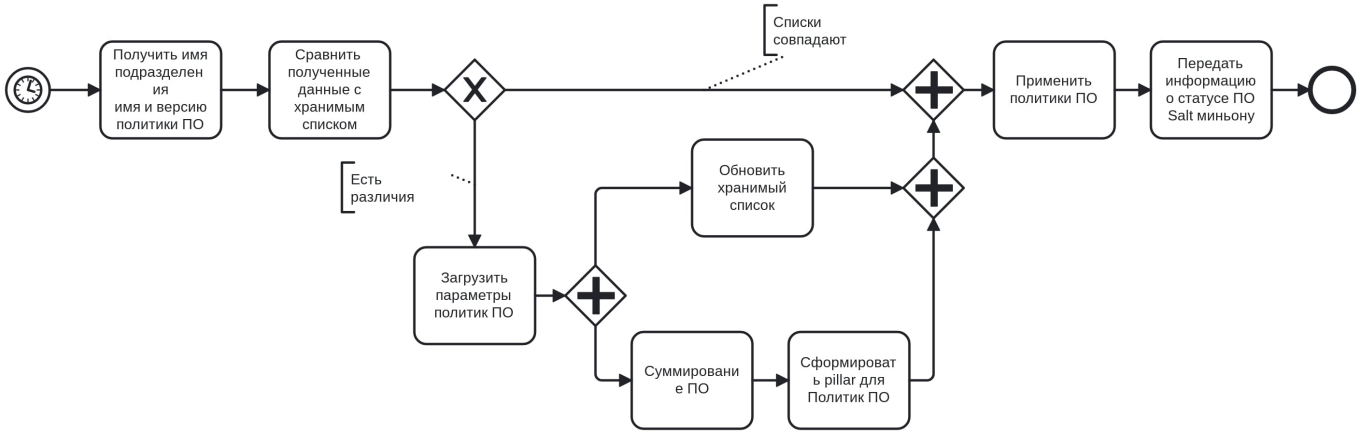
Область действия политики ПО определяется этапом назначения на подразделение. Это могут быть:

- Все компьютеры в рамках выбранного подразделения, но группы компьютеров,

входящие в подразделение, не учитываются. Стоит учесть, что корневое подразделение - главное подразделение домена, в которое входят все компьютеры, поэтому политики ПО, назначенные на корневое подразделение, назначаются всем компьютерам домена.

- Определенные компьютеры и/или группы компьютеров. При назначении на группу компьютеров, политика ПО так же применяется и ко вложенным группам (см. Как назначаются политики ПО).

6.9.3.2. Порядок формирования pillar политик ПО



Для получения данных политик ПО и их применения на компьютере используется pull-модель. Инициатором работы является ALD Pro Salt миньон (standalone minion), установленный на каждом компьютере домена, который по протоколу LDAP к серверу службы каталогов получает актуальные данные о конкретной политике.

- ALD Pro Salt миньон (Standalone миньон) — это рабочие станции или серверы, на котором работает демон-миньон Salt, обладающий широкой функциональностью, работающий автономно от мастера. Используется для запуска заданий в системе без подключения к мастеру.

Формирование конфигураций политик ПО происходит по расписанию или принудительному вызову. По расписанию, применение политик ПО гарантированно происходит каждые 80 минут, но не чаще каждых 30 минут.

При первом запуске задания на формирование pillar Standalone миньон обращается к серверу службы каталогов и получает:

1. имена подразделения компьютера и его родительских подразделений;

2. все настройки и данные назначенных на них политики ПО.

Имена подразделений, идентификаторы политик ПО и их версии сохраняются в файлах `sw-assigments.json, sw-versions.json, sw-settings.json` в директории `/opt/rpta/aldpro-salt/minion`. Настройки всех назначенных политик суммируются и сохраняются в `sw-pillar.json` той же директории. Далее политики ПО применяются.

При последующих запусках, Standalone миньон обращается к серверу службы каталогов с коротким запросом на получение:

1. имена подразделений связанных с компьютером или пользователем;
2. идентификаторы назначенных на них политик ПО;
3. версии этих политик.

Полученные данные миньон сравнивает со списками сформированными ранее `sw-assigments.json, sw-versions.json`. Если изменений нет, pillar остается без изменений, политики применяются. Если данные отличаются, перед применением, Standalone миньон предварительно запрашивает у сервера службы каталогов настройки отличающихся политик ПО, которые используются для формирования нового pillar.

6.9.3.3. Порядок суммирования политик ПО

Суммирование политик происходит согласно иерархической структуре домена, последовательно, начиная с корневого подразделения, заканчивая подразделением в котором состоит компьютер. Условно, каждое подразделение можно обозначить уровнем:

- 1 уровень - корневое подразделение
-
- N уровень - родительское подразделение компьютера.

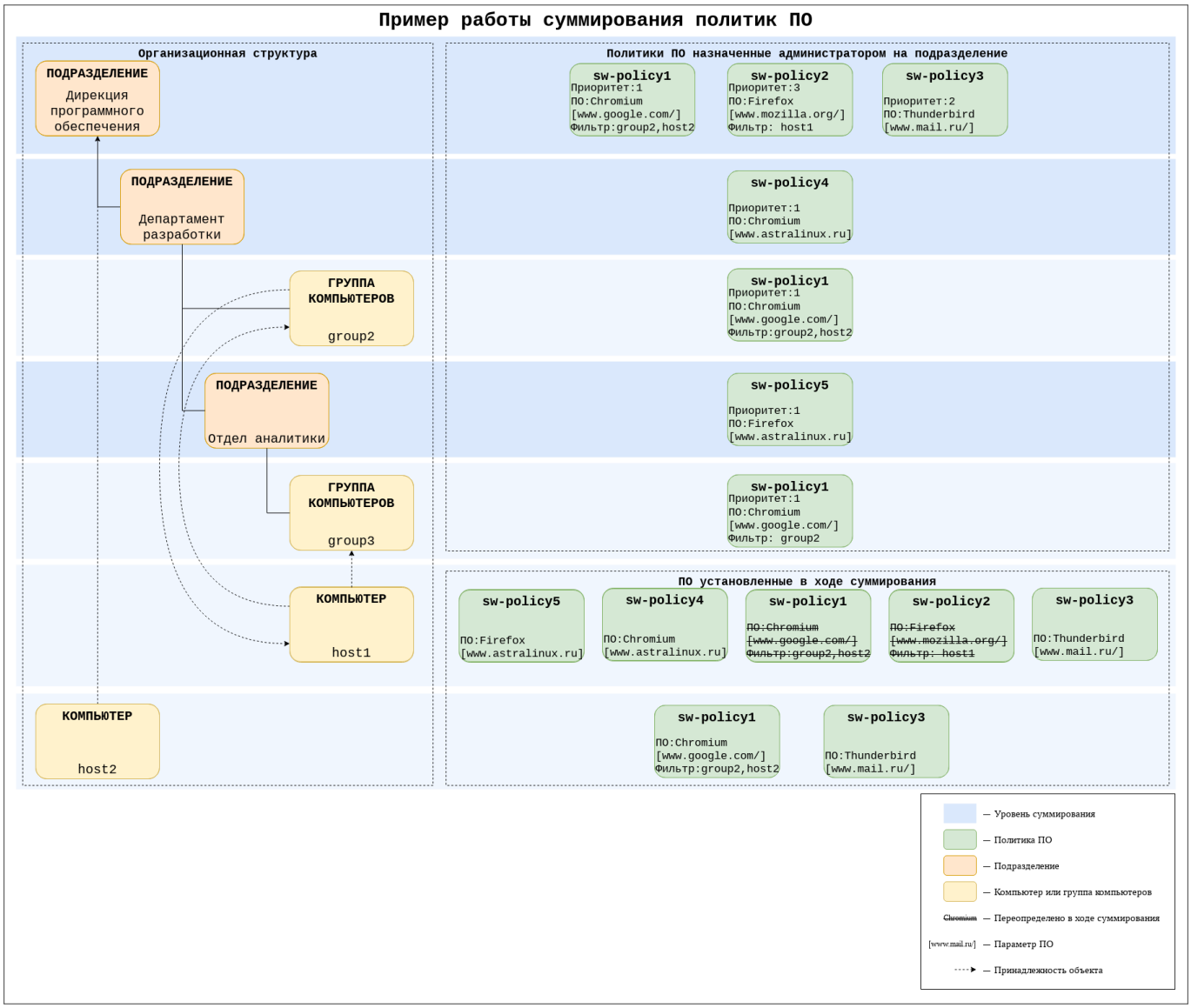
Политики ПО 1 уровня, которые применяются к компьютеру, суммируются с политиками ПО последующего уровня. Если в политиках присутствует одинаковое ПО с разными параметрами, то применяются параметры политики с приоритетом выше (наивысший приоритет 1). Если же приоритеты одинаковые, такое возможно если политики наследованы с разных подразделений, то для компьютера применяются параметры той политики, что является для данного компьютера ближайшей по структуре, поэтому одинаковые приоритеты для компьютера переопределяются, согласно иерархической структуре домена.

При возникновении конфликтов среди одинакового программного обеспечения после назначения политики ПО, то программное обеспечение, что по итогу не применилось в ходе суммирования, в карточке компьютера отображается со статусом “Переопределено”. Подробнее о статусах в разделе **Просмотр результатов применения политики**.

После суммирования, на компьютере формируется конечный pillar (см. Порядок формирования pillar политик ПО), который и задаёт основную конфигурацию устанавливаемого программного обеспечения (см. Просмотр результирующего pillar).

6.9.3.4. Схема применения ПО

Пример работы нового функционала суммирования ПО выглядит следующим образом:



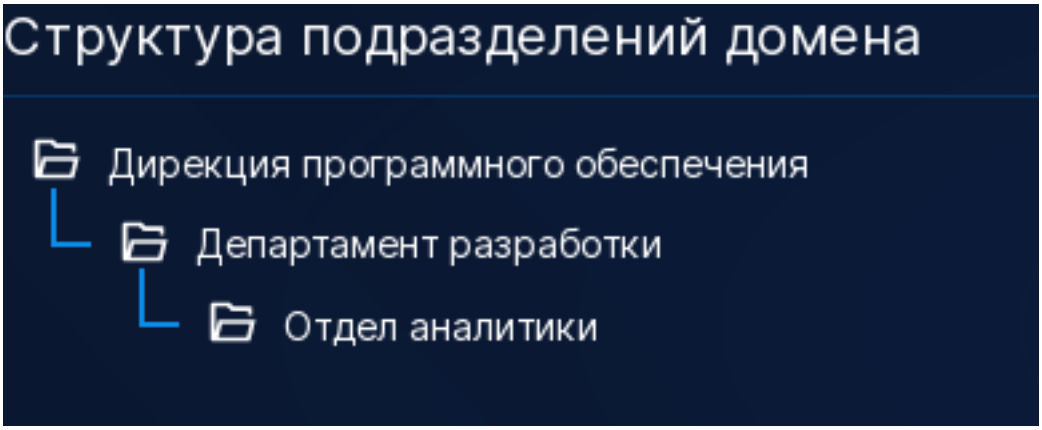
Реализация данной схемы так же представлена в портале ALD Pro (см. Пример

суммирования политик ПО через портал управления).

Пример суммирования политик ПО через портал управления

Для того, чтобы показать работу политик ПО в портале управления, приведем пример из схемы (см. Схема применения ПО).

1. Создана следующая структура домена:

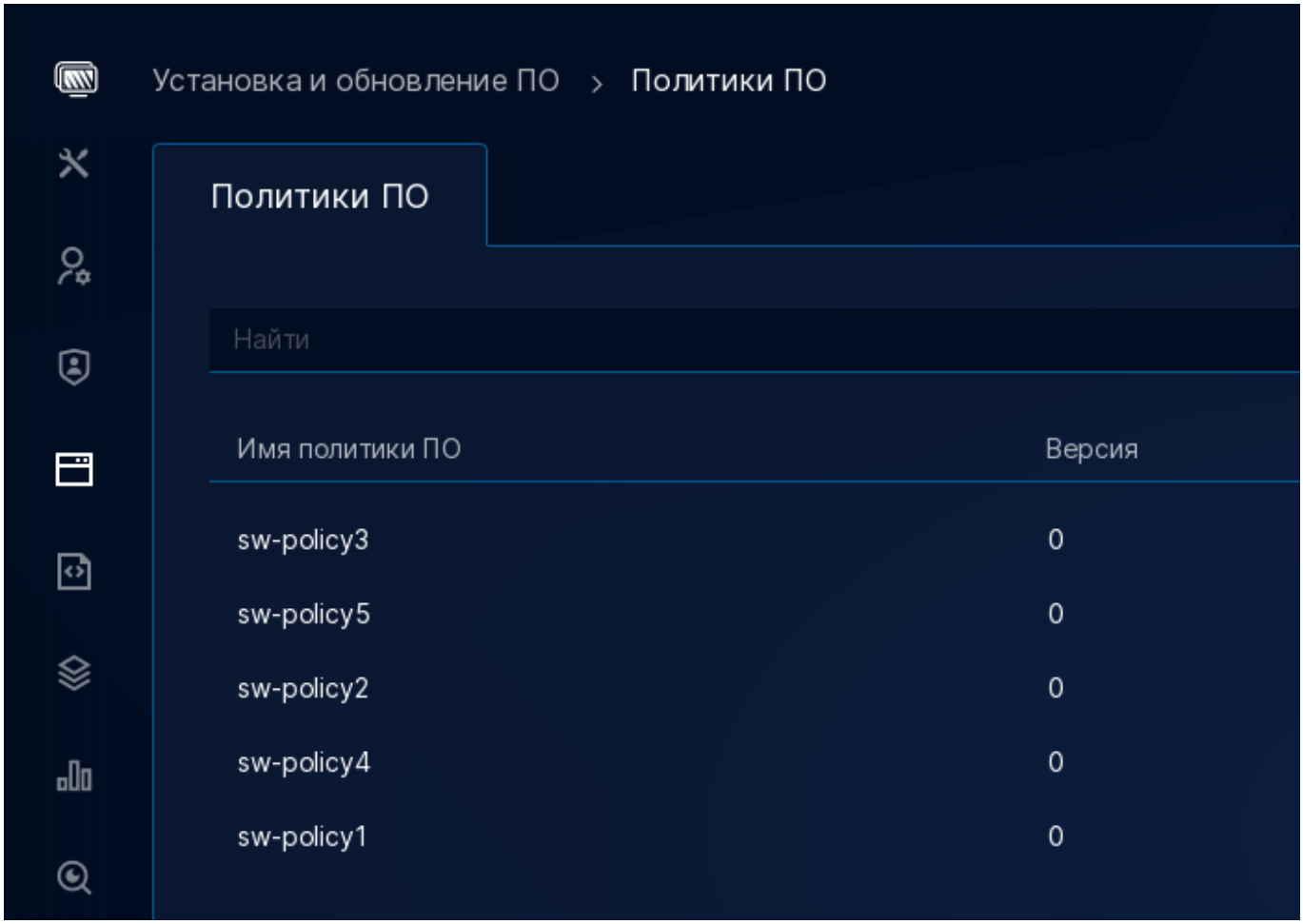


2. В системе присутствуют компьютеры “host1” и “host2”.

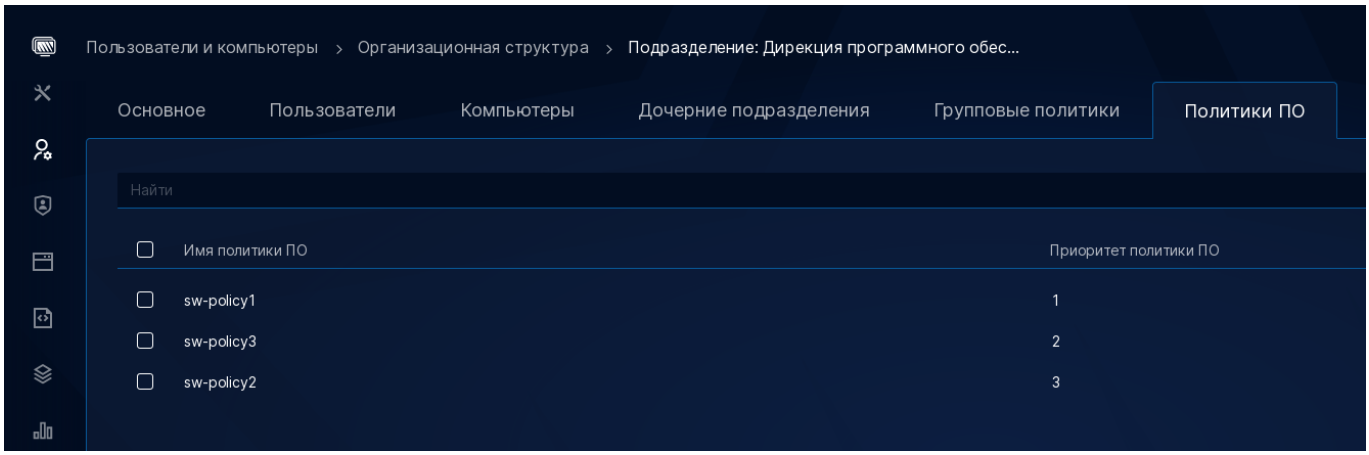
3. Созданы группы компьютеров “group3”, содержащая компьютер “host1”, и “group 2”. Группы взаимно вложены. “group2” добавлена в “Департамент разработки”, “group3” в “Отдел аналитики”.

4. В подразделение “Отдел аналитики” добавлен компьютер “host1”, а в “Дирекцию программного обеспечения” компьютер “host2”.

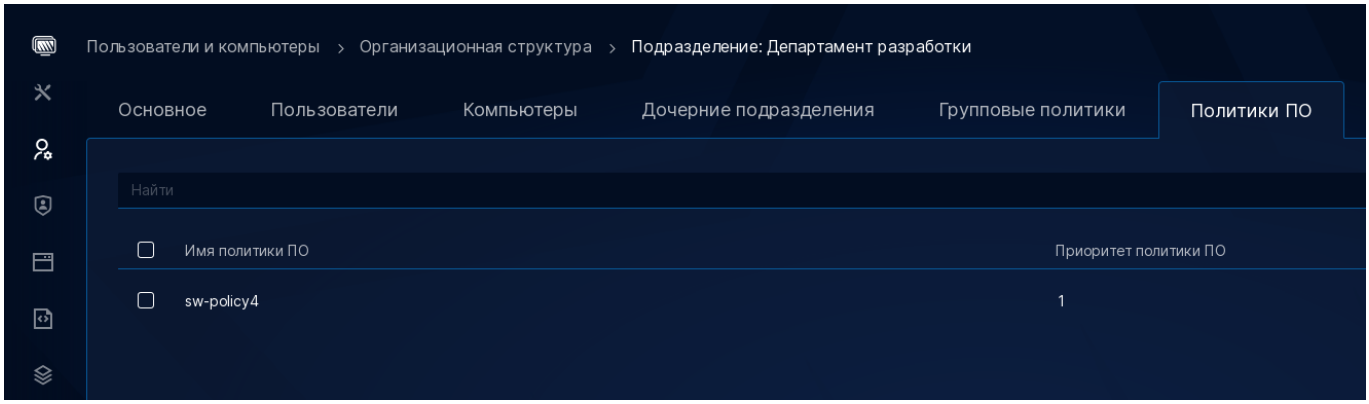
5. Созданы политики ПО, содержащие идентичное схеме программное обеспечение и параметры:



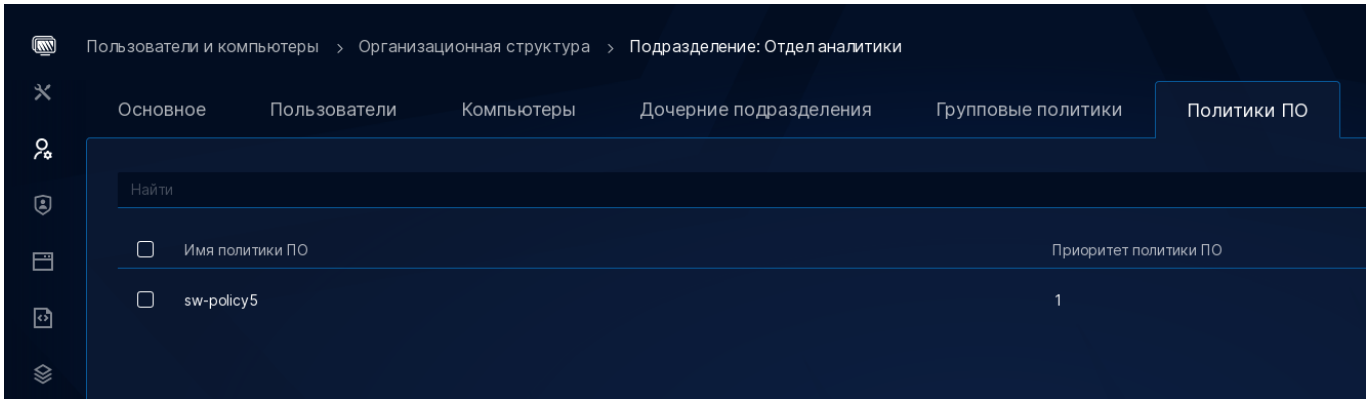
6. Политики ПО назначены на подразделения согласно схеме. Перейти: Пользователи и компьютеры → Организационная структура → Дирекция программного обеспечения. Назначенные политики ПО:



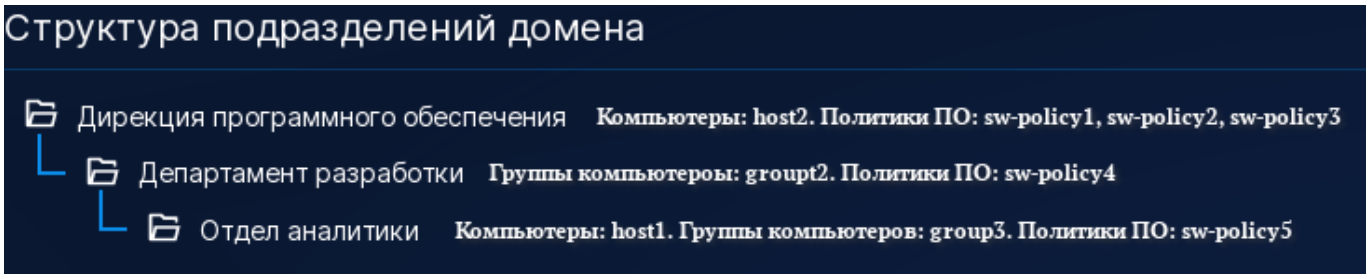
Перейти: Пользователи и компьютеры → Организационная структура → Департамент разработки. Назначенные политики ПО:



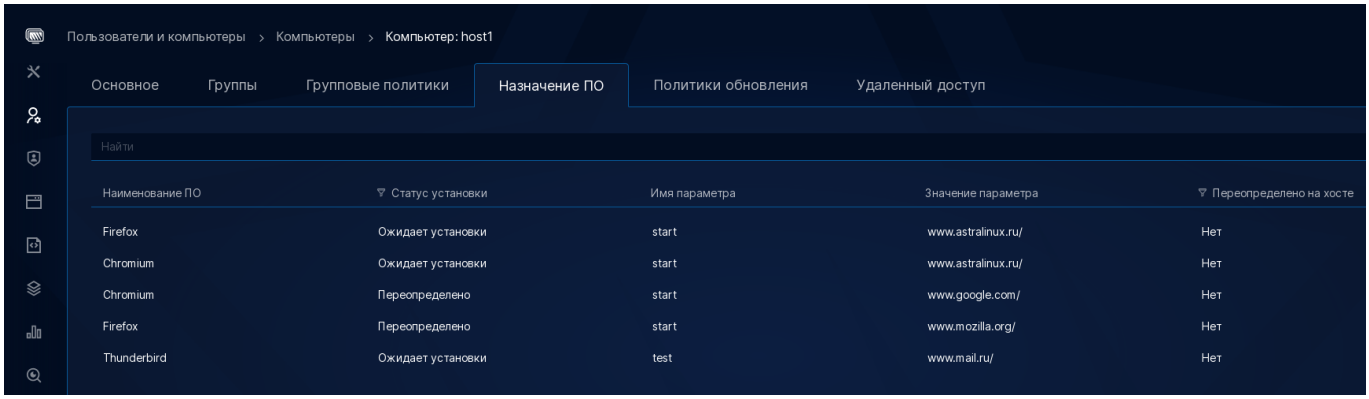
Перейти: Пользователи и компьютеры → Организационная структура → Отдел аналитики. Назначенные политики ПО:



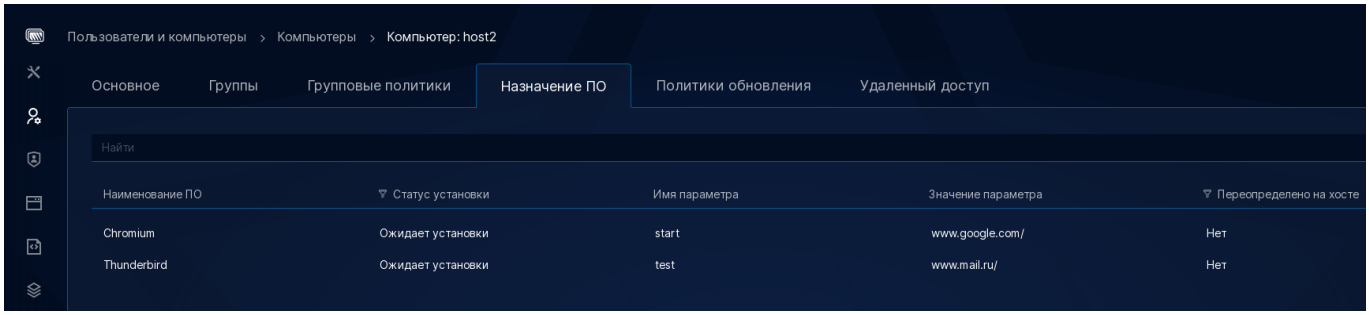
7. Не учитывая вложенность групп и точечно назначенные политики ПО, структура будет выглядеть следующим образом:



8. Перейти: **Пользователи и компьютеры → Компьютеры → host1 → Вкладка “Назначение ПО”**. Как можно увидеть, присутствует одинаковое ПО. Из него в статусе **Ожидает установки**, то программное обеспечение, что было назначено соответствующей политикой ПО на родительское подразделение компьютера, остальное отмечается как **Переопределено**:



9. Перейти: **Пользователи и компьютеры → Компьютеры → host2 → Вкладка “Назначение ПО**. Назначенное ПО определено следующим образом:



6.9.3.5. Просмотр результирующего pillar

Для просмотра назначенного программного обеспечения и его параметров на компьютере домена проверьте сформированный pillar командой:

```
aldpro-salt-call pillar.get aldpro-software
```

Вывод команды должен соответствует назначенным ПО с итоговыми параметрами (см. Просмотр результатов применения политики).

Пользователь также может посмотреть pillar с результатами суммирования в файле `/srv/aldpro-salt/roots/pillar/aldpro_po.sls`:

```
sudo cat /srv/aldpro-salt/roots/pillar/aldpro_po.sls
```

6.9.4. Отладка политик ПО

6.9.4.1. Применение политик по умолчанию

Применение политик ПО запускается при старте standalone minion. Пользователю доступна проверка текущего статуса работы standalone миньона:

```
Заменить на systemctl status aldpro-salt-minion.service
```

После старта standalone minion, начинается таймер применения Политики ПО. Просмотр таймера:

```
aldpro-salt-call schedule.show_next_fire_time build_and_run_swp
```

Система показывает примерное время, равное постоянному таймеру. Программное обеспечение установится в течение ещё 5-50 минут от указанного.

6.9.4.2. Форсированное применение ПО

Пользователь может выполнить принудительное обновление pillar и применение политик командой:

```
aldpro-salt-call state.apply gpupdate.swp pillar='{"verbose": True, "force":
↪True}'
```

Флаг pillar="{"verbose": True}" нужен для получения логов выполнения заданий применения. Флаг pillar="{"force": True}" используется для принудительного удаление сохранённых ранее pillar. Использовать флаг force: True рекомендуется не чаще одного раза в 10 минут, поскольку это приводит к повышенной нагрузке на контроллер домена.

Команда обновления pillar в обычном порядке и последующее выполнение политики:

```
aldpro-salt-call state.apply gpupdate.build
```

Ранее из интерфейса была возможность принудительно установить ПО на доменном компьютере. Теперь этот функционал из портала управления недоступен. Для принудительной установки воспользуетесь командой на доменном компьютере:

```
aldpro-salt-call state.apply policies.sw-policies
```

6.9.4.3. Редактирование времени выполнения заданий

Таймер заданий для групповых политик, политик ПО и правил аудита состоит из двух частей:

- постоянная часть таймера в 25 минут. Минимальное время между заданиями 30 минут.
- случайный разброс от 5 до 50 минут. Добавляется к постоянному таймеру, для снижения пиковой нагрузки на контроллер домена.

Таким образом, гарантированное время выполнение каждого задания 80 минут.

На доменном компьютере пользователь может установить собственные значения таймера в расписании, для внесения изменений нужно обладать sudo правами. В файле `/srv/aldpro-salt/config/minion.d/standalone_scheduler.conf` измените значения атрибутов `build_and_run_swp`:

- minutes - постоянный таймер;
- splay: * start - минимальное значение случайного таймера; * end - максимальное значение случайного таймера.

Выставлять значения таймера менее 10 минут нежелательно, это может привести к повышенной нагрузке и на контроллер домена и на компьютер.

Далее в файле `/srv/aldpro-salt/roots/_modules/utils.py` на доменном компьютере внесите изменения в строчке _25_MINUTES = dt.timedelta(minutes=25), поменяв значение 25. После внесения изменений перезапустите standalone minion:

```
systemctl restart aldpro-salt-minion.service
```

6.9.4.4. Просмотр результатов применения политики

Кроме моделирования результатов применения политики ПО непосредственно на компьютере, результат применения возможно увидеть в портале ALD Pro. Для просмотра назначенных политик ПО на компьютер или подразделение следует:

1. Перейти: **Пользователи и компьютеры → Компьютеры → {Имя компьютера} → Вкладка “Назначение ПО”**

Во вкладке отображается ПО, которое назначено на компьютер политиками ПО (см. Как назначаются политики ПО). Вкладка содержит следующие разделы:

Наименование ПО	Статус установки	Имя параметра	Значение параметра	Переопределено на хосте
Наименование ПО, которое содержится в назначенной на компьютер политике ПО. Задаётся в каталоге ПО, для просмотра перейти: Установка и обновление ПО → Каталог ПО.	Статус установки данного ПО на компьютере. Предусмотрено 4 значения: Успешно - Если ПО установилось успешно; Ошибка установки - Если при установке ПО возникли ошибки; Переопределено - Если есть аналогичное ПО из другой политики ниже приоритетом; Ожидает установки - ПО назначено, но не установлено.	ПО задаётся с определенными параметрами. Например, для браузеров параметр это часто «Стартовая страница», который задаёт первую страницу по умолчанию при открытии браузера.	Так как ПО задаётся с определенными параметрами, то для них определяются свои значения. В случае, например, параметра «Стартовая страница», его значение может быть «www.aldpro.ru».	Применяется для случаев изменения значения параметра. Если для данного ПО по итогу изменился параметр, «Переопределено на хосте» будет отображаться как «Да».

2. Перейти: **Пользователи и компьютеры → Подразделения → {Имя подразделения} → Политики ПО**. Во вкладке отображаются политики ПО, назначенные на подразделение. Вкладка содержит следующие разделы:

Имя политики	Приоритет политики
Имя политики ПО. Задаётся при создании (см. Создание политики ПО).	Приоритет назначенной политики. В рамках подразделения приоритет уникален. Задаётся при создании политики ПО.

6.10. Настройка синхронизации пользователей из ALD Pro в Keycloak

Основная настройка Keycloak подробно описана на официальном сайте Keycloak. В данной инструкции описано создание Федерации пользователей (User federation) с пользователями ALD Pro.

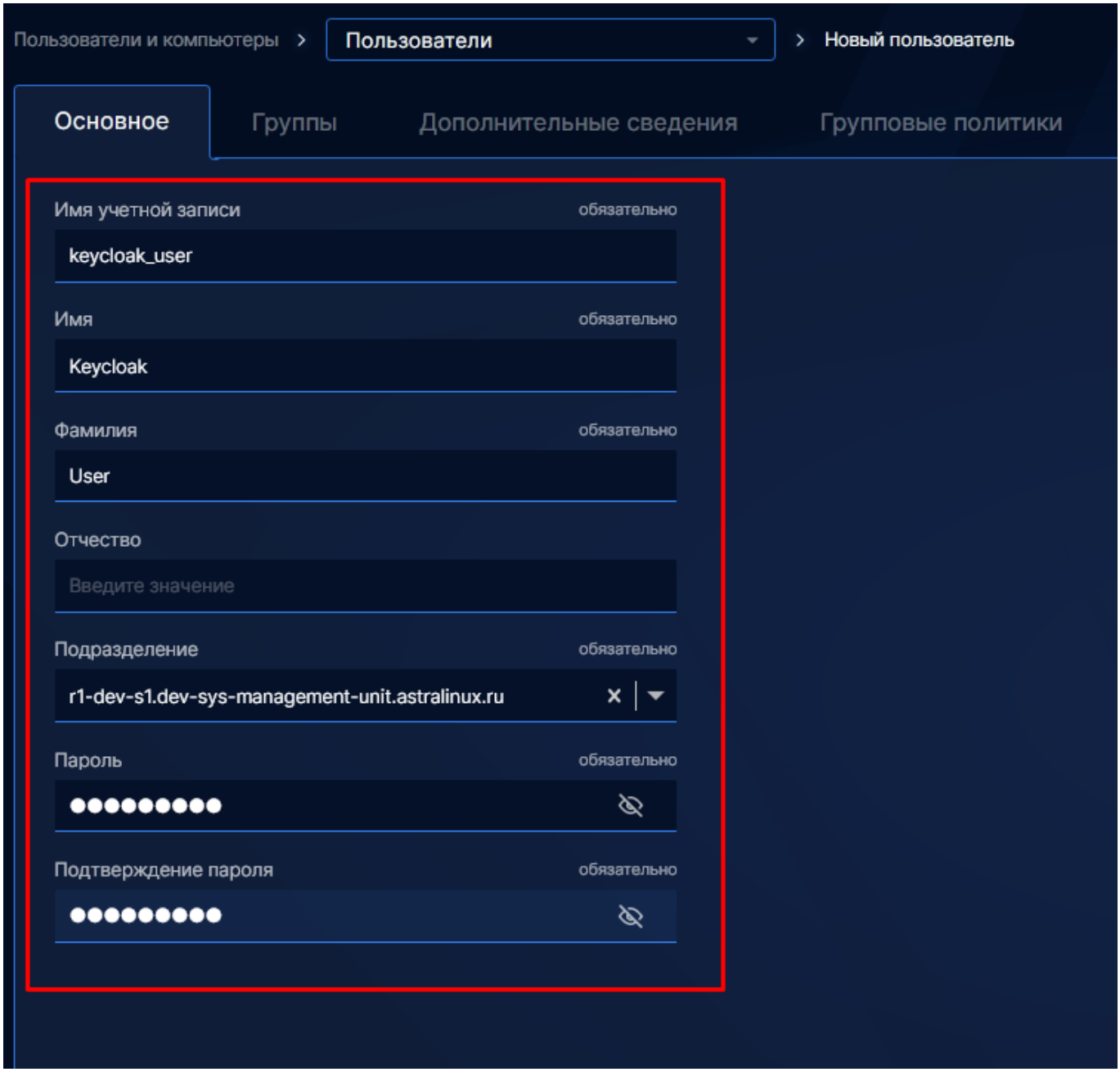
6.10.1. Термины и определения

Термин	Сокращение	Определение
Федерация пользователей		Синхронизация пользователей с серверов ALD Pro.
Single Sign-On	SSO	Метод аутентификации, который позволяет пользователям безопасно аутентифицироваться сразу в нескольких приложениях и сайтах, используя один набор учетных данных.

6.10.2. Предварительная настройка ALD Pro

6.10.2.1. Создание пользователя

В ALD Pro необходимо создать пользователя (рисунок 1), через которого будет осуществляться синхронизация пользователей, групп, подразделений. В целях безопасности пользователь не должен обладать правами администратора. Для синхронизации пользователей достаточно прав обычного пользователя.



6.10.2.2. Активация пользователя

Перед дальнейшей работы, следует авторизоваться под созданным пользователем и сменить ему пароль.

6.10.3. Краткая инструкция настройки Keycloak

Наименование	Значение
Hostname	astralinux.ru
URI/Port	ldap://astralinux.ru.ru:389 (*STARTTLS*) **Note**: Plaintext is not allowed.
	ldaps://astralinux.ru:636
BaseDN	uid=*ALD_user*,cn=users,cn=*accounts*,dc=astralinux,dc=ru
UserDN	cn=users,cn=*accounts*,dc=astralinux,dc=ru
GroupDN	ou=users,o=*YOUR_ORG_ID*,dc=jumpcloud,dc=com
Username LDAP attribute, RDN LDAP attribute	uid
U**UID LDAP attribute**	ipaUniqueId
Search scope	One level

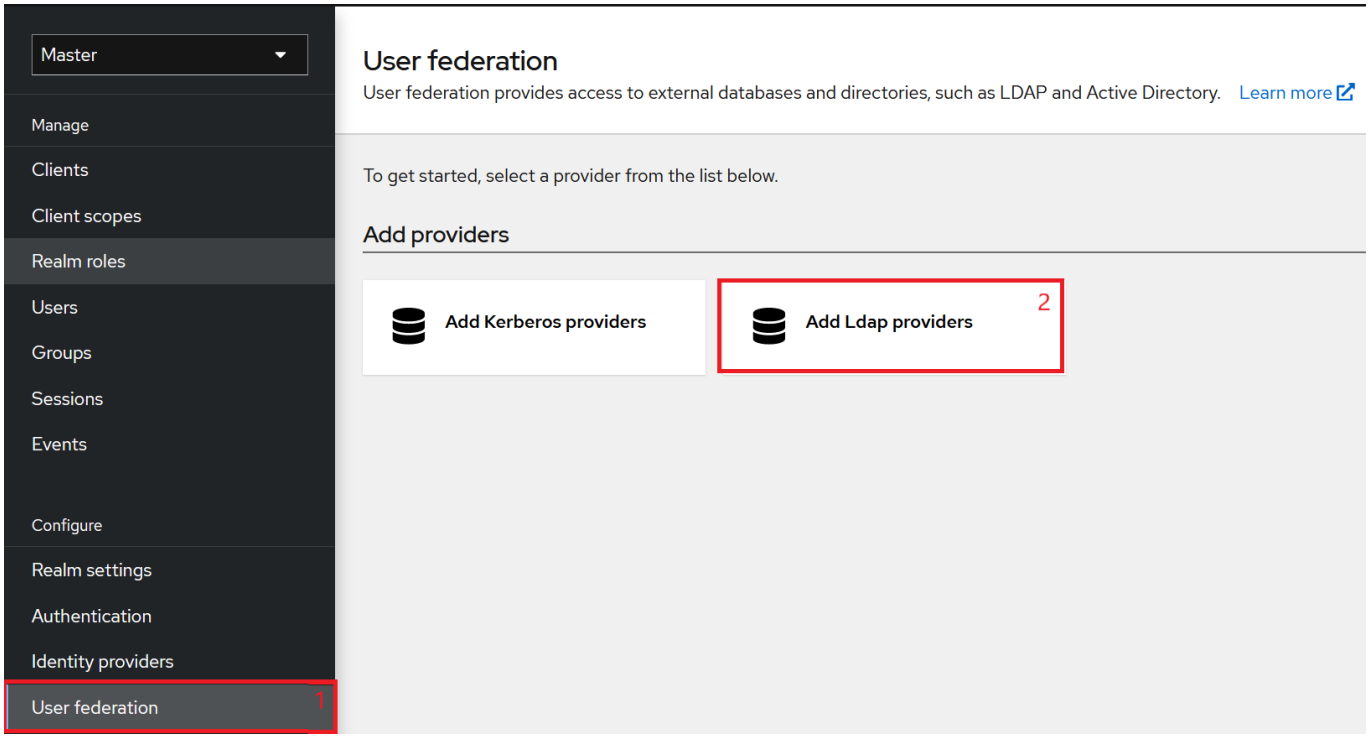
6.10.4. Подробная инструкция настройки Keycloak

Приведенная ниже инструкция актуальна для Keycloak Version 23.0.3

6.10.4.1. Создание User Federation с ALD Pro

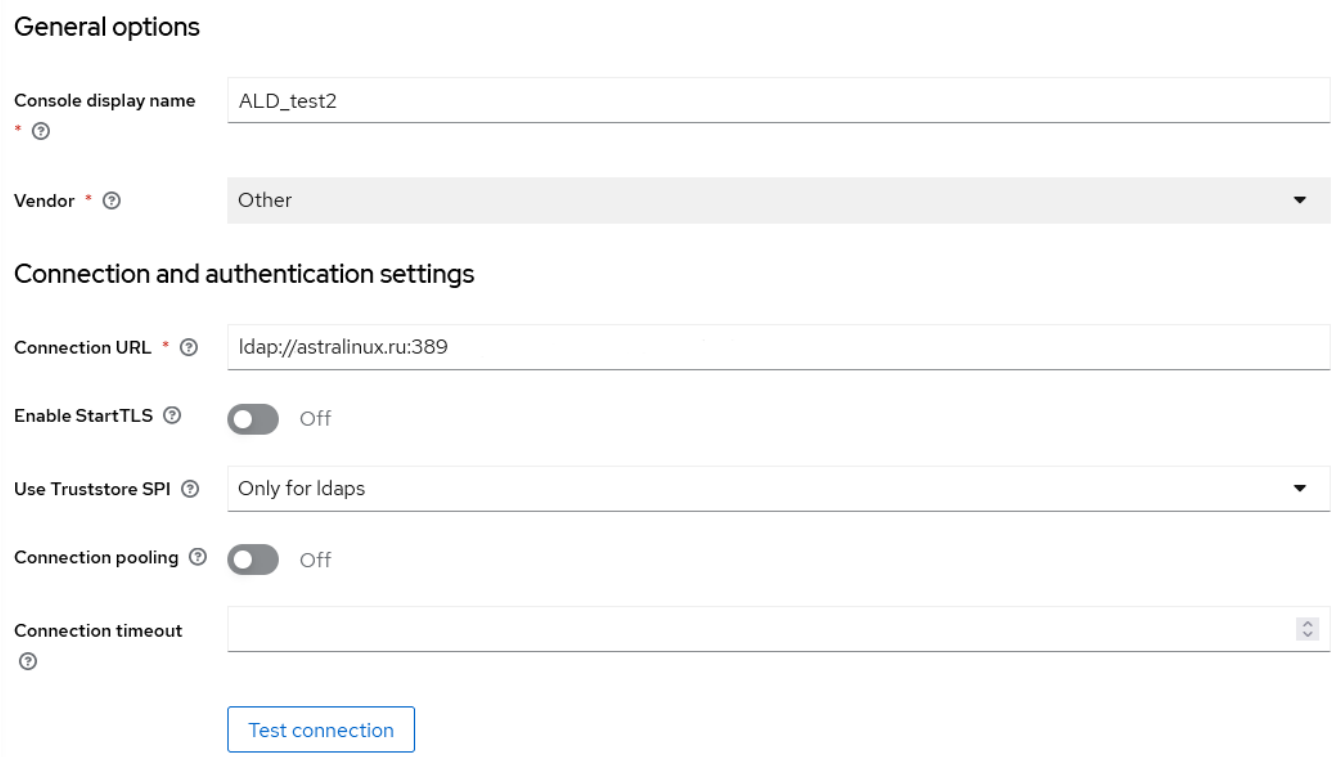
Создание User Federation

1. Выполнен вход в Keycloak с правами администратора и создан рабочий Realm.
2. На вкладке “User federation” выбран “Add Ldap providers”, во всплывающем окне выбран “LDAP” (рисунок 2).



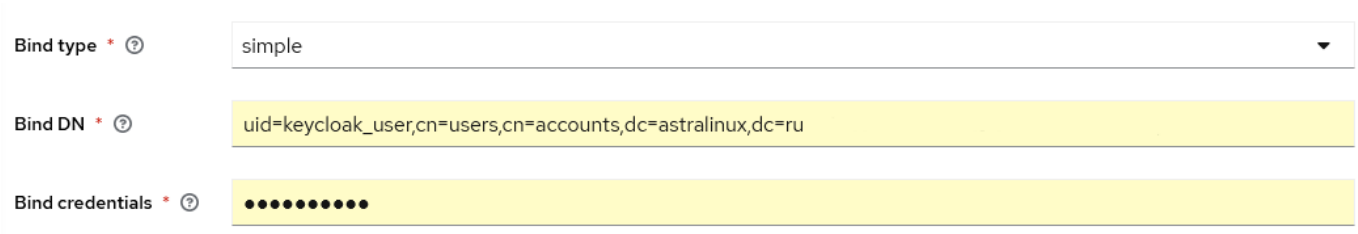
Заполнение блока General options

3. В открывшемся окне заполнены обязательные поля (рисунок 3).
4. В качестве вендора выбран “Other” или “Red Hat Directory Server”. В дальнейшем вендора для созданной федерации пользователя изменить нельзя.
5. Connection URL: URL-адрес подключения ALD Pro в формате: ldap://astralinux.ru:389
6. Остальные настройки остаются на усмотрение.



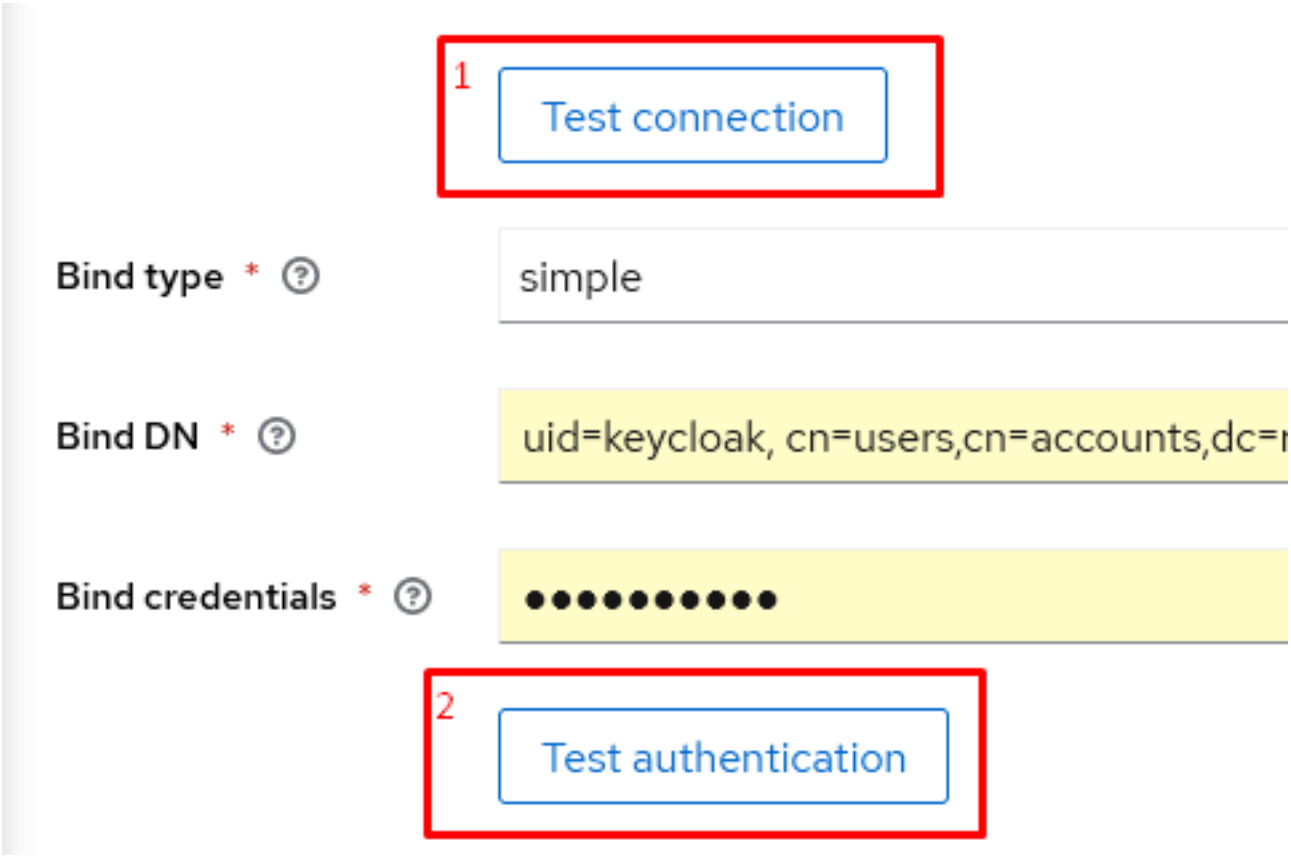
Заполнение блока Connection and authentication settings

7. Тип связи (Bind type) выбран простой (simple).
8. В “Bind DN” указан путь навигации к созданному ранее пользователю ALD Pro. DN записывается слева направо.
9. В “Bind credentials” указан пароль от привязанного пользователя ALD Pro.

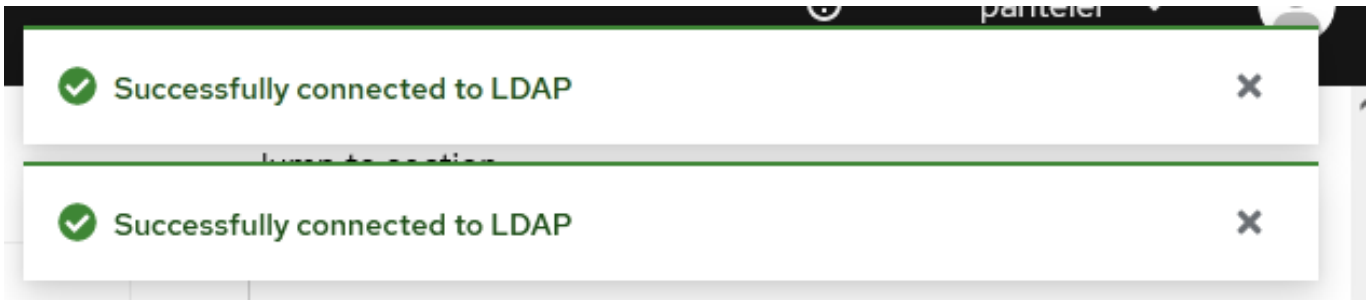


Тест соединения и аутентификации

На этом этапе можно проверить успешность заполнения форм, нажав последовательно на “Test connection” и “Test authentication” (рисунок 5).



При успешном соединении и аутентификации пользователей появится сообщения об успешном завершении операции (рисунок 6).



Заполнение блока LDAP searching and updating

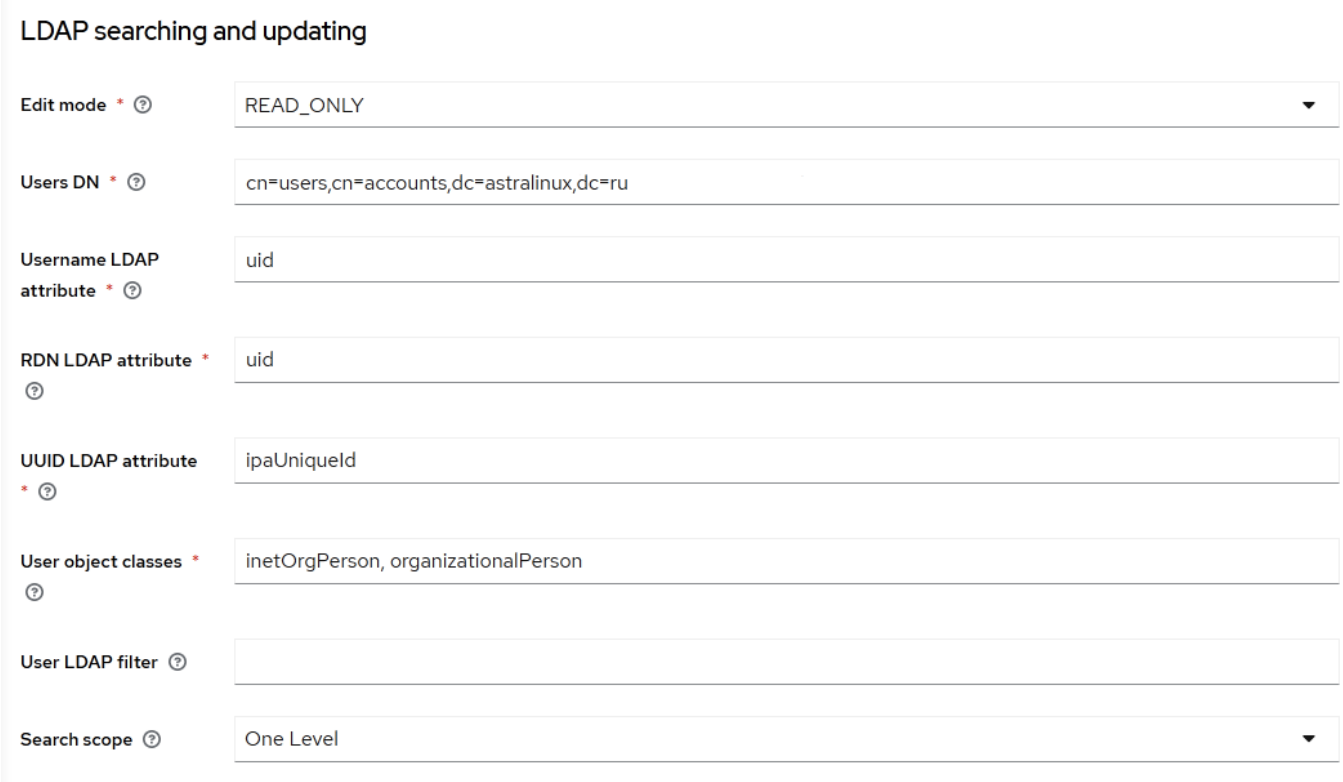
10. Режим редактирования (Edit mode) выбран “READ_ONLY”.
11. Указан путь навигации к пользователям (User DN) сервера ALD Pro.
12. Атрибуты “Username LDAP” и “RDN LDAP” - “uid”.
13. “UUID LDAP” - “ipaUniqueId”.
14. Поле “User object classes” можно оставить по умолчанию.

15. В целях безопасности каталога ALD Pro, по желанию, можно указать фильтры синхронизации для пользователей.

16. Область поиска (Search scope) для ALD Pro - “One Level”

17. Параметры тайм-аут чтения (Read timeout) и пагинация (Pagination) - на усмотрение.

Пример заполнения на рисунке 7.

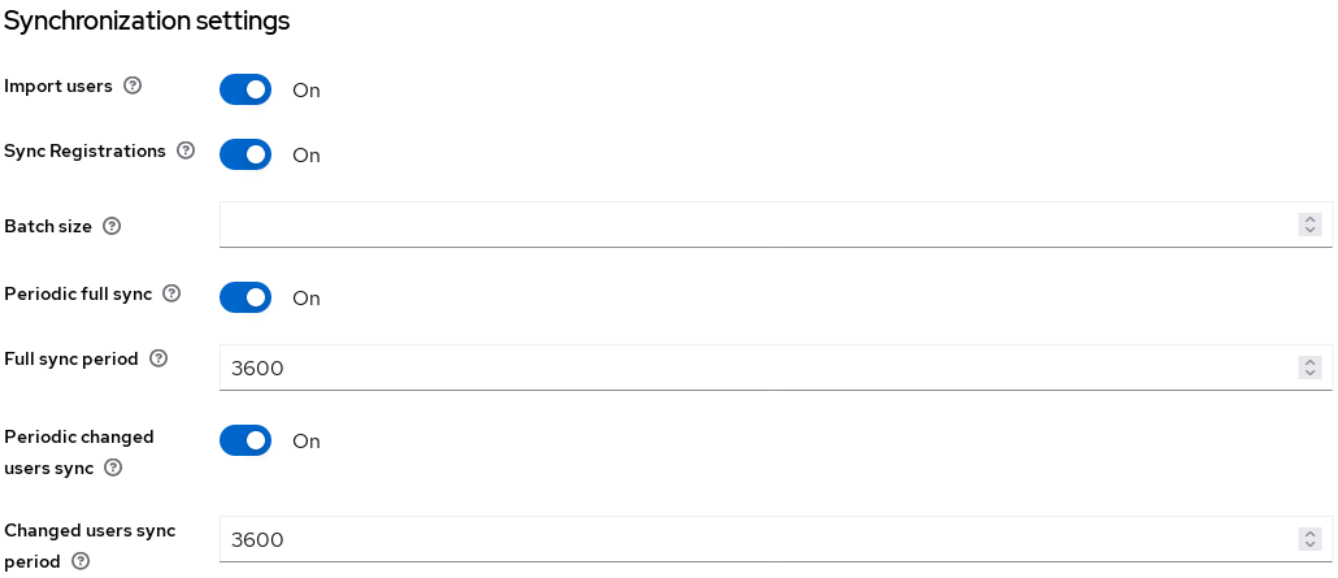


Заполнение блока Synchronization settings

18. Для настройки импорта пользователей слайдер Import users находится в положение ON.

19. Если пользователей много, рекомендуется при их импорте разбивать их на партии с помощью настройки размера партии (Batch size).

20. Для полной и периодической синхронизации пользователей рекомендуется настроить время синхронизации с помощью параметров (Periodic full sync, Periodic changed users syn). Значения указываются в секундах.

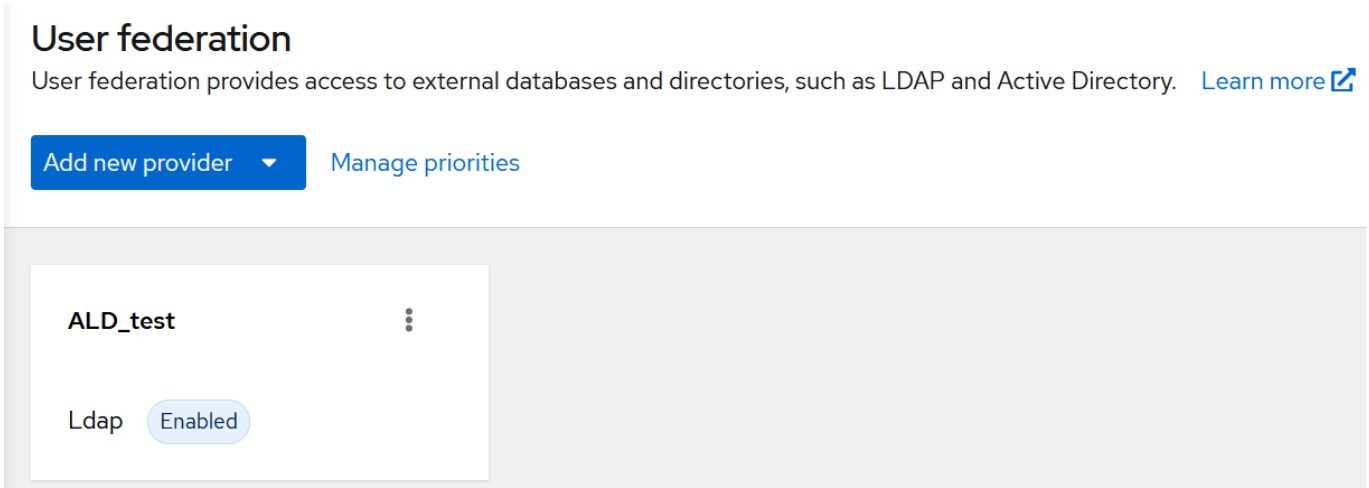


Заполнение блоков Kerberos integration, Cache settings, Advanced settings

Остальные параметры настраиваются индивидуально и позволяют гибко управлять синхронизацией и безопасностью единого входа с помощью федерации пользователей.

Проверка синхронизации пользователей

21. При завершении настроек сохранить федерацию пользователей. После сохранения федерация появится на вкладке “User federation” (рисунок 9).



22. Выполнить переход на вкладку пользователи, здесь находятся все пользователи Realm, включая пользователей федерации. Иногда в настройках Keycloak для

оптимизации работы на вкладке пользователи не отображаются пользователи федерации. Для того чтобы увидеть всех пользователей нужно в поиске ввести * и выполнить поиск.

6.10.4.2. Настройка соответствия атрибутов

Атрибуты пользователей keycloak не полностью соответствуют атрибутам ALD Pro. В таблице приведены основные соответствия атрибутов ALD Pro к Keycloak.

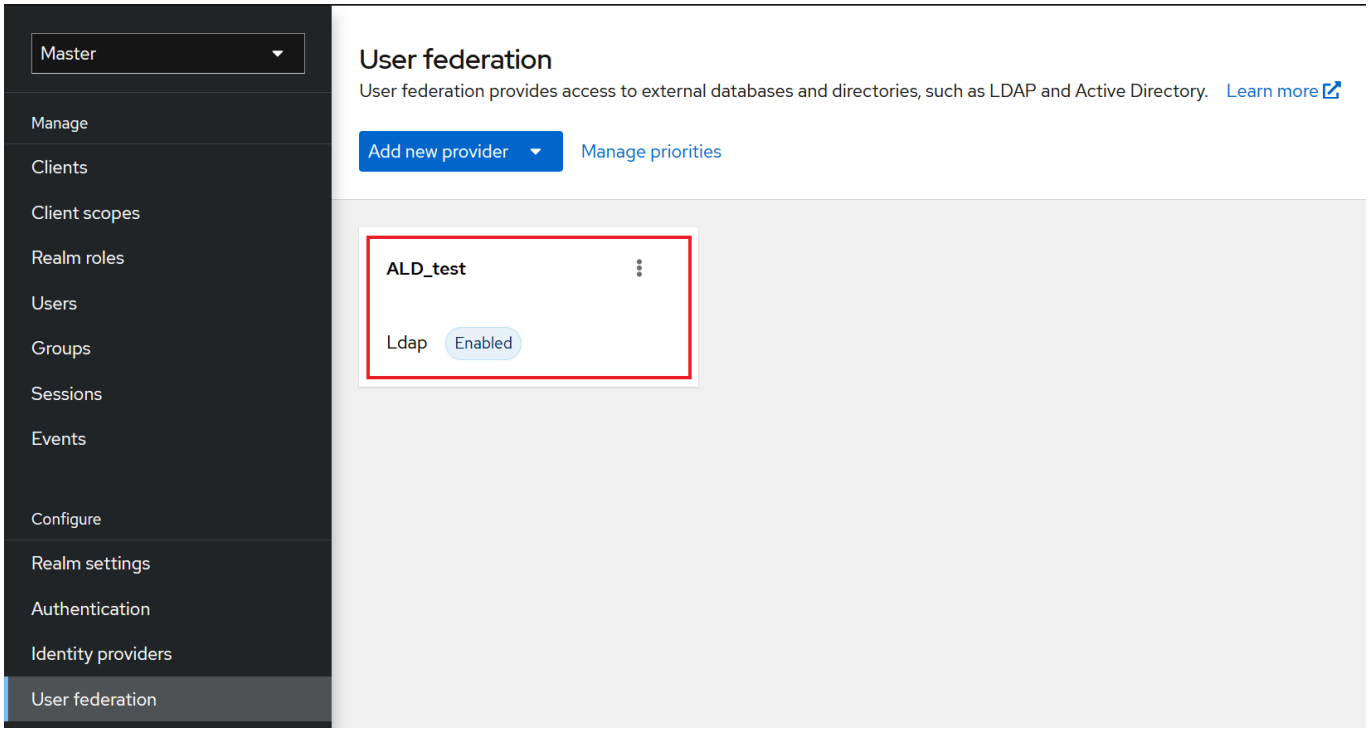
Keycloak	ALD Pro	Mapper type
email	mail	user-attribute-ldap-mapper
firstName	givenName	user-attribute-ldap-mapper

6.10.5. Синхронизация групп пользователей из ALD Pro в Keycloak

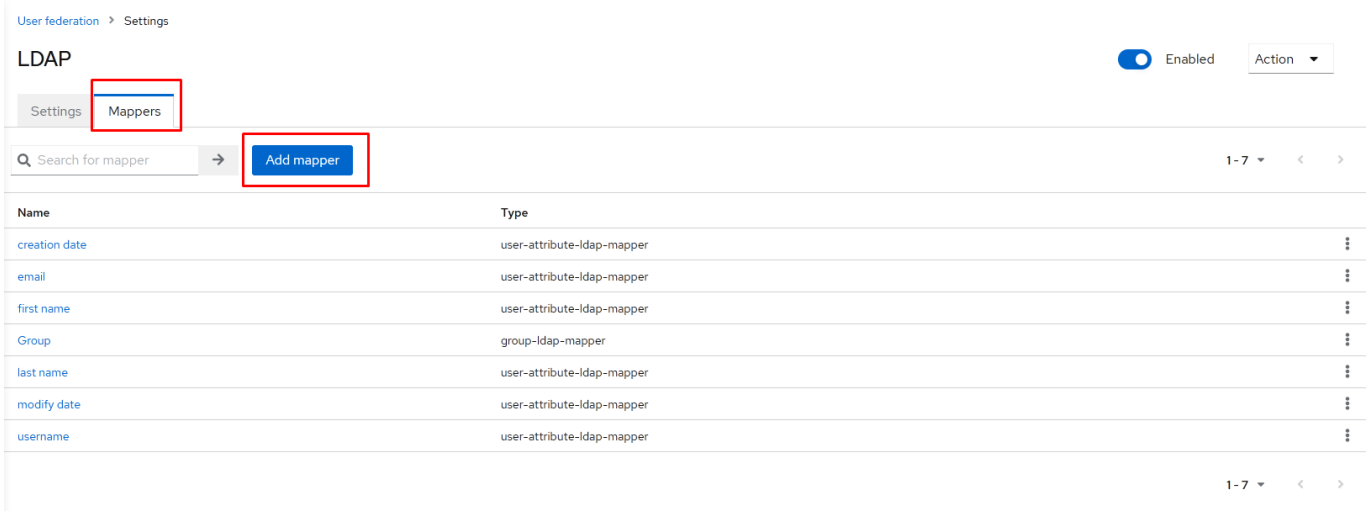
6.10.5.1. Настройка mapper в User federation

Для синхронизации пользователь должен обладать правами на чтение групп пользователей.

1. Выбрать ldap для которой нужно синхронизировать группы (рисунок 10).



2. На странице “Mappers” необходимо создать новое соответствие (рисунок 11).



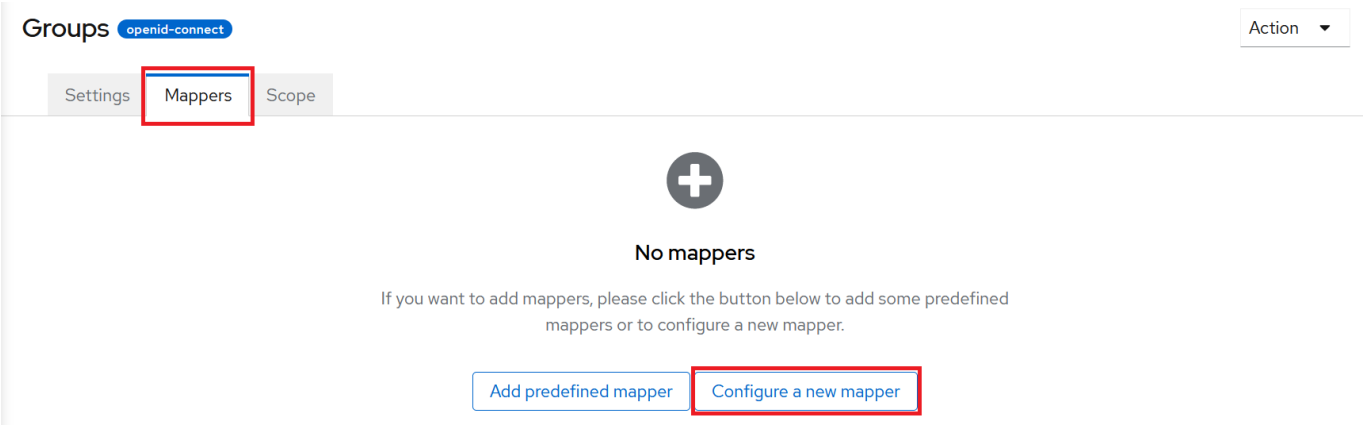
3. Для создания правильного mapper необходимо заполнить следующие поля:

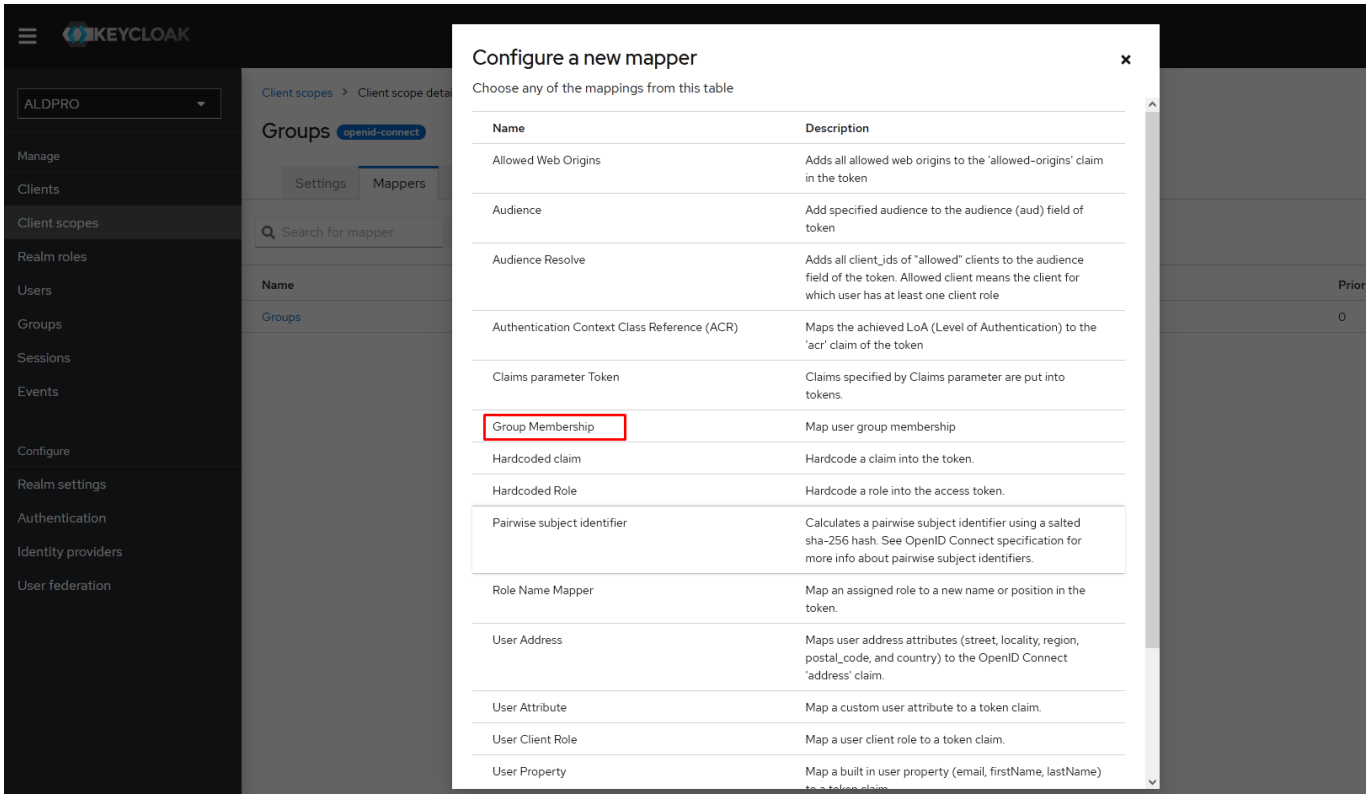
Поле	Значение
Name	<название mapper>
Mapper type	group-ldap-mapper
LDAP Groups DN (в соответствии с примером по созданию федерации пользователя)	cn=groups,cn=accounts,dc=astralinux,dc=ru
User Groups Retrieve Strategy	GET_GROUPS_FROM_USER_MEMBEROF_ATTRIBUTE

Остальные поля возможно оставить в значении по умолчанию.

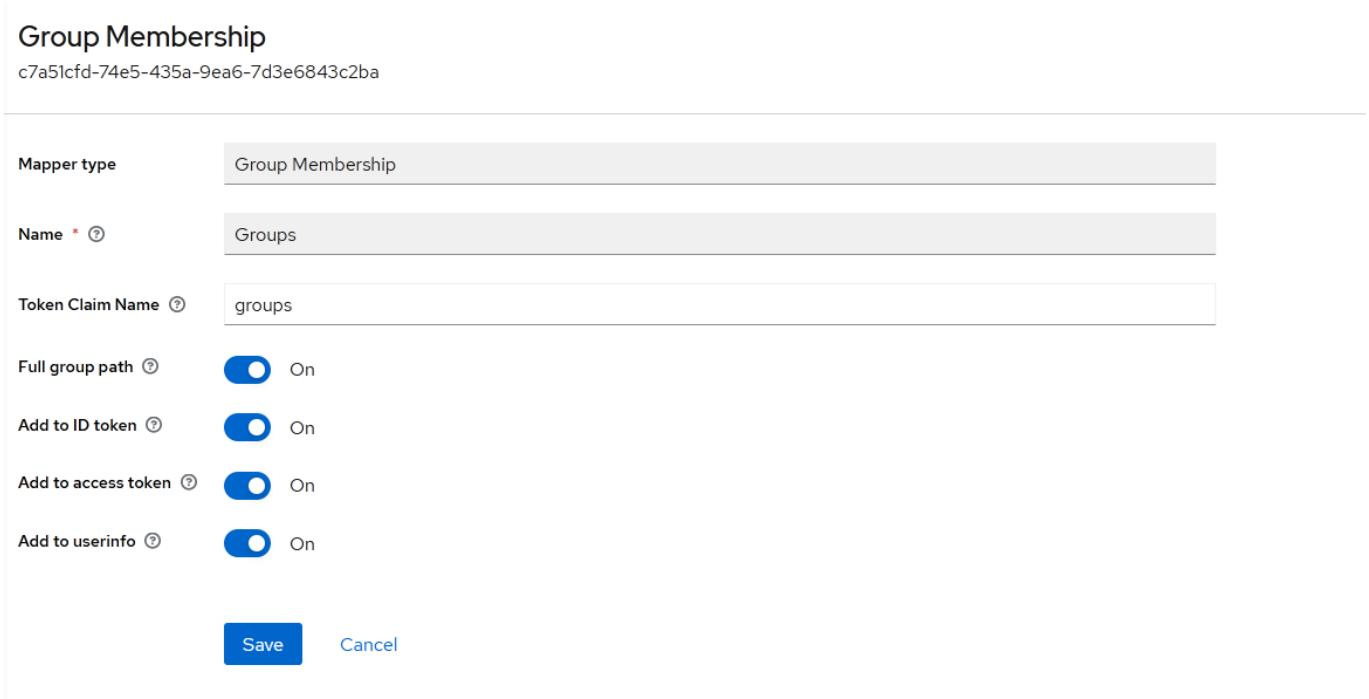
6.10.5.2. Настройка Client scope

1. На вкладке “Client Scopes” выбран “Create client scope”.
2. Параметр имя - “Groups”. Остальные по умолчанию. Сохранить.
3. Настройка mapper для этого “Client Scopes” (рисунок 12, 13).





7. mapper заполнен в соответствии с рисунком 14.



На данном шагу настройка синхронизации пользователей и групп из ALD Pro завершена.

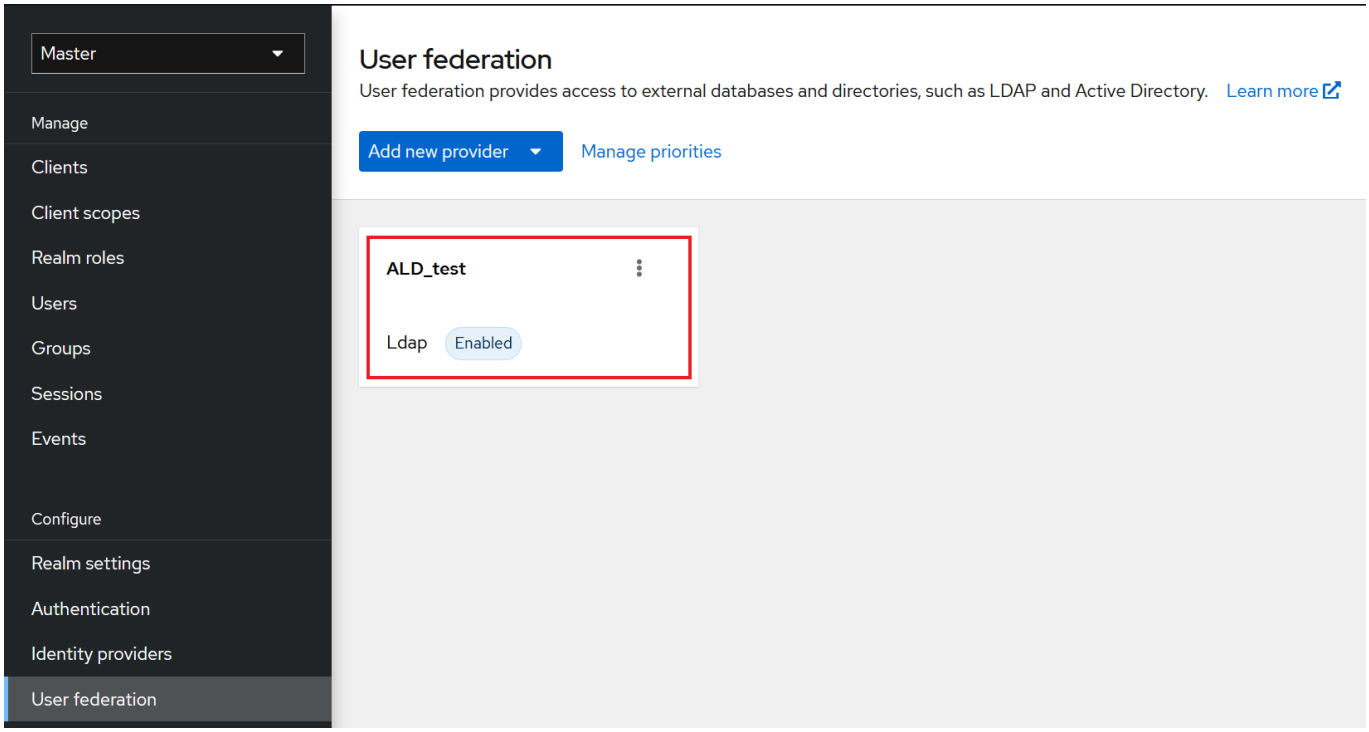
6.10.6. Синхронизация подразделений

6.10.6.1. Создание mapper

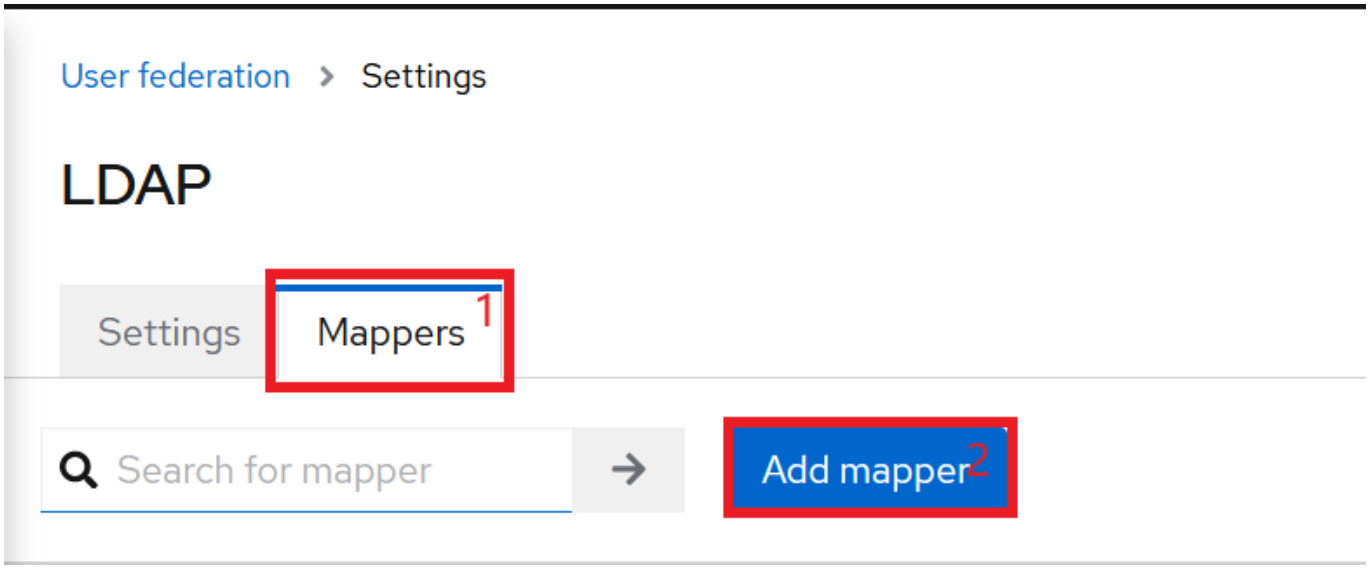
Для синхронизации подразделений сервисный пользователь должен обладать правами на просмотр подразделений.

В Keycloak из коробки нет такой сущности как подразделение, но возможна синхронизация через сущность “группы”.

1. Выполнен переход в карточку “User Federation”



2. Во вкладке “Mappers” добавлен mapper (рисунок 16).



3. Для создания правильного mapper необходимо заполнить следующие поля:

Поле	Значение
Name	<название mapper>
Mapper type	group-ldap-mapper
LDAP Groups DN (в соответствии с примером по созданию федерации пользователя)	ou=astralinux.ru,cn=orgunits,cn=accounts,dc=astralinux,dc=ru
Group Name LDAP Attribute	ou
Group Object Classes	rbta-org-unit
Member-Of LDAP Attribute	rbtadp
User Groups Retrieve Strategy	`GET_GROUPS_FROM_USER_MEMBEROF_ATTRIBUTE`

Остальные поля остаются в значении по умолчанию.

Для корректной работы протоколов может понадобиться настройка Keycloak.

6.11. Матрица совместимости ПК ALD Pro

В данной статье представлена матрица совместимости ПК ALD Pro с ОС Astra Linux и другим ПО.

Информация в таблице сформирована командой разработки ALD Pro. Тестирование с другими продуктами (не отмеченными в таблице, в том числе - с другими версиями ALSE) может быть осуществлено и подтверждено другими продуктовыми командами.

До версии 1.1.3 обновление не было кумулятивным - в связи с этим обновление с 1.0.0 до 1.1.3 необходимо осуществлять строго последовательно. В рамках версии 1.x.x рекомендуется сначала обновиться до версии 1.4.1 - например, с 1.2.0 до 1.4.1. Далее обновление может быть осуществлено до текущей версии 2.x.x (если в документации не заявлено иное) - например, можно сразу обновиться с 2.0.0 до 2.4.0.

* - Совместимость с ACM. Установка ACM на серверную группировку ALD Pro не поддерживается и не допускается. Совместимость ALD Pro и ACM предполагает возможность использования двух продуктов в одной инфраструктуре заказчика. Дополнительные проверки отсутствия аффекта решений друг на друга связаны с тем, что ряд механизмов для пользовательских ПК в обоих решениях используют SaltStack.

** - ALD Pro 2.4.0 поддерживает обновления только с двух предыдущих версий - 2.2.0 и 2.3.0.

Важно: Версия ОС Astra Linux должна совпадать на контроллере домена и подсистемах. Использования разных версий ОС может привести к возникновению проблемы или частичной неработоспособности системы.

Версия ОС Astra Linux на клиентах может не совпадать с версией на контроллере домена и подсистемах (дополнительная детализация - в таблице ниже).

ALD Pro	ОС Astra Linux	Ядро ОС	Уровень защищённости ОС	Поддержка MS AD	Браузеры	Proxmox	Vmware Workstation Pro	VirtualBox	RuPost	ACM
1.1.3	**Серверная группа ALD Pro:** 1.7.1 **Клиенты ALD Pro:** 1.7.1	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-

continues on next page

Таблица 6.1 – продолжение с предыдущей страницы

ALD Pro	ОС Astra Linux	Ядро ОС	Уровень защищённости ОС	Поддержка MS AD	Браузеры	Proxmox	Vmware Workstation Pro	VirtualBox	RuPost	ACM
1.2.0	**Серверная группа ALD Pro:** 1.7.1-1.7.2 **Клиенты ALD Pro:** 1.7.1-1.7.2	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
1.2.1	**Серверная группа ALD Pro:** 1.7.1-1.7.2 **Клиенты ALD Pro:** 1.7.1-1.7.2	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
1.3.0	**Серверная группа ALD Pro:** 1.7.1-1.7.3 **Клиенты ALD Pro:** 1.7.1-1.7.3	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
1.4.0	**Серверная группа ALD Pro:** 1.7.1 (только обновление с ALD Pro ранних версий) 1.7.2-1.7.3 **Клиенты ALD Pro:** 1.7.1-1.7.3	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
1.4.1	**Серверная группа ALD Pro:** 1.7.1 (только обновление с ALD Pro ранних версий) 1.7.2-1.7.3 **Клиенты ALD Pro:** 1.7.1-1.7.3	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
2.0.0	**Серверная группа ALD Pro:** 1.7.1 (только обновление с ALD Pro ранних версий) 1.7.2-1.7.4 **Клиенты ALD Pro:** 1.7.1-1.7.4	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
2.1.0	**Серверная группа ALD Pro:** 1.7.1 (только обновление с ALD Pro ранних версий) 1.7.2-1.7.4 **Клиенты ALD Pro:** 1.7.1-1.7.4	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
2.2.0 - 2.2.1	**Серверная группа ALD Pro:** 1.7.4 **Клиенты ALD Pro:** 1.7.1-1.7.4	**Серверная группа ALD Pro:** 5.15-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Тестирование на других браузерах до 2.3.0 не осуществлялось	7.2-7	16.2.3	от 6.1 и выше	1.0	-
2.3.0	**Серверная группа ALD Pro:** 1.7.4 1.7.5 1.7.5 UU1 **Клиенты ALD Pro:** 1.7.1-1.7.5 1.7.5 UU1 1.8.1	**Серверная группа ALD Pro:** 5.15.0-70-generic 6.1.50-1-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Google Chrome 121 и выше Chromium 111 и выше Chromium-gost 121 и выше Яндекс Браузер 23 и выше	7.2-7	16.2.3	от 6.1 и выше	1.0	1.0.0

continues on next page

Таблица 6.1 – продолжение с предыдущей страницы

ALD Pro	ОС Astra Linux	Ядро ОС	Уровень защищённости ОС	Поддержка MS AD	Браузеры	Proxmox	Vmware Workstation Pro	VirtualBox	RuPost	ACM
2.4.0* *	**Серверная группа ALD Pro:** 1.7.5 1.7.5 UU1 1.7.6 1.7.6 UU1 **Клиенты ALD Pro:** 1.7.1-1.7.5 1.7.5 UU1 1.7.6 1.7.6 UU1 1.8.1	**Серверная группа ALD Pro:** 5.15.0-70-generic 6.1.50-1-generic **Клиенты ALD Pro:** 5.4-generic 5.10-generic 5.15-generic	**Серверная группа ALD Pro:** Смоленск **Клиенты ALD Pro:** Орел Воронеж Смоленск	Windows Server 2008 R2 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Windows Server 2019	Firefox 102.0 и выше Google Chrome 121 и выше Chromium 111 и выше Chromium-gost 121 и выше Яндекс Браузер 23 и выше	7.2-7	16.2.3	от 6.1 и выше	1.0	1.1.0

Таблица 8 — Несертифицированное исполнение ALD Pro (РДЦП.10101-01)

Глоссарий

389 Directory Server

Служба каталогов уровня предприятия с открытым исходным кодом, предназначенная для централизованного управления доступом к ресурсам на множестве сетевых серверов. Ранее назывался Fedora Directory Server, а до того Netscape Directory Server.

ALD Pro

Служба каталога для Linux. Позволяет управлять парком компьютеров организации с помощью групповых политик через интуитивно понятный интерфейс. Дополнительно в системе реализованы функции автоматизированной установки ОС и ПО по сети на компьютеры в домене, удаленный доступ к рабочим столам пользователей и мониторинг состояния сервисов службы каталога.

Bind9

Открытая и наиболее распространённая реализация DNS-сервера, обеспечивающая выполнение преобразования DNS-имени в IP-адрес и наоборот. Исполняемый файл-демон сервера BIND называется named. BIND поддерживается организацией Internet Systems Consortium. Ранее назывался Berkeley Internet Name Domain или Berkeley Internet Name Daemon.

Chrony

Реализация протокола сетевого времени. Это альтернатива ntpd, эталонная реализация NTP. Он работает в Unix-подобных операционных системах и выпущен под лицензией GNU GPL v2. Это клиент и сервер NTP по умолчанию в Red Hat Enterprise Linux 8 и SUSE Linux Enterprise Server 15 и доступен во многих дистрибутивах Linux.

CUPS

CUPS (Common UNIX Printing System) — сервер печати для UNIX-подобных операционных систем. Компьютер с запущенным сервером CUPS представляет собой сетевой узел, который принимает задания на печать от клиентов, обрабатывает их и отправляет на соответствующий принтер.

FreeIPA

Акроним от англ. Free Identity, Policy and Audit — открытое программное обеспечение, специализированная служба каталогов, предназначенная для создания в ОС Linux среды, позволяющей централизованно управлять аутентификацией пользователей, устанавливать политики доступа и аудита.

IdM

Управление учётными данными (англ. Identity management, сокр. IdM, иногда IDM) — комплекс подходов, практик, технологий и специальных программных средств для управления учётными данными пользователей, системами контроля и управления доступом (СКУД), с целью повышения безопасности и производительности информационных систем при одновременном снижении затрат, оптимизации времени простоя и сокращения количества задач.

ISC DHCP

Протокол динамической настройки хоста (DHCP) - это сетевой протокол, используемый для назначения IP-адресов и предоставления информации о конфигурации таким устройствам, как серверы, настольные компьютеры или мобильные устройства, чтобы они могли обмениваться данными в сети с использованием интернет-протокола (IP). ISC DHCP - это набор программного обеспечения, которое реализует все аспекты пакета DHCP.

KDC-служба

KDC-служба это служба Kerberos представляет собой архитектуру клиент-сервер, которая обеспечивает безопасные транзакции по сетям. Она обеспечивает надежную аутентификацию пользователя, а также целостность и конфиденциальность.

LDAP-каталог

Каталог LDAP является специализированной не реляционной базой данных, файлы которой содержатся в папке */var/lib/dirsrv/slapd-ald-company-lan/db*. Информация каталога представлена в виде древовидной структуры, которую также называют Directory Information Tree или сокращенно DIT.

MIT Kerberos

Сетевой протокол аутентификации, который предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними. Kerberos выполняет аутентификацию в качестве службы

аутентификации доверенной третьей стороны, используя криптографический ключ, при условии, что пакеты, проходящие по незащищенной сети, могут быть перехвачены, модифицированы и использованы злоумышленником.

NTP

Сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей с переменной латентностью.

PAM

Pluggable Authentication Modules (PAM, подключаемые модули аутентификации) — это набор разделяемых библиотек, которые позволяют интегрировать различные низкоуровневые методы аутентификации в виде единого высокоуровневого API. Это позволяет предоставить единые механизмы для управления, встраивания прикладных программ в процесс аутентификации. Является одной из частей стандартного механизма обеспечения безопасности UNIX-систем.

PXE

PXE — среда для загрузки компьютера с помощью сетевой карты без использования локальных носителей данных.

RabbitMQ

RabbitMQ — это программный брокер сообщений на основе стандарта AMQP. Тиражируемое связующее программное обеспечение, ориентированное на обработку сообщений. Состоит из сервера, библиотек поддержки протоколов HTTP, XMPP и STOMP, клиентских библиотек AMQP для Java и .NET Framework и различных плагинов (таких как плагины для мониторинга и управления через HTTP или веб-интерфейс или плагин «Shovel» для передачи сообщений между брокерами). Имеется реализация клиентов для доступа к RabbitMQ для целого ряда языков программирования, в том числе для Perl, Python, Ruby, PHP. Поддерживается горизонтальное масштабирование для построения кластерных решений.

Reprepro

Reprepro - это инструмент для управления репозиториями APT. Он способен управлять несколькими репозиториями для нескольких версий дистрибутива и одним пулом пакетов. А так же может обрабатывать обновления из incoming каталога, копировать пакеты (ссылки) между

версиями дистрибутива, перечислять все пакеты и / или версии пакетов, доступные в репозитории, и т.д. Reprepro поддерживает внутреннюю базу данных (файл .DBM) содержимого репозитория, что делает его довольно быстрым и эффективным.

REST API

REST (от англ. Representational State Transfer — «передача репрезентативного состояния» или «передача самоописываемого состояния») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети. Другими словами, REST — это набор правил того, как программисту организовать написание кода серверного приложения, чтобы все системы легко обменивались данными и приложение можно было масштабировать.

SaltStack

Система управления конфигурациями и удалённого выполнения операций. Является программным обеспечением с открытым исходным кодом, написанным на Python.

Samba

Samba — пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB/CIFS. Имеет клиентскую и серверную части. Является свободным программным обеспечением, выпущена под лицензией GPL.

SSSD

System Security Services Daemon предоставляет набор внутренних служб для управления доступом к удалённым каталогам и механизмам проверки подлинности. Этот сервис предоставляет интерфейс NSS и PAM к операционной системе и систему подключаемых внутренних серверов для установки соединения с несколькими разными источниками учётных записей, а также интерфейс D-Bus. Также он является основой сервисов аудита и политики доступа клиентов для таких проектов, как FreeIPA.

Syslog-NG

syslog-ng - это бесплатная реализация протокола syslog с открытым исходным кодом для Unix и Unix-подобных систем. Он расширяет оригинальную модель syslogd фильтрацией на основе содержимого, широкими возможностями фильтрации, гибкими параметрами конфигурации и добавляет важные функции в системный журнал, такие как

использование TCP для транспортировки.

TFTP

Trivial File Transfer Protocol — простой протокол передачи файлов. Используется главным образом для первоначальной загрузки бездисковых рабочих станций. TFTP, в отличие от FTP, не содержит возможностей аутентификации (хотя возможна фильтрация по IP-адресу) и основан на транспортном протоколе UDP.

TGT-билет

TGT-билет это билет для выдачи билетов или Билет для получения билетов представляет собой небольшой зашифрованный идентификационный файл с ограниченным сроком действия. После аутентификации этот файл предоставляется пользователю для защиты трафика данных подсистемой центра распределения ключей служб аутентификации, такой как Kerberos.

Zabbix

Zabbix — свободная система мониторинга статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования.