Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение   
высшего профессионального образования   
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр информационной безопасности

**Лабораторный практикум**

**по теме: «Керберос настройка и организация связи»**

**Оглавление**

[Введение 3](#_Toc43188863)

[Терминология Kerberos 4](#_Toc43188864)

[Описание протокола Kerberos 5](#_Toc43188865)

[Принцип работы Kerberos 6](#_Toc43188866)

[Лабораторный практикум 11](#_Toc43188867)

[Контрольные вопросы 16](#_Toc43188868)

[Литература 17](#_Toc43188869)

# Введение

Kerberos — сетевой протокол аутентификации, который предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними, причём в протоколе учтён тот факт, что начальный обмен информацией между клиентом и сервером происходит в незащищенной среде, а передаваемые пакеты могут быть перехвачены и модифицированы.

# Терминология Kerberos

1. Учетная запись (Principal): любые пользователи, компьютеры или сервисы, предоставляемые серверами, должны быть определены как учетные записи Kerberos
2. Требования (Instances): используются для сервисных и специальных административных учетных записей
3. Области (Realms): уникальная область управления, обеспечиваемая установкой Kerberos
4. Центр распространения ключей (KDC): состоит из трех частей: базы данных всех учетных записей, сервера аутентификации и сервера предоставления билетов. Для каждой области должен быть хотя бы один KDC
5. Билет для получения билета (TGT): изданный сервером аутентификации, TGT зашифровывается на пароле пользователя, который известен только пользователю и KDC
6. Сервер распространения билетов (TGS): выпускает сервисные билеты для клиентов по запросу
7. Билеты (Tickets): подтверждение идентичности двух учетных записей. Одна учетная запись - пользователь, а другая - сервис, запрашиваемый этим пользователем. Билеты устанавливают секретный ключ, используемый для защищенного соединения во время авторизованной сессии
8. Файлы ключей (Keytab Files): файлы, извлеченные из базы учетных записей KDC и содержащие ключ шифрования для сервиса или компьютера

# Описание протокола Kerberos

Протокол Kerberos был разработан для обеспечения надежной аутентификацию пользователей. Он может использовать централизованное хранение аутентификационных данных, где пользователь вводит пароль только один раз за сессию. Таким образом, пользователи имеют доступ ко всем сервисам, на которые они авторизованы, без необходимости заново вводить пароль во время сессии. Также в протоколе используется взаимная аутоинтефикация, когда не только пользователь, но и прикладные серверы должны подтвердить свою идентичность пользователям. Kerberos поддерживает генерацию и обмен ключей шифрования после завершения этапов аутентификации и авторизации клиента и сервера. Одно из главных преимуществ протокола Kerberos, обеспечение высокого уровня безопасности, так как при любых взаимодействиях не передаются ни пароли, ни значения хеша паролей в открытом виде. Протокол основан на понятии Ticket (билет). Ticket является зашифрованным пакетом данных, который выдается доверенным центром аутентификации (KDC, центр распределения ключей). Работая с протоколом Kerberos, системные часы всех участвующих во взаимодействие узлов должны быть синхронизированы.

# Принцип работы Kerberos

***В процессе аутентификации задействованы следующие компоненты:***

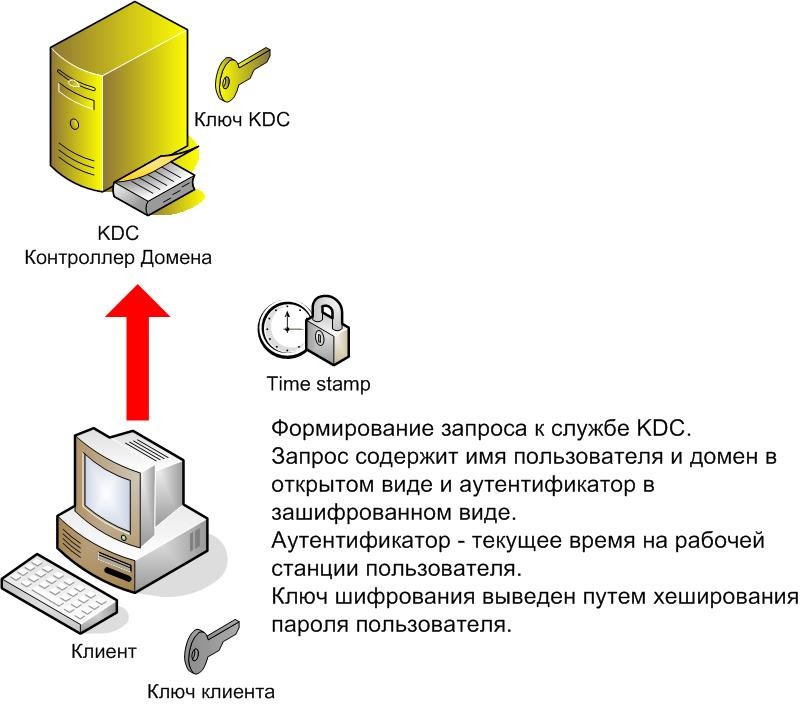
1. Клиент, запрашивающий доступ к службе или пытающийся осуществить аутентификацию.
2. Сервер, на котором работают службы, доступ к которому требуется клиенту
3. Компьютер, которому доверяет клиент (контроллер домена, на котором выполняется служба KDC)
4. KDC - служба, работающая на физическом защищенном сервере

KDC ведет базу учетных данных с информацией обо всех участниках безопасности (security principal) своей области. Вместе с информацией о каждом security principal в базе данных KDC сохраняется криптографический ключ, известный только этому объекту и службе KDC. Указанный ключ, который называют долговременным, используется для связи пользователя системы безопасности с центром распределения ключей.

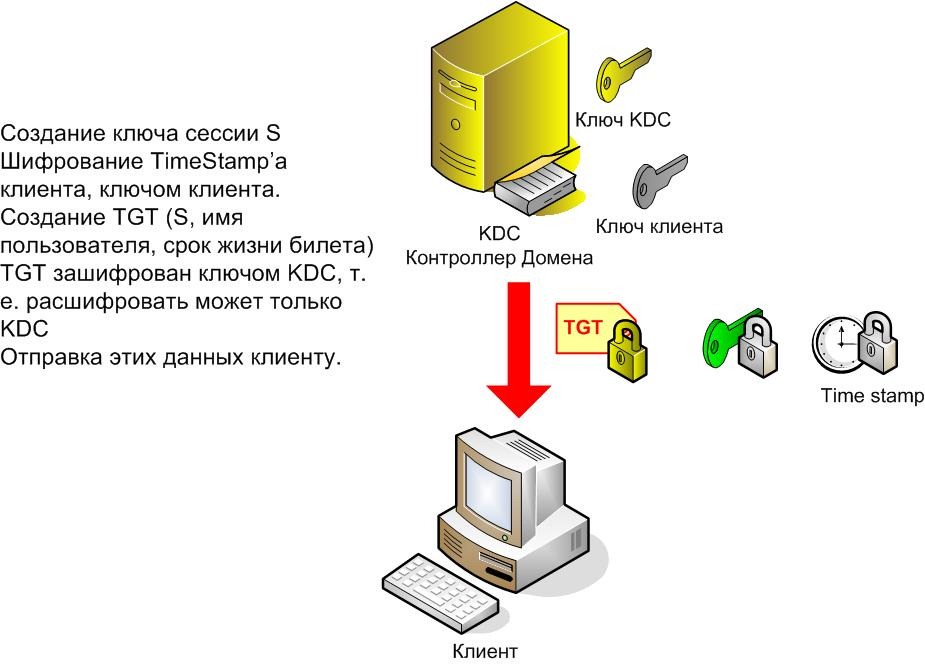
***Аутентификация клиента***

1. Получив приглашение на ввод имени пользователя, пароля и домена, пользователь указывает эти данные.

2. Затем компьютер пользователя обращается к службе KDC и передает ей имя пользователя, имя домена, а также текущее время на рабочей станции пользователя, при этом имя пользователя передается в открытом виде, текущее время на рабочей станции пользователя передается в зашифрованном виде и является аутентификатором. Ключ шифрования формируется из пароля пользователя в результате хеширования.



3. Служба KDC ищет пользователя в AD, выявляет мастер ключ пользователя, который основан на пароле пользователя и расшифровывает аутентификатор, т. е. получает время отправки запроса. Разница во времени отправки запроса и текущего времени на контроллере домена не должно превышать определенного значения, установленного политикой протокола Kerberos.



4. Затем KDC создает два объекта:

a. ключ сессии, посредством которого будет обеспечиваться зашифрование данных при обмене между клиентом и службой KDC,

b. билет на получение билета TGT. TGT включает: вторую копию ключа сессии, имя пользователя, время окончания жизни билета. Билет на получение билета шифруется с использованием собственного мастер ключа службы KDC, который известен только KDC, т. е. TGT может быть расшифрован только самой службой KDC.

5. Служба KDC зашифровывает аутентификатор пользователя (time stamp) и ключ сессии с помощью ключа клиента. После этого эти данные отправляются клиенту.

6. Компьютер клиента получает информацию от службы KDC, проверяет аутентификатор, расшифровывает ключ сессии.

7. Теперь клиент обладает ключом сессии и TGT, что предоставляет возможность безопасного взаимодействия со службой KDC. Клиент аутентифицирован в домене и получает возможность осуществлять доступ к ресурсам домена, используя протокол Kerberos.

***Получение доступа к ресурсам на других серверах в том же домене***

1. Клиент обращается к службе KDC. Клиент представляет KDC свой TGT и маркер времени, которые зашифрованы с помощью ключа сессии, известного службе KDC.

2. KDC расшифровывает TGT, используя свой собственный ключ. Маркер времени расшифровывается с помощью сессионного ключа. Теперь KDC может подтвердить, что запрос пришел от «правильного» пользователя, т. к. этот пользователь может использовать этот сессионный ключ.



3. Затем KDC создает пару билетов, один для клиента, один для сервера, к ресурсам которого клиент должен будет получать доступ. Каждый билет содержит имя пользователя, запрашивающего доступ, получателя запроса, маркер времени, показывающий, когда был создан билет, а также срок жизни билета. Оба билета будут также содержать новый ключ, K\_cs который, таким образом известен и клиенту и серверу. Этот ключ будет обеспечивать возможность безопасного взаимодействия между ними. KDC шифрует билет сервера, используя мастер – ключ сервера, затем вкладывает билет сервера внутрь билета клиента, который также содержит ключ K\_cs.

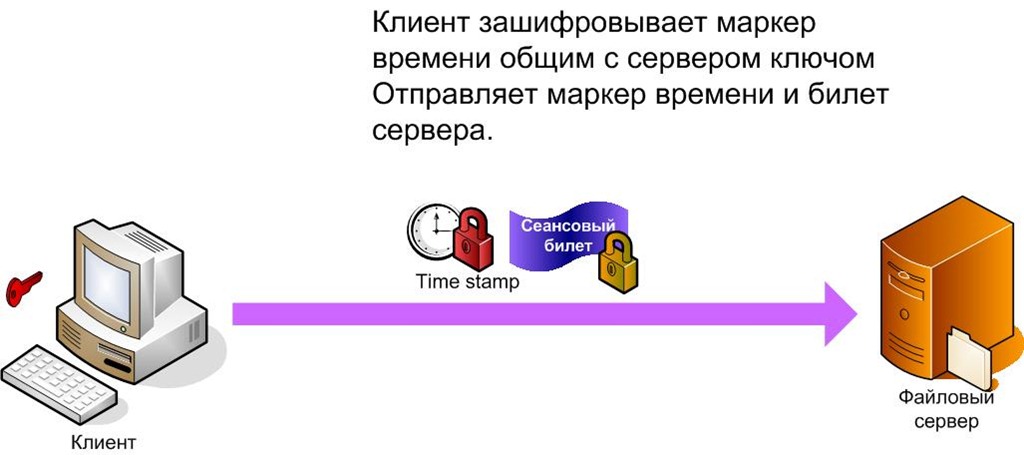
4. Вся эта структура зашифровывается с помощью сессионного ключа, который стал доступен пользователю при аутентификации. После чего эта информация отправляется клиенту.



5. Получив билет, клиент расшифровывает его с помощью сессионного ключа, т. е. K\_cs становится доступным клиенту, K\_cs доступен также и серверу. Клиент не может прочитать билет сервера, т. к. он зашифрован на ключе сервера.



6. Клиент зашифровывает маркер времени с помощью ключа, K\_cs затем отправляет маркер времени и билет сервера самому серверу, к ресурсам которого пытается получить доступ клиент

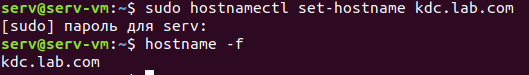


7. Получив эту информацию, на первом этапе сервер расшифровывает свой билет, используя свой долговременный ключ. Это предоставляет возможность получить доступ к K\_cs, с помощью которого будет на втором этапе расшифрован маркер времени, полученный от клиента.

8. Теперь и клиент, и сервер обладают ключом K\_cs. Следовательно, сервер может быть уверен в том, что клиент правильно идентифицирован, т. к. для шифрования маркера времени был использован K\_cs. В случае необходимости ответа сервера клиенту, сервер воспользуется ключом K\_cs. Клиент будет знать, что сервер правильно идентифицирован, поскольку сервер должен использовать, чтобы получить K\_cs.

# Лабораторный практикум

1. Создать 2-е виртуальные машины (ВМ) в среде виртуализации VMWare. (Одна ВМ будет использоваться в качестве клиентской машины, вторая будет серверной).
2. Установить на каждую ВМ дистрибутив Linux (рекомендуется Ubuntu).
3. Выполните синхронизацию времени с помощью NTP, для корректной работы Kerberos
4. Запустить серверную ВМ и авторизоваться под созданным пользователем.
5. Зададим новое сетевое имя: kdc.lab.com



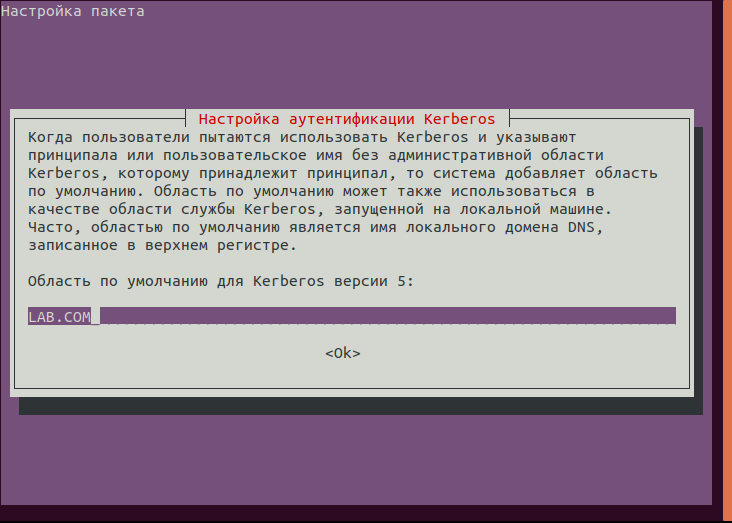
1. Первый шаг по созданию области Kerberos - это установка пакетов krb5-kdc и krb5-admin-server. Введите в терминале:

sudo apt-get install krb5-kdc krb5-admin-server

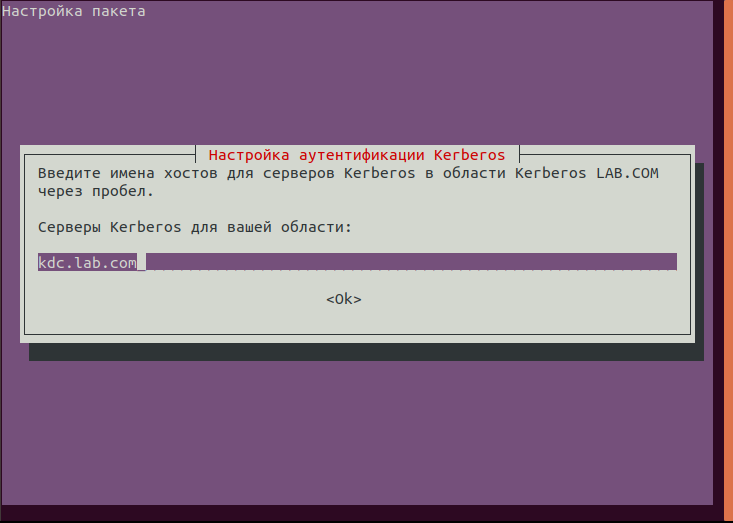
Примечание: если вылазит ошибка package has no installation candidate требуется обновить списки репозиториев это можно сделать командой

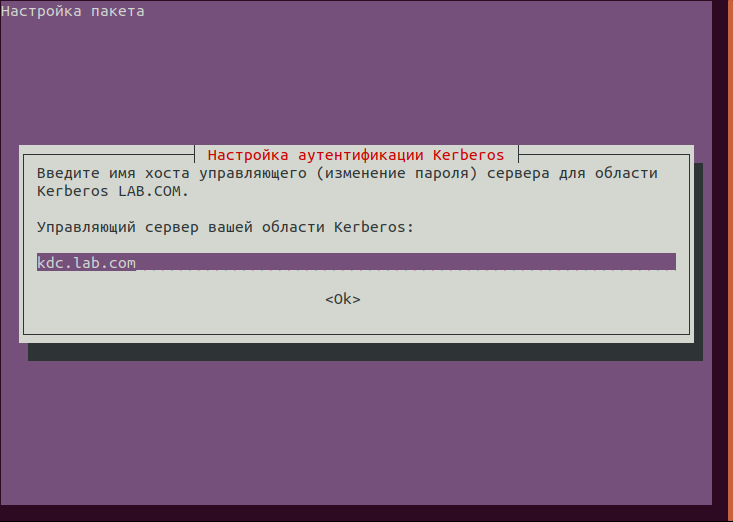
sudo apt update

1. При установке вылезет окошко с настройками



Примечание: требуется задать DNS, зададим LAB.COM



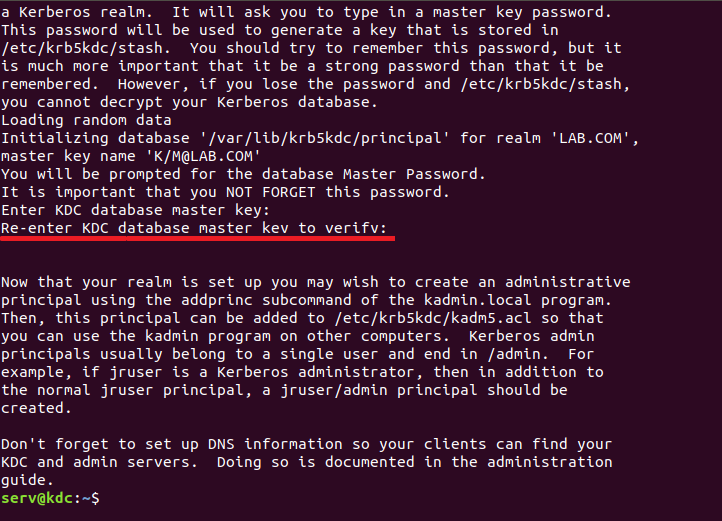


Примечание: здесь следует установить одинаковые имена как kdc.lab.com

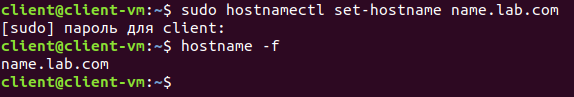
1. Далее создаем новую область с помощью утилиты kdb5\_newrealm:

sudo krb5\_newrealm

1. При создании будет запрошен ключ kdc (не забудьте его)



1. Перейдем к настройке подобным образом второй машины. Запустим ее.
2. Зададим новое сетевое имя: name.lab.com заместо name ваше имя



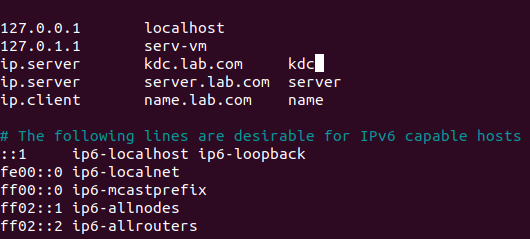
1. Нужно посмотреть ip адреса на каждой из машины сделать это можно командой

ifconfig

1. Далее с машины сервера отредактируем файл /etc/hosts можно использовать утилиту nano



1. Отредактированный файл должен выглядеть примерно следующим образом заменив требуемые значения на свои.



13) Для проверки между хостами выполните ping для каждого хоста

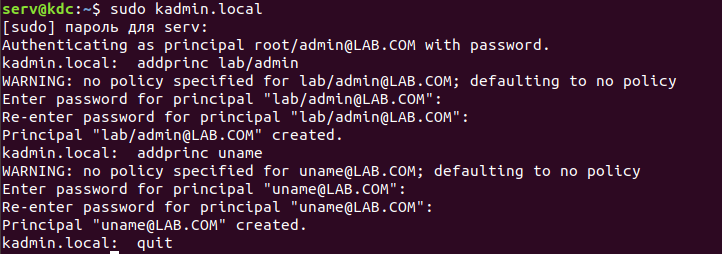
1. Создадим принципиалов (пользователей) для этого введем команду

sudo kadmin.local

1. Далее с помощью addprinc добавим несколько пользователей

* lab/admin
* uname

1. После ввода пароля должна появится надпись об успешном создании принципиала.
2. Выйти из kadmin.local можно командой quit



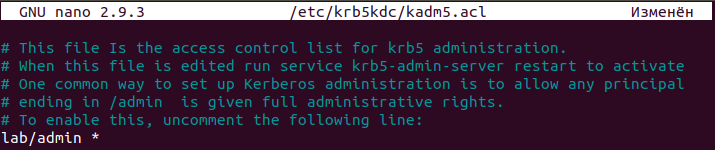
1. Перейдем к дальнейшей настройке клиентской машины. Установим на нее следующие компоненты командой

sudo apt-get install krb5-user libpam-krb5 libpam-ccreds auth-client-config

1. Для DNS вводим так-же LAB.COM, а дальше ip сервера
2. Теперь можно проверить настройку сервера и клиента используя команду kinit uname аутентифицируемся на сервере.
3. Дальше с помощью команды klist посмотрите выдаваемую информацию. Что можно узнать при выводе этой команды?
4. Проанализируем трафик для этого можно воспользоваться любым снифером например Wireshark
5. Установим его и запустим с клиентской машины командой

sudo wireshark

1. Выберете сетевой протокол и начните захват пакетов и выполните уже известные команды для взаимодействия клиента и сервера.
2. Остановите захват пакетов и проанализируйте захваченные пакеты. По каким протоколом происходит взаимодействие и каким образом посредством увиденного в захваченных пакетах.
3. Почему первый ответ сервера на попытку авторизоваться сигнализирует об ошибке?
4. С серверной машины напишите уже известную программу kadmin.local какое сообщение нам выдалось? Под каким принципиалом мы пытались пройти аунтификацию.
5. Зайдите под принципиала lab/admin и повторите команду изменился ли результат?
6. Откройте с помощью редактора нано файл /etc/krb5kdc/kadm5.acl и посмотрите содержимое. Какую строчку требуется изменить что бы принципиал получил полные права?
7. Раскоментируйте последнюю строчку и измените что бы она выглядела как lab/admin \*



1. Повторите попытку что выдало на этот раз?
2. Перезапустите сервис командой

sudo systemctl restart krb5-admin-server.service

1. Снова повторите попытку. Почему такой результат?
2. Добавьте к команде приставку sudo под каким принципиалом была аунтификация. Сделайте выводы.
3. Попробуйте авторизоваться под этим пользователем. Что сказал на это керборос?
4. Посмотрите список пользователей командой

sudo kadmin.local -q ‘listprincs \*’

1. Включите захват трафика и последовательно попробуйте авторизоваться под автоматически созданным пользователем вначале на сервере потом на клиенте. Сравните полученный трафик.
2. Почему ранее на первоначальный запрос AS\_REQ мы получали ответ KRB\_ERROR, а сейчас AS\_REP как при успешной авторизации?
3. Сравните два AS\_REP отличается ли их структура?
4. Проверьте klist какому принципиалу выдан билет и почему если структурно AS\_REP идентичны то под одного пользователя мы можем зайти а под другого нет.
5. Попробуйте создать с клиентской машины новый реалм. Почему такой результат?
6. Выполните команду

kdestroy

1. Посмотрите на результат klist что изменилось, выполните klist с другой машины объясните результат?

# Контрольные вопросы

1. Для чего служат автоматически созданные пользователи в системе kerberos?
2. Если не выполнять аунтификацию под принципиала под каким принципиалом будут выполнятся команды без приставки sudo?
3. Можно ли с клиентской машины редактировать серверные файлы?
4. Почему нельзя создать новую область с клиентской машины
5. Если установить на клиентскую машину пакеты как для серверной получится ли изменять серверные файлы? Если да то где они меняются?

# Литература

1. https://ru.wikipedia.org/wiki/Kerberos
2. Установка Kerberos в Ubuntu https://www.tux.in.ua/articles/217
3. Руководство по выживанию — Kerberos https://pro-ldap.ru/tr/zytrax/tech/kerberos.html
4. https://habr.com/ru/post/143187/