Московский институт электроники и математики НИУ ВШЭ

**Кафедра Информационно-коммуникационных технологий**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

На тему: ≪Разработка виртуальной среды для освоения протоколов распределенной аутентификации и авторизации пользователей≫

Студент: **Кузьмин Евгений Игоревич**

Руководитель проекта: **Фомин Сергей Сергеевич**

Допущен к защите \_\_\_\_\_\_\_\_\_\_\_ 2013 г.

Консультанты проекта:

Специальная часть **С. С. Фомин**

Охрана труда **Е. Б. Михайлов**

Зав. кафедрой проф. д.т.н. **В. Н. Азаров**

**Москва 2013**

# Аннотация

В данной работе была разработана и реализована виртуальная среда для освоения протоколов распределенной аутентификации и авторизации пользователей.

В ходе работы были рассмотрены существующие протоколы аутентификации и авторизации пользователей, обзор и выбор системы виртуализации и операционной системы. В процессе реализации подробно описаны каждые необходимые шаги для реализации виртуальной среды. Особое внимание уделено установке и конфигурированию протокола Kerberos, а так же реализации технологии единого входа на примере ssh. Был проведен эксперимент, подтверждающий корректную работу виртуальной среды для освоения протоколов аутентификации и авторизации пользователей.

Оглавление

[Аннотация 2](#_Toc357980698)

[Введение 5](#_Toc357980699)

[1 Обзорно-аналитическая часть 6](#_Toc357980700)

[1.1 Обзор протоколов аутентификации и авторизации пользователей 6](#_Toc357980701)

[1.1.1 Протокол аутентификации Kerberos 6](#_Toc357980702)

[1.1.1.1 Первичная аутентификация 9](#_Toc357980703)

[1.1.1.2 Получение разрешения на доступ к ресурсному серверу 12](#_Toc357980704)

[1.1.1.3 Получение доступа к ресурсу 14](#_Toc357980705)

[1.1.1.4 Технология единого входа 15](#_Toc357980706)

[1.1.2 Протокол аутентификации RADIUS 15](#_Toc357980707)

[1.1.3 Протокол аутентификации TACACS+ 18](#_Toc357980708)

[1.2 Выбор системы управления виртуальными машинами 21](#_Toc357980709)

[1.3 Выбор операционной системы 22](#_Toc357980710)

[1.3.1 Операционная система Debian 22](#_Toc357980711)

[1.3.2 Операционная система Ubuntu 23](#_Toc357980712)

[1.3.3 Операционная система Arch Linux 24](#_Toc357980713)

[1.3.4 Операционная система Fedora 25](#_Toc357980714)

[1.4 Выводы по разделу 25](#_Toc357980715)

[2. Разработка 26](#_Toc357980716)

[2.1 Разработка методики работы со стендом 26](#_Toc357980717)

[2.2 Разработка структуры виртуальной среды 28](#_Toc357980718)

[2.3. Выводы по разделу 28](#_Toc357980719)

[3. Реализация 28](#_Toc357980720)

[3.1 Установка системы управления виртуальными машинами 28](#_Toc357980721)

[3.2Создание и настройка виртуальной машины 30](#_Toc357980722)

[3.3 Настройка сетевых интерфейсов виртуальных машин 34](#_Toc357980723)

[3.4 Установка и настройка клиент-серверных компонентов протокола Kerberos 37](#_Toc357980724)

[3.4.1 Настройка конфигурационных файлов протокола Kerberos 43](#_Toc357980725)

[3.4.2 Пользователи Kerberos 48](#_Toc357980726)

[3.4.3 Настройка DNS 49](#_Toc357980727)

[3.5 Технология единого входа Single Sign On (SSO) 50](#_Toc357980728)

[3.5.1 Установка SSH 51](#_Toc357980729)

[3.5.2 Настройка сервера SSH 51](#_Toc357980730)

[3.6 Настройка клиента протокола Kerberos 51](#_Toc357980731)

[3.6.1 Настройка DNS 53](#_Toc357980732)

[3.6.2 Настройка SSH-клиента 54](#_Toc357980733)

[3.7 Выводы по разделу 54](#_Toc357980734)

[4 Экспериментальная часть 54](#_Toc357980735)

[4.1 Разработка методики испытания 54](#_Toc357980736)

[4.1.1 Объект испытаний 54](#_Toc357980737)

[4.1.2. Цель испытаний 54](#_Toc357980738)

[4.1.3. Требования к программе 54](#_Toc357980739)

[4.1.4. Средства и порядок испытаний 55](#_Toc357980740)

[4.1.5. Методы испытаний 55](#_Toc357980741)

[4.2 Проведение натурного эксперимента 56](#_Toc357980742)

[4.4. Выводы по разделу 60](#_Toc357980743)

[5. Охрана труда 62](#_Toc357980744)

[5.1 Исследование возможных опасных и вредных факторов при эксплуатации ЭВМ и их влияния на пользователей. 62](#_Toc357980745)

[5.1.1 Анализ влияния опасных и вредных факторов на пользователя 63](#_Toc357980746)

[5.1.2 Требования к помещениям и организации рабочих мест 65](#_Toc357980747)

[5.1.3 Требования к организации работы 67](#_Toc357980748)

[5.2 Методы и средства защиты пользователей от воздействия на них опасных и вредных факторов 67](#_Toc357980749)

[5.3. Выводы по разделу 69](#_Toc357980750)

[6 Итоги 70](#_Toc357980751)

[7 Список литературы 71](#_Toc357980752)

[Приложение 1 72](#_Toc357980753)

# Введение

Корпоративная сеть - это сеть, главным назначением которой является поддержание работы конкретного предприятия, владеющего данной сетью. Пользователями корпоративной сети являются только сотрудники данного предприятия. Формально корпоративной сетью является сеть предприятия любого масштаба, обычно это названия используют для сети крупного предприятия, имеющего отделения в различных городах и, возможно, разных странах.

Рассмотрим вполне себе обычную ситуацию, когда сотрудник, работающий в крупном предприятии и находящийся в одной точке земного шара, запрашивает разрешение на доступ к ресурсу центрального офиса корпорации, который в свою очередь находится в другой точке земного шара, запрос пойдет по незащищенным сетям связи где его могут перехватить злоумышленники и выдать себя за сотрудника фирмы. Тогда каким образом узнать, что человек, запросивший разрешение на доступ к ресурсу действительно является сотрудником фирмы ?

На такие вопросы отвечают механизмы аутентификация и авторизация пользователей.

Аутентификация – это проверка подлинности пользовательского идентификатора.

Авторизация - предоставление определённому лицу или группе лиц прав на выполнение определённых действий.

Корпоративные фирмы находятся в жесткой конкуренции между собой, поэтому даже незначительные просчеты в безопасности могут привести к необратимым последствиям.

Значит, нужны квалифицированные кадры в области сетевой безопасности.

Современные учебные программы по подготовке специалистов готовят таких кадров.

Процесс подготовки таких специалистов довольно сложный, поскольку ценность таких сотрудников состоит в богатом практическом опыте, который достигается работой с оборудованием. Для этого создаются целые лаборатории, которые состоят из оконечных сетевых устройств, а так же различного вида кабелей и т.д. и т.п.

Создание таких лабораторий требует больших затрат. Вот если можно было бы уместить целую лабораторию в обычный PC…

Таким образом, целью данной работы является разработка виртуальной среды для освоения протоколов распределенной аутентификации пользователей.

# 1 Обзорно-аналитическая часть

## 1.1 Обзор протоколов аутентификации и авторизации пользователей

## 1.1.1 Протокол аутентификации Kerberos

Kerberos — это сетевой протокол, предназначенный для централизованного решения задач аутентификации и авторизации в незащищенных сетях. [1]

Кроме того Kerberos является одной из возможных реализаций технологии единого входа.

Протокол основан на следующих принципах:

* В сетях, использующих технологию Kerberos, процессы аутентификации и авторизации выполняются через авторитетного (которому доверяют обе стороны процесса) посредника, и в качестве посредника выступает сама система Kerberos. Иногда этот принцип называют «аутентификация на основе доверенной третьей стороны».
* Для доступа к любой запрашиваемой службе/услуги клиенту необходимо пройти аутентификацию.
* Процесс обмена данными является защищенным с использованием алгоритма шифрования DES.

Сетевой протокол Kerberos основан на клиент-серверной архитектуре. Kerberos-клиент необходимо установить на всех компьютерах сети, которые будут запрашивать доступные сетевые услуги. Тогда Kerberos-клиент от имени пользователя передаст запрос на Kerberos-cepвep и будет поддерживать с ним диалог, необходимый для выполнения функций системы Kerberos.

В технологии Kerberos имеются следующие компоненты:

* Kerberos-cepвep
* Kerberos-клиенты
* Ресурсные серверы

Kerberos-клиенты пытаются получить доступ к сетевым ресурсам — файлам, приложениям, принтеру и т. д. Доступ может быть разрешен только зарегистрированным пользователям и при наличии соответствующих прав (полномочий), которые распределяют службы авторизации соответствующих ресурсных серверов, — файловым сервером, сервером приложений, сервером печати. Но в системе Kerberos ресурсные сервера не получают «напрямую» запросы от клиентов, а принимают запросы тогда и только тогда, когда это разрешает Kerberos-cepвep.

Таким образом, путь клиента к ресурсу в системе Kerberos состоит из трех этапов.

1. Определение валидности клиента, логический вход в сеть, получение разрешения на продолжение процесса получения доступа к ресурсу.
2. Получение разрешения на обращение к ресурсному серверу.
3. Получение разрешения на доступ к ресурсу.

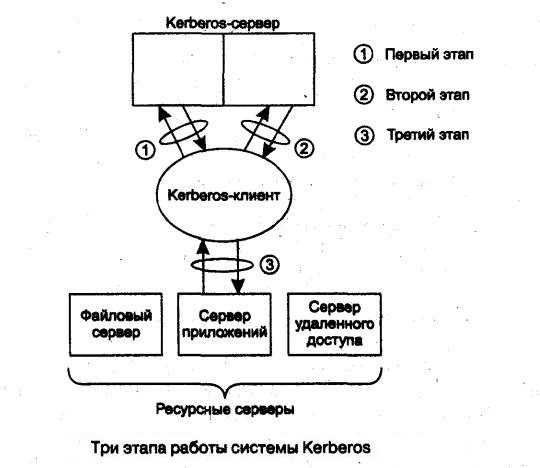


Рис. 1 Этапы работы системы Kerberos

Для решения первой и второй задач клиент обращается к Kerberos-cepвepy. Выполняется первичная аутентификации и выдается разрешение на продолжение процесса получения доступа к ресурсу, которое осуществляется сервером аутентификации (Authentication Server, AS). На этом сервере хранятся в своей личной базе данных информация о паролях и идентификаторах пользователей.

Вторую задачу, возможность обратиться к ресурсному серверу, решает другая часть Kerberos-cepвepa — сервер квитанций (Ticket Granting Server, TGS). Сервер квитанций для валидных пользователей выполняется дополнительная проверка и выдается клиенту разрешение на доступ к нужному ему ресурсному серверу, для чего он наделяет его электронной квитанцией.

Для исполнения своих функций сервер квитанций использует копии секретных ключей всех ресурсных серверов, которые хранятся в личной базе данных. Помимо этих ключей TGS-сервер имеет еще один секретный DES-ключ, общий с AS-сервером.

Третья задача — получить разрешение на доступ к ресурсу — решается на уровне ресурсного сервера.

Квитанция – электронный документ, который выдается Kerberos сервером, цель которой удостоверение личности пользователя и разрешить доступ к ресурсам.

### 1.1.1.1 Первичная аутентификация

Разрешение доступа пользователя к ресурсам заключается в два шага:

* пользователь должен пройти аутентификацию
* пользователь должен пройти авторизацию

В технологии Kerberos подразумевается, что пользователь проходит процесс аутентификации один раз во время логического входа в сеть, а затем аутентифицируется и авторизируется всякий раз, когда требуется доступ к новому ресурсному серверу.

Выполняя логический вход в сеть, пользователь, а точнее Кегberos-клиент, установленный на его компьютере, досылает серверу аутентификации AS идентификатор пользователя ID.

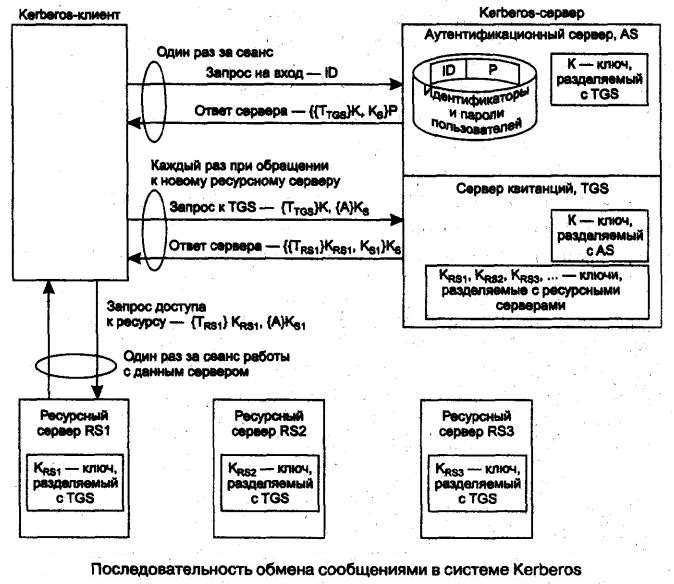


Рис. 2 Процесс обмена сообщениями в системе Kerberos

Сначала сервер аутентификации проверит в своей базе данных наличие записи о пользователе с таким идентификатором, после, если такая запись существует, извлекает из нее пароль пользователя *р*. С помощью этого пароля будет зашифрована информация, которую направит сервер аутентификаций Кегberos-клиенту в виде ответа. А ответ состоит из квитанции *TTGS*  на доступ к серверу квитанций Kerberos и ключа сеанса*KS*.

Здесь сеанс – это все время работы пользователя от момента логического входа в сеть до момента логического выхода. Ключом сеанса зашифровывается информация в процессах аутентификации в течение пользовательского сеанса. Квитанция шифруется секретным DES-ключом *К*, который разделяют серверы аутентификации и квитанций. Далее, ключ сеанса и зашифрованная квитанция – дополнительно шифруются пользовательским паролем *р.* Следовательно, квитанция шифруется дважды ключом *К* и паролем *р*. В приведенных обозначениях сообщение-ответ, которое сервер аутентификации посылает клиенту, выглядит следующим образом: *{{TTGS}K, KS}p.*

Поступившее на клиентскую машину такое ответное сообщение, активирует клиентскую часть Kerberos и потребует пользователя ввести свой пароль. После того, как пользователь ввел свой пароль, Kerberos-клиент пробует расшифровать поступившее сообщение с помощью этого пароля. Если пароль верен, то из сообщения извлекаются квитанция на доступ к серверу квитанций *{TTGS}K* (в зашифрованном виде) и ключ сеанса *КS* (в открытом виде). В случае успешного дешифрирования сообщения, можно сказать, что пользователь успешно прошел аутентификацию. Отметим, что *сервер аутентификации AS аутентифицирует пользователя без передачи пароля по сети.*

Квитанция *TTGS*  на доступ к серверу квитанций TGS является подтверждением валидности пользователя и разрешает продолжать процесс получения доступа к ресурсу. Эта квитанция содержит:

* идентификатор пользователя;
* идентификатор сервера квитанций, на доступ к которому выдана квитанция;
* отметку о текущем времени;
* период времени, в течение которого может продолжаться сеанс;
* копию ключа сеанса *KS;*

Таким образом, клиент владеет квитанцией в зашифрованном виде. Благодаря шифрованию можно с уверенностью сказать, что никто, даже сам клиент, обладатель данной квитанции, не сможет квитанцию подделать, подменить или изменить. Только TGS-сервер, получив от клиента квитанцию, сможет ее расшифровать, так как в его распоряжении имеется ключ шифрования К.

Время действия квитанции ограничивается длительностью сеанса. Длительность сеанса пользователя задается администратором и при необходимости можно установить, например, 20 минут, в других условиях (более безопасная сеть) это время может составить и 24 часа. Следовательно, информация, содержащаяся в квитанции, определяет ее срок годности. Предоставление квитанции на определенное время защищает ее от неавторизованного пользователя, который мог бы ее перехватить и использовать в личных целях.

### 1.1.1.2 Получение разрешения на доступ к ресурсному серверу

Следующим этапом для пользователя будет получение разрешения на доступ к ресурсному серверу (например, к файловому серверу или серверу приложений). Для этого нужно получить квитанцию (разрешение на доступ), а такие квитанции выдает TGS-сервер. Для получения доступа к серверу квитанций, у пользователя должна быть квитанция *{TTGS}K*, которую выдает AS-сервер. Для доказательства серверу квитанций, что пользователь имеет право на доступ к ресурсам сети, несмотря на защиту паролем и шифровании, квитанции недостаточно.

Первое сообщение от сервера аутентификации содержит не только квитанцию, но и секретный ключ сеанса *KS*, который разделяется с сервером квитанций (TGS). Клиент с помощью этого ключа шифрует ещё одну электронную форму, которая называется аутентификатором *{A}KS*. Аутентификатор *{A}KS* содержит сетевой адрес и идентификатор пользователя, а также собственную временную отметку. В отличие от квитанции *{TTGS}K*, которую можно использовать многократно в течение сеанса, аутентификатор предназначен для одноразового использования и имеет очень короткое время жизни — обычно несколько минут. Кеrberos-клиент посылает сообщение-запрос серверу квитанций, содержащее квитанцию и аутентификатор: *{TTGS)K, {A}KS.*

Сервер квитанций расшифровывает квитанцию имеющимся у него ключом К, извлекает из нее идентификатор пользователя и проверяет, не истек ли срок действия у квитанции. Затем TGS-сервер расшифровывает аутентификатор, используя ключ сеанса пользователя *Ks*, который был извлечен из квитанции. Далее сервер квитанций сверяет идентификатор пользователя и его сетевой адрес с аналогичными параметрами в квитанции и сообщении. В случае совпадения, сервер квитанций узнает, что данную квитанцию предоставил ее законный владелец.

Заметим, что просто наличие квитанции на получение доступа к серверу квитанций не доказывает идентичности пользователя. Поскольку аутентификатор действителен только в течение короткого промежутка времени, то шанс украсть одновременно и квитанцию, и аутентификатор и использовать их в течение этого времени крайне мал. Всякий раз, когда пользователь обращается к серверу квитанций для получения новой квитанции на доступ к ресурсу, он посылает многоразовую квитанцию и новый аутентификатор. Клиент обращается к серверу квитанций, который обозначен как RS1(см. рис 2), чтобы получить разрешение на доступ к ресурсному серверу. Сервер квитанций, удостоверившись в валидности запроса и личности пользователя, отсылает клиенту ответ, в котором находятся новый ключ сеанса *KS1* и многоразовая квитанция на получение доступа к запрашиваемому ресурсному серверу TRS1 *.*

Квитанция на получение доступа к ресурсу зашифровывается секретным ключом *KRS1*, общим для сервера квитанций и того сервера, к которому предоставляется доступ, в данном случае — RS1 (см. рис 2). На сервере квитанций хранятся уникальные секретные ключи для каждого сервера сети. Эти ключи распределяются между серверами сети физическим способом или каким-либо иным секретным способом при установке системы Kerberos. И в случае передачи квитанций сервером на доступ к какому-либо ресурсному серверу, то сервер шифрует ее, так что только этот сервер сможет расшифровать ее с помощью своего уникального ключа.

Новый ключ сеанса *KS1* содержится не только в самом сообщении, посылаемом клиенту, но и внутри квитанции *TRS1*. Все сообщение шифруется старым ключом сеанса клиента *KS*, так что его может прочитать только этот клиент. Используя введенные обозначения, ответ TGS-сервера клиенту можно представить в следующем виде*: {{TRS1}KRS1, KS1}KS*.

### 1.1.1.3 Получение доступа к ресурсу

После расшифровывания клиентом поступившее сообщение, он отсылает серверу, к которому он хочет получить доступ, запрос, содержащий квитанцию на получение доступа и аутентификатор, зашифрованный новым ключом сеанса:

*{TRS1}KRS1,{ A}KS1*.

Это сообщение обрабатывается аналогично запросу клиента TGS-сервером. Сперва ключом *KRS1* расшифровывается квитанция, после чего извлекается ключ сеанса *KS1* и расшифровывается аутентификатор. Далее содержащиеся в квитанции и аутентификаторе данные о пользователе сравниваются. И доступ к сетевому ресурсу будет разрешен только после того, как проверка пройдет успешно.

На этом этапе клиент перед началом работы с сервером так же может проверить аутентичность сервера. Взаимная процедура аутентификации исключает любую возможность получения доступа к секретной информации от клиента путем подмены сервера неавторизованным пользователем.

Аутентификация ресурсного сервера в технологии Kerberos выполняется следующим образом: клиент посылает запрос к серверу с предложением, чтобы сервер прислал ему сообщение, в котором повторил временную отметку из аутентификатора клиента, но увеличенную на 1. Кроме того, требуется, чтобы данное сообщение было зашифровано ключом сеанса *KS1.*Для выполнения такого запроса клиента, сервер извлекает копию ключа сеанса из квитанции на доступ, расшифровывает аутентификатор с помощью этого ключа, увеличивает на 1 значение временной отметки, заново зашифровывает сообщение с помощью ключа сеанса и возвращает сообщение клиенту. Клиент это сообщение расшифровывает, чтобы получить увеличенную на единицу отметку времени.

В случае успешного завершения вышеописанного процесса клиент и сервер получают ключ сеанса, с помощью которого шифруются будущие сообщения и появляется уверенность в секретности своих транзакций.

### 1.1.1.4 Технология единого входа

Технология единого входа (англ. Single Sign-On (SSO) ) — это технология, благодаря которой возможно воспользоваться различными услугами и ресурсами без повторной аутентификации.

Например, если в сети существует несколько независимых ресурсов (файловый сервер, сетевой принтер, ssh сервер и т. д.) то, после прохождения процедуры аутентификации в каком-либо сервисе, пользователь автоматически может получить доступ ко всем остальным, что позволяет избавить его от многократного ввода данных своей учётной записи.

## 1.1.2 Протокол аутентификации RADIUS

RADIUS – сетевой протокол для решения вопросов аутентификации, авторизации и сбора сведений об использованных ресурсах, разрабатывался специально для системы тарификации использованных ресурсов конкретным пользователем. [2]

Протокол RADIUS основан на клиент-серверной архитектуре. RADIUS сервер хранит список валидных клиентов и таким образом может проверять на подлинность запрашиваемые сервера. Между RADIUS-сервером и RADIUS-клиентами имеется общий ключ. Этот ключ не может быть пустым и используется для проверки подлинности RADIUS сервера и Network Access Server (далее NAS), что позволяет сокрыть пароль от конечного пользователя. Клиентом RADIUS выступает NAS, а сервером RADIUS считается “демон”, работающий на машине UNIX или NT. Если возникает потребность в сетевом ресурсе или услуге, RADIUS-клиент отправляет пользовательскую информацию на определенные RADIUS-серверы, которые в свою очередь отправляют ответ, после чего RADIUS-клиенты действует в соответствии с полученными от сервера указаниями.

Серверы RADIUS принимают запросы от пользователей на подключение, производят аутентификацию пользователей, а затем отправляют конфигурационные данные, которые необходимы клиенту для обслуживания пользователя. Для других серверов RADIUS сервер RADIUS выступает в роли клиента-посредника (proxy).

На рисунке 3 показано взаимодействие между пользователем с одной стороны и клиентом и сервером RADIUS с другой.

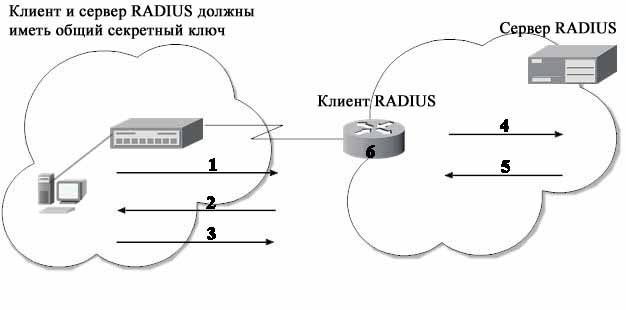


Рис. 3 Взаимодействие между пользователем и системой RADIUS

1. Пользователь инициирует соединение PPP с сервером доступа.
2. Сервер доступа запрашивает у пользователя имя и пароль.
3. Пользователь отвечает на запрос.
4. Клиент RADIUS посылает имя пользователя и зашифрованный пароль серверу RADIUS.
5. Сервер RADIUS отвечает сообщениями Accept, Reject или Challenge.
6. Клиент RADIUS обрабатывает параметры, полученные от сервера вместе с сообщениями Accept или Reject.

Регистрация пользователя состоит из поступающего от NAS на RADIUS-сервер запроса (Access Request), и соответствующего (положительного или отрицательного) ответа от RADIUS-сервера. В пакете Access Request содержится имя пользователя, зашифрованный пароль, IP-адрес системы NAS и соответствующий номер порта.

После того, как RADIUS-сервер получает от NAS запрос Access Request, начинается поиск указанного имени пользователя в базе данных. Если в базе данных такого имени нет, то сервер загружает пользовательский профиль по умолчанию, или же отправляет пользователю отрицательный ответ. В отрицательном ответе могут быть указаны причины отрицательного ответа.

В случае, если же имя пользователя в базе данных нашлось и указанный пароль верный, то RADIUS-сервер выдает положительный ответ, в котором содержится список атрибутов для данной сессии. Типичными атрибутами являются тип услуги (shell или framed), тип протокола, IP-адрес, список объектов к которым разрешен доступ или же статический маршрут, который нужно добавить в таблицу маршрутизации NAS. На рисунке №4 показана процедура аутентификации и авторизации RADIUS.

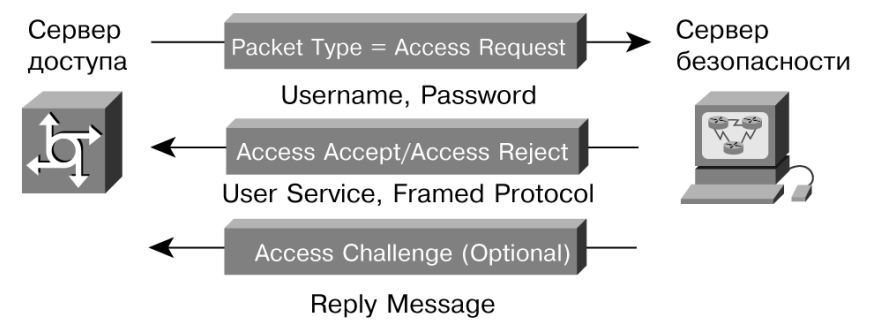


Рис. 4 Процедура идентификации и авторизации RADIUS

С помощью учетных функций RADIUS можно отслеживать количество ресурсов (количество времени, байтов и т.д.) использованных в данной сессии.

Транзакции между клиентом и сервером RADIUS идентифицируются с помощью общего “секрета”, который никогда не передается по сетевым каналам. Кроме того обмен любыми пользовательскими паролями между клиентом и сервером RADIUS производит только в зашифрованном виде.

## 1.1.3 Протокол аутентификации TACACS+

TACACS+ это сетевой протокол последнего поколения серии протоколов TACACS. [2]

TACACS - это простой протокол управления доступом, основанный на стандартах User Datagram Protocol (UDP) и разработанных компанией Bolt, Beranek, and Newman, Inc. для военной сети Military Network (MILNET). Компания Cisco улучшала и расширяла протокол TACACS, в итоге появилась собственная версия протокола TACACS, известная как TACACS+.  
Для своей работы TACACS+ использует транспортный протокол TCP. Демон сервера "прослушивает" порт №49 протокола IP (специально выделенный для протокола TACACS). Этот порт зарезервирован для специальных номеров RFC в протоколах UDP и TCP. Все текущие версии TACACS и расширенные версии этого протокола используют порт № 49.

Протокол TACACS+ основан на клиент-серверной архитектуре, где клиентом TACACS+ выступает Network Access Server (далее NAS), а в роли сервера выступает “демон” (daemon) TACACS+.

Процесс аутентификации между клиентом TACACS+ и сервером происходит с помощью общего ключа, который не передается по каналам связи. Данный ключ администратор задает на сервере и на клиенте вручную. TACACS+ можно настроить таким образом, чтобы весь трафик между клиентом TACACS+ и демоном сервера TACACS+ шифровался. На рисунке 5 изображено взаимодействие клиентом и сервером TACACS+.

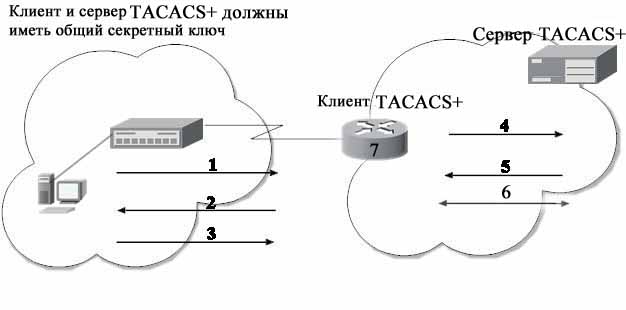


Рис. 5 Взаимодействие между пользователем и системой TACACS+.

1. Пользователь инициирует соединение PPP с сервером доступа.
2. Сервер доступа запрашивает у пользователя имя и пароль.
3. Пользователь отвечает на запрос.
4. Клиент TACACS+ посылает зашифрованный пакет серверу TACACS+.
5. Сервер TACACS+ сообщает результаты идентификации.
6. Клиент и сервер обмениваются авторизационной информацией.
7. Клиент TACACS+ обрабатывает параметры, полученные во время авторизации.

В процессе аутентификации TACACS+ посылает пакеты трех типов: START, CONTINUE и REPLY. START и CONTINUE которые отправляет клиент, а REPLY отправляет сервер.

Аутентификация начинается, когда клиент отправляет серверу сообщение START. В сообщении START описывается тип аутентификации имя пользователя и некоторые идентификационные данные. Пакет START является первым сообщением процесса аутентификации или сразу же после повторного запуска этого процесса. (Повторный запуск может запуститься по запросу сервера, который находится в пакете REPLY). Пакет START всегда имеет порядковый номер равный единице.

В ответ на пакет START сервер высылает пакет REPLY. В сообщении REPLY указывается, закончилась ли аутентификация или ее следует продолжить. Если же пакет REPLY требует продолжения аутентификации, то указывается, какую дополнительную информацию необходимо предоставить. Клиент собирает эту информацию и отправляет ее серверу в сообщении CONTINUE.

По окончании аутентификации клиент может начать процесс авторизации (если она требуется). Сессия авторизации состоит из двух сообщений: сообщения REQUEST (запрос) и следующего за ним сообщения RESPONSE (ответ). Сообщение REQUEST содержит фиксированное количество полей, которые описывают пользователя или процесс, и переменный набор аргументов, которые описывают услуги и опции, требующие авторизации.

На рисунках №6, №7 показаны процессы идентификации и авторизации TACACS+.

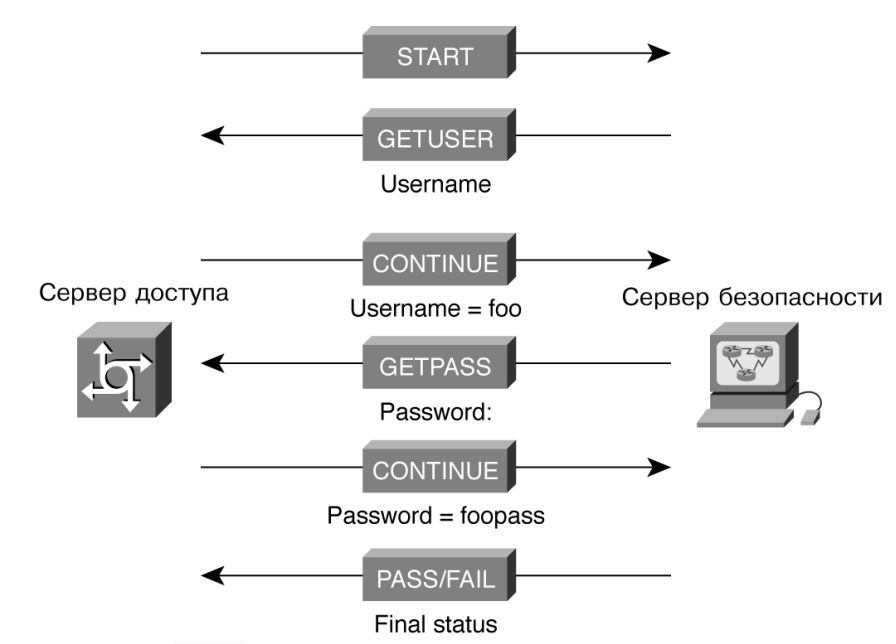
*.*

Рис. 6 Процесс авторизации TACACS+

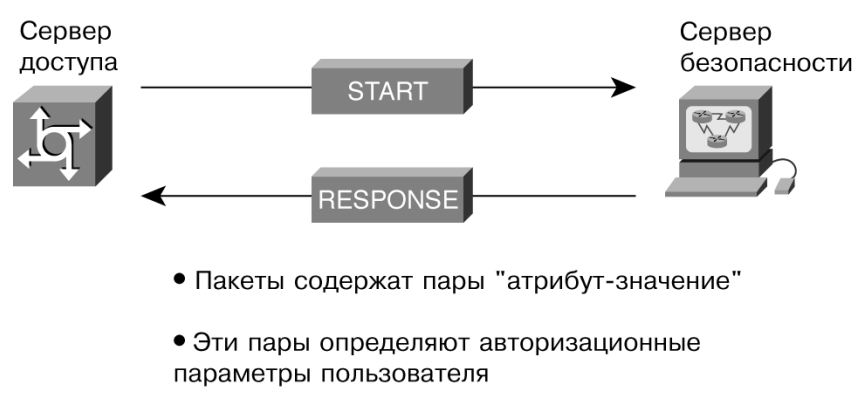


Рис. 7 Процесс идентификации TACACS+

Остановимся на протоколе Kerberos, как на наиболее распространенном и функциональном протоколе.

## 1.2 Выбор системы управления виртуальными машинами

Требования к системе управления виртуальными машинами:

* Свободная лицензия.
* Кроссплатформенность.
* Поддержка образов жёстких дисков VMDK (VMware) и VHD (Microsoft Virtual PC).
* Поддержка различных видов сетевого взаимодействия (например NAT).
* Есть возможность выбора языка интерфейса (поддерживается и русскоязычный интерфейс).

Исходя из личного опыта, а так же таблицы сравнения основных характеристик виртуальных машин, приведенной в Википедии (<http://ru.wikipedia.org/wiki/Сравнение_виртуальных_машин>**),** остановим свой выбор на системе виртуализации **VirtualBox**.

## 1.3 Выбор операционной системы

Практически все современные ОС поддерживают работу в сети. Однако в качестве ОС для настольных компьютеров чаще всего выбираются Linux и Windows семейства операционных систем. Лицензионная политика Microsoft характеризует правила приобретения и использования её программного обеспечения. Лицензирование отдельных продуктов зависит от корпоративного или частного использования и характеризуется рядом характеристик: требование активации, запрет на распространение, запрет на создание определённого числа копий. Таким образом будем рассматривать ОС свободного распространения.

### 1.3.1 Операционная система Debian

Debian — операционная система, состоящая из свободного ПО с открытым исходным кодом в основном имеющие GNU General Public License Debian или Debian GNU / Linux

Debian может эксплуатироваться в качестве и операционной системы для серверов, так и в качестве рабочих станций.

*Особенности*

Операционная система Debian заметна своими широкими возможностями.

Например, в текущую стабильную версию включено более двадцати тысяч пакетов программ для десяти архитектур на основе ядра Linux (от Intel/AMD 32-bit/64-bit, до ARM,) а так же двух архитектур на основе ядра FreeBSD (kfreebsd-i386 and kfreebsd-amd64).

Отличительными чертами Debian являются: система управления пакетами Advanced Packaging Tool (APT), серьёзная политика к пакетам, огромное количество репозиториев, а также высокое качество выпускаемых версий. Это сделало возможным простое обновление между версиями, а также автоматическую установку и удаление пакетов. Именно в Дебиане впервые был введён единый стандартный механизм выбора предпочтительного ПО среди нескольких вариантов — Alternatives.

При стандартной установке Debian используется среда рабочего стола GNOME, куда включён набор популярных программ, таких как LibreOffice, Iceweasel (модификация Firefox), почтовая программа Evolution, программы для записи CD/DVD, проигрыватели музыки и видео, программы для просмотра и редактирования изображений и программы для просмотра документов в формате PDF.

### 1.3.2 Операционная система Ubuntu

Ubuntu — операционная система, основанная на Debian. В настоящее время проект активно развивается и поддерживается свободным сообществом.

Является самым популярным дистрибутивом Linux (на момент 26 апреля 2011 года) для десктопов. Он является 4-м в списке самых популярных ОС для веб-серверов и его популярность быстро растёт.

*Особенности*

Ubuntu состоит из множества программных пакетов, большинство из которых распространяется по свободной лицензии. Основная используемая лицензия GNU General Public License (GNU GPL), которая, наряду с GNU Lesser General Public License (GNU LGPL) заявляет, что пользователи могут свободно запускать, копировать, распространять, изучать, изменять, развивать и совершенствовать программное обеспечение

Направленность политики Ubuntu на конечного пользователя позволяет устанавливать Ubuntu на жесткий диск средствами Live CD без потребности в перезапуске компьютера до установки. Начиная с версии 5.04, UTF-8 стал кодировкой символов по умолчанию. Как средство защиты, команда sudo используется, чтобы присвоить временные полномочия суперпользователя для того, чтобы выполнить задачи администрирования, позволяя корневой учетной записи остаться заблокированным, и предотвращая неопытных пользователей от непреднамеренного внесения катастрофических изменений или неявного использования возможных «дыр» в системе безопасности. Рабочий стол Ubuntu включает графическую среду. В версиях до 11.04 GUI по умолчанию была панель GNOME, но это было отброшено в пользу Unity.

После установки Ubuntu может порадовать пользователя своим широким диапазоном программного обеспечения, которое включает LibreOffice, Firefox, Thunderbird, Empathy, Transmission, и несколько легких игр (таких как Судоку и шахматы). Дополнительное программное обеспечение, которое не установлено по умолчанию (включая программное обеспечение, которое использовалось ранее в установке по умолчанию, такие как Evolution, GIMP, Pidgin, and Synaptic) может быть загружено и устанавлено из Центра программного обеспечения Ubuntu. Программы в Центре программного обеспечения главным образом бесплатные, но также есть и платные продукты, включая приложения и журналы.

Ubuntu может также использовать множество программ, разработанных для Microsoft Windows (таких как Microsoft Office) через Wine или использование Виртуальных машин (таких как Рабочая станция VMware или VirtualBox).

### 1.3.3 Операционная система Arch Linux

Arch — «легковесный», простой и гибкий дистрибутив Linux, оптимизированный для архитектур i686 и x86-64, использующий последние стабильные версии программ и дополняемый поддерживаемым сообществом репозиторием AUR.

Arch «будет тем, что вы из него сделаете», и рассчитан не на новичков, а на более опытных пользователей, т.е. подход к проектированию группы разработчиков фокусируется на простоте с точки зрения разработчика, а не точки зрения пользователя, элегантности, правильности кода и минимализма.

Arch Linux использует роллинг-релиз (плавающий релиз), потому что регулярное системное обновление – это все, что необходимо, чтобы получить последнее программное обеспечение Arch.

### 1.3.4 Операционная система Fedora

Fedora – операционная система, основанная на ядре Linux, принадлежит компании Red Hat, однако Red Hat не осуществляет поддержку пользователей дистрибутива Fedora, поддержка осуществляется открытым сообществом. Проект служит для тестирования новых технологий, которые в дальнейшем включаются в продукты Red Hat и других производителей.

Цель проекта Fedora — повысить продвижение бесплатного программного обеспечения и программного обеспечения с открытым исходным кодом, а так же свободного контента. У A версии Fedora относительно короткий жизненный цикл — интервал техобслуживания составляет только 13 месяцев: между выпусками есть 6 месяцев, и версия X поддерживается только до спустя 1 месяц после версии X+2. Это помогает продвигать передовое программное обеспечение, потому что оно освобождает разработчиков от некоторых ограничений обратной совместимости, но оно также делает Fedora плохим выбором для разработки продукта (например, встроенные системы), который обычно требует долгосрочной поддержки поставщика, недоступной с любой версией Fedora.

Таким образом, исходя из поставленных задач и личного опыта использования, остановимся на операционной системе Ubuntu.

## 1.4 Выводы по разделу

В этом разделе описывается обзор протоколов аутентификации и авторизации пользователей. Из рассмотренных протоколов был выбран протокол аутентификации и авторизации Kerberos. Был произведен обзор и выбор операционной системы и системы виртуализации. Выбрана операционная система Ubuntu, а система виртуализации – VirtualBox.

# 2. Разработка

## 2.1 Разработка методики работы со стендом

Стенд разрабатывается для специалистов в области сетевой безопасности с целью повышения уровня подготовки таких специалистов.

Специалисты в области сетевой безопасности ценятся благодаря большому практическому опыту, а это означает, что нужно постоянно, ежедневно практиковаться. Отсюда возникает требование к созданию такого стенда, с которым можно будет работать в любое удобное время и в удобном месте. Таким образом, стенд должен быть простым в управлении и настройке, кроссплатформенным, удобным в использовании, понятным даже для простых пользователей, решивших стать специалистом в области сетевой безопасности.

Поскольку появляется необходимость работы со стендом в домашних условиях (для возможности постоянной работы), то это означает, что проектируемый стенд должен учитывать аппаратные части домашних компьютеров, которые зачастую на порядок ниже аппаратных возможностей современный серверов или же современных рабочих станциях.

Так же необходимо предоставить выбор пользователям, устанавливать стенд на компьютере или же не обременять себя установкой и подключиться к удаленному учебному серверу с уже установленным и готовым к эксплуатации стендом.

Принципиальная схема работы с удаленным учебным сервером проиллюстрирована на рис. 1.

Условно схему можно разделить на 3 части, рассмотрим каждую из них.

* Learn server – Учебный сервер, на котором развернута управляющая система обучением Мoodle с учебными пособиями, система виртуализации и виртуальные машины.
* Интернет - как средство связи до удаленного учебного сервера
* Host – Рабочая машина, например домашний компьютер студента.

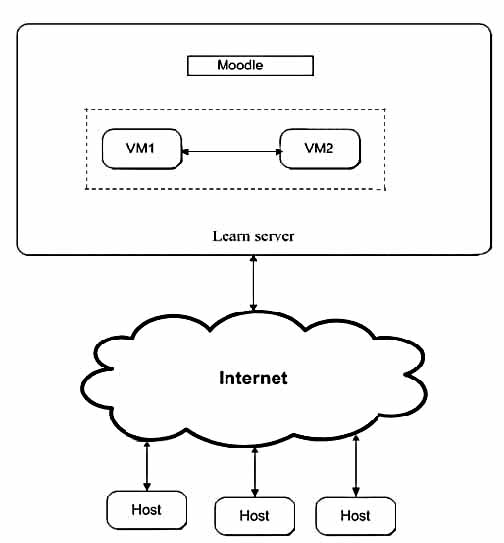


Рис. 8 Схема работы с удаленным учебным сервером

Поскольку для реализации такой схемы работы с удаленным учебным сервером необходим сервер, соответствующие права доступа к нему, то будем разрабатывать и реализовывать «домашний» вариант использования стенда, который в свою очередь является компонентом работы с удаленным учебным сервером.

## 2.2 Разработка структуры виртуальной среды

Разрабатываемая виртуальная среда должна быть аналогична реальной сети, в которой можно установить протокол аутентификации Kerberos.

Поскольку протокол Kerberos основан на клиент-серверной архитектуре, то в базовом случае виртуальная среда будет состоять из двух виртуальных машин, роль которых будет Kerberos-сервер, Kerberos-клиент.

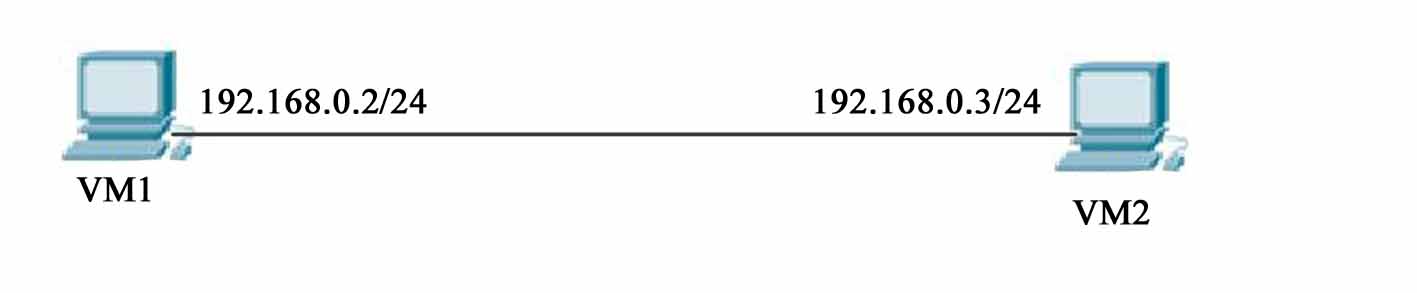


Рис. 9 Схема сети виртуальных машин

В качестве примера, первая виртуальная машина имеет название VM1 и IP-адрес 192.168.0.2 c маской подсети /24 (255.255.255.0), а вторая виртуальная машина имеет название VM2 и IP-адрес 192.168.0.3 c маской подсети /24 (255.255.255.0).

## 2.3. Выводы по разделу

В этом разделе были разработаны структура стенда. Разработана структура виртуальной среды.

# 3. Реализация

## 3.1 Установка системы управления виртуальными машинами

Для установки системы управления виртуальными машинами VirtualBox, необходимо перейти по ссылке <https://www.virtualbox.org/wiki/Downloads> и выбрать нужную версию для своей операционной системы. В нашем случае для Ubuntu, нужно выбрать раздел VirtualBox for Linux hosts, далее выбрать нужную версию Ubuntu и скачать. Производится типичная установка и комментироваться не будет.



Рис. 10 Страница в интернете <https://www.virtualbox.org/wiki/Downloads>

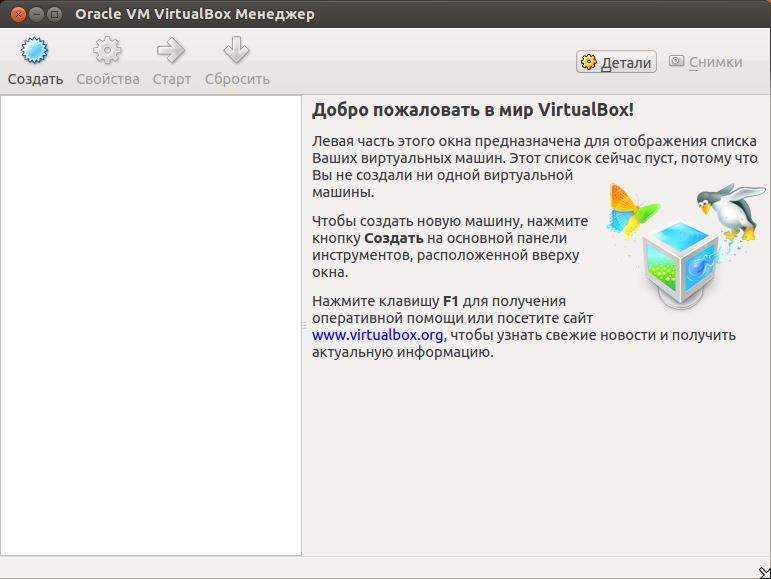


Рис. 11 Снимок запущенной программы Virtualbox

## 3.2Создание и настройка виртуальной машины

Для того чтобы создать виртуальную машину, нужно нажать на кнопку «Создать» (см. рис.11) и следовать указаниям.

Первым делом нужно указать тип и имя операционной системы.

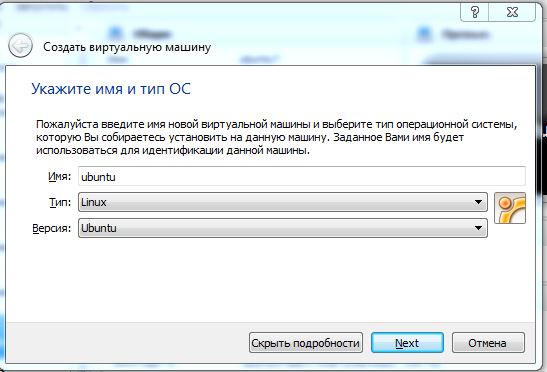


Рис. 12 Выбор имени и типа операционной системы в Virtualbox

Выбираем объем оперативной памяти виртуальной машины. Выберем рекомендуемый объем оперативной памяти 512 МБ, если аппаратная часть позволяет это сделать.

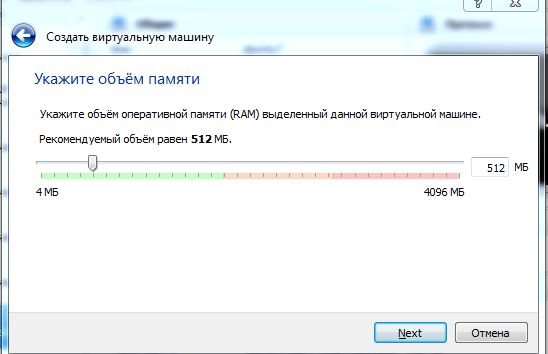


Рис. 13 Выбор объема оперативной памяти в Virtualbox

Создаем жесткий диск.

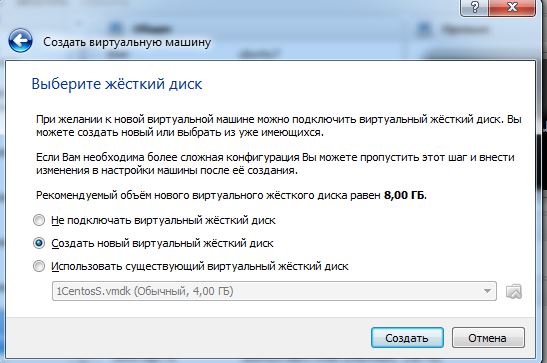


Рис. 14 Выбор жесткого диска в Virtualbox

Далее укажем тип жесткого диска

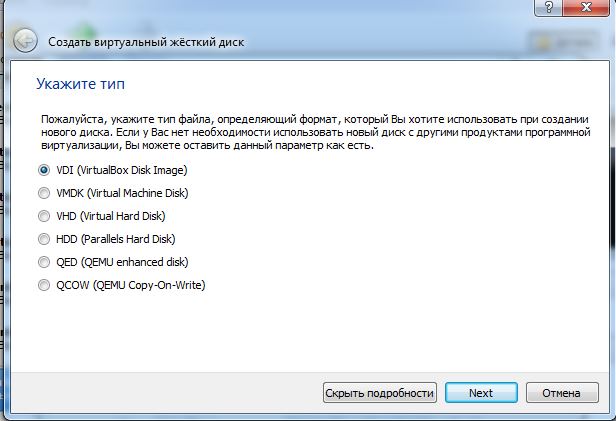


Рис. 15 Выбор тип жесткого диска в Virtualbox

Укажем формат хранения данных

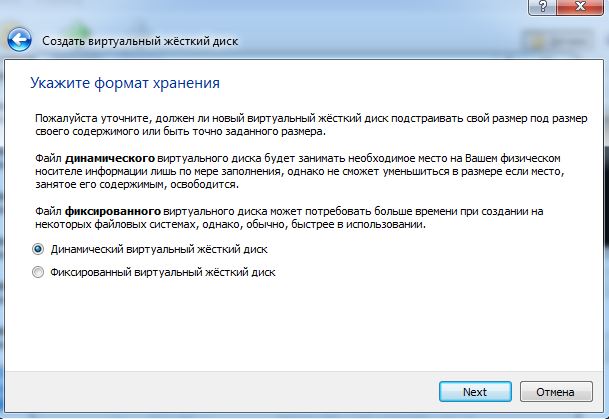


Рис. 16 Выбор формата жесткого диска в Virtualbox

Установим имя и размер файла. Выбираем рекомендуемый объем жесткого диска, аппаратная часть позволяет это сделать.

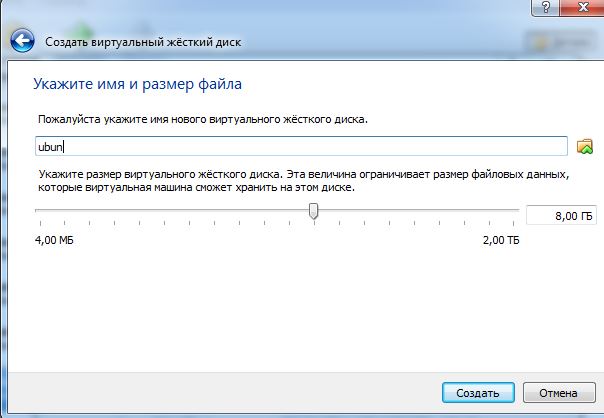


Рис.17 Выбор объема жесткого диска в Virtualbox

Теперь **установим операционную систему** на виртуальные машины.

Найти актуальную версию операционной системы можно на сайте [www.ubuntu.ru](http://www.ubuntu.ru).

Установка операционной системы является типовой и комментироваться не будет.

В данном случае была загружена и установлена Ubuntu 12.10.

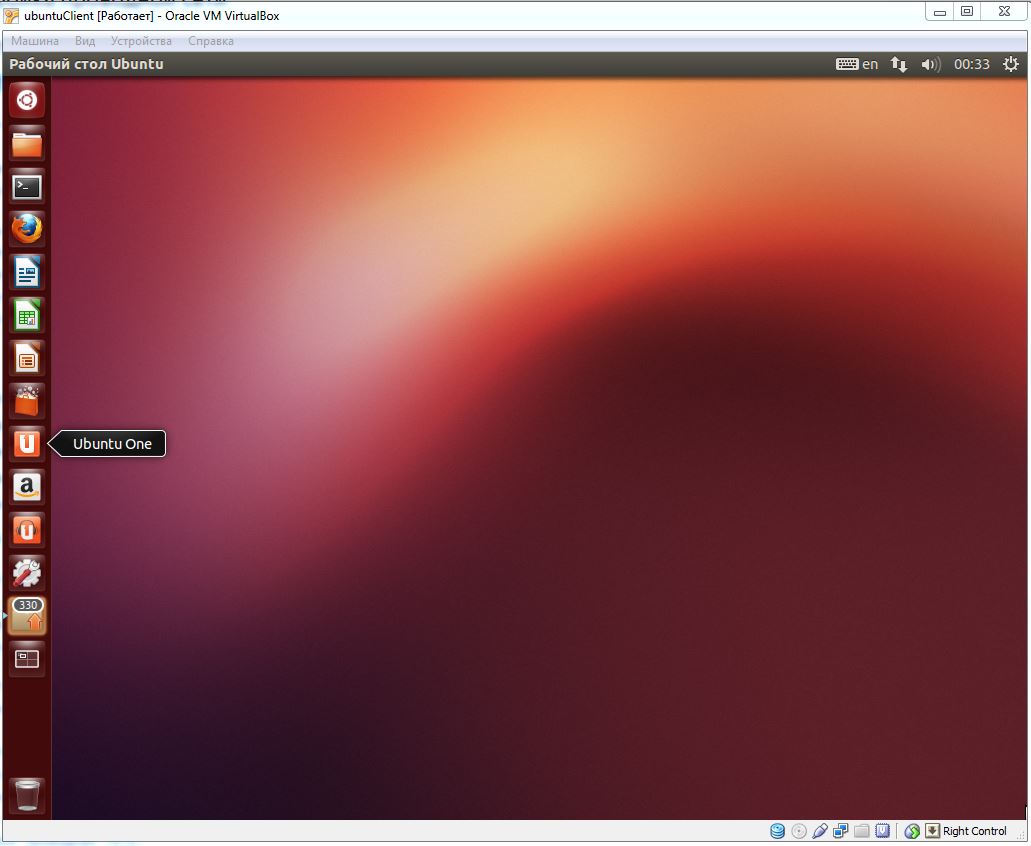


Рис. 18 Рабочий стол операционной системы Ubuntu 12.10

## 3.3 Настройка сетевых интерфейсов виртуальных машин

Необходимо настроить сетевые интерфейсы таким образом, чтобы настройки были работоспособными и после перезагрузки виртуальной машины, а так же чтобы виртуальная машина имела доступ в Интернет.

Для этого, в первую очередь нужно перевести режим сетевой карты виртуальной машины в режим «сетевой мост».

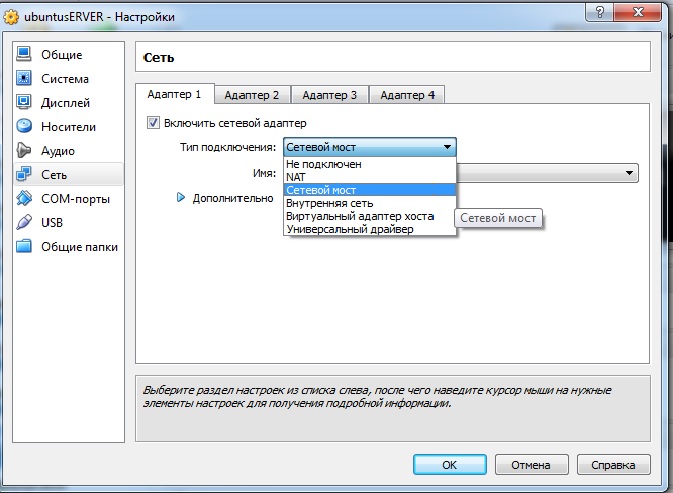


Рис. 19 Выбор типа подключения сетевого адаптера

Настроим IP-адрес, маску подсети.

Для этого воспользуемся любым доступным текстовым редактором и отредактируем файл конфигурации /etc/network/interfaces, например так:

# sudo nano /etc/network/interfaces

Допишем в него:

Для статического IP-адреса:

iface eth0 inet static

address 192.168.0.2

netmask 255.255.255.0

gateway 192.168.0.254

auto eth0

Где:

* iface eth0 inet static - указывает, что интерфейс (iface eth0) находится в диапазоне адресов IPv4 (inet) со статическим ip (static);
* address 192.168.0.1 - указывает что IP-адрес (address) сетевой карты 192.168.0.1;
* netmask 255.255.255.0 - указывает что маска подсети (netmask) имеет значение 255.255.255.0;
* gateway 192.168.0.254 - адрес шлюза (gateway) по умолчанию 192.168.0.254;
* auto eth0 - указывет системе что интерфейс eth0 необходимо включать автоматически при загрузке системы с вышеуказанными параметрами.
* eth0 - имя подключаемого своего интерфейса.

Список интерфейсов можно посмотреть командой:

# ifconfig –a

В итоге файл /etc/network/interfaces должен выглядеть примерно так:

(для одного проводного соединения со статическим IP-адресом)

# This file describes the network interfaces available on your system

# and how to activate them. For more information, see interfaces(5).

# The loopback network interface

auto lo

iface lo inet loopback

# Моя сеть.

iface eth0 inet static

address 192.168.0.1

netmask 255.255.255.0

gateway 192.168.0.254

auto eth0

Пример конфигурации для динамического IP-адреса:

iface eth0 inet dhcp

auto eth0

## 3.4 Установка и настройка клиент-серверных компонентов протокола Kerberos

Термины, которые необходимо знать перед установкой Kerberos:

* **Учетная запись (Principal):** любые пользователи, компьютеры или сервисы, предоставляемые серверами, должны быть определены как учетные записи Kerberos .
* **Требования (Instances):** используются для сервисных и специальных административных учетных записей.
* **Области (Realms):** уникальная область управления, обеспечиваемая установкой Kerberos. Представляйте ее себе как домен или группу ваших компьютеров и пользователей, ей принадлежащих. По умолчанию Ubuntu использует имя DNS домена в верхнем регистре (EXAMPLE.COM) в качестве имени области.
* **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.
* **Билет для получения билета (TGT):** изданный сервером аутентификации, TGT зашифровывается на пароле пользователя, который известен только пользователю и KDC.
* Сервер распространения билетов (TGS): выпускает сервисные билеты для клиентов по запросу.
* **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.
* **Файлы ключей (Keytab Files)**: файлы, извлеченные из базы учетных записей KDC и содержащие ключ шифрования для сервиса или компьютера.

Для корректной работы протокола Kerberos, необходимо задать соответствующую область, в которой будет работать протокол, а так же указать имена хостов на которых будет установлен Kerberos.

Пусть областью будет MIEM.RU, а хостами будут kdcbox.miem.ru, client1.miem.ru.

* kdcbox.miem.ru –будет выполнять роль Kerberos -сервера
* client1.miem.ru - будет выполнять роль Kerberos – клиента

**Начнем настройку с сервера.**

Для настройки Kerberos-сервера нужно выполнить следующие шаги:

* Определить область (realm), относящиеся к нему DNS-имена и отвечающие за него имена
* Настроить KDC (Key Distribution Server)
* Добавить принципал администратора
* Добавить принципал для клиента
* Добавить принципала для пользователя

В качестве сервера выступает виртуальная машина VM1(см.рис. 9).

Далее все действия будут производиться от пользователя root. Для того чтобы стать пользователем root, наберем в терминале:

# sudo su

Для начала необходимо изменить hostname по умолчанию на нужное нам имя.

Для этого отредактируем файл конфигурации любым доступным текстовым редактором /etc/hosts например так:

# nano /etc/hosts

И заменим значение по умолчанию на нужное нам:

kdcbox.miem.ru

Перезагрузим виртуальную машину для того чтобы настройки пришли в силу.

Kerberos - протокол, зависимый от времени. Поэтому если локальное время системы на клиентской машине и на сервере отличается более чем на 5 минут (по умолчанию), рабочая станция не будет аутентифицирована. Для решения проблемы все узлы сети должны синхронизировать свое время по одному серверу NTP.

**NTP** - это протокол синхронизации времени по сети. По существу клиенты запрашивают текущее время на сервере и используют его для установки своих собственных часов.

Ubuntu стандартно устанавливается с ntpdate и будет запускать его при каждой загрузке один раз для установки времени по NTP серверу Ubuntu.

ntpdate -s ntp.ubuntu.com

Теперь приступим непосредственно к установке серверных компонент протокола Kerberos.

Для работы Kerberos-сервера необходимы следующие серверные компоненты:

* krb5-kdc – необходим для работы Authentication Service (AS) и kdc (Key Distribution Center) (см. раздел 1.1.1)
* krb5-admin-server – необходим для работы с учетными записями Kerberos

Для их установки воспользуемся следующей командой:

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

В процессе установки нужно указать сетевые имена управляющего и административного серверов Kerberos, которые могут быть одним и тем же или разными серверами для определенной области.

Сначала зададим область по умолчанию. В нашем случае это будет MIEM.RU.



Рис. 20 Установка Kerberos

Далее нужно ввести имя сервера Kerberos для нашей зоны MIEM.RU, в нашем случае это будет kdc.miem.ru

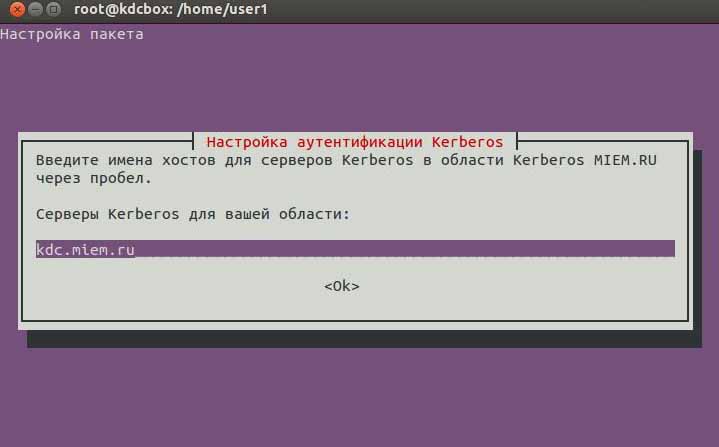


Рис. 21 Установка Kerberos

Введем имя хоста, ответственного за пользователей области Kerberos, в нашем случае это будет kdc.miem-test.ru.

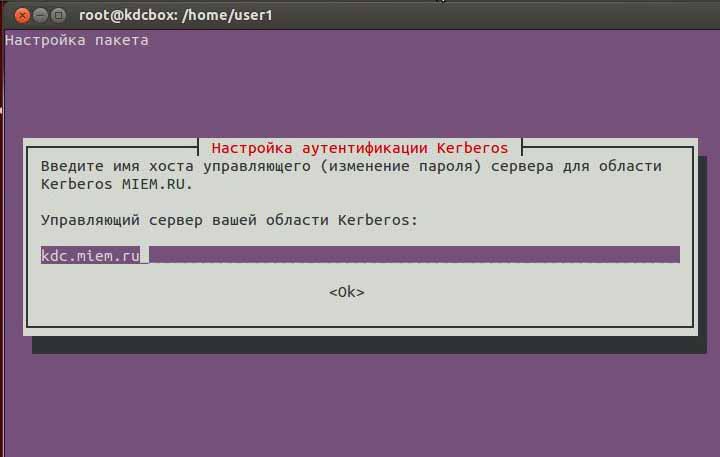


Рис. 22 Установка Kerberos

При установке пакета не выполняется автоматическая настройка области Kerberos.

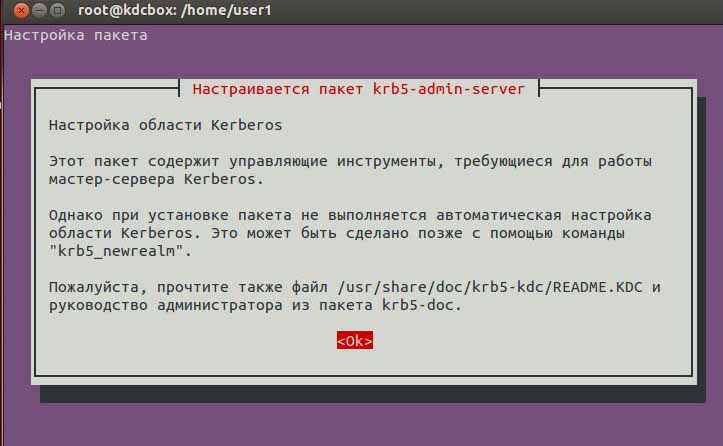


Рис. 23 Установка Kerberos

Поскольку автоматической настройки области при установке компонентов Kerberos не произошло, то настроим область вручную с помощью утилиты kdb5\_newrealm:

Для этого в терминале введем:

# krb5\_newrealm

Потребуется ввести пароль, который будет использоваться для генерации мастер – ключа для базы данных и который хранится в /etc/krb5kdc/stash. Если пароль будет утерян, то базы данных Kerberos уже не расшифровать, поэтому нельзя допустить потери пароля.

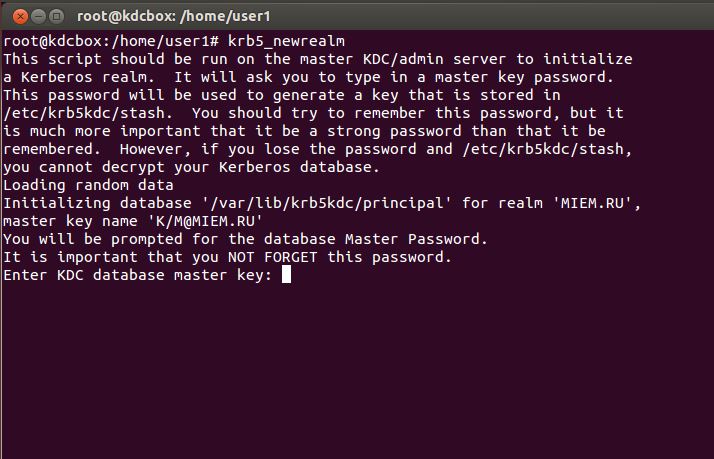


Рис. 24 Работа утилиты krb5\_newrealm

### 3.4.1 Настройка конфигурационных файлов протокола Kerberos

Конфигурационные файлы протокола Kerberos:

* /etc/krb5.conf – файл, отвечающий за области (realms) Kerberos.
* /etc/krb5kdc/kdc.conf – файл, отвечающий за настройку KDC (Key Distribution Server).

Вопросы, задаваемые в процессе установки, используются для настройки файла /etc/krb5.conf. Если требуется перенастроить Kerberos сначала, возможно для изменения имени области, моэно это сделать набрав в терминале команду:

# dpkg-reconfigure krb5-kdc

Рассмотрим конфигурационный файл /etc/krb5.conf (см. приложение 1). Он состоит из следующих секций:

* [libdefaults]
* [realms]
* [domain\_realm]
* [login]

Проанализируем каждую из них отдельно.

**[libdefaults]**

Указываются основные настройки работы Kerberos.

* default\_realm - Определяет область по умолчанию Kerberos для клиента. В нашем случае это MIEM.RU
* kdc\_timesync - Допустимые значения равны 1 или 0. Если значение отлично от нуля, клиентские машины будут вычислять разницу между их временем и временем сервера. Она будет использоваться в билетах для коррекции системных часов при запросе к услугам. Это актуально только для Kerberos, не влияет на изменение системных часов.
* ccache\_type - Этот параметр определяет формат кэша учетных данных, который создается утилитой kinit. Значение по умолчанию 4. Меньшие значения могут быть использованы для совместимости с очень старыми реализациями протокола Kerberos, которые взаимодействуют с учетными данными кэша на том же хосте.
* default\_tgs\_enctypes - Определяет список поддерживаемых типов сессионного ключа шифрования, который клиент должен запросить при принятии TGS-REQ, в порядке предпочтения от самого высокого до самого низкого. Список может быть разделен запятыми или пробелами.
* default\_tkt\_enctypes - Определяет список поддерживаемых типов с шифрования, который клиент должен запросить при принятии AS-REQ, в порядке предпочтения от самого высокого до самого низкого. Список может быть разделен запятыми или пробелами.
* permitted\_enctypes - Определяет все типы шифрования, которые разрешены для использования в сессии типов шифрования.

**[realms]**

Каждый тег в [realms] это имя области Kerberos. Значение тега является подразделом с отношениями, которые определяют свойства данной области.

* admin\_server - Определяет хост, где находится административный сервер. Как правило, это главный сервер Kerberos.
* kdc - Имя хоста с KDC для данной зоны.
* default\_domain - Задает домен, который используется для расширения имен хостов при переводе Kerberos 4 к Kerberos 5.

**[domain\_realm]**

Раздел содержит перевод доменного имени или имя хоста в имя области Kerberos.

**[login]**

Совместимость с Kerberos версией 4.

Устанавливаем все значения в false.

Отредактируем конфигурационный файл /etc/krb5.conf .

Для этого введем в терминале команду

# gedit /etc/krb5.conf

На рисунке 25 указаны новые настройки конфигурационного файла /etc/krb5.conf

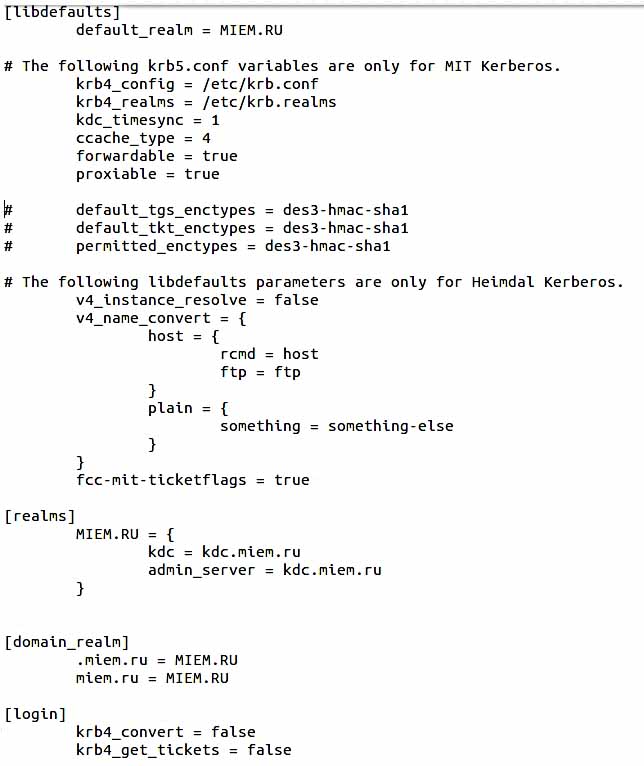


Рис. 25 Новые настройки конфигурационного файла /etc/krb5.conf

Рассмотрим конфигурационный файл /etc/krb5kdc.kdc.conf.

Он состоит из следующих секций:

* [kdcdefaults] - содержит параметры, которые контролируют общую работу KDC
* [realms] - Содержит подразделы области Kerberos, которые описывают параметры KDC на данную область.

Проанализируем каждую из них.

[kdcdefaults]

kdc\_ports - порт, который устанавливается по умолчанию для работы сервера Kerberos.

[realms]

* database\_name Определяет расположение базы данных Kerberos для этой области.
* admin\_keytab Определяет расположение keytab файла
* acl\_file Эта строка определяет расположение списка управления доступом (acl)
* kdc\_ports Эта строка определяет список портов, которые использует KDC. По умолчанию значение kdc\_ports как определено в секции [kdcdefaults].
* max\_life Определяет время жизни билета
* max\_renewable\_life Определяет время, в течене которого билет может быть возобновлен.
* supported\_enctypes – поддерживаемые типы шифрования.

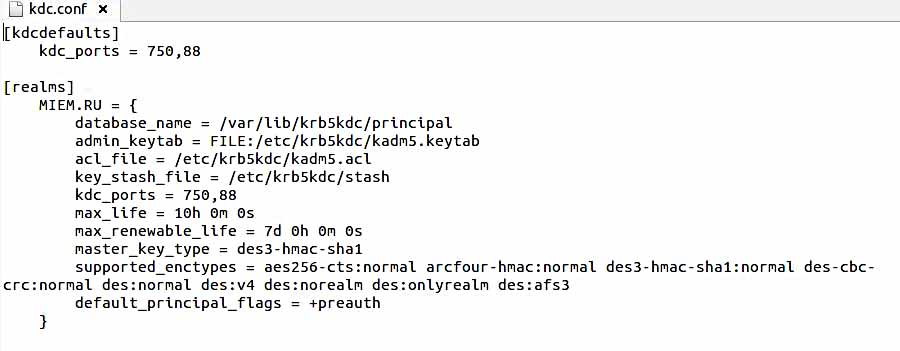


Рис. 26 Конфигурационный файл /etc/krb5kdc/kdc.conf

### 3.4.2 Пользователи Kerberos

Необходимо добавить административного пользователя (учетная запись администратора). Рекомендуется использовать имя пользователя, отличное от повседневного пользователя. Для этого воспользуемся утилитой kadmin.local:

# kadmin.local

# kadmin.local : addprinc admin01/admin

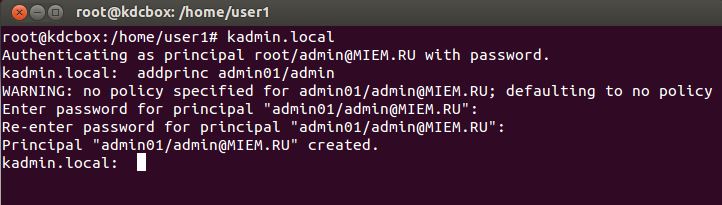


Рис. 27 Добавление административного пользователя с помощью утилиты kadmin.local

Таким образом создали административного пользователя, где admin01 - учетная запись, /admin – параметр, означающий что у пользователя admin01 административные права, а @MIEM.RU - определяет область.

Далее, нового пользователя-администратора нужно вписать в соответствующий список ACL, который настраиваются в файле /etc/krb5kdc/kadm5.acl.

Допишем в список следующую строчку:

admin01/admin@miem.ru \*

Важно !

* Обязательно установить символ звездочки «\*» в конце строчки, чтобы учетная запись администратора вступила в силу.
* Раскомментируйте строчку \*/admin \*

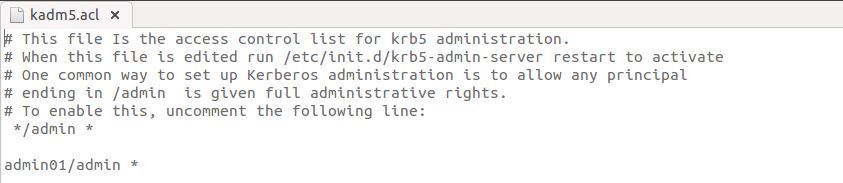


Рис. 28 Измененный список доступа kadm5.acl

Эта запись предоставляет для пользователя admin01/admin возможность выполнять любые операции над любыми учетными записями в этой (MIEM.RU) области. Так же мы можно настроить учетные записи с более ограниченными правами, которые удобны если нам потребуется учетная запись младшего администратора, которую можно использовать на клиентах Kerberos.

### 3.4.3 Настройка DNS

Настроим файл /etc/hosts.

hosts — текстовый файл, содержащий базу данных доменных имен и используемый при их трансляции в сетевые адреса узлов. Запрос к этому файлу имеет приоритет перед обращением к DNS-серверам. В отличие от DNS, содержимое файла контролируется администратором компьютера.

Для начала необходимо свой узнать IP-адрес, для этого наберем в терминале:

# ifconfig -a

В нашем случае IP-адрес: 192.168.0.102.

После того, как узнали IP-адрес, отредактируем файл hosts.

Для этого наберем в терминале:

# gedit /etc/hosts

Добавим в файл следующие строки:

192.168.0.102 miem.ru

192.168.0.102 kdc.miem.ru

192.168.0.102 kdcbox.miem.ru kdcbox

Важно !

* В этот файл дописываются все IP-адерса Kerberos-клиентов и их доменные имена.

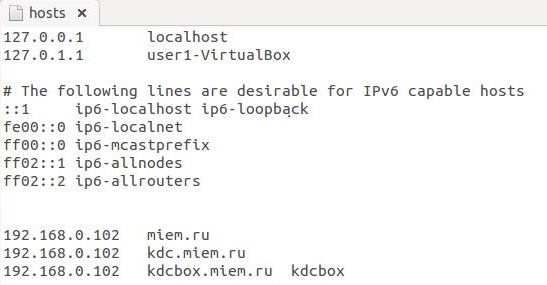


Рис. 29 Настроенный файл hosts

## 3.5 Технология единого входа Single Sign On (SSO)

Поскольку протокол Kerberos является частью технологии единого входа (Single Sign On - SSO), то продемонстрируем это на примере ssh.

SSH (Secure Shell) —это специальный сетевой протокол, позволяющий получать удаленный доступ к компьютеру.

### 3.5.1 Установка SSH

Установить SSH можно из терминала командой:

# apt-get install ssh

### 3.5.2 Настройка сервера SSH

Настроим SSH сервер таким образом, чтобы аутентификацияпользователя происходила средствами протокола Kerberos.

Отредактируем конфигурационный файл /etc/ssh/sshd\_config, для этого наберем в терминале:

# gedit /etc/ssh/sshd\_config

найдем, раскомментируем и изменим следующие строчки:

KerberosAuthentication yes

GSSAPIAuthentication yes

GSSAPIDelegateCredentials yes

Далее необходимо создать файл krb5.keytab.

Для этого наберем в терминале:

# kadmin.local

# kadmin.local: addprinc root

# kadmin.local: addprint -randkey host/miem.ru

# kadmin.local: ktadd host/miem.ru

# kadmin.local: quit

На этом настройка серверных компонентов протокола Kerberos завершена.

## 3.6 Настройка клиента протокола Kerberos

Для настройки Kerberos-клиента нужно выполнить следующие шаги:

* Определить realm, относящиеся к нему DNS-имена и отвечающие за него сервера
* Включить Kerberos в ssh

В качестве клиента выступает виртуальная машина VM2 (см. рис. 9).

Далее все действия будут производиться от пользователя root. Для того чтобы стать root пользователем, наберите в терминале:

# sudo su

Для начала необходимо изменить hostname по умолчанию на нужное нам имя.

Для этого отредактируем файл конфигурации /etc/hosts например так:

# nano /etc/hosts

Заменим значение по умолчанию на нужное нам (см. раздел 3.4):

client1.miem.ru

Перезагрузим виртуальную машину для того чтобы, настройки вошли в силу.

Далее необходимо установить и настроить клиентские компоненты протокола Kerberos.

Так же необходимо настроить синхронизацию по времени, как и в случае с сервером.

Введем в терминале:

# ntpdate -s ntp.ubuntu.com

Теперь приступим непосредственно к установке клиентских компонентов протокола Kerberos.

Для этого введем в терминале:

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

В процессе установки, так же как и в случае с сервером запросят сетевые имена управляющего и административного серверов Kerberos, которые могут быть одним и тем же или разными серверами для определенной области.

Настройки клиента на данном этапе аналогичны настройкам сервера.

Далее настроим конфигурационный файл Kerberos.

Для этого введем в терминале команду:

# gedit /etc/krb5.conf

Настройки аналогичные серверным, поэтому комментироваться не будут.

### 3.6.1 Настройка DNS

Как и в случае настройки DNS сервера (см. раздел 3.4.3), отредактируем файл /etc/hosts, так же как и в случае с сервером.

Наберем в терминале:

# gedit /etc/hosts

Добавим необходимые записи.

Итоговый файл hosts можно увидеть на рис. 30

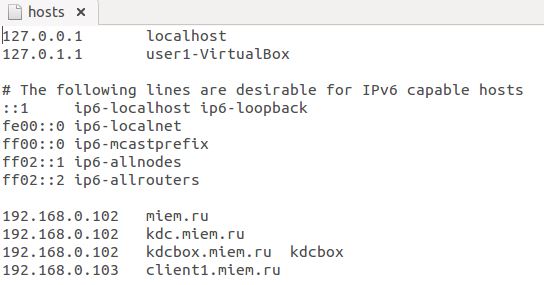


Рис. 30 Файл hosts

Стоить отметить, что настройки файла hosts, кроме тех что по умолчанию (хотя они могут быть и одинаковыми), идентичны.

### 3.6.2 Настройка SSH-клиента

Для реализации технологии единого входа Single Sign On (SSO) на примере ssh, необходимо отредактировать файл /etc/ssh/ssh\_config.

Для этого наберем в терминале:

gedit /etc/ssh/shh\_config

найдем, изменим и раскоментируем следующие строки:

GSSAPIAuthentication yes

GSSAPIDelegateCredentials yes

## 3.7 Выводы по разделу

В данном разделе были установлены и настроены серверные и клиентские компоненты протокола Kerberos на сервер и клиент соответственно. Настроены сетевые интерфейсы сервера и клиента. Добавлены пользователи Kerberos. Настроены DNS на клиенте и сервере. Реализована технология единого входа SSO на примере ssh.

# 4 Экспериментальная часть

## 4.1 Разработка методики испытания

### 4.1.1 Объект испытаний

Объектом испытаний является виртуальная среда для освоения протоколов распределенной аутентификации и авторизации пользователей.

### 4.1.2. Цель испытаний

Целью испытаний является проверка правильности работы протокола Kerberos, технологии единого входа SSO.

### 4.1.3. Требования к программе

Список требований, для проверке во время испытаний:

Для проверки работоспособности программы будут проведены следующие испытания:

* получить билет для получения билетов (TGT).
* запросить билет для сервисов, поддерживающих Kerberos, на сервере распространения билетов (TGS).
* Пройти процесс аутентификации без ввода имени и пароля.

### 4.1.4. Средства и порядок испытаний

Для проведения испытаний необходимо:

* Компьютер с ОС Windows, Linux.
* Установленная система виртуализации VirtualBox.
* Запущенные и настроенные виртуальные машины VM1 и VM2 ( роли которых Kerberos-сервер и Kerberos-клиент соответственно).

### 4.1.5. Методы испытаний

В таблице 1 описаны методы испытаний, которым будет подвержена виртуальная среда для освоения протоколов распределенной аутентификации и авторизации пользователей.

Таблица 1: Методы испытаний

|  |  |  |
| --- | --- | --- |
| Описание функции | Предполагаемые результат | Полученный результат |
| Командой kinit запросить билет для получения билетов (TGT). | Получить билет для получения билетов (TGT). |  |
| Командой klist проверить наличие выданного билета. | Наблюдать выданный билет; Пользователя, для которого выдан билет; Время действия билета. |  |
| После получения билета, получить доступ к серверу средствами ssh. | Получить доступ на сервер без ввода пароля. |  |
|  |  |  |
| Командой kdestroy удалить выданный ранее билет. | Удаление выданного ранее билета |  |
| Получить доступ к серверу без билета. | Получить доступ к серверу без ввода пароля уже не получится. Необходим ввод пароля. |  |
|  |  |  |

## 4.2 Проведение натурного эксперимента

Проверка проводилась с методикой в соответствии с таблицей 1. Результат выполнения эксперимента приведён в таблице 2.

**Команда Kinit** используется для проверки подлинности на сервере Kerberos в качестве принципала, который получает билет на выдачу билета. Проверить что билет выдан можно командой klist. Для удаления билетов используется команда kdestroy.

Более того, можно посмотреть пакеты изнутри с помощью сниффера Wireshark.

*Wireshark* - программа-анализатор трафика для компьютерных сетей Ethernet. Имеет графический пользовательский интерфейс

Для этого с клиентской машины получим билет для получения билетов (TGT), введя в терминале команду kinit.

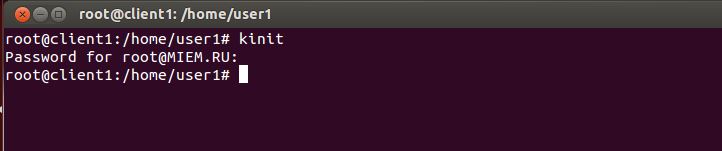


Рис. 31 Использование утилиты kinit

Проверим выданный билет командой klist.

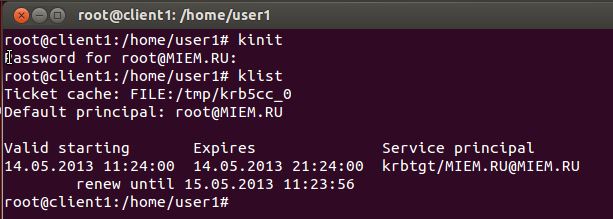


Рис. 32 Использование утилиты klist

Можно увидеть, что билет выдан и будет актуален 24 часа с момента выдачи.

На рис. 29 проиллюстрировано окно программы Wireshark, на котором рассматривается пакет с Kerberos. Можно увидеть область использования, имя пользователя и тип шифрования.

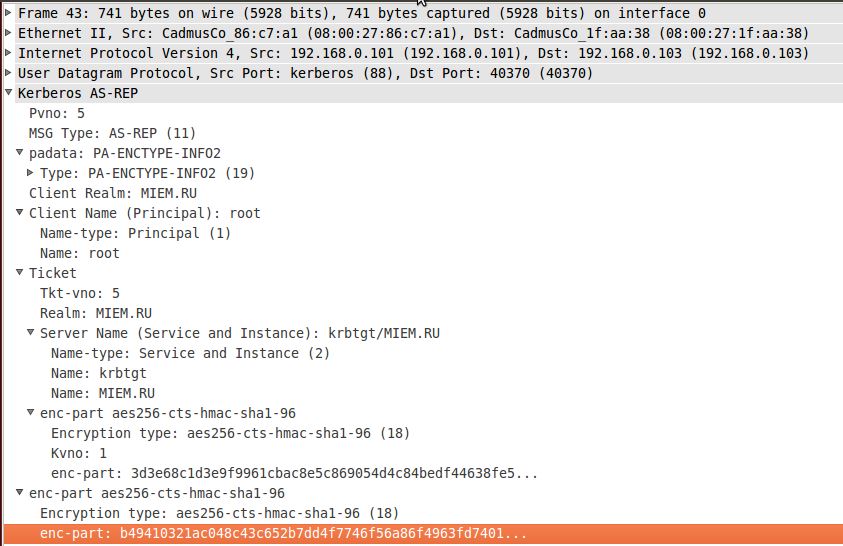


Рис. 33 Окно программы Wireshark

Таким образом, мы получили билет для получения билетов, Kerberos работает корректно.

Теперь проверим корректность работы Технологии единого входа SSO.

Подключимся к серверу через ssh набрав в терминале:

# ssh kdcbox.miem.ru

Итог проиллюстрирован на рис. 34

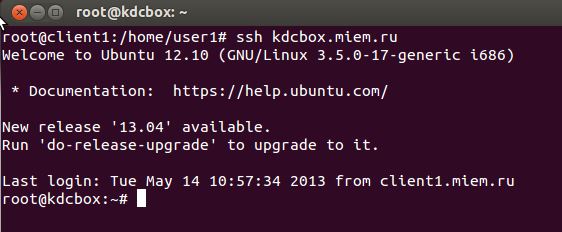


Рис. 34 Технология единого входа SSO на примере ssh

Как видно из рисунка 34 мы получили доступ к серверу без ввода каких либо данных.

Посмотрим что нам показывает программа Wireshark. На рисунке 31 видно как получаем билет на сервере распространения билетов (TGS).

Таким образом, проверили корректность работы технологии единого входа SSO на примере ssh . Работает корректно.

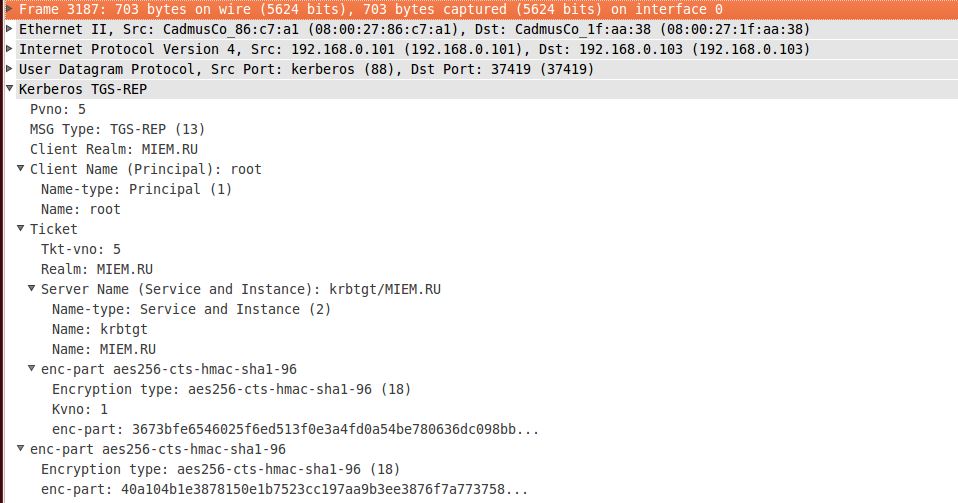


Рис. 35 Окно программы Wireshark

Теперь для чистоты эксперимента попробуем войти на сервер через ssh без билета для получения билетов (TGT).

Для этого удалим выданные ранее билеты командой kdestroy.

Проверим выданные нам билеты командой klists (будет видно что никаких билетов нет).

Подключимся к серверу набрав команду ssh kdcbox.miem.ru

Как видно из рисунка 36, запрашивается пароль на вход на сервер.

Таким образом ясно, что без билета на получение билета на сервер без ввода пароля не войти.

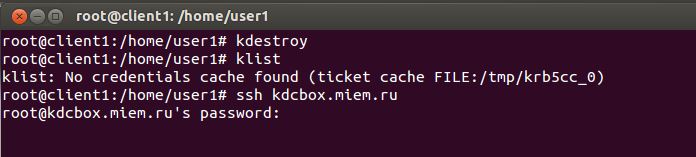


Рис. 36 Подключение к серверу без билета на выдачу билета TGT через ssh

Таблица 2: Методы испытаний

|  |  |  |
| --- | --- | --- |
| Описание функции | Предполагаемые результат | Полученный результат |
| Командой kinit запросить билет для получения билетов (TGT). | Получить билет для получения билетов (TGT). | Получили билет для получения билетов (TGT). |
| Командой klist проверить наличие выданного билета. | Наблюдать выданный билет; Пользователя, для которого выдан билет; Время действия билета. | Наблюдаем выданный билет; Пользователя, для которого выдан билет; Время действия билета. |
| После получения билета, получить доступ к серверу средствами ssh. | Получить доступ на сервер без ввода пароля. | Получили доступ на сервер без ввода пароля. |
|  |  |  |
| Командой kdestroy удалить выданный ранее билет. | Удаление выданного ранее билета | Удалили выданный ранее билета |
| Получить доступ к серверу без билета. | Получить доступ к серверу без ввода пароля уже не получится. Необходим ввод пароля. | Получить доступ к серверу без ввода пароля уже не удалось. Необходим ввод пароля. |

## 4.4. Выводы по разделу

В этом разделе описывалась разработка методики испытания для виртуальной среды для освоения протоколов распределенной аутентификации и авторизации пользователей, были определены требования, средства, порядок и методы испытаний. Проведён натурный эксперимент, где выяснилось, виртуальной среды для освоения протоколов распределенной аутентификации и авторизации пользователей полностью работоспособна и работает корректно.

# 5. Охрана труда

## 5.1 Исследование возможных опасных и вредных факторов при эксплуатации ЭВМ и их влияния на пользователей.

Во время работы с компьютером возникают угрожающие здоровью оператора ПЭВМ факторы. Существует перечень недомоганий, которые испытывают к концу рабочего дня люди, работающие с компьютером:

* головная боль
* заболевания глаз
* боли в мышцах шеи, рук
* боли в поясничных отделах спины
* кожный зуд и т.п.

Если на эти симптомы не обращать должного внимания, испытывать их каждый день, то они могут привести к мигреням, частичной потере зрения, сколиозу, кифозу, кифосколиозу, кожным воспаления и другим нежелательным заболеваниям.

При длительной работе за компьютером так же могут возникнуть следующие неприятные последствия: быстрая утомляемость, запястный синдром, стенокардия, различные стрессовые состояния, хронические головные боли, головокружения, повышенная возбудимость, депрессия, нарушение сна.

Основным источником вышеперечисленных заболеваний сотрудников, которые длительное время работают за компьютером, является монитор, особенно ЭЛТ монитор (монитор с электронно-лучевой трубкой), потому что монитор представляет собой источник вредного излучения.

ПЭВМ «питается» от электросети с напряжением 220В. Так как безопасное для человека напряжение не больше 40В, то при работе на ПЭВМ возникает потенциальная опасность поражения электрическим током.

Любые электронные устройства во время работы накапливают статическое электричество, вследствие чего происходит электризация пыли и мелких частиц, которые притягивается к устройству (например монитор компьютера). Таким образом собравшаяся на экране электризованная пыль ухудшает видимость, а при повышении подвижности воздуха, попадает на лицо и в легкие человека, что в свою очередь вызывает заболевания кожи и дыхательных путей.

### 5.1.1 Анализ влияния опасных и вредных факторов на пользователя

**Влияние электрического тока**

Результатом взаимодействия электрического тока и человека могут быть следующие травмы:

* Спазм мышц (судороги) без потери сознания/с потерей сознания
* Потеря сознания с нарушением работы органов дыхания и кровообращения
* Состояние клинической смерти
* Электрические ожоги

Взаимодействия, которые оказывает электрический ток при прохождении тела человека:

* Термическое — нагрев тканей человека (электрический ожог).
* Электролитическое — разложение крови и плазмы на ионы и нарушении их физико-химического состава и свойств.
* Биологическое — раздражение и возбуждение живых тканей организма, судорожное сокращением мышц, спазмы, нарушение внутренних биологических процессов.
* Механическое — может привести к расслоению, разрыву тканей человека, возможно образование пара из тканевой жидкости и крови.

Наиболее опасным для человека переменным током является ток с частотой 20 - 100Гц. Так как компьютер питается от сети переменного тока частотой 50Гц, то этот ток является опасным для человека.

**Влияние электромагнитных излучений**

При длительном контакте с электромагнитным полем возникает опасность развития «радиоволновой болезни», проявление которой заключается в изменении функционального состояния нервной и сердечно-сосудистой систем. Со стороны нервной системы: общая усталость, повышенная раздражительность, потеря краткосрочной памяти, малая эффективность сна. Со стороны сердечно-сосудистой системы: гипотония, боли в сердце, аритмия пульса.

Электромагнитные поля частота которых выше 60Гц в последствии приводят к изменениям в структуре костного мозга человека, что сказывается на увеличении скорости регенерации. Так же негативно сказывается и на иммунитете человека. Электромагнитные поля подавляют работу вилочковой железы, которая отвечает за вырабатывание лимфоцитов, что в свою очередь существенно повышает возможность аутоимунных реакций иммунитета на здоровые ткани человека.

**Влияние ультрафиолетового излучения**

В малых дозах ультрафиолетовое излучение (УФ) благоприятно влияет на здоровье и играет важную роль в выработке витамина «D». Но большое воздействие УФ излучения может привести различным патологиям:

* Патологии кожи – приводит к злокачественному раку кожи (меланома, плоскоклеточная карцинома кожи, базальноклеточная карцинома), потери эластичности кожи человека.
* Патологии глаз – приводит к воспалению роговицы и конъюнктивиту. Данные патологии глаз обратимы, поддаются лечению и можно предостеречь ношением солнцезащитных очков.

**Влияние статического электричества**

Благодаря статическому электричеству электризуется пыль, что способствует образованию скоплений пыли. Скопление пыли может вызвать воспаление кожи, что в свою очередь приводит к появлению угрей**.** Скопление пыли может образовываться на экранах мониторов рабочих компьютеров. Физика этого явления заключается в том, что наэлектризованный монитор притягивает к себе частички пыли из окружающего воздуха, поэтому вблизи него снижается «качество» воздуха и оператор, работающий за компьютером, вынужден работать в более запыленной атмосфере. Этим же воздухом он и дышит, что крайне негативно сказывается на его здоровье.

### 5.1.2 Требования к помещениям и организации рабочих мест

Требования к помещениям, в которых эксплуатируются компьютеры:

* Не допускается в подвальных помещениях размещение рабочих мест.
* Объем для одного рабочего места должен быть не менее 20 куб. м., а площадь не меньше 6 кв. м.

Чтобы повысить влажность воздуха в помещении используются увлажнители воздуха, которые нужно заправлять прокипяченной питьевой или дистиллированной водой. Помещения должны проветриваться до начала работы и каждый час работы.

Необходимо придерживаться следующих микроклиматических условий в помещениях при работе с ПЭВМ:

* Температура воздуха в границах 19- 21°С;
* Относительная влажность воздуха в границах 49-72%.

В помещениях, где располагаются скопление вычислительных машин (серверы, принтеры т.п.), уровень шума не должен превышать 75дБА, а в обычных помещениях не должен превышать 65 дБА.

Помещения должны освещаться как искусственным так и естественным светом. Оконные проемы должны быть оснащены регулируемыми жалюзи или занавесками, которые полностью закрывают оконные проемы. Занавески должны быть из плотной ткани, одноцветные, гармонирующие с цветом стен.

Чтобы занавески играли звукопоглощающую роль, их нужно подвесить на расстояние 13-19 см от стены с оконными проемами.

Рабочие места должна располагаться таким образом, чтобы естественный свет падал на них сбоку, преимущественно - слева.

Рабочие места должны располагаться от стен с оконными проемами на расстоянии не менее 1,5 м, от стен без оконных проемов на расстоянии не менее 1,0 м.

В рабочих помещениях пол должен быть ровным, покрыт нескользящим покрытием, обладать антистатическими свойствами.

Освещенность на рабочем месте с ПЭВМ должна быть не менее:

* монитора - 200 лк;
* клавиатуры, документов и стола - 400 лк.

При искусственном освещении в качестве источников света применяются люминесцентные лампы типа ЛБ, цветовая температура (Тцв) излучения которых находится в диапазоне 3500-4200 K.

В светильниках местного освещения возможно применение ламп накаливания. Так же нужно устранить из непосредственной близости к оператору источники света, а также убрать отражающие поверхности (например, поверхность блестящих полированных столов) для исключения ослепления оператора. При этом потолок должен быть плоским, матовым и однородным, освещенность должна быть равномерной. Для регулирования высоты подвеса светильников помещение должно быть с достаточно высоким потолком.

Расстояние между мониторами должно быть не меньше 2 метров друг от друга, и, 1,2 метра, в случае длины от задней крышки одного монитора до задней крышки другого, между их боковыми поверхностями соответственно.

Рабочий стул должен быть оснащен подъемно-поворотным механизмом и регулируемым по высоте и углам наклона сиденья и спинки стула.

Монитор должен находиться на расстоянии не менее 50 сантиметров от глаз пользователя. В рабочем помещении следует проводить влажную уборку ежедневно

### 5.1.3 Требования к организации работы

Длительность работы в компьютерных классах для преподавателей вузов и учителей средних учебных заведений составляет не более 4 часов в день.

Для инженеров - не более 6 часов в день. Для оператора ПЭВМ время непрерывной работы за компьютером не должна превышать 2 часов. Так же необходимо каждые 2 часа совершать 15-минутные перерывы.

В случае, если рабочая смена за компьютером составляет 12 часов, то в течение последних четырех часов, через каждый час нужно делать 15-минутный перерыв.

Для всех видов и категорий работ, при работе с компьютером в ночное время суток, продолжительность плановых перерывов увеличиваются на 60 минут.

Работающие с компьютером сотрудники обязаны проходить периодические медицинские осмотры. Необходимо исключить работу с компьютером женщин во время беременности и в период кормления ребенка.

В случае, если на предприятии работают сотрудники с ограниченными возможностями, требуется создать соответствующие рабочие условия, установить время рабочего дня и регламентированные перерывы в индивидуальном порядке.

## 5.2 Методы и средства защиты пользователей от воздействия на них опасных и вредных факторов

**Методы и средства защиты от поражения электрическим током**

Защитное заземление - это соединение металлических нетоковедущих частей, которые могут оказаться под напряжением, с землей или ее эквивалентом.

Защитное действие заземления основано на двух принципах:

* Уменьшить напряжение до безопасного значения между заземляемым предметом и другим проводящим предметом, с естественным заземлением.
* Отвод тока утечки, в случае контакта заземляемого проводящего элемента с фазным проводом.

Зануление – это соединение металлических нетоковедущих частей, которые могут оказаться под напряжением, с нулевым защитным проводником.

Принцип защиты занулением заключается в следующем: в случае замыкания одной из фаз на заземляющий корпус, в цепи появляется ток замыкания, отключающий от потребителя сеть.

**Методы и средства защиты от ультрафиолетового излучения**

От ультрафиолетового излучения для защиты организма человека применяют:

* Специальный защитный фильтр или специальные очки (толщина стекол 2мм, насыщенных свинцом).
* Используют одежда из фланели и поплина.
* Применяют белила для стен стен и потолка (ослабляет эффект УФ излучения на 45-50%).
* Использование штор или жалюзи.

**Защита от электромагнитных излучений осуществляется следующими способами:**

* Непрерывное время работы - не более 4 часов
* Расстояние от глаз до монитора - не менее 50 см от источника
* Экранирование мониторов защитными экранами

**Защита от статического электричества и вызванных им явлений осуществляется следующими способами:**

* Устанавливать контурное заземление.
* Подавлять статического электричества.
* Отсутствие синтетических покрытий.
* Ежедневная влажная уборка.

## 5.3. Выводы по разделу

Выбранные методы и способы защиты от опасных и вредных факторов, при соблюдении эргономических требований, обеспечивают защиту пользователей̆, работающих с вычислительной техникой̆.

# 6 Итоги

Разрабатываема виртуальная среда для освоения протоколов распределенной аутентификации и авторизации пользователей была сделана в соответствии с техническим заданием. Работа была выполнена полностью.

В ходе выполнения дипломной работы проведён обзор существующих протоколов аутентификации и авторизации пользователей, среди которых Kerberos, RADIUS, TACACS+. Итогом данного анализа стал выбор протокола Kerberos. Произведен обзор и выбор системы виртуализации, итогом которого стал Virtualbox. Так же в результате обзора и выбора операционной системы была выбрана Ubuntu 12.10. Разработана и реализована структура стенда и виртуальной среды, установлен и настроен протокол Kerberos с поддержкой технологии единого входа (SSO) на примере ssh. Разработана методика испытаний, где в ходе экспериментов была показана полная работоспособность виртуальной среды для освоения протоколов распределенной аутентификации и авторизации пользователей.

# 7 Список литературы

1. Олифер В.Г., Олифер Н.А. - Сетевые операционные системы (Учебник для ВУЗов) – 2009.
2. Решения компании Cisco Systems по обеспечению безопасности. Мерике Кео.
3. ГОСТ 12.4-99. Средства защиты от статического электричества.
4. ГОСТ Р 50948, 49-96. Общие эргономические требования и требования безопасности и ее параметры для ЭВМ.
5. СанПиН No1340-03. Гигиенические требования к персональным ЭВМ и организация работы. Санитарно-гигиенические правила и нормы.
6. Указания по проектированию и эксплуатации искусственного УФ-излучения на промышленных предприятиях.
7. ГОСТ 19.301-79 Устанавливает требования к содержанию и оформлению программного документа «Программа и методика испытаний».
8. ГОСТ 12.0.003-99. Опасные и вредные производственные факторы.
9. ГОСТ 12.0.002-80. Система стандартов безопасности труда.
10. ГОСТ 12.1.045-01. Электростатические поля. Допустимые уровни на рабочих местах.
11. Указания по проектированию и эксплуатации искусственного УФ-излучения на промышленных предприятиях.
12. Веб-сайт <http://wiki.debian.org/>
13. Веб-сайт <https://wiki.ubuntu.com/>
14. Веб-сайт <https://wiki.archlinux.org/>
15. Веб-сайт http://ru.wikipedia.org/wiki/Fedora

# Приложение 1

Исходный конфигурационный файл /etc/rkb5.conf

[libdefaults]

default\_realm = MIEM.RU

# The following krb5.conf variables are only for MIT Kerberos.

krb4\_config = /etc/krb.conf

krb4\_realms = /etc/krb.realms

kdc\_timesync = 1

ccache\_type = 4

forwardable = true

proxiable = true

# The following encryption type specification will be used by MIT Kerberos

# if uncommented. In general, the defaults in the MIT Kerberos code are

# correct and overriding these specifications only serves to disable new

# encryption types as they are added, creating interoperability problems.

#

# Thie only time when you might need to uncomment these lines and change

# the enctypes is if you have local software that will break on ticket

# caches containing ticket encryption types it doesn't know about (such as

# old versions of Sun Java).

# default\_tgs\_enctypes = des3-hmac-sha1

# default\_tkt\_enctypes = des3-hmac-sha1

# permitted\_enctypes = des3-hmac-sha1

# The following libdefaults parameters are only for Heimdal Kerberos.

v4\_instance\_resolve = false

v4\_name\_convert = {

host = {

rcmd = host

ftp = ftp

}

plain = {

something = something-else

}

}

fcc-mit-ticketflags = true

[realms]

MIEM.RU = {

kdc = kdc.miem.ru

admin\_server = kdc.miem.ru

}

ATHENA.MIT.EDU = {

kdc = kerberos.mit.edu:88

kdc = kerberos-1.mit.edu:88

kdc = kerberos-2.mit.edu:88

admin\_server = kerberos.mit.edu

default\_domain = mit.edu

}

MEDIA-LAB.MIT.EDU = {

kdc = kerberos.media.mit.edu

admin\_server = kerberos.media.mit.edu

}

ZONE.MIT.EDU = {

kdc = casio.mit.edu

kdc = seiko.mit.edu

admin\_server = casio.mit.edu

}

MOOF.MIT.EDU = {

kdc = three-headed-dogcow.mit.edu:88

kdc = three-headed-dogcow-1.mit.edu:88

admin\_server = three-headed-dogcow.mit.edu

}

CSAIL.MIT.EDU = {

kdc = kerberos-1.csail.mit.edu

kdc = kerberos-2.csail.mit.edu

admin\_server = kerberos.csail.mit.edu

default\_domain = csail.mit.edu

krb524\_server = krb524.csail.mit.edu

}

IHTFP.ORG = {

kdc = kerberos.ihtfp.org

admin\_server = kerberos.ihtfp.org

}

GNU.ORG = {

kdc = kerberos.gnu.org

kdc = kerberos-2.gnu.org

kdc = kerberos-3.gnu.org

admin\_server = kerberos.gnu.org

}

1TS.ORG = {

kdc = kerberos.1ts.org

admin\_server = kerberos.1ts.org

}

GRATUITOUS.ORG = {

kdc = kerberos.gratuitous.org

admin\_server = kerberos.gratuitous.org

}

DOOMCOM.ORG = {

kdc = kerberos.doomcom.org

admin\_server = kerberos.doomcom.org

}

ANDREW.CMU.EDU = {

kdc = kerberos.andrew.cmu.edu

kdc = kerberos2.andrew.cmu.edu

kdc = kerberos3.andrew.cmu.edu

admin\_server = kerberos.andrew.cmu.edu

default\_domain = andrew.cmu.edu

}

CS.CMU.EDU = {

kdc = kerberos.cs.cmu.edu

kdc = kerberos-2.srv.cs.cmu.edu

admin\_server = kerberos.cs.cmu.edu

}

DEMENTIA.ORG = {

kdc = kerberos.dementix.org

kdc = kerberos2.dementix.org

admin\_server = kerberos.dementix.org

}

stanford.edu = {

kdc = krb5auth1.stanford.edu

kdc = krb5auth2.stanford.edu

kdc = krb5auth3.stanford.edu

master\_kdc = krb5auth1.stanford.edu

admin\_server = krb5-admin.stanford.edu

default\_domain = stanford.edu

}

UTORONTO.CA = {

kdc = kerberos1.utoronto.ca

kdc = kerberos2.utoronto.ca

kdc = kerberos3.utoronto.ca

admin\_server = kerberos1.utoronto.ca

default\_domain = utoronto.ca

}

[domain\_realm]

.mit.edu = ATHENA.MIT.EDU

mit.edu = ATHENA.MIT.EDU

.media.mit.edu = MEDIA-LAB.MIT.EDU

media.mit.edu = MEDIA-LAB.MIT.EDU

.csail.mit.edu = CSAIL.MIT.EDU

csail.mit.edu = CSAIL.MIT.EDU

.whoi.edu = ATHENA.MIT.EDU

whoi.edu = ATHENA.MIT.EDU

.stanford.edu = stanford.edu

.slac.stanford.edu = SLAC.STANFORD.EDU

.toronto.edu = UTORONTO.CA

.utoronto.ca = UTORONTO.CA

[login]

krb4\_convert = true

krb4\_get\_tickets = false