РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян

Информационная безопасность компьютерных сетей

Лабораторные работы

Учебное пособие

Москва
Российский университета дружбы народов
2015

УДК 004.056.5+003.26
ББК
К

Утверждено
РИС Учёного совета
Российского университета
дружбы народов

Рецензенты:

К Информационная безопасность компьютерных сетей: лабораторные работы : учебное пособие / Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. — Москва : РУДН, 2015. — 64 с. : ил.

УДК 004.056.5+003.26
ББК

ISBN

Оглавление

Лабораторный практикум

1. Установка и конфигурация операционной системы на виртуальную машину

1.1. Цель работы

Приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

1.2. Информация, необходимая для начала работы

Лабораторная работа подразумевает установку на виртуальную машину VirtualBox (`https://www.virtualbox.org/`) операционной системы Linux, дистрибутив Centos.

Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук РУДН, так и дома. Описание выполнения работы приведено для дисплейного класса со следующими характеристиками техники:

- Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 8GB свободного места на жёстком диске;
- ОС Linux Gentoo (`http://www.gentoo.ru/`);
- VirtualBox верс. 4.3.18 или старше.

1.3. Последовательность выполнения работы

Загрузить в дисплейном классе операционную систему Linux. Осуществить вход в систему.

Запустить терминал. Перейти в каталог `/var/tmp`:

```
cd /var/tmp
```

Создать каталог с именем пользователя (желательно совпадающим с логином студента в дисплейном классе), например, `avkorolkova`:

```
mkdir avkorolkova
```

Перейти в общий каталог, где размещён образ виртуальной машины:

```
cd /afs/dk.sci.pfu.edu.ru/common/files
```

Скопировать образ, например, `CentOS-6.6-i386-bin-DVD1.iso` виртуальной машины в созданный на предыдущем шаге каталог:

```
cp CentOS-6.6-i386-bin-DVD1.iso /var/tmp/avkorolkova
```

Запустить виртуальную машину (рис. 1.1), введя `VirtualBox` в командной строке.

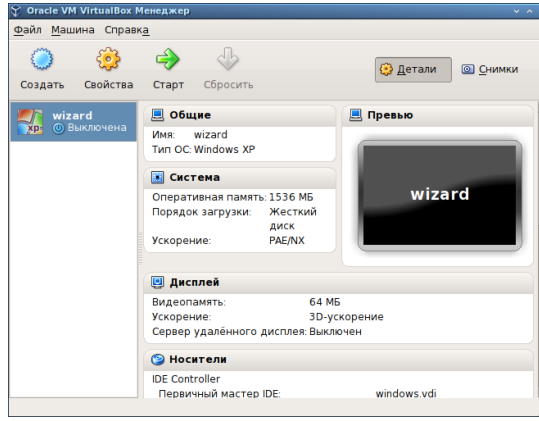

Рис. 1.1. Менеджер VirtualBox

Проверить в свойствах VirtualBox **месторасположение каталога для виртуальных машин**. Для этого в VirtualBox выбрать Файл > Свойства, вкладка Общие. В поле Папка для машин (рис. 1.2) должно стоять

`/var/tmp/имя_пользователя`

где `имя_пользователя` — логин (учётная запись) студента в дисплейном классе. Если указан другой каталог, то **изменить его**, как указано выше.

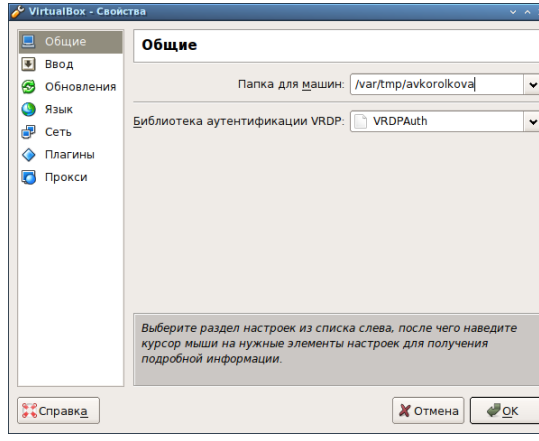

Рис. 1.2. Окно «Свойства» VirtualBox

Создать новую виртуальную машину. Для этого в VirtualBox выбрать Машина > Создать.

Указать имя виртуальной машины — Base, тип операционной системы — Linux, RedHat (рис. 1.3). Указать размер основной памяти виртуальной машины — 1024 МБ (рис. 1.4).

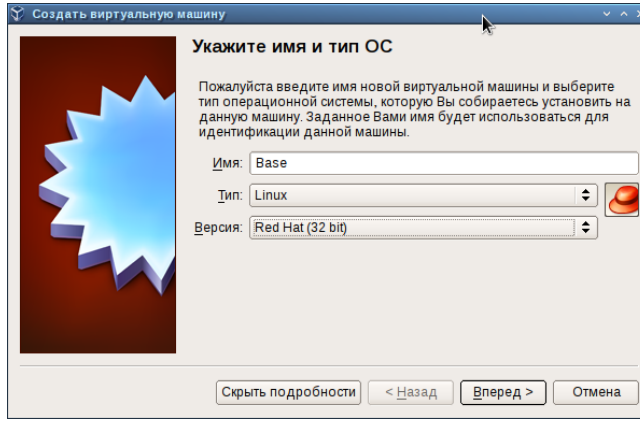

Рис. 1.3. Окно «Имя машины и тип ОС»

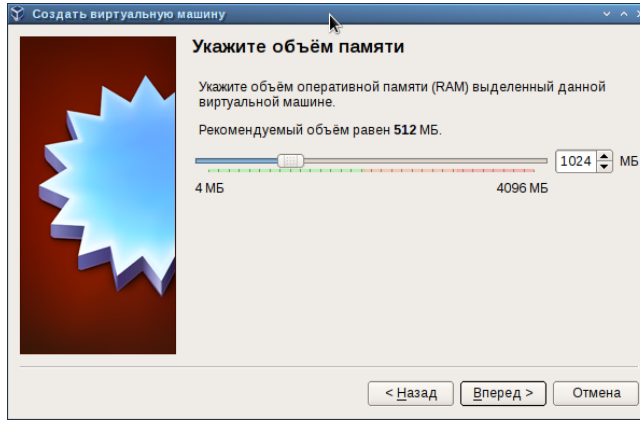

Рис. 1.4. Окно «Размер основной памяти»

Задать конфигурацию жёсткого диска — загрузочный, VDI (BirtualBox Disk Image), динамический виртуальный диск (рис. 1.5–1.7).

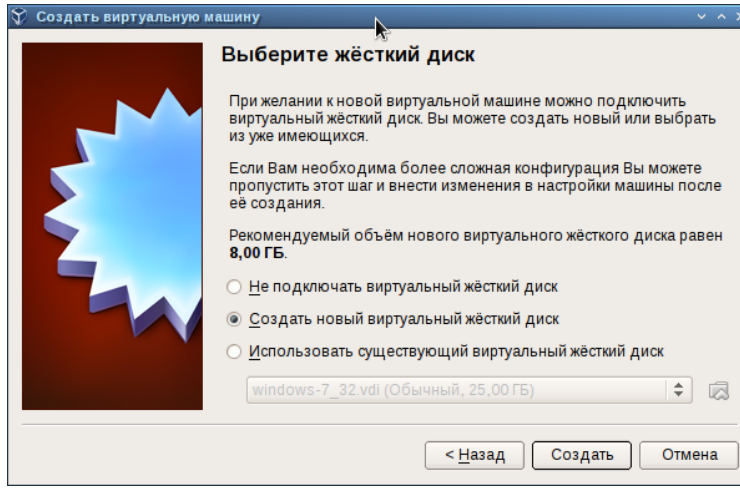

Рис. 1.5. Окно «Виртуальный жёсткий диск»

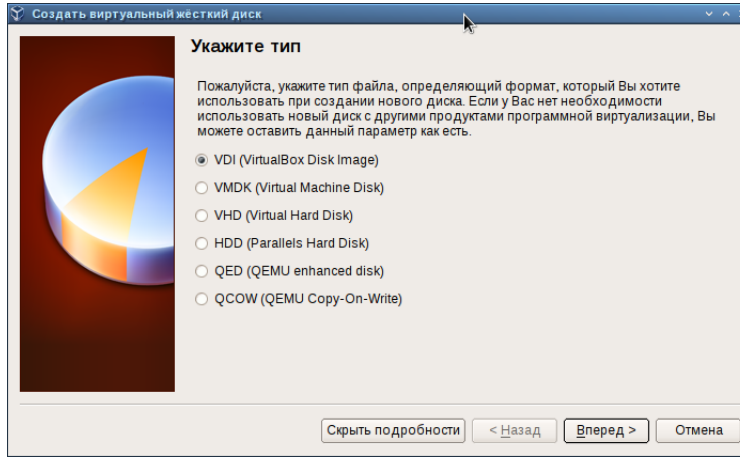

Рис. 1.6. Окно «Мастер создания нового виртуального диска»

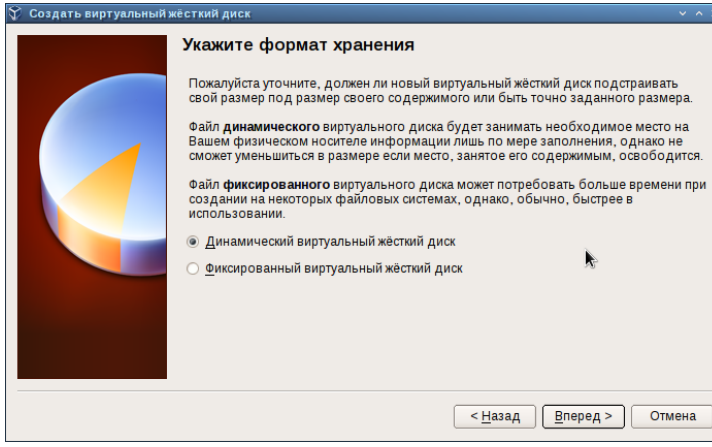

Рис. 1.7. Окно «Дополнительные атрибуты виртуального диска»

Задать размер диска — 40 ГБ, его расположение — в данном случае `/var/tmp/имя_пользователя/Base/Base.vdi` (рис. 1.8).

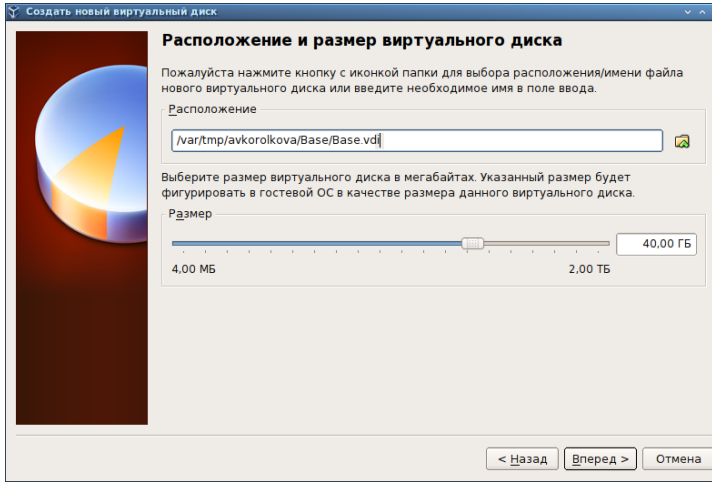

Рис. 1.8. Окно «Расположение и размер виртуального диска»

Выделить в окне менеджера VirtualBox виртуальную машину Base, и открыть окно Свойства. **Проверить**, что **папка для снимков виртуальной машины Base** имеет путь `/var/tmp/имя_пользователя/Base/Snapshots`. Для этого надо выбрать в VirtualBox Свойства виртуальной машины Base, Общие, вкладка Дополнительно (рис. 1.9).

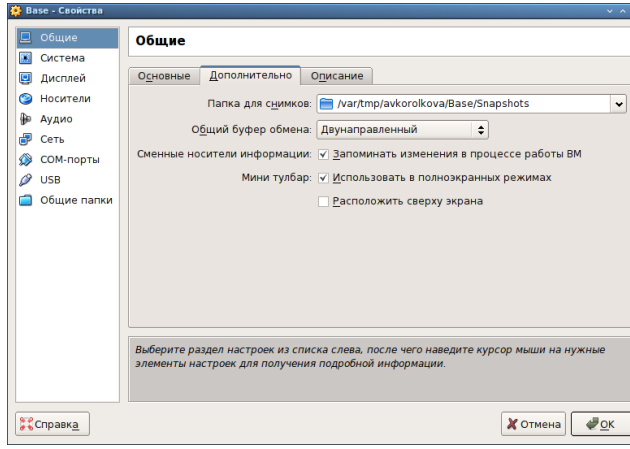

Рис. 1.9. Окно «Свойства» виртуальной машины Base

Выбрать в VirtualBox Свойства > Носители виртуальной машины Base.

Добавить новый привод оптических дисков и выбрать образ `CentOS-6.6-i386-bin-DVD1.iso` (рис. 1.10–1.11).

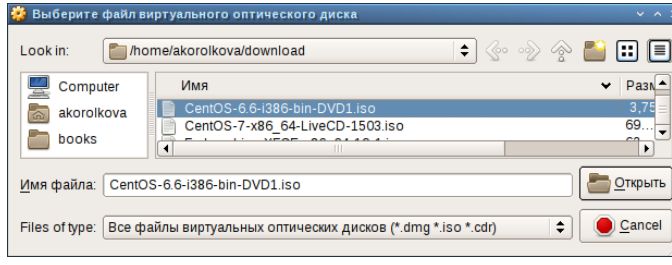

Рис. 1.10. Окно «Носители» виртуальной машины Base: выбор образа оптического диска

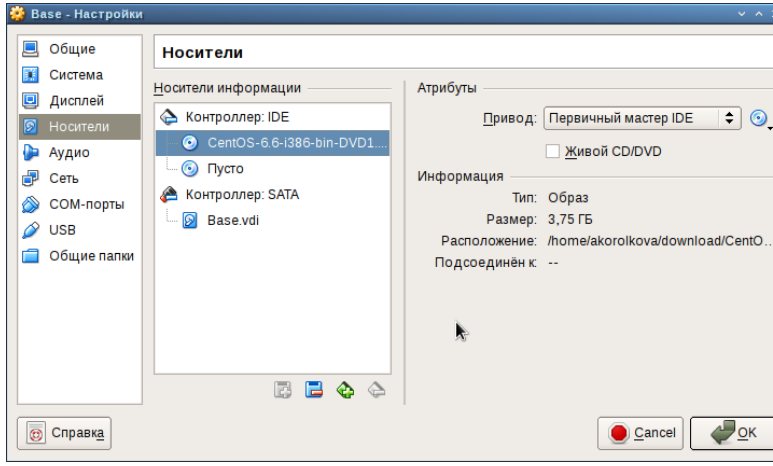

Рис. 1.11. Окно «Носители» виртуальной машины Base

Запустить виртуальную машину Base, выбрать установку системы на жёсткий диск (рис. 1.12).

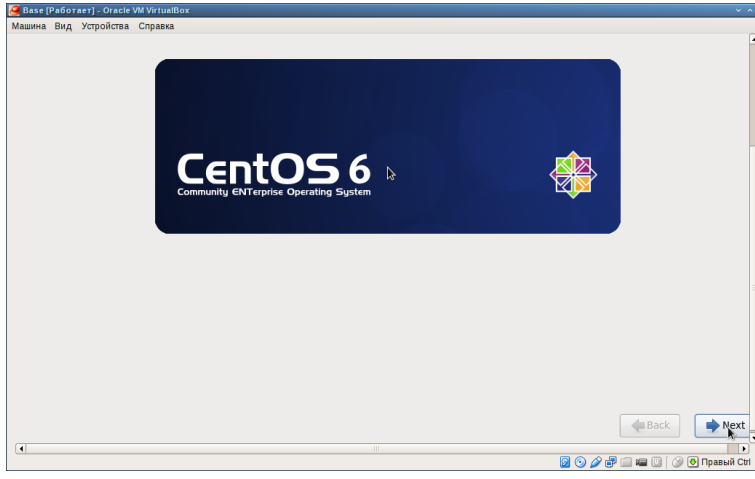

Рис. 1.12. Запуск установки системы

Установить русский язык для интерфейса (рис. 1.13) и раскладки клавиатуры (рис. 1.14).

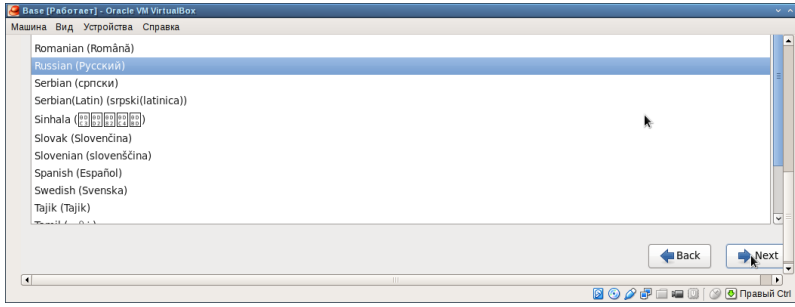

Рис. 1.13. Виртуальная машина Base. Установка русского языка

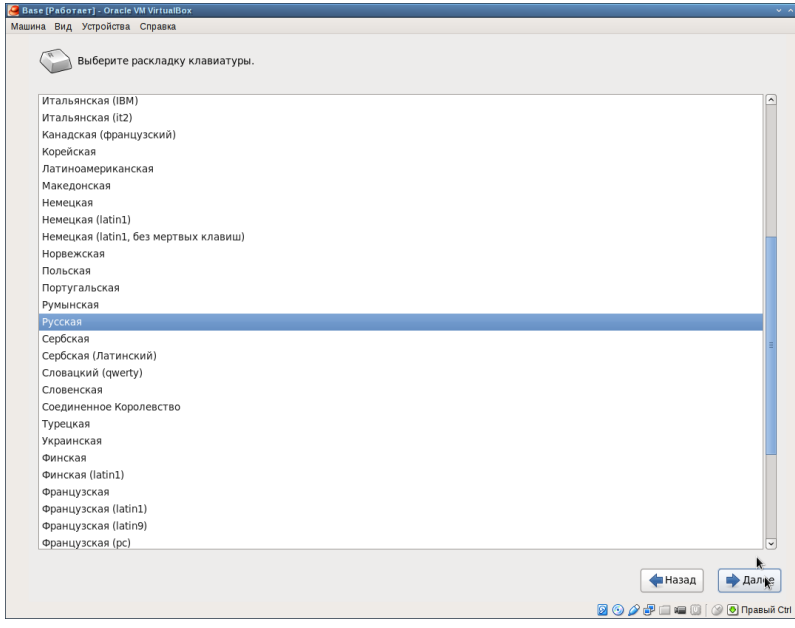

Рис. 1.14. Виртуальная машина Base. Установка русского языка для раскладки клавиатуры

Указать Стандартные накопители (рис. 1.15) для установки ОС. В окне конфигурации жёсткого диска выбрать Да, удалить данные (рис. 1.16).

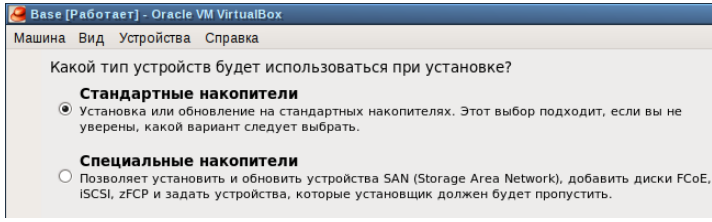

Рис. 1.15. Виртуальная машина Base

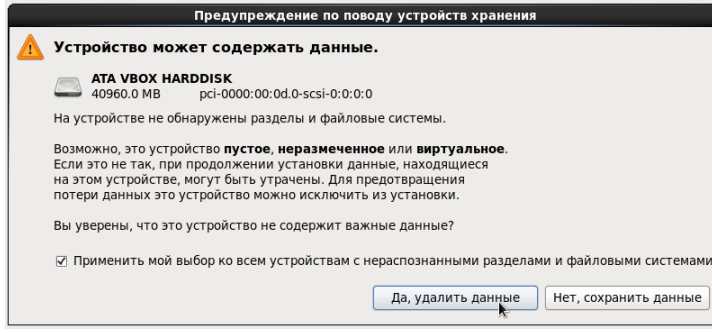

Рис. 1.16. Конфигурация жёсткого диска

В качестве имени машины указать «имя_пользователя.localdomain» (рис. 1.17). Указать часовой пояс «Москва» (рис. 1.18).

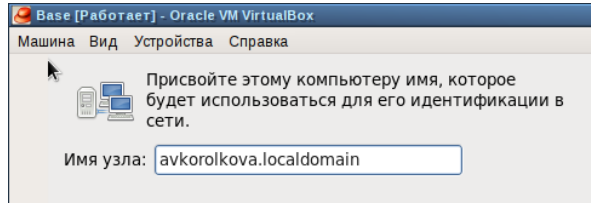

Рис. 1.17. Задать сетевое имя виртуальной машины

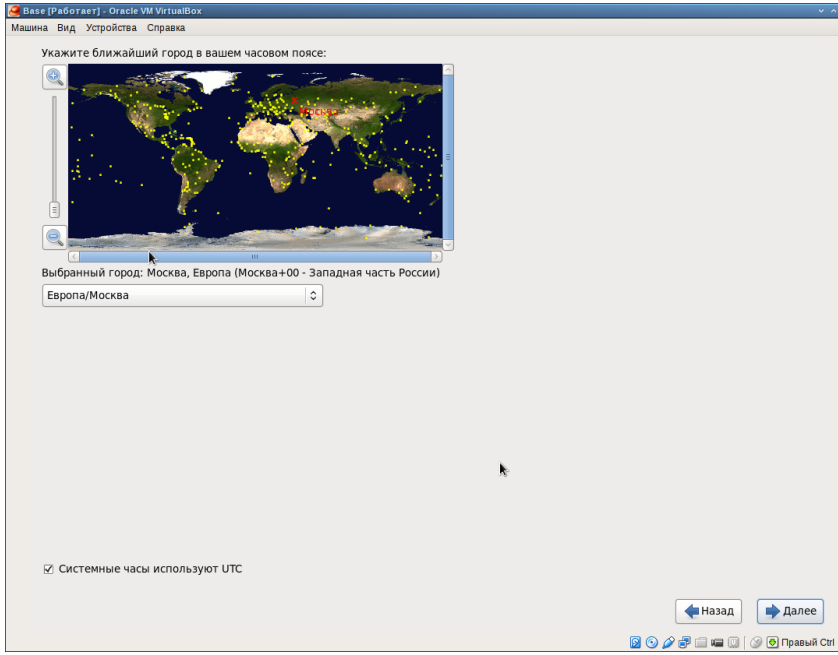

Рис. 1.18. Указать часовой пояс «Москва»

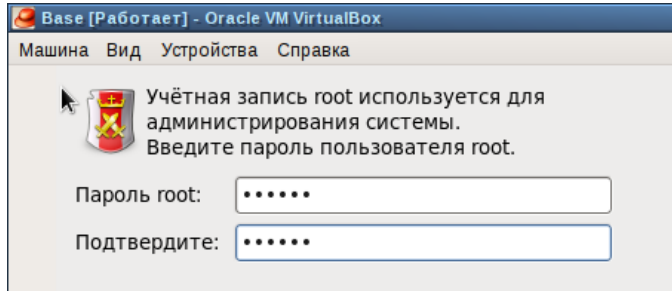

Рис. 1.19. Установка пароля для root

Установить пароль для root (рис. 1.19).

При конфигурировании размера жёсткого диска указать «Всё пространство» (рис. 1.20).

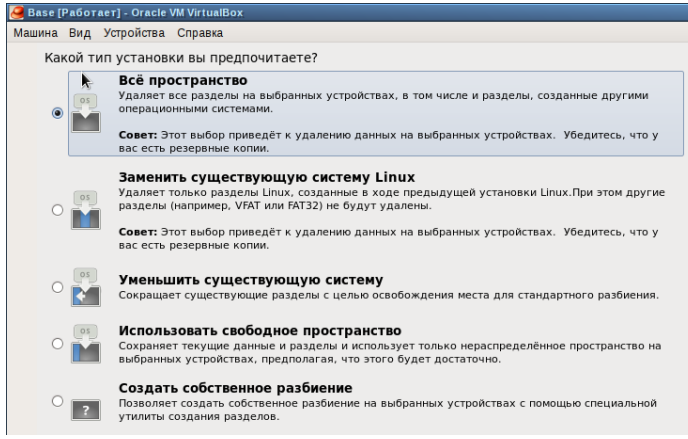

Рис. 1.20. Конфигурирование размера жёсткого диска

Выбрать вариант стандартной установки CentOS (рис. 1.21).

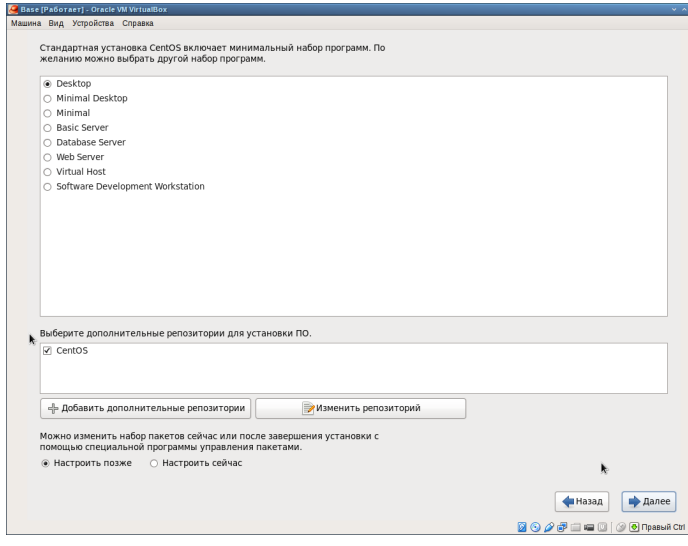

Рис. 1.21. Варианты стандартной установки CentOS

Завершить установку операционной системы (рис. 1.22) и перезагрузить её.

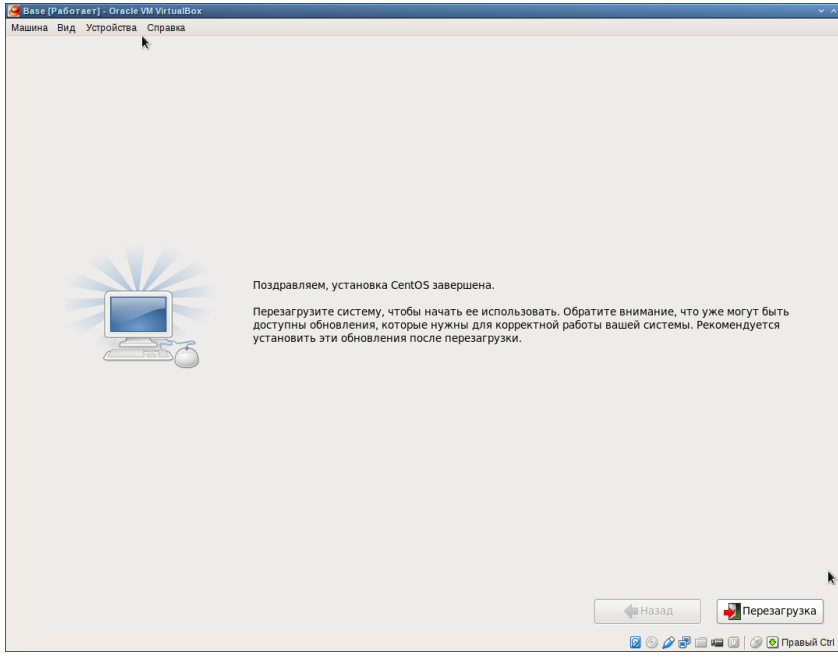

Рис. 1.22. Виртуальная машина Base. Завершение установки

В VirtualBox оптический диск должен отключиться автоматически, но если это не произошло, то необходимо отключить носитель информации с образом, выбрав Свойства » Носители » `CentOS-6.6-i386-bin-DVD1.iso` » Удалить устройство.

Запустить виртуальную машину Base и настроить её (рис. 1.23–1.26).

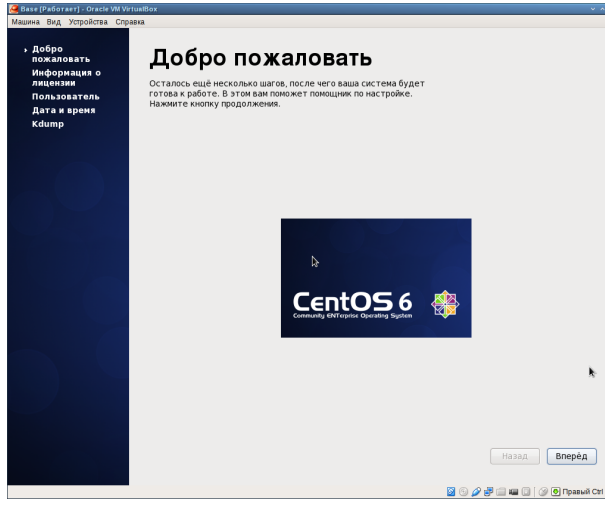

Рис. 1.23. Запуск системы

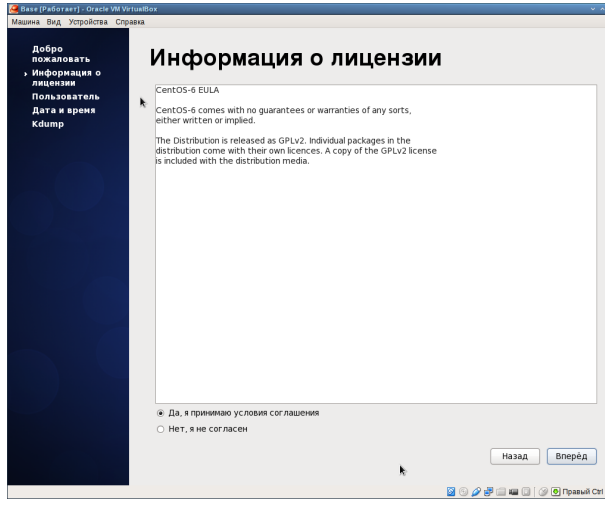

Рис. 1.24. Информация о лицензии

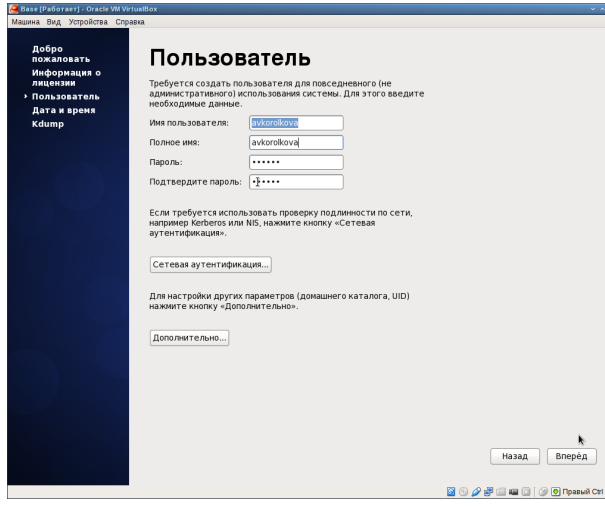

Рис. 1.25. Настройка виртуальной машины: учётная запись

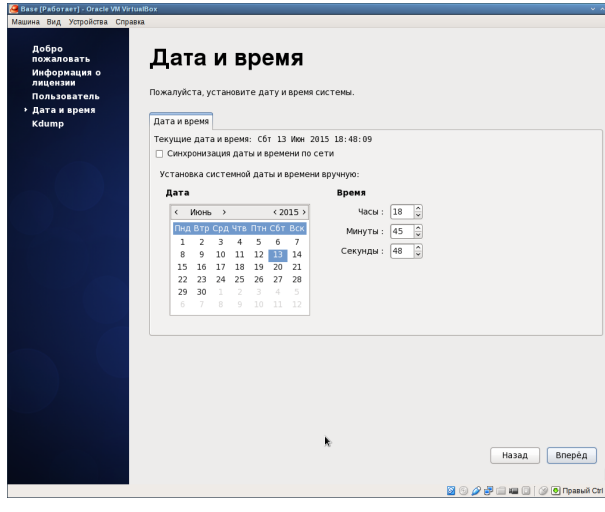

Рис. 1.26. Настройка виртуальной машины: дата и время

Подключиться к виртуальной машине с помощью созданной учётной записи (рис. 1.27).

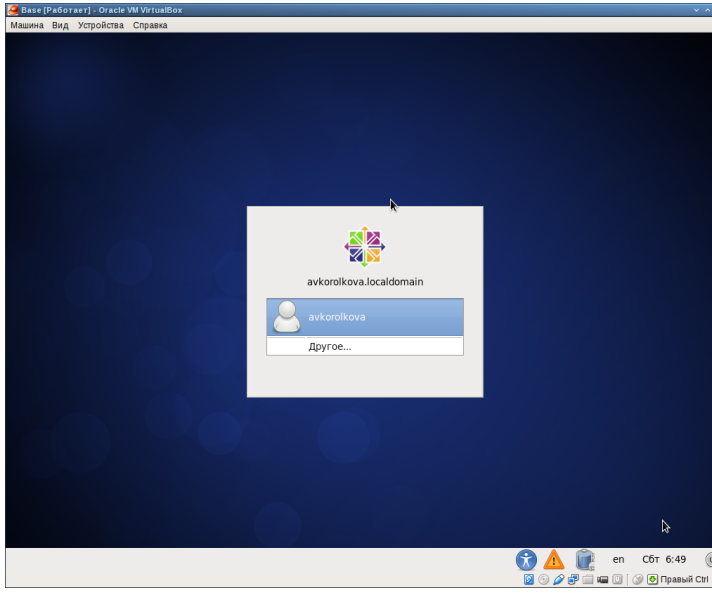

Рис. 1.27. Подключение к виртуальной машине

На виртуальной машине Base запустить терминал, перейти под учетную запись `root` с помощью команды `su`.

С помощью команды `yum update` обновить системные файлы и установить необходимые программы, например, mc:

```
yum update
yum install mc
```

После установки необходимых программ можно завершить работу виртуальной машины. Её конфигурация сохранится на жёстком диске в директории `/var/tmp/имя_пользователя/Base`.

Для того чтобы другие виртуальные машины могли использовать машину Base и её конфигурацию как базовую, необходимо произвести следующие действия. В VirtualBox в меню выбрать Файл › Менеджер виртуальных носителей › Жёсткие диски и, выделив «Base.dvi», указать Освободить (рис. 1.28–1.30).

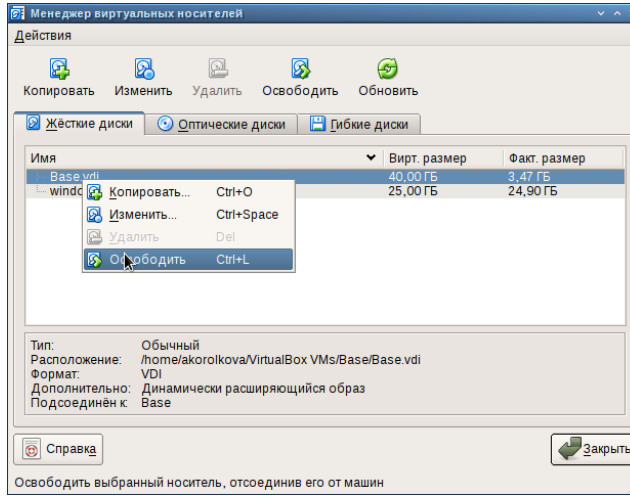

Рис. 1.28. Менеджер виртуальных носителей: освободить жёсткий диск

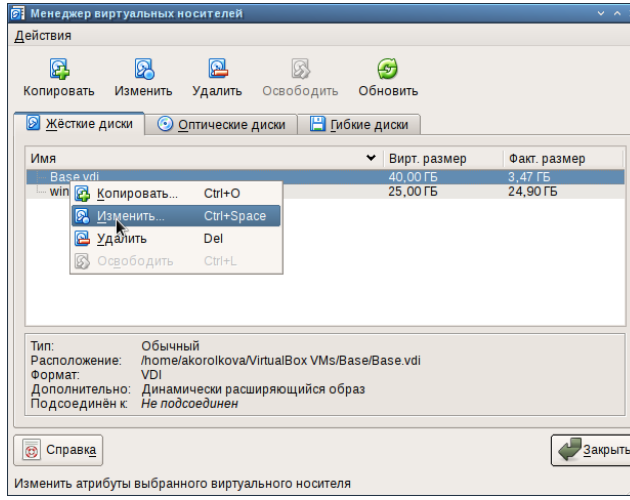

Рис. 1.29. Менеджер виртуальных носителей: изменить свойства жёсткого диска

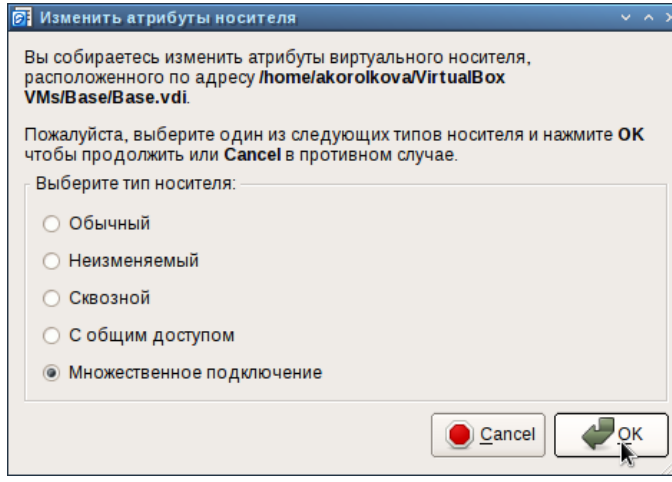

Рис. 1.30. Менеджер виртуальных носителей: множественное подключение

Теперь на основе виртуальной машины Base можно создать машину Host2, выбрав в VirtualBox Машина > Создать и в «Мастере создания новой виртуальной машины» указав в качестве имени машины Host2, в качестве типа операционной системы — Linux, версия «RedHat», а при конфигурации виртуального жёсткого диска выбрав «Использовать существующий жёсткий диск» Base.vdi.

2. Дискреционное разграничение прав в Linux

2.1. Цель работы

Получение практических навыков работы в консоли с атрибутами файлов, закрепление теоретических основ дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux[1].

2.2. Подготовка к выполнению лабораторной работы

1. В установленной при выполнении предыдущей лабораторной работы операционной системе создайте учётную запись пользователя guest:
   `useradd guest`
2. Задайте пароль для пользователя guest:
   `passwd guest`
3. Аналогично создайте второго пользователя guest2.
4. Добавьте пользователя guest2 в группу guest:
   `gpasswd -a guest2 guest`

2.3. Задание 1. Основные атрибуты

Постарайтесь последовательно выполнить все пункты, занося ваши ответы на поставленные вопросы и замечания в отчёт.

1. Войдите в систему от имени пользователя guest.
2. Определите директорию, в которой вы находитесь, командой `pwd`. Сравните её с приглашением командной строки. Определите, является ли она вашей домашней директорией? Если нет, зайдите в домашнюю директорию.
3. Уточните имя вашего пользователя командой `whoami`.
4. Уточните имя вашего пользователя, его группу, а также группы, куда входит пользователь, командой `id`. Выведенные значения `uid`, `gid` и др. запомните. Сравните вывод `id` с выводом команды `groups`.
5. Сравните полученную информацию об имени пользователя с данными, выводимыми в приглашении командной строки.
6. Просмотрите файл `/etc/passwd` командой
   `cat /etc/passwd`

[1] При составлении работы использовались материалы [**zaklyakov:2010:discret:samag:part1**; **zaklyakov:2010:discret:samag:part2**; **zaklyakov:2011:discret:samag:part3**].

Найдите в нём свою учётную запись. Определите uid пользователя. Определите gid пользователя. Сравните найденные значения с полученными в предыдущих пунктах.
Замечание: в случае, когда вывод команды не умещается на одном экране монитора, используйте прокрутку вверх–вниз (удерживая клавишу `shift`, нажимайте `page up` и `page down`) либо программу `grep` в качестве фильтра для вывода только строк, содержащих определённые буквенные сочетания:
```
cat /etc/passwd | grep guest
```
7. Определите существующие в системе директории командой
```
ls -l /home/
```
Удалось ли вам получить список поддиректорий директории `/home`? Какие права установлены на директориях?
8. Проверьте, какие расширенные атрибуты установлены на поддиректориях, находящихся в директории `/home`, командой:
```
lsattr /home
```
Удалось ли вам увидеть расширенные атрибуты директории?
Удалось ли вам увидеть расширенные атрибуты директорий других пользователей?
9. Создайте в домашней директории поддиректорию `dir1` командой
```
mkdir dir1
```
Определите командами `ls -l` и `lsattr`, какие права доступа и расширенные атрибуты были выставлены на директорию `dir1`.
10. Снимите с директории `dir1` все атрибуты командой
```
chmod 000 dir1
```
и проверьте с её помощью правильность выполнения команды
```
ls -l
```
11. Попытайтесь создать в директории `dir1` файл `file1` командой
```
echo "test" > /home/guest/dir1/file1
```
Объясните, почему вы получили отказ в выполнении операции по созданию файла?
Оцените, как сообщение об ошибке отразилось на создании файла? Проверьте командой
```
ls -l /home/guest/dir1
```
действительно ли файл `file1` не находится внутри директории `dir1`.
12. Заполните таблицу «Установленные права и разрешённые действия» (см. табл. 2.2), выполняя действия от имени владельца директории (файлов), определив опытным путём, какие операции разрешены, а какие нет. Если операция разрешена, занесите в таблицу знак «+», если не разрешена, знак «-».
Замечание 1: при заполнении табл. 2.2 рассматриваются не все атрибуты файлов и директорий, а лишь «первые три»: r, w, x, для «владельца». Остальные атрибуты также важны (особенно при использовании доступа от имени разных пользователей, входящих в те или иные группы). Проверка всех атрибутов при всех условиях значительно увеличила бы таблицу: так 9 атрибутов на директорию и 9 атрибутов на файл дают

218 строк без учёта дополнительных атрибутов, плюс таблица была бы расширена по количеству столбцов, так как все приведённые операции необходимо было бы повторить ещё как минимум для двух пользователей: входящего в группу владельца файла и не входящего в неё.
После полного заполнения табл. 2.2 и анализа полученных данных нам удалось бы выяснить, что заполнение её в таком виде излишне. Можно разделить большую таблицу на несколько малых независимых таблиц. В данном примере предлагается рассмотреть $3+3$ атрибута, т.е. $2^6 = 64$ варианта.
Замечание 2: в ряде действий при выполнении команды удаления файла вы можете столкнуться с вопросом: «удалить защищённый от записи пустой обычный файл dir1/file1?» Обратите внимание, что наличие этого вопроса не позволяет сделать правильный вывод о том, что файл можно удалить. В ряде случаев, при ответе «y» (да) на указанный вопрос, возможно получить другое сообщение: «невозможно удалить dirl /file1: Отказано в доступе».

13. На основании заполненной таблицы определите те или иные минимально необходимые права для выполнения операций внутри директории `dir1`, заполните табл. 2.1.

Таблица 2.1

Минимальные права для совершения операций

Операция	Минимальные права на директорию	Минимальные права на файл
Создание файла		
Удаление файла		
Чтение файла		
Запись в файл		
Переименование файла		
Создание поддиректории		-
Удаление поддиректории		-

Таблица 2.2

Установленные права и разрешённые действия

Права директории	Права файла	Создание файла	Удаление файла	Запись в файл	Чтение файла	Смена директории	Просмотр файлов в директории	Переименование файла	Смена атрибутов файла
d (000)	(000)	-	-	-	-	-	-	-	-
d--x------ (100)	(000)	-	-	-	-	+	-	-	+
drwx------ (700)	-rwx------ (700)	+	+	+	+	+	+	+	+

Таблица 2.3

Установленные права и разрешённые действия для групп

Права директории	Права файла	Создание файла	Удаление файла	Запись в файл	Чтение файла	Смена директории	Просмотр файлов в директории	Переименование файла	Смена атрибутов файла
d (000)	(000)	-	-	-	-	-	-	-	-
d–x– (010)	(000)	-	-	-	-	+	-	-	+
d---rwx--- (070)	---rwx--- (070)	+	+	+	+	+	+	+	+

2.4. Задание 2. Два пользователя

1. Осуществите вход в систему от двух пользователей на двух разных консолях: guest на первой консоли и guest2 на второй консоли.
2. Для обоих пользователей командой `pwd` определите директорию, в которой вы находитесь. Сравните её с приглашениями командной строки.
3. Уточните имя вашего пользователя, его группу, кто входит в неё и к каким группам принадлежит он сам. Определите командами `groups guest` и `groups guest2`, в какие группы входят пользователи guest и guest2. Сравните вывод команды `groups` с выводом команд `id -Gn` и `id -G`.
4. Сравните полученную информацию с содержимым файла `/etc/group`. Просмотрите файл командой
   `cat /etc/group`
5. От имени пользователя guest2 выполните регистрацию пользователя guest2 в группе guest командой
   `newgrp guest`
6. От имени пользователя guest измените права директории `/home/guest`, разрешив все действия для пользователей группы:
   `chmod g+rwx /home/guest`
7. От имени пользователя guest снимите с директории `/home/guest/dir1` все атрибуты командой
   `chmod 000 dirl`
   и проверьте правильность снятия атрибутов.
   Меняя атрибуты у директории `dir1` и файла `file1` от имени пользователя guest и делая проверку от пользователя guest2, заполните табл/ 2.3, определив опытным путём, какие операции разрешены, а какие нет. Если операция разрешена, занесите в таблицу знак «+», если не разрешена, знак «-».
   Сравните табл. 2.2 и табл. 2.3.
   На основании заполненной таблицы определите те или иные минимально необходимые права для выполнения пользователем guest2 операций внутри директории dir1 и заполните табл. 2.1, изменив её название на «Минимальные права для совершения операций от имени пользователей входящих в группу».

2.5. Задание 3. Расширенные атрибуты

1. От имени пользователя guest определите расширенные атрибуты файла `/home/guest/dir1/file1` командой
   `lsattr /home/guest/dir1/file1`
2. Установите командой
   `chmod 600 file1`
   на файл `file1` права, разрешающие чтение и запись для владельца файла.

3. Попробуйте установить на файл `/home/guest/dir1/file1` расширенный атрибут `a` от имени пользователя `guest`:

   ```
   chattr +a /home/guest/dir1/file1
   ```

   В ответ вы должны получить отказ от выполнения операции.
4. Зайдите на третью консоль с правами администратора либо повысьте свои права с помощью команды `su`. Попробуйте установить расширенный атрибут `a` на файл `/home/guest/dir1/file1` от имени суперпользователя:

   ```
   chattr +a /home/guest/dir1/file1
   ```
5. От пользователя guest проверьте правильность установления атрибута:

   ```
   lsattr /home/guest/dir1/file1
   ```
6. Выполните дозапись в файл `file1` слова «test» командой

   ```
   echo "test" /home/guest/dir1/file1
   ```

   После этого выполните чтение файла `file1` командой

   ```
   cat /home/guest/dir1/file1
   ```

   Убедитесь, что слово test было успешно записано в `file1`.
7. Попробуйте удалить файл `file1` либо стереть имеющуюся в нём информацию командой

   ```
   echo "abcd" > /home/guest/dirl/file1
   ```

   Попробуйте переименовать файл.
8. Попробуйте с помощью команды

   ```
   chmod 000 file1
   ```

   установить на файл `file1` права, например, запрещающие чтение и запись для владельца файла. Удалось ли вам успешно выполнить указанные команды?
9. Снимите расширенный атрибут `a` с файла `/home/guest/dirl/file1` от имени суперпользователя командой

   ```
   chattr -a /home/guest/dir1/file1
   ```

   Повторите операции, которые вам ранее не удавалось выполнить. Ваши наблюдения занесите в отчёт.
10. Повторите ваши действия по шагам, заменив атрибут «a» атрибутом «i». Удалось ли вам дозаписать информацию в файл? Ваши наблюдения занесите в отчёт.

В результате выполнения работы вы повысили свои навыки использования интерфейса командой строки (CLI), познакомились на примерах с тем, как используются основные и расширенные атрибуты при разграничении доступа. Имели возможность связать теорию дискреционного разделения доступа (дискреционная политика безопасности) с её реализацией на практике в ОС Linux. Составили наглядные таблицы, поясняющие какие операции возможны при тех или иных установленных правах. Опробовали действие на практике расширенных атрибутов «a» и «i».

2.6. Задание 4. Исследование влияния дополнительных атрибутов

Исследуются механизмы изменения идентификаторов, применение SetUID- и Sticky-битов.

Предлагается рассмотреть работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2.6.1. Подготовка лабораторного стенда

Помимо прав администратора для выполнения части заданий потребуются средства разработки приложений. В частности, при подготовке стенда следует убедиться, что в системе установлен компилятор gcc (для этого, например, можно ввести команду gcc -v). Если же gcc не установлен, то его необходимо установить, например, командой

```
yum install gcc
```

которая определит зависимости и установит следующие пакеты: gcc, cloog-ppl, cpp, glibc-devel, glibc-headers, kernel-headers, libgomp, ppl, cloog-ppl, cpp, gcc, glibc-devel, glibc-headers, kernel-headers, libgomp, libstdc++-devel, mpfr, ppl, glibc, glibc-common, libgcc, libstdc++.

Файловая система, где располагаются домашние директории и файлы пользователей (в частности, пользователя guest), не должна быть смонтирована с опцией nosuid.

Так как программы с установленным битом SetUID могут представлять большую брешь в системе безопасности, в современных системах используются дополнительные механизмы защиты. Проследите, чтобы система защиты SELinux не мешала выполнению заданий работы. Если вы не знаете, что это такое, просто отключите систему запретов до очередной перезагрузки системы командой

```
setenforce 0
```

После этого команда `getenforce` должна выводить Permissive. В этой работе система SELinux рассматриваться не будет.

2.6.2. Компилирование программ

Для выполнения четвёртой части задания вам потребуются навыки программирования, а именно, умение компилировать простые программы, написанные на языке C (C++), используя интерфейс CLI.

Само по себе создание программ не относится к теме, по которой выполняется работа, а является вспомогательной частью, позволяющей увидеть, как реализуются на практике те или иные механизмы дискреционного разграничения доступа. Если при написании (или исправлении существующих) скриптов на bash-е у большинства системных администраторов не

возникает проблем, то процесс компилирования, как показывает практика, вызывает необоснованные затруднения.

Компиляторы, доступные в Linux-системах, являются частью коллекции GNU-компиляторов, известной как GCC (GNU Compiller Collection, подробнее см. `http://gcc.gnu.org`). В неё входят компиляторы языков C, C++, Java, Objective-C, Fortran и Chill. Будем использовать лишь первые два.

Компилятор языка C называется gcc. Компилятор языка C++ называется g++ и запускается с параметрами почти так же, как gcc.

Проверить это можно следующими командами:

```
whereis gcc
whereis g++
```

Первый шаг заключается в превращении исходных файлов в объектный код:

```
gcc -c file.c
```

В случае успешного выполнения команды (отсутствие ошибок в коде) полученный объектный файл будет называться file.o.

Объектные файлы невозможно запускать и использовать, поэтому после компиляции для получения готовой программы объектные файлы необходимо скомпоновать. Компоновать можно один или несколько файлов. В случае использования хотя бы одного из файлов, написанных на C++, компоновка производится с помощью компилятора g++. Строго говоря, это тоже не вполне верно. Компоновка объектного кода, сгенерированного чем бы то ни было (хоть вручную), производится линкером ld, g++ его просто вызывает изнутри. Если же все файлы написаны на языке C, нужно использовать компилятор gcc.

Например, так:

```
gcc -o program file.o
```

В случае успешного выполнения команды будет создана программа `program` (исполняемый файл формата ELF с установленным атрибутом +x).

Компилирование — это процесс. Компилятор gcc (g++) имеет множество параметров, влияющих на процесс компиляции. Он поддерживает различные режимы оптимизации, выбор платформы назначения и пр.

Также возможно использование make-файлов (Makefile) с помощью утилиты `make` для упрощения процесса компиляции.

Такое решение подойдёт лишь для простых случаев. Если говорить про пример выше, то компилирование одного файла из двух шагов можно сократить вообще до одного, например:

```
gcc file.c
```

В этом случае готовая программа будет иметь называние a.out.

Механизм компилирования программ в данной работе не мог быть не рассмотрен потому, что использование программ, написанных на bash, для изучения SetUID- и SetGID- битов, не представляется возможным. Связано это с тем, что любая bash-программа интерпретируется в процессе своего выполнения, т.е. существует сторонняя программа-интерпретатор, которая выполняет считывание файла сценария и выполняет его последовательно.

Сам интерпретатор выполняется с правами пользователя, его запустившего, а значит, и выполняемая программа использует эти права.

При этом интерпретатору абсолютно всё равно, установлены SetUID-, SetGID-биты у текстового файла сценария, атрибут разрешения запуска «x» или нет. Важно, чтобы был установлен лишь атрибут, разрешающий чтение «r».

Также не важно, был ли вызван интерпретатор из командной строки (запуск файла, как bash file1.sh), либо внутри файла была указана строчка `#!/bin/bash`.

Логично спросить: если установление SetUID- и SetGID- битов на сценарий не приводит к нужному результату как с исполняемыми файлами, то что мешает установить эти биты на сам интерпретатор? Ничего не мешает, только их установление приведёт к тому, что, так как владельцем `/bin/bash` является root:

```
ls -l /bin/bash
```

все сценарии, выполняемые с использованием `/bin/bash`, будут иметь возможности суперпользователя — совсем не тот результат, который хотелось бы видеть.

Если сомневаетесь в выше сказанном, создайте простой файл `progl.sh` следующего содержания:

```
#!/bin/bash
/usr/bin/id /usr/bin/whoami
```

и попробуйте поменять его атрибуты в различных конфигурациях.

Подход вида: сделать копию `/bin/bash`, для нее `chown user:users` и потом SUID также плох, потому что это позволит запускать любые команды от пользователя user.

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c:

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
  uid_t uid = geteuid ();
  gid_t gid = getegid ();
  printf ("uid=%d, gid=%d\n", uid, gid);
  return 0;
}
```

3. Скомплилируйте программу и убедитесь, что файл программы создан:

```
gcc simpleid.c -o simpleid
```

4. Выполните программу simpleid:

```
./simpleid
```

5. Выполните системную программу id:

```
id
```

и сравните полученный вами результат с данными предыдущего пункта задания.

6. Усложните программу, добавив вывод действительных идентификаторов:

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
  uid_t real_uid = getuid ();
  uid_t e_uid = geteuid ();

  gid_t real_gid = getgid ();
  gid_t e_gid = getegid () ;

  printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
  printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);

  return 0;
}
```

Получившуюся программу назовите simpleid2.c.

7. Скомпилируйте и запустите simpleid2.c:

```
gcc simpleid2.c -o simpleid2
./simpleid2
```

8. От имени суперпользователя выполните команды:

```
chown root:guest /home/guest/simpleid2
chmod u+s /home/guest/simpleid2
```

9. Используйте `sudo` или повысьте временно свои права с помощью `su`. Поясните, что делают эти команды.
10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
ls -l simpleid2
```

11. Запустите simpleid2 и id:

```
./simpleid2
id
```

Сравните результаты.

12. Проделайте тоже самое относительно SetGID-бита.
13. Создайте программу readfile.c:

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int
main (int argc, char* argv[])
```

```
{
  unsigned char buffer[16];
  size_t bytes_read;
  int i;

  int fd = open (argv[1], O_RDONLY);
  do
  {
    bytes_read = read (fd, buffer, sizeof (buffer));
    for (i =0; i < bytes_read; ++i) printf("%c", buffer[i]);
  }

  while (bytes_read == sizeof (buffer));
  close (fd);
  return 0;
}
```

14. Откомпилируйте её.
    `gcc readfile.c -o readfile`
15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.
16. Проверьте, что пользователь guest не может прочитать файл readfile.c.
17. Смените у программы readfile владельца и установите SetU'D-бит.
18. Проверьте, может ли программа readfile прочитать файл readfile.c?
19. Проверьте, может ли программа readfile прочитать файл `/etc/shadow`? Отразите полученный результат и ваши объяснения в отчёте.

2.6.3. Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории `/tmp`, для чего выполните команду
   `ls -l / | grep tmp`
2. От имени пользователя guest создайте файл file01.txt в директории `/tmp` со словом test:
   `echo "test" > /tmp/file01.txt`
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»:
   ```
   ls -l /tmp/file01.txt
   chmod o+rw /tmp/file01.txt
   ls -l /tmp/file01.txt
   ```
4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл `/tmp/file01.txt`:
   `cat /tmp/file01.txt`
5. От пользователя guest2 попробуйте дозаписать в файл `/tmp/file01.txt` слово test2 командой
   `echo "test2" > /tmp/file01.txt`

Удалось ли вам выполнить операцию?

6. Проверьте содержимое файла командой
   `cat /tmp/file01.txt`
7. От пользователя guest2 попробуйте записать в файл `/tmp/file01.txt` слово test3, стерев при этом всю имеющуюся в файле информацию командой
   `echo "test3" > /tmp/file01.txt`
   Удалось ли вам выполнить операцию?
8. Проверьте содержимое файла командой
   `cat /tmp/file01.txt`
9. От пользователя guest2 попробуйте удалить файл `/tmp/file01.txt` командой
   `rm /tmp/file01.txt`
   Удалось ли вам удалить файл?
10. Повысьте свои права до суперпользователя следующей командой
    `su -`
    и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`:
    `chmod -t /tmp`
11. Покиньте режим суперпользователя командой
    `exit`
12. От пользователя guest2 проверьте, что атрибута `t` у директории `/tmp` нет:
    `ls -l / | grep tmp`
13. Повторите предыдущие шаги. Какие наблюдаются изменения?
14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.
15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`:
    ```
    su -
    chmod +t /tmp
    exit
    ```

3. Мандатное разграничение прав в Linux

3.1. Цели работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux[1].

Проверить работу SELinx на практике совместно с веб-сервером Apache.

3.2. Организация и описание лабораторного стенда

Для проведения указанной лабораторной работы на одно рабочее место требуется компьютер с установленной операционной системой Linux, поддерживающей технологию SELinux.

Предполагается использовать стандартный дистрибутив Linux CentOS с включённой политикой SELinux targeted и режимом enforcing. Для выполнения заданий требуется наличие учётной записи администратора (root) и учётной записи обычного пользователя. Постоянно работать от учётной записи root неправильно с точки зрения безопасности.

3.3. Подготовка лабораторного стенда и методические рекомендации

1. При подготовке стенда обратите внимание, что необходимая для работы и указанная выше политика targeted и режим enforcing используются в данном дистрибутиве по умолчанию, т.е. каких-то специальных настроек не требуется. При этом следует убедиться, что политика и режим включены, особенно когда работа будет проводиться повторно и велика вероятность изменений при предыдущем использовании системы.
2. При необходимости администратор должен разбираться в работе SELinux и уметь как исправить конфигурационный файл `/etc/selinux/config`, так и проверить используемый режим и политику.
3. Необходимо, чтобы был установлен веб-сервер Apache. При установке системы в конфигурации «рабочая станция» указанный пакет не ставится.
4. В конфигурационном файле `/etc/httpd/httpd.conf` необходимо задать параметр ServerName:
   ```
   ServerName test.ru
   ```

[1] При составлении работы использовались материалы [**zaklyakov:2011:selinux:samag:part1**; **zaklyakov:2011:selinux:samag:part2**].

чтобы при запуске веб-сервера не выдавались лишние сообщения об ошибках, не относящихся к лабораторной работе.

5. Также необходимо проследить, чтобы пакетный фильтр был отключён или в своей рабочей конфигурации позволял подключаться к 80-у и 81-у портам протокола tcp.
   Отключить фильтр можно командами
   ```
   iptables -F
   iptables -P INPUT ACCEPT iptables -P OUTPUT ACCEPT
   ```
   либо добавить разрешающие правила:
   ```
   iptables -I INPUT -p tcp --dport 80 -j ACCEPT
   iptables -I INPUT -p tcp --dport 81 -j ACCEPT
   iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
   iptables -I OUTPUT -p tcp --sport 81 -j ACCEPT
   ```
6. Обратите внимание, что данные правила не являются «точными» и рекомендуемыми на все случаи жизни, они лишь позволяют правильно организовать работу стенда.
7. В работе специально не делается акцент, каким браузером (или какой консольной программой) будет производиться подключение к веб-серверу. По желанию могут использоваться разные программы, такие как консольные `links`, `lynx`, `wget` и графические `konqueror`, `opera`, `firefox` или др.

3.3.1. Задание лабораторной работы

1. Войдите в систему с полученными учётными данными и убедитесь, что SELinux работает в режиме enforcing политики targeted с помощью команд `getenforce` и `sestatus`.
2. Обратитесь с помощью браузера к веб-серверу, запущенному на вашем компьютере, и убедитесь, что последний работает:
   ```
   service httpd status
   ```
   или
   ```
   /etc/rc.d/init.d/httpd status
   ```
   Если не работает, запустите его так же, но с параметром `start`.
3. Найдите веб-сервер Apache в списке процессов, определите его контекст безопасности и занесите эту информацию в отчёт. Например, можно использовать команду
   ```
   ps auxZ | grep httpd
   ```
   или
   ```
   ps -eZ | grep httpd
   ```
4. Посмотрите текущее состояние переключателей SELinux для Apache с помощью команды
   ```
   sestatus -bigrep httpd
   ```
   Обратите внимание, что многие из них находятся в положении «off».
5. Посмотрите статистику по политике с помощью команды `seinfo`, также определите множество пользователей, ролей, типов.

6. Определите тип файлов и поддиректорий, находящихся в директории `/var/www`, с помощью команды
   ```
   ls -lZ /var/www
   ```
7. Определите тип файлов, находящихся в директории `/var/www/html`:
   ```
   ls -lZ /var/www/html
   ```
8. Определите круг пользователей, которым разрешено создание файлов в директории `/var/www/html`.
9. Создайте от имени суперпользователя (так как в дистрибутиве после установки только ему разрешена запись в директорию) html-файл `/var/www/html/test.html` следующего содержания:
   ```
   <html>
   <body>test</body>
   </html>
   ```
10. Проверьте контекст созданного вами файла. Занесите в отчёт контекст, присваиваемый по умолчанию вновь созданным файлам в директории `/var/www/html`.
11. Обратитесь к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Убедитесь, что файл был успешно отображён.
12. Изучите справку `man httpd_selinux` и выясните, какие контексты файлов определены для `httpd`. Сопоставьте их с типом файла `test.html`. Проверить контекст файла можно командой `ls -Z`.
    ```
    ls -Z /var/www/html/test.html
    ```
    Рассмотрим полученный контекст детально. Обратите внимание, что так как по умолчанию пользователи CentOS являются свободными от типа (unconfined в переводе с англ. означает свободный), созданному нами файлу `test.html` был сопоставлен SELinux, пользователь unconfined_u. Это первая часть контекста.
    Далее политика ролевого разделения доступа RBAC используется процессами, но не файлами, поэтому роли не имеют никакого значения для файлов. Роль object_r используется по умолчанию для файлов на «постоянных» носителях и на сетевых файловых системах. (В директории `/proc` файлы, относящиеся к процессам, могут иметь роль system_r. Если активна политика MLS, то могут использоваться и другие роли, например, secadm_r. Данный случай мы рассматривать не будем, как и предназначение :s0).
    Тип httpd_sys_content_t позволяет процессу httpd получить доступ к файлу. Благодаря наличию последнего типа мы получили доступ к файлу при обращении к нему через браузер.
13. Измените контекст файла `/var/www/html/test.html` с httpd_sys_content_t на любой другой, к которому процесс httpd не должен иметь доступа, например, на samba_share_t:
    ```
    chcon -t samba_share_t /var/www/html/test.html
    ls -Z /var/www/html/test.html
    ```
    После этого проверьте, что контекст поменялся.

14. Попробуйте ещё раз получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1/test.html`. Вы должны получить сообщение об ошибке:
```
Forbidden
You don't have permission to access /test.html on this server.
```
15. Проанализируйте ситуацию. Почему файл не был отображён, если права доступа позволяют читать этот файл любому пользователю?
```
ls -l /var/www/html/test.html
```
Просмотрите log-файлы веб-сервера Apache. Также просмотрите системный лог-файл:
```
tail /var/log/messages
```
Если в системе окажутся запущенными процессы setroubleshootd и audtd, то вы также сможете увидеть ошибки, аналогичные указанным выше, в файле `/var/log/audit/audit.log`. Проверьте это утверждение самостоятельно.
16. Попробуйте запустить веб-сервер Apache на прослушивание TCP-порта 81 (а не 80, как рекомендует IANA и прописано в `/etc/services`). Для этого в файле `/etc/httpd/httpd.conf` найдите строчку `Listen 80` и замените её на `Listen 81`.
17. Выполните перезапуск веб-сервера Apache. Произошёл сбой? Поясните почему?
18. Проанализируйте лог-файлы:
```
tail -nl /var/log/messages
```
Просмотрите файлы `/var/log/http/error_log`, `/var/log/http/access_log` и `/var/log/audit/audit.log` и выясните, в каких файлах появились записи.
19. Выполните команду
```
semanage port -a -t http_port_t -p tcp 81
```
После этого проверьте список портов командой
```
semanage port -l | grep http_port_t
```
Убедитесь, что порт 81 появился в списке.
20. Попробуйте запустить веб-сервер Apache ещё раз. Поняли ли вы, почему он сейчас запустился, а в предыдущем случае не смог?
21. Верните контекст httpd_sys_content__t к файлу /var/www/html/test.html:
```
chcon -t httpd_sys_content_t /var/www/html/test.html
```
После этого попробуйте получить доступ к файлу через веб-сервер, введя в браузере адрес `http://127.0.0.1:81/test.html`.
Вы должны увидеть содержимое файла — слово «test».
22. Исправьте обратно конфигурационный файл apache, вернув `Listen 80`.
23. Удалите привязку http_port_t к 81 порту:
```
semanage port -d -t http_port_t -p tcp 81
```
и проверьте, что порт 81 удалён.
24. Удалите файл `/var/www/html/test.html`:
```
rm /var/www/html/test.html
```

4. Элементы криптографии

4.1. Однократное гаммирование

4.1.1. Цель работы

Освоить на практике применение режима однократного гаммирования[1].

4.1.2. Указание к работе

Предложенная Г. С. Вернамом так называемая «схема однократного использования (гаммирования)» (рис. 4.1) является простой, но надёжной схемой шифрования данных.

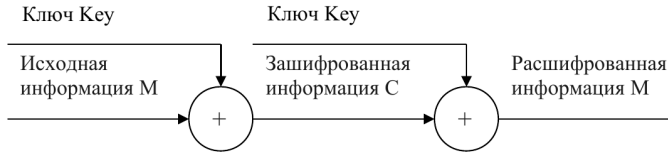

Рис. 4.1. Схема однократного использования Вернама

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования.

В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.

Наложение гаммы по сути представляет собой выполнение операции *сложения по модулю 2 (XOR)* (обозначаемая знаком $\oplus$) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$.

Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное

[1] При составлении работы использовалось пособие [**gultaeva:2012**].

значение, а шифрование и расшифрование выполняется одной и той же программой.

Если известны ключ и открытый текст, то задача нахождения шифротекста заключается в применении к каждому символу открытого текста следующего правила:

$$C_i = P_i \oplus K_i, \tag{4.1}$$

где C_i — i-й символ получившегося зашифрованного послания, P_i — i-й символ открытого текста, K_i — i-й символ ключа, $i = \overline{1, m}$. Размерности открытого текста и ключа должны совпадать, и полученный шифротекст будет такой же длины.

Если известны шифротекст и открытый текст, то задача нахождения ключа решается также в соответствии с (4.1), а именно, обе части равенства необходимо сложить по модулю 2 с P_i:

$$C_i \oplus P_i = P_i \oplus K_i \oplus P_i = K_i,$$
$$K_i = C_i \oplus P_i.$$

Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов.

К. Шеннон доказал абсолютную стойкость шифра в случае, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения. Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении C все различные ключевые последовательности K возможны и равновероятны, а значит, возможны и любые сообщения P.

Необходимые и достаточные условия абсолютной стойкости шифра:

– полная случайность ключа;
– равенство длин ключа и открытого текста;
– однократное использование ключа.

Рассмотрим пример.

Ключ Центра:

```
05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54
```

Сообщение Центра:

```
Штирлиц – Вы Герой!!
D8 F2 E8 F0 EB E8 F6 20 2D 20 C2 FB 20 C3 E5 F0 EE E9 21 21
```

Зашифрованный текст, находящийся у Мюллера:

```
DD FE FF 8F E5 A6 C1 F2 B9 30 CB D5 02 94 1A 38 E5 5B 51 75
```

Дешифровальщики попробовали ключ:

```
05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 55 F4 D3 07 BB BC 54
```

и получили текст:

```
D8 F2 E8 F0 EB E8 F6 20 2D 20 C2 FB 20 C1 EE EB E2 E0 ED 21
Штирлиц - Вы Болван!
```

Другие ключи дадут лишь новые фразы, пословицы, стихотворные строфы, словом, всевозможные тексты заданной длины.

4.1.3. Задание

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

1) определить вид шифротекста при известном ключе и известном открытом тексте;
2) определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

4.1.4. Контрольные вопросы

1. Поясните смысл однократного гаммирования.
2. Перечислите недостатки однократного гаммирования.
3. Перечислите преимущества однократного гаммирования.
4. Почему длина открытого текста должна совпадать с длиной ключа?
5. Какая операция используется в режиме однократного гаммирования, назовите её особенности?
6. Как по открытому тексту и ключу получить шифротекст?
7. Как по открытому тексту и шифротексту получить ключ?
8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

4.2. Шифрование (кодирование) различных исходных текстов одним ключом

4.2.1. Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

4.2.2. Указание к работе

Исходные данные.
Две телеграммы Центра:
P_1 = `НаВашисходящийот1204`
P_2 = `ВСевернымфилиалБанка`
Ключ Центра длиной 20 байт:
`K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 OB B2 70 54`

Режим шифрования однократного гаммирования одним ключом двух видов открытого текста реализуется в соответствии со схемой, приведённой на рис. 4.2.

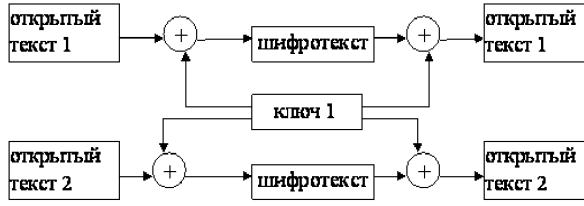

Рис. 4.2. Общая схема шифрования двух различных текстов одним ключом

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$\begin{aligned} C_1 &= P_1 \oplus K, \\ C_2 &= P_2 \oplus K. \end{aligned} \tag{4.2}$$

Открытый текст можно найти в соответствии с (4.2), зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства (4.2) складываются по модулю 2. Тогда с учётом свойства операции XOR

$$1 \oplus 1 = 0, \quad 1 \oplus 0 = 1 \tag{4.3}$$

получаем:

$$C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2.$$

Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар $C_1 \oplus C_2$ (известен вид обеих шифровок). Тогда зная P_1 и учитывая (4.3), имеем:

$$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2. \tag{4.4}$$

Таким образом, злоумышленник получает возможность определить те символы сообщения P_2, которые находятся на позициях известного шаблона сообщения P_1. В соответствии с логикой сообщения P_2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P_2. Затем вновь используется (4.4) с подстановкой вместо P_1 полученных на предыдущем шаге новых символов сообщения P_2. И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

4.2.3. Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P_1 и P_2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C_1 и C_2 обоих текстов P_1 и P_2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

4.2.4. Контрольные вопросы

1. Как, зная один из текстов (P_1 или P_2), определить другой, не зная при этом ключа?
2. Что будет при повторном использовании ключа при шифровании текста?
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?
4. Перечислите недостатки шифрования одним ключом двух открытых текстов.
5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Учебно-методический комплекс

Рекомендуется для направлений подготовки
01.04.02 «Прикладная математика и информатика»
02.04.02 — Фундаментальная информатика и информационные технологии.

Квалификация (степень) выпускника: магистр

Программа дисциплины

Цели и задачи дисциплины

Цель курса — введение учащихся в предметную область защиты современных систем и сетей телекоммуникаций.

В процессе преподавания курса решаются следующие задачи:

– изучаются основные уязвимости операционных систем;
– даётся понятие о защите компьютерных сетей.

Место дисциплины в структуре основной образовательной программы

Цикл, к которому относится дисциплина: базовая часть блока Б.1 «Дисциплины (модули)».

Требования к входным знаниям, умениям и компетенциям студента: компетенции и навыки, полученные при изучении дисциплин «Операционные системы», «Сетевые технологии», в объёме бакалавриата.

Дисциплины, для которых данная дисциплина является предшествующей: НИРм, выпускная квалификационная работа.

Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ОК: 1; ОПК: 3, 4; ПК: 2 (для направления 02.04.02):

ОК-1 способностью к абстрактному мышлению, анализу, синтезу;

ОПК-3 способностью использовать и применять углубленные теоретические и практические знания в области фундаментальной информатики и информационных технологий;

ОПК-4 способностью самостоятельно приобретать и использовать в практической деятельности новые знания и умения, в том числе, в новых областях знаний, непосредственно не связанных со сферой деятельности, расширять и углублять своё научное мировоззрение;

ПК-2 способностью использовать углубленные теоретические и практические знания в области информационных технологий и прикладной математики, фундаментальных концепций и системных методологий, международных и профессиональных стандартов в области информационных технологий;

ОПК-3, 4 (для направления 01.04.02):

ОПК3 способностью самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе, в новых областях знаний, непосредственно не связанных со сферой деятельности, расширять и углублять своё научное мировоззрение;
ОПК4 способностью использовать и применять углубленные знания в области прикладной математики и информатики.

В результате изучения дисциплины студент должен:

знать:

- основные уязвимости операционных систем;
- основные уязвимости компьютерных сетей;
- принципы дискреционного и мандатного разграничения прав доступа в информационных системах;
- теоретические основы и общие принципы использования профессиональных областей: системное администрирование, управление безопасностью информационных систем;

уметь:

- применять на практике международные и профессиональные стандарты информационных технологий, современные парадигмы и методологии, инструментальные и вычислительные средства, связанные с безопасностью информационных систем;

владеть:

- навыками применения и настройки дискреционного и мандатного разграничения прав доступа в информационных системах;
- навыками применение режима однократного гаммирования, как элемента криптографии.

Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 3 зачетные единицы.

№	Вид учебной работы	Всего часов	Семестры
			1 или 3
1.	**Аудиторные занятия (всего)**	**34**	**34**
	В том числе:		
1.1	Лекции		
1.2	Прочие занятия	34	34
	В том числе:		
1.2.1	*Практические занятия (ПЗ)*	-	-

1.2.2	*Семинары (С)*	-	-
1.2.3	*Лабораторные работы (ЛР)*	34	34
1.2.4	*Из них в интерактивной форме (ИФ)*	34	34
2	**Самостоятельная работа студентов**	**74**	**74**
	В том числе:		
2.1	Курсовой проект (работа)	-	-
2.2	Расчетно-графические работы	-	-
2.3	Реферат	-	-
2.4	Подготовка и прохождение промежуточной аттестации	27 или 36	27 или 36
2.5	*Другие виды самостоятельной работы:*		
2.5.1	Самостоятельная проработка дополнительных материалов по дисциплине	47 или 36	47 или 36
3.	**Общая трудоёмкость (ак.часов)**	**108**	**108**
4.	**Общая трудоёмкость (зач. ед.)**	**3**	**3**

Содержание дисциплины

Содержание разделов дисциплины

Тема 1. Основы безопасности сетевых информационных технологий

1. Основы безопасности сетевых информационных технологий.
   - Основные понятия информационной безопасности.
   - Типовая IP-сеть организации.
   - Классификация уязвимостей и атак.
   - Защитные механизмы и средства обеспечения безопасности.
2. Безопасность уровня сетевого взаимодействия.
   - Базовые принципы сетевого взаимодействия. Модель OSI.
   - Архитектура TCP/IP.
   - Безопасность физического и канального уровней модели OSI. Сетевые анализаторы и «снифферы».
   - Проблемы безопасности протокола разрешения адресов ARP.
   - Безопасность сетевого уровня модели OSI. Протоколы IP и ICMP.
   - Меры защиты сетевого уровня. Протокол IPSEC.
   - Безопасность транспортного уровня модели OSI. Протоколы TCP и UDP.

- Меры защиты транспортного уровня. Протоколы SSL/TLS, SSH.
- Система обнаружения атак Snort.
- Проблемы безопасности протоколов прикладного уровня (Telnet, FTP, HTTP, SMTP).

3. Понятие о моделях безопасности ОС.
4. Понятие о безопасность баз данных.

Тема 2. Защита информации в компьютерных сетях

1. Принципы обеспечения безопасности приложений.
   - Уязвимости прикладного программного обеспечения.
   - Сетевой сканер Nessus.
   - Варианты решений по обеспечению безопасности сети организации.
2. Применение межсетевых экранов для защиты корпоративных сетей.
   - Место и роль межсетевых экранов в корпоративных сетях. Типовая корпоративная сеть. Уровни информационной инфраструктуры. Понятие межсетевых экранов. Защитные механизмы, реализуемые межсетевыми экранами.
   - Обзор документов RFC, имеющих отношение к межсетевым экранам, основные термины и определения. Типы межсетевых экранов.
   - Фильтрация пакетов. Параметры фильтрации. Правила фильтрации. Реализация пакетных фильтров. Понятие демилитаризованной зоны. Особенности фильтрации различных типов трафика.
   - Пакетный фильтр на базе ОС Linux. Архитектура и схема работы. Управление правилами фильтрации с помощью утилит ipchains и iptables.
   - Пакетный фильтр на базе ОС Windows 2000. Служба RRAS. Программа управления службой RRAS.
   - Шлюзы уровня соединения. Протокол SOCKS. Трансляция адресов. Типы трансляции. Реализация трансляции адресов в ОС Linux. Реализация трансляции адресов в ОС Windows 2000.
   - Шлюзы прикладного уровня. Сервер SQUID, принципы работы, варианты конфигурации.
   - Расположение межсетевых экранов в корпоративной сети. Особенности фильтрации служб прикладного уровня DNS, FTP, SMTP.
   - Противодействие сетевым атакам при помощи межсетевых экранов. Интеграция межсетевых экранов с другими средствами защиты. Достоинства и недостатки межсетевых экранов как средств защиты.

Тема 3. Криптография

1. Место и роль криптографии в обеспечении безопасности информационных технологий.
   - Актуальность проблемы безопасности сетевых технологий. Место и роль криптографических методов и средств в системах управления и

электронной коммерции. Задачи, решаемые средствами криптографической защиты информации: обеспечение конфиденциальности, целостности и аутентичности данных, разграничение ответственности, аутентификация абонентов.

2. Криптографические примитивы и механизмы.
   – Электронные цифровые подписи. Механизмы цифровой подписи. Криптографические хэш-функции: SHA-1, SHA-256, 512, ГОСТ Р 34.10 94. Основные определения и классификация. Схемы цифровой подписи с приложением. Схемы цифровой подписи с восстановлением сообщения. Атаки на схемы подписи. RSA и родственные схемы. Схема подписи DSA и родственные схемы. Схема подписи Эль-Гамаля. ГОСТ Р 34.11 94. (2001 г.).
   – Техники управления ключами. Основные концепции. Классификация ключей по типам алгоритмов и последующему использованию. Цели, угрозы и политики управления ключами. Простые модели установления ключей. Роли третьих сторон. Техника управления использованием ключей, ограничения на их использование. Техника контроля использования асимметричных ключей.
   – Проблемы жизненного цикла ключей. Требования защиты в течение срока эксплуатации. Жизненный цикл управления ключами.
3. Теоретические основы инфраструктуры открытых ключей.
   – Концепция инфраструктуры открытых ключей. Основные термины и определения. Компоненты PKI и их функции: орган сертификации, органы регистрации, владельцы сертификатов, клиенты и клиентское программное обеспечение, хранилище сертификатов.
   – Управление ключами при наличии множества доменов. Доверие между двумя доменами. Модели доверия при наличии различных органов сертификации. Цепочки сертификатов и сертификационные пути. Доверие с разделенными доменами. Строгая иерархическая модель доверия. Реверсивные сертификаты и обобщенные модели. Ограничения в моделях доверия. Распределение и отзыв сертификатов.
4. Практические аспекты инфраструктуры открытых ключей.
   – Жизненный цикл сертификатов. Процессы генерирования, отзыва, приостановки действия и истечения сроков действия сертификатов.
   – Форматы данных. Форматы сертификатов. Перечень отозванных сертификатов (CRL). Проверка пути сертификации. Форматы сообщения транзакции. Транзакции PKI.
   – Отзыв сертификатов. Механизмы периодической публикации. Перечни отозванных сертификатов (CRLs). Полные и разделенные CRLs. CRL органов сертификации. Усовершенствованные разделенные CRL и CRL с переадресацией. Дельта CRL. Косвенные CRL. Механизмы онлайновых запросов. Онлайновый протокол проверки статуса сертификата (OCSP). Другие возможности отзыва. Характеристики, масштабируемость и своевременность.
   – Распространение информации в PKI. Публикация и хранилища (X.500 и LDAP, DNS, FTP, сервер OCSP, WEB сервера, корпоратив-

ные базы данных). Междоменные хранилища. Прямой доступ. Коллективно используемое хранилище. Междоменное дублирование. Пограничное хранилище.
  - Проблемы, возникающие при реализации PKI. Резервное копирование и восстановление ключей. Дублирование и депонирование ключей. Поддержка неотказуемости. Подписывание индивидуального ключа. Обновление и поддержание хронологий ключевых пар. Отзыв сертификата. Перекрестная сертификация.
5. Развертывание инфраструктуры открытых ключей.
  - Определение целей и путей развертывания PKI. Определение политик безопасности. Планирование и создание инфраструктуры. Управление и администрирование PKI.
  - Управление ключами (использование ключей сеанса, использование открытых/индивидуальных ключевых пар, обмен криптографическими ключами, сохранение ключей сеанса, обмен открытыми ключами, обмен ключами сеанса). Пример трёхфазного протокола обмена.
6. Протоколы аутентификации. Протокол Kerberos.

Разделы дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами

№ п/п	Наименование обеспечиваемых (последующих) дисциплин	№ № разделов данной дисциплины, необходимых для изучения обеспечиваемых (последующих) дисциплин		
		1	2	3
1.	НИРм	+	+	+
2.	ВКР	+	+	+

Разделы дисциплин и виды занятий

№ п/п	**Наименование раздела дисциплины**	**Лекц.**	**Практические занятия и лабораторные работы**			**СРС**	**Всего час.**
			ПЗ/С	**ЛР**	**Из них в ИФ**		
1.	Основы безопасности сетевых информационных технологий					12	**12**

2.	Защита информации в компьютерных сетях			24	*24*	20	**44**
3.	Криптография			8	*8*	15	**23**
4.	Контроль знаний			2	*2*	27	**29**
Итого:		**0**	**0**	**34**	***34***	**74**	**108**

Описание интерактивных занятий

№ п/п	**№ р/д**	**Тема интерактивного занятия**	**Вид занятия**	**Труд. (час.)**
1	2	Установка и конфигурация операционной системы на виртуальную машину	Лаб. раб., выполняемая малой группой (2–3 чел.)	8
2	2	Дискреционное разграничение прав в Linux	Лаб. раб., выполняемая малой группой (2–3 чел.)	8
3	2	Мандатное разграничение прав в Linux	Лаб. раб., выполняемая малой группой (2–3 чел.)	8
4	3	Элементы криптографии	Лаб. раб., выполняемая малой группой (2–3 чел.)	8

Лабораторный практикум

№ п/п	№ р/д	Наименование лабораторных работ	Труд. (час.)
1.	2	Установка и конфигурация операционной системы на виртуальную машину	8
2.	2	Дискреционное разграничение прав в Linux	8
3.	2	Мандатное разграничение прав в Linux	8
4.	3	Элементы криптографии	8
5.	1-3	Контроль знаний	2
Итого:			**34**

Практические занятия (семинары)

Практические занятия (семинары) не предусмотрены.

Примерная тематика курсовых проектов (работ)

Курсовые работы не предусмотрены.

Учебно-методическое и информационное обеспечение дисциплины

а) **Основная литература**

1. Загежда Д. П., Ивашко А. М. Основы безопасности информационных систем. — М.: Горячая линия — Телеком, 2000. — 452 с.
2. И. Д. Медведовский, П. В. Семьянов, Д. Г. Леонов Атака на Internet. — Издательство ДМК. — 1999.
3. Мэйволд Э. Безопасность сетей. Эком, 2006 г., 528 с. — http://www.intuit.ru/department/security/netsec/
4. Полянская О.Ю., Горбатов В.С. Инфраструктуры открытых ключей. БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2007. - http://www.intuit.ru/department/security/pki/
5. Галатенко В.А. Основы информационной безопасности. Интернет-университет информационных технологий - ИНТУИТ.ру, 2008 г., 208 с. - http://www.intuit.ru/department/security/secbasics/
6. Галатенко В.А. Стандарты информационной безопасности. Интернет-университет информационных технологий - ИНТУИТ.ру, 2005. - http://www.intuit.ru/department/security/secst/

б) **Дополнительная литература**

1. Шумский А. А. Системный анализ в защите информации. — Учебное пособие для вузов. — М.: Гелиос АРВ, 2005. — 224 с.

в) **Программное обеспечение:** ОС Linux (CentOS), VirtualBox.

г) **Базы данных, информационно-справочные и поисковые системы:** не требуются.

Материально-техническое обеспечение дисциплины

Москва, ул. Орджоникидзе, д.3, корп. 1, 5, лаборатория «Управление инфокоммуникациями»: ауд. 110: комплект жидко-кристаллический дисплей Sharp PNL702B, Монитор 24” Acer V243HAOBD, системный блок (процессор Intel Core i7-2600 OEM <3.40GHz, 8Mb, 95W, LGA1155(Sandy Bridge)>, 16GB ОП, HDD 2 TB), ноутбук Toshiba Satellite 17/300GB Intel Core2 2.4 GHz (9 шт.); ауд. 116: проектор DMS800 с интерактивной доской Board

1077, HP xw7800, Intel Core2 2.4 GHz (8 шт.); дисплейные классы ДК3, ДК4, ДК5, ДК6, ДК7, Intel Core i3-550 3.2 GHz – 60 шт.

Методические рекомендации по организации изучения дисциплины

Учебным планом на изучение дисциплины отводится один семестр. Промежуточный контроль знаний предусматривает: отчёты по лабораторному практикуму. В качестве итогового контроля знаний предусмотрен экзамен в форме контрольного тестирования.

Дисциплина разбита на 2 модуля:

Первый модуль трудоемкостью в 2 кредита составляют:

1) теоретический материал на темы: Основы безопасности сетевых информационных технологий. Основные понятия информационной безопасности. Типовая IP-сеть организации. Классификация уязвимостей и атак. Защитные механизмы и средства обеспечения безопасности. Безопасность уровня сетевого взаимодействия. Безопасность физического и канального уровней модели OSI. Сетевые анализаторы и «снифферы». Проблемы безопасности протокола разрешения адресов ARP. Безопасность сетевого уровня модели OSI. Протоколы IP и ICMP. Меры защиты сетевого уровня. Протокол IPSEC. Безопасность транспортного уровня модели OSI. Протоколы TCP и UDP. Меры защиты транспортного уровня. Протоколы SSL/TLS, SSH. Система обнаружения атак Snort. Проблемы безопасности протоколов прикладного уровня (Telnet, FTP, HTTP, SMTP). Понятие о моделях безопасности ОС.
2) выполнение лабораторных работ в интерактивной форме по разделу 2.

В конце этого модуля проводится промежуточный контроль знаний.

Второй модуль трудоемкостью в 1 кредит составляют:

1) теоретический материал на темы: Место и роль криптографии в обеспечении безопасности информационных технологий. Криптографические примитивы и механизмы. Электронные цифровые подписи. Механизмы цифровой подписи. Криптографические хэш-функции: SHA-1, SHA-256, 512, ГОСТ Р 34.10 94. Основные определения и классификация. Схемы цифровой подписи с приложением. Схемы цифровой подписи с восстановлением сообщения. Техники управления ключами. Основные концепции. Проблемы жизненного цикла ключей. Теоретические основы инфраструктуры открытых ключей. Концепция инфраструктуры открытых ключей (Public Key Infrastructure — PKI). Практические аспекты инфраструктуры открытых ключей. Жизненный цикл сертификатов. Процессы генерирования, отзыва, приостановки действия и истечения сроков действия сертификатов. Развертывание инфраструктуры открытых ключей. Протоколы аутентификации. Протокол Kerberos.
2) выполнение лабораторных работ в интерактивной форме по разделу 3.

В конце этого модуля проводится итоговый контроль знаний.

Фонды оценочных средств

Примерные тестовые задания

1. Kerberos обеспечивает следующие возможности:
(a) невозможность отказа
(b) масштабируемость
(c) прозрачность

2. Агрессивное потребление ресурсов является угрозой:
(a) доступности
(b) конфиденциальности
(c) целостности

3. Атака «man in the middle» является
(a) пассивной
(b) активной
(c) может быть как активной, так и пассивной

4. Аутентификация на основе пароля, переданного по сети в зашифрованном виде и снабженного открытой временной меткой, плоха, потому что не обеспечивает защиты от:
(a) перехвата
(b) воспроизведения
(c) атак на доступность

5. Аутентификация на основе пароля, переданного по сети в зашифрованном виде, плоха, потому что не обеспечивает защиты от:
(a) перехвата
(b) воспроизведения
(c) атак на доступность

6. Аутентификация — это
(a) невозможность несанкционированного доступа к данным
(b) подтверждение того, что информация получена из законного источника законным получателем
(c) невозможность несанкционированного просмотра и модификации информации

7. Большинство людей не совершают противоправных действий потому, что это:
(a) осуждается и/или наказывается обществом
(b) технически невозможно
(c) сулит одни убытки

8. В Kerberos используется:
(a) исключительно симметричное шифрование
(b) исключительно асимметричное шифрование
(c) как симметричное, так и асимметричное шифрование

9. В законопроекте «О совершенствовании информационной безопасности» (США, 2001 год) особое внимание обращено на:
(a) системы электронной коммерции
(b) инфраструктуру для электронных цифровых подписей
(c) средства электронной аутентификации

10. В рамках программы безопасности нижнего уровня определяются:
(a) совокупность целей безопасности
(b) набор используемых механизмов безопасности
(c) наиболее вероятные угрозы безопасности

11. В число архитектурных принципов, направленных на обеспечение высокой доступности информационных сервисов, входят:
(a) разделение обязанностей
(b) модульность архитектуры
(c) ориентация на простоту решений

12. В число возможных стратегий нейтрализации рисков входят:
(a) ликвидация риска
(b) игнорирование риска
(c) принятие риска

13. В число возможных стратегий нейтрализации рисков входят:
(a) уменьшение риска
(b) сокрытие риска
(c) афиширование риска

14. В число классов мер процедурного уровня входят:
(a) управление персоналом
(b) управление персоналками
(c) реагирование на нарушения режима безопасности

15. В число классов функциональных требований «Общих критериев» входят:
(a) анонимность
(b) приватность
(c) связь

16. В число мер, позволяющих структурировать средства достижения информационной безопасности, входят:
(a) законодательные меры
(b) меры обеспечения доступности
(c) профилактические меры

17. В число направлений повседневной деятельности на процедурном уровне входят:
(a) резервное копирование
(b) управление носителями
(c) изготовление резервных носителей

18. В число направлений физической защиты входят:
(a) противопожарные меры
(b) межсетевое экранирование
(c) контроль защищенности

19. В число принципов физической защиты входят:
(a) беспощадный отпор
(b) непрерывность защиты в пространстве и времени
(c) минимизация защитных средств

20. В число универсальных сервисов безопасности входят:
(a) шифрование
(b) средства построения виртуальных частных сетей
(c) туннелирование

Перечень тем для контроля знаний

1. Электронные цифровые подписи. Механизмы цифровой подписи.
2. Безопасность физического и канального уровней модели OSI. Сетевые анализаторы и «снифферы».
3. Техники управления ключами. Основные концепции. Классификация ключей по типам алгоритмов и последующему использованию. Проблемы жизненного цикла клю-чей. Требования защиты в течение срока эксплуатации. Жизненный цикл управления ключами.
4. Модели безопасности ОС.
5. Применение межсетевых экранов для защиты корпора-тивных сетей.
6. Пакетный фильтр на базе ОС Linux.
7. Шлюзы прикладного уровня. Противодействие сетевым атакам при помощи межсетевых экранов.
8. Система PGP.
9. Системы обнаружения атак.
10. Протокол Kerberos.

Календарный план

Неделя	Лабораторные занятия	Число часов
1–4	Лабораторная работа № 1	8
5–8	Лабораторная работа № 2	8
9–12	Лабораторная работа № 3	8
12–16	Лабораторная работа № 4	8
17	Контроль знаний	2

Балльно-рейтинговая система

Рейтинговая система оценки знаний студентов

Раздел	Тема	Формы контроля уровня освоения ООП			Баллы темы	Баллы раздела
		Выполнение ЛР	Интерактивная работа (доклад + презентация)	Итоговый контроль знаний (тест)		
Основы безопасности сетевых информационных технологий	Основы безопасности сетевых информационных технологий	15	5	5	25	25
Защита информации в компьютерных сетях.	Принципы обеспечения безопасности приложений	30	5	5	40	40
	Применение межсетевых экранов для защиты корпоративных сетей					
Криптография	Криптографические примитивы и механизмы	15	5	5	25	35
	Основы инфраструктуры открытых ключей		5	5	10	
	Протоколы аутентификации					
Итого:		60	20	20	100	100

Таблица соответствия баллов и оценок

Баллы БРС	Традиционные оценки в РФ	Баллы для перевода оценок	Оценки	Оценки ECTS
86–100	5	95–100 86–94	5+ 5	A B
69–85	4	69–85	4	C
51–68	3	61–68 51–60	3+ 3	D E
0–50	2	31–50 0–30	2+ 2	FX F

Правила применения БРС

1. Раздел (тема) учебной дисциплины считаются освоенными, если студент набрал более 50 % от возможного числа баллов по этому разделу (теме).
2. Студент не может быть аттестован по дисциплине, если он не освоил все темы и разделы дисциплины, указанные в сводной оценочной таблице дисциплины.
3. По решению преподавателя и с согласия студентов, не освоивших отдельные разделы (темы) изучаемой дисциплины, в течение учебного семестра могут быть повторно проведены мероприятия текущего контроля успеваемости или выданы дополнительные учебные задания по этим темам или разделам. При этом студентам за данную работу засчитывается минимально возможный положительный балл (51 % от максимального балла).
4. При выполнении студентом дополнительных учебных заданий или повторного прохождения мероприятий текущего контроля полученные им баллы засчитываются за конкретные темы. Итоговая сумма баллов не может превышать максимального количества баллов, установленного по данным темам (в соответствии с приказом Ректора № 564 от 20.06.2013). По решению преподавателя предыдущие баллы, полученные студентом по учебным заданиям, могут быть аннулированы.
5. График проведения мероприятий текущего контроля успеваемости формируется в соответствии с календарным планом курса. Студенты обязаны сдавать все задания в сроки, установленные преподавателем.
6. Время, которое отводится студенту на выполнение мероприятий текущего контроля успеваемости, устанавливается преподавателем. По завершении отведённого времени студент должен сдать работу препода-

вателю, вне зависимости от того, завершена она или нет.

7. Использование источников (в том числе конспектов лекций и лабораторных работ) во время выполнения контрольных мероприятий возможно только с разрешения преподавателя.
8. Отсрочка в прохождении мероприятий текущего контроля успеваемости считается уважительной только в случае болезни студента, что подтверждается наличием у него медицинской справки, заверенной круглой печатью в поликлинике № 25, предоставляемой преподавателю не позднее двух недель после выздоровления. В этом случае выполнение контрольных мероприятий осуществляется после выздоровления студента в срок, назначенный преподавателем. В противном случае отсутствие студента на контрольном мероприятии признается неуважительным.
9. Студент допускается к итоговому контролю знаний с любым количеством баллов, набранных в семестре, но при условии, что у студента имеется теоретическая возможность получить за весь курс не менее 31 балла.
10. Итоговая контроль знаний оценивается из 20 баллов независимо от числа баллов за семестр.
11. Если в итоге за семестр студент получил менее 31 балла, то ему выставляется оценка F и студент должен повторить эту дисциплину в установленном порядке. Если же в итоге студент получил 31–50 баллов, т. е. FX, то студенту разрешается добор необходимого (до 51) количества баллов путём повторного одноразового выполнения предусмотренных контрольных мероприятий, при этом по усмотрению преподавателя аннулируются соответствующие предыдущие результаты. Ликвидация задолженностей проводится в период с 07.02 по 28.02 (с 07.09 по 28.09) по согласованию с деканатом.

Сведения об авторах

Геворкян Мигран Нельсонович — кандидат физико-математических наук, доцент кафедры прикладной информатики и теории вероятностей РУДН.

Королькова Анна Владиславовна — кандидат физико-математических наук, доцент-исследователь кафедры прикладной информатики и теории вероятностей РУДН.

Кулябов Дмитрий Сергеевич — кандидат физико-математических наук, доцент, доцент-исследователь кафедры прикладной информатики и теории вероятностей РУДН.

Учебное издание

Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян

Информационная безопасность компьютерных сетей

Технический редактор *Н. А. Ясько*

Издание подготовлено в авторской редакции
Компьютерная вёрстка *А. В. Королькова*, *Д. С. Кулябов*

Подписано в печать ___.___.2015 г. Формат 60×84/16. Печать офсетная.
Усл. печ. л. ___. Тираж 500 экз. Заказ № ___.

Российский университет дружбы народов
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3

Типография РУДН
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3, тел. 952-04-41