Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

В. В. Подтопельный

БЕЗОПАСНОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

Учебно-методическое пособие
по выполнению лабораторных работ по дисциплине
для студентов специальности 10.05.03
«Информационная безопасность автоматизированных систем»
специализация «Безопасность открытых информационных систем»

Калининград
Издательство ФГБОУ ВО «КГТУ»
2022

УДК 004.4 (075)

Рецензент

доцент кафедры информационной безопасности института информационных технологий ФГБОУ ВО «Калининградский государственный технический университет» А. Г. Жестовский

Подтопельный, В. В.

Безопасность операционных систем: учебно-методическое пособие по выполнению лабораторных работ по дисциплине для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» специализация «Безопасность открытых информационных систем». – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 181 с.

Учебно-методическое пособие включает в себя рассмотрение практических вопросов в области защиты информации по дисциплине «Безопасность операционных систем». В учебно-методическом пособии приведен список лабораторных работ для изучения и закрепления материала дисциплины. Представлены методические указания по изучению дисциплины. Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы дисциплины.

Учебно-методическое пособие предназначено для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» и смежных специальностей.

Табл. 6, рис. 24, список лит. – 52 наименования

Учебно-методическое пособие рассмотрено и одобрено в качестве электронного методического материала кафедрой информационной безопасности 19 мая 2022 г., протокол № 7

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в учебном процессе в качестве локального электронного методического материала методической комиссией института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 28 июня 2022 г., протокол № 4

УДК 004.4 (075)

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем», специализация: «Безопасность открытых информационных систем», изучающих дисциплину «Безопасность операционных систем».

Лабораторный практикум содержит 14 лабораторных работ.

В результате выполнения лабораторных работ студенты должны:

знать:

– принципы построения и функционирования, примеры реализаций современных операционных систем;

– функции ОС, основные концепции управления процессорами, памятью, вспомогательной памятью, устройствами;

– критерии оценки эффективности и надежности средств защиты ОС;

– принципы организации и структуру подсистем защиты ОС семейств UNIX и Windows;

уметь:

– использовать средства операционных систем для обеспечения эффективного и безопасного функционирования автоматизированных систем;

– оценивать эффективность и надежность защиты операционных систем;

– планировать политику безопасности операционных систем;

владеть:

– навыками работы с операционными системами семейств Windows и Unix, восстановления операционных систем после сбоев;

– навыками установки и настройки операционных систем семейств Windows и Unix с учетом требований по обеспечению информационной безопасности.

Программное обеспечение

1. Microsoft Desktop Education. Операционные системы: Microsoft Windows Desktop operating systems, офисные приложения: Microsoft Office, по соглашению V9002148 Open Value Subscription (срок действия: три года).

2. Программное обеспечение распространяемое по лицензии GNU General Public License (лицензия на свободное программное обеспечение, созданная в рамках проекта GNU , по которой автор передаёт программное обеспечение в общественную собственность): операционная система Linux, ПО Virtual Box.

Критерии положительной оценки изложены в таблице 1.

Таблица 1 – Шкала оценок уровня

Оценка			
Неудовлетворительный	Пороговый	Углубленный	Продвинутый
«2» (неудовлетворительно)	«3» (удовлетворительно)	«4» (хорошо)	«5» (отлично)
Работа выполнена в полном объеме. Отчет не оформлен и представлен. При защите отчетных материалов правильные ответы даны менее чем на 50 % включительно. Материал излагается непоследовательно, сбивчиво, не представляет определенной системы знаний по работе	Работа выполнена в полном объеме. Отчет оформлен и представлен. При защите отчетных материалов правильные ответы на 51–64 % вопросов. Допускаются нарушения в последовательности изложения. Демонстрируются поверхностные знания вопроса. Имеются затруднения с выводами. Допускаются нарушения норм литературной речи	Работа выполнена в полном объеме. Отчет оформлен и представлен. При защите отчетных материалов правильные ответы на 65–94 % вопросов. Ответы на поставленные вопросы излагаются систематизировано и последовательно. Материал излагается уверенно. Демонстрируется умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер	Работа выполнена в полном объеме. Отчет оформлен и представлен. При защите отчетных материалов правильные ответы даны на 95–100 % вопросов. Ответы на поставленные в билете вопросы излагаются логично, последовательно и не требуют дополнительных пояснений. Делаются обоснованные выводы. Демонстрируются глубокие знания предмета

Лабораторная работа № 1
Работа с файлами и дисками в ОС Windows

Программно-аппаратные средства: стандартные средства Microsoft Office.

Цель работы: развитие навыков работы в среде операционной системы MS-DOS

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/10/2012/2012R2

1. Теоретическое введение

Операционная система представляет собой комплекс системных и служебных программных средств. С одной стороны, она опирается на базовое программное обеспечение компьютера, входящее в его систему BIOS*;* с другой стороны, она сама является опорой для программного обеспечения более высоких уровней – прикладных и большинства служебных приложений.

Одной из первых операционных систем была MS DOS (Microsoft Disk Operating System) . Она появилась на рынке еще в 1981 г. вместе с первыми компьютерами фирмы IBM.

MS DOS состоит из следующих составных частей:

– программа начальной загрузки (Boot Record);

– базовая система ввода/вывода (BIOS), состоящая из двух частей:

– BIOS – записана на жесткий диск и содержит набор подпрограмм нижнего уровня, осуществляющих непосредственный доступ к аппаратуре;

– файл IO.SYS, содержащий подпрограммы ввода/вывода для конкретной реализации, которые используют или заменяют программы, находящиеся в BIOS. Файл IO.SYS вместе с файлом MSDOS.SYS составляют системное ядро ОС;

– файл MSDOS.SYS – часть системного ядра, отвечающая за:

– управление файлами;

– управление ресурсами сети;

– обработку ошибок;

– запуск и завершения выполнения программы;

– командный процессор – файл COMMAND.COM, который организует интерфейс (то есть взаимодействие) с пользователем путем обработки команд, которые выдают ему пользователь (на клавиатуре в виде командной строки) и прикладные программы;

– драйверы загружаются в память компьютера при загрузке операционной системы, их имена указываются в специальном файле – CONFIG.SYS. Такая схема облегчает добавление новых устройств и позволяет делать это, не затрагивая системные файлы DOS;

– файл AUTOEXEC.BAT, служащий для загрузки прикладных программ (например, Norton Commander) сразу же после загрузки ОС;

– Загрузка MS DOS происходит в несколько этапов:

– Вначале специальная процедура BIOS запускает программу начальной загрузки, хранящуюся в загрузочном секторе системного диска (диска, с которого загружается ОС).

– Эта программа в свою очередь загружает системное ядро – файлы IO.SYS и MSDOS.SYS.

– Затем загружается командный процессор – файл COMMAND.COM.

Загрузка системы производится при включении компьютера или при перезагрузке.

MS-DOS является неграфической операционной системой, использующей интерфейс командной строки (консольный интерфейс). Основным устройством управления в данном случае является клавиатура. Управляющие команды вводят в поле командной строки, где их можно и редактировать. Исполнение команды начинается после ее утверждения (нажатием клавиши ENTER).

Данные о местоположении файлов хранятся в табличной структуре, однако пользователю они представляются в виде иерархической структуры, а все необходимые преобразования берет на себя операционная система.

К функции обслуживания файловой структуры относятся следующие операции, происходящие под управлением операционной системы:

– создание файлов и присвоение им имен;

– создание каталогов (папок) и присвоение им имен;

– переименование файлов и каталогов (папок);

– копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска;

– удаление файлов и каталогов (папок);

– навигация по файловой структуре с целью доступа к заданному файлу, каталогу (папке);

– управление атрибутами файлов.

Команды MS DOS бывают двух типов:

– внутренние команды, их выполняет командный процессор COMMAND.COM (например, dir, copy);

– внешние команды – программы, поставляемые вместе с ОС в виде отдельных файлов. Они размещаются на диске и выполняют действия обслуживающего характера (например, форматирование диска, очистка экрана, проверка диска).

– В операционной системе MS-DOS для осуществления выше перечисленных операций используется достаточно большое количество команд (таблица 1). Команды состоят из имени команды и, возможно, параметров, разделенных пробелами.

Основные команды MS-DOS:

1. Создание файла с консоли copy con <имя файла>

2. Удаление файла del <имя файла>

3. Переименование файла ren <имя файла 1> <имя файла 2>
4. Редактирование файла edit <имя файла>
5. Переход на диск <имя диска>
6. Переход в каталог cd <путь>
7. Сортировка по имени файлов каталога ds
8. Сортировка по расширению файлов каталога ne
9. Создание каталога md <имя каталога>
10. Удаление каталога rd <имя каталога>
11. Очистка экрана cls
12. Вывод содержимого файла на экран type <имя файла>
13. Копирование файла copy <путь 1> <путь 2>
14. Поиск файла filefind <имя файла>
15. Работа с командной строкой prompt
16. Информация о команде <команда> /?

2.Задание к лабораторной работе

Реализовать порядок работы приведенный ниже:

Через панель Пуск меня Программы закладка Стандартные загрузить командную строку. Через командную строку зайти в каталог Temp диска С.

1. В каталоге Temp создать дерево каталогов (иерархия вложенности включает три уровня).
2. В каталоге А2 создать подкаталоги В4 и В5 и удалить каталог В2.
3. В каталоге Personal создать файл Name.txt, содержащий информацию о фамилии, имени и отчестве студента. Здесь же создать файл Date.txt, содержащий информацию о дате рождения студента. В этом же каталоге создать файл School.txt, содержащий информацию о школе, которую закончил студент.
4. В каталоге University создать файл Name.txt, содержащий информацию о названии Вуза и специальность, на которой студент обучается. Здесь же создать файл Mark.txt с оценками на вступительных экзаменах и общей суммой баллов.
5. В каталоге Hobby создать файл hobby.txt с информацией об увлечениях студента.
6. Скопировать файл hobby.txt в каталог А2 и переименовать его в файл Lab_№варианта.txt.
7. Сделать копию файла Lab_№варианта.txt (например, copy_Lab_№варианта.txt) в этом же каталоге и удалить его.
8. Очистить экран от служебных записей.
9. Вывести на экран поочередно информацию, хранящуюся во всех файлах каталога Personal.
10. Отсортировать все файлы, хранящиеся в каталоге Personal, по имени.
11. Объединить все файлы, хранящиеся в каталоге Personal, в файл all.txt и вывести его содержимое на экран.
12. Отредактировать файл all.txt, добавив в него год вашего рождения, и вывести его содержимое на экран.

13. Скопировать файл all.txt в директорию A1.

14. Удалить все директории, в названии которых есть буква А или цифра 2.

15. Изменить строку приглашения MS-DOS в соответствии с номером варианта.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы:

1. Назовите основные функции операционной системы?
2. Какой интерфейс имеет операционная система MS-DOS?
3. Какие операции можно выполнять с помощью команды COPY?
4. Как осуществляется смена дисков в ОС MS-DOS?
5. В чем особенность удаления каталога в ОС MS-DOS?
6. Каким образом происходит загрузка ОС MS-DOS?
7. Какие способы создания файла существуют в ОС MS-DOS?

Лабораторная работа № 2
Командные файлы Windows

Цель работы: научиться решать типовые задачи администрирования операционной системы Windows с использованием командных файлов.

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/
2012R2

1.Теоретическое введение

Командный файл представляет собой обычный текстовый файл с расширением bat (cmd), в котором записаны допустимые команды операционной системы, а также некоторые дополнительные инструкции и ключевые слова, придающие командным файлам некоторое сходство с алгоритмическими языками программирования. Например, если записать в файл deltmp.bat следующие команды:

```
C:\
CD %TEMP%
ATTRIB -R *.tmp
DEL *.tmp
```

и запустить его на выполнение, то будут удалены все файлы во временном каталоге Windows. Таким образом, исполнение командного файла приводит к тому же результату, что и последовательный ввод записанных в нем команд. Командные файлы широко используются при выполнении многих задач, связанных с администрированием системы.

В командных файлах можно использовать комментарии, т. е. строки, которые никак не влияют на выполнение этого файла. Комментарии вносятся с помощью ключевого слова rem, например

```
REM Снимем атрибут «Только чтение» ATTRIB -R *.tmp
REM Удалим файлы с расширением tmp DEL *.tmp
```

Условное выполнение команд

В командной строке Windows можно использовать специальные символы, которые позволяют вводить несколько команд одновременно и управлять работой команд в зависимости от результатов их выполнения. С помощью таких символов условной обработки можно содержание небольшого пакетного файла записать в одной строке и выполнить полученную составную команду.

Используя символ амперсанда &, можно разделить несколько утилит в одной командной строке, при этом они будут выполняться друг за другом. Например, если набрать командную строку

```
DIR & PAUSE &  COPY /?
```

и нажать клавишу <Enter>, то вначале на экран будет выведено содержимое текущего каталога, а после нажатия любой клавиши – встроенная справка команды copy.

Символ ^ позволяет использовать командные символы как текст, т. е. при этом происходит игнорирование значения специальных символов. Например, если ввести в командной строке

ECHO Абв & COPY /?

и нажать клавишу <Enter>, то произойдет выполнение подряд двух команд: echo Абв и copy /?. Если же выполнить команду

ECHO Абв ^& COPY /?

то на экран будет выведено

Абв & COPY /?

В этом случае просто выполняется одна команда echo с соответствующими параметрами.

Условная обработка командосуществляется с помощью символов && и | | следующим образом. Двойной амперсанд && запускает команду, стоящую за ним в командной строке, только в том случае, если команда, стоящая перед амперсандами была выполнена успешно. Например, если в корневом каталоге диска c: есть файл plan.txt, то выполнение строки

TYPE C:\plan.txt && DIR

приведет к выводу на экран этого файла и содержимого текущего каталога. Если же файл c:\pian.txt не существует, то команда dir выполняться не будет.

Два символа || осуществляют в командной строке обратное действие, т. е. запускают команду, стоящую за этими символами, только в том случае, если команда, идущая перед ними, не была успешно выполнена. Таким образом, если в предыдущем примере файл c:\plan.txt будет отсутствовать, то в результате выполнения строки

TYPE C:\plan.txt || DIR

на экран выведется содержимое текущего каталога.

Отметим, что условная обработка действует только на ближайшую команду, т. е. в строке

TYPE C:\plan.txt && DIR & COPY /?

команда copy /? запустится в любом случае, независимо от результата выполнения команды TYPE C:\plan.txt.

Несколько утилит можно сгруппировать в командной строке с помощью скобок.

Рассмотрим, например, две строки:

TYPE C:\plan.txt && DIR & COPY /?

TYPE C:\plan.txt && (DIR & COPY /?)

В первой из них символ условной обработки && действует только на команду dir, во второй — одновременно на две команды: dir и copy.

Запуск программ в определенное время

С помощью утилиты AT можно запускать команды и программы в заданное время. Для работы этой команды должен быть запущен сервис расписаний, а пользователь должен являться членом локальной группы администраторов.

Для того чтобы внести новое задание в расписание, используется следующий синтаксис команды AT:

AT [\\имя_компьютера] время [/INTERACTIVE] [/EVERY:дата[,..] [/NEXT:дата[,...]] «команда»

Если запустить AT без параметров, то на экран будет выведен список всех команд и программ, которые будут запущены с ее помощью.

Параметр *\\имя_компьютера* задает удаленный компьютер, на котором могут быть запущены планируемые команды и программы. Если этот параметр не задан, то предполагается, что запуск программ будет произведен на том же компьютере, где запущена команда at.

Параметр время задает время, когда планируемая команда должна быть запущена. Время задается в 24-часовом формате часы:минуты (от 00:00 до 23:59).

Ключ /INTERACTIVE позволяет команде AT обмениваться данными с теми пользователями, которые будут подключены к системе в момент выполнения запланированной команды (в случае выполнения утилиты командной строки или пакетного файла будет создано новое командное окно).

Ключи /EVERY: дата [,...] или /NEXT: *дата* [,...] задают дату, когда должна быть запущена запланированная команда. Если указан ключ /EVERY: дата, то команда будет запускаться в заданные дни в течение недели или месяца (например, каждый четверг или каждый третий день месяца). Дни недели задаются буквами (Пн, Вт, Ср, Чт, Пт, Сб, Вс или M, T, W, Th, F, S,. Su), в зависимости от того, какая версия Windows (русифицированная или английская) установлена на компьютере), а дни месяца – цифрами (от 1 до 31) Несколько дат разделяются запятыми. Если параметр *дата* не задан, то подразумевается текущий день месяца.

Ключ /NEXT: дата[,...] позволяет запустить команду при наступлении следующей заданной даты (например, в следующий четверг). Параметр *дата* имеет то же значение, что и в ключе /every.

При помощи параметра *«команда»* задаются утилита, программа (файлы с расширением exe или com) или пакетный файл (файлы с расширением bat или cmd), которые должны быть запущены. Если для задания команды необходимо указывать ее местоположение, имя файла задается полностью, с указанием пути и диска. Если команда запускается на удаленной машине, то необходимо указать имя этой машины и имя разделяемого ресурса, а не имя сетевого диска. Параметр *«команда»* должен быть заключен в кавычки.

Команда AT не вызывает автоматически командный интерпретатор cmd.exe перед запуском запланированной команды. Поэтому если запускается внутренняя команда (например, copy или dir), а не исполняемый файл, то вначале должен быть запущен командный интерпретатор с ключом /C, например,

AT 23:00 "CMD /C DIR > C:\test.out".

После того как задание запланировано на выполнение, ему присваивается идентификационный номер. Отменить выполнение уже запланированного задания можно с помощью следующего варианта команды AT:

AT *[\\имя_компьютвра] [[код]* [/DELETE[/YES]]

Здесь параметр *код* определяет идентификационный номер, присваиваемый команде или программе, которая будет запущена. Если *код* не задан, отменены будут все запланированные на компьютере команды.

Ключ /yes задает утвердительный ответ на все запросы об отмене запланированных для запуска команд.

Замечание: Команды, запуск которых задан с помощью AT, выполняются как фоновые (если только не указан ключ /interactive), поэтому результаты их работы не выводятся на экран. Для направления вывода результатов в файл используются, как обычно, символы перенаправления > и >>. В этом случае сама команда должна быть заключена в кавычки.

Текущим каталогом для выполнения запланированных команд по умолчанию является каталог %systemRoot%. Все запланированные с помощью AT команды сохраняются в системном реестре, и, следовательно, не будут потеряны при перезапуске службы расписаний или всего компьютера.

Запланированные задания, использующие сетевые диски, не должны использовать переназначения дисков, заданные пользователем, так как служба расписаний может не получить доступа к таким дискам или диски могут оказаться не подключенными, если другой пользователь войдет в систему в момент выполнения запланированной команды. Вместо этого, запланированные задания должны использовать полный сетевой путь (*\\имя_компьютера\имя_ресурса)*.

Рассмотрим пример. Пусть у пользователя имеется сетевой диск X:, подключенный к сетевому ресурсу \\Server1\ForBackup, на который необходимо производить резервное копирование. Командный файл mybackup.bat выполняет копирование по указанному в качестве параметра командной строки пути. Чтобы запланировать запуск этого командного файла в 1 час ночи, можно использовать следующую команду:

AT 1:00 mybackup \\Server1\ForBackup,

тогда как недопустимой является следующая форма:

AT 1:00 mybackup X:

Если с помощью AT запланирована команда, использующая буквенное обозначение диска для подключения к разделяемому каталогу, то после ее выполнения должна быть запланирована команда, отключающая данный каталог от диска. В противном случае, буква, использованная для обозначения подключаемого диска, будет недоступна или не будет выводиться в командной строке.

Работа с переменными среды

В Windows различаются два вида переменных среды: переменные *среды операционной системы* (системные переменные) и переменные *среды текущего пользователя.*

Системные переменные среды определяются Windows и имеют одни и те же значения, не зависящие от того, какой пользователь вошел на компьютер. Например, переменные comspec (полный путь к командному интерпретатору, comspec=C: \WINNT\system32\cmd.exe), os (название операционной системы, os=windows_NT), windir (каталог Windows NT, windir=c:\winnt). Добавлять новые системные переменные или изменять значения существующих могут члены группы администраторов.

Переменные среды текущего пользователя могут иметь разные значения для каждого пользователя на конкретном компьютере. В число таких переменных входят переменные, определяемые в приложениях (например, путь к каталогу, в котором сохраняются файлы приложений).

По умолчанию при загрузке Windows находит файл c:\autoexec.bat, если он есть, и берет оттуда все установки для переменных среды. Скажем, если в autoexec.bat определена переменная path, то путь, задаваемый этой переменной, будет добавляться к системному пути по умолчанию каждый раз, когда какой-либо пользователь регистрируется в системе.

Когда открывается новое окно командного интерпретатора, переменные среды обоих типов копируются в переменные среды этого командного окна. При этом копирование происходит в следующем порядке:

1. Переменные из файла autoexec.bat.
2. Переменные среды операционной системы.
3. Переменные среды пользователя.
4. Переменные из файла %systemRoot%\SYSTEM32\autoexec.nt.

Например, если в файле autoexec.bat имеется строка «*SET TMP=C:\»* и, кроме этого, задана переменная среды пользователя TMP со значением D:\tempdir, то значением переменной TMP в командном окне будет d:\tempdir (конечно, если только значение переменной TMP не переопределяется еще раз в файле autoexec.nt).

Команда *SET*

Работа с переменными среды текущего командного окна осуществляется с помощью команды set. Естественно, изменения, которые вносятся в переменные среды этой команды, актуальны только в текущем командном окне.

Если режим расширенной обработки команд не включен, то синтаксис команды set следующий:

SET [переменная=[строка]]

В частности, команда SET, запущенная без параметров, выводит значения всех переменных среды текущего командного окна.

При включении расширенной обработки команд появляются новые возможности у команды SET. Рассмотрим их подробнее.

Если при вызове команды SET указать только имя переменной без знака равенства и значения, то команда выведет значения всех переменных, имя которых начинается с указанной строки. Таким образом, команда

SET P

отобразит значения всех переменных, имена которых начинаются с P (path, например).

Если имя переменной не найдено в текущей среде, то при возврате команда SET установит значение errorlevel равным 1. Это свойство можно использовать в командных файлах для определения наличия определенной переменной.

Команда SET допускает использование знака равенства (=) в любой позиции значения переменной среды, кроме первого символа.

Переменные могут рассматриваться как числа и с ними можно производить арифметические вычисления. Для этой цели в команде set имеется дополнительный ключ /A

SET /A переменная=выражение

Использование ключа /A указывает, что стоящая справа от знака равенства строка является числовым выражением, значение которого вычисляется.

Например, если задать команду

SET /A M=1+2

то значение переменной M будет равно трем (M=3). Обработчик выражений, входящих в команду SET, очень прост и поддерживает следующие операции, перечисленные в порядке убывания приоритета:

– группировка с помощью круглых скобок ();

– арифметические операторы умножения (*), целочисленного деления (/), остатка от деления (%);

– арифметические операторы сложения (+) и вычитания (-);

– двоичный сдвиг влево (<<) и вправо (>>);

– двоичное И (&);

– двоичное исключающее ИЛИ (^);

– двоичное ИЛИ (|);

– операторы присваивания =*, =/, =%, =+, =-, =, &=, ^=, |=, <<= и >>=;

– разделение операторов с помощью запятой (,).

При использовании любых логических или двоичных операторов необходимо заключить строку выражения в кавычки. Любые нечисловые строки в выражении рассматриваются как имена переменных среды, значения которых преобразуются в числовой вид перед использованием. Если переменная с указанным именем не определена в системе, вместо нее подставляется нулевое значение. Например, если переменная среды x не была предварительно задана, то в результате выполнения команды

SET /A N=X+5

значение переменной N будет равно пяти (n=5).

Таким образом, применение ключа /A позволяет выполнять арифметические операции со значениями переменных среды, причем не нужно вводить знаки % для получения их значений. Например:

SET /A M=1 SET /A N=M+1

В Windows также немного усовершенствована работа с переменными среды как со строками. Например, следующая команда:

SET *переменная!=%переменная2:строка 1=строка2%*

раскроет значение второй указанной переменной среды *(переменная2),* заменит там все вхождения *строки1* на *строку2* и запишет результат в первую переменную *(переменная1).* Скажем, если значением переменной s была строка D:\Programs\Aditor, то в результате выполнения команды

SET S=%S:Programs=IIporраммы%

переменная s приобретет значение D:\Прогрaммы\Aditor. Подставляемая строка *(строка2)* может быть пустой, что приведет к удалению всех вхождений *строки1* из раскрытого значения переменной. Если *строка1* начинается со знака звездочки, то будет заменен весь текст с начала раскрытого значения до первого вхождения оставшейся части *строки1.* Например, если выполнить следующие команды:

SET S=Раз Два Три Раз Два Три

SET M=%S:* Три=Четыре%

то значением переменной M будет строка четыре Раз Два три. Если задать команду вида

SET *переменна я 1=%переменная2:*~m,n%

то она раскроет значение второй указанной переменной *(переменнаяг),* использует из него только n символов, начиная с (m+i)-го (m определяет количество символов, на которое происходит сдвиг) и запишет результат в первую переменную *(переменная1).* Например, в результате выполнения следующих команд:

SET S=Pas Два Три Раз Два Три SET M=%S:~4,3%

значением переменной M будет слово Два.

Запуск программ и документов

Синтаксис команды start имеет следующий вид:

START [«заголовок»] *[/Dпуть]* [/I] [/MIN] [/MAX] [/SEPARATE|/SHARED] [/LOW| /NORMAL/HIGH|/REALTIME] [/WAIT] [/B] *[команда/программа] [параметры]*

Если параметр *команда/программа* определяет внутреннюю команду интерпретатора cmd.exe или пакетный файл, то для их выполнения в новом (если не указан ключ /B) окне автоматически запускается интерпретатор команд cmd.exe с ключом /K. Таким образом, в этом случае новое окно не будет закрыто после завершения команды. Если же запускается не внутренняя команда cmd.exe и не пакетный файл, то эта программа запускается в графическом или текстовом окне.

В случае, когда первым элементом командной строки является слово cmd без расширения и пути к файлу, обработчик команд перед выполнением строки заменяет слово cmd на значение переменной comspec (полный путь к командному интерпретатору), что позволяет избежать неожиданного запуска случайных версий файла CMD.exe.

Если имя запускаемой программы задано без расширения, то командный интерпретатор использует значение переменной среды PATHTEXT (в Windows 9*x* такой переменной нет), чтобы определить расширения имен исполняемых файлов и порядок поиска нужного файла. По умолчанию для переменной PATHTEXT задается следующее значение:

PATHEXT=.COM;.EXE;.BAT;.CMD

(cmd — это расширение для командных файлов в Windows NT и выше). Здесь синтаксис подобен синтаксису для переменной path, т. е. отдельные элементы разделяются точкой с запятой. Если ни одного файла с заданными по умолчанию расширениями не найдено, интерпретатор команд проверяет, задает ли указанное имя существующий каталог. Если это так, то команда START запускает Проводник Windows и открывает в нем указанный каталог.

Параметр *заголовок* в команде start определяет заголовок создаваемого окна. Например:

START «Копирование данных» copier.bat

Если команда start открывает новое командное окно, то в нем можно сразу указать рабочий каталог. Это делается с помощью параметра *путь*.

Применение ключа /I означает, что новой операционной средой станет исходная среда, переданная командным интерпретатором cmd.exe, а не текущая среда командного окна.

Если указан ключ /MIN, то запуск команды/программы происходит в свернутом окне, если /MAX – то в развернутом (максимизированном) окне.

Ключи /SEPARTR и /SHARED используются для указания режима запуска 16-разрядных приложений Windows. Если указан ключ /SEPARATE, то запуск такой программы происходит в отдельной области памяти, если /SHARED – то в общей области памяти.

Следующие четыре ключа отвечают за приоритет запускаемой задачи. Применение ключа /LOW означает, что приложение запускается с приоритетом IDLE, ключа /NORMAL – с приоритетом NORMAL, ключа /HIGH – с приоритетом HIGH, ключа /REALTIME – с приоритетом REALTIME.

Ключ /WAIT используется для запуска приложения с ожиданием его завершения.

Если указан ключ /B, то запуск приложения происходит без создания *нового* окна (конечно, если это возможно). Таким образом, если с этим ключом запускается внутренняя команда cmd.exe или пакетный файл, то новая копия командного интерпретатора будет запущена в текущем командном окне.

Команды *ASSOC* и *FTYPE*

Рассмотрим теперь, каким образом можно из командной строки сопоставить файлам с определенным расширением программу, которая будет их открывать. Для этого необходимо использовать две команды: ASSOC и FTYPE.

С помощью assoc можно устанавливать или изменять связи между расширениями и типами файлов. Синтаксис этой команды похож на синтаксис команды set:

ASSOC [.рсш[=[тип_файла]]]

Параметр .рсш здесь задает расширение для связи с типом файлов, а *тип_файла* указывает тип файла для связи с данным расширением. Например:

ASSOC .txt=txtfile

Команда assoc, запущенная без параметров, выведет информацию обо всех хекуших связях между расширениями и типами файлов. Для того чтобы вывести информацию только о типе файлов с одним заданным расширением, нужно использовать команду assoc *.рсш,* например

ASSOC .txt

Команда assoc, заданная без параметра *тип_файла,* удалит связь между данным расширением и типом файлов, например

ASSOC . txt=

С помощью команды ftype можно сопоставить определенному типу файлов программу, которая будет их открывать. Синтаксис этой команды имеет вид:

FTYPE [тип_файлов [= [командная_строка_открытия]]]

Здесь параметр *командная_строка_открытия* задает команду открытия, использующуюся при запуске файлов указанного типа.

Команда FTYPE без параметров выводит список всех типов файлов, для которых определены командные строки открытия. Если указан только тип файла, FTYPE выводит командную строку открытия для этого типа файлов. Например, если задать команду

FTYPE txtfile

то на экран выведется строка следующего вида:

txtfile=%SystemRoot%\system32\NOTEPAD.EXE %1

Если после знака равенства не указана строка открытия, FTYPE удалит текущее сопоставление для указанного типа файлов.

В параметре *командная_строка_открытия* можно использовать замещаемые параметры командной строки %0 — %9. Когда будет запускаться программа, сопоставленная заданному типу файлов, переменные %0 и %1 заменяются на имя файла, запускаемого с помощью сопоставления. Вместо переменной %* подставляются все оставшиеся параметры командной строки, а переменные %2, %3 и т. д. заменяются, соответственно, на первый, второй и другие параметры. Вместо переменной *%~n* подставляются все оставшиеся параметры, начиная с n, где n является числом от 2 до 9. Например:

.pl=PerlScript PerlScript=perl.exe %1 %*

Эти команды позволят вызывать из командной строки интерпретатор сценариев Perl следующим образом:

script.pl 1 2 3

Более того, если записать расширение pl в переменную среды PATHTEXT, то можно не вводить расширение p1 в именах файлов. Для этого команда SET используется следующим образом:

SET PATHEXT=.p1;%PATHEXT%

Теперь обработчик сценариев Perl вызывается еще проще:

script 1 2 3

Вывод сообщений и дублирование команд

По умолчанию команды пакетного файла перед исполнением выводятся на экран с помощью команды echo off можно отключить дублирование команд, идущих после нее (сама команда echo off при этом все же дублируется). Например,

```
REM Следующие две команды будут дублироваться на экране   ...
DIR C:\
ECHO OFF
REM A остальные уже не будут
DIR D:\
```

Для восстановления режима дублирования используется команда echo on. Кроме этого, можно отключить дублирование любой отдельной строки в командном файле, написав в начале этой строки символ @, например:

```
ECHO  ON
REM Команда DIR C:\  дублируется на экране
DIR C:\
REM А команда  DIR D:\ — нет
@DIR D:\
```

Таким образом, если поставить в самое начало командного файла команду

```
@ECHO OFF
```

то это решит все проблемы с дублированием команд.

Вывести строку сообщения на экран можно с помощью команды

```
ECHO сообщение
```

Например,

```
@ECHO OFF
ECHO Привет!
```

Команда ECHO, (точка должна следовать непосредственно за словом «echo») выводит на экран пустую строку. Например,

```
@ECHO OFF
ECHO Привет!
ECHO.
ECHO Пока!
```

Часто бывает удобно для просмотра сообщений, выводимых из пакетного файла, предварительно полностью очистить экран командой cls.

Используя описанный в первой главе механизм перенаправления ввода/вывода (символы > и >>), можно направить сообщения, выводимые командой echo, в заданный текстовый файл. Например,

```
@ECHO OFF
ECHO Привет! > hi.txt
ECHO Пока! >> hi.txt
```

Использование параметров командной строки и переменных среды

При запуске пакетных файлов в командной строке можно указывать произвольное число параметров, значения которых допускается использовать внутри файла Это позволяет, например, применять один и тот же командный файл для выполнения команд с различными параметрами

Для доступа к параметрам командной строки применяются символы %0, %1,…,%9. Вместо %0 подставляется имя выполняемого пакетного файла, а вместо %1, %2,…, %9 – значения первых девяти параметров командной строки соответственно Если в командной строке при вызове пакетного файла задано меньше девяти параметров, то «лишние» переменные из %1 – %9 являются пустыми строками. Рассмотрим следующий пример. Пусть имеется командный файл copier.bat такого содержания

```
@ECHO OFF
CLS
ECHO Файл %0 копирует каталог %1 в %2
XCOPY %1 %2 /S
```

Если запустить его из командной строки с двумя параметрами, например

```
copier.bat C:\Programs D:\Backup
```

то на экран выведется сообщение

```
Файл copier.bat копирует каталог C:\Programs в D:\Backup
```

и произойдет копирование каталога c:\Programs со всеми его подкат алогами в D.\Backup.

При необходимости можно использовать более девяти параметров командной строки Это достигается с помощью команды shift, которая изменяет значения замещаемых параметров с %0 по %9, копируя каждый параметр в предыдущий, т е значение %1 копируется в %0, значение %2 – в %1 и т д Замещаемому параметру %9 присваивается значение параметра, следующего в командной строке за старым значением *%9* Если же такой параметр не задан, то новое значение %9 – пустая строка.

Например, пусть командный файл my.bat вызван из командной строки следующим образом

```
my.bat p1 p2  p3
```

Тогда %0=my bat, %1=p1, °%2=p2, %3=p3, параметры %4 – %9 являются пустыми строками После выполнения команды shift значения замещаемых параметров изменятся следующим образом %0=p1, %1=p2, %2=p3, параметры %3 – %9 – пустые строки.

Обратный shift (обратный сдвиг) отсутствует, т.е. после выполнения shift уже нельзя восстановить параметр (%0), который был первым перед сдвигом Если в командной строке задано больше десяти параметров, то команду shift можно использовать несколько раз.

С помощью команды set внутри командных файлов можно работать с переменными среды, в том числе объявлять собственные переменные. В Windows *9x* все переменные среды рассматриваются как строки и для получения их значений нужно имя соответствующей переменной заключить в символы *%*. Например,

```
@ECHO OFF
CLS
REM Создание  переменной MyVar
SET MyVar=Привет
REM Изменение переменной
SET MyVar=%MyVar%
ECHO Значение переменной MyVar: %MyVar%
REM Удаление переменной MyVar
SET MyVar=
```

При запуске такого командного файла на экран выведется строка
Значение переменной MyVar: Привет!

Внутри командного файла можно использовать и переменные, которые система устанавливает автоматически. Например

```
@ECHO OFF
CLS
ECHO Каталог Windows: %WinDir%
ECHO Каталог для временных файлов: %TEMP%
```

При включенной расширенной обработке команд имеется возможность рассматривать переменные среды как числа и производить с ними арифметические вычисления. Для этого используется команда set с ключом /a. Приведем пример пакетного файла add.bat, складывающего два числа, заданных в качестве параметров командной строки, и выводящего полученную сумму на экран:

```
@ECHO OFF
REM В переменной M будет храниться сумма
SET /A M=%1+%2
ECHO Сумма  %1  и  %2 равна  %M%
```

```
REM Удалим переменную M
SET M=
```

Напомним, что все изменения, производимые с помощью команды set над переменными среды в командном файле, сохраняются и после завершения работы этого файла, но действуют только внутри текущего командного окна. В Windows имеется возможность локализовать изменения переменных среды внутри пакетного файла, т. е. автоматически восстанавливать значения всех переменных в том виде, в каком они были до начала запуска данного файла. Для этого используются две команды: setlocal и endlocal. Команда setlocal определяет начало области локальных установок переменных среды. Другими словами, изменения среды, внесенные после выполнения setlocal, будут являться локальными относительно текущего пакетного файла. Каждая команда setlocal должна иметь соответствующую команду endlocal для восстановления прежних значений переменных среды. Изменения среды, внесенные после выполнения команды endlocal, уже не являются локальными относительно текущего пакетного файла; их прежние значения не будут восстановлены по завершении выполнения этого файла.

Рассмотрим пример командного файла super.bat, который запускает приложение superApp, записывает вывод в файл c:\superapp.out и загружает этот файл в программу Notepad:

```
@ECHO OFF
REM Запоминаем значения переменных среды
SETLOCAL
REM Добавляем к переменной PATH путь
REM к каталогу G:\programs\superapp
PATH=G:\programs\superapp,%PATH%
REM Запускаем приложение
CALL superapp>c:\superapp.out
REM Восстанавливаем значения переменных среды
ENDLOCAL
REM Загружаем полученный файл в Блокнот
START notepad C:\superapp.out
```

При включении расширенной обработки команд у SETLOCAL можно указывать необязательный аргумент, который может иметь значения ENABLEEXTENSIONS или DISABLEEXTENSIONS. Это позволяет временно включить или отключить расширенную обработку команд до выполнения команды ENDLOCAL, независимо от исходного состояния режима обработки команд до вызова команды.

```
SETLOCAL
```

Если команда setlocal вызывается с аргументом, то она устанавливает код ошибки ERRORLEVEL. Если указан один из двух допустимых аргументов, то код ошибки будет равен нулю, иначе возвращается значение 1. Это свойство

можно использовать в пакетных файлах, чтобы определить доступность расширенной обработки команд, например

```
XCOPY Б: > NUL
SETLOCAL ENABLEEXTENSIONS
IF ERRORLEVEL 1 ECHO Не удается включить расширенную обработку
```

В данном примере команда XCOPY с недопустимым аргументом Б: необходима для установки ненулевого значения errorlevel, так как в прежних версиях интерпретатора cmd.exe команда setlocal не устанавливает код ошибки.

Перейдем теперь к рассмотрению заменяемых параметров командной строки (%0, %1 и т.д.) В пакетных файлах Windows работа с этими параметрами становится более удобной.

Во-первых, с помощью символов %* в пакетном файле можно обозначить все аргументы (%1 %2 %3 %4 %5) Таким образом, запустив следующий пакетный файл

```
@ECHO OFF
ECHO Файл запущен с параметрами: %*
```

с параметрами a в c, мы на экране получим следующее сообщение-файл запущен с параметрами: ABC

Во-вторых, появляются некоторые возможности синтаксического анализа заменяемых параметров для параметра с номером n (%n). При этом допустимы различные синтаксические конструкции (включение дополнительных операторов).

Операторы для заменяемых параметров:

%~Fn Переменная %n расширяется до полного имени файла
%~Dn Из переменной %n выделяется только имя диска
%~Pn Из переменной %n выделяется только путь к файлу
%~Nn Из переменной %n выделяется только имя файла
%~Xп Из переменной *%n* выделяется расширение имени файла
%~Sn Значение операторов N и x для переменной °п изменяется так, что они работают с кратким именем файла

%~$PATH:n Проводится поиск по каталогам, заданным в переменной среды PATH, и переменная %n заменяется на полное имя первого найденного файла Если переменная path не определена или в результате поиска не найден ни один файл, эта конструкция заменяется на пустую строку Естественно, здесь переменную PATH можно заменить на любое другое допустимое значение

Данные синтаксические конструкции можно объединять друг с другом, например

%~DPn – из переменной %n выделяется имя диска и путь,
%~NXn – из переменной %n выделяется имя файла и расширение

Рассмотрим следующий пример. Пусть мы находимся в каталоге c:\text и запускаем пакетный файл с параметром Рассказ.doc (0%1=Рассказ.doc). Тогда применение операторов (привидениы ниже) к параметру %1 даст следующие результаты

```
%~Fl=C:\TEXT\Рассказ.doc
%~D1=C:
%~P1=\TEXT\
%~Xl=.doc
%DPl=C.\TEXT\
%NX1=Рассказ.doc
```

Небольшое изменение произошло также в команде shift, которая производит сдвиг параметров. При включении расширенной обработки команд shift поддерживает ключ */n,* задающий начало сдвига параметров с номера n, где n может быть числом от 0 до 9.

Например, в следующей команде:

```
SHIFT   /2
```

параметр %2 заменяется на %3, %3 на %4 и т. д., а параметры %0 и %1 остаются без изменений.

Приостановка выполнения командных файлов

Для того чтобы вручную прервать выполнение запущенного bat-файла, нужно нажать клавиши <Ctrl>+<C> или <Ctrl>+<Break>. Однако часто бывает необходимо программно приостановить выполнение командного файла в определенной строке с выдачей запроса на нажатие любой клавиши. Это делается с помощью команды pause. Перед запуском этой команды полезно с помощью команды echo информировать пользователя о действиях, которые он должен произвести. Например:

```
ECHO Вставьте дискету в дисковод A:  и нажмите любую клавишу
PAUSE
```

Команду pause обязательно нужно использовать при выполнении потенциально опасных действий (удаление файлов, форматирование дисков и т. п.). Например:

```
ECHO Сейчас будут удалены все файлы в C:\TEXT
ECHO Для отмены нажмите Ctrl-C
PAUSE
DEL C:\text\*.*
```

Вызов внешних командных файлов

Из одного командного файла можно вызвать другой, просто указав его имя. Например:

```
@ECHO OFF
```

```
CLS
REM Вывод списка  log-файлов
DIR C:\*.log
REM Передача выполнения файлу f.bat
f .bat
COPY A:\*.*  C:\
PAUSE
```

Однако в этом случае после выполнения вызванного файла управление в вызывающий файл не передается, т. е. в приведенном примере команда

```
COPY A:\*.*  C:\
```

(и все следующие за ней команды) никогда не будет выполнена.

Для того чтобы вызвать внешний командный файл с последующим возвратом в первоначальный файл, нужно использовать специальную команду

CALL *файл*

Например:

@ECHO OFF Например:

```
CLS
REM Вывод списка log-файлов
DIR C:\*.log
REM Передача выполнения файлу f.bat
CALL  f.bat
COPY A:\*.*  C:\
PAUSE
```

В этом случае после завершения работы файла f.bat управление вернется в первоначальный файл на строку, следующую за командой call (в нашем примере это команда COPY A:*.* C:\).

Внутри пакетного файла нельзя явным образом использовать подпрограммы, однако можно создать несколько пакетных файлов: один основной и один или несколько вспомогательных, каждый из которых может обрабатывать параметры командной строки. Так как после вызова с помощью команды call вспомогательного файла из основного управление возвращается на следующую инструкцию основного файла, то можно считать, что таким образом была вызвана подпрограмма.

Этим приемом можно пользоваться, скажем, при работе с оператором for ... in ... do, который, задает цикл только для одной команды. Рассмотрим пример.

Пусть в командном файле proc.bat записаны следующие строки:

```
@ECHO OFF
ECHO Записываем файл %1.txt
```

ECHO Параметр вызова: %1 > %1.txt

В другом пакетном файле main.bat введем:
@ECHO OFF
FOR %%i IN (Раз,Два,Три) DO CALL proc.bat %%i

Если запустить файл main.bat на исполнение, то это приведет к троекратному выполнению файла proc.bat. В результате парной работы этих двух командных файлов создадутся три текстовых файла: Раз.txt, Два.txt и Три.txt. При этом в Раз.txt будет записана строка Параметр вызова: Раз, В Два.txt – Параметр вызова: Два, В Три.txt – Параметр вызова: Три.

Команды перехода

В расширенном режиме командного интерпретатора cmd.exe в команде перехода внутри файла goto можно задавать в качестве метки перехода строку :eof, которая передает управление в конец текущего пакетного файла. Это позволяет легко выйти из пакетного файла без определения каких-либо меток в самом его конце, например:

@ECHO OFF
REM Если файл был запущен без параметров, выходим из него,
REM иначе печатаем первый параметр
IF -%1==- GOTO :EOF
ECHO %1

При включении расширенной обработки команд произошли изменения в команде вызова внешнего пакетного или исполняемого файла call – теперь в качестве адресата вызова в этой команде можно использовать метки внутри текущего командного файла, т. е. применяется следующий синтаксис:

CALL :метка аргументы

При вызове такой команды создается новый контекст текущего пакетного файла с заданными аргументами и управление передается на инструкцию. расположенную сразу после метки. Для выхода из такого пакетного файла необходимо два раза достичь его конца. Первый выход возвращает управление на инструкцию, расположенную сразу после строки call, а второй выход завершает выполнение пакетного файла. Например, если запустить с параметром Копия-1 командный файл следующего содержания:

@ECHO OFF
ECHO %1
CALL :2 Копия-2
:2
ECHO %1

то на экран выведутся три строки:
Копия-1 Копия-2 Копия-1

Таким образом, подобное использование команды call очень похоже на обычный вызов подпрограмм (процедур) в алгоритмических языках программирования.

Операторы сравнения

IF [NOT] *строка1==строка2 команда*

IF [NOT] EXIST *файл команда*

IF [NOT] ERRORLEVEL *число команда.*

При включении расширенной обработки команд можно дополнительно применять еще три варианта этой команды:

IF [/I] *строка1 оператор__сравнения строка2 команда*

IF CMDEXTVERSION *число команда*

IF DEFINED *переменная команда*

Рассмотрим сначала оператор if в следующем виде:

IF [/I] *строка1 оператор_сравнения строка2 команда*

Операторы сравнения в IF

Приведем пример использования операторов сравнения:

```
@ECHO OFF
CLS
IF -%1 == -Вася ECHO Привет, Вася!
IF -%1 NEQ -Вася ECHO Привет, но Вы не Вася!
```

Ключ /I, если он указан, задает сравнение текстовых строк без учета регистра. Ключ /I можно также использовать и в форме *строка 1== строка2* команды if. Например, условие

IF /I DOS==dos . . .

будет истинным.

Отметим также, что сравнение проводится по общему типу данных, так что если обе сравниваемые строки содержат только цифры, то обе строки преобразуются в числа, после чего выполняется сравнение этих чисел. Например, условие

IF 002 LEQ 5 ...

будет истинным.

Операторы сравнения чисел можно применять и в операторе

IF ERRORLEVEL ...

Например:

IF ERRORLEVEL LEQ 1 GOTO Case1

При включенной расширенной обработке команд для более удобной работы с кодами завершения программ можно использовать выражение %errorlevel%. Строка %errorlevel% будет развернута в строковое представление текущего значения кода ошибки errorlevel, за исключением ситуации, когда уже

имеется переменная среды с именем errorlevel; в этом случае, естественно, подставляется значение этой переменной.

Для определения внутреннего номера версии текущей реализации расширенной обработки команд применяется оператор if в следующем виде:

IF CMDEXTVERSION *число команда*

Здесь условие cmdextversion применяется подобно условию errorlevel, но число сравнивается с вышеупомянутым внутренним номером версии. Первая версия имеет номер 1. Номер версии будет увеличиваться на единицу при каждом добавлении существенных возможностей расширенной обработки команд. Если расширенная обработка команд отключена, условие cmdextversion никогда не бывает истинно.

Для определения наличия заданной переменной среды предназначен следующий вариант команды IF:

IF DEFINED *переменная команда*

Здесь условие DEFINED применяется подобно условию EXISTS наличия заданного файла, но принимает в качестве аргумента имя переменной среды и возвращает истинное значение, если эта переменная определена. Например:

```
@ECHO OFF
CLS
IF DEFINED Username GOTO :VarExists
ECHO Переменная Username не определена
GOTO   :EOF
:VarExists
ECHO Переменная Username определена,
ECHO ее значение равно %Username%
```

Организация циклов

Рассмотрим подробно различные варианты команды FOR. Первый из них реализуется, если указать в команде FOR ключ /D:

FOR /D *%переменная* IN *(набор)* DO *команда [параметры]*

В случае если набор содержит подстановочные знаки, то команда выполняется для всех подходящих имен каталогов, а не имен файлов. Скажем, выполнив следующий командный файл:

```
@ECHO OFF
CLS
FOR /D %%f IN (C:\*.*) DO ECHO %%f.
```

мы получим список всех каталогов на диске C.

С помощью ключа /R можно задать рекурсию в команде FOR:

FOR /R *[[диск:] путь] %переменная* IN (набор) DO *команда [параметры]*

В этом случае заданная команда выполняется для каталога *[диск:]путь,* а также для всех подкаталогов этого пути. Если после ключа /R не указано имя

каталога, то выполнение команды начинается с текущего каталога. Например, для распечатки всех файлов с расширением txt в текущем каталоге и всех его подкаталогах можно использовать следующий пакетный файл:

```
@ECHO OFF
CLS
FOR /R %%f IN (*.txt) DO PRINT %%f
```

Если вместо набора указана только точка (.), то команда проверяет все подкаталоги текущего каталога. Например, если мы находимся в каталоге c:\text с двумя подкаталогами books и articles, to в результате выполнения файла:

```
@ECHO OFF
CLS
FOR /R %%f IN (.) DO ECHO %%f
```

на экран выведутся три строки:

```
C:\TEXT\.
C:\TEXT\BOOKS\.
C:\TEXT\ARTICLES\.
```

Ключ /L позволяет реализовать с помощью команды FOR арифметический цикл, в этом случае синтаксис имеет следующий вид:

FOR /L *%первменная* IN *(начало,шаг, конец)* DO *команда [параметры]*

заданная после ключевого слова in тройка *(начало,шаг,конец)* раскрывается в последовательность чисел с заданными началом, концом и шагом приращения. Так, набор (1,1,5) раскрывается в (1 2 3 4 5), а набор (5,-1,1) заменяется на (5 4 3 2 1). Например, в результате выполнения следующего командного файла:

```
@ECHO OFF
CLS
FOR /L %%f IN (1,1,5) DO ECHO %%f
```

переменная цикла %%f пробежит значения от 1 до 5, и на экране напечатаются пять чисел:

```
1
2
3
4
5
```

Числа, получаемые в результате выполнения цикла FOR /L, можно использовать в арифметических вычислениях. Рассмотрим командный файл my.bat следующего содержания:

```
@ECHO OFF
CLS
FOR /L %%f IN (1,1,5) DO CALL :2 %%f
GOTO :EOF
:2
```

```
SET  /A M=10*%1
ECHO  10*%1=%M%
```

В третьей строке в цикле происходит вызов нового контекста файла my.bat с текущим значением переменной цикла %%f в качестве параметра командной строки, причем управление передается на метку : 2. В шестой строке переменная цикла умножается на десять, и результат записывается в переменную M. Таким образом, в результате выполнения этого файла выведется следующая информация:

10*1=10 10*2=20 10*3=30 10*4=40 10*5=50

Самые мощные возможности (и одновременно самый запутанный синтаксис) имеет команда for с ключом */F*.

FOR /F [«ключи»] %переменная IN (набор) DO команда [параметры]

Здесь параметр *набор* содержит имена одного или нескольких файлов, которые по очереди открываются, читаются и обрабатываются. Обработка состоит в чтении файла, разбиении его на отдельные строки текста и выделении из каждой строки заданного числа подстрок. Затем найденная подстрока используется в качестве значения переменной при выполнении основного тела цикла (заданной команды).

По умолчанию ключ /F выделяет из каждой строки файла первое слово, очищенное от окружающих его пробелов. Пустые строки в файле пропускаются. Необязательный параметр *«ключи»* служит для переопределения заданных по умолчанию правил обработки строк.

При использовании ключа tokens=x,y,m-n создаются дополнительные переменные. Формат m-n представляет собой диапазон подстрок с номерами от м до N. Если последний символ в строке tokens= является звездочкой, то создается дополнительная переменная, значением которой будет весь текст, оставшийся в строке после обработки последней подстроки.

Разберем использование этой команды на примере пакетного файла parser.bat, который производит разбор файла myfiie.txt:

```
@ECHO OFF
IF NOT EXIST myfile.txt GOTO :NoFile
FOR /F "EOL=; TOKENS=2,3* DELIMS=, " %%i IN (myfile.txt) DO
@ECHO %%i %%j %%k
GOTO :EOF
:NoFile
ECHO Не найден файл myfile.txt!
```

Здесь во второй строке производится проверка наличия файла myfiie.txt, в случае отсутствия этого файла выводится предупреждающее сообщение.

Команда for в третьей строке обрабатывает файл myfiie.txt следующим образом:

Пропускаются все строки, которые начинаются с символа точки с запятой (EOL=;).

Вторая и третья подстроки из каждой строки передаются в тело цикла, причем подстроки разделяются пробелами (по умолчанию) и/или запятыми (delims=,).

3. В теле цикла переменная %%i используется для второй подстроки, %%j – для третьей, а %%к получает все оставшиеся подстроки после третьей.

В нашем примере переменная %%i явно описана в инструкции for, а переменные %%j и %%к задаются неявно с помощью ключа tokens=. Например, если в файле myfiie.txt были записаны следующие три строки:

ддд БББББ ВВВВ,ГГГГГ ДДДД

ЕЕЕЕЕ,ЖЖЖЖ ЗЗЗЗ

;КККК ЛЛЛЛЛ ММММММ

то в результате выполнения пакетного файла parser.bat на экран выведется

следующее:

БББББ ВВВВ ГГГГГ ДДДД

ЖЖЖЖ ЗЗЗЗ

Замечание

Ключ tokens= позволяет извлечь из одной строки файла до 26 подстрок, поэтому запрещено использовать имена переменных, начинающиеся не с букв английского алфавита (a-z). Следует помнить, что имена переменных for являются глобальными, поэтому одновременно не может быть активно более 26 переменных.

Команда for /f также позволяет обработать отдельную строку. Для этого следует ввести нужную строку в кавычках вместо набора имен файлов в скобках. Строка будет обработана так, как будто она взята из файла. Например, файл следующего содержания:

```
@ECHO OFF
FOR /f "EOL=; TOKENS=2,3* DELIMS=, " %%i IN ("AAA БББББ ВВВВ,ГГГГГ ДДДД") DO @ECHO %%i %%j %%k
```

При своем выполнении напечатает

БББББ ВВВВ ГГГГГ ДДДД

Команда for /f позволяет обработать строку вывода другой команды. Для этого следует вместо набора имен файлов в скобках ввести строку вызова команды в апострофах (не в кавычках!). Строка передается для выполнения интерпретатору команд cmd.exe, а вывод этой команды записывается в память и обрабатывается так, как будто строка вывода взята из файла. Например, следующий командный файл:

```
@ECHO OFF
```

```
CLS
ECHO Имена переменных среды:
ECHO
FOR /F "DELIMS==" %%i IN ('SET') DO ECHO %%i
```

В Windows также расширены операции подстановки ссылок на переменные команды for.

Например, следующий командный файл выведет полные имена всех файлов с расширением txt.

```
@ECHO OFF
CLS
FOR %%i IN (*.txt) DO ECHO %%~Fi
```

Команды *PUSHD* и *POPD*

Иногда бывает необходимо в командных файлах запоминать текущий каталог, переходить в другой каталог, выполнять какие-либо действия, а затем возвращаться в исходный Для этого используются две команды

PUSHD И POPD

Команда PUSHD сохраняет имя текущего каталога для команды POPD и осуществляет переход в другой каталог Ее синтаксис имеет вид

PUSHD [путь | …]

Здесь параметр *путь* задает каталог, который будет сделан текущим.

Когда включена расширенная обработка команд, команда PUSHD допускает ввод сетевых путей в дополнение к обычным именам дисков и путям Например

PUSHD \\Server1\Programs

Если указан сетевой путь, команда PUSHD создает временное имя диска, указывающее на заданный сетевой ресурс, а затем производит смену текущего диска и каталога, используя вновь определенное имя диска Выделение временных имен дисков проводится в обратном порядке, начиная с Z:, причем выбирается первое свободное имя диска

Вновь сделать текущим каталог, сохраненный командой PUSHD, можно с помощью команды POPD. Команда POPD может быть использована только один раз для изменения каталога, после чего буфер будет очищен Когда включена расширенная обработка команд, команда popd удаляет временные имена дисков, созданные pushd для сетевых ресурсов

Таким образом, PUSHD и POPD можно использовать в командных файлах для возврата в каталог, откуда была вызвана пакетная программа Приведем пример пакетного файла deltxt.bat, который удаляет все файлы с расширением txt в каталоге, заданном первым параметром командной строки.

```
@ECHO OFF
IF -%1==- GOTO :NoParam
```

```
REM Переходим в нужный каталог
PUSHD  %1
DEL *  txt
REM Возвращаемся в исходный каталог
POPD
CLS
ECHO Все текстовые файлы в каталоге %1 удалены!
GOTO :EOF
:NoParam
ECHO Не задано имя каталога!
PAUSE
```

2. Задание на лабораторную работу: написать командный файл

Вариант 0. Пусть имеется текстовый файл protokol.txt, в котором хранится журнал обработанных файлов в следующем формате:

Имя:	file1.txt	Дата: 02.01.2001	Время:	14:50
Имя:	file22.txt	Дата: 03.02.2001	Время:	23:50
Имя:	letter2.txt	Дата: 02.01.2001	Время:	12:00
Имя:	soft.txt	Дата: 10.01.2000	Время:	13:00
…				

Слово дата здесь начинается в каждой строке с двадцатой позиции. Необходимо написать командный файл, с помощью которого сделать выборку из этого файла (т. е. создать новый текстовый файл с нужной информацией) за заданный в командной строке месяц (мм) и год (гггг) в файл out.txt, сформированный файл упорядочить по дате обработки. Нужные месяц и год указать как параметры командной строки.

Если пакетный файл запускается вообще без параметров, то вывести описание его синтаксиса.

Вариант 1. Написать командный файл, который будет копировать из текущего каталога все файлы с расширением txt, кроме одного файла, указанного в качестве второго параметра командной строки, в каталог, указанный первым параметром. Если имя каталога, в который должно производиться копирование, не задано, то вывести сообщение об этом, если будет введен символ Y – продолжить работу, т. е. создать каталог и скопировать файлы, N – прервать выполнение файла.

Если пакетный файл запускается вообще без параметров, то вывести описание его синтаксиса.

Вариант 2. Создать командный файл, который выводил бы содержимое каталога, указанного в качестве параметра командной строки, причем пользователю должна быть предоставлена возможность выбора с помощью меню

устройства для вывода: на экран (информация выводится по одному экрану), в текстовый файл catalog.txt.

Командный файл должен обрабатывать два ключа:

/a – сортировка выводимой информации по алфавиту

/d – по дате создания.

Если пакетный файл запускается вообще без параметров, то вывести описание его синтаксиса.

Вариант 3. Написать командный файл, который печатал бы общее число переменных среды, а всю остальную информацию о переменных выводил в заданный в качестве параметра командной строки текстовый файл, который затем открывается в Блокноте (Notepad). Если файл для вывода не задан, то выводить список на экран. Также предусмотреть два дополнительных ключа: если задан ключ /b, то выводить только имена переменных (без значений, как в предыдущем упражнении), если задан ключ /a, то выводить имя переменной и ее значение в круглых скобках:

По умолчанию считать заданным ключ /b. Если пакетный файл запускается вообще без параметров, то вывести описание его синтаксиса.

Вариант 4. Написать пакетный файл, который автоматически удалял бы в каталоге, указанном в командной строке все подкаталоги, размер которых превышает 20 Мбайт.

Если пакетный файл запускается вообще без параметров, то вывести описание его синтаксиса.

Вариант 5. Пусть имеется текстовый файл sums.txt с разделителями следующего формата:

Фамилия|Имя|Отчество|Сумма

Например:

Петров|Петр|Петрович|1450

Иванов|Иван|Иванович|1200

Необходимо написать пакетный файл seeksum.cmd, который запускался бы с двумя параметрами командной строки:

MIN MAX

где min – минимальная сумма, мах – максимальная сумма, и искал в файле sums.txt всех людей, у которых сумма меньше либо равна мах, но больше либо равна min. Информацию выводить в файл suminfo.txt, причем фамилии должны идти в алфавитном порядке.

Если пакетный файл запускается вообще без параметров, то вывести описание его синтаксиса.

Контрольные вопросы:

1. Рассмотрите особенности описания уязвимостей информации, защищенности информации, угроз безопасности информации, защите информации,

которые используются при сканировании. Перечислите источники угроз безопасности информации.

2. Охарактеризуйте основные принципы функционирования программ поиска уязвимостей и собора данных.

3. Проясните положительные и отрицательные особенности программ поиска уязвимостей и собора данных.

Практическая работа № 3
Организация консоли администрирования в ОС Windows

Цель работы: Изучить способы централизованного управления Windows XP с помощью Microsoft Management Console

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2

1.Теоретический материал

Консоль управления **Microsoft Management Console (MMC)** – это основа администрирования и управления системы Windows XP. Это средство операционной системы, которое предоставляет своим встроенным (интегрированным) компонентам или, другими словами, системным приложениям, удобный для использования графический интерфейс.

Сама по себе MMC не содержит средств администрирования, сила ее в том, что она позволяет в любой момент добавлять новые интегрированные компоненты и координировать работу уже установленных. Консоль управления MMC работает на любой платформе Win32 (например, Windows XP, Windows 2000, Windows NT 4.0, Windows 9x). Разработчики Microsoft почти все инструменты управления Windows XP встроили в систему в виде «оснасток» (snap-ins) MMC.

С помощью MMC существует возможность объединять встроенные в систему (интегрированные) компоненты, создавая собственные надежные средства управления компьютерами предприятия. Созданные таким образом управляющие системы можно сохранить в файлах с расширением .msc (Management Saved Console – сохраненная консоль управления) и распространять их в пределах всей системы (например, задавая к ним доступ с помощью ярлыков или элементов меню **Пуск**, отправляя их по почте или размещая на страницах Web).

2. Задание к работе

Задание 1. Знакомство с Microsoft Management Console

1. Зарегистрируйтесь в системе.
2. Запустите консоль Управление компьютером:
3. Запустите Панель управления (Пуск | Настройка | Панель управления).
4. Выберите Администрирование| Управление компьютером. Другой способ вызвать эту консоль – щелкнуть правой кнопкой мыши по значку рабочего стола Мой компьютер, выбрать пункт меню Управление.
5. В оснастке **Локальные пользователи и группы** добавьте пользователя **студент** локального компьютера в группу **Администраторы.**
6. Для этого выберите Группы | Администраторы| Добавить.В появившемся диалоговом окне **Выбор** измените область поиска **Искать в** на имя локального компьютера, выберите **Студент** и нажмите кнопку **Добавить**.Щелкните кнопку ОК.
7. Создайте общую папку на локальном компьютере и назначьте на нее

права доступа.

8. В оснастке **Общие папки** выберите **Ресурсы**.

9. В меню **Действие** выберите **Новый общий файл**.

10. В поле **Общая папка** выберите каталог C:\Lab Files с помощью кнопки **Обзор**.

11. В поле **Сетевое имя** напишите Labfiles и нажмите кнопку **Далее**.

12. Выберите **Администраторы имеют общий доступ**, **остальные имеют доступ только для чтения**.

13. Нажмите **Готово**.

14. Для ответа на вопрос «Хотите создать еще одну папку?» нажмите кнопку **Нет**.

Задание 2. Создание пользовательской консоли mmc.

В системе Windows XP любой интегрируемый компонент может быть включен в состав новой или уже существующей консоли.

1. Зарегистрируйтесь в системе как **студент** локального компьютера.

2. Создайте собственную консоль **mmc**. Для этого:

3. Запустите mmc (**Пуск | Выполнить | MMC**)

4. Из меню **Консоль** выберите **Добавить или удалить оснастку,** нажмите Add, выберите **Управление дисками** и нажмите **Добавить**.

5. В окне **Выбор компьютера** выберите **Локальный компьютер**. Нажмите **Готово**.

6. Нажмите **Закрыть** в окне **Добавить изолированную оснастку**.

7. Нажмите **OK** в основном окне

Теперь у Вас есть своя собственная консоль MMC, в которой находится только Disk Management. Для того, чтобы сохранить эту консоль, выберите пункт **Сохранить как** в меню **Консоль**, введите, например, «Консоль1» в качестве названия консоли и нажмите **Сохранить**. Теперь в меню **Программы** появится новая папка My Administrative Tools, в которой будет находиться консоль MMC по имени «Консоль 1».

Выйдите из консоли **Консоль1** , выбрав **Консоль | Выход.**

Задание 3. Знакомство с интерфейсом управления системными политиками (System Policy).

1. Запустите консоль mmc.Выберите **Добавить или удалить оснастку**, нажмите **Добавить| Групповая политика. Убедитесь,** что в поле **Объект групповой политики** выбран **Локальный компьютер .** Нажмите **Готово.**

2. В окне **Добавить изолированную оснастку** нажать кнопку **Закрыть**.

3. В окне **Добавить/Удалить оснастку** нажать **OK**.

4. В левой половине окна консоли выберите **Политика «Локальный компьютер»| Конфигурация Windows| Параметры безопасности| Локальные политики | Назначение прав пользователей.**

Задание 4. Отмена изменений конфигурации операционной системы.

1. Удалите общий доступ к папке C:\Lab Files.

2. Удалите пользователя студент локального компьютера из группы Администраторы.

Контрольные вопросы

1. Консоль администрирования mmc. Назначение, функции, особенности.
2. Понятие общего ресурса. Создание общего ресурса.
3. Понятие общего ресурса. Разрешения для общего ресурса. Комбинация разрешений на общий ресурс и разрешений NTFS.
4. Понятие общего ресурса. Два способа обращения к общему ресурсу.
5. Что такое привилегии и в чем их отличие от прав доступа? Приведите примеры привилегий.
6. Что такое привилегии? С помощью какого инструмента производят настройку привилегий пользователя?
7. Какие средства администрирования Windows XP Professional Вы знаете? Назовите их основные особенности.

Лабораторная работа № 4
Мониторинг, оптимизация и аудит ОС Windows

Цель работы: Познакомиться с реестром Windows XP. Изучить основные средства наблюдения за распределением виртуальной памяти в ОС Windows XP Professional.

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2

1.Теоретический материал

Физическая память представляет собой упорядоченное множество ячеек и все они пронумерованы, то есть с каждой из них можно обратиться, указав ее порядковый номер (адрес). Количество ячеек физической памяти ограничено и фиксировано.

Виртуальная память создает иллюзию того, что каждый процесс имеет доступ к 4Гб непрерывного адресного пространства. Виртуальное адресное пространство процесса является набором адресов, доступным всем нитям этого процесса.

Windows XP распределяет адресное пространство физической и виртуальной памяти страницами (pages) – блоками по 4Кб.

Страницы виртуальной памяти имеют три состояния:

1. Большинство страниц пусто, поскольку процесс их не использует;
2. Используемые страницы отображаются с помощью невидимого для процесса указателя в область физической оперативной памяти (ОЗУ);
3. Некоторые страницы, к которым не было обращений в течение определенного времени, отображаются с помощью невидимого для процесса указателя в 4Кб раздел файла подкачки (pagefile.sys).

Процесс управления местоположением страниц – в ОЗУ или в страничном файле называется **подкачкой страниц по запросу.**

Реестр – это унифицированная база данных, в которой Windows XP/2003 хранит всю информацию о конфигурации оборудовании и программного обеспечения локального компьютера. Реестр управляет ОС Windows XP/2003, предоставляя информацию, используемую при запуске приложений и загрузке компонентов, например драйверов устройств и сетевых протоколов.

Реестр содержит следующую информацию:

1. Об оборудовании, установленном на компьютере, включая центральный процессор, тип шины, указательное устройство или мышь и клавиатуру.
2. Об установленных драйверах устройств; установленных приложениях.
3. Об установленных сетевых протоколах.
4. О настройках платы сетевого адаптера (номер прерывания, базовый адрес памяти, базовый адрес портов ввода-вывода, тип трансивера).

5. Об учетных записях пользователей (например, о принадлежности пользователей группам, их правах доступа и привилегиях).

Разделяют логическую и физическую структуру реестра. Логическая структура реестра отображена в редакторе реестра regedit.exe и состоит из ветвей, ключей и т.д. Физическая структура отражает порядок, в котором файлы реестра (кусты) хранятся на жестком диске. Всю необходимую информацию можно получить в **Центре Справки и Поддержки ОС** Windows XP.

2. Задание на лабораторную работу: выполнить задание по модификации реестра

Задание 1. Работа с реестром Windows XP, получение информации о настройках диспетчера памяти.

1. Создайте ярлык для **Редактора реестра.** Щелкните правой кнопкой в любом месте рабочего стола. Щелкните **Создать**, а затем – **Ярлык**. В поле **Укажите размещение объекта** введите **regedit.exe.** Щелкните кнопки **Далее**, а затем – **Готово**. На рабочем столе появится значок ярлыка для программы regedit.exe.

2. Познакомьтесь со структурой реестра.

3. **Чтобы просмотреть реестр, з**апустите **Редактор реестра**, дважды щелкнув его ярлык.

4. Составьте список пяти ветвей реестра.

Как и большинство компонентов Windows XP, диспетчер памяти старается автоматически оптимизировать работу систем различных масштабов и конфигураций при разных уровнях загруженности. Стандартные настройки можно изменить через параметры в разделе реестра HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management. Часть этих параметров перечислена ниже:

1. ClearPageFileAtShutdown. Указывает, надо ли заполнять нулями неактивные страницы в страничном файле при завершении работы системы. Включение этого параметра обеспечивает дополнительную защиту.

2. DisablePagingExecutive. Определяет, можно ли выгружать системный код и драйверы устройств в страничный файл на то время, когда они не используются. Если этот параметр равен 0 (по умолчанию), драйверы и системный код должны оставаться в физической памяти. Если же он равен 1, драйверы и системный код можно при необходимости выгружать в страничный файл.

3. IoPageLockLimit. Задаст максимальное число байт, блокируемых в пользовательском процессе для операций ввода-вывода. Если этот параметр равен 0, система использует лимит по умолчанию (512 Кб). Предельно возможное значение примерно равно объему физической памяти за вычетом 7 Мб.

4. LargePageMinimum. Определяет минимальный объем памяти (в Мб) для проецирования Ntoskrnl и HAL с использованием больших страниц (по 4 Мб). Этот параметр не документирован и по умолчанию отсутствует, его нужно добавлять вручную.

5. LargeSystemCache. Определяет, чему будет отдан приоритет при нехватке памяти – кэшу файловой системы или рабочим наборам процессов. Также влияет на размер кэша файловой системы. (В Windows XP Server этот параметр можно задать косвенно, через свойства службы файлового сервера).

6. NonPagedPoolQuota. Указывает максимальный объем неподкачиваемой памяти (в Мб), который можно выделять какому-либо процессу. Если этот параметр равен 0, данное значение определяется самой системой

7. NonPagedPoolSize. Задает начальный размер пула неподкачиваемой памяти (в байтах). Если этот параметр равен 0, данное значение определяется самой системой.

8. PagedPoolQuota. Указывает максимальный объем подкачиваемой памяти (в Мб), который можно выделять какому-либо процессу. Если этот параметр равен 0, данное значение определяется самой системойSystemPages. Определяет число элементов в системной таблице страниц, зарезервированных для проецирования на системное адресное пространство буферов ввода-вывода, драйверов устройств, стеков потоков ядра и страниц, используемых для программного ввода-вывода. Если этот параметр равен 0, данное значение.

9. SystemPages. Определяет число элементов в системной таблице страниц, зарезервированных для проецирования на системное адресное пространство буферов ввода-вывода, драйверов устройств, стеков потоков ядра и страниц, используемых для программного ввода-вывода. Если этот параметр равен 0, данное значение выбирается самой системойвыбирается самой системой.

С помощью Редактора реестра и Панели управления произведите настройку и мониторинг файла подкачки Windows XP.

1. Откройте раздел реестра HKLM\SYSTEM\CurrentControlSet\Control\ \Session Manager\Memory Management и, исследуя значения параметров этого раздела, найдите место расположения, название, размер файла подкачки. Обратите внимание на имена, типы и значения параметров в правом окне. Запишите значения в лабораторную тетрадь.

2. Закройте Редактора реестра.

3. Нажмите Пуск | Настройка | Панель управления.

4. Дважды щелкните на значке Система, перейдите на закладку Дополнительно, в окне настроек Быстродействие нажмите кнопку Параметры, перейдите на вкладку Дополнительно

5. В окне Виртуальная память изучите общий объем файла подкачки на всех дисках. Запишите значение в лабораторную тетрадь.

6. Нажмите кнопку Изменить. Установите размер файла подкачки на диске C: в соответствии с указаниями преподавателя.

7. Запустите Редактор реестра, откройте раздел реестра HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management и запишите новые место расположения, название, размер файла подкачки значения в лабораторную тетрадь.

8. Верните первоначальные настройки виртуальной памяти.

9. Проверьте текущий размер файла подкачки на диске, используя **Проводник** и запишите его размер в лабораторную тетрадь.

10. Дополнительно: изучите системные параметры, которые возможно изменять с помощью значка панели управления **Система.**

Задание 2. Просмотр информации о виртуальной памяти в Диспетчере задач.

Базовую информацию о системной памяти можно получить на вкладке **Быстродействие** в **Диспетчере задач**, как показано ниже. Эти сведения являются подмножеством информации о памяти, предоставляемой счетчиками производительности.

Задание 3. Наблюдение за использованием памяти с помощью утилиты Performance Monitor (Производительность).

Объекты счетчиков производительности **Память** и **Процесс** открывают доступ к большей части сведений об использовании памяти системой и процессами. Для получения информации об этих счетчиках нажмите кнопку **Объяснение** в окне **Добавить счетчики**.

1. Запустите | Системный монитор (c:\windows\system32\perfmon.exe).
2. Удалите счетчики по умолчанию.
3. Добавьте счетчики:

Объект Память| счетчик: Байт выделенной виртуальной памяти.

Объект Память| счетчик: Предел выделенной виртуальной памяти.

Объект Память | счетчик: Процент использования выделенной памяти.

4. Пронаблюдайте использование памяти процессом CPU Stress с помощью следующих счетчиков:

Объект Процесс | счетчик: Байт виртуальной памяти| вхождения: cpustress

Объект Процесс | счетчик: Байт исключительного пользования | вхождения: cpustress

Объект Процесс | счетчик: Байт файла подкачки | вхождения: cpustress

Запишите в лабораторную тетрадь средние значения этих счетчиков и их интерпретацию (то, что они означают).

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы

1. Что такое реестр? Перечислите пять ветвей реестра и их основное содержание.

2. Опишите иерархическую структуру реестра, расположение файлов реестра на диске.

3. Какие средства изменения информации в реестре Вы знаете? Перечислите типы данных параметров реестра.

4. Что представляет собой физическая память, и как вы понимаете понятие Виртуальная память? Что такое страница? Что называют рабочим набором?

Лабораторная работа № 5
Работа с Реестром ОС Windows

Цель работы: изучение и освоение порядка работы с реестром

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2

1.Теоретический материал

Реестр и его роль

Реестр (registry) представляет собой иерархически организованную базу данных, которую ОС использует для запуска приложений, хранения системных правил, профилей пользователей и прочих настроек и т. д. Также реестр используется практически всеми приложениями для хранения своих настроек.

Впервые реестр был использован в Windows 95, с тех пор он стал быстрее и лучше защищен от всевозможных ошибок.

Хотя реестр организован довольно понятным образом, он представляет собой сложную и обширную структуру.

Реестр можно рассматривать как записную книжку Windows - как только системе нужна какая-то информация, она ищет ее в реестре. Реестр очень обширен, и дать однозначное его определение невозможно. Кратко и достаточно точно можно сказать, что реестр – компонент операционной системы компьютера, который в иерархической базе данных хранит важнейшие установки и информацию о приложениях, системных операциях, пользовательской и аппаратной конфигурациях.

История реестра

Изначально каждая программа хранила нужные для себя настройки и данные своим собственным способом, как правило – в виде файлов собственной структуры. Поэтому при переносе программы с одной машины на другую достаточно было настроить эти файлы (как правило – прописать нужные имена дисков и каталоги, так как каждая машина имела свои диски). С другой стороны, эта процедура – тогдашний эквивалент инсталляции – порой была очень сложной. В Windows 3.x была сделана попытка упорядочить формат и способ хранения конфигурационных файлов. В частности, всем им было предписано носить расширение .ini.

Реестр был создан потому, что с файлами INI, которые появились в Windows 3.x, пользователям приходилось думать, какой файл INI за что отвечает и как изменить в нужном файле ту или иную настройку. Часто было трудно выяснить месторасположение таких файлов для нужной программы. Кроме того, отсутствовал способ определения того, какая программа связана с конкретным INI- файлом. С другой стороны, редактировать такие файлы было намного легче, чем реестр. Одним из самых важных различий между файлами INI и реестром являлось расположение файлов – INI-файлы принадлежали программе, в то время, как реестр – часть Windows. Именно поэтому в наше время невозможно «просто перенести» программу – требуется записать ее настройки в реестр, что, собственно, и составляет процедуру инсталляции программы.

Где расположены файлы реестра

В ОС Windows 9x реестр хранится в двух файлах: System.dat и User.dat. Эти два файла находятся в папке с Windows. Если на данном компьютере несколько пользователей, то система создает несколько файлов User.dat.

В Windows NT (2000/XP) есть специальный каталог SYSTEM32\CONFIG, хранящий в виде защищенных файлов разделы реестра.

Роль реестра

Реестр можно рассматривать как записную книжку Windows - как только системе нужна какая-то информация, она ищет ее в реестре. Реестр очень обширен, и дать однозначное его определение невозможно. Кратко и достаточно точно можно сказать, что реестр – компонент операционной системы компьютера, который в иерархической базе данных хранит важнейшие установки и информацию о приложениях, системных операциях, пользовательской и аппаратной конфигурациях.

Архитектура реестра

В целом реестр очень напоминает файловую систему с той разницей, что вместо файлов на нижнем уровне содержатся параметры.

Информация, хранящаяся в иерархической базе данных реестра, собрана в разделы (key), которые содержат один или более подразделов (subkey). Каждый подраздел содержит параметры (value):

Возможность создавать вложенные подразделы позволяет группировать параметры. В результате получается древовидная структура, которую можно просмотреть в Редакторе реестра (Registry editor, RegEdit). Каждый раздел (ветвь) соответствует определенному типу информации о пользователе, аппаратном обеспечении, приложении и т. д.

Изменяя тот или иной параметр, можно управлять работой Windows, защитить компьютер от нежелательных пользователей и просто настраивать внешний вид Windows.

В частности, в разделе

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

сдержится список параметров. Имена этих параметров не играют роли для системы, а значения представляют собой имена исполняемых файлов, которые следует запускать всякий раз при запуске системы. Добавив туда свой параметр, можно заставить систему запускать свою программу.

Редактирование реестра

RegEdit – это программа, которая позволяет редактировать файлы реестра. Запустить ее можно из командной строки, либо через меню Start->Run. В левой части окна отображается вся иерархическая структура реестра, в правой – параметры, наличествующие в текущем разделе. Следует отметить, что параметры могут находиться даже в корне реестра.

Файлы реестра

Regedit позволяет импортировать и экспортировать часть реестра в файлы. Эта возможность, например, может быть использована для создания ре-

зервных копий, либо для переноса ПО с одной машины на другую. Структура этих файлов такова:

REGEDIT4
[раздел реестра]
«параметр»=«строковое значение»

Если параметр имеет тип dword, то соответствующая строка должна иметь вид:

«параметр»=dword:00000000 (где вместо 00000000 надо задать нужное значение).

А если тип параметра двоичный, то формат строки:

«параметр»=hex:00,00,00,00 (где через запятую указываются значения байтов в шестнадцатеричном виде).

Следует обратить внимание, что в конце файла (*.reg) обязательно должна быть пустая строка.

Кроме того, следует учитывать, что в значении строкового параметра перед символами «кавычки» и «обратный слеш» (\) должен добавляться символ «обратный слеш». А параметр «(По умолчанию)» обозначается символом «@» (без кавычек).

Пример задания параметра

Чтобы присвоить параметру «(По умолчанию)» значение

"C:\Program Files\Accessories\WORDPAD.EXE" "%1", надо записать:

@="\"C:\\Program Files\\Accessories\\WORDPAD.EXE\" \"%1\""

Созданный reg-файл следует запустить на выполнение (с файлами типа *.reg по умолчанию ассоциирован редактор реестра REGEDIT.EXE).

Пример файла, содержащего параметр для автоматического запуска драйвера мыши

REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

"Gnetmous"="G:\\genius\\gnetmous.exe"

Для того чтобы полностью удалить раздел (ключ) реестра с помощью reg-файла, надо перед именем раздела поставить «-» (без кавычек).

В Windows 2000/XP используются файлы реестра 5-й версии (в windows NT – 4-й). Они отличаются поддержкой Unicode (каждый символ занимает два байта вместо одного) и заголовком «Windows Registry Editor Version 5.00» вместо «REGEDIT4». Файлы 4-й версии по-прежнему используются и поддерживаются.

В «свежеустановленной» системе размер реестра составляет примерно 12–15Мб, увеличиваясь со временем до 20–25 Мб. Поскольку реестр содержит чисто текстовую информацию, можно оценить количество параметров, учитывая, что один параметр занимает около 100 байт. В силу этого не существует полного описания реестра (не следует также забывать, что каждая программа

заносит в реестр что-то свое). Однако, есть множество программ, называемых «твикерами» или «настройщиками», предназначенных для удобного редактирования некоторых системных параметров реестра. Эти программы содержат достаточно подробное описание изменяемых параметров и рекомендации по настройке. Наиболее обширной на сегодняшний день является система Xteq X-setup, позволяющая модифицировать около 1000 различных значений. Умелое ее использование способно обеспечить 20–30 % прироста производительности системы.

2. **Задание на лабораторную работу:** выполнить задание в соответствие со своим вариантом

Задание 1. Перед выполнением заданий создайте точку восстановления системы.

1. С помощью редактора реестра изучить корневые разделы системного реестра.

Произведте экспорт реестра

Экспорт Реестра ОС или его части это одна из тех вещей, которые достаточно часто приходится делать системным администраторам и опытным пользователям. Экспорт – копирование данных в другой файл. По отношению к Реестру, этот файл имеет расширение .reg.

Экспорт настроек в **Reg**-файл может использоваться для следующих целей.

Прежде всего, это хороший способ создать резервную копию системных настроек на случай их экстренного восстановления при необходимости. Также появляется возможность передавать настройки другим пользователям на другие компьютеры сети. Имея несколько **Reg**-файлов с различными настройками системы, возможно импортировать их одним двойным щелчком мышью.

Для экспорта ветвей реестра выполните следующие действия:

a) щелкните мышью на разделе (ключе), находящемся в вершине ветви, выбранной самостоятельно, которую необходимо экспортировать (например, HKEY_CURRENT_USER);

b) в меню «ФАЙЛ» выберите пункт «ЭКСПОРТ», чтобы вывести на экран диалоговое окно «ЭКСПОРТ ФАЙЛА РЕЕСТРА»;

c) в поле «ИМЯ ФАЙЛА» введите имя файла для экспорта;

d) выберите диапазон экспорта: чтобы создать копию всего реестра, щелкните на «ВЕСЬ РЕЕСТР», чтобы создать копию выделенной ветви, щелкните на «ВЫБРАННАЯ ВЕТВЬ»;

e) в выпадающем списке «Тип файла» выберите тип файла для экспорта:

f) «Файлы Реестра *.reg», «Файлы кустов Реестра *.*», «Текстовые файлы *.txt» или

g) «Файлы Реестра Win9x/NT4 *.reg»;

h) экспортируйте ветвь, мышью щелкнув на кнопке «СОХРАНИТЬ».

Последовательность вышеописанных действий фактически представляет собой один из способов создания резервной копии Реестра ОС. Сохранение Реестра перед его редактированием является принципиальным, поскольку обеспечивает дополнительный шанс на его восстановление в случае выхода системы из строя посредством непродуманных действий пользователя.

Обратная процедура импорта Реестра практически ни чем не отличается от простого открытия Reg-файла. Для этого необходимо щелкнуть мышью на пункте «ИМПОРТ» в меню «ФАЙЛ», далее в выпадающем списке «ТИП ФАЙЛА» выбрать тип файла, который предполагается импортировать, а затем в поле «ИМЯ ФАЙЛА» ввести полный путь **Reg**-файла и подтвердить операцию, щелкнув по кнопке «ОТКРЫТЬ».

Важно! Файлы Реестра ОС Windows XP представляют собой пятую версию **Reg**файлов. Другие ОС семейства Windows имеют другие версии **Reg**-файлов. Поэтому не импортируйте **Reg**-файл, созданный в одной версии ОС Windows, в другую версию этой ОС. Это может привести к неработоспособности последней.

3. Внесение в системный реестр настроек, запрещающих пользователю полное или частичное изменение свойств Рабочего стола.

Указание: в отчет внести скриншот и полученные выводы.

С помощью ПРОВОДНИКА найти в папке Windows файл regedit.exe и запустить его.

Перейти в раздел реестра

HKEY_CURRENT_USERS\SOFTWARE\ MICROSOFT\WINDOWS\CURRENT VERSION\POLICIES\SYSTEM.

Если при открытии раздела POLICIES окажется, что в нем отсутствует раздел

SYSTEM, создать его, используя команду

ПРАВКА – СОЗДАТЬ – РАЗДЕЛ.

Свернуть окно редактирования реестра и, щелкнув правой кнопкой мыши в свободном месте Рабочего стола, с помощью контекстного меню открыть окно *СВОЙСТВА: ЭКРАН*. Записать перечень закладок окна с настройками экрана, доступными для пользователя, и закрыть окно.

Развернуть окно редактирования реестра и в разделе SYSTEM с помощью команды

ПРАВКА – СОЗДАТЬ – ПАРАМЕТР DWORD

создать ключ *NODISPSettingsPage* и, щелкнув по его имени правой кнопкой мыши, выбрать в появившемся меню команду ИЗМЕНИТЬ. Используя окно ИЗМЕНЕНИЕ ПАРАМЕТРА DWORD (вызов осуществляется через контекстное меню), присвоить созданному ключу значение «1» в шестнадцатеричной системе.

Свернуть окно редактирования реестра и вновь открыть окно *СВОЙСТВА: ЭКРАН.* Изучить перечень закладок, доступных пользователю, и сделать вывод о назначении ключа *NODISPSettingsPage.* Закрыть окно свойств экрана.

Повторить действия, описанные в пунктах 4 и 5, присваивая значение «1» следующим ключам:

− NODISPBackgroundPage;

− NODISPAppearancePage;

− NODISPScrSavPage; − NODISPCPL .

Сделать вывод об их назначении.

4. Создание файлов редактирования реестра, один из которых разрешает, а другой запрещает пользователю изменение настроек Рабочего стола.

Указание: внести в отчет скриншот и полученные выводы.

Хотя файлы редактирования реестра могут создаваться в любом текстовом редакторе (например, *Блокнот*), удобнее получить шаблон такого файла, используя *regedit.* Для этого, не закрывая редактор *regedit* после выполнения задания 3, в разделе *System* удалить все ключи кроме последнего *NODISPCPL*.

Щелкнув мышью по строке с названием раздела *System,* выполнить команду

ФАЙЛ – ЭКСПОРТ

и, указав имя создаваемого файла *file1,* сохранить его в папке *Мои документы.*

Закрыть программу *regedit.*

Перейти в папку *Мои документы* и, щелкнув правой кнопкой мыши по файлу *file1.reg,* выполнить команду

ОТКРЫТЬ С ПОМОЩЬЮ – БЛОКНОТ

Изучить структуру файла *file1.reg.* Его содержимое должно быть примерно следующим:

Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Pol icies\Sy stem]

"NODISPCPL"=dword:00000001

Заменить в последней строке файла значение параметра DWORD с 00000001 на 00000000 и, используя команду ФАЙЛ – СОХРАНИТЬ КАК, сохранить внесенные изменения в файле *file2.reg.*

Закрыть *Блокнот.* Поочередно запуская двойным щелчком на выполнение файлы *file1.reg* и *file2.reg*, произвести попытку редактирования настроек Рабочего стола. Сделать выводы, удалить оба файла.

5. Настройка визуальных опций ОС с помощью системного Реестра.

Указание: в отчет внести выводы по задачам № 1 и 2, скриншоты по заданиям № 3 и 4.

В диалоговом окне «ИЗМЕНЕНИЕ СТРОКОВОГО ПАРАМЕТРА» ключа *HKCU\Control Panel\Desktop* измените значение параметра *MenuShowDelay* на любое число, менее 400. Сделайте вывод о том, как различ-

ные значения этого параметра влияют на раскрытие вложенных списков меню ПУСК.

Скрыть все значки с рабочего стола. Для этого в разделе *HREY_CURRENT_USER\ Software\ Microsoft\ Windows\ CurrentVersion\ Policies\ Explorer* создать параметр *DWORD NoDesktop =1* (=0 - все значки видны). При необходимости выполните перезагрузку виртуальной машины.

Запретить следующие команды в меню ПУСК.

В разделе

HKEY_CURRENT_USER\ Software\ Microsoft\ Windows\ CurrentVersion\ Policies\ Explorer если параметр имеет равен 1, то команда запрещена, 0 – разрешена

a) *NoTrayContextMenu* – запретить контекстное меню панели задач,
b) *NoChangeStartMenu* – запретить контекстное меню в меню ПУСК
c) *NoStartMenuSubfolders* – скрыть подкаталоги в меню ПУСК.
d) *NoRun* – скрыть меню ВЫПОЛНИТЬ в меню ПУСК.
e) *NoFind* скрыть меню НАЙТИ в меню ПУСК.
f) *NoLogOff* скрыть меню ЗАВЕРШЕНИЕ СЕАНСА в меню ПУСК.
g) *NoClose* скрыть меню ЗАВЕРШЕНИЕ РАБОТЫ в меню ПУС
h) HKLM\SOFTWARE\Classes\lnkfile – ярлыки Windows XP

STRING IsShortcut – удаление этого параметра – отключает стрелки на ярлыках.

Не добавлять «ЯРЛЫК ДЛЯ...» для создаваемых ярлыков:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer BINARY link, значение hex:00,00, 00,00 – не добавлять.

6. Настройка меню ПУСК посредством системного реестра.

Указания:

– перенесите последовательность выполняемых действий по каждому из пунктов 1–4 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала);

– результаты применения новых значений системных параметров Реестра ОС перенесите в отчет;

– сделайте вывод о проделанной работе и запишите его в отчет.

Все настройки главного меню «Пуск» находятся в системном Реестре в одном месте *HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ Advanced.*

Для настройки меню «Пуск» посредством Реестра ОС Windows XP, выполните следующие действия:

1) самостоятельно выберите вид главного меню «ПУСК» (классический или новый), соответствующие параметры которого будут применяться в Реестре ОС.

Самостоятельно определите, какие именно параметры будут применены для конфигурирования меню «ПУСК» (в количестве не менее пяти штук),

Самостоятельно конфигурируйте меню «ПУСК» с применением выбранных параметров, результаты конфигурирования меню «ПУСК» зафиксируйте в виде графических фрагментов, сделанных с экрана командой PrintScreen.

7. Создание в системном реестре собственного обработчика произвольного расширения.

Указание:

a) выберите самостоятельно произвольное расширение, состоящее из трех символов, обработчик которого предполагается создать, в разделе HKCR Реестра ОС создайте новый раздел с названием выбранного ранее расширения; при этом обратите внимание на то, как это уже сделано для других расширений в системе, значение строкового параметра (по умолчанию), соответствующего созданному разделу, должно содержать ссылку вида ****file,* где *** – символы выбранного расширения, на раздел обработчика данного расширения, в разделе HKCR Реестра ОС создайте новый раздел обработчика расширения следующего вида ****file\shell\open\command* – для команды открытия и ****file\shell\list\command* – для команды просмотра файла;

b) в разделах *command*, каждой из ветвей, создайте по одному расширяемому строковому параметру типа REG_EXPAND_SZ с наименованием (*по умолчанию*), удалите старые строковые параметры REG_SZ, создаваемые в разделе *command* по умолчанию, в расширяемом строковом параметре раздела ****file\shell\list* измените данные значения по умолчанию на *«Мой просмотр»,* в соответствующих разделах command измените значения расширяемых строковых параметров на команды для открытия файла и его просмотра. В частности, для открытия текстового файла можно воспользоваться приложением WORDPAD.EXE, а для его просмотра выбрать NOTEPAD.EXE, проверьте работоспособность обработчика, выполнив следующее:

− выберите какой-либо файл с его стандартным расширением,

− поменяйте стандартное расширение на то, обработчик которого Вы только что создали,

− правой кнопкой манипулятора мышь выберите из контекстного меню команду с именем того файла (*filename.****),* который Вы собираетесь открыть или команду «Мой просмотр», чтобы просмотреть файл; при этом должно загрузиться соответствующее приложение обработчика.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы:

1. Укажите особенности настройки реестра.
2. Охарактеризуйте основные принципы настройки и организации реевстра.
3. Проясните положительные и отрицательные особенности рестра.

Лабораторная работа № 6
Работа с подсистемой безопасности в ОС Windows

Цель работы: изучение порядка работы с подсистемамим информационной безопасности ОС Windows

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2

1.Теоретический материал

Учетная запись пользователя представляет собой набор данных, сообщающих Windows к каким папкам и файлам пользователь имеет доступ, какие он может делать изменения в работе компьютера, а также персональные настройки пользователя, такие как фон рабочего стола, цветовое оформление и т. д. Учетные записи пользователей позволяют осуществлять работу нескольких пользователей на компьютере, каждый из которых будет иметь свои собственные файлы и настройки. Каждый пользователь получает доступ к своей учетной записи с помощью имени пользователя и пароля.

Существуют основные три типа учетных записей: стандартная, администратор и гость.

Каждый тип дает пользователю разные возможности управления компьютером. Стандартная учетная запись используется при ежедневной работе. Стандартная учетная запись позволяет использовать большую часть возможностей компьютера, но если необходимо сделать изменения, влияющие на всех пользователей или на безопасность компьютера, то потребуется разрешение администратора. Используя стандартную учетную запись, можно работать в большинстве установленных на компьютере программ, но устанавливать новые или удалять старые программы и устройства, удалять необходимые для работы компьютера файлы и изменять настройки, влияющие на всех пользователей компьютера, нельзя. Если используется стандартная учетная запись, некоторые программы могут потребовать пароль администратора для выполнения какихлибо задач.

Учетная запись «Администратор» предоставляет наиболее полный контроль над компьютером и ее рекомендуется применяться только в необходимых случаях. Учетная запись администратора представляет собой учетную запись пользователя, с помощью которой можно делать изменения, затрагивающие других пользователей компьютера. Администраторы могут менять параметры безопасности, устанавливать программное обеспечение и оборудование, а также они имеют доступ ко всем файлам на компьютере. Кроме того, администраторы могут изменять любые учетные записи пользователей.

При установке Windows потребуется создать учетную запись пользователя. Она будет являться учетной записью администратора, позволяющей настраивать компьютер и устанавливать любые программы. После окончания настройки компьютера для повседневного использования рекомендуется ис-

пользовать стандартную учетную запись. Безопаснее использовать стандартную учетную запись пользователя вместо учетной записи администратора.

Учетная запись «Гость» предназначена для временного доступа к компьютеру. Она предназначена для пользователей, не имеющих постоянной учетной записи на компьютере или в домене. Позволяет использовать компьютер без доступа к личным файлам. Пользователи, вошедшие в систему под учетной записью «Гость», не могут устанавливать программное обеспечение и оборудование, изменять настройки или создавать пароль. Перед использованием учетной записи «Гость» ее необходимо включить.

Группа пользователей представляет собой набор учетных записей пользователей, имеющих одинаковые права безопасности. Двумя наиболее распространенными группами пользователей являются группа стандартных пользователей и группа администраторов, но существуют и другие группы. Используя учетную запись администратора, можно создавать новые группы пользователей, перемещать учетные записи из одной группы в другую, добавлять учетные записи в различные группы или удалять их. При создании новой группы пользователей можно самостоятельно определить, какие права к ней будут применены.

Учетную запись часто называют по имени группы, в которую она входит (например, учетная запись, входящая в группу стандартных пользователей, называется стандартная учетная запись). Учетная запись может входить в одну или несколько групп. Группы пользователей также называют группами безопасности.

1. В окне консоли «Управление компьютером» открыть папку «Пользователи» и щелкнуть по новой записи правой кнопкой мыши. В контекстном меню выбрать команду «Свойства».

2. На вкладке «Членство в группах» убедиться, что новый пользователь принадлежит к группе «Пользователи». Это локальная группа безопасности, членам которой разрешен доступ к ресурсам данного компьютера.

Диалоговое окно свойств позволяет настроить еще только те свойства учетной записи, которые имеют отношение к профилю (вкладка «Профиль»). Для изменения остальных свойств Windows XP Professional не предоставляет графического интерфейса.

Используйте следующие шаги, чтобы просмотреть и изменить свойства учетной записи:

1. Выбрать «Пуск» – «Выполнить», появится диалоговое окно «Запуск программы».

2. В появившемся окне в строке «Открыть» ввести «cmd» (для вызова «Командной строки») и нажать «ОК» (рисунок 1).

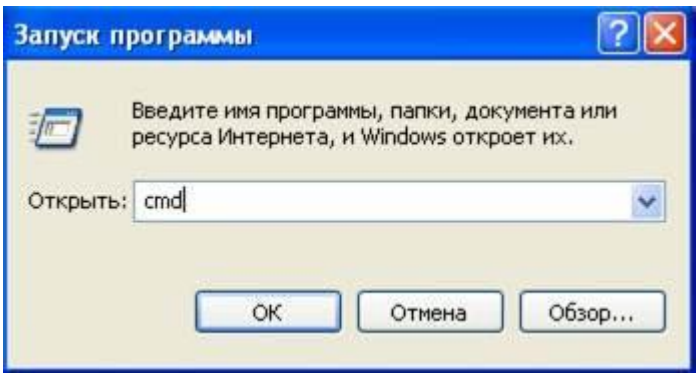

Рисунок 1. Диалоговое окно «Запуск программы»

3. В командной строке ввести команду «netuser» для вывода списка всех локальных учетных записей на этом компьютере.

4. Ввести команду «nethelpuser», которая выведет краткую справку об использовании команды «netuser» и свойствах учетной записи, которые можно настроить с ее помощью.

5. Изучить возможности команд.

6. Посмотреть поочередно все свойства созданных записей, введя команду «netuserимя_пользователя» (результаты данной работы отразить в отчете).

Ограничение срока действия учетной записи

1. Для ограничения учетной записи в командной строке ввести команду «netuserимя_пользователя /expires: (укажите дату число_месяц_год)».

2. Проконтролировать выполнение вводом команды «netuserимя_пользователя» в командной строке (результаты отразить в отчете).

После данных настроек срок действия учетной записи закончится с началом суток (указанные дата (число_месяц_год)). Дату нужно вводить в кратком формате так, как это указано на вкладке «Региональные параметры» окна «Язык и региональные стандарты».

Отменить ограничение можно командой «netuserимя_пользователя /expires: all». **Ограничение времени работы пользователя**

1. Выполнить ограничения соответственно заданию используя команду «times».

2. Проконтролировать выполнение вводом команды «netuserимя_пользователя» в командной строке.

Теперь если пользователь попробует войти в систему вне указанного времени, то он увидит предупреждающее сообщение, а регистрация выполнена не будет.

Снять ограничение входа для пользователя можно командой «netuserимя_пользователя / times: all».

3. Результаты выполнения работы отразить в отчете.

При включении компьютера с установленной на нем операционной системой Windows XP Professional, на экране входа в систему каждая локальная учетная запись представлена значком и регистрационным именем. Чтобы зарегистрироваться в системе, нужно щелкнуть по значку и ввести пароль. После

чего запустится процесс регистрации, по окончании которого перед пользователем появляется его рабочий стол.

Такое положение дел представляет некоторый риск с точки зрения безопасности. Каждый, кто включит компьютер, увидит чужие учетные записи и, если ему повезет, сможет подобрать пароль и причинить неприятности законным пользователям. Скрыть регистрационные имена можно следующим образом:

1. Зарегистрироваться под именем администратора.

2. Выбрать в главном меню «Панель управления» – «Учетные записи пользователей». (Если вы переключили главное меню на классический вид, папка «Панель управления» в группе «Настройка»).

3. В диалоговом окне «Учетные записи пользователей» щелкнуть по ссылке «Изменение входа пользователей в систему». Снять флажок «Использовать страницу приветствия» и нажать кнопку «Применение параметров» (рисунок 2).

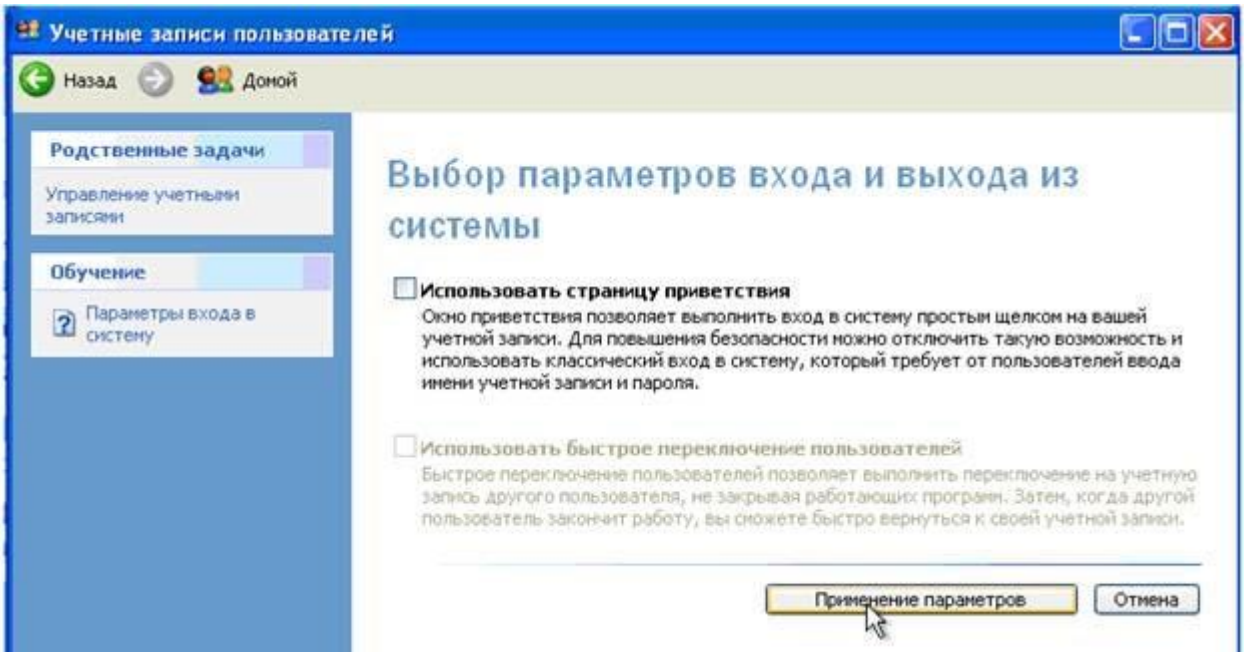

Рисунок 2. Диалоговое окно «Учетные записи пользователей»

4. Выполнить команду «Пуск»/ «Выполнить» и в поле ввода ввести команду «secpol.msc». Откроется окно консоли «Локальные параметры безопасности» (рисунок 3).

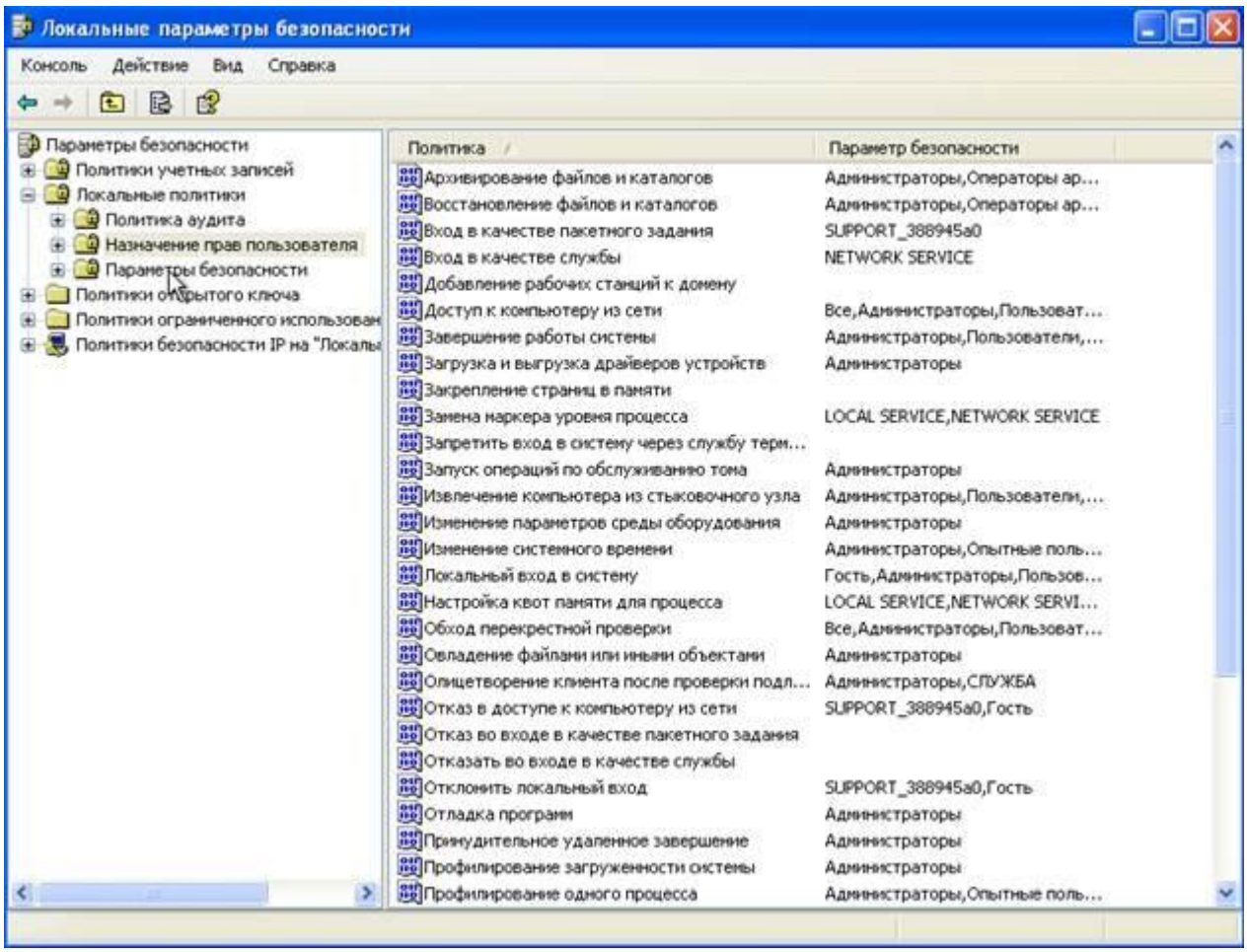

Рисунок 3. Окно консоли «Локальные параметры безопасности»

5. Развернуть группу «Локальные политики» и выбрать пункт «Параметры безопасности».

6. В правой части окна консоли дважды щелкнуть по пункту «Интерактивный вход в систему: не отображать последнего имени пользователя».

7. В появившемся окне свойств поставить переключатель в положение включен, и нажать «ОК» (рисунок 4).

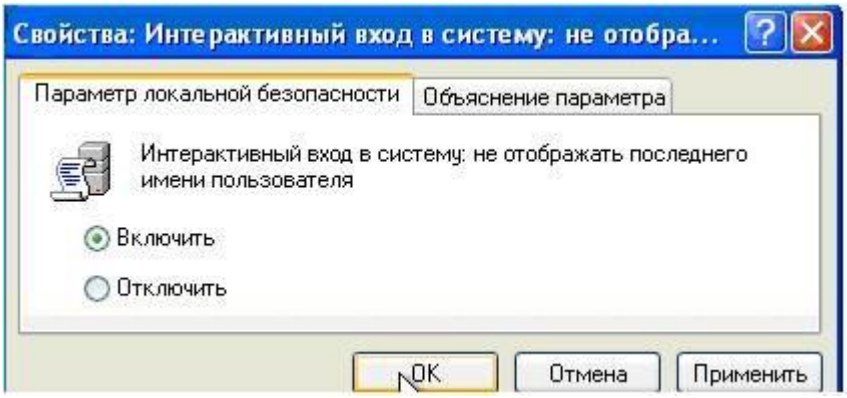

Рисунок 4. Диалоговое окно «Интерактивный вход в систему: не отображать последнего имени пользователя»

Повысить безопасность входа в систему можно, заставив пользователя перед регистрацией нажимать комбинацию клавиш «Ctrl + Alt+Del».

8. Для этого на консоли «Локальные параметры безопасности»отключить режим «Интерактивный вход в систему: не требовать нажатия CTRL+ALT+DEL».

Таким образом, вы помешаете «работе» троянских коней, имитирующих диалог входа в систему с целью перехватить вводимые пользователем имя и пароль: если окно входа принадлежало посторонней программе, то нажатие «Ctrl+Alt+Del» вызовет перезагрузку и управление перейдет к настоящей операционной системе.

9. Изучить возможности консоли «Локальные параметры безопасности».

10. Результаты включить в отчет.

Пользовательская учетная запись содержит имя и пароль для регистрации на локальном компьютере или в домене. В ActiveDirectory (AD) учетная запись пользователя может также содержать дополнительную информацию, такую как полное имя пользователя, адрес электронной почты, номер телефона, отдел и физический адрес. Кроме того, учетная запись пользователя служит средством для назначения разрешений, сценариев регистрации, профилей и домашних каталогов.

В WindowsServer 2003 определены пользовательские учетные записи двух типов: доменные учетные записи и локальные учетные записи.

Доменные учетныезаписи определены в ActiveDirectory. Посредством системы однократного ввода пароля такие учетные записи могут обращаться к ресурсам во всем домене. Они создаются в консоли «ActiveDirectory – пользователи и компьютеры».

Локальные учетные записиопределены на локальном компьютере, имеют доступ только к его ресурсам и должны аутентифицироваться, прежде чем получат доступ к сетевым ресурсам. Локальные учетные записи пользователей создают в оснастке «Локальные пользователи и группы».

Локальные учетные записи пользователей и групп хранятся только на рядовых серверах и рабочих станциях. На первом контроллере домена они перемещаются в ActiveDirectory и преобразуются в доменные учетные записи.

Все учетные записи пользователей распознаются по имени для входа в систему. В WindowsServer 2003 оно состоит из двух частей:

– «имя пользователя» – текстовое имя учетной записи;

– «домен или рабочая группа», вкоторых находится учетная запись.

Например: для пользователя mask, учетная запись которого создана в домене is4.local, полное имя для входа в WindowsServer 2003 выглядит так – mask@is4.local. Имя для предыдущих версий Windows – is4\ mask. При работе с ActiveDirectory иногда требуется полное имя домена пользователя, состоящее из DNS-имени домена в сочетании с именами контейнера и группы. У пользователя is4.local \Users\ mask, is4.local – DNSимя домена, Users – имя контейнера, а mask – имя пользователя.

С учетной записью пользователя могут сопоставляться пароль и открытый сертификат. В открытом сертификате сочетаются открытый и закрытый ключ для идентификации пользователя. Вход в систему по паролю проходит интерактивно. При входе в систему с открытым сертификатом используются смарт-карта и считывающее устройство.

Хотя для назначения привилегий и разрешений в WindowsServer 2003 применяются имена пользователей, ключевым идентификатором учетной записи является генерируемый при создании уникальный идентификатор безопасности (SID). Он состоит из идентификатора безопасности домена и уникального относительного идентификатора, который был выделен хозяином относительных идентификаторов.

С помощью SID, ОС WindowsServer 2003 способна отслеживать учетные записи независимо от имен пользователей. Благодаря наличию SID вы вправе изменять имена пользователей и удалять учетные записи, не беспокоясь, что кто-то получит доступ к ресурсам, создав учетную запись с тем же именем. Когда вы меняете имя пользователя, WindowsServer 2003 сопоставляет прежний SID с новым именем. Когда вы удаляете учетную запись, WindowsServer 2003 считает, что конкретный SID больше недействителен.Если вы затем создадите учетную запись с тем же именем, она не получит привилегий предыдущей записи, так как у нее иной SID.

Помимо учетных записей пользователей в WindowsServer 2003 используются группы, позволяющие автоматически предоставлять разрешения схожим типам пользователей и упростить администрирование учетных записей. Если пользователь – член группы, которая вправе обращаться к ресурсу, то он тоже может к нему обратиться. Чтобы предоставить пользователю доступ к нужным ресурсам, вы просто включаете его в подходящую группу. Поскольку в разных доменах ActiveDirectory могут быть группы с одинаковыми именами, на группы часто ссылаются по полному имени – домен\имя_группы.

В WindowsServer 2003 используются группы трех типов:

– локальные группы определяются и используются только на локальном компьютере, создаются в оснастке «Локальные пользователи и группы»;

– группы безопасности располагают дескрипторами защиты и определяются в доменах посредством консоли «ActiveDirectory – пользователи и компьютеры». Это те группы, для которых можно назначать права и разрешения. Права определяют, какая деятельность разрешается в домене члену подобной группы (пользователю или компьютеру), а разрешения определяют, к каким объектам в сети они будут иметь доступ. Группы безопасности можно использовать и для рассылки e-mail сообщений многим пользователям. Сообщение отсылается лишь один раз, но при этом его получают все члены группы. Для этого, впрочем, в сети должен быть установлен продукт MicrosoftExchangeServer 2003. В этом случае группы безопасности ведут себя так же, как группы распространения;

– группы распространения используются как списки рассылки электронной почты, не имеют дескрипторов безопасности и определяются в доменах посредством консоли «ActiveDirectory – пользователи и компьютеры». Эти группы предназначены только для рассылки пользователям сообщений электронной почты. Для них не определяются права доступа к сетевым объектам.

Группы безопасности имеют все свойства групп распространения, но не наоборот. Но дело в том, что некоторые приложения могут работать только с ними, а не с группами безопасности.

У групп возможны разные области действия – локальная доменная, встроенная локальная, глобальнаяи универсальная. От этого зависит, в какой части сети они действительны.

Локальные доменные группы предоставляют разрешения в одном домене, В состав локальных доменных групп входят лишь учетные записи (и пользователей, и компьютеров) и группы из домена, в котором они определены.

Встроенные локальные группы обладают особыми разрешениями в локальном домене. Для простоты их часто также называют локальными доменными группами, но в отличие от обычных групп, встроенные локальные группы нельзя создать или удалить – можно лишь изменить их состав. Как правило, говоря о локальных доменных группах, имеется в виду и обычные, и встроенные локальные группы, если не указано обратное.

Глобальные группы используются для назначения разрешений на доступ к объектам в любом домене дерева или леса. В глобальную группу входят только учетные записи и группы из домена, в котором они определены.

Универсальные группы управляют разрешениями во всем дереве или лесе; в них входят учетные записи и группы из любого домена в дереве или лесе домена. Универсальные группы доступны только в ActiveDirectory в основном режиме Windows 2000 или в режиме WindowsServer 2003.

Универсальные группы очень полезны на больших предприятиях, имеющих несколько доменов. Состав универсальных групп не должен часто меняться, так как любое изменение надо реплицировать во все глобальные каталоги в дереве или лесе. Чтобы уменьшить количество изменений, включайте в универсальную группу только группы, а не сами учетные записи.

В WindowsServer 2003 учетные записи групп, как и учетные записи пользователей, различаются по уникальным идентификаторам безопасности. Это значит, что нельзя удалить учетную запись группы, а затем создать группу с тем же именем, чтобы у нее появились прежние разрешения и привилегии. У новой группы будет новый SID, и все разрешения и привилегии старой группы будут утеряны.

Для каждого сеанса пользователя в системе WindowsServer 2003 создает маркер безопасности, содержащий идентификатор учетной записи пользователя и SID всех групп безопасности, к которым относится пользователь. Размер маркера растет по мере того, как пользователь добавляется в новые группы безопасности. Это приводит к следующим последствиям:

– чтобы пользователь вошел в систему, маркер безопасности должен быть передан процессу входа в систему. Поэтому по мере увеличения членства пользователя в группах безопасности процесс входа требует все больше времени;

– чтобы выяснить разрешения доступа, маркер безопасности пересылается на каждый компьютер, к которому обращается пользователь. Поэтому чем больше маркер безопасности, тем выше сетевой трафик.

Сведения о членстве в группах распространения не передаются в маркере безопасности, поэтому составэтих групп не влияет на размер маркера.

2. Задание на лабораторную работу:

Задания 1. Создать и настроить учетные записи**:**

1. Изучить теоретический материал по данной теме.
2. Создать и настроить учетные записи трех пользователей системы.
3. Создать группу пользователей, в которую включить учетные записи новых пользователей.
4. Ограничить срок действия первой учетной записи пользователя до определенной даты, например, до указанной даты и разрешить ему вход в систему по понедельникам и четвергам с 10.00 до 17.00.
5. Вход в систему второго пользователя задать в остальные дни недели с 10.00 до 17.00.
6. Вход в систему третьему пользователю задать в будние дни с 8.00 до 10.00.
7. Данные по каждому пользователю вставить в отчет.
8. Установить безопасный вход в систему.

Задания 2. Изучить особенности созданной учётной записи**:**

1. Создать учётную запись пользователя.
2. Изучить свойства созданной учётной записи.
3. Создать группу безопасности и группу распространения, включить учётную запись пользователя в эти группы.
4. Создать шаблон учетной записи.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы:

1. Что представляет собой учетная запись пользователя?
2. Какие типы учетных записей вам известны?
3. Какую учетную запись рекомендуется использовать после окончания настройки компьютера для повседневной работы?
4. Для чего предназначена учетная запись «Гость»?
5. Какая учетная запись позволяет настраивать ПК и устанавливать любые программы?
6. Для чего создаются группы пользователей?

Лабораторная работа № 7
Модель безопасности ОС Windows

Цель работы: изучение порядка активного аудита информационной безопасности компьютерных систем

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2, ОС Kali Linux, hping3, nmap, ScanOval

1.Теоретический материал

Защита конфиденциальных данных от несанкционированного доступа является важнейшим фактором успешного функционирования любой многопользовательской системы. ОС Windows не является исключением и требования к защите объектов файловой системы, памяти, объектов ядра операционной системы внесли существенный вклад в процесс ее проектирования и реализации.

Так, например, версии Windows NT/2000 были сертифицированы по классу C2 критериев TSSEC («Оранжевая книга»). Требования к операционной системе, защищенной по классу C2, включают:

– обязательную идентификацию и аутентификацию всех пользователей операционной системы. Под этим понимается способность операционной системы идентифицировать всех пользователей, которые получают санкционированный доступ к системе, и предоставление доступа к ресурсам только этим пользователям;

– разграничительный контроль доступа – предоставление пользователям возможности защиты принадлежащих им данных;

– системный аудит – способность системы вести подробный аудит всех действий, выполняемых пользователями и самой операционной системой;

– защита объектов от повторного использования – способность системы предотвратить доступ пользователя к информации ресурсов, с которыми до этого работал другой пользователь.

ОС Microsoft Windows XP Professional (SP2/SP3) имеет действующие сертификаты ФСТЭК России и может использоваться в составе автоматизированных систем до класса защищенности 1Г включительно в соответствии с требованиями руководящего документа «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требований по защите информации» (Гостехкомиссия России, 1992).

Все действующие в системе объекты (пользователи, группы, локальные компьютеры, домены) идентифицируются в Windows не по именам, уникальность которых не всегда удается достичь, а по **идентификаторам защиты (Security Identifiers, SID)**. SID представляет собой числовое значение переменной

S – R – I – S0 - S1 - … - Sn – RID

длины:

S – неизменный идентификатор строкового представления SID;

R – уровень ревизии (версия). На сегодня 1.

I – (identifier-authority) идентификатор полномочий. Представляет собой 48-битную строку, идентифицирующую компьютер или сеть, который(ая) выдал SID объекту. Возможные значения:

– 0 (SECURITY_NULL_SID_AUTHORITY) – используются для сравнений, когда неизвестны полномочия идентификатора;

– 1 (SECURITY_WORLD_SID_AUTHORITY) – применяются для конструирования идентификаторов SID, которые представляют всех пользователей. Например, идентификатор SID для группы *Everyone* (Все пользователи) – это *S-1-1-0*;

– 2 (SECURITY_LOCAL_SID_AUTHORITY) – используются для построения идентификаторов SID, представляющих пользователей, которые входят на локальный терминал;

– 5 (SECURITY_NT_AUTHORITY) – сама операционная система. То есть, данный идентификатор выпущен компьютером или доменом.

Sn – 32-битные коды (колчеством 0 и более) субагентов, которым было передано право выдать SID. Значение первых подчиненных полномочий общеизвестно. Они могут иметь значение:

– 5 – идентификаторы SID присваиваются сеансам регистрации для выдачи прав любому приложению, запускаемому во время определенного сеанса регистрации. У таких идентификаторов SID первые подчиненные полномочия установлены как 5 и принимают форму *S-1-5-5-x-y*;

– 6 – когда процесс регистрируется как служба, он получает специальный идентификатор SID в свой маркер для обозначения данного действия. Этот идентификатор SID имеет подчиненные полномочия 6 и всегда будет *S-1-5-6*;

– 21 (SECURITY_NT_NON_UNIQUE) – обозначают идентификатор SID пользователя и идентификатор SID компьютера, которые не являются уникальными в глобальном масштабе;

– 32 (SECURITY_BUILTIN_DOMAIN_RID) – обозначают встроенные идентификаторы SID. Например, известный идентификатор SID для встроенной группы администраторов *S-1-5-32-544*;

– 80 (SECURITY_SERVICE_ID_BASE_RID) – обозначают идентификатор SID, который принадлежит службе.

Остальные подчиненные полномочия идентификатора совместно обозначают домен или компьютер, который издал идентификатор SID.

RID – 32-битный относительный идентификатор. Он является является идентификатором уникального объекта безопасности в области, для которой был определен SID. Например, 500 – обозначает встроенную учетную запись *Administrator*, 501 – обозначает встроенную учетную запись *Guest, a* 502 – RID для билета на получение билетов протокола Kerberos .

При генерации SID Windows использует генератор случайных чисел, чтобы обеспечить уникальность SID для каждого пользователя. Для некоторого произвольного пользователя SID может выглядеть так:

S-1-5-21-789336058-484763869-725345543-1003

Предопределенным пользователям и группам Windows выдает характерные SID, состоящие из SID компьютера или домена и предопределенного RID. В таблице 2 приведен перечень некоторых общеизвестных SID.

Таблица 2. Общеизвестные SID Windows

SID	Название	Описание
S-1-1-0	Все	Группа, в которую входят все пользователи
S-1-5-2	Сеть	Группа, в которую входят все пользователи, зарегистрировавшиеся в системе из сети
S-1-5-7	Анонимный вход	Группа, в которую входят все пользователи, вошедшие в систему анонимно
S-1-5-*домен*-500	Администратор	Учетная запись администратора системы. По умолчанию только эта запись обеспечивает полный контроль системы
S-1-5-*домен*-501	Гость	Учетная запись пользователя-гостя

Полный список общеизвестных SID можно посмотреть в документации Platform SDK. Узнать SID конкретного пользователя в системе, а также SID групп, в которые он включен, можно, используя консольную команду **whoami**:

whoami /user /sid

Соответствие имени пользователя и его SID можно отследить также в ключе реестра **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ ProfileList.**

После аутентификации пользователя процессом Winlogon, все процессы, запущенные от имени этого пользователя будут идентифицироваться специальным объектом, называемым **маркером доступа (access token)**. Если процесс пользователя запускает дочерний процесс, то его маркер наследуются, поэтому маркер доступа олицетворяет пользователя для системы в каждом запущенном от его имени процессе. Основные элементы маркера представлены на рисунке 5.

SID пользователя	SID1 … SIDn Идентификаторы групп пользователя	DACL по умолчанию	Привилегии	Прочие параметры

Рисунок 5. Обобщенная структура маркера доступа

Маркер доступа содержит идентификатор доступа самого пользователя и всех групп, в которые он включен. В маркер включен также DACL по умолчанию – первоначальный список прав доступа, который присоединяется к создаваемым пользователем объектам. Еще одна важная для определения прав пользователя в системе часть маркера – список его привилегий. Привилегии – это права доверенного объекта на совершение каких-либо действий по отношению ко всей системе. В таблице 3 перечислены некоторые привилегии, которые могут быть предоставлены пользователю.

Таблица 3. Привилегии, которыми могут быть наделены пользователи

Имя и идентификатор привилегии	Описание привилегии
Увеличение приоритета диспетчирования SeIncreaseBasePriorityPrivilege	Пользователь, обладающий данной привилегией может изменять приоритет диспетчирования процесса с помощью интерфейса Диспетчера задач
Закрепление страниц в памяти SeLockMemoryPrivilege	Процесс получает возможность хранить данные в физической памяти, не прибегая к кэшированию данных в виртуальной памяти на диске
Управление аудитом и журналом безопасности SeAuditPrivilege	Пользователь получает возможность указывать параметры аудита доступа к объекту для отдельных ресурсов, таких как файлы, объекты Active Directory и разделы реестра
Овладение файлами или иными объектами SeTakeOwnershipPrivilege	Пользователь получает возможность становиться владельцем любых объектов безопасности системы, включая объекты Active Directory, файлы и папки NTFS, принтеры, разделы реестра, службы, процессы и потоки
Завершение работы системы SeShutdownPrivilege	Пользователь получает возможность завершать работу операционной системы на локальном компьютере
Обход перекрестной проверки SeChangeNotifyPrivilege	Используется для обхода проверки разрешений для промежуточных каталогов при проходе многоуровневых каталогов

Управление привилегиями пользователей осуществляется в оснастке **«Групповая политика»**, раздел **Конфигурация Windows/Локальные политики/Назначение прав пользователя**.

Чтобы посмотреть привилегии пользователя, можно также использовать команду

whoami /all

Остальные параметры маркера носят информационный характер и определяют, например, какая подсистема создала маркер, уникальный идентификатор маркера, время его действия. Необходимо также отметить возможность создания ограниченных маркеров (restricted token), которые отличаются от обычных тем, что из них удаляются некоторые привилегии и его SID-идетификаторы проверяются только на запрещающие правила. Создать ограниченный маркер можно программно, используя API-функцию **CreateRestrictedToken**, а можно запустить процесс с ограниченным маркером, используя пункт контекстного меню Windows **«Запуск от имени…»** и отметив пункт **«Защитить компьютер от несанкционированных действий этой программы»** (рисунок 6).

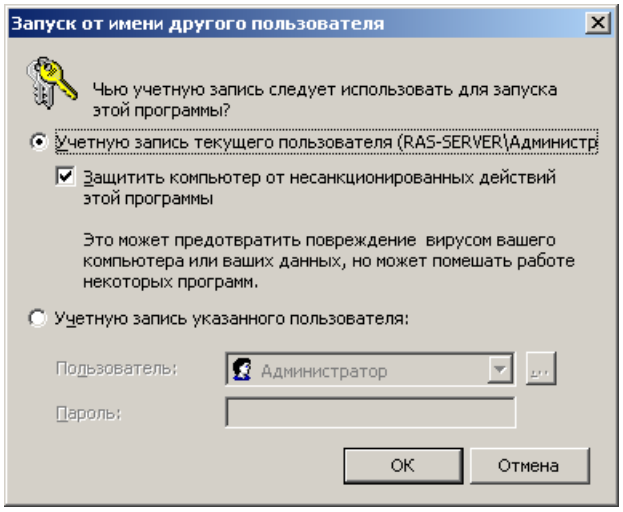

Рисунок 6. Запуск процесса с ограниченным маркером

Ограниченные маркеры используются для процессов, подменяющих клиента и выполняющих небезопасный код.

Маркер доступа может быть создан не только при первоначальном входе пользователя в систему. Windows предоставляет возможность запуска процессов от имени других пользователей, создавая для этих процессов соответствующий маркер. Для этих целей можно использовать:

– API-функции **CreateProcessAsUser, CreateProcessWithLogon**;

– оконный интерфейс (рисунок 6), инициализирующийся при выборе пункта контекстного меню **«Запуск от имени…»**;

– консольную команду **runas**:

runas /user:*имя_пользователя program* ,

где *имя_пользователя* – имя учетной записи пользователя, которая будет использована для запуска программы в формате *пользователь@домен* или *домен\пользователь*;

program – команда или программа, которая будет запущена с помощью учетной записи, указанной в параметре **/user.**

В любом варианте запуска процесса от имени другой учетной записи необходимо задать ее пароль.

Защита объектов системы

Маркер доступа идентифицирует субъектов-пользователей системы. С другой стороны, каждый объект системы, требующий защиты, содержит описание прав доступа к нему пользователей. Для этих целей используется **дескриптор безопасности (Security Descriptor, SD)**. Каждому объекту системы, включая файлы, принтеры, сетевые службы, контейнеры Active Directory и другие, присваивается дескриптор безопасности, который определяет права доступа к объекту и содержит следующие основные атрибуты (рисунок 7):

– SID владельца, идентифицирующий учетную запись пользователя-владельца объекта;

– пользовательский список управления доступом (Discretionary Access Control List, DACL), который позволяет отслеживать права и ограничения, установленные владельцем данного объекта. DACL может быть изменен пользователем, который указан как текущий владелец объекта.

– системный список управления доступом (System Access Control List, SACL), определяющий перечень действий над объектом, подлежащих аудиту;

– флаги, задающие атрибуты объекта.

Авторизация Windows основана на сопоставлении маркера доступа субъекта с дескриптором безопасности объекта. Управляя свойствами объекта, администраторы могут устанавливать разрешения, назначать право владения и отслеживать доступ пользователей.

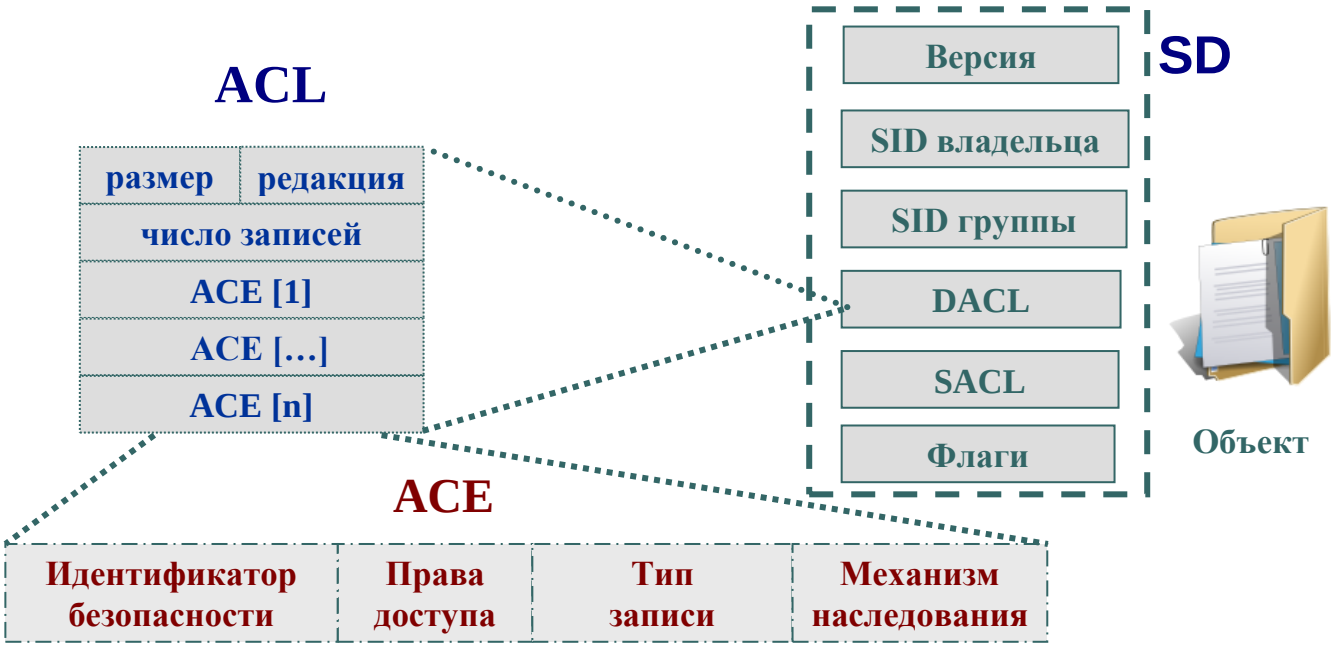

Рисунок 7. Структура дескриптора безопасности объекта Windows

Список управления доступом содержит набор элементов (Access Control Entries, ACE). В DACL каждый ACE состоит из четырех частей: в первой указываются пользователи или группы, к которым относится данная запись, во второй – права доступа, а третья информирует о том, предоставляются эти права или отбираются. Четвертая часть представляет собой набор флагов, определяющих, как данная запись будет наследоваться вложенными объектами (актуально, например, для папок файловой системы, разделов реестра).

Если список ACE в DACL пуст, к нему нет доступа ни у одного пользователя (только у владельца на изменение DACL). Если отсутствует сам DACL в SD объекта – полный доступ к нему имеют все пользователи.

Если какой-либо поток запросил доступ к объекту, подсистема SRM осуществляет проверку прав пользователя, запустившего поток, на данный объект, просматривая его список DACL. Проверка осуществляется до появления разрешающих прав **на все** запрошенные операции. Если встретится запрещающее правило хотя бы **на одну** запрошенную операцию, доступ не будет предоставлен.

Рассмотрим пример на рисунке 8. Процесс пытается получить доступ к объекту с заданным DACL. В маркере процесса указаны SID запустившего его пользователя, а также SID групп, в которые он входит. В списке DACL объекта присутствуют разрешающие правила на чтение для пользователя с SID=100, и на запись для группы с SID=205. Однако, в доступе пользователю будет отказано, поскольку раньше встречается запрещающее запись правило для группы с SID=201.

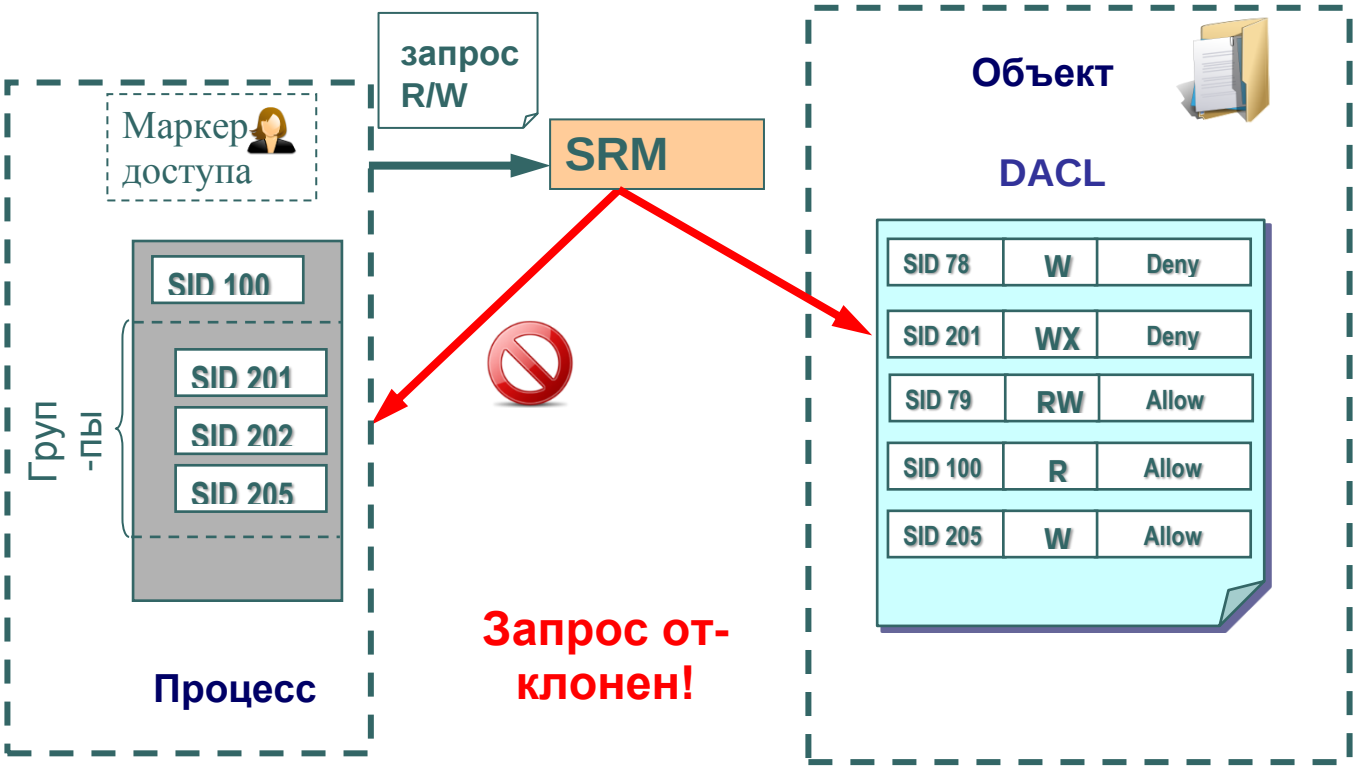

Рисунок 8. Проверка прав доступа пользователя к объекту

Необходимо отметить, что запрещающее правило помещено в списке DACL на рисунке не случайно. Запрещающие правила **всегда** размещаются перед разрешающими, то есть являются доминирующими при проверке прав доступа.

Для определения и просмотра прав доступа пользователей к ресурсам можно использовать как графические средства контроля, так и консольные команды. Стандартное окно свойств объекта файловой системы (диска, папки, файла) на вкладке **Безопасность** (рисунок 9) позволяет просмотреть текущие разрешения для пользователей и групп пользователей, редактировать их, создавать новые или удалять существующие.

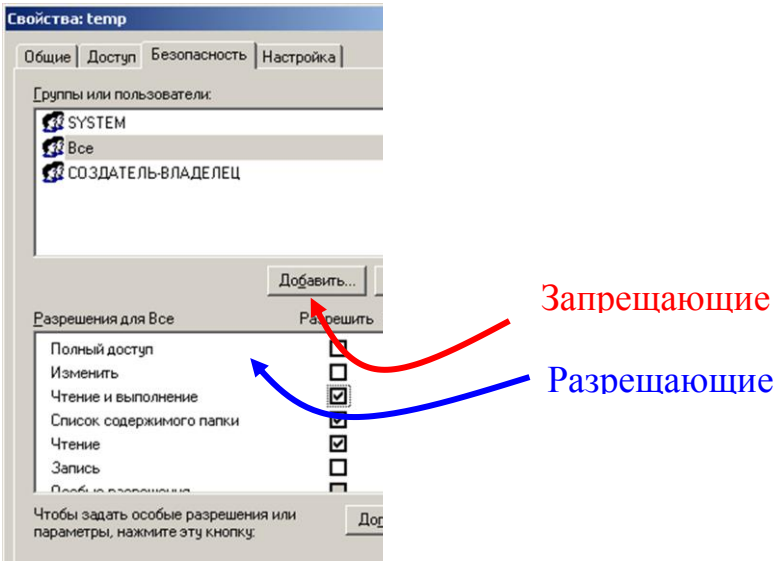

Рисунок 9. GUI-интерфейс Windows для изменения прав доступа к объектам

При определении прав доступа к объектам можно задать правила их наследования в дочерних контейнерах. В окне дополнительных параметров безопасности на вкладке **Разрешения** при выборе опции «**Наследовать от родительского объекта применимых к дочерним объектам разрешения, добавляя их к явно заданным в этом окне**» можно унаследовать разрешения и ограничения, заданные для родительского контейнера, текущему объекту.

При выборе опции **«Заменить разрешения для всех дочерних объектов заданными здесь разрешениями, применимыми к дочерним объектам»** разрешается передача определенных для объекта-контейнера правил доступа его дочерним объектам.

В этом же окне на вкладке **Владелец** допустимо узнать владельца объекта и заменить его. Владелец объекта имеет право на изменение списка его DACL, даже если к нему запрещен любой тип доступа. Администратор имеет право становиться владельцем любого объекта.

С учетом возможности вхождения пользователя в различные группы и независимости определения прав доступа к объектам для групп и пользователей, зачастую бывает сложно определить конечные права пользователя на доступ к объекту: требуется просмотреть запрещающие правила, определенные для самого объекта, для всех групп, в которые он включен, затем то же проделать для разрешающих правил. Автоматизировать процесс определения разрешенных пользователю видов доступа к объекту можно с использованием вкладки **«Действующие разрешения»** окна дополнительных параметров безопасности объекта.

Для просмотра и изменения прав доступа к объектам в режиме командной строки предназначена команда **cacls** (**icacls** в Windows Vista и Widows 7).

cacls *имя_файла* [**/t**] [**/e**] [**/c**] [**/g** *пользователь*:*разрешение*] [**/r** *пользователь* [...]] [**/p** *пользователь*:*разрешение* [...]] [**/d** *пользователь* [...]]

Назначения параметров команды приведены в таблице 4.

Таблица 4. Параметры команды cacls

<имя файла>	Задаёт файл или папку, права доступа к которой необходимо просмотреть/изменить (допустимо использовать шаблоны с символами ***** и **?**).
/t	Изменение избирательных таблиц контроля доступа (DACL) указанных файлов в текущем каталоге и всех подкаталогах
/e	Редактирование избирательной таблицы управления доступом (DACL) вместо ее замены
/c	Заставляет команду продолжить изменение прав доступа при возникновении ошибки, связанной с нарушениями прав доступа
/g <пользователь | группа: разрешение>	Предоставление прав доступа указанному пользователю
/r <пользователь | группа>	Отнимает права доступа указанного пользователя.
/p <пользователь | группа: разрешение>	Заменяет права доступа указанного пользователя
/d <пользователь | группа>	Отказывает в праве доступа указанному пользователю или группе

Для указания добавляемых или отнимаемых прав используются следующие значения:

F – полный доступ;
C – изменение (запись);
W – запись;
R – чтение;
N – нет доступа.

Рассмотрим несколько примеров.

cacls d:\test

Выдаст список DACL для папки test.

cacls d:\test /d ИмяКомпьютера\ИмяПользователя /e

Запретит доступ к объекту для указанного пользователя.

cacls d:\test /p ИмяКомпьютера\ИмяГруппы:f /e /t

Предоставит полный доступ к папке d:\test и ее подпапках всем для членов указанной группы.

Для программного просмотра и изменения списков DACL можно использовать API-функции **AddAccessAllowedAce, AddAccessDeniedAce,**

SetSecurityInfo. Подробнее с этими функциями и примерами их использования можно ознакомиться в [пособие].

Подсистема аудита.

Важный элемент политики безопасности – аудит событий в системе. ОС Windows ведет аудит событий по 9 категориям:

1. Аудит событий входа в систему.
2. Аудит управления учетными записями.
3. Аудит доступа к службе каталогов.
4. Аудит входа в систему.
5. Аудит доступа к объектам.
6. Аудит изменения политики.
7. Аудит использования привилегий.
8. Аудит отслеживания процессов.
9. Аудит системных событий.

Рассмотрим более подробно, какие события отслеживает каждая из категорий.

Аудит событий входа в систему

Аудит попыток пользователя войти в систему с другого компьютера или выйти из нее, при условии, что этот компьютер используется для проверки подлинности учетной записи.

Аудит управления учетными записями

Аудит событий, связанных с управлением учетными записями на компьютере: создание, изменение или удаление учетной записи пользователя или группы; переименование, отключение или включение учетной записи пользователя; задание или изменение пароля.

Аудит доступа к службе каталогов

Аудит событий доступа пользователя к объекту каталога Active Directory, для которого задана собственная системная таблица управления доступом (SACL).

Аудит входа в систему

Аудит попыток пользователя войти в систему с компьютера или выйти из нее.

Аудит доступа к объектам

Аудит событий доступа пользователя к объекту – например, к файлу, папке, разделу реестра, принтеру и т. п., – для которого задана собственная системная таблица управления доступом (SACL).

Аудит изменения политики

Аудит фактов изменения политик назначения прав пользователей, политик аудита или политик доверительных отношений.

Аудит использования привилегий

Аудит попыток пользователя воспользоваться предоставленным ему правом.

Аудит отслеживания процессов

Аудиту таких событий, как активизация программы, завершение процесса, повторение дескрипторов и косвенный доступ к объекту.

Аудит системных событий

Аудит событий перезагрузки или отключения компьютера, а также событий, влияющих на системную безопасность или на журнал безопасности.

Решения об аудите конкретного типа событий безопасности принимаются в соответствии с политикой аудита локальной системы. Политика аудита, также называемая локальной политикой безопасности (local security policy), является частью политики безопасности, поддерживаемой LSASS в локальной системе, и настраивается с помощью редактора локальной политики безопасности (Оснастка **gpedit.msc, Конфигурация компьютера – Конфигурация Windows – Параметры безопасности – Локальные политики – Политика аудита,** рисунок 8).

Для каждого объекта в SD содержится список SACL, состоящий из записей ACE, регламентирующих запись в журнал аудита удачных или неудачных попыток доступа к объекту. Эти ACE определяют, какие операции, выполняемые над объектами конкретными пользователями или группами, подлежат аудиту. Информация аудита хранится в системном журнале аудита. Аудиту могут подлежать как успешные, так и неудачные операции. Подобно записям ACE DACL, правила аудита объектов могут наследоваться дочерними объектами. Процедура наследования определяются набором флагов, являющихся частью структуры ACE.

Настройка списка SACL может быть осуществлена в окне дополнительных свойств объекта (**пункт «Дополнительно», закладка «Аудит»**).

Для программного просмотра и изменения списков SACL можно использовать API-функции **GetSecutityInfo** и **SetSecutityInfo**.При инициализации системы и изменении политики LSASS посылает SRM сообщения, информирующие его о текущей политике аудита. LSASS отвечает за прием записей аудита, генерируемых на основе событий аудита от SRM, их редактирование и передачу Event Logger (регистратору событий). SRM посылает записи аудита LSASS через свое LPC-соединение. После этого Event Logger заносит записи в журнал безопасности.

События аудита записываются в журналы следующих типов:

1. **Журнал приложений.** В журнале приложений содержатся данные, относящиеся к работе приложений и программ.
2. **Журнал безопасности.** Журнал безопасности содержит записи о таких событиях, как успешные и безуспешные попытки доступа в систему, а также о событиях, относящихся к использованию ресурсов.
3. **Журнал системы.** В журнале системы содержатся события системных компонентов Windows. Например, в журнале системы регистрируются сбои при загрузке драйвера или других системных компонентов при запуске системы.

4. **Журнал службы каталогов.** В журнале службы каталогов содержатся события, заносимые службой каталогов Windows (на контроллере домена AD).

5. **Журнал службы репликации.** В журнале службы репликации файлов содержатся события, заносимые службой репликации файлов Windows (на контроллере домена AD).

Просмотр журнала безопасности осуществляется в оснастке «Просмотр событий» (**eventvwr.msc**). Сами журналы хранятся в файлах **SysEvent.evt, SecEvent.evt, AppEvent.evt** в папке **%WinDir%\system32\config.**

В журнал записываются события 3 основных видов:

1. Информационные сообщения о событиях.

Описывают успешное выполнение операций, таких как запуск или некоторое действие системной службы.

2. Предупреждающие сообщения о событиях.

Описывают неожиданные действия, означающие проблему, или указывают на проблему, которая возникнет в будущем, если не будет устранена сейчас .

3. Сообщения о событиях ошибок.

Описывают ошибки, возникшие из-за неудачного выполнения задач.

Шифрующая файловая система

Начиная с версии Windows 2000, в операционных системах семейства Windows NT поддерживается шифрование данных на разделах файловой системы NTFS с использованием *шифрующей файловой системы* (**Encrypted File System, EFS**). Основное ее достоинство заключается в обеспечении конфиденциальности данных на дисках компьютера за счет использования надежных симметричных алгоритмов для шифрования данных в реальном режиме времени.

Для шифрации данных EFS использует симметричный алгоритм шифрования (AES или DESX) со случайным ключом для каждого файла (**File Encryption Key, FEK**). По умолчанию данные шифруются в Windows 2000 и Windows XP по алгоритму DESX, а в Windows XP с Service Pack 1 (или выше) и Windows Server 2003 – по алгоритму AES. В версиях Windows, разрешенных к экспорту за пределы США, драйвер EFS реализует 56-битный ключ шифрования DESX, тогда как в версии, подлежащей использованию только в США, и в версиях с пакетом для 128-битного шифрования длина ключа DESX равна 128 битам. Алгоритм AES в Windows использует 256-битные ключи.

При этом для обеспечения секретности самого ключа FEK шифруется асимметричным алгоритмом RSA открытым ключом пользователя, результат шифрации FEK – **Data Decryption Field, DDF** – добавляется в заголовок зашифрованного файла. Такой подход обеспечивает надежное шифрование без потери эффективности процесса шифрования: данные шифруются быстрым симметричным алгоритмом, а для гарантии секретности симметричного ключа используется асимметричный алгоритм шифрования.

Для шифрации файлов с использованием EFS можно использовать графический интерфейс или команду **cipher**.

Графический интерфейс доступен в стандартном окне свойств объекта по нажатию кнопки **«Дополнительно»**. Зашифрованные объекты в стандартном интерфейсе Windows Explorer отображаются зеленым цветом.

Необходимо отметить, что EFS позволяет разделять зашифрованный файл между несколькими пользователями. В этом случае FEK шифруется открытыми ключами всех пользователей, которым разрешен доступ к файлу, и каждый результат шифрации добавляется в DDF.

Шифрование файла с использованием EFS защищает файл комплексно: пользователю, не имеющему права на дешифрацию файла, недопустимы, в том числе, такие операции, как удаление, переименование и копирование файла. Необходимо помнить, что EFS является частью файловой системы NTFS, и в случае копирования защищенного файла авторизованным пользователем на другой том с файловой системой, на поддерживающей EFS (например, FAT32), он будет дешифрован и сохранен на целевом томе в открытом виде.

Консольная команда **cipher** может быть использована для шифрации/дешифрации файлов из командной строки или в bat-сценарии.

cipher [{**/e**|**/d**}] [**/s:***каталог*] [**/a**] [**/i**] [**/f**] [**/q**] [**/h**] [**/k**] [**/u**[**/n**]] [*путь* [...]] | [**/r:***имя_файла_без_расширения*]

Назначения параметров команды приведены в таблице 5.

Таблица 5. Параметры команды **cipher**

/e	Шифрует указанные папки. Папки помечаются таким образом, чтобы файлы, которые будут добавляться в папку позже, также шифровались.
/d	Расшифровывает указанные папки. Папки помечаются таким образом, чтобы файлы, которые будут добавляться в папку позже, не будут шифроваться
/s: ***каталог***	Выполняет выбранную операцию над указанной папкой и всеми подпапками в ней.
/a	Выполняет операцию над файлами и каталогами
/i	Продолжение выполнения указанной операции даже после возникновения ошибок. По умолчанию выполнение **cipher** прекращается после возникновения ошибки
/f	Выполнение повторного шифрования или расшифровывания указанных объектов. По умолчанию уже зашифрованные или расшифрованные файлы пропускаются командой **cipher**
/k	Создание ключа шифрования файла для пользователя, выполнившего команду **cipher**. Если используется данный параметр, все остальные параметры команды **cipher** не учитываются.
/u	Обновление ключа шифрования файла пользователя или ключа агента восстановления на текущие ключи во всех за-

	шифрованных файлах на локальном диске (если эти ключи были изменены). Этот параметр используется только вместе с параметром **/n**.
/n	Запрещение обновления ключей. Данный параметр служит для поиска всех зашифрованных файлов на локальных дисках. Этот параметр используется только вместе с параметром **/u**.
путь	Указывает шаблон, файл или папку.
/r: ***имя_файла***	Создание нового сертификата агента восстановления и закрытого ключа с последующей их записью в файлах с именем, указанным в параметре *имя_файла* (без расширения). Если используется данный параметр, все остальные параметры команды **cipher** не учитываются.

Например, чтобы определить, зашифрована ли какая-либо папка, необходимо использовать команду:

cipher ***путь\имя_папки***

Команда **cipher** без параметров выводит статус (зашифрован или нет) для всех объектов текущей папки.

Для шифрации файла необходимо использовать команду

cipher /e /a ***путь\имя_файла***

Для дешифрации файла, соответственно, используется команда

cipher /d /a ***путь\имя_файла***

Допустима шифрация/дешифрация группы файлов по шаблону:

cipher /e /a d:\work*.doc

Пара открытый и закрытый ключ для шифрации FEK создаются для пользователя автоматически при первой шифрации файла с использованием EFS.

Если некоторый пользователь или группа пользователей зашифровали файл с использованием EFS, то его содержимое доступно только им. Это приводит к рискам утери доступа к данным в зашифрованных файлах в случае утраты пароля данным пользователем (работник забыл пароль, уволился и т. п.). Для предотвращения подобных проблем администратор может определить некоторые учетные записи в качестве агентов восстановления.

Агенты восстановления (Recovery Agents) определяются в политике безопасности **Encrypted Data Recovery Agents (Агенты восстановления шифрованных данных)** на локальном компьютере или в домене. Эта политика доступна через оснастку **Групповая политика (gpedit.msc)** раздел **«Параметры безопасности»-> «Политика открытого ключа»-> «Файловая система EFS».** Пункт меню **«Действие»-> «Добавить агент восстановления** данных» открывает мастер добавления нового агента.

Добавляя агентов восстановления можно указать, какие криптографические пары (обозначенные их сертификатами) могут использовать эти агенты для восстановления шифрованных данных (рисунок 10). Сертификаты для агентов восстановления создаются командой **cipher** с ключом **/r** (см. таблицу 5). Для пользователя, который будет агентом восстановления, необходимо импортировать закрытый ключ агента восстановления из сертификата, созданного командой **cipher**. Это можно сделать в мастере импорта сертификатов, который автоматически загружается при двойном щелчке по файлу *.pfx.

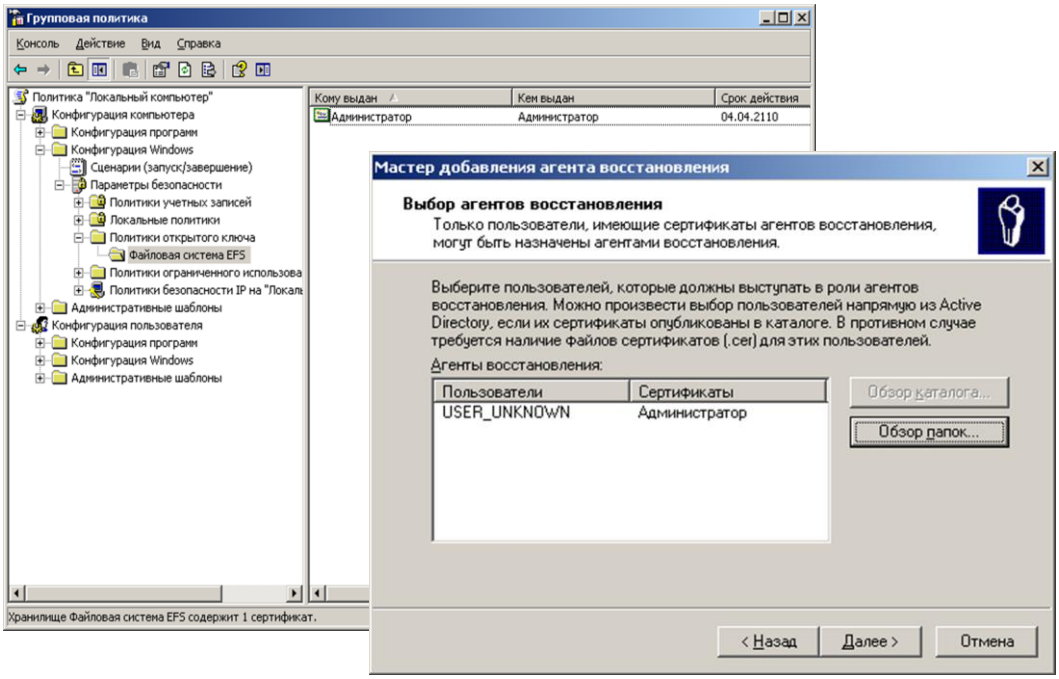

Рисунок 10. Добавление нового агента восстановления EFS

EFS создает – DRF **(Data Recovery Field)**-элементы ключей для каждого агента восстановления, используя провайдер криптографических сервисов, зарегистрированный для EFS-восстановления. DRF добавляется в зашифрованный файл и может быть использован как альтернативное средство извлечения FEK для дешифрации содержимого файла.

Windows хранит закрытые ключи в подкаталоге **Application Data\Microsoft\Crypto\RSA** каталога профиля пользователя. Для защиты закрытых ключей Windows шифрует все файлы в папке RSA на основе симметричного ключа, генерируемого случайным образом; такой ключ называется мастер-ключом пользователя. Мастер-ключ имеет длину в 64 байта и создается стойким генератором случайных чисел. Мастер-ключ также хранится в профиле пользователя в каталоге Application Data\Microsoft\Protect и зашифровывается по алгоритму 3DES с помощью ключа, который отчасти основан на пароле пользователя. Когда пользователь меняет свой пароль, мастер-ключи автоматически расшифровываются, а затем заново зашифровываются с учетом нового пароля.

Для расшифровки FEK EFS использует функции Microsoft CryptoAPl (CAPI). CryptoAPI состоит из DLL провайдеров криптографических сервисов (cryptographic service providers, CSP), которые обеспечивают приложениям доступ к различным криптографическим сервисам (шифрованию, дешифрованию и хэшированию). EFS опирается на алгоритмы шифрования RSA, предоставляемые провайдером **Microsoft Enhanced Cryptographic Provider (\Windows\ System32\Rsaenh.dll).**

Шифрацию и дешифрацию файлов можно осуществлять программно, используя API-функции **EncryptFile** и **DecryptFile**.

Задание на лабораторную работу

1. Ознакомьтесь с теоретическими основами защиты информации в ОС семейства Windows в настоящих указаниях и конспектах лекций.

2. Выполните задания 2.1–2.8

2.1. Запустите в программе **Oracle VM Virtualbox** виртуальную машину WinXP. Войдите в систему под учетной записью администратора, пароль узнайте у преподавателя. Все действия в пп 2.1–2.8 выполняйте в системе, работающей на виртуальной машине.

2.2. Создайте учетную запись нового пользователя **testUser** в оснастке **«Управление компьютером» (compmgmt.msc).** При создании новой учетной записи запретите пользователю смену пароля и снимите ограничение на срок действия его пароля. Создайте новую группу **«testGroup»** и включите в нее нового пользователя. Удалите пользователя из других групп. Создайте на диске **C:** папку **forTesting**. Создайте или скопируйте в эту папку несколько текстовых файлов (*.txt).

2.3. С помощью команды **runas** запустите сеанс командной строки (**cmd.exe**) от имени вновь созданного пользователя. Командой **whoami** посмотрите SID пользователя и всех его групп, а также текущие привилегии пользователя. Строку запуска и результат работы этой и ***всех*** следующих консольных команд копируйте в файл протокола лабораторной работы.

2.4. Убедитесь в соответствии имени пользователя и полученного SID в реестре Windows. Найдите в реестре, какому пользователю в системе присвоен SID **S-1-5-21-1957994488-492894223-170857768-1004** (Используйте ключ реестра **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList**).

2.5. Командой **whoami** определите перечень текущих привилегий пользователя **testUser**. В сеансе командной строки пользователя попробуйте изменить системное время командой **time**. Чтобы предоставить пользователю подобную привилегию, запустите оснастку **«Локальные параметры безопасности» (secpol.msc).** Добавьте пользователя в список параметров политики **«Изменение системного времени»** раздела **Локальные политики -> Назначение прав пользователя**. После этого перезапустите сеанс командной строки от имени пользователя, убедитесь, что в списке привилегий добавилась **SeSystemtime-Priviege**. Попробуйте изменить системное время командой **time**.

Убедитесь, что привилегия **«Завершение работы системы» (SeShutdownPrivilege)** предоставлена пользователю testUser . После этого попробуйте завершить работу системы из сеанса командной строки пользователя командой **shutdown –s**. Добавть ему привелегию **«Принудительное удаленное завершение» (SeRemoteShutdownPrivilege)**. Попробуйте завершить работу консольной командой еще раз (отменить команду завершения до ее непосредственного выполнения можно командой **shutdown –a**).

2.6. Ознакомьтесь с справкой по консольной команде **cacls.** Используя эту команду, просмотрите разрешения на папку **c:\forTesting**. Объясните все обозначения в описаниях прав пользователей и групп в выдаче команды.

а) Разрешите пользователю **testUse**r запись в папку **forTesting**, но запретите запись для группы **testGroup**. Попробуйте записать файлы или папки в **forTesting** от имени пользователя **testUser**. Объясните результат. Посмотрите эффективные разрешения пользователя **testUser** к папке **forTesting** в окне свойств папки.

б) Используя стандартное окно свойств папки, задайте для пользователя **testUser** такие права доступа к папке, чтобы он мог записывать информацию в папку **forTesting**, но не мог просматривать ее содержимое. Проверьте, что папка **forTesting** является теперь для пользователя **testUser** «слепой», запустив, например, от его имени файловый менеджер и попробовав записать файлы в папку, просмотреть ее содержимое, удалить файл из папки.

в) Для вложенной папки **forTesting\Docs** отмените наследование ACL от родителя и разрешите пользователю просмотр, чтение и запись в папку. Проверьте, что для пользователя папка **forTesting\Docs** перестала быть «слепой» (например, сделайте ее текущей в сеансе работы файлового менеджера от имени пользователя и создайте в ней новый файл).

г) Снимите запрет на чтение папки **forTesting** для пользователя **testUser**. Используя команду **cacls** запретите этому пользователю доступ к файлам с расширением txt в папке **forTesting**. Убедитесь в недоступности файлов для пользователя.

д) Командой **cacls** запретите пользователю все права на доступ к папке **forTesting** и разреште полный доступ к вложенной папке **forTesting\Docs**. Убедитесь в доступности папки **forTesting\Docs** для пользователя. Удалите у пользователя **testUser** привилегию **SeChangeNotifyPrivilege**. Попробуйте получить доступ к папке **forTesting\Docs**. Объясните результат.

е) Запустите файловый менеджер от имени пользователя **testUser** и создайте в нем папку **newFolder** на диске С. Для папки **newFolder** очистите весь список ACL командой **cacls**. Попробуйте теперь получить доступ к папке от имени администратора и от имени пользователя. Кто и как теперь может вернуть доступ к папке? Верните полный доступ к папке для всех пользователей.

ж) Создайте в разделе **HKLM\Software** реестра раздел **testKey**. Запретите пользователю **testUser** создание новых разделов в этом разделе реестра. Создайте для раздела **HKLM\Software\testKey** SACL, позволяющий протоколировать отказы при создании новых подразделов, а также успехи при перечисле-

нии подразделов и запросе значений (предварительно проверьте, что в локальной политике безопасности соответствующий тип аудита включен). Попробуйте от имени пользователя **testUser** запустить **regedit.exe** и создать раздел в **HKLM\Software**. Убедитесь, что записи аудита были размещены в журнале безопасности (**eventvwr.msc**).

2.7. Шифрование файлов и папок средствами EFS.

а) От имени пользователя **testUser** зашифруйте какой-нибудь файл на диске. Убедитесь, что после этого был создан сертификат пользователя, запустив оснастку **certmgr.msc** от имени пользователя (раздел **Личные**). Просмотрите основные параметры сертификата открытого ключа пользователя **testUser** (срок действия, используемые алгоритмы). Установите доверие к этому сертификату в вашей системе.

б) Создайте в папке **forTesting** новую папку **Encrypt**. В папке **Encrypt** создайте или скопируйте в нее текстовый файл. Зашифруйте папку **Encrypt** и все ее содержимое из меню свойств папки от имени администратора. Попробуйте просмотреть или скопировать какой-нибудь файл этой папки от имени пользователя **testUser**. Объясните результат. Скопируйте зашифрованный файл в незашифрованную папку (например, **forTesting**). Убедитесь что он остался зашифрованным. Добавьте пользователя **testUser** в список имеющих доступа к файлу пользователей в окне свойств шифрования файла. Повторите попытку получить доступ к файлу от имени пользователя **testUser**.

в) Создайте учетную запись нового пользователя **agentUser**, сделайте его членом группы Администраторы. Определите для пользователя **agentUser** роль агента восстановления EFS. Создайте в папке **forTesting** новый текстовый файл с произвольным содержимым. Зашифруйте этот файл от имени пользователя **testUser**. Убедитесь в окне подробностей шифрования файла, что пользователь **agentUser** является агентом восстановления для данного файла. Попробуйте прочитать содержимое файла от имени администратора и от имени пользователя **agentUser**. Объясните результат.

г) Зашифруйте все текстовые файлы папки **forTesting** с использованием консольной команды шифрования **cipher** от имени пользователя **testUser (**предварительно снимите запрет на доступ к этим файлам, установленный в задании 2.6, г**).**

д) Убедитесь, что при копировании зашифрованных файлов на том с файловой системой, не поддерживающей EFS (например, FAT32 на флешнакопителе), содержимое файла дешифруется.

2.8. После демонстрации результатов работы преподавателю восстановите исходное состояние системы: удалите созданные папки и файлы, разделы реестра, удалите учетную запись созданного пользователя и его группы, снимите с пользователя **agentUse**r роль агента восстановления.

2.9. Представьте отчёт по лабораторной работе преподавателю и отчитайте работу.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы

1. К какому классу безопасности относится ОС Windows по различным критериям оценки?
2. Каким образом пользователи идентифицируются в ОС Windows?
3. Что такое списки DACL и SACL?
4. Перечислите, каким образом можно запустить процесс от имени другого пользователя.
5. Как происходит проверка прав доступа пользователя к ресурсам в ОС Windows?
6. Что такое маркер безопасности, и какова его роль в модели безопасности Windows?
7. Как с использованием команды cacls добавить права на запись для всех файлов заданной папки?
8. Какие события подлежат аудиту в ОС Windows?
9. Каким образом шифруются файлы в файловой системе EFS? Что такое FEK? DDF? DDR?
10. Какие алгоритмы шифрования используются в EFS?

Лабораторная работа № 8
Создание и управление доменной политикой

Цель работы: Изучить средства для создания доменной политики безопасности в ОС Windows Server.

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2, ОС Kali Linux, hping3, nmap, ScanOval

1.Теоретический материал

Для защиты данных Windows использует следующие основные механизмы: аутентификация и авторизация пользователей, аудит событий в системе, шифрование данных, поддержка инфраструктуры открытых ключей, встроенные средства сетевой защиты. Эти механизмы поддерживаются такими подсистемами Windows как LSASS (Local Security Authority Subsystem Service, подсистема локальной аутентификации), SAM (Security Account Manager, диспетчер локальных записей безопасности), SRM (Security reference Monitor, монитор состояния защиты), Active Directory (служба каталогов), EFS (Encrypting File System, шифрующая файловая система) и др.

Защита объектов и аудит действий с ними в ОС Windows организованы на основе избирательного (дискреционного) доступа, когда права доступа (чтение, запись, удаление, изменение атрибутов) субъекта к объекту задается явно в специальной матрице доступа. Для укрупнения матрицы пользователи могут объединяться в группы. При попытке субъекта (одного из потоков процесса, запущенного от его имени) получить доступ к объекту указываются, какие операции пользователь собирается выполнять с объектом. Если подобный тип доступа разрешен, поток получает описатель (дескриптор) объекта и все потоки процесса могут выполнять операции с ним. Подобная схема доступа, очевидно, требует аутентификации каждого пользователя, получающего доступ к ресурсам и его надежную идентификацию в системе, а также механизмов описания прав пользователей и групп пользователей в системе, описания и проверки дискреционных прав доступа пользователей к объектам. Рассмотрим, как в ОС Windows организована аутентификация и авторизация пользователей.

2.Задание на лабораторную работу

Задание 1. Создание доменной учетной записи

Создайте доменные учетные записи, перечисленные в рисунке 11.

First Name (Имя)	Last Name (Фамилия)	User Logon Name (Имя входа пользователя)	Password (Пароль)	Change Password (Изменить пароль)
User	One	User1	(пустой)	Must
User	Three	User3	(пустой)	Must
User	Five	User5	User5	Must
User	Seven	User7	User7	Must
User	Nine	User9	User9	Cannot

Рисунок 11. Доменные учетные записи для лабораторной работы

Выполнив следующее задание, вы создадите первую учетную запись с помощью консоли Active Directory Users and Computers. Далее повторите те же действия для со щания

остальных учетных записей.

Создайте доменную учетную запись

1. Зарегистрируйтесь как Administrator (Администратор).

2. Раскройте меню Start\Programs\Administrative Tools (Пуск\Программы\Администрирование) и щелкните Active Directory Users and Computers (Active Directory – пользователи и компьютеры). Откроется одноименная консоль.

3. Раскройте узел microsoft.com (если вы используете другое имя домена, раскройте свой домен) и дважды щелкните папку Users Какие учетные записи мастер установки Active Directory создал по умолчанию?

4. Щелкните правой кнопкой мыши папку Users и выберите в контекстном меню команду New\User (Создать\Пользователь). Откроется окно New object – User.Где в Active Directory будет создана новая учетная запись?

5. В поле First Name введите **User.**

6. В поле Last Name введите One.

Заметьте: поле Full Name заполняется автоматически.

7. В поле User Logon введите **userl.**

8. В списке справа от окна User Logon выберите @microsoft.com (имя домена может отличаться, если вы не использовали microsoft.com в качестве доменного имени DNS). Имя входа пользователя в сочетании с доменным именем, появляющимся в окне справа от окна User Logon Name, – это полное имя входа пользователя в Интернете. Это имя уникально определяет пользователя в каталоге (например, userl@microsofi.com).

Заметьте: поле имени входа для предыдущих версий Windows заполняется автоматически.

В каких случаях используется имя входа предыдущих версий Windows?

9. Щелкните Next, чтобы продолжить.Windows 2000 отобразит окно New Object – User, предлагая ввести параметры пароля и ограничения.

10. В полях Password и Confirm Password введите пароль или оставьте эти поля пустыми, если вы не присваиваете пароль. Если вы вводите пароль, обра-

тите внимание, что на экране его символы заменяются звездочками (*), дабы посторонние не подсмотрели ваш пароль.

11. Определите, может ли пользователь изменять свой пароль. Каковы результаты одновременного применения флажков User Must Change Password At Next Logon и User Cannot Change Password? Поясните ответ. В каком случае следует выбрать флажок Account is Disabled при создании новой учетной записи?

12. После задания параметров пароля щелкните Next.

Откроется окно New Object – User, содержащее параметры, сконфигурированные для этой учетной записи.

13. Проверьте правильность параметров и щелкните кнопку Finish (Готово). Примечание Если настройки учетной записи оказались неверными, щелкните кнопку Back (Назад), чтобы изменить их. Заметьте: на правой панели консоли Active Directory Users and Computers появилась вновь созданная учетная запись.

14. Повторите пункты 4–13 для остальных учетных записей.

Задание 2: Администрирование учетных записей

Подключение учетной записи

1. Войдите в домен как Administrator (Администратор).
2. Откройте консоль Active Directory Users and Computers.
3. Раскройте домен microsoft.com и щелкните Users.
4. На правой панели щелкните правой кнопкой учетную запись Profile User, созданную ранее, и в контекстном меню выберите команду Disable Account (Отключить учетную запись).

Active Directory сообщит, что учетная запись была отключена. Учетная запись также помечена красным крестом.

5. Щелкните ОК, чтобы вернуться в консоль Active Directory Users and Computers.
6. На правой панели консоли Active Directory Users and Computers щелкните правой кнопкой мыши учетную запись пользователя, которую только что отключили, чтобы появилось контекстное меню.
7. Завершите сеанс Windows 2000.
8. Попытайтесь войти в систему как puser.

Дополнительно:

1. Зарегистрируйтесь в домене как Administrator (Администратор).
2. Запустите консоль Active Directory Users and Computers.
3. Раскройте домен Microsoft.com и щелкните Users.
4. На правой панели щелкните правой кнопкой мыши созданную вами учетную запись.
5. Profile User и в контекстном меню выберите команду Enable Account (Включить учетную запись).
6. Active Directory сообщит, что учетная запись подключена.

7. Щелкните ОК, чтобы вернуться в консоль Active Directory Users and Computers.

8. На правой панели консоли Active Directory Users and Computers щелкните правой кнопкой мыши учетную запись пользователя, которую только что включили, чтобы появилось контекстное меню.

Протестируйте включение учетной записи и измените ее пароль

1. Войдите в систему как puser.

2. Измените пароль на **student.**

3. Завершите сеанс Windows 2000.

Восстановление пароля для учетной записи

1. Войдите в домен как Administrator (Администратор).

2. Запустите консоль Active Directory Users and Computers.

3. Раскройте домен microsoft.com и щелкните Users.

4. На правой панели щелкните правой кнопкой учетную запись Profile User и в контекстном меню выберите команду Reset Password (Смена пароля).

Откроется одноименное окно, содержащее поле для ввода нового пароля для этой учетной записи. Заметьте: Administrator не может узнать текущий пароль.

5. В полях New Password (Новый пароль) и Confirm Password (Подтверждение) введите **password** и пометьте флажок User Must Change Password At Next Logon (Потребовать

смену пароля при следующем входе в систему). Щелкните ОК.

Active Directory сообщит, что пароль изменен.

6. Щелкните ОК, чтобы вернуться в консоль Active Directory Users and Computers

7. Завершите сеанс.

Протестируйте смену пароля

1. Войдите в систему как puser с паролем **password,**

Удалось ли это? Почему?

2. Завершите сеанс.

Задание3: Планирование новых учетных записей групп

Предположим, вы – администратор отдела по обслуживанию клиентов производственной компании и управляете доменом, входящим в дерево доменов организации. Администрированием других доменов вы не занимаетесь, однако вам может потребоваться предоставить некоторым пользователям других доменов доступ к ресурсам вашего домена. Пользователи компании работают с несколькими разделяемыми сетевыми ресурсами. Компания также планирует развернуть программу электронной почты, использующую Active Directory.

Как администратору, вам требуется определить:

– необходимые группы;

– состав каждой группы. Это могут быть как учетные записи пользователей, так и другие группы;

– тип и область действия каждой группы.

Зафиксируйте разработанные вами стратегии в тетради ≪Планирование групп≫. При

заполнении тетради укажите:

– названия всех групп в колонке ≪Имя группы≫;

– тип и область действия группы;

– состав группы.

Назначьте разрешение на доступ к папке

1. Зарегистрируйтесь в системе как Administrator и запустите Windows Explorer (Проводник).

2. Откройте локальный диск C;\.

3. Щелкните правой кнопкой мыши значок папки, для которой требуется изменить разрешения, и выберите в контекстном меню команду Properties (Свойства). Откроется окно свойств данной папки с выбранной вкладкой General (Обшие).

4. Перейдите на вкладку Security (Безопасность).

5. Если нужно изменить унаследованные разрешения для пользователя, учетной записи или группы, сбросьте флажок Allow Inheritable Permissions From Parent To Propagate To This Object (Переносить наследуемые от родительского объекта разрешения на этот объект) и, когда появится соответствующий запрос, щелкните кнопку Copy (Копировать).

6. Чтобы добавить учетным записям или группами разрешения для данной папки, щелкните кнопку Add (Добавить).

Откроется диалоговое окно Select Users, Computers, Or Groups (Выбор: Пользователи, Компьютеры или Группы).

7. Убедитесь, что в списке Look In (Искать в) в верхней части окна выбран ваш домен.

8. В списке Name (Имя) выберите имя требуемой учетной записи или группы, сверив-шись со сценарием, и щелкните кнопку Add (Добавить).

Учетная запись или группа появится в списке Name (Имя).

9. Повторите пункт 8 для каждой учетной записи или группы, перечисленной для данной папки в сценарии.

10. Щелкните ОК, чтобы вернуться к диалоговому окну свойств.

11. Если окно Properties (Свойства) содержит учетные записи и группы, не перечисленные в сценарии, выделите их и щелкните кнопку Remove (Удалить).

12. Для всех учетных записей и групп, перечисленных для данной папки в предыдущем сценарии, выберите в списке Name учетную запись или группу, затем в списке разрешений пометьте флажок Allow (Разрешить) или Deny (Отменить) согласно требованиям сценария.

13. Щелкните ОК, чтобы сохранить изменения и закрыть диалоговое окно свойств.

14. Повторите эту процедуру для каждой папки, которой назначаете разрешения (см. сценарий).

15. Завершите рабочий сеанс. проверка **разрешений NTFS**

Сейчас вы зарегистрируетесь в системе по разным учетным записям и проверите разрешения NTFS.

Задание 4: Проверьте разрешения на доступ к папке Reports пользователя UserSI

1. Зарегистрируйтесь в системе как UserSI и запустите Windows Explorer (Проводник).

2. Откройте папку C:\Data\Managers\Reports.

3. Попробуйте создать файл в папке Reports.

Удалось ли это? Почему?

4. Закройте Windows Explorer (Проводник) и завершите рабочий сеанс.

Проверьте разрешение на доступ к папке Reports пользователя User82

1. Зарегистрируйтесь в системе как User82 и запустите Windows Explorer (Проводник).

2. Откройте папку C:\Data\Managers\Reports.

3. Попробуйте создать файл в папке Reports.

Удалось ли это? Почему?

4. Завершите рабочий сеанс.

Для закрепления: Создание ОП и их объектов

Создайте ОП и три учетные записи пользователей, которые будут применяться в дальнейшем.

Задание 1: создайте экземпляры ОП и объектов User

Создайте два ОП и три объекта Lser.

1. Зарегистрируйтесь на ServerOl как Administrator (Администратор) с паролем password.

2. Откройте оснастку Active Directory Users And Computers (Active Directory – пользователи и компьютеры).

Для уверенности в правильном размещения нового ОП сначала выберите место размещения.

3. В дереве консоли щелкните microsou.com.

4. В меню Action (Действие) выберите New (Создать), а затем – команду Organizational Unit (Подразделение).

Откроется диалоговое окно New Object – Organizational Unit (Новый объект – Подразделение).

Заметьте: в имени заключена только требуемая информация. В диалоговом окне будет представлено местоположение, где будет создан объект. Это должно быть microsoft.com/.

5. В поле Name (Имя) введите Sales и щелкните кнопку ОК.

В дереве консоли появится ОП с именем Sales.

6. В microsoft.com создайте также еше одно ОП с именем Servers.

7. В дереве консоли щелкните Users.

8. В меню Action (Действие) выберите New (Создать), а затем – команду User (Пользователь).

Диалоговое окно New Object – User (Новый объект – Пользователь) сообщает, что новая учетная запись пользователя будет создана в ОП microsoft.com/Users.

9. Создайте новую учетную запись пользователя с параметрами:

Поле Значение

First name (Имя) Jane

Last name (Фамилия) Doe

User logon name (Имя входа пользователя) Jane_Doe

10. Щелкните кнопку Next (Далее).

11. Оставьте поля пароля незаполненными и не изменяйте стандартные параметры этой учетной записи. Щелкните кнопку Next (Далее).

Откроется итоговое окно, представляющее полное и регистрационное имя для пользователя Jane Doe.

12. Щелкните кнопку Finish (Готово).

13. На правой панели оснастки Active Directory Users And Computers щелкните объект Jane_Doe.

14. В меню Action (Действие) выберите команду Properties (Свойства).

Откроется диалоговое окно свойств этого объекта.

15. На вкладке General (Обшие) диалогового окна свойств в поле Telephone Number (Номер телефона) наберите 555-1234.

16. Щелкните кнопку ОК.

17. Создайте учетные записи пользователей с параметрами:

a) Поле Значение
b) First name (Имя) John
c) Last name (Фамилия) Smith
d) User logon name (Имя входа пользователя) John_Smith
e) Поле Значение
f) First name (Имя) Bob
g) Last name (Фамилия) Train
h) User logon name (Имя входа пользователя) Bob_Train

Задание 5: Управление объектами Active Directory

Сначала найдите объект User, созданный в предыдущем упражнении, а затем переместите его в другое место.

1. Зарегистрируйтесь на ServerOl как Administrator с паролем password.

2. Откройте оснастку Active Directory Users And Computers (Active Directory – пользователи и компьютеры).

3. В дереве консоли щелкните microsoft.com.

4. В меню Action (Действие) выберите команду Find (Найти).

Откроется окно Find Users, Conlacts, And Groups (Поиск: пользователи, контакты и группы).

5. Убедитесь, что в списке Find (Найти) выбрано Users, Contacts, And Groups (Пользователи, контакты и группы) и щелкните кнопку Find Now (Найти).

Заметьте: объекты User и Group обнаруживаются независимо от их местоположения.

6. Щелкните кнопку Clear All (Очистить все), а затем – кнопку ОК чтобы очистить панель результатов поиска.

7. Убедитесь, что в списке In (в) значится домен microsoft.

8. В поле Name (Имя) введите Jane.

9. Перейдите на вкладку Advanced (Дополнительно).

10. Щелкните кнопку Field (Поле), выберите User (Пользователь) и затем щелкните Telephone Number (Номер телефона).

Примечание Если в списке не видно пункта Telephone Number, прокрутите список вниз.

11. В поле View (Значение) наберите 555-12 и щелкните кнопку Add (Добавить).

12. В меню View (Вид) выберите команду Choose Columns (Выбрать столбцы).

Откроется диалоговое окно Choose Columns (Выбор столбцов).

13. В списке Columns Shown (Отображаемые столбцы) щелкните пункт Description, а затем – кнопку Remove (Удалить).

14. Прокрутите список Columns Available (Доступные столбцы), выберите X500 Distinguished Name (X500 различающееся имя) и щелкните кнопку Add (Добавить).

15. Щелкните кнопку (Ж, чтобы закрыть диалоговое окно Choose Columns.

В диалоговом окне Find Users, Contacts And Groups отображаются параметры найденного пользователя Jane Doe с типом объекта User (Пользователь) и различающимся именем CN=Jane Doe, CN=Users, DC=microsoft, DC=com.

Различающееся имя указывает, что пользователь Jane Doe находится в контейнере Users домена microsoft.com.

16. Закройте диалоговое окно Find Users, Contacts And Groups.

Переместите объект пользователя Jane Doe из контейнера Users в контейнер Sales.

1. В дереве консоли оснастки Active Directory Users And Computers (Active Directory пользователи и компьютеры) щелкните Users.

На правой панели появятся все объекты с типом User (Пользователь) и Security Group

(Группа безопасности).

2. На правой панели щелкните объект пользователя Jane Doe.

3. В меню Action (Действие) выберите команду Move (Переместить).

Откроется одноименное окно.

4. Выделите ОП Sales и щелкните кнопку ОК.

Пользователь Jane Doe переместится из ОП Users в ОП Sales.

5. В дереве консоли щелкните ОП Sales.

На правой панели появится объект пользователя Jane Doe.

6. Закройте оснастку Active Directory Users And Computers.

Задание 6: Получение файла во владение, определите разрешения для файла

1. Зарегистрируйтесь в домене как Administrator и запустите Windows Explorer (Проводник).

2. В папке C:\Data (где C:\ – имя вашего системного диска) создайте текстовый файл с именем OWNER.

3. Щелкните правой кнопкой мыши файл OWNER.TXT и выберите команду Properties (Свойства).

Откроется окно свойств файла с активной вкладкой General (Общие).

4. Перейдите на вкладку Security (Безопасность), чтобы увидеть разрешения для файла

OWNER.TXT.

Каковы текущие"разрешения для OWNER.TXT?

5. Щелкните кнопку Advanced (Дополнительно).

Откроется диалоговое окно Access Control Settings For OWNER.TXT (Параметры управления доступом для OWNER.TXT) с активной вкладкой Permissions (Разрешения).

6. Перейдите на вкладку Owner (Владелец).

Кто является текущим владельцем файла OWNER.TXT?

1. В диалоговом окне Access Control Settings For OWNER.TXT (Параметры управления доступом для OWNER.TXT) перейдите на вкладку Permissions (Разрешения).

2. Щелкните кнопку Advanced (Дополнительно).

Откроется диалоговое окно Select User, Computer, Or Group (Выбор: Пользователи, Компьютеры или Группы).

3. В списке Look In (Искать в) выберите ваш домен.

4. В списке Name (Имя) выберите User83, затем щелкните ОК.

Откроется диалоговое окно Permission Entry For OWNER.TXT.

Обратите внимание, что нет ни одного разрешения для User84.

5. В списке разрешений пометьте флажок Allow (Разрешить) у разрешения Take Ownership.

6. Щелкните ОК.

Откроется окно Access Control Settings For OWNER.TXT (Параметры управления доступом для OWNER.TXT) с выбранной вкладкой Permissions (Разрешения).

7. Щелкните ОК, чтобы вернуться в диалоговое окно свойств.

8. Щелкните ОК, чтобы сохранить изменения и закрыть диалоговое окно свойств OWNER.

TXT.

9. Закройте все приложения и завершите сеанс работы с Windows 2000.

Станьте владельцем файла

1. Зарегистрируйтесь в системе как User83 и запустите Windows Explorer (Проводник).

2. Откройте папку C:\Data.

3. Щелкните файл OWNER.TXT правой кнопкой мыши и выберите в контекстном меню команду Properties (Свойства).

Откроется окно свойств с активной вкладкой General (Общие).

4. Перейдите на вкладку Security (Безопасность) и изучите разрешения для файла OWNER.

TXT.

Появится сообщение, что вы можете лишь просматривать текущую информацию о защите файла OWNER.TXT.

5. Щелкните ОК.

Откроется диалоговое окно свойств с выбранной вкладкой Security (Безопасность).

6. Щелкните кнопку Advanced (Дополнительно), чтобы открыть диалоговое окно Access

Control Settings For OWNER.TXT {Параметры управления доступом для OWNER.TXT), и перейдите на вкладку Owner (Владелец).

7. В списке Name (Имя) выберите User83 и щелкните кнопку Apply (Применить).

Назовите текущего владельца файла OWNER.TXT.

8. Щелкните кнопку ОК, чтобы закрыть диалоговое окно Access Control Settings For

OWNER.TXT (Параметры управления доступом для OWNER.TXT).

Откроется диалоговое окно свойств OWNER.TXT с выбранной вкладкой Security (Безопасность).

9. Щелкните ОК, чтобы закрыть диалоговое окно Properties (Свойства).

Проверьте разрешения на доступ владельца к файлу

1. Зарегистрируйтесь в системе как User83. Назначьте пользователю User83 разрешение

Full Control для файла Owner.txt и щелкните кнопку Apply (Применить).

2. Сбросьте флажок Allow Inheritable Permissions From Parent To Propagate To This Object (Переносить наследуемые от родительского объекта разрешения на этот объект).

3. В диалоговом окне Security (Безопасность) щелкните кнопку Remove (Удалить), чтобы удалить разрешения групп Users (Пользователи) и Administrators (Администраторы) для файла OWNER.TXT.

4. Щелкните ОК, чтобы закрыть окно свойств файла OWNER.TXT.

5. Удалите файл OWNER.TXT.

6. Закройте все приложения.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы:

1. Каковы текущие"разрешения для OWNER.TXT?
2. Как стать текущим владельцем файла OWNER.TXT?
3. Назовите текущего владельца файла OWNER.TXT.

Лабораторная работа № 9
Конфигурирование доменной политики

Цель работы: Изучить конфигурирование доменной политики безопасности в ОС Windows Server.

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2

1.Теоретический материал

Зачастую, настройка локальной сети в операционных системах Windows Vista, Windows 7, Windows Server 2008/2008 R2 начинается с такой области конфигурирования сетевых свойств, как компонент «Центр управления сетями и общим доступом». При помощи данного средства конфигурирования сетей можно выбирать сетевое размещение, просматривать карту сети, настраивать сетевое обнаружение, общий доступ к файлам и принтерам, а также настраивать и просматривать состояние ваших текущих сетевых подключений.

Перед началом работы с данным компонентом, следует разобраться с таким понятием как сетевое расположение. Этот параметр задается для компьютеров при первом подключении к сети и во время подключения автоматически настраивается брандмауэр и параметры безопасности для того типа сети, к которому производится подключение. В отличие от операционной системы Windows Vista, где для всех сетевых подключений используется самый строгий профиль брандмауэра для сетевого размещения, операционная система Windows поддерживает несколько активных профилей, что позволяет наиболее безопасно использовать несколько сетевых адаптеров, подключенных к различным сетям. Существует четыре типа сетевого расположения

Также как и сетевые клиенты, сетевые службы являются компонентами операционной системы. Сетевые службы операционных систем Windows – это специальные процессы, которые создают прослушивающий сокет и привязывают его к определенному порту, обеспечивающие дополнительную функциональность для сетевых подключений. Системные службы запускаются операционной системой автоматически в процессе загрузки компьютера или по мере необходимости при выполнении стандартных операций. Понятное имя службы отображается в оснастке **«Службы»**, а настоящее имя службы используется в программах с интерфейсом командной строки. По умолчанию в операционных системах Microsoft ко всем локальным подключениям привязаны две сетевые службы:

2.Задание на лабораторную работу

Задание 1. Изменение свойств учетной записи пользователя домена (на закрепление)

Предполагается, что вы создали три учетные записи пользователей.

Задание 1: Измените параметры учетных записей пользователей

1. Зарегистрируйтесь на ServerOl как Administrator с паролем password.

2. Раскрыв меню $tart\Programs\Administrative Tools, щелкните ярлык Active Directory

Users And Computers. Откроется одноименная оснастка.

3. Раскройте в дереве консоли узел microsoft.com.

4. Выберите папку Users.

5. В правой панели дважды щелкните учетную запись Bob Train.

Откроется диалоговое окно свойств учетной записи с выбранной вкладкой General (Общие). На вкладке General, помимо имени и фамилии, определяются и другие свойства учетной записи. Поиск пользователей облегчают поля Office (Комната) и Telephone Number (Номер телефона).

6. На вкладке Account (Учетная запись) щелкните кнопку Logon Hours (Время входа).

Откроется диалоговое окно Logon Hours For Bob Train.

Заметьте, что пользователю Bob вход в систему разрешен в любое время.

7. Чтобы ограничить время входа в систему пользователя Bob, щелкните время начала первого периода, в течение которого Вы хотите запретить ему вход в систему, и перетащите указатель на время окончания этого периода. Для этого определите текущие день и время и запретите вход на ближайшие 3 ч.

8. Щелкните переключатель Logon Denied (Вход запретен).

Выделенный период изменит цвет на белый – пользователю запретен вход в систему в течение этого срока.

Совет: Чтобы выбрать такой же период времени для всех дней недели, щелкните в поле All (Все) серый квадрат, представляющий начало периода, и перетащите указатель на время окончания. Чтобы выбрать день полностью, щелкните серый квадрат с его названием

Чтобы закрыть диалоговое окно Logon Hours For Bob Train, щелкните кнопку ОК.

10. Примените параметры, шелкнув ОК в диалоговом окне Bob Train Properties.

11. На правой панели дважды щелкните John Smith.

Откроется диалоговое окно свойств учетной записи John Smith с выбранной вкладкой General (Общие).

12. Перейдите на вкладку Account (Учетная запись).

13. Когда окончится срок действия данной учетной записи?

14. В группе Account expires (Срок действия учетной записи) щелкните переключатель End

Of (Истекает) и задайте текущую дату.

15. Примените внесенные изменения, щелкнув кнопку ОК.

16. Щелкните папку Sales в дереве консоли.

В правой панели появится учетная запись Jane Doe.

17. Дважды щелкните учетную запись Jane Doe.

Откроется окно Jane Doe Properties с выбранной вкладкой General.

18. Перейдите на вкладку Account.

19. В списке Account Options (Параметры учетной записи) пометьте флажок User Must Change Password At Next Logon (Потребовать смену пароля при следующем входе в систему).

20. Закройте окно Jane Doe Properties (Свойства: Jane Doe), щелкнув кнопку ОК.

21. Закройте оснастку Active Directoiy Users And Computers.

22. В меню Start выберите команду Shut Down (Завершение работы).

Откроется диалоговое окно Shut Down Windows (Завершение работы Windows).

23. Выбрав в списке Log Off Administrator (Завершение сеанса Администратор), щелкните ОК.

Windows 2000 завершит сеанс пользователя Administrator и выведет на экран окно сообщения Welcome To Windows.

Попытайтесь войти на ServerOl с учетной записью пользователя

Вы используете для входа на ServerO I учетную запись Jane Doe (Jane_Doe).

1. Войдите в систему без пароля как Jane_Doe.

Появится сообщение, что срок действия Вашего пароля закончился и его следует сменить.

2. Щелкните кнопку ОК. Откроется диалоговое окно Change Password с курсором в поле Old Password.

3. Поскольку учетной записи пользователя Jane_Doe не присвоен пароль, нажмите клавишу Tab.

4. В полях New Password и Confirm New Password, введите student и щелкните ОК. Появится сообщение, что Ваш пароль изменился.

5. Закройте окно сообщения, щелкнув кнопку ОК.

Вошли ли Вы в систему? Почему?

6. Закройте окно сообщения, щелкнув кнопку ОК.

Предоставьт е учетным записям пользователей права локального входа в систему

Разрешить пользователям локальную регистрацию на контроллере домена можно по-разному. Вы добавите 3-х пользователей, созданных в главе 6, к группе Print Operators (Операторы печати), имеющей право входить на контроллер домена.

Зарегистрируйтесь в системе как Administrator с паролем password.

2. Откройте оснастку Active Directory Users And Computers и в дереве консоли раскройте ОП Sales.

3. В правой панели дважды щелкните учетную запись пользователя Jane Doe.

Откроется диалоговое окно Jane Doe Properties (Свойства: Jane Doe) с выбранной вкладкой General.

4. Перейдите на вкладку Member Of (Член групп).

5. Щелкните кнопку Add (Добавить).

Откроется диалоговое окно Select Groups (Выбор: Группа).

6. Щелкните Print Operators (Операторы печати).

7. Щелкните кнопку Add (Добавить), затем – ОК, чтобы закрыть окно Select Groups (Выбор: Группа).

8. Закройте диалоговое окно Jane Doe Properties (Свойства: Jane Doe), щелкнув ОК.

Далее Вы используете более простой способ добавить учетные записи для Bob Train и John Smith в группу Print Operators.

9. Щелкните папку Users в дереве консоли.

10. В правой панели щелкните Bob Train и, удерживая клавишу Ctrl, щелкните John Smith.

11. В меню Action (Действие) выберите команду Add Members To Group (Добавить участников в группу).

Откроется диалоговое окно Select Group.

12. Щелкните Print Operators (Операторы печати).

Active Directory сообщит об успешном добавлении пользователей в группу.

13. Щелкните кнопку ОК.

14. Закройте оснастку Active Directory Users And Computers и завершите свой сеанс.

15. Попытайтесь войти в систему как Jane_Doe с паролем student.

Заметьте: Вы можете локально войти в систему по учетной записи Jane_Doe.

16. Попытайтесь войти в систему как Bob_Train без пароля.

Вы не можете войти в систему из-за ограничения учетной записи пользователя. В задании 1 Вы ограничили время входа в систему пользователя Bob.

17. Попытайтесь войти в систему как John_Smith без пароля.

Вам разрешен вход в систему по учетной записи John_Smith, В задании 1 Вы ограничили срок действия учетной записи этого пользователя до конца дня. Завтра войти в систему с его реквизитами будет нельзя,

18. Завершите свой сеанс на Server.

Задание 2: Создание RUP и назначение домашней папки. (на закрепление)

Вы создадите профиль, применив учетную запись. Затем, чтобы создать локальный профиль для учетной записи, Вы войдете в систему как владелец учетной записи. Затем Вы войдете в систему как Administrator и с помощью приложения System (Система) в Control Panel убедитесь, что нужный профиль создан. В задании 2 с помощью этого профиля Вы создадите и проверите RUP со второго компьютера.

Создайте шаблон профиля пользователя

Вы определите и проверите локальный профиль пользователя. На ServerOl Вы создадите шаблон профиля пользователя. Обычно для создания

шаблона применяется компьютер с Windows, но здесь предполагается, что задания выполняются на Windows Server.

1. Если Вы зарегистрированы на ServerOlKaic Administrator, завершите сеанс.

2. Войдите в домен microsoft.com как Jane_Doe с паролем student.

Если при этом Вы использовали учетную запись Jane_Doe первый раз, создается стандартный локальный профиль. Он будет перенастроен и назначен другим пользователям.

3. Дважды щелкните значок My Computer на рабочем столе. Откроется одноименное окно.

4. Перетащите значок Local Disk (C:) [Локальный диск (C:)] на рабочий стол.

Вы увидите сообщение о том, что данный элемент не может быть скопирован или перемещен, но для него можно создать ярлык.

5. Щелкните кнопку Yes (Да) для создания^ ярлыка для диска C:.

6. Закройте окно My Computer (Мой компьютер).

7. Раскройте меню Start\Settings (Пуск\Настройка) и щелкните ярлык Control Panel. В открывшемся окне дважды щелкните значок Display (Экран). Откроется диалоговое окно Display Properties (Свойства: Экран).

8. Перейдите на вкладку Appearance (Оформление).

Обратите внимание на текущую цветовую схему.

9. В списке Scheme (Схема) выберите другую схему и щелкните ОК.

Рабочий стол изменится в соответствии с новой цветовой схемой.

10. Закройте окно Control Panel.

11. Завершив сеанс Jane_Doe, войдите снова как Administrator с паролем password.

12. Раскрыв меню Start\Settings (Пуск\Настройка), щелкните ярлык Control Panel. В открывшемся окне дважды щелкните значок System (Система).

13. Перейдите на вкладку User Profiles (Профили пользователей).

Заметьте: на ServerOl несколько профилей. Они представляют все учетные записи пользователей на ServerOl.

14. Не закрывайте приложение System (Система) – оно Вам еще потребуется.

Определите и назначьте обязательный RUP

Из профиля пользователя Jane_Doe Вы создадите RUP и назначите его учетной записи John Smith. Выполняйте все действия на ServerOl. Чтобы проверить RUP> можете войти в систему с Server02.

Этот этап предполагает, что Вы знаете, как создать папку и открыть к ней доступ (см. занятие 1 главы 5).

1. Создайте на диске C:\ папку с именем Profiles.

2. Создайте обший ресурс с именем Profiles для палки C:\Profiles.

3. Откройте папку Profiles и создайте подпапку Shared.

Закройте окно Profiles.

4. Найдите диалоговое окно System Properties (Свойства системы). Приложение System было открыто на предыдущем задании.

5. Перейдите на вкладку User Profiles (Профили пользователей).

6. В списке Profiles Stored On This Computer (Профили, хранящиеся на этом компьютере) выберите MICROSOFT\Jane_Doe.

7. Щелкните кнопку Copy To (Копировать).

Откроется диалоговое окно Copy To (Копирование профиля).

8. В поле Copy Profile to (Копировать профиль на) наберите \\server01\profiles\shared.

9. Щелкните кнопку Change (Изменить).

Откроется диалоговое окно Select User Or Group (Выбор: Пользователь или Группа).

10. В столбце Name (Имя) щелкните Users (Пользователи) и затем — кнопку ОК.

В группе Permitted To Use (Разрешить использование) появится надпись BUILTIN\Users.

11. Щелкните кнопку ОК для возврата в окно System Properties.

Вы увидите сообщение, что папка \\server01\profiles\shared уже существует и текущее содержимое будет удалено. Такое сообщение появилось потому, что папку для этого профиля Вы уже создали.

12. Щелкните кнопку Yes (Да).

13. Щелкните кнопку ОК для возврата в окно Control Panel.

14. Откройте оснастку Active Directory Users And Computers.

15. Раскройте узел microsoft.com и щелкните папку Users.

16. В правой панели дважды щелкните учетную запись пользователя John Smith. Откроется диалоговое окно ее свойств.

17. Ранее Вы установили срок действия учетной записи этого пользователя. Чтобы удалить этот срок действия, на вкладке Account (Учетная запись) щелкните переключатель Never в группе Account Expires (Срок действия учетной записи).

18. Перейдите на вкладку Profile.

19. В поле Profile path (Путь к профилю) наберите \\server01\profiles\shared и щелкните ОК.

Закройте оснастку Active Directory Users And Computers.

Так как Вы используете централизованный профиль, который должен быть назначен другим пользователям, сделайте его обязательным.

20. Дважды щелкните значок My Computer (Мой компьютер) на рабочем столе.

21. Дважды щелкните значок Local Disk (C:) [Локальный диск (C:)].

22. Дважды щелкните папку Profiles.

23. Дважды щелкните папку Shared.

Заметьте, что открылись папки профиля.

24. В меню Tools (Сервис) выберите команду Folder Options (Свойства папки).

Откроется одноименное диалоговое окно.

25. Перейдите на вкладку View (Вид).

26. Щелкните переключатель Select the Show Hidden Files And Folders (Показывать скрытые файлы и папки) и сбросьте флажок Hide File Extensions For Known File Types (Скрывать расширения для зарегистрированных типов файлов).

27. Щелкните кнопку ОК.

Откроется окно Shared, показывающее скрытые файлы и папки, включая файл Ntuser.dat.

28. Выберите файл Ntuser.dat.

29. В меню File (Файл) выберите команду Rename (Переименовать).

30. Измените расширение файла на .man и нажмите клавишу Enter.

31. Закройте окно Shared и окно Control Panel (Панель управления).

Завершите текущий сеанс и войдите в систему как John_Smith без пароля.

Откроется рабочий стол пользователя John_Smith. Убедитесь, что его цветовая палитра именно та, что Вы назначили шаблону профиля пользователя и что на рабочем столе появился ярлык диска С:.

33. Для проверки обязательного профиля удалите с рабочего стола ярлык Connect To The Internet (Подключение к Интернету).

34. Завершите текущий сеанс и войдите в систему как John_Smitri без пароля.

На рабочем столе появился ярлык Connect to the Internet. Это произошло потому, что Вы назначили учетной записи Jolin_Smith обязательный профиль.

Назначьте пользователя домашнюю папку

На этом этапе Вы назначите John_Smith домашнюю папку.

1. Завершите сеанс John_Smith и войдите как Administrator с паролем password.

2. Создайте на диске С: папку HomeDirs.

3. Сделайте папку HomeDirs общей

4. Откройте оснастку Active Directoiy User And Computers.

5. Открыв окно свойств учетной записи John_Smith, перейдите на вкладку Profile.

6. В разделе Home Folder (Домашняя папка) щелкните переключатель Connect (Подключить),

7. Проверьте, что справа от переключателя Connect в списке появился диск Z:.

8. В поле То (к) наберите \\serverOl\HomeDirs\%username% и щелкните кнопку ОК.

9. Закройте оснастку Active Directory Users And Computers.

10. Щелкните папку HomeDirs в Windows Explorer (Проводник).

11 . В меню File (Файл) выберите команду Properties (Свойства).

Откроется диалоговое окно свойств папки HomeDirs.

12. Перейдите на вкладку Security (Безопасность).

Заметьте, что группа Everyone (Все) имеет разрешение Full Control (Полный доступ) для этого каталога.

13. Щелкните кнопку Add (Добавить).

Откроется диалоговое окно Select Users, Computers, Or Groups (Выбор: Пользователи,

Компьютеры или Группы).

14. Выберите Users (Пользователи) и щелкните кнопку Add (Добавить).

15. Щелкните кнопку ОК.

Откроется диалоговое окно свойств палки HomeDirs, показывающее группы Everyone

(Все) и MICRQSOFT\Users. Убедитесь, что группе Users назначены права Read & Execute (Чтение и выполнение); List Folder Contents (Список содержимого папки) и Read (Чтение).

16. Сбросьте флажок Allow Inheritable Permissions From Parent To Propagate To This Object (Переносить наследуемые от родительского объекта разрешения на этот объект). Появится сообщение Security (Безопасность).

17. Прочитайте сообщение и щелкните кнопку Remove (Удалить).

Группа Everyone (Все) больше не имеет прав доступа к папке HomeDirs.

18. Щелкните кнопку Add (Добавить).

Откроется диалоговое окно Select Users, Computers, Or Groups.

19. Выберите группу Administrators и щелкните кнопку Add.

20. Щелкните кнопку ОК.

Откроется диалоговое окно свойств папки HomeDirs, показывающее группы MICRO-SOFT\Users и MICROSOFT\Administrators.

21. Выберите группу Administrators (Администраторы).

22. В списке Permissions (Разрешения) пометьте флажок Allow (Разрешить) в строке Full Control (Полный доступ).

Все флажки должны быть помечены.

23. Щелкните кнопку ОК.

24. Дважды щелкните папку HomeDirs.

25. Щелкните папку John_Smith.

26. В меню File (Файл) выберите команду Properties (Свойства).

Откроется диалоговое окно John Smith Properties.

27. Перейдите на вкладку Security (Безопасность).

Заметьте, что Administrators и John Smith получили полный контроль над этим каталогом. Эти произошло автоматически, когда Вы задали как домашнюю папкой для учетной записи John_Smith папку \\server01\HomeDirs\%username%.

28. Щелкните кнопку ОК и закройте Windows Explorer (Проводник).

29. Завершите сеанс Administrator и войдите снова как John_Smith без пароля.

30. Дважды щелкните значок My Computer (Мой компьютер).

Заметьте: появился новый значок сетевого диска Z:, указывающий на подпапку John^Smith папки \\server01\HomeDirs.

31. Закройте окно My Computer и выйдите из системы.

Задание 3: Изменение режима домена. Создание групп (на закрепление)

Измените режим своего домена с помощью оснастки Active Directory Users And Computers. Переведите домен из смешанного режима в основной

По умолчанию Windows 2000 Server работает в смешанном режиме. Чтобы задействовать все функции работы с группами, доступные в Windows 2000 Server, домен должен работать в основном режиме. Выполняйте упражнение на ServerOl.

1. Зарегистрируйтесь на ServerOl как Administrator (Администратор) с паролем password.

2. Откройте оснастку Active Directory Users And Computers.

3. В дереве консоли выберите свой домен и затем в меню Action (Действие) – команду Properties (Свойства).

Откроется диалоговое окно свойств microsoft.com.

В настоящий момент домен работает в смешанном режиме. Обратите внимание на предупреждение об изменении режима домена.

4. Щелкните кнопку Change Mode (Изменить режим).

Active Directory предупредит о невозможности отмены изменений.

Щелкните кнопку Yes (Да).

В окне свойств microsoft.com будет показано, что домен был переведен в основной режим.

5. Щелкните кнопку ОК, чтобы закрыть окно свойств microsoft.com.

Active Directory сообщит об успешном изменении режима домена и укажет, что репликация новой информации на другие контроллеры домена может занять более 15 минут.

6. Щелкните кнопку ОК.

7. Не закрывайте оснастку Active Directory Users And Computers

Проверьте разрешения на доступ к папке Reports пользователя UserSI

1. Зарегистрируйтесь в системе как UserSI и запустите Windows Explorer (Проводник).

2. Откройте папку C:\Data\Managers\Reports.

3. Попробуйте создать файл в папке Reports.

Удалось ли это? Почему?

4. Закройте Windows Explorer (Проводник) и завершите рабочий сеанс.

проверьте разрешение на доступ к папке Reports пользователя User82

1. Зарегистрируйтесь в системе как User82 и запустите Windows Explorer (Проводник).

2. Откройте папку C:\Data\Managers\Reports.

3. Попробуйте создать файл в папке Reports.

4. Завершите рабочий сеанс.

Создайте глобальную группу, добавьте участников и организуйте учетные записи пользователей

Создайте глобальную группу защиты, добавьте в нее членов и переместите пользователя из одного организационного подразделения (ОП) в другое.

1. Убедитесь, что оснастка Active Directory Users And Computers открыта и в фокусе.

2. В дереве консоли щелкните узел ОП Sales.

На правой панели появится учетная запись пользователя Jane Doe.

3. В меню Action выберите New (Создать), а затем – команду Group.

Откроется диалоговое окно New Object – Group (Новый объект – Группа).

Когда выбрана группа безопасности, доступна универсальная область действия. Это связано с тем, что служба каталогов Active Directory работает в основном режиме.

4. Убедитесь, что выбраны переключатели Global (Глобальная) и Security (Группа безопасности).

5. В поле Group Name (Имя группы) введите Sales и щелкните ОК.

На правой панели узла появится новая группа.

6. На правой панели дважды щелкните группу Sales.

Откроется диалоговое окно Sales Properties (Свойства: Sales).

7. Перейдите на вкладку Members (Члены группы).

8. Щелкните кнопку Add (Добавить).

Откроется диалоговое окно Select Users, Contacts, Computers, Or Groups (Выбор: Пользователи, Компьютеры, Контакты или Группы); в списке Look In (Искать в) будет выбрано microsoft.com.

9. В списке учетных записей, групп л компьютеров щелкните Jane_Doe и, удерживая клавишу Ctrl, щелкните John_Smith.

Будут выбраны обе учетные записи. Учетная запись Jane Doe находится в ОП microsoft .com/Sales, а John Smith — в ОП microsoft.com/Users.

10. Щелкните кнопку Add (Добавить).

Учетные записи Jane Doe и John Smith стали членами глобальной группы защиты Sales.

11. Щелкните кнопку ОК,

12. Снова щелкните кнопку ОК, чтобы закрыть диалоговое окно Sales Properties (Свой ства: Sales). В организационных целях Вы решили переместить учетную запись John Smith в ОП Sales.

13. Щелкните ОП Users.

14. На правой панели щелкните учетную запись John Smith

15. В меню Action (Действие) выберите команду Move (Переместить).

Откроется одноименное окно.

16. Выберите ОП Sales и щелкните кнопку ОК.

Учетная запись John Smith больше не отображается в правой панели ОП Users.

17. В дереве консоли щелкните ОП Sales.

В правой панели отображены учетные записи John Smith и Jane Doe и глобальная группа безопасности Sales.

18. Дважды щелкните глобальную группу Sales. Откроется диалоговое окно Sales Properties (Свойства: Sales).

19. Перейдите на вкладку Members (Члены группы).

Учетная запись John Smith по-прежнему член группы Sales, но находится теперь в папке microson.com/Sales.

20. Щелкните кнопку ОК. Создайте и используйте локальную группу домена. Создайте локальную группу домена для предоставления доступа к отчетам о продажах.

В нее Вы добавите глобальную группу безопасности, созданную на этапе 1.

1. Щелкните правую панель консоли, чтобы снять выделение с группы Sales.

2. В меню Action выберите New, а затем – команду Group.

Откроется диалоговое окно New Object – Group.

3. В поле Group Name (Имя группы) введите Reports.

4. Щелкните переключатели Security (Группа безопасности) и Domain Local (Локальная в домене).

5. Щелкните кнопку ОК.

На правой панели для ОП Sales появится локальная группа домена.

6. На правой панели дважды щелкните группу Reports,

Откроется диалоговое окно Reports Properties (Свойства: Reports).

7. Перейдите на вкладку Members (Члены группы).

8. Щелкните кнопку Add (Добавить).

Откроется диалоговое окно Select Users, Contacts, Computers, Or Groups.

9. В списке Look In (Искать в) выберите пункт Entire Directory (Вся папка).

Будут отображены учетные записи и группы всех доменов, а также расположение этих учетных записей и групп.

10. В списке учетных записей, групп и компьютеров щелкните заголовок Name (Имя).

Поле Name (Имя) будет отсортировано по алфавиту в убывающем порядке.

11. Снова щелкните этот заголовок, чтобы отсортировать поле по алфавиту в возрастающем порядке.

12. В списке учетных записей, групп и компьютеров выделите ОП Sales и шелкните кнопку Add (Добавить). Щелкните кнопку ОК.

Группа Sales стала членом доменной локальной группы Reports.

13. Щелкните кнопку ОК.

14. Закройте оснастку Active Directory Users And Computers.

Назначьте разрешения NTFS

Выполняйте упражнение на ServerOl.

1. Создайте на диске С: папку с именем Dept.

2. Сделайте эту папку обшей с именем ресурса Dept, а в поле Comment (Комментарий) введите Department share.

Для общего ресурса задавать разрешения не надо, так как папка Dept создана на томе NTFS.

3. Создайте в папке Dept подкаталог Sales.

4. Выделите папку Sales.

5. В меню File (Файл) выберите команду Properties (Свойства).

Откроется диалоговое окно Sales Properties (Свойства: Sales).

6. Перейдите на вкладку Security (Безопасность).

Системной группе Everyone (Все) предоставлены полные права управления данной папкой.

7. Снимите флажок Allows Inheritable Permissions From Parent To Propagate To This Object (Переносить наследуемые от родительского объекта разрешения на этот объект).

Появится сообщение Security (Безопасность) с описанием доступных вариантов выбора.

8. Щелкните кнопку Remove (Удалить).

Откроется диалоговое окно Sales Properties (Свойства: Sales).

9. Щелкните кнопку Add (Добавить).

Откроется диалоговое окно Select Users, Computers, Or Group.

10. В списке Look In (Искать в) выберите пункт Entire Directory (Вся папка).

11. В списке учетных записей, групп и компьютеров выделите Reports и щелкните кнопкуAdd.

12. Щелкните кнопку ОК.

В диалоговом окне свойств папки Sales показывается, что локальной группе Reports предоставлены разрешения Read & Execute (Чтение и выполнение), List Folder Contents (Просмотр содержимого папки) и Read (Чтение).

13. Пометьте флажок Write (Запись) и щелкните кнопку ОК.

14. Закройте окно Dept и завершите с;еанс Administrator (Администратор).

15. Зарегистрируйтесь в системе как Jane Doe с паролем student и откройте в окне My Computer (Мой компьютер) папку C:\Dept\SaIes.

16. В меню File выберите New, а затем – Text Document (Текстовый документ). В окне Sales появится файл New Text Document (Новый текстовый докуменг).

17. Дважды щелкните этот файл.

Откроется окно программы Notepad (Блокнот).

18. Введите несколько символов и закройте Notepad.

Появится запрос на сохранении изменений.

19. Щелкните кнопку Yes (Да).

20. Закройте окно Sales.

21. Завершите сеанс Jane_Doe и зарегистрируйтесь как Bob_Train без пароля.

В случае ошибки убедитесь, что Вы пытаетесь зарегистрироваться в интервал времени, когда пользователю Bob Train разрешено работать в системе. Вы задали этот интервал в одном из предыдущих упражнений данной главы.

22. Попробуйте обратиться к папке C:\Dept\Sales.

Сообщение Dept известит об отказе и доступе.

Дело в том, что Bob Train – не член глобальной группы Sales, которая в свою очередь включена в доменную локальную группу Report. Доступ к локальным папкам тоже. Невозможен, так как разрешения NTFS распространяются и на сетевой, и на локальный доступ.

23. Щелкните кнопку ОК и закройте окно Dept.

24. Завершите сеанс Bob Train.

Задание 4: Создание объекта групповой политики и настройка политики.

Создайте для своего домена GPO с именем Domain Policy, затем из оснастки Group Policy измените параметры безопасности GPO, чтобы разрешить группе Domain Users (Пользователи домена) локально входить на контроллеры домена. Выполняйте упражнение на ServerOl.

Создайте GPO на уровне домена.

1. Войдите в домен как Administrator с паролем password.

2. Раскройте меню Start\Programs\Administrative Tools и щелкните ярлык Active Directory

Users And Computers.

Откроется оснастка Active Directory Users And Computers.

3. В дереве консоли щелкните microsoft.com, затем в меню Action выберите команду Properties. Откроется окно свойств microsoft com.

4. Перейдите на вкладку Group Policy (Групповая политика) и щелкните кнопку Add (Добавить).

Откроется окно Add A Group Policy Object Link (Добавить ссылку на объект групповой политики).

5. Перейдите на вкладку АН (Все).

В списке имеется Default Domain Policy. Вы могли бы использовать этот GPO и изменять его по своему желанию, но сейчас создайте для домена новый GPO.

6. Щелкните среднюю из трех имеющихся на панели инструментов кнопок.

В перечне GPO появится New Group Policy Object (Новый объект групповой политики).

7. Назовите новый GPO Domain Policy и щелкните кнопку ОК.

В столбце Group Policy Object Links (ссылки на объекты групповой политики) появится объект Domain Policy.

8. Щелкните кнопку ОК, чтобы закрыть окно свойств политики.

9. Оставьте оснастку Active Directory Users And Computers открытой.

Задание 2: измените параметры безопасности

С помощью редактора Group Policy измените параметры безопасности, чтобы разрешить

группе Domain Users локально входить на ServerOl.

1. В дереве консоли раскройте microsoft.com.

2. Щелкните контейнер Domain Controllers.

3. В меню Action (Действие) выберите команду Properties (Свойства).

Откроется окно Domain Controllers Properties.

4. Перейдите на вкладку Group Policy.

Убедитесь, что в списке Group Policy Object Links выбрана строка Default Domain

Controllers Policy, и шелкните кнопку Edit.

5. Откроется оснастка Group Policy, отображающая дерево консоли Default Domain Controller Policy.

6. Проверьте, что узел Computer Configuration в дереве консоли раскрыт.

7. Раскройте в узле Computer Configuration узел Windows Settings\Security Settings\Local Policies.

8. В объекте Local Policies щелкните User Right Assignment.

На правой панели появится список атрибутов User Right Assignment.

9. На правой панели дважды щелкните Log On Locally.

Откроется окно Log On Locally. Заметьте: этот параметр политики назначен нескольким пользователям и группам.

10. Щелкните кнопку Add.

Откроется окно Add User Or Group.

11. Щелкните кнопку Browse.

Откроется окно Select Users Or Groups,

12. В списке Name выберите Domain Users (Пользователи домена), щелкните кнопку Add, затем – ОК.

Совет Если у Вас возникли затруднения при поиске группы Domain Users, просто введите Domain Users, и Windows сама найдет эту группу.

13. Еще раз щелкните кнопку ОК.

Группа Domain Users появится в списке пользователей и групп с правом локального входа.

14. Щелкните кнопку ОК и закройте оснастку Group Policy.

15. Щелкните кнопку ОК, чтобы закрыть окно Domain Controllers Properties.

16. Оставьте открытой оснастку Active Directory Users And Computers – она понадобится в следующем упражнении.

Теперь все пользователи домена могут входить на ServerOl локально.

Изменение политик ПО

Создайте и затем измените групповую политику ОП Sales, удалив из меню Start пункты Search (Найти) и Run (Выполнить). Вы также отключите политику Lock Workstation и просмотрите результаты этих изменений политики ПО. Наконец, Вы сделаете так, чтобы политика ОП Sales не перекрывала

групповую политику его родительского контейнера, домена. Выполняйте упражнение на ServerOl.

Создайте и измените политики ПО для ОП Sales

1. В оснастке Active Directory Users And Computers раскройте узел microsoft.com.

2. В дереве консоли шелкните Sales, затем в меню Action выберите команду Properties.

Откроется окно Sales Properties (Свойства: Sales).

3. Перейдите на вкладку Group Policy (Групповая политика).

4. Щелкните кнопку Add.

Откроется окно Add A Group Policy Object Link.

5. Перейдите на вкладку All и щелкните среднюю из трех кнопок на панели инструментов.

В списке Group Policy Objects Associated With This Container появится новый GPO.

6. Назовите новый GPO SalesSoftware и щелкните кнопку ОК.

Вернитесь на вкладку Group Policy окна свойства ОП Sales.

7. Выделите SalesSoftware и шелкните кнопку Edit (Изменить).

Откроется оснастка Group Policy.

8. Найдите и раскройте шаблоны Administrative в узле User Configuration.

9. В дереве консоли шелкните Start Menu & Task Bar.

На правой панели появятся политики, доступные для этой категории.

10. На правой панели дважды щелкните Remove Search Menu From Start Menu.

Откроется окно свойств этой политики.

11. Перейдите на вкладку Explain, чтобы прочитать описание этой политики.

12. Перейдите на вкладку Policy и шелкните переключатель Enabled.

13. Щелкните кнопку ОК.

14. Повторите пп. 10–13 для активизации политики Remove Run Menu From Start Menu.

15. В дереве консоли дважды щелкните System, затем Logon/Logoff.

На правой панели появятся политики, доступные для этой категории.

16. На правой панели активизируйте политику Disable Lock Computer.

17. Закройте оснастку Group Policy, затем закройте окно Sales Properties.

18. Закройте оснастку Active Directory Users And Computers.

Изучите действие политик ПО, созданных на предыдущем этапе.

После выполнения упражнений этой главы и главы 6 в ОП Sales должны находиться учетные записи пользователей Jane Doe и John Smith.

1. Завершите на ServerOl сеанс администратора.

2. Нажмите клавиши Ctrl+Alt+Delete.

3. Откроется окно Windows Security.

Кнопка Shutdown недоступна. Это контролируется политикой Shutdown Without Logon.

Windows 2000 Server не делает эту кнопку доступной по умолчанию.

4. Зарегистрируйтесь на ServerOl как Jane_Doe с паролем student.

5. Раскройте меню Start.

Заметьте: пункты Search и Run в меню Start не отображаются.

Задание 3: предотвратите перекрытие групповой политики

Вы помешаете ОП Sales перекрыть групповую политику его родительского контейнера.

1. Зарегистрируйтесь как Administrator с паролем password.

2. Раскройте меню Start\Programs\Administrative Tools и щелкните ярлык Active Directory Users And Computers.

Откроется оснастка Active Directory Users And Computers.

3. Раскройте узел microsoft.com.

4. Щелкните Sales, затем выберите в меню Action команду Properties.

Откроется окно Sales Properties.

5. Перейдите на вкладку Group Policy.

6. Проверьте, что в списке Group Policy Objects Link выбрана строка SalesSoftware, и щелкните кнопку Options.

7. Пометьте флажок No Override: Prevents Other Group Policy Objects From Overriding Policy Set In This One, затем щелкните кнопку ОК.

8. Еще раз щелкните кнопку ОК и закройте оснастку Active Directory Users And Computers.

Задание 5: Развертывание групповой политики

Создание ОГП для собственного ОП

1. Зарегистрируйтесь в домене как Administrator (Администратор).

2. Раскройте меню Start\Programs\Administrative Tools и щелкните Active Directory Users And Computers.

3. Дважды щелкните microsoft.com (или имя вашего домена).

4. Создайте новый ОП с именем Dispatch.

5. Щелкните ОП Dispatch правой кнопкой мыши, выберите команду Properties и перейдите на вкладку Group Policy.

6. Щелкните кнопку New (Создать) и введите имя нового ОГП – **DispatchPolicy.**

7. Щелкните Close.

Создание консоли ОГП

Вы создадите консоль для ОГП DispatchPolicy. Сохранив консоль, вы всегда сможете открыть ее из меню Administrative Tools.

Создайте консоль ОГП DispatchPolicy

1. В меню Stan выберите команду Run.

2. В поле Open (Открыть) введите mmc и щелкните ОК.

Откроется новая консоль управления.

3. В меню Console выберите команду Add/Remove Snap-In.

Откроетсяодноименное диалоговое окно

4. Щелкните кнопку Add.

Откроется диалоговое окно Add Standalone Snap-In.

5. Щелкните Group Policy и затем – кнопку Add.

Откроется окно Select Group Policy Object.

6. Щелкните кнопку Browse (Обзор), чтобы найти ОГП DispatchPollcy.

7. В открывшемся окне перейдите на вкладку All (Все), щелкните ОГП Dispatch Policy, затем ОК.

8. Щелкните кнопку Finish (Готово), затем – кнопку Close (Закрыть) в диалоговом окне Add Standalone Snap-In.

9. В диалоговом окне Add/Remove Snap-In щелкните ОК.

10. В меню Console выберите команду Save As (Сохранить как).

11. В окне Save As (Сохранить как) в поле File Name (Имя файла) введите **Dispatch Policy GPO** и щелкните кнопку Save (Сохранить).

Ярлык для оснастки DispatchPoticy GPO появится в программной группе Administrative Tools (Администрирование).

Делегирование управления ОГП

Вы предоставите группе Administrators административные полномочия в отношении ОГП DispathPolicy.

Делегируйте управление ОГП

1. Откройте консоль ОГП Dispatch Policy.

2. Щелкните корневой узел (DispatchPolicy [serverl.microsoft.com] Policy) консоли правой кнопкой мыши, выберите команду Properties и перейдите на вкладку Security (Безопасность).

Откроется диалоговое окно свойств для ОГП DispathPolicy.

Какие группы безопасности обладают административными полномочиями в отношении ОГП Dispatch Policy?

3. Добавьте группу Administrators (Администраторы), щелкнув кнопку Add.

4. Чтобы предоставить группе Administrators полные административные полномочия, предоставьте разрешения Read, Write, Create All Child Objects и Delete All Child Objects.

5. Щелкните ОК.

Определение параметров групповой политики

Вы настроите некоторые параметры ОГП DispatchPolicy.

Настройте параметры групповой политики для ОГП

1. В дереве консоли ОГП DispatchPolicy раскройте корневой узел.

2. Раскройте узел User Configuration\Administrative Templates (Конфигурация пользователя\Административные шаблоны).

3. Щелкните элемент Start Menu & Task Bar (Панель задач и меню ≪Пуск≫).

Что отображается в правой панели?

4. В правой панели дважды щелкните Remove Search Menu From Start Menu (Удалить меню ≪Найти≫ из главного меню).

Откроется одноименное диалоговое окно.

5. Щелкните переключатель Enabled (Включена), затем — ОК.

Как быстро определить, что этот параметр включен?

6. Повторите пункты 4 и 5, чтобы включить политику Remove Run Menu From Start Menu (Удалить команду ≪Выполнить≫ из меню ≪Пуск≫) (там же, в узле User Configuration).

7. В дереве консоли раскройте узел System (Система) и щелкните Logon/Logoff (Вход/выход из системы).

В правой панели отобразятся соответствующие политики.

8. В правой панели дважды щелкните политику Disable Lock Computer (Запретить блокировку компьютера), затем – ОК

Отключение неиспользуемых параметров групповой политики

Вы отключите узел Computer Configuration дерева консоли, поскольку все параметры в нем не заданы. Это ускорит загрузку и регистрацию в системе пользователей и компьютеров, на которые распространяется действие вашего ОГП.

Отключите узел Computer Configuration для вашего ОГП

1. Откройте консоль Dispatch Policy, щелкните корневой узел правой кнопкой мыши и выберите команду Properties.

Откроется диалоговое окно свойств ОГП Dispatch Policy.

2. На вкладке General щелкните Disable Computer Configuration Settings (Отключить параметры конфигурации компьютера).

Откроется диалоговое окно Confirm Disable (Подтвердить отключение), предлагающее подтвердить отключение *узла* Computer Configuration.

3. Щелкните Yes (Да), затем ОК.

Выявление исключений в порядке обработки ОГП

Вы настроите ОГП DispatchPolicy так, чтобы другие ОГП не могли переопределять его параметры.

Задание параметра No Override для вашего ОГП

1. Раскройте меню Start\Programs\Administrative TooIs\Active Directory Users And Computers.

2. Щелкните ОП Dispatch правой кнопкой мыши и выберите команду Properties.

3. Перейдите на вкладку Group Policy, щелкните ОГП DispatchPolicy и затем – кнопку Options (Параметры).

Откроется одноименное диалоговое окно.

4. Щелкните флажок No Override (Не перекрывать), затем – ОК.

5. В диалоговом окне свойств ОГП Dispatch щелкните ОК.

Фильтрование области действия ОГП

Вы заблокируете наследование политики для группы Sales, отозвав у последней разрешение Read для ОГП. Группа Sales и ее участники были созданы при выполнении упражнений главы 8.

1. Щелкните корневой узел консоли ОГП DispatchPolicy правой кнопкой мыши и выбе рите команду Properties.

Откроется одноименное диалоговое окно.

2. Перейдите на вкладку Security и щелкните группу безопасности Sales. Вам надо будет добавить данную группу с помощью кнопки Add.

3. Отмените для группы Sales разрешения Apply Group Policy и Read. Затем щелкните ОК.

Откроется диалоговое окно, предлагающее подтвердить отзыв разрешений.

4. Щелкните кнопку Yes,

привязка ОГП

По умолчанию параметры ОГП Dispatch Policy распространяются на ОП Dispatch. Вы создадите ссылку на ОГП DispatchPolicy для ОП Security I, созданного в главе 11.

1. Раскройте меню Start\Programs\Administrative Tools и щелкните Active Directory Users And Computers.

2. Щелкните ОП Security11 правой кнопкой мыши и выберите команду Properties

Откроется одноименное диалоговое окно.

3. Перейдите на вкладку Group Policy и щелкните кнопку Add.

Откроется диалоговое окно Add A Group Policy Object Link (Добавить ссылку на объект групповой политики).

4. Перейдите на вкладку All (Все), щелкните ОГП DispatchPolicy, затем – ОК.

5. В диалоговом окне свойств ОП Security ! щелкните ОК.

Тестирование ОГП

Сейчас вы проверите, как работает созданный вами ранее ОГП.

Проверьте ОГП DispatchPolicy

1. Зарегистрируйтесь в системе как Assistant 1, член ОП Security 1.

2. Нажмите комбинацию клавиш CtrI+Alt+Delete.

Откроется диалоговое окно Windows Security (Безопасность Windows).

Можете ли вы заблокировать рабочую станцию? Почему?

3. Щелкните кнопку Cancel (Отмена) и раскройте меню Start.

Отображаются ли в меню Start команды Search (Найти) и Run (Выполнить)?

4. Завершите сеанс работы Assistantl и затем зарегистрируйтесь в системе как Administrator.

5. Сделайте учетную запись Assistantl членом группы безопасности Sales.

6. Завершите сеанс работы Administrator и затем зарегистрируйтесь в системе как Assistant 1.

7. Нажмите комбинацию клавиш Ctrl+Alt+Delete.

Можете ли вы заблокировать рабочую станцию? Почему?

8. Завершите текущий сеанс работы.

Задание 6: определение стандартного перемещаемого профиля пользователя

Сейчас вы создадите общую папку, в которой может храниться стандартный перемещаемый профиль пользователя. Создайте учетную запись с именем Profile Template, котораябудет служить моделью для стандартного перемещаемого профиля. Задайте параметры для профиля шаблона. Скопируйте профиль пользователя Profile Template в общую папку для User2, Задайте путь к профилю для User2. Вы можете протестировать стандартный профиль, если имеете доступ к двум компьютерам сети.

1. Войдите в домен как Administrator (Администратор) на контроллере домена.

2. В папке C:\ (где C:\ – имя вашего системного диска) создайте папку с именем Profiles.

3. Щелкните правой кнопкой папку Profiles и выберите команду Properties (Свойства).

4. В окне Profiles Properties (Свойства: Profiles) перейдите на вкладку Sharing (Доступ).

5. Щелкните переключатель Share This Folder (Открыть общий доступ к этой папке), затем – кнопку Permissions (Разрешения).

6. В окне Permissions For Profiles (Разрешения для профилей) убедитесь, что выбрана группа Everyone (Все) и отмечен флажок Full Control (Полный контроль), и щелкните ОК.

7. В окне Profiles Properties (Свойства: Profiles) щелкните ОК.

Создайте шаблон профиля пользователя

1. Раскройте меню Start\Programs\Administrative Tools (Пуск\Программы\Администрирование) и щелкните Active Directory Users and Computers (Active Directory – пользователи и компьютеры).

2. В консоли Active Directory Users and Computers создайте учетную запись ptemplate. В списке справа от поля User Logon Name (Имя входа пользователя) выберите cmicrosoft.com. Добавьте ptemplate к группе Print Operators (Операторы печати), чтобы пользователь мог войти в контроллер домена.

3. Выйдите из Windows 2000.

4. Войдите в систему как ptemplate.

Локальный профиль пользователя автоматически создается для пользователей Profile Template на локальном компьютере в папке C:\Documents and Settings *регистрационное _имя_пользователя* (где C:\ – это имя вашего системного диска).

5. Щелкните правой кнопкой рабочий стол и выберите команду Properties (Свойства). Откроется окно Display Properties (Свойства: Экран).

7. Перейдите на вкладку Appearance (Оформление). Обратите внимание на текущую цветовую схему.

8. В списке Scheme (Схема) выберите другую схему и шелкните ОК. Рабочий стол немедленно изменится в соответствии с новой цветовой схемой.

9. Выйдите из системы и вновь войдите как ptemplate. Заметьте: цвета экрана сохранились в профиле пользователя.

10. Завершите сеанс Windows.

Скопируйте шаблон профиля в общую папку на сетевом сервере

1. Войдите в систему как Administrator (Администратор).

2. Воспользуйтесь консолью Active Directory Users and Computers (Active Directory -пользователи и компьютеры) для создания учетной записи User2. В списке справа от окна User Logon Name (Имя входа пользователя) выберите @microsoft.com. Добавьте User2 к группе Print Operators (Операторы печати), чтобы пользователь мог зарегистрироваться на контроллере домена.

3. Раскройте меню Start\Settings (Пуск\Настройка) и шелкните ярлык Control Panel (Панель управления).

4. На панели управления дважды шелкните значок System (Система). Откроется окно System Properties (Свойства системы).

5. Перейдите на вкладку User Profiles (Профили пользователей). Заметьте: были созданы профили для всех пользователей, ранее входивших на компьютер, в том числе и профиль пользователя MICROS O FT\ptemplate.

6. В списке Profiles Stored On This Computer (Профили, хранящиеся на этом компьютере) шелкните MICRO SO FT\ptemplate, затем – Copy To (Копировать).

7. В открывшемся окне в поле Copy Profile To (Копировать профиль на) введите \\ ***имя_компьютера*** – это SERVER1 или имя вашею компьютера). Это местоположение общей папки, где будет храниться шаблон профиля.

Определите пользователей, имеющих право применять профиль

1. В окне Copy To (Копирование профиля) в области Permitted To Use (Разрешить использование) щелкните кнопку Change (Изменить). Откроется окно Select User Or Group (Выбор: Пользователь или группа).

2. В столбце Name (Имя) щелкните User Two, затем – ОК. В столбце Permitted To Use окна Copy To появится строка MICROSOFT\user2.

3. Щелкните ОК.

В Windows Explorer (Проводник) просмотрите Profiles\user2. Обратите внимание на папки для параметров рабочего стола, хранящиеся в папке Profiles.

Задайте путь к перемещаемому профилю пользователя

1. В консоли Active Directory' Users and Computers дважды щелкните User Two. Откроется окно User Two Properties (Свойства: User Two).

2. Перейдите на вкладку Profile (Профиль).

3. В поле Profile path (Путь к профилю) введите ***\\имя_комяьютера\фгоП\е$\и$ег2*** (где *имя_компьютера* – это SERVER! или имя вашего компьютера).

4. Щелкните ОК.

5. Закройте консоль Active Directory Users and Computers.

Протестируйте перемещаемый профиль

1. Выйдите из системы и войдите как User2. Совпадают ли или отличаются цвета экрана и рабочий стол от заданных в Profile Template? Почему?

Определите тип профиля, назначенного пользователю

1. Выйдите из системы, войдите как Administrator (Администратор) и запустите панель управления.

2. Дважды щелкните строку System (Система) и перейдите на вкладку User Profiles (Профили пользователей).

Какие типы профиля перечислены для учетной записи User2?

3. Выйдите из всех программ и из Windows 2000.

Протестируйте перемещаемый профиль с другого компьютера

1. Войдите на второй компьютер как User2.

2. Если откроется окно со списком параметров профиля, щелкните кнопку Download (Загрузить). Заметьте: цвета экрана те же, что и на первом компьютере, потому что перемещаемый профиль для шаблонной учетной записи загружается с сервера и применяется к компьютеру, где регистрируются под этой записью.

3. Выйдите со второго компьютера.

Удалите профиль пользователя Profile Template

1. На вкладке User Profiles (Профили пользователей) в списке Profiles Stored On This Computer (Профили, хранящиеся на этом компьютере) щелкните профиль MSCROSOFT\ptemplate, затем – кнопку Delete (Удалить). Откроется окно сообщения Confirm Delete (Подтвердить удаление).

2. Щелкните кнопку Yes (Да), чтобы удалить локальный профиль.

Профиль пользователя Profile Template будет удален с локального компьютера.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы

1. Опишите изменение режима домена. Создание групп.
2. Опишите создание объекта групповой политики и настройка политики.
3. Опишите развертывание групповой политики
4. Опишите специфику работы стандартного перемещаемого профиля пользователя

5. Отображаются ли в меню Start команды Search (Найти) и Run (Выполнить)?

6. Можете ли вы заблокировать рабочую станцию? Почему?

Лабораторная работа № 10
Конфигурирование и использование EFS. Восстановление данных

Цель работы: Изучить конфигурирование и использование EFS. Восстановление данных в ОС Windows Server.

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2

1.Теоретический материал

Одним из аспектов сетевой безопасности NT-систем, которому часто уделяется недостаточно внимания, являются совместно используемые ресурсы (shares). Объявление общих ресурсов со слабой защищенностью, позволяющей неавторизированным пользователям просматривать системные файлы, является одной из наиболее распространенных брешей в безопасности. Система не предоставляет средств для просмотра видимых в сети ресурсов машины и их установок безопасности. Однако, используя NetBIOS, можно просмотреть доступные компьютеры внутри домена и изучить доступные дисковые ресурсы, общие принтеры, а также их установки безопасности. Данный способ наиболее эффективен для администратора домена, поскольку именно он имеет возможность просматривать все ресурсы сети.

Для отображения локальных дисков на сетевые каталоги используются функции:

WNetOpenEnum – получение манипулятора перечисления сетевых ресурсов (дисков и принтеров) с заданными параметрами;

WNetEnumResource – заполнение структуры NETRESOURCE информацией о перечисляемом сетевом ресурсе;

WNetCloseEnum – удаление манипулятора перечисления.

Для получения списка совместно используемых ресурсов (включая скрытые), а также количества подключённых к ним пользователей существуют следующие API функции:

NetShareEnum – перечисление совместно используемых ресурсов;

NetShareAdd – добавление совместно используемого ресурса;

NetShareDel – удаление совместно используемого ресурса.

2.Задание

Задание 1. Изучение конфигурирования и использования EFS

Сконфигурируйте политику нос становления данных в домене и зашифруйте папку. Упражнение выполняйте на ServerOl.

Политика восстановления конфигурируются по умолчанию, когда устанавливается первый контроллер домена. В итоге самостоятельно подписанный сертификат назначает агентом восстановления администратора домена. На этом этапе перед использованием EFS вручную добавьте администратора как агента восстановления.

1. Зарегистрируйтесь на ServerOl как Administrator с паролем password.

2. В меню Start выберите команду Run (Выполнить), в поле Open (Открыть) введите http:/ /serverOl/certsrv/ и щелкните кнопку ОК. В Internet Explorer откроется страница работы с сертификатами.

3. Выберите переключатель Request A Certificate (Запросить сертификат) и щелкните Next (Далее). Откроется страница Choose Request Type (Выбор типа запроса).

4. Щелкните переключатель Advanced Request (Расширенный запрос), а затем – кнопку Next (Далее).

Откроется страница Advanced Certificate Requests (Расширенные запросы на сертификаты).

5. Проверьте, что выбран переключатель Submit A Certificate Request To This CA Using A Form (Выдать запрос на сертификат этому ЦС, используя форму), затем щелкните кнопку Next (Далее).

6. В списке Certificate Template (Шаблон сертификата) выберите EFS Recovery Agent (Агент восстановления EFS).

7. Щелкните кнопку Submit (Выдать запрос).Откроется страница Certificate Issued (Сертификат выдан).

8. Щелкните ссылку Install This Certificate (Установить этот сертификат). Откроется страница Certificate Installed (Сертификат установлен).

9. Закройте Internet Explorer.

10. Из программной группы Administrative Tools (Администрирование) откройте оснастку Active Directory Users And Computers (Active Directory – пользователи и компьютеры).

11. В дереве консоли выберете узел microsoft.com.

12. В меню Action (Действие) выберите команду Properties (Свойства).

13. Откроется диалоговое окно microsoft. com Properties (Свойства: microsoft.com).

14. На вкладке Group Policy (Групповая политика) щелкните кнопку Edit (Изменить). Откроется оснастка Group Policy (Групповая политика).

15. Под узлом Computer Configuration (Конфигурация компьютера) раскройте контейнер Windows Settings (Конфигурация Windows), затем узел Security Settings (Параметры безопасности), контейнер Public Key Policies (Политики открытого ключа) и контейнер Encrypted Data Recovery Agents (Агенты восстановления шифрованных данных).

16. В меню Action (Действие) выберите команду Add (Добавить). Откроется окно мастера Add Recovery Agent (Мастер добавления атента восстановления).

17. Щелкните кнопку Next (Далее). Откроется окно Select Recovery Agents (Выбор агентов восстановления).

18. Щелкните кнопку Browse Directory (Обзор каталога). Откроется диалоговое окно Find Users, Contacts and Groups (Поиск: пользователи, контакты и группы).

19. Щелкните кнопку Find Now (Найти).

20. В списке найденных пользователей и групп дважды щелкните Administrator (Администратор). Откроется окно Select Recovery Agents (Выбор агентов восстановления).

21. Щелкните кнопку Next (Далее). Откроется окно завершения работы мастера,

22. Щелкните кнопку Finish (Готово). На правой панели оснастки Group Policy (Групповая политика) появится строка Administrator (Администратор).

23. Щелкните эту строку.

24. В меню Action (Действие) выберите команду Properties (Свойства). Откроется диалоговое окно Administrator Properties. Заметьте: для этого сертификата в принципе доступны все назначения, а в настоящее время лишь одно – File Recovery (Восстановление файлов).

25. Щелкните кнопку ОК.

26. Закройте оснастку Group Policy. Откроется диалоговое окно свойств microsoft.com.

27. Щелкните кнопку ОК. Откроется оснастка Active Directory Users And Computers,

28. В меню View (Вид) выберите команду Advanced Features (Дополнительные функции).

29. В дереве консоли шелкните контейнер Users.

30. На правой панели щелкните Administrator (Администратор).

31. В меню Action (Действие) выберите команду Properties (Свойства). Откроется диалоговое окно Administrator Properties (Свойства: Администратор).

32. Щелкните вкладку Published Certificates (Опубликованные сертификаты). Появится список сертификатов стандарта X.509, изданных для данной учетной записи пользователя.

33. Щелкните кнопку ОК.

34. Закройте оснастку Active Directory Users And Computers.

Зашифруйте папку с использованием EFS. Зашифруйте папку на ServerOl с помощью Windows Explorer.

1. На рабочем столе дважды щелкните ярлык My Computer (Мой компьютер).

2. Дважды щелкните диск Local Disk (C:) [Локальный диск (C:)].

3. Дважды щелкните папку Document And Settings.

4. Дважды щелкните папку Administrator (Администратор).

5. Выделите папку My Documents (Мои документы).

6. В меню File (Файл) выберите команду Properties (Свойства). Откроется диалоговое окно My Documents Properties (Свойства: Мои документы).

7. Щелкните кнопку Advanced (Другие).

Откроется диалоговое окно Advanced Attributes (Дополнительные атрибуты).

8. Пометьте флажок Encrypt Contents To Secure Data (Шифровать содержимое для зашиты данных) и дважды щелкните кнопку ОК, чтобы закрыть окно свойств папки,

9. Закройте окно Administrator.

Задание 2: Установка и конфигурирование служб сертификации.

Установите корневой ЦС предприятия и с его помощью выпустите, установите и отзовите несколько сертификатов. Установите службы сертификации на ServerOl. Он будет выполнять роль корневого ЦС предприятия.

1. Зарегистрируйтесь на ServerOl как Administrator с паролем password.

2. Раскройте меню Start\Settings (Пуск\Настройка) и щелкните ярлык Control Panel (Панель управления). Откроется окно Control Panel (Панель управления).

3. Дважды щелкните значок Add/Remove Programs (Установка и удаление программ). Откроется одноименное окно.

4. На левой панели шелкните значок Add/Remove Windows components (Добавление и удаление компонентов Windows). Откроется окно мастера компонентов Windows.

5. Пометьте флажок Certificate Services (Службы сертификации). Появится сообщение, что после установки служб сертификации данный компьютер не сможет быть переименован, включен в домен или удален из его состава.

6. Щелкните кнопку Yes (Да).

7. В окне Windows Components (Компоненты Windows) щелкните кнопку Details (Состав). Откроется окно Certificate Services (Службы сертификации).

8. Щелкните кнопку ОК.

9. В окне Windows Components (Компоненты Windows) щелкните кнопку Next (Далее). Откроется окно Certification Authority Type (Тип центра сертификации).

10. Выберите последовательно каждый переключатель и изучите описание каждого типа ЦС. Заметьте: ЦС предприятия можно установить, только если на сервере установлены службы Active Directory. Изолированный ЦС не зависит от Active Directory.

11. Щелкните переключатель Enterprise Root CA (корневой ЦС предприятия), пометьте флажок Advanced Options (Дополнительные возможности) и щелкните кнопку Next (Далее).

Откроется окно Public and Private Key Pair (Пара из открытого и закрытого ключей).

Здесь же Вы можете задать длину ключа или применить существующий ключ, установленный на компьютере, а также импортировать ключи и просматривать сертификаты.

12. Убедитесь, что в списке CSP (Поставщик CSP) выбран пункт Microsoft Base Cryptographic Provider vl.O, в списке Hash Algorithm (Алгоритм хеширования) – SHA-1, а в списке Key Length (Длина ключа) – Default (По

умолчанию). Щелкните кнопку Next (Далее). Откроется окно CA Identifying Information (Сведения о центре сертификации).

13. Введите данные из таблицы в текстовые поля этого окна:

Заметьте: данный сертификат будет иметь силу в течение двух лет.

14. Щелкните кнопку Next (Далее).

Откроется окно Data Storage Location (Размещение хранилища данных). По умолчанию БД сертификатов и ее журнал CertLog помещаются в загрузочном разделе диска. Если его емкость невелика, укажите другой защищенный раздел. Флажок Store configuration information in a shared folder (Сохранить сведения о конфигурации в общей папке) не имеет значения, если на компьютере есть Active Directory и компьютер, работающий как ЦС, входит в какой-либо домен. Информация о конфигурации данного ЦС автоматически публикуется в хранилище Active Directory.

15. Щелкните кнопку Next (Далее).

Появится сообщение, что в данный момент работает служба IIS и для продолжения установки надо завершить работу этой службы.

16. Щелкните кнопку ОК. Откроется окно Configuring Components (Настройка компонентов), а затем – окно Completing the Windows Components Wizard (Завершение работы мастера компонентов Windows).

17. Щелкните кнопку Finish (Готово), а затем в окне Add/Remove Programs – кнопку Close (Закрыть).

18. Закройте окно Control Panel.

Сгенерируйте, установите и отзовите сертификат для ServerOl (выполняется при условии наличия центра сертификации)

Задействуйте Web-страницу работы с сертификатами и оснастку Certificate Authority (Центр сертификации).

1. Раскройте меню Start\Programs\Administrative Tools и щелкните ярлык Certification Authority (Центр сертификации). Откроется оснастка Certification Authority (Центр сертификации).

2. В дереве консоли раскройте узел Enterprise CA.

3. Выберите папку Pending Requests (Запросы в ожидании) и сверните окно оснастки.

4. В меню Start (Пуск) и выберите команду Run (Выполнить). Откроется диалоговое окно Run (Запуск программы).

5. В поле Open (Открыть) наберите http://server01/certsrv и щелкните кнопку ОК.Откроется окно мастера подключения к Интернету.

6. Выберите способ подключения – I Connect Through A Local Area Network (LAN) (С помощью локальной сети). Затем щелкните кнопку Next (Далее). Откроется окно Local Area Network Internet Configuration (Параметры Интернета для локальной сети).

7. Сбросьте флажок Automatic Discovery Of Proxy Server (Recommended) (Автоматическое определение прокси-сервера) и щелкните кнопку Next (Да-

лее). Откроется окно Set Up Your Internet Mail Account (Настройте учетную запись почты Интернета).

8. Щелкните переключатель No (Нет), а затем – кнопку Next (Далее).

Откроется окно Completing The Internet Connection (Завершение настройки).

9. Щелкните кнопку Finish (Готово).

В Internet Explorer откроется страница работы с сертификатами.

10. Изучите информацию на этой странице и убедитесь, что выбран переключатель Request A Certificate (Запросить сертификат).

11. Щелкните кнопку Next (Далее).

Откроется страница Choose Request Type (Выбор типа запроса) с выбранным пунктом User Certificate Request (Запрос сертификата пользователя).

12. Щелкните кнопку Next (Далее).

Откроется страница User Certificate – Identifying Information (Сертификат программы обзора веба – Идентифицируюшая информация).

13. Заполните два первых поля и щелкните кнопку More Options (Дополнительные параметры).

Заметьте: выбран тот тип CSP, который Вы задали при установке служб сертификации.

14. Щелкните кнопку Submit (Выдать запрос).

15. Щелкните ссылку Ноте (Домой), чтобы вернуться на основную страницу работы с сертификатами.

16. Сверните окно Internet Explorer и восстановите окно оснастки Certification Authority (Центр сертификации).

На правой панели оснастки появится Ваш запрос сертификата. Если Вы не увидите его, нажмите клавишу F5, чтобы обновить содержимое панели.

17. Щелкните значок запроса правой кнопкой, в контекстном меню выберите All Tasks (Все задачи), затем – команду Issue (Выдать).

18. В дереве консоли выберите папку Issued Certificates {Выданные сертификаты). На правой панели оснастки появится Ваш сертификат. Если Вы не увидите его, нажмите клавишу F5, чтобы обновить содержимое панели.

19. Дважды щелкните этот сертификат.

Откроется окно Certificate (Сертификат) с тремя вкладками.

20. Перейдите на вкладку Details (Состав).

21. В списке под раскрывающимся списком Show (Показать) щелкните строку Issuer (Поставщик).

В нижнем текстовой области будут отражены сведения, которые Вы ввели в окне CA Identifying Information.

22. Щелкните кнопку ОК, чтобы закрыть окно свойств сертификата.

23. Сверните окно оснастки Certification Authority и восстановите окно Internet Explorer.

24. Щелкните переключатель Check on a pending certificate (Проверить ожидающий выполнения запрос на сертификат), а затем – кнопку Next (Далее).

Откроется страница с информацией об ожидающих выполнения запросах сертификатов.

25. Щелкните кнопку Next (Далее).

Откроется страница Certificate Issued (Сертификат выдан).

26. Щелкните ссылку Install This Certificate (Установить этот сертификат).

Откроется страница Certificate Installed (Сертификат установлен).

27. Закройте Internet Explorer.

28. Восстановите окно оснастки Certification Authority (Центр сертификации) и в ее правой панели выберите Ваш сертификат.

29. В меню Action (Действие) выберите Al l Tasks (Все задачи), а затем – команду Revoke Certificate (Отзыв сертификата).

Откроется диалоговое окно Certif cate Revocation (Отзыв сертификатов).

30. В списке Reason Code выберите Key Compromise (Компрометация ключа) и щелкните кнопку Yes (Да).

31. В дереве консоли щелкните папку Revoked Certificates (Отозванные сертификаты).

Отозванный сертификат появится на правой панели.

32. В меню Action (Действие) выберите All Tasks (Все задачи), а затем – команду Publish (Публикация).

Диалоговое окно Certificate Revocation List (Список отзыва сертификатов) сообщит, что предыдущий список еще действителен.

33. Щелкните кнопку Yes (Да).

34. Закройте оснастку Certification Authority (Центр сертификации).

35. В меню Start (Пуск) выберите команду Run (Выполнить).

Откроется окно Run (Запуск программы), где в поле Open (Открыть) уже будет набра на ссылка на каталог Certsrv.

36. Щелкните кнопку ОК.

В Internet Explorer будет открыта страница работы с сертификатами.

37. Щелкните переключатель Retrieve The CA Certificate Or Certificate Revocation List (Получить сертификат ЦС или список отзыва сертификатов), а затем – кнопку Next (Далее).

38. На открывшейся странице щелкните ссылку Download Latest Certificate Revocation List (Загрузить последний список отзыва сертификатов). Откроется диалоговое окно File Download (Загрузка файла).

39. Щелкните переключатель Open This File From Its Current Location (Открыть этот файлиз текущего места), а затем – кнопку ОК.

Откроется диалоговое окно Certificate Revocation List (Список отзыва сертификатов).

40. Перейдите на вкладку Revocation List (Список отзыва).

41. В списке Revoked Certificates (Отозванные сертификаты) щелкните отозванный сертификат.

В списке ниже отобразится серийный номер сертификата, дата и причина отзыва.

42. Щелкните кнопку ОК, чтобы закрыть окно Certificate Revocation List (Список отзыва сертификатов).

43. Закройте Internet Explorer.

Задание 3: Создание и использование оснастки

Security Analysis And Configuration

Создайте собственную оснастку, содержащую оснастки Security Analysis And Configuration (Анализ и настройка безопасности) и Security Templates (Шаблоны безопасности). Затем создайте шаблон и, используя его, откройте новую БД. Проанализируйте параметры безопасности ServerOl на соответствие шаблону и примените свой шаблон для задания параметров безопасности на ServerOl. Выполняйте это упражнение на ServerOl.

Создайте консоль для анализа безопасности (повторение пройдённого материала)

Запустите консоль управления MMC и добавьте к ней оснастку Security Analysis And Configuration. Версия 1.2 консоли MMC из состава Windows позволяет добавлять к нейсвои оснастки. Вместо добавления оснасток к существующей консоли создайте новую.

1 . Зарегистрируйтесь на ServerOl как Administrator с паролем password.

2. В меню Start (Пуск) выберите команду Run (Выполнить).

Откроется диалоговое окно Run (Запуск программы).

3. В поле Open (Открыть) введите mmc и щелкните кнопку ОК.

Откроется пустая консоль MMC с именем Console! (Консоль!).

4. В меню Console (Консоль) выберите команду Add/Remove Snap-in.

Откроется одноименное диалоговое окно.

5. Щелкните кнопку Add (Добавить).

Откроется диалоговое окно Add Standalone Snap-in (Добавить изолированную оснастку).

6. Выберите в списке оснастку Security Configuration And Analysis (Анализ и настройка безопасности) и щелкните кнопку Add (Добавить).

7. Щелкните кнопку Close (Закрыть).

8. Щелкните кнопку ОК.

9. В меню Console (Консоль) выберите команду Save (Сохранить).

Откроется диалоговое окно Save As (Сохранить как).

10. В поле File Name (Имя файла) введите Security и щелкните кнопку Save (Сохранить).

Добавьте и задайте параметры безопасности с помощью оснастки Security

Template в консоли Security

До проведения анализа ServerO! и задания новых параметров безопасности установите модуль Security Template в Вашу консоль Security.

1. В меню Console выберите команду Add/Remove Snap-in.

Откроется диалоговое окно Add/Remove Snap-in.

2. Щелкните кнопку Add (Добавить).

Откроется окно Add. Standalone Snap-in.

3. Выберите в списке оснасток Security Templates (Шаблоны безопасности) и щелкните кнопку Add.

4. Щелкните кнопку Close.

5. Щелкните ОК.

6. В меню Console (Консоль) выберите команду Save (Сохранить).

7. Раскройте узел Security Templates (Шаблоны безопасности), а затем – папку C:\WTNNT\Security\Templates.

Все установленные шаблоны отображаются в дереве консоли и в правой панели.

8. Раскройте узел securedc.

Этот шаблон более строгой безопасности обычно используется после применения основного шаблона безопасности.

9. Раскройте узел Account Policies (Политики учетных записей) и щелкните папку Password Policy (Политика паролей).

Параметры политики паролей появятся в правой панели.

10. На правой панели дважды щелкните параметр Minimum Password Length (Мин. Длина пароля).

Откроется диалоговое окно Template Security Policy Setting (Параметр шаблона политики безопасности).

11. В поле Password Must Be At Least (Длина пароля не менее) введите 5 и щелкните кнопку ОК.

12. В дереве консоли щелкните пункт securedc.

13. В меню Action (Действие) выберите команду Save As (Сохранить как). Откроется одноименное окно.

14. В поле File Name (Имя файла) наберите customdc и щелкните кнопку Save (Сохранить).

15. В дереве консоли щелкните пункт customdc.

16. В меню Action (Действие) выберите команду Set Description (Задать описание).

Откроется окно Security Template Description (Описание шаблона безопасности).

17. В поле Description (Описание) наберите Custom Security Template for Training и щелкните кнопку ОК.

18. В дереве консоли щелкните папку C:\WINNT\Security\Templates.

Заметьте: в правой панели для customdc теперь отображается соответствующее описание.

19. Почитайте описания других шаблонов Windows Server.

Создайте новую базу данных системы безопасности

1. В дереве консоли щелкните пункт Security Configuration And Analysis (Анализ и настройка безопасности) и прочитайте текст в правой панели.

2. В меню Action (Действие) выберите команду Open Database.(Открыть базу данных).

Откроется диалоговое окно Open Database.

3. В поле File Name (Имя файла) наберите training и щелкните кнопку Open (Открыть).

Откроется диалоговое окно Import Template (Импортировать шаблон).

4. Щелкните customdc.inf, а затем – кнопку Open (Открыть).

Это тот шаблон, который Вы создали на предыдущем этапе.

Проанализируйте текущие параметры безопасности

Проанализируйте текущие параметры ServerOl на соответствие созданному Вами шаблону.

1. Удостоверьтесь, что в дереве консоли выбран узел Security Configuration And Analysis (Анализ и настройка безопасности).

2. В меню Action (Действие) выберите команду Analyze Computer Now (Анализ компьютера).

Откроется диалоговое окно Perform Analysis (Анализ), показывающее путь к журналуошибок и его имя в следующем виде: C:\Documents and Settings\Administrator\LocaISettings\Temp\traimng.log.

3. Щелкните кнопку ОК.

Открывшееся окно Analyzing System Security (Анализ безопасности системы) сообщит о проверке различных аспектов конфигурации безопасности ServerOl на соответствие Вашему шаблону.

4. Когда закончится анализ, раскройте узел Security Configuration And Analysis.

5. Раскройте узел Account Policies (Политики учетных записей) и щелкните узел Password Policy (Политика паролей).

В правой панели для каждой политики отображены параметры шаблона и настройки компьютера. Несовпадения отмечаются красным кругом с крестом в центре. Совпадения отмечаются белым кругом с зеленой галочкой в центре. Если флажка или метки нет, то этот параметр безопасности не указан в шаблоне.

6. В дереве консоли щелкните узел Security Configuration And Analysis (Анализ и настройка безопасности).

7. В меню Action (Действие) выберите команду Configure Computer Now (Настроить компьютер).

Откроется диалоговое окно Configure System (Настройка системы).

8. Щелкните кнопку ОК.

9. В меню Action (Действие) выберите команду Analyze Computer Now (Анализ компьютера).

Откроется диалоговое окно Perform Analysis (Анализ).

10. Щелкните кнопку ОК.

П. Просмотрите параметры политики и удостоверьтесь, что столбцы Database Settings и Computer Setting совпадают.

12. Закройте оснастку Security.

Откроется окно сообщения с предложением сохранить консоль.

13. Щелкните кнопку Yes (Да).

14. Если откроется окно сохранения шаблонов безопасности, щелкните кнопку Yes (Да).

Задание 4: С помощью мастера Backup Вы создадите резервные копии некоторых файлов на жестком диске. Затем, используя Task Scheduler, Вы создадите отложенное задание резервного копирования. Создайте, выполните и проверьте задание резервного копирования

Для создания резервной копии файлов на локальном лиске ServerOl Вы запустите утилиту

Backup и поработаете с мастером архивации.

1. Зарегистрируйтесь на ServerOl как Administrator с паролем password.

2. В меню Start (Пуск) выберите команду Run (Выполнить).

Откроется диалоговое окно Run (Запуск программы).

3. В поле Open (Открыть) наберите ntbackup и щелкните кнопку ОК.

Откроется диалоговое окно Backup – [Untitled] (Архивация – [Безымянный]).

4. Прочтите описания трех вариантов работы с утилитой на вкладке Welcome (Добро пожаловать) и щелкните кнопку Backup Wizard.

Откроется окно Welcome To The Windows 2000 Backup And Recovery Tools (Мастер архивации и восстановления Windows 2000).

5. Щелкните кнопку Next (Далее).

Откроется окно What To Back Up (Что следует архивировать), где Вам предлагается выбрать копируемые данные.

6. Щелкните переключатель Back Up Selected Files, Drives, Or Network Data (Архивировать файлы, диски или сетевые данные), а затем – кнопку Next (Далее).

Откроется окно Items To Back Up (Элементы для архивации), где надо выбрать локальные и сетевые диски, папки и файлы, которые будут включены в архив.

7. Раскройте узел My Computer (Мой компьютер).

8. Щелкните пункт System State (Состояние системы). Не щелкайте флажок слева от System State!

На правой панели указано, что будут созданы резервные копии хранилища Active

Directory, загрузочных файлов, реестра, базы данных регистрации классов COM+, папки SYSVOL и базы данных служб сертификации.

9. На левой панели окна раскройте узел диска С и щелкните букву С. Не щелкайте флажок слева от С.

10. На правой панели пометьте флажок рядом с Boot.ini и щелкните кнопку Next (Далее),

Откроется окно Where To Store The Backup (Где хранить архив).

Если к компьютеру не подключен накопитель на магнитной ленте, раскрывающийся список Backup Media Type (Тип носителя архива) будет недоступен.

В этом случае File (Файл) – единственно доступный тип носителя для архива.

11. В поле Backup Media Or File Name (Носитель архива или имя файла) наберите c:\backupl.bkf и щелкните кнопку Next (Далее).

Обычно резервное копирование выполняется на ленту или в файл, сохраняемый на другой жесткий диск, устройство со сменным диском (типа Iomega Zip или Jaz), записываемый компакт- или оптический диск. Мы для простоты сохраним архив на то же устройство, где расположен файл, копия которого создается Откроется окно Completing The Backup Wizard (Завершение работы мастера архивации), тде приведена сводка параметров задания.

12. Для задания дополнительных параметров щелкните кнопку Advanced (Дополнительно). Откроется окно Type Of Backup (Тип архива),

13. Просмотрите типы резервного копирования, перечисленные в списке Select The Type Of Backup Operation To Perform (Выберите нужный тип операции архивирования).

Типы архивов были описаны выше.

14. Убедитесь, что выбран тип Normal (Обычный).

15. Убедитесь, что флажок Backup Migrated Remote Storage Data (Архивировать данные из внешних хранилищ) сброшен.

Этот параметр включает поддержку возможностей HSM в Windows 2000 Server.

16. Щелкните кнопку Next (Далее).

Откроется окно How To Backup (Способы архивации), где Вам предлагается включить проверку данных резервной копии после ее создания.

17. Пометьте флажок Verify Data After Backup (Проверять данные после архивации) и щелкните кнопку Next (Далее).

Откроется окно Media Options (Параметры носителей), где Вам предлагается добавить текущую копию к существующей, либо перезаписать старую копию.

18. Пометьте флажок Replace The Data On The Media With This Backup (Затереть данные носителя этим архивом).

Обратите внимание на флажок Allow Only The Owner And The Administrator Access To The Backup Data And Any Backups Appended To This Media (Разрешать доступ к данным этого архива и всем дозаписываемым на этот носитель архивам только владельцу и администратору). Этот параметр обеспечивает более высокий уровень безопасности.

При его выборе восстановить данные из резервной копии сможет только владелец файла или Administrator (Администратор). Убедитесь, что этот флажок сброшен.

19. Щелкните кнопку Next (Далее).

Откроется окно Backup Label (Метка архива), в котором вводится название задания резервного копирования и носителя архива.

Мастер задает эти метки на основе текущей даты и времени.

20. В поле Backup Label (Метка архива) наберите Boot.ini backup set created on <дата>, где<дата> – текущая дата и время.

21. Не меняйте текст в поле Media Label (Метка носителя). Щелкните кнопку Next.

Откроется окно When To Back Up (Когда архивировать), где предлагается начать резервное копирование немедленно или позже по указанному расписанию.

22. Убедитесь, что выбран переключатель Now (Сейчас). Щелкните кнопку Next.

Откроется окно Completing The Backup Wizard (Завершение работы мастера архивации).

23. Для начала резервного копирования щелкните кнопку Finish (Готово).

Откроется диалоговое окно Selection Information (Информация о выборе), где отображается краткая информация о размере копируемых данных и предполагаемой длительности копирования.

Откроется диалоговое окно Backup Progress (Ход архивации), где отображается текущая информация о состоянии задания резервного копирования, примерном суммарном размере и текущем размере обработанных данных, текущем времени выполнения задания и времени, оставшемся до его завершения.

24. Увидев сообщение о завершении задания, щелкните кнопку Report (Отчет).

Запустится программа Notepad со сформированным отчетом о произведенном резервном копировании. Отчет содержит основную информацию о задании резервного копирования: время его начала и количество скопированных файлов.

25. Закройте программу Notepad.

26. В диалоговом окне Backup Progress (Ход архивации) щелкните кнопку Close (Закрыть).

Откроется диалоговое окно Backup – [Untitled] с открытой вкладкой Welcome.

Создайте, выполните и проверьте автоматически выполняемые задание архивации

Вы создадите отложенное задание резервного копирования с помощью Task Scheduler.

1. На вкладке Welcome (Добро пожаловать) щелкните кнопку Backup Wizard (Мастер архивации).

После запуска мастера откроется окно Welcome To The Windows 2000 Backup And Recovery Tools (Мастер архивации и восстановления Windows 2000).

2. Щелкните кнопку Next (Далее).

Откроется окно What To Back Up (Что следует архивировать), где надо выбрать данные, предназначенные для копирования.

3. Щелкните переключатель Back Up Selected Files, Drives, Or Network Data (Архивировать выбранные файлы, диски и сетевые данные) и щелкните кнопку Next.

Откроется окно Items To Back Up (Элементы для архивации), где надо выбрать локальные и сетевые диски, папки и файлы, которые будут скопированы.

4. Раскройте узел My Computer (Мой компьютер), затем диск C и пометьте флажком папку Inetpub.

5. Щелкните кнопку Next (Далее).

Откроется окно Where To Store The Backup (Где хранить архив), где надо выбрать расположение резервной копии.

6. В поле Backup Media Or File Name (Носитель архива или имя файла) наберите C:\backupZ.bkf и щелкните кнопку Next (Далее).

Откроется окно Completing The Backup Wizard (Завершение работы мастера архивации).

7. Для задания дополнительных параметров щелкните кнопку Advanced (Дополнительно).

Откроется окно Type Of Backup (Тип архива), где Вам предлагается выбрать тип создаваемого архива.

8. Убедитесь, что в списке Type Of Backup Operation To Perform (Выберите нужный тип операции архивирования) выбрано Normal (Обычный).

9. Щелкните кнопку Next (Далее).

Откроется окно How To Backup (Способы архивации).

10. Пометьте флажок Verify Data After Backup (Проверять данные после архивации) и щелкните кнопку Next (Далее).

Откроется окно Media Options (Параметры носителей).

11. Щелкните переключатель Replace The Data On The Media With This Backup (Затереть данные носителя этим архивом).

12. Убедитесь, что флажок Allow Only The Owner And The Administrator Access To The Backup Data And Any Backups Appended To This Media (Разрешать доступ к данным этого архива и всем дозаписывагмым на этот носитель архивам только владельцу и администратору) сброшен, и щелкните кнопку Next (Далее).

Откроется окно Backup Label (Метка архива), предлагающее ввести метку архива и носителя.

13. В поле Backup Label (Метка архива) наберите Inetpub backup set created on <дата>, где <дата> – текущая дата и время.

14. Не изменяйте текст в поле Media Label (Метка носителя). Щелкните кнопку Next. Откроется окно When To Back Up (Когда архивировать).

15. Щелкните переключатель Later (Позже).

Откроется диалоговое окно Set Account Information (Указание учетной записи), в котором надо ввести пароль учетной записи MlCROSOFT\administrator. Если служба TaskScheduler не настроена для автозапуска, сначала откроется дишюговое окно с предложением запустить эту

службу. Щелкните кнопку ОК, после чего откроется диалоговое окно Set Account Information.

Поскольку служба Task Scheduler автоматически запускает приложения, не проверяя параметров безопасности и прав пользователя компьютера или домена, нужно указать имя и пароль пользователя для запуска отложенного резервного копирования. Для назначенного задания архивации Вы должны быть членом группы Backup Operators (Операторы архива) с разрешениями доступа ко всем копируемым файлам и палкам.

Для упрощения задачи при настройке задания архивации используйте учетную запись Administrator (Администратор).

16. Убедитесь, что в поле Run As (Пользователь) появился текст MICROSOFT\admmistrator.

Затем в полях ввода пароля наберите password,

17. Щелкните кнопку ОК,

18. В поле Job Name (Имя задания) наберите Inetpub Backup и щелкните кнопку Set Schedule (Установить расписание).

Откроется диалоговое окно Schedule Job, в котором надо назначить время начала и параметры расписания резервного копирования.

19. В списке Schedule Task выберите Daily (Ежедневно), а в поле Start Time (Время начала) введите текущее время, прибавив к нему 5 минут.

20. Щелкните кнопку Advanced (Дополнительно).

Откроется окно Advanced Schedule Options (Дополнительные параметры расписания).

21. Пометьте флажок End Date (Дата окончания), выберите в списке завтрашнюю дату и щелкните кнопку ОК.

Откроется окно Schedule Job (Запланированное задание).

22. Щелкните кнопку ОК.

Откроется окно When To Backup (Когда архивировать).

23. Щелкните кнопку Next (Далее).

Откроется окно мастера Completing The Backup Wizard (Завершение работы мастераархивации), отображающее сводку выбранных Вами параметров задания.

24. Для запуска задания резервного копирования щелкните кнопку Finish (Готово).

Откроется диалоговое окно Backup – [Untitled] (Архивация – [Безымянный]) с открытой вкладкой Welcome (Добро пожаловать).

25. Закройте окно утилиты Backup.

Задание резервного копирования запустится в назначенное время.

26. Запустите Windows Explorer (Проводник Windows), щелкните диск C: и убедитесь в наличии файла Backup2.bkf.

Просмотрите и настройте задания

Вы просмотрите назначенные задания резервного копирования и назначите новые.

1. Выберите Start\Accessories\System Tools (Пуск\Стандартные\Служебные) и щелкните ярлык Scheduled Tasks (Назначенные задания).

Откроется одноименное окно. Заметьте, что в списке заданий присутствует Inetpub Backup.

2. Дважды щелкните значок задания Inetpub Backup.

Обратите внимание на текст в поле Run (Выполнить). Это команда консольного приложения ntbackup с параметрами, сгенерированными мастером для архивации папки Inetpub.

Если перед запуском резервного копирования надо остановить какую-либо службу, например, Certificate Services, создайте командный файл (.cmd или .bat) с командами остановки службы, запуска задачи резервного копирования и последующего запуска остановленной службы. Команда для остановки службы Certificate Services: net stop «certificate services», а для запуска net start «certificate services»

3. Щелкните вкладку Schedule (Расписание).

В списке указано задание, созданное с помощью мастера Backup.

4. Чтобы закрыть окно Inetpub Backup шелкните кнопку ОК.

Откроется окно Scheduled Tasks (Назначенные задания).

5. В меню File (Файл) выберите команду Delete (Удалить).

Откроется окно Confirm File Delete (Подтверждение удаления файла).

6. Щелкните кнопку Yes (Да).

7. Закройте окно Scheduled Tasks (Назначенные задания).

Задание 5: Включение дисковых квот

Измените используемые по умолчанию параметры управления дисковыми квотами и ограничьте объем информации, которую пользователи могут хранить на диске С: компьютера ServerOl. На диске С: имеется разделяемый общедоступный каталог HomeDirs, созданный Вами для John Smith. Затем настройте для пользователя персональную дисковую квоту – увеличьте максимальный объем дискового пространства, доступный пользователю, до 20 Мб, и установите порог выдачи предупреждений равным] 6 Мб. После этого отключите управление квотами для диска С:. Упражнение выполняйте на ServerOl.

Настройте параметры управления квотами

Настройте параметры управления квотами для диска С: и ограничьте объем информации, которую пользователи могут хранить в томе.

1. Зарегистрируйтесь на ServerOl как Administrator (Администратор) с паролем password.

2. Дважды щелкните значок My Computer (Мой компьютер).

3. Щелкните значок Local Disk (C:) [Локальный диск (C:)], затем в меню File (Файл) выберите команду Properties (Свойства).

Откроется диалоговое окно свойств локального диска С:.

4. Перейдите на вкладку Quota (Квота).

Заметьте: по умолчанию дисковые квоты отключены.

5. Пометьте флажок Enable Quota Management (Включить управление квотами).

6. Щелкните переключатель Limit Disk Space To (Выделять на диске не более).

7. В списке рядом с этим полем введите 10, а в поле Set Warning Level To (Порог выдачи предупреждений) – 6.

Заметьте: по умолчанию размер указывается в килобайтах.

8. Измените единицу измерения на мегабайт и щелкните кнопку Apply (Применить).

Откроется диалоговое окно Disk Quota (Дисковая квота), предупреждающее, что при включении дисковых квот в целях обновления статистики об использовании дискового пространства будет произведено сканирование тома.

9. Щелкните кнопку ОК, чтобы включить дисковые квоты.

10. Не закрывайте окно свойств диска — оно Вам понадобится.

Настройте персональную дисковую квоту для пользователя

Настройте для пользователя John Smith персональную дисковую квоту.

1. На вкладке Quota (Квота) диалогового окна свойств диска С: щелкните кнопку Quota Entries (Записи квот).

Откроется диалоговое окно Quota Entries For Local Disk (C:) [Записей квот для Локальный диск (C:)]. Обратите внимание, что в окне перечислены созданные Вами учетные записи пользователей, группы NT AUTHORITY\SYSTEM и BUILTIN\Adniinistmtors (\Администраторы). Учетные записи (Jane_Doe, Jonh_ Smith и Bob_Train) перечислены потому, что соответствующим пользователям принадлежат файлы на диске С:.

2. Дважды щелкните запись John Smith.

Откроется диалоговое окно Quota Settings For John Smith (Параметры квоты для John Smith).

3. Введите в поле Limit Disk Space To (Выделять на диске не более) 20, а в поле Set Warning

Level To (Порог выдачи предупреждений) – 16.

4. Щелкните кнопку ОК, чтобы вернуться к диалоговому окну Quota Entries For Local Disk (C:) [Записей квот для Локальный диск (C:)].

5. Закройте диалоговое окно Quota Entries For Local Disk (C:).

6. Не закрывайте окно свойств диска – оно Вам понадобится.

Отключите управление дисковыми квотами

Отключите управление дисковыми квотами для диска С:.

1. На вкладке Quota (Квота) сбросьте флажок Enable Quota Management (Включить управление квотами). Параметры квот для диска С: станут недоступны.

2. Щелкните кнопку Apply (Применить).

Диалоговое окно Disk Quota (Дисковая квота) предупредит, что при повторной активации системы дисковых квот будет произведено сканирование тома.

3. Щелкните кнопку ОК, чтобы закрыть диалоговое окно Disk Quota.

4. Щелкните кнопку ОК, чтобы закрыть диалоговое окно Local Disk (C:) Properties.

5. Закройте окно My Computer (Мой компьютер).

Задание 6. Восстановление данных

Вы удалите папку Inetpub, а затем восстановите ее. Выполните упражнение на компьютере ServerOl. Удалите важные данные

Вы преднамеренно удалите файл Booi.ini. Обычно удаление системных файлов происходит ошибочно или в результате сбоя аппаратуры.

1. Щелкните дважды значок My Computer (Мой компьютер), а затем – Local Disk (C:)[Локальный диск (C:)].

Откроется одноименное окно.

2. Разверните окно на весь экран.

3. В меню Tools (Сервис) выберите команду Folder Options (Свойства папки).

Откроется одноименное окно,

4. Перейдите на вкладку View (Вид).

5. Сбросьте флажок Hide Protected Operating System Files (Recommended) (Скрывать защищенные системные файлы).

6. В окне сообщения щелкните кнопку Yes (Да).

7. В окне Folder Options щелкните кнопку ОК.

Количество отображаемых в окне Local Disk (C:) файлов увеличится.

8. Щелкните файл boot.ini.

9. В меню File (Файл) выберите команду Delete (Удалить).

10. В окне Confirm File Delete (Подтверждение удаления файла) щелкните кнопку Yes (Да).

Файл boot.ini будет удален. Хотя его еще можно восстановить из корзины, мы помним, что в упражнении] была сделана резервная копия этого файла. На следующем этапе используем программу восстановления данных.

Вы восстановите файл Boot.ini из архивного набора.

1. В окне Local Disk (C:) дважды щелкните файл Backupl.bkf.

Откроется диалоговое окно Backup – [Untitled] (Архивация – (Безымянный]).

2. Щелкните кнопку Restore Wizard (Мастер восстановления).

Откроется окно Welcome To The Restore Wizard (Мастер восстановления).

3. Щелкните кнопку Next (Далее).

Откроется окна What To Restore (Что следует восстановить), где Вам предлагается выбрать носитель архива, с которого будут восстановлены файлы. Заметьте: в данном случае Вам доступен один тип носителя информации – файл и что файлы архивов отсортированы по именам.

4. В окне What To Restore раскройте узел первого задания архивации, созданного в упражнении 1.

Заметьте: первой папкой в файле резервной копии является диск С: Утилита Backup создает отдельные архивные наборы для каждого сохраняемого тома. Все файлы и папки с одного и того же устройства обозначаются соответствующей тому буквой.

5. Раскройте узел диска С.

Откроется диалоговое окно Backup File Name (Имя архивного набора), В поле Catalog Backup File (Каталогизировать архивный файл) будет выведено C:\Backupl.bkf. Если там будет текст C:\Backup2.bkf, измените имя на C:\Backupl.bkf.

6. Щелкните кнопку ОК.

7. После возврата в окно What To Restore (Что следует восстановить) щелкните С:. В колонке Name (Имя) появится файл Boot.ini.

8. Пометьте флажком Boot.ini и щелкните кнопку Next (Далее).

Откроется окно Completing The Restore Wizard (Завершение работы мастера восстановления).

9. Щелкните кнопку Advanced (Дополнительно).

Откроется окно Where To Restore (Выбор места для восстановления), где предлагается ввести путь для восстанавливаемых файлов.

10. Для просмотра параметров восстановления щелкните раскрывающийся список.

11. Проверьте правильность выбора пути для восстановления файла и щелкните кнопку Next (Далее).

12. Откроется окно How To Restore (Способ восстановления), где назначается порядок восстановления файлов с одинаковыми именами.

13. Проверьте, что выбран параметр Do Not Replace The File On My Disk (Не заменять имеющийся на диске файл) и щелкните кнопку Next (Далее).

14. Откроется окно Advanced Restore Options (Дополнительные параметры восстановления), где для задания резервного [солирования выбирают параметры безопасности.

15. Убедитесь, что помечен флажок Restore Security (Восстановление безопасности), сбросьте флажок Restore Junction Points, Not The Folders And File Data They Reference (Восстановление точек соединения, а не папок и файлов, на которые они ссылаются) и щелкните кнопку Next (Далее).

Откроется окно Completing The Restore Wizard (Завершение работы мастера восстановления), отображающее сводку выбранных Вами параметров.

16. Для начала восстановления файлов щелкните кнопку Finish (Готово).

Откроется диалоговое окно Enter Backup File Name (Ввод имени архивного файла), где при необходимости указывают имя архивного файла, содержащего восстанавливаемые файлы и папки.

17. Убедитесь, что в поле Restore From Backup File (Восстанавливать из архивного файла) введено C:\Backupl.bkf, и щелкните кнопку ОК.

Откроется диалоговое окно Restore Progress (Ход восстановления), где отображается информация о состоянии задания резервного копирования, при-

мерном суммарном размере и текущем размере обработанных данных, текущем времени выполнения задания и времени, оставшемся до его завершения.

18. По завершении восстановления щелкните кнопку Report (Отчет).

Запустится программа Notepad со сформированным отчетом о восстановлении. В журнал резервного копирования добавятся подробные сведения о произведенном восстановлении. В журнале централизованно хранится вся информация об операциях резервного копирования и восстановления.

19. После просмотра отчета закройте программу Notepad.

20. В диалоговом окне Restore Progress (Ход восстановления) щелкните кнопку Close (Закрыть).

21. Закройте диалоговое Backup – [U:ititled].

Откроется окно Local Disk (C;).

22. Удостоверьтесь, что файл Boot.ini был успешно восстановлен.

23. Закройте окно Local Disk (C:).

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы

1. Назовите основные источники уязвимостей ОС.
2. Для чего в реальных системах может понадобиться монитор запускаемых процессов?
3. Каким образом приложение может изменить привилегии и установки безопасности порождаемых процессов?
4. Можно ли штатными средствами ОС Windows NT изменить контекст безопасности запускаемого процесса, нити?
5. Назовите главное отличие ОС Windows NT от семейства 9x, проявляющееся при выделении ресурсов.
6. Чем отличается сервис от обычной программы?
7. Каким способом можно запустить собственную программу в начале загрузки Windows.

Лабораторная работа № 11.
ОС семейства UNIX. Работа с файлами и каталогами. Управление пользователями. Защита файлов. Резервное копирование данных

Цель: изучить основные команды ОС для работы с файлами и каталогами, получить практические навыки администрирования системы.

Программное обеспечение: ОС Kali Linux

1.Теоретический материал

ОС семейства UNIX поддерживает многочисленные утилиты, позволяющие работать с файловой системой и доступные как команды командного интерпретатора. Рассмотрим наиболее распространенные команды работы с файлами и каталогами (табл. 1).

Ядро. Выполняет функции управления памятью, процессорами. Осуществляет диспетчеризацию выполнения всех программ и обслуживание внешних устройств. Все действия, связанные с вводом/выводом и выполнением системных операций, выполняются с помощью системных вызовов. Системные вызовы реализуют программный интерфейс между программами и ядром. Имеется возможность динамического конфигурирования ядра.

Диспетчер процессов Init. Активизирует процессы, необходимые для нормальной работы системы и производит их начальную инициализацию. Обеспечивает завершение работы системы, организует сеансы работы пользователей, в том числе, для удаленных терминалов.

Интерпретатор команд Shell. Анализирует команды, вводимые с терминала либо из командного файла, и передает их для выполнения в ядро системы. Команды обычно имеют аргументы и параметры, которые обеспечивают модернизацию выполняемых действий. Shell является также языком программирования, на котором можно создавать командные файлы (shell-файлы). При входе в ОС пользователь получает копию интерпретатора **Shell** в качестве родительского процесса. Далее, после ввода команды пользователем создается порожденный процесс, называемый процессом-потомком. То есть после запуска ОС каждый новый процесс функционирует только как процесс – потомок уже существующего процесса. В ОС Linux имеется возможность динамического порождения и управления процессами.

Обязательным в системе является интерпретатор Bash, полностью соответствующий стандарту POSIX. В качестве Shell может быть использована оболочка **mc** с интерфейсом, подобным Norton Commander.

Сетевой графический интерфейс X-сервер (X-Windows). Обеспечивает поддержку графических оболочек.

Графические оболочки KDE, Gnome. Отличительными свойствами KDE являются: минимальные требования к аппаратуре, высокая надежность, интернационализация. Базовые библиотеки KDE (qt, kde-libs) признаны одними из лучших продуктов по созданию графического интерфейса, обеспечивают простое написание программ с использованием передовых технологий. Gnome

имеет развитые графические возможности, но более требователен к аппаратным средствам.

Сетевая поддержка NFS, SMB, TCP/IP. NFS – программный комплекс PC-NFS (Network File System) для выполнения сетевых функций. PC-NFS ориентирован для конкретной ОС персонального компьютера (PC) и включает драйверы для работы в сети и дополнительные утилиты. SMB – сетевая файловая система, совместимая с Windows NT. TCP/IP – протокол контроля передачи данных (Transfer Control Protocol/Internet Protocol). Сеть по протоколам TCP/IP является неотъемлемой частью ОС семейства UNIX. Поддерживаются любые сети, от локальных до Internet, с использованием только встроенных сетевых средств.

Инструментальные средства программирования. Основой средств программирования является компилятор CC или GCC для языков C и C++; модули поддержки других языков программирования (Obective C, Фортран, Паскаль, Modula-3, Ада, Java и др.); интегрированные среды и средства визуального проектирования: Kdevelop, Xwpe; средства адаптации привязки программ AUTOCONFIG, AUTOMAKE.

Таблица 6. Основные команды для работы с файлами и каталогами

Выполняемая функция	**Наименование команды**	**Спецификация команды**	**Пример использования**
Просмотр содержимого каталога	ls	ls [ключи] [имя_каталога]	ls –l rabota
Создание каталога	mkdir	mkdir [ключи] имя_каталога	mkdir rabota
Перемещение в каталоговой структуре	cd	cd имя_каталога	cd rabota
Удаление каталога	rmdir	rmdir [ключи] имя_каталога	rmdir rabota
Создание файла	touch	touch имя_файла	cd /home touch primer1
Просмотр файла	more less cat	more имя_файла	more primer Примечание: Для выхода из режима просмотра нажмите клавишу q.
Счетчик строк в файле	wc	wc [ключи] имя_файла	wc primer
Просмотр типа файла	file	file имя_файла	file primer Примечание На экране по-

Выполняемая функция	Наименование команды	Спецификация команды	Пример использования
			явится сообщение: primer: ASCII text, т.е. файл текстовый
Копирование файла в каталог	cp	cp имя_файла_источника имя_принимающего_каталога	cp primer /home/rabota
Копирование файла в файл	cp	cp имя_файла_источника имя_принимающего_файла	cp primer1 primer2 Примечание. Имя файла-получателя затирается информацией файла-источника
Копирование каталоговой структуры	cp	cp -r имя_исходного_каталога имя_принимающего_каталога	cp -r 1 2
Переименование файла	mv	mv старое_имя_файла новое_имя_файла	mv primer pr
Перемещение файла	mv	mv имя_перемещаемого_файла имя_принимающего_каталога	mv primer /
Удаление файла	rm	rm имя_файла	rm pr
Определение месторасположения	pwd	pwd	pwd
Останов системы	shutdown	shutdown [ключи]	shutdown –h now
Перезагрузка системы	shutdown	shutdown [ключи]	shutdown –r now
Вызов справки	man info	man <имя_команды>	man ls

Пояснения к некоторым командам, представленным в таблице 6.

Выполнение простых команд

Формат команд в ОС LINUX следующий:

имя команды [аргументы] [параметры] [метасимволы]

Имя команды может содержать любое допустимое имя файла; аргументы – одна или несколько букв со знаком минус (-); параметры – передаваемые значения для обработки; метасимволы интерпретируются как специальные операции. В квадратных скобках указываются необязательные части команд.

Введите команду **echo**, которая выдает на экран свои параметры:

echo good morning

и нажмите клавишу *Enter*. На экране появится приветствие «*good morning*» – параметр команды **echo.** Командный интерпретатор *shell* вызвал команду **echo**, реализованную в виде программы на языке СИ, и передал ей параметры. После этого интерпретатор команд вывел знак-приглашение. Синтаксис команды **echo**:

echo [-n] [arg1] [arg2] [arg3]...

Команда помещает в стандартный вывод свои параметры, разделенные пробелами и завершаемые символом перевода строки. При наличии флага *-n* символ перевода строки исключается.

who [am i] – получение информации о работающих пользователях.

В квадратных скобках указываются параметры команды, которые можно опустить. Ответ представляется в виде таблицы, которая содержит следующую информацию:

– идентификатор пользователя;

– идентификатор терминала;

– дата подключения;

– время подключения.

Команда ls

Команда ls выводит содержание каталога. Она может иметь параметры, которые влияют на ее выполнение.

– a – выводит все файлы, включая скрытые;

– C – выводит в несколько колонок;

– F – показывает тип файла;

– l – выводит в длинном формате;

Если выведенный файл помечен звездочкой, то такой файл является исполняемым. Если выведенный файл помечен слешем, то это каталоги. Скрытые файлы помечены точкой. Опция –l выводит файлы и каталоги в длинном формате. Этот формат показывает пользователю большую часть необходимой информации.

Команда cd

Пользователь может обращаться к файлам и каталогам с помощью абсолютных и относительных имен маршрутов. Абсолютные имена всегда начинаются со слеша. **Пример** команды cd с абсолютным именем маршрута: cd /home/rabota. В данном примере становится текущим подкаталог rabota каталога

home. **Пример** команды cd с относительным именем маршрута: cd rabota. cd ./ rabota (точка – это относительное имя, оно относится к текущему каталогу). Переход в родительский каталог – cd .

date – вывод на экран текущей даты и текущего времени.

cal [[месяц]год] – календарь; если календарь не помещается на одном экране, то используется команда **cal год | more** и клавишей пробела производится постраничный вывод информации.

man <название команды> – вызов электронного справочника об указанной команде. Выход из справочника – нажатие клавиши Q.

Команда **man man** сообщает информацию о том, как пользоваться справочником.

tty – сообщение имени специального файла стандартного вывода, соответствующего терминалу пользователя.

cat <имя файла> – вывод содержимого файла на экран. Команда **cat > text.1** создает новый файл с именем text.1, который можно заполнить символьными строками, вводя их с клавиатуры. Нажатие клавиши *Enter* создает новую строку. Завершение ввода – нажатие *Ctrl - d*. Команда **cat text.1 > text.2** пересылает содержимое файла text.1 в файл text.2. Слияние файлов осуществляется командой **cat text.1 text.2 > text.3.**

ls [-alrstu] [имя] – вывод содержимого каталога на экран. Если параметр не указан, выдается содержимое текущего каталога.

Аргументы команды:

– a – выводит список всех файлов и каталогов, в том числе и скрытых;

– l – выводит список файлов в расширенном формате, показывая тип каждого элемента, полномочия, владельца, размер и дату последней модификации;

– r – выводит список в порядке, обратном заданному;

– s – выводит размеры каждого файла;

– t – перечисляет файлы и каталоги в соответствии с датой их последней модификации;

– u – перечисляет файлы и каталоги в порядке, обратном их последней модификации.

rm <имя файла> – удаление файла (файлов). Команда **rm text.1 text.2 text.3** удаляет файлы text.1, text.2, text.3. Другие варианты этой команды – **rm text.[123]** или **rm text.[1-3].**

wc [имя файла] – вывод числа строк, слов и символов в файле.

clear – очистка экрана.

Группирование команд

Группы команд или сложные команды могут формироваться с помощью специальных символов (метасимволов):

& – процесс выполняется в фоновом режиме, не дожидаясь окончания предыдущих процессов;

? – шаблон, распространяется только на один символ;

* – шаблон, распространяется на все оставшиеся символы;

| – программный канал – стандартный вывод одного процесса является стандартным вводом другого;

> – переадресация вывода в файл;

< – переадресация ввода из файла;

; – если в списке команд команды отделяются друг от друга точкой с запятой, то они выполняются друг за другом;

&& – эта конструкция между командами означает, что последующая команда выполняется только при нормальном завершении предыдущей команды (код возврата 0);

|| – последующая команда выполняется только, если не выполнилась предыдущая команда (код возврата 1);

() – группирование команд в скобки;

{ } – группирование команд с объединенным выводом;

[] – указание диапазона или явное перечисление (без запятых);

>> – добавление содержимого файла в конец другого файла.

Примеры.

who | wc – подсчет количества работающих пользователей командой **wc** (word count – счет слов);

cat text.1 > text.2 – содержимое файла text.1 пересылается в файл text.2;

mail student < file.txt – электронная почта передает файл file.txt всем пользователям, перечисленным в командной строке;

cat text.1,text.2 – просматриваются файлы text.1 и text.2;

cat text.1 >> text.2 – добавление файла text.1 в конец файла text.2;

cc primer.c & – трансляция Си – программы в фоновом режиме. Имя выполняемой программы по умолчанию a.out.

cc -o primer.o primer.c – трансляция Си-программы с образованием файла выполняемой программы с именем primer.o;

rm text.* – удаление всех файлов с именем text;

{cat text.1; cat text.2} | lpr – просмотр файлов text.1 и text.2 и вывод их на печать;

ps [-al] [number] – команда для вывода информации о процессах:

-a – вывод информации обо всех активных процессах, запущенных с вашего терминала;

-l – полная информация о процессах;

number – номер процесса.

Команда **ps** без параметров выводит информацию только об активных процессах, запущенных с данного терминала, в том числе и фоновых. На экран выводится подробная информация обо всех активных процессах в следующей форме:

```
F  S  UID  PID  PPID  C  PRI  NI  ADDR  SZ  WCHAN  TTY  TIME  CMD

1  S  200  210  7     0  2    20  80    30  703a   03   0:07  cc
1  R  12   419  7     11 5    20  56    20         03   0:12  ps
```

F – флаг процесса (1 – в оперативной памяти, 2 – системный процесс, 4 – заблокирован в ОЗУ, 20 – находится под управлением другого процесса, 10 – подвергнут свопингу);

S – состояние процесса (O – выполняется процессором , S – задержан, R – готов к выполнению, I – создается);

UID – идентификатор пользователя;

PID – идентификатор процесса;

PPID – номер родительского процесса;

C – степень загруженности процессора;

PRI – приоритет процесса, вычисляется по значению переменной NICE и чем больше число, тем меньше его приоритет;

NI – значение переменной NICE для вычисления динамического приоритета, принимает величины от 0 до 39;

ADDR – адрес процесса в памяти;

SZ – объем ОЗУ, занимаемый процессом;

WCHAN – имя события, до которого процесс задержан, для активного процесса – пробел;

TTY – номер управляющего терминала для процесса;

TIME – время выполнения процесса;

CMD – команда, которая породила процесс.

nice [-приращение приоритета] команда[аргументы] – команда изменения приоритета. Каждое запущенное задание (процесс) имеет номер приоритета в диапазоне от 0 до 39, на основе которого ядро вычисляет фактический приоритет, используемый для планирования процесса. Значение 0 представляет наивысший приоритет, а 39 – самый низший. Увеличение номера приоритета приводит к понижению приоритета, присвоенного процессу. Команда **nice -10 ls -l** увеличивает номер приоритета, присвоенный процессу **ls -l** на 10.

renice 5 1836 – команда устанавливает значение номера приоритета процесса с идентификатором 1836 равным 5. Увеличить приоритет процесса может только администратор системы.

kill [-sig] <идентификатор процесса> – прекращение процесса до его программного завершения. Sig – номер сигнала. Sig = -15 означает программное (нормальное) завершение процесса, номер сигнала = -9 – уничтожение процесса. По умолчанию sig= -9. Вывести себя из системы можно командой kill -9 0. Пользователь с низким приоритетом может прервать процессы, связанные только с его терминалом.

mc – вызов файлового менеджера (программы – оболочки) *Midnight Commander*, аналогичного *Norton Commander*.

sort [-dr] – сортировка входных файлов и вывод результата на экран.

4. Краткое описание командного интерпретатора Shell

Интерпретатор команд **Shell** анализирует команды, вводимые с терминала либо из командного файла, и передает их для выполнения в ядро системы. Команды обычно имеют аргументы и параметры, которые обеспечивают модернизацию выполняемых действий. **Shell** является также языком программирования, на котором можно создавать командные файлы (shell-файлы). При входе в ОС пользователь получает копию интерпретатора **Shell** в качестве родительского процесса. Далее, после ввода команды пользователем создается порожденный процесс, называемый процессом-потомком. То есть после запуска ОС каждый новый процесс функционирует только как процесс – потомок уже существующего процесса. В ОС Linux имеется возможность динамического порождения и управления процессами.

Обязательным в системе является интерпретатор **Bash**, полностью соответствующий стандарту POSIX. В качестве **Shell** может быть использована оболочка **mc** с интерфейсом, подобным Norton Commander.

2.Задание к лабораторной работе

1.Задание А. Освоить использование оснвных команда ОС Linux**.**

№ п/п	Задание
1.	Войти в систему
2.	Определить свое месторасположение
3.	Перейти в корневой каталог
4.	Просмотреть содержимое корневого каталога
5.	Просмотреть содержимое корневого каталога в длинном формате
6.	Просмотреть содержимое каталога /etc
7.	Просмотреть содержимое каталога /etc в длинном формате
8.	Просмотреть содержимое каталога /home
9.	Переместиться в каталог /home
10.	Определить свое месторасположение
11.	Создать следующую структуру каталогов:
	/home/my/vhod;
	/home/my/vihod.
12.	Создать в каталоге /home/my/vhod несколько пустых файлов
13.	Просмотреть типы файлов каталога /home/my/vhod
14.	Переименовать один из файлов
15.	Переместить этот файл в каталог /home/my/vihod
16.	Создать в корневом каталоге каталог /test
17.	Скопировать файл каталога /home/my/vihod в каталог /test
18.	Скопировать файлы каталога /home/my/vhod с копированием каталогов структуры в каталог /test
19.	Просмотреть содержимое файла /etc/group. Подсчитать число групп пользователей в системе

№ п/п	Задание
20.	Перенаправить информацию файла /etc/passwd в файл каталога /home/my/vhod
21.	Просмотреть содержимое этого файла
22.	Предъявить работу преподавателю
23.	Удалить свои файлы и каталоги (используйте символы-заместители)

2. Задание Б. Получить информацию о работающих пользователях, подсчитать их количество и запомнить в файле**:**

1. Ознакомиться с теоретической частью к лабораторной работе.
2. Определить день недели, в который Вы родились.
3. Получить подробную информацию обо всех активных процессах.
4. Используя редактор VI (см. приложение), создать два текстовых файла (с расширением TXT) и командой CAT просмотреть их на экране.
5. Получить информацию о работающих пользователях, подсчитать их количество и запомнить в файле.
6. Объединить текстовые файлы в единый файл и посмотреть его на экране.
7. Посмотреть приоритет своего процесса и уменьшить скорость его выполнение за счет повышения номера приоритета.
8. Используя редактор VI, написать программу на языке Си и запустить ее на трансляцию в фоновом режиме.
9. Показать преподавателю исходный текст программы на языке Си, текстовый файл, файл с сохранением количества пользователей.
10. Продемонстрировать выполнение Си – программы.
11. Удалить свои файлы и выйти из системы.

3. Задание В. Ознакомиться с файловой структурой ОС LINUX. Изучить команды работы с файлами**:**

1. Ознакомиться с файловой структурой ОС LINUX. Изучить команды работы с файлами.
2. Используя команды ОС LINUX, создать два текстовых файла.
3. Полученные файлы объединить в один файл и его содержимое просмотреть на экране.
4. Создать новую директорию и переместить в нее полученные файлы.
5. Вывести полную информацию обо всех файлах и проанализировать уровни доступа.
6. Добавить для всех трех файлов право выполнения членам группы и остальным пользователям.
7. Просмотреть атрибуты файлов.
8. Создать еще один каталог.
9. Установить дополнительную связь объединенного файла с новым каталогом, но под другим именем.

10. Создать символическую связь.

11. Сделать текущим новый каталог и вывести на экран расширенный список информации о его файлах.

12. Произвести поиск заданной последовательности символов в файлах текущей директории и получить перечень соответствующих файлов.

13. Получить информацию об активных процессах и имена других пользователей.

14. Сдать отчет о работе и удалить свои файлы и каталоги.

15. Выйти из системы.

4. Задание Г. Изучите работу с пользователями:

1. Создать два пользователя «students и prepodavateli».
2. Задать этим пользователям пароли.
3. Создать две группы «SAIT и SAPR».
4. Добавить пользователей в обе группы.
5. Проверить наличие этих пользователей в группах, путем вывода всех пользователей состоящих в каждой группе.
6. Удалить пользователя «students» из группы «SAIT».
7. Создать каталог «newfolder».
8. Вывести информацию о каталоге, а именно о правах доступа к нему.
9. Изменить правообладателя каталога «newfolder» на «students», и убедится в этом.
10. Удалить каталог и раннее созданные группы и пользователей.
11. Используя первое терминальное устройство, войти в систему суперпользователем.
12. Создать группу.
13. Создать первого пользователя, принадлежащего этой группе.
14. Создать второго пользователя, принадлежащего этой группе.
15. Переключиться на второе терминальное устройство. Используя команду login, войти в систему под именем одного из созданных пользователей.
16. Создать несколько текстовых файлов в своем каталоге (использовать команды: cat, touch, vi).
17. Переключиться на первое терминальное устройство. Заблокировать вход второму пользователю.
18. Переключиться на третье терминальное устройство. Попытаться войти в систему под именем заблокированного пользователя.
19. Переключиться на первое терминальное устройство. Изменить пароль первому пользователю. Разблокировать второго пользователя.
20. Переключиться на второе терминальное устройство. Войти в систему под именем пользователя, у которого изменен пароль.
21. Переключиться на третье терминальное устройство. Войти в систему под именем разблокированного пользователя.
22. Создать (второму пользователю) несколько текстовых файлов.
23. Определить их пользовательскую и групповую принадлежность.

24. Попытаться отредактировать файлы, принадлежащие первому (другому) пользователю.

25. Попытаться удалить файлы, принадлежащие первому (другому) пользователю.

26. Переключиться на первое терминальное устройство. Определить активных пользователей в системе.

27. Отсортировать файл /etc/passwd, сохранив отсортированную информацию в файле /etc/new_passwd. Просмотреть содержимое этого файла. Отобразить имена пользователей с именами от a до k.

28. Отработать команду finger.

29. Переключиться на второе терминальное устройство. Просмотреть содержимое своего каталога в длинном формате.

30. Изменить принадлежность файлов другому пользователю.

31. Переключиться на третье терминальное устройство. Попытаться отредактировать файлы, принадлежащие ранее первому (другому) пользователю.

32. Предъявить работу преподавателю.

33. Удалить созданные файлы.

34. Найти строку (указать имя пользователя) в файле /etc/passwd. Удалить одного из созданных пользователей (удалив учетную запись в файле).

4. Задание Д. Резервное копирование данных

1. Создать командой tar архивный файл текущего каталога. Архивный файл поместить в текущий каталог.

2. Просмотреть содержание архивного файла.

3. Просмотреть содержимое текущего каталога в длинном формате с указанием типа файлов.

4. Определить размер созданного архивного файла.

5. Используя команду gzip, сжать данные архивного файла.

6. Определить размер созданного архива.

7. Подмонтировать дискету.

8. Скопировать созданный архив на дискету/usb-хранилище.

9. Просмотреть содержимое дискет/usb-хранилище

10. Отмонтировать дискету. /usb-хранилище.

11. Найти в каталогах архив небольшого размера с расширением *.tar.gz

12. Определить размер найденного архива.

13. Просмотреть содержимое этого архива.

14. Распаковать данные, поместив их в каталог /home/my/bin

15. Определить размер распакованных данных.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного

материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы

1. Как осуществить вход в систему?
2. Как корректно выйти из системы?
3. Как создать каталог? Как создать подкаталог?
4. Как осуществить перемещение в каталоговой структуре?
5. Какая команда используется для удаления каталогов?
6. Как создать файл, как просмотреть его содержимое? Как просмотреть тип файла, его размер, количество строк, столбцов.
7. Назовите команду, с помощью которой можно копировать файл в файл, файл в каталог, структуру файлов и каталогов в каталог.
8. Как осуществить перемещение файла из каталога в каталог?
9. Какая команда используется для переименования файла? Как удалить файл, каталог?
10. Как определить свое месторасположение в системе?
11. Какие способы запуска справочной системы Вы знаете?

Лабораторная работа № 12
Работа с процессами в операционной системе LINUX

Цель работы: целью работы является изучение методов программирования по созданию пользовательских процессов в ОС Linux**.**

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2, ОС Kali Linux

1. Теоретическая часть

Идентификаторы процессов. Каждый процесс в Linux помечается уникальным идентификатором (PID, process identifer). Идентификаторы – это 16-разрядные числа, назначаемые последовательно по мере создания процессов. У всякого процесса имеется также родительский процесс за исключением специального суперсервера init с идентификатором 1. Таким образом, все процессы Linux организованы в виде сложной иерархии, на вершине которой находится процесс **init**. Иерархию процессов можно увидеть, выполнив команду **ps – axf** . К атрибутам процесса относится идентификатор его родительского процесса (PPID, parent process identifer).

Работая с идентификаторами процессов в программах, написанных на языках C и C++, следует объявить соответствующие переменные как имеющие тип **pid_t** (этот тип определяется в файле **<sys/types.h>)**. Программа может узнать идентификатор своего собственного процесса с помощью системного вызова **getpid()**, а идентификатор своего родительского процесса с помощью системного вызова **getppid()**.

Пример 1:

```
#include<stdio.h>
#include<unistd.h>
int main()
{
printf (“Номер процесса: %d\n”, (int) getpid() );
printf («Номер родительского процесса: %d\n», (int) getppid());
return 0;
}
```

Обратите внимание на важную особенность: при каждом вызове программа сообщает о разных идентификаторах, поскольку всякий раз запускается новый процесс. Тем не менее, если программа вызывается из одного и того же интерпретатора команд, то родительский идентификатор оказывается одинаковым.

Создание процессов

Существуют два способа создания процессов. Первый из них относительно прост, применяется редко, поскольку неэффективен и связан со значительным риском для безопасности системы. Второй способ сложнее, но избавлен от

недостатков первого. Первый способ основан на применении функции **system()**, а второй на основе применения функций **fork()** и **exec()**.

Функция system()

Функция **system()** определена в стандартной библиотеке языка С и позволяет вызвать из программы системную команду, как если бы она была набрана в командной строке.

По сути, эта функция запускает стандартный интерпретатор и передает ему команду на выполнение.

Пример 2:

```
#include<stdlib.h>
int main()
{
int return_value;
return_value = system("ls –l /");
return return_value;
}
```

Функция **system()** возвращает код завершения указной команды. Если интерпретатор не может быть запущен, возвращается значения 127, а в случае возникновения других ошибок (-1).

Поскольку функция **system()** запускает интерпретатор команд, она подвержена всем тем ограничениям безопасности, что и командный интерпретатор.

В большинстве Unix-системах программа /bin/sh представляет собой символическую ссылку на другой интерпретатор. В Linux это в основном bash.

Вызов из функции **system()** программы с привилегиями пользователя root также может иметь неодинаковые последствия в разных системах. Таким образом, лучше создавать процессы с помощью функций **fork()** и **exec()**.

Функции fork() и exec()

В DOS и Windows API имеется семейство функций **spawn()**. Они принимают в качестве аргумента имя программы, создают новый экземпляр ее процесса и запускают его.

В Linux нет такой функции, которая выполнила бы все это за одни заход. Вместо этого имеются функция **fork()**, создающая дочерний процесс, который является точной копией родительского процесса, и семейство функций **exec()**, заставляющих требуемый процесс перестать быть вторым экземпляром одной программы и превратиться в экземпляр другой программы.

Чтобы создать новый процесс, нужно сначала с помощью функции **fork()** создать копию текущего процесса, а затем с помощью функции **exec()** преобразовать одну из копий в экземпляр запускаемой программы.

Вызов функции fork()

Вызывая функцию **fork()**, программа создает свой дубликат, называемый дочерним процессом. Родительский процесс продолжает выполнять программу

с той точки, где была вызвана функция **fork()**. То же самое делает и дочерний процесс.

Процессы отличаются своими идентификаторами. Таким образом, программа может вызывать функцию **getpid()** и узнать где именно она находится.

Но сама функция **fork()** реализует другой способ: она возвращает разные значения в родительском и дочернем процессах. В родительском процессе функция **fork()** равна идентификатору своего потомка, а в дочернем процессе она равна 0. Рассмотрим данную ситуацию на примере, учтите, что первая часть инструкции **if** выполняется только в родительском процессе, тогда как ветвь **else** – только в дочернем.

Пример 3:

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
int main()
{
pid_t child_pid;
printf("ID процесса основной программы: %d\n", (int) getpid() );
child_pid = fork();
if (child_pid)
{
printf(«Это родительский процесс, с ID %d\n», (int) getpid() );
printf("Дочерний процесс, с ID %d\n", (int) child_pid );
}
else
printf(«Дочерний процесс с ID %d\n», (int) getpid() );
return 0;
}
```

Семейство функций exec()

Функции семейства **exec()** заменяют программу, выполняющуюся в текущем процессе, другой программой. Когда программа вызывает функцию **exec()**, ее выполнение немедленно прекращается и начинает работу новая программа.

Функции, в название которых присутствует суффикс 'p' (**execvp()** и **execlp()**), принимают в качестве аргумента имя программы и ищут эту программу в каталогах, определяемых переменной среды PATH. Всем остальным функциям нужно передавать полное путевое имя программы.

Функции, в названии которых присутствует суффикс 'v' (**execv()**, **execvp()**, **execve()**), принимают список аргументов программы в виде массива строковых указателей, оканчивающегося NULL-указателем. Функции с суффиксом 'l' (**excevl()**, **execlp()**, **execlve()**), принимают список аргументов переменного размера.

Функции, в названии которых присутствует суффикс 'e' (**execve()**, **execle()**), в качестве дополнительного аргумента принимают массив переменных среды. Этот массив содержит строковые указатели и оканчивается пустым указателем. Каждая строка должна иметь вид «Переменная = значение».

Поскольку функция **exec()** заменяет одну программу другой, она никогда не возвращает значение – только если вызов программы оказался невозможен в случае ошибки.

Список аргументов, передаваемых программе, аналогичен аргументам командной строки, указываемым при запуске программы в интерактивном режиме. Их тоже можно получить с помощью параметров **argc** и **argv** функции **main()**. Когда программу запускает интерпретатор команд, первый элемент массива **argv** будет содержать имя программы, а далее будут находиться переданные программе аргументы. Аналогичным образом следует поступить, формируя список аргументов для функции **exec()**.

Совместное использование функций fork() и exec()

Стандартная методика запуска одной программы из другой такова: сначала с помощью функции **fork()** создается дочерний процесс, затем в нем вызывается функция **exec()**.

Это позволяет главной программе продолжать выполнение в родительском процессе. В качестве примера напишем программу, которая отображает корневой каталог.

Пример 4:

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>
int spawn(char* program, char** arg_list)
{
pid_t child_pid;
child_pid = fork();
if(child_pid)
return child_pid;
else
{
execvp (program, arg_list);
fprintf (stderr, “an error 152роцесс152 in execvp\n”);
abort();
}
}
int main()
{
int child_status;
char* arg_list[] = {“ls”,”-l”,”/”,NULL};
```

```
spawn (“ls”, arg_list);
wait (&child_status);
printf(“done\n”);
return 0;
}
```

Системные вызовы wait()

Самая простая функция в семействе называется **wait()**. Она блокирует вызывающий процесс до тех пор, пока один из его дочерних процессов не завершиться (или не произойдет ошибка).

Пример использования данной функции приведен выше.

Функция **waitpid()** позволяет дождаться завершения конкретного дочернего процесса.

Функция **wait3()** возвращает информацию о статистике использования центрального процессора завершившемся дочерним 153 процесссом.

Функция **wait4()** позволяет задать дополнительную информацию о том, каких процессов следует дождаться.

Краткое описание языка программирования Си

Си – универсальный язык программирования. Он тесно связан с системой UNIX, так как был разработан в этой системе, которая, как и большинство программ работающих в ней, написаны на Си.

В отличие от «безтиповых» языков Си обеспечивает разнообразие типов данных. Базовыми типами являются символы, а также целые и числа с плавающей точкой различных размеров. Кроме того, имеется возможность получать целую иерархию производных типов данных из указателей, массивов, структур и объединений. Выражения формируются из операторов и операндов. Любое выражение, включая присваивание и вызов функции, может быть инструкцией. Указатели обеспечивают машинно-независимую адресную арифметику. В Си имеются основные управляющие конструкции, используемые в хорошо структурированных программах: составная инструкция ({...}), ветвление по условию (if-else), выбор одной альтернативы из многих (switch), циклы с проверкой наверху (while, for) и с проверкой внизу (do), а также средство прерывания цикла (break). В качестве результата функции могут возвращать значения базовых типов, структур, объединений и указателей. Любая функция допускает рекурсивное обращение к себе. Функции программы на Си могут храниться в отдельных исходных файлах и компилироваться независимо. Переменные по отношению к функции могут быть внутренними и внешними. Последние могут быть доступными в пределах одного исходного файла или всей программы. Си – язык сравнительно «низкого уровня». Однако это вовсе не умаляет его достоинств, просто Си имеет дело с теми же объектами, что и большинство компьютеров, т. е. с символами, числами и адресами. С ними, можно оперировать при помощи арифметических и логических операций, выполняемых реальными машинами. В Си нет прямых операций над составными объектами, такими как

строки символов, множества, списки и массивы. В нем нет операций, которые бы манипулировали с целыми массивами или строками символов, хотя структуры разрешается копировать целиком как единые объекты. В языке нет каких-либо средств распределения памяти, помимо возможности определения статических переменных и стекового механизма при выделении места для локальных переменных внутри функций. Наконец, в самом Си нет средств ввода-вывода, инструкций READ (читать) и WRITE (писать) и каких-либо методов доступа к файлам. Все это – механизмы высокого уровня, которые в Си обеспечиваются исключительно с помощью явно вызываемых функций. Большинство реализованных Си-систем содержат в себе разумный стандартный набор этих функций. Си предоставляет средства лишь последовательного управления ходом вычислений: механизм ветвления по условиям, циклы, составные инструкции, подпрограммы – и не содержит средств мультипрограммирования, параллельных процессов, синхронизации и организации сопрограмм. Однако компактность языка имеет реальные выгоды. Поскольку Си относительно мал, то и описание его кратко, и овладеть им можно быстро. Программист может реально рассчитывать на то, что он будет знать, понимать и на практике регулярно пользоваться всеми возможностями языка. Важный аспект языка – это определение библиотеки, поставляемой вместе с Си-компилятором, в которой специфицируются функции доступа к возможностям операционной системы (например, чтения-записи файлов), форматного ввода-вывода, динамического выделения памяти, манипуляций со строками символов и т. д. Набор стандартных заголовочных файлов обеспечивает единообразный доступ к объявлениям функций и типов данных. Почти все программы, написанные на Си, если они не касаются каких-либо скрытых в операционной системе деталей, переносимы на другие машины. Си соответствует аппаратным возможностям многих машин, однако он не привязан к архитектуре какой-либо конкретной машины. Основной философией Си остается то, что программисты сами знают, что делают; язык лишь требует явного указания об их намерениях. Си, как и любой другой язык программирования, не свободен от недостатков. Тем не менее, как оказалось, Си – чрезвычайно эффективный и выразительный язык, пригодный для широкого класса задач.

2. Задание на лабораторную работу

С помощью компилятора С создать и выполнить программы, исходный текст которых приведен в примерах 1–4.

3. Методика выполнения задания

Порядок выполнения работы:

1. Прочитать методический материал.
2. Изучить характеристики и синтаксис функций и системных вызовов.
3. Набрать код примеров в текстовые файлы и произвести компиляцию программ.

4. Проверить работоспособность программ.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы:

1. Что является атрибутами процесса?
2. Как организуется взаимодействие процессов?
3. Каким образом программные средства Linux позволяют динамически порождать процессы?
4. Какие существуют формы системного вызова exec()?

Лабораторная работа № 13
Особенности ОС Linux

Цель работы: целью работы является изучение методов программирования по работе с файлами через системные вызовы и стандартную библиотеку ввода-вывода для языка С**.**

Программное обеспечение: ОС Microsoft Windows XP/2003/ 2008R2/Vista/7/8/8.1/2012/2012R2, ОС Kali Linux

1. Теоретическая часть

Среди всех категорий средств коммуникации наиболее употребительными являются каналы связи, обеспечивающие достаточно безопасное и достаточно информативное взаимодействие процессов.

Существует две модели передачи данных по каналам связи – поток ввода-вывода и сообщения. Из них более простой является потоковая модель, в которой операции передачи/приема информации вообще не интересуются содержимым того, что передается или принимается. Вся информация в канале связи рассматривается как непрерывный поток байт, не обладающий никакой внутренней структурой.

Потоковая передача информации может осуществляться не только между процессами, но и между процессом и устройством ввода-вывода, например между процессом и диском, на котором данные представляются в виде файла.

Функции работы с файлами из стандартной библиотеки ввода-вывода, такие как *fopen(), fread(), fwrite(), fprintf(), fscanf(), fgets()* и т. д. входят как неотъемлемая часть в стандарт ANSI на язык С. Они позволяют программисту получать информацию из файла или записывать ее в файл при условии, что программист обладает определенными знаниями о содержимом передаваемых данных. Так, например, функция *fgets()* используется для ввода из файла последовательности символов, заканчивающейся символом «\n» – перевод каретки. Функция *fscanf()* производит ввод информации, соответствующей заданному формату, и т. д. С точки зрения потоковой модели операции, определяемые функциями стандартной библиотеки ввода-вывода, не являются потоковыми операциями, так как каждая из них требует наличия некоторой структуры передаваемых данных.

В операционной системе Linux эти функции представляют собой надстройку – сервисный интерфейс – над системными вызовами, осуществляющими прямые потоковые операции обмена информацией между процессом и файлом и не требующими никаких знаний о том, что она содержит.

Файловый дескриптор. Информация о файлах, используемых процессом, входит в состав его системного контекста и хранится в его блоке управления – PCB. В операционной системе Linux можно упрощенно полагать, что ин-

формация о файлах, с которыми процесс осуществляет операции потокового обмена, наряду с информацией о потоковых линиях связи, соединяющих процесс с другими процессами и устройствами ввода-вывода, хранится в некотором массиве, получившем название таблицы открытых файлов или таблицы файловых дескрипторов. Индекс элемента этого массива, соответствующий определенному потоку ввода-вывода, получил название файлового дескриптора для этого потока. Таким образом, файловый дескриптор представляет собой небольшое целое неотрицательное число, которое для текущего процесса в данный момент времени однозначно определяет некоторый действующий канал ввода-вывода. Некоторые файловые дескрипторы на этапе старта любой программы ассоциируются со стандартными потоками ввода-вывода. Так, например, файловый дескриптор 0 соответствует стандартному потоку ввода, файловый дескриптор 1 – стандартному потоку вывода, файловый дескриптор 2 – стандартному потоку для вывода ошибок. В нормальном интерактивном режиме работы стандартный поток ввода связывает процесс с клавиатурой, а стандартные потоки вывода и вывода ошибок – с текущим терминалом.

Открытие файла. Системный вызов *open()*. Файловый дескриптор используется в качестве параметра, описывающего поток ввода-вывода, для системных вызовов, выполняющих операции над этим потоком. Поэтому прежде чем совершать операции чтения данных из файла и записи их в файл, необходимо поместить информацию о файле в таблицу открытых файлов и определить соответствующий файловый дескриптор. Для этого применяется процедура открытия файла, осуществляемая системным вызовом *open()*.

Прототип системного вызова:

```
#include <fcntl.h>
int open(char *path, int flags);
int open(char *path, int flags, int mode);
```

Системный вызов *open()* предназначен для выполнения операции открытия файла и, в случае ее удачного осуществления, возвращает файловый дескриптор открытого файла (небольшое неотрицательное целое число, которое используется в дальнейшем для других операций с этим файлом). Параметр *path* является указателем на строку, содержащую полное или относительное имя файла. Параметр *flags* может принимать одно из следующих трех значений:

O_RDONLY – если над файлом в дальнейшем будут совершаться только операции чтения;
O_WRONLY – если над файлом в дальнейшем будут осуществляться только операции записи;
O_RDWR – если над файлом будут осуществляться и операции чтения, и операции записи.

Каждое из этих значений может быть скомбинировано посредством операции «побитовое или (|)» с одним или несколькими флагами:
O_CREAT – если файла с указанным именем не существует, он должен быть создан;
O_EXCL – применяется совместно с флагом O_CREAT. При совместном их использовании и существовании файла с указанным именем, открытие файла не производится и констатируется ошибочная ситуация;
O_NDELAY – запрещает перевод процесса в состояние ожидание при выполнении операции открытия и любых последующих операциях над этим файлом;
O_APPEND – при открытии файла и перед выполнением каждой операции записи (если она, конечно, разрешена) указатель текущей позиции в файле устанавливается на конец файла;
O_TRUNC – если файл существует, уменьшить его размер до 0, с сохранением существующих атрибутов файла, кроме, быть может, времен последнего доступа к файлу и его последней модификации.

Кроме того, в некоторых версиях операционной системы UNIX могут применяться дополнительные значения флагов:

O_SYNC – любая операция записи в файл будет блокироваться (т. е. процесс будет переведен в состояние ожидание) до тех пор, пока записанная информация не будет физически помещена на соответствующий нижележащий уровень hardware;
O_NOCTTY – если имя файла относится к терминальному устройству, оно не становится управляющим терминалом процесса, даже если до этого процесс не имел управляющего терминала.

Параметр *mode* устанавливает атрибуты прав доступа различных категорий пользователей к новому файлу при его создании. Он обязателен, если среди заданных флагов присутствует флаг O_CREAT, и может быть опущен в противном случае. Этот параметр задается как сумма следующих восьмеричных значений:

0400 – разрешено чтение для пользователя, создавшего файл;
0200 – разрешена запись для пользователя, создавшего файл;
0100 – разрешено исполнение для пользователя, создавшего файл;
0040 – разрешено чтение для группы пользователя, создавшего файл;
0020 – разрешена запись для группы пользователя, создавшего файл;
0010 – разрешено исполнение для группы пользователя, создавшего файл;
0004 – разрешено чтение для всех остальных пользователей;
0002 – разрешена запись для всех остальных пользователей;
0001 – разрешено исполнение для всех остальных пользователей.

При создании файла реально устанавливаемые права доступа получаются из стандартной комбинации параметра *mode* и маски создания файлов текущего процесса *umask*, а именно – они равны *mode & ~umask*.

Системный вызов возвращает значение файлового дескриптора для открытого файла при нормальном завершении и значение -1 при возникновении ошибки.

Системный вызов *open()* использует набор флагов для того, чтобы специфицировать операции, которые предполагается применять к файлу в дальнейшем или которые должны быть выполнены непосредственно в момент открытия файла. Из всего возможного набора флагов на лабораторной работе понадобятся только флаги O_RDONLY, O_WRONLY, O_RDWR, O_CREAT и O_EXCL. Первые три флага являются взаимоисключающими: хотя бы один из них должен быть применен и наличие одного из них не допускает наличия двух других. Эти флаги описывают набор операций, которые, при успешном открытии файла, будут разрешены над файлом в дальнейшем: только чтение, только запись, чтение и запись. У каждого файла существуют атрибуты прав доступа для различных категорий пользователей. Если файл с заданным именем существует на диске, и права доступа к нему не противоречат запрошенному набору операций, то операционная система сканирует таблицу открытых файлов от ее начала к концу в поисках первого свободного элемента, заполняет его и возвращает индекс этого элемента в качестве файлового дескриптора открытого файла. Если файла на диске нет, не хватает прав или отсутствует свободное место в таблице открытых файлов, то констатируется возникновение ошибки.

Если файл на диске отсутствует и его нужно создать, флаг для набора операций должен использоваться в комбинации с флагом O_CREAT. Если файл существует, то все происходит по рассмотренному выше сценарию. Если файла нет, сначала выполняется создание файла с набором прав, указанным в параметрах системного вызова.

Чтобы создать файл в момент открытия, флаг для набора операций должен использоваться в комбинации с флагами O_CREAT и O_EXCL.

Системные вызовы *read(), write(), close()*. Для совершения потоковых операций чтения информации из файла и ее записи в файл применяются системные вызовы *read()* и *write()*.

Прототипы системных вызовов

```
#include <sys/types.h>
#include <unistd.h>
size_t read(int fd, void *addr, size_t nbytes);
size_t write(int fd, void *addr, size_t nbytes);
```

Системные вызовы *read* и *write* предназначены для осуществления потоковых операций ввода (чтения) и вывода (записи) информации над каналами связи, описываемыми файловыми дескрипторами, т. е. для файлов, *pipe*, *FIFO* и *socket*.

Параметр *fd* является файловым дескриптором созданного ранее потокового канала связи, через который будет отсылаться или получаться информация, т. е. значением, которое вернул один из системных вызовов *open(), pipe()* или *socket().*

Параметр *addr* представляет собой адрес области памяти, начиная с которого будет браться информация для передачи или размещаться принятая информация.

Параметр *nbytes* для системного вызова *write* определяет количество байт, которое должно быть передано, начиная с адреса памяти *addr*. Параметр *nbytes* для системного вызова *read* определяет количество байт, которое необходимо получить из канала связи и разместить в памяти, начиная с адреса *addr*.

В случае успешного завершения системный вызов возвращает количество реально посланных или принятых байт. Это значение (большее или равное 0) может не совпадать с заданным значением параметра *nbytes*, а быть меньше, чем оно, в силу отсутствия места на диске или в линии связи при передаче данных или отсутствия информации при ее приеме. При возникновении какой-либо ошибки возвращается отрицательное значение.

При работе с файлами информация записывается в файл или читается из файла, начиная с места, определяемого указателем текущей позиции в файле. Значение указателя увеличивается на количество реально прочитанных или записанных байт. При чтении информации из файла она не пропадает из него. Если системный вызов *read* возвращает значение 0, то это означает, что файл прочитан до конца.

После завершения потоковых операций процесс должен выполнить операцию закрытия потока ввода-вывода, во время которой произойдет окончательный сброс буферов на линии связи, освободятся выделенные ресурсы операционной системы, и элемент таблицы открытых файлов, соответствующий файловому дескриптору, будет отмечен как свободный. За эти действия отвечает системный вызов *close()*. При завершении работы процесса с помощью явного или неявного вызова функции *exit()* происходит автоматическое закрытие всех открытых потоков ввода-вывода.

Системный вызов *close*. Прототип системного вызова:

```
#include <unistd.h>
int close(int fd);
```

Системный вызов *close* предназначен для корректного завершения работы с файлами и другими объектами ввода-вывода, которые описываются в операционной системе через файловые дескрипторы: *pipe, FIFO, socket*.

Параметр *fd* является дескриптором соответствующего объекта, т. е. значением, которое вернул один из системных вызовов *open(), pipe()* или *socket()*.

Системный вызов возвращает значение 0 при нормальном завершении и значение -1 при возникновении ошибки.

В качестве иллюстрации рассмотрим следующую программу:

```
/*Программа 05
-1.c, иллюстрирующая использование системных вызовов
open(), write() и close() для записи информации в файл */
#include <sys/types.h>
#include <fcntl.h>
#include <stdio.h>
int main(){
   int fd;
   size_t size;
   char string[] = "Hello, world!";
   /* Обнуляем маску создания файлов текущего процесса для того,
   чтобы права доступа у создаваемого файла точно соответствовали
   параметру вызова open() */
   (void)umask(0);
   /* Попытаемся открыть файл с именем myfile в текущей директории
    только для операций вывода. Если файла не существует, попробуем
    его создать с правами доступа 0666, т. е. read-write для всех
    категорий пользователей */
   if((fd = open("myfile", O_WRONLY | O_CREAT, 0666)) < 0){
      /* Если файл открыть не удалось, печатаем об этом сообщение и
       прекращаем работу */
      printf("Can\'t open file\n");
      exit(-1);
   }
   /* Пробуем записать в файл 14 байт из нашего массива, т.е. всю
   строку "Hello, world!" вместе с признаком конца строки */
   size = write(fd, string, 14);
   if(size != 14){
      /* Если записалось меньшее количество байт, сообщаем об ошибке */
      printf("Can\'t write all string\n");
      exit(-1);
   }
   /* Закрываем файл */
   if(close(fd) < 0){
      printf("Can\'t close file\n");
   }
   return 0;
}
```

Обратите внимание на использование системного вызова *umask()* с параметром 0 для того, чтобы права доступа к созданному файлу точно соответствовали указанным в системном вызове *open()*.

Краткое описание языка программирования Си

Си – универсальный язык программирования. Он тесно связан с системой UNIX, так как был разработан в этой системе, которая, как и большинство программ работающих в ней, написаны на Си.

В отличие от «безтиповых» языков Си обеспечивает разнообразие типов данных. Базовыми типами являются символы, а также целые и числа с плавающей точкой различных размеров. Кроме того, имеется возможность получать целую иерархию производных типов данных из указателей, массивов, структур и объединений. Выражения формируются из операторов и операндов. Любое выражение, включая присваивание и вызов функции, может быть инструкцией. Указатели обеспечивают машинно-независимую адресную арифметику. В Си имеются основные управляющие конструкции, используемые в хорошо структурированных программах: составная инструкция ({...}), ветвление по условию (if-else), выбор одной альтернативы из многих (switch), циклы с проверкой наверху (while, for) и с проверкой внизу (do), а также средство прерывания цикла (break). В качестве результата функции могут возвращать значения базовых типов, структур, объединений и указателей. Любая функция допускает рекурсивное обращение к себе. Функции программы на Си могут храниться в отдельных исходных файлах и компилироваться независимо. Переменные по отношению к функции могут быть внутренними и внешними. Последние могут быть доступными в пределах одного исходного файла или всей программы. Си – язык сравнительно «низкого уровня». Однако это вовсе не умаляет его достоинств, просто Си имеет дело с теми же объектами, что и большинство компьютеров, т. е. с символами, числами и адресами. С ними, можно оперировать при помощи арифметических и логических операций, выполняемых реальными машинами. В Си нет прямых операций над составными объектами, такими как строки символов, множества, списки и массивы. В нем нет операций, которые бы манипулировали с целыми массивами или строками символов, хотя структуры разрешается копировать целиком как единые объекты. В языке нет каких-либо средств распределения памяти, помимо возможности определения статических переменных и стекового механизма при выделении места для локальных переменных внутри функций. Наконец, в самом Си нет средств ввода-вывода, инструкций READ (читать) и WRITE (писать) и каких-либо методов доступа к файлам. Все это – механизмы высокого уровня, которые в Си обеспечиваются исключительно с помощью явно вызываемых функций. Большинство реализованных Си-систем содержат в себе разумный стандартный набор этих функций. Си предоставляет средства лишь последовательного управления ходом вычислений: механизм ветвления по условиям, циклы, составные инструкции, подпрограммы – и не содержит средств мультипрограммирования, параллельных процессов, синхронизации и организации сопрограмм. Однако компактность языка имеет реальные выгоды. Поскольку Си относительно мал, то и описание его кратко, и овладеть им можно быстро. Программист может реально рассчитывать на то, что он будет знать, понимать и на практике регулярно пользовать-

ся всеми возможностями языка. Важный аспект языка – это определение библиотеки, поставляемой вместе с Си-компилятором, в которой специфицируются функции доступа к возможностям операционной системы (например, чтения-записи файлов), форматного ввода-вывода, динамического выделения памяти, манипуляций со строками символов и т. д. Набор стандартных заголовочных файлов обеспечивает единообразный доступ к объявлениям функций и типов данных. Почти все программы, написанные на Си, если они не касаются каких-либо скрытых в операционной системе деталей, переносимы на другие машины. Си соответствует аппаратным возможностям многих машин, однако он не привязан к архитектуре какой-либо конкретной машины. Основной философией Си остается то, что программисты сами знают, что делают; язык лишь требует явного указания об их намерениях. Си, как и любой другой язык программирования, не свободен от недостатков. Тем не менее, как оказалось, Си – чрезвычайно эффективный и выразительный язык, пригодный для широкого класса задач.

2. Задание на лабораторную работу

С помощью компилятора С создать и выполнить программу, исходный текст которой приведен в примере 1.

Модифицировать эту программу для вывода на экран информации о результате создания файла.

Порядок выполнения работы:

1. Прочитать методический материал.
2. Изучить характеристики и синтаксис функций и системных вызовов.
3. Набрать код примера 1 в текстовый файл и произвести компиляцию программы.
4. Проверить работоспособность программы.
5. Модифицировать программу для вывода на экран сообщение о результате создания файла.
6. Проверить работоспособность модифицированной программы.

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы:

1. Какие системные вызовы используются для записи информации в файл?
2. Как формируются права доступа к файлу, создаваемому пользовательской программой?
3. Для чего предназначен системный вызов *close*?
4. По каким признакам можно определить неудачную попытку открыть файл?
5. Чем отличаются системные вызовы *fread()* и *read()*?

Лабораторная работа № 14
Механизмы безопасности в Linux

Цель работы:

– изучение основных возможностей по обеспечению безопасности информационных систем

– получение опыта работы с основными утилитами, обеспечивающими различные аспекты безопасности информационные систем в операционной системе Linux классификация основных угроз безопасности информационных систем

1.Теоретический материал

Основные сведения

В данной лабораторной работе будет проведен краткий экскурс в наиболее распространенные средства, связанные с безопасностью Linux. Информация предоставлена в сжатом виде, и если какое-то средство вас заинтересует, можно пройтись по ссылкам и прочитать более подробно. Будут рассмотрены следующие средства: ssh, sudo, firewall (iptables).

sudo: выполнение программ от чужого имени

Описание: Выполнение программ от своего и/или чужого имени.

Механизм работы: При вызове команды sudo/sudoedit система считывает файл

/etc/sudoers, и на его основе определяет, какие команды может вызывать пользователь.

Пример использования: Вся конфигурация определяется в файле /etc/sudoers. Например, можно разрешать выполнять только определенные команды и только от определенного пользователя:

WEBMASTERS www = (www) ALL, (root) /usr/bin/su www

Данная строчка говорит о том, что пользователи, определенные в алиасе *WEBMASTERS* могут выполнять все команды от имени пользователя *www*, или делать *su* только в *www*.

Пакет sudo позволяет системному администратору давать права определенным пользователям (или группам) на исполнение конкретных программ с правами другого пользователя (и записывать эти действия в журнал). Возможна привязка списка допустимых команд к имени хоста, что позволяет использовать один файл настройки на нескольких хостах с различными полномочиями. Обычно требует аутентификации пользователя (например, ввода пароля). Позволяет избегать слишком частого ввода пароля (по умолчанию - 5 минут). Для борьбы с подменой динамических библиотек из окружения удаляются переменные типа LD_* и т. п., а также IFS, ENV, BASH_ENV, KRB_CONF, KRB5_CONFIG, LOCALDOMAIN, RES_OPTIONS, HOSTALIASES.

Можно также удалять текущую директорию из PATH.

Состоит из файла настройки */etc/sudoers*, программы его редактирования

visudo и клиентской программы sudo. В лабораторной работе также описывается процедура установки из исходных текстов и ключи configure.

visudo

visudo позволяет осуществить безопасное редактирование sudoers, она проверяет файл на корректность после выхода из текстового редактора.

По умолчанию, простая аккредитация пользователя означает, что при попытке исполнить команду от имени другого ему будет предложено ввести свой пароль – свой собственный, обратите внимание, а не пароль того, от чьего имени он собирается действовать. Вдобавок, «эффект памяти» после первого ввода пароля составляет по тому же умолчанию всего пять минут, и если команду sudo вновь отдать по истечении этого срока, она снова затребует пароль. Однако изменением настроек в файле

/etc/sudoers можно преодолеть эти ограничения, уместные для больших многопользовательских систем, но никак не для домашних компьютеров.

Кстати, если sudo попытается воспользоваться неаккредитованный пользователь, сообщение об этом в виде электронного письма незамедлительно отправится на локальный адрес root, и если даже машина не подключена к Интернету, суперпользователь получит это письмо. Кроме того, запись о нарушителе появится в системном журнале.

Чтобы определить, какими полномочиями наделѐн тот или иной пользователь для исполнения команды sudo, достаточно от его имени сделать запрос sudo -l

Изначально только root имеет право вершить при помощи sudo всѐ, что угодно, – но суперпользователю эта утилита ни к чему, он и так способен на всѐ. Владельцам же прочих учѐтных записей на данном компьютере следует позаботиться о своей аккредитации. Заведение в */etc/sudoers* новых правил следует производить – внимание!

– не при помощи привычного вам редактора, а посредством особой утилиты *visudo*. Еѐ аналоги, кстати, имеются и для редактирования файлов паролей */etc/passwd* и групп

/etc/group – *vipw* и *vigr*, соответственно. Главная их особенность – в том, что перед запуском самого редактора они блокируют (lock) редактируемый файл, так что ни один другой пользователь не сможет в то же самое время его править. Vi ведь оперирует не с самим файлом, а с его копией в памяти, как мы помним, и до самого момента явной записи информации на диск в исходный файл не вносятся изменения. Значит, возможна ситуация, когда один и тот же файл открывают на редактирование два пользователя (в случае */etc/sudo*, например, им обоим должнен быть известен пароль *root*). Тогда в итоге файл останется таким, каким сохранит его на диск последний из завершивших работу пользователей. Если же файл блокирован, открыть его кому-то ещѐ в то же самое время будет непросто.

Утилита sudo – мощный и гибкий инструмент, и с его помощью системный администратор может существенно облегчить свою жизнь. Скажем, можно

аккредитовать в */etc/sudoers* своего заместителя для выполнения некоторых рутинных обязанностей и не терзаться мыслью о том, что пароль root'а знает кто-то ещѐ: теперь это ни к чему. Однако вряд ли для домашнего компьютера потребуется вся потаѐнная мощь *sudo* – все-таки уровень подозрительности к окружающим, если это члены семьи, даже у самого параноидального сисадмина должен быть понижен. Так что наиболее распространѐнная опция в sudo «для дома, для семьи» – это разрешение на запуск той или иной полезной утилиты от имени непривилегированного пользователя без дополнительного введения пароля.

Аккредитация в этом случае выглядит чрезвычайно просто (если вас интересуют более глубокие подробности, к вашим услугам man sudo и man 5 sudoers). Запустите команду visudo, и в самой последней строке открывшегося файла допишите строчку:

[имя пользователя] localhost.localdomail = NOPASSWD: [полный путь исполнения]

Все множество средств обеспечения безопасности можно разделить на следующие группы или категории:

- Средства управления доступом к системе (доступ с консоли, доступ по сети) и разграничения доступа
- Обеспечение контроля целостности и неизменности программного обеспечения (сюда же я отношу средства антивирусной защиты, поскольку внедрение вируса есть изменение ПО)
- Средства криптографической защиты
- Средства защиты от вторжения извне (внешнего воздействия)
- Средства протоколирования действий пользователей, которые тоже служат обеспечению безопасности (хотя и не только)
- Средства обнаружения вторжений
- Средства контроля состояния безопасности системы (обнаружения уязвимостей)

Как перезагрузить компьютер, находясь за пару тысяч километров от него или выполнить другие действия удаленно и безопасно? Ответ на эти вопросы – протокол SSH.

SSH (Secure Shell) – это сетевой протокол, используемый для удаленного управления компьютером и для передачи файлов. SSH похож по функциональности на протоколы telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли.

SSH позволяет передавать через безопасный канал любой другой сетевой протокол, таким образом, можно не только удаленно работать на компьютере, но и передавать по шифрованному каналу звуковой поток или видео (например, с веб-камеры). Также, SSH может использовать сжатие передаваемых данных для последующей их шифрации.

Большинство хостинг-провайдеров за определенную плату предоставляют клиентам доступ к их домашнему каталогу по SSH. Это очень удобно как для работы в командной строке, так и для удаленного запуска графических

приложений. Через SSH можно работать и в локальной сети. Если вам лень ходить к серверам – администрируйте их удаленно, используя любой SSH-клиент.

SSH-клиенты и SSH-сервера написаны под большинство платформ: Windows, Linux, Mac OS X, Java, Solaris, Symbian OS, Windows Mobile и даже Palm OS.

Первая версия протокола, SSH-1, была разработана в 1995 году исследователем Tatu Yl'nen из Технологического университета Хельсинк и, Финляндия. SSH-1 был написан для обеспечения большей конфиденциальности, чем протоколы rlogin, telnet и rsh. В 1996 году была разработана более безопасная версия протокола, SSH-2, уже несовместимая с SSH-1. Протокол приобрел еще большую популярность, и к 2000 году его использовало уже порядка двух миллионов пользователей. В 2006 году протокол был утвержден рабочей группой IETF в качестве Интернет- стандарта.

Однако, до сих пор в некоторых странах (Франция, Россия, Ирак и Пакистан) требуется специальное разрешение в соответствующих структурах для использования определенных методов шифрования, включая SSH. См. закон Российской Федерации «О федеральных органах правительственной связи и информации».

Распространены две реализации SSH: коммерческая (закрытая) и свободная (открытая). Свободная реализация называется OpenSSH. К 2006 году 80 % компьютеров Интернет использовало именно OpenSSH. Коммерческая реализация разрабатывается организацией SSH Inc., http://ssh.com/ – закрытая реализация, бесплатная для некоммерческого использования. Свободная и коммерческая реализации SSH содержат практически одинаковый набор команд.

Существуют две версии протокола SSH: SSH-1 и SSH-2. В первой версии протокола есть существенные недостатки, поэтому в настоящее время SSH-1 практически нигде не применяется.

Многие взломщики сканируют сеть в поиске открытого порта SSH, особенно – адреса хостинг-провайдеров. Так что, в целях безопасности – запрещайте доступ по ssh для суперпользователя. Обычно злоумышленники подбирают именно пароль суперпользователя. См. ниже рекомендации по безопасности использования SSH.

Протокол SSH-2 устойчив в атакам «man-in-middle», в отличие от протокола telnet. То есть, прослушивание трафика, «снифинг», ничего не дает злоумышленнику. Протокол SSH-2 также устойчив к атакам путем присоединения посредине (session hijacking) и обманом сервера имен (DNS spoffing).

Далее по тексту, мы будем иметь ввиду под SSH вторую версию протокола, SSH-2.

SSH-сервера

Debian GNU/Linux: dropbear, lsh-server, openssh-server, ssh

MS Windows: freeSSHd, OpenSSH sshd, WinSSHD, ProSHHD, Dropbear SSH Server

SSH-клиенты и оболочки

- Debian GNU/Linux: kdessh, lsh-client, openssh-client, putty, ssh

- MS Windows: PuTTY, SecureCRT, ShellGuard, Axessh, ZOC, SSHWindows, ProSSHD
- Mac OS: NiftyTelnet SSH
- Symbian OS: PuTTY
- Java: MindTerm, AppGate Security Server

Для работы по SSH нужен SSH-сервер и SSH-клиент. Сервер прослушивает соединения от клиентск их машин и при установлении связи производит аутентификацию, после чего начинает обслуживание клиента. Клиент используется для входа на удаленную машину и выполнения команд.

Предположим, что сервер у нас настроен и работает. Для подключения клиента требуется сгенерировать пару из открытого и закрытого ключей. Если Вы используете PuTTY под Windows – это делается утилитой puttygen.exe.

Под Linux обычно используется команда puttygen (для PuTTY) или ssh-keygen (для OpenSSH). Далее указываем клиенту – где находится закрытый ключ, и соединяемся с вводом пароля. Возможно также беспарольное соединение, а чем упомянуто ниже.

Рекомендации по безопасности использования SSH:

1. Запрещение удаленного root-доступа.
2. Запрещение подключения с пустым паролем или отключение входа по паролю.
3. Выбор нестандартного порта для SSH-сервера.
4. Использование длинных SSH2 RSA-ключей (2048 бит и более). По состоянию на 2006 год система шифрования на основе RSA считалась надѐжной, если длина ключа не менее 1024 бит.
5. Ограничение списка IP-адресов, с которых разрешен доступ. Например, настройкой файрвола.
6. Запрещение доступа с некоторых, потенциально опасных адресов.
7. Отказ от использования распространенных или широко известных системных логинов для доступа по SSH.
8. Регулярный просмотр сообщений об ошибках аутентификации.
9. Установка детекторов атак (IDS, Intrusion Detection System).
10. Использование ловушек, подделывающих SSH-сервис (honeypots).

Как подключиться к удаленному серверу из Linux или FreeBSD? Команда подключения к локальному SSH-серверу из командной строки для пользователя pacify (сервер слушает нестандартный порт 30000):

$ ssh -p30000 pacify@127.0.0.1

Под Windows можно воспользоваться программой PuTTY. Она имеет простой графический интерфейс, через который удобно настраивать подключения. На вкладке SSH\Auth надо выбрать закрытый ключ, который будет использоваться PuTTY.

К удаленному компьютеру можно подключиться по SSH, не вводя пароль, а используя открытый ключ. Разумеется, открытый ключ лучше передавать на сервер по защищенному каналу.

Генерация пары SSH-2 RSA-ключей длиной 4096 бита программой

puttygen под Linux/UNIX надо выполнить команду:

$ puttygen - t rsa -b 4096 -o sample

Под Windows у этой программы (puttygen.exe) есть пользовательский интерфейс.

Установка SSH в Linux на примере Debian

Итак, всѐ, что нам нужно для установки полного комплекта удалѐнного управления компьютером (**SSH-клиент** и **SSH-сервер**) давным-давно лежит в репозитории. Лѐгким движением ставим пакет:

 # apt-get install ssh

и ждѐм несколько мгновений, когда оно настроится. После этого мы получим возможность **SSH** доступа в систему и управления ей. Так как технология эта кросс- платформенная, то можно управлять по SSH Linux или FreeBSD можно и из Windows. Для этого есть **putty,** SSH Windows клиент.

На стороне клиента пеперь надо поправить настройки, которые лежат в каталоге

/etc/ssh – конфиг для клиента называется ssh-config, конфиг для сервера, соответственно, sshd-config. На своей, клиентской, стороне, настраиваем возможность приѐма X11Forward, ищем и меняем ключи на:

ForwardX11 yes ForwardX11Trusted yes

Клиентская машина теперь может запускать удалѐнно графические приложения на сервере. Настройка **SSH** на стороне клиента закончена, теперь идѐм к админу далѐкого сервера. В принципе, можно на клиентской стороне ничего не менять, а логиниться на удалѐнную машину так:

*$ ssh –**X** user@server1.mydomain.com*

Или

*$ ssh –**X** user@192.168.x.x*

На стороне сервера пеперь нужно настроить SSH сервер: в конфигах машины-сервера, к которой будем подсоединяться (у вас ведь есть еѐ рутовый пароль, так ведь?) в

/etc/ssh/sshd-config ищем и меняем ключи на:

X11Forwarding yes X11DisplayOffset 10 X11UseLocalhost yes

Этим мы разрешаем серверу запускать удалѐнно графические приложения и отправлять их на клиентскую машину. Перестартуем сервис:

$sudo /etc/init.d/ssh restart

Теперь мы сможем заходить на машину не только в консольном режиме, но и с запуском иксовых приложений. Если хочется разрешить вход только с определѐнных машин, нужно подправить строки в конфиге */etc/ssh/sshd_config*

AllowUsers hacker@* AllowUsers *@192.168.1.* **SSH в действии**

Всѐ готово, и теперь будут приведены несколько команд SSH для примера. Открываем консольку и пишем в ней:

$ ssh

имя_пользователя_удалѐнной_машины@ip_адрес_или_сетевое_имя_удалѐнной_маши ны

После этого нас могут спросить: данный айпишник ещѐ не опознан, как

доверительный, стоит ему доверять? Пишем yes, стоит, конечно! Далее вводим пароль пользователя удалѐнной машины, под которым мы заходим и, если он правильный, попадаем в консоль ***удалённой*** машины. В процессе набора пароля вы его не увидите – набирайте всѐ равно; даѐтся три попытки – потом соединение обрывается.

Итак, нас поприветствуют как-то вроде этого:

penta4@penta4rce:~$ ssh beast@192.168.1.5 Password:

Last login: Tue Oct 10 19:23:57 2006 from 192.168.1.1 beast@notebeast:~$

Теперь, в окошке терминала, который на нашей машине, мы рулим компьютером, к которому мы подключились. Не перепутайте терминалы, а то вырубите не тот компьютер. Здесь всѐ просто и логично, но нам бы хотелось ещѐ и запускать графические приложения на удалѐнной системе.

Запуск графических приложений удалѐнно. Вводим, как обычно, лог ин и пароль ***удалённой*** машины. И вот перед нами та же самая консоль. Как вызвать графическое приложение? Просто наберите имя вызываемой программы и поставьте после имени знак амперсанд:

$ gimp &;

Это запустит на ***удалённо*** машине GiMP в фоне и вернѐт вам консоль для дальнейших действий. Если вы не поставите амперсанд после имени приложения, то управление в консоль будет возвращено только после завершения приложения.

Принцип работы файрвол

Брандмауэр (файрвол) предназначен для фильтрации и обработки пакетов, проходящих через сеть. Когда пакет прибывает, брандмауэр анализирует заголовки пакета и принимает решение, «выбросить» пакет (DROP), принять пакет (ACCEPT, пакет может пройти дальше) или сделать с ним что-то еще более сложное.

В Linux брандмауэр является модулем ядра, его неотъемлемой частью. С его помощью мы можем делать с пакетами множество хитроумных вещей, но основной принцип манипуляции трафиком сохраняется: просматриваются заголовки пакетов и решается их дальнейшая судьба. Интерфейсом для модификации правил, по которым брандмауэр обрабатывает пакеты, служит iptables.

Итак, пребывающий пакет проходит по цепочке правил. Каждое правило содержит *условие* и *цель* (действ ие). Если пакет *удовлетворяет условию* то он *передается на цель*, в противном случае к пакету применяется следующее правило в цепочке. Если пакет не удовлетворил ни одному из условий в цепочке, то к нему применяется *действие по умолчанию*.

Входящий пакет начинает обрабатываться брандмауэром с цепочки PREROUTING в таблице **mangle**. Затем он обрабатывается правилами цепочки PREROUTING таблицы **nat.** На этом этапе проверяется, не требуется ли модификация назначения пакета (DNAT). Важно сменить назначение сейчас, потому что маршрут пакета определяется сразу после того, как он покинет цепочку PREROUTING. После этого он будет отправлен на цепочку INPUT (если целью пакета является этот компьютер) или FORWARD (если его целью является дру-

гой компьютер в сети).

Если целью пакета является другой компьютер, то пакет фильтруется правилами цепочки FORWARD таблиц **mangle** и **filter**, а затем к нему применяются правила цепочки POSTROUTING. На данном этапе можно использовать

SNAT/MASQUARADE (подмена источника/маскировка). После этих действий пакет (если выжил) будет отправлен в сеть

Если назначением пакета является сам компьютер с брандмауэром, то, после маршрутизации, он обрабатывается правилами цепочек INPUT таблиц **mangle** и **filter**. В случае прохождения цепочек пакет передается приложению.

Когда приложение, на машине с брандмауэром, отвечает на запрос или отправляет собственный пакет, то он обрабатывается цепочкой OUTPUT таблицы **filter**. Затем к нему применяются правила цепочки OUTPUT таблицы **nat**, для определения, требуется ли использовать DNAT (модификация назначения), пакет фильтруется цепочкой OUTPUT таблицы **filter** и выпускается в цепочку POSTROUTING которая может использовать SNAT и QoS. В случае успешного прохождения POSTROUTING пакет выходит в сеть.

Для добавления правила в цепочку используется ключ **-A**

```
#iptables -A INPUT правило
```

добавит правило в цепочку INPUT таблицы **filter** (по умолчанию). Для указания таблицы, в цепочку которой следует добавить правило, используйте ключ **-t**:

```
#iptables -t nat -A INPUT правило
```

добавит правило в цепочку INPUT таблицы **nat**. Цель по умолчанию задается с помощью ключа **-P**:

```
#iptables -P INPUT DROP
```

Условия для отбора пакетов

Теперь мы знаем как пакеты проходят сквозь различные таблицы и цепочки iptables. Пришло время разъяснить принципы, по которым строятся условия накладываемые на пакеты:

Немного о протоколе TCP/IP

TCP/IP является протоколом, в котором соединение устанавливается в 3 фазы. Если компьютер А пытается установить соединение с компьютером Б они обмениваются специальными TCP пакетами.

После чего соединение считается **установленным** (ESTABLISHED). iptables различает эти состояния как NEW и ESTABLISHED.

2. Задание на лабораторную работу

1. Проверьте, установлена ли на Вашем рабочем месте утилита sudo и, при необходимости, инсталлируйте ее.

2. Отредактируйте файл с конфигурацией sudo, добавив своего пользователя, и запустите команду ifconfig от своего имени.

3.

```
┌──(root㉿kali)-[~]
└─# echo "annak  ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/annak
annak  ALL=(ALL) NOPASSWD:ALL
```

Рисунок 12. Добавление своего пользователя в файл с конфигурацией sudo

```
┌──(annak㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe96:1898  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:96:18:98  txqueuelen 1000  (Ethernet)
        RX packets 314  bytes 158863 (155.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 336  bytes 43181 (42.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рисунок 13. Результат работы команды ifconfig

4. Проверьте, какими правами наделен Ваш пользователь?

```
annak@kali:~$ sudo -l
Matching Defaults entries for annak on kali:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User annak may run the following commands on kali:
    (ALL) NOPASSWD: ALL
```

Рисунок 14. Просмотр прав пользователя

5. Установите серверный пакет Open SSH и настройте его

```
┌──(root㉿kali)-[~]
└─# apt-get install ssh
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  openssh-client openssh-server openssh-sftp-server
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard ufw
The following NEW packages will be installed:
  ssh
```

Рисунок 15. Установка серверного пакета Open SSH

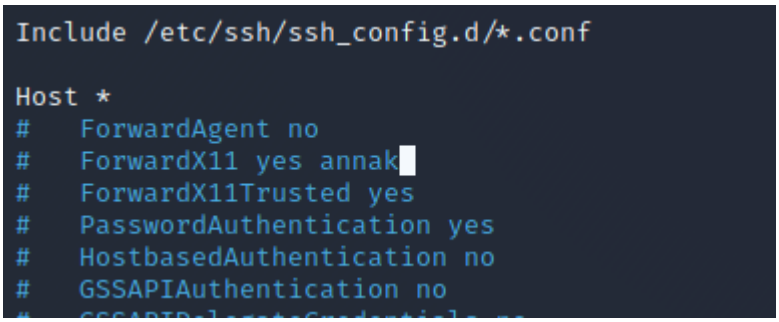

Рисунок 16. Настройка SSH на компьютере

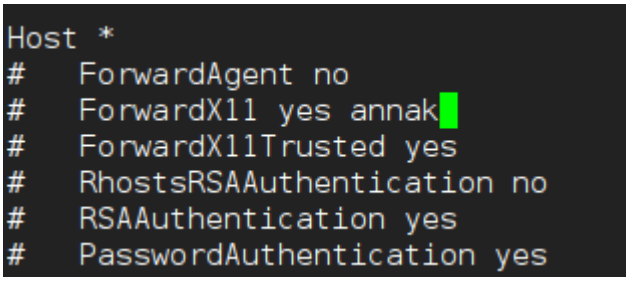

Рисунок 17. Настройка SSH на сервере

5. Попробуйте зайти по протоколу ssh на рабочее место соседнего компьютера и откройте свое место для соседа по лабораторной работе.

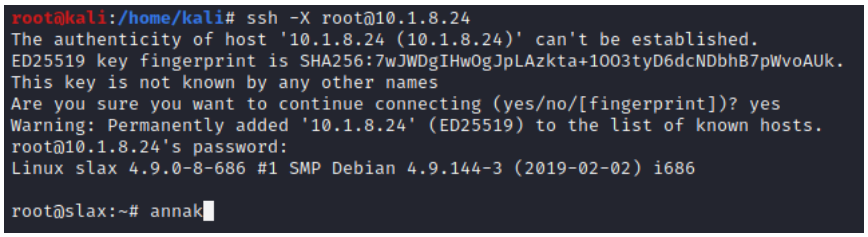

Рисунок 18. Удачная попытка входа

6. Сгенерируйте и установите открытый ключ для доступа, попробуйте беспарольный вход.

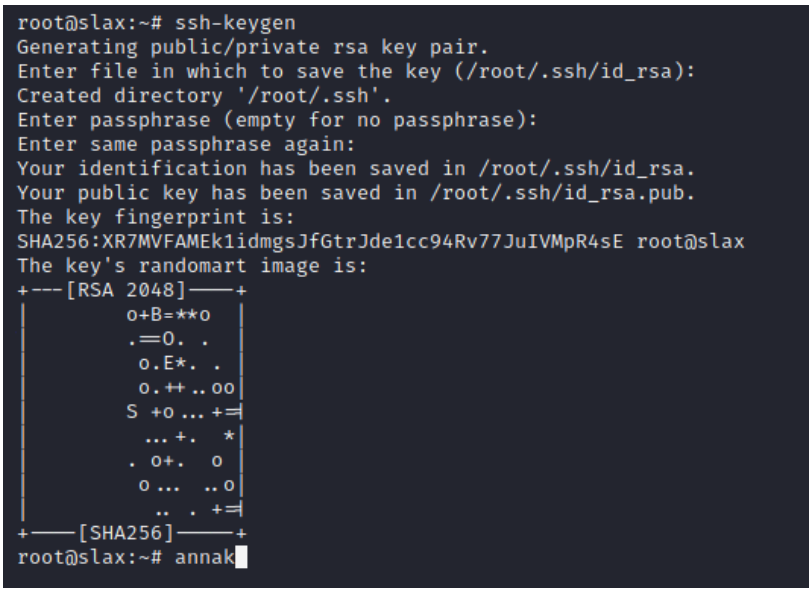

Рисунок 19. Генерация ключа

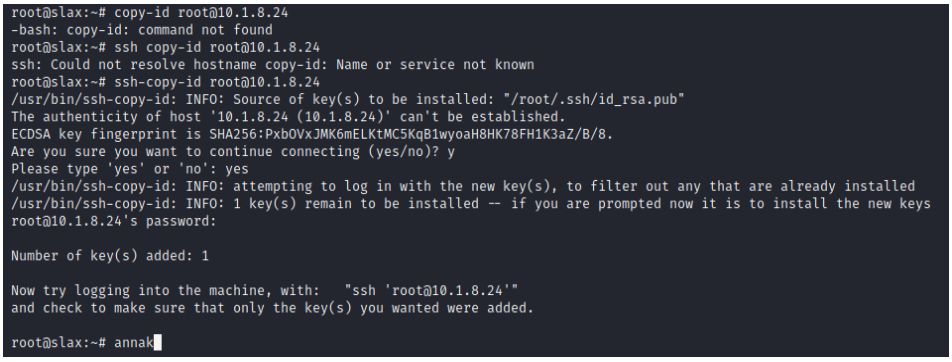

Рисунок 20. Установка ключа

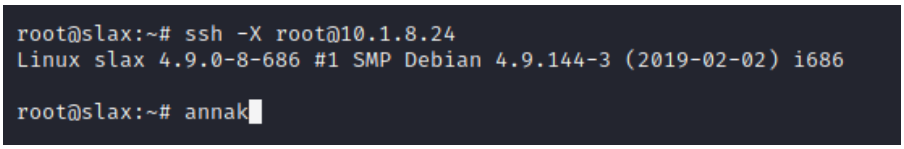

Рисунок 21. Вход без пароля

7. Установите правило, блокирующее пакеты от соседнего рабочего места и проверьте работоспособность фильтра. По окончании теста отмените это правило.

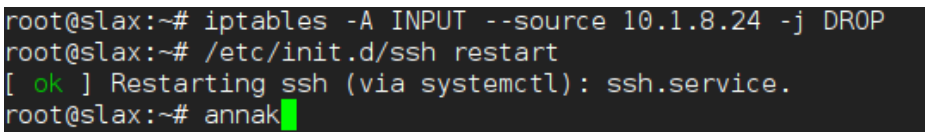

Рисунок 22. Установка правила

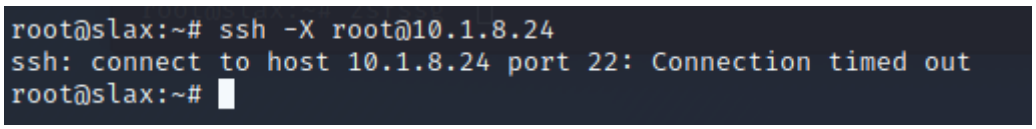

Рисунок 23. Невозможно установить соединение

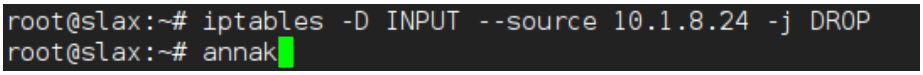

Рисунок 24. Отмена правила

Требования к отчету и защите

Защита предполагает опрос по материалу, изложенному в отчете.

Отчет должен быть оформлен: требуется наличие титульного листа, указание цели лабораторной работы, последовательное изложение разработанного материала, подписанные скриншоты хода работы, если использовалась компьютерная техника.

Проверяется знание теоретического материала с учетом знаний ответов на контрольные вопросы, приведенные в пункте «Контрольные вопросы».

Задаются вопросы по пунктам, приведенным в пункте «Методические указания и порядок выполнения работы».

Критерии к оценке защиты отчета студентом приведены в пункте «Общие сведения».

Контрольные вопросы

1. При помощи каких основных утилит обеспечивается безопасность в операционной системе Linux?
2. При помощи какой утилиты производится выполнение программ от своего и/или чужого имени?
3. Какой файл отвечает за настройки прав для утилиты ***sudo***?
4. Каковы основные отличия удаленного доступа ***telnet*** от сетевого протокола ***ssh***?
5. Назовите основные программные пакеты, реализующие ***ssh*** в различных операционных системах?
6. Какая утилита реализует механизм генерации открытого ключа ***ssh***?
7. Каковы основные принципы работы файрвол?
8. Приведите примеры фильтрации пакетов при помощи ***iptables***

ЗАКЛЮЧЕНИЕ

Правильная организация практических учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет привить студентам умения и навыки в овладении, изучении, усвоении и систематизации приобретаемых знаний в процессе обучения, обеспечивать высокий уровень успеваемости в период обучения, привить навыки повышения профессионального уровня в течение всей трудовой деятельности.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 27 декабря 2006 г. N 373-ст: дата введения 2008-02-01. – URL: https://docs.cntd.ru/document/1200058320 (дата обращения: 15.04.2021). – Текст: электронный.

2. ГОСТ Р ИСО 19011-2012. Руководящие указания по аудиту систем менеджмента: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 19 июля 2012 г. N 196-ст: дата введения 2013-02-01. – URL: https://docs.cntd.ru/document/1200095049 (дата обращения: 15.04.2021). – Текст: электронный.

3. Аудит информационной безопасности органов исполнительной власти: учебное пособие / В. И. Аверичников, М. Ю. Рытов, А. В. Кувылкин, М. В. Рудановский. – Москва: Флинта, 2011. – 100 с. – ISBN 978-5-9765-1277-1.

4. Аверченков, В. И. Оптимизация выбора состава средств инженерно-технической защиты информации на основе модели Клементса-Хоффмана / В. И. Аверченков, М. Ю. Рытов, Т. Р. Гайнулин // Вестник Брянского государственного технического университета. – 2008. – № 1. – С. 61–66.

5. Аверченков, В. И. Аудит информационной безопасности: учеб. пособие для вузов / В. И. Аверченков. – 2-е изд., стер. – Брянск: БГТУ, 2010. – 268 с. – ISBN 978-89838-487-6.

6. Аверченков, В. И. Аудит информационной безопасности: учеб. пособие для вузов / В. И. Аверченков. – 3-е изд., стер. – Москва: Флинта, 2016. – 269 с. – ISBN 978-5-9765-1256-6.

7. Банк данных угроз безопасности информации: официальный сайт. – URL: https://bdu.fstec.ru/scanoval (дата обращения: 15.05.2021). – Текст: электронный.

8. Колегов Д. Н. Проблемы синтеза и анализа графов атак [Электронный ресурс]. Режим доступа: http://www.securitylab.ru

9. Котенко И. В. Многоагентные технологии анализа уязвимостей и обнаружения вторжений в компьютерных сетях // Новости искусственного интеллекта. – 2004. – № 1. – С. 56–72.

10. Джексон П. Введение в экспертные системы. Москва: Вильямс, 2001. – 624 с.

11. Люгер Д. Ф. Искусственный интеллект, стратегии и методы решения сложных проблем [Текст] – 4-е изд. – Вильямс, 2003. – 864 с.

12. Schneier B. Attack Trees. – Dr. Dobbs Journal, December 1999.

13. Управление рисками: обзор употребительных подходов //Jet Info, №12, 2006 г.

14. Мониторинг и аудит информационной безопасности автоматизированных систем / В. В. Кульба, А. Б. Шелков, Ю. М. Гладков, С. В. Павельев. – Москва: ИПУ им. В.А. Трапезникова РАН, 2009. – 94 с. – ISBN 5-201-15025-8.

15. Макаренко, С. И. Аудит безопасности критической инфраструктуры специальными информационными воздействиями: монография / С. И. Макаренко. – Санкт-Петербург: Наукоемкие технологии, 2018. – 122 с. – ISBN 978-5-6041427-8-3.

16. Nmap network scaning: официальный сайт. – URL: https://nmap.org/book/ (дата обращения: 18.05.2021). – Текст: электронный.

17. Организация аудита информационной безопасности: сайт – URL: https://accounting.fa.ru/jour/article/viewFile/129/130.pdf (дата обращения: 22.05.2021). – Текст: электронный.

18. Различные приемы сканирования портов: сайт. – URL: https://nmap.org/man/ru/man-port-scanning-techniques.html (дата обращения: 15.05.2021). – Текст: электронный.

19. Сканер уязвимостей Nessus: сайт. – URL: https://networkguru.ru/tenable-nessus-vulnerability-scanner/ (дата обращения: 20.05.2021). – Текст: электронный.

20. Сканер уязвимостей XSpider: сайт. – URL: http://www.s-t.ru/2014-03-16-16-26-58/237--xspider.html (дата обращения: 22.05.2021). – Текст: электронный.

21. Хомяков, В. А. Аудит как метод модернизации системы обеспечения информационной безопасности/В. А. Хомяков // Экономический вестник Ярославского университета. – 2013. – № 29. – С. 48–52.

22. Список литературы.

23. Уэлш. М. и др., Руководство по установке и использованию системы Linux. – Москва: ИЛКиРЛ, 1999.

24. Александр Боковой, Александр Колотов, Александр Прокудин, Алексей Новодворский, Алексей Смирнов, Анатолий Якушин, Антон Бояршинов, Антон Ионов, Вадим Виниченко, Виталий Липатов, Георгий Курячий, Даниил Смирнов, Дмитрий Аленичев, Дмитрий Левин, Илья Трунин, Кирилл Маслинский, Максим Отставнов, Мэтт Уэлш, Олег Власенко, Сергей Турчин, Станислав Иевлев, Юрий Коновалов и другие; ALT Linux снаружи. ALT Linux изнутри, ISBN 5-9706-0029-6, Издатель: ДМК пресс, 2006 г. Москва.

25. Марк Г. Собелл, Практическое руководство по Red Hat Linux: Fedora Core и Red Hat Enterprise Linux, 2-е издание (Practical Guide to Red Hat Linux: Fedora Core and Red Hat Enterprise Linux), 1072 стр., с ил.; ISBN 5-8459-0841-8, 0-13- 147024-8; формат 70x100/16;твердый переплет DVD-ROM; 2005, 2 кв.; Вильямс.

26. Разработка приложений в среде Linux. Программирование для linux, 2-е издание, Майкл К. Джонсон, Эрик В. Троан.

27. Руководство администратора Linux. Установка и настройка. 2-е издание, Эви Немет, Гарт Снайдер, Трент Хейн.

28. Linux. Библия пользователя, Кристофер Негус.

29. Linux для чайников , 6-е издание, Ди-Анн Лебланк.

30. Разработка ядра Linux, 2-е издание, Роберт Лав.

31. Библиотека Qt 4. Программирование прикладных приложений в среде Linux., Чебота рев Арсений Викторович.

32. Red Hat Linux Fedora 4. Полное руководство, Пол Хадсон, Эндрю Хадсон, Билл Болл, Хойт Дафф.

33. Искусство программирования для Unix, Эрик С. Реймонд.

34. Linux для «чайников», 5-е издание, Ди-Анн Лебланк.

35. Red Hat Linux. Секреты профессионала, Наба Баркакати.

36. Использование Linux, Apache, MySQL и PHP для разработки Web- приложений, Джеймс Ли, Брент Уэр.

37. Секреты хакеров. Безопасность сетей – готовые решения, 4-е издание, Стюарт Мак-Клар, Джоэл Скембрей, Джордж Курц.

38. FreeBSD: полный справочник, Родерик Смит.

39. Секреты хакеров. Безопасность Linux – готовые решения, 2-е издание, Брайан Хатч, Джеймс Ли, Джордж Курц.

40. Red Hat Linux 8. Библия пользователя, Кристофер Негус.

41. Серверы Linux. Самоучитель, Птицын Константин Александрович.

42. Безопасность Linux, 2-е издание, Скотт Манн, Эллен Л. Митчелл, Митчелл Крелл.

43. Сетевые средства Linux, Родерик Смит.

44. Руководство администратора Linux, Эви Немет, Гарт Снайдер, Трент Хейн.

45. Сети TCP/IP, том 3. Разработка приложений типа клиент/сервер для Linux/POSIX, Дуглас Камер, Дэвид Л. Стивенс.

46. Секреты хакеров. Безопасность Linux – готовые решения, Брайан Хатч, Джеймс Ли, Джордж Курц.

47. Программирование для Linux. Профессиональный подход, Марк Митчелл, Джеффри Оулдем, Алекс Самьюэл.

48. Использование Linux, 6-е издание. Специальное издание, Дэвид Бендел, Роберт Нейпир.

49. Создание сетевых приложений в среде Linux, Шон Уолтон.

50. Освой самостоятельно Linux за 24 часа, 3-е издание.

51. Система электронной почты на основе Linux. Руководство администратора, Ричард Блам.

52. Системное администрирование Linux, М. Карлинг, Стефан Деглер, Джеймс Деннис.

Локальный электронный методический материал

Владислав Владимирович Подтопельный

БЕЗОПАСНОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

Редактор С. Кондрашова
Корректор Т. Звада

Уч.-изд. л. 14,3. Печ. л. 11,3.

Федеральное государственное
бюджетное образовательное учреждение высшего образования
«Калининградский государственный технический университет»,
236022, Калининград, Советский проспект, 1