Федеральное агентство по образованию
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГОУВПО «АмГУ»
Факультет математики и информатики

*УТВЕРЖДАЮ*
Зав. кафедрой МАиМ
Т.В. Труфанова
«__» _____ 200__г.

# Администрирование и защита локальных сетей

## *Учебно – методический комплекс дисциплины*

для специальности 010501 – «Прикладная математика»

Составитель: Павлов М. С.

Благовещенск
2008

*ББК*

К

*Печатается по решению редакционно-издательского совета факультета математики и информатики Амурского государственного университета*

Павлов М.С..

Администрирование и защита локальных сетей**.** Учебно – методический комплекс дисциплины для студентов очной формы обучения специальности 010501 – «Прикладная математика». – Благовещенск: Амурский гос. ун–т, 2008. − 57 с.

Учебно – методический комплекс дисциплины " Администрирование и защита локальных сетей" содержит рабочую программу дисциплины, краткий конспект лекций, материалы для проведения лабораторных занятий, и библиографический список.

# I. РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ.

По дисциплине «Администрирование и защита локальных сетей»

Для специальности 010501 – «Прикладная математика»

Курс 5 Семестр 9

Лекций 34 час. Экзамен (нет)

Практические занятия 34 час. Зачет 9 семестр

Лабораторные занятия (нет)

Самостоятельная работа 44 час.

Типовой расчет 9 семестр

Всего 112 час.

Составитель Павлов М.С. ассистент кафедры МАиМ

Факультет математики и информатики

Кафедра математического анализа и моделирования

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### 1.1 ЦЕЛИ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ.

Основной целью дисциплины является формирование у будущих специалистов практических навыков по обеспечению безопасности и надежности работы компьютеров, объединенных в локальные сети, навыков по администрированию серверов и рабочих станций, их защите от несанкционированного доступа и вредоносных программ, навыков работы с персоналом рабочих станций по правилам хранения паролей и порядку пользования доступными сервисами, обучение работе с научно-технической литературой и технической документацией по администрированию и защите ПЭВМ.

## 1.2ЗАДАЧИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Задачей изучения дисциплины является реализация требований, установленных в квалификационной характеристике, в подготовке специалистов 010501 в области использования вычислительной техники и ее программного обеспечения в системах машинной обработки информации, проектирования и разработки этих систем.

**Математик, системный программист** должен знать и уметь использовать принципы организации, состав и схемы работы операционных систем, принципы управления ресурсами, методы организации файловых систем, принципы построения сетевого взаимодействия, основные методы разработки программного обеспечения;

**Математик, системный программист** должен имеет опыт работы на различных типах ЭВМ, применения стандартных алгоритмических языков, использования приближенных методов и стандартного программного обеспечения для решения прикладных задач.

**Математик, системный программист** должен обладать знаниями и умениями, позволяющими применять современные математические методы и программное обеспечение для решения задач науки, техники, экономики, и управления и использования информационных технологий в проектно-конструкторской, управленческой и финансовой деятельности.

**Математик, системный программист** должен обладать теоретическими знаниями и практическими навыками, соответствующими той специализации, по которой он проходил подготовку.

**Математик, системный программист** должен быть способен к совершенствованию своей профессиональной деятельности в области прикладной математики и информатики.

Дисциплина «Администрирование и защита локальных сетей» тесно связана с изучением курсов государственного образовательного стандарта «Информатика», «Системное и прикладное программное обеспечение», и «Операционные системы и сетевые технологии» и является основой для

изучения дальнейших дисциплин, использующих ЭВМ и практику программирования.

## 2. ТЕМАТИКА ЛЕКЦИЙ ПО КУРСУ «АДМИНИСТРИРОВАНИЕ И ЗАЩИТА ЛОКАЛНЫХ СЕТЕЙ» (34 часа)

### 2.1 Оценка информационных ресурсов вычислительной сети (2 часа)

Оценка информационных ресурсов. Жизненный цикл приложений, операционных систем и аппаратного обеспечения ПК. Инвентаризация ресурсов.

### 2.2 Безопасность информации (4 часа)

Понятие безопасности информации, Классификация угроз безопасности информации. Формы атак на объекты информационных систем. Анализ угроз и каналов утечки информации. Принципы политики информации. Виды политики информации. Модели типовых политик безопасности: модели на основе дискрементных компонент, модели на основе анализа угроз системы, модели конечных состояний. Классификация способов защиты информации. Структура системы защиты информации. Основные концепции безопасности в операционных системах. Ядро безопасности. Проблемы безопасности в Windows 2000 и XP. Защита информации в Unix-системах.

### 2.3 Защита информации в сетях (4 часа)

Проблема защиты компьютерных сетей. Файлы и базы данных как информационные объекты защиты. Управление доступа. Безопасность программного обеспечения. Понятие брандмауэра. Защита данных при передаче по каналам сети. Защита электронной почты.

### 2.4Многоуровневая защита корпоративных сетей (4 часа)

Концепция безопасности модели OSI. Механизмы обеспечения безопасности. Обзор средств защиты информации в компьютерных сетях.

### 2.5Администрирование системы безопасности (10 часов)

Принципы организации и контроля системы защиты. Управление защитой. Административная группа управления защитой. Реализация политики безопасности. Администрирование сетей. Базовые технологии безопасности – аутентификация, авторизация, ресурсные квоты, аудит. Учетные записи и защита рабочих станций. Типы учетных записей: локальные и доменные учетные записи, перемещаемый профиль, профиль по умолчанию, групповые учетные записи, учетные записи служб, резервные учетные записи. Управление учетными записями. Квоты дискового пространства. Журнализация.

### 2.6Администрирование серверов (10 часов)

Установка, настройка и администрирование файлового сервера. Политика назначения имен пользователей, прав доступа к своим и общим ресурсам сервера. Установка, конфигурирование и настройка сервера приложений. Настройка и администрирование «тонкого» клиента. Установка и администрирование прокси-сервера. Политика безопасности при работе в Интернете. Защита от вирусов, троянских программ и несанкционированного доступа. Настройка почтового сервера. Проблема безопасности почтовых сообщений. Защита от вредоносных вложений и спама.

## 3. ТЕМАТИКА ЛАБОРАТОРНЫХ ЗАНЯТИЙ (34 часа)

1. Настройка локального профиля рабочих станций. Права доступа к программам и ресурсам (2 часа).
2. Процедура установки и настройки сервера сети (2 часа)
3. Администрирование учетных записей (2 часа)
4. Защита файлов и каталогов (2 часа)
5. Защита реестра (2 часа)
6. Установка и настройка журналов аудита (2 часа)
7. Работа с журналами аудита. Поиск нужной информации в журналах с помощью специальных программ и из командной строки (4 часа)
8. Работа с реестром рабочих станций (2 часа)

9. Работа со сканером сети. Защита от сканера (2 часа)
10. Передача информации по сети. Приемы работы со снифферером. Защита от сниффера (2 часа)
11. Настройка файерволла (2 часа)
12. Установка и настройка антивирусных программ. Настройка обновления антивирусных баз локальных машин из серверной папки (2 часа)
13. Настройка файлового сервера (2 часа)
14. Настройка прокси-сервера (2 часа)
15. Настройка почтового сервера (2 часа)
16. Приемы работы при восстановлении системы после сбоев. Работа с образами дисков ( программы Ghost Acronis True Image) (2 часа)
17. Правила и приемы резервного копирования информации (2 часа)

## 4. ТЕМАТИКА ЗАДАНИЙ ДЛЯ ТИПОВОГО РАСЧЕТА

1) Настроить локальные профили пользователя. Создать профили пользователей для работы с математическими пакетами (профиль Mathematica), для работы с пакетом Microsoft Office (профиль Office), профиль для работы с web-станциями (web). Результаты работы. Представить в виде снимков с экрана своего компьютера.
2) Создать загрузочный диск (Live CD) операционных систем Windows 98, Windows 2000, Windows XP (на выбор). Предусмотреть на диске средства антивирусной защиты и программы для работы с диском.
3) Создать диск аварийного восстановления системы при помощи программ создания образов дисков.
4) Создать компакт-диск автоматизированной установки (Unattended CD) ОС Windows XP. Предусмотреть автоматическую установку ОС со всеми сервис-паками, всех драйверов, пакета Microsoft Office, Total Commander, антивирусного ПО. Результаты представить в виде готового CD.

## 5. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ (44 часа)

1) Создание загрузочного диска аварийного восстановления (6 часа)
2) Создание диска автоматизированной установки операционной системы (Unattended CD). Работа с файлом winnt.sif. Режимы установки. Создание файлов ответов (8 часов).
3) Настройка сервера сети Samba в среде ОС Linux (4 часа).
4) Администрирование ОС Linux (6 часов)
5) Выполнение задания типового расчета (20 часов)

## 6. ВОПРОСЫ К ЗАЧЕТУ

1) Жизненный цикл приложений.
2) Жизненный цикл ОС.
3) Жизненный цикл аппаратного обеспечения ПК.
4) Инвентаризация ресурсов.
5) Понятие безопасности информации. Классификация угроз безопасности информации.
6) Формы атак на объекты информационных систем. Анализ угроз и каналов утечки информации.
7) Принципы политики информации. Виды политики информации. Модели типовых политик безопасности: модели на основе дискрементных компонент, модели на основе анализа угроз системе, модели конечных состояний.
8) Классификация способов защиты информации. Структура системы защиты информации.
9) Основные концепции безопасности в операционных системах. Ядро безопасности.
10) Проблемы безопасности в Windows 2000 и XP.
11) Защита информации в Unix-системах.

12) Проблема защиты компьютерных сетей. Файлы и базы данных как информационные объекты защиты.
13) Управление доступа. Безопасность программного обеспечения.
14) Понятие брандмауэра. Настройка брандмауэра.
15) Процедура установки и настройки сервера сети.
16) Защита данных при передаче по каналам сети. Защита электронной почты.
17) Администрирование сетей. Базовые технологии безопасности – аутентификация, авторизация, ресурсные квоты, аудит.
18) Учетные записи и защита рабочих станций.
19) Администрирование учетных записей.
20) Защита файлов и каталогов.
21) Защита реестра.
22) Настройка локального профиля рабочих станций. Права доступа к программам и ресурсам.
23) Типы учетных записей: локальные и доменные учетные записи, перемещаемый профиль, профиль по умолчанию, групповые учетные записи, учетные записи служб, резервные учетные записи.
24) Управление учетными записями.
25) Квоты дискового пространства.
26) Журнализация.
27) Установка и настройка журналов аудита.
28) Работа с журналами аудита. Поиск нужной информации в журналах с помощью специальных программ и из командной строки.
29) Установка, настройка и администрирование файлового сервера.
30) Политика назначения имен пользователей, прав доступа к своим и общим ресурсам сети.
31) Установка, конфигурирование и настройка сервера приложений.
32) Настройка и администрирование «тонкого» клиента.
33) Установка и администрирование прокси-сервера.

34) Политика безопасности при работе в Интернете.

35)Защита от вирусов, троянских программ и несанкционированного доступа.

36)Настройка почтового сервера.

37)Проблема безопасности почтовых сообщений. Защита от вредоносных вложений и спама.

38)Работа со сканером сети. Защита от сканера

39) Передача информации по сети. Приемы работы со сниффером. Защита от сниффера.

40) Настройка файерволла.

41)Установка и настройка антивирусных программ. Настройка обновления антивирусных баз локальных машин из серверной папки

42)Настройка файлового сервера

43)Настройка прокси-сервера. Настройка почтового сервера.

44) Приемы работы при восстановлении системы после сбоев. Работа с образами дисков ( программы Ghost Acronis True Image)

45)Правила и приемы резервного копирования информации

## 6.1ТРЕБОВАНИЯ К ЗНАНИЯМ СТУДЕНТОВ, ПРЕДЬЯВЛЯЕМЫЕ ПРИ ЗАЧЕТЕ

Необходимым условием допуска на зачет является выполнение все лабораторных работ по дисциплине.

На зачете студенту предлагается ответить на два вопроса из предлагаемого выше списка и ответить на дополнительные вопросы по теме.

Знания студента оценивается на «зачтено» при ответе на оба вопроса и удовлетворительном ответе на дополнительные вопросы преподавателя.

Оценка «не зачтено» ставится при незнании одного из вопросов, предлагаемого студенту на зачетном занятии, либо при полном отсутствии ответов на дополнительные вопросы преподавателя.

## 7. ЛИТЕРАТУРА.

Основная:


1. Джонс Ален Руководство системного администратора Windows для профессионалов. – СПб.: Питер, 2000.
2. Домарев В. С. Безопасность информационных технологий. – М.: BHV, 2002.
3. Иртегов Д. Введение в операционные системы. – СПб.: БХВ – Петербург, 2002.
4. Олифер В. Г., Олифер Н. А. Сетевые операционные системы. – СПб.: Питер, 2001.
5. Петров В. Н. Информационные системы. – СПб.: Питер, 2002.
6. Щербаков А. М. Введение в теорию и практику компьютерной безопасности – М.: 2001


Дополнительная:


1. Анин Б. Защита компьютерной информации. – СПб.: БХВ – Санкт-Петербург, 2000.
2. Баурн С. Операционная система UNIX. – М.: Мир, 1986
3. Даркин Р. Профессиональная работа с MS DOS. М.:ДМК,2000.
4. Карлан Нильсен Windows 2000 изнутри. – М.: ДМК, 2000.
5. Панкратов Е. Операционная система MS DOS 6.22. Справочное пособие. – М.: Позновательная книга плюс, 2001.
6. Робачевский А. Операционная система UNIX. – М.: BHV, 1999.

## II. КРАТКИЙ КОНСПЕКТ ЛЕКЦИИ ПО ПРЕДМЕТУ.

### Оценка информационных ресурсов вычислительной сети.

**Операционная система** компьютера представляет собой комплекс взаимосвязанных программ, который действует как интерфейс между приложениями и пользователями с одной стороны, и аппаратурой компьютера с другой стороны.

В соответствии с этим определением ОС выполняет две группы функций:

- предоставление пользователю или программисту вместо реальной аппаратуры компьютера расширенной виртуальной машины, с которой удобней работать и которую легче программировать;
- повышение эффективности использования компьютера путем рационального управления его ресурсами в соответствии с некоторым критерием.

Операционная система не только предоставляет пользователям и программистам удобный интерфейс к аппаратным средствам компьютера, но и является механизмом, распределяющим ресурсы компьютера.

К числу основных ресурсов современных вычислительных систем могут быть отнесены такие ресурсы, как процессоры, основная память, таймеры, наборы данных, диски, накопители на магнитных лентах, принтеры, сетевые устройства и некоторые другие. Ресурсы распределяются между процессами. **Процесс (задача)** представляет собой базовое понятие большинства современных ОС и часто кратко определяется как программа в стадии выполнения. **Программа** — это статический объект, представляющий собой файл с кодами и данными. **Процесс** — это динамический объект, который возникает в операционной системе после того, как пользователь или сама операционная система решает «запустить программу на выполнение», то есть создать новую единицу вычислительной работы.

Управление ресурсами вычислительной системы с целью наиболее эффективного их использования является назначением операционной

системы. ОС также отслеживает и разрешает конфликты, возникающие при обращении нескольких процессов к одному и тому же устройству ввода-вывода или к одним и тем же данным. Критерий эффективности, в соответствии с которым ОС организует управление ресурсами компьютера, может быть различным. Например, в одних системах важен такой критерий, как пропускная способность вычислительной системы, в других — время ее реакции. Соответственно выбранному критерию эффективности операционные системы по-разному организуют вычислительный процесс.

**Управление ресурсами** включает решение следующих общих, не зависящих от типа ресурса задач:

- планирование ресурса — то есть определение, какому процессу, когда и в каком количестве (если ресурс может выделяться частями) следует выделить данный ресурс;
- удовлетворение запросов на ресурсы;
- отслеживание состояния и учет использования ресурса — то есть поддержание оперативной информации о том, занят или свободен ресурс и какая доля ресурса уже распределена;
- разрешение конфликтов между процессами.

Для решения этих общих задач управления ресурсами разные ОС используют различные алгоритмы, особенности которых, в конечном счете, и определяют облик ОС в целом, включая характеристики производительности, область применения и даже пользовательский интерфейс.

Безопасность данных вычислительной системы обеспечивается средствами отказоустойчивости ОС, направленными на защиту от сбоев и отказов аппаратуры и ошибок программного обеспечения, а также средствами защиты от несанкционированного доступа. В последнем случае ОС защищает данные от ошибочного или злонамеренного поведения пользователей системы.

Первым рубежом обороны при защите данных от несанкционированного доступа является процедура логического входа. Операционная система должна убедиться, что в систему пытается войти пользователь, вход которого разрешен администратором. Функции защиты ОС тесно связаны с функциями **администрирования**, так как именно администратор определяет права пользователей при их обращении к разным ресурсам системы — файлам, каталогам, принтерам, сканерам и т. п. Кроме того, администратор ограничивает возможности пользователей в выполнении тех или иных системных действий. Например, пользователю может быть запрещено выполнять процедуру завершения работы ОС, устанавливать системное время, завершать чужие процессы, создавать учетные записи пользователей, изменять права доступа к некоторым каталогам и файлам. Администратор может также урезать возможности пользовательского интерфейса, убрав, например, некоторые пункты из меню операционной системы, выводимого на дисплей пользователя.

Важным средством защиты данных являются функции **аудита** ОС, заключающиеся в фиксации всех событий, от которых зависит безопасность системы. Например, попытки удачного и неудачного логического входа в систему, операции доступа к некоторым каталогам и файлам, использование принтеров и т. п. Список событий, которые необходимо отслеживать, определяет администратор ОС.

Поддержка отказоустойчивости реализуется операционной системой, как правило, на основе **резервирования**. Чаще всего в функции ОС входит поддержание нескольких копий данных на разных дисках или разных дисковых накопителях. Резервируются также принтеры и другие устройства ввода-вывода. При отказе одного из избыточных устройств операционная система должна быстро и прозрачным для пользователя образом произвести реконфигурацию системы и продолжить работу с резервным устройством. Особым случаем обеспечения отказоустойчивости является использование нескольких процессоров, то есть мультипроцессирование, когда система

продолжает работу при отказе одного из процессоров, хотя и с меньшей производительностью.

В общем смысле под инвентаризацией можно понимать процедуру сбора информации о пользователях и ресурсах компьютерной системы (компьютерной сети). Процесс инвентаризации предполагает установку активного соединения с исследуемой системой и генерацию направленных запросов. Такая деятельность может и должна регистрироваться исследуемой системой. Инвентаризация может быть полезна как для администратора компьютерной сети, так и для нарушителя.

Информация, которую можно получить при инвентаризации, можно разделить на следующие категории.

- Сетевые ресурсы, в том числе открытые для совместного доступа.
- Пользователи и группы пользователей.
- Приложения и идентификационные маркеры (banner).

Методика инвентаризации в значительной степени зависит от операционной системы. Зная, какая информация может заинтересовать нарушителя и насколько хорошо ваша система ее скрывает, вы можете предпринять ответные меры, которые позволят защитить самые уязвимые участки.

**Безопасность информации.**

Любой ресурс компьютерной системы может рассматриваться как некоторая информационная структура. Все ресурсы можно условно разделить на разделяемые (ресурсы которые допускают совместное одновременное использование несколькими процессами (потоками)) и не разделяемые (ресурсы, одновременное использование которых несколькими процессами недопустимо по тем или иным причинам).

Файлы — это частный, хотя и самый популярный, вид разделяемых ресурсов, доступ к которым операционная система должна контролировать. Существуют и другие виды ресурсов, с которыми пользователи работают в режиме совместного использования. Прежде всего, это различные внешние

устройства: принтеры, модемы, графопостроители и т. п. Область памяти, используемая для обмена данными между процессами, также является примером разделяемого ресурса. Да и сами процессы в некоторых случаях выступают в этой роли, например, когда пользователи ОС посылают процессам сигналы, на которые процесс должен реагировать.

Во всех этих случаях действует общая схема: пользователи пытаются выполнить с разделяемым ресурсом определенные операции, а ОС должна решать, имеют ли пользователи на это право. Пользователи являются субъектами доступа, а разделяемые ресурсы — объектами. Пользователь осуществляет доступ к объектам операционной системы не непосредственно, а с помощью прикладных процессов, которые запускаются от его имени. Для каждого типа объектов существует набор операций, которые с ними можно выполнять. Система контроля доступа ОС должна предоставлять средства для задания прав пользователей по отношению к объектам дифференцированно по операциям, например, пользователю может быть разрешена операция чтения и выполнения файла, а операция удаления — запрещена.

Во многих операционных системах реализованы механизмы, которые позволяют управлять доступом к объектам различного типа с единых позиций. Так, представление устройств ввода-вывода в виде специальных файлов в операционных системах UNIX является примером такого подхода: в этом случае при доступе к устройствам используются те же атрибуты безопасности и алгоритмы, что и при доступе к обычным файлам и каталогам. Еще дальше продвинулась в этом направлении операционная система Windows NT. В ней используется унифицированная структура — **объект безопасности.** Данная структура создается не только для файлов и внешних устройств, но и для любых разделяемых ресурсов: секций памяти, синхронизирующих примитивов типа семафоров и мьютексов и т. п. Это позволяет использовать в Windows NT для контроля доступа к ресурсам любого вида общий модуль ядра — **менеджер безопасности**.

**Угроза безопасности информации** – потенциально возможное воздействие на информацию, которое прямо или косвенно может нанести урон пользователям или владельцам информации

Основные свойства информации:

- Конфиденциальность;
- Целостность;
- Доступность;

Классификация угроз информации:

- Угроза конфиденциальности информации;
- Угроза целостности информации;
- Угроза доступности информации;
- Угроза раскрытия параметров компьютерной системы;

Опасность угрозы раскрытия параметров компьютерной системы и конфиденциальности нередко происходит с использованием **неблагоприятных информационных потоков (каналов утечки информации)**. Выделяют два основных вида каналов утечки информации: по времени и по памяти.

**Политика безопасности -** определяет правила обращения с информацией так, чтобы исключить или снизить ущерб, в случае реализации угроз. Единая политика нужна также, для того чтобы исключить противоречия между правилами обращения с одной и той же информацией в разных подразделениях организации.

Политика информационной безопасности должна отвечать следующим требованиям:

- интегрированность;
- комплексность;
- достаточность.

Политики безопасности должны быть подробными, четко определёнными и обязательными для компьютерной системы. Есть несколько основных политик безопасности:

- Избирательный (или дискреционный) доступ имеет место, когда для каждого объекта сам владелец может определить допустимые операции с объектами. Этот подход называется также произвольным (от `discretionary` — предоставленный на собственное усмотрение) доступом, так как позволяет администратору и владельцам объектов определить права доступа произвольным образом, по их желанию. Между пользователями и группами пользователей в системах с избирательным доступом нет жестких иерархических взаимоотношений, то есть взаимоотношений, которые определены по умолчанию и которые нельзя изменить. Исключение делается только для администратора, по умолчанию наделяемого всеми правами. **Матрица доступов –** матрица размером |O|x|S|, строки которой соответствуют объектам защиты, а столбцы субъектам защиты. При этом элемент матрицы задает права доступа для пользователя на объект из множества прав доступа.
- Мандатный доступ (от `mandatory` — обязательный, принудительный) — это такой подход к определению прав доступа, при котором система наделяет пользователя определенными правами по отношению к каждому разделяемому ресурсу (в данном случае файлу) в зависимости от того, к какой группе пользователь отнесен. От имени системы выступает администратор, а владельцы объектов лишены возможности управлять доступом к ним по своему усмотрению. Все группы пользователей в такой системе образуют строгую иерархию, причем каждая группа пользуется всеми правами группы более низкого уровня иерархии, к которым добавляются права данного уровня. Членам какой-либо группы не разрешается предоставлять свои права членам групп более низких уровней иерархии. Мандатный способ доступа близок к схемам, применяемым для

доступа к секретным документам: пользователь может входить в одну из групп, отличающихся правом на доступ к документам с соответствующим грифом секретности, например «для служебного пользования», «секретно», «совершенно секретно» и «государственная тайна». При этом пользователи группы «совершенно секретно» имеют право работать с документами «секретно» и «для служебного пользования», так как эти виды доступа разрешены для более низких в иерархии групп. Однако сами пользователи не распоряжаются правами доступа — этой возможностью наделен только особый чиновник учреждения.

**Ядро безопасности** это локализованная, минимизированная, четко ограниченная и надежно изолированная совокупность программно-аппаратных механизмов, доказательно правильно реализующих функции диспетчера доступа (и ряд других сопутствующих служебных функций, например, доверенных процессов).

Осенью 1999 года компания Microsoft открыла доступ через Internet к нескольким тестовым серверам с бета-версией операционной системы Windows 2000 Server на узле Windows2000test.com, предлагая всем желающим попытаться взломать этот программный продукт. Несколько недель спустя, после многочисленных успешных атак со стороны хакеров, этот эксперимент был отменен. Правда, следует отметить, что хакерам не удалось достичь уровня операционной системы. Они лишь обнаружили бреши в приложении Guestbook, основанном на использовании Web-технологии и работающем "на переднем крае" операционной системы. Аналогичные результаты были получены в процессе проведения других подобных тестов.

Такие тесты включают множество параметров, и мы не станем обсуждать их реальные возможности по выявлению преимуществ системы безопасности Windows 2000 по сравнению с конкурирующими программными продуктами. Однако в результате такого тестирования

совершенно очевидно одно: правильно сконфигурированные серверы под управлением Windows 2000 на уровне операционной системы столь же надежны, как любые другие серверные платформы. Наиболее уязвимым местом этой операционной системы является уровень приложений, посредством которого можно обойти системные средства обеспечения безопасности. Защищенность Windows 2000 подкреплена множеством новых средств обеспечения безопасности, встроенных в операционные системы Windows нового поколения. К ним относятся реализация собственного протокола IP Security (IPSec), кодирующая файловая система EPS (Encrypting File System), возможность выбора политики безопасности для групп пользователей, шаблоны защиты Security Templates, средство конфигурирования и анализа политики безопасности, а также возможность централизованного удаленного управления на основе использования сервера удаленной аутентификации RADIUS (Remote Authentication Dial-In User Service) и многое другое. При этом бросается в глаза соответствие всех средств обеспечения безопасности общепризнанным стандартам, что свидетельствует об изменении общеизвестной сепаратистской политики компании Microsoft в вопросах обеспечения безопасности.

Перечисленные технологии обеспечили пользователям NT долгожданные возможности, но достаточно ли корректно они реализованы? Радикальные изменения в Windows 2000, особенно касающиеся нового средства Active Directory (AD), заставят сетевых администраторов изрядно потрудиться при переходе на новую операционную систему. И как показывает опыт использования, из-за неполной реализации протоколов и проблем совместимости с более ранними версиями говорить о полной безопасности операционной системы можно после появления сервисного пакета Service Pack 4.

UNIX — это сложная система, для адекватной защиты которой необходимо предпринимать целый ряд комплексных мероприятий. Мощь и элегантность UNIX обеспечивают ее популярность, однако они же являются

и причиной ее уязвимости. Мириады методов удаленного и локального взлома позволяют злоумышленникам нарушать подсистему безопасности даже самых защищенных систем UNIX. Чуть ли не ежедневно обнаруживаются новые методы взлома путем переполнения буфера. Программисты мало задумываются о безопасности, а средства обнаружения несанкционированных действий устаревают практически в течение нескольких недель.

**Защита информации в сетях**.

На современном рынке преобладает два типа брандмауэров: программные посредники (application proxy) и шлюзы фильтрации пакетов (packet filtering gateways). Хотя программные посредники считаются более надежными, чем шлюзы фильтрации пакетов, их ограниченность и невысокая производительность обуславливают их применение в основном к исходящему трафику, а не к входящему потоку сообщений, поступающему на Web-серверы многих компаний. В то же время, шлюзы фильтрации пакетов или более сложные шлюзы с сохранением состояния (stateful) можно найти во многих крупных организациях, в которых высокие требования предъявляются к исходящему трафику.

После появления первого брандмауэра они стали защищать многочисленные сети от глаз и нападок злоумышленников. Однако брандмауэры все же нельзя рассматривать как панацею от всех бед. Ежегодно новые слабые места обнаруживаются в системе защиты практически каждого брандмауэра, присутствующего на рынке. Что еще хуже, большинство брандмауэров зачастую неправильно настраивается, обслуживается и проверяется.

Идентификация брандмауэров

Почти каждый брандмауэр имеет свои отличительные особенности. Поэтому с помощью сканирования портов, инвентаризации и сбора идентификационных маркеров взломщики могут правильно определить тип,

версию и набор правил практически каждого брандмауэра в сети. Почему так важна подобная идентификация? Как только вся эта информация будет получена, взломщик может приступать к ее анализу, поиску уязвимых мест.

### Прямое сканирование

Самый простой способ поиска брандмауэров заключается в сканировании определенных портов, используемых ими по умолчанию. Некоторые современные брандмауэры можно уникально идентифицировать, выполнив простое сканирование портов. Для этого необходимо лишь знать, что именно вы хотите найти. Например, брандмауэры Firewall-1 компании Checkpoint ожидают поступления запросов с TCP-портов 256, 257 и 258, а Proxy Server компании Microsoft обычно прослушивает TCP-порты с номерами 1080 и 1745. Обладая такой информацией, поиск этих типов брандмауэров окажется тривиальным

Для предотвращения подобной ситуации необходимо заблокировать попытки такого сканирования на пограничном маршрутизаторе или воспользоваться одним из средств выявления вторжений (свободно распространяемым или коммерческим). Однако и в этом случае нельзя предотвратить простое сканирование портов, поскольку по умолчанию большинство систем IDS не позволяет обнаружить подобную деятельность. Так что перед их использованием необходимо выполнить соответствующую настройку.

### Отслеживание маршрута

Более скрытый и изощренный метод поиска брандмауэров в сети заключается в использовании утилиты traceroute. Для поиска каждого сегмента пути к целевому узлу можно воспользоваться утилитой traceroute системы UNIX или аналогичной утилитой tracert.exe системы NT. Затем на

основании полученной информации можно сделать некоторые логические предположения. В версии утилиты traceroute из системы Linux имеется параметр -I, при указании которого для поиска сегментов будут посылаться пакеты ICMP, а не UDP-пакеты, используемые по умолчанию.

Для того чтобы предотвратить получение информации с помощью утилиты traceroute, запретите передачу ответных пакетов на пакеты с истекшим временем TTL на всех брандмауэрах и маршрутизаторах, на которых это возможно. Однако помните о том, что полностью решить этот вопрос можно далеко не всегда.

### Сбор маркеров

Сканирование портов позволяет определить местоположение брандмауэров, однако многие из них, подобно продуктам компаний Checkpoint и Microsoft, не прослушивают порты, установленные по умолчанию. Так что для подтверждения имеющихся сведений требуется получить дополнительные данные. В главе 3 подробно рассматривались методы получения имен запущенных приложений и их версий. Для этого необходимо подключиться к активным службам и извлечь связанные с ними идентификационные маркеры. Многие популярные брандмауэры предоставляют всю необходимую информацию сразу же после установки с ними соединения. Многие промежуточные узлы информируют злоумышленника о том, что они являются брандмауэрами, а некоторые из них дополнительно сообщают свой тип, а также версию. Например, при подключении с помощью утилиты netcat к порту 21 (FTP) узла, который, очевидно, является брандмауэром, можно получить некоторую важную информацию.

Для того чтобы избежать утечки информации, необходимо ограничить количество данных, предоставляемых по внешним запросам. К каждому маркеру можно добавить также сообщение о юридической

ответственности, а, кроме того, все попытки получения этой информации должны регистрироваться в системных журналах. Специфика изменения маркеров, используемых по умолчанию, сильно зависит от типа используемого брандмауэра, так что перед выполнением подобных действий внимательно прочтите документацию или проконсультируйтесь с производителем.

**Многоуровневая защита корпоративных сетей.**

Организация взаимодействия между устройствами в сети является сложной проблемой, она включает много аспектов, начиная с согласования уровней электрических сигналов, формирования кадров, проверки контрольных сумм и кончая вопросами аутентификации приложений. Для ее решения используется универсальный прием — разбиение одной сложной задачи на несколько частных, более простых задач. Средства решения отдельных задач упорядочены в виде иерархии уровней. Для решения задачи некоторого уровня могут быть использованы средства непосредственно примыкающего нижележащего уровня. С другой стороны, результаты работы средств некоторого уровня могут быть переданы только средствам соседнего вышележащего уровня.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют две машины, то есть в данном случае необходимо организовать согласованную работу двух «иерархий». При передаче сообщений оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать способ кодирования электрических сигналов, правило определения длины сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого уровня передачи битов до самого высокого уровня, предоставляющего услуги пользователям сети.

Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются протоколом.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другое в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть интерфейсом. Интерфейс определяет услуги, предоставляемые данным уровнем соседнему уровню.

В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы — модулей соседних уровней в одном узле.

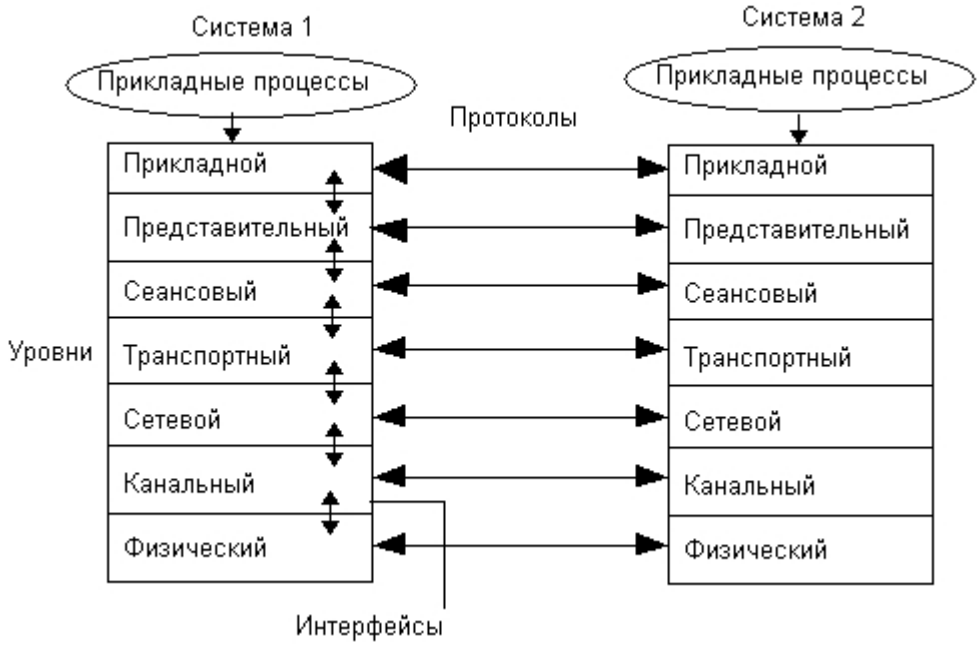
Средства каждого уровня должны отрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями. Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется стеком коммуникационных протоколов.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средства протоколы верхних уровней, как правило, чисто программными средствами.

В начале 80-х годов - ряд международных организаций по стандартизации — ISO, ITU-T и некоторые другие - разработали модель, которая сыграла значительную роль в развитии сетей. Эта модель называется моделью взаимодействия открытых систем (Open System interconnection, OSI), или моделью OSI. Модель OSI определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень.

В модели OSI средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой,

канальный и физический. Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств.




Модель взаимодействия открытых систем ISO/OSI

**Физический уровень**

Физический уровень (Physical layer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с

волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

**Канальный уровень**

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня (Data Link layer) является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный уровень обеспечивает корректность передачи кадров, помещая для выделения каждого кадра специальную последовательность бит в его начало и конец, а также вычисляет контрольную сумму, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру.

В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

В глобальных сетях, которые в отличие от локальных сетей редко обладают регулярной топологией, канальный уровень обеспечивает обмен сообщениями только между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов «точка-точка» (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B.

**Сетевой уровень**

Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной; структурой связей.

Сети соединяются между Собой специальными устройствами, называемыми маршрутизаторами. Маршрутизатор — это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Для того чтобы передать сообщения сетевого уровня, или, как их принято называть, пакеты (packets), от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет. Проблема выбора наилучшего пути называется маршрутизацией, и ее решение является одной из главных задач сетевого уровня.

Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол межсетевого обмена пакетами IPX стека Novell.

**Транспортный уровень**

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Работа транспортного уровня (Transport layer) заключается в том, чтобы обеспечить приложениям или верхним уровням стека — прикладному и сеансовому — передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов услуг, предоставляемых транспортным уровнем. Эти виды услуг отличаются

качеством: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью к обнаружению и исправлению ошибок передачи, таких как,, искажение, потеря и дублирование пакетов.

Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети — компонентами их сетевых операционных систем. В качестве примера .транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell.

**Сеансовый уровень**

Сеансовый уровень (Session layer) обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке вместо того, чтобы начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

**Уровень представления**

Уровень представления (Presentation layer) имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда будет понятна прикладному уровню в другой системе. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия кодов символов, например кодов ASCII и EBCDIC. На этом уровне может выполняться шифрование и дешифрирование данных, благодаря которому секретность 'обмена данными

обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

**Прикладной уровень**

Прикладной уровень (Application layer) — это в действительности прости *набер разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например с помощью протокола электронной почты. Единица данных, которой оперирует Прикладной уровень, обычно называется сообщением (message).

Существует очень большое разнообразие служб прикладного уровня. Приведем в качестве примеров протоколов прикладного уровня хотя бы несколько наиболее распространенных реализаций файловых служб: NCP в операционной системе Novell NetWare, 8MB в Microsoft Windows NT, NFS, FTP и TFTP, входящие в стек TCP/IP.

## Администрирование системы безопасности.
## Администрирование серверов.

Универсальной классификации угроз не существует, возможно, и потому, что нет предела творческим способностям человека, и каждый день применяются новые способы незаконного проникновения в сеть, разрабатываются новые средства мониторинга сетевого трафика, появляются новые вирусы, находятся новые изъяны в существующих программных и аппаратных сетевых продуктах. В ответ на это разрабатываются все более изощренные средства защиты, которые ставят преграду на пути многих типов угроз, но затем сами становятся новыми объектами атак. Тем не менее, попытаемся сделать некоторые обобщения. Угрозы могут быть разделены на умышленные и неумышленные.

Неумышленные угрозы вызываются ошибочными действиями лояльных сотрудников, становятся следствием их низкой квалификации или безответственности. Кроме того, к такому роду угроз относятся последствия ненадежной работы программных и аппаратных средств системы. Так, например, из-за отказа диска, контроллера диска или всего файлового сервера могут оказаться недоступными данные, критически важные для работы предприятия. Поэтому вопросы безопасности так тесно переплетаются с вопросами надежности, отказоустойчивости технических средств. Угрозы безопасности, которые вытекают из ненадежности работы программно-аппаратных средств, предотвращаются путем их совершенствования, использования резервирования на уровне аппаратуры (RAID-массивы, многопроцессорные компьютеры, источники бесперебойного питания, кластерные архитектуры) или на уровне массивов данных (тиражирование файлов, резервное копирование).

Умышленные угрозы могут ограничиваться либо пассивным чтением данных или мониторингом системы, либо включать в себя активные действия, например нарушение целостности и доступности информации, приведение в нерабочее состояние приложений и устройств. Так, умышленные угрозы возникают в результате деятельности хакеров и явно направлены на нанесение ущерба предприятию.

В вычислительных сетях можно выделить следующие типы умышленных угроз:

- незаконное проникновение в один из компьютеров сети под видом легального пользователя;
- разрушение системы с помощью программ-вирусов;
- нелегальные действия легального пользователя;
- «подслушивание» внутри сетевого трафика.

Незаконное проникновение может быть реализовано через уязвимые места в системе безопасности с использованием недокументированных возможностей операционной системы. Эти возможности могут позволить

злоумышленнику «обойти» стандартную процедуру, контролирующую вход в сеть.

Другим способом незаконного проникновения в сеть является использование «чужих» паролей, полученных путем подглядывания, расшифровки файла паролей, подбора паролей или получения пароля путем анализа сетевого трафика. Особенно опасно проникновение злоумышленника под именем пользователя, наделенного большими полномочиями, например администратора сети. Для того чтобы завладеть паролем администратора, злоумышленник может попытаться войти в сеть под именем простого пользователя. Поэтому очень важно, чтобы все пользователи сети сохраняли свои пароли в тайне, а также выбирали их так, чтобы максимально затруднить угадывание.

Подбор паролей злоумышленник выполняет с использованием специальных программ, которые работают путем перебора слов из некоторого файла, содержащего большое количество слов. Содержимое файла-словаря формируется с учетом психологических особенностей человека, которые выражаются в том, что человек выбирает в качестве пароля легко запоминаемые слова или буквенные сочетания.

Еще один способ получения пароля — это внедрение в чужой компьютер «троянского коня». Так называют резидентную программу, работающую без ведома хозяина данного компьютера и выполняющую действия, заданные злоумышленником. В частности, такого рода программа может считывать коды пароля, вводимого пользователем во время логического входа в систему.

Программа - «троянский конь» всегда маскируется под какую-нибудь полезную утилиту или игру, а производит действия, разрушающие систему. По такому принципу действуют и программы-вирусы, отличительной особенностью которых является способность «заражать» другие файлы, внедряя в них свои собственные копии. Чаще всего вирусы поражают исполняемые файлы. Когда такой исполняемый код загружается в

оперативную память для выполнения, вместе с ним получает возможность исполнить свои вредительские действия вирус. Вирусы могут привести к повреждению или даже полной утрате информации.

Нелегальные действия легального пользователя — этот тип угроз исходит от легальных пользователей сети, которые, используя свои полномочия, пытаются выполнять действия, выходящие за рамки их должностных обязанностей. Например, администратор сети имеет практически неограниченные права на доступ ко всем сетевым ресурсам. Однако на предприятии может быть информация, доступ к которой администратору сети запрещен. Для реализации этих ограничений могут быть предприняты специальные меры, такие, например, как шифрование данных, но и в этом случае администратор может попытаться получить доступ к ключу. Нелегальные действия может попытаться предпринять и обычный пользователь сети. Существующая статистика говорит о том, что едва ли не половина всех попыток нарушения безопасности системы исходит от сотрудников предприятия, которые как раз и являются легальными пользователями сети.

«Подслушивание» внутри сетевого трафика — это незаконный мониторинг сети, захват и анализ сетевых сообщений. Существует много доступных программных и аппаратных анализаторов трафика, которые делают эту задачу достаточно тривиальной. Еще более усложняется защита от этого типа угроз в сетях с глобальными связями. Глобальные связи, простирающиеся на десятки и тысячи километров, по своей природе являются менее защищенными, чем локальные связи (больше возможностей для прослушивания трафика, более удобная для злоумышленника позиция при проведении процедур аутентификации). Такая опасность одинаково присуща всем видам территориальных каналов связи и никак не зависит от того, используются собственные, арендуемые каналы или услуги общедоступных территориальных сетей, подобных Интернету.

Однако использование общественных сетей (речь в основном идет об Интернете) еще более усугубляет ситуацию. Действительно, использование Интернета добавляет к опасности перехвата данных, передаваемых по линиям связи, опасность несанкционированного входа в узлы сети, поскольку наличие огромного числа хакеров в Интернете увеличивает вероятность попыток незаконного проникновения в компьютер. Это представляет постоянную угрозу для сетей, подсоединенных к Интернету.

Интернет сам является целью для разного рода злоумышленников. Поскольку Интернет создавался как открытая система, предназначенная для свободного обмена информацией, совсем не удивительно, что практически все протоколы стека TCP/IP имеют «врожденные» недостатки защиты. Используя эти недостатки, злоумышленники все чаще предпринимают попытки несанкционированного доступа к информации, хранящейся на узлах Интернета.

Построение и поддержка безопасной системы требует системного подхода. В соответствии с этим подходом, необходимо осознать весь спектр возможных угроз для конкретной сети и для каждой из этих угроз продумать тактику ее отражения. В этой борьбе можно и нужно использовать самые разноплановые средства и приемы — морально-этические и законодательные, административные и психологические, защитные возможности программных и аппаратных средств сети.

К **морально-этическим средствам** защиты можно отнести всевозможные нормы, которые сложились по мере распространения вычислительных средств в той или иной стране. Например, подобно тому как в борьбе против пиратского копирования программ в настоящее время в основном используются меры воспитательного плана, необходимо внедрять в сознание людей аморальность всяческих покушений на нарушение конфиденциальности, целостности и доступности чужих информационных ресурсов.

**Законодательные средства** защиты — это законы, постановления правительства и указы президента, нормативные акты и стандарты, которыми регламентируются правила использования и обработки информации ограниченного доступа, а также вводятся меры ответственности за нарушения этих правил. Правовая регламентация деятельности в области защиты информации имеет целью защиту информации, составляющей государственную тайну, обеспечение прав потребителей на получение качественных продуктов, защиту конституционных прав граждан на сохранение личной тайны, борьбу с организованной преступностью.

**Административные меры** — это действия, предпринимаемые руководством предприятия или организации для обеспечения информационной безопасности. К таким мерам относятся конкретные правила работы сотрудников предприятия, например режим работы сотрудников, их должностные инструкции, строго определяющие порядок работы с конфиденциальной информацией на компьютере. К административным мерам также относятся правила приобретения предприятием средств безопасности. Представители администрации, которые несут ответственность за защиту информации, должны выяснить, насколько безопасным является использование продуктов, приобретенных у зарубежных поставщиков. Особенно это касается продуктов, связанных с шифрованием. В таких случаях желательно проверить наличие у продукта сертификата, выданного российскими тестирующими организациями.

**Психологические меры** безопасности могут играть значительную роль в укреплении безопасности системы. Пренебрежение учетом психологических моментов в неформальных процедурах, связанных с безопасностью, может привести к нарушениям защиты. Рассмотрим, например, сеть предприятия, в которой работает много удаленных пользователей. Время от времени пользователи должны менять пароли (обычная практика для предотвращения их подбора). В данной системе выбор паролей осуществляет администратор. В таких условиях

злоумышленник может позвонить администратору по телефону и от имени легального пользователя попробовать получить пароль. При большом количестве удаленных пользователей не исключено, что такой простой психологический прием может сработать.

К **физическим средствам** защиты относятся экранирование помещений для защиты от излучения, проверка поставляемой аппаратуры на соответствие ее спецификациям и отсутствие аппаратных «жучков», средства наружного наблюдения, устройства, блокирующие физический доступ к отдельным блокам компьютера, различные замки и другое оборудование, защищающие помещения, где находятся носители информации, от незаконного проникновения и т. д. и т. п.

**Технические средства** информационной безопасности реализуются программным и аппаратным обеспечением вычислительных сетей. Такие средства, называемые также службами сетевой безопасности, решают самые разнообразные задачи по защите системы, например контроль доступа, включающий процедуры аутентификации и авторизации, аудит, шифрование информации, антивирусную защиту, контроль сетевого трафика и много других задач. Технические средства безопасности могут быть либо встроены в программное (операционные системы и приложения) и аппаратное (компьютеры и коммуникационное оборудование) обеспечение сети, либо реализованы в виде отдельных продуктов, созданных специально для решения проблем безопасности.

В разных программных и аппаратных продуктах, предназначенных для защиты данных, часто используются одинаковые подходы, приемы и технические решения. К таким базовым технологиям безопасности относятся аутентификация, авторизация, аудит и технология защищенного канала.

**Шифрование**

**Шифрование** — это краеугольный камень всех служб информационной безопасности, будь то система аутентификации или

авторизации, средства создания защищенного канала или способ безопасного хранения данных.

Любая процедура шифрования, превращающая информацию из обычного «понятного» вида в «нечитабельный» зашифрованный вид, естественно, должна быть дополнена процедурой **дешифрирования**, которая, будучи примененной к зашифрованному тексту, снова приводит его в понятный вид. Пара процедур — шифрование и дешифрирование — называется **криптосистемой**.

Информацию, над которой выполняются функции шифрования и дешифрирования, будем условно называть «текст», учитывая, что это может быть также числовой массив или графические данные.

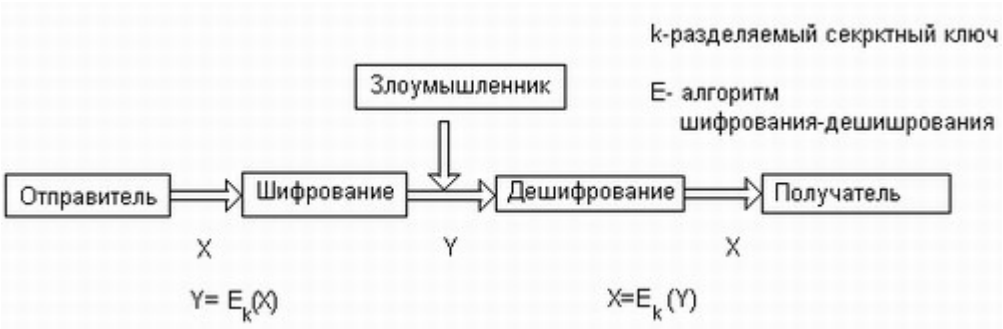
В современных алгоритмах шифрования предусматривается Наличие параметра — **секретного ключа**. В криптографии принято правило Керкхоффа: «Стойкость шифра должна определяться только секретностью ключа». Так, все стандартные алгоритмы шифрования (например, DES, PGP) широко известны, их детальное описание содержится в легко доступных документах, но от этого их эффективность не снижается. Злоумышленнику может быть все известно об алгоритме шифрования, кроме секретного ключа (следует отметить, однако, что существует немало фирменных алгоритмов, описание которых не публикуется).

Алгоритм шифрования считается раскрытым, если найдена процедура, позволяющая подобрать ключ за реальное время. Сложность алгоритма раскрытия является одной из важных характеристик криптосистемы и называется **криптостойкостью**.

Существуют два класса криптосистем — симметричные и асимметричные. В симметричных схемах шифрования (классическая криптография) секретный ключ зашифровки совпадает с секретным ключом расшифровки. В асимметричных схемах шифрования (криптография с открытым ключом) открытый ключ зашифровки не совпадает с секретным ключом расшифровки.

**Симметричные алгоритмы шифрования**

В классической модели шифрования три участника: отправитель, получатель, злоумышленник. Задача отправителя заключается в том, чтобы по открытому каналу передать некоторое сообщение в защищенном виде. Для этого он на ключе k зашифровывает открытый текст X и передает шифрованный текст Y. Задача получателя заключается в том, чтобы расшифровать Y и прочитать сообщение X. Предполагается, что отправитель имеет свой источник ключа. Сгенерированный ключ заранее по надежному каналу передается получателю. Задача злоумышленника заключается в перехвате и чтении передаваемых сообщений, а также в имитации ложных сообщений.




Модель симметричного шифрования

Модель является универсальной — если зашифрованные; данные хранятся в компьютере и никуда не передаются, отправитель и получатель совмещаются в одном лице, а в роли злоумышленника выступает некто, имеющий доступ к компьютеру в ваше отсутствие.

Наиболее популярным стандартным симметричным алгоритмом шифрования данных является DES (Data Encryption Standard). Алгоритм разработан фирмой IBM и в 1976 году был рекомендован Национальным бюро стандартов к использованию в открытых секторах экономики. Суть этого алгоритма заключается в следующем.

Данные шифруются поблочно. Перед шифрованием любая форма представления данных преобразуется в числовую. Эти числа получают путем любой открытой процедуры преобразования блока текста в число. Например, ими могли бы быть значения двоичных чисел, полученных слиянием ASCII-

кодов последовательных символов соответствующего блока текста. На вход шифрующей функции поступает блок данных размером 64 бита, он делится пополам на левую (L) и правую (R) части. На первом этапе на место левой части результирующего блока помещается правая часть исходного блока. Правая часть результирующего блока вычисляется как сумма по модулю 2 (операция XOR) левой и правой частей исходного блока. Затем на основе случайной двоичной последовательности по определенной схеме в полученном результате выполняются побитные замены и перестановки. Используемая двоичная последовательность, представляющая собой ключ данного алгоритма, имеет длину 64 бита, из которых 56 действительно случайны, а 8 предназначены для контроля ключа.

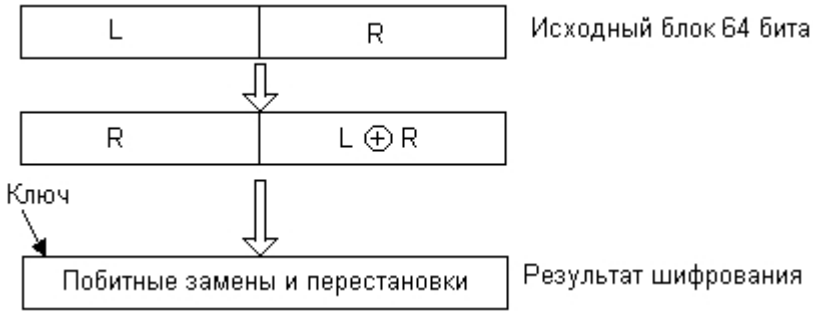



Схема шифрования по алгоритму DES

Вот уже в течение двух десятков лет алгоритм DES испытывается на стойкость. И хотя существуют примеры успешных попыток «взлома» данного алгоритма, в целом можно считать, что он выдержал испытания. Алгоритм DES широко используется в различных технологиях и продуктах безопасности информационных систем. Для того чтобы повысить криптостойкость алгоритма DES, иногда применяют его усиленный вариант, называемый «тройным DES», который включает троекратное шифрование с использованием двух разных ключей. При этом можно считать, что длина ключа увеличивается с 56 бит до 112 бит, а значит, криптостойкость алгоритма существенно повышается. Но за это приходится платить производительностью — «тройной DES» требует в три раза больше времени, чем «обычный» DES.

В симметричных алгоритмах главную проблему представляют ключи. Во-первых, криптостойкость многих симметричных алгоритмов зависит от качества ключа, это предъявляет повышенные требования к службе генерации ключей. Во-вторых, принципиальной является надежность канала передачи ключа второму участнику секретных переговоров. Проблема с ключами возникает даже в системе с двумя абонентами, а в системе с п абонентами, желающими обмениваться секретными данными по принципу «каждый с каждым», потребуется их(и-1)/2 ключей, которые должны быть сгенерированы и распределены надежным образом. То есть количество ключей пропорционально квадрату количества абонентов, что при большом числе абонентов делает задачу чрезвычайно сложной. Несимметричные алгоритмы, основанные на использовании открытых ключей, снимают эту проблему.

**Несимметричные алгоритмы шифрования**

В середине 70-х двое ученых — Винфилд Диффи и Мартин Хеллман — описали принципы шифрования с открытыми ключами.

Особенность шифрования на основе открытых ключей состоит в том, что одновременно генерируется уникальная пара ключей, таких, что текст, зашифрованный одним ключом, может быть расшифрован только с использованием второго ключа и наоборот.

В модели криптосхемы с открытым ключом также три участника: отправитель, получатель, злоумышленник. Задача отправителя заключается в том, чтобы по открытому каналу связи передать некоторое сообщение в защищенном виде. Получатель генерирует на своей стороне два ключа: открытый E и закрытый D. Закрытый ключ D (часто называемый также личным ключом) абонент должен сохранять в защищенном месте, а открытый ключ E он может передать всем, с кем он хочет поддерживать защищенные отношения. Открытый ключ используется для шифрования текста, но расшифровать текст можно только с помощью закрытого ключа. Поэтому открытый ключ передается отправителю в незащищенном виде.

Отправитель, используя открытый ключ получателя, шифрует сообщение X и передает его получателю. Получатель расшифровывает сообщение своим закрытым ключом D.

Очевидно, что числа, одно из которых используется для шифрования текста, а другое — для дешифрирования, не могут быть независимыми друг от друга, а значит, есть теоретическая возможность вычисления закрытого ключа по открытому, но это связано с огромным количеством вычислений, которые требуют соответственно огромного времени. Поясним принципиальную связь между закрытым и открытым ключами следующей аналогией.


Модель криптосхемы с открытым ключом

Пусть абонент 1 решает вести секретную переписку со своими сотрудниками на малоизвестном языке, например санскрите. Для этого он обзаводится санскритско-русским словарем, а всем своим абонентам посылает русско-санскритские словари. Каждый из них, пользуясь словарем, пишет сообщения на санскрите и посылает их абоненту 1, который переводит их на русский язык, пользуясь доступным только ему санскритско-русским словарем. Очевидно, что здесь роль открытого ключа E играет русско-санскритский словарь, а роль закрытого ключа D — санскритско-русский словарь. Могут ли абоненты 2, 3 и 4 прочитать чужие сообщения $S_2$, $S_3$, $S_4$, которые посылает каждый из них абоненту 1? Вообще-то нет, так как, для этого им нужен санскритско-русский словарь, обладателем которого является только абонент 1. Но теоретическая возможность этого имеется, так как затратив массу времени, можно прямым перебором составить санскритско-

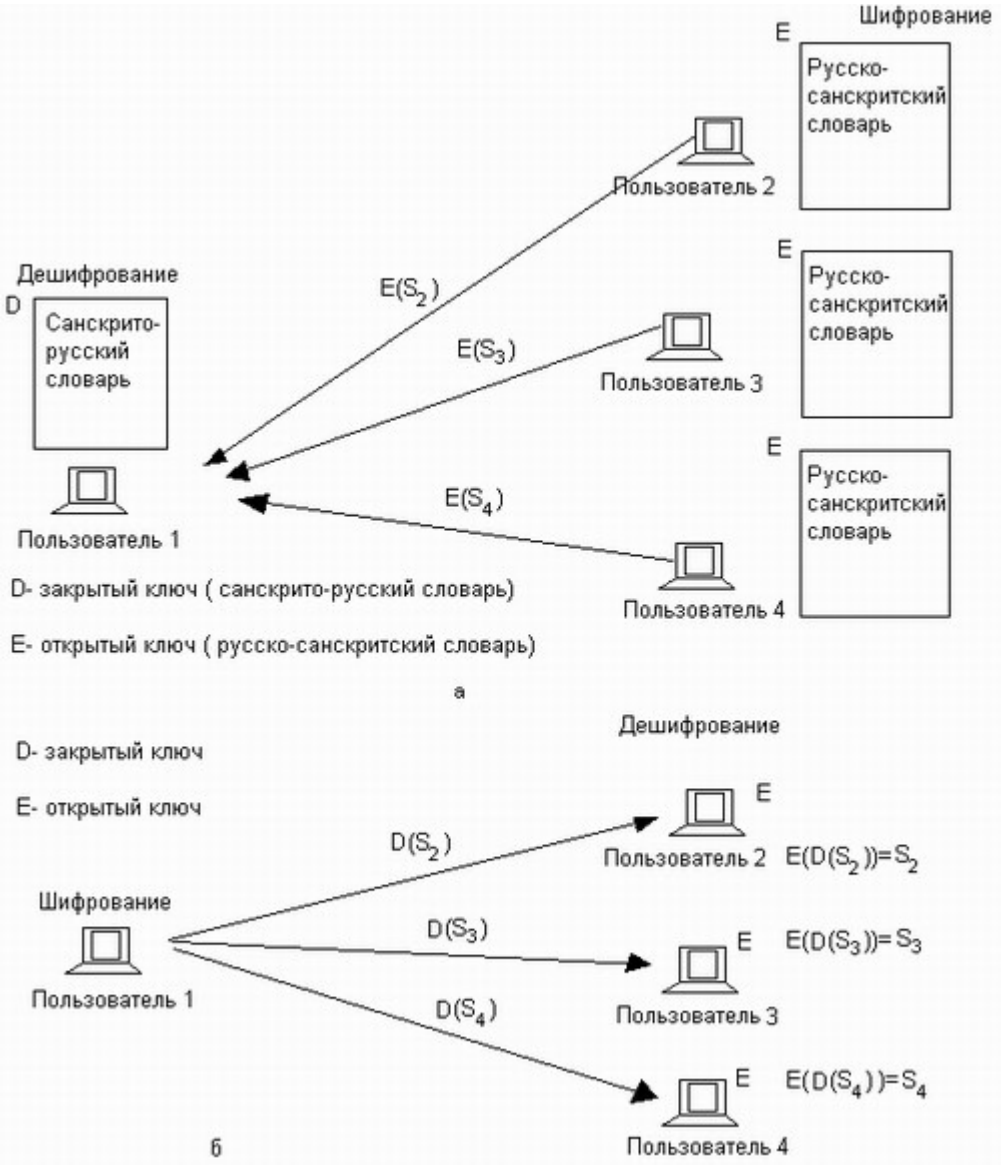
русский словарь по русско-санскритскому словарю. Такая процедура, требующая больших временных затрат, является отдаленной аналогией восстановления закрытого ключа по открытому.

Другая схема использования открытого и закрытого ключей, целью которой является подтверждение авторства (аутентификация или электронная подпись) посылаемого сообщения. В этом случае поток сообщений имеет обратное направление — от абонента 1, обладателя закрытого ключа D, к его корреспондентам, обладателям открытого ключа E. Если абонент 1 хочет аутентифицировать себя (поставить электронную подпись), то он шифрует известный текст своим закрытым ключом D и передает шифровку своим корреспондентам. Если им удается расшифровать текст открытым ключом абонента 1, то это доказывает, что текст был зашифрован его же закрытым ключом, а значит, именно он является автором этого сообщения. Заметим, что в этом случае сообщения $S_2$, $S_3$, 84, адресованные разным абонентам, не являются секретными, так как все они — обладатели одного и того же открытого ключа, с помощью которого они могут расшифровывать все сообщения, поступающие от абонента 1.

Если же нужна взаимная аутентификация и двунаправленный секретный обмен сообщениями, то каждая из общающихся сторон генерирует собственную пару ключей и посылает открытый ключ своему корреспонденту.

Для того чтобы в сети все n абонентов имели возможность не только принимать зашифрованные сообщения, но и сами посылать таковые, каждый абонент должен обладать своей собственной парой ключей E и D. Всего в сети будет 2n ключей: n открытых ключей для шифрования и n секретных ключей для дешифрирования. Таким образом, решается проблема масштабируемости — квадратичная зависимость количества ключей от числа абонентов в симметричных алгоритмах заменяется линейной зависимостью в несимметричных алгоритмах. Исчезает и задача секретной доставки ключа. Злоумышленнику нет смысла стремиться завладеть открытым ключом,

поскольку это не дает возможности расшифровывать текст или вычислить закрытый ключ.




Две схемы использования открытого и закрытого ключей

Хотя информация об открытом ключе не является секретной, ее нужно защищать от подлогов, чтобы злоумышленник под именем легального пользователя не навязал свой открытый ключ, после чего с помощью своего закрытого ключа он может расшифровывать все сообщения, посылаемые легальному пользователю и отправлять свои сообщения от его имени. Проще всего было бы распространять списки, связывающие имена пользователей с их открытыми ключами широковещательно, путем публикаций в средствах массовой информации (бюллетени, специализированные журналы и т. п.). Однако при таком подходе мы снова, как и в случае с паролями,

сталкиваемся с плохой масштабируемостью. Решением этой проблемы является технология цифровых сертификатов. Сертификат — это электронный документ, который связывает конкретного пользователя с конкретным ключом.

В настоящее время одним из наиболее популярных криптоалгоритмов с открытым ключом является криптоалгоритм RSA.

**Аутентификация**

**Аутентификация** (authentication) предотвращает доступ к сети нежелательных лиц и разрешает вход для легальных пользователей. Термин «аутентификация» в переводе с латинского означает «установление подлинности». Аутентификацию следует отличать от идентификации. Идентификаторы пользователей используются в системе с теми же целями, что и идентификаторы любых других объектов, файлов, процессов, структур данных, но они не связаны непосредственно с обеспечением безопасности. Идентификация заключается в сообщении пользователем системе своего идентификатора, в то время как аутентификация — это процедура доказательства пользователем того, что он есть тот, за кого себя выдает, в частности, доказательство того, что именно ему принадлежит введенный им идентификатор.

В процедуре аутентификации участвуют две стороны: одна сторона доказывает свою аутентичность, предъявляя некоторые доказательства, а другая сторона — аутентификатор — проверяет эти доказательства и принимает решение. В качестве доказательства аутентичности используются самые разнообразные приемы:

- аутентифицируемый может продемонстрировать знание некоего общего для обеих сторон секрета: слова (пароля) или факта (даты и места события, прозвища человека и т. п.);
- аутентифицируемый может продемонстрировать, что он владеет неким уникальным предметом (физическим

ключом), в качестве которого может выступать, например, электронная магнитная карта;

- аутентифицируемый может доказать свою идентичность, используя собственные биохарактеристики: рисунок радужной оболочки глаза или отпечатки пальцев, которые предварительно были занесены в базу данных аутентификатора.

Сетевые службы аутентификации строятся на основе всех этих приемов, но чаще всего для доказательства идентичности пользователя используются пароли. Простота и логическая ясность механизмов аутентификации на основе паролей в какой-то степени компенсирует известные слабости паролей. Это, во-первых, возможность раскрытия и разгадывания паролей, а во-вторых, возможность «подслушивания» пароля путем анализа сетевого трафика. Для снижения уровня угрозы от раскрытия паролей администраторы сети, как правило, применяют встроенные программные средства для формирования политики назначения и использования паролей: задание максимального и минимального сроков действия пароля, хранение списка уже использованных паролей, управление поведением системы после нескольких неудачных попыток логического входа и т. п. Перехват паролей по сети можно предупредить путем их шифрования перед передачей в сеть.

Легальность пользователя может устанавливаться по отношению к различным системам. Так, работая в сети, пользователь может проходить процедуру аутентификации и как локальный пользователь, который претендует на использование ресурсов только данного компьютера, и как пользователь сети, который хочет получить доступ ко всем сетевым ресурсам. При локальной аутентификации пользователь вводит свои идентификатор и пароль, которые автономно обрабатываются операционной системой, установленной на данном компьютере. При логическом входе в сеть данные о пользователе (идентификатор и пароль) передаются на сервер,

который хранит учетные записи обо всех пользователях сети. Многие приложения имеют свои средства определения, является ли пользователь законным. И тогда пользователю приходится проходить дополнительные этапы проверки.

В качестве объектов, требующих аутентификации, могут выступать не только пользователи, но и различные устройства, приложения, текстовая и другая информация. Так, например, пользователь, обращающийся с запросом к корпоративному серверу, должен доказать ему свою легальность, но он также должен убедиться сам, что ведет диалог действительно с сервером своего предприятия. Другими словами, сервер и клиент должны пройти процедуру взаимной аутентификации'. Здесь мы имеем дело с аутентификацией на уровне приложений. При установлении сеанса связи между двумя устройствами также часто предусматриваются процедуры взаимной аутентификации на более низком, канальном уровне. Примером такой процедуры является аутентификация по протоколам PAP и CHAP, входящим в семейство протоколов PPP. Аутентификация данных означает доказательство целостности этих данных, а также того, что они поступили именно от того человека, который объявил об этом. Для этого используется механизм электронной подписи.

В вычислительных сетях процедуры аутентификации часто реализуются теми же программными средствами, что и процедуры авторизации. В отличие от аутентификации, которая распознает легальных и нелегальных пользователей, система авторизации имеет дело только с легальными пользователями, которые уже успешно прошли процедуру аутентификации. Цель подсистем авторизации состоит в том, чтобы предоставить каждому легальному пользователю именно те виды доступа и к тем ресурсам, которые были для него определены администратором системы.

**Авторизация доступа**

**Средства авторизации** (authorization) контролируют доступ легальных пользователей к ресурсам системы, предоставляя каждому из них

именно те права, которые ему были определены администратором. Кроме предоставления прав доступа пользователям к каталогам, файлам и принтерам система авторизации может контролировать возможность выполнения пользователями различных системных функций, таких как локальный доступ к серверу, установка системного времени, создание резервных копий данных, выключение сервера и т. п.

Система авторизации наделяет пользователя сети правами выполнять определенные действия над определенными ресурсами. Для этого могут быть использованы различные формы предоставления правил доступа, которые часто делят на два класса:

- избирательный доступ;
- мандатный доступ.

Избирательные права доступа реализуются в операционных системах универсального назначения. В наиболее распространенном варианте такого подхода определенные операции над определенным ресурсом разрешаются или запрещаются пользователям или группам пользователей, явно указанным своими идентификаторами. Например, пользователю, имеющему идентификатор User_T, может быть разрешено выполнять операции чтения и записи по отношению к файлу Filet. Модификацией этого способа является использование для идентификации пользователей их должностей, или факта их принадлежности к персоналу того или иного производственного подразделения, или еще каких-либо других позиционирующих характеристик. Примером такого правила может служить следующее: файл бухгалтерской отчетности BUCH могут читать работники бухгалтерии и руководитель предприятия.

Мандатный подход к определению прав доступа заключается в том, что вся информация делится на уровни в зависимости от степени секретности, а все пользователи сети также делятся на группы, образующие иерархию в соответствии с уровнем допуска к этой информации. Такой подход используется в известном делении информации на информацию для

служебного пользования, «секретно», «совершенно секретно». При этом пользователи этой информации в зависимости от определенного для них статуса получают различные формы допуска: первую, вторую или третью. В отличие от систем с избирательными правами доступа в системах с мандатным подходом пользователи в принципе не имеют возможности изменить уровень доступности информации. Например, пользователь более высокого уровня не может разрешить читать данные из своего файла пользователю, относящемуся к более низкому уровню. Отсюда видно, что мандатный подход является более строгим, он в корне пресекает всякий волюнтаризм со стороны пользователя. Именно поэтому он часто используется в системах военного назначения.

Процедуры авторизации реализуются программными средствами, которые могут быть встроены в операционную систему или в приложение, а также могут поставляться в виде отдельных программных продуктов. При этом программные системы авторизации могут строиться на базе двух схем:

- централизованная схема авторизации, базирующаяся на сервере;
- децентрализованная схема, базирующаяся на рабочих станциях.

В первой схеме сервер управляет процессом предоставления ресурсов пользователю. Главная цель таких систем — реализовать «принцип единого входа». В соответствии с централизованной схемой пользователь один раз логически входит в сеть и получает на все время работы некоторый набор разрешений по доступу к ресурсам сети. Система Kerberos с ее сервером безопасности и архитектурой клиент-сервер является наиболее известной системой этого типа. Системы TACACS и RADIUS, часто применяемые совместно с системами удаленного доступа, также реализуют этот подход.

При втором подходе рабочая станция сама является защищенной — средства защиты работают на каждой машине, и сервер не требуется. Рассмотрим работу системы, в которой не предусмотрена процедура

однократного логического входа. Теоретически доступ к каждому приложению должен контролироваться

средствами безопасности самого приложения или же средствами, существующими в той операционной среде, в которой оно работает. В корпоративной сети администратору придется отслеживать работу механизмов безопасности, используемых всеми типами приложений — электронной почтой, службой каталогов локальной сети, базами данных хостов и т. п. Когда администратору приходится добавлять или удалять пользователей, то часто требуется вручную конфигурировать доступ к каждой программе или системе.

В крупных сетях часто применяется комбинированный подход предоставления пользователю прав доступа к ресурсам сети: сервер удаленного доступа ограничивает доступ пользователя к подсетям или серверам корпоративной сети, то есть к укрупненным элементам сети, а каждый отдельный сервер сети сам по себе ограничивает доступ пользователя к своим внутренним ресурсам: разделяемым каталогам, принтерам или приложениям. Сервер удаленного доступа предоставляет доступ на основании имеющегося у него списка прав доступа пользователя (Access Control List, ACL), а каждый отдельный сервер сети предоставляет доступ к своим ресурсам на основании хранящегося у него списка прав доступа, например ACL файловой системы.

Подчеркнем, что системы аутентификации и авторизации совместно выполняют одну задачу, поэтому необходимо предъявлять одинаковый уровень требований к системам авторизации и аутентификации. Ненадежность одного звена здесь не может быть компенсирована высоким качеством другого звена. Если при аутентификации используются пароли, то требуются чрезвычайные меры по их защите. Однажды украденный пароль открывает двери ко всем приложениям и данным, к которым пользователь с этим паролем имел легальный доступ.

**Аудит**

**Аудит** (auditing) — фиксация в системном журнале событий, связанных с доступом к защищаемым системным ресурсам. Подсистема аудита современных ОС позволяет дифференцирование задавать перечень интересующих администратора событий с помощью удобного графического интерфейса. Средства учета и наблюдения обеспечивают возможность обнаружить и зафиксировать важные события, связанные с безопасностью, или любые попытки создать, получить доступ или удалить системные ресурсы. Аудит используется для того, чтобы засекать даже неудачные попытки «взлома» системы.

Учет и наблюдение означает способность системы безопасности «шпионить» за выбранными объектами и их пользователями и выдавать сообщения тревоги, когда кто-нибудь пытается читать или модифицировать системный файл. Если кто-то пытается выполнить действия, определенные системой безопасности для отслеживания, то система аудита пишет сообщение в журнал регистрации, идентифицируя пользователя. Системный менеджер может создавать отчеты о безопасности, которые содержат информацию из журнала регистрации. Для «сверхбезопасных» систем предусматриваются аудио- и видеосигналы тревоги, устанавливаемые на машинах администраторов, отвечающих за безопасность.

Поскольку никакая система безопасности не гарантирует защиту на уровне 100 %, то последним рубежом в борьбе с нарушениями оказывается система аудита.

Действительно, после того как злоумышленнику удалось провести успешную атаку, пострадавшей стороне не остается ничего другого, как обратиться к службе аудита. Если при настройке службы аудита были правильно заданы события, которые требуется отслеживать, то подробный анализ записей в журнале может дать много полезной информации. Эта информация, возможно, позволит найти злоумышленника или по крайней мере предотвратить повторение подобных атак путем устранения уязвимых мест в системе защиты.

**Схема использования сертификатов**

Аутентификация личности на основе сертификатов происходит примерно так же, как на проходной большого предприятия. Вахтер пропускает людей на территорию на основании пропуска, который содержит фотографию и подпись сотрудника, удостоверенных печатью предприятия и подписью лица, выдавшего пропуск. Сертификат является аналогом пропуска и выдается по запросам специальными сертифицирующими центрами при выполнении определенных условий.

Сертификат представляет собой электронную форму, в которой содержится следующая информация:

- открытый ключ владельца данного сертификата;
- сведения о владельце сертификата, такие, например, как имя, адрес электронной почты, наименование организации, в которой он работает, и т. п.;
- наименование сертифицирующей организации, выдавшей данный сертификат.

Кроме того, сертификат содержит электронную подпись сертифицирующей организации — зашифрованные закрытым ключом этой организации данные, содержащиеся в сертификате.

Использование сертификатов основано на предположении, что сертифицирующих организаций немного и их открытые ключи могут быть всем известны каким-либо способом, например, из публикаций в журналах.

Когда пользователь хочет подтвердить свою личность, он предъявляет свой сертификат в двух формах — открытой (то есть такой, в которой он получил его в сертифицирующей организации) и зашифрованной с применением своего закрытого ключа. Сторона, проводящая аутентификацию, берет из открытого сертификата открытый ключ пользователя и расшифровывает с помощью него зашифрованный сертификат. Совпадение результата с открытым сертификатом подтверждает

факт, что предъявитель действительно является владельцем закрытого ключа, парного с указанным открытым

Затем с помощью известного открытого ключа указанной в сертификате организации проводится расшифровка подписи этой организации в сертификате. Если в результате получается тот же сертификат с тем же именем пользователя и его открытым ключом — значит, он действительно прошел регистрацию в сертификационном центре, является тем, за кого себя выдает, и указанный в сертификате открытый ключ действительно принадлежит ему.

Сертификаты можно использовать не только для аутентификации, но и для предоставления избирательных прав доступа. Для этого в сертификат могут вводиться дополнительные поля, в которых указывается принадлежность его владельцев той или иной категории пользователей. Эта категория назначается сертифицирующей организацией в зависимости от условий, на которых выдается сертификат. Например, организация, поставляющая через Интернет на коммерческой основе информацию, может выдавать сертификаты определенной категории пользователям, оплатившим годовую подписку на некоторый бюллетень, а Web-сервер будет предоставлять доступ к страницам бюллетеня только пользователям, предъявившим сертификат данной категории.

Подчеркнем тесную связь открытых ключей с сертификатами. Сертификат является не только удостоверением личности, но и удостоверением принадлежности открытого ключа. Цифровой сертификат устанавливает и гарантирует соответствие между открытым ключом и его владельцем. Это предотвращает угрозу подмены открытого ключа. Если некоторому абоненту поступает открытый ключ в составе сертификата, то он может быть уверен, что этот открытый ключ гарантированно принадлежит отправителю, адрес и другие сведения о котором содержатся в этом сертификате.

При использовании сертификатов отпадает необходимость хранить на серверах корпораций списки пользователей с их паролями, вместо этого

достаточно иметь на сервере список имен и открытых ключей сертифицирующих организаций. Может также понадобиться некоторый механизм отображений категорий владельцев сертификатов на традиционные группы пользователей для того, чтобы можно было использовать в неизменном виде механизмы управления избирательным доступом большинства операционных систем или приложений.

**Прокси-сервер**

**Прокси-сервер** (от англ. proxy — «представитель, уполномоченный») — служба в компьютерных сетях, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам. Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс (например, e-mail), расположенный на другом сервере. Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кеша (в случаях, если прокси имеет свой кеш). В некоторых случаях запрос клиента или ответ сервера может быть изменён прокси-сервером в определённых целях. Также прокси-сервер позволяет защищать клиентский компьютер от некоторых сетевых атак.

Чаще всего прокси-серверы применяются для следующих целей:

- Обеспечение доступа с компьютеров локальной сети в Интернет.
- Кеширование данных: если часто происходят обращения к одним и тем же внешним ресурсам, то можно держать их копию на прокси-сервере и выдавать по запросу, снижая тем самым нагрузку на канал во внешнюю сеть и ускоряя получение клиентом запрошенной информации.
- Сжатие данных: прокси-сервер загружает информацию из Интернета и передаёт информацию конечному пользователю в сжатом виде. Такие прокси-серверы используются в основном с целью экономии внешнего трафика.
- Защита локальной сети от внешнего доступа: например, можно настроить прокси-сервер так, что локальные компьютеры будут

обращаться к внешним ресурсам только через него, а внешние компьютеры не смогут обращаться к локальным вообще (они «видят» только прокси-сервер). См. также NAT.

- Ограничение доступа из локальной сети к внешней: например, можно запретить доступ к определённым веб-сайтам, ограничить использование интернета каким-то локальным пользователям, устанавливать квоты на трафик или полосу пропускания, фильтровать рекламу и вирусы.
- Анонимизация доступа к различным ресурсам. Прокси-сервер может скрывать сведения об источнике запроса или пользователе. В таком случае целевой сервер видит лишь информацию о прокси-сервере, например, IP-адрес, но не имеет возможности определить истинный источник запроса. Существуют также искажающие прокси-серверы, которые передают целевому серверу ложную информацию об истинном пользователе.

Многие прокси-серверы используются для нескольких целей одновременно. Некоторые прокси-серверы ограничивают работу несколькими портами: 80 (Браузер), 443 (Шифрованное соединение (HTTPS)), 20,21 (FTP).

В отличие от шлюза прокси-сервер чаще всего не пропускает ICMP-трафик (невозможно проверить доступность машины командами ping и tracert).

Прокси-сервер, к которому может получить доступ любой пользователь сети интернет, называется открытым.

**«Тонкий» клиент**

В компьютерных технологиях **тонкий клиент** (англ. thin client) — это компьютер-клиент сети с клиент-серверной архитектурой (точнее с терминальной архитектурой), который переносит все задачи по обработке информации на сервер. Таким образом, для работы тонкого клиента необходим терминальный сервер. Этим тонкий клиент отличается от

**толстого клиента**, который, напротив, производит обработку информации независимо от сервера, используя последний в основном лишь для хранения данных. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями. Кроме общего случая, следует выделить аппаратный тонкий клиент (например, Windows-терминал) - специализированное устройство, принципиально отличное от ПК. Аппаратный тонкий клиент не имеет жёсткого диска, использует специализированную локальную ОС (одна из задач которой организовать сессию с терминальным сервером для работы пользователя), не имеет в своём составе подвижных деталей, выполняется в специализированных корпусах с полностью пассивным охлаждением. Для расширения функциональности тонкого клиента прибегают к его "утолщению", например, добавляют возможности автономной работы, сохраняя главное отличие - работу в сессии с терминальным сервером. Когда в клиенте появляются подвижные детали (драйвы), появляются возможности автономной работы, он перестаёт быть тонким клиентом в чистом виде, а становится универсальным клиентом.

**Тонкий клиент** в большинстве случаев обладает минимальной аппаратной конфигурацией, вместо жесткого диска для загрузки локальной специализированной ОС используется DOM (DiskOnModule) [модуль с разъёмом IDE, флэш-памятью и микросхемой, реализующей логику обычного жёсткого диска - в BIOS определяется как обычный жёсткий диск, только размер его обычно на 2-3 порядка меньше]. В некоторых конфигурациях системы тонкий клиент загружает операционную систему по сети с сервера, используя протоколы PXE, BOOTP, DHCP, TFTP и Remote Installation Services (RIS).

Протоколы, используемые тонкими клиентами

- X11 — используется в Unix
- Telnet — мультиплатформенный

- SSH — мультиплатформенный защищённый аналог Telnet
- NX NoMachine — протокол X11 со сжатием данных
- Virtual Network Computing
- Citrix ICA
- Remote Desktop Protocol (RDP), протокол для удалённой работы с использованием графического интерфейса пользователя для Microsoft Windows.

## III. **КАРТА ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ КАДРАМИ ПРОФЕССОРСКО-ПРЕПОДАВАТЕЛЬСКОГО СОСТАВА**.

Дисциплину ведет Павлов Михаил Сергеевич ассистент кафедры математического анализа и моделирования.

**СОДЕРЖАНИЕ**