МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

Методические указания для проведения лабораторных
работ по дисциплине

Безопасность информационных процессов в компьютерных системах и сетях

(направление подготовки 09.03.01 - Информатика и вычислительная техника)

Ростов-на-Дону
2022

Методические указания для проведения лабораторных работ по дисциплине

Безопасность информационных процессов в компьютерных системах и сетях

Составил: И.А. Сосновский, доцент кафедры ИТСС

Рассмотрено и одобрено
на заседании кафедры
Протокол от «19» декабря 2022 г. № 5

***Лабораторная работа № 1*. Модели взаимодействия прикладной программы и программной закладки. Методы перехвата и навязывания информации. Методы внедрения программных закладок**

***Цель работы*:** ознакомиться с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.

Модели взаимодействия прикладной программы и программной закладки

Общая модель РПС можно представить в виде совокупности моделей, каждая из которых соответствует описанным выше типам РПС и характеризует действия злоумышленника, исходя из его образа мыслей и возможностей [ПАС].

1. *Модель «перехват».* Программная закладка встраивается (внедряется) в ПЗУ, ОС или прикладное программное обеспечение и сохраняет все или избранные фрагменты вводимой или выводимой информации в скрытой области локальной или удаленной внешней памяти прямого доступа. Объектом сохранения может быть клавиатурный ввод, документы, выводимые на принтер, или уничтожаемые файлы-документы. Для данной модели существенно наличие во внешней памяти места хранения информации, которое должно быть организовано таким образом, чтобы обеспечить ее сохранность на протяжении заданного промежутка времени и возможность последующего съема. Важно также, чтобы сохраняемая информация была каким-либо образом замаскирована от просмотра легальными пользователями.

2. *Модель «троянский конь».* Закладка встраивается в постоянно используемое программное обеспечение и по некоторому активизирующему событию моделирует сбойную ситуацию на средствах хранения информации или в оборудовании компьютера (сети). Тем самым могут быть достигнуты две различные цели: во-первых, парализована нормальная работа компьютерной системы и, во-вторых, злоумышленник (например, под видом обслуживания или ремонта) может ознакомиться с имеющейся в системе или накопленной

посредством использования модели «перехват» информацией. Событием, активизирующим закладку, может быть некоторый момент времени, либо сигнал из канала модемной связи (явный или замаскированный), либо состояние некоторых счетчиков (например, число запусков программ).

3. *Модель «наблюдатель».* Закладка встраивается в сетевое или телекоммуникационное программное обеспечение. Пользуясь тем, что данное ПО, как правило, всегда активно, программная закладка осуществляет контроль за процессами обработки информации на данном компьютере, установку и удаление закладок, а также съем накопленной информации. Закладка может инициировать события для ранее внедренных закладок, действующих по модели «троянский конь».

4. *Модель «компрометация».* Закладка либо передает заданную злоумышленником информацию (например, клавиатурный ввод) в канал связи, либо сохраняет ее, не полагаясь на гарантированную возможность последующего приема или снятия. Более экзотический случай – закладка инициирует постоянное обращение к информации, приводящее к росту отношения сигнал/шум при перехвате побочных излучений.

5. *Модель «искажение или инициатор ошибок».* Программная закладка искажает потоки данных, возникающие при работе прикладных программ (выходные потоки), либо искажает входные потоки информации, либо инициирует (или подавляет) возникающие при работе прикладных программ ошибки.

6. *Модель «сборка мусора».* В данном случае прямого воздействия РПС может и не быть; изучаются «остатки» информации. В случае применения программной закладки навязывается такой порядок работы, чтобы максимизировать количество остающихся фрагментов ценной информации. Злоумышленник получает либо данные фрагменты, используя закладки моделей 2 и 3, либо непосредственный доступ к компьютеру под видом ремонта или профилактики.

У рассмотренных различных по целям воздействия моделей закладок имеется важная общая черта - наличие операции записи, производимой закладкой (в оперативную или внешнюю память). При отсутствии данной операции никакое негативное влияние невозможно. Вполне понятно, что для направленного воздействия закладка должна также выполнять и операции чтения. Так, например, можно реализовать функцию целенаправленной модификации данных в каком-либо секторе жесткого диска, которая возможна только после их прочтения.

Таким образом, исполнение кода закладки может быть сопровождено операциями несанкционированной записи (НСЗ), например, для сохранения некоторых фрагментов информации, и несанкционированного считывания (НСЧ), которое может происходить отдельно от операций чтения прикладной программы или совместно с ними. При этом операции считывания и записи могут быть не связаны с получением информации, например считывание параметров устройства или его инициализация - закладка может использовать для своей работы и такие операции, в частности, для инициирования сбойных ситуаций или переназначения ввода вывода.

Возможные комбинации несанкционированных действий (НСЧ и НСЗ), а также санкционированных действий прикладной программы или операционной среды по записи (СЗ) или считыванию (СЧ) приведены в табл. 1.2 [ПАС].

Ситуации 1 - 4 соответствуют нормальной работе прикладной программы, когда закладка не оказывает на нее никакого воздействия.

Ситуация 5 может быть связана либо с разрушением кода прикладной программы в оперативной памяти ЭВМ, поскольку прикладная программа не выполняет санкционированные действия по записи и считыванию, либо с сохранением уже накопленной в ОП информации.

Ситуация 6 связана с разрушением или с сохранением информации (искажение или сохранение выходного потока), записываемой прикладной программой.

Ситуация 7 связана с сохранением информации (сохранение входного потока) считываемой прикладной программой.

Ситуация 8 связана с сохранением информации закладкой при считывании или записи информации прикладной программой.

Ситуация 9 не связана с прямым негативным воздействием, поскольку прикладная программа не активна, а закладка производит только НСЧ (процесс «настройки»).

Ситуация 10 может быть связана с сохранением выводимой информации в оперативную память.

Ситуация 11 может быть связана с сохранением вводимой информации в оперативную память либо с изменением параметров процесса санкционированного чтения закладкой.

Ситуация 12 может быть связана с сохранением как вводимой, так и выводимой прикладной программой информации в оперативную память.

Ситуация 13 может быть связана с размножением закладки, сохранением накопленной в буферах ОП информации или с разрушением кода и данных в файлах, поскольку прикладная программа не активна.

Таблица 1.2

Номер ситуации	***НСЧ***	***НСЗ***	***Действия РПС***	***СЧ***	***СЗ***
1	0	0	Нет	0	0
2	0	0	Нет	0	1
3	0	0	Нет	1	0
4	0	0	Нет	1	1
5	0	1	Изменение (разрушение) кода прикладной программы в ОП	0	0
6	0	1	Разрушение или сохранение выводимых прикладной программой данных	0	1
7	0	1	Разрушение или сохранение вводимых прикладной программой данных	1	0

8	0	1	Разрушение или сохранение вводимых и выводимых данных	1	1
9	1	0	Нет	0	0
10	1	0	Перенос выводимых прикладной программой данных в ОП	0	1
11	1	0	Перенос вводимых прикладной программой данных в ОП	1	0
12	1	0	Перенос вводимых и выводимых данных в ОП	1	1
13	1	1	Процедуры типа «размножения вируса» (действия закладки независимо от операций прикладной программы)	0	0
14	1	1	Те же действия, что и в процедурах 6 – 8	0	1
15	1	1		1	0
16	1	1		1	1

Ситуации 14 - 16 могут быть связаны как с сохранением, так и с разрушением данных или кода и аналогичны ситуациям 6 - 8.

Несанкционированная запись закладкой может происходить:

- в массив данных, не совпадающий с пользовательской информацией, - сохранение информации;
- в массив данных, совпадающий с пользовательской информацией или ее подмножеством, - искажение, уничтожение или навязывание информации закладкой.

Следовательно, можно рассматривать три основные группы деструктивных функций, которые могут выполняться РПС [ПАС]:

- сохранение фрагментов информации, возникающей при работе пользователя, прикладных программ, вводе/выводе данных, во внешней памяти (локальной или удаленной) в сети или выделенном компьютере, в том числе сохранение различных паролей, ключей и кодов доступа, собственно конфиденциальных документов в электронном виде, либо безадресная

компрометация фрагментов ценной информации (модели «перехват», «компрометация»);

- изменение алгоритмов функционирования прикладных программ (т.е. целенаправленное воздействие во внешней или оперативной памяти) - происходит изменение собственно исходных алгоритмов работы программ, например, программа разграничения доступа станет пропускать пользователей по любому паролю (модели «искажение», «троянский конь»);
- навязывание некоторого режима работы (например, при уничтожении информации - блокирование записи на диск, при этом информация, естественно, не уничтожается) либо замена записываемой информации данными, навязанными закладкой.

Методы внедрения рпс

Можно выделить следующие основные методы внедрения РПС [ПАС].

Маскировка закладки под «безобидное» программное обеспечение. Данный метод заключается в том, что программная закладка внедряется в систему под видом новой программы, на первый взгляд абсолютно безобидной. Программная закладка может быть внедрена в текстовый или графический редакторы, системную утилиту, компьютерную игру, хранитель экрана и т.д. После внедрения закладки ее присутствие в системе не нужно маскировать - даже если администратор заметит факт появления в системе новой программы, он не придаст этому значения, поскольку эта программа внешне совершенно безобидна.

Маскировка закладки под «безобидный» модуль расширения программной среды. Многие программные среды допускают свое расширение дополнительными программными модулями. Например, для операционных систем семейства Microsoft Windows модулями расширения могут выступать динамически подгружаемые библиотеки (DLL) и драйверы устройств. В таких модулях расширения может содержаться РПС, которое может потенциально внедрено в систему. Данный метод фактически является частным случаем предыдущего метода и отличается от него только тем, что закладка

представляет собой не прикладную программу, а модуль расширения программной среды.

Подмена закладкой одного или нескольких программных модулей атакуемой среды. Данный метод внедрения в систему программной закладки заключается в том, что в атакуемой программной среде выбирается один или несколько программных модулей, подмена которых фрагментами программной закладки позволяет оказывать на среду требуемые негативные воздействия. Программная закладка должна полностью реализовывать все функции подменяемых программных модулей.

Основная проблема, возникающая при практической реализации данного метода, заключается в том, что программист, разрабатывающий программную закладку, никогда не может быть уверен, что созданная им закладка точно реализует все функции подменяемого программного модуля. Если подменяемый модуль достаточно велик по объему или недостаточно подробно документирован, точно запрограммировать все его функции практически невозможно. Поэтому описываемый метод целесообразно применять только для тех программных модулей атакуемой среды, для которых доступна полная или почти полная документация. Оптимальной является ситуация, когда доступен исходный текст подменяемого модуля.

Прямое ассоциирование. Данный метод внедрения в систему программной закладки заключается в ассоциировании закладки с исполняемыми файлами одной или нескольких легальных программ системы. Сложность задачи прямого ассоциирования программной закладки с программой атакуемой среды существенно зависит от того, является атакуемая среда однозадачной или многозадачной, однопользовательской или многопользовательской. Для однозадачных однопользовательских систем эта задача решается достаточно просто. В то же время при внедрении закладки в многозадачную или многопользовательскую программную среду прямое ассоциирование закладки с легальным программным обеспечением является весьма нетривиальной задачей.

Косвенное ассоциирование. Косвенное ассоциирование закладки с программным модулем атакуемой среды заключается в ассоциировании закладки с кодом программного модуля, загруженным в оперативную память. При косвенном ассоциировании исполняемый файл программного модуля остается неизменным, что затрудняет выявление программной закладки.

Для того чтобы косвенное ассоциирование стало возможным, необходимо, чтобы инсталлирующая часть закладки уже присутствовала в системе. Другими словами, программная закладка, внедряемая в систему с помощью косвенного ассоциирования, должна быть составной.

***Лабораторная работа № 2*. Защита от закладок и дизассемблирования. Способы встраивания защитных механизмов в программное обеспечение. Понятие разрушающего программного воздействия.**

***Цель работы*:** освоить технологию работы с дизассемблером и декомпилятором.

Основные принципы обеспечения безопасности ПО.

В качестве объекта обеспечения технологической и эксплуатационной безопасности ПО рассматривается вся совокупность его компонентов в рамках конкретной КС. В качестве доминирующей должна использоваться стратегия сквозного тотального контроля технологического и эксплуатационного этапов жизненного цикла компонентов ПО. Совокупность мероприятий по обеспечению технологической и эксплуатационной безопасности компонентов ПО должна носить, по возможности, конфиденциальный характер. Необходимо обеспечить постоянный, комплексный и действенный контроль за деятельностью разработчиков и пользователей компонентов ПО. Кроме общих принципов, обычно необходимо конкретизировать принципы обеспечения безопасности ПО на каждом этапе его жизненного цикла. Далее приводятся один из вариантов разработки таких принципов.

Принципы обеспечения технологической безопасности при обосновании, планировании работ и проектном анализе ПО.

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Комплексности обеспечения безопасности ПО, предполагающей рассмотрение проблемы безопасности информационно - вычислительных процессов с учетом всех структур КС, возможных каналов утечки информации и несанкционированного доступа к ней, времени и условий их возникновения, комплексного применения организационных и технических мероприятий.

Планируемости применения средств безопасности программ, предполагающей перенос акцента на совместное системное проектирование ПО и средств его безопасности, планирование их использования в предполагаемых условиях эксплуатации.

Обоснованности средств обеспечения безопасности ПО, заключающейся в глубоком научно-обоснованном подходе к принятию проектных решений по оценке степени безопасности, прогнозированию угроз безопасности, всесторонней априорной оценке показателей средств защиты.

Достаточности защищенности программ, отражающей необходимость поиска наиболее эффективных и надежных мер безопасности при одновременной минимизации их стоимости.

Гибкости управления защитой программ, требующей от системы контроля и управления обеспечением безопасности ПО способности к диагностированию, опережающей нейтрализации, оперативному и эффективному устранению возникающих угроз,

Заблаговременности разработки средств обеспечения безопасности и контроля производства ПО, заключающейся в предупредительном характере мер обеспечения технологической безопасности работ в интересах недопущения снижения эффективности системы безопасности процесса создания ПО.

Документируемости технологии создания программ, подразумевающей разработку пакета нормативно-технических документов по контролю программных средств на наличие преднамеренных дефектов.

Принципы достижения технологической безопасности по в процессе его разработки.

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Регламентации технологических этапов разработки ПО, включающей упорядоченные фазы промежуточного контроля, спецификацию программных модулей и стандартизацию функций и формата представления данных.

*Автоматизации средств контроля управляющих и вычислительных программ*на наличие преднамеренных дефектов.

Создания типовой общей информационной базы алгоритмов, исходных текстов и программных средств, позволяющих выявлять преднамеренные программные дефекты.

*Последовательной многоуровневой фильтрации программных модулей*в процессе их создания с применением функционального дублирования разработок и поэтапного контроля.

Типизации алгоритмов, программ и средств информационной безопасности, обеспечивающей информационную, технологическую и программную совместимость, на основе максимальной их унификации по всем компонентам и интерфейсам.

*Централизованного управления базами данных проектов ПО и администрирование технологии их разработки*с жестким разграничением функций, ограничением доступа в соответствии со средствами диагностики, контроля и защиты.

*Блокирования несанкционированного доступа*соисполнителей и абонентов государственных и негосударственных сетей связи, подключенных к стендам для разработки программ.

*Статистического учета и ведения системных журналов*о всех процессах разработки ПО с целью контроля технологической безопасности.

Использования только сертифицированных и выбранных в качестве единых инструментальных средств разработки программ для новых технологий обработки информации и перспективных архитектур вычислительных систем.

Дизассе́мблер — **транслятор, преобразующий машинный код, объектный файл или библиотечные модули в текст программы на языке ассемблера.**

- По режиму работы с пользователем делятся на Автоматические
- Интерактивные

Примером автоматических дизассемблеров может служить Sourcer. Такие дизассемблеры генерируют готовый листинг, который можно затем править в текстовом редакторе. Пример интерактивного — IDA. Он позволяет изменять правила дизассемблирования и является весьма удобным инструментом для исследования программ.

Дизассемблеры бывают **однопроходные и многопроходные**. Основная трудность при работе дизассемблера — отличить данные от машинного кода, поэтому на первых проходах автоматически или интерактивно собирается информация о границах процедур и функций, а на последнем проходе формируется итоговый листинг. Интерактивность позволяет улучшить этот процесс, так как просматривая дамп дизассемблируемой области памяти, программист может сразу выделить строковые константы, дать содержательные имена известным точкам входа, прокомментировать разобранные им фрагменты программы.

Чаще всего дизассемблер используют для анализа программы (или ее части), исходный текст которой неизвестен — с целью модификации, копирования или взлома. Реже — для поиска ошибок (багов) в программах и компиляторах, а также для анализа оптимизации создаваемого компилятором машинного кода. Обычно однопроходный дизассемблер (как и построчный ассемблер) является составной частью отладчика.

Защита от дизассемблирования

Первое направление защиты, как правило, реализуется значительно легче, чем второе, поэтому будет приведен лишь краткий обзор данного направления. При реализации защиты программ от дизассемблирования можно применять различные приемы.

Среди них наиболее часто используемым и эффективным приемом является зашифровка и \ или запаковка отдельных участков исходного кода или всего кода целиком, при этом необходимо позаботиться о распаковке \ расшифровке программы на точке входа. Таким образом, при просмотре

исполняемого машинного кода исполняемого файла вместо рабочего кода программы будет отображен лишь бессмысленный набор операций. При реализации защиты от дизассемблирования используется также множество приемов, которые реализуются с целью запутать потенциального взломщика. Можно навести несколько примеров такого вида приемов:

- увеличение исходного кода программы добавлением множества «бессмысленных» операций, а рабочий участок программы записать в определенное место этого множества;

- замена местами адресов обработчиков (векторов) прерываний, например, поменять местами вектор прерывания видео сервиса (INT 10h) с вектором прерывания сервиса DOS (INT 21h), после такой замены для вызова из программы какой-либо функции прерывания INT 21h необходимо пользоваться вызовом прерывания INT 10h.

Для достижения наиболее надежной и эффективной защиты используется комбинация нескольких приемов.

Защита от отладки

Для защиты программы от трассировки отладчиком также существует несколько способов. Наиболее распространенными являются два из них.

Первый способ.

Идея:

При трассировке программы команды выполняются по команде человека, поэтому длительность выполнения операций(время от начала одной операции до начала следующей) изменяется. Поэтому в программу можно включать точки для проверки времени выполнения одинаковых участков кода программы. Если время выполнения выполнения одинаковых участков различна, то это означает, что программа трассируется в данный момент, необходимо выйти из программы, иначе - продолжить выполнение.

Алгоритм реализации:

1. Запомнить текущее время;

2. Выполнить контрольный участок кода;

3. Запомнить текущее время и разность текущего и предыдущего запомненного времени;

4. Выполнить контрольный участок кода повторно;

5. Сравнить разность текущего времени и предыдущего запомненного текущего времени с предыдущей запомненной разностью;

6. Если разности совпадают, продолжить выполнение, иначе – выйти из программы.

- метаморфическое преобразование кода программы, позволяющее защитить программу от дизассемблирования и модификации;

- защита ключом отдельных участков кода программы (поддерживается только в зарегистрированной версии);

- полное разрушение логики защищенных фрагментов кода, не позволяющее анализировать программу с помощью дизассемблера или отладчика;

- обнаружение и противодействие отладчикам SoftIce, NtIce, TD и др.;

- защита точки входа;

- защита от модификации кода;

- защищенная работа с реестром, не позволяющая программам вроде RegMon определить, к какому ключу реестра обращается твоя программа;

- технология "динамического импорта", которая разрушает имена всех импортируемых функций, а также не использует функцию GetProcAddress;

- сжатие ресурсов и исполнимого кода приложения;

- поддержка коротких серийных номеров (12 символов);

- поддержка внешнего генератора серийных номеров с OLE/DLL-интерфейсом;

- технология OneTouch Trial (о ней читай ниже).

Самое главное, что нас интересует – это метаморфическое преобразование кода программы и поддержка серийных номеров. Метаморфическое кодирование позволяет изменить код программы до

неузнаваемости и запутать отладчик и человека, который запустил этот отладчик.

Декомпилятор.

Он переводит двоичный код в символьный на языке команд какого-нибудь языка. Например, диассемблеры, деклиппер и многие другие. Эти средства появились раньше отладчиков, т.к. вначале не было архитектуры со встроенными средствами отлаживания программ. С помощью декомпиляторов можно изменять исходный код программы. Допустим необходимо внести крупные изменения в код программы. Прямая вставка двоичных кодов не помогает, т.к. нарушается расположение меток перехода и процедур. Программа – это линейка кода, по которой нужно перемещаться нелинейно, переходить с определенным смещением. Если линейка удлиняется из-за добавления чего-то в середине, все смещения будут показывать не туда куда нужно. Повторная перекомпиляция вписывает новые смещения.

Среди декомпиляторов можно выделить: Hacker-VIEW (HVIEW), IDA (интерактивный дизассемблер).

С помощью Hacker-VIEW можно посмотреть любой исполняемый файл по любому смещению. Можно выполнить какую-то часть программы. Это позволяет расшифровывать программы и обходить защиту от дизассемблирования. Этот декомпилятор «понимает» как старые форматы исполняемых файлов DOS-COM и DOS-EXE, так и форматы исполняемых файлов Windows.

IDA очень мощное средство работы с ассемблерными текстами программ. Обладает широким спектром возможностей, имеет более удобный интерфейс, чем Hacker-VIEW. Очень хорошо предусмотрена архитектура работы программ в Windows (такие вещи, как DLL, расширенный режим работы с памятью и т.д.).

Декомпиляторы программ занимают свое место в инструментарии взломщика. В основном это совместное использование с отладчиками.

Второе средство – отладчики.

Отладчики позволяют запускать отдельные части программы и следить за изменениями, которые она производит, за результатами ее работы.

Защите от отладки не стоит уделять много времени, т.к. все возможные хитрости и приемы уже известны и взломщикам и программистам. Так же и шифрование. Любой хакер, если получает заказ на взлом, имеет доступ к нормальной копии программы. То есть он ее либо может купить, либо попользоваться ею на компьютере покупателя.

Среди отладчиков выделим: SOFTICE и WINICE.

С появлением Windows отладка программ стала на порядок проще и намного удобнее дизассемблирования. Принципиально изменился стиль некоторых атак на защиту программ. Теперь не надо шаг за шагом смотреть на ассемблерный код, «продираться» сквозь дебри незначащих кодов и защит. Теперь надо отловить нужное событие и понять как на него реагирует программа. Это, конечно, не всегда бывает так просто, как выглядит на словах. Как и ранее, отладка требует знание архитектуры операционной системы.

Неважно насколько сложным был бы механизм защиты, все сводится к простейшей проверке или дешифровке. И взлом, в случае с проверкой, можно разбить на два этапа: установка «брейков» на «подозрительные» флаги, обнаруженные в процедуре защиты; анализ обращений к флагам. По реакции программы можно судить флаг это или просто переменная.

Практическая часть.

1. Изучить теоретическую часть. Сделать записи в тетради.
2. Провести сравнение декомпилятора и отладчика. По данным составить таблицу сравнений.
3. Ответить на контрольные вопросы

Способы встраивания защитных механизмов в программное обеспечение

Встраивание защитных механизмов можно выполнить следующими основными способами:

- вставкой фрагмента проверочного кода в исполняемый файл;
- преобразованием исполняемого файла к неисполняемому виду (шифрование, архивация с неизвестным параметром и т.д.) и применением для загрузки не средств операционной среды, а некоторой программы, в теле которой и осуществляются необходимые проверки;
- вставкой проверочного механизма в исходный код на этапе разработки и отладки программного продукта;
- комбинированием указанных методов.

Применительно к конкретной реализации защитных механизмов для конкретной вычислительной архитектуры можно говорить о защитном фрагменте в исполняемом или исходном коде. К процессу и результату встраивания защитных механизмов можно предъявить следующие требования:

- высокая трудоемкость обнаружения защитного фрагмента при статическом исследовании (особенно актуальна при встраивании в исходный код программного продукта);
- высокая трудоемкость обнаружения защитного фрагмента при динамическом исследовании (при отладке и трассировке по внешним событиям);
- высокая трудоемкость обхода или редуцирования защитного файла.

Возможность встраивания защитных фрагментов в исполняемый код обусловлена типовой архитектурой исполняемых модулей различных операционных сред, содержащих, как правило, адрес точки входа в исполняемый модуль. В этом случае добавление защитного фрагмента происходит следующим образом. Защитный фрагмент добавляется к началу или концу исполняемого файла, точка входа корректируется таким образом, чтобы при загрузке управление передалось дополнительному защитному фрагменту, а в составе защитного фрагмента предусматривается процедура возврата к оригинальной точке входа. Достаточно часто оригинальный исполняемый файл подвергается преобразованию. В этом случае перед возвратом управления оригинальной точке входа производится преобразование

образа оперативной памяти загруженного исполняемого файла к исходному виду.

В случае дополнения динамических библиотек возможна коррекция указанным образом отдельных функций.

Существенным недостатком рассмотренного метода является его легкая обнаруживаемость и в случае отсутствия преобразования оригинального кода исполняемого файла – легкая возможность обхода защитного фрагмента путем восстановления оригинальной точки входа.

Обфускация программ

В данном подразделе кратко затрагиваются вопросы, связанные с активно развивающимися теорией и практикой обфускации программ. Неформально говоря, под *обфускацией программ*здесь понимается преобразование программ с целью максимального затруднения их анализа и модификации, при сохранении, в то же время, их функциональных возможностей. Известные на сегодня методы обфускации, как правило, носят эмпирический характер и слабо обоснованы теоретически. В работе [CTL] приведена классификация методов обфускации, а в работе [BGI] предпринята, скорее всего, первая попытка формализации и теоретического обоснования задачи обфускации программ.

Неформально, *обфускатор*– это (эффективный, вероятностный) «компилятор», который в качестве входа имеет программуPи производит новую программу$\boldsymbol{O}(P)$, которая имеет те же самые функциональные возможности, как иP, но, в то же время, программа$\boldsymbol{O}(P)$ является «*неясной*», («*непонятной*») для противника (наблюдателя, постороннего лица) в некотором заранее определенном смысле. Обфускаторы, если будет доказано, что они существуют, могут применяться для защиты программного обеспечения и, кроме того, могут иметь широкую область криптографических и теоретико-сложностных приложений. Существование обфускаторов для этих приложений

должно быть основано на формализации определения понятия «неясности» [BGI].

Методы компенсации угроз информационной безопасности

Таблица 15.2. Методы компенсации угроз информационной безопасности

Атаки, осуществляемые атакующей стороной класса ВН (внешний наблюдатель)		
A1	Перехват, накопление, анализ информации	защита трафика
A2	Разведка топологии сети	туннелирование трафика
A3	Атаки на систему управления сетью	аутентификация трафика, защита трафика, авторизованный доступ
A4	Проникновение в сеть с целью компрометации информации на серверах сети	аутентификация трафика, защита трафика, авторизованный доступ
A5	Атаки путем фальсификации или повторной передачи	аутентификация трафика, защита трафика и проверка целостности
Атаки, осуществляемые атакующей стороной класса ВА (внешний абонент)		
B1	Компрометация ресурсов сети для повышения эффективности атак нижележащих классов	Аутентификация трафика, авторизованный доступ
B2	Маскировка под пользователей своей сети	Аутентификация трафика, авторизованный доступ
Атаки, осуществляемые атакующей стороной класса ЗП (зарегистрированный пользователь) и ЗПП (зарегистрированный привилегированный пользователь)		
C1	Компрометация ресурсов сети для повышения эффективности атак нижележащих классов	Аутентификация трафика, авторизованный доступ
Атаки, осуществляемые атакующей стороной класса СА (системный администратор)		
D1	Атаки на систему защиты сети	Аутентификация трафика, событийное протоколирование

D2	Усиление атак нижележащих классов путем эффективной маскировки под другого участника, кооперация с другими атакующими сторонами	Протоколирование действий оператора, оперативная сигнализация
D3	Компрометация системы управления	Протоколирование действий оператора, разделение ответственности, взаимный контроль
Атаки, осуществляемые атакующей стороной класса АБ (администратор безопасности)		
E1	Атаки на систему протоколирования сети	Протоколирование действий оператора, разделение ответственности, взаимный контроль
E2	Атаки на систему защиты трафика и аутентификации	Протоколирование действий оператора, разделение ответственности, взаимный контроль
E3	Компрометация сертификатов	Организационно-штатные мероприятия, организация учета и контроля
E4	Усиление атак нижележащих классов путем маскировки под другого участника, кооперация с другими атакующими сторонами	Протоколирование действий оператора, разделение ответственности, взаимный контроль

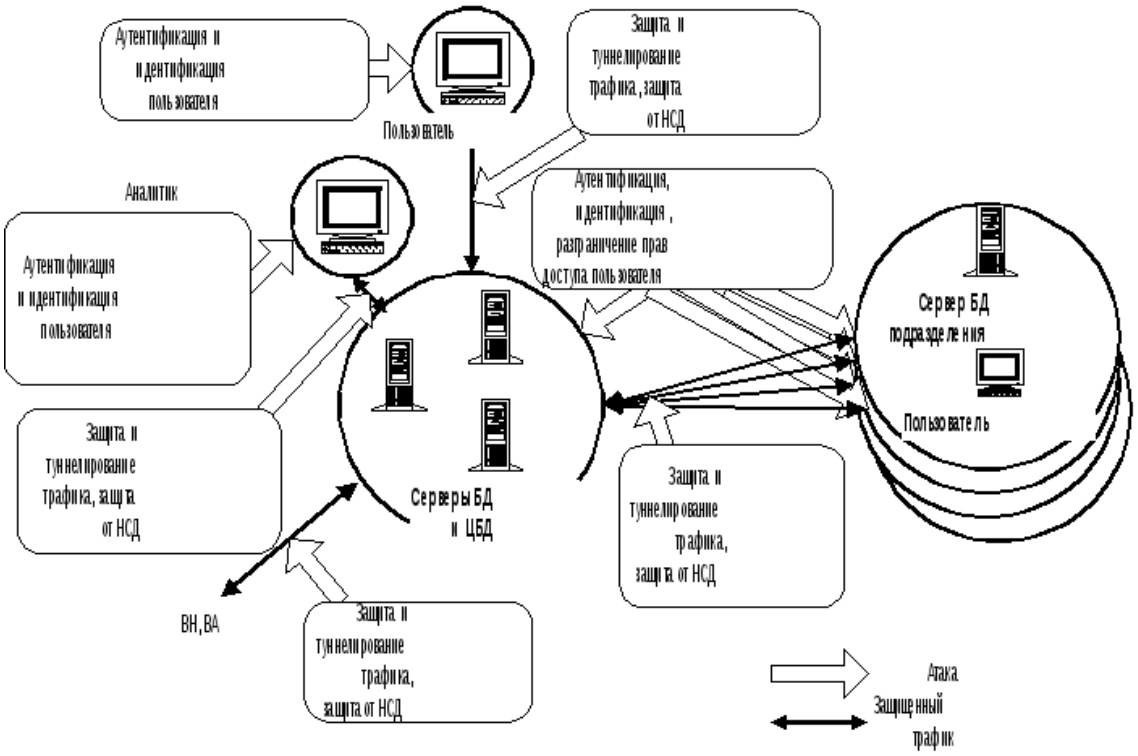

Рис. 1 - Методы компенсации угроз информационной безопасности

Понятие разрушающего программного воздействия

Существующая на сегодняшний день концепция построения защищенных компьютерных систем (КС) подразумевает использование в едином комплексе программных средств различного назначения. Так, система автоматизированного документооборота банка может использовать на одних и тех же аппаратных средствах (например, компьютере, аппаратуре передачи данных (модемах) и т.д.) взаимосвязанный комплекс программных средств: операционную среду, программные средства управления базой данных (СУБД), телекоммуникационные средства, средства обработки текстов (редакторы, текстовые процессоры), возможно, также средства антивирусного контроля, разграничения доступа к программам и данным, средства криптографической защиты передаваемой и хранимой информации, средства криптографической идентификации и аутентификации (электронная цифровая подпись) и многое другое.

Важным моментом при работе прикладных программ, в особенности средств защиты информации, является необходимость потенциального невмешательства иных присутствующих в КС прикладных или системных программ в процесс обработки информации.

Напомним, что под несанкционированным доступом (НСД) к ресурсам защищенной КС понимаются действия по использованию, изменению и уничтожению исполняемых модулей и массивов данных, принадлежащих указанной системе, производимые субъектом, не имеющим права на такие действия. Данного субъекта будем называть злоумышленником (нарушителем). Остальные субъекты именуются легальными пользователями. Данное априорно деление предполагает несколько существенно важных моментов:

- система имеет механизм различения злоумышленников и легальных пользователей;
- в системе имеется пассивная и активная компоненты (исполняемые модули и данные), пользование которыми злоумышленником нежелательно;
- в системе имеется механизм установления соответствия субъекта и информации, к которой он имеет доступ.

Как уже отмечалось, в настоящее время для интегрального обозначения процедур обеспечения безопасности информации употребляют термин "политика безопасности", а всевозможные ситуации нарушения априорно предписанных правил называют нарушениями безопасности.

Считая, что злоумышленник в совершенстве владеет всем программным и аппаратным обеспечением системы, можно предполагать, что несанкционированный доступ может быть вызван следующими причинами:

- отключением или видоизменением защитных механизмов злоумышленником (сюда же можно отнести процедуры доступа "мимо" средств контроля, например, при нарушении уровня иерархии информационных объектов – доступ к объектам типа "файл" как к последовательности секторов и др.);

- входом злоумышленника в систему под именем и с полномочиями легального пользователя.

В первом случае злоумышленник должен видоизменить защитные механизмы в системе (например, отключить программу запросов паролей пользователей). Во втором — каким-либо образом выяснить или подделать идентификатор реального пользователя (например, "подсмотреть" пароль, вводимый с клавиатуры). В обоих случаях НСД можно представить моделью опосредованного несанкционированного доступа – когда проникновение в систему осуществляется на основе некоторого воздействия, произведенного предварительно внедренной в систему программой (программами).

Например, злоумышленник пользуется информацией, которая извлечена из некоторого массива данных, созданного при совместной работе программного средства злоумышленника и системы проверки прав доступа (т.е. предварительно внедренная в систему программа при доступе легального пользователя перехватит его пароль и сохранит в заранее известном и доступном злоумышленнику месте, а затем злоумышленник воспользуется данным паролем для входа в систему). Либо злоумышленник изменит часть системы защиты так, чтобы она перестала выполнять свои функции. Например, модифицирует систему проверки прав доступа

так, чтобы она пропускала любого пользователя, или изменит программу шифрования вручную (или при помощи некоторой другой программы) таким образом, чтобы она перестала шифровать или изменила алгоритм шифрования на более простой.

Программой с потенциально опасными последствиями (badware – "вредные программы") назовем некоторую программу (осмысленный набор инструкций для какого-либо процессора), которая способна выполнить любое непустое подмножество перечисленных функций:

1) скрыть признаки своего присутствия в программной среде КС;

2) реализовать самодублирование, ассоциирование себя с другими программами и/или перенос своих фрагментов в иные (не занимаемые изначально указанной программой) области оперативной или внешней памяти;

3) разрушить (исказить произвольным образом) код программ (отличных от нее) в оперативной памяти КС;

4) перенести (сохранить) фрагменты информации из оперативной памяти в некоторые области оперативной или внешней памяти прямого доступа (локальных или удаленных);

5) имеет потенциальную возможность исказить произвольным образом, заблокировать и/или подменить выводимый во внешнюю память или в канал связи массив информации, образовавшийся в результате работы прикладных программ или уже находящийся во внешней памяти, либо изменить его параметры.

Программы с потенциально опасными последствиями можно (весьма условно) разделить на три класса.

1. Классические программы — "вирусы" (термин применен в 1984 г. Ф. Коэном). Особенность данного класса вредных программ заключается в ненаправленности их воздействия на конкретные типы прикладных программы, а также в том, что во главу угла ставится самодублирование вируса. Разрушение информации вирусом не направлено на конкретные программы и встречается у 10...20% вирусов.

2. Программы типа "программный червь" или "троянский конь" и фрагменты программ типа "логический люк". В данном случае имеет место обратная ситуация — самодублирование не всегда присуще такого рода программам или фрагментам программ, но они обладают возможностью перехвата конфиденциальной информации, или извлечения информации из сегментов систем безопасности, или разграничения доступа.

3. Программные закладки или разрушающие программные воздействия (РПВ) — обобщенный класс программ (в смысле отсутствия конкретных признаков) с потенциально опасными последствиями, обязательно

реализующие хотя бы один из п.п. 3 - 5 определения программы с потенциально опасными последствиями.

Далее наряду с термином "программа с потенциально опасными последствиями" будут использованы термины "закладка", "программная закладка" либо сокращение РПВ.

Кроме того, программные закладки можно классифицировать по методу и месту их внедрения и применения (т.е. по "способу доставки" в компьютерную систему):

- закладки, ассоциированные с программно-аппаратной средой компьютерной системы (основная BIOS или расширенные BIOS);
- закладки, ассоциированные с программами первичной загрузки (находящиеся в Master Boot Record или BOOT-секторах активных разделов) – загрузочные закладки;
- закладки, ассоциированные с загрузкой драйверов DOS, драйверов внешних устройств других ОС, командного интерпретатора, сетевых драйверов, т.е. с загрузкой операционной среды;
- закладки, ассоциированные с прикладным программным обеспечением общего назначения (встроенные в клавиатурные и экранные драйверы, программы тестирования компьютеров, утилиты и оболочки типа NORTON);
- исполняемые модули, содержащие только код закладки (как правило, внедряемые в файлы пакетной обработки типа .BAT);
- модули-имитаторы, совпадающие по внешнему виду с некоторыми программами, требующими ввода конфиденциальной информации — наиболее характерны для Unix-систем;
- закладки, маскируемые под программные средства оптимизационного назначения (архиваторы, ускорители обмена с диском и т.д.);

- закладки, маскируемые под программные средства игрового и развлекательного назначения (как правило, используются для первичного внедрения закладок).

Как видно, программные закладки имеют много общего с классическими вирусами, особенно в части ассоциирования себя с исполняемым кодом (загрузочные вирусы, вирусы-драйверы, файловые вирусы). Кроме того, программные закладки, как и многие известные вирусы классического типа, имеют развитые средства борьбы с отладчиками и дизассемблерами.

Для того чтобы закладка смогла выполнить какие-либо действия по отношению к прикладной программе или данным, она должна получить управление, т.е. процессор должен начать выполнять инструкции (команды), относящиеся к коду закладки. Это возможно только при одновременном выполнении двух условий:

1) закладка должна находиться в оперативной памяти до начала работы программы, которая является целью воздействия закладки, следовательно, она должна быть загружена раньше или одновременно с этой программой;

2) закладка должна активизироваться по некоторому общему как для закладки, так и для программы событию, т.е. при выполнении ряда условий в программно-аппаратной среде управление должно быть передано программе-закладке. Данное событие далее будем называть активизирующим.

Обычно выполнение указанных условий достигается путем анализа и обработки закладкой общих относительно закладки и прикладной программы воздействий (как правило, прерываний) либо событий (в зависимости от типа и архитектуры операционной среды). Причем прерывания должны сопровождать работу прикладной программы или работу всего

компьютера. Данные условия являются необходимыми (но недостаточными), т.е. если они не выполнены, то активизация кода закладки не произойдет и код не сможет оказать какого-либо воздействия на работу прикладной программы.

Кроме того, возможен случай, когда при запуске программы (активизирующим событием является запуск программы) закладка разрушает некоторую часть кода программы, уже загруженной в оперативную память (ОП), и, возможно, систему контроля целостности кода или контроля иных событий и на этом заканчивает свою работу. Данный случай не противоречит необходимым условиям.

С учетом замечания о том, что закладка должна быть загружена в ОП раньше, чем цель ее воздействий, можно выделить закладки двух типов.

1. Закладки резидентного типа — находятся в памяти постоянно с некоторого момента времени до окончания сеанса работы компьютера (выключения питания или перезагрузки). Закладка может быть загружена в память при начальной загрузке компьютера, загрузке операционной среды или запуске некоторой программы (которая по традиции называется вирусоносителем или просто носителем), а также запущена отдельно.

2. Закладки нерезидентного типа — начинают работу, как и закладки резидентного типа, но заканчивают ее самостоятельно через некоторый промежуток времени или по некоторому событию, при этом выгружая себя из памяти целиком.

Контрольные вопросы:

1. Что такое дизассемблер?
2. Как происходит защита программ от дизассемблирования?
3. Как происходит защита программ от отладки?
4. Какие виды отладчиков вы знаете?
5. Что такое декомпилятор?
6. Какие он функции выполняет?
7. Что такое трассировка?
8. Какие виды дизассемблеров вам известны?
9. Какие приемы дизассемблирования вам известны?

***Лабораторная работа № 3*. Восстановление зараженных файлов. Профилактика проникновения «троянских программ». Настройка безопасности почтового клиента.**

***Цель работы*:** ознакомиться с возможностями профилактики вирусов и восстановления заражённых файлов.

Краткие теоретические сведения.

Макровирусы заражают файлы – документы и электронные таблицы популярных офисных приложений [10].

Для анализа макровирусов необходимо получить текст их макросов. Для нешифрованных («не - стелс») вирусов это достигается при помощи меню Сервис/Макрос. Если же вирус шифрует свои макросы или использует «стелс»-приемы, то необходимо воспользоваться специальными утилитами просмотра макросов.

Задание: восстановить файл, зараженный макровирусом

Алгоритм выполнения работы.

Для восстановления документов Word и Excel достаточно сохранить пораженные файлы в текстовый формат RTF, содержащий практически всю информацию из первоначальных документов и не содержащий макросы.

Для этого выполните следующие действия.

1. В программе **WinWord** выберите пункты меню «**Файл**» – «**Сохранить как**».

2. В открывшемся окне в поле «**Тип файла**» выберите «**Текст в формате RTF**» (рис. 7).

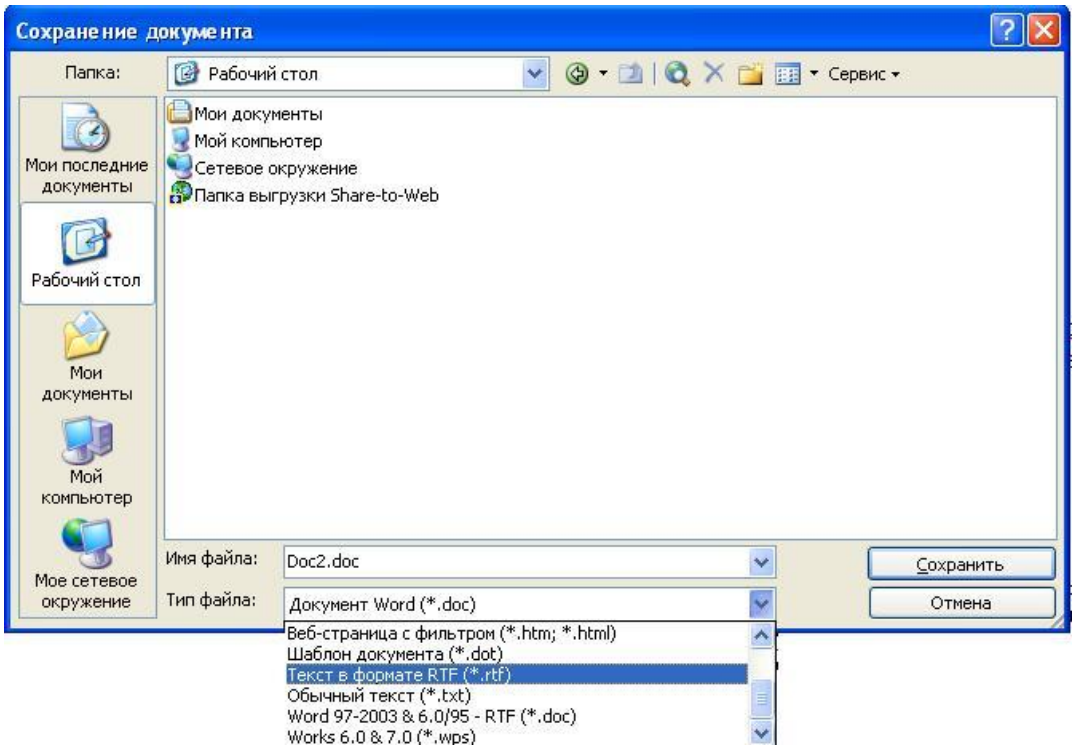

Рис. 7

3. Выберите команду **Сохранить**, при этом имя файла оставьте прежним.

4. В результате появится новый файл с именем существующего, но с другим расширением.

5. Далее закройте **WinWord** и удалите все зараженные Word-документы и файл-шаблон **NORMAL.DOT** в папке **WinWord**.

6. Запустите **WinWord** и восстановите документы из RTF-файлов в соответствующий формат файла (рис. 8) с расширением (.doc).

7. В результате этой процедуры вирус будет удален из системы, а практически вся информация останется без изменений.

Примечание:

а) этот метод рекомендуется использовать, если нет соответствующих антивирусных программ;

б) при конвертировании файлов происходит потеря невирусных макросов, используемых при работе. Поэтому перед запуском описанной процедуры следует сохранить их исходный текст, а после обезвреживания вируса – восстановить необходимые макросы в первоначальном виде.

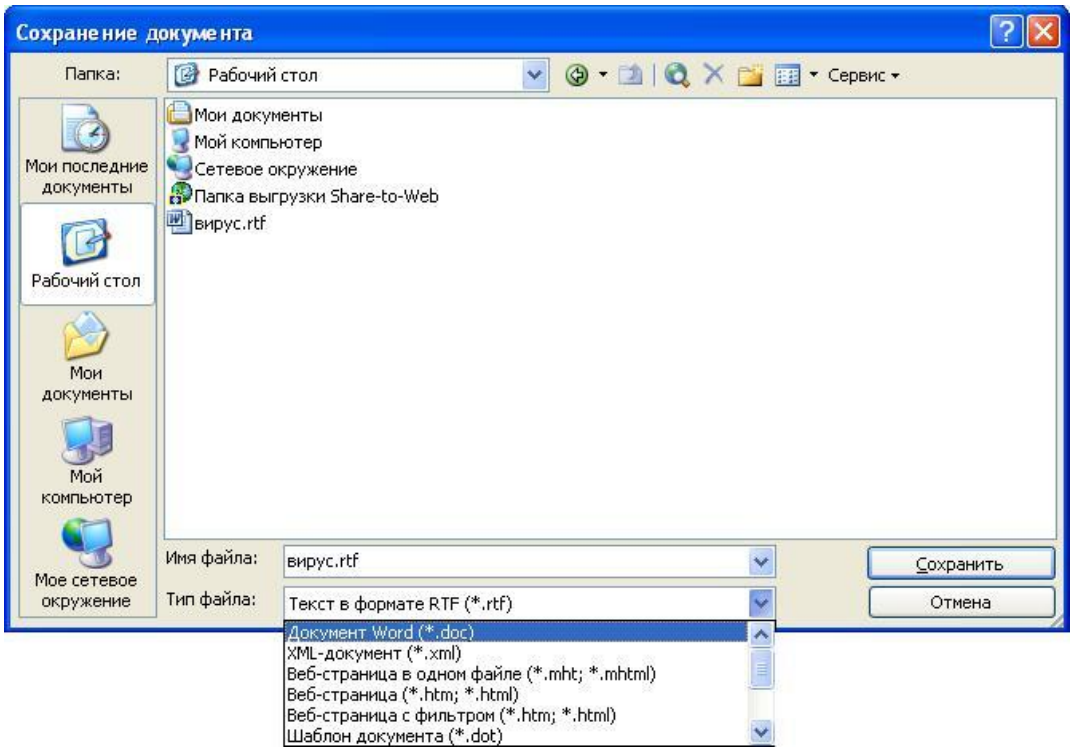

Рис. 8

8. Для последующей защиты файлов от макровирусов включите защиту от запуска макросов.

9. Для этого в **WinWord** выберите последовательно пункты меню: **Сервис - Макрос - Безопасность** (рис. 9).

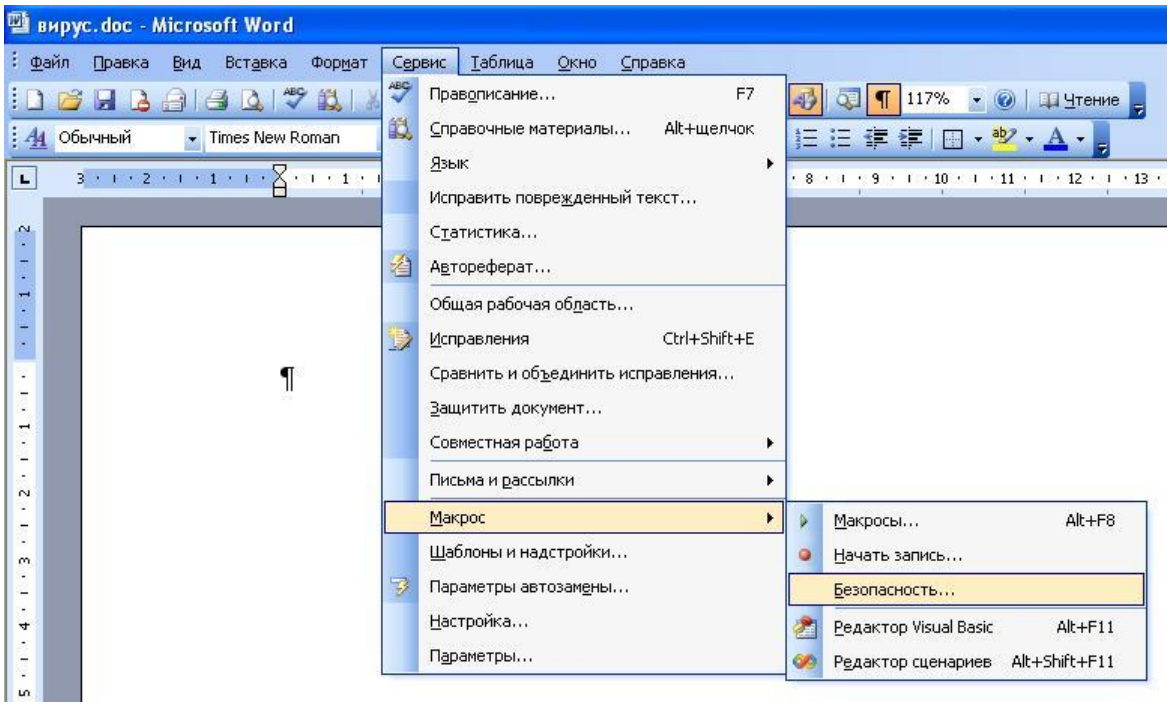

Рис. 9

10. В открывшемся окне на закладке **Уровень безопасности** отметьте пункт **Высокая** (рис. 10).

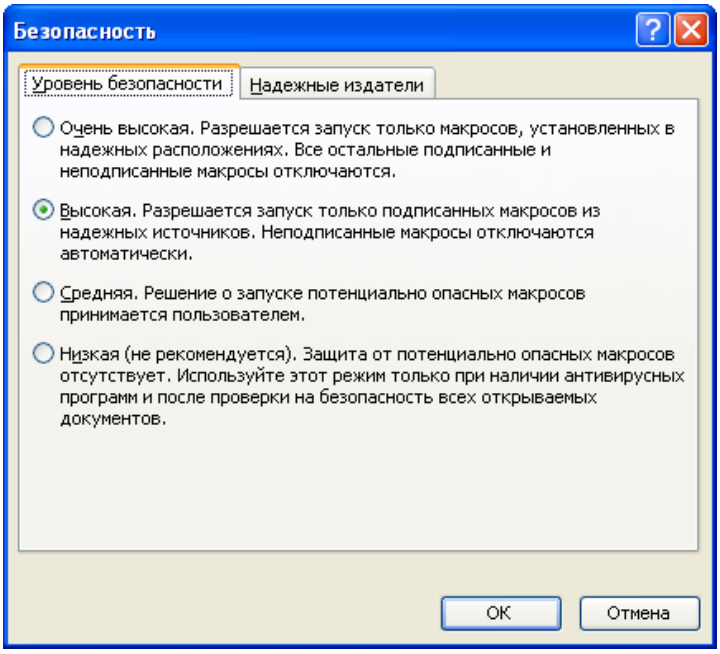

Рис. 10

Задания для самостоятельной работы

1. Создайте файл **virus.doc**(содержание – чистый лист) и выполните алгоритм восстановления файла (в предположении его заражения макровирусом).

2. Зафиксируйте этапы работы, используя команду **PrintScreen**клавиатуры (скопированные таким образом файлы вставьте в новый Word-документ для отчета преподавателю).

3. Сравните размеры файлов **virus.doc**и**virus.rtf**, используя пункт контекстного меню**Свойства**(для этого выделите в**Проводнике**файл, нажмите правую кнопку мыши и выберите пункт**Свойства**).

Контрольные вопросы

1. Какие файлы заражают макровирусы?
2. Как просмотреть код макровируса?
3. Как восстановить файл, зараженный макровирусом?

Практическая работа 2 Профилактика проникновения «троянских программ»

Краткие теоретические сведения

Главное отличие «троянских программ» от компьютерных вирусов состоит в том, что они не размножаются на зараженном компьютере и не имеют встроенных возможностей к самораспространению. “Троянские кони” засылаются пользователям (обычно через электронную почту) непосредственно их авторами под видом каких-нибудь полезных утилит. На самом деле они производят несанкционированное внедрение на компьютеры и в корпоративные сети различного рода нежелательных программ. Именно этой особенности “Троянские кони” обязаны своим названием.

Троянские программы различаются между собой по тем действиям, которые они производят на зараженном компьютере.

Backdoor – троянские утилиты удаленного администрирования

Троянские программы этого класса являются утилитами удаленного администрирования компьютеров в сети. По своей функциональности они во многом напоминают различные системы администрирования, разрабатываемые и распространяемые фирмами-производителями программных продуктов.

Единственная особенность этих программ заставляет классифицировать их как вредные троянские программы: отсутствие предупреждения об инсталляции и запуске. При запуске «троянец» устанавливает себя в системе и затем следит за ней, при этом пользователю не выдается никаких сообщений о действиях троянца в системе. Более того, ссылка на «троянца» может отсутствовать в списке активных приложений. В результате «пользователь» этой троянской программы может и не знать о ее присутствии в системе, в то время как его компьютер открыт для удаленного управления.

Утилиты скрытого управления позволяют делать с компьютером все, что в них заложил автор: принимать или отсылать файлы, запускать и уничтожать их, выводить сообщения, стирать информацию, перезагружать компьютер и т. д. В результате эти троянцы могут быть использованы для обнаружения и передачи конфиденциальной информации, для запуска вирусов, уничтожения данных и т.п. – пораженные компьютеры оказываются открытыми для злоумышленных действий хакеров.

Таким образом, троянские программы данного типа являются одним из самых опасных видов вредоносного программного обеспечения, поскольку в них заложена возможность самых разнообразных злоумышленных действий, присущих другим видам троянских программ.

Отдельно следует отметить группу бэкдоров, способных распространяться по сети и внедряться в другие компьютеры, как это делают компьютерные черви. Отличает такие «троянцы» от червей тот факт, что они распространяются по сети не самопроизвольно (как черви), а только по специальной команде «хозяина», управляющего данной копией троянской программы.

Trojan-PSW – воровство паролей

Данное семейство объединяет троянские программы, «ворующие» различную информацию с зараженного компьютера, обычно –системные пароли (PSW – Password-Stealing-Ware). При запуске PSW-троянцы ищут системные файлы, хранящие различную конфиденциальную информацию (обычно номера телефонов и пароли доступа к интернету) и отсылают ее по указанному в коде «троянца» электронному адресу или адресам.

Существуют PSW-троянцы, которые сообщают и другую информацию о зараженном компьютере, например, информацию о системе (размер памяти и дискового пространства, версия операционной системы), тип используемого почтового клиента, IP-адрес и т.п. Некоторые троянцы данного типа «воруют» регистрационную информацию к различному программному обеспечению, коды доступа к сетевым играм и прочее.

Trojan-AOL – семейство троянских программ, «ворующих» коды доступа к сети AOL (America Online). Выделены в особую группу по причине своей многочисленности.

Trojan-Clicker – интернет-кликеры

Семейство троянских программ, основная функция которых – организация несанкционированных обращений к интернет-ресурсам (обычно к веб-страницам). Достигается это либо посылкой соответствующих команд

браузеру, либо заменой системных файлов, в которых указаны «стандартные» адреса интернет-ресурсов (например, файл hosts в MS Windows).

У злоумышленника могут быть следующие цели для подобных действий:

- увеличение посещаемости каких-либо сайтов с целью увеличения показов рекламы;
- организация DoS-атаки (Denial of Service) на какой-либо сервер;
- привлечение потенциальных жертв для заражения вирусами или троянскими программами.

Trojan-Downloader – доставка прочих вредоносных программ

Троянские программы этого класса предназначены для загрузки и установки на компьютер-жертву новых версий вредоносных программ, установки «троянцев» или рекламных систем. Загруженные из интернета программы затем либо запускаются на выполнение, либо регистрируются «троянцем» на автозагрузку в соответствии с возможностями операционной системы. Данные действия при этом происходят без ведома пользователя.

Информация об именах и расположении загружаемых программ содержится в коде и данных троянца или скачивается троянцем с «управляющего» интернет-ресурса (обычно с веб-страницы).

Trojan-Dropper – инсталляторы прочих вредоносных программ

Троянские программы этого класса написаны в целях скрытной инсталляции других программ и практически всегда используются для «подсовывания» на компьютер-жертву вирусов или других троянских программ.

Данные троянцы обычно без каких-либо сообщений (либо с ложными сообщениями об ошибке в архиве или неверной версии операционной системы) сбрасывают на диск в какой-либо каталог (в корень диска C:, во временный каталог, в каталоги Windows) другие файлы и запускают их на выполнение.

Обычно структура таких программ следующая:

Основной код
Файл 1
Файл 2
...

«Основной код» выделяет из своего файла остальные компоненты (файл 1, файл 2, ...), записывает их на диск и открывает их (запускает на выполнение).

Обычно один (или более) компонент является троянской программой, и как минимум один компонент является «обманкой»: программой-шуткой, игрой, картинкой или чем-то подобным. «Обманка» должна отвлечь внимание пользователя и/или продемонстрировать то, что запускаемый файл действительно делает что-то «полезное», в то время как троянская компонента инсталлируется в систему.

В результате использования программ данного класса хакеры достигают двух целей:

- скрытная инсталляция троянских программ и/или вирусов;
- защита от антивирусных программ, поскольку не все из них в состоянии проверить все компоненты внутри файлов этого типа.

Trojan-Proxy – троянские прокси-сервера

Семейство троянских программ, скрытно осуществляющих анонимный доступ к различным интернет-ресурсам. Обычно используются для рассылки спама.

Trojan-Spy – шпионские программы

Данные троянцы осуществляют электронный шпионаж за пользователем зараженного компьютера: вводимая с клавиатуры информация, снимки экрана, список активных приложений и действия пользователя с ними сохраняются в какой-либо файл на диске и периодически отправляются злоумышленнику.

Троянские программы этого типа часто используются для кражи информации пользователей различных систем онлайновых платежей и банковских систем.

Trojan – прочие троянские программы

К данным троянцам относятся те из них, которые осуществляют прочие действия, попадающие под определение троянских программ, т.е. разрушение или злонамеренная модификация данных, нарушение работоспособности компьютера и прочее.

В данной категории также присутствуют «многоцелевые» троянские программы, например, те из них, которые одновременно шпионят за пользователем и предоставляют proxy-сервис удаленному злоумышленнику.

Trojan-Notifier – оповещение об успешной атаке

Троянцы данного типа предназначены для сообщения своему «хозяину» о зараженном компьютере. При этом на адрес «хозяина» отправляется информация о компьютере, например, IP-адрес компьютера, номер открытого порта, адрес электронной почты и т.п. Отсылка осуществляется различными способами: электронным письмом, специально оформленным обращением к веб-странице «хозяина», ICQ-сообщением.

Данные троянские программы используются в многокомпонентных троянских наборах для извещения своего «хозяина» об успешной инсталляции троянских компонент в атакуемую систему.

Реестр операционной системы Windows– это большая база данных, где хранится информация о конфигурации системы. Этой информацией пользуются как операционная система Windows, так и другие программы. Реестр содержит данные, к которым Windows XP постоянно обращается во время загрузки, работы и её завершения, а именно:

- профили всех пользователей, то есть их настройки;
- конфигурация оборудования, установленного в операционной системе.
- данные об установленных программах и типах документов, создаваемых каждой программой;
- свойства папок и значков программ;
- данные об используемых портах.

Реестр имеет иерархическую древовидную структуру, состоящую из разделов, подразделов и ключей (параметров).

В некоторых случаях восстановить работоспособность системы после сбоя можно, загрузив работоспособную версию реестра, но для этого, естественно, необходимо иметь копию реестра. Основным средством для просмотра и редактирования записей реестра служит специализированная утилита «**Редактор реестра**».

Файл редактора реестра находится в папке Windows. Называется он **regedit.exe**. Для того, чтобы запустить эту программу, необходимо выбрать**Пуск**–**Выполнить**–**regedit.exe**. После запуска появится окно редактора реестра. Вы увидите список из 5 разделов (рис. 11):

HKEY_CLASSES_ROOT.

HKEY_CURRENT_USER.

HKEY_LOCAL_MACHINE.

HKEY_USERS.

HKEY_CURRENT_CONFIG.

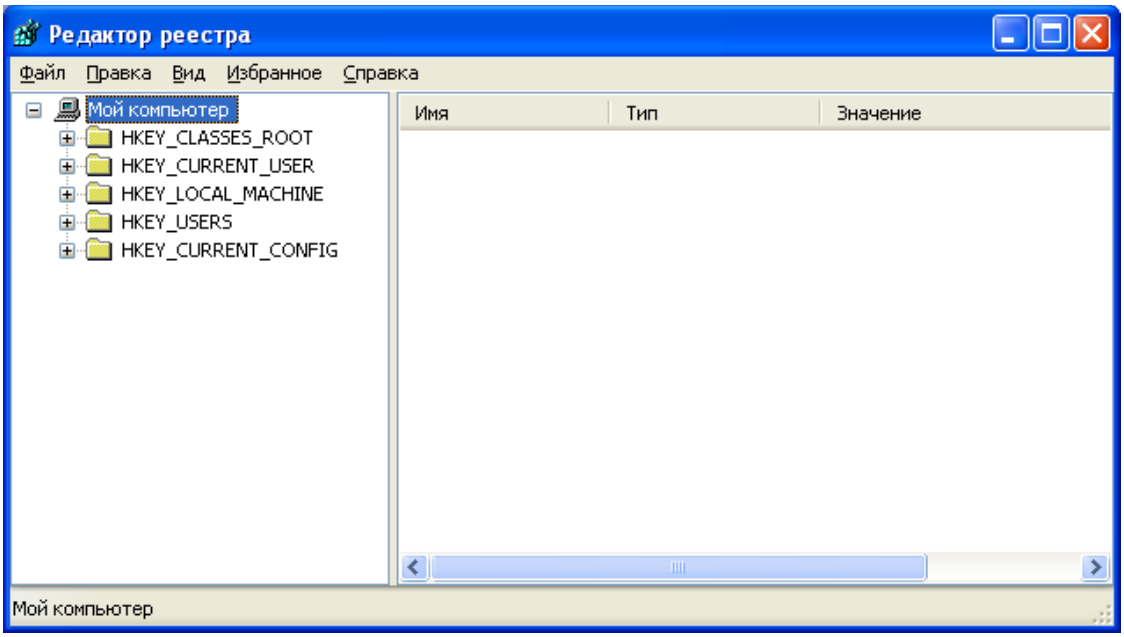

Рис. 11

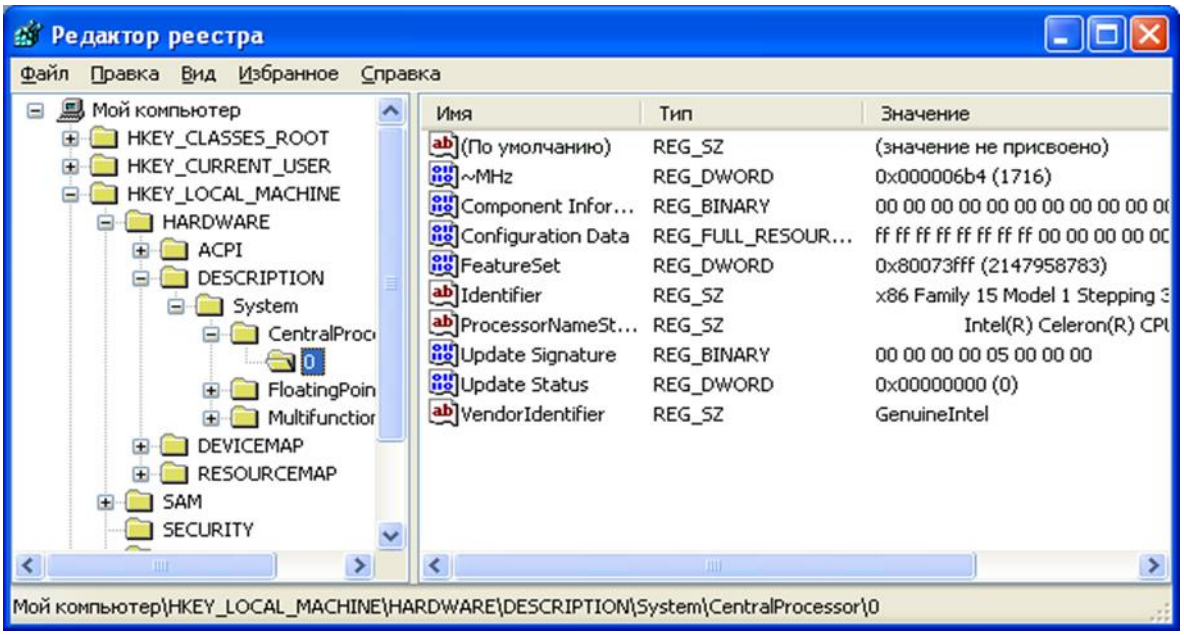

Рис. 12

Работа с разделами реестра аналогична работе с папками в Проводнике. Конечным элементом дерева реестра являются ключи или параметры, делящиеся на три типа (рис. 12):

- строковые (напр. «C:\Windows»);
- двоичные (напр. 10 82 AO 8F);

DWORD - этот тип ключа занимает 4 байта и отображается в шестнадцатеричном и в десятичном виде (например, 0x00000020 (32)).

В Windows системная информация разбита на так называемые ульи (hive). Это обусловлено принципиальным отличием концепции безопасности этих операционных систем. Имена файлов ульев и пути к каталогам, в которых они хранятся, расположены в разделе **HKEY_LOCAL_MACH1NE\SYSTEM\CurrentControlSet\Control\hivelist**(рис. 13).

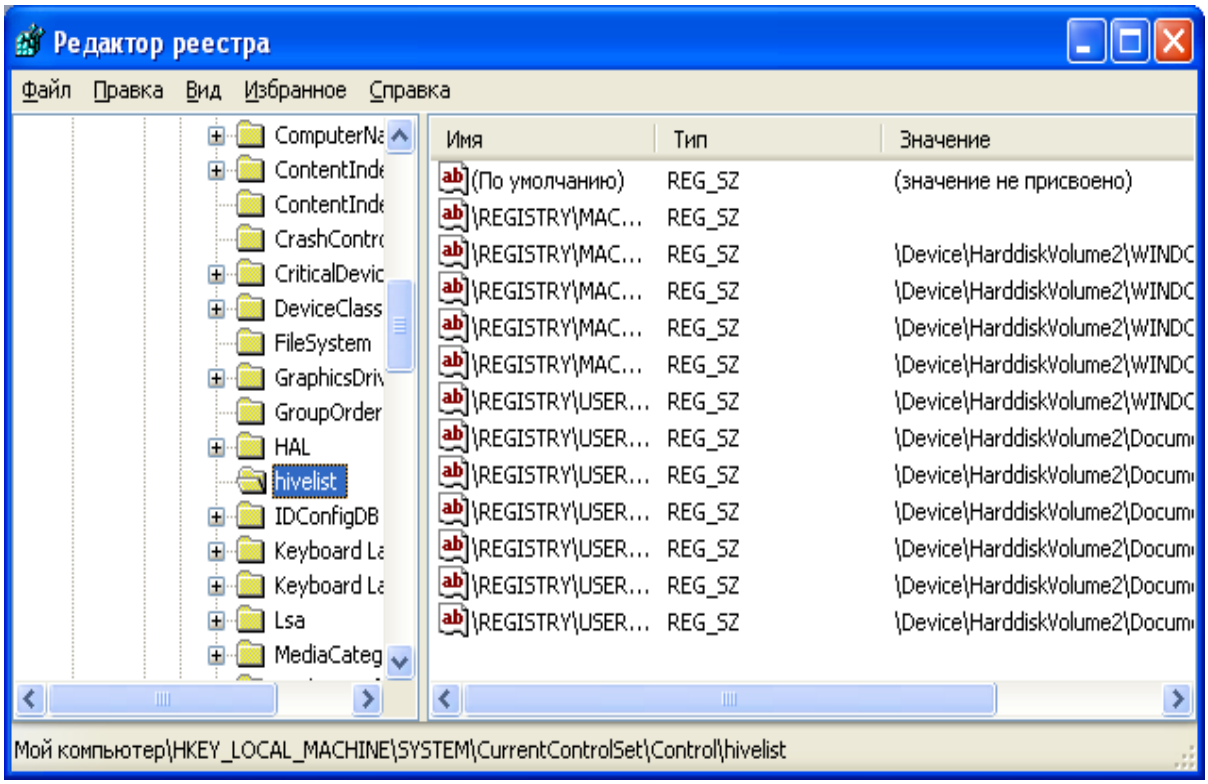

Рис. 13

В таблице 1 даны краткие описания ульев реестра и файлов, в которых хранятся параметры безопасности.

Таблица 5

Характеристика основных разделов системного реестра

HKEY_LOCAL_MACHINE\SAM	Содержит информацию SAM (Security Access Manager), хранящуюся в файлах SAM, SAM.LOG, SAM.SAV в папке %System-root%\System32\Config
HKEY_LOCAL_MACHINE\SECURITY	Содержит информацию безопасности в файлах SECURITY, SECURITY.LOG, SECURITY.SAV в папке \%Systemroot%\System32\Config
HKEY_LOCAL_MACHINE\SYSTEM	Содержит информацию об аппаратных профилях этого подраздела. Информация хранится в файлах SYSTEM, SYSTEM.LOG, SYSTEM.SAV в папке \%Systemroot%\System32\Config
HKEY_CURRENT_CONFIG	Содержит информацию о подразделе System этого улья, которая хранится в файлах SYSTEM.SAV и SYSTEM.ALT в

	папке \%Systemroot%\System32\Config
HKEY_USERS\.DEFAULT	Содержит информацию, которая будет использоваться для создания профиля нового пользователя, впервые регистрирующегося в системе. Информация хранится в файлах DEFAULT, DEFAULT.LOG, DEFAULT.SAV в папке \%Systemroot%\System32\Config
HKEY_CURRENT_USER	Содержит информацию о пользователе, зарегистрированном в системе на текущий момент. Эта информация хранится в файлах NTUSER.DAT и NTUSER.DAT.LOG, расположенных в каталоге \%Systemroot%\Profiles\User name, где User name – имя пользователя

Задание: проверить потенциальные места записей «троянских программ» в системном реестре операционной системы Windows 2000 (XP).

Алгоритм выполнения работы

Потенциальными местами записей «троянских программ» в системном реестре являются разделы, описывающие программы, запускаемые автоматически при загрузке операционной системы от имени пользователей и системы.

1. Запустите программу **regedit.exe**.
2. В открывшемся окне выберите ветвь **HKEY_LOCAL_MACHINE**и далее**Software\Microsoft\WindowsNT\CurrentVersion\Winlogon**(щелкнуть по значку «папка»).
3. В правой половине открытого окна программы **regedit.exe**появится список ключей.
4. Найдите ключ **Userinit**(**REG_SZ**) и проверьте его содержимое.
5. По умолчанию (исходное состояние) 151 этот ключ содержит следующую запись **C:\WINDOWS\system32\userinit.exe**(рис. 14).

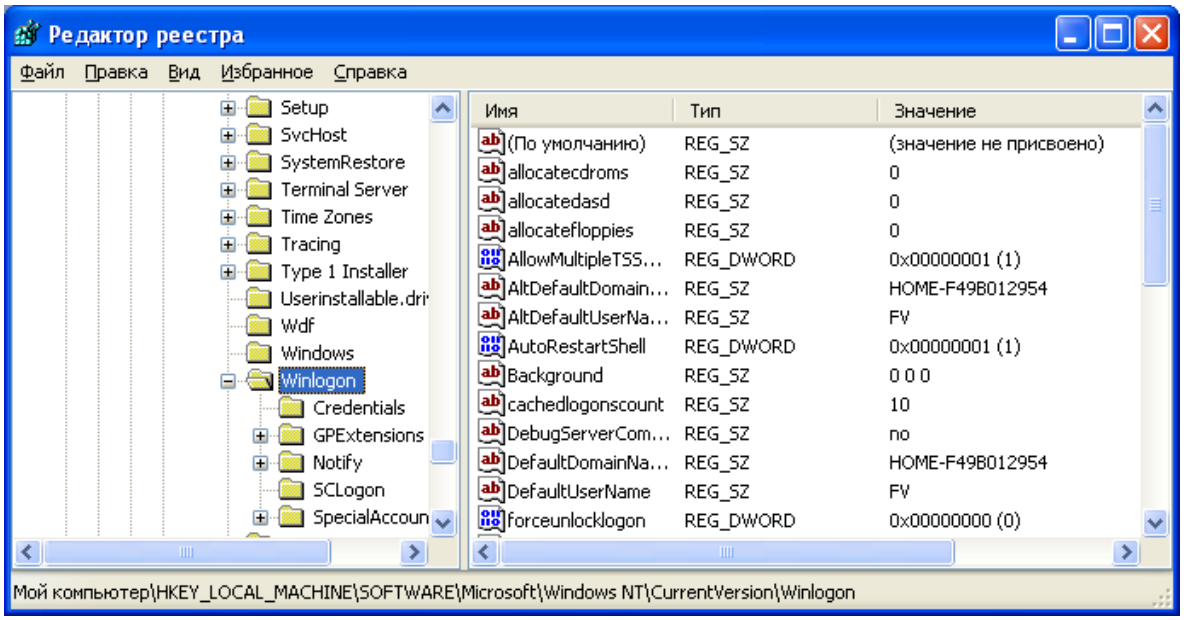

Рис. 14

6. Если в указанном ключе содержатся дополнительные записи, то это могут быть «троянские программы».

7. В этом случае проанализируйте место расположения программы, обратите внимание на время создания файла и сопоставьте с Вашими действиями в это время.

8. Если время создания файла совпадает со временем Вашей работы в Интернете, то возможно, что в это время Ваш компьютер был заражен «троянским конем».

9. Для удаления этой записи необходимо дважды щелкнуть на названии ключа (или при выделенном ключе выбрать команду **Изменить**из меню**Правка**программы**regedit.exe**).

10. В открывшемся окне в поле **Значение**(рис. 15) удалите ссылку на подозрительный файл.

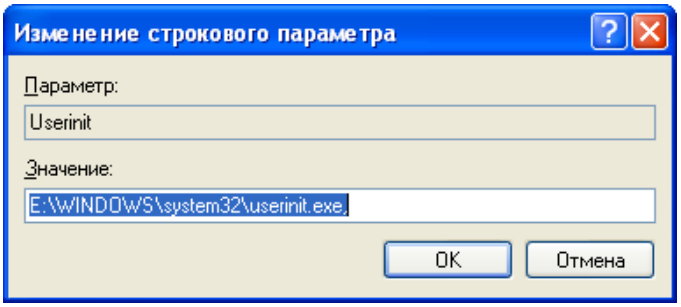

Рис. 15

11. Закройте программу **regedit.exe**.

12. Перейдите в папку с подозрительным файлом и удалите его.

13. Перезагрузите операционную систему и выполните пункты задания 1-4.

14. Если содержимое рассматриваемого ключа не изменилось, то предполагаемый «троянский конь» удален из Вашей системы.

Еще одним потенциальным местом записей на запуск «троянских программ» является раздел автозапуска **Run**.

Для его проверки выполните следующее.

1. Запустите программу **regedit.exe**.

2. В открывшемся окне выберите ветвь **HKEY_LOCAL_MACHINE**и далее**Software\Microsoft\Windows\CurrentVersion\Run** ... (**REG_SZ**) (рис. 16).

3.

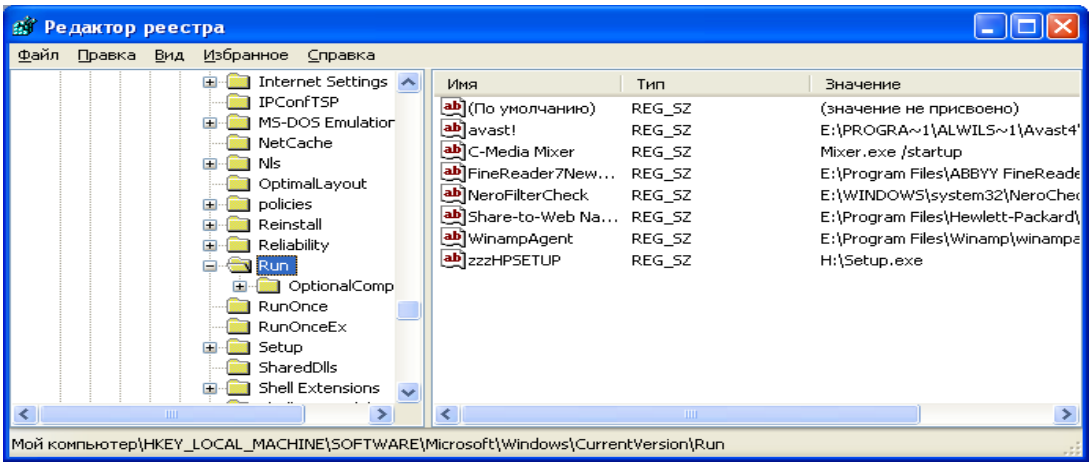

Рис. 16

3. В рассматриваемом примере автоматически запускается резидентный антивирус и его планировщик заданий, а также утилита, относящаяся к программе Nero (запись на CD).

4. Если в указанном разделе есть записи вызывающие подозрения, то выполните пункты 6-14 предыдущего задания.

Задания для самостоятельной работы

1. Проверьте содержимое ключа HKEY_LOCAL_MACHINE\ Software\ Microsoft\ WindowsNT\ CurrentVersion\ Winlogon\ System (REG_SZ).

2. Зафиксируйте этапы работы, используя команду PrintScreen клавиатуры.

3. Составьте отчет о результатах проверки.

***Лабораторная работа № 4*. Структура комплексной системы защиты информации от несанкционированного доступа (НСД); мониторинг и контроль окружающей среды; ведение специальной информационной базы данных КСИБ.**

***Цель работы*:** ознакомиться со структурой комплексной системы защиты информации от несанкционированного доступа.

На рынке защиты информации предлагается много отдельных инженерно-технических, программно-аппаратных, криптографических средств защиты информации. В литературе по защите информации можно найти описание методов и средств на их основе, теоретических моделей защиты. Однако для того, чтобы создать на предприятии условия эффективной защиты информации, необходимо объединить отдельные средства защиты в систему. При этом надо помнить, что главным элементом этой системы является человек. Причем человек является ключевым элементом системы и вместе с тем самым трудно формализуемым и потенциально слабым ее звеном.

Создание системы защиты информации (СЗИ) не является главной задачей предприятия, как, например, производство продукции и получение прибыли. Поэтому создаваемая СЗИ не должна приводить к ощутимым трудностям в работе предприятия, а создание СЗИ должно быть экономически оправданным. Тем не менее она должна обеспечивать защиту важных информационных ресурсов предприятия от всех реальных угроз.

В книге предложен комплексный подход к организации защиты информации (ЗИ) на предприятии. При этом объектом исследования является не только информационная система, но и предприятие в целом.

Рассматриваются концептуальные основы защиты информации, раскрывающие сущность, цели, структуру и стратегию защиты.

Анализируются источники, способы и результаты дестабилизирующего воздействия на информацию, а также каналы и методы несанкционированного

доступа к информации. Определяются методологические подходы к организации и технологическому обеспечению защиты информации на предприятии. Представлена архитектура, этапы построения, принципы управления комплексной системой защиты информации (КСЗИ). Особое внимание уделено проблеме «человеческого фактора».

Предложенный подход к защите информации обеспечит целостное видение проблемы, повышение качества, следовательно, и надежности защиты информации.

Следует подчеркнуть, что автор умышленно уходит от понятия «информационная безопасность», используя термин «защита информации».

Информационную безопасность принято рассматривать как обеспечение состояния защищенности:

1) личности, общества, государства от воздействия недоброкачественной информации;

2) информации и информационных ресурсов от неправомерного и несанкционированного воздействия посторонних лиц;

3) информационных прав и свобод гражданина и человека.

Поскольку в книге не рассматриваются вопросы защиты от воздействия недобросовестной информации, автор посчитал необходимым использовать более «узкий» термин.

1. Сущность и задачи комплексной системы защиты информации

1.1. Подходы к проектированию систем защиты информации

Бытует мнение, что проблемы защиты информации относятся исключительно к информации, обрабатываемой компьютером. Это, по-видимому, связано с тем, что компьютер, и в частности персональный компьютер, является «ядром», центром хранения информации. Объект информатизации, по отношению к которому направлены действия по защите информации, представляется более широким понятием по сравнению с персональным компьютером. Что же представляет собой объект информатизации и каково его место на предприятии?

ГОСТ РФ 51275-99 определяет объект информатизации как «совокупность информационных ресурсов, средств и систем обработки информации, используемых в соответствии с заданной информационной технологией, средств обеспечения объекта информатизации, помещений или объектов (зданий, сооружений, технических средств), в которых они установлены, или помещения и объекты, предназначенные для ведения конфиденциальных переговоров»[1].

Слово «совокупность» в данном определении указывает на то, что объект информатизации это единая информационная система, охватывающая в целом предприятие, учреждение, организацию.

В реальной жизни все эти отдельные «объекты информатизации» расположены в пределах одного предприятия и представляют собой единый комплекс компонентов, связанных общими целями, задачами, структурными отношениями, технологией информационного обмена и т. д.

Современное предприятие — большое количество разнородных компонентов, объединенных в сложную систему для выполнения поставленных целей, которые в процессе функционирования предприятия могут модифицироваться. Многообразие и сложность влияния внутренних и внешних факторов, которые часто не поддаются строгой количественной оценке, приводят к тому, что эта сложная система может обретать новые качества, не свойственные составляющим ее компонентам.

Характерной особенностью подобных систем является прежде всего наличие человека в каждой из составляющих ее подсистем и отдаленность (разделенность) человека от объекта его деятельности. Это происходит в связи с тем, что множество компонентов, составляющих объект информатизации, интегрально может быть представлено совокупностью трех групп систем: 1) люди (биосоциальные системы); 2) техника (технические системы и помещения, в которых они расположены); 3) программное обеспечение, которое является интеллектуальным посредником между человеком и техникой (интеллектуальные системы). Совокупность этих трех групп образует

социотехническую систему. Такое представление о социотехнических системах является достаточно широким и может быть распространено на многие объекты. Круг наших интересов ограничивается исследованием безопасности систем, предназначенных для обработки поступающей на их вход информации и выдачи результата, т. е. социотехнических систем информационного типа.

Если обратиться к истории этой проблемы, то можно условно выделить три периода развития средств защиты информации (ЗИ):

— первый относится к тому времени, когда обработка информации осуществлялась по традиционным (ручным, бумажным) технологиям;

— второй — когда для обработки информации на регулярной основе применялись средства электронной вычислительной техники первых поколений;

— третий — когда использование средств электронно-вычислительной техники приняло массовый и повсеместный характер (появление персональных компьютеров).

В 60–70 гг. проблема защиты информации решалась достаточно эффективно применением в основном организационных мер. К ним относились: режимные мероприятия, охрана, сигнализация и простейшие программные средства защиты информации. Эффективность использования этих средств достигалась за счет концентрации информации в определенных местах (спец. хранилища, вычислительные центры), что способствовало обеспечению защиты относительно малыми средствами.

«Рассосредоточение» информации по местам хранения и обработки обострило ситуацию с ее защитой. Появились дешевые персональные компьютеры. Это дало возможность построения сетей ЭВМ (локальных, глобальных, национальных и транснациональных), которые могут использовать различные каналы связи. Эти факторы способствуют созданию высокоэффективных систем разведки и получения информации. Они нашли отражение и в современных предприятиях.

Современное предприятие представляет собой сложную систему, в рамках которой осуществляется защита информации.

Рассмотрим основные особенности современного предприятия:

— сложная организационная структура;

— многоаспектность функционирования;

— высокая техническая оснащенность;

— широкие связи по кооперации;

— необходимость расширения доступа к информации;

— всевозрастающий удельный вес безбумажной технологии обработки информации;

• возрастающий удельный вес автоматизированных процедур в общем объеме процессов обработки данных;

• важность и ответственность решений, принимаемых в автоматизированном режиме, на основе автоматизированной обработки информации;

• высокая концентрация в автоматизированных системах информационных ресурсов;

• большая территориальная распределенность компонентов автоматизированных систем;

• накопление на технических носителях огромных объемов информации;

• интеграция в единых базах данных информации различного назначения и различной принадлежности;

• долговременное хранение больших объемов информации на машинных носителях;

• непосредственный и одновременный доступ к ресурсам (в т. ч. и к информации) автоматизированных систем большого числа пользователей различных категорий и различных учреждений;

• интенсивная циркуляция информации между компонентами автоматизированных систем, в том числе и удаленных друг от друга.

Таким образом, создание индустрии переработки информации, с одной стороны, создает объективные предпосылки для повышения уровня производительности труда и жизнедеятельности человека, с другой стороны, порождает целый ряд сложных и крупномасштабных проблем. Одной из них является обеспечение сохранности и установленного статуса информации, циркулирующей и обрабатываемой на предприятии.

1.2. Понятие комплексной системы защиты информации

Работы по защите информации у нас в стране ведутся достаточно интенсивно и уже продолжительное время. Накоплен существенный опыт. Сейчас уже никто не думает, что достаточно провести на предприятии ряд организационных мероприятий, включить в состав автоматизированных систем некоторые технические и программные средства — и этого будет достаточно для обеспечения безопасности.

Главное направление поиска новых путей защиты информации заключается не просто в создании соответствующих механизмов, а представляет собой реализацию регулярного процесса, осуществляемого на всех этапах жизненного цикла систем обработки информации при комплексном использовании всех имеющихся средств защиты. При этом все средства, методы и мероприятия, используемые для ЗИ, наиболее рациональным образом объединяются в единый целостный механизм — причем не только от злоумышленников, но и от некомпетентных или недостаточно подготовленных пользователей и персонала, а также нештатных ситуаций технического характера.

Основной проблемой реализации систем защиты является:

— с одной стороны, обеспечение надежной защиты, находящейся в системе информации: исключение случайного и преднамеренного получения информации посторонними лицами, разграничение доступа к устройствам и ресурсам системы всех пользователей, администрации и о обслуживающего персонала;

— с другой стороны, системы защиты не должны создавать заметных неудобств пользователям в ходе их работы с ресурсами системы.

Проблема обеспечения желаемого уровня защиты информации весьма сложная, требующая для своего решения не просто осуществления некоторой совокупности научных, научно-технических и организационных мероприятий и применения специальных средств и методов, а создания целостной системы организационно-технологических мероприятий и применения комплекса специальных средств и методов по ЗИ.

На основе теоретических исследований и практических работ в области ЗИ сформулирован системно-концептуальный подход к защите информации.

Под системностью как основной частью системно-концептуального похода понимается:

— системность целевая, т. е. защищенность информации рассматривается как основная часть общего понятия качества информации;

— системность пространственная, предлагающая взаимоувязанное решение всех вопросов защиты на всех компонентах предприятия;

— системность временная, означающая непрерывность работ по ЗИ, осуществляемых в соответствии планам;

— системность организационная, означающая единство организации всех работ по ЗИ и управления ими.

Концептуальность подхода предполагает разработку единой концепции как полной совокупности научно обоснованных взглядов, положений и решений, необходимых и достаточных для оптимальной организации и обеспечения надежности защиты информации, а также целенаправленной организации всех работ по ЗИ.

Комплексный (системный) подход к построению любой системы включает в себя: прежде всего, изучение объекта внедряемой системы; оценку угроз безопасности объекта; анализ средств, которыми будем оперировать при построении системы; оценку экономической целесообразности; изучение самой системы, ее свойств, принципов работы и возможность увеличения ее

эффективности; соотношение всех внутренних и внешних факторов; возможность дополнительных изменений в процессе построения системы и полную организацию всего процесса от начала до конца.

Комплексный (системный) подход — это принцип рассмотрения проекта, при котором анализируется система в целом, а не ее отдельные части. Его задачей является оптимизация всей системы в совокупности, а не улучшение эффективности отдельных частей. Это объясняется тем, что, как показывает практика, улучшение одних параметров часто приводит к ухудшению других, поэтому необходимо стараться обеспечить баланс противоречий требований и характеристик.

Комплексный (системный) подход не рекомендует приступать к созданию системы до тех пор, пока не определены следующие ее компоненты:

1. *Входные элементы.* Это те элементы, для обработки которых создается система. В качестве входных элементов выступают виды угроз безопасности, возможные на данном объекте;

2. *Ресурсы.* Это средства, которые обеспечивают создание и функционирование системы (например, материальные затраты, энергопотребление, допустимые размеры и т. д.). Обычно рекомендуется четко определять виды и допустимое потребление каждого вида ресурса как в процессе создания системы, так и в ходе ее эксплуатации;

3. *Окружающая среда.* Следует помнить, что любая реальная система всегда взаимодействует с другими системами, каждый объект связан с другими объектами. Очень важно установить границы области других систем, не подчиняющихся руководителю данного предприятия и не входящих в сферу его ответственности.

Характерным примером важности решения этой задачи является распределение функций по защите информации, передаваемой сигналами в кабельной линии, проходящей по территориям различных объектов. Как бы ни устанавливались границы системы, нельзя игнорировать ее взаимодействие с окружающей средой, ибо в этом случае принятые решения могут оказаться

бессмысленными. Это справедливо как для границ защищаемого объекта, так и для границ системы защиты;

4. *Назначение и функции.* Для каждой системы должна быть сформулирована цель, к которой она (система) стремится. Эта цель может быть описана как назначение системы, как ее функция. Чем точнее и конкретнее указано назначение или перечислены функции системы, тем быстрее и правильнее можно выбрать лучший вариант ее построения. Так, например, цель, сформулированная в самом общем виде как обеспечение безопасности объекта, заставит рассматривать варианты создания глобальной системы защиты. Если уточнить ее, определив, например, как обеспечение безопасности информации, передаваемой по каналам связи внутри здания, то круг возможных решений существенно сузится. Следует иметь в виду, что, как правило, глобальная цель достигается через достижение множества менее общих локальных целей (подцелей). Построение такого «дерева целей» значительно облегчает, ускоряет и удешевляет процесс создания системы;

5. *Критерий эффективности.* Необходимо всегда рассматривать несколько путей, ведущих к цели, в частности нескольких вариантов построения системы, обеспечивающей заданные цели функционирования. Для того чтобы оценить, какой из путей лучше, необходимо иметь инструмент сравнения — критерий эффективности. Он должен: характеризовать качество реализации заданных функций; учитывать затраты ресурсов, необходимых для выполнения функционального назначения системы; иметь ясный и однозначный физический смысл; быть связанным с основными характеристиками системы и допускать количественную оценку на всех этапах создания системы.

Таким образом, учитывая многообразие потенциальных угроз информации на предприятии, сложность его структуры, а также участие человека в технологическом процессе обработки информации, цели защиты информации могут быть достигнуты только путем создания СЗИ на основе комплексного подхода.

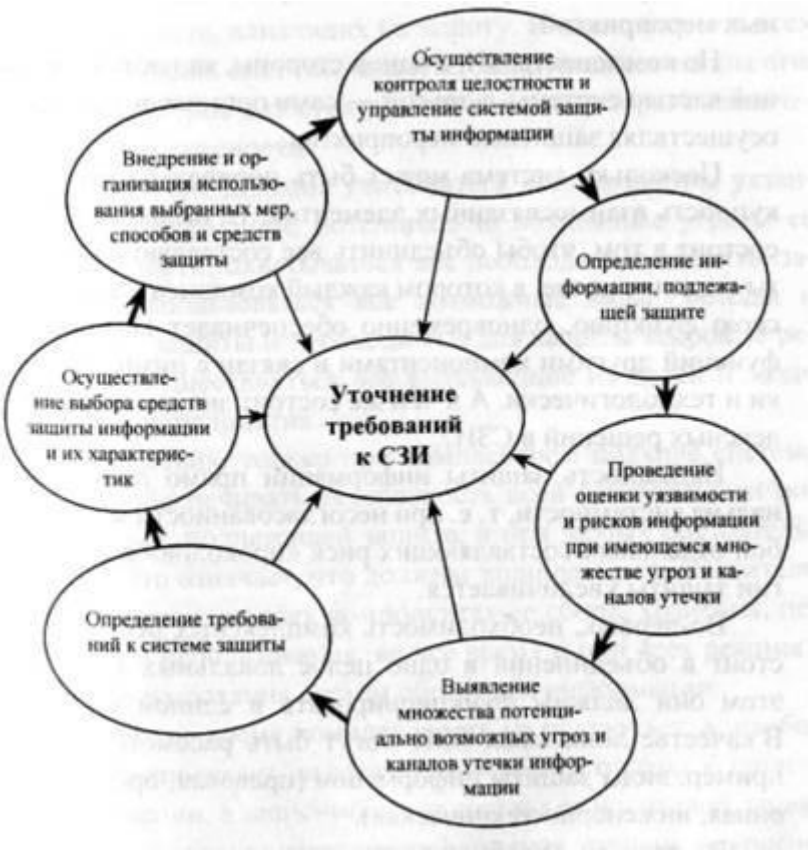

Рис. 1. Непрерывный цикл создания СЗИ

Процесс создания комплексной системы защиты информации может быть представлен в виде непрерывного цикла, так как это показано на рис. 1.

1.3. Назначение комплексной системы защиты информации

Главная цель создания системы защиты информации — ее надежность. Система ЗИ — это организованная совокупность объектов и субъектов ЗИ, используемых методов и средств защиты, а также осуществляемых защитных мероприятий.

Но компоненты ЗИ, с одной стороны, являются составной частью системы, с другой — сами организуют систему, осуществляя защитные мероприятия.

Поскольку система может быть определена как совокупность взаимосвязанных элементов, то назначение СЗИ состоит в том, чтобы объединить все составляющие защиты в единое целое, в котором каждый компонент, выполняя свою функцию, одновременно обеспечивает выполнение

функций другими компонентами и связан с ними логически и технологически. А в чем же состоит значимость комплексных решений в СЗИ?

Надежность защиты информации прямо пропорциональна системности, т. е. при несогласованности между собой отдельных составляющих риск «проколов» в технологии защиты увеличивается.

Во-первых, необходимость комплексных решений состоит в объединении в одно целое локальных СЗИ, при этом они должны функционировать в единой «связке». В качестве локальных СЗИ могут быть рассмотрены, например, виды защиты информации (правовая, организационная, инженерно-техническая).

Во-вторых, необходимость комплексных решений обусловлена назначением самой системы. Система должна объединить логически и технологически все составляющие защиты. Но из ее сферы выпадают вопросы полноты этих составляющих, она не учитывает всех факторов, которые оказывают или могут оказывать влияние на качество защиты. Например, система включает в себя какие-то объекты защиты, а все они включены или нет — это уже вне пределов системы.

Поэтому качество, надежность защиты зависят не только от видов составляющих системы, но и от их полноты, которая обеспечивается при учете всех факторов и обстоятельств, влияющих на защиту. Именно полнота всех составляющих системы защиты, базирующаяся на анализе таких факторов и обстоятельств, является вторым назначением комплексности.

При этом должны учитываться все параметры уязвимости информации, потенциально возможные угрозы ее безопасности, охватываться все необходимые объекты защиты, использоваться все возможные виды, методы и средства защиты и необходимые для защиты кадровые ресурсы, осуществляться все вытекающие из целей и задач защиты мероприятия.

В-третьих, только при комплексном подходе система может обеспечивать безопасность всей совокупности информации, подлежащей защите, и при любых обстоятельствах. Это означает, что должны защищаться все носители информации, во всех компонентах ее сбора, хранения, передачи и

использования, во все время и при всех режимах функционирования систем обработки информации.

В то же время комплексность не исключает, а, наоборот, предполагает дифференцированный подход к защите информации, в зависимости от состава ее носителей, видов тайны, к которым отнесена информация, степени ее конфиденциальности, средств хранения и обработки, форм и условии проявления уязвимости, каналов и методов несанкционированного доступа к информации.

Таким образом, значимость комплексного подхода к защите информации состоит:

— в интеграции локальных систем защиты;

— в обеспечении полноты всех составляющих системы защиты;

— в обеспечении всеохватности защиты информации.

Исходя из этого, можно сформулировать следующее определение:

«Комплексная система защиты информации — система, полно и всесторонне охватывающая все предметы, процессы и факторы, которые обеспечивают безопасность всей защищаемой информации»[2].

1.4. Принципы построения комплексной системы защиты информации

При построении любой системы необходимо определить принципы, в соответствии с которыми она будет построена. КСЗИ — сложная система, функционирующая, как правило, в условиях неопределенности, требующая значительных материальных затрат. Поэтому определение основных принципов КСЗИ позволит определить основные подходы к ее построению.

Принцип законности заключается в соответствии принимаемых мер законодательству РФ о защите информации, а в случае отсутствия соответствующих законов — другим государственным нормативным документам по защите.

В соответствии с принципом полноты защищаемой информации защите подлежит не только информация, составляющая государственную, коммерческую или служебную тайну, но и та часть несекретной информации,

утрата которой может нанести ущерб ее собственнику либо владельцу. Реализация этого принципа позволяет обеспечить и охрану интеллектуальной собственности.

Принцип обоснованности защиты информации заключается в установлении путем экспертной оценки целесообразности засекречивания и защиты той или другой информации, вероятных экономических и других последствий такой защиты исходя из баланса жизненно важных интересов государства, общества и граждан. Это, в свою очередь, позволяет расходовать средства на защиту только той информации, утрата или утечка которой может нанести действительный ущерб ее владельцу.

Принцип создания специализированных подразделений по защите информации заключается в том, что такие подразделения являются непременным условием организации комплексной защиты, поскольку только специализированные службы способны должным образом разрабатывать и внедрять защитные мероприятия и осуществлять контроль за их выполнением.

Принцип участия в защите информации всех соприкасающихся с нею лиц исходит из того, что защита информации является служебной обязанностью каждого лица, имеющего по роду выполняемой работы отношение к защищаемой информации, и такое участие дает возможность повысить качество защиты.

Принцип персональной ответственности за защиту информации требует, чтобы каждое лицо персонально отвечало за сохранность и неразглашение вверенной ему защищаемой информации, а за утрату или распространение такой информации оно несет уголовную, административную или иную ответственность.

Принцип наличия и использования всех необходимых правил и средств для защиты заключается в том, что КСЗИ требует, с одной стороны, участия в ней руководства предприятия и специальной службы защиты информации и всех исполнителей, работающих с защищаемой информацией, с другой стороны, использования различных организационных форм и методов защиты,

с третьей стороны, наличие необходимых материально-технических ресурсов, включая технические средства защиты.

Принцип превентивности принимаемых мер по защите информации предполагает априорное опережающее заблаговременное принятие мер по защите до начала разработки или получения информации. Из этого принципа вытекает, в частности, необходимость разработки защищенных информационных технологий.

Среди рассмотренных принципов едва ли можно выделить более, или менее важные. А при построении КСЗИ важно использовать их в совокупности.

Главная цель создания СЗИ — достижение максимальной эффективности защиты за счет одновременного использования всех необходимых ресурсов, методов и средств, исключающих несанкционированный доступ к защищаемой информации и обеспечивающих физическую сохранность ее носителей.

Организация — это совокупность элементов (людей, органов, подразделений) объединенных для достижения какой-либо цели, решения какой-либо задачи на основе разделения труда, распределения обязанностей и иерархической структуры.

СЗИ относится к системам организационно-технологического (социотехнического) типа, т. к. общую организацию защиты и решение значительной части задач осуществляют люди (организационная составляющая), а защита информации осуществляется параллельно с технологическим процессами ее обработки (технологическая составляющая).

Серьезным побудительным мотивом к проведению перспективных исследований в области защиты информации послужили те постоянно нарастающие количественные и качественные изменения в сфере информатизации, которые имели место в последнее время и которые, безусловно, должны быть учтены в концепциях защиты, информации.

Постановка задачи защиты информации в настоящее время приобретает ряд особенностей: во-первых, ставится вопрос о комплексной защите информации; во-вторых, защита информации становится все более актуальной

для массы объектов (больших и малых, государственной и негосударственной принадлежности); в-третьих, резко расширяется разнообразие подлежащей защите информации (государственная, промышленная, коммерческая, персональная и т. п.). Осуществление мероприятий по защите информации носит массовый характер, занимается этой проблемой большое количество специалистов различного профиля. Но успешное осуществление указанных мероприятий при такой их масштабности возможно только при наличии хорошего инструментария в виде методов и средств решения соответствующих задач. Разработка такого инструментария требует наличия развитых научно-методологических основ защиты информации.

Под научно-методологическими основами комплексной защиты информации (как решения любой другой проблемы) понимается совокупность принципов, подходов и методов (научно-технических направлений), необходимых и достаточных для анализа (изучения, исследования) проблемы комплексной защиты, построения оптимальных механизмов защиты и управления механизмами защиты в процессе их функционирования. Уже из приведенного определения следует, что основными компонентами научно-методологических основ являются принципы, подходы и методы. При этом под принципами понимается основное исходное положение какой-либо теории, учения, науки, мировоззрения; под подходом — совокупность приемов, способов изучения и разработки какой-либо проблемы; под методом — способ достижения какой-либо цели, решения конкретной задачи. Например, при реализации принципа разграничения доступа в качестве подхода можно выбрать моделирование, а в качестве метода реализации — построение матрицы доступа.

Общее назначение методологического базиса заключается в

— формировании обобщенного взгляда на организацию и управление КСЗИ, отражающего наиболее существенные аспекты проблемы;

— формировании полной системы принципов, следование которым обеспечивает наиболее полное решение основных задач;

— формировании совокупности методов, необходимых и достаточных для решения всей совокупности задач управления.

Предмет нашего исследования — рассмотрение различных аспектов обеспечения безопасности социотехнической системы, характерным примером которой является современный объект информатизации.

Поэтому состав научно-методологических основ можно определить следующим образом:

— так как речь идет об организации и построении КСЗИ, то общеметодологической основой будут выступать основные положения теории систем;

— так как речь идет об управлении, то в качестве научно-методической основы будут выступать общие законы кибернетики (как науки об управлении в системах любой природы);

— так как процессы управления связаны с решением большого количества разноплановых задач, то в основе Должны быть принципы и методы моделирования больших систем и процессов их функционирования.

Состав научно-методологических основ комплексной системы защиты информации представлен на рис. 2.

Рис. 2. Состав научно-методологических основ КСЗИ

2.2. Основные положения теории систем

Я считаю, что познать части без знания целого так же невозможно, как познать целое без знания его частей (Блез Паскаль 1623–1662).

Эти слова очень точно отражают суть теории систем. Но давайте по-порядку.

Начнем с определения системы.

Система — совокупность или множество связанных между собой элементов.

Под системой может пониматься естественное соединение составных частей, самостоятельно существующих в природе, а также нечто абстрактное, порожденное воображением человека. Такой подход к определению понятия системы заранее предлагает существование связей между ее элементами.

Всякая система состоит из взаимосвязанных и взаимодействующих между собой и с внешней средой частей ем в определенном смысле представляет собой замкнутое целое.

Система взаимодействует с внешней средой и может быть количественно оценена через свои входы и выходы.

Входами могут быть, в общем смысле, перерабатываемое сырье, его количество, состав, температура и т. д.; выходами могут быть количество готового продукта, его качество и т. п. (см. рис. 3).

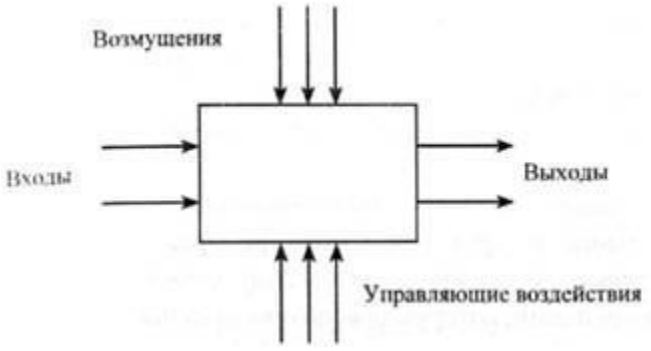

Рис. 3. Обобщенное представление системы

Обычно система подвержена возмущениям, для их компенсации, т. е. для того, чтобы система работала в заданном направлении, используют управляющие воздействия.

Система — это достаточно сложный объект, который можно расчленить (провести декомпозицию) на составляющие элементы или подсистемы. Элементы связаны друг с другом и с окружающей средой объекта. Совокупность связей образует структуру системы. Система имеет алгоритм функционирования, направленный на достижение определенной цели.

Все системы можно условно разделить на малые и большие.

Малые системы однозначно определяются свойствами процесса и обычно ограничены одним типовым процессом, его внутренними связями, а также особенностями функционирования.

Большие системы представляют собой сложную совокупность малых (подсистем) систем и отличаются от них в количественном и качественном отношениях.

Рассмотрим составляющие системы и ее основные свойства.

Элементы — это объекты, части, компоненты системы. Причем их число ограничено.

Свойства — качества элементов, дающие возможность количественного описания системы, выражая ее в определенных величинах.

Связи — это то, что соединяет элементы и свойства системы в целое.

При анализе систем значительный интерес представляет изучение их структуры. Структура отражает наиболее существенные, устойчивые связи между элементами системы и их группами, которые обеспечивают основные свойства системы. То есть структура — это форма организации системы. Структура системы может претерпевать определенные изменения в зависимости от факторов (причин) внутренней и внешней природы, от времени.

Понятие «состояние» обычно выявляют на основании исследования, ситуационного анализа, исследуя, например, входные воздействия и выходные результаты системы.

Поведение системы характеризует возможность устойчивого, контролируемого перехода системы из одного состояния в другое.

Понятие «равновесие» определяется как способность системы в отсутствие внешних воздействий сохранять заранее заданное состояние.

Устойчивость характеризуется как способность системы возвращаться в состояние равновесия после того, как она была выведена из него под влиянием внешнего воздействия. На рисунке 4 схематично показана система в устойчивом и неустойчивом состояниях. Реально устойчивость систем может достигаться только в определенных пределах.

Понятие «развитие» характеризует совершенствование структуры и функций системы под влиянием внутренних факторов, в связи с чем поведение системы приобретает более упорядоченный и предсказуемый характер.

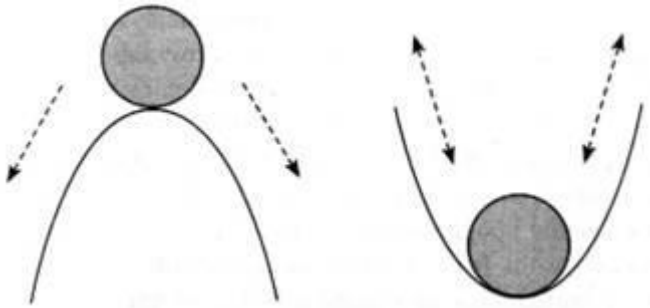

Рис. 4. Система в устойчивом состоянии (справа) и неустойчивом (слева)

Главное свойство системы в том, что она приобретает особенности, не свойственные ее элементам. Здесь можно привести множество примеров: компьютер, как система, состоящая из определенного набора деталей и программного обеспечения. И если все собрано и отлажено правильно (организована система), то получаем новые качества входящих в эту систему элементов. Это свойство называется принципом эмерджентности.

Общая теория систем — междисциплинарная область научных исследований, в задачи которой входит разработка обобщенных моделей систем, построение методологического аппарата, описание функционирования и поведения системных объектов, рассмотрение динамики систем, их поведения, развития, иерархического строения и процессов управления в

системах. Теория систем оперирует такими понятиями, как системный анализ и системный подход.

Системный анализ — это стратегия изучения сложных систем. В качестве метода исследования в нем используется математическое моделирование, а основным принципом является декомпозиция сложной системы на более простые подсистемы (принципы иерархии системы). В этом случае математическая модель строится по блочному принципу: общая модель подразделяется на блоки, которым можно дать сравнительно простые математические описания.

В основе стратегии системного анализа лежат следующие общие положения: 1) четкая формулировка цели исследования; 2) постановка задачи по реализации этой цели и определение критерия эффективности решения задачи; 3) разработка развернутого плана исследования с указанием основных этапов и направлений решения задачи; 4) последовательное продвижение по всему комплексу взаимосвязанных этапов и возможных направлений; 5) организация последовательных приближений и повторных циклов исследований на отдельных этапах; 6) принцип нисходящей иерархии анализа и восходящей иерархии синтеза в решении составных задач и т. п.

Системный анализ позволяет организовать наши знания об объекте таким образом, чтобы помочь выбрать нужную стратегию либо предсказать результаты одной или нескольких стратегий, представляющихся целесообразными для тех, кто должен принимать решение.

С позиций системного анализа решаются задачи моделирования, оптимизации, управления и оптимального проектирования систем.

Особый вклад (важность) системного анализа в решении различных проблем заключается в том, что он позволяет выявить факторы и взаимосвязи, которые впоследствии могут оказаться весьма существенными, дает возможность спланировать методику наблюдений и построить эксперимент так, чтобы эти факторы были включены в рассмотрение, освещает слабые места гипотез и допущений. Как научный подход системный анализ создает

инструментарий познания физического мира и объединяет его в систему гибкого исследования сложных явлений.

Системный подход — направление методологии научного познания и социальной практики, в основе которого лежит рассмотрение объектов как систем. Системный подход ориентирует исследование на раскрытие целостности объекта, на выявление разных личных типов связей в нем и сведения в единую теоретическую картину.

Системный подход основан на представлении о системе как о чем-то целостном, обладающем новыми свойствами (качествами) по сравнению со свойствами составляющих ее элементов. Новые свойства при этом понимаются очень широко. Они могут выражаться, в частности, в способности решать новые проблемы или достигать новые цели. Для этого требуется определить границы системы, выделив ее из окружающего мира, и затем соответствующим образом изменить (преобразовать), или, говоря математическим языком, перевести систему в желаемое состояние. Академик В. М. Глушков выделил в системном подходе следующие этапы[4]:

1. Постановка задачи (проблемы): определение Объекта исследования, постановка целей, задание критериев для изучения объекта и управления им;

2. Очерчивание границ изучаемой системы и ее (первичная) структуризация. На этом этапе вся совокупность объектов и процессов, имеющих отношение к поставленной цели, разбивается на два класса — собственно изучаемая система и внешняя среда;

3. Составление математической модели изучаемой системы: параметризация системы, задание области определения параметров, установление зависимостей между введенными параметрами;

4. Исследование построенной модели: прогноз развития изучаемой системы на основе ее модели, анализ результатов моделирования;

5. Выбор оптимального управления.

Выбор оптимального управления как раз и позволяет перевести систему в желаемое (целевое) состояние и тем самым решить проблему.

Несмотря на четкую математическую трактовку системного подхода, он не получил, однако, однозначном практической интерпретации. В связи с этим развиваются несколько направлений его практической реализации. Наибольшее распространение получили АСУПовское и системотехническое направления, суть которых заключается в совершенствовании существующих систем управления. Для этого проводится их обследование (диагностическим анализ), выявляются недостатки и пути устранения последних, формируются мероприятия по совершенствованию систем, разрабатываются проекты систем, внедрение которых рассматривается как способ преобразования существующих систем управления.

Значительную роль в этих методах играют понятие системы, подсистемы, окружающей среды, классификация основных свойств и процессов в системах, классификация систем и т. д.

Остановимся на обобщенном определении системы.

Система, с одной стороны, может быть описана динамически как процесс, а с другой — статически, с точки зрения либо внешних, либо внутренних характеристик.

Кроме того, внутреннее строение системы может быть представлено в виде функциональных зависимостей и в виде структуры, реализующей эти зависимости.

Таким образом, можно выделить пять основных системных представлений: процессуальное, функциональное, макроскопическое, иерархическое и микроскопическое.

В процессуальном плане система рассматривается динамически как процесс, остальные системные представления отражают ее статический аспект.

В макроскопическом представлении описываются внешние характеристики системы, в функциональном, иерархическом и микроскопическом — внутренние.

Микроскопическое представление системы основано на понимании ее как совокупности взаимосвязанных элементов, неразложимых далее «кирпичиков».

Центральными понятием микроскопического системного представления является понятие элемента. Конечно, в общем виде элемент лишь относительно неделим, однако для данной системы он является абсолютно неделимым. Элементы также могут быть рассмотрены как системы, но это будут системы другого типа, по отношению к исследуемой. Кроме того, система понимается как совокупность разнородных элементов, которые могут отличаться по принципу действия, техническому исполнению и ряду других характеристик. Система сводится к ансамблю простых частей.

Элементы системы обладают связями, которые объединяют их в целостную систему. Элементы могут существовать только в «связанном» виде — между элементами обязательно устанавливаются связи.

Например, в электрической цепи, если по ней не течет ток, нет электрических связей, следовательно, нет и элементов; когда цепь подключена к источнику электрической энергии, в ней образуются реальные электрические связи, и можно говорить о существовании элементов, которые они связывают.

Элементы в системе обязательно взаимодействуют, в результате одни свойства (переменные) изменяются, другие остаются неизменными (константы).

Важную роль в системных исследованиях играет поиск системообразующих связей, благодаря которым все элементы системы оказываются связанными воедино.

Функциональное представление системы связано с пониманием системы как совокупности функций (действий) Для достижения определенной цели. Каждый элемент в системе выполняет определенную функцию.

Синонимом понятия «структура» для функционального представления служит понятие функциональной структуры, или организации.

Организация может быть реализована различными структурами (при этом функциональная сущность систему остается той же, меняется только способ реализации).

Для макроскопического представления характерно понимание системы как нерасчленимого целого. Здесь важно понятие системного окружения.

Под окружающей средой системы понимается совокупность всех объектов, изменение свойств которых влияет на систему и на которые влияет изменение свойств системы. Ни одна система объектов не может быть рассмотрена вне системного окружения. Системное окружение позволяет охарактеризовать систему множеством внешних связей (или внешней структурой), так и совокупностью внешних отношений.

Иерархическое представление системы (как иерархической упорядоченности) основано на понятии подсистемы, или единицы, которые следует отличать от понятия «элемент». Единица обладает функциональной спецификой целого (системы). Система может быть представлена в виде совокупности единиц, составляющих системную иерархию. (Единица может быть разложена на элементы.)

Можно выделить два типа функциональных связей между единицами системной иерархии — горизонтальные — между единицами одного уровня и вертикальные — между единицами различных уровней. Единицы каждого уровня описываются набором вертикальных и горизонтальных связей.

Процессуальное представление системы предполагает понимание системного объекта как совокупности процессов, характеризуемых последовательностью состояний во времени. Основным понятием здесь является понятие периода жизни — временного интервала, в течение которого функционирует данный процесс.

Комплексная система защиты информации — это система организационно-технологического типа. Она характеризуется рядом признаков.

КСЗИ — это система:

— искусственная, т. е. создана человеком;

— материальная, что подразумевает не только объективность ее существования, но и тот или иной уровень материальных и финансовых затрат на реализацию;

— открытая, т. е. возможно ее расширение;

— динамическая — подвержена старению, развитию, движению, прогрессу и регрессу, делению, слиянию и т. д.;

— вероятностная — система характеризуется вероятностью структуры, функции, целей, задач, ресурсов.

***Лабораторная работа № 5*. Настройка и использование межсетевого экрана. Создание VPN- подключения средствами Windows. Сетевые протоколы для секретной передачи данных.**

***Цель работы*:** ознакомиться с возможностями по настройке и использованию межсетевого экрана и создания VPN- подключения средствами Windows.

1. Настроить брандмауэр на работу с **Веб-сервером** (HTTP), FTP-сервером и зафиксировать соответствующее окно для отчета (PrinScren).
2. Включить **журнал безопасности**.
3. После выполнения задания 1 и 2 подключиться к Интернету и посетить любой веб-сервер.
4. Просмотреть **журнал безопасности**.
5. Зафиксировать записи **журнала безопасности** для отчета. Сделать выводы.

Рекомендации по выполнению заданий практического занятия

Для подготовки докладов на семинар рекомендуются следующие темы:

1. Технологии и методы защиты информации и информационных систем в таможенных органах РФ: идентификация и аутентификация.
2. Технологии и методы защиты информации и информационных систем в таможенных органах РФ: криптография и шифрование.
3. Технологии и методы защиты информации и информационных систем в таможенных органах РФ: регистрация и аудит.
4. Технологии и методы защиты информации и информационных систем в таможенных органах РФ: межсетевое экранирование; виртуальные частные сети (VPN).

5. Основные направления и задачи обеспечения информационной безопасности таможенных органов РФ на период до 2020 года.

Для подготовки докладов формируются четыре творческие группы, каждая из которых готовит доклад по одной из перечисленных тем. По каждому докладу необходимо раскрыть содержание технологии (метода) защиты информации, его место роль в обеспечении защиты информации, характер угроз, в отношении которых применяется технология (метод) и примеры информационных систем (желательно таможенных информационных систем), в которых реализованы соответствующие технологии и методы защиты информации.

Доклад по последней теме готовит один из наиболее подготовленных студентов. Этот доклад завершает изучение дисциплины.

Все доклады должны сопровождаться презентациями.

Базовая теория по теме практического занятия и алгоритм его выполнения. Межсетевое экранирование повышает безопасность объектов внутренней сети за счет игнорирования неавторизованных запросов из внешней среды, тем самым, обеспечивая все составляющие информационной безопасности. Кроме функций разграничения доступа, экранирование обеспечивает регистрацию информационных обменов.

Функции экранирования выполняет **межсетевой экран** или брандмауэр (firewall), под которым понимают программную или программно-аппаратную систему, которая выполняет контроль информационных потоков, поступающих в информационную систему и/или выходящих из нее, и обеспечивает защиту информационной системы посредством фильтрации информации. Фильтрация информации состоит в анализе информации по совокупности критериев и принятии решения о ее приеме и/или передаче.

Брандмауэрв**Windows –** это система защиты подключения к Интернету (Internet Connection Firewall, ICF), представляет собой программу настройки ограничений, регулирующих обмен данными между Интернетом и небольшой

сетью или локальным компьютером. Брандмауэр ICF необходимо установить для любого компьютера, имеющего прямое подключение к Интернету.

При включении брандмауэра для локального компьютера, подключенного к Интернету с помощью модема удаленного доступа, брандмауэр ICF обеспечивает защиту этого подключения.

Задание:Активизировать встроенный брандмауэр операционной системыWindowsи настроить его параметры.

Все действия выполняются в режиме удаленного рабочего стола с учетной записью userXX, где XX номер.

Алгоритм выполнения работы.

А) Активизация встроенного брандмауэра.

Для активизации встроенного брандмауэра операционной системы Windows выполните следующие действия.

1. Последовательно выполните **Пуск/Панель управления /Брандмауэр Windows.**

2. В открывшемся окне выберите **Включение и отключение брандмауэр Windows (рисунок 1).**

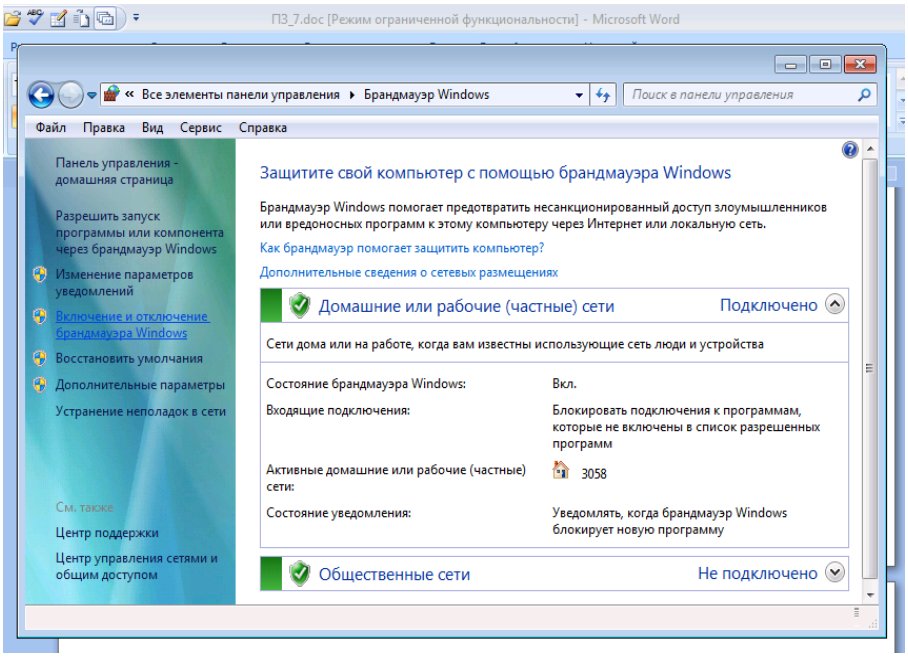

Рисунок 1 – Окно настройки брандмауэра

3. Щелкните **Включение брандмауэра Windows** под каждым сетевым размещением (рисунок 2), которое следует защитить, и нажмите кнопку **ОК**.

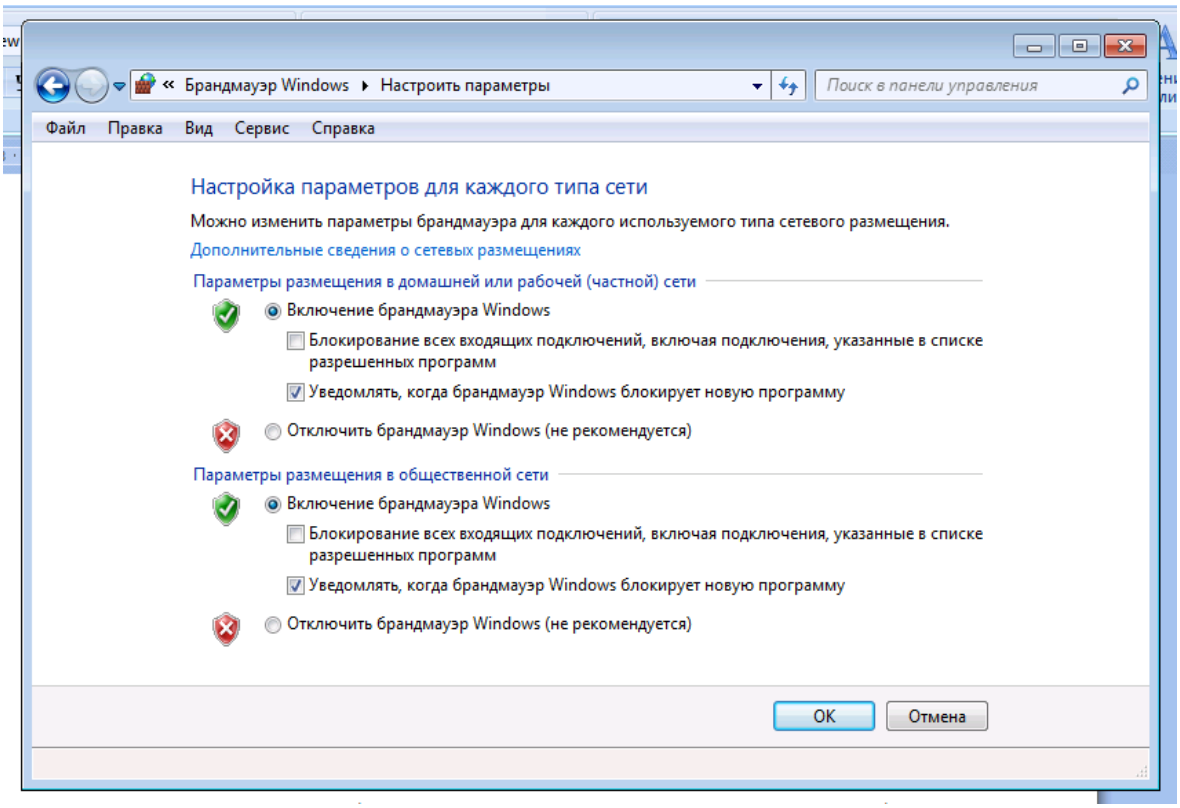

Рисунок 2 – Окно включения/отключения брандмауэра

4. Если брандмауэр должен блокировать все, включая программы, которым ранее было разрешено устанавливать связь через брандмауэр, установите флажок **Блокирование всех входящих подключений**, включая подключения, указанные в списке разрешенных программ.

В) Настройка параметров брандмауэра.

Если требуется настроить разрешения для конкретной программы или компонента ОС, щелкните **Разрешить запуск программы или компонента через брандмауэр Windows** в левой панели.

1. В открывшемся окне просмотрите разрешенные для соединения программы;

2. Для добавления новой программы в открывшемся окне нажмите **Разрешить другую программу** и выберите любую программу.

3. Нажмите **Ок** и убедитесь, что программа появилась в списке;

Для настройки брандмауэра в режиме расширенной безопасности выполните следующие действия.

4. Последовательно выполните **Пуск/Панель управления/Брандмауэр Windows.**

5. В открывшемся окне выберите **Дополнительные параметры (рисунок 3).**

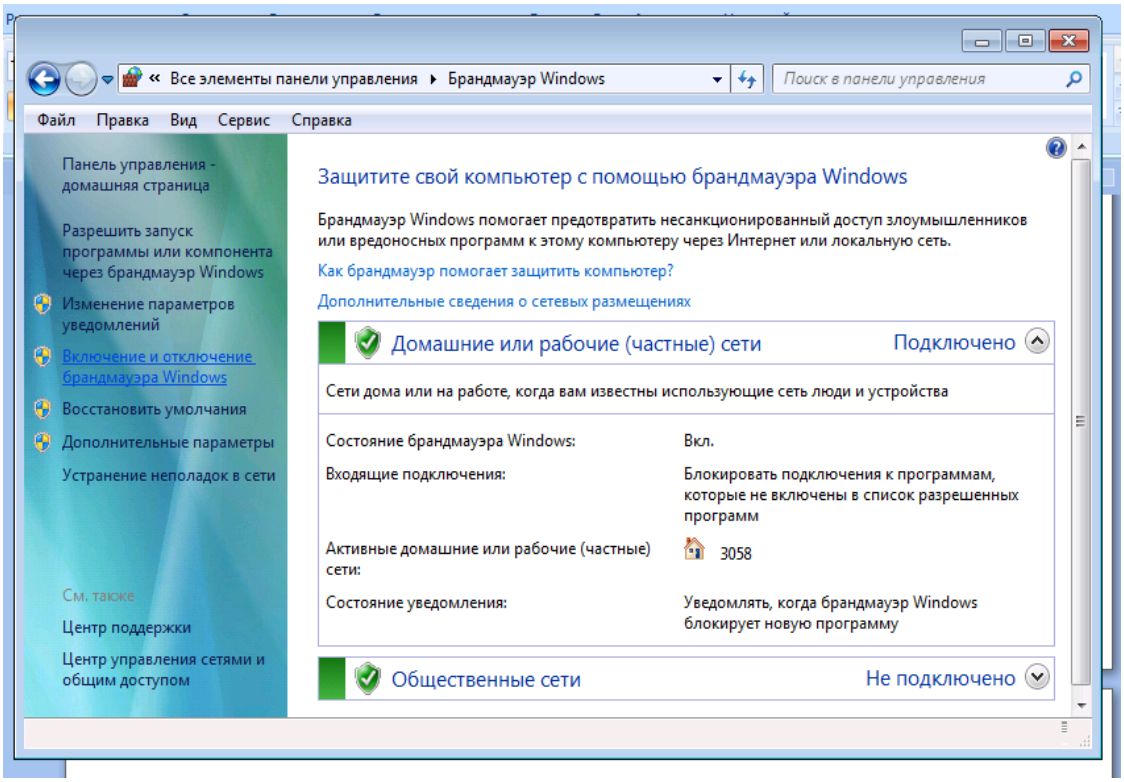

Рисунок 3 – Окно выбора дополнительных параметров брандмауэра

6. В результате откроется окно **Брандмауэр Windows в режиме повышенной безопасности** (рисунок 4) с тремя полями.

7. Изучите все закладки открытого окна, найдите закладку **Правила безопасности** и изучите порядок создания правил.

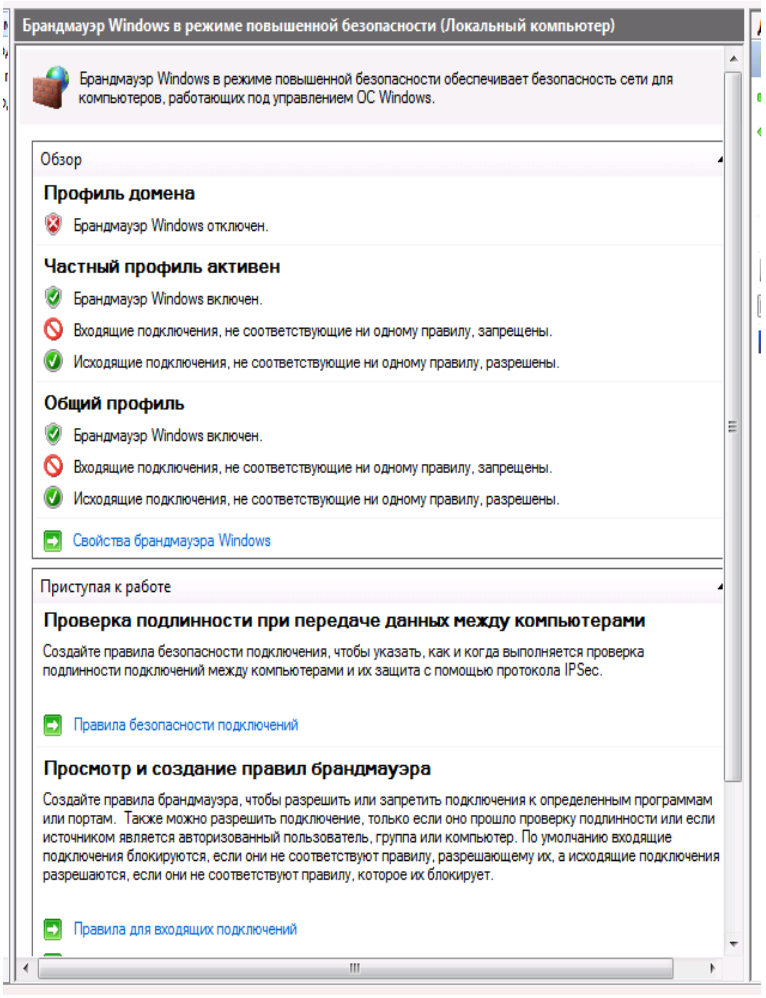

Рисунок 4 – Окно брандмауэра в режиме повышенной безопасности

Создание vpn-подключения средствами Windows 2000 (xp)

Технология виртуальных частных сетей (VPN – Virtual Private Network) является одним из эффективных механизмов обеспечения информационной безопасности при передаче данных в распределенных вычислительных сетях.

Виртуальные частные сети являются комбинацией нескольких самостоятельных сервисов (механизмов) безопасности: шифрования, экранирования и туннелирования.

Задание: создать VPN-подключение и выполнить его настройку.

Алгоритм выполнения работы

А. Создание VPN-подключения.

1. Откройте компонент **Сетевые подключения**. Для этого выберите последовательно **Пуск** – **Панель управления** – **Сетевыеподключения**.

2. Выберите пункт **Создание нового подключения** и нажмите кнопку **Далее**.

3. В зависимости от операционной системы выполните следующие действия:

- для Windows XP – в открывшемся окне выберите пункт **Подключить к сети на рабочем месте** (рис. 5, только для XP) и нажмите **Далее**. После этого выберите **Подключение к виртуальной частной сети** (рис. 6) и нажмите **Далее**.
- для Windows 2000 – в открывшемся окне выберите пункт **Подключение к виртуальной частной сети через Интернет** и нажмите **Далее**.

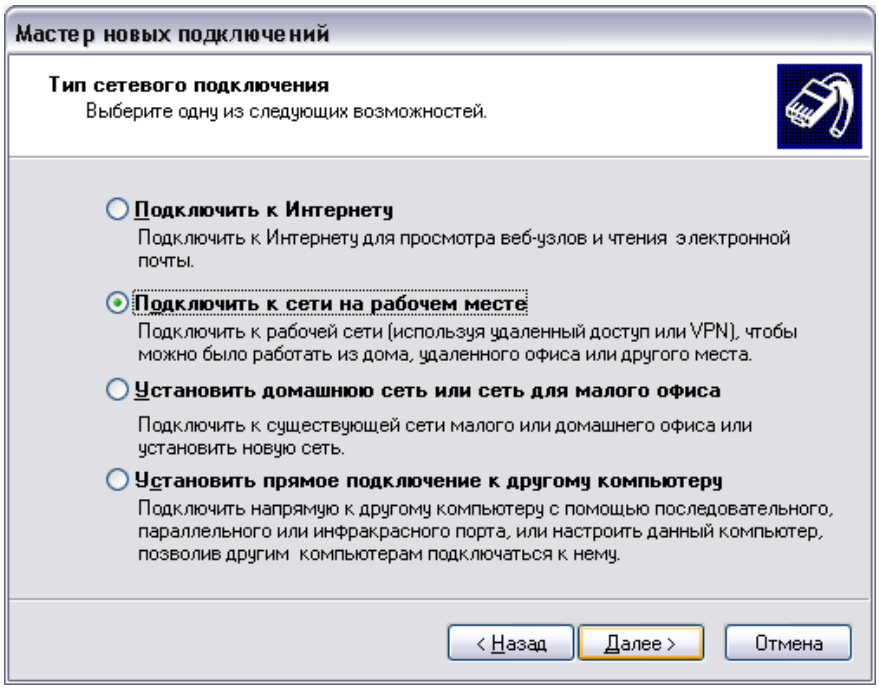

Рис. 5

4. Введите имя подключения и перейдите к следующему шагу командой **Далее**.

5. Если перед установкой «туннельного доступа» требуется подключение к провайдеру услуг Интернета, то выберите (рис. 7).

Набрать номер для следующего предварительного подключения и, выбрав нужное подключение, нажмите **Далее**. В противном случае, выберите **Не набирать номер для предварительного подключения** и нажмите **Далее**.

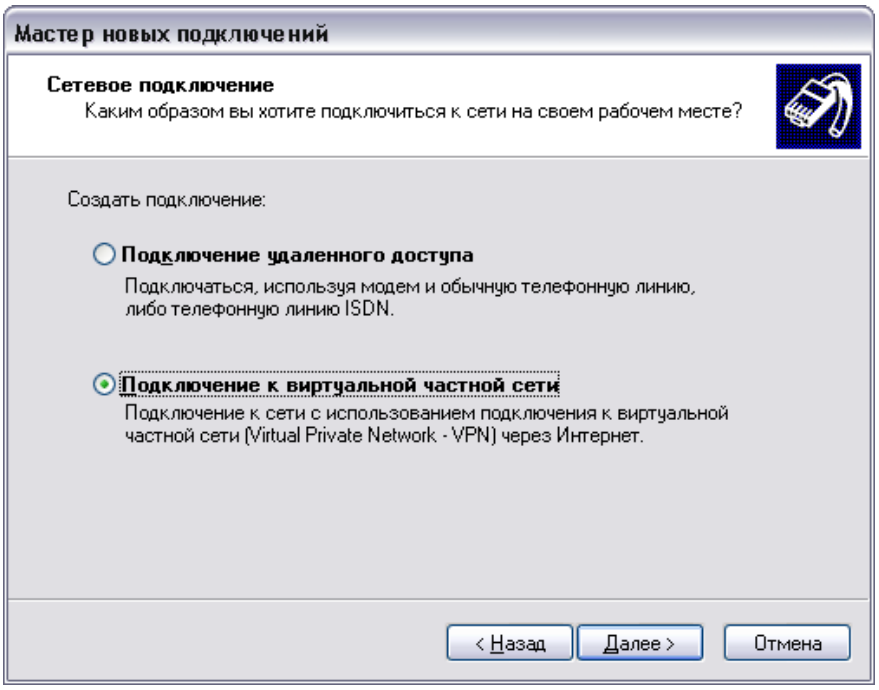

Рис. 6

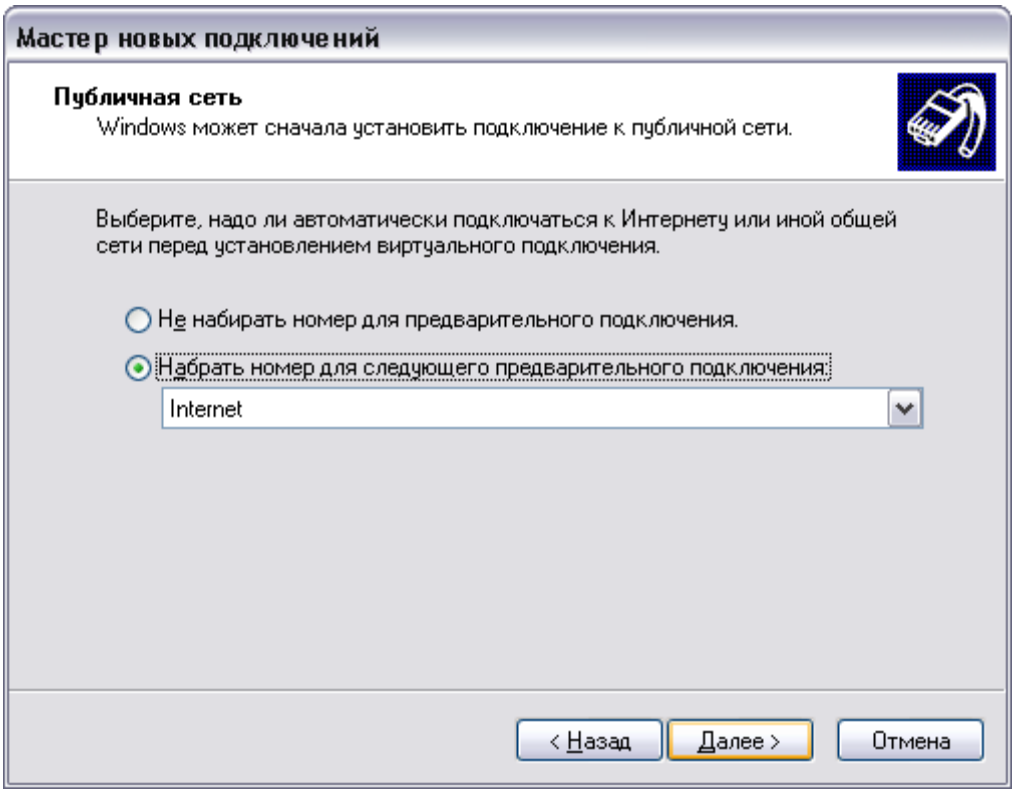

Рис.7

6. Введите имя узла (сети) или его IP-адрес (например 122.122.122.122), к которому идет подключение (рис.8).

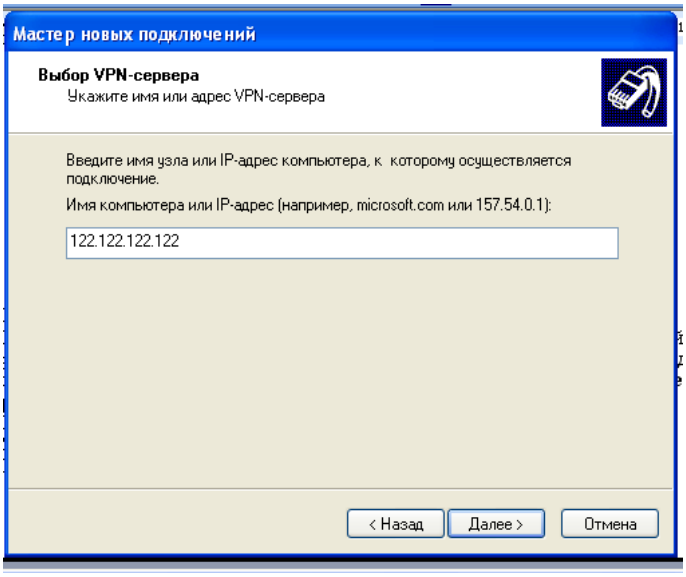

Рис. 8

7. Завершите работу Мастера сетевых подключений.

8. В результате в папке Подключения появится новое подключение (рис. 9).

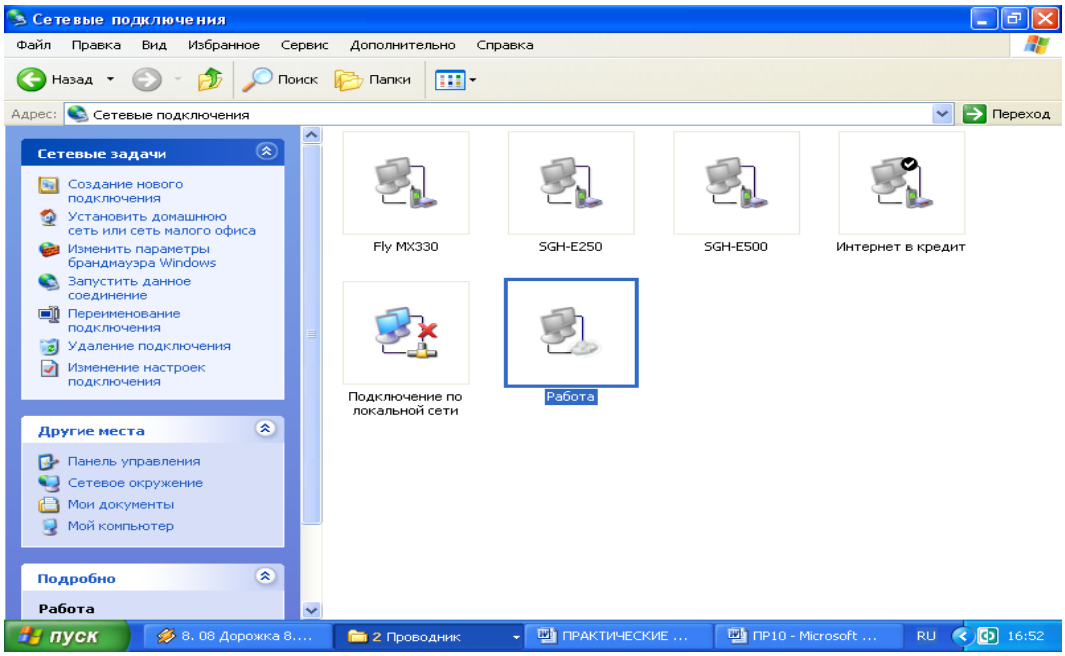

Рис.9

9. Для настройки параметров подключения выделите подключение VPN и вызовите его свойства из контекстного меню (нажатие правой клавиши мыши).

10. Рассмотрите все имеющиеся параметры VPN-подключения и при необходимости воспользуйтесь соответствующими разделами справки.

Задания для самостоятельной работы

Создайте VPN-подключение к узлу с адресом 122.122.122.122 и зафиксируйте окно его свойств (Print Screen) на закладке Общие (как показано на рис. 5) в качестве отчета.

Контрольные вопросы

1. Какие механизмы безопасности используются при реализации VPN-подключения?
2. Что такое «туннель» и в чем состоит принцип «туннелирования»?
3. В чем заключаются защитные функции виртуальных частных сетей?