А.Ю. Щеглов

ЗАЩИТА КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Анализ защищенности современных операционных систем. Особенности и недостатки встроенных систем защиты ОС Windows и ОС Unix.

Методология проектирования и принципы построения эффективной системы защиты от НСД.

Модели и механизмы управления доступом к ресурсам. Обеспечение полноты и корректности разграничения доступа.

Разработка добавочной системы защиты. Подробное описание и обоснование добавочных механизмов для ОС Windows и ОС Unix.

Основы теории надежности средств защиты информации. Задачи и принципы резервирования механизмов защиты.

Технологий программно-аппаратной защиты. Принципы комплексирования средств защиты.

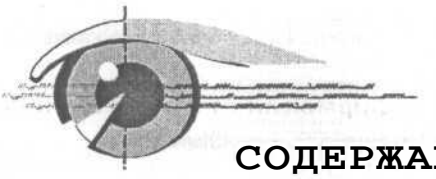
Профи

НиТ
Издательство

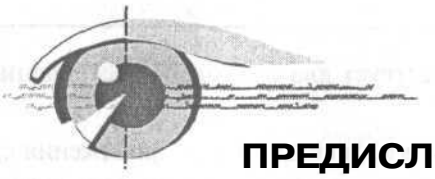
А.Ю. Щеглов

ЗАЩИТА
КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ
ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Наука и Техника, Санкт-Петербург
2004



СОДЕРЖАНИЕ



ПРЕДИСЛОВИЕ

Современными тенденциями развития информационных технологий является ярко выраженный переход в сторону создания корпоративных информационных систем. При этом основной характеристикой этих систем является разграничение доступа сотрудникам корпорации к информационным и иным ресурсам вычислительной системы. Причем данные тенденции проявляются практически для всех уровней иерархии современных информационных технологий, начиная с архитектурного уровня в целом (Internet и Intranet), включая сетевые технологии (например, IP v.4.0 и IP Sec), и заканчивая уровнем общесистемных средств (ОС, СУБД) и приложений.

Масштабы применения и приложения информационных технологий стали таковы, что наряду с проблемами производительности, надежности и устойчивости функционирования информационных систем, остро встает проблема защиты циркулирующей в системах информации от Несанкционированного доступа. Статистика фактов несанкционированного доступа к информации (НСД) показывает, что большинство современных информационных систем достаточно уязвимо с точки зрения безопасности.

- Средняя стоимость финансовых потерь компании от НСД составляет от $5 тыс. до $12 тыс. (CSI, 2002 г.).
- Общий ущерб, который понесли компании в 2002 году от НСД, оценивается в $24 млрд. (Forrest Research, 2002 г.).
- В 80% утечка информации из компании происходит непосредственно по вине ее сотрудников (IDC, 2002 г.).

И это несмотря на устойчивую тенденцию к усилению встроенных в них механизмов защиты.

Другой аспект приведенной статистики — это **возможность локализации угроз корпоративной информации**, так как большая их часть связана с угрозой НСД, исходящей от самих сотрудников компании. При этом следует учитывать и сетевые ресурсы (прежде всего в составе ЛВС), к которым сотрудник имеет доступ со своего компьютера в рамках своей служебной деятельности.

В связи с этим именно компьютер (особенно находящийся в составе сети) следует в первую очередь рассматривать в качестве объекта защиты, а конечного пользователя -- в качестве ее наиболее вероятного потенциального нарушителя. Как следствие, **под сомнение ставится обоснованность концепции реализованной системы защиты в современных универсальных ОС.** Эта система защиты заключается в построении распределенной схемы администрирования механизмов защиты, элементами которой, помимо администратора, выступают пользователи, имеющие возможность назначать и изменять права доступа к создаваемым ими файловым объектам.

На практике сегодня существует два подхода к обеспечению компьютерной безопасности:

1. Использование только встроенных в ОС и приложения средств защиты.
2. Применение, наряду со встроенными, дополнительных механизмов защиты. Этот подход заключается в использовании так называемых технических средств добавочной защиты -- программных, либо программно-аппаратных комплексов, устанавливаемых на защищаемые объекты.

Существующая статистика ошибок, обнаруженных в ОС, а также сведения о недостаточной эффективности встроенных в ОС и приложения механизмов защиты, заставляет специалистов сомневаться в достижении гарантированной защиты от НСД, при использовании встроенных механизмов, и все большее внимание уделять средствам добавочной защиты информации.

Данная книга представляет собою попытку системного изложения проблем защиты компьютерной информации, а также теоретических основ применения добавочных средств защиты компьютерной информации.

Книга посвящена рассмотрению подходов к построению встроенных средств защиты современных ОС и приложений, выявлению причин их уязвимости на основе существующей статистики угроз. На основании этого формулируются и теоретически обосновываются общие требования к механизмам защиты, в том числе добавочной. Также в книге рассматриваются вопросы построения и проектирования средств защиты компьютерной информации, определяются цеди их применения и решаемые задачи. Излагаются подходы к построению добавочных средств (методы комплексирования встроенных и добавочных механизмов защиты), с учетом изменяемого при их включении в систему поля угроз информационной безопасности.

В книге приведено исследование системных и прикладных вопросов защиты компьютерной информации от НСД. Вопросы, относящиеся к применению механизмов криптографической защиты, остались за рамками настоящей работы.

При подготовке книги использованы полученные в последнее время результаты отечественных и зарубежных специалистов в области проектирования и построения систем защиты компьютерной информации. Однако в первую очередь обсуждаются теоретические и практические результаты проведенных исследований и проектирования механизмов защиты от НСД, а также учтен опыт, приобретенный автором при разработке добавочных средств защиты в рамках работы НПП «Информационные технологии в бизнесе» (http://www.npp-itb.spb.ru). Так, в книге излагаются новые технологии обеспечения компьютерной безопасности и технические решения по реализации механизмов добавочной защиты, апробированные при разработке комплексной системы защиты информации (КСЗИ) «Панцирь» для ОС Windows 9x/NT/2000 [33], HP-UX, Linux, Free BSD. Здесь же стоит отметить, что большинство описываемых технологий запатентовано [24...32].

Целью книги является не только рассмотрение перспективных технологий и методов защиты информации от НСД, в том числе добавочной, но и обоснование требований к системе защиты. Эти требования позволят потребителю ориентироваться на рынке средств информационной защиты при выборе оптимального решения. Кроме того, немалое внимание уделено иллюстрации тех задач, которые должны решаться администратором безопасности, эксплуатирующим средства защиты. При этом следует понимать, что хорошая система защиты — это своего рода «конструктор», в котором заложены механизмы противодействия как явным, так и скрытым угрозам информационной безопасности. Чтобы достигнуть необходимого уровня безопасности защищаемых объектов, администратор должен не только знать и понимать возможности механизмов защиты, но и умело их настраивать, применительно к непрерывно изменяющейся статистике угроз.

Очевидно, что попытка задания типовых настроек механизмов защиты разработчиком с целью снижения требований к квалификации администратора безопасности — это не только бессмысленный, но и чрезвычайно вредный подход, приводящий к урезанию возможностей средств защиты и к появлению ложной уверенности потребителя в достижении им положительного эффекта. Единственно правильным подходом является повышение квалификации сотрудников, эксплуатирующих средства защиты. При этом необходимо понимать, что процесс защиты информации непрерывен, равно как непрерывен процесс изменения статистики угроз.

Таким образом, **важнейшим условием защищенности компьютерной информации является квалификация администраторов безопасности и сотрудников эксплуатирующих служб,** которая, по крайней мере, не должна уступать квалификации злоумышленников. В противном случае не помогут никакие средства защиты (то же, кстати говоря, относится и к разработчикам средств защиты и защищенных информационных систем).

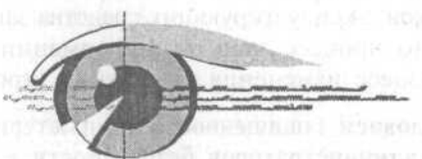
Автор надеется, что книга будет полезна научным и инженерно-техническим работникам, занимающимся исследованиями в области защиты информации и разрабатывающим, как собственно средства защиты информации (в том числе добавочные), так и информационные системы в защищенном исполнении. Кроме того, определенный интерес книга представляет для сотрудников предприятий, призванных решать задачи обеспечения информационной безопасности.

Материалы книги могут оказаться полезными для изучения аспирантами и студентами ВУЗов, ориентированных на подготовку специалистов в области различных приложений информационных технологий, в первую очередь, в области защиты информации.

Автор с благодарностью воспримет отзывы, пожелания и предложения. По возможности ответит на все вопросы, которые могут направляться по адресу: info@npp-itb.spb.ru.

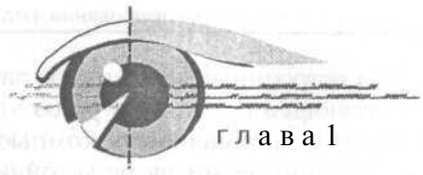
Компьютерная безопасность:

современные требования, подходы, статистика угроз



Часть I

- Требования к защите компьютерной информации
- Анализ защищенности современных операционных систем
- Подходы к проектированию системы защиты



глава 1

Требования к защите компьютерной информации

1.1. Общие положения

Вопросы защиты от НСД и методика их рассмотрения в книге

Естественным ходом развития информационных технологий явился принципиальный переход от открытости к защищенности при построении информационных систем. На сегодняшний день большинство программных продуктов, применяющихся для построения информационных систем, обладают встроенными средствами защиты. Это касается не только ОС, но СУБД и других приложений. Например, взамен протокола IP v4.0, изначально созданного для реализации единого открытого сетевого пространства, предлагается версия протокола IPSec, содержащая развитые возможности обеспечения информационной безопасности. Таким образом, наблюдается общая тенденция усиления роли механизмов защиты в современных информационных и сетевых технологиях.

Данную тенденцию наглядно иллюстрирует развитие ОС MS Windows. Можно четко проследить развитие встроенных в ОС механизмов защиты от Windows 3.1 (где механизмы защиты практически отсутствовали), к Windows NT (где механизмы защиты интегрированы в ядро ОС) и к Windows 2000 (где интеграция захватывает и внешние по отношению к разработчикам технологии защиты, такие как Kerberos, IP-тунелирование и т.д.). То же самое можно сказать и о других семействах ОС. Например, в ОС FreeBSD в каждой из новых версий появляются новые механизмы защиты (firewall, nat). Такое же развитие средств защиты касается и прикладного программного обеспечения (ПО). В СУБД (Oracle) развитие защитных механизмов выражается в шифровании трафика, усложнении авторизации, добавлении разграничения доступа к элементам таблиц и т.п.

С учетом сказанного возникает ряд вопросов:

1. Достаточно ли встроенных в современные ОС и приложения механизмов защиты для обеспечения гарантированной защиты информации от НСД?

2. В предположении, что встроенных механизмов защиты недостаточно (а с учетом существующей статистики угроз это именно так), то чем это вызвано? Почему защищенность компьютерной информации остается недостаточной, несмотря на устойчивую тенденцию к усилению встроенных в современные универсальные ОС и приложения механизмов защиты? Каким образом следует усиливать встроенные механизмы добавочными средствами защиты?
3. Какие функции должны обеспечивать и какими характеристиками должны обладать системы встроенной и добавочной защиты, чтобы обеспечить надежное противодействие попыткам НСД?
4. В предположении, что добавочные механизмы защиты необходимы, каким образом комплексировать (взаимосвязывать) в защищаемой вычислительной системе встроенные и добавочные механизмы защиты?

Используемые в настоящее время на практике (а, естественно, именно это нас и будет интересовать) подходы к защите компьютерной информации определяются следующим характеристиками:

* формализованными требованиями к набору и параметрам механизмов защиты, регламентирующими современные требования к обеспечению компьютерной безопасности (требованиями, определяющими что должно быть);
» реальными механизмами защиты, реализуемыми при защите компьютерной информации. К таковым относятся прежде всего средства защиты ОС, т.к. большинство приложений используют встроенные в ОС механизмы защиты (определяющими, что есть);
« существующей статистикой угроз компьютерной безопасности -- существующими успешными атаками на информационные компьютерные ресурсы (определяющими, насколько эффективно то, что есть и дающими оценку достаточности требований к тому, что должно быть).

С учетом сказанного построим свое изложение и анализ существующих подходов к защите компьютерной информации следующим образом. Для начала рассмотрим формализованные требования, то есть требования соответствующих нормативных документов, регламентирующих требования к защите компьютерной информации от несанкционированного доступа. Далее рассмотрим механизмы защиты, реализованные в современных ОС и проанализируем, в какой мере ими выполняются требования соответствующих нормативных документов.

Такой анализ необходим, чтобы определиться с причиной низкой защищенности компьютерной информации. Если встроенные в современные ОС механизмы защиты в полной мере соответствуют формализованным требованиям к ним, то, следовательно, эти требования необходимо усиливать. В противном случае необходимо усиливать механизмы защиты с целью выполнения соответствующих требований.

Следующим шагом будет рассмотрение и анализ существующей статистики угроз компьютерной информации и определение причины уязвимости современных ОС. На основе этого, а также на основе нормативных документов мы определим дополнительные требования к защите компьютерной информации, которые не выполняются встроенными в ОС механизмами защиты.

Затем рассмотрим непосредственно методы и механизмы, которые должны быть реализованы средствами добавочной защиты в дополнение к встроенным в ОС защитным механизмам. Особое внимание уделим методам и механизмам добавочной защиты, позволяющим функционально расширять встроенные механизмы защиты в предположении возможности потенциальных (еще не известных из опубликованной статистики) угроз. Также отдельно будут рассмотрены архитектурные и технические решения по реализации добавочной защиты.

Классификация требований к системам защиты

В общем случае следует говорить о необходимости учета двух (дополняющих друг друга) групп требований к системе защиты.

Первая группа требований (необходимые требования) заключается в необходимости реализации системой защиты формализованных мер безопасности (то есть мер, заданных соответствующими нормативными документами в области защиты информации).

Формализованные требования к необходимым механизмам защиты информационных систем, в части их защиты от НСД, сформулированы в руководящих документах ГОСТЕХКОМИССИИ РОССИИ (для нашей страны). Для других стран эти требования сформулированы в документах соответствующих организаций, например:

* *«Оранжевая книга»* КРИТЕРИИ ОЦЕНКИ НАДЕЖНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ Министерства обороны США.
» Согласованные критерии оценки безопасности информационных технологий, Information Technology Security Evaluation Criteria, ITSEC, Европейские страны.

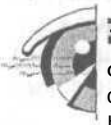
При этом отметим, что данные документы носят общий характер. В них не в полной мере предусматривается классификация объектов, для которых должна быть реализована защита. Да наверное, это и невозможно в рамках одного документа. В частности, эти документы предъявляют единые требования для всех семейств ОС. И это несмотря на то, что ОС различных семейств имеют принципиально отличные принципы построения, а значит, для них различаются и способы НСД.

В указанных руководящих документах не дается рекомендаций по способам построения и администрирования защищенных систем, то есть не сказано, как их строить. В этих документах лишь сформулированы тре-

бования (что должно быть реализовано) к механизмам защиты информации и, отчасти, требования к их количественным характеристикам.

Данный подход к заданию формализованных требований, наверное, в целом оправдан, т.к. невозможно учесть в нормативных документах все тонкости построения и проектирования средств защиты сложных информационных систем, особенно при существующей динамике их развития.

Далее в работе по понятным причинам будут рассматриваться формализованные требования, принятые в нашей стране на момент написания книги.



Примечание

Формализованные требования носят основополагающий характер — в том смысле, что в них формализованы требования к основополагающим механизмам защиты информации. Поэтому их возможное со временем изменение не может быть сколько-нибудь существенным без появления новых научно обоснованных технологий защиты информации. Само же изменение неизбежно и естественно ввиду меняющейся со временем статистики угроз информационной безопасности.

Вторая группа требований (дополнительные требования) заключается в необходимости учета существующей (текущей) статистики угроз для конкретного типа защищаемого объекта, а также потенциально возможных угроз (на данный момент отсутствующих в опубликованных угрозах, но гипотетически возможных). Необходимость этой группы требований обусловлена тем, что формализованные требования не могут учитывать все возможные угрозы объектам всех типов, требующих защиты. Также формализованные требования не могут соперничать по скорости обновления со скоростью изменения статистики угроз.

С учетом сказанного может быть сделан вывод о целесообразности рассмотрения условий необходимости и достаточности требований к защите информации. Необходимыми являются формализованные требования, определяемые соответствующими нормативными документами в области защиты информации. Достаточной является совокупность формализованных и дополнительных требований, формулируемых на основе анализа текущей статистики угроз защищаемому объекту, а также потенциально возможных угроз (на данный момент отсутствующих в опубликованных угрозах, но гипотетически возможных).

1.2. Формализованные требования к защите информации от НСД. Общие подходы к построению систем защиты компьютерной информации

1.2.1. Формализованные требования к защите и их классификация

Как было сказано ранее, общие подходы в системах защиты целесообразно рассматривать относительно соответствия их принятым формализованным требованиям. Эти требования предъявляются к механизмам защиты, которые в той или иной мере реализуются современными ОС, приложениями и добавочными средствами защиты.

Защиту информации от НСД в нашей стране на сегодняшний день регламентируют нормативные документы [1, 2] ГОСТЕХКОМИССИИ РОССИИ:

- **Требования к защите средств вычислительной техники** (СВТ) [1], формализуют условия защищенности отдельно взятого средства — ОС, СУБД, приложения.
- **Требования к защите автоматизированных систем** (АС) [2], формализуют условия защищенности объекта с учетом:
  - совокупности механизмов защиты, реализуемых установленными на защищаемом объекте средствами, включая ОС, СУБД (если есть), приложениями, добавочными механизмами защиты (если есть);
  - дополнительных организационных мер, принимаемых для безопасного функционирования АС.

При этом отметим, что, как правило, приложения используют механизмы защиты ОС. Что касается СУБД, то механизмы защиты СУБД дополняют защитные механизмы ОС, так как возникает необходимость защиты дополнительных объектов доступа — «таблиц». И действительно, для ОС защищаемыми файловыми объектами являются: логические диски (тома), каталоги, файлы. При работе с базами данных сложность обусловлена тем, что таблицы различных пользователей, к которым должен разграничиваться доступ, могут находиться в одном файле. Таким образом, получается, что в общем случае невозможно управлять доступом к базам данных только механизмами защиты ОС. Однако это относится только к СУБД. Остальные приложения обычно довольствуются защитными механизмами ОС. То есть, можно сказать, что, **как правило, все механизмы защиты автоматизированных систем** (АС) **по умолчанию реализуются собственно ОС.**

В общем случае формализованные требования к обеспечению защиты компьютерной информации от НСД (т.е. к средствам защиты ОС) зада-

ются формализованными требованиями к защите средств вычислительной техники (СВТ) [1J. Однако, в связи с вышесказанным, будем для оценки эффективности защитных механизмов ОС также использовать формализованные требования из документа [2], применяемые к автоматизированным системам (АС). Это следует делать, потому что именно защитные механизмы ОС призваны обеспечивать необходимый уровень защиты автоматизированной системы (АС) в целом.

Примечание

Противоречия здесь нет, т.к. требования к СВТ (к которым относится ОС) и АС формально зависимы по соответствующим классам защиты. Однако требования к АС несколько расширены по сравнению с требованиями к СВТ. Требования же к СВТ в большей части детализированы.

Аналогичные рассуждения могут быть проведены и относительно средства добавочной защиты. При этом, во-первых, оно может рассматриваться как отдельное СВТ, выполняющее формализованные требования документа [1]. В этом случае встроенные в ОС механизмы защиты не должны рассматриваться. А во-вторых, оно может рассматриваться как средство в составе АС (наряду с механизмами ОС, дополняя их). В этом случае встроенные в ОС и добавочные механизмы защиты должны в совокупности выполнять формализованные требования документа [2]. Кроме того, для определенных классов защиты встроенные механизмы защиты должны быть сертифицированы.

Рассмотрим формализованные требования к защите компьютерной информации АС в соответствии с документом [2]. При этом будем рассматривать первую группу АС (в соответствии с используемой в [2] классификацией), как включающую в себя наиболее распространенные многопользовательские АС, в которых одновременно обрабатывается и/или хранится информация разных уровней конфиденциальности. Причем не все пользователи имеют право доступа ко всей информации АС.

Первая группа АС содержит пять классов — 1Д, 1Г, 1В, 1Б и 1А. Деление АС на соответствующие классы по условиям их функционирования с точки зрения защиты информации необходимо в целях разработки и применения обоснованных мер по достижению требуемого уровня защиты. Дифференциация подхода к выбору методов и средств защиты определяется важностью обрабатываемой информации, а также различием АС по своему составу, структуре, способам обработки информации, количественному и качественному составу пользователей и обслуживающего персонала [2].

Требования к АС первой группы сведены в табл. 1.1. При этом используются следующие обозначения:

« — » ..... нет требований к данному классу;

« + » ..... есть требования к данному классу.

формализованные требования к защите информации от НСД *Таблица 1.1*

Подсистемы и требования	Классы				
	1Д	1Г	1В	1Б	1А
1. Подсистема управления доступом					
1.1. Идентификация, проверка подлинности и контроль доступа субъектов:					
• в систему	+	+	+	+	+
• к терминалам, ЭВМ, узлам сети *ЭВМ,* каналам связи, внешним устройствам ЭВМ	–	+	+	+	+
• к программам	–	+	+	+	+
• к томам, каталогам, файлам, записям, полям записей	–	+	+	+	+
1.2. Управление потоками информации	–	–	+	+	+
2. Подсистема регистрации и учета					
2.1. Регистрация и учет:					
• входа (выхода) субъектов доступа в (из) систему (узел сети)	+	+	+	+	+
« выдачи печатных (графических) выходных документов	–	+	+	+	+
• запуска (завершения) программ и процессов (заданий, задач)	–	+	+	+	+
• доступа программ субъектов доступа к защищаемым файлам, включая их создание, удаление, передачу по линиям и каналам связи	–	+	+	+	+
• доступа программ субъектов доступа к терминалам, ЭВМ , узлам сети ЭВМ, программам, томам, каталогам, файлам, записям, полям записей	–	+	+	+	+
• изменения полномочий субъектов доступа	–	–	+	+	+
• создаваемых защищаемых объектов доступа	–	–	+	+	+
2.2. Учет носителей информации	+	+	+	+	+
2.3. Очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей	–	+	+	+	+
2.4. Сигнализация попыток нарушения защиты	–	–	4	+	+
3. Криптографическая подсистема					
3.1. Шифрование конфиденциальной информации	–	–	–	+	+
3.2. Шифрование информации, принадлежащей различным субъектам доступа (группам субъектов) на разных ключах	–	–	–	–	+
3.3. Использование аттестованных (сертифицированных) криптографических средств	–	–	–	+	+
4. Подсистема обеспечения целостности					
4.1. Обеспечение целостности программных средств и обрабатываемой информации	+	+	+	+	+
4.2. Физическая охрана средств вычислительной техники и носителей информации	+	*	+	+	+
4.3. Наличие администратора (службы) защиты информации в АС	–	–	+	+	+
4.4. Периодическое тестирование СЗИ НСД	+	+	+	+	+
4.5. Наличие средств восстановления СЗИ НСД	+	+	+	+	+
4.6. Использование сертифицированных средств защиты	–	–	+	+	+

Как видим, рассматриваемыми требованиями выделяются следующие основные группы механизмов защиты:

» механизмы управления доступом;
• механизмы регистрации и учета;
* механизмы криптографической защиты;
* механизмы контроля целостности.

Отметим, что первая группа «Подсистема управления доступом» является основополагающей для реализации защиты от НСД, т.к. именно механизмы защиты данной группы призваны непосредственно противодействовать несанкционированному доступу к компьютерной информации.

Остальные же группы механизмов реализуются в предположении, что механизмы защиты первой группы могут быть преодолены злоумышленником. В частности они могут использоваться:

* для контроля действий пользователя — группа «Подсистема регистрации и учета»;
* для противодействия возможности прочтения похищенной информации (например, значений паролей и данных) — группа «Криптографическая подсистема»;
* для контроля осуществленных злоумышленником изменений защищаемых объектов (исполняемых файлов и файлов данных) при осуществлении к ним НСД и для восстановления защищаемой информации из резервных копий • • группа «Подсистема обеспечения целостности».

Кроме того, эти группы механизмов могут использоваться для проведения расследования по факту НСД.

Рассмотрим более подробно требования различных групп (согласно [2]), а также соответсвующие им основные подходы к защите компьютерной информации, реализуемые на сегодняшний день на практике. При этом имеет смысл остановиться лишь на двух классах:

* 1Г, задающим необходимые (минимальные) требования для обработки конфиденциальной информации;
* 1В, задающим необходимые (минимальные) требования для обработки информации, являющейся собственностью государства и отнесенной к категории секретной.

1.2.2. Требования к защите конфиденциальной информации

Подсистема управления доступом должна удовлетворять следующим требованиям:

1. Идентифицировать и проверять подлинность субъектов доступа при входе в систему. Причем это должно осуществляться по идентификатору (коду) и паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов.
2. Идентифицировать терминалы, ЭВМ, узлы компьютерной сети, каналы связи, внешние устройства ЭВМ по их логическим адресам (номерам).
3. По именам идентифицировать программы, тома, каталоги, файлы, записи и поля записей.
4. Осуществлять контроль доступа субъектов к защищаемым ресурсам в соответствии с матрицей доступа.

Подсистема регистрации и учета должна:

1. Регистрировать вход (выход) субъектов доступа в систему (из системы), либо регистрировать загрузку и инициализацию операционной системы и ее программного останова. При этом в параметрах регистрации указываются:
   - дата и время входа (выхода) субъекта доступа в систему (из системы) или загрузки (останова) системы;
   - результат попытки входа — успешная или неуспешная (при НСД);
   - идентификатор (код или фамилия) субъекта, предъявленный при попытке доступа;
   - код или пароль, предъявленный при неуспешной попытке.

Регистрация выхода из системы или останова не проводится в моменты аппаратурного отключения АС.

2. Регистрировать выдачу печатных (графических) документов на «твердую» копию. При этом в параметрах регистрации указываются:
   - дата и время выдачи (обращения к подсистеме вывода);
   - краткое содержание документа (наименование, вид, код, шифр) и уровень его конфиденциальности;
   - спецификация устройства выдачи (логическое имя (номер) внешнего устройства);
   - идентификатор субъекта доступа, запросившего документ.
3. Регистрировать запуск (завершение) программ и процессов (заданий, задач), предназначенных для обработки защищаемых файлов. При этом в параметрах регистрации указывается:
   - дата и время запуска;
   - имя (идентификатор) программы (процесса, задания);
   - идентификатор субъекта доступа, запросившего программу (процесс, задание);
   - результат запуска (успешный, неуспешный — несанкционированный).
4. Регистрировать попытки доступа программных средств (программ, процессов, задач, заданий) к защищаемым файлам. В параметрах регистрации указывается:
   - дата и время попытки доступа к защищаемому файлу с указанием ее результата (успешная, неуспешная — несанкционированная);
   - идентификатор субъекта доступа;
   - спецификация защищаемого файла.
5. Регистрировать попытки доступа программных средств к следующим дополнительным защищаемым объектам доступа: терминалам, ЭВМ, узлам сети ЭВМ, линиям (каналам) связи, внешним устрой-

ствам ЭВМ, программам, томам, каталогам, файлам, записям, полям записей. При этом в параметрах регистрации указывается:

- дата и время попытки доступа к защищаемому файлу с указанием ее результата: успешная, неуспешная, несанкционированная;
- идентификатор субъекта доступа;
- спецификация защищаемого объекта [логическое имя (номер)].

6. Проводить учет всех защищаемых носителей информации с помощью их маркировки и с занесением учетных данных в журнал (учетную карточку).
7. Регистрировать выдачу (приемку) защищаемых носителей.
8. Осуществлять очистку (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей. При этом очистка должна производиться однократной произвольной записью в освобождаемую область памяти, ранее использованную для хранения защищаемых данных (файлов).

Подсистема **обеспечения** целостности должна:

1. Обеспечивать целостность программных средств системы защиты информации от НСД (СЗИ НСД), обрабатываемой информации, а также неизменность программной среды. При этом:
   - целостность СЗИ НСД проверяется при загрузке системы по контрольным суммам компонент СЗИ;
   - целостность программной среды обеспечивается использованием трансляторов с языка высокого уровня и отсутствием средств модификации объектного кода программ в процессе обработки и (или) хранения защищаемой информации.
2. Осуществлять физическую охрану СВТ (устройств и носителей информации). При этом должны предусматриваться контроль доступа в помещение АС посторонних лиц, а также наличие надежных препятствий для несанкционированного проникновения в помещение АС и хранилище носителей информации. Особенно в нерабочее время.
3. Проводить периодическое тестирование функций СЗИ НСД при изменении программной среды и персонала АС с помощью тест-программ, имитирующих попытки НСД.
4. Иметь в наличии средства восстановления СЗИ НСД. При этом предусматривается ведение двух копий программных средств СЗИ НСД, а также их периодическое обновление и контроль работоспособности.

1.2.3. Требования к защите секретной информации

Подсистема управления доступом должна:

1. Идентифицировать и проверять подлинность субъектов доступа при входе в систему. Причем это должно осуществляться по идентификатору (коду) и паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов.
2. Идентифицировать терминалы, ЭВМ, узлы компьютерной сети, каналы связи, внешние устройства ЭВМ по их логическим адресам (номерам).
3. По именам идентифицировать программы, тома, каталоги, файлы, записи и поля записей.
4. Осуществлять контроль доступа субъектов к защищаемым ресурсам в соответствии с матрицей доступа.
5. Управлять потоками информации с помощью меток конфиденциальности. При этом уровень конфиденциальности накопителя должен быть не ниже уровня конфиденциальности записываемой на него информации.

Подсистема регистрации и учета должна:

1. Регистрировать вход (выход) субъектов доступа в систему (из системы) либо регистрировать загрузку и инициализацию операционной системы и ее программного останова. При этом в параметрах регистрации указываются:
   - дата и время входа (выхода) субъекта доступа в систему (из системы) или загрузки (останова) системы;
   - результат попытки входа — успешная или неуспешная (приНСД);
   - идентификатор (код или фамилия) субъекта, предъявленный при попытке доступа;
   - код или пароль, предъявленный при неуспешной попытке.

   Регистрация выхода из системы или останова не проводится в моменты аппаратурного отключения АС.
2. Регистрировать выдачу печатных (графических) документов на «твердую» копию. Выдача должна сопровождаться автоматической маркировкой каждого листа (страницы) документа порядковым номером и учетными реквизитами АС с указанием на последнем листе документа общего количества страниц. В параметрах регистрации указываются:
   - дата и время выдачи (обращения к подсистеме вывода);
   - краткое содержание документа (наименование, вид, код, шифр) и уровень его конфиденциальности;

- спецификация устройства выдачи (логическое имя (номер) внешнего устройства);
- идентификатор субъекта доступа, запросившего документ;
- объем фактически выданного документа (количество страниц, листов, копий) и результат выдачи (успешный — весь объем, неуспешный).

3. Регистрировать запуск (завершение) программ и процессов (заданий, задач), предназначенных для обработки защищаемых файлов. В параметрах регистрации указывается:
   - дата и время запуска;
   - имя (идентификатор) программы (процесса, задания);
   - идентификатор субъекта доступа, запросившего программу (процесс, задание);
   - результат запуска (успешный, неуспешный - - несанкционированный).
4. Регистрировать попытки доступа программных средств (программ, процессов, задач, заданий) к защищаемым файлам. В параметрах регистрации указывается:
   - дата и время попытки доступа к защищаемому файлу с указанием ее результата: (успешная, неуспешная — несанкционированная);
   - идентификатор субъекта доступа;
   - спецификация защищаемого файла;
   - имя программы (процесса, задания, задачи), осуществляющих доступ к файлам;
   - вид запрашиваемой операции (чтение, запись, удаление, выполнение, расширение и т.п.).
5. Регистрировать попытки доступа программных средств к следующим дополнительным защищаемым объектам доступа: терминалам, ЭВМ, узлам сети ЭВМ, линиям (каналам) связи, внешним устройствам ЭВМ, программам, томам, каталогам, файлам, записям, полям записей. В параметрах регистрации указывается:
   - дата и время попытки доступа к защищаемому файлу с указанием ее результата (успешная, неуспешная — несанкционированная);
   - идентификатор субъекта доступа;
   - спецификация защищаемого объекта [логическое имя (номер)];
   - имя программы (процесса, задания, задачи), осуществляющих доступ к файлам;
   - вид запрашиваемой операции (чтение, запись, монтирование, захват и т.п.).

6. Регистрировать изменения полномочий субъектов доступа, а также статуса объектов доступа. В параметрах регистрации указывается:
   - дата и время изменения полномочий;
   - идентификатор субъекта доступа (администратора), осуществившего изменения.
7. Осуществлять автоматический учет создаваемых защищаемых файлов с помощью их дополнительной маркировки, используемой в подсистеме управления доступом. Маркировка должна отражать уровень конфиденциальности объекта.
8. Проводить учет всех защищаемых носителей информации с помощью их маркировки и с занесением учетных данных в журнал (учетную карточку).
9. Проводить учет защищаемых носителей с регистрацией их выдачи (приема) в специальном журнале (картотеке).
10. Проводить несколько видов учета (дублирующих) защищаемых носителей информации.
11. Осуществлять очистку (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей. Причем очистка должна осуществляется двукратной произвольной записью в освобождаемую область памяти, ранее использованную для хранения защищаемых данных (файлов).
12. Сигнализировать о попытках нарушения защиты.

Подсистема обеспечения целостности должна:

1. Обеспечивать целостность программных средств СЗИ НСД, обрабатываемой информации, а также неизменность программной среды. При этом:
   - целостность СЗИ НСД проверяется при загрузке системы по контрольным суммам компонент СЗИ;
   - целостность программной среды обеспечивается использованием трансляторов с языка высокого уровня и отсутствием средств модификации объектного кода программ в процессе обработки и (или) хранения защищаемой информации.
2. Осуществлять физическую охрану СВТ (устройств и носителей информации). При этом должно предусматриваться постоянное наличие охраны на территории здания и помещений, где находится АС. Охрана должна производиться с помощью технических средств охраны и специального персонала, а также с использованием строгого пропускного режима и специального оборудования в помещении АС.
3. Предусматривать наличие администратора или целой службы защиты информации, ответственных за ведение, нормальное функционирование и контроль работы СЗИ НСД. Администратор должен

иметь свой терминал и необходимые средства оперативного контроля и воздействия на безопасность АС.

4. Проводить периодическое тестирование функций СЗИ НСД при изменении программной среды и персонала АС с помощью специальных программных средств не реже одного раза в год.
5. Иметь в наличии средства восстановления СЗИ НСД, предусматривающие ведение двух копий программных средств СЗИ НСД и их периодическое обновление и контроль работоспособности.
6. Использовать только сертифицированные средства защиты. Их сертификацию проводят специальные сертификационные центры или специализированные предприятия, имеющие лицензию на проведение сертификации средств защиты СЗИ НСД.

1.2.4. Различия требований и основополагающих механизмов защиты от НСД

Сравним две рассмотренные выше группы требований и их особенности для защиты информации различных категорий (конфиденциальной и секретной).

Ясно, что ключевыми механизмами защиты, образующими основную группу механизмов защиты от НСД («Подсистема управления доступом») являются:

* идентификация и проверка подлинности субъектов доступа при входе в систему по идентификатору (коду) и паролю условно-постоянного действия;
» контроль доступа субъектов к защищаемым ресурсам в соответствии с матрицей доступа.

Дополнительным требованием и принципиальным отличием при защите секретной информации является то, что механизмом защиты должно осуществляться управление потоками информации с помощью меток конфиденциальности. При этом уровень конфиденциальности накопителя должен быть не ниже уровня конфиденциальности записываемой на него информации.

Все три перечисленных механизма являются основополагающими. Связаны они следующим образом: все права доступа к ресурсам (разграничительная политика доступа к ресурсам) задаются для конкретного субъекта доступа (пользователя). Поэтому субъект доступа (пользователь) должен быть идентифицирован при входе в систему, соответственно, должна быть проконтролирована его подлинность. Обычно это делается путем использования секретного слова — пароля.

Теперь, чтобы было понятно дальнейшее изложение материала, остановимся более подробно на рассмотрении механизмов, реализующих основу

защиты компьютерной информации от НСД — разграничительную политику доступа к ресурсам. Таковыми механизмами являются механизмы контроля доступа. Более подробно речь об этом пойдет в четвертой части книги. Однако уже сейчас необходимо понимать, о чем идет речь.

Следуя формализованным требованиям к системе защиты информации, основу реализации разграничительной политики доступа к ресурсам при обработке сведений конфиденциального характера является **дискреционный механизм управления доступом.** При этом реализуется дискреционная модель доступа к ресурсам. Принципы организации и функционирования этой модели мы подробно рассмотрим в главе 11. Сейчас лишь отметим, что при дискреционной модели права доступа задаются матрицей доступа, элементами которой являются разрешенные права доступа субъекта к объекту. Что касается контроля доступа, то он осуществляется непосредственно путем анализа прав доступа к объекту запрашивающего доступ субъекта. При этом анализируется, есть ли в матрице информация о разрешении доступа данного субъекта к данному объекту, или нет.

При защите секретной информации в основе разграничительной политики доступа (РПД) к ресурсам должен лежать помимо дискреционного (дискреционная модель управления доступом), **мандатный механизм управления доступом** (мандатная модель управления доступом). Мы его также будем подробно рассматривать в главе 11.

В рамках мандатного механизма каждому субъекту (пользователю, приложению и т.д.) и каждому объекту (файлу, каталогу и т.д.) ставятся в соответствие специальные классификационные метки. Посредством этих меток субъектам и объектам назначаются классификационные уровни (уровни уязвимости, категории секретности и т. п.), являющиеся комбинациями иерархических и неиерархических категорий. Сам контроль и управление доступом осуществляется путем сопоставления классификационных меток субъекта и объекта доступа, отражающих их место в соответствующей иерархии. В общих чертах в этом и заключается мандатный механизм управления доступом. То есть право доступа дается на основе сравнения меток объекта и субъекта. При этом, чтобы субъект получил доступ к объекту, его уровень конфиденциальности должен быть не ниже уровня конфиденциальности объекта.

Требования к практической реализации дискретного и мандатного механизмов управления доступом, как и сами эти методы, рассмотрены в главе 11. Тем не менее, уже сейчас стоит отметить, что эти требования тесно связаны с требованиями к обеспечению целостности и очистки памяти.

Для системы защиты выполнение требования «Должна быть обеспечена целостность программных средств системы защиты информации от НСД (СЗИ НСД), обрабатываемой информации, а также замкнутость программной среды» обеспечивает возможность противодействия

скрытым действиям пользователей, направленных на получение НСД к информации в обход РПД. При этом не важно, чем будет пользоваться потенциальный нарушитель — программами собственной разработки, всевозможными отладочными средствами или иным ПО. Далее будет показано, что обеспечение замкнутости программной среды -- это важнейший механизм защиты в части противодействия скрытым атакам.

Требование к очистке памяти «Должна осуществляться очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей. Очистка осуществляется двукратной произвольной записью в освобождаемую область памяти, ранее использованную для хранения защищаемых данных (файлов)» обусловливает невозможность доступа пользователя к остаточной информации. Его необходимость вызвана тем, что при удалении файла на внешнем накопителе системными средствами осуществляется изменение разметки накопителя, но собственно информация остается. Она так и называется «остаточная информация». При этом ввиду того, что остаточная информация уже не является объектом доступа (она соответствующим образом не размечена на диске — не является файлом), дискреционный и мандатный механизмы контроля доступа не могут осуществлять РПД к этой информации. А это значит, что доступ к ней может быть осуществлен в обход подсистемы управления доступом.

глава2

Анализ защищенности современных операционных систем

2.1. Основные механизмы защиты ОС. Анализ выполнения современными ОС формализованных требований к защите информации от НСД

2.1.1. Принципиальные различия в подходах обеспечения защиты. Разность концепций

Сразу отметим, что анализировать выполнение современными универсальными ОС требований, задаваемых для класса защищенности автоматизированных систем 1В (защита секретной информации), не имеет смысла в принципе. Для большинства ОС либо полностью не реализуется основной для данных приложений мандатный механизм управления доступом к ресурсам, либо не выполняется его важнейшее требование «Должно осуществляться управление потоками информации с помощью меток конфиденциальности. При этом уровень конфиденциальности накопителя должен быть не ниже уровня конфиденциальности записываемой на него информации».

В связи с этим далее будем говорить лишь о возможном соответствии средств защиты современных ОС классу автоматизированных систем 1Г (защита конфиденциальной информации).

Примечание

В общем случае возможна неоднозначная трактовка некоторых формализованных требований. Проиллюстрируем сказанное примером. Рассмотрим требование «Для каждой пары (субъект — объект) в средстве вычислительной техники (СВТ) должно быть задано явное и недвусмысленное перечисление допустимых типов доступа (читать, писать и т.д.), т.е. тех типов доступа, которые являются санкционированными для данного субъекта (индивида или

группы индивидов) к данному ресурсу СВТ (объекту)». Возникают вопросы: что понимается под условием «для каждой пары...», т.е. причисляются ли в данном требовании к «объектам» программы (процессы или исполняемые файлы), устройства и т.д, а к субъектам — процессы (программы)? Далее, что понимается под условием «должно быть задано явное и недвусмысленное перечисление допустимых типов доступа (читать, писать и т.д.)», что значит «и т.д.», относится ли к нему, например, «исполнение» и др. В связи с этим автор далее будет трактовать все неоднозначности в пользу усиления механизмов защиты. Обоснованность подобного подхода мы увидим ниже, при анализе существующей статистики угроз.

В качестве альтернативных реализаций ОС рассмотрим семейства UNIX и Windows (естественно, Windows NT/2000, т.к. о встроенных механизмах защиты ОС Windows 9x/Me говорить вообще не приходится).

Сначала остановимся на принципиальном, даже можно сказать, концептуальном противоречии между реализованными в ОС механизмами защиты и принятыми формализованными требованиями. Концептуальном в том смысле, что это противоречие характеризует не какой-либо один механизм защиты, а общий подход к построению системы защиты.

Противоречие состоит в принципиальном различии подходов (соответственно требований) к построению схемы администрирования механизмов защиты. Как следствие, это коренным образом сказывается на формировании общих принципов задания и реализации политики безопасности на предприятии, распределения ответственности за защиту информации на предприятии, а также на определении того, кого относить к потенциальным злоумышленникам (от кого защищать информацию). То есть сказывается на ключевых вопросах защиты информации.

Для иллюстрации из совокупности формализованных требований к системе защиты конфиденциальной информации (требований к дискреционному механизму управления доступом) рассмотрим следующие два требования [1]:

1. Право изменять правила разграничения доступа (ПРД) должно предоставляться выделенным субъектам (администрации, службе безопасности и т.д.).
2. Должны быть предусмотрены средства управления, ограничивающие распространения прав на доступ.

Примечание

С полным перечнем требований вы ознакомитесь в главе 11, когда непосредственно приступите к рассмотрению методов управления доступом.

Данные требования жестко регламентируют схему (или модель) администрирования механизмов защиты. Это должна быть централизованная схема, единственным элементом которой выступает выделенный субъект, в частности, администратор (в общем случае следует говорить об адми-

нистраторе безопасности). При этом конечный пользователь исключен в принципе из схемы администрирования механизмов защиты.

При реализации концепции построения системы защиты, регламентируемой рассматриваемыми требованиями, пользователь не наделяется элементом доверия. Таким образом, он может считаться потенциальным злоумышленником, что и имеет место на практике -- этот вопрос будет рассмотрен далее. Ответственным за информационную безопасность на предприятии является доверенное с его стороны лицо — выделенный субъект, в частности, администратор безопасности.

Теперь в общих чертах рассмотрим концепцию, реализуемую в современных универсальных ОС (подробнее см. гл. 2). Здесь «владельцем» файлового объекта, то есть лицом, получающим право на задание атрибутов (или ПРД) доступа к файловому объекту, является лицо, создающее файловый объект. Т.к. файловые объекты создают конечные пользователи (иначе, для чего нужен компьютер?), то именно они и назначают ПРД к создаваемым им файловым объектам. Другими словами, в ОС реализуется распределенная схема назначения ПРД, где элементами схемы администрирования являются собственно конечные пользователи.

В данной схеме пользователь должен со стороны предприятия наделяться практически таким же доверием, как и администратор безопасности, при этом нести наряду с ним ответственность за обеспечение компьютерной безопасности. Отметим, что данная концепция реализуется и большинством современных приложений, в частности СУБД, где пользователь может распространять свои права на доступ к защищаемым ресурсам.

Кроме того, не имея в полном объеме механизмов защиты компьютерной информации от конечного пользователя, в рамках данной концепции невозможно рассматривать пользователя в качестве потенциального злоумышленника. А как мы увидим далее, именно с несанкционированными действиями пользователя на защищаемом компьютере (причем как сознательными, так и нет) связана большая часть угроз компьютерной безопасности.

Отметим, что централизованная и распределенная схемы администрирования — это две диаметрально противоположные точки зрения на защиту, требующие совершенно различных подходов к построению моделей и механизмов защиты. На наш взгляд, сколько-нибудь гарантированную защиту информации можно реализовать только при принятии концепции **полностью централизованной схемы администрирования.** Кстати, это подтверждается известными угрозами ОС.

Возможности моделей, методов и средств защиты будем далее рассматривать применительно к реализации именно концепции централизованного администрирования. Одним из элементов данной концепции является рассмотрение пользователя в качестве потенциального злоумышленника, способного осуществить НСД к защищаемой информации.

2.1.2. Основные встроенные механизмы защиты ОС и их недостатки

Кратко остановимся на основных механизмах защиты, встроенных в современные универсальные ОС. Сделаем это применительно к возможности реализации ими принятой нами для рассмотрения концепции защиты конфиденциальной информации.

Основные защитные механизмы ОС семейства UNIX

Защита ОС семейства UNIX в общем случае базируется на трех основных механизмах:

- « идентификации и аутентификация пользователя при входе в систему;
- » разграничении прав доступа *к* файловой системе, в основе которого лежит реализация дискреционной модели доступа;
- аудит, то есть регистрация событий.

При этом отметим, что для различных клонов ОС семейства UNIX возможности механизмов защиты могут незначительно различаться, однако будем рассматривать ОС UNIX в общем случае, без учета некоторых незначительных особенностей отдельных ОС этого семейства.

Построение файловой системы и разграничение доступа к файловым объектам имеет особенности, присущие данному семейству ОС. Рассмотрим кратко эти особенности. Все дисковые накопители (тома) объединяются в единую ВИРТУАЛЬНУЮ ФАЙЛОВУЮ СИСТЕМУ путем операции монтирования тома. При этом содержимое тома проецируется на выбранный каталог файловой системы. Элементами файловой системы являются также все устройства, подключаемые к защищаемому компьютеру (монтируемые к файловой системе). Поэтому разграничение доступа к ним осуществляется через файловою систему.

Каждый файловый объект имеет индексный дескриптор (описатель), в котором среди прочего хранится информация о разграничении доступа к данному файловому объекту. Права доступа делятся на три категории: доступ для владельца, доступ для группы и доступ для остальных пользователей. В каждой категории определяются права на чтение, запись и исполнение (в случае каталога — просмотр).

Пользователь имеет уникальные символьный идентификатор (имя) и числовой идентификатор (UID). Символьный идентификатор предъявляется пользователем при входе в систему, числовой используется операционной системой для определения прав пользователя в системе (доступ к файлам и т.д.).

Принципиальные недостатки защитных механизмов ОС семейства UNIX

Рассмотрим в общем случае недостатки реализации системы защиты ОС семейства UNIX в части невыполнения требований к защите конфиденциальной информации. При этом прежде всего рассмотрим принципиальные недостатки защиты, напрямую связанные с возможностью НСД к информации.

Для начала отметим, что в ОС семейства UNIX, вследствие реализуемой ею концепции администрирования (не централизованная), **невозможно обеспечить замкнутость (или целостность) программной среды.** Это связано с невозможностью установки атрибута «исполнение» на каталог (для каталога данный атрибут ограничивает возможность «обзора» содержимого каталога). Поэтому при разграничении администратором доступа пользователей к каталогам, пользователь, как «владелец» создаваемого им файла, может занести в свой каталог исполняемый файл и, как его «владелец», установить на файл атрибут «исполнение», после чего запустить записанную им программу. Эта проблема непосредственно связана с реализуемой в ОС концепцией защиты информации.

Не в полном объеме реализуется дискреционная модель доступа, в частности не могут разграничиваться права доступа для пользователя «root» (UID = 0). Т.е. данный субъект доступа исключается из схемы управления доступом к ресурсам. Соответственно все запускаемые им процессы имеют неограниченный доступ к защищаемым ресурсам. Как мы увидим позднее, с этим недостатком системы защиты связано множество атак, в частности:

* несанкционированное получение прав root;
» запуск с правами root собственного исполняемого файла (локально либо удаленно внедренного). При этом несанкционированная программа получает полный доступ к защищаемым ресурсам и т.д.

Кроме того, в ОС семейства UNIX **невозможно встроенными средствами гарантированно удалять остаточную информацию.** Для этого в системе абсолютно отсутствуют соответствующие механизмы.

Необходимо также отметить, что большинство ОС данного семейства **не обладают возможностью контроля целостности файловой системы,** то есть не содержат соответствующих встроенных средств. В лучшем случае дополнительными утилитами может быть реализован контроль конфигурационных файлов ОС по расписанию, в то время, как важнейшей возможностью данного механизма можно считать контроль целостности программ (приложений) перед их запуском, контроль файлов данных пользователя и др.

Что касается регистрации (аудита), то в ОС семейства Unix **не обеспечивается регистрация выдачи документов на «твердую копию», а также некоторые другие требования к регистрации событий.**

Если же трактовать требования к управлению доступом в общем случае, то при защите компьютера в составе ЛВС, необходимо управление доступом к хостам (распределенный пакетный фильтр). Однако **встроенными средствами зашиты некоторых ОС семейства** UNIX **управление доступом к хостам не реализуется.**

Из приведенного анализа видно, что многие механизмы, необходимые с точки зрения выполнения формализованных требований, большинством ОС семейства UNIX не реализуется в принципе, либо реализуется лишь частично.

Основные защитные механизмы ОС семейства Windows(NT/2000/XP)

Теперь кратко остановимся на основных механизмах защиты, реализованных в ОС семейства Windows, и проведем анализ защищенности ОС семейства Windows (NT/2000). Отметим, что здесь ряд объектов доступа (в частности, устройства, реестр ОС и т.д.) не являются объектами файловой системы. Поэтому возникает вопрос, как следует трактовать требование «Система защиты должна контролировать доступ наименованных субъектов (пользователей) к наименованным объектам (файлам, программам, томам и т.д.)». То есть, не ясно, являются ли объектами доступа, к которым, следуя формальным требованиям, необходимо разграничивать доступ пользователей, например, реестр ОС и т.д.

В отличие от семейства ОС UNIX, где все задачи разграничительной политики доступа к ресурсам решаются средствами управления доступом к объектам файловой системы, доступ в данных ОС разграничивается собственным механизмом для каждого ресурса. Другими словами, при рассмотрении механизмов защиты ОС Windows встает задача определения и задания требований к полноте разграничений (это определяется тем, что считать объектом доступа).

Также, как и для семейства ОС UNIX, здесь основными механизмами защиты являются:

- идентификация и аутентификация пользователя при входе в систему;
- разграничение прав доступа к ресурсам, в основе которого лежит реализация дискреционной модели доступа (отдельно к объектам файловой системы, к устройствам, к реестру ОС, к принтерам и др.);
- аудит, то есть регистрация событий.

Здесь явно выделяются (в лучшую сторону) возможности разграничений прав доступа к файловым объектам (для NTFS) — существенно расширены атрибуты доступа, устанавливаемые на различные иерархические объекты файловой системы (логические диски, каталоги, файлы). В частности, атрибут «исполнение» может устанавливаться и на каталог, тогда он наследуется соответствующими файлами (в отличие от ОС семейства UNIX).

При этом существенно ограничены возможности управления доступом к другим защищаемым ресурсам, в частности, к устройствам ввода. Например, здесь отсутствует атрибут «исполнение», т.е. невозможно запретить запуск несанкционированной программы с устройств ввода.

Принципиальные недостатки защитных механизмов ОС семейства Windows (NT/2000/XP)

Прежде всего рассмотрим принципиальные недостатки защиты ОС семейства Windows, напрямую связанные с возможностью НСД к информации. При этом в отличие от ОС семейства UNIX в **ОС Windows невозможна в общем случае реализация централизованной схемы администрирования** механизмов защиты или соответствующих формализованных требований. Вспомним, что в ОС UNIX это распространялось лишь на запуск процессов. Связано это с тем, что в ОС Windows принята иная концепция реализации разграничительной политики доступа к ресурсам (для NTFS).

В рамках этой концепции разграничения для файла приоритетнее, чем для каталога, а в общем случае — разграничения для включаемого файлового объекта приоритетнее, чем для включающего (более подробно данный вопрос анализируется в четвертой части книги). Это приводит к тому, что пользователь, создавая файл и являясь его «владельцем», может назначить любые атрибуты доступа к такому файлу (т.е. разрешить к нему доступ любому иному пользователю). При этом обратиться к этому файлу может пользователь (которому назначил права доступа «владелец») вне зависимости от установленных администратором атрибутов доступа на каталог, в котором пользователь создает файл. Данная проблема непосредственно связана с реализуемой в ОС Windows концепцией защиты информации.

Далее, в ОС семейства Windows (NT/2000/XP) **не в полном объеме реализуется дискреционная модель доступа,** в частности, не могут разграничиваться права доступа для пользователя «Система». В ОС присутствуют не только пользовательские, но и системные процессы, которые запускаются непосредственно системой. При этом доступ системных процессов не может быть разграничен. Соответственно, все запускаемые системные процессы имеют неограниченный доступ к защищаемым ресурсам. Как увидим ниже, с этим недостатком системы защиты связано множество атак, в частности, несанкционированный запуск собственного процесса с правами системного. Кстати, позднее мы увидим, что это возможно и вследствие некорректной реализации механизма обеспечения замкнутости программной среды.

В ОС семейства Windows (NT/2000/XP) **невозможно в общем случае обеспечить замкнутость (или целостность) программной среды.** Это связано совершено с иными проблемами, чем в ОС семейства UNIX, в которых невозможно установить атрибут «исполнение» на каталог. Давайте выясним, в чем сложности у ОС Windows в этом вопросе.

Рассмотрим два способа, которыми в общем случае можно реализовать данный механизм (подробнее об этом читайте в соответсвующих главах), и покажем их несостоятельность в ОС Windows. Итак, механизм замкнутости программной среды в общем случае может быть обеспечен:

- Заданием списка разрешенных к запуску процессов с предоставлением возможности пользователям запускать процессы только из этого списка. При этом процессы задаются **полнопутевыми именами,** причем средствами разграничения доступа обеспечивается невозможность их модернизации пользователем. Данный подход просто не реализуется встроенными в ОС механизмами.
- Разрешением запуска пользователями программ только из заданных каталогов при невозможности модернизации этих каталогов. Одним из условий корректной реализации данного подхода является запрет пользователям запуска программ иначе, чем из соответствующих каталогов. Некорректность реализации ОС Windows данного подхода связана с невозможностью установки атрибута «исполнение» на устройства ввода (дисковод или CD-ROM). В связи с этим при разграничении доступа пользователь может запустить несанкционированную программу с дискеты, либо с диска CD-ROM (как далее увидим – это очень распространенная атака на ОС данного семейства).

Здесь же стоит отметить, что с точки зрения обеспечения замкнутости программной среды (т.е. реализации механизма, обеспечивающего возможность пользователям запускать только санкционированные процессы (программы)) действия пользователя по запуску процесса могут быть как явными, так и скрытыми.

Явные действия предполагают запуск процессов (исполняемых файлов), которые однозначно идентифицируются своим именем. Скрытые действия позволяют осуществлять встроенные в приложения интерпретаторы команд. Примером таковых могут служить офисные приложения. При этом скрытыми действиями пользователя будет запуск макроса.

В данном случае идентификации подлежит лишь собственно приложение, например, процесс winword.exe. При этом он может помимо своих регламентированных действий выполнять те скрытые действия, которые задаются макросом (соответственно, те, которые допускаются интерпретатором), хранящимся в открываемом документе. То же относится и к любой виртуальной машине, содержащей встроенный интерпретатор команд. При этом отметим, что при использовании приложений, имеющих встроенные интерпретаторы команд (в том числе офисных приложений), не в полном объеме обеспечивается выполнение требования по идентификации программ.

Возвращаясь к обсуждению недостатков, отметим, что в ОС семейства Windows (NT/2000/XP) **невозможно встроенными средствами гарантированно удалять остаточную информацию.** В системе просто отсутствуют соответствующие механизмы.

Кроме того, ОС семейства Windows (NT/2000/XP) **не обладают в полном объеме возможностью контроля целостности файловой системы.** Встроенные механизмы системы позволяют контролировать только собственные системные файлы, не обеспечивая контроль целостности файлов пользователя. Кроме того, они не решают важнейшую задачу данных механизмов — контроль целостности программ (приложений) перед их запуском, контроль файлов данных пользователя и др.

Что касается регистрации (аудита), то в ОС семейства Windows (NT/2000/XP) **не обеспечивается регистрация выдачи документов на «твердую копию», а также некоторые другие требования к регистрации событий.**

Опять же, если трактовать требования к управлению доступом в общем случае, то при защите компьютера в составе ЛВС необходимо управление доступом к хостам (распределенный пакетный фильтр). В ОС семейства Windows (NT/2000/XP) **механизм управления доступа к хостам в полном объеме не реализуется.**

Что касается разделяемых сетевых ресурсов, то фильтрации подвергается только входящий доступ к разделяемому ресурсу, а запрос доступа на компьютере, с которого он осуществляется, фильтрации не подлежит. Это принципиально, т.к. не могут подлежать фильтрации приложения, которыми пользователь осуществляет доступ к разделяемым ресурсам. Благодаря этому, очень распространенными являются атаки на протокол NETBIOS.

Кроме того, в полном объеме (в части фильтрации только входящего трафика) управлять доступом к разделяемым ресурсам возможно только при установленной на всей компьютерах ЛВС файловой системы NTFS. В противном случае невозможно запретить запуск несанкционированной программы с удаленного компьютера, то есть обеспечить замкнутость программной среды в этой части.

Из приведенного анализа можем видеть, что многие механизмы, необходимые с точки зрения выполнения формализованных требований, ОС семейства Windows не реализуют в принципе, либо реализуют лишь частично.

Выводы

С учетом сказанного можем сделать важный вывод относительно того, что большинством современных универсальных ОС не выполняются в полном объеме требования к защите АС по классу 1Г. Это значит, что, учитывая требования нормативных документов [1, 2], они не могут без использования добавочных средств защиты применяться для защиты даже конфиденциальной информации. При этом следует отметить, что основные проблемы защиты здесь вызваны не невыполнимостью ОС требований к отдельным механизмам защиты, а принципиальными причинами, обусловленными реализуемой в ОС концепцией защиты. Концепция эта

основана на реализации распределенной схемы администрирования механизмов защиты, что само по себе является невыполнением формализованных требований к основным механизмам защиты.

2.2. Анализ существующей статистики угроз для современных универсальных ОС. Дополнительные требования к защите компьютерной информации

В данном разделе мы рассмотрим существующие (опубликованные) атаки на ОС. При этом мы определим, в какой мере невыполнение рассмотренных ранее требований к защите сказывается на уязвимости ОС. Таким образом, мы попытаемся практически подтвердить необходимость усиления встроенных механизмов защиты.

2.2.1. Семейства ОС и общая статистика угроз

На сегодняшний день существует достаточно большая статистика угроз ОС [5, 34, 35], направленных на преодоление встроенных в ОС механизмов защиты, позволяющих изменить настройки механизмов безопасности, обойти разграничения доступа и т.д.

Таким образом, статистика фактов несанкционированного доступа к информации показывает, что большинство распространенных систем (универсального назначения) довольно уязвимы с точки зрения безопасности. И это несмотря на отчетливую тенденцию к повышению уровня защищенности этих систем.

Здесь же необходимо отметить, что на практике современные информационные системы, предназначенные для обработки конфиденциальной информации, строятся уже с учетом дополнительных мер безопасности. А это также косвенно подтверждает изначальную уязвимость современных ОС.

Проиллюстрируем сказанное. Для этого рассмотрим операционные системы, фигурирующие в публикуемых списках системных и прикладных ошибок, то есть ошибок, позволяющих получить несанкционированный доступ к системе, понизить степень ее защищенности или добиться отказа в обслуживании (системного сбоя). Итак, вот эти операционные системы:

- MS Windows 9X
- MS Windows NT
- MS Windows 2000
- Novell NetWare
- BSD
- BSDI
- Solaris
- Sun OS
- Digital Unix
- HPUX
- AIX
- SCO
- IOS (Cisco)
- Linux
- IRIX

Общее количество известных успешных атак для различных ОС (по данным RootShell, Rhino9, SecurityFocus) [34, 35], представлено в табл. 2.1, а их процентное соотношение для различных типов ОС - на диаграмме рис. 2.1.

Общее количество известных успешных атак для различных ОС *Таблица 2.1*

Тип ОС	Количество атак	Тип ОС	Количество атак
MS Windows NT/2000	130	Linux	167
MS Windows 9X/ME	120	IRIX	84
BSD	64	HPUX	65
BSDI	10	AIX	42
Solaris	125	SCO	40
Sun OS	40	Novell NetWare	10
Digital Unix	25	IOS (Cisco)	7

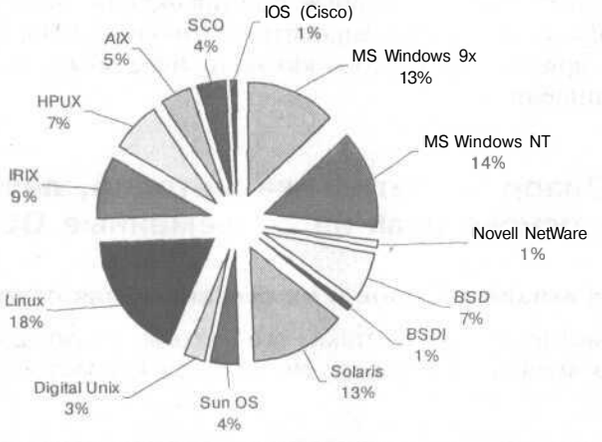


Рис. 2.1. *Статистика соотношения угроз для различных ОС*

Вследствие того, что большинство атак для операционных систем, построенных на базе UNIX (BSD или AT&T), достаточно похожи, целесообразно объединить их в одну группу. Тоже самое можно сказать и об ОС семейства Windows. Таким образом, в дальнейшем будем рассматривать только семейства ОС:

- UNIX
- MS Windows
- Novell NetWare

Общее количество известных успешных атак для различных групп ОС представлено в табл. 2.2, а их процентное соотношение для различных типов ОС — на диаграмме рис. 2.2.

Таблица 2.2
Общее количество известных успешных атак для различных групп ОС

Тип ОС	Количество атак
MS Windows	230
UNIX	660
Novell Netware	10

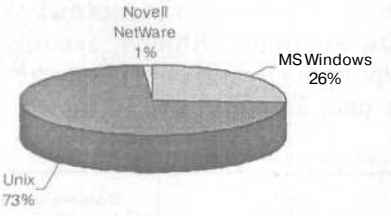


Рис. 2.2. *Статистика соотношения угроз для семейств ОС*

Относительно ОС Novell следует заметить, что данная ОС изначально создавалась как защищенная (не универсального назначения) ОС, основной функцией которой был защищенный файловый сервис. Это, с одной стороны, должно было обеспечить ее более высокий уровень защищенности, с другой стороны, налагало определенные ограничения по использованию. Однако, начиная с пятой версии, данная ОС начала приобретать свойства универсальности (с точки зрения применяемых протоколов и приложений), что в какой-то мере может сказаться и на уровне ее защищенности.

2.2.2. Обзор и статистика методов, лежащих в основе атак на современные ОС

Классификация методов и их сравнительная статистика

Анализируя рассматриваемые атаки, все методы, позволяющие несанкционированно вмешаться в работу системы, можно разделить на следующие группы:

1. Позволяющие несанкционированно запустить исполняемый код.
2. Позволяющие осуществить несанкционированные операции чтения/записи файловых или других объектов.
3. Позволяющие обойти установленные разграничения прав доступа.
4. Приводящие к отказу (Denial of Service) в обслуживании (системный сбой).
5. Использующие встроенные недокументированные возможности (ошибки и закладки).
6. Использующие недостатки системы хранения или выбора (недостаточная длина) данных об аутентификации (пароли) и позволяющие путем реверсирования, подбора или полного перебора всех вариантов получить эти данные.
7. Троянские программы.
8. Прочие.

Диаграмма, представляющая собой соотношение групп атак (для представленной выше их классификации) для ОС семейства Windows, представлена на рис. 2.3, для ОС семейства UNIX — на рис. 2.4.

Давайте очень кратко проиллюстрируем выделенные группы угроз.

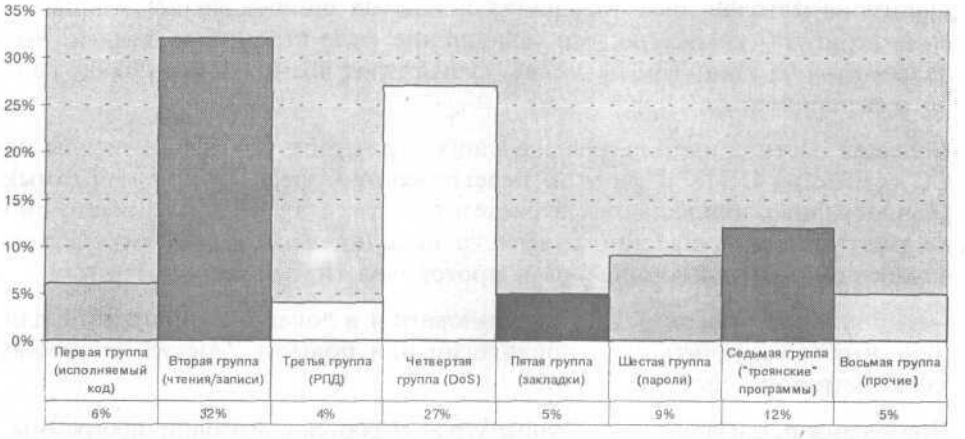


Рис. 2.3. *Соотношение групп атак для ОС семейства Windows*

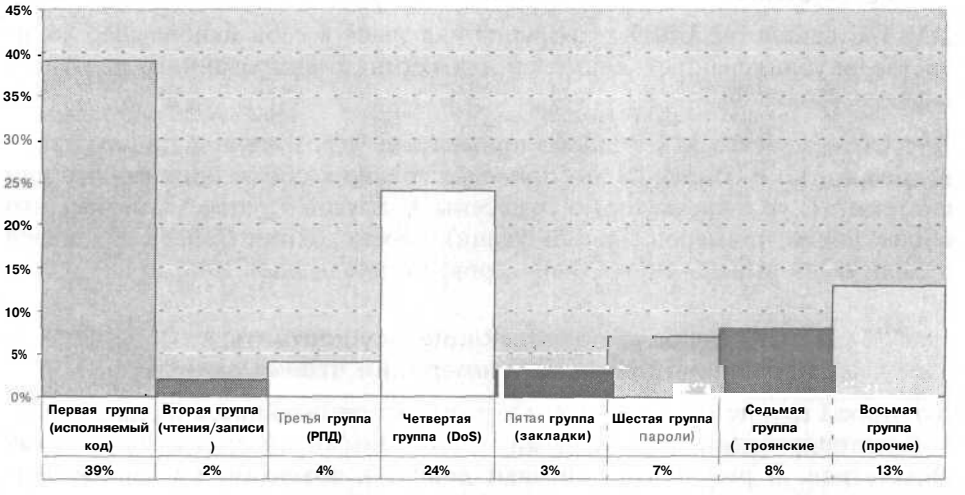


Рис. 2.4. *Соотношение групп атак для ОС семейства UNIX*

Угрозы, позволяющие несанкционированно запустить исполняемый код

К данной группе относятся угрозы, которые основываются на переполнении буфера для входных данных (переполнение стека) и последующей передачи управления на исполняемый код, занесенный при этом в стек. Для переполнения стека используется тот факт, что часто при выполнении функций работы со строками, переменными среды исполнения и т.д.,

разработчики ПО не заботятся о проверке размерности входных данных. А это приводит к выходу за границы массивов, выделенных для работы с этими данными. В последнее время появилось целое направление системных средств по борьбе с угрозами данной группы (Pax, StackGuard).

Одним из методов предотвращения подобных ошибок является присвоение атрибута, исключающего исполнение кода страницам памяти, выделенным под стек. Тем не менее, существуют возможности обхода данного ограничения.

Большая часть примеров, реализующих эту группу угроз, рассчитаны на ОС семейства UNIX. При этом переполнение буфера возможно в самых разнообразных приложениях и системных утилитах. Наиболее часто оно используется для удаленного запуска исполняемого кода, посредством обработчиков сетевых запросов и протоколов (ftp, telnet, pop3 и др.).

Переполнение буфера можно использовать и в локальном контексте, для того, чтобы увеличить свои привилегии или получить доступ на уровне администратора системы (root).

Примерами реализации этой группы угроз являются следующие программы:

» Zgv_exploit.c » Kmemthief.c » Imapd_exploit.c и др.

Для ОС семейства UNIX эта группа включает в себя наибольшее количество опубликованных примеров для несанкционированного доступа к системе (более 30%).

Для ОС семейства MS Windows применение угроз данной группы также возможно, но в основном это приводит только к сбоям прикладного или системного уровня, которые отнесены к другой группе. Заметим, что общее число примеров, использующих переполнения буфера для целей отличных от вывода системы из строя, не превышает 10%.

Угрозы, позволяющие осуществить несанкционированные операции чтения/записи

Ко второй группе можно отнести угрозы, основывающиеся на неправильной интерпретации прикладными и системными программами входных параметров. В результате они дают доступ к объектам, не перечисленным в списках санкционированного доступа.

Неправильная интерпретация входных параметров связана с некорректной программной реализацией их обработки. Это происходит потому, что программы, обрабатывающие данные запросы, являются либо системными утилитами, либо прикладными программами, запущенными в контексте безопасности системы. Поэтому они имеют непосредственный доступ к любым файловым (и другим) объектам, и могут предоставить этот доступ пользователями, не обладающими достаточными правами для непосредственной работы с этими объектами.

Наибольшее распространение получили реализации данных методов для ОС семейства MS Windows. В основном ошибки встречаются в стандартных включенных в состав операционных систем Internet/Intranet-приложениях, которые включены в состав ОС, таких как IIS (Internet Information Server), почтовые клиенты (MS Mail, Exchange) и др.

Достаточно большое количество ошибок данного рода можно встретить в системных утилитах, реализующих взаимодействие по сетевым протоколам прикладного уровня (NETBIOS и др.).

Например, ошибка в IIS заключается в следующем. IIS, обрабатывая запросы в формате UNICODE, может неправильно интерпретировать символы '\' , '/' и т.п. (%c0%af , %c1%9c и т.п.), что приводит в дальнейшем к генерации некорректных команд (недоступных в нормальной ситуации) и получению несанкционированного доступа к объектам.

Большое количество ошибок встречается *в* реализации Java-апплетов, VB-скриптов и т.д. в браузерах фирм Microsoft и Netscape. Через них с помощью соответствующих апплетов можно получить несанкционированный доступ к файловым объекта. А поскольку обе фирмы выпускают свои браузеры не только для ОС семейства MS Windows, но и для ОС семейства UNIX, то ошибки в большинстве случаев дублируются в версиях ПО для разных платформ. Здесь же стоит отметить, что проблема апплетов относится собственно не к языку Java, а к его реализации, например, Microsoft Java VM.

Угрозы, позволяющие обойти установленные разграничения прав доступа

К третьей группе угроз можно отнести примеры, основывающиеся на недоработках (ошибках) в ядре и системных утилитах ОС, позволяющих программными методами обходить установленные разграничения доступа к объектам системы.

Примеры ошибок, составляющих эту группу, немногочисленны, т.к. требуют детального анализа работы механизмов (функций API) ОС и соответствующей квалификации нарушителя. При этом нужно учитывать, что при рассмотрении коммерческих ОС (не имеющих общедоступных исходных текстов) данный анализ сильно затруднен, поскольку производители, по понятным причинам, крайне неохотно документируют внутреннюю архитектуру систем.

В качестве примера для данной группы можно привести известную программу «GetAdmin», реализующую получение администраторских прав, используя некорректную работу функции NTAddAtom, позволяющую записывать значения в любую область адресного пространства.

В системе Windows NT есть некий глобальный флаг NtGlobalFlag, имеющий адрес примерно 0x801XXXXX. Изменением одного из битов этого флага существует возможность превратить Windows NT в Windows NT

Checked Build. В результате право «SeDebugPrivilege» не будет необходимо для внедрения в системные процессы. Далее, внедряя свой исполняемый код (для чего нужна была привилегия «SeDebugPrivilege») в системные процессы, можно обойти любые ограничения, связанные с политикой безопасности (в данном случае создавался пользователь с администраторскими правами).

Угрозы, приводящие к отказу в обслуживании (Denial of Service — системный сбой)

К этой группе можно отнести угрозы, приводящие к отказу в обслуживании (системный сбой). Большую часть этой группы составляют примеры, основанные на недостаточной надежности реализации стека сетевых протоколов ОС. Сбои в работе ОС достигаются посылкой групп пакетов с некорректными заголовками, параметрами и т.п.

Примерами подобных программ служат:

- teardrop
- jolt/jolt2
- lornuke
- winnuke
- winfreez
- win95ping и др.

Другую часть этой группы составляют угрозы, не использующие напрямую (или совсем не использующие) детали реализации стека сетевых протоколов конкретной ОС. Они провоцируют отказ в обслуживании путем чрезмерной загрузки канала. Простейшим примером может служить посылка большого количества пакетов из источника, обладающего более скоростным каналом, приемнику, обладающему менее скоростным каналом. Таким образом полностью исчерпывается ресурс приемника, приводя к его полному или частичному отказу в обслуживания.

Более сложным примером является так называемый флудер-множитель. При отправке на удаленный хост сообщения, состоящего из 20-и байт IP-заголовка, в поле Protocol которого содержится значение 00 (что соответствует IPPROTO_RAW), удаленная система (или ближайший к провоцируемой системе маршрутизатор), получив такое сообщение, ответит сообщением ICMP-Destination Unreachable-Protocol Unreachable, длиной от 68 до 84 байт. Очевидно, что, заменяя Source Address на адрес атакуемого, провоцируется поток с коэффициентом умножения 4 (если рассчитывать динамическое сжатие, то много больше).

Следует отметить, что программы, представляющие данную группу, не нарушают напрямую безопасность атакуемой системы, а просто выводят ее из строя. Но можно представить себе пример более сложных атак, где угрозами, приводящими к отказу от обслуживания, можно устранять, на-

пример, реально действующие в системе узлы, а затем от их имени получать несанкционированный доступ к защищенным данным.

Угрозы, использующие встроенные недокументированные возможности (закладки)

К пятой группе можно отнести методы, использующие встроенные недокументированные возможности (закладки). К таким закладкам относятся:

* встроенные инженерные пароли для входа в систему;
* специальные возможности (последовательность действий) для недокументированных действий (например, в одном из хранителей экрана фирмы Microsoft присутствует сетевой код);
* закладки в разнообразных прикладных приложениях и т.п.

Примером использования встроенного инженерного пароля может служить широко известный пароль фирмы Award «AWARD_SW», позволяющий получить весь спектр прав для работы с BIOS.

Угрозы, использующие недостатки системы хранения или выбора (недостаточная длина) данных об аутентификации (пароли)

К шестой группе можно отнести угрозы, использующие недостатки системы хранения или выбора (недостаточная длина) данных об аутентификации (пароли) и позволяющие путем реверсирования, подбора или полного перебора всех вариантов получить эти данные. Эти программы основываются на недостатках алгоритмов кодирования (хеширования) паролей на защищаемые ресурсы или на вход в ОС.

Примером может служить реализация защиты разделяемых ресурсов в Windows 9X, где при разграничении доступа на уровне ресурса (по паролю), пароль для доступа хранится в реестре (HKLM\Software\Microsoft\Windows\CurrentVersion\Network\LanMan\<ИМЯ КАТАЛОГА>, в ключе ParmlEnc), зашифрованный с помощью алгоритма, который легко поддается расшифровке, поэтому легко получить исходный пароль.

Также неудачен сам алгоритм аутентификации в Windows 9X через NETBIOS. Если клиент посылает вместо полного пароля открытым текстом только его первый символ (байт), то при совпадении этого символа пароль считается правильным.

Следует отметить, что существует большое количество программ (не только для ОС семейства MS Windows), предназначенных для перебора паролей по различным алгоритмам, учитывающим слабость реализации систем аутентификации и выбора паролей. К таким программам относятся: lOphtcrack; pwlhack; pwlview; John the Ripper и др.

«Троянские» программы

Это программы, которые прописываются в автозагрузку ОС или подменяют собой системные компоненты (утилиты) ОС и выполняют несанкционированные действия. Для того, чтобы такая программа появилась в системе, пользователь должен сам (преднамеренно, либо нет) первоначально выполнить ее.

Обычно «троянские» программы распространяются под видом полезных утилит (в том числе они могут присутствовать и в некоммерческих средствах добавочной защиты информации), посылаются по почте в виде присоединяемых замаскированных исполняемых файлов, скриптов, устанавливаются злоумышленником на защищаемом компьютере вручную и т.п. После первого запуска программа заменяет собой часть системных файлов или просто добавляет себя в список загрузки и предоставляет нарушителю доступ к системе или защищаемым ресурсам.

Примером подменяющей программы может служить динамическая библиотека клиента Novell NetWare для ОС Windows NT «FPNWCLNT.DLL», перехватывающая и хранящая передаваемые пароли в открытом виде. Другие программы из этой группы: Back Orifice; Net Bus; Priority и др.

Прочие угрозы

К последней группе отнесем все остальные угрозы и программные реализации, влияющие на функционирование и безопасность компьютерной системы. В частности, большую часть угроз данной группы составляют всевозможные программы-сниферы, позволяющие в пассивном режиме «прослушивать» каналы ввода/вывода, передачи и обработки данных.

Например, сюда можно отнести сниферы клавиатуры — программы, устанавливаемые злоумышленником на защищаемый объект, с целью «прослушивания» канала ввода пароля (пароль с клавиатуры вводится в открытом виде, соответственно, в открытом виде снимается снифером). Отдельно отметим сниферы канала — программы, устанавливаемые на какой-либо компьютер в ЛВС и «прослушивающие» весь трафик в канале (особенно это критично для ЛВС, не реализующих физической сегментации канала связи).

2.2.3. Выводы из анализа существующей статистики угроз

Из приведенного анализа можем сделать следующий важный вывод: **угрозы, описанные в большинстве групп, напрямую используют различные недостатки ОС и системных приложений и позволяют при полностью сконфигурированных и работающих встроенных в ОС механизмах защиты осуществлять НСД, что подтверждает необходимость усиления встроенных механизмов защиты.**

Кроме того, анализируя представленную статистику угроз, можем сделать вывод, что большая их часть связана именно с недостатками средств защиты ОС, отмеченными выше, т.е. недостатками, связанными с невыполнением (полным, либо частичным) формализованных требований к защите, среди которых, в первую очередь, могут быть выделены:

1. Некорректная реализация механизма управления доступом, прежде всего при разграничении доступа к защищаемым объектам системных процессов и пользователей, имеющих права администратора.
2. Отсутствие обеспечения замкнутости (целостности) программной среды. Как мы видим, большинство атак осуществлялось либо с использованием некоторых прикладных программ, либо с применением встроенных в виртуальные машины средств программирования. То есть, возможность большинства атак напрямую связана с возможностью запуска злоумышленником соответствующей программы. При этом запуск может быть осуществлен как явно, так и скрыто, в рамках возможностей встроенных в приложения интерпретаторов команд.

Далее, можем сделать еще один вывод: проведенный **анализ известных угроз современным универсальным ОС полностью подтверждает, что большая их часть обусловлена именно реализуемым в ОС концептуальным подходом, состоящим в реализации схемы распределенного администрирования механизмов защиты. В** рамках этой схемы пользователь рассматривается как доверенное лицо, являющееся элементом схемы администрирования и имеющее возможность назначать/изменять ПРД. При этом он не воспринимается как потенциальный злоумышленник, который может сознательно или несознательно осуществить НСД к информации. Отсюда можем сделать и вывод об основном назначении механизмов добавочной защиты ОС — реализация централизованной схемы администрирования механизмов защиты, в рамках которой будет осуществляться противодействие НСД пользователя к информации.

Примечание

По понятным соображениям в качестве примеров в книге приводятся те ошибки, которые известны и разработчиками устранены (патчами и т.п.). Однако данные примеры иллюстрируют общие тенденции и позволяют ввести обобщенную классификацию угроз.

глава 3

Подходы к проектированию системы защиты

3.1. Оценка надежности систем защиты информации

3.1.1. Отказоустойчивость системы защиты. Понятие отказа

Ранее, исследуя вопросы защиты компьютерной информации, мы не затрагивали вопросы надежности системы защиты. При этом следует понимать, что интерпретация самого понятия «надежность системы защиты», а также основных параметров и характеристик надежности системы защиты, принципиально иная, чем для свойств надежности вычислительной системы.

В общем случае надежность вычислительной системы — это свойство системы выполнять возложенные на нее функции в течение заданного промежутка времени. Применительно к системе защиты информации от НСД **надежность — это свойство системы защиты обеспечивать защиту компьютерной информации от НСД в течение заданного промежутка времени.**

В общем случае отказ системы — это случайное событие, приводящее к невозможности выполнения системой в течение некоторого времени возложенных на нее функций.

Для системы защиты понятие «отказ» может трактоваться совсем иначе, чем при рассмотрении любого иного технического средства, т.к. с отказом связан не только переход системы защиты в состояние неработоспособности (данная составляющая «отказа» присуща любому техническому средству и нами в работе не рассматривается), но и обнаружение в системе защиты уязвимости.

Действительно, пусть обнаружена ошибка в механизме защиты, использование которой злоумышленником прямо, либо косвенно, приводит к

НСД. Это можно трактовать как отказ системы защиты, т.к. до тех пор, пока подобная ошибка не будет исправлена, система защиты не выполняет своих функций, и в данной ситуации существует канал несанкционированного доступа к информации.

Кстати говоря, неважно, что на какой-то момент времени для системы защиты известен только один-единственный канал НСД к информации. Этого вполне достаточно, чтобы говорить об отказе системы защиты в целом, т.к. характеристикой защиты является вероятность отражения ею атак на защищаемый объект, которая в данном случае имеет нулевое значение.

Определение.

Под **«отказом»** системы защиты будем понимать обнаружение злоумышленником канала НСД к информации (например, ошибки *в* ОС, либо в приложении, которая может привести к несанкционированному доступу). Под **«отказоустойчивостью»** — способность системы защиты обеспечивать свои функции (обеспечивать защиту компьютерной информации) в условиях обнаружения канала НСД к информации.

Отметим, что, вообще говоря, вопрос обеспечения надежности функционирования любой системы является одним из важнейших при ее проектировании. И совершенно непонятно, почему на сегодняшний день ему уделяется столь незначительное внимание при построении систем защиты (в данном случае под надежностью понимаем ее отказоустойчивость к обнаружению канала НСД к информации).

Действительно, представим себе, что вся сеть предприятия реализована на единой платформе — установлен единый тип ОС, вся защита обеспечивается встроенными в ОС средствами защиты. В этом случае с момента обнаружения канала НСД к информации и до момента его ликвидации (например, посредством установки некоторого дополнительного программного обеспечения, которое должен поставить разработчик ОС), можно считать систему защиты отказавшей, а сеть предприятия незащищенной.

Аналогично тому, как это выполнено в теории надежности вычислительных систем, в данном случае можно ввести еще два важнейших параметра, характеризующих систему защиты: интенсивность отказов — среднее число отказов в единицу времени, А; время восстановления системы после отказа T_e.

Под интенсивностью отказов системы защиты от НСД следует понимать интенсивность обнаружения в ней каналов НСД к информации в единицу времени.

Численные значения данного параметра могут быть получены на основании статистики угроз НСД, которая для современных универсальных ОС приведена выше.

При расчете надежности принимается, что интенсивность отказов является постоянной во времени величиной. Если предположить, что угрозы НСД

взаимонезависимы и любая i-я (i = 1,...,I) угроза носит катастрофический характер, предоставляя злоумышленнику несанкционированный доступ к информации, то интенсивность отказов системы защиты равна сумме интенсивностей угроз НСД к соответствующей системе защиты:

$$\lambda = \sum_{i=1}^{I} \lambda_i.$$

Тогда вероятность исправной работы системы защиты в течение произвольного интервала времени t определяется следующим образом:

$$p(t) = e^{-\lambda t}.$$

Соответственно, обратная величина интенсивности отказов системы равна среднему промежутку времени между двумя отказами и называется **временем наработки на отказ:**

$$T = 1/\lambda.$$

Интенсивность отказов системы определяется рядом параметров, в том числе, сложностью исследования защитных механизмов в системе, квалификацией злоумышленника и временным интервалом эксплуатации системы защиты.

Сложность исследования механизмов защиты зависит не только от качества разработки системы защиты (уровня квалификации разработчика), но и от доступности информации о системе защиты для исследования злоумышленником. С точки зрения доступности для исследования можно выделить следующие категории систем, характеризуемые возможностью обнаружения злоумышленником каналом НСД к информации:

- **Очень высокая** — некоммерческие средства (в том числе ОС), характеризуемые свободным доступом злоумышленника к их исходным кодам.
- **Высокая** -- широко используемые на практике коммерческие средства (в том числе ОС), характеризуемые отсутствием доступа злоумышленника к их исходным кодам.
- **Низкая** — ограниченно (по различным причинам) используемые на практике коммерческие средства (в том числе ОС), характеризуемые отсутствием доступа злоумышленника к их исходным кодам.
- **Очень низкая** — используемые на практике коммерческие средства, имеющие формализованные правила ограниченного распространения, характеризуемые отсутствием доступа злоумышленника к исходным кодам ПО. К этой категории, в первую очередь, относятся коммерческие средства защиты отечественных производителей.

Очень важно при построении системы защиты правильно учесть **квалификацию злоумышленника.** При этом в определенных случаях его квалификация может быть очень высокой, т.к. для обнаружения некорректно-

стей в реализации механизма защиты, либо ошибок, требуется проведение некоторого системного и архитектурного анализа защищаемого объекта и т.д. В других же случаях злоумышленник вполне может обойтись небольшим набором знаний и умений. Очевидно, что квалификация злоумышленника в большой мере определяется областью применения защищаемого объекта, т.е. ценностью той информации, которая обрабатывается защищаемым объектом.

Временной интервал эксплуатации системы защиты — также весьма важный аспект, влияющий на интенсивность отказов. Ведь понятно, что сначала злоумышленником выявляются наиболее очевидные каналы НСД к информации, которые соответствующим образом ликвидируются разработчиком. Соответственно, чем дольше эксплуатируется система, тем сложнее злоумышленнику найти канал НСД к информации. Однако это при условии, что вносимые разработчиком исправления, не приведут к новым, еще более очевидным некорректностям и ошибкам.

3.1.2. Время восстановления системы защиты. Коэффициент готовности

Ранее нами рассматривалась целесообразность усиления средств защиты в части увеличения параметра «среднее время наработки на отказ». Однако не менее (если не более) важным является другая составляющая надежности системы защиты — так называемое «время восстановления».

Интервал времени, в течение которого после возникновения отказа системы защиты обнаруженный канал НСД к информации устраняется, будем называть временем восстановления ($T_в$).

В общем случае время восстановления является случайной величиной. Однако принято учитывать его среднее значение. Характеризуется среднее время восстановления следующими компонентами:

- временем устранения соответствующего канала НСД к информации разработчиком ($T_у$);
- временем внедрения на защищаемый объект исправленной версии системы защиты, включая ее настройку ($T_{вн}$).

Очевидно, можем принять, что среднее время восстановления определяется временем устранения соответствующего канала НСД к информации разработчиком. Другой компонентой, ввиду ее относительной малости, можем пренебречь. Поэтому примем:

$$T_в = T_у$$

С учетом сказанного можем отметить, что в рассматриваемом случае среднее время восстановления -- это одна из основных характеристик надежности функционирования системы защиты. Определяется она тем,

насколько предприятие-разработчик системы защиты оперативно исправляет обнаруживаемые каналы НСД. Причем при использовании встроенных механизмов защиты этим разработчиком является сам разработчик ОС (приложения).

Отметим, что характеристика «время восстановления» объективно зависит от сложности системы. Действительно, зачастую исправление одной ошибки требует тестирования практически всех функциональных модулей системы заново, поскольку в сложной технической системе все функциональные модули сильно взаимосвязаны.

Кстати говоря, исправление одной ошибки может привести к появлению других ошибок. Поэтому исправление ошибок в ОС и сложных приложениях требует реализации соответствующей технологии внесения исправлений. Эта технология предполагает серьезное тестирование ПО практически по всем функциям, что может составлять месяцы. Например, для ОС семейства Windows данный параметр можно охарактеризовать, как половину среднего интервала времени между выходами в свет доработок ОС — Servise Pack.

Отметим, что с учетом сложности исправления ошибок, на практике разработчики сложных систем стараются не исправлять ошибки поодиночке, а приступать к тестированию системы уже по факту исправления некоторой совокупности ошибок (с учетом совокупности внесенных изменений).

При этом необходимо учитывать, что в течение всего времени восстановления можно считать систему защиты отказавшей, а защищаемый объект — незащищенным.

Следовательно, такая характеристика надежности системы защиты, как «время восстановления», может служить требованием к предприятию-разработчику системы защиты.

С позиций надежности эксплуатационные свойства системы защиты можно охарактеризовать **коэффициентом готовности:**

$$K_г = T/(T + T_в).$$

Коэффициент готовности, во-первых, характеризует долю времени, в течение которого система защиты работоспособна, а во-вторых, определяет вероятность того, что в любой произвольный момент времени система защиты работоспособна. Соответственно получаем долю времени, в течение которого объект находится в незащищенном состоянии, а также вероятность того, что в любой момент времени объект незащищен:

$$K_{нг} = 1 - K_г.$$

Проведем грубую оценку характеристики «коэффициент готовности», чтобы оценить, насколько критична характеристика «время восстановления» при построении системы защиты. Примем интенсивность обнаружения

ошибки, которая может привести к НСД к информации, равной 1 в год (практика показывает, что для ряда ОС и приложений она значительно выше). В табл. 3.1 показано изменение коэффициента готовности системы (вероятности нахождения системы в защищенном виде) при различных значениях характеристики времени восстановления.

Таблица 3.1

Изменение коэффициента готовности системы (вероятности нахождения системы в защищенном виде) при различных значениях характеристики времени восстановления

Время восстановления	3 месяца	1 месяц	2 недели	1 неделя	3 дня	1 день
Коэффициент готовности	0,8	0,92	0,96	0,98	0.992	0,997

3.1.3. Требования к системе защиты информации, исходя из отказоустойчивости

Требования к надежности вычислительной системы на практике определяются, как правило, значением коэффициента готовности не ниже 0,99 (в большинстве случаев -- 0,999 и выше). Соответственно, на основе представленных в табл. 1.5 исследований мы можем сделать вывод, что время восстановления системы защиты должно определяться днями, а не месяцами, как это происходит на практике при использовании встроенных в ОС (и приложения) защитных механизмов. При этом отметим, что нами исследовался наиболее благоприятный случай, когда интенсивность отказов принималась равной 1 в год. А если их будет два и более, какова будет вероятность защищенности системы?

Из сказанного могут быть сделаны следующие выводы:

1. Усиление средств защиты целесообразно для повышения надежности системы защиты — уменьшения времени восстановления системы при обнаружении канала НСД к информации. В этом случае следует говорить о целесообразности замещения всех встроенных механизмов защиты на механизмы систем защиты отечественной разработки. Возможность же существенного снижения характеристики «времени восстановления» при этом обусловливается двумя причинами: во-первых, объективной — система защиты как таковая имеет на порядки меньшую сложность, чем ОС в целом (исправление ошибки в ней потребует на столько же меньших трудозатрат, чем аналогичные исправления в ОС), во-вторых, субъективной — отечественные производители могут с большей оперативностью обеспечить взаимодействие с конечным потребителем средств защиты, в частности, при распространении исправленной версии ПО.
2. К разработчику системы защиты, при ее практическом использовании, должно выдвигаться требование к времени восстановления си-

стемы защиты — устранения каналов НСД к информации. Так как время восстановления системы защиты, с целью достижения высокого уровня ее отказоустойчивости, должно определяться днями (что на практике невозможно), то при построении системы защиты должны решаться вопросы резервирования.

3. Недопустимо использовать средства добавочной защиты, не поддерживаемые конкретным предприятием-разработчиком (в частности, свободно распространяемые и др.), а также распространяемые предприятием, не способным обеспечить необходимую оперативность восстановления системы защиты при обнаружении канала НСД к информации.

Таким образом, как видим, требования к времени восстановления системы защиты очень высоки даже при использовании на защищаемом объекте средств защиты отечественного производителя.

3.2. Задача и методы резервирования встроенных в ОС механизмов защиты для повышения отказоустойчивости системы защиты

Для увеличения надежности любой вычислительной системы применяется резервирование. Любое резервирование основывается на включении в состав системы защиты дополнительных средств. В нашем случае — дополнительных механизмов защиты. Причем в отличие от классических способов резервирования средств вычислительной техники, предполагающих параллельное включение резервного оборудования, однотипного с резервируемым оборудованием, в данном случае резервные механизмы (дополнительные механизмы защиты) должны включаться последовательно.

Определение.

Под резервированием механизмов защиты будем понимать последовательное включение в систему защиты дополнительных механизмов, реализующих те же функции защиты, что и основные механизмы, но иным способом и средствами.

Требование к реализации механизмов защиты различными способами и средствами обусловлено необходимостью резервного противодействия угрозе в случае преодоления злоумышленником основного механизма защиты. То есть одна и та же угроза должна распространяться либо на резервируемый, либо на резервный механизмы, в противном случае факт резервирования отсутствует, как таковой.

Таким образом, использование в системе защиты дополнительных механизмов можно рассматривать не только с целью расширения функций встроенных механизмов защиты, но и с целью их резервирования.

Если надежность системы защиты характеризуется вероятностью p безотказной работы за время t и определяется для встроенных средств защиты надежностью p_0:

$$P = p_0,$$

то при использовании дополнительных механизмов защиты, обеспечивающих резервирование встроенных механизмов, и характеризуемых надежностью p_1, имеем:

$$p = (1 - (1 - p_0)(1 - p_1)).$$

Заметим, что резервирование приводит не только к увеличению вероятности безотказной работы системы защиты, но и к снижению требований к времени восстановления.

В общем случае **можно выделить три режима резервирования системы защиты дополнительными механизмами защиты:**

- **Горячий резерв,** при котором основные и дополнительные механизмы защиты настроены и включены. В этом случае ограничений на время восстановления системы практически не накладывается (естественно, в разумных пределах). Этот подход обеспечивает наиболее высокий уровень защиты, т.к. при преодолении основного механизма защиты противодействие угрозе оказывает резервный механизм (в предположении, что угроза в равной мере не распространяется на основной и резервный механизмы защиты).

  Для систем защиты информации, критичной к НСД, следствием сказанного будет вывод о целесообразности резервной реализации основных механизмов защиты от НСД, дополнительными механизмами защиты с их включением в режиме «горячего резерва» совместно со встроенными механизмами защиты. При этом к основным механизмам защиты от НСД прежде всего относятся механизм идетификации и аутентификации, а также механизм управления доступом к ресурсам.
- **Активный холодный резерв,** при котором основной и дополнительный механизмы защиты настроены, но включен только один из них. В этом случае время восстановления определяется продолжительностью запуска резервной системы при отказе основной. Как правило, это время составляет несколько часов. Однако, по сравнению с горячим резервирование здесь достигается снижение влияния системы защиты на загрузку вычислительного ресурса защищаемого объекта.
- **Пассивный холодный резерв,** при котором только одно из средств защиты (основное или дополнительное) настроено и включено. В этом случае время восстановления определяется продолжительностью настройки и запуска резервной системы при отказе основной. Обычно это составляет от нескольких часов до несколько дней. Однако, по сравнению с активным холодным резервирование здесь достигается упрощение администрирования системы защиты, как при ее внедрении, так и в процессе функционирования.

Ранее нами рассматривались вопросы необходимости усиления механизмов зашиты, встроенных в ОС. Однако, на основании анализа, проведенного в данном разделе, можем сделать вывод, что наряду с этим важнейшей задачей усиления системы защиты следует считать резервирование встроенных в ОС и приложения механизмов защиты. Связано это с тем, что даже при потенциально максимальной оперативности предприятия-разработчика по устранению ошибок (выявленных каналов НСД к информации), только этим невозможно обеспечить сколько-нибудь высокую отказоустойчивость системы защиты (ее способность выполнять свои функции). Таким образом, по мнению автора, без резервирования механизмов защиты о гарантированной защите не приходится говорить в принципе.

3.3. Понятие добавочной защиты информации. Способы комплексирования механизмов защиты

3.3.1. Понятие встроенной и добавочной защиты

По способу реализации системы защиты компьютерной информации от НСД принято подразделять на встроенные и добавочные.

Определения.

Под **встроенной** понимается система защиты, механизмы которой встроены в соответствующее функциональное ПО (системное и прикладное), являются его неотъемлемой частью и реализуют дополнительную для данного ПО функцию — обеспечение защиты компьютерной информации.

Под **добавочной** понимается система защиты, основное функциональное назначение которой является решение задач защиты информации. Используется добавочная система совместно с функциональным ПО и имеет своей целью усиление защитных свойств встроенных в ПО механизмов.

Изначально средства добавочной защиты появились для незащищенных ОС, не обладающих втроенными механизмами защиты. Сначала это были системы для ОС семейства DOS, затем для ОС семейства Windows 9x. Подобные системы отечественной разработки сегодня достаточно широко представлены на рынке отечественных средств защиты информации. К ним могут быть отнесены системы защиты: «Secret Net 4.0», «Dallas Lock 4.1», «Аккорд-АМДЗ» (версии ПО 1.35, 1.95), «Спектр-Z», «Панцирь» v1.0 и другие. Понятно, что о способах комплексирования механизмов защиты, равно как и о возможности их резервирования, здесь говорить не приходится, т.к. используются только добавочные механизмы.

Примечание

В книге конкретные системы защиты как таковые не рассматриваются, а исследуются собственно подходы к их построению, Дело в том, что невозможно осуществить корректное сравнение средств добавочной защиты между собой в принципе, т.к. при этом необходимо учитывать не только, какие методы и функции защиты реализуются, но и то, каким образом они реализуются. Однако это является НОУ-ХАУ разработчика, а значит, по понятным причинам, информация об их уязвимости в открытой печати отсутствует. Поэтому сравнивать можно лишь общие подходы и технологии, реализуемые в системах защиты. При необходимости читатель может найти описание данных систем защиты на сайтах разработчиков.

С точки же зрения обоснованности подходов к защите ОС (в которых изначально разработчиком ОС не продумывались и не закладывались свойства защищенности) отметим, что на наш взгляд, построить гарантированную защиту ОС семейства Windows 9x/Me (не говоря уже об ОС семейства DOS) добавочными средствами крайне затруднительно. Связано это с тем, что практически невозможно выявить и предотвратить все скрытые возможности НСД, поскольку им не оказывается никакого противодействия на уровне ядра ОС. В данных случаях возрастает роль механизмов криптографической защиты.

Однако, во-первых, следует учитывать, что криптографическая защита не является панацеей. Причем это касается как противодействия несанкционированному удалению файлов, так и несанкционированного получения информации. Например, используя сниффер клавиатуры, можно перехватить все данные, в том числе и значение пароля, вводимых пользователем, т.е. еще до их криптографического преобразования. Во-вторых, осуществление криптографических преобразований не может не сказываться на загрузке вычислительного ресурса компьютера.

Поэтому на наш взгляд, с учетом появившихся более защищенных ОС семейства Windows, где многие скрытые угрозы НСД предотвращаются системными средствами, ОС семейства Windows 9x/Me, даже при использовании средств добавочной защиты, целесообразно применять лишь в некритичных к НСД приложениях. В данных же приложениях достаточным является использование средств защиты, основу которых составляет автоматическое («прозрачное» для пользователя) шифрование, либо иное преобразование данных при их сохранении на диске. К таким средствам относятся, например, Secret Disk, система защиты данных (СЗД) «Панцирь» и др. Кстати, для СЗД «Панцирь» также предусмотрено гарантированное удаление файлов и очистка остаточной информации на диске или внешнем носителе.

Другое дело ОС Windows NT/2000/XP, а также современные ОС семейства UNIX, обладающие развитыми встроенными механизмами защиты. Они уже обладают реализацией ядра ОС с заложенными принципами противодействия как явным, так и скрытым угрозам. Поэтому для этих ОС уже становится актуальным выбор концептуального решения (определения подходов) при построении системы добавочной защиты, а также методов комплексирования механизмов встроенной и добавочной защиты.

3.3.2. Подходы к построению добавочной защиты

С учетом сказанного можно признать обоснованными следующие подходы к построению добавочных средств защиты для защищенных ОС (т.е. обладающих встроенными механизмами защиты):

1. Добавление механизмов защиты, отсутствующих в ОС, и усиление добавочными средствами штатных встроенных механизмов защиты.
2. Реализация всех механизмов защиты добавочными средствами и их применение наряду со встроенными механизмами.

Очевидно, что первому подходу присущи два недостатка: во-первых, невозможность решения концептуальных вопросов защиты информации, рассмотренных ранее, во-вторых -- невозможность резервирования основных механизмов защиты. При этом к основным механизмами защиты от НСД относятся механизмы идентификации и аутентификации пользователя при входе в систему, а также механизмы управления доступом к ресурсам.

Что касается второго подхода, то он, возможно, характеризуется некоторой избыточностью. Вместе с тем он позволяет в полном объеме решать рассматриваемые проблемы защиты информации

В качестве замечания следует отметить, что, на наш взгляд, в критичных приложениях по изложенным ранее причинам недопустимо использование свободно распространяемых (не коммерческих) средств защиты, а также свободно распространяемых защищенных систем (в частности ОС) без применения средств добавочной защиты. Говоря же о средствах добавочной защиты, прежде всего следует иметь в виду использование коммерческих отечественных продуктов, сертифицированных по принятым требованиям информационной безопасности.

3.3.3. Требования и задачи, которые должна выполнять система добавочной защиты от НСД

Обобщая все сказанное ранее, сформулируем необходимые требования к системе добавочной защиты от НСД для защищенных ОС (ОС, обладающих встроенными механизмами защиты):

1. Механизмы добавочной защиты должны применяться наряду со встроенными в ОС и приложения механизмами защиты.
2. Механизмы добавочной защиты должны применяться с целью усиления защитных свойств встроенных механизмов (задачи добавочной защиты в части усиления защитных свойств встроенных механизмов рассмотрены выше).
3. Механизмы добавочной защиты должны применяться с целью контроля корректности функционирования встроенных механизмов за-

щиты, обеспечивая противодействие ошибкам и закладкам в системном и прикладном ПО.

4. Механизмы добавочной защиты должны применяться с целью резервирования встроенных механизмов, прежде всего, основных механизмов противодействия НСД. Это будет приводить к повышению надежности системы защиты информации.

Итак, мы показали, что добавочные механизмы защиты необходимы и определили общий круг решаемых ими задач. Вместе с тем следует отметить, что с включением в систему защиты добавочных механизмов связано и появление новых угроз — угроз именно механизмам добавочной защиты. Эти угрозы связаны с тем, что механизмы добавочной защиты по ряду причин (в том числе и ввиду необходимости выполнять разработчиком систем добавочной защиты требования лицензионного соглашения для ОС) реализуются не на уровне ядра, а на уровне драйверов и приложений. Т.е. на них не распространяется защита системными средствами, реализуемая ядром ОС. Они являются внешними по отношению к ОС.

Опасность этого проиллюстрируем примером. Для ОС Windows NT/2000/XP для запуска режима загрузки системы Safe Mode достаточно осуществить авторизацию пользователя. В этом режиме пользователем возможна выборочная загрузка системы, при которой часть драйверов и приложений по усмотрению пользователя может не загружаться. И хотя это не распространяется на встроенные средства защиты (они загружаются всегда), пользователем могут быть не загружены драйверы и приложения добавочных средств защиты.

Таким образом, при включении в систему механизмов добавочной защиты дополнительно должны быть решены задачи противодействия угрозе перевода добавочной защиты в пассивное состояние. То есть должно осуществляться противодействие загрузке системы без механизмов добавочной системы. Кроме того, должно быть реализовано противодействие удалению механизмов добавочной защиты, а также переводу их в пассивное состояние при функционировании системы.

Итак, теперь мы можем сформулировать еще одну задачу, которая должна решаться при реализации в системе добавочной защиты информации от НСД:

5. Механизмы добавочной защиты должны оказывать противодействие как явным, так и скрытым угрозам их переводу в паесивное состояние в процессе функционирования системы, либо загрузке системы без механизмов добавочной защиты.

Кроме того, отметим, что ранее рассматривались требования к встроенным в ОС механизмам защиты и их возможности в предположении, что данные механизмы присутствуют в системе. Вместе с тем, существует возможность загрузки на защищаемом компьютере другой ОС, в частно-

сти, с внешнего носителя. Функции защиты загрузки системы уже не входят в задачи ОС. В частности данная задача решается BIOS, с реализацией которой, как было показано выше, также связаны некоторые угрозы. Поэтому добавочные средства защиты также могут использоваться и с целью дополнительной защиты загрузки системы, т.е. можем сформулировать еще одно требование к механизмам добавочной защиты:

6. Механизмы добавочной защиты должны применяться с целью усиления защитных свойств от несанкционированной загрузки ОС, в частности, реализуемых отдельным программным средством BIOS.

Мы рассмотрели основные задачи механизмов добавочной защиты, однако, кроме них для добавочных механизмов защиты могут рассматриваться также и некоторые дополнительные задачи. Эти задачи связаны с корректным использованием приложений и встроенных в приложения механизмов защиты, с решением задачи централизации схемы администрирования защитных механизмов в иерархической информационной системе, с решением задач упрощения настройки защитных механизмов и др. Все это будет рассмотрено в последующих главах.

3.4. Вопросы оценки эффективности и проектирования системы защиты

Итак, ранее нами было приведено обоснование целесообразности применения добавочных средств защиты при построении современных информационных систем. При этом были выделены две группы требований к защищенности, которые должны учитываться при построении системы защиты — формализованные требования и требования, формулируемые на основе существующей статистики угроз. Невозможность в общем случае формализовать требования второй группы не позволяет и формализовать сравнительный анализ систем защиты, отнесенных к одному классу защищенности (в соответствии с классификацией нормативных документов).

В частности, две системы защиты, отнесенные к одному классу защищенности, могут принципиально различаться по своим возможностям. При этом необходимо отметить, что в случае предполагаемой идентичности реализации формализованных требований, для них важнейшей характеристикой становится уровень квалификации их разработчиков.

Примечание

Предположение идентичности обычно не соответствует действительности, т.к. в нормативных документах не указывается, каким способом должен быть реализован каждый механизм защиты. Поэтому существующие системы добавочной защиты, отнесенные к одному классу защищенности, принципиально различаются и в реализации формализованных требований.

Понятно, что вопросы оценки эффективности и вопросы проектирования системы защиты тесно связаны, т.к. в их основе лежит единый математический аппарат решения соответствующей оптимизационной задачи.

Рассмотрим возможный подход к проектированию (оценке эффективности) системы защиты. Отметим, что в данном случае мы не разделяем механизмы защиты на встроенные и добавочные, поскольку это уже вопрос реализации тех требований к набору и функциям механизмов защиты, которые будут сформулированы в результате проектирования системы защиты. При этом будем понимать, что необходимо решать задачу многокритериальной оптимизации, т.к. система защиты в общем случае характеризуется целым рядом параметров, которые должны учитываться при ее проектировании.

3.4.1. Общий подход к оценке эффективности системы добавочной защиты

Критерий и параметры проектирования оптимальной системы защиты

Будем оценивать **защищенность системы** (Z) количественно в зависимости от стоимости защищаемой информации, вероятности взлома, стоимости самой системы защиты, производительности системы:

$$Z = f(C_{инф}, p_{взл}, Ц_{СЗИ}, П),$$

где $C_{инф}$ -- стоимость защищаемой информации;
$p_{взл}$ - вероятность взлома;
$Ц_{СЗИ}$ — стоимость СЗИ;
$П$ - производительность системы.

С учетом введенного понятия защищенности системы оптимизационная задача состоит в обеспечении максимального уровня защищенности (как функции стоимости защищаемой информации и вероятности взлома) при минимальной стоимости системы защиты и минимальном влиянии ее на производительность системы:

$$Z^{opt} = max\, Z(C_{инф}, p_{взл}, Ц_{СЗИ}, П).$$

С учетом сказанного может быть сделан важный вывод о многокритериальном характере задачи проектирования системы защиты. При этом, кроме обеспечиваемого уровня защищенности, должен учитываться еще ряд важнейших характеристик системы. Например, обязательно должно учитываться влияние системы защиты на загрузку вычислительного ресурса защищаемого объекта.

Примечание

В общем случае Загрузка вычислительного ресурса определяется количеством прикладных задач, решаемых объектом в единицу времени.

Исходные параметры для задачи проектирования системы защиты, а также возможности сведения задачи к однокритериальной проиллюстрированы рис. 3.1.

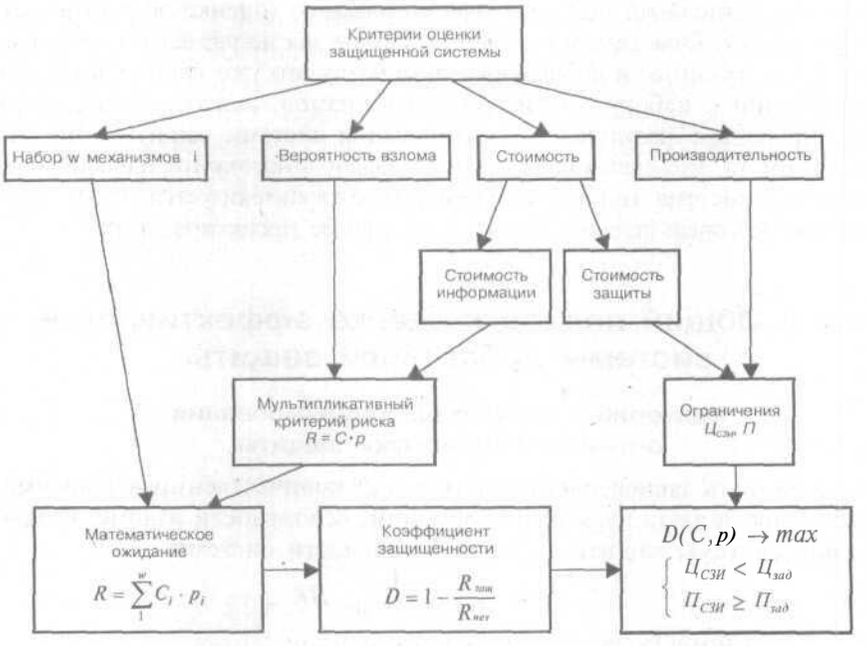


Рис. 3.1. *Критерии оценки защищенности*

Защищенность системы с точки зрения риска

Рассмотрим защищенность системы с точки зрения риска. Заметим, что использование теории рисков для оценки уровня защищенности на сегодняшний день является наиболее часто используемым на практике подходом. *Риск (R)* — это потенциальные потери от угроз защищенности:

$$R(p) = C_{инф} \cdot p_{взл},$$

По существу, параметр риска здесь вводится как мультипликативная свертка двух основных параметров защищенности.

С другой стороны, можно рассматривать риск как потери в единицу времени:

$$R(\lambda) = C_{инф} \cdot \lambda_{взл},$$

где $\lambda_{взл}$ — интенсивность потока взломов (под взломом будем понимать удачную попытку несанкционированного доступа к информации).

Эти две формулы связаны следующим соотношением:

$$p_{взл} = \frac{\lambda_{взл}}{\Lambda},$$

где Л - общая интенсивность потока несанкционированных попыток доступа злоумышленниками к информации.

Основной критерий защищенности. Общее решение задачи проектирования оптимальной системы защиты

В качестве основного критерия защищенности будем использовать *коэффициент защищенности (D),* показывающий относительное уменьшение риска в защищенной системе по сравнению с незащищенной системой.

$$D\% = \left(1 - \frac{R_{защ}}{R_{нез}}\right) \times 100\% \qquad (1\ 1)$$

где $R_{защ}$ — риск в защищенной системе;

$R_{нез}$ -- риск в незащищенной системе.

Таким образом, в данном случае задача оптимизации выглядит следующим образом:

$$\begin{cases} D(C_{шф}, p\_\_) \to max; \\ Ц_{СЗИ} \to min; \\ П_{СЗИ} \to max. \end{cases}$$

Для решения этой задачи сведем ее к однокритериальной посредством введения ограничений. В результате получим:

$$\begin{cases} D\left(C_{шф}, p_{взл}\right) \to max; \\ Ц_{СЗИ} \le Ц_{зад}; \\ П_{сзи} \ge П_{зад}, \end{cases}$$

где $Ц_{зад}$ и $П_{зад}$ — заданные ограничения на стоимость системы защиты и производительность системы.

Целевая функция выбрана исходя из того, что именно она отражает основное функциональное назначение системы защиты -- обеспечение безопасности информации.

Производительность системы $П_{СЗИ}$ рассчитывается с применением моделей и методов теории массового обслуживания и теории расписаний (в зависимости от того, защищается ли система оперативной обработки, либо реального времени). На практике возможно задание ограничения по производительности (влияние на загрузку вычислительного ресурса защищаемой системы) не непосредственно в виде требуемой производительнос-

ти системы, а как снижение производительности ($d\Pi_{СЗИ}$)информационной системы от установки системы защиты. В этом случае задача оптимизации будет выглядеть следующим образом:

$$\begin{cases} D(C,p) \to max\ ; \\ Ц_{СЗИ} \to min\ ; \\ d\Pi_{СЗИ} \to min\ , \end{cases}$$

или после сведения ее к однокритериальной:

$$\begin{cases} D(C,p) \to max\ ; \\ Ц_{СЗИ} \le Ц_{зад}\ ; \\ d\Pi_{СЗИ} \le d\Pi_{зад}\ , \end{cases}$$

где $Ц_{зад}$ и $d\Pi_{зад}$ — заданные ограничения на стоимость системы защиты и снижение производительности.

Заметим, что на наш взгляд именно такой принцип сведения задачи к однокритериальнои целесообразен, т.к. в любом техническом задании на разработку системы защиты указывается, в какой мере система защиты должна оказывать влияние на производительность системы. Как правило, внедрение системы защиты не должно снижать производительность системы более чем на 10%. Кроме того, обычно вводится ограничение на стоимость системы защиты.

Если рассчитанное значение коэффициента защищенности (D) не удовлетворяет требованиям к системе защиты, то в допустимых пределах можно изменять заданные ограничения и решить задачу методом последовательного выбора уступок (рассмотрен ниже). При этом задается приращение стоимости и снижение производительности:

$$Ц^{*}_{зад} = Ц_{зад} + \Delta Ц,$$

$$\Pi^{*}_{зад} = \Pi_{зад} - А/7 \text{ или } d\Pi^{*}_{зад} = d\Pi_{зад} + МП.$$

В таком виде задача решается в результате реализации итерационной процедуры путем отсеивания вариантов, не удовлетворяющих ограничительным условиям, и последующего выбора из оставшихся варианта с максимальным коэффициентом защищенности.

Теперь выразим коэффициент защищенности через параметры угроз. В общем случае в системе присутствует множество видов угроз. В этих условиях зададим следующие величины:

w - количество видов угроз, воздействующих на систему;

$C_i\left(i = \overline{1,w}\right)$ — стоимость (потери) от взлома i-того вида;

$\lambda_i\left(i = \overline{1,w}\right)$ — интенсивность потока взломов i-того вида, соответственно;

$Q_i\left(i=\overline{1,w}\right)$ — вероятность появления угроз i-того вида в общем потоке попыток несанкционированного доступа к информации, причем $Q_i = \frac{\lambda_i}{\Lambda}$;

$p_i\left(i=1,\overline{w}\right)$ -- вероятность отражения угроз i-того вида системой защиты.

Соответственно, для коэффициента потерь от взломов системы защиты имеем:

$$R(p)=\sum_1^w R_i(p)=\sum_1^w C_i \cdot p_{взл\,i},$$

где $R_i(p)$ — коэффициент потерь от взлома i-того типа; показывает, какие в среднем потери приходятся на один взлом i-того типа.

Для незащищенной системы $p_{угр\,i}=Q_i$, для защищенной системы $p_{угр\,i}=Q_i\cdot(1-p_i)$.

Соответственно, для коэффициента потерь от взломов системы защиты в единицу времени имеем:

$$R(\lambda)=\sum_1^w R_i(\lambda)=\sum_1^w C_i \cdot \lambda_{угр\,i},$$

где Д.(я) — коэффициент потерь от взломов i-того типа в единицу времени.

Для незащищенной системы $\lambda_{угр\,i}=\lambda_i$, для защищенной системы $\lambda_{угр\,i}=\lambda_i\cdot(1-p_i)$. Соответственно, из (1.1) имеем:

$$D=1-\frac{\sum_1^w C_i\cdot Q_i\cdot(1-p_i)}{\sum_1^w C_i\cdot Q_i}=1-\frac{\sum_1^w C_i\cdot\lambda_i\cdot(1-p_i)}{\sum_1^w C_i\cdot\lambda_i}. \qquad (1.2)$$

Если в качестве исходных параметров заданы вероятности появления угроз Q_i, то коэффициент защищенности удобно считать через вероятности появления угроз. Если же в качестве исходных параметров заданы интенсивности потоков угроз λ_i, то, естественно, коэффициент защищенности считается через интенсивность.

3.4.2. Способы задания исходных параметров для оценки защищенности

Очевидно, что при использовании любого математического метода проектирования системы защиты необходимо задавать определенные исходные параметры для оценки ее защищенности. Однако именно с этим связаны основные проблемы формализации задачи синтеза системы защиты. Поэтому мы отдельно рассмотрим основные пути решения данной задачи.

Способы задания вероятностей и интенсивностей угроз

Основной проблемой проведения количественной оценки уровня защищенности является задание входных параметров для системы защиты — вероятностей и интенсивностей угроз. Рассмотрим возможные способы задания вероятностей и интенсивностей угроз.

1. Метод статистической оценки $\lambda_i(Q_i)$ и p_i . Основным способом задания интенсивностей потоков угроз λ_i (вероятностей угроз Q_i) и вероятностей взломов p_i является получение этих значений на основе имеющейся статистики угроз безопасности информационных систем, в которых реализуется система защиты. Если существует статистика для аналогичной информационной системы, то задавать исходные параметры для оценки защищенности можно на ее основе. При этом желательно, чтобы сходные информационные системы эксплуатировалась на предприятиях со сходной спецификой деятельности.

Заметим, что статистика угроз периодически публикуется достаточно авторитетными изданиями, т.е. всегда существуют исходные данные для использования данного подхода для большинства приложений средств защиты информации. Обычно эта статистика доступна в Интернете на сайтах специализированных организаций.

Если же необходимая статистика по угрозам безопасности отсутствует, то можно воспользоваться одним из других подходов, описанных далее.

2. Оптимистически-пессимистический подход. В рамках данного подхода предусмотрено два разных способа.

Первый способ — **это способ равных интенсивностей** $\forall\lambda_i = a, a = const.$ При этом способе для расчета защищенности константа a может быть выбрана любой. В формуле (1.2) она будет вынесена за скобки и в конечном итоге сократится, так что защищенность в данном случае будет зависеть только от потерь:

$$D = 1 - \frac{\sum_1^w C_i \cdot \lambda_i \cdot (1 - p_i)}{\sum_1^w C_i \cdot \lambda_i} = 1 - \frac{\sum_1^w C_i \cdot \alpha \cdot (1 - p_i)}{\sum_1^w C_i \cdot \alpha} =$$

$$= 1 - \frac{\alpha \cdot \sum_1^w C_i \cdot (1 - p_i)}{\alpha \cdot \sum_1^w C_i} = 1 - \frac{\sum_1^w C_i \cdot (1 - p_i)}{\sum_1^w C_i}.$$

Второй способ — это **способ пропорциональности потерям** $\lambda_i = a \cdot C_i, a = const.$ При этом способе предполагается, что чем больше потери от взлома, тем чаще осуществляются попытки несанкционированного доступа к этой информации. То есть интенсивности потоков угроз прямо пропорциональны потерям. В этом случае защищенность будет зависеть от квадрата потерь:

$$D = 1 - \frac{\sum_{1}^{w} C_i \cdot \lambda_i \cdot (1 - p_i)}{\sum_{1}^{w} C_i \cdot \lambda_i} = 1 - \frac{\sum_{1}^{w} C_i \cdot \alpha \cdot C_i \cdot (1 - p_i)}{\sum_{1}^{w} {}_{i} \cdot \alpha \cdot C_i} =$$

$$= 1 - \frac{\alpha \cdot \sum_{1}^{w} C_i^2 \cdot (1 - p_i)}{\alpha \cdot \sum_{1}^{w} C_i^2} = 1 - \frac{\sum_{1}^{w} C_i^2 \cdot (1 - p_i)}{\sum_{1}^{w} C_i^2}.$$

3. Метод экспертной оценки. Экспертная оценка исходных параметров для расчета защищенности может осуществляться с использованием так называемой **дельфийской группы.** Дельфийская группа — это группа экспертов, созданная в целях сбора информации из определенных источников по определенной проблеме.

При этом необходимо задать лингвистический словарь возможных оценок экспертов, определить набор вопросов и условных значений квалификаций отдельных экспертов. После определения всех входных переменных производится поочередный опрос каждого эксперта. После опроса всех экспертов с учетом их квалификации определяется общая оценка группы и согласованность (достоверность) ответов для каждого вопроса.

Эксперт оценивает эффективность (вероятность) отражения угроз элементами защиты p_i и вероятность появления угроз Q_i. Вероятности эксперт задает лингвистическими оценками: отлично, хорошо, удовлетворительно, плохо, не отражает; вероятно, близко к нулю, близко к единице, весьма вероятно и т.п. Затем эти лингвистические оценки при помощи словаря переводятся в числа p_i и Q_i в диапазоне [0; 1].

Для задания вероятности появления угрозы возможна оценка вероятности появления угрозы i-того вида в общем потоке угроз:

$$Q_i = \frac{\lambda_i}{\sum \lambda_i}.$$

Исходя из заданной квалификации экспертов, рассчитываются их веса (значимость) в группе по формуле:

$$k_e = \frac{s_e}{\sum s_e},$$

где s_e - - квалификация эксперта, задаваемая в некотором диапазоне, например, от 0 до 10 в зависимости от опыта, образования и других качеств эксперта.

Затем оценки суммируются с учетом весов экспертов:

$$p_i = \sum p_{ie} \cdot k_e,$$

$$Q_i = \sum Q_{ie} \cdot k_e,$$

где p_{ie} и Q_{ie} — оценка вероятностей отражения и появления угроз, сделанные одним экспертом;

k_e - «вес» эксперта в группе.

После расчета общей оценки всей группы рассчитывается согласованность ответов, которая может использоваться для оценки достоверности результатов. Согласованность рассчитывается при помощи среднеквадратического отклонения и выражается в процентах.

Максимальная согласованность достигается при одинаковых значениях оценок экспертов и в этом случае равняется 100%. Минимальная согласованность достижима при максимальном разбросе оценок экспертов.

Способы задания стоимости потерь

Важнейшей характеристикой защищаемого объекта (как следствие, и системы защиты) является стоимость потерь от взлома. Рассмотрим возможные способы задания стоимости потерь.

1. Стоимость похищенной/искаженной/утерянной информации.

Исходные данные:

c_i*[руб./бит]* ..... удельная цена информации;

v[бит/с].....скорость получения/искажения/уничтожения информации;

t[c].....время нахождения субъекта в системе;

V_i*[бит]*.....объем информации.

$$C_i = min(c_i \cdot v \cdot t, c_i \cdot V_i).$$

***2.* Затраты от невозможности получения доступа к информации.**

Исходные данные:

c_i*[руб./бит]*.....удельная цена недоступности информации;

t[c] ..... время восстановления системы.

$$C_i = c_i \cdot t.$$

Способы задания соответствия между параметрами угроз, параметрами защищаемых объектов и параметрами элементов защиты

Существуют различные классификации угроз:

- » по принципам и характеру воздействия на систему;
- » по используемым техническим средствам;
- * по целям атаки и т.п.

Очевидно, что стоимость потерь С. удобнее задавать для угроз, классифицированных по целям атаки. Что касается характеристики интенсивности угроз, то она определяется с помощью средств аудита и сетевого мониторинга, которые различают угрозы по принципам и характеру воздействия на систему (механизму атаки, способу проникновения). Вероятность отражения угрозы средствами защиты *р.* определяется *в* соответствии с теми механизмами, которые реализованы в каждом средстве. Причем каждый из механизмов в общем случае может отражать несколько видов атак.

Таким образом, необходимо задавать соответствие между всеми этими параметрами (см. рис. 3.2). Для успешного приведения в соответствие различных параметров оценки защищенности необходимо корректное построение модели нарушителя. В этой модели должны быть отражены практические и теоретические возможности нарушителя, его априорные знания, время и место действия.

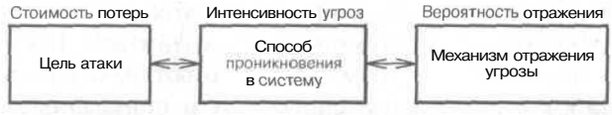


Рис. 3.2. *Взаимозависимость параметров защиты*

Задание соответствия между стоимостью потерь и интенсивностью угроз

Задание соответствия между стоимостью потерь и интенсивностью угроз можно осуществлять следующим образом:

1. Статистический подход. Статистический подход является основным, как обладающий большей достоверностью. Из анализа статистики можно выявить вероятности нанесения определенных видов ущерба при определенных видах взломов. Однако на практике далеко не всегда подобная статистика существует, в частности, при внедрении новых технологий защиты информации, новых версий ОС или приложений и т.д., т.к. для ее сбора требуется некоторое время. В этом случае может использоваться пессимистический подход.

2. Пессимистический подход. Если не имеется достаточной статистики, можно воспользоваться другим способом. Будем считать, что при проникновении в систему злоумышленник наносит наибольший вред, какой он только может причинить.

При задании соответствия между интенсивностью угроз и вероятностью их отражения нужно учитывать, что, если в системе реализовано несколько механизмов, отражающих некоторую атаку, вероятность преодоления защиты рассчитывается следующим образом.

Если p_k есть вероятность отражения i-той угрозы каждым средством защиты, то вероятность взлома системы ($\overline{p}_i$) будет:

$$\overline{p}_i = \prod_k (1 - p_k),$$

а вероятность отражения угрозы системой защиты

$$A = 1 - A \cdot$$

3.4.3. Особенности проектирования системы защиты на основе оценки защищенности системы

Этапы проектирования системы защиты

Задача проектирования системы защиты принципиально отличается от задач проектирования иных информационных систем. Дело в том, что проектирование осуществляется с учетом статистических данных об уже существующих угрозах. Однако, в процессе функционирования системы защиты поле угроз может принципиально измениться. В частности, это связано с тем, что многие угрозы предполагают нахождение злоумышленниками ошибок в реализации системных и прикладных средств, которые могут быть неизвестны на момент создания системы защиты, но должны быть учтены в процессе ее функционирования.

Поэтому проектирование системы защиты — процедура итерационная, в общем случае предполагающая следующие этапы (см. рис. 3.3):

- проектирование первоначальной системы защиты (исходный вариант);
- анализ защищенности на основе статистических данных, полученных в процессе эксплуатации системы защиты;
- модификация «узких мест» системы защиты (настройка/замена/дополнение отдельных механизмов защиты информации).

Рис. 3.3. *Иллюстрация проектирования системы защиты (СЗИ)*

После модификации «узких мест» происходит возврат к эксплуатации системы защиты и накопление статистической информации.

Этапы оценки защищенности и выбора оптимального варианта системы защиты

Оценка защищенности с учетом приведенных выше расчетных формул и выбор оптимального варианта системы защиты (необходимого набора механизмов защиты при разработке новой системы) осуществляется следующим образом:

1. Расчет параметров C_i, λ_i, p_i для оценки защищенности по исходным данным, полученным статистическим или, в случае недостатка статистики, одним из приведенных выше способов (оптимистически-пессимистический подход, метод экспертной оценки).
2. Расчет критериев защищенности D, $Ц_{СЗИ}$, $П_{СЗИ}$ ($dП_{СЗИ}$) для каждого варианта системы защиты (набора механизмов защиты при разработке новой системы).
3. Выбор системы защиты (набора механизмов защиты при разработке системы) с максимальным коэффициентом защищенности D, удовлетворяющей ограничениям по стоимости $Ц_{СЗИ}$ и производительности $П_{СЗИ}$.
4. Анализ изменения коэффициента защищенности dD при задании приращений для критериев $Ц_{СЗИ}$ и $dП_{СЗИ}$ методом последовательного выбора уступок с оценкой целесообразности выбора системы, удовлетворяющей новым ограничениям.

Последовательность задач, решаемых при проектировании системы защиты. Требование непрерывности проектирования

Последовательность задач, решаемых при проектировании системы защиты проиллюстрирована на рис. 3.4.

Рис. *3.4.* *Последовательность задач, решаемых при проектировании системы защиты*

С учетом сказанного ранее можем сделать следующий важный вывод: **проектирование системы защиты — это непрерывный процесс, осуществляемый в течение всего жизненного цикла системы, предполагающий исходное проектирование системы по расчетным значениям параметров и последующую ее модификацию (доработку), основанную на непрерывном анализе текущего состояния обеспечиваемого системой уровня защищенности средствами мониторинга с учетом меняющегося поля угроз.**

Модификация «узких мест» системы защиты предполагает выявление новых угроз и анализ их отра-

жения механизмами защиты, присутствующими в системе. При необходимости следует либо модифицировать существующие механизмы, либо встраивать новые.

Таким образом, с учетом сложности формализации задачи (особенно при задании исходных параметров), можно выделить характерную особенность проектирования системы защиты. В общем случае она заключается в невозможности реализовать гарантированную защиту информационной системы при условии ее проектирования исключительно по расчетным значениям параметров без последующего мониторинга защищенной системы. При этом в задачи мониторинга входит сбор статистики для задания параметров конкретной системы защиты, а также ее повторный расчет в рамках полученной статистики. Результатом повторного расчета будет обоснование оптимальности принятых на этапе проектирования технических решений и их соответствующая корректировка.

3.4.4. Метод последовательного выбора уступок

Качественная зависимость изменения основных параметров, характеризующих систему защиты, от ее сложности — используемого набора механизмов защиты, представлена на рис. 3.5.

Проанализировав характер зависимостей от сложности системы, можем сказать, что стоимость системы защиты возрастает неограниченно, а производительность снижается в пределе до нуля.

В то же время кривая коэффициента защищенности (D) стремится к предельному значению — к единице (100%) и в некоторый момент достигает насыщения. Это в свою очередь приводит к тому, что при дальнейшем нарастании сложности (и, соответственно, увеличении цены, а также снижении производительности) увеличение коэффициента защищенности происходит незначительно.

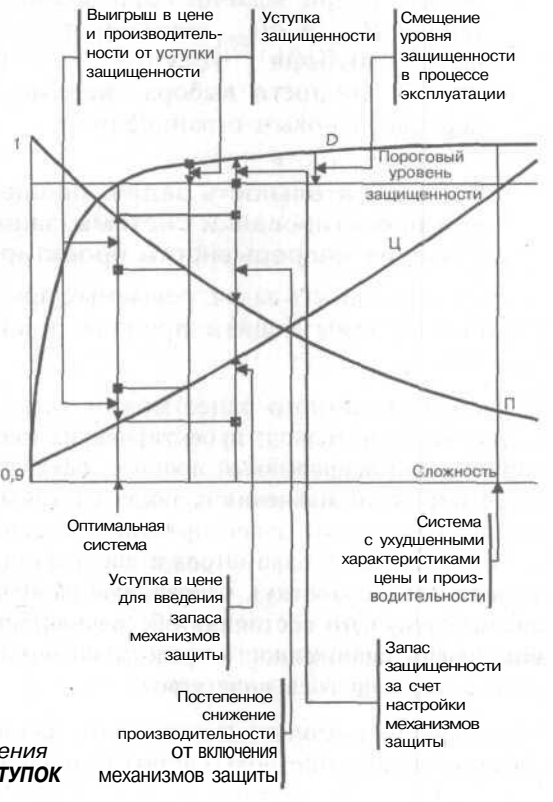


Рис. 3.5. *Иллюстрация применения **Метода последовательного выбора УСТУПОК***

Следовательно, при проектировании системы защиты, параметры защищенности которой расположены в области насыщения, целесообразно проанализировать параметры альтернативных вариантов. То есть целесообразно исследовать возможность использования менее сложных систем защиты и, задав некоторый промежуток снижения коэффициента защищенности *(dD),* выбрать систему, уровень защищенности которой удовлетворяет полученному *(D — dD).* Конечно, если таковые имеются. При этом может быть получен ощутимый выигрыш в цене и производительности.

В этом и состоит применение известного метода последовательных уступок при выборе оптимальной системы защиты (набора реализуемых механизмов защиты при ее проектировании). Этот метод, как уже упоминалось, подразумевает сведение многокритериальной задачи оптимизации к однокритериальной.

Метод последовательных уступок представляет собою итерационную человеко-машинную процедуру, используя которую разработчик, давая допустимые приращения одним параметрам (в частности, задавая снижение коэффициента защищенности), анализирует изменение других, принимая решение о допустимости вводимых уступок.

3.4.5. Методы проектирования системы защиты с избыточными механизмами защиты

Необходимость избыточных механизмов в системе защиты

Выше мы особо отмечали одну принципиальную особенность функционирования системы защиты. Суть ее заключается в том, что коэффициент защищенности непрерывно снижается в процессе функционирования защищенной системы. Это связано с накоплением информации злоумышленником о системе защиты, а также с накоплением статистики об ошибках реализации системных и прикладных средств.

Возникает следующая проблема. Если строить систему с требуемым уровнем защищенности (учитывающим текущую статистику угроз), то через небольшой промежуток времени функционирования защищаемой системы систему защиты потребуется проектировать вновь, что связано с большими накладными расходами. Поэтому ранее был сделан вывод о необходимости разработки системы защиты с учетом не только существующей статистики угроз, но и с учетом потенциально возможных (неизвестных) угроз. Тем не менее приходится учитывать тот факт, что наличие запаса по параметрам защищенности приводит к значительным накладным расходам, прежде всего, по параметрам производительности.

Что касается стоимости, то лучше раз создать систему с избыточными механизмами защиты и заплатить изначально больше, чем практически сразу после ее внедрения вновь проектировать систему защиты, т.е. вновь

оплачивать все расходы, связанные с этим проектированием, да еще думать об очередном ее внедрении в функционирующую систему, что, как правило, протекает весьма болезненно.

Из сказанного можем сделать важнейший вывод: **с целью увеличения жизненного цикла системы защиты (без необходимости ее модификации) разработчикам следует ориентироваться не только на существующую статистику угроз, но и закладывать технические решения, позволяющие противодействовать неизвестным на момент проектирования системы защиты угрозам (потенциальным угрозам).**

Обратите внимание на принципиальную разницу между подходами резервирования и подходами включения избыточных механизмов. При резервировании, рассмотренном нами ранее в п. 3.2, механизмы защиты дублируются такими же механизмами (решают те же функциональные задачи), но реализуются иным способом. Резервный призван заменить основной механизм при выходе его из строя (при обнаружении его уязвимости к атакам).

Что касается включения избыточных механизмов, то здесь рассматривается добавление новых механизмов с иными, чем у механизмов, реализованных в системе защиты, свойствами (решающих частично, либо полностью иные функциональные задачи). Другое дело, что данные возможности на первых этапах функционирования системы защиты могут быть не востребованными. Поэтому, чтобы не загружать излишне вычислительные ресурсы системы, их рекомендуется не активизировать – благодаря избыточным механизмам реализуется запас защищенности.

Подходы к проектированию систем защиты, обладающих избыточными механизмами. Метод динамического анализа

Классификация возможных подходов к проектированию рассматриваемых систем защиты представлена на рис. 3.6. Альтернативные подходы составляют **метод необходимого минимума и метод полного перекрытия.**

Метод необходимого минимума состоит в поиске варианта построения системы по характеристике требуемого уровня защищенности. Метод полного перекрытия состоит в исходной реализации избыточности механизмов, что изначально позволяет построить систему защиты с большим запасом в характеристике защищенности.

Как отмечалось, данные подходы характеризуются следующими недостатками:

- Метод необходимого минимума с учетом специфики функционирования рассматриваемого класса систем позволяет построить систему практически без запаса защищенности, что потребует ее достаточно скорого проектирования заново.
- Метод полного перекрытия связан с существенными затратами, прежде всего в производительности системы.

Рис. 3.6. *Классификация методов проектирования системы защиты*

Рассмотренные недостатки данных альтернативных подходов проиллюстрированы рис. 3.7. На этом же рисунке проиллюстрирована идея **метода динамического анализа.** На наш взгляд это наиболее обоснованный подход к проектированию для исследуемой области приложений. Суть этого метода состоит в выполнении следующих действий:

1. Исходя из возможных затрат на систему защиты, выбирается вариант, обеспечивающий максимальный уровень защищенности (при условии, что он не ниже требуемого). Причем это делается в рамках заданных ограничений на параметр производительности.
2. Затем, в рамках выбранного варианта, определяется набор механизмов защиты и их параметры, которые должны быть включены для обеспечения требуемого уровня защищенности. Остальные механизмы находятся в резерве - - выключены (по существу данная задача аналогична исходной, так как определение набора изначально включенных механизмов защиты является той же задачей, что и определение варианта построения системы защиты).
3. При внедрении в систему включаются лишь необходимые механизмы, но система обладает резервом в защищенности, который может постепенно выбираться при снижении уровня защищенности.
4. В процессе функционирования системы собирается необходимая статистика о параметрах системы (реализуется метод статистической оценки) с использованием которой непрерывно оценивается уровень защищенности системы (посредством представленных ранее формул).

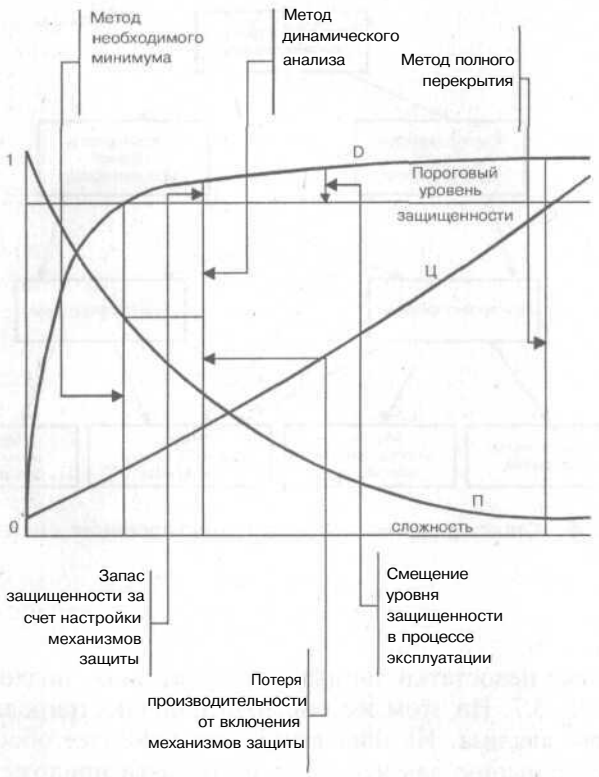


Рис. 3.7. Иллюстрация метода динамического анализа

5. При снижении уровня защищенности ниже заданного порогового значения включаются избыточные механизмы защиты, что позволяет поддерживать требуемый уровень защищенности в течение некоторого времени функционирования системы (естественно, тем большего, чем выше исходные затраты на систему защиты).

Таким образом, особенностью метода динамического анализа является то, что исходно создается некоторый запас в механизмах защиты, что обеспечивает уровень защищенности выше необходимого на текущий момент. Вместе с тем, данные механизмы при внедрении системы защиты в эксплуатацию изначально не используются. Они находятся в резерве.

В результате использование избыточных механизмов в системе защиты позволяет реализовать требуемый уровень защищенности без дополнительного снижения производительности. Это по существу и отличает данный подход от метода полного перекрытия. При снижении же уровня защищенности ниже требуемого порога (обнаружении новых угроз) будут подключаться избыточные механизмы.

Активация избыточных механизмов (которые при этом становятся основными) защиты приводит к соответствующему снижению производительности системы. Однако надо отметить, что вывод из резерва новых механизмов может привести к целесообразности отключения некоторых задействованных механизмов на основании частичного дублирования выполняемых ими функций. В результате это позволит минимизировать влияние системы защиты на производительность защищаемого объекта.

Таким образом, метод динамического анализа представляет собою разумный компромисс, состоящий в следующем: чтобы обеспечить жизнеспособность системы защиты, вносятся дополнительные функции защиты (дополнительные относительно возможности обеспечения текущего уровня защищенности). Однако они не включаются при внедрении системы защиты, а помещаются в резерв. Это позволяет обеспечить максимальную производительность системы при требуемой текущей защищенности. Снижение производительности будет происходить по мере подключения резервных механизмов защиты при необходимости повышения уровня защищенности.

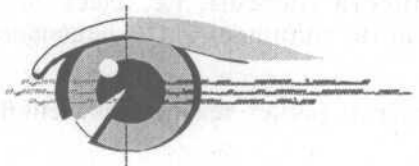
Важнейшим требованием к реализации описанного метода является непрерывная оценка защищенности системы, т.е. здесь должна быть реализована система мониторинга защищенности, решающая следующие задачи:

* Непрерывный сбор статистики и расчет текущих значений параметров защищенности.
* Оценка уровня защищенности системы и его сравнение с пороговым значением.
* Выработка рекомендаций по включению зарезервированных механизмов защиты при снижении текущего уровня защищенности ниже порогового значения.

Отметим, что для выработки рекомендаций опять же осуществляется сравнение возможных вариантов защиты с использованием рассмотренных подходов. При этом варианты различаются набором включаемых механизмов, а параметры защищенности определяются уже на основании собранной статистики о конкретной системе.

Архитектурные принципы

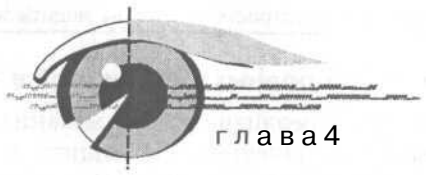
построения системы защиты информации



Часть II

4 Общие принципы. Системный подход в проектировании систем защиты

- Архитектура системы защиты
- Особенности архитектуры сетевой системы защиты. Состав и назначение функциональных блоков



Общие принципы. Системный подход в проектировании систем защиты

В данной главе рассмотрим основные архитектурные принципы построения системы защиты, определим ее состав и назначение основных функциональных блоков.

Система защиты компьютерной информации от НСД может рассматриваться в двух приложениях:

* защита автономного компьютера;
» защита компьютера в составе сети (в работе рассматриваются вопросы защиты компьютера в составе ЛВС).

Очевидно, что задача защиты компьютера как самостоятельного объекта, предназначенного для хранения и обработки информации, должна решаться в обоих случаях.

4.1. Системный подход к проектированию системы защиты компьютерной информации от НСД

До этого каждый из механизмов защиты рассматривался и анализировался по отдельности. Однако на практике все они реализуются в рамках одной системы защиты, а потому нельзя не учитывать их взаимное влияние. Поэтому уже на начальном этапе проектирования системы защиты необходимо придерживаться системного подхода, в рамках которого механизмы защиты рассматриваются и проектируются в совокупности.

Таким образом, при проектирования системы необходимо учитывать следующие аспекты:

» комплексирование разнородных механизмов в единой системе;
» взаимное влияние защитных механизмов;
« оптимальность системы защиты;
» ориентацию на статистику угроз.

Каждый из этих аспектов подробно рассмотрен далее.

Комплексирование разнородных механизмов в единой системе

Никакой, даже великолепно реализованный, механизм защиты в отдельности не сможет обеспечить качественной защиты в целом. Защита информации требует комплексирования разнородных механизмов в единой системе. Причем при проектировании системы следует анализировать не отдельно взятые механизмы, а решение конкретной задачи обеспечения безопасности всей системой защиты в целом, то есть реализованной совокупностью механизмов.

Таким образом, далее будем говорить о требованиях к заданным свойствам защищенности, обеспечиваемых совокупностью реализованных защитных механизмов. На наш взгляд, например, целесообразно говорить не о том, что в системе защиты должен быть реализован механизм авторизации пользователя при длине пароля не менее определенного числа символов, а о том, что в системе должна быть обеспечена защита входа в систему с определенными характеристиками, а также и о совокупности механизмов защиты, обеспечивающих надежную авторизацию пользователя. Причем делать это нужно с учетом того, что существуют способы преодоления данного механизма в отдельности, например, с использованием программ-сниферов, программ подбора паролей и др.

Необходимые характеристики могут быть заданы либо количественно, либо качественно. В частности, при качественном задании характеристик целесообразно задать угрозы (из существующей статистики и возможные — потенциальные), которым необходимо противодействовать при решении задачи входа в систему.

Взаимное влияние различных механизмов защиты

Другая сторона проблемы заключается в том, что использование одного механизма защиты может существенно изменить требования к реализации другого механизма. То есть, требования к отдельному механизму могут быть существенно снижены за счет расширения функций другого механизма.

Рассмотрим пример. В системе защиты должен быть реализован механизм гарантированной очистки остаточной информации на жестком диске. Но для просмотра остаточной информации злоумышленнику необходимо запустить соответствующую специальную программу, поскольку из обычных широко используемых приложений доступ к такой информации не получить (приложения не предоставляют подобной возможности). Однако, если в системе корректно реализован и активен важнейший механизм защиты – механизм обеспечения замкнутости программной среды, то данная скрытая угроза ликвидируется этим механизмом. При этом пользователю разрешается запускать только санкционированные для него процессы, а значит, пользователь не сможет запустить программу, предоставляющую возможность доступа к остаточной информации.

Получаем, что требования к очистке внешней памяти при корректной реализации механизма обеспечения замкнутости программной среды могут быть существенно снижены. При этом в качестве замечания отметим, что реализация данного требования весьма ресурсоемка, то есть приводит к существенному увеличению загрузки вычислительного ресурса.

Оптимальность системы защиты

Как отмечено ранее, реализация системы защиты влияет на другие параметры защищаемого объекта, прежде всего, на его производительность (загрузку вычислительного ресурса защищаемого компьютера). Это обусловливает необходимость выбора оптимальных решений, при этом оптимальность должна рассматриваться по совокупности параметров. Способ учета разнородных параметров при проектировании системы защиты уже был нами рассмотрен в первой главе.

Ориентация на статистику угроз

При проектировании системы защиты целесообразно ориентироваться на существующую статистику угроз. При этом ориентироваться нужно только в смысле иллюстрации возможностей того, либо иного механизма защиты. Недопустимой является разработка механизмов защиты под конкретную угрозу.

Системный подход предполагает проектирование системы защиты в предположении, что существуют потенциальные (неизвестные) угрозы, которые характеризуются скрытыми (неявными) каналами НСД. Поэтому при проектировании системы защиты необходимо рассматривать не конкретные угрозы, а характерные признаки их возникновения и противодействовать не угрозам, а возможным каналам их возникновения.

Принципы системного подхода при проектировании системы защиты

Все вышесказанное в этом разделе позволяет сформулировать принципы системного подхода при проектировании системы защиты. Принципы эти таковы:

1. Любой механизм защиты должен проектироваться с учетом его влияния на безопасность системы в целом и с учетом функций защиты, реализуемых другими механизмами. То есть должно выполняться классическое условие системного проектирования — учитываться влияние подсистемы на систему в целом, состоящую из совокупности подсистем.
2. Проектирование системы защиты - - многокритериальная задача. Поэтому при разработке механизма защиты должен учитываться не только обеспечиваемый им уровень безопасности, но и, по крайней мере, его влияние на производительность защищаемого объекта.

3. Проектирование системы защиты должно осуществляться в предположении существования потенциальных (неизвестных) угроз, которые характеризуются скрытыми (неявными) каналами НСД. При этом необходимо рассматривать не конкретные угрозы, а характерные признаки их возникновения и противодействовать не угрозам, а возможным каналам их возникновения.

4.2. Особенности системного подхода к проектированию системы защиты компьютерной информации в составе ЛВС

Особенности и проблемы защиты компьютерной информации в составе ЛВС

В части защиты компьютерной информации в составе ЛВС, кроме решения задачи защиты отдельного компьютера, возникает и задача защиты взаимодействия защищаемых объектов между собой. При этом компьютер уже должен рассматриваться не только (а при определенной технологии обработки и хранения информации — и не столько) как самостоятельный объект, требующий защиты, но и как средство удаленного доступа к другим защищаемым объектам информации.

В частности, рабочую станцию можно рассматривать, как средство удаленного, в том числе, возможно, и несанкционированного доступа к серверу, на котором хранится защищаемая информация. Причем на самой рабочей станции защищаемой информации может и не храниться.

Отметим, что защита информации в составе ЛВС по-прежнему реализуется средствами, устанавливаемыми на компьютеры (рабочие станции и серверы) в составе ЛВС. При этом возможности механизмов защиты должны быть соответствующим образом функционально расширены.

Другой особенностью построения системы защиты в составе ЛВС является возможность реализации в ЛВС выделенного рабочего места администратора безопасности (сервера безопасности). С учетом требований к оперативности принятия решений в распределенной системе защиты это становится необходимостью.

В функции администратора входят:

- удаленное управление средствами защиты информации, установленными на компьютеры в составе ЛВС;
- контроль за действиями пользователей на защищаемом объекте;
- оперативная обработка регистрационной информации (данных аудита), формируемой механизмами защиты;
- выработка реакций на определенные виды событий и т.д.

В общем случае проблема проектирования систем защиты информации в ЛВС связана либо с различными уровнями конфиденциальности, либо с различным функциональным назначением информации, обрабатываемой и циркулирующей в сети. Как следствие, возникают проблемы с требованиями к ее защите в рамках одной системы.

Очевидно, что говорить о разумной достаточности средств защиты невозможно без учета уровня критичности или конфиденциальности (назначения) обрабатываемой информации. При этом, как правило, реализуемые в сети информационные системы характеризуются несколькими типами потоков обрабатываемой информации. Примерами могут служить: внешний Web-сервер корпорации, доступ к которому разрешен практически с любой рабочей станции сети Internet; внешний и внутренний файловые серверы корпорации, предполагающие уже существенное разграничение прав доступа к ним; внутренний сервер баз данных, на котором может храниться конфиденциальная (например, финансовая) информация корпорации и др.

Задачи системного подхода при проектировании системы защиты ЛВС

Системный подход при проектировании системы защиты ЛВС с различным уровнем конфиденциальности (назначения) обрабатываемой в ней информации состоит в решении следующей совокупности задач [18]:

- « выделение фрагментов сети (информационных подсистем), обрабатывающих соответствующие информационные потоки различных уровней конфиденциальности (назначения);
- « проектирование систем защиты каждого отдельного информационного потока в соответствии с требованиями к защищенности системы, обрабатывающей информацию данного уровня конфиденциальности (назначения). Заметим, что если несколько различных (в смысле уровня конфиденциальности) потоков обрабатываются одной подсистемой, то подсистема защищается по требованиям к параметрам защиты более конфиденциального потока информации;
- » проектирование системы защиты для интерфейсов взаимодействия подсистем обработки информационных потоков различных уровней конфиденциальности (назначения).

Таким образом, основу рассматриваемого подхода составляет деление (фрагментация) системы на подсистемы с последующей защитой отдельных подсистем и соответствующих интерфейсов.

Информационные потоки будем считать различными при различии их конфиденциальности (назначения), в конечном итоге — при различии требований к их информационной безопасности.

Информационные потоки будем считать одинаковыми (без анализа их функционального назначения) при совпадении требований к их информационной безопасности.

Задачу разделения различных информационных потоков можно решать либо на физическом уровне, либо на виртуальном — с использованием средств защиты, устанавливаемых на компьютер.

Практические шаги по построению системы информационной защиты ЛВС в рамках системного подхода

Основу применения системного подхода составляет решение следующей совокупности задач проектирования системы защиты:

1. Выделяются информационные потоки, защиту которых необходимо обеспечить.
2. Для защиты каждого информационного потока устанавливается **виртуальная система защиты информационного потока** (ВСЗИП). «Виртуальная» -- понимается в том смысле, что система может представлять собою некоторое программное обеспечение, устанавливаемое на существующих технических средствах обработки информации. Однако она может требовать использования выделенного технического средства защиты, либо некоторой их совокупности.
3. Для разделения информационных потоков и защиты интерфейсов взаимодействия подсистем в местах их функционального объединения устанавливаются **виртуальные системы разделения информационных потоков** (ВСРИП).
4. Разрабатываются достаточные требования к защищенности информационных потоков информационной системы с учетом конфиденциальности (назначения) информации, принадлежащей анализируемому информационному потоку (вопросы достаточности требований рассмотрены в предыдущей главе).
5. Разрабатываются требования к ВСЗИП.
6. Разрабатываются требования к дополнительным системам защиты информации, устанавливаемым на компьютеры в составе ЛВС.
7. Разрабатываются требования к ВСРИП, к которым могут быть отнесены:
   - требования к архитектуре информационной системы в целом;
   - требования к дополнительным средствам защиты информации, используемым для разделения информационных потоков.

Отметим, что описанный системный подход к проектированию систем защиты целесообразно использовать и для проектирования комплекса организационно-технических мероприятий по обеспечению информационной безопасности информационных систем. Важным при этом будет то, что организационно-технические мероприятия разрабатываются применительно к конкретному информационному потоку с учетом его конфиденциальности (назначения).

Пример разделения и защиты информационных потоков в рамках типовой ЛВС организации

Проиллюстрируем сказанное ранее. Рассмотрим потоки информации, циркулирующие в рамках некой типовой ЛВС корпорации и требующие защиты. При этом можно выделить следующие потоки:

- » внутренний информационный поток взаимодействия рабочих станций (РС) с информационными серверами (ИС) баз данных;
- « внутренний информационный поток взаимодействия РС с внутренними файл-серверами ЛВС корпорации;
- * внутренний информационный поток взаимодействия внутренних и внешних файл-серверов между србой;
- * внешний информационный поток взаимодействия внешних файл-серверов с удаленными рабочими станциями и серверами по виртуальным каналам сети передачи данных общего пользования (СПДОП).

Структура ЛВС корпорации с ВСЗИП и ВСРИП, характеризуемая наличием всех перечисленных информационных потоков, представлена на рис. 4.1.

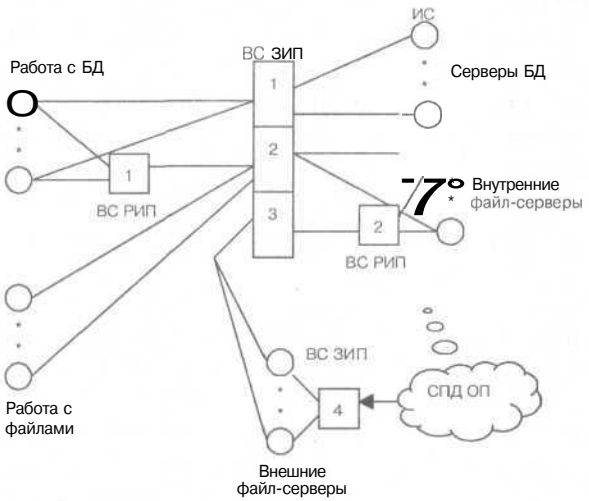


Рис. 4.1. *Структура виртуальной системы защиты ЛВС*

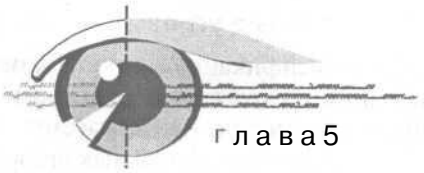
Реализация первой ВСРИП в общем случае необходима, так как предполагается, что одни и те же рабочие станции могут иметь доступ как к серверам БД, так и к внутренним файл-серверам. Вторая ВСРИП используется потому, что внутренние файл-серверы могут обмениваться информацией как с соответствующими рабочими станциями, так и с внешними файл-серверами.

Четыре ВСЗИП предназначаются для защиты соответствующих информационных потоков:

1. .....взаимодействия рабочей станции с серверами БД;
2. .....взаимодействия рабочей станции с внутренними файл-серверами;
3. .....взаимодействия внутренних файл-серверов с внешними файл-серверами;
4. .....взаимодействия внешних файл-серверов с удаленными рабочими станциями и серверами.

Необходимо отметить, что схема, приведенная на рис. 4.1, составлена в предположении, что все четыре потока характеризуются различными уровнями конфиденциальности. Например, если 2 и 3 потоки имеют равную конфиденциальность — они сольются, соответственно объединятся 2 и 3 ВСЗИП и не потребуется два ВСРИП.

В рамках данной структуры должны быть разработаны требования к ВСЗИП и ВСРИП, а также распределение задач защиты между средствами защиты информации, устанавливаемых на компьютеры ЛВС (рабочие станции и серверы).



глава 5

Архитектура системы защиты

5.1. Объекты угроз

5.1.1. Классификация угроз по способу их осуществления

В главе 2 (п. 2.2.2) была представлена классификация угроз, введенная исходя из проанализированной статистики. Здесь рассмотрим иную классификацию (которую также можно ввести, анализируя существующую статистику угроз), а именно классификацию угроз по способу их осуществления. Эта классификация необходима нам для формирования задач добавочной защиты информации и приведена она на рис. 5.1.

Рис. 5.1. *Классификация угроз по способу их осуществления*

Явные угрозы

В соответствии с данной классификацией угрозы компьютерной безопасности подразделяем на явные и скрытые. Под явными понимаем такие угрозы, которые понятны и однозначно предсказуемы. Они не требуют для противодействия им каких-либо дополнительных сведений о статистике угроз и неочевидных предположений о возможных атаках злоумышленника.

Явные угрозы связаны с некорректной реализацией и настройкой системы защиты. К таковым могут быть отнесены:

* некорректность реализации механизма защиты;
* неполнота покрытия каналов доступа к информации механизмами защиты;
* некорректность (противоречивость) возможных настроек механизмов защиты.

В двух словах прокомментируем классификацию явных угроз (подробно этот вопрос будет исследован в последующих главах).

Некорректность реализации механизма защиты может быть проиллюстрирована невозможностью в ОС Windows 9x/Me запретить доступ к системному диску «на запись», а также невозможностью управлять доступом к неразделяемым системой и приложениями каталогам (например, «Temp», «Корзина», «Мои документы» и т.д.) Это в свою очередь не позволяет говорить о корректности реализации механизма управления доступом к файловым объектам.

Если рассматривать ОС Windows NT/2000/XP, то здесь можно отметить невозможность разграничить доступ к устройствам ввода «на исполнение», что позволяет пользователю запускать любые программы с внешних носителей.

Неполнота покрытия каналов доступа к информации механизмами защиты может быть проиллюстрирована невозможностью управления доступом к некоторым ресурсам защищаемого объекта, например, к виртуальному каналу связи и т.д. При этом ресурсы могут быть как локальными, так и сетевыми — в составе ЛВС.

Некорректность (противоречивость) возможных настроек механизмов защиты может рассматриваться в двух аспектах: собственно некорректность настроек и некорректность механизма (способа) задания настроек.

В первом случае речь идет о невозможности для механизма защиты задания корректных настроек как таковых, например, если для мандатного механизма управления доступом не реализован принцип задания настроек «все, что не разрешено, то запрещено».

Во втором случае речь идет о настройке механизмов защиты в иерархической системе, где настройки может осуществлять администратор безопасно-

сти, администратор СУБД и приложения, пользователь и т.д. Например, пользователь в ОС самостоятельно может разделять (делать доступными из сети) ресурсы, к которым ему разрешен доступ, администратор СУБД может осуществлять настройки встроенных в СУБД механизмов защиты, а пользователь самостоятельно может делегировать свою роль. В то же время ранее было показано, что задача управления безопасностью должна решаться централизованно с доминирующей ролью администратора безопасности.

Скрытые угрозы

Под скрытыми понимаем такие угрозы, которые не очевидны, и требуют для противодействия им дополнительных предположений о возможных атаках злоумышленника.

Скрытые угрозы связаны с нерегламентированными действиями пользователя прежде всего посредством запуска собственных программ, а также с использованием злоумышленником ошибок и закладок в системном и прикладном ПО.

При этом скрытая угроза может быть охарактеризована двумя свойствами:

- характеристикой объекта угрозы (например, учетная запись пользователя);
- общей характеристикой атаки злоумышленника (например, модификация учетной записи с применением собственно запущенной программы. Действий подобной программы множество, как известных, так и неизвестных).

С учетом сказанного можно сделать следующий важнейший вывод: **любой механизм защиты должен проектироваться с учетом как явных, так и скрытых (в том числе и неизвестных) угроз информационной безопасности, так как только в этом случае можно говорить о возможности реализации механизмом защитных свойств.**

5.1.2. Классификация объектов угроз

Для обоснования структуры системы защиты необходимо рассмотреть классификацию объектов угроз. Она представлена на рис. 5.2.

С учетом этой классификации к объектам защиты могут быть отнесены:

1. Информационные и иные ресурсы защищаемого объекта, включая локальные и сетевые (в составе ЛВС). К таковым могут быть отнесены собственно система (вход в систему), файловые объекты (локальные и разделяемые) и т.д.
2. Программные средства защищаемого объекта, включая программные средства ОС и приложений, а также ПО системы защиты. При этом должна быть обеспечена неизменность, а при необходимости — активность процессов, драйверов, динамических библиотек и т.д.

Рис. 5.2. *Классификация объектов угроз*

3. Настройки программного обеспечения, включая настройки системного и прикладного ПО (реестр ОС, файлы настроек ОС и приложений, настройки BIOS и т.д.), а также настройки системы защиты (файлы настроек, реестр ОС).
4. Аппаратные средства защищаемого объекта, включая собственно оборудование компьютера, а также оборудование системы защиты. При этом с целью усиления защищенности может использоваться дополнительное оборудование, в частности плата, обеспечивающая функциональное расширение BIOS в части ввода пароля перед загрузкой системы с внешнего носителя.

5.2. Функциональная модель системы защиты. Состав и назначение функциональных блоков

5.2.1. Основные группы механизмов защиты. Функциональная модель

Современными нормативными документами в области защиты информации в части защиты от НСД [1, 2] выделяются следующие основные группы механизмов защиты:

1. Механизмы авторизации пользователей.
2. Механизмы управления доступом пользователей к ресурсам.
3. Механизмы контроля целостности.
4. Механизмы регистрации (аудита).

Функционально (с учетом действий пользователя при доступе к ресурсам, а также с учетом противодействия НСД к информации механизмами защиты) система защиты должна строиться как иерархическая система - - могут быть выделены несколько основных уровней иерархии защиты. Выделение данных уровней и их реализация является необходимым (определяется формализованными требованиями) условием построения системы защиты.

Функциональная модель системы защиты, которая может быть получена на основании анализа формализованных требований к системе защиты, представлена на рис. 5.3.

Примечание

Для конкретных уровней функциональной модели системы защиты сопоставление механизмов защиты не приводится, т.к. оно здесь достаточно очевидно.

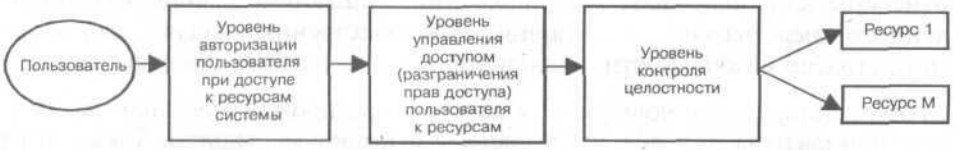


Рис. 5.3. *Функциональная модель системы защиты информации на основе формализованных требований*

Функциональная модель системы добавочной защиты [9, 10, 12], решающей рассмотренные выше задачи (п. 3.3.3), представлена на рис. 5.4.

Рис. 5.4. *Функциональная модель системы защиты информации на основе разработанных требований к добавочной защите*

Из сравнения функциональных моделей, представленных на рис. 5.3 и рис. 5.4 видно, что с целью решения сформулированных выше задач добавочной защиты в модель защиты включены:

- уровень контроля (мониторинга) активности ПО системы защиты;
- уровень контроля (мониторинга) наличия оборудования системы защиты;
- кроме того, принципиально изменены функции уровня контроля целостности — данный уровень защиты здесь функционально предназначен для контроля (мониторинга) корректности выполнения функций защиты и контроля целостности.

Рассмотрим назначение уровней защиты в приведенной функциональной модели. Задачи, решаемые на различных уровнях, реализуются с учетом сформулированных для них требований. При этом каждый уровень защиты будем рассматривать как функциональный блок в представленной схеме.

Уровень авторизации пользователя при доступе к ресурсам системы

Для начала надо определиться, кого мы будем понимать под пользователем. К пользователям, в рамках уровневой модели защиты, могут быть отнесены как пользователи приложений, решающие с использованием данного средства соответствующие производственные задачи, так и администратор безопасности, являющийся пользователем системы защиты.

Уровень авторизации пользователя обеспечивает проверку учетных параметров пользователя при доступе в систему и к системе защиты. Также этим уровнем защиты решается ряд вспомогательных задач, например, запуск процесса (приложения) после авторизации ответственного лица и др. Рассмотрению авторизации целиком посвящена третья часть данной книги.

Уровень управления доступом (разграничения прав доступа) пользователя к ресурсам

Уровень управления доступом (разграничения прав доступа) реализует собственно разграничительную схему доступа пользователей к ресурсам защищаемого объекта, а также политику администрирования системы защиты в рамках выполнения политики информационной безопасности. Под системой защиты здесь понимаем соответствующие механизмы, встроенные в ОС, СУБД, приложения, а также добавочные механизмы защиты.

Для решения задачи управления доступом к ресурсам на этом уровне выделяются локальные и сетевые ресурсы. К локальным ресурсам, требующим разграничения доступа пользователей, относятся:

- файловые объекты (логические диски, каталоги, файлы);
- устройства со сменными носителями (в частности, дисковод и CD-ROM);
- отчуждаемые физические носители информации (в частности дискеты и CD-ROM диски);
- коммуникационные порты;
- локальные принтеры;

- процессы (исполняемые файлы), в том числе процессы ОС, системы защиты и приложений — в части их модификации и запуска;
- настройки ОС (для ОС Windows — реестр ОС);
- файлы настроек системы защиты;
- файлы настроек приложений;
- при использовании СУБД — таблицы данных и таблицы настроек;
- настройки «рабочего стола» ОС и т.д.

К сетевым ресурсам (в составе ЛВС), требующим разграничения доступа пользователей, относятся:

- разделяемые сетевые ресурсы (по протоколу NetBios для сети Microsoft), к которым относятся разделяемые файловые объекты, устройства со сменными носителями (виртуальные каналы связи сети Microsoft);
- сетевые ресурсы, например, по протоколу TCP/IP (хосты, протоколы), виртуальные каналы связи сети TCP/IP;
- сетевые принтеры;
- сетевые службы и приложения (в том числе приложения информационных систем, например, СУБД), в части их модификации и запуска;
- файлы настроек сетевых служб и приложений и т.д.

Разграничительная политика рассматривается как в виде разграничения доступа к ресурсам, так и в виде функций, реализующих разрешенный доступ (например, чтение, запись, исполнение и др). Решение задач разграничения доступа пользователей к ресурсам предполагает и реализацию процедур возврата коллективно используемого ресурса в исходное состояние для его предоставления другому пользователю (например, очистка оперативной и внешней памяти).

На этом уровне также решается задача распределения функций администрирования безопасностью системы между пользователями, системным (сетевым) администратором, администраторами СУБД и приложений, администратором безопасности. При этом решается задача централизации схемы администрирования безопасности, в рамках которой изменение настроек безопасности на различных уровнях иерархии системы должно осуществляться только при непосредственном контроле со стороны администратора безопасности.

Вопросы корректности и полноты, а также возможности противодействия группе скрытых угроз, связанных с нерегламентированными действиями пользователя, для данного уровня защиты будут рассматриваться в четвертой части.

Уровень контроля (мониторинга) корректности выполнения функций защиты и контроля целостности

Отметим, что сама по себе задача контроля целостности предполагает возможность несанкционированного доступа к информации (в против-

ном случае достаточно первых двух уровней защиты). Таким образом, данный механизм априори служит противодействию скрытым угрозам в предположении, что злоумышленником преодолены первые два уровня защиты, которые реализуют разграничительную политику доступа к ресурсам защищаемого объекта.

В рамках формализованных требований недостаток механизма контроля целостности состоит в том, что данный механизм, во-первых, реализует очень ограниченный набор функций, а во-вторых, не позволяет обеспечивать эффективную реакцию на скрытую атаку в реальном времени. По сути он только фиксирует ее факт и последствия.

С учетом сказанного функциональные задачи этого уровня защиты имеет смысл принципиально расширить. При этом к уже существующим задачам контроля целесообразно добавить контроль (мониторинг) корректности выполнения разграничительной политики доступа, реализуемой на предыдущем уровне. В том числе на этом уровне следует в реальном времени фиксировать факты использования злоумышленником ошибок и закладок в системном и прикладном ПО, а также оказывать противодействие данной группе скрытых угроз. К этому уровню также следует отнести контроль целостности программ и данных, то есть контроль объектов файловой системы.

В отличие от двух предыдущих уровней, где соответствующие механизмы (программные модули системы защиты) запускаются асинхронно по факту запроса в системе на доступ к ресурсу, данный уровень реализует синхронную процедуру контроля. При этом он контролирует соответствующие события периодически, что связано с его более сильным влиянием на загрузку вычислительного ресурса защищаемого объекта по сравнению с большинством механизмов защиты двух предыдущих уровней.

Исключение составляют лишь реализуемые на предыдущем уровне механизмы распределения функций администрирования безопасностью системы между пользователями, системным (сетевым) администратором, администраторами СУБД и приложений, а также администратором безопасности. Эти механизмы также реализуются с использованием синхронной процедуры контроля.

Подробно эти технологии и особенности их реализации рассматриваются в соответсвующих главах книги (п. 16.2 и гл. 17).

Уровень контроля (мониторинга) активности системы защиты

Очевидно, что большинство задач защиты информации решаются программно, т.е. защищенность компьютерной информации обеспечивается до тех пор, пока активно ПО системы защиты. При этом включение в систему механизмов добавочной защиты, которые могут быть реализованы на различных уровнях (как на системном, так и на прикладном), связано с появлением дополнительной группы угроз — угроз загрузки сис-

темы без механизмов добавочной защиты и угроз перевода механизмов защиты в пассивное состояние (отключение) в процессе функционирования защищаемого объекта.

В задачи данного уровня входит контроль активности системы защиты, с предотвращением возможности функционирования системы в незащищенном состоянии. Последнее может быть связано как с возможностью загрузки ОС без системы защиты (либо с усеченными функциями), так и с возможностью перевода системы защиты, либо ее компонент, в пассивное состояние в процессе функционирования защищаемого объекта.

Очевидно, что активность одной программы не имеет смысла контролировать другой программой, запущенной на том же компьютере. Поэтому в рамках данного уровня должны быть реализованы следующие две возможности анализа активности системы защиты:

* локальная — с использованием аппаратной компоненты (платы);
* сетевая — удаленно, администратором с сервера безопасности.

При реализации данного уровня защиты должно оказываться противодействие как явным, так и скрытым угрозам. Соответствующая технология защиты рассматривается в шестой части книги.

Уровень контроля (мониторинга) наличия оборудования системы защиты

Данный уровень защиты целесообразно реализовывать в том случае, если система защиты содержит аппаратную составляющую. При этом защита от угрозы удаления аппаратной компоненты, помимо организационных мероприятий, может обеспечиваться техническими средствами.

Изначально данный уровень обеспечивает техническую защиту от угрозы удаления аппаратной компоненты системы защиты. Однако его реализация позволяет комплексировать в единой технической системе защиты информации различные функции защиты: защиты компьютерной информации, контроля доступа в помещения, противопожарной безопасности и т.д. Причем все это будет доступно с единого рабочего места администратора.

Функции аппаратной компоненты защиты и методы ее реализации будут рассмотрены в шестой части книги.

5.2.2. Сводные рекомендации по отдельным уровням функциональной модели

С учетом сказанного в данном разделе можем сделать следующие выводы:

1. При построении системы добавочной защиты информации целесообразно принципиально пересмотреть функции уровня контроля

целостности. Будем позиционировать данный уровень защиты, как уровень контроля (мониторинга) корректности выполнения функций защиты и контроля целостности.

2. В уровневую функциональную модель защиты имеет смысл включить уровень контроля (мониторинга) активности системы защиты, а при использовании аппаратной компоненты защиты функциональная модель может быть дополнена уровнем контроля (мониторинга) наличия оборудования системы защиты.
3. Уровни же авторизации пользователя и управления доступом должны обеспечивать решение задач корректности и полноты разграничений доступа пользователя к ресурсам. Кроме того, должна быть реализована также возможность противодействия группе скрытых угроз, связанных с нерегламентированными действиями пользователя.

5.3. Регистрация (аудит) событий

Регистрация (аудит) событий осуществляется каждым реализованным в системе механизмом защиты. С учетом включения в функциональную модель системы защиты уровня контроля (мониторинга) корректности выполнения функций защиты и контроля целостности имеет смысл принципиально изменить концепцию сбора и обработки регистрационной информации.

Особенностью реализованной модели защиты с точки зрения построения подсистемы регистрации (аудита) является введение в системе двух уровней аудита.

* **Аудит первого уровня.** Этот аудит осуществляется уровнем авторизации пользователя при доступе к ресурсам системы, а также уровнем управления доступом (разграничения прав доступа) пользователя к ресурсам. На этих уровнях ведется полный аудит действий пользователя системы. При этом фиксируются все действия, связанные как с правомерными, так и неправомерными попытками доступа пользователя к ресурсам защищаемого объекта. Регистрируемыми фактами НСД здесь являются неправомерные (противоречащие реализуемой политике информационной безопасности) действия пользователя (как ошибочные, так и сознательные), которые предотвращены механизмами защиты рассматриваемых уровней.
* **Аудит второго уровня.** Осуществляется уровнем контроля (мониторинга) корректности выполнения функций защиты и контроля целостности. Аудит второго уровня уже фиксирует не все действия пользователя, включая НСД, а только критичные факты НСД, связанные с преодолением злоумышленником механизмов защиты первых двух уровней рассматриваемой модели. По существу, на данном уровне осуществляется мониторинг корректности функционирования разграничительных механизмов защиты. При этом здесь уже речь идет не об ошибках пользователя, а об осознанных действиях нарушителя.

Таким образом, реализация уровневой модели защиты позволяет ввести уровневую модель аудита. При этом принципиально различается функциональное назначение уровней аудита. На первом уровне регистрируются все действия пользователей, в том числе и попытки НСД. Причем регистрации подлежат как сознательные нарушения, так и нарушения, связанные с ошибками пользователей. На втором уровне регистрируются только факты НСД, обусловленные сознательными нарушениями пользователей, связанные с преодолением защиты первых двух уровней модели защиты (либо с некорректным функционированием данных уровней защиты).

Другими словами, рассматриваемый подход позволяет разделить регистрируемые события на некритичные (аудит первого уровня) и критичные (аудит второго уровня). При этом различаются как требования к оперативности обработки регистрационной информации уровней аудита, так и вероятность регистрируемых событий на данных уровнях, а также объемы регистрируемой информации.

Так, на первом уровне аудита ведется непрерывная регистрация событий, т.е. накапливаются большие объемы некритичной к оперативности обработки регистрационной информации. Подобную информацию администратор может получать на сервер по своему запросу в моменты минимальной нагрузки на опорную сеть, а затем обрабатывать ее в рамках проведения расследований по факту НСД.

Напротив, на втором уровне аудита регистрируется критичная к оперативности обработки информация по фактам НСД. Данные этого уровня аудита должны поступать администратору безопасности немедленно и обрабатываться в реальном времени. При этом данные обычно поступают на специальный сервер ошибок — специальную программу сервера безопасности, а их обработка производится либо автоматически, либо с привлечением администратора безопасности.

Что касается мониторинга функционирования системы защиты, то отметим, что здесь также реализуются два уровня мониторинга: мониторинг корректности выполнения системой защиты своих функций и мониторинг активности системы защиты (возможности выполнения ею своих функций).

Итак, мы можем сделать вывод, что включение в функциональную модель защиты уровня контроля (мониторинга) корректности выполнения функций защиты и контроля целостности позволяет реализовать в системе двухуровневую схему обработки данных регистрации событий (данных аудита). Это позволяет выделить лишь малую часть регистрационной информации, к которой выставляются требования по обработке в реальном времени. Таким образом, реализация двухуровневой модели аудита значительно повышает оперативность обработки регистрационной информации администратором безопасности и снижает нагрузку на трафик опорной сети при реализации сетевой системы защиты.

глава 6

Особенности архитектуры сетевой системы защиты. Состав и назначение функциональных блоков

6.1. Архитектура сетевой системы защиты

Общим элементом архитектуры любой современной информационной системы, в том числе, сетевой системы защиты, является использование в той или иной мере распределенной модели межсетевого взаимодействия клиент—сервер. Причем в большинстве случаев это взаимодействие основывается на протокольных стеках TCP/IP.

В общем случае в ЛВС могут быть реализованы следующие варианты архитектур системы защиты:

- распределенная архитектура;
- централизованная архитектура;
- централизованно-распределенная архитектура.

Возникает необходимость обоснованного выбора архитектуры системы защиты, а в случае выбора централизованно-распределенной архитектуры — задача функционального распределения задач защиты между архитектурными компонентами.

Распределенная архитектура системы защиты

Распределенная архитектура представляет собой исходный вариант системы без обеспечения функций сетевого контроля и управления с выделенного рабочего места администратора безопасности (которое для данной архитектуры отсутствует). Все механизмы защиты и функции их администрирования реализуются непосредственно на защищаемом объекте. При этом какой-либо элемент централизации системы отсутствует как таковой.

К неоспоримым достоинствам подобной архитектуры следует отнести:

* максимальный обеспечиваемый уровень надежности системы защиты, так как отсутствует какой-либо структурообразующий элемент, отказ которого приводит к отказу всей системы в целом;

» отсутствие какого-либо влияния системы защиты на пропускную способность связного ресурса защищаемой сети.

К принципиальным недостаткам архитектуры относится:

» невозможность обеспечения оперативного управления рассредоточенной системой защиты в целом. То есть невозможно управление защитой одного объекта при зарегистрированных событиях на других защищаемых объектах. Можно управлять лишь отдельным элементом защиты с локальной консоли;

* невозможность оперативной обработки регистрационной информации, а также невозможность оперативного контроля действий пользователей на защищаемых сетевых объектах. Как следствие, невозможно оперативное расследование по фактам НСД.

Централизованная архитектура системы защиты

В основу централизованного типа архитектуры положен принцип удаленной реализации защиты объектов с центральной консоли администратора безопасности. Все функции по обеспечению информационной безопасности защищаемых объектов делегированы одной компоненте системы защиты — серверу безопасности. Таким образом в основе централизованных систем лежит построение виртуальной сетевой системы защиты, наложенной на опорную сеть передачи данных (ЛВС).

Естественно, что данной архитектуре присущи достоинства и недостатки, диаметрально противоположные распределенной архитектуре.

К неоспоримым недостаткам подобной архитектуры следует отнести:

» минимальный обеспечиваемый уровень надежности системы защиты. В системе присутствует один структурообразующий элемент — сервер безопасности, отказ которого приводит к отказу всей системы в целом;

» максимальное влияние системы защиты на пропускную способность связного ресурса защищаемой сети.

К принципиальным достоинствам архитектуры относится:

« обеспечение максимально оперативного управления рассредоточенной системой защиты в целом (управление защитой одного объекта при зарегистрированных событиях на других защищаемых объектах). Возможность управлять всеми элементами защиты с локальной консоли сервера безопасности;

» максимальные возможности по оперативной обработке регистрационной информации и контролю над пользователями. Как следствие, можно выделить максимальную оперативность при расследовании фактов НСД.

6.2. Централизованно-распределенная архитектура системы защиты

Достоинства и недостатки

Реализация данной архитектуры призвана объединить в себе все достоинства архитектурных решений, рассмотренных выше (соответственно, устранить их недостатки).

Останавливаясь на рассмотрении достоинств и недостатков распределенной и централизованной архитектур системы защиты, естественно (интуитивно понятно) будет сделать предположение об оптимальности следующего распределения функциональных задач между компонентами централизованно-распределенной сетевой системы защиты:

- полностью распределенно должны решаться задачи защиты рабочих станций и серверов ЛВС. Реализовываться это должно устанавливаемой на них клиентской частью системы защиты;
- централизованно должны решаться:
  - администрирования клиентских частей системы защиты;
  - обработка регистрационной информации, собираемой клиентскими частями системы защиты;
  - контроль за действиями пользователей на защищаемых объектах.

  Все это должно производиться с серверной части системы защиты – сервера безопасности.

Обратим внимание, что при централизованно-распределенной системе защиты, ее администрирование осуществляется полностью централизованно. То есть выполняется то принципиальное условие, которое было нами обосновано ранее. Описываемое далее распределение и масштабирование административных функций означает лишь распределение ресурсов, призванное снизить вред от возможного выхода из строя сервера безопасности. При этом сам принцип централизованного администрирования сохраняется, так как вершиной системы защиты является не сервер безопасности, а администратор безопасности. То есть именно администратор безопасаности является централизующим звеном.

Итак для поддержания функционирования системы в случае отказа сервера безопасности функции администрирования клиентских частей должны дублироваться распределенно. Кроме того сетевым агентам (клиентским частям системы защиты) должна отводится часть общей задачи контроля и управления. При этом настройка клиентских частей производится самим администратором безопасности (в рамках централизованной схемы администрирования), а база данных администратора безопасности оказывается распределенной.

Режимы защиты, обеспечиваемые в рамках данной архитектуры, являются более интеллектуальными, поскольку сетевой агент располагает информацией, необходимой для экстренного принятия решений без непосредственного сигнала от менеджера. При этом потоки информации, передаваемые по сети, существенно сокращаются.

Структура сетевой системы защиты в рамках централизованно-распределенной архитектуры

Структура сетевой системы защиты определяется следующим набором компонент.

- **Клиентская часть системы защиты** — обеспечивает реализацию механизмов защиты на объекте. Используется для проведения контрольных проверок и регистрации действий пользователей на локальных рабочих станциях и информационных серверах ЛВС. Обеспечивает формирование системных журналов в соответствующем формате и отображение их на локальной консоли при запуске соответствующего интерфейсного модуля.
- **Модуль управления локальной базы данных (ЛБД) узла ЛВС —** обеспечивает формирование ЛБД администратора безопасности на основе сбора и предварительной обработки локальных системных журналов (собственных и базовых журналов используемой платформы и прикладного ПО), сигналов синхронизации от центральной базы данных (ЦБД), а также сигналов администратора по настройке и обеспечению соответствующей политики безопасности в ЛВС.
- **Сетевой агент** - - программный модуль, обеспечивает маскирующее кодирование (шифрование) и передачу сигналов управления, сигналов синхронизации между локальными и удаленными модулями системы защиты, а также обеспечивающий целостность соединений агент—элемент—менеджер.
- **Сетевой менеджер** — обеспечивает в дополнение к агенту мультиплексирование/демультиплексирование сигналов, передаваемых между ЦБД и ЛБД. Таким образом, им предоставляется связь точка-многоточка на прикладном уровне модели протоколов ISO/OSI. Кроме того, сетевой менеджер реализует сеансовую авторизацию клиентских частей системы защиты при их соединении с серверной частью.
- **Сетевая подсистема** - - обеспечивает эмуляцию консоли удаленной станции с передачей сигналов управления и обратной связи по сетевому интерфейсу агент—элемент—менеджер.
- **Модуль ЦБД** — обеспечивает хранение и синхронизацию данных в ЛБД и ЦБД, а также инициализацию учетных данных пользователей ресурсов ЛВС.
- **Интерфейсный модуль** - - обеспечивает просмотр и редактирование ЦБД в соответствии с принятой политикой обеспечения безопасности. Также он осуществляет инициализацию функций расширенного

контроля удаленного узла (сканирование удаленной консоли) и вывод результатов на консоль администратора безопасности.

На рис. 6.1 представлена диаграмма потоков управления и компонентный состав системы контроля на основе рассматриваемой архитектуры.

Сплошными стрелками на диаграмме выделен информационный поток, обеспечивающий запросы удаленного выполнения программных процедур, запускаемых из интерфейсного модуля администратора в специализированном формате, представляя собой по существу некоторое подобие сетевых программ удаленного доступа.

Пунктирными стрелками обозначен поток данных, обеспечивающий автоматическую передачу журналов регистрации, фрагментов системных структур ядра ОС, а также любых настроек программного обеспечения сетевого агента узла ЛВС.

Незакрашенными (полыми) стрелками обозначен поток данных, инкапсулирующий в себе команды, данные и параметры конфигурации сетевого агента, а также несущий в себе последовательность меток временной синхронизации. Защищенность этого потока обеспечивается в режиме канального шифрования из конца в конец (то есть при непосредственной передаче информации от агента к менеджеру через ЛВС).

Серыми стрелками на диаграмме выделяются данные, передаваемые через стыковочный шлюз сетевой подсистемы защиты при выполнении контроля в режиме эмуляции удаленного узла. В данном случае имеется в виду, что внешний вид системы и интерфейс администратора сохраняются, однако система выбирает информацию для контроля не из структуры ядра и журнальных файлов, а из центральной базы данных (ЦБД) администратора безопасности.

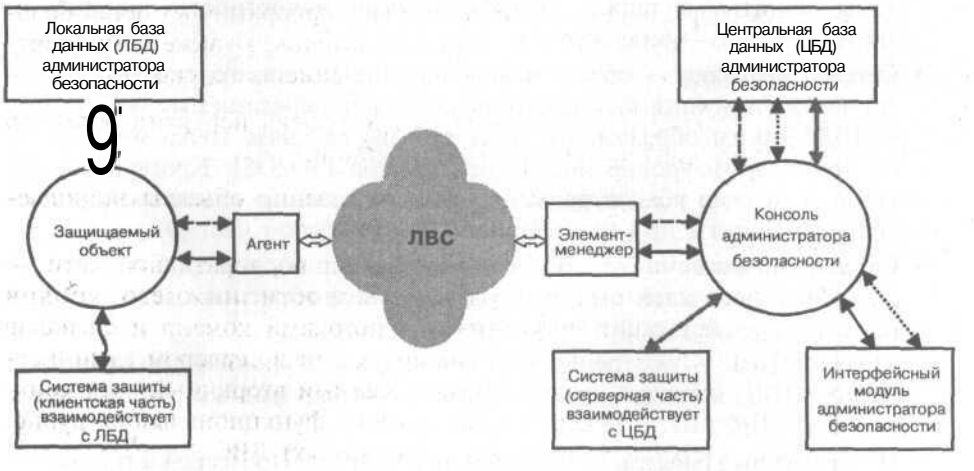


Рис. 6.1 Диаграмма потоков управления и компонентный состав системы контроля на основе централизованно-распределенной архитектуры

Стрелками, состоящими из точек, выделяется взаимодействие интерфейсного модуля администратора и ЦБД, в рамках которого реализуются основные функции удаленного управления (смена паролей, установка таймеров протокола, настройка параметров бюджета пользователей и узлов ЛВС, и т.д.).

Волнистыми стрелками выделяются локальные взаимодействия обеспечивающие функции защиты на основе технических средств системы защиты, реализуемых клиентской частью.

Централизованно-распределенная архитектура характеризуется незначительным усложнением клиентской части системы защиты по сравнению с распределенной архитектурой, при существенном упрощении серверной компоненты системы защиты по сравнению с централизованной архитектурой.

Многоуровневая централизованно-распределенная система

В рамках реализации централизованно-распределенной архитектуры очевидна возможность распределения функций удаленного управления системой защиты, в частности, посредством реализации многоуровневой схемы администрирования. В централизованно-распределенной одноуровневой архитектуре, рассмотренной выше, использовался единственный узел концентрации управления безопасностью в ЛВС.

Рассмотрим вариант большей степени децентрализации, при котором происходит дальнейшее наращивание уровней контроля и управления безопасностью защищаемых объектов. Реализуется этот вариант с использованием промежуточных систем, которые будут концентрировать управляющую нагрузку различных ЛВС и далее транслировать ее уже непосредственно на рабочее место администратора безопасности (сервер безопасности более высокого уровня). Таким образом можно обеспечить масштабирование задач управления безопасностью распределенных корпоративных сетей большой сложности. Структура информационных потоков, а также компонентный состав подобной архитектуры условно изображены на рис. 6.2.

На приведенной схеме (рис. 6.2) имеется три уровня иерархии объектов сбора и обработки контролируемой информации.

- Первичный узел концентрации — непосредственно объекты защищаемой ЛВС.
- Вторичный узел - - узел в составе сегмента корпоративной сети — ЛВС (сервер администратора безопасности нижнего уровня иерархии), отвечающий за управление потоками команд и сигналов «клиент-менеджер» между первичными узлами и сервером администратора безопасности верхнего уровня. Каждый вторичный узел контролирует объекты, принадлежащие адресно-функциональной группе, включенной в состав конкретной (конкретных) ЛВС.
- Третичный узел — рабочее место администратора безопасности верхнего уровня иерархии.

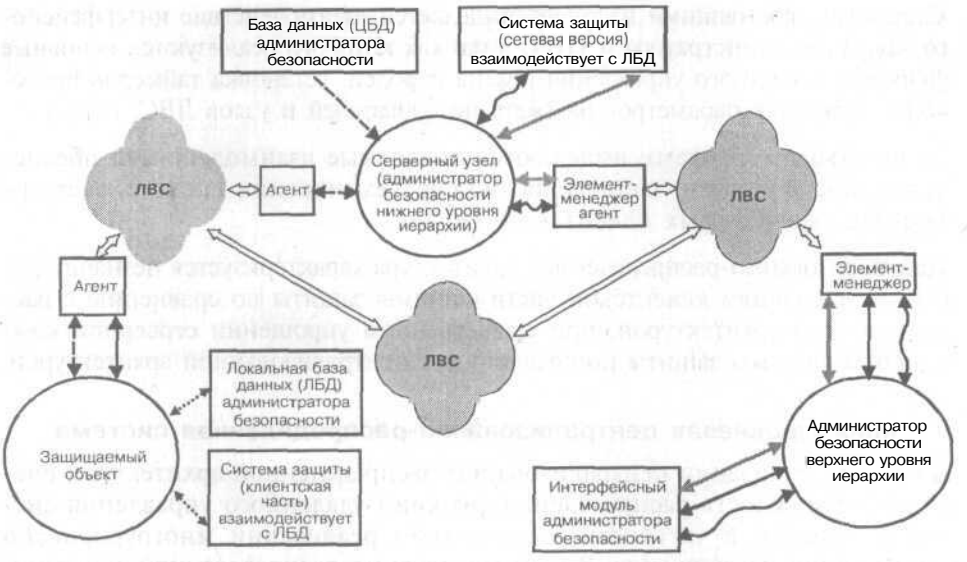


Рис. 6.2. *Многоуровневая централизованно-распределенная система*

Таким образом, на основании всего сказанного выше очевидно, что в современных системах защиты ЛВС централизованно-распределенная архитектура может рассматриваться как основная (типовая) архитектура сетевой системы защиты ЛВС. Принципы распределения функций между компонентами рассмотрены выше.

6.3. Анализ эффективности централизованно-распределенной системы защиты

6.3.1. Параметры и характеристики эффективности. Информационная сигнатура

С точки зрения анализа эффективности централизованно-распределенной архитектуры системы защиты будем оценивать влияние, оказываемое на опорную сеть передачи данных (на пропускную способность) наложенной сетью системы защиты. Цель такого анализа — выработка рекомендаций по централизации задач, решаемых системой защиты в рамках сетевой реализации системы.

При этом выделим три основных функции системы защиты, характеризующих ее функционирование в штатном режиме (не рассматривается начальный этап функционирования системы, связанный с заданием настроек ее параметров). Эти три функции таковы:

- контроль изменений состояния защищаемых объектов, выработка реакции на изменение состояния;
- контроль пользователя, работающего на защищаемом объекте;
- регистрация (аудит) событий на защищаемом объекте.

Введем следующие обозначения:

- параметр *λвход1* определяется, как интенсивность изменения состояний защищаемого узла ЛВС, которые должны быть зафиксированы системой защиты (именно на них должно быть выработано управляющее воздействие);
- параметр *λвход2* определяется, как интенсивность проведения контроля действий пользователя на защищаемом объекте;
- параметр *λвход3* определяется, как интенсивность получения запросов администратором безопасности на серверную часть регистрационной информации (данных аудита) с защищаемого объекта;

Примечание

Будем считать, что все входные потоки пуассоновские (простейшие), что позволяет осуществлять суммирование их интенсивностей, при определении интенсивности входного потока в канале связи.

- параметр P — мера централизации системы защиты, определяется как вероятность того, что событие изменения состояния защищаемого объекта — узла ЛВС, будет обрабатываться системой защиты распределенно (клиентской частью - - без участия сервера безопасности). Соответственно, с вероятностью $(1-P)$ событие будет обрабатываться централизованно — серверной частью системы защиты (сервером безопасности).

Далее будем использовать понятие информационной сигнатуры, генерируемой в канале связи системой защиты в единицу времени. Под информационной сигнатурой будем понимать общий объем информации, измеряемый в килобайтах секунду и передаваемый системой защиты по каналу связи.

Определим объемы информационных взаимодействий клиентской и серверной компонент системы защиты:

Vu..... объем информации, идентифицирующей изменение состояния защищаемого объекта (дата, время, идентифицирующие параметры события события, текущий пользователь и т.д.);

Vy..... объем информации, идентифицирующей управляющее воздействие системы защиты на факт изменения состояния защищаемого объекта;

Vзк..... объем информации, передаваемый в запросе на проведение контроля действий пользователя;

Уза ..... объем информации, передаваемый в запросе на получение регистрационной информации (аудита) с защищаемого объекта;

Vк..... объем информации контроля действий пользователя на защищаемом объекте (например, экранная копия);

Уа ..... объем регистрационной информации (аудита).

Так как в ЛВС может одновременно находиться несколько десятков и сотен защищаемых объектов, взаимодействующих с одним сервером безопасности, через *N* обозначим число защищаемых узлов в ЛВС.

С учетом введенных обозначений информационная сигнатура *S,* генерируемая в канале связи системой защиты в единицу времени, может быть определена следующим образом:

$$5 = N \cdot [P \cdot \lambda вход1 \cdot (Vи + Vу) + \lambda вход2 \cdot (Vзк + Vк) + \lambda вход3 \cdot Va].$$

Так как объем запрашиваемой с сервера регистрационной информации зависит от параметра *P* (на сервер следует запрашивать только регистрационную информацию, обрабатываемую в системе распределенно; информация, обрабатываемая централизованно, поступает на сервер в реальном масштабе времени с вероятностью $1-P$), то объем информации регистрации определяется следующим образом:

$$Va = (1 - P) \cdot Vи \cdot \lambda вход1/\lambda вход3.$$

Естественно, что если пропускную способность связного ресурса обозначить через *C,* то доля пропускной способности связного ресурса, занимаемая системой защиты *K* составит:

$$K = S/C.$$

При этом под пропускной способностью связного ресурса будем понимать не физическую скорость передачи данных в канале связи, а скорость передачи собственно информации, без учета передачи заголовков, подтверждений, затрат времени на арбитраж требований связного ресурса в ЛВС и т.д. Заметим также, что обработка заявок на обслуживание системой защиты по сравнению с обработкой заявок от прикладных задач, должна осуществляться в приоритетном режиме.

Проанализируем полученное выражение. Если в него подставить выражение, задающее параметр *Уа,* то получим следующее приближение характеристики *S*:

$$S = N \cdot [\lambda вход1 \cdot (Vи + Уу) + \lambda вход2 \cdot (Vзк + Vк)].$$

Откуда можно сделать важный вывод, что осуществление регистрации (аудита) событий в реальном времени на сервере безопасности, по существу, приводит к выполнению условия $P = 1$. Это означает получение характеристик, практически полностью совпадающих с характеристиками централизованной архитектуры системы защиты.

Таким образом, при построении системы защиты должно решаться следующее противоречие. С одной стороны, регистрировать события на сервере безопасности целесообразно в реальном масштабе времени (в противном случае, администратор безопасности не будет обладать необходимой текущей информацией о защищаемом объекте). С другой стороны, реализация данного условия приводит к нивелированию преимуществ централизованно-распределенной архитектуры с точки зрения эффективности использования связного ресурса в защищаемой сети (не с точки зрения оперативности реакций на регистрируемые события).

Ранее было отмечено, что исходя из соображений оперативности реакции на возможные факты НСД, а также исходя из обеспечения высокой надежности системы защиты, все механизмы защиты должны быть реализованы распределенно (должно выполняется условие: $P = 0$). То есть они должны исполняться клиентской частью системы защиты, устанавливаемой на защищаемые объекты. Характеристика S в этом случае задается следующим выражением:

$$S = S\backslash + S2 = N \cdot [\lambda вход1 \cdot Vu + \lambda вход2 \cdot (Vж + Vк)].$$

Здесь характеристика $S1 — N \cdot [\lambda вход1 \cdot Vu]$ определяет влияние на пропускную способность опорной сети подсистемы регистрации событий (аудита), а характеристика $S2 = N \cdot [\lambda вход2 \cdot (Vзк + Vк)]$ — влияние подсистемы контроля действий пользователя на защищаемом объекте.

6.3.2. Количественная оценка характеристик эффективности централизованно-распределенных систем защиты

Приведем грубую количественную оценку рассматриваемых характеристик. Рассмотрим влияние подсистемы аудита (характеристика $S1$) на пропускную способность связного ресурса. При этом зададим следующие значения параметров.

Пусть в сети находится 250 защищаемых объектов: $N = 250$ (это ЛВС средних масштабов). Параметр Vu определяется объемом регистрационной информации, характеризующим изменение состояние защищаемого объекта (тип события, текущий пользователь, время, дата и т.д.); примем: $Vu = 100$ байт. Параметр $\lambda вход1$ обоснованно задать наиболее сложно, т.к. события, характеризуемые состояние станции, имеющие различный физический смысл, могут генерироваться с частотой, различающейся на порядки.

Примем $\lambda вход1 = 10\ с^{-1}$ (например, за 1 секунду запускается 10 различных процессов, осуществляется 10 обращений к файловым объектам и т.д.), понимая, что реальное значение данного параметра при осуществлении атаки может быть значительно больше.

При заданных значениях параметров получаем, что 51 = 0.25 Мбайт/с. Оценим значение искомой характеристики С. Физическая скорость в канале для ЛВС Ethernet составляет 10 Мбит/с (соответственно, Fast Ethernet — 100 Мбит/с). Скорость передачи данных грубо можно принять на порядок меньше физической скорости в канале связи (затраты времени на передачу служебных частей заголовков, подтверждений передачи, разрешения коллизий и т.д.), откуда для сети Ethernet получаем порядка: С = 0.1 Мбайт/с, а для Fast Ethernet получаем порядка: $C = 1$ Мбайт/с.

Таким образом, для введенных предположений получаем, что для сети Ethernet трафик системы защиты в рамках реализации подсистемы аудита превышает исходную пропускную способность сети Ethernet и снижает пропускную способность сети Fast Ethernet на 25% ($K = S1/C = 0.25$).

6.3.3. Поиск оптимального решения

Возможные подходы

Из полученных результатов можем сделать вывод о невозможности в системе защиты реализовать передачу регистрационной информации в полном объеме на сервер безопасности в реальном времени. Мы показали, что это обусловлено слишком большими потерями производительности связного ресурса, то есть доли полосы пропускания канала связи ЛВС, занимаемой системой защиты информации.

С целью поиска оптимального решения проанализируем две составляющие — *S1* и *S2*. Из выражения $S1 = N \cdot [\lambda вход1 \cdot Vu]$ можем сделать вывод, что уменьшение характеристики *S1* возможно только за счет уменьшения значения параметра $\lambda вход1$ (т.к. уменьшить параметр *Vu* практически невозможно). Собственно параметр $\lambda вход1$ уменьшить невозможно, он определяется пользователем, а также работой ОС и приложений. Однако можно значительно снизить долю регистрационной информации, которую следует выдавать на сервер в реальном времени. При этом всю остальную информацию можно получать на сервер по запросу администратора безопасности в моменты наименьшей загрузки канала связи (например, после завершения рабочего дня, во время обеденного перерыва и т.д.).

Резонно предположить, что в реальном времени на сервер безопасности следует выдавать только информацию, связанную с НСД, то есть информацию о преднамеренных действиях пользователя, противоречащих разграничительной политики доступа к ресурсам. Это позволит снизить значение параметра $\lambda вход1$ на несколько порядков, обеспечивая долю потерь производительности связного ресурса в пределах единиц (долей) процентов.

Однако здесь появляется угроза генерирования злоумышленником событий НСД (ошибок) с целью перегрузки канала связи ЛВС и временного вывода из строя ЛВС как опорной сети связи предприятия. Поскольку

заявки системы защиты в канале связи обрабатываются с более высоким приоритетом, чем заявки на занятие связного ресурса, поступающие от приложений, то у злоумышленника такая возможность имеется.

Для противодействия этому необходимо выделять и обрабатывать в реальном времени (передавать на сервер безопасности) только те зарегистрированные события НСД, которые каким-либо образом преодолевают механизмы защиты, реализуемые клиентской частью системы защиты, и потому требуют немедленного вмешательства в процесс обработки информации администратора безопасности.

Характеристика $S2 = N \cdot [\lambda вход2 \cdot (Vзк + Vк)]$ отражает влияние на пропускную способность опорной сети подсистемы контроля действий пользователя на защищаемом объекте. Проанализируем данную зависимость. Во-первых, очевидно, что $Vзк << VK$, т.е. далее параметр VK в выражении для $S2$ можем опустить. Во-вторых, контроль действий пользователя осуществляется подсистемой оперативной обработки, реакцию которой задает человеческий фактор. Реальный интервал времени, с которым администратор безопасности может просматривать контролируемую информацию (например, изображение на мониторе защищаемого объекта) — это 5...15 секунд, причем вне зависимости от числа подключенных к ЛВС защищаемых объектов. Поэтому здесь примем: $N \cdot \lambda вход2 = 0.1\ с^{-1}$.

Объем контролируемой информации может быть достаточно велик, однако для передачи по каналу связи контролируемая информация может сжиматься. Поэтому примем $VK = 10$ Кбайт.

С учетом сказанного получаем: $S2 = 1$ Кбайт/с, соответственно, для сети Ethernet получаем: $K = S2/C = 0.01$, для сети Fast Ethernet получаем: $K = 0.001$, т.е. контроль действий пользователя на защищаемом объекте снижает пропускную способность сети, соответственно, на 1% и на 0.1%.

Таким образом, несмотря на большие объемы контролируемой информации, ввиду низкого уровня оперативности подсистемы контроля, обусловливаемой человеческим фактором, реализация данной подсистемы в системе защиты не оказывает сколько-нибудь заметного влияния на производительность связного ресурса ЛВС. Причем это верно вне зависимости от числа подключаемых к сети контролируемых объектов защиты.

Сводные рекомендации по построению централизованно-распределенных сетевых систем защиты, исходя из анализа эффективности

Система защиты должна содержать три основных компоненты:

- собственно компоненту защиты, реализующая разграничительную политику доступа пользователей к ресурсам;

- компоненту регистрации (аудита) изменений событий на защищаемом объекте;
- компоненту контроля действий пользователя на защищаемом объекте.

На основании проведенного анализа эффективности централизованно-распределенной архитектуры системы защиты могут быть даны следующие рекомендации по ее построению.

1. Компонента защиты должна быть реализована практически полностью распределенно. При этом все ее механизмы защиты должны быть реализованы клиентской частью системы, устанавливаемой на защищаемом объекте. Централизация для этой компоненты состоит только лишь в возможности удаленного администрирования механизмов защиты с сервера безопасности, причем с дублированием данной функции с консоли защищаемого объекта (т.е. распределенно).
2. Компонента регистрации (аудита) должна быть реализована по централизованно-распределенной схеме с двумя уровнями обработки регистрационной информации. Распределенность компоненты состоит в том, что все события аудита должны регистрироваться клиентской частью. При этом на сервер должны поступать только те данные, которые зарегистрированы клиентской частью системы защиты.

   Уровни обработки определяют механизмы построения централизованной компоненты аудита и задают способы передачи регистрационной информации на сервер безопасности. Первый уровень — обработка в реальном времени - - предполагает немедленную отправку регистрационной информации клиентской частью на сервер безопасности, второй уровень — уровень оперативной обработки — предполагает выдачу регистрационной информации по удаленному запросу администратором безопасности с сервера в моменты минимальной загрузки канала связи.

   С целью минимизации влияния системы защиты на производительность связного ресурса на первом уровне должны обрабатываться минимальные объемы регистрационной информации — факты НСД (в пределе — факты преодоления злоумышленником распределенно реализуемых в системе механизмов защиты информации).
3. Компонента контроля пользователей на защищаемом объекте представляет собою реализацию централизованной схемы сбора и обработки информации. При этом контролируемая информация генерируется и выдается на сервер безопасности по команде с сервера. Несмотря на большие объемы передаваемой информации, реализация данной компоненты не оказывает сколько-нибудь заметного влияния на производительность связного ресурса (занимает доли процента полосы пропускания канала связи). Причем это независимо от числа контролируемых системой защиты объектов (рабочих станций и серверов ЛВС).

6.4. Функциональные подсистемы и модули центрально-распределенной системы защиты

Функциональная структура сетевой системы защиты

С учетом особенностей рассматриваемого в работе класса систем (систем защиты, основные функции которых реализуются на системном уровне), система должна быть хорошо структурирована. Функциональная структура системы защиты задает распределение реализуемых функций по отдельным функциональным подсистемам. Функциональная подсистема состоит из нескольких модулей, реализующих набор механизмов, необходимых для покрытия требований данной подсистемы.

Функциональный модуль определяется как структурный элемент, реализующий набор механизмов или используемый для вспомогательных целей (организации взаимодействия между другими модулями в рамках решения всего комплекса задач).

Используя определенные выше компоненты системы защиты и сформулированные в предыдущем разделе рекомендации, получаем структуру сетевой системы защиты, основанную на реализации централизованно-распределенной архитектуры, представленную на рис. 6.3. Там указаны информационные и управляющие связи между модулями в пределах одной функциональной подсистемы, а также между отдельными функциональными подсистемами.

Из рис. 6.3. можно видеть, что в общем случае могут быть выделены следующие типы функциональных подсистем:

1. Подсистема защиты рабочих станций и информационных серверов (клиентская часть системы защиты). К этой же подсистеме отнесем функции регистрации событий.
2. Подсистема удаленного контроля рабочих станций и информационных серверов.
3. Подсистема удаленного управления механизмами защиты рабочих станций и серверов.

Подсистема защиты рабочих станций и информационных серверов (клиентская часть системы защиты)

Подсистема защиты рабочих станций и информационных серверов (клиентская часть системы защиты) является структурообразующим элементом, призванным решать собственно задачи защиты информации и содержащем в себе следующие основные функциональные модули:

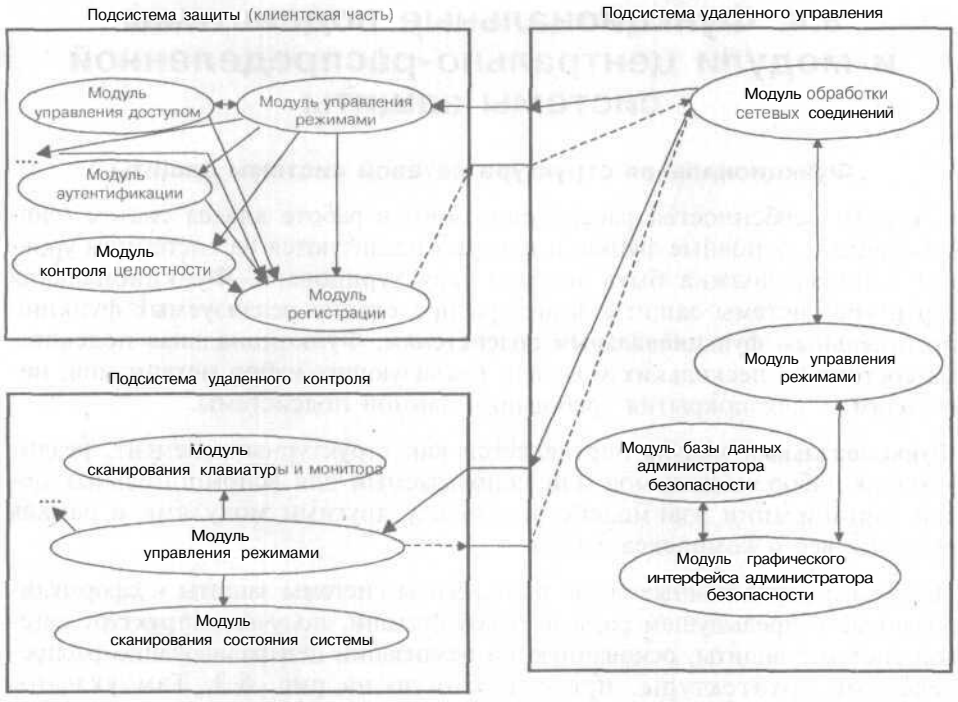


Рис. 6.3. *Функциональная структура системы защиты*

1. Модули, реализующие механизмы защиты (каждый механизм защиты реализуется отдельным функциональным модулем):
   - модуль аутентификации;
   - модуль управления доступом;
   - модуль контроля целостности;
   - модуль противодействия ошибкам и закладкам в системном и функциональном программном обеспечении;
   - модуль очистки памяти и изоляции программных модулей и др. (в зависимости от реализуемых в системе механизмов защиты).
2. Модуль регистрации (аудита).
3. Модуль управления режимами, который решает задачи инициализации механизмов при доступе к защищаемым ресурсам.

Подсистема удаленного контроля рабочих станций и информационных серверов

Подсистема удаленного контроля рабочих станций и информационных серверов реализует, в дополнение к перечисленным функциям защиты, контроль служебной деятельности сотрудников и информации, располагаемой на защищаемом объекте.

Данная подсистема состоит из следующего набора функциональных модулей:

1. **Модуль сканирования клавиатуры и монитора.** Модуль сканирования клавиатурного буфера удаленной консоли осуществляет копирование информации, набираемой на клавиатуре видеотерминала, с целью последующего архивирования и передачи на рабочее место администратора безопасности. При этом передача может осуществляться либо в синхронном режиме — по расписанию, либо в асинхронном режиме — по внешним запросам администратора безопасности.

   Предварительное накопление информации по результатам контроля состояния осуществляется либо в специальной, защищенной от просмотра области внешнего диска, либо в виде, не пригодном для непосредственного просмотра и удаления пользователями. Режим функционирования модуля определяется согласно установке соответствующих параметров локальной базы данных системы защиты по сигналам от модуля управления режимами. Команды управления режимами инкапсулируются в кадрах протокола управления сетевыми соединениями системы защиты и непосредственно передаются модулю управления режимами.

   Модуль сканирования видеобуфера удаленной консоли реализует аналогичные контролирующие функции по отношению к видеотерминалу удаленной консоли. В дополнение к перечисленному, данным модулем осуществляется автоматическое архивирование полученной информации непосредственно перед передачей на рабочее место администратора безопасности. Период выполнения контролирующих функций данного вида определяется необходимостью выборочного контроля пользователей.

2. **Модуль сканирования состояния системы** реализует функции удаленного контроля состояния системы:
   - контроля файловой системы защищаемого объекта. Реализуется возможность удаленного просмотра объектов файловой системы, создания, записи, удаления файла (каталога) на защищаемом объекте, запрета доступа пользователей к удаленно создаваемому файлу;
   - контроля среды исполнения. Реализуется возможность удаленного контроля запущенных на защищаемом объекте процессов, удаленного запуска процесса, в том числе в невидимой для пользователя консоли;
   - контроля настроек ОС. Реализуется возможность удаленного просмотра объектов реестра ОС, создания, записи, удаления ключа реестра на защищаемом объекте, запрета доступа пользователей к ветви или ключу реестра ОС;
   - контроля параметров сетевого доступа, в частности, занятых TCP портов и т.д. и др. (в зависимости от реализуемых в системе механизмов контроля).
3. **Модуль управления режимами** решает задачи инициализации механизмов при запуске функций контроля.

Подсистема удаленного управления механизмами защиты рабочих станций и серверов (серверная часть системы защиты — сервер безопасности)

Подсистема удаленного управления механизмами защиты рабочих станций и серверов (серверная часть системы защиты — сервер безопасности) обеспечивает централизацию управления техническими средствами клиентской части системы защиты и контроля (по соответствующим журналам) со стороны серверной части в удаленном режиме. Функциональные возможности подсистемы обеспечиваются следующим набором модулей:

1. **Модуль обработки сетевых соединений** предоставляет интерфейс служб представительского уровня MOC/BOC для подсистемы системы защиты и интерфейс сеансового уровня для подсистемы контроля рабочих станций и информационных серверов. Межсетевое взаимодействие целесообразно реализовать на основе служб транспортного протокола TCP и сетевого протокола IP. В качестве транспортного интерфейса для организации передачи команд и ответов реализуется закрытый протокол, позволяющий шифровать трафик. Для организации сетевых соединений и мониторинга подключения рабочих станций используется интерфейс сокетов высокого уровня. При установлении виртуального канала и передаче данных должна использоваться сеансовая аутентификация агентов и контроль целостности соединений по последовательным номерам команд-ответов и таймерам ожидания/блокировки соединения. Передача кадров протокола реализуется с использованием внешних подключаемых алгоритмов маскирования и шифрования информации.

   Модуль непосредственно взаимодействует с модулем доступа к базе данных администратора безопасности при выполнении репликации (согласования) элементов учетных записей централизованной базы данных и локальных сегментов баз данных удаленных узлов. Модуль инициализируется при запуске сервера безопасности и остается активен до момента остановки серверной части.

2. **Модуль базы данных администратора безопасности** используется для хранения, предварительной обработки (сортировки, фильтрации) и выборки информации следующих типов:
   - регистрационные записи журнальных файлов;
   - параметры конфигурации и настройки системы защиты;
   - сообщения об ошибках работы ОС и системы защиты (журналы ошибок).

3. **Модуль графического интерфейса администратора безопасности** используется для доступа к базе данных выделенного сервера администратора безопасности и осуществления прямого контроля удаленных узлов в режиме реального времени, включая выдачу команд чтения буферов клавиатуры и видеотерминала, проверки подключе-

ния пользователя к сети, команд обновления конфигурации и опроса статуса удаленного узла. Интерфейс целесообразно строить в удобной для использования форме, базирующейся на стандартных для Windows-приложений графических примитивах. При этом должны предусматриваться различные способы группирования объектов контроля в различные уровни иерархии, включая:

- группирование пользователей в функциональные группы по особенностям служебной деятельности;
- группирование пользователей в функциональные группы в соответствии с особенностями физического их расположения на территории предприятия;
- группирование узлов в сетевые сегменты в соответствии с особенностями физического их расположения и способов подключения к ресурсам сети;
- группирование узлов в информационные (автоматизированные) системы в соответствии с их принадлежностью к системам обработки данных;
- комбинированное группирование с учетом перечисленных возможностей.

По соображениям удобства мониторинг объектов контроля осуществляется с применением одного из указанных представлений.

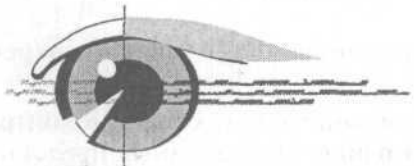
4. Модуль управления режимами решает задачи инициализации режимов мониторинга и управления.

Функционирование системы защиты должно осуществляться в двух режимах в зависимости от наличия активного соединения клиентской части с сервером безопасности.

- При работе в автономном режиме (серверная часть отсутствует) функционирование системы защиты начинается с загрузки операционной системы на защищаемом сервере или рабочей станции. При этом в качестве сервиса ОС автоматически загружается система защиты с заданными настройками параметров механизмов защиты.
- При работе в сетевом варианте (с сервером безопасности) после загрузки клиентской части осуществляется автоматический поиск сервера безопасности (по его заданному IP-адресу или имени), подключение к серверу по протоколу TCP/IP на заданный порт и проведение сеансовой аутентификации компонент системы защиты.

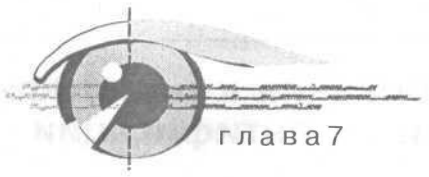
Авторизация

Методы идентификации и аутентификации пользователя



Часть III

- Авторизация и ее задачи
- Парольная защита
- Задачи и методы добавочных механизмов в рамках усиления парольной защиты
- * Сетевая авторизация



Авторизация и ее задачи

7.1. Понятие идентификации и аутентификации. Процедура авторизации

Идентификация призвана каждому пользователю (группе пользователей) сопоставить соответствующую ему разграничительную политику доступа на защищаемом объекте. Для этого пользователь должен себя идентифицировать — указать свое «имя» (идентификатор). Таким образом проверяется, относится ли регистрирующийся пользователь к пользователям, идентифицируемым системой. В соответствии с введенным идентификатором пользователю будут сопоставлены соответствующие права доступа.

Аутентификация предназначена для контроля процедуры идентификации. Для этого пользователь должен ввести секретное слово — пароль. Правильность вводимого пароля подтверждает однозначное соответствие между регистрирующимся пользователем и идентифицированным пользователем.

В общем случае, как будет показано далее, идентифицируются и аутентифицируются не только пользователи, но и другие субъекты доступа к ресурсам.

Совокупность выполнения процедур идентификации и аутентификации принято называть **процедурой авторизации.** При этом заметим, что иногда не требуется идентифицировать пользователя, а достаточно только выполнения процедуры аутентификации. Например, это происходит когда требуется подтвердить текущего (уже зарегистрированного) пользователя при выполнении каких-либо действий, требующих дополнительной защиты. В свою очередь, не всегда требуется осуществлять контроль идентификации, то есть в некоторых случаях аутентификация может не производиться.

Процедура авторизации имеет ключевое значение при защите компьютерной информации, т.к. вся разграничительная политика доступа к ресурсам реализуется относительно идентификаторов пользователей. То есть, войдя в систему с чужим идентификатором, злоумышленник получает права доступа к ресурсу того пользователя, идентификатор которого был им предъявлен при входе в систему.

7.2. Требования к идентификации и аутентификации

7.2.1. Формализованные требования

Формализованные требования к данным механизмам защиты состоят в следующем:

- Должны осуществляться идентификация и проверка подлинности субъектов доступа при входе в систему по идентификатору (коду) и паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов (для классов защищенности 1Г и 1В по классификации АС) [1].
- Система защиты должна требовать от пользователей идентифицировать себя при запросах на доступ.
- Система защиты должна подвергать проверке подлинность идентификации — осуществлять аутентификацию. Для этого она должна располагать необходимыми данными для идентификации и аутентификации.
- Система защиты должна препятствовать доступу к защищаемым ресурсам неидентифицированных пользователей и пользователей, подлинность идентификации которых при аутентификации не подтвердилась (для 5 класса защищенности по классификации СВТ) [2] Для 3 класса защищенности по классификации СВТ вводится дополнительное требование: система защиты должна обладать способностью надежно связывать полученную идентификацию со всеми действиями данного пользователя [2].

Очевидно, что кроме ограничения «...паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов...» данные требования никак не формализуют подходы к реализации механизмов парольной защиты. Кроме того, данные требования не определяют, каким образом должны быть реализованы механизмы парольной защиты, а также не накладывают дополнительных ограничений, связанных с повышением стойкости пароля к подбору. В частности, они не регламентируют использование внешних носителей парольной информации — дискет, смарт-карт и т.д.

7.2.2. Дополнительные требования

Как отмечалось ранее, существует целая группа угроз, связанная с некорректностью реализации процедуры авторизации в современных ОС, а также с наличием ошибок в реализации соответствующих механизмов защиты. Это обусловливает целесообразность рассмотрения механизмов авторизации с целью их добавочной защиты. Кроме того, механизмы идентификации и аутентификации являются важнейшими для противодействия НСД

к информации, а значит, следует рассматривать возможные варианты их резервирования (как «горячего», так и «холодного»).

Кроме того, в рамках декларируемого системного подхода к проектированию системы защиты (рассмотренного нами в четвертой главе), при разработке механизмов авторизации следует рассматривать как явные, так и скрытые угрозы преодоления защиты.

И еще: в рамках системного подхода следует рассматривать не отдельно механизм авторизации как таковой, а необходимую и достаточную совокупность механизмов защиты, призванных в комплексе решать рассматриваемую задачу защиты. Далее мы покажем, что как бы хорошо не был исполнен механизм авторизации, в отдельности он не позволяет обеспечить надежную защиту от несанкционированного входа пользователя в систему.

7.3. Авторизация в контексте количества и вида зарегистрированных пользователей

7.3.1. Кого следует воспринимать в качестве потенциального злоумышленника

В системе зарегистрирован один пользователь

В общем случае в системе может быть зарегистрирован один либо несколько пользователей. Случай, когда в системе зарегистрирован только один пользователь, характеризуется тем, что данный пользователь является и прикладным пользователем, и администратором безопасности. Здесь источником потенциальной угрозы является только сторонний сотрудник предприятия, а вся задача защиты сводится к контролю доступа в компьютер (либо в систему), т.е. к парольной защите.

Данный случай является вырожденным и нами далее не рассматривается, т.к. в соответствии с формализованными требованиями к защите информации от НСД (см. главу 1) даже при защите конфиденциальной информации предполагается обязательное наличие администратора безопасности.

В системе зарегистрированы администратор безопасности и один прикладной пользователь

Общий случай функционирования системы с одним прикладным пользователем — это наличие в системе администратора безопасности и только одного прикладного пользователя. В задачи администратора безопасности здесь входит ограничение прав прикладного пользователя по доступу к системным (администратора безопасности) и иным ресурсам компью-

тера. В частности, может ограничиваться набор задач, разрешенных для решения на компьютере, набор устройств, которые могут быть подключены к компьютеру (например, внешний модем, принтер и т.д.), способ сохранения обрабатываемых данных (например, на дискетах только в шифрованном виде) и т.д.

В данном случае потенциальным злоумышленником в части несанкционированного использования ресурсов защищаемого объекта может являться как сторонний сотрудник предприятия, так и собственно прикладной пользователь. Заметим, что прикладной пользователь здесь может выступать в роли сознательного нарушителя, либо стать «инструментом» в роли стороннего нарушителя, например, запустив по чьей-либо просьбе какую-нибудь программу).

В системе зарегистрированы администратор безопасности и несколько прикладных пользователей

Кроме администратора безопасности, в системе может быть заведено несколько прикладных пользователей. При этом ресурсами защищаемого компьютера могут пользоваться несколько сотрудников, решая различные задачи. Ввиду этого информационные и иные ресурсы защищаемого объекта должны между ними разграничиваться.

В данном случае к потенциальным нарушителям добавляется санкционированный прикладной пользователь, целью которого может служить НСД к информации, хранимой на защищаемом объекте другим пользователем.

При использовании компьютера (прежде всего, рабочей станции) в составе ЛВС, помимо локальных ресурсов защищаемого объекта, защите подлежат сетевые ресурсы. В этом случае между пользователями могут разграничиваться права по доступу к серверам, сетевым службам, разделенным сетевым ресурсам (общим папкам и устройствам, например, к сетевым принтерам) и т.д. Здесь злоумышленник (санкционированный пользователь) может осуществлять попытку получить НСД к сетевому ресурсу, к которому ему доступ не разрешен, с целью осуществления на него атаки с рабочей станции.

7.3.2. Рекомендации по построению авторизации, исходя из вида и количества зарегистрированных пользователей

Отметим, что наиболее простой в реализации защитой является защита от стороннего сотрудника. В этом случае все мероприятия по защите возлагаются на использование механизма парольного входа. Простота же состоит в том, что, как увидим далее, в этом случае следует оказывать противодействие только явным угрозам преодоления парольной защиты, от которых защититься не представляет большого труда.

Однако основной угрозой служат преднамеренные или неумышленные действия санкционированного пользователя, который обладает возможностью осуществления скрытой атаки на защищаемый ресурс (например, запустив какую-либо программу собственной разработки).

Очевидно, что механизмы идентификации и аутентификации должны предусматривать противодействие всем потенциальным злоумышленникам, т.е. как сторонним по отношению к защищаемому объекту, так и санкционированным пользователям, зарегистрированным на компьютере. При этом речь идет о прикладных пользователях, т.к. осуществить какую-либо защиту от НСД к информации от администратора безопасности невозможно, даже включая применение механизмов криптографической защиты (он сумеет снять информацию до момента ее поступления в драйвер шифрования).

С учетом сказанного в этом разделе мы можем сделать следующие выводы:

1. На защищаемом объекте, как правило, зарегистрированы, по крайней мере, два пользователя — прикладной пользователь и администратор безопасности. Поэтому в качестве потенциального злоумышленника при реализации механизмов парольной защиты в общем случае следует рассматривать не только стороннее по отношению к защищаемому объекту лицо, но и санкционированного пользователя, который преднамеренно либо неумышленно может осуществить атаку на механизм парольной защиты.
2. Рассматривая атаки на парольную защиту следует учитывать, что по сравнению со сторонним лицом, которое может характеризоваться явными угрозами парольной защите, защита от атак санкционированного пользователя качественно сложнее, т.к. им могут быть реализованы скрытые угрозы.

7.4. Классификация задач, решаемых механизмами идентификации и аутентификации

7.4.1. Классификация задач по назначению защищаемого объекта

Основу классификации задач, решаемых механизмами парольной защиты, составляет назначение защищаемого объекта (компьютера). Именно в соответствии с назначением объекта определяется перечень защищаемых ресурсов и источников угроз (потенциальных злоумышленников). Соответствующая классификация приведена на рис. 7.1.

Кстати говоря, рассматриваемые процедуры парольной защиты могут устанавливаться на любые действия системы и пользователя (например, на запуск каждого процесса). Однако необходимо понимать, что это, как правило, не оправдано, поскольку приводит к существенной дополнительной загрузке вычислительного ресурса, т.к. данная процедура выполняется не автоматически.

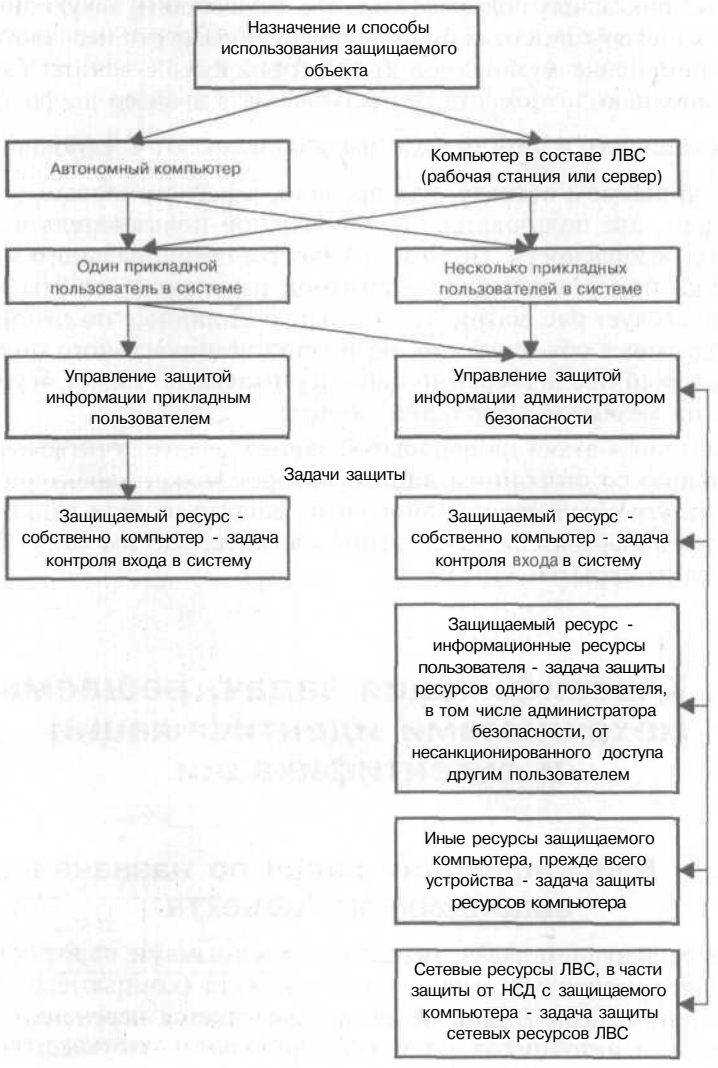


Рис. 7.1. *Классификация задач защиты в соответствии с назначением и способами использования защищаемого объекта*

7.4.2. Возможные классификации механизмов авторизации, реализованных в современных системах защиты

Рассмотрим возможные классификации механизмов идентификации и аутентификации, полученные на основе анализа применения механизмов парольной защиты в ОС (прежде всего семейства Windows), приложениях и современных добавочных средствах защиты ОС.

Классификация по функциональному назначению (классификация решаемых задач) процедур идентификации и аутентификации, применяемых на практике в системах защиты (в том числе и в добавочных средствах защиты), представлена на рис. 7.2.

Классификация по принадлежности идентификаторов и паролей приведена на рис. 7.3, по способу их задания — на рис. 7.4, по способу их ввода — на рис. 7.5, по способу их хранения — на рис. 7.6.

Рис. 7.2. *Классификация по функциональному назначению (цели использования) процедур идентификации и аутентификации, применяемых в системе защиты*

Рис. 7.3. *Классификация по принадлежности идентификатора и пароля*

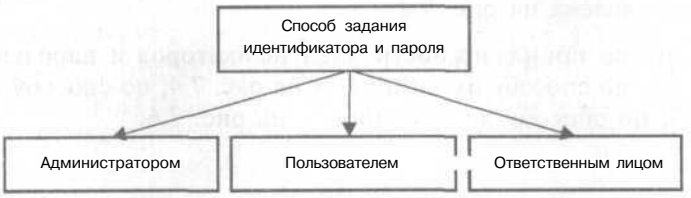


Рис. 7.4. *Классификация по способу задания идентификатора и пароля*

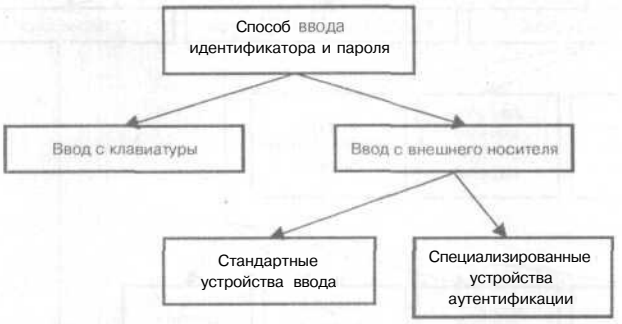


Рис. 7.5. *Классификация по способу ввода идентификатора и пароля*

Рис. 7.6. *Классификация по способу хранения идентификатора и пароля*

глава 8

Парольная защита

8.1. Механизмы парольной защиты

Функциональное назначение механизмов парольной защиты

По функциональному назначению парольный вход, как правило, используется для контроля загрузки системы, контроля функционирования и с целью блокировки. С целью контроля загрузки может устанавливаться процедура идентификации и аутентификации пользователя перед началом загрузки системы, например, встроенными средствами BIOS. В этом случае выполнить загрузку системы сможет только санкционированный пользователь.

Доступ к заданию режима загрузки контролируется штатными средствами BIOS, где после аутентификации пользователь может установить, откуда загружается система — с жесткого диска или с внешнего носителя, а также указать очередность выбора средств загрузки. В качестве контроля доступа к заданию режима загрузки может устанавливаться парольный вход на возможность загрузки в безопасном режиме. Например, для загрузки в режиме Safe Mode для ОС Windows NT/2000/XP пользователю необходимо пройти авторизацию. Загрузиться в аналогичном безопасном режиме в ОС семейства UNIX после авторизации может только пользователь с правами «root».

Для решения задачи контроля функционирования вычислительной системы выделяются:

- **Контроль пользователя при доступе в систему.** Реализуется в том числе штатными средствами ОС.
- **Контроль при запуске процесса.** Благодаря этому при запуске некоторых приложений может быть установлена парольная защита. Прежде всего, здесь интерес представляет установка пароля ответственного лица, например, начальника подразделения (будет рассмотрено ниже).
- **Контроль при доступе к локальным ресурсам.** Например, при доступе к локальному принтеру и т.д. также может использоваться аутентификация ответственного лица.
- **Контроль при доступе к сетевым ресурсам.** Реализуется в том числе штатными средствами ОС. Например, доступ к ресурсам можно разделить паролем. Так осуществляется сетевой доступ к общим ресурсам по протоколу NETBIOS для ОС семейства Windows.

В качестве реакции на несанкционированные действия пользователя системой защиты может устанавливаться блокировка некоторых функций: загрузки системы, доступа в систему, учетных записей пользователя (идентификаторов), запуска определенных приложений. Для снятия блокировки необходима авторизация администратора безопасности или ответственного лица.

Кроме того, пользователь может сам выставить блокировку на доступ к системе и к приложениям и т.д., чтобы доступ в систему и к этим приложениям в его отсутствии был блокирован. Для разблокировки приложения необходимо авторизоваться текущему пользователю. При этом администратор безопасности может блокировать учетные записи пользователей для входа в систему в нерабочее время.

С учетом введенной классификации может быть сделан вывод о функциональном назначении применения механизмов парольной защиты:

- С целью контроля загрузки может устанавливаться возможность контроля пользователя перед началом загрузки системы. Кроме того, контроль пользователя может осуществляться при задании способа и при доступе к заданию режима Загрузки.
- С целью контроля доступа выделяется контроль пользователя при доступе в систему. Также могут иметь место контроль при запуске процесса (прежде всего, здесь интерес представляет установка пароля ответственного лица) и контроль при доступе к локальным и сетевым ресурсам.
- С целью снятия блокировки (реакции) используется контроль администратора безопасности или ответственного лица. Кроме того, пользователь может выставить блокировку на некоторые приложения и т.д. Для их снятия осуществляется контроль пользователя.

Особенности парольной защиты, исходя из принадлежности пароля

С точки зрения принадлежности пароля в классификации выделены «пользователь», к которому относится прикладной пользователь системы и администратор, а также «ответственное лицо», в качестве которого может, например, выступать начальник подразделения. Авторизация ответственного лица может устанавливаться для реализации физического контроля доступа пользователя к ресурсам, прежде всего, к запуску процесса. При этом особенностью здесь является то, что авторизация ответственного лица осуществляется не при доступе в систему, а в процессе функционирования текущего пользователя.

Рассмотрим пример. Пусть требуется обеспечить физически контролируемый доступ к внешней сети, например, к сети Internet. На запуск соответствующего приложения устанавливается механизм авторизации ответственного лица

(его учетные данные хранятся в системе защиты). Тогда при запуске соответствующего приложения появится окно авторизации ответственного лица, и приложение может быть запущено только после его успешной авторизации. При этом приложение запускается только на один сеанс.

Таким образом, приложение физически запускается ответственным лицом с локальной консоли защищаемого объекта. В результате ответственное лицо будет знать, кто и когда запросил доступ в сеть Internet, так как сам принимает решение — разрешать доступ или нет. Если доступ разрешается, ответственное лицо может полностью контролировать данный доступ, т.к. запуск приложения возможен только в его присутствии.

В соответствии с классификацией принадлежности учетной записи введена и классификация способов задания учетных данных (идентификаторов и паролей). Соответсвенно назначение учетных данных могут осуществлять как владелец учетной записи, так и администратор (принудительно).

Реализация механизмов парольной защиты

Ввод идентификатора и пароля может осуществляться, как с применением штатных средств компьютера — клавиатуры, устройств ввода (например, дисковод — с дискеты), так и с использованием специализированных устройств аутентификации — всевозможных аппаратных ключей, биометрических устройств ввода параметров и т.д.

Естественно, что для сравнения вводимой и эталонной информации, эталонные учетные данные пользователей должны где-то храниться. Возможно хранение эталонных учетных данных непосредственно на защищаемом объекте. Тогда при вводе учетных данных из памяти считываются эталонные значения и сравниваются с вводимыми данными.

Кроме того, эталонные данные могут располагаться на сервере. Тогда эталонные значения на защищаемом объекте не хранятся, а вводимые данные передаются на сервер, где и сравниваются с эталоном. При этом именно с сервера разрешается или запрещается доступ субъекту, который ввел учетные данные.

Очевидно, что хранить эталонный пароль как на защищаемом объекте, так и на сервере в открытом виде недопустимо. Поэтому для хранения пароля используется необратимое преобразование (Хеш-функция), позволяющая создавать некий образ пароля -- прямое преобразование. Этот образ однозначно соответствует паролю, но не позволяет осуществить обратное преобразование — из образа восстановить пароль. Образы паролей уже могут храниться на защищаемом объекте, т.к. их знание не позволяет злоумышленнику восстановить исходный пароль. Для реализации необратимого преобразования наиболее часто на сегодняшний день используется алгоритм хеширования MD5.

8.2. Угрозы преодоления парольной защиты

Обобщенная классификация основных угроз парольной защите представлена на рис. 8.1. Данная классификация вводится как в соответствии со статистикой известных угроз, так и в соответствии с потенциально возможными угрозами. Кроме того, при построении данной классификации учитывался анализ принципов работы механизмов идентификации и аутентификации.

Рассмотрим представленные угрозы. Наиболее очевидными явными угрозами являются физические — хищение носителя (например, дискеты с паролем, электронного ключа с парольной информацией и т.д.), а также визуальный съем пароля при вводе (с клавиатуры, либо с монитора). Кроме того, при использовании длинных сложных паролей пользователи подчас записывают свой пароль, что также является объектом физического хищения.

К техническим явным угрозам можно отнести подбор пароля -- либо автоматизированный (вручную пользователем), либо автоматический, предполагающий запуск пользователем специальной программы подбора

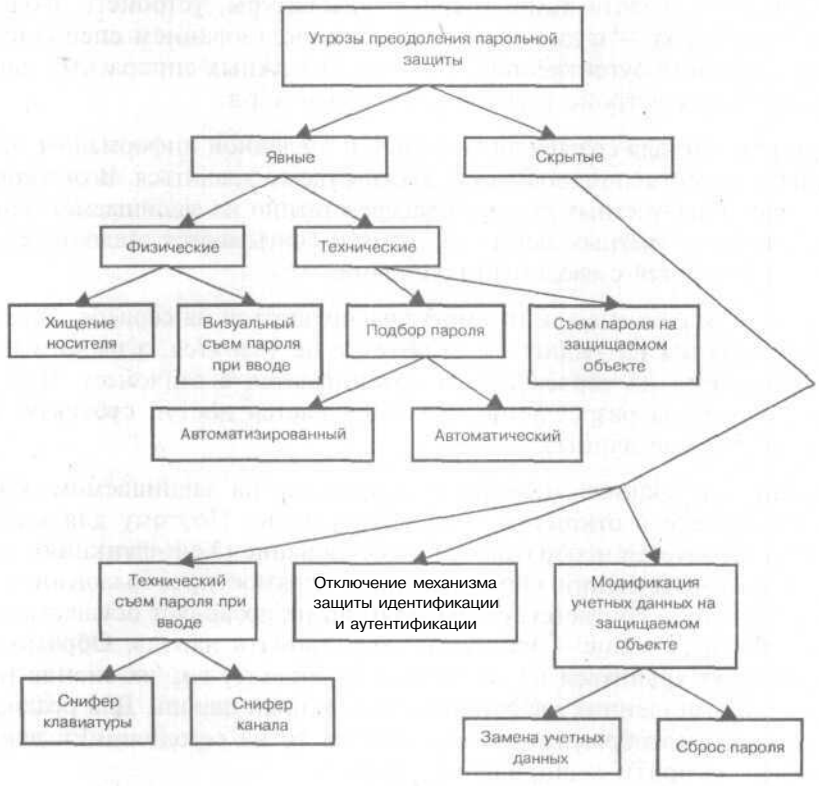


Рис. 8.1. *Обобщенная классификация угроз преодоления парольной защиты*

паролей. Кроме того, для сравнения вводимого и эталонного значений пароля, эталонное значение пароля должно храниться на защищаемом объекте (либо на сервере в сети). Это эталонное значение без соблюдения соответствующих мер по хранению паролей (хеширование, разграничение доступа к области памяти или реестра, где хранятся пароли), может быть похищено злоумышленником.

Естественно, что наиболее опасными являются скрытые угрозы, например:

- технический съем пароля при вводе;
- модификация механизма парольной защиты;
- модификация учетных данных на защищаемом объекте.

Первая группа скрытых угроз наиболее очевидна. Пароль должен быть каким-либо образом введен в систему — с клавиатуры, со встроенного или дополнительного устройства ввода, из сети (по каналу связи). При этом злоумышленником может быть установлена соответствующая программа, позволяющая перехватывать поступающую на защищаемый объект информацию. Развитые подобные программы позволяют автоматически фильтровать перехватываемую информацию по определенным признакам — в том числе, с целью обнаружения паролей. Примером таких программ могут служить сниферы клавиатуры и канала связи. Например, снифер клавиатуры позволяет запоминать все последовательности нажатий кнопок на клавиатуре (здесь пароль вводится в явном виде), а затем фильтровать события по типам приложений.

Злоумышленник, установив подобную программу, и задав режим ее запуска при входе в систему какого-либо пользователя, получит его пароль в открытом виде. Затем, например, троянская программа может выдать этот пароль по сети на другую рабочую станцию. Таким образом, если в системе зарегистрировано несколько пользователей, то один пользователь может узнать пароль другого пользователя, а затем осуществить доступ в систему с правами последнего и т.д.

Второй тип скрытых угроз предполагает возможность отключить механизм парольной защиты злоумышленником, например, загрузить систему с внешнего носителя (дисковода или CD-ROM). Если механизм парольной защиты представляет собой некий процесс (в добавочной системе защиты), то выполнение данного процесса можно остановить средствами системного монитора, либо монитора приложений, например, встроенными средствами в оболочку Far. Подобная возможность существует для ОС Windows 9X/Me.

Третья группа скрытых угроз заключается в модификации учетных данных на защищаемом объекте. Это осуществляется либо путем их замены, либо путем сброса в исходное состояние настроек механизма защиты. Примером может служить известная программная атака на BIOS –

сброс настроек BIOS в исходное состояние посредством изменения контрольных сумм BIOS.

Из сказанного может быть сделан весьма важный вывод, подтверждающий выводы, сделанные ранее: **каким бы надежным ни был механизм парольной защиты, он сам по себе в отдельности, без применения иных механизмов защиты, не может обеспечить сколько-нибудь высокого уровня безопасности защищаемого объекта.**

Другой вывод состоит в том, что невозможно сравнивать между собою альтернативные подходы к реализации механизма защиты (в частности, механизма парольной защиты), так как можно оценивать лишь уровень защищенности, обеспечиваемый всей системой защиты в целом, то есть обеспечиваемый совокупностью механизмов защиты (с учетом их реализации), комплексированных в системе.

8.3. Способы усиления парольной защиты

8.3.1. Основные механизмы ввода пароля. Усиление парольной защиты за счет усовершенствования механизма ввода пароля

В этом разделе мы рассмотрим основные известные на сегодняшний день способы ввода пароля. Все они представлены на рис. 8.2.

Рис. 8.2. *Способы ввода пароля*

Наиболее очевидный способ ввода пароля, который реализован практически во всех ОС, состоит в вводе пароля с клавиатуры. Недостатком данного способа является возможность визуального съема пароля злоумышленником. При этом в меньшей степени опасность представляет набор пароля пользователем на клавиатуре — этому можно противодействовать организационными мерами. В большей степени угроза состоит в том, что при задании сложного пароля пользователь стремится его куда-нибудь записать, чтобы не забыть.

В качестве противодействия угрозе визуального съема пароля могут использоваться внешние носители информации. При этом могут использоваться как стандартные средства ввода информации (например, дискета), так и средства, предполагающие подключение специальных средств ввода парольной информации — всевозможные электронные ключи, «таблетки» и т.д. На этих носителях записывается пароль, который считывается системой при аутентификации пользователя. Здесь может задаваться достаточно большая длина пароля без угрозы его визуального съема. Применение для ввода пароля стандартного или специального носителя с точки зрения обеспечиваемого уровня безопасности практически равноценно. Вопрос выбора носителя определяется его ценой, долговечностью, удобством хранения.

Примечание

Дополнительные носители парольной информации для усиления парольной защиты ОС используются практическими всеми современными средствами добавочной защиты.

Недостатком применения внешних носителей информации для ввода пароля является потенциальная угроза его хищения злоумышленником.

Для противодействия хищению злоумышленником носителя информации с паролем могут рассматриваться следующие альтернативные способы защиты:

- Использование биометрических характеристик пользователя — подход, позволяющий отказаться от внешнего носителя с паролем как такового. При этом идентификатором пользователя становятся его биометрические параметры. Причем ввиду их однозначного соответствия пользователю эти параметры служат одновременно и паролем.
- Комбинирование способа ввода пароля с клавиатуры и способа ввода пароля с внешнего носителя. Например, механизм ввода пароля с клавиатуры может рассматриваться как дополнение к механизму ввода пароля с внешнего носителя.

Комбинированный способ осуществляется двумя механизмами, один из который является основным, а другой — дополнительным. На наш взгляд, при защите компьютеров имеет смысл использовать следующий комбинированный способ ввода пароля:

- основной — с внешнего носителя (целесообразно реализовать добавочными средствами защиты),
- дополнительный - - с клавиатуры (для этого могут использоваться встроенные в ОС механизмы авторизации пользователя).

Таким образом можно выделить следующие приоритеты в использовании механизмов ввода пароля (расположены в порядке убывания):

1. Биометрический способ идентификации пользователя.
2. Комбинированный способ (консольный ввод + использование внешнего носителя).

3. Ввод пароля с внешнего носителя.
4. Консольный ввод (ввод пароля с клавиатуры).

Соответственно механизмы парольной защиты можно усиливать уже на этапе ввода пароля. Для этого необходимо в системе предусмотреть наиболее приоритетный способ ввода пароля.

8.3.2. Основное достоинство биометрических систем контроля доступа

В двух словах остановимся на рассмотрении новых свойств парольной защиты, реализуемых на основе контроля биометрических характеристик пользователя.

Гипотетически возможна угроза, связанная с тем, что один пользователь передает свои парольные данные другому пользователю, а тот воспользуется ими для несанкционированного входя в систему последним (в определенном смысле это можно трактовать, как изменение ПРД к ресурсам пользователем не администратором безопасности, что противоречит формальным требованиям к системе защиты).

В общем же случае механизмы биометрической идентификации пользователя (естественно, при их корректной реализации) предотвращают возможность какой-либо передачи парольной информации между пользователями. А это достаточно важно при реализации централизованной (без участия пользователя) схемы администрирования механизмов защиты. В этом и заключается несомненное достоинство данных подходов парольной защиты по сравнению с применением внешних аппаратных носителей парольных данных (всевозможных ключей, смарт-карт и т.д.). Другими словами, **корректно в общем случае концепция централизованного администрирования системы защиты может быть реализована с применением биометрических систем контроля доступа.**

Достоинством же применения внешних аппаратных носителей парольных данных является большая универсальность в смысле возможности хранения учетных данных. То есть на них может храниться не только информация, идентифицирующая пользователя, но и ключи шифрования, а также иные данные.

8.3.3. Основные способы усиления парольной защиты, используемые в современных ОС и приложениях

Классификация основных способов усиления пароля, используемых в современных ОС и в приложениях, представлена на рис. 8.3. Все они призваны воспрепятствовать подбору пароля злоумышленником.

Рис. *8.3.* *Способы усиления пароля*

8.3.4. Анализ способов усиления парольной защиты

Проиллюстрируем цели применения рассматриваемых способов усиления пароля. При этом опустим некоторые уникальные подходы, как например, учет параметров, характеризующих процедуру ввода пароля пользователем — скорость набора символов и др. То есть не будем рассматривать методы, которые едва ли применимы в распространенных приложениях. Собственно расчетные формулы оценки сложности подбора пароля в работе не приводятся ввиду их тривиальности [8, 13].

Пусть A — исходный алфавит для задания пароля (некоторое число символов, включая их типы, для назначения пароля), а L — длина пароля. В этих предположениях число возможных парольных комбинаций составит:

$$R = f(A,L).$$

Обозначим вероятность подбора злоумышленником пароля с одной попытки $P\backslash$ (в предположении, что все парольные комбинации равновероятны: $P\backslash = 1/R$). Если для подбора пароля злоумышленником совершается п попыток в единицу времени t, то за интервал времени T (число единиц времени), вероятность подбора пароля злоумышленником будет описываться следующей зависимостью:

$$P = F(A,\ L,\ P1,\ n(t),\ T).$$

Теперь, в соответствии с полученной зависимостью, рассмотрим, какие способы усиления пароля (рис. 8.3) на какой параметр призваны влиять.

Применение способов усиления пароля, посредством задания дополнительных требований к параметрам пароля в соответствии с зависимостью $R = / (A, L)$ призваны увеличить число возможных парольных комбинаций.

Ограничения на «число типов символов в пароле», «возможность задания простых паролей», и на «повторяемость паролей» задаются с целью уменьшения параметра P1, то есть с целью, по возможности, обеспечить равновероятность для злоумышленника всех исходных парольных комбинаций.

Ограничение на «число неверно введенных значений пароля» реализует возможность совершить пользователю только заданное число N попыток подбора пароля. В случае превышения этого количества может либо блокироваться учетная запись данного пользователя, либо блокироваться защищаемый объект в целом.

Данное ограничение является одним из важнейших, т.к. оно призвано противодействовать возможности автоматического подбора паролей. В этом случае число попыток подбора злоумышленником жестко фиксировано параметром N и исходная зависимость для вероятности подбора пароля злоумышленником принимает следующий вид:

$$P= F(A, L, P1, N).$$

Данное ограничение можно рассматривать как альтернативу применению способов усиления пароля, основанных на дополнительных требованиях к параметрам пароля в соответствии с зависимостью $R = f(A, L)$. То есть, уменьшая параметр N, тем самым можно снижать требования к параметру L. А это, как отмечали ранее, крайне важно при использовании механизма парольной защиты, предполагающего ввод пароля с клавиатуры. Очевидно, что без использования данного ограничения с учетом больших темпов роста производительности компьютеров, приводящего к заметному увеличению параметра $n(t)$, без применения данного ограничения соответственно возрастают требования к параметрам A, L.

Ранее было отмечено, что в основе проектирования системы защиты должен лежать системный подход. В рамках этого подхода при проектировании механизмов парольной защиты необходимо учитывать наличие других механизмов в системе защиты.

Так, если в системе защиты обеспечена замкнутость программной среды, которая не позволит пользователю запустить программу подбора паролей, т.е. реализовать автоматический способ подбора, то не столь актуальным становится и реализация ограничения на число неверно введенных значений пароля. Действительно, в этом случае параметр $n(t)$ уже имеет значе-

ние: 1 попытка за 5...15 с. (определяется скоростью ручного ввода пароля пользователем), что не позволит сколько-нибудь эффективно противодействовать парольной защите.

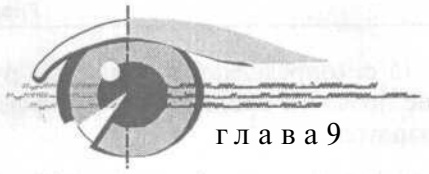
С учетом сказанного может быть сделан следующий важный вывод: существуют альтернативные подходы к усилению пароля с целью преодоления возможности его подбора. Наиболее эффективным из них можно признать «Ограничение на число неверно введенных значений пароля», использование которого позволяет существенно снизить требования к длине пароля. Однако аналогичный результат достигается, если в системе защиты обеспечивается замкнутость программной среды, противодействующая выполнению автоматического (программного) подбора пароля.

Кроме того, в случае, если не используется ограничение на число неверно введенных значений пароля, вероятность подбора пароля зависит не только от параметра $n(t)$, но и от параметра T. *При* этом данное ограничение устанавливается на параметр T, позволяя снизить суммарное число попыток подбора для одного установленного значения пароля (за счет ограничения отведенного на это времени).

Примечание

Возможность ограничения «на число неверно введенных значений пароля» является важнейшей в части усиления парольной защиты и, наряду с другими рассмотренными выше возможностями, присутствует в механизмах парольной защиты практически всех современных ОС большинства приложений. При этом в различных семействах ОС возможности установки ограничений на пароль различаются незначительно.

Дополнительно в системах парольной защиты может использоваться ограничение на периодичность смены пароля пользователем, которое призвано ограничить возможность использования одного и того же значения пароля в течение сколько-нибудь продолжительного времени (например, более месяца).



глава 9

Задачи и методы добавочных механизмов в рамках усиления парольной защиты

9.1. Требования к добавочным механизмам в рамках усиления парольной защиты

Как отмечалось ранее, важнейшими способами усиления парольной защиты является комплексное использование механизмов, так как сам по себе механизм парольной защиты не может обеспечить гарантированной защиты от несанкционированного входа в систему.

Рассмотрим группу задач, которые должны решаться средствами добавочной защиты в общем случае, без учета реализации встроенных в ОС механизмов защиты.

Добавочной защитой в рамках противодействия скрытым угрозам преодоления механизма парольной защиты должны решаться следующие задачи:

- должна обеспечиваться замкнутость программной среды, не позволяющая запускать пользователю в системе программы с целью осуществления нерегламентированных действий, например, запуск программ-сниферов (канала связи, каналов ввода с внешних устройств и с клавиатуры) и иных потенциально опасных программ, в частности, позволяющих модифицировать механизм парольной защиты;
- должно обеспечиваться разграничение прав доступа к реестру ОС (для ОС семейства Windows), к каталогам и файлам, хранящим учетные данные механизмов парольной защиты для ОС UNIX;
- должен обеспечиваться контроль целостности (с возможностью автоматического восстановления из эталонной копии) ключей и ветвей реестра, а также файлов, хранящих учетные данные механизмов парольной защиты, применяемый в предположении, что разграничение прав доступа к реестру ОС, каталогам и файлам злоумышленником преодолены.

» должна обеспечиваться защита от возможности отключения механизма защиты, в частности, возможности загрузки системы с внешних устройств, останова процесса или драйвера, реализующего парольную защиту добавочными средствами;

* должно обеспечиваться хеширование паролей, применяемое в предположении, что разграничение прав доступа к реестру ОС, каталогам и файлам злоумышленником преодолены, в том числе с использованием ошибок и закладок в ПО.

9.2. Необходимые механизмы добавочной защиты, направленные на усиление парольной защиты

С учетом сказанного можно выделить ряд механизмов, которые должны быть реализованы с целью усиления парольной защиты от скрытых угроз на защищаемом объекте. В общем случае эти механизмы таковы:

» Механизм обеспечения замкнутости программной среды.

◆ Механизм разграничения прав доступа к настройкам ОС и приложений (к реестру ОС для ОС семейства Windows, к каталогам и файлам настроек для ОС семейства Unix).

◆ Механизм контроля целостности с возможностью автоматического восстановления из резервной копии настроек ОС и приложений (ключей реестра ОС для ОС семейства Windows, файлов для ОС семейства Unix).

• Механизм защиты от возможности отключения парольной защиты, в частности, возможности загрузки системы с внешних устройств, останова процесса или драйвера, реализующего парольную защиту добавочными средствами.

» Механизм хеширования парольной информации и др.

9.3. Применяющиеся на сегодняшний день подходы. Двухуровневая авторизация

Сформулировав требования, рассмотрим, какие же на сегодняшний день используются подходы для усиления механизмов парольной защиты добавочными средствами.

9.3.1. Двухуровневая авторизация на уровне ОС

Ранее отмечалось, что целесообразно усилить парольную защиту за счет включения в механизм идентификации и аутентификации возможности ввода пароля с внешнего носителя.

Реализация данной возможности имеет смысл, прежде всего, для однопользовательских ОС, к которым относятся ОС семейства Windows (под однопользовательской здесь понимаем систему, в которой в каждый момент времени может присутствовать только один прикладной пользователь). Здесь пользователь авторизуется при консольном входе в систему. Для многопользовательских ОС, к которым относятся ОС семейства UNIX, характерна удаленная работа на компьютере многих пользователей, поэтому рассматриваемая задача дополнительной авторизации здесь не столь актуальна.

В простейшем случае рассматриваемая возможность усиления парольной защиты достигается подключением внешних носителей пароля к встроенным (прежде всего для ОС Windows) механизмам идентификации и аутентификации. При этом средства разработчика, поставляемые изготовителем аппаратных носителей парольной информации, как правило, содержат соответствующие примеры решения данной задачи.

Однако, как отмечалось ранее, имеет смысл реализовать режим двухуровневой авторизации — так называемый комбинированный режим. При этом авторизация производится сначала с внешнего устройства, а затем с клавиатуры. Это позволяет оказывать противодействие несанкционированному входу в систему при хищении носителя с парольной информацией.

Для современных ОС может быть реализован комбинированный способ авторизации, использующий развитый механизм авторизации пользователя при входе в систему с консоли (например, ОС Windows NT/2000). Это, собственно, и составляет одну из основных задач добавочной защиты ОС.

Рассмотрим возможное техническое решение по комбинированию механизмов добавочной и встроенной парольной защиты входа в систему. Схема двухуровневой авторизации представлена на рис. 9.1.

Примечание

На практике это решение реализовано, например, в КСЗИ «Панцирь».

Работает схема следующим образом. Сначала со входа 8 задается пароль для аутентификации механизмом добавочной защиты (значение этого пароля записывается на внешний носитель, например, на дискету). Затем со входа 9 задается пароль для аутентификации встроенным механизмом защиты. Этот пароль заносится и в блок 4, и в блок 2. При запросе доступа в систему (со входа 7) блок 2 запускает интерфейс для ввода пользователя пароля со входа 6 (с внешнего носителя).

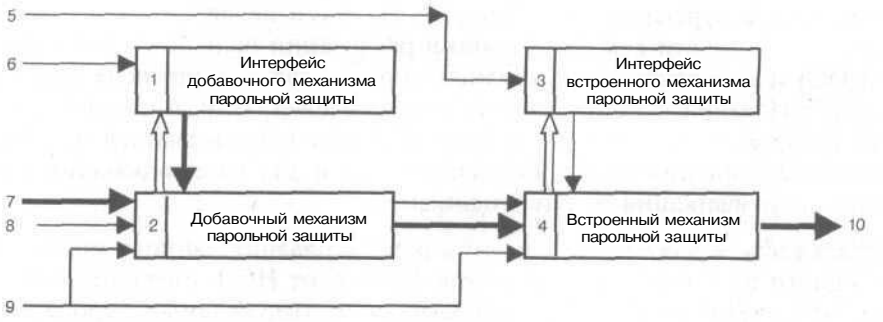


Рис. 9.1. Схема двухуровневой авторизации

При успешной аутентификации блоком *2* выдается в блок 4 запрос пользователя на доступ в систему и системный пароль. Причем системный пароль подставляется блоком 2 по запросу аутентификации пользователя блоком 4 без запуска интерфейса 3. Если пароль совпадает, то на выход 10 выдается сигнал об успешном прохождении пользователем процедуры аутентификации. В противном случае, блоком 4 запускается блок 3, и пользователю предлагается пройти аутентификацию со входа 5 в стандартном окне системной аутентификации — в блоке 3. При успешной аутентификации вырабатывается сигнал на выходе 10.

Таким образом, представленная схема позволяет реализовать два режима аутентификации — одноуровневый и двухуровневый. При одноуровневом режиме со входа *9* в блоки 2 и 4 заносятся одинаковые значения системного пароля для пользователя, а пользователь аутентифицируется только в интерфейсе добавочного механизма парольной защиты. При этом системную аутентификацию за пользователя реализует уже собственно механизм добавочной парольной защиты. Делает он это без выдачи соответствующего окна системной аутентификации на экран монитора.

При двухуровневом режиме (со входа *9* в блоки 2 и 4 заносятся различные значения системного пароля для пользователя), пользователь сначала аутентифицируется в интерфейсе добавочного механизма парольной защиты с использованием внешнего носителя с паролем. Затем, при успешной аутентификации, ему предлагается пройти второй уровень – уровень системной аутентификации в окне ОС. Для этого ему нужно ввести пароль с клавиатуры.

Таким образом, помимо ввода пароля с внешнего носителя, пользователю дополнительно предлагается ввести пароль с клавиатуры. Это предотвращает неконтролируемый вход пользователя в систему при хищении внешнего носителя с паролем. При этом заметим, что требования к дополнительному паролю уже могут быть не столь жесткими, как к основному.

Требованием к реализации данного механизма является отсутствие какой-либо общности как собственно в реализации основного (встроенного в ОС) и добавочного механизмов защиты, так и в способах хранения и преобразования учетных данных пользователей. При выполнении данного требования добавочный механизм может использоваться не только в качестве усиления парольной защиты, но и для резервирования механизма идентификации и аутентификации.

Актуальность и даже необходимость резервирования данного механизма, как одного из основных механизмов защиты от НСД, очевидна. Напомним, что учетные данные пользователя, подтверждаемые паролем при входе пользователя в систему, являются основой задания разграничительной политики доступа к ресурсам.

9.3.2. Двухуровневая авторизация на уровне BIOS

Альтернативный известный вариант реализации двухуровневой парольной защиты состоит в реализации добавочного механизма не на уровне входа в ОС, а перед загрузкой системы (средствами расширения BIOS). При этом перед загрузкой системы добавочное средство защиты предлагает пользователю авторизоваться с использованием внешнего носителя пароля. Затем уже при входе в систему могут использоваться штатные средства авторизации ОС. Таким образом, в отличие от рассмотренного выше подхода, здесь заменяется (дополняется) не механизм парольной защиты входа в систему, а механизм парольной защиты BIOS (контроля загрузки системы). Данный подход используется в специальных дополнительных программно-аппаратных средствах защиты, так называемых «Электронных замках». (При этом, несмотря на использование аппаратной компоненты, собственно защита осуществляется программно).

К достоинствам данного подхода, по сравнению с рассмотренным выше, можно отнести противодействие скрытым угрозам непосредственно в процессе загрузки системы. В частности, загрузочное меню ОС здесь уже может быть вызвано только санкционированным пользователем.

К очевидным недостаткам данного подхода можно отнести необходимость использования дополнительной аппаратной компоненты системы защиты — платы, реализующей функцию расширения BIOS.

Примечание

В дальнейшем (в шестой части книги) будет показано, что на наш взгляд аппаратную компоненту можно наделить несколько иными свойствами.

Сетевая авторизация

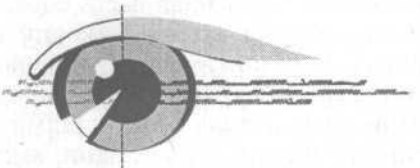
Описанные ранее способы двухуровневой авторизации могут быть реализованы как в локальном, так и в сетевом исполнении. Оба данных подхода могут осуществлять локальную авторизацию добавочным средством при использовании сетевой авторизации пользователя на контроллере домена встроенным в ОС механизмом. Кроме того, для двухуровневой авторизации на уровне ОС оба уровня защиты могут быть выполнены в виде сетевой авторизации. При этом дополнительная авторизация осуществляется на сервере безопасности (т.е. пароли добавочного средства защиты хранятся не на защищаемом компьютере, а на сервере безопасности).

В этом случае механизм добавочной защиты содержит клиентскую часть, устанавливаемую на защищаемый компьютер, и серверную часть, устанавливаемую на сервер безопасности. Серверная часть содержит учетную запись пользователя, а также пароли для аутентификации пользователя добавочным механизмом защиты и на контроллере домена при выполнении процедуры авторизации, серверная часть функционирует аналогично контроллеру домена. При этом наряду с функциями контроллера домена она реализует собственно процедуру авторизации, выдавая на клиентскую часть (на компьютер, где осуществлена загрузка ОС) окно ввода идентификатора и пароля пользователем. Клиентская же часть осуществляет сокрытие окна авторизации контроллера домена и подстановку пароля ОС, который поступает с серверной части при прохождении авторизации пользователем в добавочном механизме защиты.

Отметим, что применение сетевой авторизации в данном случае, не давая каких-либо преимуществ по защите пароля (предполагаем, что на защищаемом компьютере установлено добавочное средство защиты, локально обеспечивающее надежную защиту учетных и парольных данных пользователя), существенно сказывается на надежности функционирования защищаемой сети. Действительно, выход из строя сервера безопасности приводит к отказу всей сети. При этом ни на одном компьютере сети не сможет быть осуществлен вход пользователя в систему. Таким образом, надежность сети в целом (а в общем случае и всех информационных систем, реализованных на базе данной сети) в данном случае определяется надежностью одного компьютера. Это, по мнению автора, недопустимо в большинстве приложений информационных систем. И прежде всего это касается тех из них, которые предполагают распределенную обработку данных.

С учетом сказанного могут быть сформулированы следующие рекомендации по реализации механизмов двухуровневой авторизации: дополнительный механизм защиты должен быть реализован локально. При этом системой защиты должно обеспечиваться надежное хранение парольной информации на защищаемом компьютере. Встроенный в ОС механизм может быть настроен как в режиме локальной, так и в режиме сетевой авторизации, в зависимости от задач, решаемых в сети.

Управление
доступом к ресурсам



Часть IV

- Требования, подходы и задачи управления доступом
- Модели управления доступом
- Реализация моделей доступа механизмами добавочной и встроенной защиты
- Субъект доступа «Процесс» и его учет при разграничении доступа
- Диспетчер доступа
- Практическая реализация механизмов добавочной защиты

глава 11

Требования, подходы и задачи управления доступом

11.1. Общие положения

Механизмы управления доступом являются основой защиты ресурсов, обеспечивая решение задачи разграничения доступа субъектов к защищаемым информационным и техническим ресурсам -- объектам [3]. В качестве субъектов в простейшем случае понимается пользователь. Однако в дальнейшем это понятие будет нами расширено.

На практике наличие механизмов управления доступом необходимо, даже если в системе может находиться только один прикладной пользователь. Это вызвано тем, что, как правило, в системе должен быть также заведен пользователь с правами администратора, который настраивает параметры системы защиты и права доступа к ресурсам защищаемого объекта. При этом у администратора принципиально иные права, чем у прикладного пользователя. Обо всем этом мы говорили в предыдущих главах книги.

11.1.1. Абстрактные модели доступа

Механизм управления доступом реализует на практике некоторую абстрактную (или формальную) модель [3, 4, 8], определяющую правила задания разграничительной политики доступа к защищаемым ресурсам и правила обработки запросов доступа к защищаемым ресурсам.

Модель Биба

Одной из первых моделей была опубликованная в 1977 модель Биба (Biba). Согласно этой модели все субъекты и объекты предварительно разделяются по нескольким уровням доступа. Затем на их взаимодействия накладываются следующие ограничения:

« субъект не может вызывать на исполнение субъекты с более низким уровнем доступа;
» субъект не может модифицировать объекты с более высоким уровнем доступа.

Эта модель очень напоминает ограничения, введенные в защищенном режиме микропроцессоров Intel 80386+ относительно уровней привилегий.

Модель Гогена-Мезигера

Модель Гогена-Мезигера (Goguen-Meseguer), представленная ими в 1982 году, основана на теории автоматов. Согласно этой модели система может при каждом действии переходить из одного разрешенного состояния только в несколько других. Субъекты и объекты в данной модели защиты разбиваются на группы — домены.

Переход системы из одного состояния в другое выполняется только в соответствии с так называемой таблицей разрешений, в которой указано, какие операции может выполнять субъект, например, из домена С над объектом из домена D. В данной модели при переходе системы из одного разрешенного состояния в другое используются транзакции, что обеспечивает общую целостность системы.

Сазерлендская модель

Сазерлендская (от англ. Sutherland) модель защиты, опубликованная в 1986 году, основана на взаимодействии субъектов и потоков информации. Так же как и в предыдущей модели, здесь используется машина состояний со множеством разрешенных комбинаций состояний и некоторым набором начальных позиций. В данной модели исследуется поведение множественных композиций функций перехода из одного состояния в другое.

Модель Кларка-Вильсона

Важную роль в теории защиты информации играет модель защиты Кларка-Вильсона (Clark-Wilson), опубликованная в 1987 году и модифицированная в 1989. Основана данная модель на повсеместном использовании транзакций и тщательном оформлении прав доступа субъектов к объектам. В данной модели впервые исследована защищенность третьей стороны •- стороны, поддерживающей всю систему безопасности. Эту роль в информационных системах обычно играет программа-супервизор.

Кроме того, в модели Кларка-Вильсона транзакции впервые были построены по методу верификации, то есть идентификация субъекта производилась не только перед выполнением команды от него, но и повторно после выполнения. Это позволило снять проблему подмены субъекта в момент между его идентификацией и собственно командой. Модель Кларка-Вильсона считается одной из самых совершенных в отношении поддержания целостности информационных систем.

Дискреционная (матричная) модель

Рассмотрим так называемую матричную модель защиты (ее еще называют дискреционной моделью), получившую на сегодняшний день наибольшее распространение на практике. В терминах матричной модели, состояние системы защиты описывается следующей тройкой:

(S, O, M),

где *S* — множество субъектов, являющихся активными структурными элементами модели;

O — множество объектов доступа, являющихся пассивными защищаемыми элементами модели. Каждый объект однозначно идентифицируется с помощью имени объекта;

M -- матрица доступа. Значение элемента матрицы *M [S, 0}* определяет права доступа субъекта *S к* объекту *O.*

Права доступа регламентируют способы обращения субъекта *S* к различным типам объектов доступа. В частности, права доступа субъектов к файловым объектам обычно определяют как чтение (*R*), запись (*W*) и выполнение (*E*).

Основу реализации управления доступом составляет анализ строки матрицы доступа при обращении субъекта к объекту. При этом проверяется строка матрицы, соответствующая объекту, и анализируется есть ли в ней разрешенные прав доступа для субъекта или нет. На основе этого принимается решение о предоставлении доступа.

При всей наглядности и гибкости возможных настроек разграничительной политики доступа к ресурсам, матричным моделям присущи серьезные недостатки. Основной из них -- это излишне детализированный уровень описания отношений субъектов и объектов. Из-за этого усложняется процедура администрирования системы защиты. Причем это происходит как при задании настроек, так и при поддержании их в актуальном состоянии при включении в схему разграничения доступа новых субъектов и объектов. Как следствие, усложнение администрирования может приводить к возникновению ошибок.

Более подробно дискреционная модель управления доступом рассмотрена в п. 11.1.2.

Многоуровневые (мандатные) модели

С целью устранения недостатков матричных моделей были разработаны так называемые многоуровневые модели защиты, классическими примерами которых являются модель конечных состояний Белла и Ла-Падулы, а также решетчатая модель Д. Деннинг. Многоуровневые модели предполагают формализацию процедуры назначения прав доступа посредством использования так называемых меток конфиденциальности или мандатов, назначаемых субъектам и объектам доступа.

Так, для субъекта доступа метки, например, могут определяться в соответствии с уровнем допуска лица к информации, а для объекта доступа (собственно данные) -- признаками конфиденциальности информации. Признаки конфиденциальности фиксируются в метке объекта.

Примечание

В связи с использованием терминов «мандат», «метка», «полномочия» многоуровневую защиту часто называют соответственно либо мандатной защитой, либо защитой с метками конфиденциальности, либо полномочной защитой.

Права доступа каждого субъекта и характеристики конфиденциальности каждого объекта отображаются в виде совокупности уровня конфиденциальности и набора категорий конфиденциальности. Уровень конфиденциальности может принимать одно из строго упорядоченного ряда фиксированных значений, например: конфиденциально, секретно, для служебного пользования, несекретно и т.п.

Основу реализации управления доступом составляют:

1. Формальное сравнение метки субъекта, запросившего доступ, и метки объекта, к которому запрошен доступ.
2. Принятие решений о предоставлении доступа на основе некоторых правил, основу которых составляет противодействие снижению уровня конфиденциальности защищаемой информации.

Таким образом, многоуровневая модель предупреждает возможность преднамеренного или случайного снижения уровня конфиденциальности защищаемой информации за счет ее утечки (умышленного переноса). То есть эта модель препятствует переходу информации из объектов с высоким уровнем конфиденциальности и узким набором категорий доступа в объекты с меньшим уровнем конфиденциальности и более широким набором категорий доступа.

Практика показывает, что многоуровневые модели защиты находятся гораздо ближе к потребностям реальной жизни, нежели матричные модели, и представляют собой хорошую основу для построения автоматизированных систем разграничения доступа. Причем, так как отдельно взятые категории одного уровня равнозначны, то, чтобы их разграничить наряду с многоуровневой (мандатной) моделью, требуется применение матричной модели.

С помощью многоуровневых моделей возможно существенное упрощение задачи администрирования (настройки). Причем это касается как исходной настройки разграничительной политики доступа (не требуется столь высокого уровня детализации задания отношения субъект-объект), так и последующего включения в схему администрирования новых объектов и субъектов доступа.

Более подробно мандатная модель доступа рассмотрена в п. 11.1.3.

Выбор модели

Следуя формализованным требованиям к системе защиты информации, основой реализации разграничительной политики доступа к ресурсам при обработке сведений конфиденциального характера является **дискреционный механизм управления доступом,** а секретных сведений -- **мандатный механизм управления доступом** [1, 2].

Как было показано ранее, в теории защиты информации известны и иные модели управления доступом [8]. Причем нами были рассмотрены лишь основные из них. Однако, ввиду их более теоретического, нежели практического интереса, а также с учетом ярко выраженной практической направленности книги на рассмотрении данных подходов мы останавливаться не станем. Мы будем рассматривать решения, реализованные на практике встроенными и добавочными средствами защиты.

При этом в данной работе рассматриваются две основных модели, нашедших отражение в соответствующих нормативных документах в области защиты информации [1, 2] и реализуемых на практике современными ОС и добавочными средствами защиты. Это дискреционная и мандатная модели управления доступом.

Дальнейшее описание дискреционного и мандатного механизмов управления доступом представим в виде рассмотрения формализованных требований к соответствующим механизмам защиты.

11.1.2. Дискреционная модель управления доступом

Следуя формализованным требованиям к системе защиты информации, основой реализации разграничительной политики доступа к ресурсам при обработке сведений конфиденциального характера является дискреционный механизм управления доступом. При этом к нему предъявляются следующие требования [1]:

* Система защиты должна контролировать доступ наименованных субъектов (пользователей) к наименованным объектам (файлам, программам, томам и т.д.).
« Для каждой пары (субъект — объект) в средстве вычислительной техники (СВТ) должно быть задано явное и недвусмысленное перечисление допустимых типов доступа (читать, писать и т.д.), т.е. тех типов доступа, которые являются санкционированными для данного субъекта (индивида или группы индивидов) к данному ресурсу СВТ (объекту).
» Система защиты должна содержать механизм, претворяющий в жизнь дискреционные правила разграничения доступа.
» Контроль доступа должен быть применим к каждому объекту и каждому субъекту (индивиду или группе равноправных индивидов).

» Механизм, реализующий дискреционный принцип контроля доступа, должен предусматривать возможности санкционированного изменения правил или прав разграничения доступа (ПРД), в том числе возможность санкционированного изменения списка пользователей СВТ и списка защищаемых объектов.

» Право изменять ПРД должно предоставляться выделенным субъектам (администрации, службе безопасности и т.д.).

* Должны быть предусмотрены средства управления, ограничивающие распространения прав на доступ.

11.1.3. Мандатная модель управления доступом

Основу реализации разграничительной политики доступа к ресурсам при защите секретной информации является требование к реализации, помимо дискреционного, мандатного механизма управления доступом. Требования к мандатному механизму состоят в следующем [1]:

* Каждому субъекту и объекту доступа должны сопоставляться классификационные метки, отражающие их место в соответсвующей иерархии (метки конфиденциальности). Посредством этих меток субъектам и объектам должны назначаться классификационные уровни (уровни уязвимости, категории секретности и т. п.), являющиеся комбинациями иерархических и неиерархических категорий. Данные метки должны служить основой мандатного принципа разграничения доступа.

» Система защиты при вводе новых данных в систему должна запрашивать и получать от санкционированного пользователя классификационные метки этих данных. При санкционированном занесении в список пользователей нового субъекта ему должны назначаться классификационные метки. Внешние классификационные метки (субъектов, объектов) должны точно соответствовать внутренним меткам (внутри системы защиты).

» Система защиты должна реализовывать мандатный принцип контроля доступа применительно ко всем объектам при явном и скрытом доступе со стороны любого из субъектов:

- субъект может читать объект, только если иерархическая классификация в классификационном уровне субъекта не меньше, чем иерархическая классификация в классификационном уровне объекта. При этом иерархические категории в классификационном уровне субъекта должны включать в себя все иерархические категории в классификационном уровне объекта;
- субъект осуществляет запись в объект, только если классификационный уровень субъекта в иерархической классификации не больше, чем классификационный уровень объекта в иерархической классификации. При этом все иерархические категории в

классификационном уровне субъекта должны включаться в иерархические категории в классификационном уровне объекта.

- Реализация мандатных ПРД должна предусматривать возможность сопровождения, изменения классификационных уровней субъектов и объектов специально выделенными субъектами.
- В СВТ должен быть реализован диспетчер доступа, т.е. средство, во-первых, осуществляющее перехват всех обращений субъектов к объектам, а во-вторых, разграничивающее доступ в соответствии с заданным принципом разграничения доступа. При этом решение о санкционированности запроса на доступ должно приниматься только при одновременном разрешении его и дискреционными, и мандатными ПРД. Таким образом, должны контролироваться не только единичный акт доступа, но и потоки информации.

11.1.4. Дополнительные требования к защите секретной информации в контексте использования дискреционной и мандатной моделей управления доступом

Требования непосредственно к дискреционному и мандатному механизмам

При защите секретной информации используется и дискреционная и мандатная модели управления доступом. Требования к реализации мандатной модели в рамках защиты секретной информации были приведены в предыдущем пункте (в п. 11.1.3). Что касается требований к дискреционному механизму, то при защите секретной информации следует придерживаться тех же требований, что и для защиты конфиденциальной информации (см. п. 11.1.2). Однако в дополнение к последним добавляется еще пара требований:

- Система защиты должна содержать механизм, претворяющий в жизнь дискреционные ПРД, как для явных действий пользователя, так и для скрытых, обеспечивая тем самым защиту объектов от НСД (т.е. от доступа, недопустимого с точки зрения заданного ПРД). Под «явными» [1] подразумеваются действия, осуществляемые с использованием системных средств — системных макрокоманд, инструкций языков высокого уровня и т.д. Под «скрытыми» — иные действия, в том числе с использованием злоумышленником собственных программ работы с устройствами.
- Дискреционные ПРД для систем данного класса являются дополнением мандатных ПРД.

Кроме того, отдельно сформулированы требования к управлению доступом к устройствам. Эти требования рассмотрены ниже.

Защита ввода и вывода на отчуждаемый физический носитель информации

» Система защиты должна различать каждое устройство ввода-вывода и каждый канал связи как произвольно используемые или идентифицированные («помеченные»). При вводе с «помеченного» устройства (выводе на «помеченное» устройство) система защиты должна обеспечивать соответствие между меткой вводимого (выводимого) объекта (классификационным уровнем) и меткой устройства. Такое же соответствие должно обеспечиваться при работе с «помеченным» каналом связи.

» Изменения в назначении и разметке устройств и каналов должны осуществляться только под контролем системы защиты.

Сопоставление пользователя с устройством

* Система защиты должна обеспечивать вывод информации на запрошенное пользователем устройство как для произвольно используемых устройств, так для идентифицированных (при совпадении маркировки).

• Система защиты должна включать в себя механизм, посредством которого санкционированный пользователь надежно сопоставляется с выделенным ему конкретным устройством.

11.2. Определение и классификация задач, решаемых механизмами управления доступом к ресурсам

11.2.1. Понятия «владелец» и «собственник» информации

Принципиальным моментом при исследовании проблем управления доступом к ресурсам является трактовка понятия «владельца» информации (в терминологии, применяемой при реализации механизмов управления доступом в ОС) или соответственно «собственника» информации (как увидим далее, это не одно и то же). Сформулируем эти понятия, исходя из приведенных выше формализованных требований к механизмам защиты [1, 2], а также из принципов реализации встроенной защиты в современных универсальных ОС. При этом будем учитывать, что «владелец» файлового объекта во встроенных в ОС механизмах защиты может устанавливать и изменять атрибуты доступа, т.е. назначать и распространять ПРД.

На сегодняшний -день в качестве «владельца» информации рассматривается либо пользователь, либо некое ответственное лицо. В качестве последнего как правило выступает сотрудник подразделения безопасности, в частности, администратор безопасности. Здесь же отметим, что в су-

ществующих ОС пользователь сам может устанавливать атрибуты на создаваемые ими файловые объекты и не во всех случаях данные действия пользователя могут осуществляться в рамках задаваемых администратором разграничений прав доступа к ресурсам.

Таким образом, в рамках существующих ОС «владелец» объекта файловой системы -- это лицо, которое может устанавливать права доступа (атрибуты) к данному файловому объекту. В общем случае это может быть либо администратор, либо пользователь, создающий файловый объект.

Однако права «владельца» в конечном счете определяются тем, кто является собственником информации, т.к. именно собственник информации может принимать решение о ее передаче другим лицам. Естественно, что когда речь идет о домашнем компьютере, то собственником и «владельцем» является непосредственный хозяин компьютера, обрабатывающий на нем собственную информацию.

Другое дело -- применение вычислительных средств (соответственно средств защиты) на предприятии. Использование защищаемого компьютера на предприятии, как правило, связано с защитой служебной информации, конфиденциальных данных и т.д. Но эта информация уже не является собственностью пользователя, следовательно, не пользователь должен являться ее конечным «владельцем». При этом также необходимо учитывать, что, по статистике, большинство хищений информации на предприятии (умышленно или нет) осуществляется непосредственно сотрудниками, в частности, пользователями защищаемых компьютеров. Естественно, то же (но в большей мере) относится к защите секретной информации, собственником которой является государство.

Таким образом, по мнению автора, следует говорить о подходах к защите информации, «владельцем» которой априори не является пользователь. В частности, владельцем служебной информации является предприятие. Что касается конфиденциальной информации, то здесь все зависит от ее типа. В связи с этим большинство приложений систем защиты связано именно с защитой данных, собственником которых пользователь не является.

Следует отметить, что на практике существует некоторое противоречие в определении дискреционного управления доступом и требований, формулируемых к его реализации. Так, определение дискреционного управления доступом предполагает: «Разграничение доступа между поименованными субъектами и поименованными объектами. Субъект с определенным правом доступа может передать это право любому другому субъекту». Другими словами, «владельцем» файлового объекта здесь непосредственно пользователь. Именно данная концепция принимается в качестве основы создания разграничительной политики доступа к ресурсам в современных ОС, прежде всего, в ОС семейства Windows (само собой подразумевается, что говоря о разграничении к файловым объектам для ОС Windows, мы подразумеваем файловую систему NTFS).

Вместе с тем, возвращаясь к формализованным требованиям к дискреционному управлению (т.е. требований к системе защиты, предназначенной для обработки конфиденциальной информации) среди этих требований можем видеть:

* контроль доступа должен быть применим к каждому объекту и каждому субъекту (индивиду или группе равноправных индивидов);
* механизм, реализующий дискреционный принцип контроля доступа, должен предусматривать возможности санкционированного изменения правил или прав разграничения доступа (ПРД), в том числе возможность санкционированного изменения списка пользователей СВТ и списка защищаемых объектов;
* право изменять ПРД должно предоставляться выделенным субъектам (администрации, службе безопасности и т.д.);
* должны быть предусмотрены средства управления, ограничивающие распространения прав на доступ.

Другими словами, рассмотренными требованиями к дискреционному управлению доступом регламентируется, что все права по назначению и изменению ПРД предоставляются не субъектам (пользователям), а выделенному субъекту — администратору безопасности. При этом должны быть предусмотрены средства, ограничивающие распространение прав на доступ, т.е. препятствующие передаче прав одним субъектом другому. Из данных требований вытекает, что в качестве «владельца» ресурса должен рассматриваться администратор безопасности, т.к. «владелец» имеет возможность назначить, изменить и распространить права на доступ.

Видим, что здесь вступают в противоречие концептуальные подходы к определению «владельца» информации, а как следствие и подходы к построению разграничительной политики доступа к ресурсам. В одном случае, если «владельцем» информации считать пользователя, то естественным будет предоставить ему возможность передачи прав доступа к своим ресурсам («владельцем» которых он является) другим пользователям. В другом случае, если «владельцем» информации пользователь не является (что чаще всего), то все права по назначению и распространению (переназначению) прав доступа должны принадлежать вьщеленным субъектам — ответственным лицам «владельца» информации, например, администраторам безопасности. На наш взгляд, в общем случае правомерен именно второй подход. Поэтому, говоря далее о дискреционном управлении доступом, будем предполагать, что назначение и изменение ПРД в системе предоставляется вьщеленным субъектам — администраторам безопасности. При этом модели дискреционного доступа будем рассматривать именно в данных предположениях. Соответсвенно они будут отличаться от моделей, предусматривающих, что «владельцем» файлового объекта является пользователь.

С учетом сказанного можем сделать вывод о том, что в основу разграничительной политики доступа должен быть положен следующий тезис: **Пользо-**

ватель не является собственником обрабатываемой им информации, как следствие, не может рассматриваться ее «владельцем», т.е. не должен иметь прав назначать и изменять ПРД к объектам файловой системы. «Владельцем» объектов файловой системы должен рассматриваться администратор безопасности, являющийся ответственным лицом собственника, в частности, предприятия.

11.2.2. Корректность и полнота реализации разграничительной политики доступа

Последующие исследования будем проводить с учетом того, что пользователь не является «владельцем» информации, им обрабатываемой. Для механизмов управления доступом к ресурсам данный подход связан с понятиями корректности и полноты реализации разграничительной политики доступа к ресурсам.

Под корректностью реализации разграничительной политики доступа к ресурсу будем понимать свойство механизма управления доступом полностью разделять ресурс между пользователями системы. При этом разграничение доступа должно быть реализовано таким образом, чтобы различные пользователи имели доступ к непересекающимся элементам разделяемого ресурса, т.е. чтобы обеспечивалась невозможность несанкционированного обмена ими информацией между собой по средством данного ресурса.

Под полнотой реализации разграничительной политики доступа к ресурсам будем понимать свойство системы защиты обеспечивать корректную реализацию разграничительной политики доступа ко всем ресурсам системы, посредством которых возможен несанкционированный обмен информацией пользователей между собой.

Данные определения вытекают из рассмотренных выше формализованных требований к дискреционному управлению доступом к ресурсам.

Системой защиты, используемой в автоматизированных системах обработки информации, владельцем которой не является пользователь (информация не является собственностью пользователя), должно выполняться требование к полноте реализации разграничительной политики доступа к ресурсам. Причем это требование должно выполняться по отношению к каждому защищаемому ресурсу, что в совокупности обеспечивает невозможность несанкционированного обмена информацией субъектов доступа в системе между собой.

Очевидно, что требование к полноте разграничений доступа к ресурсам может выполняться только в том случае, когда доверенным лицом владельца информации (предприятия, либо государства) выступает не пользователь, а некий субъект, внешний по отношению к информации, обрабатываемой на защищаемом компьютере. Этот субъект наделяется специальными полномочиями и обладает необходимым элементом доверия со стороны владельца информации (в частности, предприятия). Таким субъектом и является администратор безопасности.

11.2.3. Классификация субъектов и объектов доступа

Общая классификация субъектов и объектов доступа

Немаловажным является вопрос классификации субъектов и объектов доступа. Именно на основе этой классификации определяются задачи, которые должны решаться механизмами управления доступом. При этом разграничивается доступ каждого субъекта к каждому объекту.

Прежде всего, введем общую классификацию субъектов и объектов доступа, которые потенциально могут присутствовать в защищаемой системе. Общая классификация субъектов доступа приведена на рис. 11.1, а общая классификация объектов доступа — на рис. 11.2.

Горизонтальная связь на рис. 11.1 отражает, что субъекты «ПОЛЬЗОВАТЕЛЬ» и «ПРОЦЕСС» не могут рассматриваться независимо, т.к. запрос доступа к ресурсу генерирует процесс, запускаемый пользователем.

Рис. 11.1. *Общая классификация субъектов доступа*

Рис. 11.2. *Общая классификация объектов доступа*

Очевидно, что представленные на рис. 11.1 и 11.2 классификации содержат всю возможную совокупность субъектов и объектов доступа защищаемого компьютера (реестр ОС относим к объектам файловой системы).

Теперь введем детальные классификации возможных в системе субъектов и объектов доступа, доступ которых (к которым) должен разграничивать диспетчер доступа. Здесь же определим механизмы управления доступом, решающие задачи разграничения прав доступа.

Детальная классификация субъектов доступа

Как отмечалось, субъектами доступа к ресурсам компьютера являются пользователи и процессы. При этом каждый из этих субъектов, в свою очередь, может быть классифицирован. Классификация пользователей представлена на рис. 11.3. Классификация процессов представлена на рис. 11.4.

Детальная классификация объектов доступа

Так же как и субъекты доступа, в рамках представленной общей классификации могут быть более детально классифицированы и объекты доступа.

Классификация файловых объектов данных, с учетом разделяемых в ЛВС представлена на рис. 11.5. Классификация файловых объектов программ представлена на рис. 11.6. Классификация устройств, с учетом разделяемых в ЛВС представлена на рис. 11.7.

Классификация каналов связи (виртуальные) ЛВС, в предположении, что защищаемый объект находится в составе ЛВС, представлена на рис. 1.8.

Рис. 11.3. *Классификация пользователей*

Рис. 11.4. *Классификация процессов*

Рис. 11.5. *Классификация файловых объектов данных*

Рис. 11.6. *Классификация файловых объектов программ*

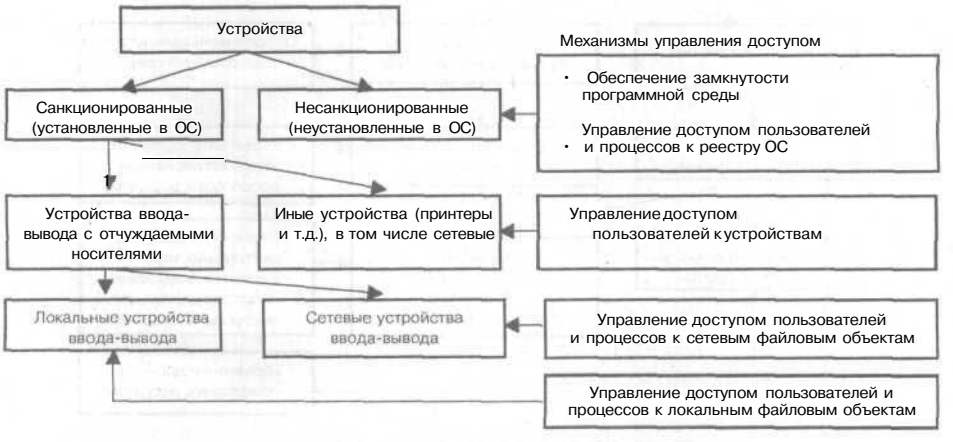


Рис. 11.7. *Классификация устройств*

Рис. 11.8. *Классификация каналов связи (виртуальные) ЛВС*

Требования к механизмам управления доступом

С учетом сказанного можем сделать вывод о необходимости управления доступом для каждой пары «субъект — объект». Без этого не может быть обеспечена полнота разграничительной политики доступа к ресурсам защищаемого объекта. При этом должна быть реализована следующая совокупностьмеханизмов:

1. Механизм управления доступом пользователей (прикладных пользователей и администратора) к ресурсам.
2. Механизм управления доступом процессов (прикладных и системных) к ресурсам.
3. Механизм комбинированного доступа пользователя и процесса к ресурсам.

В качестве ресурсов, к которым должен разграничиваться доступ, должны выступать:

1. При автономном (не в составе сети) функционировании защищаемого объекта:
   - логические диски (тома), каталоги, файлы данных;
   - каталоги, не разделяемые ОС и приложениями (например, TEMP, «Корзина» и др.);
   - каталоги с исполняемыми файлами, исполняемые файлы (обеспечение замкнутости программной среды);
   - системный диск (где располагаются каталоги и файлы ОС);
   - объекты, хранящие настройки ОС, приложений, системы защиты (для ОС Windows — реестр ОС);
   - устройства;
   - отчуждаемые внешние накопители (дискеты, CD-ROM диски и т.д.).

2. При функционировании защищаемого объекта в составе ЛВС помимо вышеуказанных должны дополнительно рассматриваться следующие ресурсы:
   - разделяемые в сети файловые объекты;
   - разделяемые в сети устройства;
   - хосты;
   - сетевые информационные технологии (сетевые приложения и службы).

С учетом формализованных требований к системе защиты при реализации системы защиты конфиденциальной информации основу данных механизмов должен составлять дискреционный принцип разграничения прав доступа. При реализации системы защиты секретной информации основу данных механизмов должен составлять мандатный принцип разграничения прав доступа, а дискреционный принцип в этом случае реализуется в качестве дополнения к мандатному.

11.3. Угрозы преодоления разграничительной политики доступа к ресурсам. Противодействие угрозам современными ОС

11.3.1. Классификация угроз преодоления разграничительной политики доступа к ресурсам

Классификация угроз преодоления разграничительной политики доступа к ресурсам в общем случае представлена на рис. 11.9. Как и ранее, здесь могут быть выделены явные и скрытые угрозы. Явные угрозы связаны с некорректностью реализации как собственно механизма защиты, так и механизма его администрирования. Кроме того, явные угрозы могут быть связаны с неполнотой разграничений, реализуемых в системе, что позволяет пользователю без применения каких-либо вспомогательных средств осуществить НСД к ресурсам.

Скрытые же угрозы наоборот предполагают для осуществления НСД применение пользователем своего программного обеспечения (и здесь невозможно предположить, каким образом и с какой целью они реализуются), а также знаний об ошибках и потенциально возможных закладках в реализации механизмов управления доступом к ресурсам.

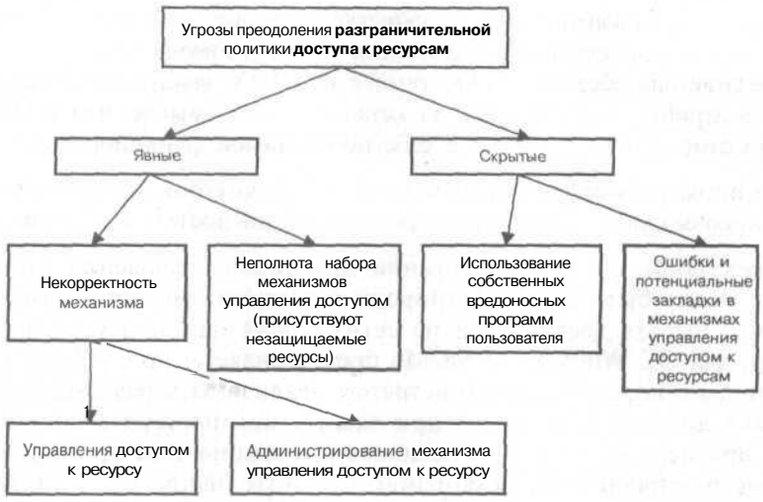


Рис. 11.9. Угрозы преодоления разграничительной политики доступа к ресурсам

11.3.2. Практический анализ современных ОС в контексте принятой классификации угроз преодоления разграничительной политики доступа

Проиллюстрируем отмеченные классы угроз простыми примерами и проанализируем возможность противодействия им современными универсальными ОС.

Под некорректностью механизма управления доступом к ресурсам (в отличие от ошибки в реализации механизма) будем понимать возможность появления (или существование) канала НСД к информации, обусловленного в идеологически неверном определении его задач и функций, что критически сказывается на реализации механизма. Например, ранее отмечалось, что важнейшим механизмом защиты является обеспечение замкнутости программной среды. При этом должны рассматриваться возможности запуска программ из всех объектов доступа, а не только с жесткого диска, в частности, с локальных и сетевых устройств ввода, общих папок (разделяемых ресурсов файловой системы).

Разграничение доступа на запуск программ только из объектов файловой системы, располагаемых на жестком диске, является некорректным решением задачи обеспечения замкнутости программной среды. Подобный недостаток, например, присущ ОС Windows NT/2000, где невозможно установить атрибут доступа «на исполнение» к устройствам ввода. Таким образом, важнейший механизм защиты — механизм обеспечения замкнутости программ-

ной среды, противодействующий скрытым угрозам, — здесь не может быть реализован без физического отключения устройств ввода в принципе. Гораздо хуже ситуация обстоит для ОС семейства UNIX, ввиду невозможности установить атрибут «исполнение» на каталог (в том смысле, что в UNIX для каталога этот атрибут не несет в себе необходимой функциональности).

Другой примернекорректности механизма замкнутости программной среды — невозможность разграничить исходящий доступ к общим папкам.

Некорректность администрирования механизма управления доступом к ресурсу может быть проиллюстрирована следующими примерами. Право разделять (делать доступными из сети) общие папки и устройства, например, для ОС Windows 95/98/Me предоставляется пользователю, что в принципе не позволяет администратору реализовать разграничительную политику доступа к ресурсам при защите компьютера в составе ЛВС. Далее, при использовании на защищаемом компьютере приложения, обладающего встроенными механизмами защиты (например, СУБД), появляется несколько уровней администрирования безопасностью, при этом не решается задача централизации подобной системы управления.

Неполнота набора механизмов управления доступом делает современные ОС уязвимыми. В частности, без разграничений доступа для субъекта «процесс» трудно обеспечить какую-либо серьезную защиту информации. Так, для ОС семейства Windows существуют системные процессы, запускаемые под текущим пользователем. При этом невозможно разграничить доступ для системного процесса.

Относительно скрытых угроз можем отметить, что они, в первую очередь, связаны с некорректностью реализации механизма обеспечения замкнутости программной среды, который, как отмечалось, большинством современных универсальных ОС реализован не в полном объеме.

11.4. Структура диспетчера доступа. Требования к механизмам управления доступом к ресурсам

11.4.1. Диспетчер доступа, как центральный элемент системы защиты

Разграничение доступа осуществляется центральным элементом системы защиты -- диспетчером доступа. Диспетчер доступа идентифицирует субъекты, объекты и параметры запрашиваемого доступа субъектом к объекту. Это именно диспетчер доступа предоставляет запрашиваемый доступ или запрещает его.

Идентифицирующую информацию субъектов и объектов доступа принято называть **учетной информацией.** Правила, на основании которых диспетчер доступа принимает решение о предоставлении (либо отказе) субъекту доступа к объекту, называют правилами (или политикой) разграничения доступа.

Обобщенная схема управления доступом представлена на рис. 11.10.

Рис. 11.10. *Обобщенная схема управления доступом*

11.4.2. Требования, предъявляемые к диспетчеру доступа

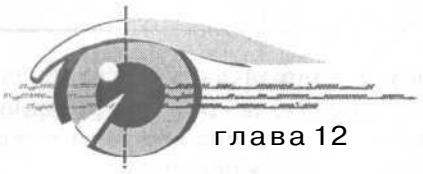
Следуя формализованным требованиям к механизмам управления доступом к ресурсам, могут быть сформулированы требования к диспетчеру доступа, которые состоят в следующем:

- диспетчер доступа должен обеспечивать корректность и однозначность реализации разграничительной политики доступа к ресурсам. Для каждой пары «субъект»—«объект» в диспетчере доступа должно быть задано явное и недвусмысленное перечисление допустимых типов доступа (читать, писать и т. д.), т.е. тех типов доступа, которые являются санкционированными для данного субъекта к данному ресурсу (объекту);
- диспетчер доступа должен выполнять требования к полноте реализации разграничительной политики доступа, то есть управление доступом должно быть применимо к каждому объекту и каждому субъекту;
- диспетчер доступа должен обнаруживать как явные, так и скрытые действия субъекта при доступе к объекту, т.е. противодействовать как явным, так и скрытым угрозам НСД к ресурсам. Под «явными» подразумеваются действия, осуществляемые с использованием санкционированных системных средств, а под «скрытыми» — иные действия, в том числе с использованием собственных программ субъекта;

* диспетчер доступа должен обеспечивать реализацию централизованной схемы администрирования (задания и изменения разграничительной политики доступа к ресурсам). Кроме того, он должен предусматривать возможность санкционированного изменения учетной информации субъектов и объектов доступа, а также и правил разграничения доступа. В том числе, должна быть предусмотрена возможность санкционированного изменения списка субъектов доступа- и списка защищаемых объектов. Право изменять учетную информацию субъектов и объектов доступа и правил разграничения доступа должно предоставляться только выделенным субъектам (администрации, службе безопасности и т.д.);
« диспетчер доступа должен содержать в своем составе средства управления, ограничивающие распространение прав на доступ.

При практической реализации схемы управления доступом важнейшими вопросами, решению которых должно быть уделено особое внимание, являются:

1. Обоснование корректности реализуемого диспетчером доступа механизма управления доступом и полноты механизмов, реализуемых диспетчером доступа. Без возможности обоснования корректности механизма управления доступом к ресурсам невозможно говорить об эффективности системы защиты.
2. Разработка подходов к противодействию «скрытым» каналам несанкционированного доступа к ресурсам (в обход диспетчера доступа, при условии корректной реализации им механизма управления доступом к ресурсам). Особенно эта задача актуальна при реализации добавочного средства защиты, используемого для ОС и приложений, в которых могут присутствовать ошибки в реализации, а также системные условия, требующие решения дополнительных задач для корректного разграничения доступа к ресурсам.



глава 12

Модели управления доступом

В данном разделе введем и теоретически обоснуем основополагающие принципы решения задачи управления доступом к ресурсам, в предположении, что пользователь не является «владельцем» информации. Соответственно, должно выполняться требование к корректности реализации разграничительной политики доступа к ресурсу.

В данных предположениях получим и исследуем модели дискреционного и мандатного управления доступом к ресурсам. Рассмотрим, в чем состоит их принципиальное различие, сформулируем требования к соответствующим механизмам защиты и подходы к их реализации. Рассмотрим возможности реализации разрабатываемых моделей встроенными в ОС механизмами защиты.

Сразу отметим, что общим для всех моделей являет то, что администратор безопасности является «владельцем» любого объекта файловой системы. То есть только он обладает правом назначить (изменить) атрибуты доступа. Поэтому механизмы изменения прав доступа в моделях не рассматриваются.

12.1. Каноническая модель управления доступом. Условие корректности механизма управления доступом

Введем следующие обозначения. Пусть множества $C = \{C1,...,Ck\}$ и $O = \{O1,...,Ok\}$ — соответственно линейно упорядоченные множества субъектов и объектов доступа. В качестве субъекта доступа Ci, $i = 1,...,k$ рассматривается как отдельный субъект, так и группа субъектов, обладающих одинаковыми правами доступа. Соответственно, в качестве объекта доступа Oi, $i = 1,...,k$ может также рассматриваться как отдельный объект, так и группа объектов, характеризуемых одинаковыми правами доступа к ним.

Пусть S = {0, 1} — множество прав доступа, где «О» обозначает запрещение доступа субъекта к объекту, а «1» -- разрешение полного доступа. Тогда каноническую модель управления доступом можно представить матрицей доступа D, имеющей следующий вид:

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \end{array}$$

Под канонической моделью управления доступом для линейно упорядоченных множеств субъектов (групп субъектов) и объектов (групп объектов) доступа понимается модель, описываемая матрицей доступа, элементы главной диагонали которой «1» разрешают полный доступ субъектов к объектам, остальные элементы «О» запрещают доступ субъектов к объектам.

Говоря о доступе, нами в этот атрибут не включается право назначения (изменения) «владельца» и право назначения (изменения) атрибутов доступа к объекту. Как отмечали ранее, данные права выведены из схемы рассмотрения, поскольку они принадлежат администратору безопасности, который является «владельцем» любого объекта файловой системы.

Данная модель управления доступом формально может быть описана следующим образом:

- Dij = 1, если i = j,
- иначе Dij = 0.

Утверждение

Диспетчер доступа реализует механизм управления доступом корректно только в том случае, если его настройками (заданием учетных записей субъектов и объектов доступа и правил разграничения доступа) можно реализовать каноническую модель управления доступом.

Доказывается утверждение от обратного. Если каноническую модель управления доступом реализовать невозможно (т.е. присутствуют элементы «1» вне главной диагонали матрицы доступа), то в системе присутствует, по крайней мере, один объект, доступ к которому невозможно разграничить в полном объеме. При этом объект включается одновременно в несколько групп объектов, априори характеризуемых различными правами доступа к ним.

Следствие 1.

Любой механизм управления доступом должен позволять настройками диспетчера доступа сводить реализуемую им модель доступа к каноническому виду.

Объекты доступа могут по своей сути существенно различаться: файловые объекты, ветви и ключи реестра ОС (для ОС Windows), принтеры, разделяемые сетевые ресурсы, устройства, ресурсы внешней сети (хосты) и т.д. К ним могут различаться типы доступа (например, файловые объекты и принтеры). Кроме того, на практике может быть ограничение на число ресурсов (например, принтеров в системе, к которым разграничивается доступ, может быть существенно меньше, чем субъектов доступа к ним).

Однако все это не противоречит общности сформулированного утверждения, требования которого должны выполняться механизмом управления доступом при соответствующих настройках системы и диспетчера доступа к объекту любого вида. Например, диспетчер доступа к принтерам должен при включении в систему принтеров по числу субъектов доступа (в частности, пользователей) обеспечивать реализацию канонической модели управления доступом, где элементами матрицы доступа D будут «1» — доступ субъекту к принтеру разрешен, «О» — доступ субъекту к принтеру запрещен.

Следствие 2.

К каждому защищаемому ресурсу системы, требующему разграничения доступа, должен быть реализован диспетчер доступа, позволяющий соответствующими настройками сводить реализуемую им модель доступа к каноническому виду. Таким образом, механизм управления доступом к ресурсу реализован корректно только в том случае, если настройками диспетчера доступа реализуемая им модель доступа может быть приведена к каноническому виду.

12.2. Понятие и классификация каналов взаимодействия субъектов доступа

Рассмотренная выше каноническая модель управления доступом характеризуется полным разграничением доступа субъектов к объектам, при котором субъекты не имеют каналов взаимодействия — каналов обмена информацией. Однако такая возможность должна предусматриваться. Если в системе может находиться несколько субъектов (например, пользователей), то появляется задача предоставления субъектам возможности обмена информацией.

Введем несколько определений:

- **Под каналом взаимодействия субъектов доступа** понимается реализуемая диспетчером доступа возможность обмена субъектами доступа информацией в рамках канонической модели управления доступом.
- **Под симплексным каналом взаимодействия субъектов доступа** понимается канал, обеспечивающий возможность одностороннего обмена субъектами доступа информацией. Будем его обозначать Ci → Cj (информация может передаваться в одну сторону — от субъекта Ci к субъекту Cj).

» **Под дуплексным каналом взаимодействия субъектов доступа** понимается канал, обеспечивающий возможность двухстороннего обмена субъектами доступа информацией. Будем его обозначать Ci ↔ Cj.

Дуплексный канал взаимодействия субъектов доступа представляет собою два встречно-направленных симплексных канала.

» **Под активным симплексным каналом взаимодействия субъектов доступа** Ci a Cj понимается канал взаимодействия, в котором активным является отправитель информации (отправитель, в данном случае субъект Ci выдает информацию получателю).

Активный симплексный канал взаимодействия субъектов доступа может быть реализован с использованием операций «запись», «модификация» или «добавление». Операция «модификация» отличается от операции «запись» тем, что не позволяет читать объект перед занесением в него информации.

Операция «добавление» отличается от операций «запись» и «модификация» тем, что ее выполнение не позволяет ни читать, ни модифицировать информацию, располагаемую в объекте, в который добавляется информация по каналу взаимодействия - - данная операция позволяет лишь добавить информацию, предотвращая возможность изменить существующую в объекте информацию.

* **Под пассивным симплексным каналом взаимодействия субъектов доступа** Ci a Cj понимается канал взаимодействия, в котором активным является получатель информации (получатель, в данном случае субъект Cj) забирает информацию у отправителя. Пассивный симплексный канал взаимодействия субъектов доступа реализуется с использованием операции «чтение».

Отметим, что в задачу управления доступом в общем случае входит не только защита данных субъектов (в данном случае пользователей) от несанкционированного их прочтения другими субъектами, но и защита данных субъектов от возможности их искажения другими субъектами. Этим, кстати говоря, защита данных средствами защиты от НСД принципиально отличается от защиты данных (управления доступом) с использованием средств криптографической защиты.

Для иллюстрации сказанного, рассмотрим каноническую матрицу доступа D, реализуемую с использованием средств криптографической защиты.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Зп & \dots & Зп & Зп \\ Зп & Зп/Чт & \dots & Зп & Зп \\ \dots & \dots & \dots & \dots & \dots \\ Зп & Зп & \dots & Зп/Чт & Зп \\ Зп & Зп & \dots & Зп & Зп/Чт \end{bmatrix} \end{array}$$

где элемент «Зп» обозначает разрешение доступа с использованием операции «запись» (прочитать информацию пользователь может, но не имеет возможности ее преобразовать к читаемому виду, т.к. информация зашифрована, поэтому обозначаем подобное право доступа, как «Зп»), «Зп/Чт» — соответственно, операции «запись» и «чтение».

С учетом введенных выше понятий и определений данную матрицу доступа можно охарактеризовать, как каноническую матрицу доступа, расширенную дуплексными каналами взаимодействия между собою всех субъектов. При этом дуплексные каналы реализованы на основе активных симплексных каналов взаимодействия субъектов доступа с использованием операции «запись». Другими словами, это частный случай управления доступом, не обеспечивающий защиту данных от их несанкционированной модификации (удаления).

12.3. Модель управления доступом с взаимодействием субъектов доступа посредством выделенного канала

Особенностью данной модели является включение в систему дополнительного объекта доступа (группы объектов), используемого субъектами доступа в качестве выделенного канала взаимодействия. Отметим, что в качестве канала взаимодействия здесь должны использоваться дуплексные каналы. Причем в обе стороны взаимодействия должны быть реализованы как активный, так и пассивный симплексные каналы.

Ниже представлена каноническая матрица доступа D с выделенным каналом взаимодействия субъектов доступа. При этом полный доступ очевидно задается операциями «запись» и «чтение». Для организации выделенного канала взаимодействия в систему включен объект доступа Ok+1.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \\ O_{k+1} \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & 0 & \dots & 0 & 0 \\ 0 & Зп/Чт & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & Зп/Чт & 0 \\ 0 & 0 & \dots & 0 & Зп/Чт \\ Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт \end{bmatrix} \end{array}$$

Недостатком данной модели, ограничивающим возможность ее практического использования, является доступность находящейся в канале информации одновременно всем субъектам доступа. Данный недостаток может преодолеваться созданием группы каналов взаимодействия (включением группы дополнительных объектов доступа, в пределе, свой объект доступа для каждого канала взаимодействия субъектов доступа) с соответствующим разграничением к ним доступа субъектов. Однако это приводит к неэффективному использованию ресурсов защищаемого объекта.

12.4. Модели управления доступом с взаимодействием субъектов доступа посредством виртуальных каналов

Под **виртуальным каналом взаимодействия субъектов доступа** будем понимать канал взаимодействия, реализованный с использованием только существующих объектов доступа без включения в систему дополнительных объектов.

Естественно, что в соответствии с классификацией (см. п. 12.2) виртуальные каналы для рассматриваемых моделей должны быть дуплексными. При этом они должны строиться на основе реализации либо пассивных симплексных каналов, либо активных симплексных каналов, реализованных с использованием операции «добавления». Использование операции «запись» здесь недопустимо ввиду необходимости защиты информации субъектов от возможности ее модификации.

Модель управления доступом с дуплексными виртуальными каналами взаимодействия на основе пассивных симплексных каналов

Рассмотрим матрицу доступа D для модели управления доступом с дуплексными виртуальными каналами взаимодействия на основе пассивных симплексных каналов.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Чт & \dots & Чт & Чт \\ Чт & Зп/Чт & \dots & Чт & Чт \\ \dots & \dots & \dots & \dots & \dots \\ Чт & Чт & \dots & Зп/Чт & Чт \\ Чт & Чт & \dots & Чт & Зп/Чт \end{bmatrix} \end{array}$$

Модель управления доступом формально может быть описана следующим образом:

- Dij = Зп/Чт, если i = j,
- * иначе Dij = Чт.

К недостаткам данной модели можно отнести то, что она предотвращает лишь возможность несанкционированной модификации информации объектов, при этом не разграничивая субъектам доступ к объектам «по чтению». Очевидно, что в таком виде виртуальный канал взаимодействия субъектов не применим. Он может использоваться лишь при введении дополнительных разграничений или условий для канала взаимодействия субъектов доступа.

Модель управления доступом с дуплексными виртуальными каналами взаимодействия на основе активных симплексных каналов

При организации канала взаимодействия субъектов доступа целесообразно рассматривать множество прав доступа {Чт, Д}, используемое для реализации канала взаимодействия субъектов доступа. Здесь элемент «Д» обозначает добавление, то есть возможность пользователя добавить информацию в объект без возможности чтения объекта и без возможности модификации существующей в объекте информации. Право доступа «Зп/Чт» обозначает право пользователя на чтение и запись информации.

Рассмотрим матрицу доступа D для модели управления доступом с дуплексными виртуальными каналами взаимодействия на основе активных симплексных каналов.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Д & & Д & Д \\ Д & Зп/Чт & & Д & Д \\ \dots & \dots & \dots & \dots & \dots \\ Д & Д & \dots & Зп/Чт & Д \\ Д & Д & & Д & Зп/Чт \end{bmatrix} \end{array}$$

Модель управления доступом формально может быть описана следующим образом:

- Dij = Зп/Чт, если i = j,
- иначе Dij = Д.

Данная модель практически лишена недостатков, присущих ранее рассмотренным моделям. Здесь в полном объеме реализуется каноническая модель управления доступом. При этом обеспечивается возможность полноценного корректного взаимодействия субъектов доступа: каждый субъект доступа может взаимодействовать со всеми другими субъектами доступа системы без снижения уровня защищенности от НСД.

Будем считать данную модель, наиболее приемлемой для использования в рассматриваемых приложениях. Именно эту модель будем считать канонической моделью управления доступом с взаимодействием субъектов. Далее дадим ее определение.

Определение.

Под **канонической моделью управления доступом с взаимодействием субъектов (групп субъектов) и объектов (групп объектов) доступа** понимается модель, описываемая матрицей доступа, элементы главной диагонали которой «Зп/Чт» задают полный доступ субъектов к объектам, остальные элементы «Д» задают активные симплексные каналы взаимодействия с использованием операции «добавление».

12.5. Различия и общность альтернативных моделей

Выше были рассмотрены различные модели управления доступом. С учетом сказанного, можем сделать следующие выводы относительно различия и общности альтернативных моделей:

1. Модели управления доступом различаются реализуемым в них каналом (каналами) взаимодействия субъектов доступа. При этом канал взаимодействия может быть выделенным, либо виртуальным; пассивным, либо активным; симплексным, либо дуплексным.

2. Общим для рассматриваемых моделей управления доступом является то, что для корректной реализации канала взаимодействия субъектов доступа следует рассматривать не право доступа «Зп» — «запись», а право доступа «Д» -- «добавление». Право «Д» предоставляет возможность пользователю добавить информацию в объект без возможности чтения объекта и без возможности модификации существующей в объекте информации при добавлении новой информации.

12.6. Методы управления виртуальными каналами взаимодействия и соответсвующие канонические модели управления доступом

Методы произвольного и принудительного управления виртуальными каналами взаимодействия субъектов доступа

Выше рассматривались способы организации виртуального канала взаимодействия субъектов доступа. Рассмотрим методы управления ими. Методы управления виртуальными каналами взаимодействия субъектов доступа могут быть классифицированы, как методы произвольного и принудительного управления. Соответственно метод, комбинирующий данные подходы -- метод произвольно-принудительного управления.

Под методом произвольного управления виртуальными каналами взаимодействия субъектов доступа понимается управление по усмотрению субъекта (владельца информации). При этом решение о предоставлении другому субъекту информации из объекта по виртуальному каналу принимается непосредственно субъектом, имеющим полный доступ к этой информации.

Под методом принудительного управления виртуальными каналами взаимодействия субъектов доступа понимается управление, реализуемое не по усмотрению субъекта, а на основании некоторой параметрической шкалы оценки субъектов и объектов доступа.

Метод произвольно-принудительного (комбинированного) управления виртуальными каналами взаимодействия субъектов доступа включает в себя элементы произвольного и принудительного управления доступом.

Таким образом, произвольное управление доступом реализуется в том случае, когда не может (либо не имеет смысла) быть введена какая-либо шкала оценки субъектов и объектов доступа. Как следствие, каналы взаимодействия субъектов идентичны для всех субъектов доступа. Естественно, что в данных предположениях нецелесообразна какая-либо схема принудительного управления.

Каноническая модель на основе произвольного управления виртуальными каналами взаимодействия субъектов

В соответствии с введенным ранее определением, каноническая модель управления доступом на основе произвольного управления виртуальными каналами определяется канонической моделью управления доступом с взаимодействием субъектов для линейно упорядоченных множеств субъектов (групп субъектов) и множеств объектов (групп объектов). Описывается эта модель матрицей доступа D, имеющей следующий вид.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{cccccc} C & C & & C_{k-1} & C \end{array} \\ \begin{bmatrix} Зп/Чт & Д & \dots & Д & Д \\ Д & Зп/Чт & \dots & Д & Д \\ \dots & \dots & \dots & \dots & \dots \\ Д & Д & \dots & Зп/Чт & Д \\ Д & Д & & Д & Зп/Чт \end{bmatrix} \end{array}$$

Модель управления доступом формально может быть описана следующим образом:

- элемент (Dij) матрицы Dij = Зп/Чт, если i = j,
- иначе Dij = Д.

В данной модели каждому субъекту (группе субъектов) доступа предоставляется активный симплексный канал взаимодействия со всеми остальными субъектами доступа (с субъектами других групп) с использованием операции «добавление».

Под канонической моделью управления доступом на основе метода произвольного управления виртуальными каналами взаимодействия субъектов доступа для линейно упорядоченных множеств субъектов (групп субъектов) и объектов (групп объектов) доступа понимается модель, описываемая матрицей доступа, элементы главной диагонали которой «Зп/Чт» задают полный доступ субъектов к объектам, остальные элементы «Д» задают активные симплексные каналы взаимодействия с использованием операции «добавление».

Принудительное управление виртуальными каналами и полномочное управление доступом

Основу принудительного управления виртуальными каналами взаимодействия субъектов доступа составляет введение некоторой параметрической шкалы оценки субъектов и объектов доступа. При принудительном управлении виртуальными каналами реализуется полномочное управление доступом (управление доступом с учетом параметрической шкалы оценки субъектов и объектов доступа).

Введем следующие обозначения. Пусть множества C = {C1,..., Cк} и O = {O1,..., Ok} — соответственно, линейно полномочно упорядоченные множества субъектов и объектов доступа, упорядоченные таким образом, что, чем меньше порядковый номер субъекта доступа, тем он обладает большими полномочиями по доступу к объектам. Соответственно, чем меньше порядковый номер объекта доступа, тем большие полномочия необходимы для доступа к нему.

В качестве субъекта доступа Ci, i = l,...,k рассматривается как отдельный субъект, так и группа субъектов, обладающих одинаковыми правами доступа. Соответственно, в качестве объекта доступа Oi, i = l,...,k может также рассматриваться как отдельный объект, так и группа объектов, характеризуемых одинаковыми к ним правами доступа.

При линейно полномочном упорядочивании множеств субъектов и объектов доступа C = {C1,..., Cк} и O = {O1,..., Ok} i-й субъект может иметь доступ к объектам с порядковым номером не меньше i-го (к объектам Oi,...,Ok). Соответственно, к i-му объекту могут иметь доступ субъекты с порядковым номером не больше i-го (субъекты O1,...,Oi).

Утверждение

При полномочном управлении доступом недопустимым является передача информации (организация виртуального канала взаимодействия субъектов доступа) из объектов с более высокими полномочиями (меньшим порядковым номером) доступа к ним в объекты с меньшими полномочиями (большим порядковым номером) доступа к ним.

Утверждение доказывается от обратного. Если подобное взаимодействие допустимо, то априори отсутствует полномочное упорядочивание объектов, т.е. собственно нивелируется параметрическая шкала оценки субъектов и объектов доступа.

Полномочная модель управления доступом с произвольным управлением виртуальными каналами взаимодействия субъектов доступа

Особенностью данной модели управления доступом является то, что в каноническую модель управления доступом добавляются активные симплексные каналы, обеспечивающие взаимодействия субъектов, име-

ющих более низкие полномочия, с субъектами-обладателями более высоких полномочий.

Матрица доступа D, описывающая данную модель управления доступом, имеет следующий вид.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Д & \dots & Д & Д \\ 0 & Зп/Чт & \dots & Д & Д \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & Зп/Чт & Д \\ 0 & 0 & \dots & О & Зп/Чт \end{bmatrix} \end{array}$$

Модель управления доступом формально может быть описана следующим образом:

» Dij = Зп/Чт, если i = j,
» Dij = 0, если i > j,
« Dij = Д, если i < j,

где i — порядковый номер объекта (номер строки в матрице доступа);
j — порядковый номер субъекта (номер столбца в матрице доступа).

Для данной канонической модели управления доступом характерно то, что для доступа к объекту с полномочиями i, субъект должен иметь полномочия не ниже, чем i, где i = 1,...,k. При этом полномочия линейно упорядочены по возрастанию -- чем меньше номер, тем выше полномочия.

В данной модели применяется метод произвольного управления виртуальными каналами взаимодействия субъектов доступа. В рамках этого метода субъекты с меньшими полномочиями могут «добавлять» информацию в объекты с большими полномочиями.

Таким образом, **под канонической моделью управления доступом на основе метода произвольного управления виртуальными каналами** взаимодействия субъектов доступа для линейно полномочно упорядоченных множеств субъектов (групп субъектов) и объектов (групп объектов) доступа понимается модель, описываемая матрицей доступа, элементы главной диагонали которой «Зп/Чт» задают полный доступ субъектов к объектам равных полномочий, элементы, расположенные выше главной диагонали, а «Д» задают активные симплексные каналы взаимодействия с использованием операции «добавление». Таким образом субъекты с меньшими полномочиями могут «добавлять» информацию в объекты с большими полномочиями.

Полномочная модель управления доступом с принудительным управлением виртуальными каналами взаимодействия субъектов доступа

Особенностью данной модели управления доступом является то, что в каноническую модель управления доступом добавляются пассивные симплексные каналы, обеспечивающие взаимодействия субъектов, имеющих более высокие полномочия, с субъектами-обладателями более низких полномочий.

Матрица доступа D, описывающая данную модель управления доступом, имеет следующий вид.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \bullet\bullet\bullet & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & 0 & \dots & О & О \\ Чт & Зп/Чт & & О & О \\ \dots & \dots & \dots & \dots & \dots \\ Чт & Чт & \dots & Зп/Чт & О \\ Чт & Чт & \dots & Чт & Зп/Чт \end{bmatrix} \end{array}$$

Модель управления доступом формально может быть описана следующим образом:

- Dij = Зп/Чт, если i = j,
- Dij = Чт, если i > j,
- Dij = 0, если i < j,

где i — порядковый номер объекта (номер строки в матрице доступа);
j — порядковый номер субъекта (номер столбца в матрице доступа).

Для данной канонической модели управления доступом характерно то, что для доступа к объекту с полномочиями i, субъект должен иметь полномочия не ниже, чем i, где i = l,...,k. При этом полномочия линейно упорядочены по возрастанию — чем меньше номер, тем выше полномочия. В данной модели применяется метод принудительного управления виртуальными каналами взаимодействия субъектов доступа: субъекты с большими полномочиями имеют принудительное для объектов с меньшими полномочиями право «читать» из них информацию в объекты с большими полномочиями.

Под канонической моделью управления доступом на основе метода принудительного управления виртуальными каналами взаимодействия субъектов доступа для линейно полномочно упорядоченных множеств субъектов (групп субъектов) и объектов (групп объектов) доступа понимается модель, описываемая матрицей доступа, элементы главной диагонали которой «Зп/Чт» задают полный доступ субъектов к объектам равных полномочий, элементы, расположенные ниже главной диагонали, «Чт» задают пассивные симплексные каналы взаимодействия с использованием операции «чтение». Таким образом, субъекты с большими полномочиями имеют принудительное

для объектов с меньшими полномочиями право «читать» из них информацию в объекты с большими полномочиями.

Полномочная модель управления доступом с комбинированным управлением виртуальными каналами взаимодействия субъектов доступа

Особенностью данной модели управления доступом является то, что в каноническую модель управления доступом добавляются дуплексные каналы взаимодействия субъектов доступа:

- активные симплексные каналы, обеспечивающие взаимодействия субъектов, имеющих более низкие полномочия, с субъектами-обладателями более высоких полномочий;
- пассивные симплексные каналы, обеспечивающие взаимодействия субъектов, имеющих более высокие полномочия, ссубъектами-обладателями более низких полномочий.

Матрица доступа D, описывающая данную модель управления доступом, имеет следующий вид:

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Д & & Д & Д \\ Чт & Зп/Чт & \dots & Д & Д \\ \dots & \dots & \dots & \dots & \dots \\ Чт & Чт & \dots & Зп/Чт & Д \\ Чт & Чт & & Чт & Зп/Чт \end{bmatrix} \end{array}.$$

Модель управления доступом формально может быть описана следующим образом:

- Dij = Зп/Чт, если i = j,
- Dij = Чт, если i > j,
- Dij = Д, если i < j,

где i — порядковый номер объекта (номер строки в матрице доступа);
j — порядковый номер субъекта (номер столбца в матрице доступа).

Для данной модели управления доступом характерно то, что для доступа к объекту с полномочиями i, субъект должен иметь полномочия не ниже, чем i, где i = 1,...,k (полномочия линейно упорядочены по возрастанию - чем меньше номер, тем выше полномочия).

Данная модель является комбинированной и в ней применяются оба метода управления виртуальными каналами:

- метод произвольного управления виртуальными каналами взаимодействия субъектов доступа: субъекты с меньшими полномочиями могут «добавлять» информацию в объекты с большими полномочиями;

» метод принудительного управления виртуальными каналами взаимодействия субъектов доступа: субъекты с большими полномочиями имеют принудительное для объектов с меньшими полномочиями право «читать» из них информацию в объекты с большими полномочиями (комбинация методов).

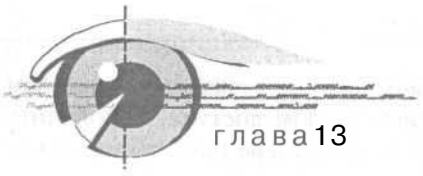
Под канонической моделью управления доступом на основе метода комбинированного управления виртуальными каналами взаимодействия субъектов доступа для линейно полномочно упорядоченных множеств субъектов (групп субъектов) и объектов (групп объектов) доступа понимается модель, описываемая следующей матрицей доступа:

- элементы главной диагонали которой «Зп/Чт», задают полный доступ субъектов к объектам равных полномочий;
* элементы «Д», расположенные выше главной диагонали, задают активные симплексные каналы взаимодействия с использованием операции «добавление» - - субъекты с меньшими полномочиями могут «добавлять» информацию в объекты с большими полномочиями;
» элементы «Чт», расположенные ниже главной диагонали, задают пассивные симплексные каналы взаимодействия с использованием операции «чтение» — субъекты с большими полномочиями имеют принудительное для объектов с меньшими полномочиями право «читать» из них информацию в объекты с большими полномочиями.

12.7. ВЫВОДЫ

Подводя итог всему сказанному, в данном разделе можно сделать следующие выводы:

1. В общем случае различие моделей управления доступом базируется на отличиях способов реализации канала взаимодействия субъектов доступа и методов управления им.
2. Могут быть выделены методы произвольного и принудительного управления виртуальными каналами взаимодействия субъектов доступа. Под методом произвольного управления понимается управление по усмотрению субъекта (владельца информации). При этом решение о предоставлении другому субъекту информации из объекта по виртуальному каналу принимается непосредственно субъектом, имеющим полный доступ к этой информации. Под методом принудительного управления понимается управление, реализуемое не по усмотрению субъекта (владельца информации), а на основании некоторой параметрической шкапы оценки субъектов и объектов доступа (полномочий).
3. Задача управления доступом решена корректно в том случае, если диспетчером доступа реализуется одна из рассмотренных канонических моделей (для полномочных моделей в предположении, что при полномочном управлении корректно задана параметрическая шкала оценки субъектов и объектов доступа).



глава 13

Реализация моделей доступа механизмами добавочной и встроенной защиты

13.1. Механизмы реализации дискреционной и мандатной моделей управления доступом

Итак, выше нами были определены канонические модели управления доступом, канонические в том смысле, что обеспечивают корректное решение задачи управления доступом и имеют общий вид для соответствующих условий решения задачи.

Показано, что отличие моделей управления доступом базируется на отличиях способов реализации канала взаимодействия субъектов доступа и методов управления им.

Теперь рассмотрим возможные способы реализации моделей диспетчером доступа. Для этого определим механизмы, реализуемые диспетчером доступа, а также используемую диспетчером учетную информацию субъектов и объектов доступа.

Очевидно, что **различные механизмы, реализующие одну и ту же модель управления доступом, по свой сути идентичны. Их отличие состоит лишь в способе задания и обработки учетной информации субъектов и объектов доступа. Механизм управления доступом, реализуемый диспетчером доступа, это лишь способ обработки запросов доступа в соответствии с реализуемой моделью управления доступом.**

13.1.1. Механизмы реализации дискреционной модели доступа

На сегодняшний день на практике используются понятия дискреционного и мандатного механизмов управления доступом [1, 2, 4, 8, 13]. Дадим этим механизмам определения с учетом введенных ранее моделей.

Под дискреционным механизмом управления доступом понимается способ обработки запросов диспетчером доступа, основанный на задании правил разграничения доступа в диспетчере непосредственно матрицей доступа D.

Таким образом, дискреционный механизм управления доступом предполагает задание в качестве учетной информации субъектов и объектов их идентификаторов (например, имя пользователя и имя файлового объекта), а в качестве правил разграничения доступа — матрицы доступа D. При запросе доступа, поступающего в диспетчер доступа от субъекта, диспетчер из запроса получает идентификаторы субъекта и объекта. Затем он находит элемент матрицы доступа на основе учетных данных субъекта и объекта, осуществляет управление запросом доступа на основании выбранного элемента матрицы доступа.

С учетом того, что при управлении доступом диспетчером анализируется собственно матрица доступа, дискреционный механизм управления доступом является универсальным в том смысле, что им может быть реализована любая из рассмотренных выше модель управления доступом, в том числе и модель полномочного управления.

Ранее были представлены канонические модели управления доступом, задающие корректные разграничения в общем случае. Общность определяется тем, что однотипный канал взаимодействия субъектов доступа создается одновременно для всех субъектов. В частном случае подобный канал может устанавливаться не для всех субъектов. При этом разграничение диспетчером доступа должно осуществляться на основе частной матрицы доступа, получаемой из соответствующей канонической матрицы путем запрещения каналов взаимодействия. Пример частной матрицы доступа для модели управления доступом с дуплексными виртуальными каналами взаимодействия на основе активных симплексным каналов представлен ниже.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & \partial & \dots & 0 & \partial \\ \partial & Зп/Чт & \dots & \partial & \partial \\ \dots & \dots & \dots & \dots & \dots \\ \partial & Д & \dots & Зп/Чт & Д \\ 0 & 0 & \dots & Д & Зп/Чт \end{bmatrix} \end{array}$$

13.1.2. Механизмы реализации мандатной модели доступа

Метки безопасности

В отличие от дискреционного механизма управления доступом, с применением которого может быть реализована любая модель управления до-

ступом (посредством задания правил разграничения доступа в диспетчере матрицей доступа), мандатный механизм реализует полномочные модели управления доступом.

Как уже говорилось, основой мандатного механизма является включение в схему управления доступом так называемых меток безопасности (иерархических, так как в системе реализуется иерархия полномочий). Эти метки призваны отражать полномочия субъектов и объектов. При этом разграничение прав доступа в диспетчере уже может задаваться не матрицей доступа, а правилами обработки меток, на основании которых диспетчер принимает решение о предоставлении запрашиваемого доступа к ресурсу. В качестве же учетной информации субъекта и объекта доступа является метка безопасности. Впервые разграничение доступа на основе задания и обработки меток безопасности было предложено Белл ом и Ла-Падулой [4].

Метки безопасности являются элементами линейно упорядоченного множества M = {M1,...,Mk} и задаются субъектам и объектам доступа. Метки безопасности назначаются субъектам и объектам (группам субъектов и объектов). Они служат для формализованного представления их уровня полномочий.

Будем считать, что чем выше полномочия субъекта и объекта (меньше их порядковый номер в линейно полномочно упорядоченных множествах субъектов и объектов — C = {C1,..., Ск} и O = {O1,..., Ok}), тем меньшее значение метки безопасности Mi, i = 1,...,k им присваивается, т.е.: Ml < M2 < M3<...<Mk.

Таким образом, в качестве учетной информации субъектов и объектов доступа, кроме их идентификаторов (имен), в диспетчере доступа каждому субъекту и объекту задаются метки безопасности из множества M.

Правила разграничения доступа

Разграничение доступа диспетчером реализуется на основе правил, определяющих отношение линейного порядка на множестве M, где для любой пары элементов из множества M, задается один из типов отношения: {>,<,=}.

Примечание

На практике реализуется выбор подмножества M, изоморфного конечному подмножеству натуральных чисел — такой выбор делает естественным арифметическое сравнение меток безопасности.

Рассмотрим правила разграничения доступа для различных полномочных моделей управления доступом. При этом введем следующие обозначения:

- Ms — метка безопасности субъекта (группы субъектов) доступа;
- Mo — метка безопасности объекта (группы объектов) доступа;
- метка безопасности с порядковым номером i -- Mi устанавливается для субъекта доступа с порядковым номером i -- Ci и для объекта доступа с порядковым номером i — Oi.

Рассмотрение правил оформим в виде списка:

1. Полномочная модель управления доступом с произвольным управлением виртуальными каналами взаимодействия субъектов доступа.
   - субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс = Мо;
   - субъект С имеет доступ к объекту О в режиме «Записи» в случае, если выполняется условие: Мс = Мо;
   - субъект С имеет доступ к объекту О в режиме «Добавления» в случае, если выполняется условие: Мс > Мо.
2. Полномочная модель управления доступом с принудительным управлением виртуальными каналами взаимодействия субъектов доступа:
   - субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс <= Мо;
   - субъект С имеет доступ к объекту О в режиме «Записи» в случае, если выполняется условие: Мс = Мо.
3. Полномочная модель управления доступом с комбинированным управлением виртуальными каналами взаимодействия субъектов доступа:
   - субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс <= Мо;
   - субъект С имеет доступ к объекту О в режиме «Записи» в случае, если выполняется условие: Мс = Мо;
   - субъект С имеет доступ к объекту О в режиме «Добавления» в случае, если выполняется условие: Мс > Мо.

Примечание

Данная модель, при определенных допущениях, представляет собою модель Белла—Ла-Падулы [4].

Возможности мандатной модели доступа

Дадим мандатному механизму управления доступом определение с учетом сказанного ранее.

Под мандатным механизмом управления доступом, реализующим канонические полномочные модели управления доступом, понимается способ обработки запросов диспетчером доступа, основанный на формальном сравнении меток безопасности субъектов и объектов доступа в соответствии с заданными правилами.

Основой же мандатного механизма является реализация принудительного управления виртуальными каналами взаимодействия субъектов доступа. При этом основным требованием к принудительному управлению является обеспечение невозможности перенесения информации из объекта более высокого уровня конфиденциальности в объект с информацией более низкого уровня конфиденциальности. Поэтому к механизмам уп-

равления доступом, реализующим принудительное управление виртуальными каналами взаимодействия субъектов доступа, накладываются дополнительные ограничения к корректности реализации.

На практике для дискреционного механизма управления доступом, как правило, используется модель произвольного управления виртуальными каналами взаимодействия субъектов доступа. Однако управление каналами может быть и принудительным — все зависит от реализуемой матрицы доступа. В этом случае мандатный и дискреционный механизмы различаются только способом задания разграничительной политики в диспетчере доступа и обработки запросов на доступ.

Утверждение

Мандатный механизм управления доступом позволяет корректно реализовать полномочные модели управления доступом при условии, что всем субъектам и объектам доступа сопоставлены метки безопасности. Для дискреционного механизма, реализующего принудительное управление виртуальными каналами, это соответственно означает необходимость задания разграничений для всех субъектов и объектов доступа.

Доказательство данного утверждения очевидно. Нетрудно показать, что представленные правила реализуют разграничения доступа полностью адекватные соответствующим каноническим матрицам доступа D, отображающим полномочные модели управления доступом, в случае, если всем субъектам из множества C и объектам из множества O сопоставлены метки безопасности. При этом несопоставление метки безопасности какому-либо субъекту или объекту означает вычеркивание соответствующей строки или столбца из матрицы доступа D.

Таким образом, если существует объект, который не включен в схему мандатного управления доступом, то этот объект может являться средством несанкционированного взаимодействия пользователей, имеющих различные метки безопасности.

Таким образом, метки безопасности должны устанавливаться на все объекты файловой системы (логические диски (тома), каталоги, подкаталоги, файлы), а также на все иные объекты доступа — на устройства ввода/вывода и отчуждаемые носители информации, виртуальные каналы связи и т.д. Речь о требованиях к полноте разграничительной политики доступа к ресурсам пойдет ниже.

Утверждение

Мандатный механизм управления доступом позволяет корректно реализовать полномочные модели управления доступом при условии, что системой защиты реализуется требование к изоляции программных модулей (процессов) различных пользователей. То же справедливо и для дискреционного механизма, реализующего принудительное управление виртуальными каналами взаимодействия субъектов доступа.

Доказательство данного утверждения состоит в необходимости противодействовать скрытым каналам взаимодействия субъектов доступа. Если под явным каналом понимается объект, доступ к которому может быть разграничен, то под скрытым — любые иные возможности взаимодействия субъектов доступа, которые в этом случае должны исключаться, например, передача информации между субъектами доступа через буфер обмена и т.д. Очевидно, что если существует возможность передачи информации между процессами, запускаемыми с правами пользователей, которым назначены различные метки безопасности, то реализуется возможность переноса информации из объекта более высокого уровня конфиденциальности в объект более низкого уровня конфиденциальности.

Данное требование относится к процессам прикладных пользователей (которым назначаются метки безопасности). Поэтому, в первую очередь, данное требование выдвигается при реализации системы защиты для ОС семейства UNIX, где одновременно в системе могут быть запущены процессы различных пользователей. Для ОС семейства Windows одновременно запускаются процессы двух пользователей — текущего прикладного пользователя и виртуального пользователя «СИСТЕМА» (системные процессы). Однако, так как системные процессы не имеют средств управления пользователем, то выполнение рассматриваемого требования для ОС семейства Windows становится неактуальным.

Отметим, что мандатный механизм управления доступом может применяться лишь для реализации канонических матриц доступа. Для реализации частных матриц доступа мандатный механизм должен функционировать в диспетчере наряду с дискреционным механизмом. Дискреционный механизм здесь служит для разграничения прав доступа пользователей, обладающих одинаковой меткой безопасности.

Проиллюстрируем сказанное простым примером. Рассмотрим частную матрицу, представленную ниже.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Д & & Д & Д \\ Чт & Зп/Чт & & Д & Д \\ \dots & \dots & \dots & \dots & \dots \\ Чт & Чт & & Зп/Чт & Д \\ О & Чт & \dots & Чт & Зп/Чт \end{bmatrix} \end{array}$$

Чтобы реализовать данную матрицу доступа в дополнение к мандатному механизму управления доступом, реализующему каноническую полномочную модель управления доступом с комбинированным управлением виртуальными каналами взаимодействия субъектов доступа, следует запретить дискреционным механизмом управления доступом чтение субъектом C1 объекта Ok (закрыть соответствующий пассивный симплексный канал взаимодействия субъектов доступа Ck → C1).

Таким образом, обобщая сказанное, отметим, что **мандатный механизм управления доступом можно рассматривать как альтернативный дискреционному механизму способ реализации полномочных моделей управления доступом. Более того, с точки зрения реализуемых возможностей управления доступом данные механизмы адекватны при условии реализации одной и той же матрицы доступа.**

Преимуществом мандатного механизма является интуитивная понятность, а как следствие, и простота настройки диспетчера доступа в предположении, что интуитивно понятен механизм включения шкалы полномочий и назначения меток безопасности. При этом не требуется задания в диспетчере доступа матрицы доступа как таковой. Достаточно задать правила доступа, соответствующие реализуемой полномочной модели управления доступа, и метки безопасности.

Недостатком механизма является необходимость в общем случае наряду с мандатным механизмом использовать дискреционный механизм управления доступом. То есть реализация диспетчера доступа усложняется и остается необходимость в том или ином виде задания матрицы доступа.

Выше было показано, что все функции управления доступом (все матрицы доступа и соответствующие модели) могут быть реализованы дискреционным механизмом. При этом мандатный механизм является частным случаем дискреционного и задает лишь некоторые правила. Эти правила с одной стороны упрощают администрирование диспетчера доступа (за счет включения меток безопасности), а с другой стороны ограничивают возможные ошибки в администрировании.

Реализуется это за счет выполнения мандатным механизмом следующего требования: любой субъект и объект доступа, которому не присвоена метка безопасности, автоматически исключается из схемы управления доступа (какой-либо доступ непомеченного субъекта/доступ к непомеченному объекту — невозможны). При этом одно из основных требований мандатного механизма управления доступом -- управление потоками -- может быть реализовано только в рамках канонической модели, а разграничение диспетчером доступа должно осуществляться для всех субъектов ко всем объектам доступа на защищаемом объекте.

13.1.3. Общие положения по реализации управления доступом

Все вышеприведенные рассуждения можно оформить в виде следующих общих положений.

1. В общем случае могут быть выделены дискреционный и мандатный механизмы управления доступом. Под дискреционным механизмом управления доступом понимается способ обработки запросов дис-

петчером доступа, основанный на задании правил разграничения доступа в диспетчере непосредственно матрицей доступа D.

Под мандатным механизмом управления доступом, реализующим канонические полномочные модели управления доступом, понимается способ обработки запросов диспетчером доступа, основанный на формальном сравнении меток безопасности субъектов и объектов доступа в соответствии с заданными правилами. Причем мандатный механизм управления доступом корректно может реализовывать лишь канонические матрицы доступа.

Для реализации частных матриц доступа на основе мандатных разграничений данный механизм должен функционировать в диспетчере наряду с дискреционным механизмом (что, кстати говоря, задается и формализованными требованиями к механизмам управления доступом).

2. Мандатный механизм управления доступом позволяет корректно реализовать полномочные модели управления доступом при условии, что всем субъектам и объектам доступа сопоставлены метки безопасности. Для дискреционного механизма, реализующего принудительное управление виртуальными каналами, это соответственно означает необходимость задания разграничений для всех субъектов и объектов доступа.
3. Мандатный механизм управления доступом позволяет корректно реализовать полномочные модели управления доступом при условии, что системой защиты реализуется требование к изоляции программных модулей (процессов) различных пользователей. То же справедливо и для дискреционного механизма, реализующего принудительное управление виртуальными каналами взаимодействия субъектов доступа.

   Данное требование относится к процессам прикладных пользователей (которым назначаются метки безопасности). Поэтому, в первую очередь, данное требование выдвигается при реализации системы защиты для ОС семейства UNIX, где одновременно в системе могут быть запущены процессы различных пользователей. Для ОС семейства Windows одновременно запускаются процессы двух пользователей – текущего прикладного пользователя и собственно системы (системные процессы). Однако, так как системные процессы не имеют средств управления пользователем, то выполнение рассматриваемого требования для ОС семейства Windows становится неактуальным.
4. Так как все функции управления доступом (все матрицы доступа и соответствующие модели) могут быть реализованы дискреционным механизмом, мандатный является частным случаем дискреционного и задает лишь некоторые правила, с одной стороны упрощающие администрирование диспетчера доступа (за счет включения меток безопасности), с другой стороны ограничивающие возможные ошибки в администрировании за счет реализации механизмом управления доступом требования: любой субъект и объект доступа, кото-

рому не присвоена метка безопасности автоматически исключается из схемы управления доступа (какой-либо доступ непомеченного субъекта/доступ к непомеченному объекту — невозможны). При этом одно из основных требований мандатного механизма управления доступом -- управление потоками, — может быть реализовано только в рамках канонической модели, т.е. разграничение диспетчером доступа должно осуществляться для всех субъектов ко всем объектам доступа на защищаемом объекте.

5. В общем случае, говоря о требованиях к реализации управления доступом к ресурсам, следует иметь в виду требования к реализуемым моделям и матрицам доступа.

13.2. Механизмы задания меток безопасности. Категорирование прав доступа

Общие положения

Ввиду максимальной интуитивной понятности наиболее используемым на сегодняшний день механизмом задания меток безопасности является задание меток на основе уровня конфиденциальности информации и уровня прав доступа (допуска). При этом метки безопасности принято называть метками конфиденциальности.

В основе иерархической классификации информации находится достаточно хорошо формализованное ее категорирование, определяемое отнесением информации к соответствующему уровню конфиденциальности. Делается это на основании соответствующих нормативных документов и распоряжений. В рамках этих документов определены такие категории как: «открытая», «служебная», «конфиденциальная», «строго конфиденциальная», «секретная», «совершенно секретная» и т.д. Соответственно и пользователь для обработки соответствующей информации должен обладать требуемой формой допуска к информации.

Формализованное категорирование информации и категорирование формы допуска пользователей к информации без труда позволяет задать метки конфиденциальности, что существенно может упрощать задание канонической матрицы при настройке диспетчера доступа.

В общем случае категорирование прав доступа может проводиться и по другим принципам. При этом для различных принципов категорирования могут различаться и соответствующие модели. Например, категорирование может осуществляться на основе принципа «начальник—подчиненный». Для такого категорирования целесообразно запретить любой доступ подчиненного к файловым объектам начальника, а начальнику разрешить доступ ко всем файловым объектам подчиненного «на чтение». Реализуется такой подход путем использования полномочной модели

управления доступом с принудительным управлением виртуальными каналами взаимодействия субъектов доступа.

Нетрудно показать, что мандатный механизм управления доступом на практике может быть реализован и при отсутствии иерархии обрабатываемой в системе информации. Дело в том, что в системе присутствует информация пользователя и информация системы. Информация системы, как правило, располагается на системном диске — это различные конфигурационные файлы системы, файлы настроек приложений и т.д. Естественно, что к данной информации должен быть запрещен доступ пользователям «на запись» и разрешен только «на чтение», чтобы могли нормально функционировать система и приложения, запускаемые с правами текущего пользователя.

Таким образом, и в данном случае получаем схему мандатного управления доступом, где файловые объекты с данными пользователя должны иметь ту же метку, что и пользователь (разрешены «запись» и «чтение»), а системный диск должен иметь метку безопасности, разрешающую к нему доступ пользователей только «на чтение». То есть получаем, что уровень конфиденциальности пользовательской информации выше, чем системной, а значит при назначении соответствующим образом меток безопасности пользователь не сможет модифицировать системную информацию.

Неиерархические метки

Также на практике находит применение неиерархических меток (это можно рассматривать как вырожденный случай мандатного управления). На самом деле при этом реализуется дискреционный механизм, метки же служат для незначительного упрощения настройки механизма управления доступом (для формализации задания дискреционных разграничений).

Идея назначения неиерархических меток заключается в следующем. Обрабатываемая информация разделяется по функциональному назначению, например, бухгалтерская, персональные данные и т.д. По числу обрабатывемых типов информации вводятся метки Mi. Далее однотипная метка присваивается типу информации и пользователю (группе пользователей), которые имеют право обработки данной информации. Назначение однотипной метки предполагает полный доступ пользователя к информации, несовпадение меток — запрет доступа.

Таким образом, правило управления доступом при задании разграничений неиерархическими метками задается следующим образом:

1. Субъект С имеет полный доступ к объекту О в случае, если выполняется условие: Mc = Mo.
2. Субъект С не имеет доступа к объекту О в противном случае.

Далее, говоря о мандатном механизме управления доступом, будем предполагать, что используются иерархические метки безопасности.

13.3. Правила назначения меток безопасности иерархическим объектам доступа

Ранее объект доступа нами рассматривался как элемент, имеющий неиерархическую структуру. Вместе с тем, ряд объектов, например, файловые объекты, характеризуются иерархической структурой. Например, объект доступа файл может находиться в подкаталоге, который в свою очередь располагается в каталоге логического диска (или тома).

Возникает вопрос: каким образом назначать метки безопасности иерархическому объекту, с учетом того, что не только включаемый элемент (например, каталог для логического диска, подкаталог для каталога и т.д.) является объектом доступа, но и каждый включающий элемент иерархии (например, логический диск для каталога, каталог для подкаталога и т.д.) также априори является объектом доступа? То есть в соответствии с требованиями к корректности реализации мандатного механизма управления доступом метки безопасности должны устанавливаться всем объектам доступа, как включаемым, так и включающим.

Напомним, что современными универсальными ОС мандатный механизм управления доступом не реализуется, поэтому данная задача может решаться лишь средствами добавочной защиты.

13.3.1. Общие правила назначения меток безопасности иерархическим объектам доступа

Основу назначения меток безопасности иерархическим объектам доступа составляет использование следующего подхода. Пусть некоторому включаемому объекту (например, файлу) необходимо назначить метку безопасности. Это означает, что пользователю с соответствующей меткой будет дано право на чтение данного объекта и на запись в него. Соответственно включающим элементам должна присваиваться метка, не позволяющая рассматриваемому пользователю осуществить в них запись, но позволяющая осуществить чтение (чтобы просмотреть структуру включающего элемента) — т.е. метка ниже, чем метка включаемого элемента.

Реализация данного подхода позволяет сформулировать следующие общие правила назначения меток безопасности иерархическим объектам доступа.

1. Метки безопасности из множества M = {M1,...,Mk}, используемого в полномочной модели управления доступом, присваиваются объектам доступа (без учета их иерархии), к которым следует разграничивать доступ. Процедура назначение меток безопасности начинается с разметки данных объектов.

2. Метки безопасности должны присваиваться всем включающим элементам иерархии, вплоть до элемента, являющегося объектом доступа (к которому разграничивается доступ). Для разметки включающих элементов, не являющихся непосредственно объектами доступа, но к которым следует разграничить доступ, вводится метка Mk+1. Причем для элементов множества M должно выполняться условие: M1 < M2 < M3< ... <Mk < Mk+1.
3. Включаемому элементу может не присваиваться метка безопасности, тогда включаемый элемент наследует метку безопасности (имеет то же значение метки) включающего его элемента.
4. Вводится группа старших (корневых) элементов иерархии Ok+1, включающих объекты доступа. Данной группе объектов должна присваиваться метка безопасности Mk+1.
5. К группе старших (корневых) элементов иерархии Ok+1 при сопоставлении ей метки Mk+1 разрешается доступ по «чтению».

13.3.2. Правила разграничения доступа для различных полномочных моделей управления доступом к иерархическим объектам

Рассмотрим правила разграничения доступа для различных полномочных моделей управления доступом. Сделаем это применительно к иерархическим объектам, с учетом сформулированных выше правил назначения меток безопасности иерархическим объектам доступа. Правила для неирархических объектов приведены ранее, в п.13.2.

Полномочная модель управления доступом с произвольным управлением виртуальными каналами взаимодействия субъектов доступа

Правила разграничения доступа, реализуемые диспетчером доступа имеют следующий вид.

1. Для объектов Ol,...,Ok:
   - субъект C имеет доступ к объекту O в режиме «Чтения» в случае, если выполняется условие: Mc = Mo;
   - субъект C имеет доступ к объекту O в режиме «Записи» в случае, если выполняется условие: Mc = Mo;
   - субъект C имеет доступ к объекту O в режиме «Добавления» в случае, если выполняется условие: Mc > Mo.
2. Любой субъект C имеет доступ к объекту Ok+1 в режиме «Чтения».

Полномочная модель управления доступом с принудительным управлением виртуальными каналами взаимодействия субъектов доступа

Правила разграничения доступа, реализуемые диспетчером доступа имеют следующий вид.

1. Для объектов Ol,...,Ok:
   - субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс <, = Мо;
   - субъект С имеет доступ к объекту О в режиме «Записи» в случае, если выполняется условие: Мс = Мо.
2. Любой субъект С имеет доступ к объекту Ok+1 в режиме «Чтения».

Полномочная модель управления доступом с комбинированным управлением виртуальными каналами взаимодействия субъектов доступа

Правила разграничения доступа, реализуемые диспетчером доступа имеют следующий вид.

1. Для объектов O1,...,Ok:
   - субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс <, = Мо;
   - субъект С имеет доступ к объекту О в режиме «Записи» в случае, если выполняется условие: Мс = Мо;
   - субъект С имеет доступ к объекту О в режиме «Добавления» в случае, если выполняется условие: Мс > Мо.
2. Любой субъект С имеет доступ к объекту Ok+1 в режиме «Чтения».

13.3.3. Обоснование корректности механизма мандатного управления доступом к иерархическим объектам

При использовании приведенных правил назначения меток безопасности в матрице доступа, описывающей полномочную модель управления доступом, появляется дополнительная строка, соответствующая группе объектов Ok+1, элементами которой будут «Чт» — операция «чтение».

Пример матрицы доступа D для полномочной модели управления доступом с комбинированным управлением виртуальными каналами взаимодействия субъектов доступа представлена ниже.

$$D = \begin{array}{c} \\ 0, \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \\ O_{k+1} \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_{k-1} & C^* \end{array} \\ \begin{bmatrix} Зп/Чт & Д & \dots & Д & Д \\ Чт & Зп/Чт & \dots & Д & Д \\ \dots & \dots & \dots & \dots & \dots \\ Чт & Чт & \dots & Зп/Чт & Д \\ Чт & Чт & \dots & Чт & Зп/Чт \\ Чт & Чт & \dots & Чт & Чт \end{bmatrix} \end{array}$$

Модель управления доступом формально может быть описана следующим образом. Элемент (Dij) матрицы Dij = Зп/Чт, если i = j; Dij = Чт, если k+1 > i > j. Соотвественно Dij = Д, если i < j < k+1, и Dij = Чт, если i = k+1, где i — порядковый номер объекта (номер строки в матрице доступа), а j — порядковый номер субъекта (номер столбца в матрице доступа).

Утверждение

Диспетчер доступа, осуществляющий разграничение доступа на основе представленных правил, реализует корректно механизм мандатного управления доступом к иерархическим объектам доступа.

Доказательство утверждения достаточно очевидно. Матрица доступа D дополняется строкой Ok+1, для которой создается виртуальный пассивный симплексный канал взаимодействия субъектов доступа — все элементы строки «Чт». Поэтому в объекты группы Ok+1 ни один субъект не может записать информацию. Следовательно, данный канал взаимодействия субъектов доступа не позволяет получить несанкционированный доступ к информации (несанкционированно переместить информацию). Он необходим только для чтения структуры включающих элементов иерархии.

Введение рассмотренных правил назначения меток безопасности иерархическим объектам доступа обусловливает необходимость противодействия доступу пользователей к остаточной информации. Ранее мы отмечали, что основой данного противодействия является реализация механизма обес-. печения замкнутости программной среды, в дополнение к которой может осуществляться гарантированная очистка остаточной информации.

Поясним необходимость сказанного. Пусть в объекте Ok+1 (таким образом разметили логический диск) находятся два файла: объекты Ok-m и Ok−1. И пусть данным объектам присвоены различные метки безопасности. Тогда при удалении одного из объектов на диске сохраняется остаточная информация, к которой штатными средствами доступ пользователи получить не могут. Однако при определенных условиях (средствами прямого доступа к диску) они могут получить доступ вне рамок мандат-

ного механизма, поскольку остаточная информация не имеет признаков объекта и к ней не может разграничиваться доступ.

Примечание

Ради справедливости стоит отметить что подобными средствами (если разрешить их запуск на компьютере) пользователь может получить доступ не только к остаточной, но и к актуальной информации, т.к. они обращаются не к объекту, а напрямую к диску, минуя механизм управления доступом.

13.3.4. Примеры назначения меток безопасности

Рассмотрим примеры применения введенных правил.

Пример 1. Пусть на логическом диске D: вводится три каталога, соответственно, D:\2, D:\3, D:\4 (реализуется мандатный механизм управления доступа к этим каталогам) и пусть соответствующим образом назначаются метки безопасности каталогам — 2, 3, 4. В этом случае корневым включающим элементом является диск D (объект Ok+1). Ему должна быть присвоена метка Mk+1=5. Иллюстрация назначаемых меток безопасности объектам доступа для рассматриваемого примера приведена в табл. 13.1.

Пример назначения меток безопасности иерархическим объектам доступа *Таблица 13.1*

Метка безопасности	Субъекты доступа	Объекты доступа
2		D:\2
3		D:\3
4		D:\4
5		D:

Пример 2. Рассмотрим более сложную иерархическую структуру. Пусть на логическом диске D: вводится два каталога, соответственно, D:\2, D:\3, в каталоге D:\2 расположен объект доступа -- файл User 1, который должен иметь метку 2, и пусть в каталоге D:\3 присутствуют два объекта доступа — файлы User 2 и User 3, соответствующим образом размечаемые — имеют метки 3 и 4. В этом случае включающим корневым элементом является диск D (объект Ok+1) -- ему присваивается метка Mk+1 = 5. Иллюстрация назначаемых меток безопасности для рассматриваемого примера приведена в табл. 13.2.

Пример назначения меток безопасности иерархическим объектам доступа *Таблица 13.2*

Метка безопасности	**Субъекты доступа**	**Объекты доступа**
?		**D:\2**
3		**D: \3 \User 2**
4		**D: \3 \User 3**
5		**D:**

В качестве замечания необходимо отметить, что в примере разметки, изложенном в табл. 13.2, элемент `D:\з` наследует метку включающего элемента `D:`. Для файла User `1` достаточно назначить метку включающему его каталогу `D:\2`, которую он наследует.

13.3.5. Настройка мандатного механизма доступа к иерархическим объектам

Настройка мандатного механизма управления доступом с иерархической структурой объектов доступа отличается от настройки мандатного механизма управления доступом с неиерархической структурой объектов доступа следующим:

* Вводится дополнительная группа объектов (Ok+1), которую образуют корневые включающие объекты доступа элементы.
* Вводится дополнительная метка безопасности Mk+1 (причем Ml < M2 < M3<...<Mk < Mk+1), которая присваивается корневым включающим объекты доступа элементам (для файловой системы – логическим дискам или томам), образующим группу дополнительно вводимую группу объектов Ok+1.
* Метка Mk+1 наследуется всеми включаемыми элементами иерархии, вплоть до первого размеченного объекта в иерархии (которые уже, в свою очередь, должны размечаться метками безопасности из исходного множества M).

При реализации в диспетчере доступа рассмотренных выше правил назначения и обработки меток безопасности иерархических объектов доступа, можем говорить об общности мандатного механизма управления доступом применительно к структуре объекта доступа.

Рассмотренная реализация мандатного механизма управления доступом является универсальной в том смысле, что объектом доступа (ресурсом, к которому разграничивается доступ посредством назначения меток безопасности) может являться любой элемент иерархии (например, для файловой системы — логический диск, каталог, подкаталог, файл).

Процедура создания группы объектов Ok+1 и назначение ей метки безопасности Mk+1 легко формализуется и может быть реализована диспетчером доступа автоматически. При такой реализации диспетчера доступа настройка мандатного механизма (задание правил разграничения доступа в диспетчере доступа) с иерархическими объектами доступа не сложнее, чем с неиерархическими объектами доступа.

13.4. Анализ возможности корректной реализации моделей управления доступом встроенными в ОС механизмами защиты

13.4.1. Анализ возможности корректной реализации канонических моделей управления доступом

Теперь проведем анализ возможности корректной реализации моделей управления доступом встроенными в ОС механизмами защиты. Мандатный механизм современными универсальными ОС не реализуется в принципе (данная возможность может быть обеспечена исключительно средствами добавочной защиты). Поэтому остановимся на рассмотрении реализации дискреционного механизма. При этом будем рассматривать возможность корректной реализации механизмов в ОС, когда «владельцем» объекта файловой системы является администратор безопасности.

Исследования проведем для ОС семейства Windows (NT/2000/XP) и ОС семейства UNIX. При этом отметим, что для обеих веток развития ОС семейства UNIX — System V и BSD возможности дискреционного механизма практически совпадают.

Прежде всего отметим принципиальное отличие в реализации механизма для данных семейств ОС (несмотря на внешнюю схожесть). Для ОС семейства Windows разграничение для файла (включаемого объекта) приоритетнее, чем разграничение для каталога, включающего данный файл. Другими словами, если для ОС UNIX установить запрет доступа пользователю к каталогу, то какие бы атрибуты доступа не были установлены на файл, расположенный в данном каталоге, доступ к файлу пользователь получить не сможет.

Для ОС Windows в аналогичной ситуации пользователь сможет получить доступ к файлу (если в файле установлены права доступа к нему пользователем), т.к. разграничения доступа включаемого объекта приоритетнее, чем включающего. Правда, ради справедливости, отметим, что такой доступ возможен посредством программного средства, позволяющего непосредственно обратиться к файлу, минуя процедуру обзора содержимого каталога. К таким средствам не относится большинство широко применяемых на практике программ-проводников. Вместе с тем, не представляется сколько-нибудь сложным написать подобную программу самостоятельно. Ниже представлен пример программы, позволяющей обращаться непосредственно к файлу, минуя обзор содержимого каталога:

```
int main(int argc, char* argv[])
{
  HANDLE hFile;
  BOOLEAN bResult;
```

```
  hFile = CreateFile(argv[1] , GENERIC_READ,
     FILE_SHARE_READ,
     0, OPEN_EXISTING,
     FILE_ATTRIBUTE_NORMAL, 0);
     CHAR buf[0x140]; ULONG length=0x140, nBytesRead,
     nBytesWrite, keyKbd;
  do
  {
   bResult = ReadFile(hFile, buf, length, &nBytesRead, NULL) ;
   WriteFile(GetStdHandle(STD_OUTPUT_HANDLE), buf, nBytesRead,
             &nBytesWrite, 0);
   keyKbd = getch();
  }
  while (bResult && nBytesRead && keyKbd!=27);
  CloseHandle(hFile);
  return 0;
}
```

Теперь остановимся на рассмотрении возможностей по реализации приведенных моделей встроенными в ОС средствами. Общим здесь будет то, что «владельцем» любого создаваемого файла должен являться «администратор безопасности». Разграничения для пользователей устанавливается включением пользователя в группу и назначением прав доступа группе.

Естественно, что основным объектом разграничения доступа становится включающий объект файловой системы (например, каталог), т.к. если устанавливать разграничения для файлов, то администратор вынужден будет лично (как «владелец») устанавливать права на каждый вновь создаваемый файл. По умолчанию же можно установить только одинаковые права доступа на все создаваемые файлы, что не может быть достаточно.

Таким образом, разграничение доступа устанавливается на включающие объекты (каталоги), а «владельцем» всех каталогов является администратор. Созданием групп пользователей и заданием разграничений для них устанавливаются различные права доступа пользователей к каталогу. В пределе каждый пользователь получает полный доступ к своему каталогу — реализуется каноническая модель. Далее устанавливается наследование настроек для создаваемых в каталогах файлов.

Рассмотрим, что произойдет в различных ОС при подобных настройках (используемых для задания «владельцем» администратора безопасности).

Общим будет то, что пользователь, создавая новый файл в каталоге, может изменить «владельца», а как следствие, и права доступа к создаваемому файлу. Таким образом, права доступа к включаемому и включающему объектам файловой системы входят в противоречие. В результате пользователь (либо какая-либо программа от его лица) может назначить для файла несанкционированные права доступа. Возможность получения данных прав определяется тем, какие разграничения (к включающему, либо к включаемому) объекту приоритетны. Как было отмечено ранее, в

этом и состоит принципиальное различие реализации дискреционного механизма для ОС UNIX и Windows.

Для ОС семейства UNIX приоритетны разграничения на включающий элемент, для ОС Windows -- на включаемый. Таким образом, в ОС UNIX пользователь не сумеет получить доступа к файлу, в ОС Windows — сумеет.

С учетом проведенного исследования можем сделать следующий вывод: **каноническая модель дискреционного доступа может быть реализована встроенными механизмами защиты ОС семейства UNIX и не реализуема в принципе для ОС семейства Windows.**

13.4.2. Анализ возможности корректной реализации моделей управления доступом с каналом взаимодействия субъектов доступа

Выше мы рассматривали возможность реализации канонической модели, характеризуемой полным разграничением доступа. Теперь рассмотрим, какие типы каналов взаимодействия субъектов доступа могут быть реализованы для ОС семейства UNIX. Ключевым вопросом здесь является возможность реализации атрибута «добавление» встроенными средствами.

Примечание

Как таковой, атрибут «добавление» в ОС UNIX отсутствует. Кстати говоря, он существует в ОС Windows, но для этой операционной системы, как показано выше, каноническая модель встроенными средствами не может быть реализована в принципе.

«Добавление» встроенными средствами может быть реализовано следующим образом. Пользователю, которому должно разрешаться добавление данных в каталог другого пользователя, следует разрешить доступ к этому каталогу «на запись». При этом данному пользователю должен быть запрещен доступ «на запись» к уже существующим в данном каталоге файлам другого пользователя. Однако, как отмечали ранее, запрет доступа к файлам, расположенным в каталоге, возможно лишь непосредственно пользователем, создавшим эти файлы, что, строго говоря, противоречит требованию «права доступа к объекту файловой системы должны назначаться (изменяться) только администратором безопасности».

С учетом проведенного исследования можем сделать следующий вывод: корректно (в рамках рассматриваемых предположений) дискреционный механизм управления доступом может быть реализован лишь в рамках построения модели с физическим каналом взаимодействия субъектов доступа. Недостатки данной модели были нами рассмотрены ранее.

Отсюда следует вывод о необходимости применения добавочных средств защиты:

- для ОС Windows с целью возможной корректной модели дискреционного управления доступом к ресурсам в принципе;
- для ОС семейства UNIX с целью эффективной реализации канала взаимодействия субъектов доступа (реализации виртуального канала) для дискреционного механизма управления.

13.5. Механизм исключения субъектов и объектов из схемы управления доступом

Особенность реализации мандатного механизма управления доступом состоит в том, что все субъекты и объекты доступа должны быть помеченными (им должна быть назначена метка безопасности). Вместе с тем, не все субъекты и объекты доступа могут подпадать под разграничительную политику, реализуемую мандатным механизмом управления доступом. Таким образом, возникает ряд противоречий, не позволяющих корректно реализовать мандатный механизм управления доступом в полном объеме. Например, пользователю «Администратор» явно недостаточно прав доступа с использованием операции «запись» только к собственному объекту, не говоря уже о виртуальном пользователе ОС Windows «Система». То же самое можно сказать и об объектах доступа, например, об объекте «Корзина» для ОС Windows и др.

С целью разрешения данных противоречий в схему мандатного управления доступом могут быть включены группы субъектов и объектов с номером «О»: СО, ОО. Для разметки данных субъектов и объектов будем говорить о включении в схему управления доступом нулевой метки МО (признака «0»). Присвоение данной метки позволяет исключить субъект, либо объект доступа из схемы мандатного управления доступом. Соответственно, доступ субъекта, обладающего меткой МО, не подпадает под типовые мандатные разграничения, а доступ к объекту, обладающему меткой МО, не разграничивается (запрос диспетчером доступа не обрабатывается).

Обратите внимание, что речь идет только об исключении из схемы мандатного управления доступом. При этом исключаемые из мандатной схемы субъекты и (или) объекты доступа остаются элементами схемы дискреционного управления доступом, реализуемого в дополнение к мандатному.

Пример матрицы доступа D с возможностью исключения субъектов и объектов из схемы мандатного управления доступом, приведен ниже.

$$D = \begin{array}{c} \\ O_0 \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \\ O_{k+1} \end{array} \begin{array}{c} \begin{array}{ccccccc} C_0 & C_1 & C_2 & \dots & C_{k-1} & C, & \end{array} \\ \begin{bmatrix} НО & НО & НО & \dots & НО & НО \\ НО & Зп/Чт & Д & \dots & \partial & \partial \\ НО & Чт & Зп/Чт & \dots & \partial & \partial \\ \dots & \dots & \dots & \dots & \dots & \dots \\ НО & Чт & Чт & \dots & Зп/Чт & Д \\ НО & Чт & Чт & \dots & Чт & Зп/Чт \\ НО & Чт & Чт & \dots & Чт & Чт \end{bmatrix} \end{array}$$

Здесь исходное множество прав доступа, используемое для реализации канала взаимодействия субъектов доступа, дополняется элементом «НО» {Зп/Чт, Чт, Д, НО}, где элемент «НО» означает, что запрос не обрабатывается мандатным механизмом управления доступа.

Приведенная модель управления доступом формально может быть описана следующим образом:

- Dij = Зп/Чт, если i = j;
- Dij = Чт, если k+1 > i > j; i,j > 0;
- Dij = Д, если i < j < k+1; ij > 0;
- Dij = Чт, если i = k+1; ij > 0;
- Dij = НО, если i = 0 или j = 0,

где i -- порядковый номер объекта (номер строки в матрице доступа);
j — порядковый номер субъекта (номер столбца в матрице доступа).

Таким образом, включение в схему мандатного управления метки МО (и соответствующих ей правил доступа) разрешает отмеченное выше противоречие. Теперь всем субъектам и объектам доступа могут и должны назначаться мандатные метки, и в то же время как субъекты, так и объекты доступа могут исключаться из схемы типовых мандатных разграничений доступа. В этом и заключается использование механизма исключения субъектов и объектов доступа из схемы мандатного управления.

13.6. Управление доступом к устройствам и отчуждаемым накопителям (дискетам, CD-ROM-дискам)

13.6.1. Общий подход к реализации

Ранее было отмечено, что мандатный механизм управления доступом реализуется корректно только в том случае, когда элементами схемы мандатных разграничений являются все субъекты и объекты доступа защищаемого объекта.

Рассмотрим изложенные ранее возможности управления доступом применительно к устройствам ввода/вывода (съемным устройствам) и к отчуждаемым физическим накопителям — дискетам, CD-ROMам и т.д.

Итак, общий формат определения ресурса устройства (накопителя) выглядит следующим образом:

- имя съемного `устройства\каталог\подкаталог\...\файл` — для доступа к файлу;
- имя съемного `устройства\каталог\подкаталог` — для доступа к каталогу (подкаталогу);
- имя съемного устройства -- для доступа ко всему устройству в целом, которое может содержать каталоги и файлы (например, к устройству ввода данных — дисководу или CD-ROM).

Таким образом, управление доступом как к устройствам в целом, так и к ресурсам накопителей, полностью аналогично управлению доступом к файловым объектам.

13.6.2. Способы назначения ресурсам меток безопасности (способы разметки)

Определим способы разметки (назначения меток безопасности) ресурсов. Размечаться могут как собственно устройства (например, дисковод), так и накопители, размещаемые в устройстве (например, дискета).

Разметка устройства

При разметке устройства метка безопасности должна присваиваться собственно устройству, например, A:. Присвоение устройству метки означает разрешение полного доступа (чтение и запись) к устройству только пользователей с аналогичной меткой. Правила доступа остальных пользователей к устройству определяются реализуемыми каналами взаимодействия субъектов доступа в матрице доступа.

Устройства могут быть как размеченные (на них распространяются разграничения доступа), так и неразмеченные. Если доступ к устройству, например, E: не следует разграничивать мандатным механизмом управления доступом, то данному устройству следует сопоставить метку «МО». При этом к устройству реализуется только дискреционное разграничение доступа, осуществляемое в дополнение к мандатному механизму. В этом случае устройство по прежнему считается размеченным. Если же и дескреционные разграничения не устанавливаются, то к устройству получают полный доступ все пользователи и устройство считается неразмеченным. Соответственно, в этом случае на накопителе могут быть сохранены любые объекты и все объекты могут быть считаны любым пользователем.

Разметка накопителя

Существуют различные возможности разметки накопителя.

1. **Разметка накопителя для возможности сохранения на нем объектов только одного уровня.** На накопителе (например, дискете, размещаемой в дисководе A:) санкционированным пользователем создается новый объект — каталог (или файл). Объект помечается, то есть ему присваивается имя. При этом имя соответствует метке объектов файловой системы (либо метке пользователей), которым разрешена запись на данный накопитель. Например, создается каталог A:\3 — накопителю устанавливается метка 3. После этого на данный накопитель могут записываться (соответственно в каталог A:\3 дискеты) только объекты *с* меткой 3. Читать объекты с размеченного накопителя и добавлять в них информацию могут пользователи в соответствии с реализуемым каналом взаимодействия субъектов доступа в матрице доступа.
2. **Разметка накопителя для возможности сохранения на нем объектов нескольких уровней.** На накопителе (например, дискете, размещаемой в дисководе B:) санкционированным пользователем создаются новые объекты — каталоги (или файлы). Объекты размечаются, то есть им присваиваются имена, соответствующие меткам объектов файловой системы (либо меткам пользователей), которым разрешена запись на данный накопитель. Например, если на накопителе создаются каталоги B:\2 и B:\3, то соответствующим объектам накопителя устанавливаются метки 2 и 3.

   После этого на данный накопитель могут записываться соответствующие объекты файловой системы (осуществлять запись соответствующие пользователи), соответственно, в каталог B:\3, только объекты с меткой 3, в каталог B:\2, только объекты с меткой 2.

   Читать данные каталоги и добавлять в них информацию пользователи могут только в соответствии с реализуемым каналом взаимодействия субъектов доступа в матрице доступа.

Таким образом, при вводе объекта с «помеченного» устройства (или выводе на «помеченное» устройство) диспетчер доступа обеспечивает соответствие между меткой вводимого (выводимого) объекта и меткой устройства. Аналогичное соответствие обеспечивается при работе с «помеченным» отчуждаемым накопителем.

Итак, на основании вышеприведенных рассуждений можно сделать заключение, что **все рассмотренные модели управления доступом могут применяться как для разграничения прав доступа к файловым объектам, так и к устройствам ввода/вывода и к отчуждаемым накопителям.**

13.7. Управление доступом к разделяемым сетевым ресурсам

В разделяемых сетевых ресурсах — устройствах и файловых объектах (общих папках), как в объектах файловой системы, могут быть реализованы все рассмотренные выше возможности, т.е. дискреционный и мандатный механизмы разграничения прав доступа.

Однако особенностью данных ресурсов является то, что они сетевые – т.е. расположены на другом компьютере. Вследствие этого возможны альтернативные подходы к реализации механизма управления доступом к разделяемым ресурсам (распределенная и централизованные схемы управления). Это определяется тем, что при реализации разграничительной политики доступа к разделяемым ресурсам возможны два подхода:

- разграничение на компьютере, с которого осуществляется запрос (схема распределенного разграничения доступа);
- разграничение на компьютере, на котором располагается разделяемый ресурс (схема централизованного разграничения доступа).

В частности, второй подход реализован в ОС Windows для NTFS. К достоинствам данного подхода можно отнести простоту настройки разграничительной политики доступа, в предположении, что все разделяемые ресурсы располагаются на одном компьютере (файл-сервере). В общем же случае — на каждом компьютере присутствуют свои разделяемые ресурсы (при распределенном общем ресурсе). Соответсвенно настройку разграничительной политики доступа следует осуществлять для каждого компьютера, что огарничивает применение данного подхода. К недостаткам также можно отнести ограниченные, по сравнению с первым подходом, возможности разграничений. В частности, любой доступ к разделяемому ресурсу может задаваться для пользователей сети, вне зависимости от их расположения (на каких компьютерах в сети они зарегистрированы). При этом отметим, что в принципе не предотвращается возможность атаки собственно на протокол NETBIOS, т.к. фильтруется не исходящий, а входящий трафик.

Альтернативный подход состоит в реализации управления доступом к разделяемым ресурсам на компьютере, с которого осуществляется доступ. В этом случае реализуются все возможности разграничений доступа, которые рассмотрены ранее для управления доступом к локальным файловым объектам — можно не только задавать, какой пользователь, с какого компьютера имеет право доступа к разделяемому ресурсу (какому ресурсу), но и с применением какого приложения разрешен подобный доступ (разграничение доступа по процессам), что позволяет противодействовать атакам NETBIOS, использовать приложения, обладающие дополнительными возможностями защиты, в частности, шифрования и т.д.

При этом будем иметь ввиду, что при распределенном управлении доступом необходимо учитывать следующее:

* так как фильтруется только исходящий трафик, то на всех компьютерах ЛВС должна устанавливаться система защиты;
« так как регистрация (аудит) доступа к общему ресурсу ведется распределенно, в сети должна обеспечиваться синхронизация времени на всех компьютерах (например, с сервера безопасности).

Отметим, что и при централизованном управлении доступом также необходима защита всех компьютеров ЛВС. В противном случае на компьютере, с которого осуществляется доступ к разделяемому ресурсу, может быть несанкционированно заведен пользователь, которому разрешен удаленный доступ к разделенному ресурсу (т.к. разграничение осуществляется только по именам пользователей).

При распределенном управлении доступом к разделяемым ресурсам встает задача синхронизации времени на компьютерах ЛВС (вообще говоря, данная задача существует и вне данного механизма защиты, здесь она , лишь более ярко выражена). Это связано с тем, что на компьютере, к которому удаленно обращаются, можно зарегистрировать процесс, который обратился к ресурсу, а самого пользователя нельзя.

По имени процесса (системный или прикладной) можно определить, локальный или удаленный доступ произведен к ресурсу. Удаленного же пользователя, обратившегося к ресурсу, можно идентифицировать в этом случае только по данным аудита компьютера, с которого произведено обращение. Так как обращение может одновременно осуществляться с нескольких компьютеров, то для однозначной идентификации пользователя время на компьютерах в сети должно быть синхронизировано.

В рамках синхронизации должны решаться две задачи:

» установка времени на включаемом компьютере из единого центра (с этой целью целесообразно использовать сервер безопасности, т.к. при включении компьютера он устанавливает соединение с сервером безопасности);
» запрет доступа пользователей к локальным настройкам системного времени на защищаемом компьютере.

На практике, на наш взгляд, целесообразно использовать оба подхода к разграничению доступа к разделяемым сетевым ресурсам. При этом, наряду со встроенными возможностями централизованного разграничения доступа следует применять добавочный механизм распределенного разграничения, т.к., во-первых, они функционально дополняют друг друга и, во-вторых, могут рассматриваться при решении задачи резервирования основных механизмов защиты, к которым относятся механизмы управления доступом, в том числе, и к разделяемым ресурсам.

глава 14

Субъект доступа «ПРОЦЕСС» и его учет при разграничении доступа

14.1. Включение субъекта «ПРОЦЕСС» в схему управления доступом

14.1.1. Процесс, как субъект доступа

Выше рассматривались классические схемы управления доступом к ресурсам, реализуемые на основе дискреционного и мандатного механизмов управления доступом. В качестве субъекта доступа для них понимается «ПОЛЬЗОВАТЕЛЬ». При этом доступ к объектам, в соответсвии с заданными правилами, разграничивается именно для пользователей.

Тем не менее, ранее было отмечено, что в качестве самостоятельного субъекта доступа может выступать процесс, так как в общем случае он может запускаться не от лица текущего пользователя (например, системные процессы).

Возможности механизмов управления доступом могут быть существенно расширены при включении в субъекты доступа субъекта «ПРОЦЕСС» [11, 12, 19], для которого, аналогично субъекту доступа «пользователь», могут разграничиваться права доступа на основе задаваемой матрицы доступа D.

Все сказанное ранее (применительно к субъекту доступа «пользователь») может быть отнесено и к случаю, когда в качестве субъекта доступа выступает «процесс». Соответственно множество $C = \{C1,...,Ck\}$ — линейно упорядоченные множества процессов. В качестве субъекта доступа «процесс» Ci, $i = 1,...,k$ рассматривается как отдельный процесс, так и группа процессов, характеризуемая одинаковыми правами доступа к объектам.

В частности, каноническую модель управления доступом можно представить матрицей доступа D, имеющей следующий вид:

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \end{array}$$

где «О» обозначает отсутствие доступа процесса к объекту, а «1» — полный доступ (например, разрешены типы доступа «Запись» и «Чтение» для файловых объектов).

Под канонической моделью управления доступом для линейно упорядоченных множеств процессов (групп процессов) и объектов (групп объектов) доступа будем понимать модель, описываемую матрицей доступа, элементы главной диагонали которой «1» задают полный доступ процессов к объектам, остальные элементы «О» задают запрет доступа процессов к объектам.

Аналогично сказанному ранее для субъектов доступа «процесс» в модели управления доступом могут быть реализованы либо выделенные, либо виртуальные каналы взаимодействия субъектов доступа.

Следуя определению канонической модели, можем сделать вывод, что включение в схему управления субъекта доступа «процесс» позволяет в системе локализовать объекты доступа (например, области дисковой памяти, устройства и т.д.) для отдельных приложений и иных групп процессов. Верно и обратное: можно локализовать процессы (приложения) для отдельных объектов доступа (данных).

14.1.2. Схема управления доступом для субъекта «ПРОЦЕСС»

Итак, в общем случае следует различать два вида субъекта доступа — «пользователь» и «процесс». Поэтому в общем случае диспетчером доступа по отношению к объектам должны реализовываться следующие возможности:

- разграничение прав доступа к объектам процессов вне разграничений пользователей;
- разграничение прав доступа к объектам пользователей вне разграничений процессов;
- комбинированное разграничение прав доступа — разграничение прав доступа к объектам процессов в рамках разграничений пользователей (совместное разграничение доступа процессов и пользователей).

Субъект доступа «процесс» является вторичным по отношению к субъекту доступа «пользователь». В отличие от субъекта «пользователь» он может либо присутствовать, либо отсутствовать в схеме разграничения прав доступа.

На рис. 14.1 приведена логика обработки запроса доступа к объекту, реализуемая диспетчером доступа для двух случаев: при разграничении прав пользователей и при комбинированном разграничении прав доступа [12].

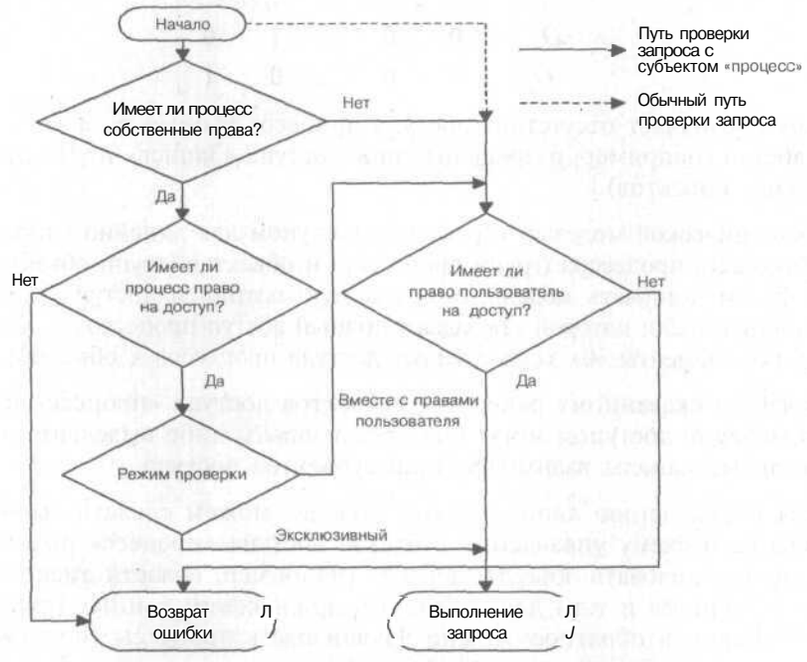


Рис. 14.1. *Логика обработки запроса доступа к объекту диспетчером доступа*

Рис. 14.2. *Схема обработки запроса доступа к объекту*

На рис. 14.2 приведена схема обработки запроса доступа к объекту, реализуемая диспетчером доступа в КСЗИ «Панцирь» при разграничении прав доступа для субъектов «пользователь» и «процесс» [31, 32].

Обозначения используемых на схеме функциональных блоков: блок анализа запроса доступа к ресурсу — 1, блок формирования отказа в доступе к ресурсу — 2, блок анализа прав доступа процесса — 3, блок разрешения/запрета доступа процессу к ресурсу — 4, блок анализа прав доступа пользователя — 5, блок разрешения/запрета доступа пользователя к ресурсу — 6, блок формирования запроса доступа к ресурсу — 7.

Работает схема реализации диспетчера доступа следующим образом. В блок 1 поступает запрос пользователя на доступ к ресурсу, содержащий идентификатор ресурса и требуемое действие (чтение, запись, выполнение). Блок 1 формирует из запроса учетную информацию запрашиваемого доступа — идентификатор ресурса и требуемое действие над ним. Затем блок 1 выдает данную информацию в блоки 3 и 5.

В блок 5 поступает имя (идентификатор) пользователя, запросившего ресурс, в блок 3 — идентификатор (полный путь) процесса, запросившего ресурс. Сначала блок 3 определяет, какие права доступа к ресурсу разрешены процессу, запросившему ресурс, и какие права доступа к ресурсу им запрошены. Затем он сравнивает заданные и запрошенные права доступа. Кроме того, блок 3 анализирует, каким способом обрабатываются права доступа процесса к ресурсу: эксклюзивно, либо с правами пользователя.

При совпадении заданных и запрошенных прав блоком 3 вырабатывается код разрешающего сигнала, поступающий в блок 4. При несовпадении заданных и запрошенных для процесса прав блок 4 подаст сигнал в блок 2, и пользователю будет сообщено об отказе в доступе.

При эксклюзивном режиме обработки процесса блок 4 пропускает запрос сразу в блок 7. При этом права пользователя не учитываются.

В режиме обработки запроса процесса вместе с правами пользователя, запрос блоком 4 в блок 7 не выдается. В этом случае, при совпадении для процесса заданных и запрошенных прав, блок 4 передает информацию в блок 6.

Теперь рассмотрим как производится анализ прав пользователя. А производится он блоком 5, и результат передается в блок 6. Соответсвенно, в случае несовпадения для пользователя заданных и запрошенных прав, ему блоком 6 через блок 2 будет отказано в доступе. Если права пользователя оказались достаточными, то блок 6 будет ждать разрешающего сигнала от блока 4, который разрешает/запрещает доступ процессу.

14.1.3. ВЫВОДЫ

1. Ввиду того, что в схеме управления доступом присутствуют два субъекта доступа «ПОЛЬЗОВАТЕЛЬ» и «ПРОЦЕСС», причем процесс может запускаться и не от лица текущего пользователя (например, системные процессы), с целью корректного решения задачи управления доступом к ресурсам следует осуществлять разграничения для обоих этих субъектов доступа.
2. Необходимо учитывать, что процесс может запускаться и не с правами текущего пользователя, то есть в системе существуют привилегированные (системные) процессы. В связи с этим для корректной реализации управления доступом должны предусматриваться следующие схемы разграничений для субъектов доступа:
   - Разграничение прав доступа к объектам процессов вне разграничений пользователей.
   - Разграничение прав доступа к объектам пользователей вне разграничений процессов.
   - Комбинированное разграничение прав доступа — разграничение прав доступа к объектам процессов в рамках разграничений пользователей (совместное разграничение доступа процессов и пользователей).

14.2. Разграничение доступа к системному диску

14.2.1. Процессы в ОС Windows 95/98/ME

В ОС семейства Windows возможен запуск пользовательских (прикладных пользователей) и системных (виртуального пользователя «система») процессов. При этом некоторые ОС, например, Windows 95/98/ME, не позволяют идентифицировать, относится ли запускаемый процесс к системному, либо к пользовательскому. То есть любой процесс может быть идентифицирован только его именем и именем пользователя, от лица которого он был запущен.

Особенностью системных процессов является то, что для нормального функционирования системы они должны осуществлять не только чтение данных из системного диска (области внешней памяти, где хранятся каталоги и файлы собственно ОС), но и запись на системный диск.

Из-за того, что все процессы в ОС Windows 95/98/Me сопоставляются с пользователем, то разграничение доступа для процессов может сводиться к разграничению доступа для пользователей. Если же рассматривать механизмы защиты, реализующие разграничение только для одного типа

субъектов доступа — субъекта «пользователь», то в общем случае доступ к системному диску не может быть запрещен ни для чтения, ни для записи. Таким образом, не может быть корректно реализован ни дискреционный, ни мандатный механизмы управления доступом, т.к. присутствуют объекты, доступ к которым не может быть разграничен между пользователями системы.

Проиллюстрируем сказанное с использованием матриц доступа D, где системный диск обозначим как Ос.

В соответствии с введенным ранее определением модель управления доступом на основе произвольного управления виртуальными каналами взаимодействия субъектов доступа в рассматриваемом случае описывается матрицей доступа D, имеющей следующий вид:

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_c \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Д & \dots & Д & \partial \\ Д & Зп/Чт & \dots & \partial & \partial \\ \dots & \dots & \dots & \dots & \dots \\ Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт \\ \dots & \dots & \dots & \dots & \dots \\ Д & Д & \dots & Зп/Чт & Д \\ Д & Д & \dots & Д & Зп/Чт \end{bmatrix} \end{array}$$

Аналогично матрица доступа D для полномочной модели управления доступом с комбинированным управлением виртуальными каналами взаимодействия субъектов доступа в рассматриваемом случае принимает вид, представленный ниже:

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_c \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Д & \dots & Д & Д \\ Чт & Зп/Чт & \dots & \partial & \partial \\ \dots & \dots & \dots & \dots & \dots \\ Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт \\ \dots & \dots & \dots & \dots & \dots \\ Чт & Чт & \dots & Зп/Чт & Д \\ Чт & Чт & \dots & Чт & Зп/Чт \end{bmatrix} \end{array}$$

Из анализа приведенных моделей можем сделать вывод о невозможности корректного решения задачи управления доступом к ресурсам (не реализуема каноническая модель доступа) так как существует объект (системный диск), доступ к которому для субъектов «пользователь» разграничить невозможно.

14.2.2. Процессы в ОС Windows NT/2000 и UNIX

Для ОС Windows NT/2000, в которых различают пользовательские и системные процессы встроенными в ОС средствами, а также для ОС UNIX, где системные процессы запускаются с правами «root», рассматриваемая проблема имеет иное трактование. Механизмами разграничения прав доступа этих систем может разграничиваться доступ только для пользователей (для пользовательских процессов). Для системных же процессов разграничения установить невозможно в принципе. Это обусловливает очень распространенную атаку — запуск несанкционированного процесса с системными правами.

Кроме того, не представляется возможным осуществить разграничение для приложений, запускаемых с системными правами. Это усугубляет последствия ошибок в данном ПО. Так, если вернуться к существующей статистике угроз, то можем увидеть, что данным угрозам подвержены практически все существующие ОС. Причем доля подобных угроз весьма существенна.

Если обозначить группу системных процессов Сс, то, например, модель управления доступом на основе произвольного управления виртуальными каналами взаимодействия субъектов доступа описывается матрицей доступа D, имеющей следующий вид:

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccccc} C_1 & C_2 & \dots & C_c & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & Д & \dots & Зп/Чт & \dots & Д & Д \\ Д & Зп/Чт & \dots & Зп/Чт & \dots & Д & Д \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Д & Д & \dots & Зп/Чт & \dots & Зп/Чт & Д \\ Д & Д & \dots & Зп/Чт & \dots & Д & Зп/Чт \end{bmatrix} \end{array}$$

Это наглядно показывает невозможность корректной реализации моделей разграничения доступа для современных ОС без возможности разграничения доступа для субъекта «ПРОЦЕСС».

14.2.3. Субъект доступа «ПРОЦЕСС», системные процессы и доступ к системному диску

Введем в схему управления доступом субъект доступа «процесс». Разделим процессы на системные и пользовательские. При этом для системных процессов установим режим эксклюзивной проверки прав доступа (вне анализа прав доступа пользователей). Этим процессам разрешим в простейшем случае полный доступ только к системному диску («на запись» и «на чтение») и запретим доступ к каталогам пользователей.

Всем пользователям запретим права доступа к системному диску «на запись». Права доступа «на чтение» системного диска пользователям оставим, так как в противном случае могут некорректно функционировать приложения.

В результате произведенных установок получаем, что пользователь не сможет обратиться к системному диску прикладным процессом «на запись». Таким образом, управление доступом реализуется корректно. При этом любой системный процесс не сможет получить доступ к данным пользователя, что в принципе снимает большой круг проблем, связанных с НСД к информации с правами системного процесса. В то же время системные процессы имеют полный доступ к системному диску, либо к необходимым объектам системного диска.

К системным процессам для различных ОС семейства Windows, которым требуется разрешить доступ к системному диску «на запись» для корректного функционирования ОС, относятся KERNEL32, SYSTEM и др.

14.2.4. Привилегированные процессы

Отметим, что в общем случае доступ «на запись» к системному диску может потребоваться разрешить не только системным процессам, но и некоторым процессам приложений. Например, если в системе реализуется добавочная система защиты, которая располагается на системном диске, то ее процессам потребуется обращаться к системному диску «на запись».

Определение.

Процессы, которым при функционировании ОС и приложений требуется обращаться к системному диску для записи на него информации, т.е. процессы, доступ которых к системному диску требуется разграничивать эксклюзивно (вне прав пользователей), будем называть **привилегированными.** Привилегированные процессы — это системные процессы (процессы ОС) и процессы приложений, которым для корректного функционирования необходим доступ «на запись» на системный диск. Как уже отмечалось, примером таких процессов могут служить процессы системы защиты.

Необходимость введения привилегированных процессов обусловлена тем, что на пути реализации мандатного механизма управления доступом системных процессов к файловым объектам возникают определенные сложности. Например, в ОС Windows 9X/Me системные процессы запускаются с правами текущего пользователя. А значит всем пользователям, независимо от их меток, должен быть разрешен доступ «на запись» и «на чтение» к системному диску. Но при этом мандатный механизм не может быть реализован в принципе, так как к одному объекту имеют право «на запись» пользователи с разными метками.

Данную проблему как раз и позволяет решить выделение привилегированных процессов, которые должны рассматриваться вне прав пользователей. При этом всем пользователям может быть запрещен доступ к системному диску.

Рассмотрим реализацию мандатного механизма управления доступом применительно к способу назначения субъектам и объектам мандатных меток. Сделаем это с учетом разграничений, вводимых для привелигерованных процессов. Для этого рассмотрим матрицу доступа D для мандатного управления доступом. При этом будем использовать дополнительные метки МО и Mk+1, обоснованность которых показана ниже.

$$D = \begin{array}{c} \\ O_0 \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccccc} C_0 & C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} НО & НО & НО & \dots & НО & НО \\ НО & Зп/Чт & Д & \dots & \partial & Д \\ НО & Чт & Зп/Чт & \dots & \partial & \partial \\ \dots & \dots & \dots & \dots & \dots & \dots \\ НО & Чт & Чт & \dots & Зп/Чт & \partial \\ НО & Чт & Чт & \dots & Чт & Зп/Чт \end{bmatrix} \end{array}$$

Введем следующие предположения:

* будем считать, что привилегированные процессы (соответствующие исполняемые файлы) не могут быть каким-либо образом модифицированы пользователем и под их именем не могут быть запущены пользовательские процессы (эта задача решается механизмами обеспечения замкнутости программной среды, которые будут рассмотрены ниже);
* будем считать, что на системном диске не располагаются пользовательские каталоги и файлы (информация). Кстати говоря, это общепринятая практика организации файловых объектов.

В данных предположениях привилегированным процессам можно назначить метку МО — ввести их в группу субъектов СО, а системному диску метку Mk+1 — ввести его в группу объектов Ok+1. При таком назначении меток безопасности доступ привилегированных процессов не разграничивается и они имеют доступ ко всем файловым объектам. В том числе они имеют полный доступ к системному диску.

Примечание

В качестве дополнения можно дискреционным механизмом ограничить доступ системным процессам только к системному диску (причем только к необходимой его части).

Все пользовательские приложения имеют доступ к системному диску только «на чтение». Таким образом мандатный механизм управления доступом реализуется корректно. При этом системный диск не становится тем объектом, доступ к которому «на запись» необходимо разрешать всем пользователям.

14.2.5. Общие рекомендации

С учетом сказанного ранее сформулируем общие рекомендации по управлению доступом к системному диску.

1. Запрещать доступ к системному диску «на запись» следует не только с целью обеспечения корректности реализации управления доступом (отсутствуют несанкционированные каналы взаимодействия субъектов доступа), но и с целью предотвращения модификации файловых объектов ОС. Без защиты ОС от модификации пользователями вообще нельзя говорить о какой-либо защите информации на компьютере.
2. С целью возможного резервирования механизма управления доступом к ресурсам (как основного механизма противодействия НСД) функции разделения процессов на системные и пользовательские должны быть выполнены средствами механизма добавочной защиты, т.к. эти функции являются важнейшими функциями механизма управления доступом к ресурсам. Резервирование было рассмотрено в п. 3.2.
3. Применительно к созданию диспетчера доступа добавочными средствами защиты необходимо реализовывать перечисленные ранее требования к нему. В том числе должно быть выполнено требование: «управление доступом должно осуществляться, как для явных действий субъекта, так и для скрытых».

В частности, при реализации диспетчера доступа для ОС Windows NT/2000 следует учитывать то, что при использовании длинных имен файловых объектов к ним можно обращаться и по длинному, и по короткому имени. Например, к каталогу «\Program files\» можно обратиться по короткому имени «\Progra~1\». Поэтому при задании правил разграничения доступом при указании пути к файлам или каталогам, следует устанавливать права доступа для обоих имен файловых объектов. Понятно, что они должны быть одинаковыми.

Необходимо также учитывать, что в ОС Windows NT/2000 имена (каталогов, файлов), набранные русскими (либо в иной кодировке) буквами, также имеют короткое имя, которое формируется с использованием кодировки Unicode (внешне они могут существенно различаться). Например, короткое имя для каталога «C:\Documents and Settings\USER1\Главное меню» выглядит как «C:\Docume~1\USER1\5D29~1\». Поэтому при использовании русских имен (или иной кодировки) в обозначении файловых объектов для покрытия всех видов обращения к таким ресурсам, также следует устанавливать права доступа (одинаковые) для соответсвующих имен файловых объектов. Отметим, что при разработке диспетчера доступа добавочного средства защиты информации данные функции могут быть реализованы автоматически.

На основании всего вышесказанного можно сделать вывод, что включение в схему управления доступом субъекта «ПРОЦЕСС», позволяющего средствами механизмов защиты различать пользовательские и привилегированные процессы, предоставляет возможность корректной реализации разграничений прав доступа к системному диску и для системных процессов. Корректно реализовать это можно только с помощью средств добавочной защиты. Самими современными универсальными ОС это в полном объеме не обеспечивается.

14.3. Разграничение доступа к реестру ОС семейства Windows

Все сказанное выше по поводу разграничения доступа к системному диску относится и к настроечному реестру ОС. При этом объектами доступа здесь являются ветви и ключи реестра (по аналогии с файловыми объектами). В данных объектах находятся настройки как собственно ОС, так и установленных на защищаемом компьютере приложений. Соответственно, доступ «на запись» пользователям к реестру ОС должен быть запрещен.

Запрещать пользователям доступ «на запись» к реестру ОС следует не только с целью предотвращения возможной модификации ими настроек ОС и приложений, но и с целью предотвращения несанкционированной инсталляции пользователем программных средств (как правило, при инсталляции ПО осуществляется запись необходимой для программы настроечной информации в реестр ОС).

Однако как сама ОС, так и приложения в процессе функционирования могут обращаться к реестру не только «на чтение», но и «на запись». Поэтому, при организации доступа к реестру так же как и к системному диску, следует разграничивать доступ для процессов и для пользователей. Права доступа процессов должны рассматриваться как эксклюзивно, так и совместно с правами пользователя.

При этом могут выделяться привилегированные процессы, которым необходимо разрешить права доступа к реестру «на запись» эксклюзивно. К таким процессам могут быть отнесены системные процессы ОС и приложений, а также процессы системы защиты информации.

Заметим, что ОС Windows, реализующие технологию NT, содержат встроенный механизм разграничения доступа к реестру для пользователей. Однако он не обладает возможностями разграничивать доступ к реестру ОС для процессов, то есть не обеспечивает корректного решения задачи управления доступом.

Таким образом, диспетчер доступа к реестру ОС должен обеспечивать управление доступом к реестру ОС субъектам «ПОЛЬЗОВАТЕЛЬ» и

«ПРОЦЕСС». Причем права доступа субъекта «ПРОЦЕСС» могут устанавливаться эксклюзивно. Эксклюзивные права доступа «на запись» следует устанавливать привилегированным процессам, доступ же пользователям к реестру ОС должен разрешаться только «на чтение».

Отметим, что ключ реестра гипотетически может служить и для передачи информации между пользователями (если пользователь туда может записывать информацию). Поэтому реестр ОС следует рассматривать в качестве объекта, доступ к которому должен быть разграничен при реализации в системе мандатного механизма управления доступом. При этом назначать какие-либо метки безопасности элементам реестра ОС (ветвям и ключам) не имеет никакого смысла. Однако необходимо учитывать реестр ОС как потенциально возможный объект, через который может передаваться информация между пользователями с различающимися метками безопасности.

Поэтому будем говорить, что реестр ОС — это объект мандатного разграничения доступа (обязательно должен учитываться в схеме мандатного управления), которому при мандатном управлении доступом назначена метка МО. Напомним, что метка МО — признак исключения из схемы мандатного управления, то есть разграничения осуществляются дискреционным механизмом.

Дискреционным механизмом должны устанавливаться эксклюзивные права доступа «на запись» привилегированным процессам. Доступ же пользователям к реестру ОС должен разрешаться только «на чтение».

14.4. Ввод новых данных в систему при мандатном управлении доступом

Использование субъекта доступа «ПРОЦЕСС», а также возможности исключения его из схемы управления доступом, позволяют корректно решить задачу санкционированного ввода новых данных в систему.

Для мандатного механизма управления доступом выдвигается следующее формализованное требование [1]: **система защиты при вводе новых данных в систему должна запрашивать и получать от санкционированного пользователя классификационные уровни (метки безопасности) этих данных.**

Подобное требование предполагает, что в систему могут вводиться новые данные, в частности, с внешних устройств ввода, например, с дисковода. Кроме того, предполагается, что в системе имеется санкционированный (следовательно, авторизуемый) пользователь, который может назначать метки безопасности вновь вводимым данным — относить вводимые объекты к соответствующему классификационному уровню. При этом отметим, что текущий пользователь может ввести данные только определенного классификационного уровня (в соответствии с его допус-

ком к информации). Поэтому в общем случае существует некоторое ответсвенное лицо, которое должно иметь возможность в соответсвии со своими правами вводить новые данные в компьютер. Причем это лицо может и не являться пользователем.

Благодаря возможности разграничивать доступ эксклюзивно для процессов, задачу санкционированного ввода данных можно решить следующим образом:

1. В системе устанавливается программа-проводник (например, программа Far, Windows Commander и т.д.), предназначенная исключительно для решения рассматриваемой задачи ввода новых данных.
2. Данной программе-проводнику назначается метка МО, т.е. программа получает доступ к ресурсам вне мандатных разграничений.
3. Дискреционным механизмом управления доступом для данной программы (соответствующего ей процесса) устанавливаются права доступа к тем файловым объектам, куда могут быть внесены новые данные, и к тем устройствам, с которых могут быть внесены новые данные.
4. Чтобы разрешить ввод новых данных в систему под любым текущим пользователем (без перезагрузки компьютера), необходимо, чтобы все пользователи имели право на запуск данной программы. При этом для данной программы-проводника реализуется режим запуска с авторизацией ответственного лица (санкционированного пользователя). При этом задается имя и пароль ответственного лица, которые могут не совпадать с учетными данными ни одного зарегистрированного в системе пользователя. Причем запуск программы-проводника возможен только после авторизации ответственного лица под любым текущим пользователем.

Только ответственное лицо (санкционированный пользователь) под любым текущим пользователем может запустить программу-проводник. При этом программе-проводнику назначена метка безопасности МО, т.е. для нее не разграничиваются права доступа мандатным механизмом управления доступом. Соответственно, запустив данную программу, санкционированный пользователь (под текущим пользователем — без перезагрузки компьютера) может ввести с внешнего устройства новые данные и разместить их в любом файловом объекте, т.е. отнести вводимые объекты к соответствующему классификационному уровню.

14.5. Механизмы принудительного использования оригинальных приложений

14.5.1. Принудительное использование оригинальных приложений при доступе к ресурсам

Включение в схему управления доступом субъекта «ПРОЦЕСС» предоставляет широкие возможности по принудительному применению на защищаемом объекте оригинальных приложений. Важность этого обусловлена тем, что большинство существующих приложений, применяемых на защищаемом компьютере, ориентированы на максимальную универсальность, а не на максимальную защищенность. Кроме того, для решения некоторых функциональных задач обеспечения информационной безопасности может потребоваться применение оригинальных приложений, при условии, что пользователю будет дана возможность использовать именно эти приложения для доступа к ресурсам.

Принудительное использование оригинальных приложений состоит в том, что доступ к ресурсу разрешается только определенному приложению и не разрешается стандартным приложениям.

Например, в качестве приложения-проводника может использоваться специальная программа, дополнительно обеспечивающая обязательное шифрование файлового объекта при его сохранении (копировании) в общей папке (разделяемый сетевой ресурс на сервере). При этом данные шифруются, как при передаче по каналу связи, так и при хранении на файл-сервере, при его вводе/выводе с/на дискету (например, диск A:, как файловый объект). Благодаря этому данные не могут быть перенесены с использованием отчуждаемого физического носителя на незащищенный компьютер. Аналогично может быть реализована работа пользователя при доступе во внешнюю сеть.

Заметим, что данные задачи могут решаться и посредством разработки соответствующих драйверов, осуществляющих необходимые преобразования автоматически («прозрачно» для пользователя). Данный же подход позволяет решать рассматриваемые задачи на уровне приложений.

14.5.2. Реализация активного симплексного канала взаимодействия субъектов доступа

Общие положения

Ранее отмечалось, что необходимым условием корректности используемого механизма управления доступом является реализация активного симплексного канала взаимодействия субъектов доступа, основанного на использовании операции «добавление» (что позволяет реализовать вир-

туальные каналы взаимодействия субъектов доступа). Операция «добавление» -- это запись с предотвращением модификации существующих файловых объектов. То есть запрещается копировать каталог или файл, если в объекте, куда записываются данные, уже существует каталог или файл с таким именем.

Однако, как отмечалось ранее, для ОС семейства UNIX данная возможность не поддерживается, а для ОС Windows вообще не приходится говорить о корректности реализации дискреционной модели управления доступом.

Рассмотрим, как реализовать данную возможность механизмом добавочной защиты на прикладном уровне.

Пусть добавочная система защиты содержит прикладную программу-проводник, которая обладает следующими возможностями:

» может идентифицировать текущего пользователя в системе;
* имеет функцию «Обзор». Отличительной особенностью реализации данной функции является то, что в обзоре копируемых объектов должны отображаться только те файловые объекты, к которым текущий пользователь имеет доступ «на запись», а в обзоре объектов, куда осуществляется копирование, только те объекты, к которым текущий пользователь имеет доступ «на добавление»;
« выполнять функцию «копирование» («запись» без возможности стирания и модификации существующих файловых объектов).

Можно отметить, что практически всеми подобными возможностями обладает проводник Far. Однако при попытке скопировать файл, приводящей к модификации существующего файла, Far осуществляет запрос у пользователя на модификацию существующего файла. Рассматриваемый нами проводник в подобной ситуации должен просто запрещать процедуру копирования, сообщая пользователю, что копирование файла с таким именем невозможно.

Установим разграничение доступа для процессов. При этом программе-проводнику, обеспечивающей выполнение операции «добавление», средствами диспетчера доступа разрешим доступ «на запись» в каталоги, в которые пользователям необходимо добавлять информацию (таким образом организуем виртуальный канал взаимодействия субъектов доступа). Остальным программам-проводникам (не поддерживающим выполнение операции «добавление») доступ к данным каталогам запретим.

Таким образом, оригинальная программа-проводник позволяет реализовать виртуальный канал взаимодействия субъектов доступа с реализацией атрибута доступа «добавление».

При реализации мандатного механизма управления доступом функциональная сложность оригинальной программы-проводника несколько возрастает. Это обусловливается тем, что операция «добавление» должна

быть реализована по-разному для различных файловых объектов — для каталогов (логических дисков или томов) и файлов.

Так если метки безопасности установлены не на файлы, а на включающие объекты — каталоги, то при «добавлении» в каталог более высокого уровня (меньшее значение метки безопасности) должна выполняться операция копирования в данный каталог без возможности стирания и модификации существующих в каталоге файловых объектов.

Если метки безопасности устанавливаются на файлы, то операция «добавление» состоит не в переносе (копировании) файла, а в изменении метки безопасности данного файла. При этом физически файл не переносится — не копируется.

Программа-проводник для реализации активного симплексного канала при назначении меток каталогам

Оригинальная программа-проводник для реализации активного симплексного канала взаимодействия субъектов доступа для мандатного механизма при разметке (назначении меток безопасности) каталогов должна обеспечивать решение следующих задач:

- идентифицировать текущего пользователя в системе;
- иметь функцию «Обзор». Отличительной особенностью реализации данной функции является то, что в обзоре копируемых объектов должны отображаться только те файловые объекты (каталоги), к которым текущий пользователь имеет доступ «на запись», т.е. имеющих ту же метку, что и у текущего пользователя. Что касается обзора объектов, куда осуществляется копирование, то должны отображаться только те объекты, к которым текущий пользователь имеет доступ «на добавление», то есть объекты, имеющие уровень выше (значение метки безопасности меньше). Если в дополнение к мандатному реализуется дискреционный механизм, то в обоих случаях должны учитываться ограничения «на запись», налагаемые этим механизмом;
- выполнять функцию «копирование», то есть «запись» без возможности стирания и модификации существующих файловых объектов.

Программа-проводник для реализации активного симплексного канала при назначении меток файлам

Оригинальная программа-проводник для реализации активного симплексного канала взаимодействия субъектов доступа для мандатного механизма при разметке (назначении меток безопасности) файлов должна обеспечивать решение следующих задач:

- идентифицировать текущего пользователя в системе;
- иметь функцию «Обзор». Отличительной особенностью реализации данной функции является то, что в обзоре объектов (файлов), кото-

рым может быть переназначена метка безопасности текущим пользователем, должны отображаться только те файловые объекты, к которым текущий пользователь имеет доступ «на запись», то есть имеющие ту же метку, что и у текущего пользователя. Если в дополнение к мандатному реализуется дискреционный механизм, должны учитываться ограничения «на запись», налагаемые этим механизмом;

- Вместо обзора объектов, куда осуществляется копирование, должен выводиться обзор меток безопасности (имеющих меньшее значение, чем у текущего пользователя), которые текущий пользователь может присваивать файлу;
- выполнять функцию переназначения метки безопасности файла — уменьшать значение метки безопасности файла.

Примечание

При переназначении метки безопасности файлу, файл становится недоступен пользователю, переназначившему метку.

Таким образом, возможность разграничения прав доступа для субъекта «ПРОЦЕСС» позволяет сделать доступ пользователя к защищаемым объектам только через принудительное использование специальной программы-проводника, характеризуемой определенными свойствам. Это может широко использоваться при построении добавочной защиты, в частности, с целью корректной реализации дискреционного и мандатного механизмов управления доступом к ресурсам.

14.6. Локализация прав доступа стандартных приложений к ресурсам

Включение в схему управления доступом субъекта «ПРОЦЕСС» предоставляет широкие возможности по локализации прав доступа стандартных приложений к ресурсам, что, прежде всего, позволяет эффективно противодействовать скрытым угрозам. Рассмотрим лишь некоторые примеры реализации дополнительных функциональных возможностей защиты информации.

14.6.1. Локализация прав доступа к ресурсам виртуальных машин

Как отмечалось ранее, важнейшим требованием при реализации диспетчера доступа является следующее: «управление доступом должно осуществляться как для явных действий субъекта, так и для скрытых». Под «явными» подразумеваются действия, осуществляемые с использованием санкционированных системных средств, а под «скрытыми» — иные действия, в том числе с использованием собственных программ субъектов доступа.

Наиболее критичными, с точки зрения обеспечения информационной безопасности, являются скрытые каналы взаимодействия виртуальных машин. К таковым, например, относятся макросы для офисных приложений и нерегламентированные действия JVM (например, виртуальная машина JAVA).

Серьезные проблемы безопасности, связанные с защитой от несанкционированных действий виртуальных машин, вызваны тем, что виртуальная машина «видна» только именем своего исполняемого файла (например, winword.exe) или процессом внешнего интерпретатора команд (если интерпретатор команд не встроен в приложение). При этом машина имеет встроенные средства программирования, например, VBA для создания макросов в редакторе Word и других офисных приложениях.

Встроенными средствами программирования виртуальной машины могут быть запрограммированы скрытые каналы доступа к ресурсам. При этом невозможно будет определить запуск соответствующей несанкционированной программы (соответственно, разграничить для нее доступ к ресурсам), т.к. в качестве запущенного процесса будет отображаться имя процесса виртуальной машины вне зависимости от того, какую внутреннюю программу она исполняет.

Настройкой прав доступа виртуальных машин (процесса виртуальной машины, либо процесса внешнего интерпретатора команд) к различным частям файловой системы можно создать некоторую «рабочую область» для данной виртуальной машины (например, каталог), вне которой она не сможет что-либо изменять. Сделать это возможно благодаря средствам разграничения доступа для субъекта «ПРОЦЕСС». При этом для виртуальных машин, доступ которых к ресурсам следует разграничивать отдельно, можно:

* предотвратить запуск каких-либо программ, установленных на компьютере (если запретить к ним доступ виртуальной машины);
* предотвратить доступ к настроечным и критичным файлам данных, располагающемся на компьютере (в том числе к системному диску).

Таким образом, можно минимизировать возможный ущерб от скрытых каналов доступа к ресурсам, которые априори могут быть организованы средствами программирования виртуальной машины [19]. Решение данной задачи является весьма существенным моментом в рамках реализации механизма обеспечения замкнутости программной среды.

14.6.2. Локализация прав доступа к ресурсам стандартных приложений со встроенными средствами защиты информации

Ранее рассматривались вопросы защиты информации на уровне ОС. Тем не менее ряд приложений имеют собственные механизмы защиты. В результате возникает задача эффективной интеграции механизмов защи-

ты ОС и приложений в единой автоматизированной системе. Проиллюстрируем сказанное на примере реализации СУБД.

Выше рассматривались механизмы управления доступом к файловым объектам, где объектами доступа являлись структурные элементы файловой системы — логический диск (или том), каталог, файл. Важной проблемой защиты информации является включение в объекты доступа элементов приложений, в частности, таблиц для СУБД. Рассмотрим возникающие при этом проблемы на примере реализации разграничений доступа для СУБД.

СУБД осуществляет управление доступом к таблицам. Все таблицы, доступ к которым разграничивается между пользователями, могут находиться в одном или нескольких файлах, т.е. разграничения для файлов и таблиц в общем случае не совпадают. Однозначное соответствие реализуется только в том случае, когда каждая таблица базы данных располагается в собственном файле.

Т.к. разграничения доступа для файлов и таблиц в одной системе в общем случае не совпадают, возникает противоречие. С одной стороны, чтобы разрешить доступ пользователю к таблице на более низком уровне иерархии (системном), ему должен быть разрешен доступ ко всему файлу. С другой стороны, один файл в СУБД, как правило, содержит таблицы нескольких различных пользователей. Таким образом, не выполняется требование к диспетчеру доступа, в рамках которого управление доступом должно осуществляться как для явных действий субъекта (доступ к файлу), так и для скрытых (доступ к таблицам внутри файла).

Корректно задача управления доступом решается здесь только в том случае, если функционируют разграничения доступом обоих уровней — и для файлов, и для таблиц. Это становится возможным только при использовании специальных приложений, которые при доступе к файлам БД позволяют осуществлять управление доступом к таблицам. Используя иные приложения, например, проводники, пользователь сумеет получить доступ ко всему файлу, т.е. получить несанкционированный доступ к таблицам других пользователей — имеет место скрытый канал доступа к ресурсам.

С целью разрешения отмеченного противоречия в современных распространенных СУБД реализуется следующий подход. При инсталляции СУБД в ОС создается специальный пользователь, от имени которого работает сам процесс СУБД. Иным пользователям средствами защиты ОС доступ к файлам БД должен запрещаться (средствами администрирования механизмов защиты ОС). Внутри СУБД осуществляется разграничение прав доступа на уровне таблиц БД непосредственно самой СУБД. Т.е. СУБД, по сути, встраивается в ОС, создавая собственный поток обращений к таблицам БД. Очевидно, что подобный подход к решению задачи интеграции механизмов защиты обусловлен возможностями механизмов управления доступом, встроенных в ОС (т.к. СУБД здесь выступает в качестве приложения и должна опираться на встроенные возможности ОС).

Включение в субъекты доступа субъекта «ПРОЦЕСС», позволяет принципиально иным образом решать данную задачу при построении СУБД, т.к. в этом случае настройками диспетчера доступа можно организовать доступ к файлам, содержащим таблицы, только приложением, имеющим встроенные средства разграничения доступа к таблицам. Иным же приложениям (в частности проводникам, работающим с файлами) доступ к файлам, содержащим таблицы, должен быть запрещен. Таким образом, применение рассматриваемой возможности управления доступом позволяет кардинально изменить подход к интеграции механизмов защиты ОС и приложений.

Аналогичным образом может быть локализован доступ к ресурсам и другим стандартным приложениям со встроенными механизмами защиты информации. При этом рассматриваемая возможность управления доступом может в значительной мере упростить подходы к реализации приложений со встроенными механизмами защиты, в смысле их интеграции со средствами защиты, реализуемыми на уровне ОС.

14.7. Управление доступом, посредством назначения меток безопасности субъектам доступа «ПРОЦЕСС». Мандатный механизм управления доступом процессов к ресурсам защищаемого объекта

14.7.1. Метки безопасности процессов

Ранее рассматривалось назначение меток безопасности субъекту доступа «ПОЛЬЗОВАТЕЛЬ». В этом случае основой для назначения метки безопасности служили уровень конфиденциальности информации и уровень доступа пользователя.

Теперь необходимо определиться исходя из чего следует назначать метки процессам (приложениям). В качестве критерия в этом случае может рассматриваться уровень защиты обрабатываемой информации, обеспечиваемой самим приложением (его внутренними механизмами). При этом могут быть сопоставлены метка безопасности объекта (категории информации) и метка безопасности приложения (процесса). Делается это в соответствии с требованиями к механизмам защиты по обработке информации соответствующей категории.

Итак, **под меткой безопасности процесса (приложения) будем понимать классификационную метку, назначаемую в соответствии с уровнем защищенности, обеспечиваемым механизмами защиты приложения.**

При этом условиями корректной настройки механизма мандатного управления доступом процессов являются следующие положения:

1. Метка безопасности, назначаемая процессу (сетевой службе), определяется меткой безопасности информации, обрабатываемой этой службой (должна совпадать с ней).
2. Метка безопасности, устанавливаемая на исполняемый файл процесса для задания полномочий на его запуск, должна совпадать с меткой безопасности, назначаемой пользователю.

14.7.2. Управление доступом с использованием меток процессов

Возможности управления доступом на основе меток безопасности, назначаемых приложениям (процессам), рассмотрим на примере мандатного управления доступом к виртуальным каналам внешней сети. Под виртуальным каналом будем понимать сетевую службу, определяющую информационную технологию обработки данных во внешней сети. Примером такой службы могут служить служба открытой почты, служба конфиденциальной почты и т.д.

С учетом сказанного ранее, метка безопасности присваивается сетевой службе (процессу). При этом мандатные разграничения для процесса действуют вне мандатных разграничений для пользователя. В соответствии со своей меткой безопасности помеченный процесс может обращаться к объектам доступа. При этом механизмом разграничения доступа к сети заносятся дополнительные дискреционные разграничения для помеченной службы. В частности с какими компьютерами разрешено взаимодействие и т.д.

Рассмотрим пример организации простого взаимодействия по внешней сети. Пусть система содержит два почтовых приложения для передачи почтовых сообщений, соответственно, в открытом и в защищенном виде (например, почтовые сообщения шифруются при передаче). Назначим почтовым приложениям (процессам) следующие метки безопасности:

- метку Мп-1 процессу конфиденциальной почтовой службы;
- метку Мп процессу открытой почтовой службы.

Данные настройки реализуют следующую возможность взаимодействия в рамках помеченной сетевой службы (почтового взаимодействия). В рамках конфиденциальной почтовой службы любым пользователем (отметим, что метки безопасности пользователям не назначались) по защищенному почтовому каналу могут передаваться объекты, которым сопоставлены метки Мп-1 и Мп. То есть только эти объекты служба сможет читать из файловой системы. Любое получаемое по защищенной почте сообщение любой пользователь сможет записать только в объект (например, каталог), которому сопоставлена метка Мп-1. Таким образом исклю-

чается возможность записи полученных конфиденциальных данных в объект с большей меткой (с меньшим уровнем конфиденциальности).

Что касается открытого почтового канала, то по нему любой пользователь может передавать только объекты, которым сопоставлена метка Мп (только эти объекты служба сможет читать из файловой системы). Любое получаемое по открытой почте сообщение любой пользователь сможет записать только в объект (например, каталог), которому сопоставлена метка Мп.

Таким образом, при вводе (выводе) объекта с «помеченного» виртуального канала, использующего помеченную сетевую службу, диспетчер доступа обеспечивает соответствие между меткой вводимого (выводимого) объекта и меткой виртуального канала (сетевой службы).

В общем случае модель мандатного управления доступом может быть расширена (по аналогии с дискреционным механизмом). Метки безопасности могут назначаться объекту доступа и обоим видам субъекта доступа – пользователю и процессу (приложению). При этом обработка запроса процесса может осуществляться как эксклюзивно (привилегированный процесс — см. рис. 11.10), так и вместе с правами пользователя. Т.е. в схеме управления доступом одновременно можно учитывать:

* категорию пользователя, то есть его допуск к информации, в соответствии с которым назначается метка безопасности;
* уровень защищенности информации, обеспечиваемый приложением (метка безопасности процесса).

14.7.3. Задачи мандатного управления доступом к виртуальным каналам связи

Рассмотрим мандатный механизм управления доступом к виртуальным каналам сети связи в общем случае. При этом под виртуальным каналом связи будем понимать канал связи между двумя оконечными устройствами сети (защищаемыми компьютерами). Канал этот характеризуется:

» уровнем конфиденциальности передаваемой по каналу информации;
» информационной технологией, использующей канал, то есть какой процесс (сетевая служба) его использует;
» характеристиками оконечных устройств (компьютеров), взаимодействующих по сети: IP-адрес, TCP-порт.

Назначение меток безопасности процессам рассматривается в предположении, что компьютерами, подключенными к виртуальному каналу связи, обрабатывается информация, различных уровней конфиденциальности. Таким образом, в схему управления доступом к файловым объектам может быть включены иерархические метки безопасности.

Рассмотрим задачи мандатного управления доступом к виртуальным каналам связи.

Задача 1. Подключение защищаемого компьютера к сети Internet

Итак, требуется подключить к сети Internet защищаемый компьютер, характеризуемый обработкой информации различных уровней конфиденциальности. При этом сеть Internet воспринимается как открытый виртуальный канал связи.

Пусть в системе обрабатывается следующая информация:

- секретная;
- конфиденциальная;
- служебная;
- открытая.

Соответственно введем 4 иерархические метки безопасности для данных — M1, M2, M3, M4.

Пусть доступ к информации имеют четыре пользователя, которым, в соответствии с их допуском к информации, также назначим метки безопасности — M1, M2, M3, M4.

Подключим компьютер к сети Internet. Поскольку при этом создается открытый виртуальный канал связи, то по нему следует разрешить передачу только открытой информации. Используя классический мандатный механизм, организовать доступ к открытому виртуальному каналу можно только следующим образом — разрешить запускать процесс (сетевую службу) только пользователю с меткой M4. Таким образом, взаимодействовать с сетью будет дозволено только пользователям, имеющим полный доступ к открытой информации и не имеющим доступа к информации иных уровней. Именно за счет этого исключается возможность попадания в открытый виртуальный канал конфиденциальных данных.

Недостаток данного решения заключается в том, что пользователи с метками M1...M3 отключены от сети Internet.

Теперь рассмотрим возможности, которые можно реализовать назначая метки процессам. При этом процесс с меткой будет обслуживаться эксклюзивно, то есть без учета прав доступа пользователей (меток безопасности).

Назначим метки безопасности пользователям и данным также, как представлено выше. Присвоим процессу (сетевой службе) Internet метку безопасности M4 (соответствующую метке открытых данных). Кроме того, обеспечим, чтобы все пользователи могли запустить данный процесс. Для этого назначим исполняемому файлу соответствующей сетевой службы метку M4.

Получаем:

* любой пользователь может запустить процесс доступа к сети Internet;
« запущенный процесс имеет права вне прав пользователей, т.е. этот процесс имеет полный доступ к открытым данным и не имеет доступа к иным данным на компьютере;
» любой пользователь имеет доступ к сети Internet, при этом выдаваться в сеть могут только открытые данные, т.е. процесс с меткой М4 может «читать» данные только с меткой М4. Открытые данные, получаемые из сети, могут записываться только в объекты, предназначенные для открытых данных, т.е. процесс с меткой М4 может «записывать» данные только в объекты с меткой М4;
* механизм управления доступом обеспечивает следующие разграничения для пользователей с учетом их меток безопасности. Пользователь с меткой М4 (допущенный к открытым данным) может выдавать в сеть и читать из сети информацию. Остальные пользователи (с меньшей меткой) могут только читать данные из сети, т.к. они имеют доступ к объекту, в который процесс с меткой М4 может записать данные только «на чтение». Поэтому для отправки данных пользователем с меньшей меткой должны быть реализованы организационные мероприятия.

На практике это может выглядеть следующим образом:

* все пользователи имеют возможность работы с Web-сервисами и иными службами сети Internet, предполагающими получение информации из сети. Информация сохраняется в объекте с меткой М4, из которого любой пользователь с меньшей меткой по правилам мандатного разграничения может его скопировать в собственный файловый объект, либо в его объект данную информацию может записать (дополнить) пользователь с меткой М4. С электронной почтой все пользователи с меткой меньше М4 могут работать только на прием сообщений из сети;
« только пользователь с меткой М4 получает право на отправку электронных почтовых сообщений, что определяется меткой виртуального канала;
* для отправки сообщений пользователями с меньшей, чем М4 меткой (что возможно только через пользователя с меткой М4, и не может быть осуществлено автоматически - - не позволит мандатный механизм управления доступа пользователей к файловым объектам) должны быть реализованы организационные мероприятия с привлечением ответственного лица, либо службы безопасности.

Достоинства данного подхода очевидны. Все пользователи могут работать с открытым виртуальным каналом связи, причем каждый в рамках своих полномочий, задаваемых иерархической меткой безопасности. При этом в полном объеме выполняются исходные требования мандатного механизма управления доступом к файловым объектам.

Задача 2. Подключение защищаемого компьютера к сети Intranet

Допустим теперь, что защищаемый компьютер требуется подключить к сети Intranet (корпоративный виртуальный канал связи). При этом, как и в предыдущей задаче, на компьютере осуществляется обработка информации различных уровней конфиденциальности.

Пусть в системе обрабатывается следующая информация:

* секретная;
* конфиденциальная;
* служебная;
* открытая.

Соответственно, введем 4 иерархические метки безопасности для данных- M1, M2, M3, M4.

Пусть доступ к информации имеют четыре пользователя, которым, в соответствии с их допуском к информации также назначим метки безопасности — M1, M2, M3, M4.

Подключим компьютер к сети Intranet. Т.к. при этом создается корпоративный виртуальный канал связи, то по нему следует разрешить передачу только служебной и открытой информации. При этом доступ к виртуальному каналу не должен иметь пользователь, допущенный к открытой информации, т.е. пользователь с меткой M4.

Примечание

Для создания служебного виртуального канала может применяться фильтрация доступа к сети (по IP адресам и TCP портам). Соответствующий процесс должен обеспечивать защиту передаваемой по каналу связи информации криптографическими методами.

Используя классический мандатный механизм управления доступом, организовать доступ к служебному виртуальному каналу можно только следующим образом — разрешить запускать процесс (сетевую службу) только пользователю с меткой M3. Таким образом, взаимодействовать с сетью будет разрешено только пользователю, имеющему полный доступ к служебной информации и не имеющему доступа к информации иных уровней. В противном случае в открытый виртуальный канал могут попасть данные, не предназначенные для передачи по нему.

Недостаток данного решения заключается в том, что пользователи с метками M1, M2, M4 будут отключены от сети Intranet. К тому же по служебному каналу не может передаваться открытая информация.

Теперь рассмотрим возможности, которые можно реализовать назначая метки процессам. При этом процесс с меткой будет обслуживаться эксклюзивно, то есть без учета прав доступа пользователей (меток безопасности).

Назначим метки безопасности пользователям и данным так же, как представлено выше. Присвоим процессу (сетевой службе) Intranet метку безопасности М3, соответствующую метке служебных данных. Далее назначим исполняемому файлу соответствующей сетевой службы метку М3. Тем самым мы обеспечим, чтобы пользователи M1...M3 имели возможность запустить данный процесс, а пользователь М4 нет.

Получаем:

- запустить процесс доступа к сети Intranet может любой пользователь, имеющий доступ не ниже, чем к служебной информации — обладающие метками M1...M3;
- запущенный процесс имеет права вне прав пользователей, т.е. этот процесс имеет полный доступ к служебным данным, доступ «на чтение» к открытым данным и не имеет доступа к иным данным на компьютере;
- любой пользователь с уровнем доступа не ниже доступа к служебным данным имеет доступ к сети Intranet. При этом в сеть могут выдаваться как служебные, так и открытые данные (процесс с меткой М3 может «читать» данные с метками М3 и М4). Любые данные, получаемые из сети, могут записываться только в объекты, предназначенные для служебных данных (процесс с меткой М3 может «записывать» данные только в объекты с меткой М3), что предотвращает доступ к этим данным пользователя с меткой М4;
- механизм управления доступом обеспечивает следующие разграничения для пользователей с учетом их меток безопасности. Пользователь с меткой М4 (допущенный к открытым данным) не может выдавать в сеть и читать из сети информацию. Пользователь с меткой М3 (допущенный к служебным данным) может выдавать в сеть и читать из сети информацию. Остальные пользователи (с меньшей меткой) могут только читать данные из сети, т.к. они имеют доступ к объекту, в который процесс с меткой М3 может записать данные, только «на чтение». Для отправки данных пользователем с меньшей меткой должны быть реализованы организационные мероприятия.

Пример практического использования:

- все пользователи, кроме пользователя с меткой М4, имеют возможность работы с Web-сервисами и иными службами сети Intranet, предполагающими получение информации из сети. Информация сохраняется в объекта (каталоге) с меткой М3, из которого любой пользователь с меньшей меткой по правилам мандатного разграничения может его скопировать в собственный файловый объект, либо в его объект данную информацию может записать (дополнить) пользователь с меткой М3. С электронной почтой все пользователи с меткой больше М3 могут работать только на прием сообщений из сети;

» только пользователь с меткой М3 получает право на отправку электронных почтовых сообщений, что определяется меткой виртуального канала;

* для отправки сообщений пользователями с меньшей, чем М3, меткой (что возможно только через пользователя с меткой М3, и не может быть осуществлено автоматически — не позволит мандатный механизм управления доступа пользователей к файловым объектам) должны быть реализованы организационные мероприятия с привлечением ответственного лица либо службы безопасности.

Таким образом все пользователи, имеющие допуск не ниже, чем допуск к служебной информации, могут работать со служебным виртуальным каналом связи, каждый в рамках своих полномочий, задаваемых иерархической меткой безопасности. При этом в полном объеме выполняются исходные требования мандатного механизма управления доступом к файловым объектам.

Задача 3 (общий случай). Подключение защищаемого компьютера к сети Internet и Intranet

Пусть требуется подключить к сети Internet (открытый виртуальный канал связи) и к сети Intranet (корпоративный виртуальный канал связи) защищаемый компьютер, характеризуемый обработкой информации различных уровней конфиденциальности. Таким образом, задача 3 является комбинацией задач 1 и 2).

При тех же предположениях, что и для предыдущих задач, включим в систему два процесса: один для передачи по сети Intranet служебной информации -- процесс 1 с меткой М3 (метка исполняемого файла процесса 1 — М3), ограничим доступ данного процесса к сетевым ресурсам Intranet по IP адресам и TCP портам; и процесс 2 для передачи по сети Internet открытой информации — с меткой М4 (метка исполняемого файла процесса 1 -- М4).

Получаем совокупность свойств, представленную при решении задач 1 и 2.

Достоинства:

* один компьютер, в котором обрабатывается информация различных уровней конфиденциальности, одновременно может быть подключен к виртуальным каналам, предназначенным для передачи информации различных уровней конфиденциальности (в частности, к сети Internet и Intranet);
* за счет назначения меток безопасности процессам (сетевым службам), характеризующим виртуальный канал, в системе может быть реализовано мандатное управления доступом к виртуальным каналам связи;
* все пользователи могут работать с открытым виртуальным каналом связи, каждый в рамках своих полномочий, задаваемых иерархичес-

кой меткой безопасности. При этом в полном объеме выполняются исходные требования мандатного механизма управления доступом к файловым объектам;

» все пользователи, имеющие допуск не ниже, чем допуск к служебной информации, могут работать со служебным виртуальным каналом связи, каждый в рамках своих полномочий, задаваемых иерархической меткой безопасности. При этом в полном объеме выполняются исходные требования мандатного механизма управления доступом к файловым объектам.

Соответственно, если требуется реализовать мандатное разграничение доступа к виртуальному каналу связи, предназначенному для передачи конфиденциальной информации, то необходимо осуществить следующие настройки мандатного механизма управления доступом процессов к виртуальному каналу связи:

1. Назначить метку безопасности процессу. Эта метка задается меткой безопасности той информации, для передачи которой он предназначен (такая же).
2. Назначить метку безопасности исполняемому файлу процесса для его запуска пользователем. Эта метка задается меткой безопасности, назначенной процессу (такая же).
3. При необходимости, следует разграничить права доступа процесса к сетевым ресурсам (в рамках реализации виртуального канала) – по IP адресам и TCP портам.

14.7.4. Требование к изоляции процессов

Следует отметить, что в случае назначения метки безопасности процессам возникает проблема, которая отсутствует для ОС семейства Windows (в силу своих особенностей). Связана эта проблема с тем, что в системе одновременно могут быть запущены несколько процессов *с* различными правами доступа к ресурсам (с различными метками безопасности). Это значит, что в системе могут быть одновременно запущены несколько процессов, которые могут осуществлять запись информации в объекты различных уровней иерархии. Поэтому назначение меток безопасности в общем случае возможно только при реализации в системе требования к изоляции модулей [1], т.е. требования, состоящего в том, что процессы с различными метками безопасности должны быть изолированы друг от друга — между ними не может осуществляться копирование данных (например, через буфер обмена).

В противном случае невозможна корректная реализация мандатного управления доступом в принципе, т.к. посредством взаимодействия процессов (копирования между ними данных) данные из объекта более высокого уровня иерархии могут быть скопированы в объект с меньшим уровнем иерархии.

14.7.5. Сводные положения по назначению и использованию меток процессов

Ранее мы показали возможность назначения метки безопасности для субъекта доступа «ПРОЦЕСС», что кардинально меняет возможности применения мандатного механизма управления доступом. Здесь сформулируем некоторые общие положения по назначению и использованию меток безопасности для субъекта доступа «ПРОЦЕСС».

1. Метка безопасности может быть присвоена приложению (процессу). В качестве критерия назначения метки может рассматриваться уровень защиты обрабатываемой информации, обеспечиваемой самим приложением (встроенными в него механизмами защиты). В этом случае могут быть сопоставлены метка безопасности объекта (категории информации) и метка безопасности приложения (процесса). Делается это в соответствии с требованиями к механизмам защиты по обработке информации соответствующей категории.
2. При назначении меток безопасности одновременно пользователю и процессу в схеме управления доступом одновременно можно учитывать и **категорию пользователя** — его допуск к информации, в соответствии с которым назначается метка безопасности, и **уровень защищенности информации,** обеспечиваемый приложением (метка безопасности процесса). Данный подход позволяет принципиально расширять возможности мандатного управления доступом к ресурсам.
3. Назначение меток безопасности в общем случае возможно только при реализации в системе требования к изоляции модулей. В рамках этого требования процессы с различными метками безопасности должны быть изолированы друг от друга - - между ними не должно осуществляться копирование данных (например, через буфер обмена). В противном случае невозможна корректная реализация мандатного управления доступом в принципе, т.к. посредством взаимодействия процессов (копирования между ними данных) данные из объекта более высокого уровня конфиденциальности могут быть скопированы в объект с меньшим уровнем конфиденциальности.

14.8. Разграничение доступа к объекту «ПРОЦЕСС» (исполняемым файлам)

Особое место среди файловых объектов занимают исполняемые файлы (программы), доступ субъектов к которым также может разграничиваться. Именно исполняемые файлы изначально порождают процессы. Поэтому в качестве разграничения доступа к процессам (как к объектам доступа) прежде всего следует разграничить доступ к исполняемым файлам.

Исполняемые файлы в общем случае имеют признаки, по которым их всегда можно отличить от файлов данных. Например, для ОС семейства Windows исполняемые файлы по внешним признакам отличаются расширениями (например, .com, .exe и т.д.). В ОС UNIX отличительным признаком исполняемых файлов может служить их атрибут исполнения. Однако интерпретаторы команд (например, встроенный в ОС процесс CMD.exe) позволяют запускать файлы с иным расширением. Поэтому в общем случае необходимо уметь выделять исполняемые файлы, исходя из их структуры, то есть определять, исходя из внутренних признаков.

Примечание

Здесь будет идти речь об исполняемых файлах прикладных программ. Вопросы разграничения доступа к системным исполняемым файлам (к привилегированным процессам) рассмотрены выше.

Управление доступом к исполняемым файлам реализует разграничение прав доступа субъектов на запуск прикладных программ.

Для разграничения доступа субъектов *к* исполняемым файлам введем право доступа «И» -- «исполнение» — чтение исполняемого файла (запуск программы), тогда множество прав доступа в общем случае имеет вид {Зп, Чт, Д, И}.

14.8.1. Каноническая модель управления доступом к исполняемым файлам

Используя обозначения, введенные *ранее,* введем следующее обозначение: пусть S = {0,И} — множество прав доступа, где «О» обозначает отсутствие доступа субъекта к объекту, «И» — доступ к исполняемому файлу (разрешение чтения исполняемого файла). Тогда, по аналогии со сказанным ранее, каноническую модель управления доступом к исполняемым файлам можно представить матрицей доступа D, имеющей следующий вид:

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} И & 0 & \dots & 0 & 0 \\ 0 & И & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & И & 0 \\ 0 & 0 & \dots & 0 & И \end{bmatrix} \end{array}$$

Модель управления доступом формально может быть описана следующим образом: элемент (Dij) матрицы Dij = И, если i = j, иначе Dij = 0.

Под канонической моделью управления доступом к исполняемым файлам для линейно упорядоченных множеств субъектов доступа (групп субъектов) и объектов доступа (групп объектов) понимается модель, описываемая матрицей доступа, элементы главной диагонали которой «И» задают доступ к исполняемому файлу (разрешение на запуск программы), остальные элементы «О» задают запрет доступа субъектов к исполняемым файлам.

Утверждение

Диспетчер доступа реализует механизм управления доступом к исполняемым файлам корректно только в том случае, если его настройками (заданием учетных записей субъектов и объектов доступа и правил разграничения доступа) можно реализовать каноническую модель управления доступом.

Доказывается утверждение от обратного. Если каноническую модель управления доступом реализовать невозможно — присутствуют элементы «И» вне главной диагонали матрицы доступа, то в системе присутствует по крайней мере один объект — программа, доступ к запуску которой невозможно разграничить в полном объеме (объект включается одновременно в несколько групп объектов, априори характеризуемых различными правами доступа к ним).

По аналогии со сказанным ранее, в рассматриваемом случае в каноническую модель управления доступом также должны быть включены каналы взаимодействия субъектов доступа. В противном случае, субъекты, имеющие возможность обрабатывать информацию только различными приложениями, не смогут обменяться информацией.

Таким образом, **включение канала взаимодействия субъектов в матрицу доступа означает разрешение запуска субъектами одного и того же приложения.**

В матрице доступа D это означает включение группы (групп) объектов, для которых несколько субъектов будут иметь доступ. Пример такой матрицы приведен ниже. При этом в нее введена группа объектов (программ) Ok+1, доступ к которым (запуск которых) разрешен всем субъектам.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \\ O_{k+1} \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} \text{Я} & 0 & \dots & 0 & 0 \\ 0 & \text{Я} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \text{Я} & 0 \\ 0 & 0 & \dots & 0 & \text{Я} \\ \text{Я} & \text{Я} & \text{Я} & \text{Я} & \text{Я} \end{bmatrix} \end{array}$$

По аналогии управления доступом к файлам данных, здесь может быть реализован дискреционный и мандатный механизмы управления. Дискреционный механизм предполагает реализацию доступа диспетчером на основе заданной матрицы доступа D, а мандатный — на основе меток безопасности.

Метки безопасности являются элементами линейно упорядоченного множества M = {M1,...,Mk} и задаются субъектам и объектам доступа. Метки безопасности назначаются субъектам и объектам (группам субъектов и объектов) и служат для формализованного представления их уровня полномочий.

Как и ранее, будем считать, что чем выше полномочия субъекта и объекта, тем меньшее значение метки безопасности Mi, i = l,...,k им присваивается, т.е.: Ml < M2 < M3<...<Mk. При этом в линейно полномочно упорядоченных множествах C = {C1,...,Ck} и O = {O1,...,Ok} субъекты и объекты располагаются в порядке уменьшения полномочий (уровня безопасности).

14.8.2. Каноническая полномочная модель управления доступом к исполняемым файлам

Каноническая полномочная модель управления доступом к исполняемым файлам может быть представлена следующей матрицей доступа D.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} И & И & \dots & И & И \\ 0 & И & \dots & И & И \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & И & И \\ 0 & 0 & \dots & 0 & И \end{bmatrix} \end{array},$$

Модель управления доступом формально может быть описана следующим образом: элемент матрицы Dij = И, если i <= j, иначе Dij = 0.

Под канонической моделью управления доступом к исполняемым файлам с виртуальными каналами взаимодействия субъектов доступа для линейно полномочно упорядоченных множеств субъектов (групп субъектов) и объектов (групп объектов) доступа понимается модель, описываемая матрицей доступа, элементы главной диагонали которой и элементы, расположенные выше главной диагонали, «И» — задают право доступа «Исполнение», остальные элементы матрицы «О» - - запрет запуска исполняемого файла.

Рассмотрим правила разграничения доступа для модели управления доступом к исполняемым файлам. При этом воспользуемся введенными ранее обозначениями:

- Ms — метка безопасности субъекта (группы субъектов) доступа;
- Mo — метка безопасности объекта (группы объектов) доступа;
- метка безопасности Mi с порядковым номером i устанавливается для субъекта доступа Ci с порядковым номером i и для объекта доступа Oi с порядковым номером i.

Правила разграничения доступа для модели управления доступом к исполняемым файлам:

1. субъект C имеет доступ к объекту O в режиме «Исполнение» в случае, если выполняется условие: Mc => Mo;
2. субъект C не имеет доступ к объекту O в режиме «Исполнение» в случае, если выполняется условие: Mc < Mo.

Выводы:

1. В схеме управления доступом в качестве отдельного объекта доступа следует рассматривать исполняемый файл (процесс). В этом случае должно рассматриваться дополнительное право доступа «исполнение». Использование данного объекта позволяет разграничивать права доступа для субъектов (в общем случае для пользователей и процессов) по запуску процессов.
2. Условие корректности разграничений для объекта доступа «исполняемый файл» задаются точно так же (аналогичные канонические модели), как и для файловых объектов данных, с учетом права доступа «исполнение».

14.9. Механизм обеспечения замкнутости программной среды

14.9.1. Механизм обеспечения замкнутости программной среды и его роль в системе защиты

Под обеспечением замкнутости программной среды понимается локализация прикладных программ для пользователей. Осуществляется это механизмом управления доступом к исполняемым файлам.

Утверждение

Механизм обеспечения замкнутости программной среды реализован корректно только в том случае, если выполняются требования к полноте управления доступом к исполняемым файлам.

Проиллюстрируем сказанное примером. Например, для ОС Windows NT/2000 могут устанавливаться разграничения на исполнение файлов (запуск программ) с жесткого диска. Однако данные ОС не позволяют управлять

запуском программ с внешних устройств ввода. Поэтому корректно обеспечить замкнутость программной среды, с целью локализации прикладных программ для пользователя, возможно лишь при отключении внешних устройств ввода.

Таким образом, разграничение доступа на запуск исполняемого файла и обеспечение замкнутости программной среды не одно и то же. Замкнутость программной среды реализуется посредством механизма управления доступом на запуск к исполняемым файлом, но можно говорить о его реализации в системе лишь при выполнении требований к полноте подобных разграничений.

Как отмечалось ранее, несмотря на то, что в современных универсальных ОС присутствует механизм управления доступом к запуску процессов, механизм обеспечения замкнутости программной среды в них не реализован. Это объясняется невыполнением требований к полноте разграничений – для ОС Windows исполняемый файл может быть запущен с внешнего устройства ввода, например, посредством дискеты. В ОС семейства UNIX невозможно установить атрибут «исполнение» на каталог (под этим атрибутом для каталога подразумевается «обзор»), т.е. невозможно запретить запуск из каталога. При этом (этот вопрос ранее рассматривался) пользователь сам может присвоить атрибут создаваемому им файлу, т.е. пользователь может записать в свой каталог исполняемый файл, установить на него для себя право исполнения, затем запустить данный файл.

Обеспечение замкнутости программной среды -- это важнейший механизм противодействия организации пользователем скрытых каналов доступа к ресурсам защищаемого объекта. Целью применения механизма является предоставление пользователям возможности запускать только санкционированные программы из заданных для них списков. При этом предотвращается возможность запуска пользователем собственной программы, которая может содержать скрытый канал доступа к ресурсам. Без реализации данного механизма в системе защиты вообще не приходится говорить о какой-либо защищенности объекта [11, 12, 19].

В общем случае механизм обеспечения замкнутости программной среды предназначен для локализации программного обеспечения на компьютере (системного, функционального и прикладного ПО) и обеспечение невозможности запуска пользователем несанкционированного процесса (программы). Благодаря этому обеспечивается то, что пользователь может работать на компьютере только жестко в рамках своих функциональных обязанностей и инструкций, т.е. в границах списка санкционированных действий. К этой же задаче относится защита от проникновения (прежде всего из сети) и запуска на компьютере деструктивных программ (троянов, программ sniffer клавиатуры и канала, программ взлома или подбора паролей, программ перепрограммирования BIOS, инструментальных средств и т.д.).

Если вернуться к рассмотрению существующей статистики угроз (приведенной в п. 2.2), то можем сделать вывод, что подавляющая их часть требовала от пользователя запуска программного средства, реализующего данные угрозы. Таким образом, путем предотвращения возможности запуска пользователем собственных программ злоумышленник лишается собственно инструментария взлома. При этом неважно, о какой угрозе, о какой цели и способе атаки идет речь. Поэтому данный механизм позиционируется нами как основной механизм защиты, позволяющий противодействовать скрытым угрозам НСД к информации.

Ранее нами отмечалось, что защита информации — это комплексная задача, решение которой достигается реализацией совокупности механизмов защиты и что невозможно рассматривать механизм защиты в отдельности, т.к. найдутся угрозы данному механизму, которые должны предотвращаться другими механизмами. Рассмотрим иную сторону этой проблемы, когда использование в системе защиты одного механизма может принципиально изменить требование к другим механизмам.

С учетом применения механизма обеспечения замкнутости программной среды могут существенно снижаться требования к реализации других механизмов защиты. Рассмотрим пример.

Ввиду того, что при удалении информации с диска осуществляется лишь переразметка диска, а собственно информация на нем остается, появляется скрытый канал в виде возможности несанкционированного доступа к остаточной информации на диске. Данный канал может быть устранен, если реализовать механизм гарантированного удаления информации, который будет заключаться:

1. В перехвате обращения приложения к ядру ОС «на удаление» объекта.
2. В записи в объект маскирующей информации (осуществляется N — кратная запись «О» и «1»).
3. В передаче ядру ОС запрос «на удаление» объекта.

При этом остаточной информации на диске не остается. Реализация данного механизма защиты связана с существенными потерями производительности защищаемого объекта. Остаточная информация также может оставаться в оперативной памяти после завершения процесса (если ОС или приложение не осуществляет ее очистку).

Однако стандартным приложением прочитать остаточную информацию не представляется возможным, для этого нужны специальные программы, запуск которых на компьютере предотвращается механизмом обеспечения замкнутости программной среды. Таким образом, прочитать остаточную информацию без запуска несанкционированнорго процесса на защищенном компьютере становится невозможно. Остается только одна возможность ее прочтения — это удалить с компьютера жесткий диск и считать информацию с него на другом компьютере. Но если такое воз-

можно, то оберегать уже следует не остаточную, а актуальную информацию на диске, для чего, в рассматриваемых предположениях, должны применяться криптографические методы защиты информации на диске. Сказанное в полной мере относится и к защите оперативной памяти. Вместе с тем, учитывая, что критичной ситуацией здесь является запуск несанкционированного процесса пользователем, механизм очистки памяти, на наш взгляд, следует запускать в качестве реакции на обнаружение запущенного несанкционированного процесса. Механизм, реализующий данную возможность, рассматривается в следующей главе.

Могут быть приведены и иные примеры, когда использование в системе механизма обеспечения замкнутости программной среды либо делает необязательными целый ряд других механизмов защиты, либо существенно изменяются требования к решаемым ими задачам.

Таким образом, можно сделать следующий вывод: **механизм обеспечения замкнутости программной среды -- важнейший механизм, который обязательно должен присутствовать в системе защиты, т.к. этот механизм можно позиционировать, как основной механизм противодействия скрытым угрозам (возможности реализации неизвестной угрозы, присутсвующей в несанкционированной пользовательской программе).**

Ввиду того, что данный механизм не имеет корректной реализации в современных ОС, его следует реализовать добавочными средствами защиты информации. В противном случае невозможно говорить о возможности противодействия скрытым угрозам в принципе.

Механизм обеспечения замкнутости программной среды может быть реализован двумя способами:

- в виде задания списков исполняемых файлов;
- в виде задания каталогов исполняемых файлов.

Рассмотрению и анализу обоих этих подходов посвящен следующий раздел книги.

14.9.2. Реализация механизма обеспечения замкнутости программной среды

Посредством задания списков исполняемых файлов

Данный механизм состоит в задании для каждого пользователя списка процессов (исполняемых файлов), которые ему разрешено запускать. Корректность функционирования механизма связана с предотвращением явных и скрытых каналов модификации списка санкционированных исполняемых файлов, а также запуска исполняемых файлов, не вошедших в список.

Примечание

При данном походе объект доступа *в* матрице доступа к исполняемым файлам представляет собой список исполняемых файлов.

Требования к корректности функционирования механизма обеспечения замкнутости программной среды таковы [30]:

- исполняемый файл должен быть задан с указанием его полного пути, что предотвращает возможность запуска несанкционированого процесса с таким же именем из другого места. При этом пользователю должен быть запрещен запуск программ с внешних устройств ввода (локальных и общих — разделяемых в сети), а также из общих папок (разделяемых в сети файловых объектов), т.к. иначе невозможно противодействовать запуску несанкционированной программы пользователя с полнопутевым именем легальной программы;
- пользователю должен быть задан список санкционированных для его запуска исполняемых файлов, к которым ему должен быть разрешен доступ «на исполнение». Ко всем другим файловым объектам пользователю должен быть запрещен доступ «на исполнение»;
- пользователю должен быть запрещен доступ «на запись» и «модификацию» к исполняемым файлам. В противном случае он сможет подменить санкционированный файл другого пользователя. Особенно это критично по отношению к системным процессам, поскольку нельзя допускать, чтобы пользователь мог подменить его собственным несанкционированным процессом. Из сказанного следует, что механизм обеспечения замкнутости программной среды не может быть корректно реализован без запрета доступа «на запись» и «на модификацию» к системному диску и без возможности управлять доступом на исполнение программ с устройств ввода, с разделяемых в сети ресурсов (устройств и файловых объектов).

Рассмотренный подход интуитивно понятен, но имеет существенные недостатки. Главным из них является необходимость перечислять в списках все процессы, разрешенные к запуску пользователем, в том числе и процессы, порождаемые уже разрешенными процессами. Из-за этого списки разрешенных процессов становятся очень громоздкими, существенно снижается производительность системы за счет больших затрат времени на анализ таких списков при каждом обращении к объекту файловой системы

Сильно усложняется администрирование подобной системы защиты, так как администратору приходится составлять такие списки для каждого пользователя, а при добавлении новых программ изменять списки у каждого пользователя. Соответственно, при удалении некоторой программы приходится удалять из множества списков не только саму программу, но и все процессы, порождаемые удаляемой программой. Кроме того, возникает проблема, связанная с необходимостью введения в список разре-

щенных некоторому пользователю процессов отладочных и иных программ, которые необходимо запускать эксплуатационным службам при работе этого пользователя в системе. С другой стороны, необходимо исключить возможность прямого запуска этих отладочных профамм несанкционированным пользователем.

Посредством каталогов исполняемых файлов

Обеспечить замкнутость программной среды можно не непосредственно заданием списков разрешенных к запуску процессов, а областью дискового пространства (каталогом), откуда пользователю можно запускать процессы. То есть пользователю можно разрешать/запрещать не что запускать, а откуда запускать. Другими словами, для каждого пользователя может быть выделен каталог, только из которого ему будет дозволено запускать программы. На этот каталог должно быть установлено право «Исполнение» для пользователя. Причем в данный каталог ему должен быть запрещен доступ «на запись» и «на модификацию», с целью предотвращения возможности внесения в данный каталог других, не разрешенных к запуску программ, а также с целью предотвращения возможности модификации расположенных в каталоге программ.

При этом из всех остальных каталогов (а также с устройств ввода), кроме системного, пользователю должен быть запрещен запуск программ. Кроме того, пользователю должен быть запрещен доступ «на запись» и «на модификацию» в системный каталог. Это необходимо с целью предотвращения возможности модификации системного каталога — занесения в него несанкционированных процессов).

Список разрешенных к запуску профамм (не процессов) определяется набором профамм, инсталлированных администратором безопасности в каталог, откуда пользователю разрешен их запуск [11, 12, 19].

Можем сформулировать следующие требования к корректности функционирования рассматриваемого механизма:

« для пользователя должен быть задан каталог, откуда ему разрешено запускать программы. На доступ к этому каталогу пользователю должно быть установлено право «Исполнение», а доступ «на запись» и «на модификацию» должен быть запрещен;

* в этот выделенный каталог администратором должны быть инсталлированы профаммы, разрешенные пользователю для запуска;

» ко всем остальным каталогам, а также к усфойствам (дисководу, CD-ROM и т.д.), разделяемым сетевым ресурсам пользователю должен быть запрещен доступ «Исполнение»;

* к системному диску, а также к каталогам с исполняемыми файлами остальных пользователей, пользователю должен быть запрещен доступ «на запись» и «на модификацию».

Достоинством рассмотренного здесь механизма является серьезное упрощение его администрирования, т.к. исключается необходимость перечисления всех процессов, которые запускает программа, при разрешении ее запуска пользователем (при установке и удалении программы из списка разрешенных к запуску). Кроме того, в предыдущей реализации механизма обеспечения замкнутости программной среды необходимость подобного перечисления могла привести к тупикам. Например, программе для работы может потребоваться некоторый процесс, а разрешать запускать этот процесс пользователю нельзя.

На рис. 14.3 приведена схема обработки запроса доступа к объекту, реализуемая диспетчером доступа в КСЗИ «Панцирь» с целью обеспечения замкнутости программной среды [31].

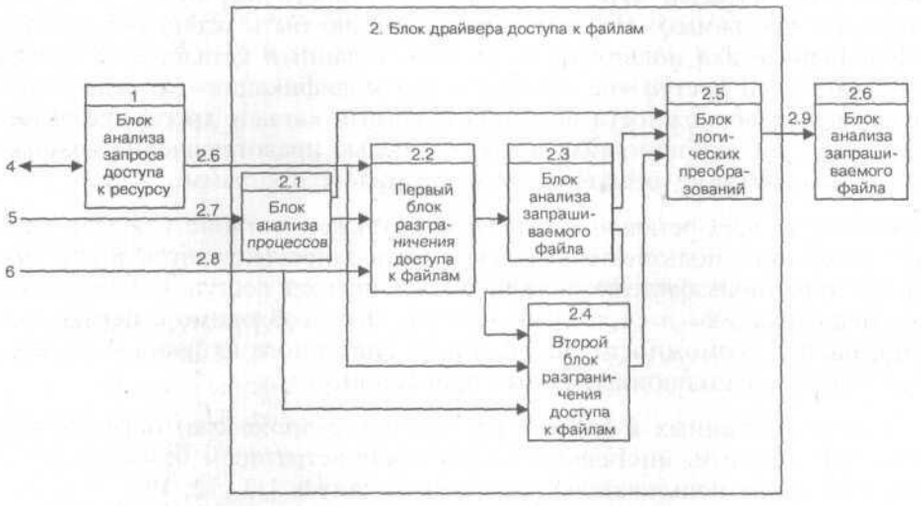


Рис. 14.3. *Схема обработки запроса доступа к объекту при реализации механизма обеспечения замкнутости программной среды*

Работает система обеспечения замкнутости программной среды следующим образом. С входа/выхода 4 производится авторизация пользователя (идентификация и аутентификация) при входе в систему. В случае, если ему разрешен вход в систему, то его имя через вход 2.6 поступает на вход блоков 2.2. и 2.4. В противном случае блоком 1 со входа/выхода 4 пользователь извещается об отсутствии у него прав на доступ в систему.

После получения доступа в систему, при обращении к файлу запрос на доступ к файлу со входа 5 поступает через вход 2.7 в блок 2.1. Запрос содержит: имя процесса, обращающегося к файлу, полный путь расположения файла в блоке 3, команду над файлом (чтение, запись и т.д.). Данный запрос анализируется блоком 2.1, который выявляет, является ли этот запрос от системного или пользовательского процесса.

Если запрос от системного процесса, то он транслируется с первого выхода блока 2.1 в блок 2.5, которым далее выдается в блок 3 (запрос поступает в обход блока 2.2). Если выявлен запрос от пользовательского процесса, то он транслируется в блок 2.2 (со второго выхода блока 2.1), где для каждого пользователя (по его имени) содержится матрица его прав доступа (полные пути к каталогам и файлам, к которым пользователь может обращаться, и команды (например, только чтение), которые пользователь может производить над файлами).

Матрица прав доступа задается администратором безопасности (после его авторизации в блоке 1) со входа 6 через вход 2.8. В случае, если запрос пользователя удовлетворяет разфаничениям, хранящимся для него в блоке 2.2, то блоком 2.2 этот запрос транслируется в блок 2.3. Блок 2.3 анализирует расширение файла, к которому обращается пользователь, с целью определения, является ли этот файл исполняемым (программой), если имеет расширения, например, .com, .exe, либо данными, например, расширения .doc, .rtf, .txt.

Если пользователь обращается к данным, то его запрос блоком 2.3 транслируется в блок 3. Если запрос не удовлетворяет требованиям разграничений в блоке 2.2, запрос блоком 2.2 не пропускается — игнорируется системой. Если блоком 2.3 выявляется, что пользователь обращается в программе в рамках общих разграничений к каталогам и файлам в блоке 2.2, то запрос им транслируется в блок 2.4 (в блок 2.5 не поступает).

В блоке 2.4 каждого пользователя (по его имени) содержится матрица его прав доступа к запускаемым программам (полные пути к каталогам и файлам к которым пользователь может обращаться и команды (например, только чтение), которые пользователь может производить над исполняемыми файлами). Например, здесь задаются разграничения доступа к каталогам, из которых пользователь может запускать программы (обращаться «на чтение» к исполняемым файлам). В частности, пользователю может быть задан режим запуска профамм только из системного диска, тогда все запросы, поступающие в блок 2.4 будут им игнорированы. Может быть запрещена запись программ (исполняемых файлов) в пользовательские каталоги и т.д. Матрица прав доступа к запуску профамм также задается администратором безопасности (после его авторизации в блоке 1) со входа 6 через вход 2.8.

Если запрос пользователя удовлетворяет разграничениям в блоке 2.4, то его запрос через блок 2.5 в блок 3. Если запрос не удовлетворяет требованиям разграничений в блоке 2.4, запрос блоком 2.4 не пропускается — игнорируется системой.

При всех достоинствах данного подхода он не может быть использован в том случае, если приложение при инсталляции должно размещать исполняемые и конфигурационные файлы (куда необходимо разрешить запись) в одном каталоге. В этом случае невозможно на данный каталог

установить атрибут «исполнение», не установив атрибута «запись», что противоречит самой концепции реализации механизма защиты.

14.9.3. Расширение возможностей механизма обеспечения замкнутости программной среды

Расширение возможностей механизма защиты обеспечения замкнутости программной среды состоит в том, чтобы и в качестве субъекта доступа, и в качестве объекта доступа рассматривать «ПРОЦЕСС». В этом случае можно разграничивать права доступа для процессов на запуск процессов, т.е. можно обеспечивать замкнутость программной среды не на уровне списков санкционированных процессов (разрешения запуска пользователем отдельных программ), а уже на уровне последовательностей запуска процессов (технологий обработки данных). Другими словами, можно задавать последовательности обработки — каким процессом какой процесс может быть запущен.

14.10. Управление доступом к каталогам, не разделяемым системой и приложениями

14.10.1. Наличие в системе каталогов, не разделяемых между пользователями, и связанные с этим сложности

При работе ОС семейства Windows (прежде всего, Windows 9x/Me) и приложений существует ряд каталогов, в общем случае не разделяемых системой или приложениями между пользователями. К таким каталогам можно отнести «корзину» (каталог RECYCLED), переменные окружения (каталоги TEMP, TMP), каталог «Мои документы», различные каталоги приложений для хранения временной информации и др. При этом для некоторых приложений, предполагающих автосохранение информации, нельзя запретить доступ какому-либо пользователю в данные каталоги. Причем информация в них записывается автоматически приложением.

Таким образом, существуют каталоги, для которых невозможно разграничить доступ пользователям. При этом либо пользователи, которым запрещен доступ к рассматриваемым каталогам, не смогут работать с приложением, либо им не сможет предоставляться необходимый сервис, например, работа с «Корзиной».

Наличие подобных объектов в системе не позволяет говорить о корректности решения задачи управления доступа в целом. В результате они должны быть реализованы добавочными средствами управления.

Если обозначить группу объектов файловой системы (каталогов), не разделяемых системой между пользователями, как Он, то матрица доступа D принимает вид, представленный ниже.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_n \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & пд & \dots & ПД & ПД \\ пд & Зп/Чт & \dots & ПД & ПД \\ \dots & \dots & \dots & \dots & \dots \\ Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт & Зп/Чт \\ \dots & \dots & \dots & \dots & \dots \\ ПД & ПД & \dots & Зп/Чт & ПД \\ пд & ПД & \dots & ПД & Зп/Чт \end{bmatrix} \end{array}$$

В матрице D использовано обозначение «ПД» -- права доступа, назначаемые в зависимости от реализуемого канала взаимодействия субъектов доступа. Как следует из представленной матрицы, корректно реализовать управление доступом (особенно мандатный механизм) в данном случае невозможно -- все пользователи имеют полный доступ к группе каталогов Он.

14.10.2. Технология переадресации запросов

Возможный подход к решению рассматриваемой проблемы заключается в технологии переадресации запросов доступа к объектам файловой системы (каталогам), не разделяемым системой между пользователями. При этом предлагается следующее решение. Для каждого пользователя средствами диспетчера доступа для неразделяемого объекта реализуется соответствующий собственный объект. Например, для каталога «Корзина» заводятся каталоги «Корзина 1» для первого пользователя, «Корзина 2» для второго пользователя и т.д.

При записи информации системой или приложением в неразделяемый каталог (соответственно, чтении из каталога) диспетчер доступа перенаправляет информацию в (из) соответствующий каталог текущего пользователя. Например, если текущим пользователем является первый пользователь, то при сохранении информации в каталог «Корзина» данная информация будет перенаправлена диспетчером доступа и сохранена в каталоге «Корзина 1».

При этом механизм перенаправления запросов к неразделяемым системой объектам должен обрабатывать запрос перед механизмом управления доступом. Средствами механизма управления доступом к файловой системе разграничиваются права доступа к каталогам, в которые перенаправляется информация. Например, доступ к каталогу «Корзина 1» следует разрешить только первому пользователю, а остальным — запретить.

Таким образом данный механизм позволяет обеспечить отсутствие общих ресурсов файловой системы для пользователей.

В результате непосредственно в исходных неразделяемых системой каталогах TMP, TEMP, «Корзина», «Мои документы» и т.д., запрос доступа к которым переадресуется, информация сохраняться не будет, т.е. данные каталоги становятся в системе виртуальными и к ним нет необходимости разграничивать доступ.

Матрица доступа D при реализации данной технологии переадресации запросов, примет следующий вид.

$$D = \begin{array}{c} \\ O_1 \\ O_2 \\ \dots \\ O_н \\ O_{н1} \\ O_{н2} \\ \dots \\ O_{нk} \\ \dots \\ O_{k-1} \\ O_k \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \dots & C_{k-1} & C_k \end{array} \\ \begin{bmatrix} Зп/Чт & пд & \dots & пд & пд \\ ПД & Зп/Чт & \dots & пд & пд \\ \dots & \dots & \dots & \dots & \dots \\ - & - & - & - & - \\ Зп/Чт & 0 & \dots & 0 & 0 \\ 0 & Зп/Чт & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & Зп/Чт \\ \dots & \dots & \dots & \dots & \dots \\ ПД & ПД & \dots & Зп/Чт & ПД \\ ПД & пд & \dots & ПД & Зп/Чт \end{bmatrix} \end{array}$$

В данной матрице введены дополнительные объекты доступа: множество объектов {Он1,..., Онk} — это объекты (группы объектов), созданные для субъектов C1,..., Ck для переадресации в них запросов, осуществляемых пользователями к неразделяемому системой объекту (группе объектов). Знак «—» обозначает, что это виртуальный объект, в котором не производится сохранение данных и к которому не может быть доступа субъекта.

Как видим из данной матрицы (реализована каноническая модель), технология перенаправления запросов к неразделяемым системой объектам позволяет выполнять требования к корректности управления доступом.

14.10.3. Практическая реализация

На рис. 14.4 приведена схема обработки запроса доступа к неразделяемому системой объекту, реализованная автором в КСЗИ «Панцирь» [31].

Работает система следующим образом. Для каждого каталога, создаваемого ОС или приложениями, доступ к которому не может быть разгра-

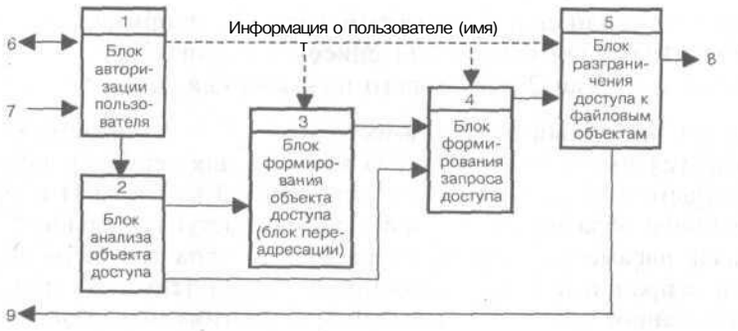


Рис. 14.4. *Схема обработки запроса доступа к неразделяемому системой объекту*

ничен для пользователей, например, `C:\RECYCLED`, создаются соответствующие каталоги, к которым осуществляется переадресация запроса доступа, например, каталог `C:\user 1` для первого пользователя, `C:\user 2` для второго пользователя и т.д. В результате каталог доступ к которому не может быть разграничен для пользователей становится виртуальным (физически в него не может быть записан и, соответственно, из него не может быть считан ни один файл).

В блоке 5 прописываются разграничения доступа для данных каталогов. В частности к каталогу `C:\user 1` разрешается доступ только первому пользователю (остальным запрещается), к каталогу `C:\user 2` разрешается доступ только второму пользователю (остальным запрещается) и т.д.

При обращении приложением (например, программой Word) к файловым объектам, приложение выдает на вход 7 запрос доступа, в котором указывается, к какому файловому объекту запрашивается доступ и какую операцию необходимо выполнить над файловым объектом.

При входе пользователя в систему (со входа 6) блоком 1 осуществляется его авторизация (проверяется имя и пароль). Имя текущего пользователя блоком 1 передается в блок 5, который содержит данные о разграничительной политике доступа каждого пользователя к файловым объектам (логическим дискам, каталогам, файлам) -- к каким файловым объектам доступ разрешен пользователю и какие права доступа к файловому объекту ему разрешены. Кроме того, имя текущего пользователя передается в блок 3, который формирует переадресацию запроса доступа для текущего пользователя.

Блок 2 выявляет, к какому каталогу запрошен доступ, если к каталогу, к которому не производится переадресации, то блок 2 транслирует исходный запрос через блок 4 в блок 5. В противном случае — передает запрос в блок 3. В блоке 3 для каждого каталога, доступ к которому переадресуется, содержится список переадресуемых каталогов — для каждого

пользователя задан переадресуемый каталог, например, для каталога C:\RECYCLED задан следующий список: каталог C:\user 1 для первого пользователя, C:\user 2 для второго пользователя, и т.д.

Блоком 3 в исходный запрос вместо имени запрашиваемого каталога подставляется имя переадресуемого каталога для текущего пользователя (имя текущего пользователя поступает в блок 3 из блока 1) и сформированный таким образом запрос через блок 4 поступает в блок 5, который сравнивает параметры запроса с правами доступа текущего пользователя. Если запрашиваемый приложением доступ текущему пользователю разрешен, запрос выдается на выход 8, в противном случае, на выход 9 блоком 5 формируется отказ приложению в запрашиваемом доступе.

Рассмотренный механизм может использоваться и для разделения объектов средствами защиты информации с целью решения различных функциональных задач защиты. Например, может быть осуществлено перенаправление к файлу Normal.dot каталога «Шаблоны» для пользователей. В этом файле располагаются макросы Microsoft Office. Перенаправление позволит задавать для каждого пользователя свой набор макросов.

В совокупности с механизмом контроля целостности данных перенаправленных файлов (этот механизм будет рассмотрен ниже) можно, с одной стороны, противодействовать атакам большой группы вирусов, а с другой стороны — фиксировать эталонный набор макросов не для защищаемого компьютера в целом, а для каждого пользователя в отдельности.

глава 15

Диспетчер доступа

15.1. Состав диспетчера доступа. Требования к полноте разграничительной политики доступа к ресурсам

15.1.1. Общие положения и принятые обозначения

Как отмечали ранее, диспетчер доступа содержит в своем составе набор механизмов управления доступом к ресурсам защищаемого объекта. Возникает проблема построения детальной модели диспетчера доступа в предположении, что корректно реализованы механизмы управления доступом, входящие в состав диспетчера.

Очевидно, что полнота модели диспетчера доступа определяется тем, для всех ли субъектов и объектов доступа реализованы разграничения. То есть, присутствуют ли в системе явные каналы несанкционированного взаимодействия субъектов доступа или нет.

Состав диспетчера доступа может быть определен на основании классификации возможных субъектов и объектов доступа, представленной ранее.

Сформулируем требование к полноте разграничительной политики доступа к ресурсам, реализуемой диспетчером доступа. Реализация данного требования, как отмечалось ранее, необходима, в первую очередь, для корректности мандатного управления доступом к ресурсам и обеспечения замкнутости программной среды на защищаемом объекте, т.е. важнейших механизмов защиты.

Введем следующие обозначения:

1. **Множества субъектов доступа.**

» **Пользователи.** Обозначим через P множество возможных пользователей в системе. Выделим три класса пользователей — возможных элементов множества P:

Pa ..... администратор;

Pп ..... пользователь, решающий прикладные задачи, соответственно, Pпп — n-й пользователь;

Pc ..... пользователь «система» — виртуальный пользователь ОС.

» **Процессы.** Обозначим через Q множество возможных процессов в системе. Выделим четыре класса процессов — возможных элементов множества Q:

Qc..... системные (привилегированные) процессы;

Qп..... прикладные процессы;

QCK..... скрытые или неидентифицируемые (процессы виртуальных машин).

2. Множество объектов доступа.

* **Файловые объекты данных.** Обозначим через F множество возможных файловых объектов данных в системе. Выделим три класса объектов — возможных элементов множества F:

Fc..... системные каталоги и файлы, каталоги и файлы настроек ОС;

Fп..... пользовательские каталоги и файлы данных, включая сетевые (в сети Microsoft — разделяемые сетевые ресурсы по протоколу Net Bios);

Fo..... неразделяемые системой и приложениями для пользователей каталоги и файлы (TEMP и т.д. для ОС семейства Windows).

* **Файловые объекты программ (исполняемые файлы).** Обозначим через S множество возможных файловых объектов программ. Выделим два класса объектов — возможных элементов множества S:

Sc..... системные исполняемые файлы привилегированных процессов — системных процессов ОС и процессов защиты;

Sп..... пользовательские исполняемые файлы (исполняемые файлы пользовательских приложений).

* **Установленные в ОС (санкционированные) устройства.** Обозначим:

U..... устройства (дисковод, CD-ROM, принтер и т.д.), как локальные, так и сетевые (разделяемые в сети);

Nu..... отчуждаемые физические носители информации для устройств ввода/вывода (дисковод, CD-ROM и т.д.);

Nuf..... файловые объекты (каталоги и файлы) на отчуждаемых физических носителях информации для устройств ввода/вывода (дисковод, CD-ROM и т.д.).

« **Неустановленные в системе (несанкционированные) устройства ввода/вывода (коммуникационные порты).** Обозначим:

E..... коммуникационные порты компьютера, к которым могут быть подключены устройства.

◆ **Каналы связи (виртуальные) ЛВС.** Обозначим:

K..... множество возможных виртуальных каналов ЛВС (определяются адресами в сети);

Кст ..... множество возможных сетевых технологий (определяется номерами портов и приложений), обеспечивающих взаимодействие в ЛВС.

3. Множество действий (устанавливаемые разграничения) субъектами доступа над объектами.

Множество действий (устанавливаемые разграничения) субъектами доступа над объектами обозначим через R. При этом выделим следующие категории доступа:

Rз ..... категория доступа «запись» (установка категории доступа «запись» означает разрешение полного доступа — запись и чтение, т.к. не имеет смысла разрешать запись, не разрешая чтение);

Rч ..... категория доступа «чтение»;

Rд ..... категория доступа «добавление» (установка категории доступа «добавление» означает разрешение записи с предотвращением возможности затирания и модификации информации, располагаемой в объекте доступа, и запрет чтения);

Rи ..... категория доступа «исполнение» («запуск»).

$$R = Rз \cup Rч \cup Rд \cup Rи;\ Rз \cap Rч = Rч.$$

Введем также категорию $\overline{R}$ — любой доступ запрещен. При этом установление категории R подразумевает, что в рамках реализации механизма управления доступом могут применяться комбинации соответствующих категорий: Rз, Rч, Rд, Rи.

4. Модель диспетчера доступа.

Обозначим:

- Wn — модель доступа n-го пользователя к объектам (индекс n будем использовать для выделения тех субъектов и тех объектов, на которые устанавливаются разграничения для n-го пользователя, если индекс не установлен — разграничения не устанавливаются);
- для указания действия при доступе к объекту будем использовать запись — [действие][объект], например, RF - - обозначает полный доступ ко всем классам объектов F, RчFn -- обозначает доступ по чтению к разрешенным для чтения пользователю n объектам F, RnFn — обозначает разграничение по действию над объектами Fn для пользователя n;
- если существуют разграничения A и B, то для указания в модели необходимости задания обоих ограничений одновременно будем использовать знак «*» (A*B); для указания того, что может использоваться любое из ограничений (A или B), используем знак «+» (A+B).

Диспетчер доступа системы защиты в любой момент времени его функционирования предполагает реализацию двух видов разграничения прав доступа:

» прикладное разграничение доступа (разграничение доступа собственно для пользователей и прикладных процессов), обозначим модель доступа через Wpn для n-го пользователя;

* системное разграничение доступа (разграничение доступа для системных процессов и процессов системы защиты — привилегированных процессов), соответственно, обозначим модель доступа для него через Wc.

15.1.2. Формулировка и доказательство требований к полноте разграничительной политики диспетчера доступа

С учетом сказанного (привилегированные процессы всегда присутствуют в системе), т.е. для модели Wn имеем:

$$Wn = Wpn \cdot Wc.$$

Утверждение. Модель диспетчера доступа функционально полна (корректна) в том случае, когда для нее выполняются следующие условия:

1. Для двух любых пользователей системы n1 и п2, решающих прикладные задачи, доступ ко всем объектам может быть полностью разграничен. При этом модели доступа имеют следующий вид:

$$Wn1 = Wpn1 \cdot Wc,$$
$$Wn2 = Wpn2 \cdot Wc,$$

причем выполняется условие: $Wpn1 \cap Wpn2 = 0$, соответственно:

$$Wn1 \cap Wn2 = Wc.$$

2. Для всех субъектов и объектов доступа реализованы разграничения, т.е. в системе отсутствуют каналы (включая скрытые) несанкционированного взаимодействия субъектов доступа, что формально может быть представлено в следующем виде:

$$P = Pa \cup Pп \cup PC;\ Pa \cap Pп \cap Pc = 0$$
$$Q = Qc \cup Qп \cup QCK;\ Qc \cap Qп \cap QCK = 0$$
$$F = FH \cup Fп \cup Fo;\ FH \cap Fп \cap Fo = 0$$
$$S = Sc \cup Sп;\ Sc \cap Sп = 0$$
$$Up = U\ (Nup = Nu,\ Nufp = Nuf)$$
$$Ep = E$$
$$Kp = K\ (Kстр = Kст),$$

где индекс «р» у объекта означает, что к нему разграничен доступ.

Доказательство утверждения очевидно — при выполнении заданных условий для всех субъектов и объектов доступа реализованы разграничения, т.е. в системе отсутствуют каналы (включая скрытые) несанкционированного взаимодействия субъектов доступа.

15.1.3. Диспетчер доступа для ОС семейства Windows

Состав диспетчера доступа

С учетом введенной выше классификации субъектов и объектов доступа рассмотрим состав диспетчера доступа для ОС семейства Windows (с учетом описанных ранее механизмов защиты и описания их применения для решения задач управления доступом) и построим его модель, определяющую требования как к диспетчеру доступа в целом, так и к его компонентам.

Состав диспетчера доступа, обеспечивающего полноту разграничительной политики доступа к ресурсам, представлен на рис. 15.1.

Рис. 15.1. *Состав диспетчера доступа для ОС семейства Windows*

Анализ полноты разграничительной политики доступа, реализуемой разработанным диспетчером доступа для ОС Windows

С учетом всего сказанного ранее проанализируем полноту разграничительной политики доступа, реализуемую диспетчером доступа для ОС семейства Windows, состав которого определен выше (см. рис. 15.1). При этом для ОС семейства Windows введем новое обозначение Н для управляющего реестра ОС.

Модель доступа Wpn для n-го пользователя задается следующим выражением:

$$Wpn = RnFnn\,(Pnn + Qnn) \cdot RnnSnn\,(Pnn + Qnn) \times$$
$$\times RnFon\,(Pnn + Qnn) \cdot RnUn\,(Pnn + Qnn) \cdot RnNu\,(Pnn + Qnn) \times$$
$$\times RnNufn\,(Pnn + Qnn) + RnKmn\,(Pnn + Qnn) \cdot RnKctn\,(Pnn + Qnn) \times$$
$$\times \overline{RFc}\,(Pnn + Qnn) \cdot \overline{RSc}\,(Pnn + Qnn)\,\overline{RE} \cdot RH\,(Pnn + Qnn).$$

Итак, оценим полноту модели диспетчера доступа. С учетом сформулированных ранее требований к реализации механизмов управления доступом, в предположении, что эти требования выполняются, для пользователей n1 и n2 имеем:

$$Wpn1 \cap Wpn2 = \overline{RFc}\,(Pnn + Qnn) \cdot \overline{RSc}\,(Pnn + Qnn) \times$$
$$\times \overline{RE} \cdot \overline{RH}\,(Pnn + Qnn).$$

Модель Wc имеет в рассматриваемом случае следующий вид:

$$Wc = RFc\,(Pnc + Qnc) \cdot RSc\,(Pnc + Qnc) \cdot RE \cdot RH\,(Pnc + Qnc),$$

где индекс «с» означает пользователя «СИСТЕМА».

Таким образом, общим в разграничениях пользователей будет запрет доступа пользователей с использованием любых прикладных процессов к системному диску (системным данным и запуску системных процессов), доступа к портам, к которым могут быть подключены несанкционированные устройства, доступа к управляющему реестру ОС. Все это подтверждает корректность рассматриваемой модели диспетчера доступа для ОС семейства Windows.

15.1.4. Построение диспетчера доступа к сетевым ресурсам

Задачи диспетчера доступа к сетевым ресурсам

В предыдущих главах отмечалось, что при использовании защищаемого объекта в составе ЛВС встает задача изоляции информационных потоков, циркулирующих в ЛВС.

В данном разделе мы рассмотрим построение диспетчера доступа к сетевым устройствам (компьютерам) в составе ЛВС по протоколу TCP/IP. Механизмы управления доступом к разделяемым сетевым ресурсам — файловым объектам (общим папкам) — были рассмотрены ранее (п. 13.7).

Согласно формализованным требованиям система защиты должна обеспечивать защищенный механизм ввода и вывода информации для объекта доступа. В данном случае таким объектом является канал связи.

Примечание

В общем случае разграничение прав доступа должно осуществляться не только применительно к каналу связи, но и к сетевым ресурсам. Под сетевым ресурсом здесь понимаем виртуальный канал связи, образуемый собственно физическим каналом между компьютерами и реализуемой в рамках их взаимодействия сетевой службой (ftp, telnet и т.д.). При этом физический канал связи характеризуется IP-адресом хоста, с которым осуществляется взаимодействие, сетевая служба — номером TCP-порта.

Разграничение доступа к узлам ЛВС предназначено для изоляции информационных потоков ЛВС — виртуальной сегментации сетевого пространства ЛВС. При этом каждому пользователю (на рабочей станции) разрешается взаимодействие с определенным набором рабочих станций и серверов с использованием фиксированного набора сетевых служб.

Разграничение доступа к внешним сетевым ресурсам предназначено для защиты доступа к сети Internet и Intranet. При этом зарегистрированным в системе пользователям должен разрешаться или запрещаться доступ к внешним по отношению к защищаемой системе ресурсам — рабочим станциям и серверам с использованием соответствующей сетевой службы.

Диспетчером доступа к сетевым ресурсам должна решаться следующая совокупность задач:

1. Должно обеспечиваться разграничение доступа к узлам и к хостам сети Internet и Intranet (на стеке протоколов TCP/IP) на уровне IP-адресов и TCP-портов, то есть на уровне сетевых служб и процессов, обеспечивающих доступ к сетевым ресурсам. Таким образом, должно обеспечиваться разграничение доступа по следующим параметрам:
   - пользователям;
   - процессам;
   - времени доступа;
   - по службам доступа (портам);
   - политике безопасности (запрещенные/разрешенные хосты и службы).
2. Должна обеспечиваться виртуальная сегментация сетевого пространства внутренней защищаемой сети (ЛВС), посредством разграничения прав доступа к внутренним серверам и рабочим станциям.

Виртуальная сегментация сетевого пространства осуществляется на уровне пользователей, что принципиально отличает данный подход логического деления сети на подсети от способов, предполагающих использование дополнительных технических средств физической сегментации на подсети — маршрутизаторов, межсетевых экранов и т.д.

Другое принципиальное отличие состоит в реализации логического (виртуального) деления внутренней сети на подсети, инвариантного к структуре сети (не требуется изменения связного ресурса — включения дополнительных осуществляющих физическое деление сети на подсети устройств -

коммутаторов, межсетевых экранов и т.д.). Для подобного деления нет необходимости в доработке структуры сети. Задача решается средствами диспетчера доступа, устанавливаемого на рабочие станции и серверы ЛВС.

Схема задания разграничений

В общем случае схема задания разграничений имеет вид: **пользователь — разрешенные компьютеры (IP-адреса) — разрешенные сетевые службы (TCP-порты).** То есть для каждого пользователя задаются разрешенные компьютеры, а на разрешенных компьютерах — разрешенные сетевые службы. При этом целесообразно отдельно задавать параметры исходящих и входящих соединений, так как они могут отличаться. Эти функции выполняют так называемые пакетные фильтры, входящие в состав некоторых ОС семейства UNIX.

В указанную схему в качестве субъектов доступа имеет смысл дополнительно ввести «ПРОЦЕССЫ» и «РАСПИСАНИЕ». Расписанием задаются интервалы времени, в течение которых пользователю разрешается доступ к сети. Для различных временных интервалов (расписаний) для каждого пользователя могут быть разрешены свои объекты доступа (IP-адреса и TCP-порты).

В качестве процессов могут приниматься сетевые приложения, с использованием которых пользователь может взаимодействовать с разрешенными компьютерами в рамках разрешенных сетевых служб — TCP-портов, соответственно, протоколов. Таким образом, введение в расмотрение субъекта «ПРОЦЕСС» позволяет использовать оригинальные приложения для доступа к сети, в частности, приложения с усеченными функциями (наборами команд). Данная возможность позволяет частично либо полностью отказаться от фильтрации пакетов на прикланом уровне (в части усечения функций взаимодействия). При этом пользователь сможет взаимодействовать с сетью только в рамках тех функций, которыми будет обладать приложение.

Итак, возвращаясь к схеме задания разграничений, можно сделать вывод, что она сводится к следующему виду: **пользователь — процесс (сетевое приложение) -- расписание — IP-адрес — TCP-порт (сетевая служба или протокол).**

Реализация мандатного механизма разграничения доступа к ресурсам сети

Под реализацией мандатного доступа к ресурсам сети следует понимать мандатное управление доступом к виртуальному каналу. В качестве такого канала здесь понимается сетевая служба, определяющая информационную технологию. Примерами могут служить: служба открытой почты, служба конфиденциальной почты, служба обработки SQL запросов и т.д.

Управление доступом к виртуальным каналам на практических примерах подробно было описано в п. 14.7.3. Однако позволим себе еще раз, с общей точки зрения, рассмотреть этот подход.

Итак, в рамках этого подхода учитываются метки безопасности сетевых служб (процесса), а не пользователей. Метка для процесса устанавливается (помечается сетевая служба) в настройках механизма мандатного управления доступом. Если для процесса не указана метка безопасности, для него доступ разграничивается с учетом меток пользователей.

В соответствии с указанной меткой помеченный процесс может обращаться к объектам файловой системы и к устройствам, что реализуется мандатным механизмом управления доступом к файловым объектам. Механизмом управления доступом к сети заносятся дополнительные дискреционные разграничения для помеченной службы, в частности, с какими компьютерами разрешено взаимодействие и т.д. Пример мандатного разграничения доступа к внешней сети представлен в табл. 15.1.

Пример мандатного разграничения доступа к внешней сети *Таблица 15.1*

Метка безопасности	Субъекты доступа	Объекты доступа
1		
...		
N-1	Процесс (программа) конфиденциальной почтовой службы	
N	Процесс (программа) открытой почтовой службы	

При данных настройках мандатного доступа к объектам файловой системы метки безопасности назначаются процессам:

» метка N−1 -- процессу конфиденциальной почтовой службы;
« метка N — процессу открытой почтовой службы.

Данные настройки реализуют следующую возможность взаимодействия в рамках помеченной сетевой службы (информационной технологии). Любым пользователем по защищенному почтовому каналу (конфиденциальная почтовая служба) могут передаваться объекты, которым сопоставлены метки N−1 и N. То есть только эти объекты служба сможет читать из файловой системы.

Данные с меньшей, чем N-1 меткой (например, секретные данные), по конфиденциальному каналу пользователи передать не могут. Соответственно, любое получаемое по защищенной почте сообщение любой пользователь сможет записать только в объект (например, каталог), которому сопоставлена метка N−1, то есть не может записать конфиденциальные данные, полученные по почте в объект с большей меткой.

Любым пользователем по открытому почтовому каналу могут передаваться только объекты, которым сопоставлена метка N (только эти объекты служба сможет читать из файловой системы). Соответственно, любое получаемое по открытой почте сообщение любой пользователь сможет записать только в объект (например, каталог), которому сопоставлена метка N. Таким образом, реализуется мандатный доступ к виртуальному каналу сети.

В дополнение можно установить пароль ответственного лица на запуск сетевой службы (этот вопрос рассматривался ранее). При этом можно сделать так, чтобы доступ к сети (например, к Интернету) мог состояться только в присутствии или под непосредственным контролем ответственно лица. В качестве такового может выступать, например, начальник подразделения. Кроме того, дискреционным механизмом можно дополнительно задать с какими хостами (по их IP-адресам) может устанавливаться помечаемый виртуальный канал (может взаимодействовать соответствующая служба), в какие интервалы времени может осуществляться взаимодействие и т.д.

С учетом появления скрытого канала взаимодействия субъектов — канала связи, по которому передаются данные между рабочими станциями и серверами, в дополнение здесь целесообразно реализовать шифрование трафика в виртуальном канале ЛВС. Подобные функции реализуют ряд добавочных средств защиты, в частности, система Vip Net.

Схема обработки запроса и функциональная схема использования диспетчера доступа к сетевым ресурсам

На рис. 15.2 приведена схема механизма виртуальной сегментации канала, реализуемой диспетчерами доступа, установленными на защищаемые объекты (рабочие станции и серверы ЛВС), а на рис. 15.3 — схема обработки запроса доступа к объекту — сетевому ресурсу.

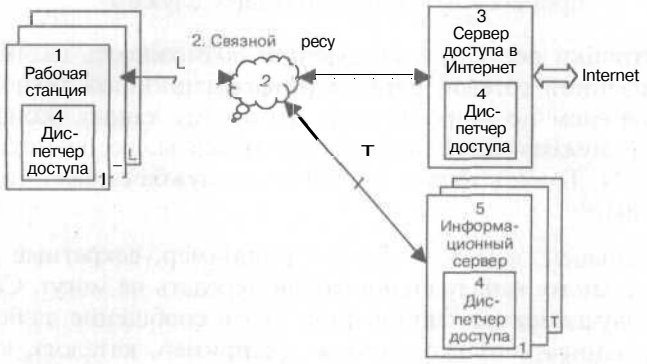
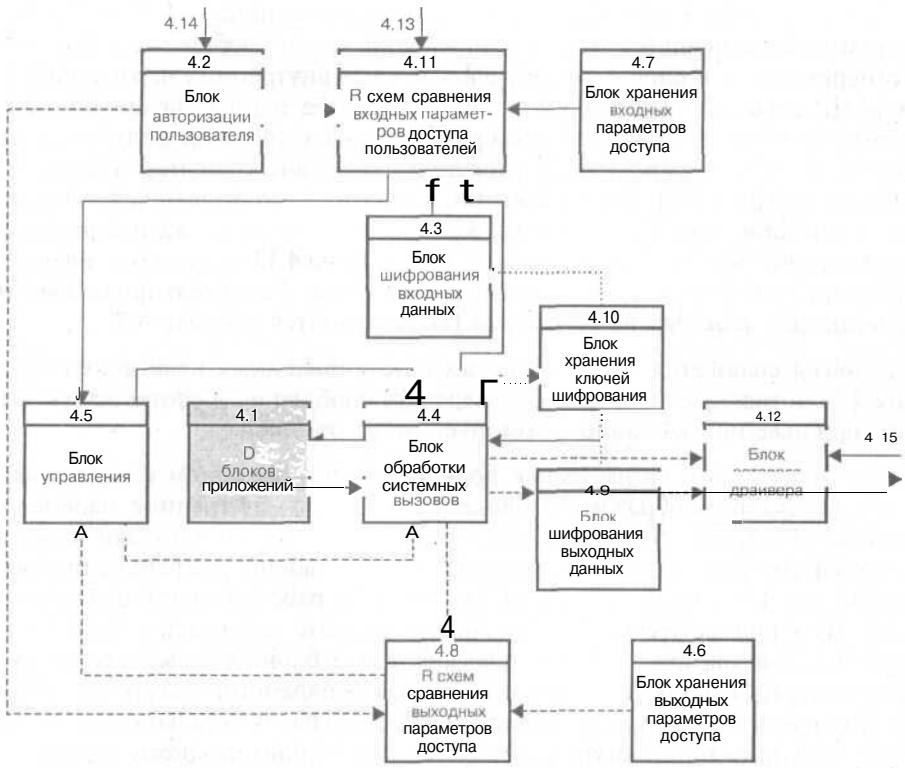


Рис. *15.2. Схема системы виртуальной сегментации канала, реализуемой диспетчерами доступа рабочих станций и серверов*



Сплошными стрелками показан входящий трафик
Пунктирными стрелками показан исходящий трафик
Точечными линиями показано функционирование системы шифрования

Рис. 15.3. *Схема диспетчера доступа*

На схеме, представленной на рис. 15.2, использованы следующие обозначения — ЛВС содержит L рабочих станций 1, связной ресурс 2 (канал связи, объединяющий хосты, сервер доступа к сети Internet 3, а в каждой рабочей станции и сервере — диспетчер доступа, реализующий разграничения прав доступа 4, T информационных серверов 5.

На схеме, представленной на рис. 15.3, использованы следующие обозначения — D блоков приложений 4.1, блок авторизации пользователя 4.2, блок шифрования входных данных 4.3, блок обработки системных вызовов 4.4, блок управления 4.5, блок хранения выходных параметров доступа 4.6, блок хранения входных параметров доступа 4.7, R схем сравнения выходных параметров доступа 4.8, блок шифрования выходных данных 4.9, блок хранения ключей шифрования 4.10, R схем сравнения входных параметров доступа пользователей 4.11, блок сетевого драйвера 4.12, Q входов параметров доступа 4.13, вход авторизации пользователя 4.14, сетевой вход-выход 4.15.

Рассмотрим работу схемы. Входной канал разграничения прав доступа реализуется следующим образом. При поступлении сообщения в блок 4.12 оно переводится в блок 4.4, где выделяются S внутренних параметров доступа (IP адрес, № TCP порта — сетевая служба и т.д.). Затем эти параметры подаются в блок 4.11, где сравниваются с правами получения информации, заданными для пользователя (его идентификатор в блок 4.11 попадает с блока 4.2). Установленные права для пользователей, заведенных в системе, хранятся в блоке 4.7 и выбираются по идентификатору пользователя. Кроме того, в блок 4.11 со входа 4.13 поступают внешние параметры доступа — день недели, время и т.д. Регламентированные их значения для каждого пользователя также хранятся в блоке 4.7.

Результаты сравнения запрашиваемых и установленных прав поступают в блок 4.5, который либо разрешает передать сообщение в соответствующий блок приложений 4.1, либо формирует отказ от этого.

Разграничение доступа на уровне пользователей в выходном канале реализуется следующим образом. В блоке 4.6 хранятся внутренние параметры доступа (IP-адреса, № TCP-порта — сетевая служба и т.д.) для каждого пользователя. При этом диспетчер содержит L таблиц разграничения прав доступа, где L — число зарегистрированных на рабочей станции пользователей. Имя пользователя, находящегося в системе поступает в блок 4.6 из блока 4.2. Для авторизованного пользователя в блоке 4.8 выбирается своя таблица доступа, в соответствии с которой (ее параметры выдаются в блок 4.8) и осуществляется разграничение прав доступа. В остальном выходной канал разграничения доступа реализуется аналогично входному каналу.

Шифрование передаваемых данных в рамках отдельной подсети реализуется следующим образом. Выходное сообщение из блока 4.4 в блок 4.12 поступает через блок 4.9. Аналогично входное сообщение из блока 4.4 в блок 4.11 поступает через блок 4.3, т.е. через блоки шифрования (расшифровки). Для каждого IP-адреса при приеме и передаче сообщений используется ключ шифрования, хранящийся в блоке 4.10. Этот ключ может быть один на всю подсеть, выделяя эту подсеть по шифрованию данных на едином сетевом пространстве. IP-адрес сообщения (выдаваемого или принимаемого) поступает в блок 4.10 из блока 4.4, где по этому адресу выбирается ключ шифрования. Затем этот ключ передается либо в блок 4.9 при выдаче сообщения, либо в блок 4.3 при приеме сообщения. Соответственно, в блок 4.12 поступает сообщение, зашифрованное данным ключом, либо в блок 4.11 сообщение, расшифрованное выбранным по IP-адресу ключом.

Всеми действиями по осуществлению доступа управляет блок 4.5. В общем случае в блоке 4.10 хранится и собственно IP-адрес компьютера, на котором установлена система. Это необходимо для того, чтобы осуществить возможность присутствия одного компьютера сразу в нескольких подсетях, для которых ключи шифрования различны. Для однозначного выбора ключа шифрования в этом случае необходимо знание двух параметров — собственного IP-адреса и IP-адреса, с которым осуществляется взаимодействие.

15.2. Оценка влияния, оказываемая на вычислительную систему системой защиты

Итак, как мы увидели выше, большинство требований к корректности реализации механизмов защиты, а также существенные функциональные расширения их возможностей связаны с включением в схему управления доступом субъекта «ПРОЦЕСС». Заметим, что возможность разграничения прав доступа процесса к ресурсам, как самостоятельного субъекта доступа, отсутствует в механизмах управления доступом, реализуемых современными универсальными ОС, что существенно ограничивает возможности защиты информации, реализуемые встроенными в ОС механизмами защиты.

В этом разделе мы проведем оценку, в какой мере влияет на загрузку вычислительного ресурса защищаемого объекта включение в схему управления доступом дополнительного субъекта доступа «ПРОЦЕСС». Для этого построим следующие математические модели:

- « математическую модель рабочей станции без системы защиты;
- * математическую модель рабочей станции с реализацией управления доступом к файловым объектам для субъекта доступа «ПОЛЬЗОВАТЕЛЬ»;
- » математическую модель рабочей станции с реализацией управления доступом к файловым объектам для субъектов доступа «ПОЛЬЗОВАТЕЛЬ» и «ПРОЦЕСС».

Будем строить математическую модель функционального объекта, где элементы модели в общем случае отображают не структурные элементы физического объекта (что, естественно, является менее корректным решением задачи, с точки зрения подходов к математическому моделированию), а функциональные взаимодействия, что более наглядно иллюстрирует получаемый результат (и, что нетрудно подтвердить, приводит к незначительной погрешности в результатах).

15.2.1. Модель рабочей станции без системы защиты

Общие положения

Для построения модели необходимо решить, на какой круг задач будет рассчитана вычислительная система. Будем рассматривать типовые задачи, с которыми ч.аще всего приходится сталкиваться в современных вычислительных системах общего назначения. Задачи гипотетической вычислительной системы, модель которой требуется построить, ограничим следующим списком:

1. Обработка текстов. Редактирование и верстка.
2. Математическая обработка данных.
3. Разработка программ. Редактирование и сборка.

В качестве типовых приложений, реализующих вышеперечисленные задачи, рассмотрены следующие программы:

* текстовый редактор — Word 97;
* текстовый компилятор — MikTeX 2.2;
* электронные таблицы — Excel 97;
* разработка программ — MS Visual C++ 5.0.

Основные параметры (число операций счета, количества обращений к внешним устройствам и т.п.) этих программ определялись экспериментально. В качестве экспериментального стенда использовался компьютер на базе процессора Intel Celeron 1000 МГц, объем памяти (ОЗУ) 384 Мб, внешнее запоминающее устройство -- накопитель на жестком магнитном диске (НЖМД) IBM IC35L040 AVER07-0, ОС -- MS Windows 2000 Professional.

Параметры программ определялись с помощью следующих специальных средств [36]:

* Filemon -- монитор обращений к файловой системе;
* Regmon — монитор обращений к реестру;
* Diskmon -- монитор обращений к физическим носителям, например, к НЖМД;
* Cpumon — монитор центрального процессора;
* системный монитор, входящий в состав ОС MS Windows 2000 Professional.

Параметры приложений, полученные в результате серии экспериментов и последующих расчетов с использованием моделей (собственно модели и соответствующие им расчетные формулы приведены ниже), представлены в табл. 15.2.

Параметры приложений *Таблица 15.2*

Параметр	MS Word	TEX	MS Excel	MS VC
Количество обращений к ФС, п2	4725	9375	2631	63125
Среднее время выполнения запроса к ФС, u2, мкс	190	475	425	534
Количество обращений к реестру ОС, п3	4548	1064	2176	17542
Среднее время выполнения запроса к реестру, u3, мкс	611	500	130	741
Среднее время выполнения, U, с	5,7	55	5,77	59
Интенсивность потока заявок , λ0, 1/с	0,2	0,125	1,2	0,04
Загрузка процессора Q, %	62	93	22	60
Количество этапов счета N = п2 + п3 +1	9274	10440	4808	80668

Общий вид несетевой модели рабочей станции без системы защиты

Математическая модель вычислительной системы (рабочей станции) без системы защиты представлена на рис. 15.4. Она представляет собой разомкнутую сеть массового обслуживания (СеМО). Эта СеМО состоит из трех узлов, в свою очередь представляющих собой отдельные системы массового обслуживания (СМО):

* система массового обслуживания «Процессор — оперативная память»;
» система массового обслуживания «Файловая система»;
» система массового обслуживания «Реестр».

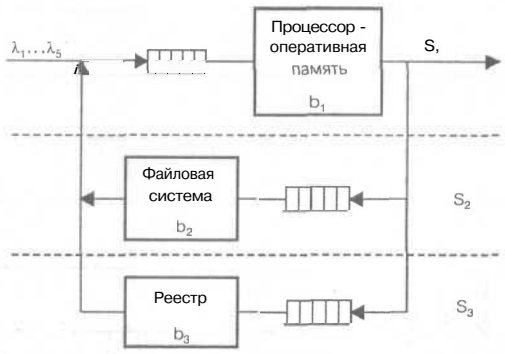


Рис. 15.4. *Несетевая модель рабочей станции без системы защиты*

В качестве входного потока примем стационарный пуассоновский поток.

Стационарный пуассоновский поток является простейшим потоком, а для простейшего потока характерно, что поступление заявки через короткий промежуток времени более вероятно, чем через длинный промежуток времени.

Следствием этого свойства является то, что простейший поток создает наиболее тяжелый режим работы для системы [6]. Поэтому простейший поток в моделях позволяет получать предельные значения входных характеристик, и, следовательно, если входной поток в реальной системе отличен от простейшего, то система будет иметь характеристики функционирования, по крайней мере, не хуже, чем при простейшем входном потоке.

Интенсивность входного потока определена экспериментально. Для задания характеристики среднего времени обслуживания в каждом узле сети массового обслуживания рассмотрим эти СМО по отдельности.

Система массового обслуживания «Файловая система»

Параметры этой СМО определялись с помощью экспериментальных данных. Для определения закона распределения длительности обслуживания

были построены гистограммы распределения значений длительностей обслуживания для каждого приложения. Все эти гистограммы имели практически одинаковый вид, представленный на рис. 15.5.

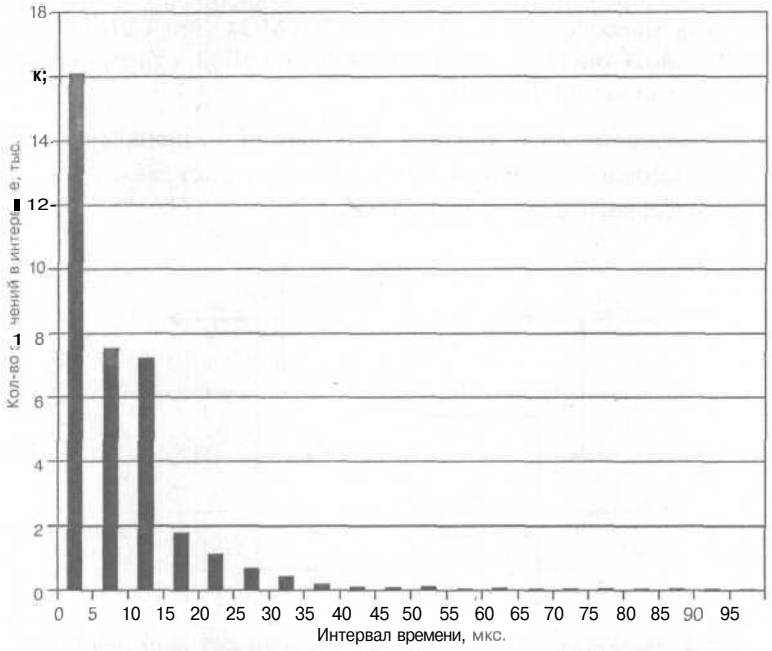


Рис. 15.5. *Распределение длительности обслуживания в СМО «файловая система»*

Как видим, полученное распределение при моделировании СМО «файловая система» при определенных оговорках подчиняется экспоненциальному закону.

Для определения среднего времени обслуживания в СМО «файловая система» используем следующее отношение:

$$u = \frac{b}{1 - p}, \quad \text{соответственно, } p = \lambda b = \alpha\lambda_0 b, \quad \text{получаем } o = \frac{u}{1 + \alpha\lambda_0 u}, \quad (15.1)$$

где: a — коэффициент передачи в соответствующий узел СеМО;

λ_0 — интенсивность потока заявок на входе СеМО;

u - - среднее время пребывания заявки в СеМО.

Среднее время обслуживания в СМО «файловая система» определялось по результатам экспериментов, с использованием отношения (15.1), и для каждого приложения представлено в табл. 15.3.

Система массового обслуживания «Реестр»

Параметры этой СМО также определялись экспериментально.

Гистограмма распределения длительности обслуживания для всех приложений имеет вид, очень похожий на гистограмму, представленную на рис. 15.4. Поэтому, следуя заключениям, сделаным для СМО «файловая система», будем также считать, что распределение длительности обслуживания в СМО «реестр» подчиняется экспоненциальному закону.

Среднее время обслуживания в СМО «реестр» определялось по результатам экспериментов, с использованием отношения (15.1), и для каждого приложения представлено в табл. 15.3.

Система массового обслуживания «Процессор — оперативная память»

Рассмотрим СМО «Процессор — оперативная память». Данных, указывающих на закон распределения времени обслуживания в СМО «процессор — оперативная память» нет. Для упрощения расчета модели и по аналогии с рассмотренными ранее СМО предположим что распределение длительности обслуживания в СМО «Процессор — оперативная память» подчиняется экспоненциальному закону.

Для определения средней длительности обслуживания будем использовать формулу (15.1) и следующее соотношение:

$$U = Nu_1 + n_2u_2 + n_3u_3,$$

откуда получаем:

$$u_1 = \frac{U - n_2u_2 - n_3u_3}{N},$$

где: u_1 - среднее время пребывания в СМО «процессор—оперативная память»;

u_2 - - среднее время пребывания в СМО «файловая система»;

u_3 •- среднее время пребывания в СМО «реестр»;

n_2 - - количество обращений к файловой системе;

n_3 - - количество обращений к реестру;

N - количество этапов счета;

U - среднее время пребывания в СеМО.

Анализ несетевой модели рабочей станции без системы защиты

Согласно формуле (15.1) и табл. 15.2 определяется для каждого приложения среднее время обслуживания в узлах модели. Результаты вычислений представлены в табл. 15.3.

Среднее время обслуживания в узлах модели *Таблица 15.3*

Параметр	MS Word	TEX	MS Excel	MS VC
Время обслуживания в СМО «Процессор — оперативная память», b1, с	0,000155	0,000660	0,000145	0,000102
Время обслуживания в СМО «Файловая система», Ь2, с	0,000161	0,000305	0,000181	0,000227
Время обслуживания в СМО «Реестр», Ь3, с	0,000392	0,000468	0,000097	0,000487

Сетевая модель рабочей станции без системы защиты

Теперь перейдем к рассмотрению сети массового обслуживания (СеМО). Обозначим СМО «Процессор — оперативная память» через S1, СМО «Файловая система» через S2, а СМО «Реестр» через S3. Введем также узел SO, через который в систему поступают заявки на обслуживание. Граф передач такой сети представлен на рис. 15.6.

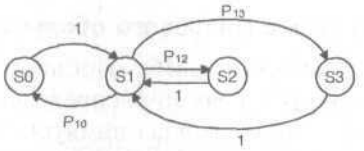


Рис. 15.6. *Граф передач СеМО для сетевой модели*

Соответствующая этому графу матрица вероятностей передач выглядит следующим образом:

$$P = \begin{vmatrix} 0 & 1 & 0 & 0 \\ P_{10} & 0 & P_{12} & P_{13} \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$$

Используя эту матрицу, получаем систему уравнений, связывающую между собой интенсивности потоков в узлах СеМО и вероятности передач:

$$\begin{cases} -\lambda_0 + P_{10}\lambda_1 = 0; \\ \lambda_1 + \lambda_2 + \lambda_3 = 0; \\ P_{12}\lambda_1 - \lambda_2 = 0; \\ P_{13}\lambda_1 - \lambda_3 = 0. \end{cases}$$

Решая эту систему уравнений, получаем выражения для определения коэффициентов передач:

$$\alpha_j = \frac{\lambda_j}{\lambda_0} \cdot \alpha_1 = \frac{1}{P_{10}}; \quad \alpha_2 = \frac{P_{12}}{P_{10}} \cdot \alpha_3 = \frac{P_{13}}{P_{10}},$$

где у — номер СМО.

Для нахождения с помощью таких выражений численного значения коэффициентов передач необходимо определить вероятности перехода заявки из одного узла СеМО в другой.

Очевидно, что $P_{10i} = 1/N_i$, откуда:

$$\alpha_{1i} = N_i,$$

где N_i — количество этапов счета для i-го типа приложения (в нашем случае $I = 5$).

Вероятность перехода заявки в СМО «Файловая система» есть отношение количества обращений к файловой системе к количеству этапов счета, то есть:

$$P_{12i} = \frac{n_{2i}}{N_i},$$

откуда:

$$\alpha_{2i} = \frac{n_{2i}}{N_i} \cdot N_i = n_{2i},$$

где n_{2i} - - количество обращений к ФС;
N_i количество этапов счета, i — тип приложения.

Таким же образом определяем вероятность перехода заявки в СМО «Реестр»:

$$P_{13i} = \frac{n_{3i}}{N_i},$$

откуда:

$$\alpha_{3i} = \frac{n_{3i}}{N_i} \cdot N_i = n_{3i},$$

где n_{3i} — количество обращений к реестру;
N_i — количество этапов счета;
i - тип приложения.

Расчет характеристик сети далее будет производиться, исходя из условия, что сеть работает в стационарном режиме, то есть вероятностные характеристики сети не зависят от времени. Условие существования стационарного режима в сети определяется существованием стационарного режима во всех узлах этой сети.

Стационарный режим существует в отдельной СМО с одним обслуживающим прибором, если выполняется следующее условие:

$$\lambda < 1/b,$$

то есть, если загрузка прибора меньше единицы.

Исходя из этого, можно сформировать условие существования стационарного режима в рассматриваемой сети:

$$\lambda_0 < min\left\{\frac{1}{\alpha_1 b_1}; \frac{1}{\alpha_2 b_2}; \frac{1}{\alpha_3 b_3}\right\}$$

и руководствоваться им при выборе максимальной интенсивности входного потока заявок каждого типа, при расчете характеристик сети.

Самой важной характеристикой сети для нас является время пребывания заявки в сети. Время пребывания заявки в сети определяется выражением:

$$U = \sum_{i=1}^{M} \alpha_i u_i ,$$

где $u_i = \frac{b_i}{1-\rho_i}$; M — число СМО в сети.

Так как:

$$\rho_i = \lambda_i b_i = \alpha_i \lambda_0 b_i ,$$

то выражение, определяющее время пребывания заявки в рассматриваемой сети, можно записать следующим образом:

$$U \; \Pi \; \frac{\alpha_i b_i}{1-\alpha_i \lambda_0 b_i} .$$

15.2.2. Модель рабочей станции с системой защиты

Под системой защиты в данном случае подразумевается система разграничения прав доступа к ресурсам файловой системы и реестра, с учетом «ПРОЦЕССА» как самостоятельного субъекта доступа.

Из блок-схемы алгоритма управления доступом с учетом субъекта доступа «ПРОЦЕСС» (приведена ранее в п. 14.1.2), среднюю трудоемкость T алгоритма анализа запроса доступа к ресурсу можно выразить следующим образом:

$$T = (P_1 + P_3)x + 2P_2x ,$$

где: x • трудоемкость проверки прав доступа;

P1 - вероятность поступления запроса от процесса, не упомянутого в разграничениях прав доступа;

P2 - вероятность поступления запроса от процесса, имеющего совместный (вместе с субъектом «пользователь») режим проверки прав доступа;

P3 - вероятность поступления запроса от процесса, имеющего эксклюзивный режим проверки прав доступа.

Исследование средств добавочной защиты для ОС семейства Windows показало, что вероятность поступления запроса от привилегированного процесса составляет порядка 0.3, процесса с совместным режимом проверки запроса порядка 0.1.

Поэтому для дальнейших расчетов примем среднее время проверки запроса с учетом субъекта «процесс», как самостоятельного субъекта доступа, равным ((1 - 0.3 - 0.1) + 0.3)x + 2 · 0.1x = 1,1x.

Из всех типов запросов, обращающихся к файловой системе и реестру, проверке (анализу прав доступа) подлежит лишь некоторая их часть. Бессмысленно проверять, например, операцию закрытия файла или операцию сброса буферов ввода-вывода. В результате проведенных экспериментов определено, что контролировать целесообразно 90% запросов (в общем потоке) к ресурсам реестра и 34% запросов к файловой системе.

Экспериментально было определено время, требуемое на проведение проверки (контроля прав доступа) для одного запроса. В среднем это время составляет 130 мкс.

Исходя из вышеизложенного, для дальнейших расчетов примем следующее выражение для определения суммарного среднего времени обслуживания в системах массового обслуживания:

$$b_4 = 130x\frac{(0.9+0.340)}{2}.$$

Модель рабочей станции с системой защиты, изображенная на рис. 15.7, представляет собой разомкнутую сеть массового обслуживания (СеМО). По сравнению с рассматриваемой ранее моделью без системы защиты, в этой СеМО появилась новая система массового обслуживания — СМО «Система защиты». Граф передач такой сети представлен на рис. 15.8.

Соответствующая этому графу матрица вероятностей передач выглядит следующим образом:

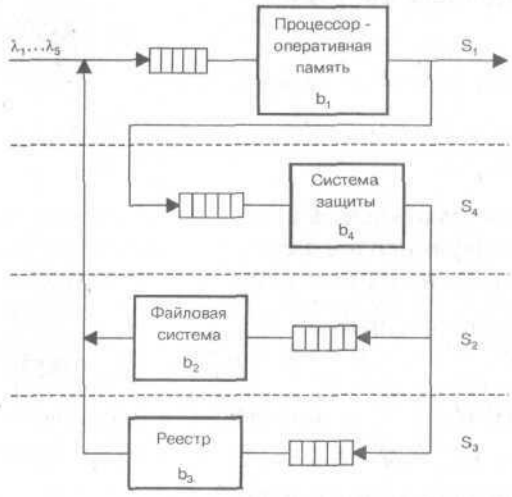


Рис. 15.7. *Модель рабочей станции с системой защиты*

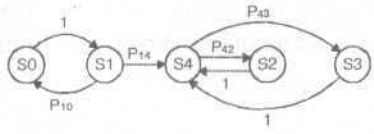


Рис. 15.8. *Граф передач СеМО*

$$p = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ P_{10} & 0 & 0 & 0 & P_{14} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & P_{42} & P_{43} & 0 \end{vmatrix}$$

Используя эту матрицу, получаем систему уравнений, связывающую между собой интенсивности потоков в узлах СеМО и вероятности передач:

$$
\begin{aligned}
-\lambda_0 + P_{10}\lambda_1 &= 0 \\
\lambda_0 - \lambda_1 + \lambda_2 + \lambda_3 &= 0 \\
P_{42}\lambda_4 - \lambda_2 &= 0 \\
P_{43}\lambda_4 - \lambda_3 &= 0 \\
P_{14}\lambda_1 - \lambda_4 &= 0
\end{aligned}
$$

Решая эту систему уравнений, получаем выражение для определния коэффициентов передач:

$$a/ = \lambda_i / \lambda_0,$$

где $i = 1...4$ — номер СМО:

$$\alpha_1 = \frac{1}{P_{10}}; \alpha_4 = \frac{P_{14}}{P_{10}}; \alpha_2 = \frac{P_{42}}{P_{10}}; \alpha_3 = \frac{P_{43}}{P_{10}}.$$

Для нахождения с помощью данных выражений численных значений коэффициентов передач необходимо определить вероятности перехода заявки из одного узла СеМО в другой.

Очевидно, что:

$$P_{10i} = 1 / N_i, \text{ откуда : } \alpha_{1i} = N_i,$$

где: Ж - количество этапов счета;

$i = 1...I$ - - тип приложения (в нашем случае I = 5).

Исходя из этого, и так как:

$$P_{14} = 1 - P_{10} = 1 - 1/N_i, \text{ то } \alpha_{4i} = N_i - 1.$$

Вероятность перехода заявки в СМО «файловая система» есть отношение количества обращений к файловой системе к количеству этапов счета, то есть:

$$P_{42i} = \frac{n_{2i}}{N_i},$$

откуда:

$$\alpha_{2i} = \frac{n_{2i}}{N_i} \cdot N_i = n_{2i},$$

где: n_{2i} количество обращений к ФС;

Ni — количество этапов счета;

$i = 1...I$ - - тип приложения (в нашем случае $I = 5$).

Аналогично определяется вероятность перехода заявки в СМО «Реестр»:

$$P_{32i} = \frac{n_{5i}}{N_i},$$

откуда:

$$\alpha_{3i} = \frac{n_{3i}}{N_i} \cdot N_i = n_{3i},$$

где: n_{3i} - количество обращений к ФС;
N_i — количество этапов счета;
$i = 1...I$ - тип приложения (в нашем случае $I = 5$).

Принимая во внимание все сказанное выше, получаем для проведения дальнейших расчетов следующее выражение, позволяющее определять среднее время обслуживания в СМО «Системы защиты»:

$$b_4 = 180 \cdot \frac{\left(0{,}9\frac{n_{3i}}{N_i^T} + 0{,}34\frac{n_{2i}}{N_i^T}\right)}{2}.$$

Таким образом, можно сделать вывод, что влияние на загрузку вычислительной системы при реализации механизма управления доступом к ресурсам пропорционально количеству решаемых задач и обратно пропорционально трудоемкости задачи. Поэтому наибольшее влияние в данном случае оказывается при загрузке системы задачами с низкой трудоемкостью этапа счета и большим количеством обращений к ресурсам при выполнении задачи.

15.2.3. Анализ эффективности механизма управления доступом

Мы рассмотрели три модели:

- модель без системы защиты;
- модель с управлением доступом к ресурсам для субъекта доступа «ПОЛЬЗОВАТЕЛЬ»;
- модель с управлением доступом к ресурсам для субъектов доступа «ПОЛЬЗОВАТЕЛЬ» и «ПРОЦЕСС».

Характеристика времени пребывания заявки в системе для рассматриваемых типовых приложений, получаемая подстановкой экспериментально измеренных данных в приведенные выражения, представлена на рис. 15.9...15.12.

Из приведенных зависимостей можно сделать следующие выводы:

1. Эффективность использования вычислительного ресурса при реализации механизма управления доступом незначительно зависит от используемых приложений (типа используемого приложения).

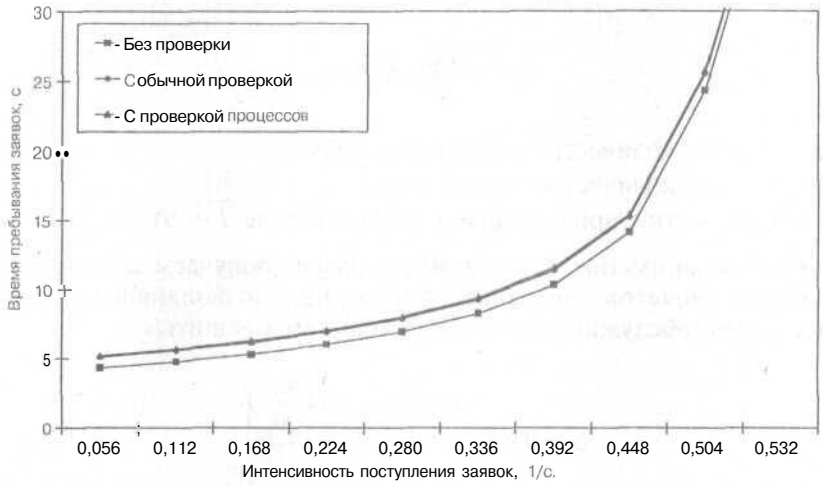


Рис. 75.9. *Характеристики моделей с потоком заявок MS Word*

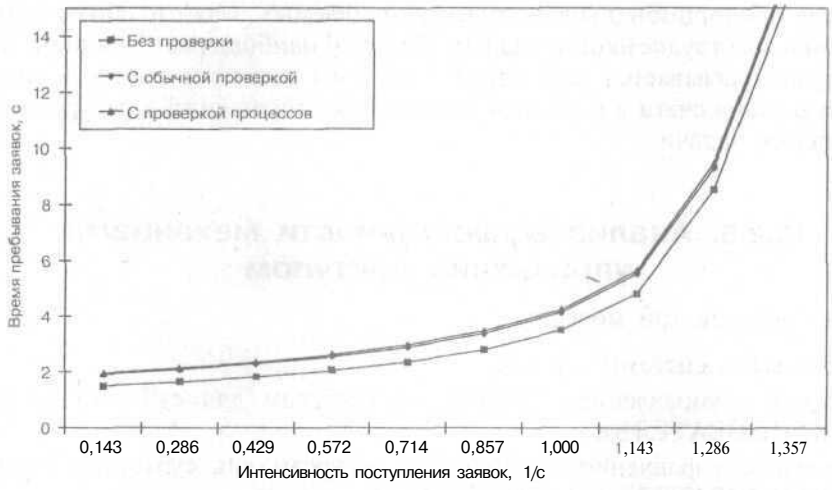


Рис. 15.10. *Характеристики моделей с потоком заявок MS Excel*

2. Дополнительное повышение загрузки вычислительного ресурса системы, вызванное использованием в схеме управления доступом к ресурсам дополнительного субъекта «ПРОЦЕСС», составляет единицы и доли единицы процентов, что является весьма низкой ценой за совокупность принципиально новых свойств защиты, обеспечиваемых при использовании в схеме управления доступом к ресурсам субъекта доступа «ПРОЦЕСС».

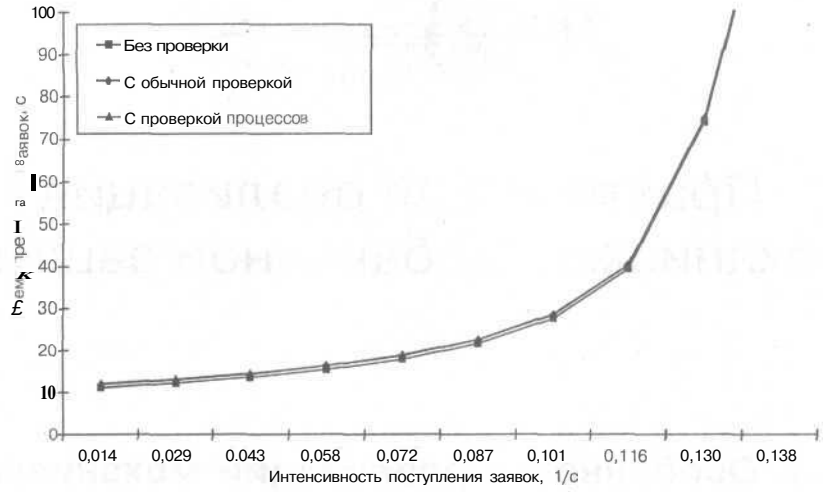


Рис. 15.11. *Характеристики моделей с потоком заявок TEX*

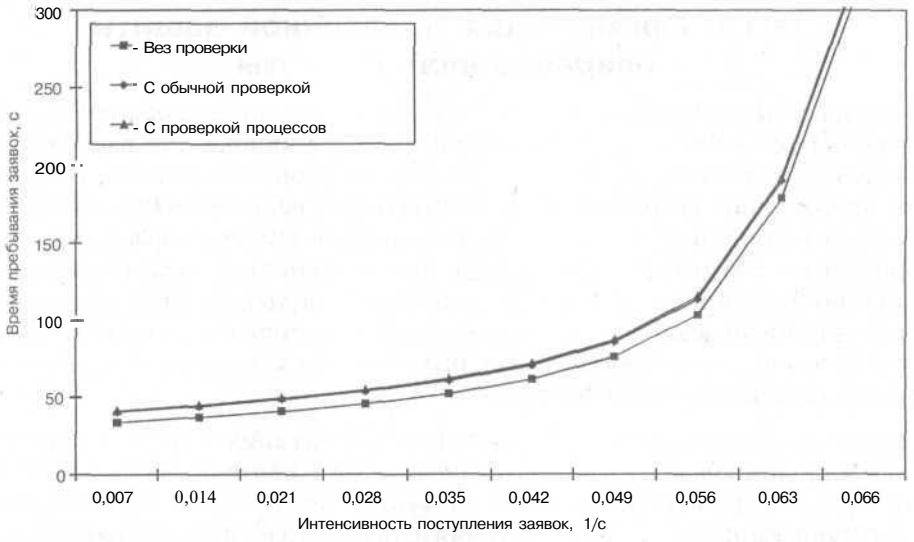
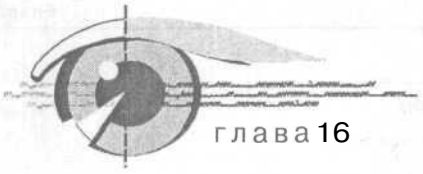


Рис. 15.12. *Характеристики моделей с потоком заявок MS VC*

3. Влияние на загрузку вычислительной системы при реализации механизма управления доступом к ресурсам пропорционально количеству решаемых задач и обратно пропорционально трудоемкости задачи. Поэтому наибольшее влияние в данном случае оказывается при загрузке системы задачами с низкой трудоемкостью этапа счета и большим количеством обращений к ресурсам при выполнении задачи.



Практическая реализация механизмов добавочной защиты

16.1. Особенности реализации механизмов добавочной защиты

16.1.1. Организация добавочной защиты операционной системы

Механизм управления доступом к ресурсам (диспетчер доступа) должен выполняться в виде системного драйвера и функционально располагаться между приложением и ядром ОС. Это вызвано следующим. На уровне приложения невозможно осуществить разграничительную политику доступа в принципе. Здесь можно только регистрировать факт уже свершившегося события и осуществлять противодействие, если событие несанкционированное. Реализация диспетчера на уровне ядра, во-первых, практически невозможна для коммерческих систем (без наличия исходных текстов), а во-вторых, может быть связана с нарушением лицензионных соглашений разработчика.

Любой запрос на доступ к ресурсам от приложения к ядру перехватывается диспетчером доступа и анализируется в соответствии с заданными правами доступа к объектам от субъектов. В случае противоречия запрашиваемого доступа разграничительной политике, он отвергается диспетчером, в противном случае доступ к ресурсу разрешается — транслируется в ядро ОС.

При наличии в системе встроенного диспетчера доступа механизм добавочной защиты должен располагаться перед ним и первым анализировать запрос. Это вызвано необходимостью реализации перенаправления запросов при доступе к общим каталогам («Корзина» и т.д.). Т.е. обработка запроса на доступ к ресурсам в этом случае должна осуществляться по схеме, приведенной на рис. 16.1.

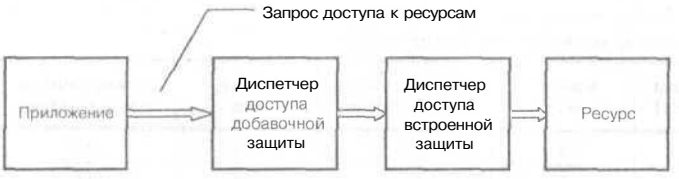


Рис. 16.1. *Схема обработки запроса доступа к ресурсам*

Особенностью реализации механизма управления доступом (диспетчера доступа) к ресурсам добавочной защиты является то, что он полностью должен быть развязан с соответствующим механизмом встроенной защиты. В частности, не должен использоваться механизм хранения атрибутов доступа файловой системой (например, NTFS). В противном случае настройки механизма добавочной защиты могут быть изменены средствами настройки встроенного механизма защиты. Кроме того, в этом случае невозможно говорить о резервировании механизмов защиты.

Остается единственный способ хранения прав доступа к ресурсам — матрицы доступа (либо соответственно, меток конфиденциальности). Этот способ предусматривает хранение прав доступа в файле (либо в реестре), доступ к которому также соответствующим образом разграничивается. Кстати говоря, реализация данного подхода позволяет строить диспетчер доступа с едиными принципами построения для различных типов файловых систем и для различных видов ресурсов. Принципиальным отличием хранения матрицы доступа в файле является то, что на производительности системы сказывается величина таблицы прав разграничения доступа, анализируемой при каждом доступе к ресурсам.

Рассмотрим, в какой мере сказывается данное влияние (на примере хранения таблицы разграничения прав доступа в файле), а также возможные подходы к его уменьшению.

16.1.2. Оценка влияния, оказываемого на вычислительную систему добавочными механизмами защиты

Для исследования влияния на загрузку вычислительного ресурса выберем приложение MS Excel, имеющее наибольший относительный показатель потери производительности, и выберем среднюю интенсивность поступления заявок, полученную для MS Excel при расчете характеристики моделей.

В табл. 16.1 представлены экспериментальные значения времени, затрачиваемого на проверку одного запроса к файловому объекту.

Экспериментальные значения времени, затрачиваемого на проверку одного запроса к файловому объекту

Таблица 16.1

Длина списка (число строк)	Время, мкс	Длина списка (число строк)	Время, мкс	Длина списка (число строк)	Время, мкс
10	9	300	278	550	509
50	46	350	324	600	556
100	92	400	370	650	602
150	139	450	417	700	648
250	231	520	481	750	694

На рис. 16.2 приведен график роста влияния на загрузку вычислительной системы при увеличении списка (строк в списке) правил разграничения доступа в рамках реализации диспетчера доступа добавочными средствами защиты (при фиксированной интенсивности поступления заявок на обслуживание). Результаты исследований получены с использованием модели, представленной ранее.

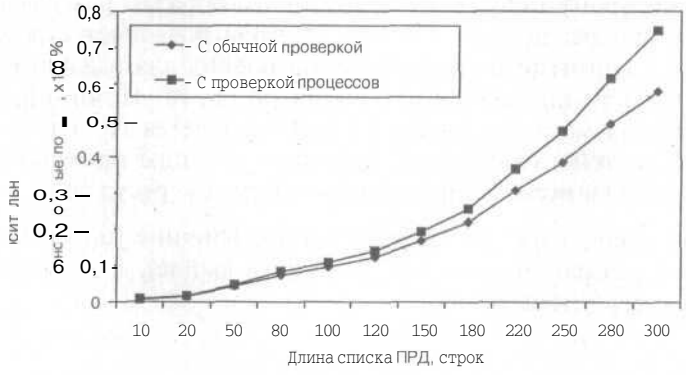


Рис. 16.2. *Влияние на загрузку вычислительной системы*

Видно, что рост влияния на загрузку вычислительной системы при контроле процессов, как самостоятельных субъектов доступа, незначителен, по сравнению с ростом общего процента потерь. Однако общие потери могут быть весьма существенны. И хотя рост времени проверки при увеличении списка правил разграничения доступа имеет линейную зависимость, рост процента потерь производительности имеет показательный характер (например, при 100 строках в таблице разграничения прав доступа потери производительности могут достигать 10%).

16.1.3. Пути уменьшения потерь производительности вычислительной системы из-за средств защиты

Так как длину списка в 100 строк в общем случае нельзя считать достаточной для задания правил разграничения доступа (в реальных системах длина списка прав разграничения доступа может достигать 150...200 строк) рассмотрим некоторые пути уменьшения потерь из-за средств защиты. Ясно, что бороться с потерями от введения средств добавочной защиты можно следующими путями:

- путем уменьшения списка правил разграничения доступа;
- путем уменьшением времени проверки одного запроса;
- сразу обоими методами.

Первый способ интуитивно очевиден: сократить длину имени ресурса (т.е. среднюю длину строки в списке правил разграничения доступа). Например, ресуры файловой системы, доступ к которым подлежит разграничению, следует располагать как можно ближе к корневому каталогу. Это естественно уменьшит длину полного имени ресурса, т.е. длину строки в списке.

Недостатком данного подхода является то, что, во-первых, данное ограничение (по-сути, разграничение доступа к каталогам) не всегда выполнимо в реальной системе, во-вторых, такой способ мало применим для остальных ресурсов вычислительной системы, например реестра или сетевых ресурсов, так как в этом случае администратор безопасности не может изменять полные имена ресурсов.

Второй способ состоит в использовании масок. Маска — это регулярное выражение, содержащее метасимволы «*», «?» и т.п., покрывающие несколько имен ресурсов сразу. Таким образом, вместо внесения некоторого количества очень похожих имен ресурсов в списки правил разграничения доступа вносится только одна маска, содержащая общую часть имен этих ресурсов и метасимволы, задающие правила последующего сравнения маски и имен ресурсов [7].

При использовании механизма масок в ОС семейства MS Windows есть своя особенность. Для таких ОС характерно, что имена файловых ресурсов не подходящие под формат «8.3» (так называемые, «длинные имена») имеют фактически два имени — длинное и короткое в формате «8.3». Это вынуждает либо вносить в списки правил разграничения доступа оба имени, либо использовать в качестве имен ресурсов в этих списках маски.

Так, например, в ОС MS Windows 95/98/Me, а также в MS Windows NT, короткое имя, получаемое из длинного, содержащего символы, не входящие в латинский алфавит, формируется по тому же принципу, что и для длинного имени, содержащего только символы латинского алфавита. То есть, например, длинному имени C:\Program Files\Абвгдежзик\abc

соответствует короткое имя C:\Progra~ 1\Абвгде~1\abc. Оба имени покрываются маской следующего вида: C:\Progra*\Абвгде*\abc.

В ОС MS Windows 2000/Xp, в случае, если длинное имя, содержащее символы, не входящие в латинский алфавит, располагается на файловой системе NTFS, короткое имя формируется с использованием четырехзначного шестнадцатеричного числа, вместо части пути, содержащего символы, не входящие в латинский алфавит. То есть, для приведенного выше длинного имени, короткое имя в таких условиях будет выглядеть так: C:\Progra~l\C6F5~l\abc.

Понятно, что попытка покрытия этих двух имен (длинного и короткого) простой маской обречена на неудачу, так как часть пути к ресурсу, содержащего символы, не входящие в латинский алфавит, выглядит различно в длинном и коротком имени (не имеет ни одного общего символа). Требуемую маску создать можно, но помимо упомянутых имен такая маска обязательно покроет еще некоторые имена ресурсов.

Подобная ситуация исправляется в случае применения более сложных регулярных выражений для масок. Однако очевидно, что проще организовывать структуру файловой системы NTFS, избегая использования длинных имен, содержащих символы, не входящие в латинский алфавит (что под силу администратору безопасности при настройке системы защиты).

Таким образом, уменьшение времени проверки одного запроса достигается применением усовершенствованного алгоритма сравнения строк. Кроме того, можно использовать различные математические методы ускорения поиска (организация списков прав разграничения доступа в виде сбалансированных деревьев и т.п.).

Учитывая все сказанное, проведем исследование потерь производительности в случае применения модифицированного алгоритма проверки прав доступа. Модификация заключается в замене процедуры посимвольного сравнения строк на процедуру с использованием простейших регулярных выражений. В табл. 16.2 представлены экспериментальные данные времени, требуемого при проверке прав доступа для одного запроса.

Экспериментальные значения времени, затрачиваемого на проверку одного запроса к файловому объекту *Таблица 16.2*

Длина списка (число строк)	Время, мкс	Длина списка (число строк)	Время, мкс	Длина списка (число строк)	Время, мкс
10	9	300	278	550	509
50	46	350	324	600	556
100	92	400	370	650	602
150	139	450	417	700	648
250	231	520	481	750	694

При сравнении данных, представленных в табл. 16.1 и 16.2, можем сделать вывод о том, что в случае применения модифицированного алгоритма проверки и сокращения длины имен ресурсов достигается почти двукратное снижение потерь производительности вычислительной системы, связанных с реализацией механизма управления доступом к ресурсам.

16.1.4. ПОДВОДЯ ИТОГИ

Обобщим сказанное ранее в плане рассмотрения преимуществ, реализуемых рассмотренными выше механизмами управления доступом к ресурсам. Общая классификация этих преимуществ приведена на рис. 16.3.

Рис. 16.3. *Преимущества, предоставляемые рассмотренными подходами*

Введение новых свойств, например, включение в схему управления доступа субъекта «процесс», как ранее показано, позволяет противодействовать целым группам скрытых угроз, например, макросам офисных приложений.

Реализация принципиально новых подходов, как отмечалось выше, связана с возможностью локализации прав стандартных и оригинальных приложений, возможностями внесения в схему мандатного разграничения субъектов и объектов доступа «ПРОЦЕСС» и т.д. При этом необходимо учитываить, что в работе отмечены лишь явные и интуитивно понятные подобные возможности, круг которых на практике может быть существенно расширен администратором безопасности с применением заложенных в рассмотренные механизмы технологий управления доступом.

В части упрощения администрирования механизмов защиты рассматривается принципиально новый подход к администрированию важнейшего механизма — обеспечения замкнутости программной среды. Посредством контроля списков процессов, разрешенных к запуску, данный механизм корректно настроить, а уж тем более модифицировать исходные настройки в процессе функционирования системы, если не невозможно, то чрез-

вычайно сложно. Рассмотренный же подход к администрированию, позволяющий администратору работать не со списками, а с каталогами, из которых разрешается запуск процессов, принципиально упрощает как процедуру исходной настройки, так и модификацию настроек в процессе функционирования системы защиты.

В качестве примера противодействия скрытым угрозам приведем пример атаки, на первый взгляд, казалось бы, никак не связанной с механизмами управления доступом к ресурсам, и рассмотрим возможность противодействия данной атаке с использованием рассмотренных подходов.

Рассмотрим одну из атак, позволяющих несанкционированно запустить исполняемый код. Атаки этого типа основываются на переполнении буфера входных данных различных программ. Атакам данного типа в основном подвержены Unix-системы, что во многом обусловливается организацией разграничения прав в подобных системах встроенными средствами.

Целью атаки является повышение привилегий злоумышленником. Это связано с тем, что некоторые программы (обычно системные демоны или программы настройки) требуют для своей работы привилегий администратора системы (root). Например, сетевые серверы ftp, http, bind и др. работают в системе с правами root.

Кроме того, программы настройки чего-либо обычно тоже требуют повышения привилегий, т.к. им зачастую необходимо модифицировать файлы в системном каталоге /etc, что позволено только root.

Атака на переполнение буфера входных данных используется для того, чтобы заставить атакуемую программу запустить другую (например, стандартную для всех Unix-систем командную оболочку /bin/sh). Атакуемая программа является программой настройки и имеет привилегии root.

В результате переполнения стека программа начинает выполнять код, занесенный в стек злоумышленником. Например, этот код может выполнять системные вызовы setuid() и exec(). Это приводит к порождению нового процесса (командной оболочки) также с правами root, что позволяет злоумышленнику отдавать команды системе уже от имени root.

Противодействовать подобным атакам с использованием встроенных механизмов защиты очень сложно. Необходимо постоянно обновлять программное обеспечение, где обнаруживаются уязвимости такого типа. И все равно, даже в обновленной версии программ, нет гарантии, что все ошибки, приводящие к переполнению буфера входных данных, исправлены. Значительно усложняет ситуацию и то, что с помощью стандартных, т.е. входящих в состав ОС, средств разграничения прав доступа, невозможно ограничить набор доступных для привилегированного процесса ресурсов.

Реализация рассмотренного метода проверки запросов к ресурсам вычислительной системы позволяет исключить в принципе возможность по-

добных атак. Так как в рамках данного метода возможно управлять правами конкретного процесса, то становится возможным ограничить набор доступных ресурсов для привилегированного процесса.

Таким образом, указанием в списке доступных для процесса ресурсов имен только тех ресурсов, доступ к которым ему необходим, автоматически исключается возможность осуществить с помощью этого процесса НСД к другим ресурсам, в том числе, и с целью выполнения.

Например, для противодействия возможности рассматриваемой атаки достаточно указать в списке доступных для программы ресурсов только файлы, доступ к которым ей необходим. Если такие файлы неизвестны, то можно разрешить доступ программе (/usr/openwin/bin/kcms_configure) только в каталоги настроек (/etc, /usr/openwin/etc), тем самым исключая доступ к каталогам, содержащим исполняемые файлы.

Таким образом, исключается не только возможность запустить с помощью такой атаки некоторый процесс с привилегиями root, но также осуществить НСД к другим объектам системы. Действительно, поскольку атакуемая программа обладает неограниченными привилегиями (в случае использования встроенных в ОС механизмов защиты), можно с помощью атаки подобного типа осуществить НСД к ресурсам, недоступным злоумышленнику по существующим разграничениям прав доступа. И тем же способом, то есть ограничивая доступ программе только файлами, непосредственно принадлежащими ей, исключаются и атаки, использующие переполнение буфера для НСД к ресурсам системы.

Мы привели только один пример противодействия целой группе атак с целью иллюстрации эффективности рассматриваемых подходов. Однако данный пример, с одной стороны, иллюстрирует эффективность рассматриваемых подходов, с другой стороны, показывают и потенциальные сложности, связанные с противодействием скрытым атакам.

Кроме того, на этом примере иллюстрируется следующая идея: любая система защиты, в том числе и добавочной, предлагает администратору безопасности лишь набор механизмов защиты. Эффективность же использования возможностей механизмов защиты является важнейшей задачей администратора безопасности.

Другими словами, защищенность системы определяется не только возможностями системы защиты, но и квалификацией администратора безопасности, настраивающего данную систему -- реализующего возможности системы защиты. При этом в функции администратора должно быть включено не только решение задачи реализации разграничительной политики доступа к ресурсам, предоставляемыми возможностями системы защиты, но и настройка системы защиты с целью противодействия возможным угрозам, в том числе скрытым.

В нашей работе рассматриваются лишь общие подходы к построению системы защиты, определить же требования к настройке рассматриваемых механизмов защиты для противодействия известным группам атак, используя данные материалы, интересующемуся читателю предлагается самостоятельно.

16.2. Метод централизации схемы администрирования встроенных и добавочных механизмов защиты

16.2.1. Способы локализации настроек диспетчера доступа

Ранее отмечалось, что одним из важнейших требований к реализации диспетчера доступа является санкционированное изменение учетной информации субъектов и объектов доступа, а также правил разграничения доступа. При настройке механизмов управления доступом, рассмотренных выше, способы локализации настройки диспетчера доступа администратором безопасности достаточно очевидны. Они включают в себя выполнение следующих условий:

* интерфейс настроек механизмов управления доступом должен открываться при запуске соответствующего приложения только после авторизации администратора безопасности;
* средствами разграничения прав доступа к исполняемым файлам (механизм обеспечения замкнутости программной среды) запуск данного приложения, предоставляющего интерфейс настроек механизмов управления доступом, должен быть разрешен только администратору безопасности;
* средствами разграничения прав доступа к файловым объектам и к реестру ОС доступ к настройкам механизмов защиты (по крайней мере, «на запись») должен быть разрешен только следующим субъектам: «пользователю» — администратору безопасности; «процессу» — интерфейсу настроек механизмов управления доступом. При этом должны использоваться одновременно разграничения и по пользователям, и по процессам.

При выполнении данных мероприятий иной пользователь системы (не администратор безопасности) не сможет получить доступ ни к интерфейсу настроек диспетчера доступа, ни к файлам (ключам реестра) настроек диспетчера доступа.

16.2.2. Проблемы централизации схемы администрирования

Серьезные проблемы возникают при построении схемы администрирования приложений, которые требуют настроек собственным администратором (ввиду сложности и специфики их администрирования) и которые имеют встроенные механизмы защиты информации. В качестве примера таких приложений можно рассмотреть СУБД. С одной стороны, возлагать функции по администрированию СУБД на администратора безопасности, наверное, нецелесообразно. С другой стороны, СУБД вносит новый объект доступа — «ТАБЛИЦА», разграничение к которому реализуется встроенными механизмами защиты СУБД и администратор должен иметь контроль над ними.

Примечание

Корректное сопряжение защитных механизмов системы и встроенных ™ защитных механизмов СУБД было рассмотрено в п. 14.6.2.

16.2.3. Метод централизации схемы администрирования

Для решения задачи централизации схемы администрирования может использоваться следующий подход, реализованный в КСЗИ «Панцирь». Пусть в общем случае в сложной информационной системе присутствует несколько уровней иерархии, к которым можно отнести системный уровень, уровень СУБД и уровень приложений. На каждом уровне решаются задачи управления доступом к информационным ресурсам собственными средствами — для каждого уровня реализован собственный диспетчер доступа (например, к таблицам на уровне СУБД), администрирование которыми осуществляется администраторами соответствующих уровней — системным администратором, администратором СУБД, администратором приложений (некоторые задачи, например, создание разделяемого ресурса для ОС Windows 95/98 вообще решаются пользователем). Сказанное проиллюстрировано рис. 16.4.

Для обеспечения информационной безопасности сложной системы должна обеспечиваться централизация администрирования средствами защиты информации. Однако в данном случае это невозможно, ввиду того, что централизация может быть достигнута только в случае, если администратор безопасности будет сам осуществлять администрирование безопасностью системы на всех ее уровнях иерархии. Это неприемлемо сложная задача и не всегда достигаемая в принципе. При децентрализации схемы администрирования информационной безопасности снижается ее защищенность (администраторы соответствующих уровней иерархии

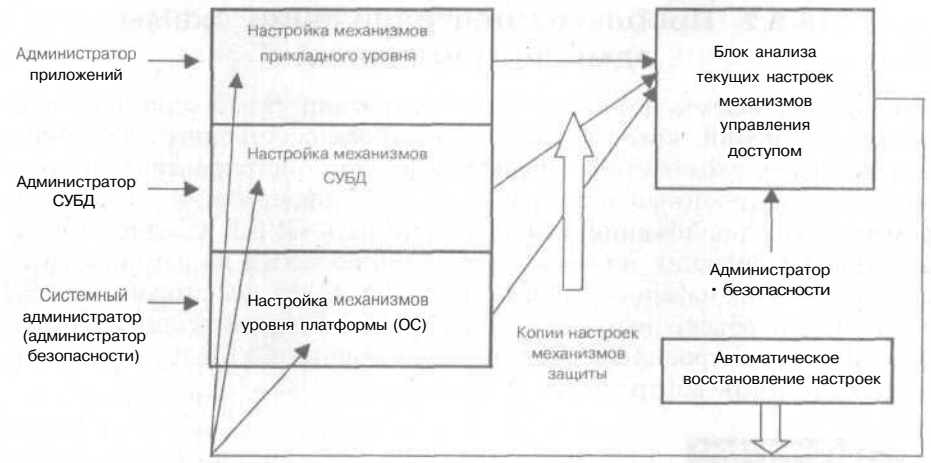


Рис. 16.4. *Иллюстрация распределения задач администрирования информационной безопасностью*

имеют возможность управлять безопасностью на своих уровнях бесконтрольно со стороны администратора безопасности).

Идея метода централизации схемы администрирования состоит в следующем [9, 17]: администратором безопасности, который должен осуществлять администрирование системы на нижнем уровне — уровне платформы (ОС) формируются эталонные копии настроек механизмов защиты (в частности, таблиц настроек безопасности СУБД — таблицы пользователей и паролей, таблицы ролей и т.д.). Эти эталонные копии настроек хранятся в блоке анализа текущих настроек механизмов управления доступом (в соответствующих объектах, доступ к которым разрешен только администратору безопасности). В процессе функционирования системы синхронно (с заданным интервалом) анализируются текущие настройки механизмов. При этом текущие настройки сравниваются эталонными (например, при помощи контрольных сумм). Если обнаруживается расхождение, то текущие настройки восстанавливаются из эталонных копий, что предотвращает их несанкционированное изменение.

Таким образом, достигается следующий результат — на уровне СУБД и приложений настройку механизмов управления доступом осуществляют администраторы соответствующих уровней, но введение их в действие (равно как и любое изменение) возможно только при участии администратора безопасности, осуществляющего настройки на нижнем уровне - уровне платформы, т.к. изменения настроек могут вступить в силу только после отключения администратором безопасности механизма автоматического восстановления настроек из эталонных копий.

16.2.4. Схема централизации администрирования сложного иерархического объекта

На рис. 16.5 приведена схема централизации администрирования механизмами управления доступом в сложной иерархической системе (имеющей несколько уровней администрирования).

Работает схема следующим образом. Перед началом работы пользователь должен пройти авторизацию со входа 10, которая осуществляется блоком 1. Параметры авторизации — имя и пароль пользователя блок 1 запрашивает и получает от блока 5. Данные текущего пользователя блоком 1 выдаются в блок 4, которым запрашивается в блоке 5 таблица разграничения прав доступа зарегистрированного в системе пользователя. При запросе пользователем доступа к информационным ресурсам со входа 11 (запрашивается объект доступа, например, файл и действия — чтение или запись и т.п.) блок 4 анализирует заданные для пользователя права доступа и запрашиваемые пользователем параметры доступа. Если они не противоречивы — запрашиваемый доступ системой разрешен. При этом вырабатывается сигнал разрешения доступа к информационному ресурсу, который поступает в блок 7.

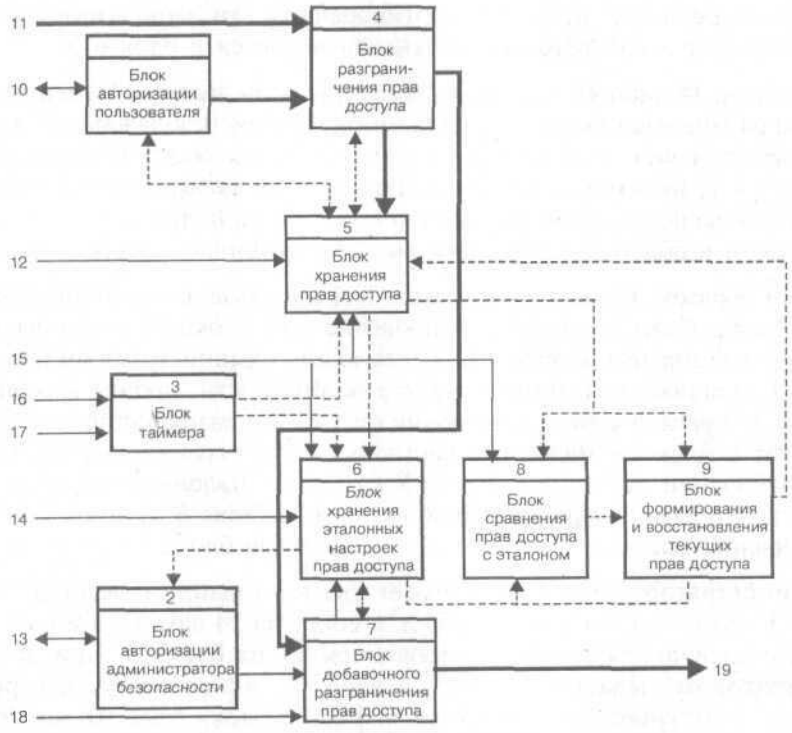


Рис. *16.5. Схема централизации администрирования механизмами управления доступом*

Со входа 12 задаются параметры прав доступа. Изменять данные права разрешается пользователю, имеющему соответствующие полномочия (в зависимости от иерархического уровня системы это может быть как собственно пользователь, так и один из администраторов: системный, СУБД, приложения). При этом должен быть выработан соответсвующий сигнал от блока 4, формируемый после авторизации пользователя и его запроса (со входа 11) на получение доступа к блоку 5 (блок *5* в общем случае также является файловым объектом).

Аналогично работает схема авторизации и разграничения доступа администратора безопасности. Перед началом работы администратор безопасности должен пройти авторизацию со входа 13, которая осуществляется блоком 2. Параметры авторизации — имя и пароль администратора безопасности блок 2 запрашивает и получает от блока 6. Данные текущего пользователя (администратора безопасности) блоком 2 выдаются в блок 7, которым запрашивается в блоке 6 таблица разграничения прав доступа зарегистрированного в системе пользователя (администратора безопасности).

При запросе пользователем (администратором безопасности) доступа к информационным ресурсам со входа 18 блок 7 анализирует заданные для пользователя права доступа и запрашиваемые пользователем параметры доступа. Если эти права не противоречивы, то запрашиваемый доступ разрешается и соответсвующий сигнал подается в блок 6.

Со входов 14 либо 15 задаются параметры прав доступа. Изменять данные права разрешается пользователю (администратору безопасности), имеющему соответствующие полномочия — если выдается соответствующий сигнал от блока 7, формируемый после авторизации пользователя (администратора безопасности) и его запроса (со входа 18) на получение доступа к блоку 6. Блок 6 в общем случае также является файловым объектом).

Таким образом, блоки 1, 4, 5 служат для контроля доступа пользователей к файловым объектам. Администрирование этих блоков осуществляется либо самим пользователем, либо соответствующим администратором (системным, СУБД, приложений). Администратор безопасности, пройдя авторизацию в блоке 2, в рамках своих полномочий (его доступ разграничивается блоком 7) создает в блоке 6 эталонные настройки либо ограничения на настройки разграничений, заданных в блоке 5. Создание эталонных настроек предполагает, что администратором безопасности в блоке 6 задается таблица разграничений доступа, являющаяся эталоном для блока 5.

Администратор безопасности имеет две возможности задания таких настроек: либо занесением их в блок 6 со входа 14 самостоятельно (например, с клавиатуры), либо копированием их из блока 5. При этом администратором выдается сигнал на вход 15, по которому настройки из блока 5 перезаписываются в блок 6. Работа блока 8 на это время блокируется. Другой режим — это задание ограничений на возможные в блоке 5 настройки. Например, разрешить пользователю для разделения только

какой-либо диск, каталог, файл, либо, наоборот, запретить пользователю для разделения диск, каталог, файл и т.д.

Данные ограничения также в блок *6* выдаются со входа 14 после обработки соответствующего запроса администратора безопасности со входа 18 (соответственно, происходит авторизация администратора и контролируется его доступ блоком 7).

Итак, администратором безопасности в блоке 6 создаются эталонные настройки (с клавиатуры — со входа 14, либо копированием из блока 5 - со входа 15) либо ограничения на разграничения прав доступа, в соответствии с которыми должен обрабатываться доступ пользователей к информационным ресурса со входа 11.

Далее администратором безопасности со входа 16 задается режим контроля настроек (интервал выдачи сигналов контроля таймером) и со входа 3 запускается таймер — блок 3. Сигналами с выхода блока 3 в блок 8 с выхода блока 5 и с выхода блока 6 заносятся текущие параметры разграничения доступа (из блока 5) и эталонные настройки, либо ограничения (с блока 6).

Блок 8 осуществляет сравнение текущих и эталонных настроек либо выполнение текущими настройками ограничений, задаваемых в блоке 6. При обнаружении некорректности текущих настроек, блок 8 выдает об этом сигнал в блок 9. Блок 9, получая текущие настройки из блока 5 и эталонные настройки либо ограничения на настройки из блока 6, корректирует текущие настройки в соответствии с разграничениями, задаваемыми администратором безопасности, и заносит их в блок 5 на место некорректных текущих настроек. При занесении настроек в блок 6 таймер (блок таймера 3) отключается со входа 17.

Ввиду того, что блок 6 также представляет собою файловый объект, разрешение доступа пользователя к информационному ресурсу с первого выхода блока 4 проходит дополнительное разграничение на блоке 7 (блок 7 запрещает корректировку информации в блоке 6 всем, кроме администратора безопасности). В случае корректного запроса доступа пользователем блоком 7 вырабатывается сигнал на выход 19.

В общем случае приведенная схема позволяет реализовать различные уровни компромисса в централизации администрирования информационной безопасностью в иерархической информационной системе.

Возможны следующие режимы функционирования вышеприведенной схемы администрирования:

* **Все управление безопасностью осуществляется администратором безопасности.** В этом случае он самостоятельно (со входа 14) заносит в блок 6 эталонные настройки доступа. После этого запускает таймер (блок 3) со входа 17. Система переходит в режим контроля текущих

настроек разграничения доступа в блоке 5 — заносит туда настройки из блока 6 при первом обнаружении несовпадения, фиксируемом блоком 8. При этом любые изменения настроек блока 5 будут автоматически восстановлены. Интервал времени, через который необходимо осуществлять проверки, задается со входа 16. Малое значение этого интервала не позволит возможным несанкционированным изменениям вступить в действие.

* **Администратором безопасности решаются задачи контроля и противодействия несанкционированному изменению настроек безопасности.** В этом режиме администратор безопасности после соответствующего визуального контроля настроек, заданных пользователем либо соответствующим администратором, в блоке 5, переносит данные настройки сигналом со входа 15 в блок 6 — эти настройки считаются корректными (эталонными). Затем запускает подсистему контроля настроек, запустив блок таймера 3 со входа 17.

  Система переходит в режим контроля, при котором без участия администратора безопасности невозможно изменить настройки разграничения доступа к информационным ресурсам, хранящиеся в блоке 5, в том числе и легальным пользователем, либо администратором в рамках разграничений задаваемых блоками 1 и 4 (т.е. любые изменения, проводимые без ведома администратора безопасности будут немедленно устранены).

  Для изменения эталонных настроек необходимо отключить блок таймера 3 (со входа 17), изменить настройки таблицы разграничений прав доступа в блоке 5 (соответствующим пользователем или администратором, в рамках полномочий, заданных блоками 1 и 4, в случае необходимости, при визуальном контроле со стороны администратора безопасности) администратором безопасности, после чего (блоком 2) со входа 15 переписать в блок 6 новые корректные настройки. Затем запускается блок 3, система переходит в штатный режим.

* **Администратором безопасности вносятся ограничения на возможности задания параметров безопасности.** В этом режиме предполагается, что администратором безопасности в блоке 6 задаются не эталонные таблицы настроек для блока 5, а ограничения на настройки. Возможны два режима ограничений — запрещение и разрешение.

  При реализации режима запрещения вносятся запреты на какие-либо действия, связанные с доступом к информационным ресурсам, например, запретить доступ к какому-нибудь файловому объекту или таблице, запретить возможность разделения какого-либо ресурса (например, создать общую папку) запретить доступ к файловому объекту на запись и т.д. Т.е. данный режим реализует правило: все, что не запрещено администратором безопасности — разрешено в системе.

  Другой режим реализует альтернативное правило: все, что не разрешено администратором безопасности, запрещено в системе. В этом слу-

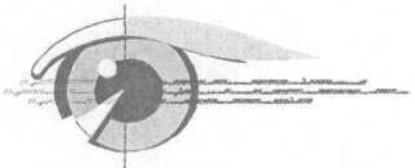
чае администратором безопасности задаются разрешения, в рамках которых уже может осуществлять свои разграничения пользователь или иной администратор. Например, администратор безопасности может разрешить доступ только к одному логическому диску. Доступ к каталогам и файлам, расположенным на этом диске, может разграничивать соответствующий пользователь или администратор со входа 12.

В данном режиме ограничения заносятся администратором безопасности (после его авторизации блоком 2) в блок 6, после чего со входа 17 запускается блок таймера 3. Текущие настройки, расположенные в блоке 5, и ограничения из блока 6 с интервалом, заданным со входа 16, поступают в блок 8, который анализирует, не противоречат ли настройки в блоке 5 ограничениям, задаваемым блоком 6. Если противоречат, то блок 9 осуществляет их корректировку и заносит в блок 5 корректные настройки.

Рассмотренный метод, основу которого составляет контроль неизменности настроек механизмов защиты всех уровней иерархии, с функцией их автоматического восстановления из эталонной копии, позволяет ввести уровень администратора безопасности системы, являющийся центральным звеном схемы администрирования, обеспечивающий возможность только санкционированного изменения настроек механизмов защиты в процессе функционирования защищаемого объекта.

Контроль корректности

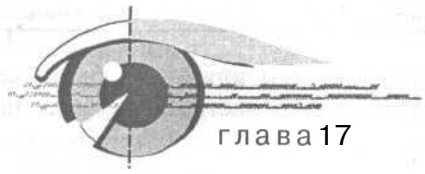
функционирования механизмов защиты. Методы контроля целостности



Часть V

- Метод уровневого контроля списков санкционированных событий
- Разработка и оптимизация механизма уровневого контроля, как механизма реального времени

4 Механизмы контроля целостности файловых объектов



глава 17

Метод уровневого контроля списков санкционированных событий

Ранее в книге рассматривались механизмы разграничительной политики доступа к ресурсам. Обсуждались различные приемы. Вырабатывалось оптимальное решение, а также рекомендации и требования к нему. Однако в общем случае задача защиты должна строиться в предположении потенциальной уязвимости механизмов защиты, реализующих разграничительную политику доступа к ресурсам защищаемого объекта. Исходя из этого, в системе защиты должны быть предусмотрены следующие механизмы:

- **Контроль (мониторинг) корректности функционирования механизмов защиты.** Этот контроль должен позволять анализировать корректность работы как самих механизмов защиты, так и корректность работы пользователя с этими механизмами, что позволит уже на предварительном этапе вырабатывать реакцию системы защиты и отсекать попытки несанкционированных действий пользователей по отношению к ней.
- **Контроль целостности файловых объектов.** Если предыдущий механизм защиты призван препятствовать преодолению разграничительной политики доступа, то данный механизм исходит из того, что эта политика была преодолена. В соответствии с этим контроль целостности призван свести к минимуму возможные потери от свершившегося НСД. При этом он контролирует целостность заданных файловых объектов и в случае необходимости может восстановить файловые объекты в соответствии с эталонами.

С целью решения данной задачи в функциональную модель системы защиты (см. главу 2) введен дополнительный уровень защиты. Этот уровень как раз и призван осуществлять контроль (мониторинг) корректности функционирования механизмов управления доступом, а также контроль целостности файловых объектов.

Кстати, вышеуказанные механизмы могут применяться как для усиления механизмов добавочной защиты, так и для усиления встроенных в ОС

механизмов защиты. Ведь и те и другие в общем случае могут иметь ошибки в реализации и ошибки в настройке при их администрировании.

17.1. Основы метода уровневого контроля списков санкционированных событий

В качестве возможного способа противодействия ошибкам и закладкам в системном и прикладном ПО, а также контроля (мониторинга) корректности функционирования механизмов, реализующих разграничительную политику доступа к ресурсам защищаемого объекта, рассмотрим **метод уровневого контроля списков санкционированных событий (МУКССС).**

Ранее было сказано, что контроль корректности должен реализовывать как контроль функционирования самой системы защиты, так и контроль за действиями пользователей. Рассмотрим, как оба этих направления реализуются предлагаемым методом уровневого контроля списков.

17.1.1. Контроль за действиями пользователей

В общем случае получение пользователем доступа к ресурсу можно представить набором некоторых событий. Например, какой пользователь обращается к ресурсу, какой процесс запущен пользователем, к какому ресурсу обращается пользователь и т.д.

Таким образом, доступ пользователя к ресурсу представляет собой некоторую последовательность действий, каждое из которых может быть охарактеризовано соответствующим списком санкционированных (разрешенных), либо несанкционированных (запрещенных) событий. Такими списками, например, могут быть:

* список зарегистрированных в системе пользователей;
* список разрешенных к запуску процессов;
* список запрещенных событий в системе;
* список ключей реестра ОС, разрешенных для изменения;
* список устройств, к которым разрешен доступ пользователям, и т.д.

Анализируя известные атаки, прежде всего, с использованием знания ошибок в ПО, можно показать, что большинство преодолений политики доступа к ресурсам осуществляется с использованием события, которое не определено как санкционированное. Таким образом, контролируя текущие события системы в соответствии со списками разрешенных событий (списками санкционированных событий), можно контролировать корректность работы пользователя в системе. При этом в случае выполнения пользователем несанкционированных действий будет вырабатывать-

ся соответствующая реакция системы защиты: завершение несанкционированного процесса, завершение сеанса несанкционированного пользователя в системе и т.п.

Отличительной особенностью рассматриваемого подхода является то, что в принципе неважно, каким образом злоумышленник пытается осуществить несанкционированный доступ к информации, т.к. фиксируются не конкретные описания, а косвенные признаки атаки -- несанкционированные для системы события. Принципиально важным является то, что рассматриваемый подход позволяет противодействовать скрытым угрозам, а также использованию злоумышленником ошибок и закладок в системном, функциональном и прикладном программном обеспечении без необходимости выявления последних.

17.1.2. Контроль корректности функционирования системы защиты

Контроль корректности функционирования системы защиты производится так же как и контроль за действиями пользователей. Только при этом регламентируются действия не пользователей, а самой системы защиты. При этом также составляются списки санкционированных событий-эталонов, с которыми периодически сравниваются текущие события. Система защиты считается нарушенной в случае расхождения эталонной копии и оригинала. В ответ на это вырабатывается соответствующая реакция системы.

Для системы защиты также составляется и контролируется список обязательных событий. В общем случае нарушение этого списка также расценивается как нарушение защиты и влечет за собой ответную реакцию системы.

17.1.3. Общие принципы построения и функционирования механизма уровневого контроля списков санкционированных событий

Общие принципы реализации уровневого контроля

Механизм уровневого контроля базируется на расширении технологии аудита. Происходящие (зарегистрированные) события анализируются автоматически (без участия администратора) в реальном масштабе времени. В зависимости от результатов анализа принимаются те или иные решения, влияющие на функционирование системы, то есть вырабатывается реакции системы защиты.

Как уже отмечалось, изначально в системе выделяются те события, которые нужно контролировать. При большом количестве таких событий они объединяются в группы (списки) по функциональному признаку или

другому общему свойству. Затем создается эталонная копия свойств данной группы (списка). Таким образом, каждый из списков формирует отдельный уровень системы. Свойства системы считаются нарушенными в случае расхождения эталонной копии и оригинала.

В процессе работы защищаемой системы осуществляется ее периодический контроль путем сравнения эталонных копий списков и их реальных значений. Проверка списков осуществляется последовательно. При этом проверка следующего уровня в общем случае зависит от результатов проверки на предыдущем уровне. В случае расхождения оригинала с эталоном генерируется состояние нарушения безопасности системы, реакцией на которое может быть восстановление нарушенного списка или выполнение какого-либо другого сценария, предусмотренного в системе.

Уровневые списки (группы) выбираются и составляются таким образом, чтобы контроль, по возможности, не требовал большого количества вычислительных ресурсов. Также условием выбора и составления списков является косвенная (реже прямая) зависимость безопасности системы от события, включаемого в список.

В общем случае нарушение события по одному конкретному списку может еще не являться прямым доказательством нарушения безопасности системы, но указывает на повышение вероятности такого события. Таким образом, при быстрой проверке косвенных признаков нарушения безопасности системы, в случае обнаружения расхождений с проверяемым списком санкционированных событий можно немедленно запустить внеочередную проверку других более важных для безопасности системы событий. Обычно проверка таких событий требует больших затрат производительности и производится уже при фиксировании каких-либо некорректностей в функционировании системы.

При правильном выборе уровневых списков и их взаимодействия (зависимости проверок) можно утверждать, что если при проверках уровневых списков не выявлено расхождений, то безопасность системы соблюдена на данный момент времени.

Таким образом, **уровневый контроль списков санкционированных событий представляет собой подход к защите информации, основанный на определении событий, происходящих при доступе к ресурсу, выявлении списков санкционированных событий и непрерывном контроле происходящих в процессе функционирования системы событий на соответствие заданным эталонным спискам.**

Обобщенная схема механизма уровневого контроля

Обощенная иллюстрация реализации механизма уровневого контроля [9, 10, 16, 17] представлена на рис. 17.1.

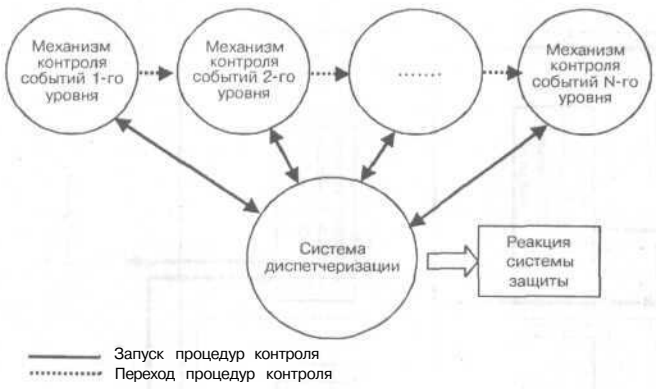


Рис. 17.1. *Обобщенная схема реализации механизма уровневого контроля*

Схема реализации рассматриваемого механизма защиты состоит из трех основных типов блоков:

- Блок (блоки) контроля списков санкционированных событий соответствующих уровней. Число таких блоков определяется числом контролируемых событий. Механизмы, реализуемые данными блоками, могут существенно различаться.
- Блок диспетчеризации, который реализует управление схемой с учетом реализуемой дисциплины обслуживания (либо иных условий). В его функции входит активизация соответствующих механизмов контроля.
- Блок реакции системы защиты, который осуществляет выработку реакции при обнаружении расхождения текущих событий с эталонным списком.

Функциональная схема системы защиты, реализующая механизм уровневого контроля

Функциональная схема, реализующая механизм уровневого контроля [24, 25, 27], представлена на рис. 17.2.

На схеме введены следующие обозначения функциональных блоков:

- блок хранения контрольных сумм 1;
- блок формирования контрольных сумм 2;
- блок сравнения контрольных сумм 3;
- блок управления 4;
- блок хранения восстанавливаемых файлов 5;
- блок идентификации и аутентификации прав доступа 6;
- блок хранения прав доступа 7;
- M блоков хранения списка разрешенных событий 8;
- блок сравнения разрешенных событий 9;

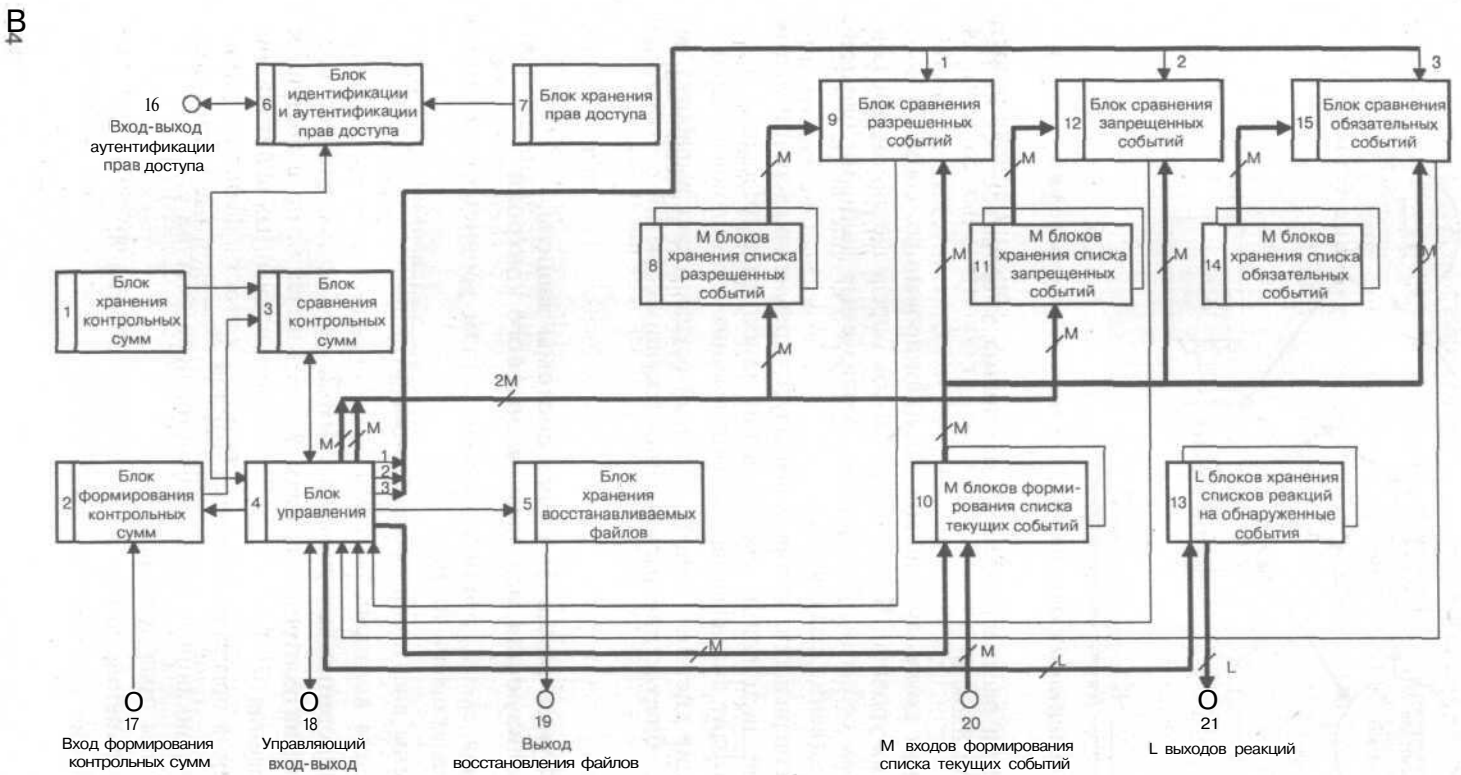


Рис. *17.2. Схема, реализующая механизм уровневого контроля*

* М блоков формирования списка текущих событий 10;
* М блоков хранения списка запрещенных событий 11;
* блок сравнения запрещенных событий 12;
* L блоков хранения списков реакций на обнаруженные события 13;
* М блоков хранения списка обязательных событий 14;
* блок сравнения обязательных событий 15;
* вход-выход аутентификации прав доступа 16;
* вход формирования контрольных сумм 17;
* управляющий вход-выход 18;
* выход восстановления файлов 19;
* М входов формирования списка текущих событий 20;
* L выходов реакций 21.

Работает система следующим образом. В блоках хранения списка разрешенных событий 8, запрещенных событий 11, обязательных событий 14 задаются списки соответствующих событий, контролируемых системой. Например, можно определить:

* разрешенные к запуску процессы, запрещенные к запуску процессы, процессы, которые обязательно должны быть запущены (к последним относятся, прежде всего, процессы, обеспечивающие защиту информации);
* устройства, разрешенные либо запрещенные к использованию (например, дисковод, COM-порт и т.д.);
* адреса хостов, к которым разрешено или запрещено адресоваться;
* сетевые службы, которые разрешены либо запрещены и т.д.

При появлении соответствующего события (одного из М, для каждого из которых существует свой список) оно фиксируется блоком формирования текущих событий и выдается на сравнение с разрешенным, запрещенными, обязательными событиями. Сравнение, соответственно, осуществляется блоками 9, 12, 15.

Информация о несовпадении передается на блок управления 4, который дает команду на соответствующий блок хранения списка реакций на обнаруженные события 13. Блок 13 в свою очередь формирует соответствующую команду, например, завершить несанкционированный процесс, восстановить исходную (эталонную) таблицу разграничения прав доступа к файлам и т.д. Вместе с тем, попытка несанкционированных действий является причиной осуществления проверки целостности файлов, например, настроечных файлов ОС и системы защиты. Эталонные копии последних хранятся в блоке 5, а контрольные суммы в блоке 1.

После выработки реакции на несанкционированное действие блок управления 4 запускает блок формирования контрольных сумм 2, форми-

рующий контрольные суммы файлов, затем блок сравнения контрольных сумм 3 (эталонных и текущих файлов).

При несовпадении блок управления 4 выдает команду в блок хранения восстанавливаемых файлов 5 для восстановления контролируемых файлов, в которых обнаружена ошибка. Для предоставления возможности динамической конфигурации списков санкционированных событий, в процессе функционирования системы, предусмотрена возможность изменения блоком управления 4 списка разрешенных и запрещенных событий, хранящихся, соответственно, в блоках 8 и И. Это может быть проделано лицом, прошедшим идентификацию и аутентификацию блоком идентификации и аутентификации прав доступа 6. Списки хранения прав доступа (разрешенных лиц для динамического конфигурирования списка) располагаются в блоке хранения прав доступа 7.

17.2. Уровневый контроль списков, как механизм реального времени

Исходя из целей механизма уровневого контроля, его основу составляет не аудит с целью обнаружения НСД, а противодействие НСД. При этом автоматическая реакция на зафиксированное несанкционированное событие должна быть осуществлена до того, как злоумышленник получит несанкционированный доступ к ресурсу. В противном случае рассматриваемый метод противодействия атакам преобразуется в метод протоколирования событий -- обычный аудит.

17.2.1. Необходимое условие работы механизма контроля списков, как механизма реального времени

Для оценки возможности применения метода уровневого контроля в реальном масштабе времени необходимо определить временные интервалы, через которые нужно запускать процедуры контроля соответствующих списков. Эти интервалы должны быть таковы, чтобы обеспечивалась своевременная реакция на несанкционированное событие. При этом под несанкционированным событием понимается нарушение одного или нескольких контролируемых базовых списков.

Кроме того, механизм контроля событий, как механизм реального времени, должен запускаться строго через определенный интервал времени (рис. 17.3). Продолжительность проверки тоже должна укладываться в заданный интервал времени, определяющийся временем нарушения соответствующего списка и продолжительностью реакции системы защиты на зафиксированное событие.

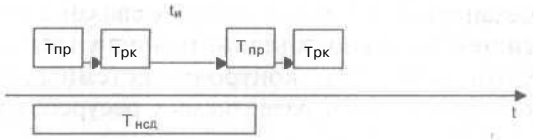


Рис. 77.3. *Временная диаграмма проведения проверок*

Таким образом, необходимым признаком работы механизма уровневого контроля событий, как механизма реального масштаба времени, является следующее условие:

$$T_{пр} + T_{рк} + t_{и} < T_{нсд}, \qquad (17.1)$$

где: $T_{пр}$ — время проверки (контроля события);
$T_{рк}$ время реакции системы на несанкционированное событие;
$t_{и}$ интервал времени до следующей проверки;
$T_{нсд}$ - время, необходимое злоумышленнику на осуществление НСД посредством изменения соответствующего события.

Если не выполняется условие (17.1), то в реальном времени не может быть оказано противодействие, т.е. имеем систему аудита с отложенной реакцией на несанкционированное событие.

Однако в операционной системе помимо механизмов защиты, реализующих контроль соответствующих списков, выполняются прикладные задачи, которым тоже необходимо передавать управление и выделять под них процессорное время. В противном случае вычислительная система сможет решать только задачи защиты информации. Поэтому в условие (17.1) необходимо ввести величину $T_{квант}$, которая будет обозначать время, отдаваемое ОС другим задачам. В соответствии с этим условие (17.1) преобразуется к виду:

$$T_{нсд} - (T_{пр} + T_{рк} + t_{и}) >= T_{квант}.$$

17.2.2. Оценка возможности применения в современных системах механизма уровневого контроля в реальном масштабе времени

Общие положения

При реализации механизма уровневого контроля списков для предотвращения НСД необходимо учитывать большое количество временных параметров. Каждый из контролируемых списков имеет собственное время проверки, восстановления и т.д. Однако на практике большинство контролируемых параметров системы имеют однотипные временные характеристики, т.е. можно выделить группы, которые имеют схожие параметры.

Например, все механизмы, контроль которых связан с работой над объектами файловой системы, можно объединить в одну группу. В другую группу можно выделить механизмы контроля системных структур (список запущенных процессов, список разделяемых ресурсов) и т.д.

Рассмотрим три типа событий (механизмов контроля), отражающих функции защиты от НСД. При этом данные механизмы выбраны исходя из того, что их временные параметры отражают поведение большинства остальных групп механизмов контроля. Выбранные механизмы таковы:

- **механизм контроля процессов,** осуществляющий контроль списков запрещенных/разрешенных/обязательных к запуску процессов. Кроме прикладных и системных процессов, к этой группе также можно отнести драйверы, соответственно, запрещенные/разрешенные/обязательные к запуску;
- **механизм контроля списка** разрешенных пользователей в системе;
- **механизм контроля целостности** объектов файловой системы (например, с помощью контрольных сумм).

Анализ трех вышеназванных механизмов позволит сделать вывод и относительно возможности применения остальных механизмов метода уровневого контроля в силу общности их временных параметров. Поэтому этот анализ мы сейчас непосредственно и приведем.

Определим для выбранных нами групп механизмов контроля временные параметры и сделаем вывод. Далее на основании проведенного анализа, сделаем вывод о принципиальной возможности применения в современных системах рассматриваемого механизма контроля как механизма реального времени.

Анализ механизма контроля процессов

Для механизма контроля процессов рассмотрим следующие параметры:

$T_{зп}$ ..... время получения управления процессом (от момента запуска процесса до того момента когда он может начать свою работу);

$T_{кп}$ ..... время контроля списка процессов (составление текущего списка запущенных процессов);

$T_{оп}$ ..... время завершения процесса (время, необходимое на завершение процесса, нарушающего целостность списка запрещенных/разрешенных процессов).

Для того, чтобы гарантированно защитить систему от влияния программ, нарушающих целостность списка запрещенных/разрешенных процессов (завершить их до того, как они получат управление), необходимо выполнение следующего условия:

$$T_{мп} < T_{кп} + T_{оп} < T_{зп},$$

где $T_{мп}$ - максимальный интервал времени, с которым должна запускаться процедура контроля списка запрещенных/разрешенных процессов. При этом предполагается, что при запуске процесса он сразу же попадает в список текущих запущенных задач.

Посмотрим, возможно ли выполнение такого условия в реальной системе в принципе. В качестве тестовой системы используем ОС Windows 98. Ниже приведены результаты измерений для вычислительной системы C333/112RAM/3GBHDD.

В рамках этих измерений:

- Параметр $T_{кп}$ (время контроля списка процессов) принимал значения, представленные в табл. 17.1. При этом измерения производились:
  - для 15 процессов ..... малая загрузка ЦП;
  - для 30 процессов ..... средняя загрузка ЦП;
  - для 100 процессов ..... большая загрузка ЦП.
- Параметр $T_{оп}$ — время завершения процесса (время реакции) — имеет измеренные значения, представленные в табл. 17.2.
- Параметр $T_{зп}$ — время получения управления процессом (от момента запуска процесса до того момента, когда он может начать свою работу) — имеет измеренные значения, представленные в табл. 17.3.

Как видно из вышеприведенных измерений, величина $T_{мп} = T_{кп} + T_{оп}$ более, чем на порядок меньше величины $T_{зп}$, что позволяет сделать вывод о возможности контроля списков запущенных процессов в реальном времени.

Время контроля списка процессов *Таблица 17.1*

Параметр	1	2	3	4	5	6	7	Среднее
$T_{кп}$(ms) для 15 процессов	0.415	0.465	0.340	0.360	0.360	0.422	0.387	0.389
$T_{кп}$(ms) для 30 процессов	0.726	0.446	0.382	0.339	0.686	0.496	0.489	0.491
$T_{кп}$(ms) для 100 процессов	0.874	0.470	0.963	0.460	0.903	0.513	0.570	0.666

Время завершения процесса (время реакции) *Таблица 17.2*

Параметр	1	2	3	4	5	6	7	Среднее
$T_{оп}$(ms)	0.0010	0.0009	0.0015	0.0014	0.0013	0.0014	0.0011	0.0012

Время получения управления процессом *Таблица 17.3*

Параметр	1	2	3	4	5	6	7	Среднее
$T_{зп}$ (ms)	13	9	13	12	9	10	10	10.8

Анализ механизма контроля списка разрешенных пользователей в системе

Для механизма контроля списка разрешенных пользователей параметрами являются:

- $T_{...}$ - время регистрации нового пользователя;
- $T_{...}$ - время контроля списка пользователей (составление текущего списка работающих пользователей);
- $T_{...}$ - время завершения сеанса пользователя.

Для того, чтобы гарантировано защитить систему от действий незарегистрированного пользователя (не указанного в списке запрещенных/разрешенных пользователей), необходимо выполнение следующего условия:

$$T_{мю} < T_{кю} + T_{ою} < T_{зю},$$

где $T_{...}$ - максимальный интервал времени, с которым должна запускаться процедура контроля пользователей.

Определим, возможно ли выполнение такого условия в реальной системе. В качестве тестовой системы используем ОС Windows NT. Ниже приведены результаты измерений для вычислительной системы C333/112RAM/3GBHDD.

В рамках этих измерений:

- Параметр $T_{кю}$ (время контроля списка пользователей) принимал значения, представленные в табл. 17.4. При этом измерения производились для разного числа пользователей в системе. Количество пользователей увеличивалось за счет запуска дополнительных сервисов с правами нового пользователя.
- Параметр $T_{ою}$ (время завершения сеанса пользователя) имеет измеренные значения, представленные в табл. 17.5.
- Параметр $T_{зю}$ (время регистрации нового пользователя) примем как минимальное время регистрации 500 ms (нажатие кнопки OK в окне регистрации).

Время контроля списка пользователей *Таблица 17.4*

Параметр	1	2	3	4	5	6	7	Среднее
$T_{кю}$(ms) для 5 пользователей	0.023	0.025	0.017	0.022	0.023	0.025	0.022	0.023
$T_{кю}$(ms) для 20 пользователей	0.24	0.022	0.020	0.023	0.023	0.24	0.23	0.23

Время завершения сеанса пользователя *Таблица 17.5*

Параметр	1	2	3	4	5	6	7	Среднее
$T_{ою}$(ms)	0.0015	0.0014	0.0016	0.0014	0.0013	0.0012	0.0012	0.0013

Как видно из вышеприведенных измерений, величина $T_{мю} = T_{кю} + T_{ою}$ на порядки меньше величины $T_{зю}$, что позволяет сделать вывод о возможности контроля списков в реальном времени..

Анализ механизма контроля целостности файловой системы

С самого начала необходимо отметить, что эту группу механизмов образуют не списки прав доступа (к файловым объектам, к ключам реестра, к устройствам и т.д.), которые также могут контролироваться, образуя соответствующие списки событий, а собственно файловые объекты. Таким образом, для данной группы контролируется не собственно процесс доступа, а непосредственно состояние файлового объекта.

Для механизма контроля целостности объектов файловой системы параметрами являются:

$T_{зф}$ ..... время доступа к объекту файловой системы (на запись);

$T_{кф}$ ..... время контроля целостности объектов файловой системы.

Для того чтобы гарантировано защитить систему от действий, связанных с нарушением целостности объектов файловой системы, должно выполняться условие:

$$T_{мф} < T_{кф} < T_{зф},$$

где $T_{мф}$ - максимальный интервал времени с которым должна запускаться процедура контроля целостности объектов файловой системы.

Очевидно, что данное условие практически никогда выполняться не будет, т.к., чтобы проконтролировать целостность файлового объекта, его нужно сначала прочитать, что сопоставимо по продолжительности с записью объекта. Исключение может составить лишь включение искусственой задержки доступа к файловым объектам, но это связано с большими затратами вычислительного ресурса.

Обобщенная оценка

На основе проведенных измерений может быть сделан вывод, что для большинства параметров системы (за исключением контроля объектов файловой системы) возможно применение уровневого контроля событий в реальном масштабе времени для защиты их от НСД.

С учетом существенного различия ограничений на время реакции системы при контроле списков различных событий может быть рассмотрена и исследована задача назначения приоритетов в схеме обслуживания запросов контроля событий. При этом необходимо учитывать, что включение приоритетов не должно нарушать выполнения требований к функционированию системы в реальном времени.

17.3. Двухуровневая модель аудита на базе механизма уровневого контроля списков санкционированных событий

Система защиты должна осуществлять регистрацию основных событий при функционировании защищаемого объекта.

В рамках построения иерархической системы защиты, как следствие, может строиться иерархическая система аудита, содержащая следующие уровни иерархии.

Первый уровень иерархии аудита — регистрация событий уровней защиты, реализующих разграничительную политику доступа к ресурсам. В рамках данной функции аудита решаются следующие задачи:

* ведется аудит всех событий, связанных с действиями пользователей по доступу к ресурсам защищаемой системы (вход в систему, запрос и получение доступа к защищаемым ресурсам и т.д.);
« ведется аудит событий, связанных с действиями администратора безопасности по созданию и переназначению прав доступа пользователей к ресурсам защищаемой системы (создание и уничтожение субъектов и объектов доступа, действий по переназначению прав разграничения доступа и т.д.).

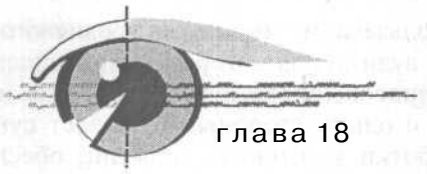
Второй уровень иерархии аудита — регистрация событий, относящихся к контролю корректности функционирования механизмов защиты, реализующих разграничительную политику доступа к ресурсам. Данный контроль ведется механизмами уровневого контроля списков санкционированных событий.

Принципиальным отличием в регистрации событий для рассмотренных уровней иерархии аудита является то, что на первом уровне иерархии фиксируются все события, связанные с доступом к защищаемым ресурсам (ведутся журналы регистрации). Что касается второго уровня, то на нем регистрируются только ошибки (ведутся журналы ошибок) и факты некорректности срабатывания механизмов защиты, реализующих разграничительную политику доступа к ресурсам, а также факты НСД.

Реализация иерархической модели аудита позволяет ввести в системе защиты различные схемы обработки журналов, в частности при удаленном управлении системой защиты — ее клиентской частью и с сервера безопасности. При этом регистрационная информация первого уровня аудита сохраняется в соответствующих файлах клиентской части системы защиты и могут быть переданы на сервер безопасности по запросу администратора безопасности (затем соответствующим образом им обработаны с применением фильтров, входящих в состав системы защиты).

Регистрационная информация второго уровня аудита в реальном времени выдается клиентской частью системы защиты на сервер безопасности и отображается в специальном окне интерфейса сервера — в окне сервера ошибок. Кроме того, на каждый тип ошибки администратор безопасности может задать реакцию, как на клиентской части системы защиты, так и на сервере безопасности.

Таким образом, использование механизма уровневого контроля позволяет выделить два уровня аудита, для которых принципиально различаются режимы обработки запросов (оперативная обработка и обработка в реальном времени). Это, с одной стороны, позволяет существенно повысить оперативность обработки критичных событий, обеспечивая реакцию на критичные события в реальном масштабе времени. С другой стороны это позволяет существенно снизить нагрузку на трафик. И действительно, в реальном времени по сети на сервер безопасности передаются только данные второго уровня аудита, т.е. нагрузка на сетевой трафик подсистемой аудита сводится к минимуму. При этом отметим, что в штатном режиме функционирования сетевой системы защиты именно интенсивность и объемы передачи данных аудита определяют дополнительную нагрузку, генерируемую системой защиты на сетевой ресурс.



глава 18

Разработка и оптимизация механизма уровневого контроля, как механизма реального времени

18.1. Задача оптимизации механизма уровневого контроля в рамках вычислительной системы

Очевидно, что как любая синхронная процедура (синхронно — то есть по расписанию) рассматриваемый метод защиты информации потенциально может оказывать весьма существенное влияние на производительность защищаемого объекта. Особенно это может сказываться при жестких временных ограничениях на время автоматической реакции на обнаруживаемое несанкционированное событие (система реального времени).

Уже на основе описания метода можно предположить, что потребуется решать оптимизационные задачи с целью снижения влияния рассматриваемого подхода на производительность системы.

Так как исследования будут проводиться для систем реального времени (или реального масштаба времени — РМВ), прежде всего рассмотрим основные принципы построения и синтеза подобных систем в общем случае.

18.2. Теоретические основы обслуживания заявок в вычислительных системах в реальном времени (по расписаниям)

18.2.1. Общая модель системы реального времени (в общем виде)

Общая модель и условие обслуживания заявки в реальном времени

Рассмотрим модель системы реального времени в общем случае, то есть для распределенных вычислительных систем (ВС). Соответственно, в частном случае, когда необходимо решать задачу в рамках сосредоточенной вычислительной системы (в рамках одного защищаемого компьютера), соответствующие параметры, используемые в модели, могут быть опущены [20, 21].

Так как основными требованиями к качеству построения и функционирования ВС реального времени является выполнение временных ограничений в обслуживании заявок, то распределенную ВС реального времени можно описать детерминированной моделью. Эта модель характеризуется не вероятностными, а граничными значениями параметров.

К временным параметрам обслуживания заявок относятся величины:

$T_{\_m}$ . . . продолжительность занятия ресурса m-м абонентом, $m = 1,...,M$ для информационного взаимодействия с ним;

$T_{пп_m}$ .... продолжительность передачи прав m-му абоненту после освобождения ресурса в системе;

$K_{i(m)}$ .. коэффициент частоты занятия ресурса i-м абонентом $i = 1,...,M$ относительно m-го $(i \neq m)$.

Качество обслуживания заявок можно описать характеристиками:

T_{a_m} . . . . продолжительность арбитража требования m-го абонента (с момента появления заявки до момента предоставления абоненту права занять ресурс) или, соответственно, продолжительность ожидания заявкой обслуживания;

T_{o_m} . . . . продолжительность обслуживания заявки системой.

Для систем реального времени интерес представляют граничные (худшие для любой заявки) значения рассмотренных характеристик, которые соответственно обозначим: $T_{гр_m}$, $T_{гпп_m}$, $K_{гi(m)}$, $T_{га_m}$, $T_{го_m}$, $i,m = \overline{1,...,M}$, $i \neq m$, откуда получаем параметры обслуживания заявок в системе реального времени: $T_{га_m}$, $T_{го_m}$.

С учетом сказанного получаем модель системы реального времени в общем виде:

$$\begin{cases} T_{гa_m} \sim T_{гnn_m} \sum_{i=1}^{M} K_{гi(m,i\neq m)}(T_{гnn_i}, T_{гp_i}) \\ T = T_{гa_m} + T_{гp_m} \\ \forall m, m = 1,\ldots,M \quad T_{гp_m} \neq \infty \end{cases} \qquad (18.1)$$

Таким образом, обслуживание заявки в реальном масштабе времени корректно, если для любого абонента системы *m (m = \...M)* выполняются условия (18.1). Если условие (18.1) хотя бы для одного абонента системы не выполняется, то нельзя считать, что его заявки обслуживаются в реальном масштабе времени. В этом случае параметры обслуживания T_{a_m} и T_{o_m} не могут быть ограничены сверху, и, следовательно, всегда найдутся условия функционирования системы, при которых $T_{гa_m} < T_{a_m}$, $T_{го_m} < T_{o_m}$ или заявка будет обслужена не в реальном времени.

Понятие «цикл расписания»

Условие (18.1) выполняется только в том случае, если в системе реализуется дисциплина обслуживания заявок по расписаниям, т.к. любая иная дисциплина обеспечивает выполнение данных условий с некоторой вероятностью (без учета эксплуатационных характеристик), что для рассматриваемого случая недопустимо.

Расписание будем характеризовать его циклом — повторяющейся очередностью передачи прав на занятие ресурса. Цикл расписания будем задавать в круглых скобках. Например, цикл (1, 2, 3, 4) означает следующую циклическую последовательность передачи прав на занятие ресурса: первому абоненту, затем второму, затем третьему, затем четвертому.

Приоритетное и бесприоритетное обслуживание в реальном времени

Особенностью обслуживания заявок в реальном времени будет то, что каждая заявка гарантированно должна быть обслужена за время $T_{го_m}$. Исходя из этого, в данном случае приоритет заявки нельзя трактовать, как преимущественное право одной заявки перед другой быть обслуженной (как, например, в случаях относительных и абсолютных приоритетов). Все заявки должны быть обслужены. При этом приоритеты заявок здесь представляют собой отношение гарантированных продолжительностей их обслуживания: $T_{го_m}$.

Будем говорить, что в системе реализована **бесприоритетная дисциплина обслуживания** требований общего ресурса реального времени, если для всех абонентов совпадают значения параметра $T_{го_m}$. Соответственно, при-

оритетная дисциплина обслуживания требований общего ресурса реального времени имеет место, если хотя бы для двух любых абонентов не совпадают значения параметра $T_{го_m}$.

В качестве параметра приоритетности обслуживания заявок в реальном времени может быть введена количественная оценка — относительный уровень приоритетности реального времени (или относительный приоритет реального времени) двух абонентов m и m'; $m, m' = 1,...,M$.

Под **относительным уровнем приоритетности** реального времени (или относительным приоритетом реального времени) двух абонентов *m и m'*; $m, m' = 1,...,M$ понимается отношение $\delta_{m-m'} = T_{го_m} / T_{го_{m'}}$.

Соответственно, в системе реализована бесприоритетная дисциплина обслуживания требований ресурса, если для любых двух абонентов системы т, m' выполняется: $\delta_{m-m'} = 1 (m \neq m')$; если $\delta_{m-m'} < 1$ приоритет *m* абонента в $\delta_{m-m'}$ выше, в противном случае — приоритет в $\delta_{m-m'}$ ниже.

Способы задания приоритетного обслуживания

Системой (18.1) в общем случае определяются три способа задания приоритетного обслуживания заявок в реальном времени для распределенных ВС и, соответственно, их комбинации:

1. Изменением параметров $K_{гi(m)}$ - именно этот параметр может изменяться при изменении цикла расписания (например, (1, 2, 3, 4) и (1, 2, 1, 3, 1, 4)).
2. Изменением параметров $T_{гпп_m}$.
3. Изменением параметров $T_{гр_m}$.

При этом очевидно, что выбор способов задания приоритетов абонентов определяется соотношением параметров $T_{гр_m}$ и $T_{гпп_m}$. При $T_{гр_m} >> T_{гпп_m}$ целесообразно использовать способы 1 и 3 (это рассматриваемый нами далее случай сосредоточенной системы — в рамках одного компьютера), а при сопоставимости $T_{гр_m}$ и $T_{гпп_m}$ - соответственно, способы 1 и 2.

18.2.2. Условия эффективности приоритетного обслуживания в реальном времени. Критерий оптимальности

Гарантированная продолжительность реакции системы

Рассмотрим, с какой целью и при каких условиях следует назначать приоритеты заявкам на обслуживания в реальном времени [21]. При этом, говоря о заявках на обслуживание, будем понимать, что заявки принадлежат к различным типам и каждый тип заявок образует свою очередь.

Пусть в систему реального времени поступает M типов заявок на обслуживание. Тогда условием корректности функционирования системы будет выполнение M неравенств:

$$T_{грс_m} \leq T^{з}_{грс_m},$$

где: $T_{грс_m}$ - гарантированная продолжительность реакции системы на воздействие, реализуемое в системе,

$T^{з}_{грс}$ - задаваемое условиями функционирования ограничение на параметр $T_{грс_m}$.

Требуемая реактивность системы по каждому m-му воздействию определяется следующими характеристиками обслуживания заявок системой:

* $T_{грз_m}$ - продолжительностью собственно решения m-й задачи по заранее известной программе (здесь для простоты считаем, что обрабатывается только один соответствующий тип воздействий);
* числом информационных взаимодействий с ресурсом $L_{г_m}$, необходимым для выработки адекватного воздействия в соответствии с заранее известным алгоритмом (далее для простоты будем считать, что $L_{г_m} = 1$, $m = 1,...,M$).
* $T_{го_m}$ — максимальной продолжительностью взаимодействия с ресурсом.

Тогда гарантированная продолжительность реакции системы на m-е воздействие составит:

$$T_{грс_m} = T_{грз_m} + L_{г_m} T_{го_m}.$$

Рассмотрим следующие возможные случаи:

1. Пусть требуется обеспечить равное время реакции системы на все воздействия — $T^{з}_{грс_m}$ для всех m, $m = 1,...,M$ воздействий совпадают. В этом случае идеальным является бесприоритетное обслуживание — в реальном времени это передача полномочий в циклической очередности.
2. Пусть $T^{з}_{грс_m}$ заданы различными, то есть различен функциональный смысл воздействий. Именно это имеет место в рассматриваемом нами случае, когда контролируемые объекты имеют различный физический смысл. В этом случае при бесприоритетной дисциплине обслуживания требований система может функционировать корректно лишь при выполнении для всех M заявок самого жесткого для системы ограничения.

Критерий оптимальности дисциплины обслуживания

Предположим, что рассматриваемое жесткое ограничение $T^{з}_{грс}$ в системе при выбранной производительности технических средств не выполняется для некоторых заявок. При этом, в общем случае, некоторые ограничения в си-

стеме могут выполняться с большим запасом по производительности ресурса. Тогда можно сформулировать задачу об эффективном перераспределении производительности вычислительных средств системы с учетом выполнения требований к корректности функционирования системы в целом.

Критерием оптимальности задания очередности передачи прав на занятие ресурса (соответственно дисциплины обслуживания) будет относительный коэффициент избыточности в эффективности обслуживания:

$$\delta_{T_{грс_m}} = \frac{T^{э}_{грс_m}}{T_{грс_m}}.$$

При этом очевидно, что **наиболее эффективно система с общими ресурсами будет реализована в том случае, если выполняются условия** $\delta_{T_{грс_m}} = \frac{T^{э}_{грс_m}}{T_{грс_m}} - 1,\ m = \overline{1,M}$. **Это можно считать условием оптимальности дисциплины обслуживания реального времени, а параметр** $\delta_{T_{грс_m}}$ **критериемоптимальности.**

В общем случае для критерия оптимальности $\delta_{T_{грс_m}} = \frac{T^{э}_{грс_m}}{T_{грс_m}}$ при реализации бесприоритетного обслуживания имеем:

$$\delta_{T_{грс_m}} = \frac{T^{э}_{грс_m}}{\min\{T^{э}_{грс_m}, m = \overline{1,M}\}}.$$

Выигрыш в производительности системы реального времени при реализации приоритетного обслуживания

Оценим, какой выигрыш может дать реализация в системе приоритетного обслуживания, где приоритет вводится с целью эффективного использования производительности общего ресурса (при связном ресурсе – эффективного использования пропускной способности канала связи). С учетом $L_{з_m} = 1$ для всех M заявок, для системы с бесприоритетной дисциплиной обслуживания (характеристика обозначена БП) требований ресурса имеем:

$$T^{БП} = MT_{гр_m},$$

где $T_{гр_m}$ - продолжительность занятия ресурса с учетом потерь времени на передачу ему прав системой занять ресурс. При это считаем, что $T_{гр_m}$ совпадают для всех M очередей заявок.

Пусть требования к времени реакции системы на 1-е входное воздействие существенно выше, чем требования к любому иному входному воздействию $m = 2,\ldots,M:\ T^{э}_{грс_{m1}} << T^{э}_{грс_{m=2\ldots M}}$. Реализуем для рассматриваемой системы приоритетную дисциплину обслуживания требований, в которой очередность предоставления прав заявкам на занятие ресурса (цикл расписания) выглядит следующим образом: (1, 2, 1, 3, 1, 4, 1, ... , 1, $M-1$, 1, M).

Для данной дисциплины обслуживания имеем следующие характеристики обслуживания заявок:

$$T^{ВП}_{го_{m=1}} = 2T_{гр_m},$$

$$T^{НП}_{го_{m=\overline{2,M}}} = 2(m-1)T_{гр_m},$$

где ВП и НП -- соответственно характеристики абонентов с высоким и низким приоритетом, откуда выигрыш для более приоритетной заяки составит:

$$\delta_{T^{ВП}_{го_m}} = \frac{T^{БП}_{го_m}}{T^{ВП}_{го_m}} = \frac{M}{2} \quad (\text{для всех } m \ T_{гр_m} = T_{гр}).$$

Однако, наряду с выигрышем для более приоритетной заявки, имеем и проигрыш для менее приоритетной заявки:

$$\delta_{T^{НП}_{го_m}} = \frac{T^{НП}_{го_m}}{T^{БП}_{го_m}} = \frac{2(M-1)}{M} = 2.$$

Откуда результирующий выигрыш в производительности ресурса:

$$\delta_{T_{грс_m}} = \frac{T^{ВП}_{го_m}}{T^{НП}_{го_m}} = \frac{M}{4}.$$

Количественная оценка выигрыша

Рассмотрим количественную оценку получаемого выигрыша. Пусть $T_{грэ_m} = kT_{го_m}$, где $0 < k < \infty$. В предположении, что $L_{гm} = 1$, $m = \overline{1,M}$ для бесприоритетного обслуживания имеем:

$$T^{БП}_{грс_m} = (M+k)T_{гр_m},$$

Соответственно, для приоритетного обслуживания при двух уровнях приоритета получаем:

$$T^{ВП}_{грс_m} = (2+k)T_{гр_m},$$

$$T^{НП}_{грс_m} = (2(M-1)+k)T_{гр_m},$$

отсюда получаем результирующий выигрыш в производительности системы:

$$\delta_{T_{грс_m}} = \frac{(M+k)^2}{(2+k)(2(M-1)+k)}.$$

Зависимости $\delta_{T_{грс_m}} = f(k)$ для различных M, при $M = 16$ и $M = 64$ представлены на рис. 18.1. Можно сделать следующие выводы.

1. При существенном различии ограничений $T^{з}_{грс}$ для M заявок (типов заявок или очередей) при условии $k <$ [появляется возможность существенного повышения производительности системы ре-

ального времени, за счет реализации эффективного управления использованием вычислительного ресурса заявками в результате введения приоритетного обслуживания. Обычно это является «узким местом» системы. Получаемый выигрыш в производительности системы в целом здесь может составить десятки раз, возрастая при увеличении числа типов заявок (очередей) в системе *M*.

2. Получаемый от реализации приоритетного расписания выигрыш снижается при увеличении коэффициента *k,* или при опережающем росте $T_{грз_m}$ над $T_{го_m}$, что определяет переход к классу сосредоточенных систем.
3. При $k > 1$ из (18.1) имеем $\delta_{T_{гр.m}} = k^2/k^2 = 1$, то есть отсутствует выигрыш как таковой. Это может иметь место для некоторых приложений сосредоточенных систем, когда $T_{грз_m} >> T_{го_m}$. В этом случае теряет актуальность задача приоритетного обслуживания заявок с целью эффективного использования ресурса.

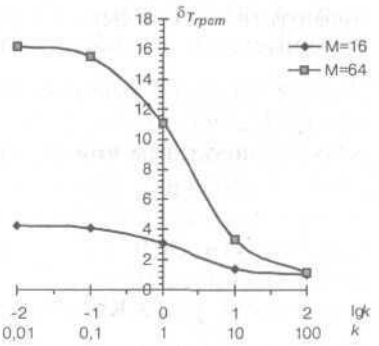


***Рис 18.1** Ил ация выигрыша в производительности системы реального времени при реализации приоритетного обслуживания*

Нетрудно показать, что предельным случаем системы, где наиболее эффективно включение приоритетного обслуживания и «узким местом» которой будет общий вычислительный ресурс ресурс, можно считать многозадачную операционную систему реального времени, т.е. именно те приложения, которые нами рассматриваются в работе.

Выводы

С учетом сказанного можем сформулировать цель назначения приоритетов при обслуживании запросов по расписанию для систем реального времени. При этом помним, что в этих системах имеется общий ресурс и именно он является «узким местом».

Итак, цель реализации приоритетного обслуживания в системах реального времени состоит в перераспределении прав между заявками (очередями заявок) на доступ к ресурсу в соответствии с заданными ограничениями времени реакции системы на входное воздействие. Выполнение данных ограничений при минимальной производительности ресурса достигается при выполнении условия: $\delta_{T_{гр.m}} = 1$, $m = \overline{1,...,M}$. Таким образом, условие $\delta_{T_{гр.m}} = 1$ можно считать условием оптимальности приоритетной дисциплины обслуживания реального времени.

18.3. Реализация приоритетных расписаний в современных ОС

Принципы приоритетного обслуживания заявок в современных универсальных ОС

Прежде чем перейти к дальнейшему рассмотрению вопросов синтеза приоритетных расписаний для рассматриваемых приложений, рассмотрим, каким образом можно реализовать обслуживание заявок по расписаниям в современных ОС.

В большинстве современных универсальных ОС алгоритм передачи управления (системных ресурсов центрального процессора) базируется на механизме назначения относительных приоритетов. В соответсвии с этим механизмом каждой программе может быть назначен свой приоритет. Диапазон приоритетов в разных ОС колеблется от 6 до нескольких десятков.

Диспетчер задач распределяет кванты времени между задачами в соответствии с их приоритетами. При этом ресурсы центрального процессора будут переданы той задаче, у которой более высокий приоритет. При этом до истечения кванта выданного ей процессорного времени, программа не может быть прервана (если только сама не отдаст управление). Таким образом, речь идет об обслуживании заявок по дисциплине с относительными приоритетами. Эта дисциплина не позволяет корректно реализовать расписания реального времени.

Если же у всех текущих задач приоритеты равны, то в общем случае ресурсы центрального процессора будут передаваться задачам в порядке их поступления (бесприоритетная дисциплина обслуживания, но опять же не реального времени).

Основу реализации приоритетного расписания на базе дисциплины обслуживания с относительными приоритетами составляет изменение относительных приоритетов по расписанию. Другими словами, в каждый момент времени реализуется дисциплина обслуживания с относительными приоритетами, которые изменяются в процессе функционирования системы (при каждом предоставлении ресурса задаче). Таким образом и реализуется приоритетное расписание. Возникает вопрос: «Какое средство (задача) будет осуществлять перераспределение относительных приоритетов по расписанию?» Поиском ответа мы и займемся в следующем пункте.

Подход к реализации в современных ОС управления по приоритетным расписаниям

Рассмотрим подход к реализации в ОС передачи управления в соответствии с приоритетным расписанием. Данный подход состоит в следующем:

» Выделяются задачи, реализующие собственно защиту от НСД механизмом уровневого контроля списков. Им назначается высокий относительный приоритет.

» Среди данных задач выделяется одна, осуществляющая диспетчеризацию заявок (передачу полномочий между заявками на занятие ресурса) –- ей назначается максимальный относительный приоритет.

« Выделяются прикладные задачи. Им назначается низкий приоритет.

Таким образом, из очереди будет выбираться всегда задача, имеющая максимальный приоритет — задача, обеспечивающая диспетчеризацию заявок. Получив управление, задача может либо занять вычислительный ресурс, либо передать его на квант времени (обозначим это время как $T_{квант}$) прикладным задачам. Т.е. задача, реализующая защиту от НСД, будет распределять время между собой и прикладными задачами (реализовывать расписание). Схема такой передачи управления приведена на рис. 18.2.

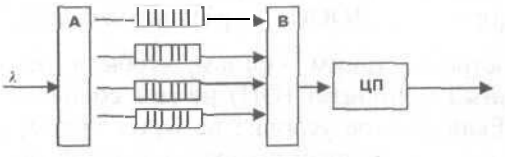


Рис. 18.2. *Схема передачи управления между задачами*

На схеме обозначен входной поток задач λ. Прибор постановки в очередь (А) выбирает из входного потока задачи и распределяет их по очередям в соответствии с их приоритетами.

В случае, если задача принудительно отдала управление на заданное время, прибор постановки отложит размещение этой задачи в очередь. Прибор выбора задачи из очереди (В) выбирает задачи из очередей в соответствии с их приоритетами и передает их на обслуживание в центральный процессор (ЦП).

На схеме задачам одного приоритета выдается одна очередь, из которой происходит выборка опять же в соответствии с расписанием.

Выводы

Подытожить вышесказанное можно следующими высказываниями:

1. В современных универсальных ОС основу подхода к приоритетному обслуживанию заявок на контроль событий в реальном времени может составлять обслуживание с динамическими относительными приоритетами, изменяемыми по расписанию, при каждом занятии вычислительного ресурса.
2. Посредством реализации рассмотренного подхода, в большинстве современных ОС можно реализовывать расписания (обслуживание в реальном времени) любого уровня сложности. При этом передачей управления занимается уже не диспетчер задач ОС (он осуществляет это на более низком уровне), а тот механизм, который специально наделен такими функциями.

18.4. Построение и использование эффективных приоритетных расписаний

18.4.1. Основы построения приоритетных расписаний

Основные понятия

Идея рассматриваемого подхода состоит в реализации дисциплин обслуживания реального времени с передачей прав по расписанию (ДОР), по средством смены относительных приоритетов (ДООП) по расписанию [20].

При этом система должна быть построена таким образом, чтобы в любой момент времени t_s относительный приоритет (ОП) не мог совпасть у заявок из нескольких очередей. Если данное условие не будет выполнено, то в системе неминуем конфликт при занятии ресурса, т.к. несколько абонентов одновременно получат право занять ресурс после его освобождения.

Для описания ДООП используем матрицу приоритетов (МП), представляющую собою квадратную матрицу $Q = [q_{ij}]$ размерности $M \times M$ по числу M абонентов. Элемент матрицы q_{ij} задает ОП заявки i по отношению к j (здесь и далее имеется в виду очередь или, соответственно, класс заявок): «0» — нет приоритета, «1» — есть приоритет. Например, первая строка первой таблицы «0111» означает, что первая заявка имеет более высокий приоритет чем вторая, третья и четвертая заявки. Для описания ДОР (в общем случае ДОСП) используем граф изменения матрицы ОП в моменты времени t_s — моменты занятия ресурса в соответствии с расписанием.

На рис 18.3 представлен пример графа бесприоритетной ДОР, реализуемой методом динамической смены ОП, для случая $M = 4$, цикл расписания которой имеет вид: (1, 2, 3, 4). Соответственно, в каждый момент времени состояние системы описывается своей МП.

Бесприоритетность расписания обеспечивается тем, что каждый абонент входит в расписание равное число раз. Причем это число в общем случае может быть более одного, например (1, 1, 2, 2, 3, 3, 4, 4).

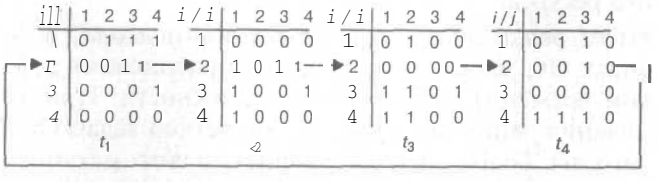


Рис. 18.3. *Пример графа бесприоритетной дисциплины обслуживания*

Требования к матрице приоритетов

Элементы МП должны удовлетворять следующим требованиям:

* $q_{i=j} = 0$, т.к. между заявками одного класса не могут быть установлены приоритеты;
» если $q_{ij} = 1$, то $q_{ji} = 0$, т.е., если заявки класса i имеют приоритет по отношению к заявкам класса j, то последние не могут иметь приоритет по отношению к заявкам i;
* в матрице приоритетов не должны совпасть любые две строки i, i', любые два столбца у, j'; $i = 1,...,M$, $i \neq m$, $j, j' = 1,...,M$, $j \neq j'$.

В графе смены МП (в цикле расписания), по крайней мере, по одному разу должны присутствовать МП, задающие высший ОП каждого из М классов заявок.

Правило высших относительных приоритетов

Для реализации приоритетной ДОР в цикле расписания по крайней мере двум классам заявок высший относительный приоритет должен присваиваться различное число раз, например (1, 2, 1, 3, 1, 4).

В противном случае получим совпадение значений $T_{го_m}$, т.е. при совпадении значений параметров $T_{гр3_m}$ получаем равный приоритет заявок — совпадают $T_{грс_m}$.

Правило изменения относительных приоритетов

Изменение ОП заявок по расписанию в процессе функционирования системы должно быть реализовано по следующему правилу.

Относительные приоритеты (ОП) в рамках ДОР однозначно задаются расписанием, где в каждый момент времени t_s приоритет заявок соответствует порядку передачи полномочий, исключая повторные передачи прав одной очереди в цикле ОР (эта возможность реализуется изменением длины кванта времени). Например, для расписания (1, 2, 1, 3) в момент времени t_1 -- ОП [1, 2, 3], в момент времени t_2 -- [2, 1, 3], в момент времени t_3 — [1, 3, 2], в t_4 — [3, 1, 2].

Для дисциплин обслуживания с динамическими ОП, изменяемыми по расписанию, функция изменения приоритета заявки m, $m = \overline{1,...,M}$ имеет вид:

$$\varphi_m(t) = \alpha_{ms}(t_s \le t^n < t_{s+1}) + \Delta\alpha_{md}(t_d, d = 1,...,G),$$

где: α_{ms} - - исходный ОП заявки поступающей в момент t'', соответствующий s-му состоянию цикла расписания, длиной G: t_s $s = 1,...,G$;

$\Delta\alpha_{md}$ — приращение приоритета заявки, получаемое при смене состояний цикла расписания t_d $d = \overline{1,...,G}$ (может иметь отрицательные значения).

Для заявок, обслуживаемых с ОП, для $\forall t_d$, $\Delta\alpha_{md} = 0$ и для $\forall t^n = t_s$, $s = 1,...,G$, $\alpha_{ms} = const$.

Режимы использования относительного приоритета

Рассмотренный подход позволяет комбинировать в одной системе обслуживание заявок и по расписаниям и с относительным приоритетом (посредством смены относительных приоритетов). Естественно, что если какой-либо заявке относительный приоритет не изменять в процессе функционирования системы, то запросы данной заявки будут тем самым исключены из расписания. В дальнейшем будем обозначать дисциплину осблуживания со смешанными приоритетами (относительного и реального времени) квадратными скобками.

Можно выделить два режима использования относительного приоритета, которые определяются его отношением к приоритетам реального времени:

- Первый режим предполагает задание относительного приоритета заявки относительно заявок, обслуживаемых в реальном времени, например, [1, (2, 3, 4)]. Данный режим может выделяться для внеочередного обслуживания каких-либо заявок (в нашем примере заявка 1 всегда приоритетнее, чем заявки 2, 3, 4, которые обслуживаются между собой по бесприоритетному расписанию).

  Этот режим может использоваться для задания относительного приоритета задаче, осуществляющей диспетчеризацию заявок.
- Второй режим предполагает задание относительного приоритета заявок реального времени относительно остальных заявок, например, [(1, 2, 3), 4] (в нашем примере заявка 4 всегда менее приоритетна, чем заявки 1, 2, 3, которые обслуживаются между собой по бесприоритетному расписанию).

  Данный режим может выделяться для фонового обслуживания каких-либо заявок (в нашем примере заявка 4 будет обслуживаться в фоновом режиме, например, в фоновом режиме могут обслуживаться заявки приложений).

18.4.2. Теоретические основы синтеза приоритетных расписаний

Детерминированная задача синтеза расписаний реального времени

Выше отмечалось, что под приоритетным понимается расписание, в цикле которого различные классы или очереди заявок встречаются различное число раз. Возникает вопрос сравнения между собой приоритетных расписаний

с целью выработки требований к их оптимальности. Например, рассмотрим два расписания (1, 1, 2, 3) и (1, 2, 1, 3). В обоих расписаниях первый абонент имеет более высокий приоритет (в 2 раза чаще встречается в цикле, чем второй и третий абоненты). Возникает вопрос — одинаковы или различны эти расписания с точки зрения эффективности их применения в системах реального времени. Попытаемся проанализировать данный вопрос и выработать требования к оптимальному приоритетному расписанию [22].

Пусть имеем R различных приоритетов заявок в системе, обслуживаемых по расписанию. Соответственно, $r = 1,...,R$, и M_r заявок имеют равный r приоритет $m_r = 1,...,M_r$: $\delta_{m_r - m'} = 1$, $m_r \neq m'_r$, $m_r, m'_r = 1,...,M_r$. Таким образом, система содержит M очередей заявок, которым соответствует R уровней приоритетов.

Детерминированная задача синтеза расписаний реального времени может быть сформулирована следующим образом : $M|1|T_{om_r} \to max$, $T_{om_r} \leq T_{zom_r}$, $m_r = 1,...,M_r$, $r = \overline{1,...,R}$, где T_{om_r} — характеристика обслуживания m-й очереди заявок r-го приоритета, T_{zom_r} - ограничение, накладываемое на время обслуживания заявки системой, функционирующей в реальном масштабе времени.

Оптимальное приоритетное расписание и радиальный граф

С учетом сформулированной задачи синтеза расписаний реального времени дадим определение канонического (в данном случае — оптимального) расписания.

Под каноническим (оптимальным) расписанием будем понимать расписание, обеспечивающее минимальные значения характеристик T_{zom_r} в рамках заданного разбиения M классов заявок на R уровней приоритета, получаемое поочередной передачей прав очередям заявок в рамках радиального графа.

Под радиальным графом передачи прав понимается граф, удовлетворяющий следующим условиям:

- Этот граф содержит M_r вершин приоритета r, $r = \overline{1,...,R}$, причем в каждую вершину нижестоящего приоритета входит только одна дуга от одной вершины вышестоящего приоритета.
- Из каждой вершины всех приоритетов, кроме R-го, выходит $[M_{r+1}/M_r]$ дуг, где $[a]$ — есть большее целое числа a.
- Все вершины приоритета R соединяются дугой с вершиной приоритета 1 (из вершин R выходит только одна дуга) — в этом случае дуга направлена в сторону вершины 1, в то время как в остальных случаях дуга направлена в вершину, соответствующую очереди заявки с более низким приоритетом.

Радиальный граф передачи прав представлен на рис. 18.4.

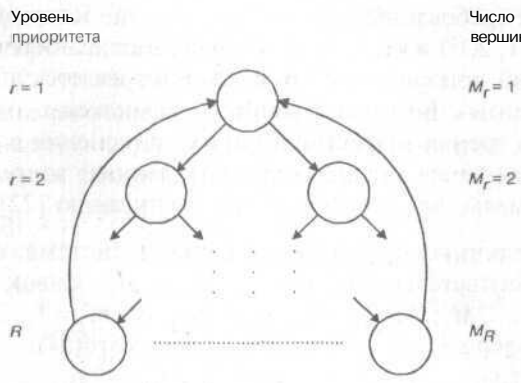


Рис. 18.4. *Радиальный граф передачи прав на занятие ресурса*

Радиальный граф, у которого по крайней мере одна вершина r-го (не R-го) уровня приоритета соединена только с одной вершиной приоритета уровня $r + 1$, назовем **вырожденным.** Соединение двух вершин одной дугой является альтернативным способом задания равного приоритета двум абонентам системы. Под **предельно вырожденным** понимаем граф, в котором из каждой вершины только одна дуга выходит и в каждую вершину только одна дуга входит — этот граф задает бесприоритетное расписание, соответствующее обслуживанию в циклическом порядке.

Радиальный граф с поочередной передачей прав

Под поочередной будем понимать такую передачу прав на занятие ресурса заявками, при которой заявки одного приоритета получают полномочия на доступ к ресурсу с равной частотой или бесприоритетно друг относительно друга. Для радиального графа такая очередность может быть определена в результате выполнения приведенной ниже итерационной процедуры. При этом, в общем случае реализуется R итераций.

1. Произвольным образом через одну вершину каждого приоритета проводится первый путь.
2. Второй путь должен пройти через вершины всех приоритетов, кроме первого (через первый путь проходит всегда), не соединенные первым путем; третий — не соединенные первыми двумя путями и т.д.
3. Если не осталось на рассмотрении вершин г-го приоритета, через которые не прошел по крайней мере один путь (для невырожденного графа прежде всего это приоритет 2) из проведенных r путей (2 путей). Дуга $r + 1$ проходит как и первая, но уже через другие вершины приоритета $r + 1$ и т.д., вплоть до соединения путями вершины первого приоритета со всеми вершинами приоритета R.
4. Все вершины R приоритета соединяются дугой с вершиной высшего приоритета.

5. Нумеруются пути в соответствии с очередностью их получения. Именно в соответствии с этой нумерацией (с этим порядком) в системе должны передаваться полномочия на доступ к ресурсу.
6. Для каждого полученного пути определяется очередность передачи полномочий между вершинами графа (очередями заявок) перечислением очередности появления соответствующих вершин на пути.

Иллюстрация получения поочередной передачи прав, с использованием приведенной процедуры представлена на рис. 18.5.

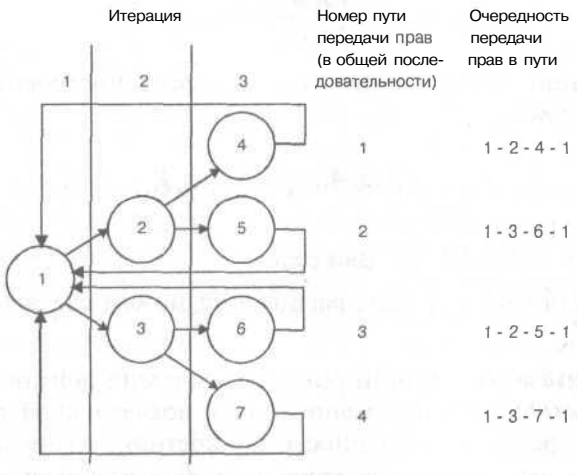


Рис. 18.5. *Радиальный граф с поочередной передачей прав*

Свойства канонических (оптимальных) расписаний реального времени

Можно выделить следующие свойства канонических расписаний реального времени:

1. В общем случае для каждой очереди заявок в цикле расписания может быть несколько очередностей передачи прав на доступ к ресурсу $s = 1,...,S$, например, (1, 2, 1, 3, 4). При этом параметры обслуживания заявок определяются гарантированной продолжительностью обслуживания $T_{го_{m,S}}$:

$$T^{г}_{го_m} = sup\{T_{го_{m(s)}}, s = \overline{1,S}\},$$

и средней продолжительностью гарантированного обслуживания требований с учетом различных очередностей S:

$$T^{*}_{го_{m(S)}} = \sum_{s=1}^{S} pT_{го_{m(S)}}.$$

Таким образом, в общем случае при нескольких очередностях в системе, дисциплина обслуживания реального времени задается параметрами $T^{z}_{zo_{m(S)}}$ и $T^{*}_{zo_{m(S)}}$. Для канонического расписания значения $T^{z}_{zo_{m(S)}}$ и $T^{*}_{zo_{m(S)}}$ соответственно совпадают для заявок всех R приоритетов.

2. В системе реализуются минимально возможные значения $T_{zo_{m(S)}}$ для всех M_r заявок всех R приоритетов.
3. Относительный уровень (коэффициент) приоритетности заявок m_r, m_r: $\delta_{r-r'}$ вырожденного графа при минимальных T_{zo_m} составляет:

$$\delta_{r-r'} = \frac{(RM_r - 1)}{(RM_{r'} - 1)}.$$

Соответственно, ограничением на общность построения радиального графа будет:

$$\frac{M_{r+1}}{M_r} = 2, 3, 4, \ldots; \; r = 1, \ldots, R.$$

Выводы

Из анализа свойств канонических расписаний можем сделать следующие выводы:

1. В основе синтеза расписаний реального времени должно находиться построение канонического расписания, с последующей его модификацией, учитывающей особенности конкретной задачи синтеза.
2. При минимизации значений параметра $T_{zo_{m_r}}$ для всех очередей заявок всех уровней приоритета системы R (эффективное использование ресурса) строго задается соотношение уровней коэффициента приоритетности заявок.
3. Исходное соотношение коэффициентов приоритетности может быть изменено либо подбором соответствующих значений параметров T_{zp_m} и T_{znn_m} в рамках канонического расписания, либо изменением значений T_{zo_m} или $T^{*}_{zo_m}$ (в общем случае они могут не совпадать), путем их увеличения для заявок отдельных уровней приоритета (уменьшить их уже невозможно, что следует из второго свойства канонических расписаний).

18.4.3. Задача включения приоритетного обслуживания в механизм контроля списков

Практическая реализация

Очевидно, что затраты на контроль необходимо минимизировать. Это позволит более эффективно использовать вычислительный ресурс для решения прикладных задач. На простом примере проиллюстрируем задачу включения в схему контроля приоритетного обслуживания.

Выше было показано, что ввиду различия физического смысла контролируемых событий, требования к обслуживанию в реальном масштабе времени для них могут сильно различаться (в разы). Это определяется как собственно процедурой контроля объектов, имеющих различный физический смысл, так и различным физическим смыслом вырабатываемых реакций.

Рассмотрим следующий вариант, когда одно из времен несанкционированного изменения списка (верхний предел времени для проведения проверки по данному списку) намного меньше остальных.

Введем следующее обозначение: $T_{ог.i}$ — задаваемое требованием к обслуживанию в реальном времени офаничение на время (продолжительность) контроля события (с учетом возможной реакции системы защиты).

Таким образом, исходным условием является то, что время $T_{ог.1} = T_{ог.}/k$, т.е. верхний предел времени для проведения проверки первого списка в k раз меньше, чем у остальных.

Длительность всех проверок для простоты примем равной $(T_{контр})$.

Очевидно, что при реализации системы реального времени все заявки на обслуживание должны обрабатываться за фиксированное время $T_{...}$. Тогда для случая с бесприоритетным обслуживанием (циклическое обслуживание запросов на проведение контроля всеми механизмами) получим временные затраты на проведение всех проверок:

$$T_{БП} = k\sum_{i=1} T_{контр_i} = kT_{контр_1} + k\sum_{i=2} T_{контр_i}.$$

Пусть контроль по первому списку, характеризуемому более жесткими ограничениями на параметр $T_{...i}$, осуществляется в k раз чаще. Введем приоритет обслуживания для контроля первого списка. В результате получим временные затраты на проведение всех проверок:

$$T_{ОП} = kT_{контр_1} + \sum_{i=2} T_{контр_i}.$$

Очередности обслуживания в бесприоритетном режиме и с приоритетным режимом для контроля первого списка представлены на рис. 18.6 и рис. 18.7.

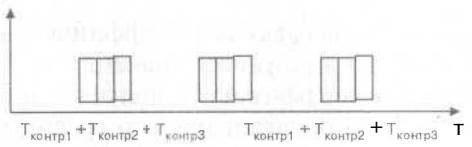


Рис 18.6. Очередность обслуживания при отсутствии приоритетов

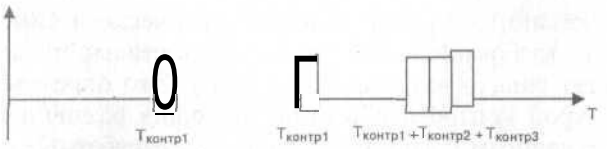


Рис 18.7. *Очередность обслуживания с приоритетом контроля первого списка*

Оценка выигрыша

Получаемый выигрыш от использования расписания проиллюстрирован на рис 18.8, 18.9, где:

$$\Delta = T_{БП} - T_{ОП} = \left(kT_{контр_1} + k\sum_{i=2} T_{контр_i}\right) - \left(kT_{контр_1} + \sum_{i=2} T_{контр_i}\right),$$

$$\Delta = T_{БП} - T_{ОП} = (k-1)\cdot\sum_{i=2} T_{контр_i}.$$

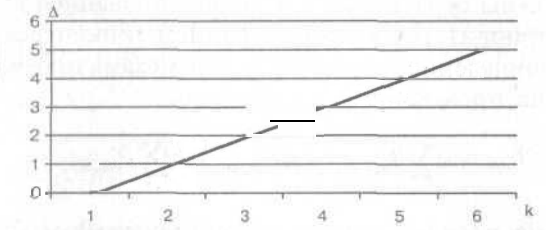


Рис. 18.8. *Выигрыш от включения в схему контроля приоритетного обслуживания, как функция* $\Delta = f(k)$

На рис 18.8 по оси k откладывался коэффициент k, по оси Д — получаемый выигрыш от использования приоритетного обслуживания по сравнению с бесприоритетным обслуживанием. По оси Д единицей измерения является $\sum_{i=2} T_{контр_i}$.

Из приведенного графика видно, что зависимость получаемого выигрыша (в единицах $\sum_{i=2} T_{контр_i}$)является линейной (прямая с коэффициентом наклона k — 1).

На рис. 18.9 по оси *(k) (i)* откладываются коэффициенты k и i. Коэффициент i для различных кривых принимает значения k, $3k$ и $1/2k$. По оси ординат отложены значения коэффициента получаемого выигрыша Д от использования приоритетного обслуживания по сравнению с бесприоритетным (получаемый выигрыш равен Д • $T_{к}$). Полученная зависимость является нелинейной.

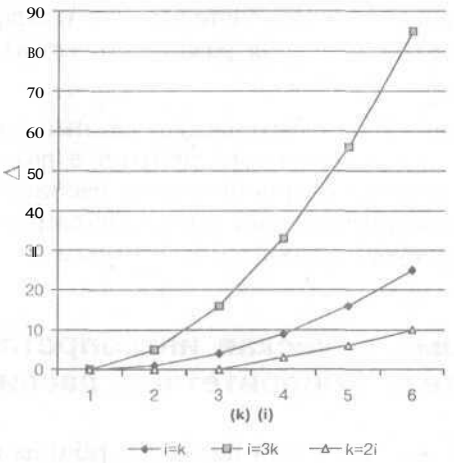


Рис. 18.9. *Выигрыш от включения в схему контроля приоритетного обслуживания, как функция A = f(k,i)*

Оценим полученный выигрыш в процентном отношении при заданных параметрах.

Пусть:

* отношение временных ограничений для первого списка и всех остальных $k = T_{огр i} / T_{огр1} = 10$;
* количество проверяемых списков $i = 10$;
* время контроля каждого списка $T_к = T_{конт} = 10$ [с].

При данных условиях получаемый выигрыш будет оцениваться:

$$\Delta = (k-1)\cdot \sum_{i=2} T_{контр_i} = (10-1)\cdot \sum_{i=2,10} 10 = 9\cdot 90 = 810 \text{ [с]},$$

$$T_{БП} = k\sum_{i=1} T_{контр_i} = 10\sum_{i=1,10} 10 = 1000 \text{ [с]},$$

$$T_{ОП} = kT_{контр_1} + \sum_{i=2} T_{контр_i} = 10\cdot 10 + \sum_{i=2,10} 10 = 100 + 90 = 190 \text{ [с]}.$$

Т.е. в данном случае выигрыш от реализации в системе приоритетного обслуживания составит:

$$\Delta / T_{БП} \cdot 100\% = 810/1000 \cdot 100 = 81\%.$$

Из графика зависимости полученного выигрыша A (единицей измерения является $T_{контр}$) видно, что с увеличением коэффициента i величина выигрыша растет быстрее. Т.е. наибольший выигрыш получается при боль-

шом количестве контролируемых списков. От коэффициента k (отношение временных ограничений для разных проверок) он зависит в меньшей степени.

С учетом сказанного может быть сделан важный вывод о том, что контроль списков санкционированных событий в реальном времени целесообразно осуществлять по приоритетным расписаниям. При этом может быть достигнуто значительное снижение затрат производительности вычислительного ресурса защищаемого объекта.

18.4.4. Геометрическая интерпретация задачи синтеза приоритетных расписаний

Геометрический подход в общем виде

Как было показано выше, в схему контроля списков санкционированных событий целесообразно включать приоритеты обслуживания. При этом особенность построения системы состоит в том, что обслуживание запросов на контроль должно осуществляться в реальном масштабе времени, причем как приоритетных, так и бесприоритетных, т.е. запросы должны обслуживаться по расписанию. Другими словами, в рассматриваемых приложениях приоритет заявки на обслуживание вводится не с целью предоставления преимущественного права одной заявки перед другой быть обслуженной, а с целью быть обслуженной за меньшее гарантированное время, определяемое с учетом исходных ограничений на систему контроля в реальном времени.

Задача синтеза расписания реального времени для метода уровневого контроля (для тех списков, где такое расписание возможно) заключается в составлении такого расписания, в котором промежутки времени между проверками будут наиболее велики, для того, чтобы высвободить максимум процессорного времени для выполнения прикладных программ. Другими словами, цель составления такого расписания — минимальное снижение системной производительности механизмами обеспечения безопасности. При этом, естественно, должно гарантироваться соблюдение правил защиты от несанкционированного воздействия.

Рассмотрим геометрическую интерпретацию задачи синтеза расписаний реального времени (в трехмерном пространстве) [22]. Пусть по осям n-мерной системы координат откладываются ограничения $T_{ог\ п}$ (верхний предел времени для проведения проверки соответствующего списка, который не должен нарушаться по требованиям функционирования системы в реальном времени). Время, необходимое для проведения контроля соответствующего списка, задается вектором $T_{конт\ п}$. Направление этого вектора указывает, какой список контролируется). Вектор $T_{\text{прикл}}$ соответствует оси времени для прикладных задач (см. рис. 18.10).

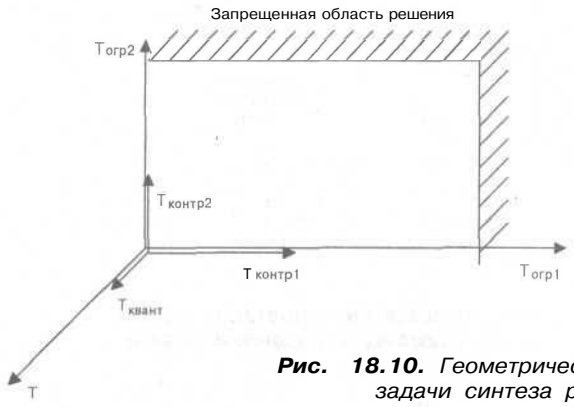


Рис. 18.10. *Геометрическая интерпретация задачи синтеза расписания*

Отложение вектора по соответствующей оси означает предоставление данному запросу кванта времени на обслуживание (соответственно $T_{к\ \ n}$ или T). Очевидно, при отложении вектора $T_{кон\ \ n}$ должна быть перенесена система координат.

Поясним сказанное. Допустим, мы отложили вектор $T_{контр1}$. Это означает, что проведен контроль по первому списку. При этом (после контроля) ограничения по первому списку будут иметь исходный вид $T_{огр\ 1}$. Текущее же значение ограничения по второму списку $T_{о\ 2}$ уменьшится и станет равным $T_{огр}2 \sim T_{контр\ 1}$, т.е. начало координат должно быть перемещено по оси $T_{огр\ 2}$ на величину вектора $T_{к\ \ 1}$. Значения $T_{огр1}$ и $T_{огр\ 2}$ задают на осях ограничения, которые не должны пересекаться откладываемыми векторами, образуя запрещенную область решения, пересечение которой означает невыполнения ограничений по обслуживанию в реальном времени.

Таким образом, задача синтеза расписаний состоит в выборе такой последовательности откладываемых векторов в рассматриваемой системе кординат (составление такого цикла — цикла расписания), при которой все события контролируются без пересечения запрещенной области решений. Соответственно, те кванты времени (отложенные векторы $T_{квант}$), которые могут быть размещены в промежутках между контролем, предоставляются для решения прикладной задачи.

Геометрическая интерпретация задачи синтеза расписания в ортонормированном базисе

Введем параметр «скорость продвижения к заданным ограничениям», для чего пронормируем исходную систему. Иллюстрация для двухмерного пространства приведена на рис. 18.11. По осям координат в нормированной системе откладываются векторы, соответствующие параметру «скорость продвижения к соответствующему временному ограничению»:

$$V_{контр.n} = T_{контр.\ n}\ /T_{огр\ n}.$$

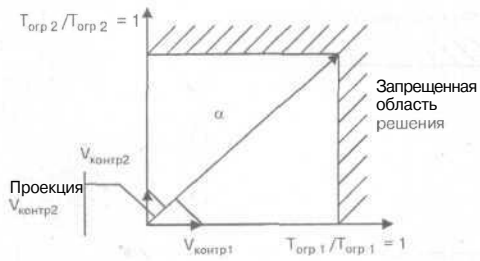


Рис. ***18.11.*** *Геометрическая интерпретация задачи синтеза расписания в ортонормированном базисе*

Задача выбора направления движения (проведения одной из проверок) интерпретируется следующим образом. Из начала системы координат к точке пересечения ограничений (в ортонормированном базисе задается значением $T_{огр\,i} / T_{огр\,i} = 1$) проводится идеальный вектор направления движения (а). Составление расписания в этом случае соответствует откладыванию по одной из осей скорости продвижения к ограничению. Выбор оси производится по критерию максимальной проекции на идеальный вектор. Т.е. движение происходит в направлении, максимально соответствующем идеальному вектору, который определяет направление, задающее максимальное расстояние до запрещенной области ограничений.

После проведения проверки, что соответствует откладыванию соответствующего вектора, система координат сдвигается на величину отложенного вектора по всем осям, кроме выбранной (на интервал, равный отложенной скорости). Данное действие соответствует уменьшению временных ограничений для проведения остальных проверок.

Последовательность откладываемых векторов образует цикл расписания. Требования к циклу таковы, что должен осуществиться контроль по всем событиям (по крайней мере, один вектор должен быть отложен по каждой оси). При этом векторы не должны пересечь запрещенную область ограничений. Таким образом определяется как расписание, так и требования к производительности вычислительного ресурса, позволяющего осуществлять рассматриваемую процедуру контроля в реальном времени. Расписание является в этой ситуации приоритетным, если в цикле расписания ветор по одному из направлений ограничений откладывается чаще, чем по другим направлениям.

Если цикл расписания не может быть реализован без пересечения запрещенной области, необходимо уменьшить параметры $T_{\quad n}$. Это соответствует увеличению мощности вычислительного ресурса.

Методика синтеза расписания

Рассмотрим в укрупненном виде методику синтеза расписания, которая состоит в выполнении последовательности шагов.

Прежде всего, исходя из требований к реализуемой системе, задаются значения следующих параметров: $T_{огрi}$; $T_{контрi}$; $T_{квант}$.

Затем проверяется условие возможности составления расписания реального времени (возможности обслуживания в реальном времени) для контроля списков:

$$\begin{cases} T_{контр_i} > T_{квант}, \\ \sum T_{контр_i} + T_{квант} < min\left\{T_{огр_i}\right\} \end{cases}$$

Если данное условие не выполняется, то необходимо изменить начальные ограничения ($T_{огрi}$; $T_{конт}$; $T_{квант}$) за счет увеличения производительности системы (уменьшая тем самым $T_{конт_i}$) или за счет уменьшения количества проверок (числа контролируемых списков).

После проверки выполнимости условия реализуется пошаговая процедура синтеза расписания, где на каждом шаге выполняются следующие действия:

1. Выбирается откладываемый вектор (контролируемый список).
2. Откладывается выбранный вектор (переносится система координат).
3. Проверяется выполнение условия реального времени -- отложенный вектор не должен пересечь запрещнную область решения. Если пересекает, то решить задачу для заданных условий невозможно – требуется либо увеличить производительность системы (уменьшая тем самым $T_{контрi}$), либо уменьшить количество проверок.
4. Проверяется, установился ли цикл расписания, т.е. обнаружена ли повторяющаяся последовательность контроля списков. Если не обнаружена — переход к следующему шагу. Если обнаружена — синтез расписания завершен.

18.5. Оценка влияния механизма уровневого контроля списков на загрузку вычислительного ресурса системы

Анализ влияния механизма на загрузку вычислительного ресурса в общем случае

Оценим влияние механизма уровневого контроля списков на загрузку вычислительного ресурса системы при решении системой прикладных задач, требующих различных временных затрат.

Выделим три класса задач по продолжительности их решения системой:

« быстрые;

» средние;

» медленные.

Пусть задачи данных классов будут иметь различную фиксированную продолжительность решения системой. Соответсвенно обозначим продолжительности как b1, *b2*, *b3*, соответствующие интенсивности обслуживания — m_1, m_2, m_3:

$$b_1 = 0{,}01(\text{с});\ m_1 = 100\ (\text{с}^{-1})$$
$$b_2 = 0{,}1(\text{с});\ m_2 = 10\ (\text{с}^{-1})$$
$$b_3 = 10(\text{с});\ m_3 = 1/10\ (\text{с}^{-1})$$

Пусть системой защиты реализуется три типа проверок (три контролируемых списка) и пусть соответственно параметрами для данных проверок будут:

$$T_{огр1} = 0{,}005\ (\text{с})$$
$$T_{огр2м} = 0{,}05\ (\text{с})$$
$$T_{огр3} = 5\ (\text{с})$$
$$T_{контр1} = 0{,}001\ (\text{с})$$
$$T_{контр2} = 0{,}001\ (\text{с})$$
$$T_{контр3} = 0{,}001\ (\text{с})$$

Для оценки влияния механизма уровневого контроля списков на загрузку вычислительного ресурса системы при решении этих задач воспользуемся моделью массового обслуживания. При этом в качестве критерия возьмем среднее время ожидания задачи в очереди.

Модель будет состоять из одного обслуживающего прибора и очереди заявок на обслуживание [6]. Входные параметры это модели таковы:

Я .... интенсивность входного потока заявок (имеет пуассоновский закон распределения. Данный закон распределения выбран, т.к. он описывает наиболее случайное распределение интенсивности входного потока заявок);

b .... время обслуживания заявки в приборе (для каждого класса задач).

Средняя длительность ожидания заявки в очереди (для пуассоновского потока) в общем случае пределяется по формуле:

$$W = \lambda/2 \cdot (\lambda - \rho) \cdot (\sigma^2 + 1/\mu^2),$$

где W - средняя длительность ожидания заявки в очереди;
μ - интенсивность обслуживания;
p - коэффициент загрузки системы;
a - отклонение длительности обслуживания;
$p = \lambda \cdot b = \lambda / \mu$ — коэффициент загрузки системы.

При постоянном времени обслуживания (s = 0) имеем:

$$W = \lambda/(2\mu^2 - 2\lambda*\mu^2 \text{ ДО } = \rho/2\mu \cdot (A - p).$$

Для каждого из введенных классов задач построим графики зависимости среднего времени ожидания задачи в очереди W от интенсивности входного потока задач $W = f(\lambda)$. Для оценки влияния механизма уровневого контроля списков на загрузку вычислительного ресурса системы сравним графики рассматриваемой зависимости для следующих случаев:

* без реализации в системе механизма контроля списков (решаются только прикладные задачи);
« при реализации механизма для двух случаев — с бесприоритетным и приоритетных обслуживанием.

Полученные зависимости для первого класса задач (быстрые) приведены на рис. 18.12. Полученные зависимости для второго класса задач (средние) приведены на рис. 18.13. Полученные зависимости для третьего класса задач (медленные) приведены на рис. 18.14.

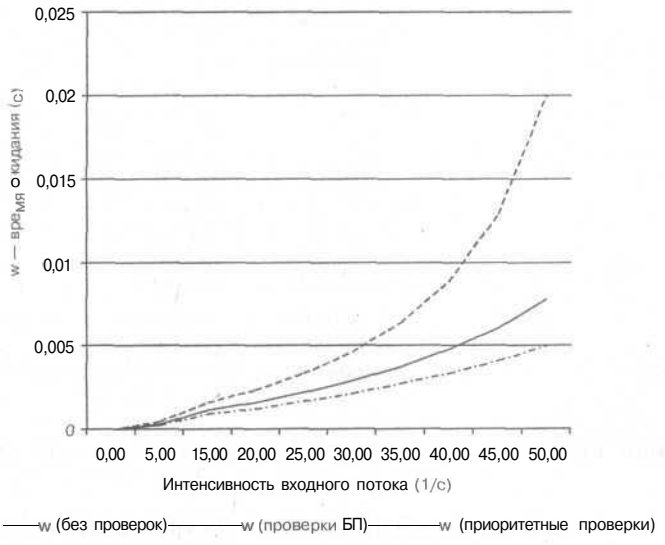


Рис. 18.12. *Среднее время ожидания задачи в очереди (первый класс)*

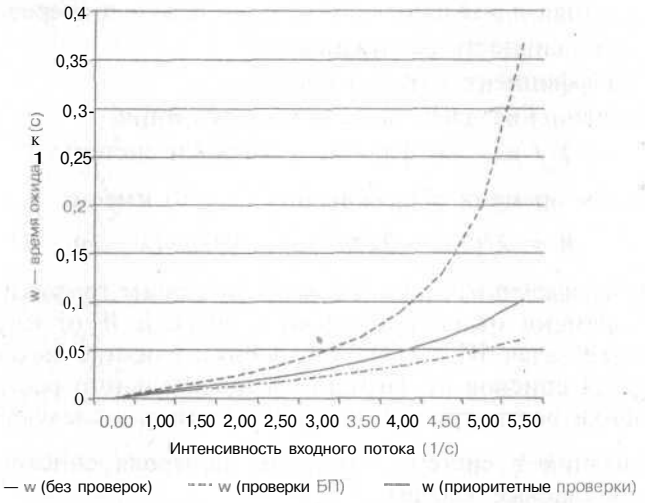


Рис. 18.13. *Среднее время ожидания задачи в очереди (второй класс)*

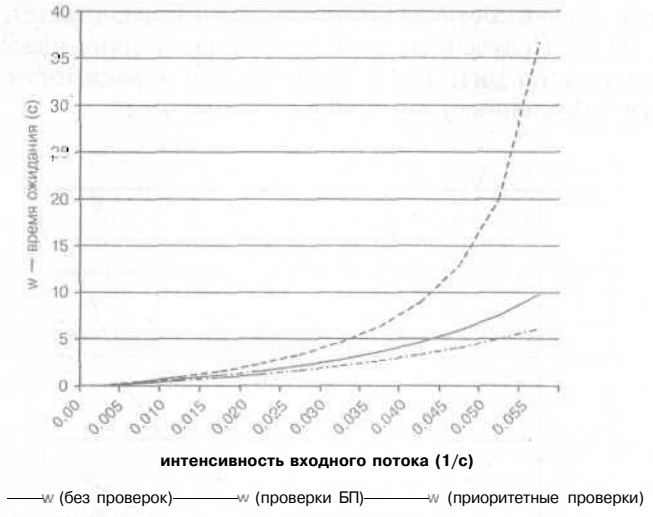


Рис. 18.14. *Среднее время ожидания задачи в очереди (третий класс)*

На основании проведенных исследований могут быть сделаны следующие выводы:

1. Увеличение загрузки вычислительного ресурса системы при использовании механизма уровневого контроля может быть достаточно велико, особенно при высокой загрузке системы. Это обусловливает необходимость обоснованного выбора контролируемых списков.

2. С целью более эффективного использования механизма целесообразна реализация приоритетного обслуживания заявок на контроль списков в реальном времени. Как видим из представленных графиков, потери производительности системы в этом случае могут быть сведены к минимуму.
3. Возможность более эффективного использования механизма также кроется в увеличении параметра $T_{оз}$. Данная возможность состоит в том, чтобы затруднить автоматизацию несанкционированных действий пользователя, в частности, если атака будет проводиться исполнением каких-либо действий с клавиатуры (не автоматически из созданной и несанкционированно запущенной им программы), то значение данного параметра увеличится на порядки и влияния на загрузку вычислительного ресурса при реализации данного уровня защиты будут минимальными.

Анализ контролируемых списков санкционированных событий

Теперь в двух словах остановимся на анализе контролируемых списков санкционированных событий с точки зрения задания ограничений $T_{озi}$. Очевидно, что можно выделить две группы атак на защищаемый объект – автоматическая и автоматизированная. Автоматическая атака реализуется разработкой и запуском на защищаемом объекте злоумышленником деструктивной программы (процесса). Эта программа автоматически выполняет все несанкционированные действия.

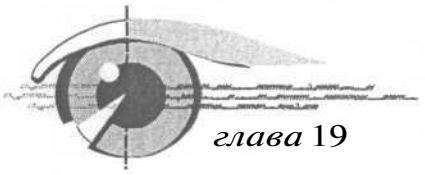
Очевидно, что значение ограничения $T_{оз\,i}$ для контроля деструктивных программ (процессов) должно быть минимально. Контролю списков запущенных процессов следует назначить максимальный приоритет. Другая группа атак, например, регистрация в системе несанкционированного пользователя, изменение ключа реестра ОС и т.д., предполагает либо выполнение ручных действий злоумышленника (в этом случае ограничения $T_{\text{ввода}}$. должны устанавливаться на несколько порядков больше) либо опять же запуск деструктивной программы, но, как отмечалось, контроль данного события должен проводиться с максимальным приоритетом.

Сказанное в полной мере подтверждает сделанные ранее выводы, что основным способом противодействия скрытым угрозам является противодействие запуску на защищаемом объекте несанкционированных пользовательских процессов (собственных программ пользователя).

18.6. Практические рекомендации по реализации механизма уровневого контроля

С учетом проведенного анализа сформулируем некоторые основные рекомендации по практической реализации механизма уровневого контроля в системе добавочной защиты.

1. При реализации механизма уровневого контроля могут быть выделены две группы событий. В первую группу включается контроль процессов, во вторую — все остальные функции контроля.
2. Для механизма контроля процессов должен устанавливаться максимальный приоритет, т.к. данный контроль характеризуется жесткими ограничениями $T_{огрi}$. Ограничения $T_{ог\,i}$ для остальных событий на несколько порядков больше. Таким образом, как было показано выше, в системе реализуются максимально благоприятные условия для приоритетного обслуживания (может быть достигнут максимальный результат за счет реализации приоритетных расписаний).
3. Как следует из рис. 18.12...18.14, при реализации приоритетной дисциплины обслуживания с выделением максимального приоритета одному механизму контроля — контролю списков запущенных процессов, достигается минимальная дополнительная загрузка вычислительного ресурса защищаемого объекта при реализации механизма уровневого контроля на всем интервале загрузки системы (в штатном режиме функционирования системы - - без перегрузок, данные потери находятся в пределе единиц процентов).
4. Возвращаясь к схеме реализации механизма уровневого контроля, представленной на рис. 17.2, отметим, что в задачи блока управления 4 должна быть включена задача диспетчеризации заявок по приоритетному расписанию, реализуемая в соответствии с рассмотренным выше способом (реализуется схемой, представленной на рис. 18.7).



глава 19

Механизмы контроля целостности файловых объектов

19.1. Задачи и проблемы реализации механизмов контроля целостности

Контроль целостности файловых объектов представляет собой самостоятельную задачу защиты информации. При этом основу механизмов контроля целостности файловых объектов представляет проверка соответствия контролируемого объекта эталонному образцу. Для контроля могут использоваться контрольные суммы и ряд иных признаков, например, дата последней модификации объекта и т.д. При необходимости содержать контролируемый объект в эталонном состоянии данные механизмы могут осуществлять автоматическое либо автомтизированное (под управлением пользователя или администратора безопасности) восстановление несанкционированно измененного файлового объекта из эталонной копии.

Основной проблемой реализации механизмов контроля целостности файловых объектов является их весьма сильное влияние на загрузку вычислительного ресурса системы, что обусловливается следующими причинами. Во-первых, может потребоваться контролировать большие объемы информации, что связано с большой продолжительностью выполнения процедуры контроля. Во-вторых, может потребоваться непрерывное поддержание файлового объекта в эталонном состоянии. При этом возникает вопрос: с какой частотой запускать продолжительную процедуру контроля? Если делать это часто, то получим существенное снижение производительности системы, если редко, то такой контроль, в частности, автоматическое восстановление объектов из эталонной копии, вообще имеет мало смысла.

Таким образом, основным вопросом при реализации механизмов контроля целостности файловых объектов является выбор принципов и механизмов запуска процедуры контроля (очевидно, что синхронный запуск — запуск контроля по расписанию, который без существенного снижения производительности системы может производиться достаточно редко, следует рассматривать в качестве дополнительного механизма).

Контролируемые файловые объекты могут быть классифицированы в соответствии с функциональным назначением: исполняемые файлы (программы) и файлы данных.

19.2. Асинхронный запуск процедуры контроля целостности и его реализация

Под асинхронным запуском понимаем запуск процедуры не периодически во времени, а по причине возникновения какого-либо события. Реализация асинхронного запуска процедуры контроля, в предположении оптимального задания подобных событий, может существенно снизить ее влияние на загрузку вычислительного ресурса защищаемого компьютера.

Рассмотрим возможные подходы к реализации асинхронного (причинного) запуска процедуры контроля целостности.

Запуск контроля целостности исполняемого файла

С точки зрения контроля исполняемых файлов асинхронный механизм запуска процедуры контроля интуитивно понятен — контроль следует запускать перед запуском программы -- перед чтением соответствующего исполняемого файла. Другими словами, система защиты должна перехватывать функцию чтения исполняемого файла (соответствующую API функцию — запрос от приложения к ядру ОС), запускать процедуру контроля и затем (при необходимости, уже после восстановления файла из эталонной копии) выдавать данную функцию на обработку в ядро ОС.

В данном случае достигается минимальное падение производительности системы, так как контроль осуществляется только в необходимые моменты — только при запуске программы. При этом предполагается, что программа запускается — значительно реже, чем осуществляется обращение к файлам данных.

Запуск контроля целостности как реакция механизма контроля списков санкционированных событий

С точки зрения контроля целостности файлов данных имеем принципиально более сложную ситуацию. Отличие от контроля исполняемых файлов здесь состоит в том, что к файлам данных обращения могут быть частыми. Поэтому механизм контроля перед чтением файла данных (по аналогии с контролем исполняемых файлов) может быть не всегда применим (использование данного подхода может привести к существенному влиянию механизма защиты на производительность системы).

В качестве альтернативного подхода к асинхронному запуску процедуры контроля целостности файлов данных может быть рассмотрен способ запуска процедуры контроля как реакции, вырабатываемой механизмом уровневого контроля списков санкционированных событий (механизм описан выше), что проиллюстрировано рис. 19.1.

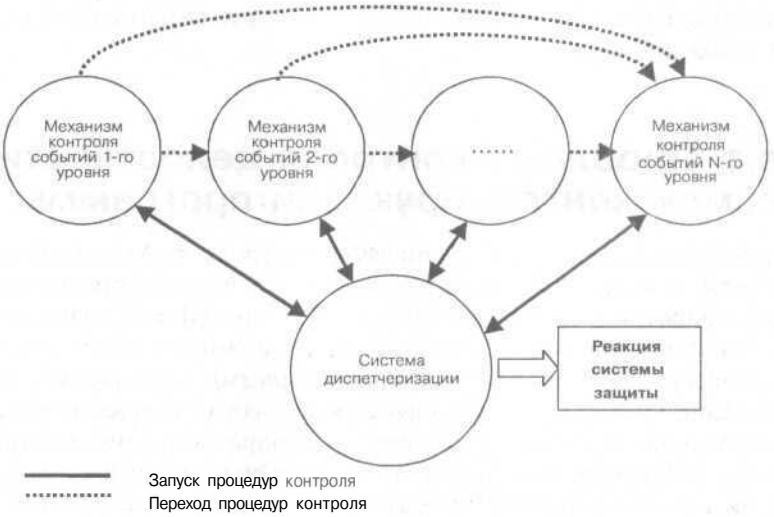


Рис. 19.1. *Схема асинхронного запуска процедуры контроля целостности*

Здесь механизм контроля событий N-го уровня не что иное, как контроль целостности файлов данных. Управление данному механизму передается остальными механизмами контроля событий (уровней 1, 2, ..., N~l) при обнаружении любым из этих уровней несанкционированного события в системе.

Таким образом, при данном способе контроля причиной запуска процедуры контроля файлов данных (асинхронный запуск) является наличие косвенных признаков несанкционированного доступа. При этом запуск процедуры контроля файлов данных осуществляется только при обнаружении в системе потенциально опасной ситуации — угрозы НСД. Благодаря этому минимизируются затраты вычислительного ресурса системы на выполнение процедуры контроля, т.к. в штатном режиме функционирования системы (при отсутствии угрозы НСД) контроль целостности не осуществляется.

Примечание

Синхронный контроль (по расписанию) здесь может использоваться в дополнение к процедуре причинного запуска процедуры контроля. Запуск процедуры по расписанию реализуется системой диспетчеризации,

Аналогичный асинхронный способ запуска может применяться и для решения задач очистки памяти. Кроме интуитивно понятных асинхронных способов, например, при запуске или завершении процесса, удалении файла, авторизации пользователя и т.д. (различные подходы применяются для гарантированной очистки оперативной и дисковой памяти), принудительная очистка памяти (например, оперативной) может осуществляться как реакция на факт обнаружения несанкционированного события.

19.3. Проблема контроля целостности самой контролирующей программы

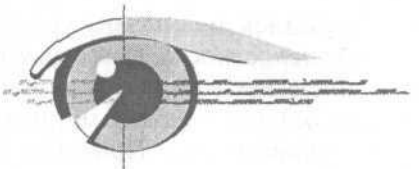
Другая проблема реализации механизмов контроля целостности состоит в следующем — если контроль целостности реализуется программно, то возникает вопрос контроля целостности и корректности функционирования собственно контролирующей программы (можно же исполняемый файл данной программы модифицировать первым). Естественно, что для решения данной задачи в общем случае необходимо осуществление контроля каким-либо внешним средством — аппаратной компонентой. Такой подход, например, используется в некоторых вариантах реализации аппаратной системы защиты «Электронный замок». Здесь платой контролируется целостность файлов еще до загрузки системы — загрузка системы (как следствие, системы защиты) при этом разрешается только в эталонном виде.

Однако это лишь частичное решение проблемы, т.к. остается задача контроля в процессе функционирования системы, которая вновь возлагается на программное средство. При этом опять же контролирующая программа может быть модифицирована, например, остановлено ее выполнение.

Для решения данной задачи может рассматриваться следующий альтернативный подход — контроль целостности возлагается на программную компоненту, аппаратная же компонента системы защиты обеспечивает корректное выполнение контролирующей программы. Данный подход позволяет решать задачу контроля в течение всего времени функционирования системы, предотвращая при этом возможность некорректного функционирования контролирующей программы, т.е. решать задачу в общем случае.

Поскольку решение данной задачи напрямую связано с функциями аппаратной компоненты системы защиты, рассматриваемый подход будет изложен в следующих главах (в части VI книги), посвященной применению аппаратных средств обеспечения компьютерной безопасности.

Применение средств аппаратной защиты



Часть VI

- Необходимость и принципы использования аппаратных средств защиты
- Технология программно-аппаратной защиты
- Метод контроля вскрытия аппаратуры

глава 20

Необходимость и принципы использования аппаратных средств защиты

20.1. Общие положения

В предыдущих главах рассматривались механизмы уровневого контроля списков и механизмы контроля целостности файловых объектов. Было сказано, что эти механизмы призваны воспрепятствовать преодолению системы защиты (контроль списков) или свести к минимуму последствия такого преодоления (контроль целостности). При этом было отмечено, что в общем случае невозможно осуществлять контроль активности одной программы над другой программой, запущенной на том же компьютере. Поэтому данная функция должна возлагаться на аппаратную компоненту системы защиты -- плату, устанавливаемую в свободный слот защищаемого компьютера.

В этом разделе книги будут рассмотрены методы и подходы по использованию аппаратных средств защиты, а также способы их комплексирования с программными средствами.

Сразу же отметим, что в связи с использованием аппаратной компоненты появляется новая угроза информационной безопасности — несанкционированное удаление аппаратной компоненты системы защиты. Поэтому в функциональную модель защиты необходимо привнесение отдельного уровня защиты -- **уровня контроля (мониторинга) наличия оборудования системы защиты.**

Примечание

Кстати говоря, данная угроза носит общий характер. Если злоумышленник получает доступ к аппаратуре, т.е. может бесконтрольно снять крышку корпуса защищаемого компьютера, у него появляется масса возможностей преодоления системы защиты, например, посредством сброса BIOS в исходное состояние.

20.2. Угрозы перевода системы защиты в пассивное состояние, их реализация

20.2.1. Классификация угроз перевода системы защиты в пассивное состояние

В общем случае угрозы перевода ПО системы защиты в пассивное состояние могут быть классифицированы следующим образом:

- угроза загрузки системы без механизмов защиты (загрузки ОС без полностью либо частично системы защиты);
- угроза перевода механизмов защиты (или системы защиты в целом) в пассивное состояние в процессе функционирования защищаемого объекта;
- если с целью добавочной защиты используется аппаратная компонента (осуществляющая мониторинг активности ПО системы защиты), то появляется дополнительная угроза — угроза удаления оборудования системы защиты (аппаратной компоненты) с целью последующего перевода ПО системы защиты в пассивное состояние одним из перечисленных выше способов.

Как и ранее, данные угрозы могут быть классифицированы как явные и скрытые. Скрытые угрозы могут быть осуществлены с использованием собственных программ злоумышленника, а также с использованием ошибок и закладок в ПО. Обобщенная классификация угроз перевода ПО системы защиты в пассивное состояние представлена на рис. 20.1.

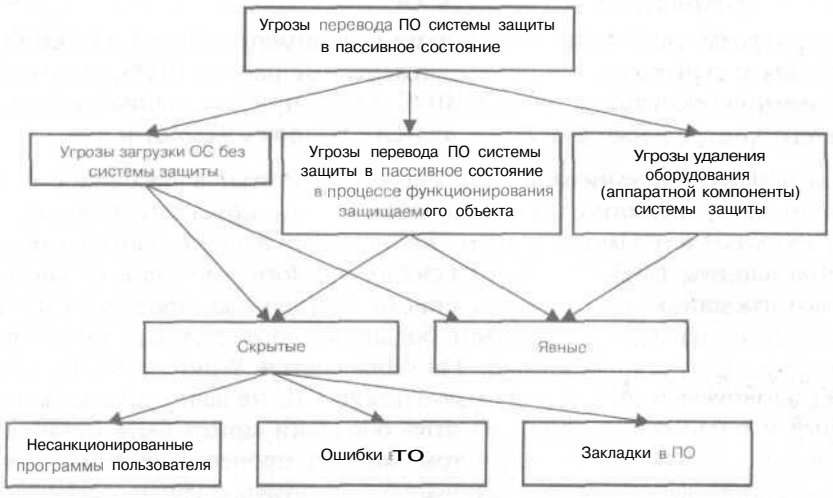


Рис. 20.1. *Классификация угроз перевода ПО системы защиты в пассивное состояние*

20.2.2. Анализ рассматриваемых угроз применительно к современным ОС

Рассмотрим данные группы угроз, применительно к ОС семейства Windows, Угроза загрузки системы без механизмов защиты (в предположении, что система защиты запускается как сервис при старте ОС) связана со следующими возможностями:

* **загрузка системы с альтернативных носителей,** например, с дискеты. В этом случае, поскольку загрузка ОС, в качестве сервиса которой должна запускаться система защиты, осуществляется не с жесткого диска, система защиты не загружается;
* **загрузка системы в безопасном режиме** (например, Safe Mode для ОС Windows), **либо в ином режиме, например, в DOS.** При этом можно блокировать загрузку драйверов (основных механизмов защиты), т.е. загрузить систему с урезанными функциями. Это не страшно для встроенный механизмов ОС — встроенная система защиты не позволит их отключить, но критично для механизмов добавочной защиты, т.к. для ОС драйверы и прикладные средства добавочной защиты являются внешними средствами. Поэтому она не обеспечивает противодействие возможному их отключению. Здесь же отметим, что принципиальным отличием загрузки в безопасном режиме для ОС UNIX и Windows (естественно, рассматриваем технологию NT) будет то, что для ОС UNIX подобную загрузку может осуществить только пользователь «root» после авторизации, в ОС Windows — любой текущий пользователь после авторизации.

Практически в полном объеме те же угрозы присущи и для ОС семейства UNIX, т.е. рассматриваемая задача защиты характеризуется общностью для альтернативных семейств ОС.

Данные угрозы могут быть как явными (например, загрузка в режиме Safe Mode), так и скрытыми, например, инженерные пароли BIOS, возможность программно сбросить настройки BIOS в исходное состояние (модифицировав его контрольные суммы — для ОС Windows 9x/Me) и т.д.

Угроза перевода механизмов (или системы защиты) в пассивное состояние в процессе функционирования защищаемого объекта связана с явными и скрытыми возможностями по остановке выполнения процесса системы защиты (либо вообще удаления данного процесса из системы). Так, возможностью отображения списка запущенных процессов и удаления процесса из списка обладают большинство системных мониторов и множество оболочек, например, Far. Для систем Windows 95/98, где все процессы запускаются как пользовательские (ОС не делит процессы на системные и пользовательские), из этой оболочки может быть остановлено выполнение любого процесса, в том числе и процесса системы защиты. Это очевидная явная угроза, основанная на архитектурных особенностях данной ОС. Однако можно предположить, что существуют и аналогичные скрытые угрозы. По крайней мере утверждать обратное невозможно.

С целью противодействия данным угрозам в систему защиты может добавляться аппаратная компонента. В этом случае, чтобы модифицировать ПО системы защиты злоумышленник сначало должен удалить аппаратную компоненту из слота защищаемого объекта, предварительно сняв крышку корпуса компьютера. Т.е. здесь появляется угроза физического удаления аппаратной компоненты системы защиты.

20.3. Методы противодействия угрозам перевода системы защиты в пассивное состояние

20.3.1. Методы противодействия загрузке ОС без ПО системы защиты

По способу реализации защиты в рассматриваемом случае могут быть выделены два класса методов:

» методы, реализующие программную защиту;

* методы, реализующие аппаратную защиту.

Все программные механизмы защиты в той или иной мере связаны с использованием или модификацией программного средства управления защитой — BIOS.

BIOS позволяет установить различные режимы загрузки. В частности, позволяет задать загрузку только с системного диска, на котором установлена ОС. При этом в качестве сервиса (или службы) последней запускается система защиты. На изменение заданного способа загрузки ОС средствами BIOS может быть установлен пароль.

С реализацией защиты подобным образом связаны следующие недостатки:

« известно применение в ряде программ BIOS инженерных паролей - паролей, ввод которых, вне зависимости от установленного пароля, позволяет получить доступ к настройкам способа загрузки ОС;

* с помощью специальных программ в ОС Windows 9x/Me настройки BIOS могут быть сброшены в исходное состояние, в котором по умолчанию задана критичная с точки зрения защиты системы очередность загрузки: в первую очередь загрузка производится с дисковода. Поэтому, если для защиты загрузки используется встроенный механизм аутентификации BIOS, то в системе защиты обязательно должен быть активным механизм обеспечения замкнутости программной среды, не позволяющий пользователю запустить программу сброса BIOS, а также осуществить иные действия по модификации BIOS.

Очевидно, что возможны два пути преодоления недостатков, связанных с использованием встроенных средств аутентификации пользователя BIOS, которые применяются на практике:

- расширение BIOS путем использования оригинальной (не встроенной) программы авторизации пользователя при доступе к настройкам BIOS;
- модификация программы BIOS путем усечения предоставляемых ею пользователю вариантов загрузки системы.

Первый подход реализуется с использованием широко применяемых сегодня аппаратных компонент (в частности, «Электронных замков»). Эти компоненты выполнены в виде специальных плат, устанавливаемых в свободный слот компьютера. В состав платы входит программно-аппаратное средство расширения функций BIOS.

При этом один из возможных известных способов защиты состоит в том, что при загрузке система BIOS наделяется дополнительной функцией, реализуемой платой — функцией авторизации пользователя (для ввода пароля пользователя здесь часто используются различные аппаратные ключи).

Второй подход связан с возможностью программным образом (перепрограммированием) закомментировать некоторые строки текста программы BIOS. При этом в качестве способа загрузки может быть установлен только один способ — загрузка с необходимого логического диска винчестера. Причем этот способ загрузки может быть установлен как способ загрузки по умолчанию. Тогда даже знание пароля не позволит изменить способ загрузки, т.к. он задан в BIOS как единственный. Естественно, что реализации данного подхода также присущи все недостатки программной реализации. В частности, здесь также обязательно использование механизма обеспечения замкнутости программной среды. Иначе тем же программным средством можно вернуть прошивку BIOS в исходное состояние.

По своей сути реализация функций расширения BIOS относится к программным методам защиты. Плата просто используется для размещения соответствующей микросхемы. Поэтому в дополнение в большинстве «Электронных замков» реализуется отключение внешних устройств на аппаратном уровне. Таким образом дисковод, CD-ROM и иные устройства ввода отключаются аппаратно. На них либо не подается питание, либо физически «разорвана» шина данных. Подключаться устройства должны каким-либо внешним сигналом, например, после завершения авторизации пользователя.

Таким образом, можем сделать следующий вывод: **противодействие исходной загрузке ОС без ПО системы защиты может быть эффективно оказано применением «Электронного замка», реализующего расширение BIOS в виде дополнительной авторизации пользователя перед загрузкой системы. При этом должно соблюдаться условие аппаратного отключения внешних устройств ввода до завершения авторизации санкционированного пользователя. Благодаря этому на данном этапе загрузки не может быть запущен пользовательский процесс, не могут использоваться инженерные пароли BIOS. Режим загрузки может быть изменен только санкционированным пользователем.**

20.3.2. Метод противодействия переводу ПО системы защиты в пассивное состояние в процессе функционирования системы

В предыдущем разделе было рассмотрено, каким образом защититься от несанкционированной загрузки ОС без системы защиты. Однако, как отмечалось ранее, это лишь одна составляющая рассматриваемой проблемы. Другая составляющая — это противодействие отключению ПО системы защиты во время функционирования защищаемого объекта.

Все главы предыдущей части были посвящены этому вопросу. При этом предлагалось использовать для этого метод уровневого контроля списков. Однако, как уже неоднократно упоминалось, невозможно осуществлять контроль активности одной программы другой программой, запущенной на том же компьютере. Поэтому в данной части будет изложено использование механизмов аппаратной защиты, призванных решить эту проблему. В рамках этого подхода предлагается использование программно-аппаратного комплекса, общая схема которого показана на рис. 20.2.

Рис. 20.2. *Иллюстрация программно-аппаратной технологии контроля (мониторинга) активности системы защиты*

глава 21

Технология программно-аппаратной защиты

21.1. Реализация программно-аппаратного контроля (мониторинга) активности системы защиты

Назначение аппаратной компоненты защиты загрузки ОС

Основу рассматриваемой технологии составляет реализация аппаратной компоненты защиты, т.к. невозможно в принципе контролировать выполнение одной программой другой программы, установленной на том же компьютере.

Реализуется аппаратная защита загрузки ОС следующим образом [10, 14, 15]. В компьютер в свободный слот устанавливается аппаратная компонента системы защиты (плата). Данная плата выполняет следующие функции:

* осуществляет в исходном состоянии отключение внешних устройств загрузки (дисковод, CD-ROM);
* обеспечивает возможность подключения внешних устройств при получении сигнала от системы защиты (например, питание внешних устройств заведено через реле на плате; подавая сигнал на реле, плата может подключать внешние устройства);
* обеспечивает возможность получения командного сигнала на включение внешних устройств со стороны программной компоненты системы защиты. Таким образом, плата пассивна, взаимодействие с нею программной компоненты осуществляется через заданный на плате и в программе защиты порт. С целью предотвращения возможности выдачи командного сигнала на плату пользовательской программой, в качестве командного сигнала используется парольное слово.

Примечание

Можно усилить процедуру аутентификации платой системы защиты, например, ограничив число получения неверных парольных командных слов. При превышении их числа плата может блокировать работу пользователя. В простейшем случае — выдавать команду «Reset» на перезагрузку компьютера, либо разрывать шину данных, отключать устройства ввода и т.д.

Подход к реализации программно-аппаратного контроля активности системы защиты

Подход к реализации программно-аппаратного контроля активности системы защиты проиллюстрирован рис. 21.1.

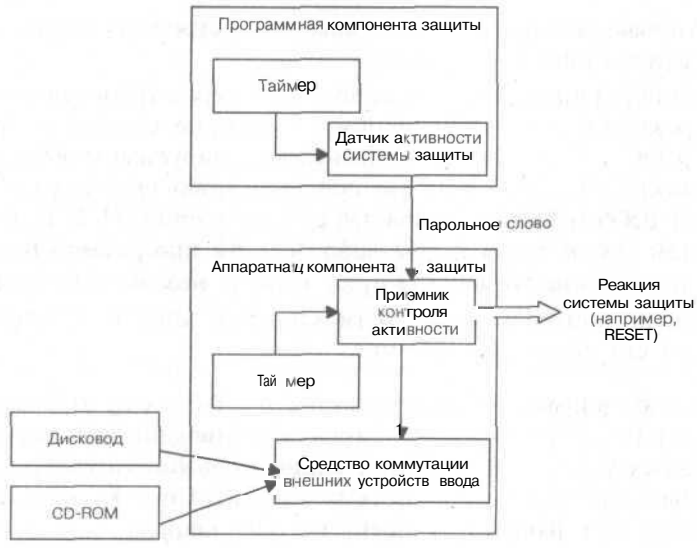


Рис. 21.1. *Укрупненная схема реализации программно-аппаратного контроля активности системы защиты*

Контроль (мониторинг активности) осуществляется следующим образом. Аппаратная компонента имеет в своем составе:

- средство коммутации внешних устройств ввода (например, через реле размыкается питание либо шина данных);
- приемник сигнала контроля активности;
- таймер.

Программная компонента содержит таймер и датчик активности системы защиты. Датчик активности с задаваемым таймером периодом выдает парольное слово на определенный порт (в плату). Получая данный сигнал, плата определяет активность ПО системы защиты. Если за заданный интервал времени плата не получит парольное слово (сигнал активности), то это будет означать перевод ПО системы защиты в пассивное со-

стояние. В ответ на это плата выработает либо сигнал «Reset», который осуществит перезагрузку системы, либо другую заданную реакцию.

Внешние устройства в исходном состоянии отключены. Подключаются они платой только после получения ею первого сигнала (парольного слова) от программной компоненты.

С целью предотвращения эмуляции ПО системы защиты вводится парольное слово. При N-кратной неверной передаче пароля на плату, плата считает, что система защиты не выполняет своих функций и отправляет компьютер на перезагрузку.

Таким образом, преимуществом данного подхода является следующее:

- « гарантированное подключение внешних устройств только при условии загрузки системы защиты;
- аппаратная защита загрузки системы с альтернативных носителей. При этом реализация защиты никоим образом не связана с программой BIOS и ОС, с их потенциальными недекларируемыми возможностями, с возможными действиями по использованию штатных свойств программы BIOS и ОС для отключения расширений BIOS и платы. Если в данном случае плата каким-либо образом программно будет отключена, то внешние устройства будет вообще невозможно подключить;
- » аппартный контроль активности системы защиты в течение всего времени его функционирования.

Особенностью данного подхода является то, что он (в отличие от аппаратного средства «Электронный замок») противодействует всей совокупности скрытых угроз, т.к. неважно, каким образом злоумышленник модифицировал или отключил систему защиты (это не анализируется), анализируется сам факт активности системы защиты.

Очевидно, что данный механизм может быть использован и для защиты загрузки системы в безопасном режиме (режим Safe Mode для ОС семейства Windows), т.к. его применение не позволит функционировать системе в частично защищенном виде (все необходимые программы и драйвера системы защиты должны быть загружены и активны).

Функциональная схема программно-аппаратного контроля

Пример реализации схемы программно-аппаратной защиты [28, 29] приведен на рис. 21.2.

На схеме использованы следующие обозначения: программная компонента системы защиты 1, аппаратная компонента системы защиты 2. При этом, программная компонента системы защиты 1 содержит блок управления 3, аппаратная компонента системы защиты 2 содержит блок аутентификации программной компоненты 7, блок контроля и управления 8, блок отключения внешних устройств 9.

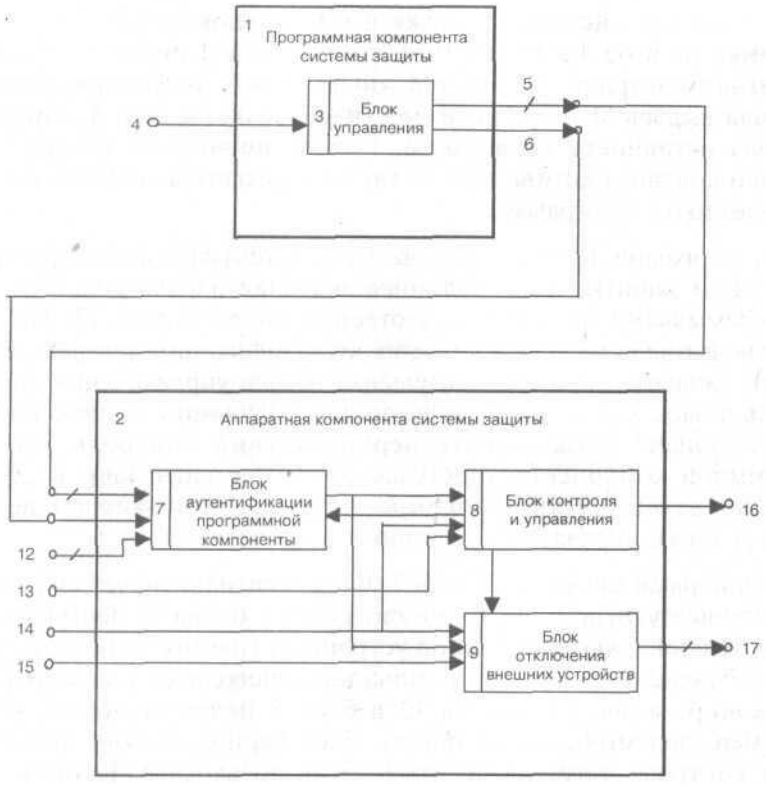


Рис. 21.2. *Функциональная схема реализации программно-аппаратной защиты*

Работает схема следующим образом. Программная компонента 1 встраивается в общем случае в ту программу, контроль активности которой необходимо осуществлять (в нашем случае в программный комплекс защиты информации). Аппаратная компонента системы 2 реализуется на плате, встраиваемой в свободный слот компьютера. При этом крышка компьютера должна быть надежно закрыта, а компьютер опечатан во избежание удаления платы из слота.

Питающее напряжение на все внешние носители (дисковод, CD-ROM), кроме винчестера, подаются на данные устройства с соответствующего выхода 17 аппаратной компоненты 2, куда оно подается со входа 15. В функции блока отключения внешних устройств 9 входит обеспечение возможности загрузки системы с того носителя, где располагается контролируемая программа — программная компонента 1 (винчестер). Остальные внешние устройства в исходном состоянии отключены.

При включении системы, а также в нормальном режиме ее функционирования на вход 4 программной компоненты 1 подается периодический сигнал контроля активности, по которому устройство управления 3 сначала выдает М-разрядный код (пароль) на выходы 5, затем сигнал контроля активности на выход 6. Пароль необходим для того, чтобы функцию контроля активности не могла перехватить подставленная злоумышленником программа.

Пароль со входов 10 поступает на блок аутентификации программной компоненты защиты, куда эталонное значение пароля подается (например, перемычками на плате) с соответствующих входов 12. По сигналу контроля активности осуществляется аутентификация программной компоненты защиты и при положительном исходе управляющий сигнал передается в блок контроля и управления 8. В функции данного блока входит следующее: осуществлять периодический контроль активности программной компоненты (поступления от нее сигналов); в случае обнаружения дезактивации программы — перевод компьютера в нерабочий режим, а также управление внешними устройствами.

Получив первый сигнал от блока 7, блок 8 сигнализирует ему, что контроль прошел успешно. Этим же сигналом с блока 7, поступающим на блок 9, разблокируются внешние устройства (на них с соответствующих входов 17 подается питание), компьютер переводится в штатный режим функционирования. Со входов 13 в блок 8 подается период контроля (например, перемычками на плате). Этот период должен превосходить период контроля активности, задаваемый со входа 4. Если за период, задаваемый со входов 13, программная компонента 1 передала сигнал (с предварительной аутентификацией), компьютер остается в штатном режиме. В противном случае блок контроля и управления 8 формируется сигнал «Сброс (Reset)» на выходе 16 и одновременно размыкает внешние устройства с выхода 17. После этого компьютер может быть включен лишь посредством загрузки операционной системы с винчестера, при условии, что одновременно загружается и контролируемая программная компонента защиты, в противном случае опять будет выработан сигнал сброса. Входным сигналом «Сброс (Reset)» система переводится в исходное состояние.

С учетом сказанного можем сделать вывод, что преимуществом реализации рассмотренной технологии программно-аппаратной защиты является то, что противодействие оказывается всей совокупности угроз (как угрозе загрузки системы без ПО системы защиты, так и угрозе перевода ПО системы защиты в пассивное состояние в процессе функционирования). Причем противодействие осуществляется вне зависимости от того, какой канал перевода программной компоненты системы защиты в пассивное состояние использован злоумышленником, в том числе и неизвестный скрытый канал, т.е. задача защиты решается в общем виде.

21.2. Метод контроля целостности и активности программных компонент системы защиты программно-аппаратными средствами

Как отмечалось ранее, при загрузке системы должна контролироваться целостность (неизменность) системы защиты. Только при реализации подобного контроля можно гарантировать, что система защиты потенциально способна реализовать свои функции корректно и в полном объеме (почему потенциально, объясним в следующем разделе). Также должна контролироваться целостность (неизменность) объектов ОС и системы защиты, включая настройки механизмов защиты, и в процессе ее функционирования.

В общем случае система защиты, как правило, представляет собою достаточно сложный программный комплекс. Этот комплекс состоит из нескольких программных модулей и содержит в своем составе как драйверы, так и прикладные программы (программирование системы защиты реализуется, как на системном, так и на прикладном уровнях).

При этом механизмы контроля целостности, осуществляющие контроль неизменности объектов файловой системы не гарантируют, что соответствующие процессы и драйверы системы защиты запущены. Кроме того, на защищаемом объекте могут находиться дополнительные программные средства, активность которых в процессе функционирования системы должны гарантироваться. Например, может находиться отдельная программа шифрования дисков, либо канала связи, программа антивирусного контроля. Если на защищаемом компьютере реализуется межсетевой экран, то всегда должна быть активна программа межсетевого экранирования и т.д. При этом данные программы могут быть различных производителей, что затрудняет встраивание в каждую из них программной компоненты защиты от перевода в пассивное состояние.

Предлагаемый метод контроля (мониторинга), осуществляемый программной компонентой системы защиты, состоит в комплексировании механизма уровневого контроля списков санкционированных событий с механизмом аппаратного контроля активности программного обеспечения, рассмотренным выше. В качестве санкционированных событий здесь следует рассматривать списки программ (процессов) и драйверов, которые обязательно должны быть запущены в системе [24, **25,** 27], а также контрольные суммы (и другие данные), характеризующие целостность ПО и настроек системы защиты.

Если не запущена какая-либо программа или драйвер из списка, либо изменены контрольные суммы контролируемых объектов системы защиты, то в качестве реакции на данное событие можно рассматривать пре-

кращение подачи парольного кодового слова программной компонентой контроля активности в аппаратную компоненту, что приведет к блокированию функционирования системы в нештатном для нее виде. Если драйверы и приложения должны загружаться при старте системы, то после перезагрузки системы, запускаемой аппаратной компонентой, возможно возвращение системы в исходное (штатное) состояние.

Таким образом, реализуется следующий подход к решению задачи контроля целостности. Контроль целостности осуществляется программно, но активность и корректность контролирующей программы обеспечивается применением аппаратной компоненты защиты (т.е. контролирующая программа, в свою очередь, контролируется аппаратной компонентой системы защиты).

Иллюстрация реализации механизма уровневого контроля, в части контроля целостности и активности программных компонент системы защиты, представлена на рис. 21.3.

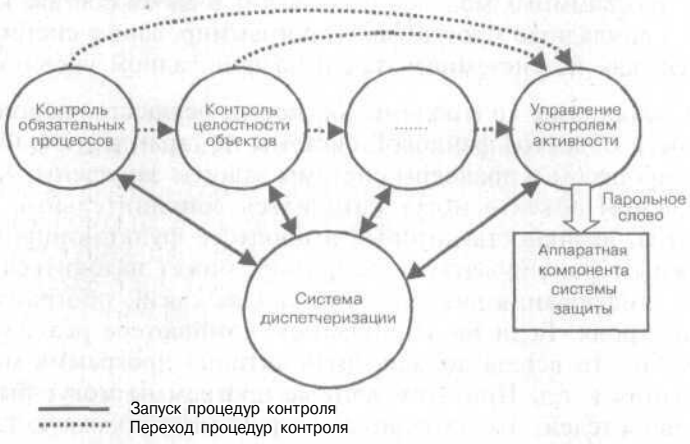


Рис. 21.3. *Иллюстрация реализации механизма уровневого контроля в части контроля целостности и активности программных компонент системы защиты*

21.3. Механизм удаленного (сетевого) мониторинга активности системы защиты, как альтернатива применению аппаратной компоненте защиты

Очевидно, что наиболее эффективное противодействие переводу системы защиты в пассивное состояние может быть оказано с использованием аппаратной компоненты защиты, которая обеспечивает мониторинг активности системы защиты и автоматическую реакцию на обнаруженный факт

перевода системы безопасности в пассивное состояние. Однако использование аппаратной компоненты приводит к дополнительному увеличению стоимости системы защиты (дополнительная компонента должна устанавливаться на каждый защищаемый объект, соответственно, на наиболее критичные объекты). Рассмотрим, можно ли осуществлять мониторинг активности системы защиты без применения аппаратной компоненты и каковы ограничения на эффективность использования данного подхода.

Как отмечалось ранее осуществлять мониторинг активности одной программы другой программой, установленной на том же компьютере, не имеет смысла. Остается единственная возможность осуществления мониторинга без применения аппаратной компоненты защиты -- осуществлять его удаленно из сети: с рабочего места (выделенного компьютера) сервера безопасности.

Примечание

При защите компьютеров в составе ЛВС, как правило, используются сетевые системы защиты, в состав которых, помимо клиентской части, устанавливаемой на защищаемый объект и реализующей собственно механизмы защиты, включается серверная часть — выделенный компьютер администратора безопасности, либо сервер безопасности.

Различаемые состояния защищаемого объекта и их выявление

В задачи сервера безопасности, как удаленного интерфейсного модуля системы защиты, входят: удаленная настройка механизмов защиты, удаленная обработка журналов аудита и удаленный контроль действий пользователя на защищаемом объекте. В соответсвии с рассматриваемым подходом сервер безопасности решает задачу удаленного мониторинга активности системы защиты.

Основу подобного мониторинга составляет возможность различать три состояния защищаемого объекта:

- **штатный режим функционирования:** защищаемый объект включен по питанию, находится в сети, и на нем активна система защиты;
- **режим отключения объекта:** объект отключен либо по питанию, либо находится не в сети (например, отсоединен сетевой кабель);
- **опасный режим функционирования:** объект включен, находится в сети, но на нем система защиты переведена в пассивное состояние.

Выявление опасного режима крайне важно, т.к. при этом фиксируется возможность удаленной атаки на сетевые объекты (компьютер не защищен) [10, 16].

Очевидно, что при использовании подобной модели контроля активности системы защиты (без применения аппаратной компоненты) должна использоваться технология хранения и обработки данных на файл-серверах. Соот-

ветственно, данная технология может эффективно использоваться в системах обработки баз данных, например, с использованием СУБД, реализующей централизованное хранение и обработку баз данных на сервере.

Выявление опасного режима функционирования компьютера означает, что на данном компьютере пассивна система защиты. С учетом этого реакцией администратора безопасности может служить только противодействие доступа с этого компьютера к другим защищаемым объектам. Для этого администратор должен воспользоваться настройками системы защиты, установленной на сервере, и запретить удаленный доступ к ней с незащищенной рабочей станции. При этом какие-либо действия с рабочей станцией, на которой система защиты переведена в пассивное состояние, администратор безопасности удаленно совершить не может, так как удален соответствующий инструментарий — клиентская часть системы защиты.

Способ реализации удаленного мониторинга системы защиты

Метод удаленного мониторинга активности системы защиты с целью различия трех состояний защищаемого объекта состоит в следующем. После установления соединения клиентской части системы защиты с сервером безопасности, сервер безопасности находится в режиме установленного соединения с клиентской частью. Соединение клиентской части с серверной реализуется по защищенному протоколу (как правило, поверх стека протоколов TCP/IP) с сеансовой авторизацией и шифрованием трафика.

При разрыве соединения сервер безопасности осуществляет попытку установить тестовое соединение с клиентской частью с использованием какого-либо открытого протокола или команды, например, ping, реализуемых на уровне ядра ОС. Если активность рабочей станции подтверждена, то делается вывод, что она находится в опасном режиме — включена по питанию, в сети (т.к. отвечает на стандартный запрос), но на ней отсутсвует система защиты (т.к. не устанавливается клиент-серверное соединение системы защиты).

Зафиксировав данный режим функционирования рабочей станции, администратор безопасности с сервера безопасности средствами удаленной настройки механизмов защиты клиентской части может осуществить логическое отключение серверов от опасной рабочей станции. Для оперативности реакции данная процедура может быть частично либо полностью автоматизирована.

Функциональная схема системы удаленного мониторинга активности системы

Пример реализации схемы удаленного мониторинга активности системы защиты [26], использованной автором в КСЗИ «Панцирь», приведен на рис. 21.4. Пример реализации системы-менджера защиты информации, устанавливаемой на выделенный сервер безопасности, приведен на рис. 21.5.

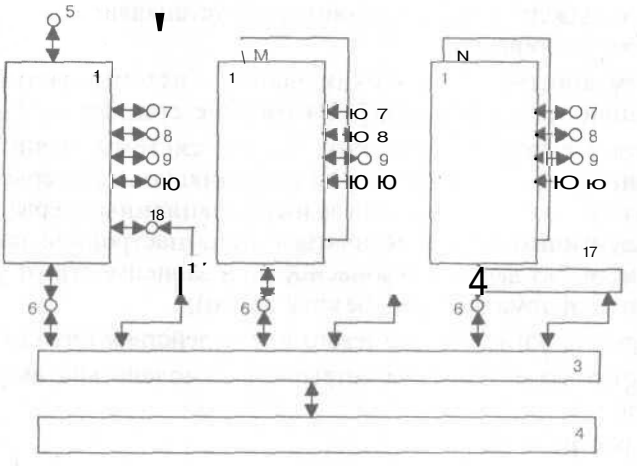


Рис. 21.4. *Пример реализации схемы удаленного мониторинга активности системы защиты*

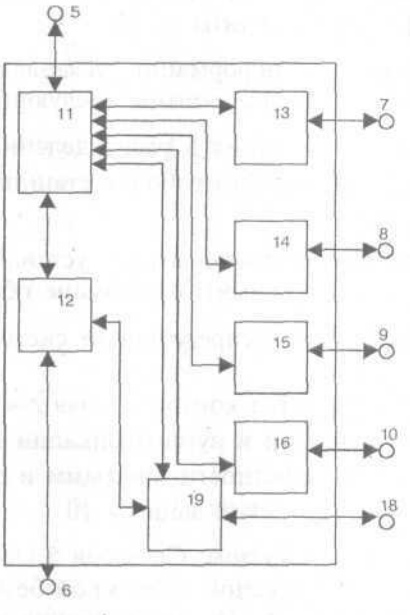


Рис. 21.5. *Пример реализации системы-менджера защиты информации, устанавливаемой на выделенный сервер безопасности*

На схемах, приведенных на рис. 21.4 и рис. 21.5, использованы следующие обозначения:

- система-менджер защиты информации, устанавливаемая на выделенный сервер безопасности -- 1;
- M систем-агентов (клиентских частей системы защиты) защиты информации, устанавливаемых на рабочие станции — 2;
- N систем-агентов (клиентских частей системы защиты) защиты информации, устанавливаемых на информационные серверы — 2 (системы-агенты рабочих станций и информационных серверов в общем случае идентичны, могут отличаться лишь настройкой параметров -- содержимым баз данных безопасности, в зависимости от реализуемой политики информационной безопасности);
- блок открытого интерфейса сетевого взаимодействия низкого уровня — 3;
- блок открытого интерфейса сетевого взаимодействия высокого уровня — 17;
- связной ресурс — 4;
- блок администрирования системы защиты — 11;
- блок закрытого интерфейса сетевого взаимодействия высокого уровня -- 12;
- блок разграничения и контроля прав доступа -- 13;
- блок идентификации и аутентификации -- 14;
- блок контроля целостности программ и данных — 15;
- блок криптографической защиты -- 16.

В системе менеджер-защиты информации, устанавливаемой на выделенный сервер безопасности 1,использованы следующие обозначения:

- вход/выход администрирования 5 распределенной системы защиты;
- вход/выход анализа активности рабочих станций и информационных серверов 18.

В N системах-агентах защиты информации, устанавливаемых на информационные серверы 2, использованы следующие обозначения:

- вход/выход подключения распределенной системы защиты 6 к связному ресурсу;
- вход/выход разграничения и контроля прав доступа 7;
- вход/выход идентификации и аутентификации пользователя 8;
- вход/выход контроля целостности программ и данных 9;
- вход/выход криптографической защиты 10.

Работает схема следующим образом. Со входа 5 администратор безопасности в соответствии с реализуемой политикой безопасности настраивает параметры блоков 13, 14, 15, 16, реализующих известные механизмы защиты информации, собственно системы-менеджера защиты информа-

ции, устанавливаемой на выделенный сервер безопасности 1, а через блоки 12, 3, посредством связного ресурса 4 настраивает параметры блоков 13, 14, 15, 16 всех M систем-агентов защиты информации, устанавливаемых на рабочие станции 2; и всех N систем-агентов защиты информации, устанавливаемых на информационные серверы 2.

Со входов 7, 8, 9, 10 системы защиты 1 и 2 реализуют принципы локальной защиты сервера безопасности, рабочих станций и информационных серверов. Адаптивно к обнаруженным попыткам несанкционированного доступа к информации на отдельных рабочих станциях и информационных серверах администратор безопасности со входа 5 системы защиты 1 имеет возможность осуществить перенастройку параметров отдельных блоков (13, 14, 15, 16) отдельных систем защиты 2, чем достигается противодействие угрозам (причем изменяющимся во времени) информационной безопасности вычислительной системе или сети.

В случае, если блоком 19 системы-менеджера 1 фиксируется невозможность взаимодействия менеджера с каким-либо агентом (рабочей станцией, либо информационным сервером) с использованием закрытого протокола распределенной системы защиты, данный блок запускает команду обращения менеджера системы защиты к рассматриваемому агенту с использованием открытого сетевого протокола высокого уровня — запускается взаимодействие через блок 19 (например, запускается команда Ping). Отметим, что блок 19 входит в состав ядра операционной системы.

Если блок 19 агента, к которому осуществляется обращение, отвечает, это означает, что на тестируемой рабочей станции отсутствует агент системы защиты, в противном случае — рабочая станция 19 отключена от питания либо от сети.

Примечание

Без блока 19, реализующего открытый интерфейс сетевого взаимодействия высокого уровня, для данной рабочей станции невозможно взаимодействие с другими рабочими станциями и информационными серверами, т.е. заблокирован доступ к информационным серверам.

Таким образом, обеспечивается возможность разделения двух событий — событие «рабочая станция отключена» и событие «на рабочей станции удален агент системы защиты». Дальнейшие действия администратора безопасности должны быть определены политикой информационной безопасности предприятия. В частности, в случае обнаружения факта удаления агента системы защиты на рабочей станции (фиксируется попытка несанкционированного доступа к информации), администратору безопасности через вход 5 системы 1 целесообразно изменить параметры блока 13 информационных серверов (систем 2), в части запрета доступа к ним с рабочей станции, на которой зафиксирована попытка несанкционированного доступа. Затем уже могут быть приняты соответствующие организационные мероприятия по восстановлению агента системы защиты и поиска злоумышленника.

Выводы по использованию метода удаленного мониторинга:

Итак, в рамках изложенного выше можно сделать следующие выводы:

1. В качестве альтернативы использования аппаратной компоненты для реализации контроля (мониторинга) активности системы защиты может использоваться метод сетевого контроля. При этом не требуется установка аппартной компоненты, а контроль осуществляется с сервера безопасности.
2. Условиями возможного использования сетевого метода контроля являются:
   - централизованный характер обработки и хранения данных в ЛВС (файл-серверы, технологии, использующие СУБД и т.д.) — защищаемые данные не хранятся и не обрабатываются на рабочих станциях в составе ЛВС;
   - возможность удаленного разграничения доступа к серверам (установленными на них механизмами управления доступом), обрабатывающим и хранящим защищаемые данные, для администратора безопасности. Эта возможность должна быть реализована для осуществления реакции на обнаружение удаления системы защиты на рабочей станции, которая при этом становится незащищенным инструментарием несанкционированного доступа по сети к защищаемым данным.

глава22

Метод контроля вскрытия аппаратуры

22.1. Общий подход к контролю вскрытия аппаратуры техническими средствами защиты

Выше был рассмотрен метод мониторинга и противодействия переводу системы защиты (реализуемой программно) в пассивное состояние, при котором объект становится незащищенным. Показано, что наиболее эффективная реализация данного механизма защиты достигается с применением аппаратной компоненты

При этом, чтобы удалить (модифицировать) программную компоненту (собственно, реализующую все механизмы разграничительной политики доступа пользователей к защищаемым ресурсам), предварительно необходимо удалить из защищаемого компьютера плату системы защиты. А это можно сделать, только сняв предварительно крышку корпуса с системного блока компьютера.

Противодействие таким действиям может быть реализовано с использованием организационных мер:

- опечатывание корпуса;
- реализация объектовой защиты — пропускной режим и т.д.

Конечно, применение организационных мероприятий существенно повысит защищенность объекта. Однако противодействовать появляющейся с применением аппаратной компоненты системы защиты угрозе несанкционированного вскрытия аппаратуры с целью удаления платы защиты можно и техническими средствами. В составе ЛВС данную функцию можно возложить на администратора безопасности.

Рассмотрим метод защиты от несанкционированного вскрытия аппаратуры, основанный на использовании в качестве дополнительного средства защиты системы контроля вскрытия аппаратуры (СКВА), предназначенной техническими средствами противодействовать несанкционированному вскрытию аппаратуры, в том числе, корпусов защищаемых компьютеров.

СКВА должна состоять из датчиков вскрытия аппаратуры, цепи сбора сигналов и специализированного рабочего места центрального контроля (сервера безопасности). В качестве датчиков могут использоваться контактные датчики, фиксирующие открывание корпусов системных блоков.

Известно, что при использовании витой пары для прокладки вычислительной сети (например, в стандарте 10 BASE-T) используются четыре из восьми линий связи для обмена информацией. Еще четыре провода остаются свободными и могут быть использованы для построения системы контроля вскрытия аппаратуры.

Использование незадействованных в передаче данных проводов также не создает помехи передачи данных, что не уменьшает производительности сети в целом.

Общий вид СКВА [14, 15] представлен на рис. 22.1.

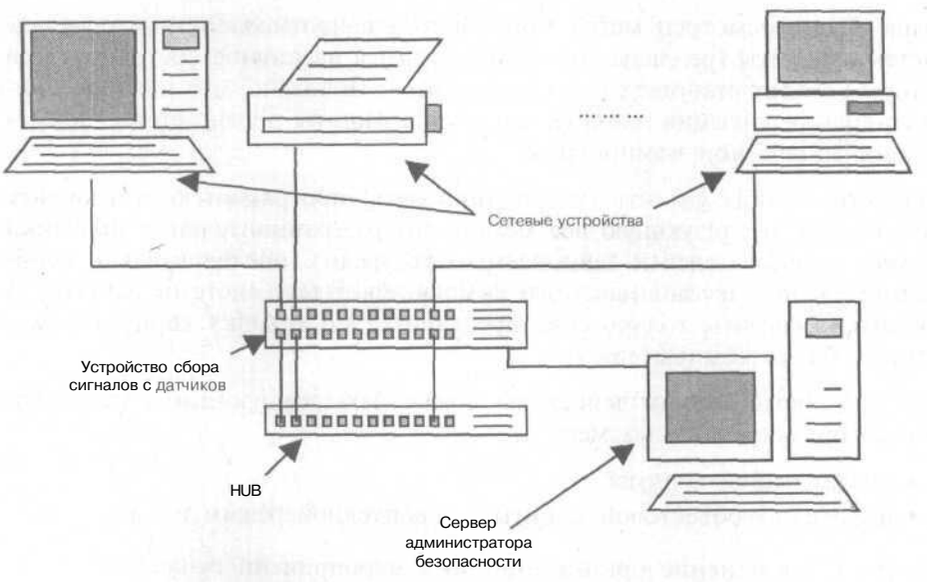


Рис. 22.1. *Система контроля вскрытия аппаратуры*

Идея рассматриваемого механизма защиты, проиллюстрированная рис. 22.1, состоит в следующем. К связному ресурсу (hub) подключается дополнительное устройство, подающее напряжение в контур, замыкаемый через размыкатель контура (выключатель), помещаемый в защищаемом компьютере. При этом используются свободные линии связи витой пары, подводимой к компьютеру. В случае снятии крышки компьютера (либо выдергивании витой пары — канала связи из розетки компьютера) размыкается контур и устройство отключает компьютер от ресурсов ЛВС, т.к. при этом компьютер становится опасным для общих ресурсов ЛВС.

Примечание

Может быть реализована дополнительная защита контура, если подавать в него не просто напряжение, а некую последовательность сигналов, однако при этом датчик должен быть активен и обладать способностью преобразовывать полученный сигнал каким-либо образом (возможны и иные подходы).

22.2. Реализация системы контроля вскрытия аппаратуры

Пример реализации схемы СКВА приведен на рис. 22.2. Пример схемы реализации сетевой компоненты защиты информации приведен на рис. 22.3. Пример схемы реализации системы защиты защищаемых компьютеров приведен на рис. 22.4.

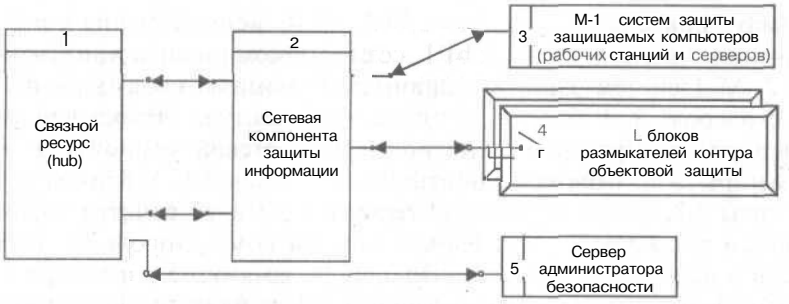


Рис. 22.2 Пример схемы реализации СКВА*

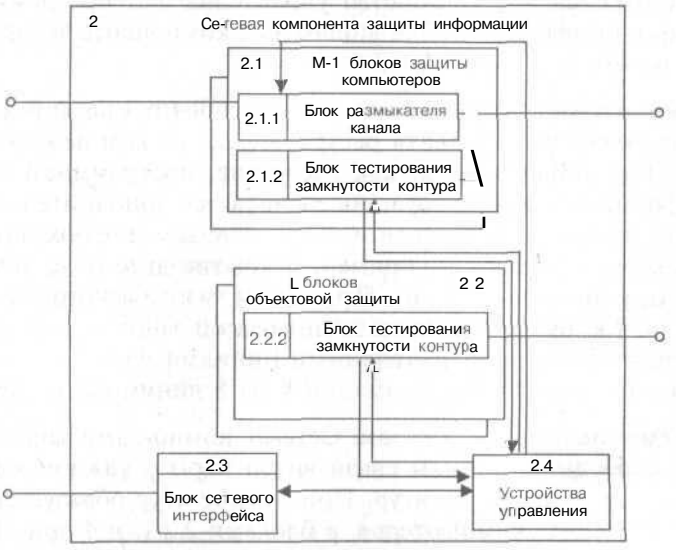


Рис. *22.3.* *Пример схемы реализации сетевой компоненты защиты информации*

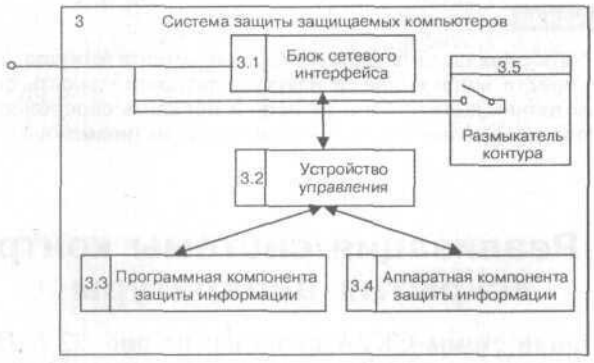


Рис. 22.4. *Пример схемы реализации системы защиты компьютеров*

На схемах, приведенных на рис. 22.2...22.4, использованы следующие обозначения: связной ресурс (hub) 1, сетевая компонента защиты информации 2, M-1 систем защиты защищаемых компьютеров (рабочих станций и серверов) 3, L блоков размыкателей контура объектовой защиты 4, сервер администратора безопасности 5, в сетевой компоненте защиты информации 2: M-1 блоков защиты компьютеров 2.1, L блоков объектовой защиты 2.2, блока сетевого интерфейса 2.3 и устройства управления 2.4, причем в каждом из M-1 блоков защиты компьютеров 2.1: блок размыкателя канала 2.1.1, блока тестирования замкнутости контура 2.1.2, в каждом из L блоков объектовой защиты 2.2 — блок тестирования замкнутости контура 2.2.1, в системе защиты защищаемых компьютеров 3: блок сетевого интерфейса 3.1, устройство управления 3.2, программная компонента защиты информации 3.3, аппаратная компонента защиты информации 3.4, размыкатель контура 3.5.

Система обеспечивает комплексную защиту рабочих станций и серверов. В частности, реализуется защита ресурсов ЛВС от компьютера, где возможно удаление аппаратной и, как следствие, программной компонент защиты информации. Кроме того, обеспечивается дополнительная объектовая защита помещений, сейфов и т.д. с выводом информации о нарушении объектовой защиты (например, открытие двери) на монитор администратору безопасности. При этом минимизируется объем оборудования, т.к. при реализации комплексной защиты компьютеров не требуется соединений дополнительными линиями связи -- используются свободные от передачи информации 4 из 8 линий связи витой пары.

Работает схема следующим образом. Сетевая компонента защиты информации 2 по свободным линиям связи витой пары в каждом компьютере и объекте защиты образует контур. При этом контур образуется блоками 2.1.2 и 3.5 при защите компьютеров, а блоками 2.2.1. и 4 при объектовой защите. Нарпяжение в контур, соответственно, подается блоками 2.1.2 и

2.2.1. Размыкатель 3.5 устанавливается в компьютере таким образом, что при снятии крышки компьютера он размыкается. Об этом через блок 2.1.2 уведомляется устройство управления 2.4, которое посредством блока 2.1.1 размыкает информационный канал, отключая опасный компьютер от сети.

Одновременно устройство управления 2.4 через блок 2.3 передает информацию администратору безопасности 5 об отключении компьютера от сети. Подключение компьютера заново возможно при замыкании контура блоком 3.5 (крышка корпуса установлена) по команде от администратора 5. Сигнал подключения информационного канала — возвращение в исходное состояние — формируется устройством управления 2.4).

При объектовой защите (двери, дверцы сейфов и т.д.) отсутствует информационный канал и размыкание (а также замыкание) контура блоком 4 через блок 2.2.1 фиксируется блоком 2.4, о чем уведомляется администратор безопасности 5.

Таким образом, с использованием рассмотренного механизма защиты обеспечивается возможность не только контроля вскрытия аппаратуры защищаемых объектов, но и автоматической реакции на факт данной угрозы - опасный объект (объект, на котором зарегистрирован факт вскрытия аппаратуры) автоматически отключается от сети, благодаря чему с него не может быть осуществлен удаленный доступ к сетевым ресурсам ЛВС.

Кроме того, реализация данного метода в системе защиты предотвращает возможность несанкционированного подключения к сетевому ресурсу (hub) незащищенного компьютера (например, подключение к каналу hub вместо санкционированного защищенного компьютера переносного компьютера с целью осуществления атаки на сетевые ресурсы).

22.3. Принципы комплексирования средств защиты информации

22.3.1. Комплексирование механизмов защиты информации от НСД

При использовании рассмотренных в данной главе методов защиты может быть построена комплексная система защиты, в которой обеспечивается реализация информационной безопасности защищаемых объектов на всех функциональных уровнях защиты. Это проиллюстрировано на рис. 22.5. Естественно, некоторые уровни могут не реализовываться — при этом взамен их должны выполняться соответствующей организационные мероприятия.

Каждая компонента рассчитана на защиту определенного набора возможных каналов НСД. Таким образом, программная, аппаратная компоненты

Рис. 22.5. *Принципы комплексирования механизмов защиты*

защиты и система контроля вскрытия аппаратуры в совокупности (при совместном использовании в системе защиты) представляют собой замкнутую многоуровневую систему защиты, схематично изображенную на рис 22.5.

В комплексной системе защиты информации реализуется иерархия мониторинга (контроля) корректности функционирования механизмов защиты, представленная на рис. 22.6.

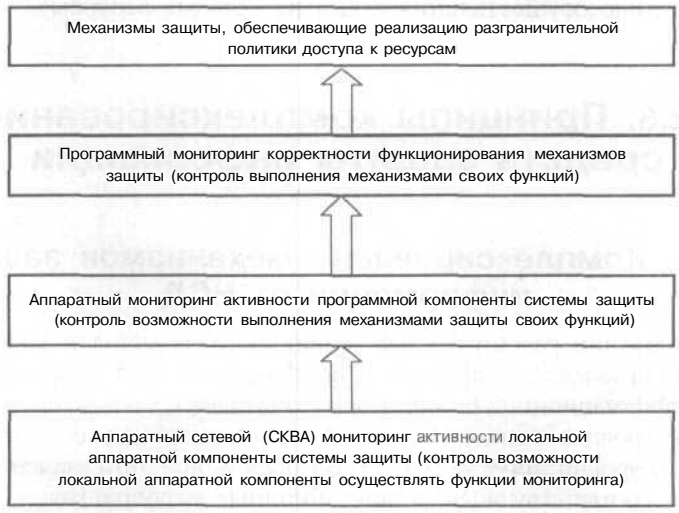


Рис. 22.6. *Иерархия мониторинга (контроля) корректности функционирования механизмов защиты в комплексной системе защиты*

22.3.2. Комплексирование в одной системе механизмов технической и объектовой защиты информации с единым сервером безопасности

При рассмотрении СКВА отмечалось, что контролируемыми объектами могут служить не только замкнутость корпусов защищаемых объектов, но и другие компоненты объектовой защиты — двери в помещение, сейфы и т.д. Таким образом, появляется возможность комплексирования в единой системе защиты (с единым выделенным сервером безопасности, соответственно с возможностью комплексирования регистрационной информации в едином интерфейсном модуле) средств технической и объектовой защиты. Причем принципиальным отличием описываемого подхода является то, что комплексирование осуществляется на технических средствах существующей ЛВС. При этом не требуется установка дополнительного коммуникационного оборудования. Как следствие, можно избежать принципиального увеличения кабельного хозяйства предприятия.

Общий анализ проблемы автоматизации систем управления доступом показывает, что для контроля над состоянием дверей может быть использована система контроля вскрытия аппаратуры. Для этого вместо того, чтобы устанавливать конечное оборудования с переключателем на системный блок, такое же оборудование можно установить на дверь. В этом случае открытие и закрытие двери будет регистрироваться сервером безопасности, что проиллюстрировано рис. 22.7.

Если в качестве конечного оборудования использовать не датчики (переключатели), имеющие только два положения «открыто/закрыто», а более интеллектуальное оборудование, например, электронные идентификаторы,

Рис. *22.7. Установка датчиков вскрытия дверей при построении автоматизированной системы контроля доступа*

то с помощью системы контроля вскрытия аппаратуры можно построить систему контроля доступа в помещение, как показано на рис. 22.8.

В рамках показанной на рис. 22.8, схемы всем сотрудникам компании, в которой установлена система контроля доступа, выдаются специальные электронные пропуска, например, «Touch Memory», представляющие собой электронный идентификатор в виде «таблетки», который содержит персональные коды доступа.

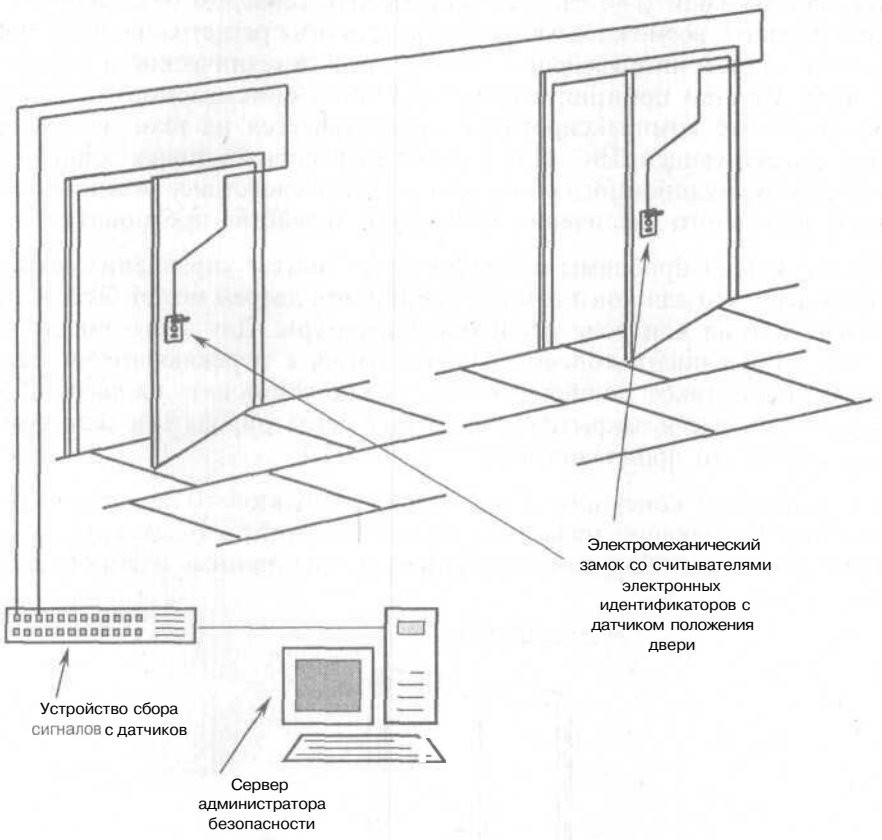


Рис. 22.8. *Применение СКВА для построения системы контроля доступа в помещения*

Считыватели, устанавливаемые у входа в контролируемое помещение, распознают код идентификаторов. Информация поступает в систему контроля доступа (сервер администратора безопасности) через устройство сбора сигналов с датчиков, такое же, как в системе контроля вскрытия аппаратуры.

Сервер администратора безопасности на основании анализа данных о владельце идентификатора, принимает решение о допуске или запрете прохода сотрудника на охраняемую территорию. В случае разрешения доступа, система приводит в действие исполнительные устройства, такие как электрозамки, шлагбаумы, электроприводы ворот и т.п. В противном случае двери блокируются, включается сигнализация и оповещается охрана.

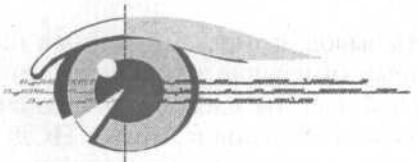
Дополнительно, с помощью СКВА [14, 15] можно реализовать такие технические решения, как:

* система охраны;
* оснащение проходной;
* система учета посетителей;
* система защиты от краж оборудования;
* система пожарной безопасности.

Таким образом, можем сделать вывод, что рассмотренный подход позволяет комплексировать различные (имеющие различное функциональное назначение) системы объектовой защиты с системой технической безопасности, реализующей защиту компьютеров и ЛВС от НСД. Причем все это в одной технической системе, на одном коммуникационном оборудовании, с единым сервером безопасности.

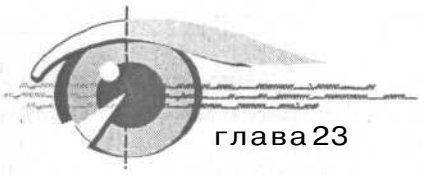
Дополнительные

вопросы защиты от НСД



Часть VII

- Антивирусная защита
- Межсетевое экранирование
- Вместо заключения. Политика информационной безопасности предприятия. Общий подход к выбору технического средства защиты компьютерной информации предприятия



глава 23

Антивирусная защита

Сразу оговоримся, что задача антивирусной защиты — это весьма самостоятельная задача в области защиты информации. Поэтому в данной книге мы не будем проводить сколько-нибудь серьезного исследования проблемы. Вместе с тем, кратко покажем, какие существенные возможности антивирусного противодействия предоставляют рассмотренные нами расширенные возможности механизмов управления доступом к ресурсам.

23.1. Общепринятый подход к антивирусной защите и его недостатки

Отметим, что доминирующим на сегодняшний день подходом к антивирусной защите является создание средств выявления вирусов по определенным признакам с последующим их удалением из системы. Такому подходу присущи два принципиальных недостатка:

- Чтобы выявить вирус в системе, необходимо однозначно определить его классификационные признаки, в частности определить сигнатуру. Другими словами, выявление вируса и, соответственно, осуществление антивирусного противодействие возможно только после его «опубликования», т.е. уже после осуществления вирусной атаки. С учетом необходимого времени на определение формализованных признаков вируса, а также на обновление вирусных баз, противодействие вирусу в общем случае осуществляется уже после нанесения ущерба вирусной атакой. При этом заметим, что эти базы, как правило, обновляются синхронно, т.е. их обновление по умолчанию задается временным интервалом.
- Так как интенсивность появления новых вирусов сегодня весьма высока, с большой скоростью возрастают и базы данных формальных признаков вирусов. Это, в свою очередь, естественно, приводит к возрастанию нагрузки антивирусных средств защиты на вычислительный ресурс защищаемого компьютера.

Таким образом, можем сделать вывод относительно того, что широко используемый на сегодняшний день подход к антивирусной защите трудно признать сколько-нибудь оправданным. При этом основными проблемами его применения является невозможность противодействия неизвестной вирусной атаке и значительное влияние системы защиты на загрузку вычислительного ресурса.

Примечания

Сколько-нибудь серьезного исследования влияния данных механизмов на производительность вычислительной системы автором не проводилось. Однако из практического опыта использования антивирусных средств можно заключить, что данное влияние заметно даже «на глаз», т.е. речь идет не об единицах процентов.

23.2. Использование расширенных возможностей механизмов управления доступом к ресурсам в решении задач антивирусного противодействия

Общие положения. Анализ современных ОС на предмет противостояния вирусным атакам

Чтобы рассмотреть возможность использования в системах антивирусной защиты рассмотренных выше методов защиты информации от НСД, прежде всего, попытаемся сформулировать основную цель вирусной атаки (не способ, их может быть множество). Естественно, что основной целью является удаленное -- без непосредственного контакта с защищаемым компьютером -- совершение какого-либо действия. Но действие, которое можно осуществить на компьютере, напрямую связано с запуском какой-либо программы (процесса). Таким образом, в основу антивирусного противодействия может быть положен механизм обеспечения замкнутости программной среды, призванный противодействовать несанкционированным действиям (запуску несанкционированных программ).

Как отмечалось, попытка несанкционированно осуществить действие (запустить программу) может быть как явной, так и скрытой.

Под явной мы понимаем внедрение на компьютер и запуск какой-либо программы, которая может быть идентифицирована. Таким образом, если добавочной защитой определен круг программ, которые разрешено запускать на компьютере, то вирусная атака, включающая все троянские и иные программы, будет полностью ликвидирована. При этом на компьютер данные программы смогут попадать, но не смогут быть там запущены. Заметим, что данный вывод делаем в предположении корректной реализации соответствующей модели управления доступом. В рамках этой

модели должны быть реализованы: защита доступа к системному диску, возможность задания разграничений для системных процессов и процессов, запускаемых с правами «root» и т.д., т.е. при управлении доступом к ресурсам должна быть реализована возможность разграничить права доступа для субъекта процесс.

Скрытые угрозы связаны с желанием разработчиков (прежде всего это относится к Microsoft) добиться максимальной универсальности и открытости технологий. Это приводит к реализации приложений с собственным встроенным в приложение интерпретатором команд, т.е. приложений, имеющих свои собственные средства программирования.

Усложнение задачи обеспечения замкнутости программной среды здесь состоит в том, что в качестве контролируемого процесса идентифицируется собственно приложение (интерпретатор команд). При этом скрыто все то, что непосредственно запрограммировано и выполняется при запуске интерпретатора. Самый распространенный пример — макросы в офисных приложениях.

Откажись разработчики приложений от возможности создания макросов, или обеспечь возможность идентификации макроса, по крайней мере, выдели интерпретатор команд как отдельный процесс, и проблема антивирусной защиты была бы практически полностью решена механизмом обеспечения замкнутости программной среды. На сегодняшний момент можно лишь локализовать воздействия вирусов. И то это можно сделать лишь средсвами добавочной защиты.

Кроме офисных приложений для ОС Windows, это относится также к виртуальным машинам JVM, также являющимся интерпретаторами команд, например, виртуальная машина JAVA. Не пользуйтесь виртуальными машинами и офисными приложениями Microsoft, и при реализации механизма обеспечения замкнутости программной среды вы практически забудете о вирусах. Не случайно, что статистика вирусных атак на ОС семейства UNIX на порядки меньше, чем на ОС семейства Windows.

Уместным будет здесь будет высказать удивление по поводу того, что, создавая защищенные приложения (системы электронного документооборота, защищенную «почтовую» связь и т.д.), отечественные разработчики иногда используют офисные приложения Microsoft, хотя известны аналогичные приложения, не позволяющие создавать макросы, т.е. решая одну проблему защиты информации, привносят тем самым другую проблему — проблему противодействия вирусным атакам.

Вирусные атаки

Вообще говоря, на взгляд автора, куда менее затратным было бы создать офисные приложения, не подверженные вирусным атакам, чем «всем миром» бороться с макросами.

Однако будем рассматривать проблему антивирусного противодействия в общем случае -- при наличии виртуальных машин и встроенных интерпретаторов команд офисных приложений. В общем случае можно выделить два класса атак, реализуемых на основе их свойств:

* это атаки, целью которых является совершение каких-либо действий (в общем случае нам неизвестных),
* атаки, реализуемые с целью распространения вируса, как внутри защищаемого компьютера, так и вне его — по сети.

Рассмотрим первый класс атак, которые более критичны с точки зрения НСД к информации. Успешно противодействовать данным атакам можно средствами разграничения прав доступа для субъекта «процесс». При этом процессу, содержащему встроенные возможности программирования, например, процессу winword.exe следует разрешить доступ только к каталогам (файлам) данных пользователя. Тогда любая скрытая атака сможет быть проведена только на каталоги и файлы с данными, т.е. непосредственно компьютер (с системной точки зрения, а не с точки зрения данных пользователей) с использованием рассмотренных подходов может быть надежно защищен. Таким образом производится локализация области действия вируса. Атака здесь представляет собою программу, выполненную в виде макроса и содержащуюся в документе. Эта программа запускается при чтении данного документа интерпретатором команд, в данном случае процессом winword.exe.

Защита же непосредственно данных пользователя напрямую связана с противодействием распространению вируса на защищаемом компьютере. В простейшем случае, здесь можно использовать рассмотренные механизмы управления доступом. Так, если установить права доступа к пользовательским данным одновременно пользователю и процессу, то данные одного пользователя не смогут «заразить» данные другого пользователя (т.е. локализация распространения вирусов осуществляется на уровне пользователей).

Динамическое управление доступом пользователей к ресурсам

Для усиления локализации возможности распространения вируса может применяться модель динамического управления доступом пользователей к ресурсам из приложения. Суть этой модели состоит в следующем.

Рассмотрим работу пользователя с приложением, предназначенным для обработки данных, например с текстовым редактором, и проиллюстрируем идею динамического управления доступом из приложения соответствующим алгоритмом.

1. В исходном состоянии пользователю должен быть разрешен доступ «на чтение» ко всем своим данным (нет основания задать какие-либо иные разграничения). Данные разграничения остаются неиз-

менными до момента обращения к файловым объектам из соответствующего приложения.

2. При чтении документа (файла), осуществляемого заданием пользователем из соответствующего приложения полнопутевого имени файла, разграничения доступа меняются. При этом пользователю разрешается записывать файл только по данному полнопутевому имени. Ко всем остальным файлам остается разрешен доступ только «на чтение».
3. Пользователь может сохранить данные по тому же адресу (доступ ему разрешен), либо задать из приложения иной адрес сохранения (данная возможность существует и без предварительного чтения файлового объекта). При задании этого адреса пользователем (санкционированная запись) опять же изменяются разграничения — пользователю разрешается запись по задаваемому ему из приложения адресу.
4. При закрытии документа разграничения восстанавливаются в исходное состояние.

Как видим, при использовании данного алгоритма обеспечивается только санкционированная запись данных, что предотвращает распространение вируса. Серьезные задачи здесь возлагаются на механизм аудита. Любая попытка обратиться по запрещенному адресу должна фиксироваться, о чем должен оповещаться пользователь. Такую попытку можно трактовать, как попытку распространения вируса, т.е. каждая попытка записи открытого документа под другим именем или в другое место может рассматриваться как подозрительная с точки зрения нахождения в документе вируса.

Аналогичный подход может использоваться и с целью локализации распространения вируса по сети. Например, это может применяться при использовании почтового приложения, содержащего встроенные средства программирования, в частности, программы Outlook (офисное приложение Microsoft). Отличие здесь состоит в том, что динамическому разграничению доступа подлежат сетевые адреса. В остальном алгоритм тот же.

Таким образом, средства динамического разграничения доступа предотвращают несанкционированное распространение вируса, решая задачу локализации, а средства аудита позволяют выявить «зараженный» документ. При этом отметим, что контролируется действие вируса, а не его потенциальное наличие в структуре документа. Таким образом, данный подход позволяет выявлять неизвестный вирус.

Однако естественно, что данный способ антивирусного противодействия не позволяет «лечить» документ, т.к. не анализирует его структуру. Поэтому данный подход может применяться как самостоятельно (в этом случае «зараженные» документы подлежат удалению), так и совместно со средствами анализа документа на наличие вируса, в качестве реакции

которых является «лечение» документа. При этом рассмотренный подход обеспечивает устранение принципиальных недостатков существующих методов антивирусной защиты. В частности он обеспечивает противодействие неизвестным вирусам и позволяет выявлять документы, требующие проверки, т.е. позволяет существенно снизить затраты вычислительных ресурсов защищаемого компьютера, связанные с антивирусным противодействием.

В качестве замечания отметим, что здесь рассмотрен лишь общий подход (на момент написания книги работа над созданием подобной системы только начата). При этом существуют как дополнительные проблемы (например, автосохранение документа и др.), так и дополнительные возможности, в частности:

- возможность контроля целостности исполняемого файла;
- возможность управления доступом к каталогам, не разделяемым системой и приложениями и др.

Однако корректная реализация данного подхода — вопрос технический, при этом уже из приведенного описания видны его возможности и существенные преимущества.

глава 24

Межсетевое экранирование

24.1. Межсетевой экран и его назначение

Основная исследуемая в книге задача — это эффективная защита компьютера (автономного, либо в составе ЛВС), предназначенного для обработки информации, требующей защиты от НСД. Вместе с тем, компьютер может решать и отдельные функциональные задачи, не связанные непосредственно с обработкой данных пользователями. В частности, компьютер с установленным на него специальным программным обеспечением может решать задачи межсетевого экранирования.

Межсетевой экран, как правило, служит для фильтрации исходящего и входящего трафика, циркулирующего между ЛВС и внешней сетью связи. Направление теории защиты информации, связанное с созданием средств межсетевого экранирования, наверное, наиболее полно освещено на сегодняшний день в литературе, поэтому мы не будем подробно останавливаться на этом вопросе. Для нас важным является вопрос, насколько актуальным является задача защиты компьютера, на который установлен межсетевой экран и в какой мере рассмотренные ранее подходы могут усилить межсетевую защиту.

Межсетевой экран может быть выполнен, как приложение, так и входить в состав ядра ОС. В частности, пакетные фильтры, обеспечивающие фильтрацию на сетевом и транспортном уровнях (в соответствии с моделью OSI), присутствуют в некоторых реализациях ОС UNIX.

24.2. Атаки на межсетевые экраны

Не останавливаясь подробно на исследовании атак на межсетевые экраны, в очень общем виде их цели (опять же не способы, которых может быть множество) можем классифицировать следующим образом:

» В первую очередь, это осуществление каких-либо действий для преодоления разграничений, реализуемых межсетевым экраном для возможности, прежде всего, получения несанкционированного доступа к ресурсам ЛВС извне.

• Во вторую очередь, это так называемые DoS атаки, призванные привести межсетевой экран к отказу для последующего обращения к другим ресурсам корпоративной сети от лица этого межсетевого экрана (в сети не может одновременно находиться два вычислительных устройства с одним адресом).

Разграничение доступа реализуется с использованием пакетных фильтров, а противодействия DoS атакам -- фильтрами прикладного уровня. Естественно, что как по назначению атаки, так и по возможности противодействия атакам рассматриваемыми в книге способами нас будут интересовать атаки, целью которых является НСД к информации ЛВС в обход разграничительной политики доступа к ресурсам, реализуемой пакетным фильтром межсетевого экрана.

Примечание

Вопрос фильтрации на прикладном уровне не рассматривается, так как он выходит за рамки рассматриваемых в данной книге задач.

24.3. Использование расширенных возможностей механизмов управления доступом к ресурсам в решении задач межсетевого экранирования

Итак, межсетевой экран содержит в своем составе средство управления удаленным доступом к ресурсам ЛВС -- пакетный фильтр. Настроенный пакетный фильтр разграничивает доступ к узлам ЛВС по IP-адресам и TCP(UDP)-портам. Пакетный фильтр, пропуская через себя весь исходящий и входящий трафик, фильтрует каждый запрос к ресурсу (ресурс - адрес и порт), пропуская только санкционированные (запросы, не соответствующие заданным разграничениям, им отклоняются). Таким образом, задача преодоления разграничительной политики доступа к ресурсам здесь полностью аналогична по постановке и по решению соответствующей задаче, рассмотренной выше для локальных ресурсов защищаемого компьютера.

Таже как и раньше можем выделить явные и скрытые угрозы. Явные угрозы связаны с корректностью построения пакетного фильтра (корректность реализуемой им модели управления доступом к ресурсам). Данная задача нами не рассматривается — это вопрос разработчиков межсе-

тевых экранов (приложений, фильтрующих запросы на доступ к ресурсам). Вторая задача — задача противодействия скрытым угрозам, напрямую связана с рассматриваемой нами задачей защиты компьютера, на котором установлено средство межсетевого экранирования.

Как отмечали ранее, любая попытка преодоления механизма управления доступом (в предположении корректности его реализации) связана с выполнением какого-либо действия. Для межсетевых экранов — это удаленное внедрение какой-либо программы с последующим ее запуском с системными правами (либо с правами «root»). Очевидно, что основная возможность противодействия данным атакам состоит в возможности разграничения прав доступа для процессов, а также в обеспечении замкнутости программной среды. При этом сетевым службам и порождаемым ими процессам должен запрещаться доступ к системному диску и файлам «на запись» с целью подмены системных программ на собственные, соответственно, к файлам настройки ОС (к реестру ОС) и т.д.

Кроме того, в системе следует предотвратить механизмом обеспечения замкнутости программной среды возможность запуска каких-либо внешних программ. К исполняемым же файлам санкционированных программ сетевым приложением следует запретить доступ «на запись» с целью противодействия их модификации. При этом гарантируется невозможность осуществления злоумышленником какого-либо действия (причем в общем случае — противодействие осуществляется скрытой угрозе) по преодолению разграничительной политики доступа к ресурсам, реализуемой пакетным фильтром межсетевого экрана.

Отметим, что более подробно в работе данный вопрос не рассматривается, т.к. это не является исследуемой областью защиты информации. Однако сказанного достаточно, чтобы сделать вывод о целесообразности использования рассмотренных подходов при защите компьютера, решающего задачи межсетевого экранирования, т.к. при их использовании возможность преодоления разграничительной политики доступа к ресурсам защищаемой ЛВС решается в принципе (при условии корректной реализации пакетного фильтра — модели управления доступом). При этом противодействие оказывается не конкретной атаке, а скрытой угрозе, т.е. цели атаки. Другими словами, задача защиты решается в общем случае.

глава 25
Вместо заключения

Политика информационной безопасности предприятия. Общий подход к выбору технического средства защиты компьютерной информации предприятия

Ранее мы подробно рассмотрели вопросы построения технических средств защиты информации от НСД, в частности — добавочных средств защиты. Общем случае были определены их цели и задачи.

Вместе с тем, применение технических средств может обеспечить эффективную защиту только в совокупности с организационными мероприятиями. Таким образом, комплексный подход к защите информации следует рассматривать как в узком (изложен в предыдущих главах), так и в широком (комплексирование организационных и технических мер защиты информации) смыслах. В данной главе рассмотрим вопросы комплексирования в широком смысле.

В рамках данного исследования необходимо ответить на следующие вопросы, возникающие при построении системы защиты корпоративной сети предприятия:

- С чего начать построение системы защиты?
- Как определить цели и задачи организационных и технических мер защиты информации?
- Как выбрать (либо спроектировать) средство добавочной защиты, наилучшим образом отвечающее потребностям предприятия?

Попробуем дать ответы на эти вопросы.

25.1. Понятие и содержание политики информационной безопасности предприятия

Под политикой информационной безопасности предприятия в общем случае будем понимать совокупность распорядительных документов, регламентирующих все аспекты обработки информации на предприятии.

Проиллюстрируем, какие же аспекты в общем случае должны найти свое отражение в политике информационной безопасности предприятия.

Перечень и классификация обрабатываемой на предприятии информации

Прежде всего, необходимо определиться с тем, какая информация обрабатывается на предприятии и как она может быть классифицирована (секретная, конфиденциальная, служебная тайна, ограниченного доступа и др.). Не определив, что защищать, невозможно в принципе решать вопрос о том, как защищать.

При классификации информации необходимо учитывать формализованные признаки ее отнесения к какой-либо категории, например, к секретной или конфиденциальной. При выявлении соответствующих признаков обрабатываемой информации автоматически задаются формализованные требования к ее защите. Например, для обработки секретных данных требования к защите информации в автоматизированной системе задаются классом защищенности не ниже 1В, а для обработки конфиденциальных данных — не ниже 1Г [2] (заметим, что в общем случае рассматриваем ОС, позволяющие использовать компьютер нескольким пользователям).

Концепция обеспечения информационной безопасности предприятия техническими мерами защиты

Концепция обеспечения информационной безопасности задает основные принципы обработки классифицированной информации на предприятии и ее технической защиты. В общем случае эта концепция должна включать:

1. Порядок обработки защищаемой информации на предприятии — определяется, какая информация обрабатывается вручную, какая с использованием средств вычислительной техники, какая информация предназначена для внутреннего использования, какая предполагает обмен вне предприятия и т.д.
2. Характеристику вычислительных средств предприятия, применяемых для обработки информации, требующей защиты. Сюда относятся архитектурные принципы построения корпоративной сети, используемые информационные технологии, ОС, приложения и т.д.

3. Характеристику предполагаемого использования вычислительных средств для обработки защищаемой информации -- где хранится, где и как обрабатывается, каким образом вводится и выводится, как передается по сети и т.д. Таким образом, эта характеристика позволяет выявить объекты корпоративной сети (компьютеры, информационные потоки и т.д.), требующие защиты. Результатом этого будут сформулированные требования к распределению ресурсов защищаемых объектов (компьютеров) между субъектами доступа – разграничительная политика доступа к ресурсам.
4. Характеристику распределения всей совокупности задач управления функционированием корпоративной сети (в том числе, защищаемыми объектами), каким-либо образом влияющих на безопасность защищаемой информации:
   - задачи защиты информации;
   - задачи системного и иного администрирования;
   - задачи инсталляции ПО;
   - задачи обработки информации и т.д.

   Здесь концептуально должно быть задано распределение задач между всеми субъектами доступа к защищаемой информации: сотрудники службы безопасности, администраторы (безопасности, системный, приложений, в частности СУБД, сети и т.д.), пользователи (возможно с предоставлением каких-либо привилегий отдельным группам пользователей), сотрудники службы эксплуатации, начальники подразделений и т.д.

   В рамках решения данной совокупности задач определяется потенциальный злоумышленник, то есть, от кого же следует защищать информацию. Данные субъекты должны исключаться из схемы управления функционированием корпоративной сети, прежде всего, из схемы управления информационной безопасностью.

   Результатом должна быть разработка концепции администрирования информационной безопасности корпоративной сети предприятия. При этом должны быть определены субъекты (сотрудники службы безопасности, администратор безопасности и т.д.) и объекты (рабочие станции, серверы, информационные технологии и т.д.) администрирования. Для объектов, где обрабатывается защищаемая информация, должны учитываться соответствующие формализованные требования.

   В качестве замечания отметим, что нами в предыдущих главах обосновывалась целесообразность централизованной схемы администрирования, которая состоит в следующем: сотрудники службы безопасности по представлению начальников подразделений формируют разграничительную политику доступа к ресурсам, администратор безопасности осуществляет непосредственное управление системой защиты (реали-

зует разграничительную политику доступа техническими средствами защиты информации и контролирует ее выполнение пользователями). Остальные субъекты исключаются из схемы администрирования. Им либо вообще не дается доступ к настройкам технических средств защиты в принципы, либо он возможен только при непосредственном контроле со стороны администратора безопасности (данные задачи должны решаться техническими средствами защиты).

5. Требования к технологии защиты информации, включающие:
   - требования к механизмам защиты в части реализации заданной разграничительной политики доступа к ресурсам;
   - требования к механизмам защиты в части противодействия НСД определенных потенциальных злоумышленников. Данные требования должны выдвигаться с учетом существующей статистики и потенциальных возможностей осуществления угроз, создаваемых определенными потенциальными злоумышленниками;
   - требования к механизмам, реализующим выбранную схему управления (администрирования) техническими средствами защиты информации.

   В качестве замечания отметим, что если защищаемая информация подпадает под формальные признаки, то при разработке требований к технологии защиты информации должны быть учтены соответствующие формализованные требования к механизмам защиты (в частности, для защиты от НСД, регламентируемые документами [1, 2].
6. Характеристику взаимодействия субъектов с целью формирования, задания и контроля разграничительной политики доступа к ресурсам. При этом определяется:
   - порядок назначения и изменения прав доступа субъекта к ресурсу;
   - порядок их задания в конфигурационных файлах средств технической защиты;
   - порядок контроля за выполнением субъектами заданных разграничений и порядок принятия решений при обнаружении фактов НСД;
   - порядок проведения регламентных работ, инсталляции ПО и т.д.
7. Инструкции всех субъектов доступа к конфиденциальной информации, где должны быть определены их права, обязанности, ответственность за нарушение разграничительной политики доступа. Данные инструкции должны быть доведены (за подписью) до каждого субъекта доступа к защищаемой информации, т.к. невозможно обеспечить защиту, не формализовав их возможные действия и ответственность за НСД к информации.

Концепция обеспечения информационной безопасности предприятия организационными мерами защиты

Поскольку книга посвящена вопросам технической защиты информации от НСД, не станем подробно останавливаться на рассмотрении данного вопроса. Это самостоятельный вопрос, требующий отдельного изложения. Здесь же отметим, что в рамках данного документа регламентируются все организационные мероприятия, реализуемые с целью защиты информации:

- порядок контроля доступа в помещения;
- порядок работы с внешними носителями (выдача пользователям, хранение, утилизация) и др.

Заметим, что организационные меры должны рассматриваться в совокупности с возможностями (требованиями) средств технической защиты и соответствующим образом их дополнять.

При этом, в общем случае политика информационной безопасности должна рассматривать задачу обеспечения информационной безопасности на предприятии куда шире, чем представлено выше. Здесь рассматриваются задачи политики только в части регламентирования вопросов защиты информации от НСД исследуемыми в работе средствами (ряд вопросов, в частности, защита от побочных электромагнитных излучений и т.д., нами в работе оставлен без внимания).

25.2. Выбор технического средства защиты информации от НСД

Роль политики информационной безопасности

В предыдущем разделе нами сформулированы задачи, решаемые в рамках разработки политики информационной безопасности. При этом политика информационной безопасности — это основополагающий документ, регламентирующий все мероприятия, реализуемые на предприятии с целью обеспечения компьютерной безопасности. Политика информационной безопасности — это та основа, без которой невозможно приступать ни к выбору, ни к проектированию средств защиты информации, в том числе добавочных.

Невозможно в общем случае сказать, какое средство защиты лучше (при прочих равных условиях — надежности функционирования, производительности и т.д.). Следует лишь говорить, какое средство в большей мере удовлетворяет требованиям, формализуемым политикой информационной безопасности.

Например, если политикой регламентируется схема централизованного администрирования механизмов защиты (в качестве потенциального злоумышленника рассматривается пользователь), то, как показано ранее, не подходят средства защиты, встроенные в современные универсальные ОС, а также ряд средств добавочной защиты, использующих встроенные в ОС механизмы (в частности, механизм управления доступом).

Следовательно, прежде всего, следует разработать политику информационной безопасности, а уже с ее использованием приступать к разработке, либо к выбору из представленных на рынке технических средств защиты информации.

Однако здесь существует противоречие, состоящее в следующем: с одной стороны, политика информационной безопасности выставляет требования к системе защиты, с другой стороны, может она быть реализована только на основе тех возможностей (механизмов защиты), которые реализуются системой защиты. То есть именно система защиты является центральным звеном, определяющим возможности защиты информации на предприятии в принципе.

Разработка политики информационной безопасности

Разработка политики информационной безопасности в общем случае является итерационной процедурой, состоящей из выполнения следующих шагов.

Первый шаг — разработка гипотетически идеальной для предприятия политики, куда закладываются те требования, которые идеально подходят для данного предприятия — формулируется некий идеальный профиль защиты.

Второй шаг — выбор (либо разработка) системы защиты, максимально обеспечивающей выполнение требований гипотетически идеальной политики информационной безопасности.

Третий шаг — определение требований политики информационной безопасности, которые не выполняются системой защиты.

Далее возможны следующие пути:

- осуществление доработки выбранного средства защиты в части выполнения ими сформулированных требований;
- выполнение (по возможности) данных требований организационными мерами. Здесь главное — не свести все к абсурду, например, если рассматривать для ОС Windows NT/2000/XP в качестве потенциального злоумышленника пользователя, то с учетом того, что пользователь может запустить любую программу с внешнего устройства ввода, т.е. получает инструментарий преодоления системы защиты, в качестве организационных мероприятий может быть предусмотрено использование компьютеров без внешних устройств ввода);
- пересмотр политики информационной безопасности (если это возможно) с учетом возможностей выбранного средства защиты информации. Сделано это должно быть таким образом, чтобы чтобы не выполняемые системой защиты требования не выдвигались в качестве основных требований к обеспечению информационной безопасности (однако заметим, что это не должно привести к ослаблению защиты информации на предприятии: мы говорим о пересмотре, не об усечении возможностей).

Проблемы реализации принятого профиля информационной безопасности

В результате выполнения рассмотренной в предыдущем пункте процедуры будет сформирована политика информационной безопасности, обеспечиваемая техническими возможностями выбранной системы защиты.

Возникает резонный вопрос: а может ли случиться так, что разработанный на предприятии профиль защиты принципиально не достижим, либо, наоборот, реализуется с большой избыточностью (это тоже плохо, т.к. в данном случае необоснованно снижается производительность, повышается цена системы защиты и т.д.)?

Гипотетически подобное возможно. Это обусловлено тем, что встроенные в современные универсальные ОС механизмы защиты еще весьма далеки от совершенства (их недостатки рассмотрены в предыдущих главах), а добавочные средства защиты на отечественном рынке еще не столь широко представлены, чтобы обеспечить всевозможные профили защиты, потребность в которых может возникнуть на практике.

Существующие системы защиты скорее имеют больше общего, чем различий. Пожалуй, значительно среди них выделяется КСЗИ «Панцирь» – здесь реализовано 9 патентов на изобретения сотрудников ЗАО «НПП «Информационные технологии в бизнесе» (на сегодняшний день по реализованным техническим решения подано 15 заявок на изобретения), Это объективно иллюстрирует существенные отличия данной системы.

Вместе с тем отметим, что если в полном объеме политику информационной безопасности обеспечить техническими средствами защиты не удается, должны быть выделены «слабые» места защиты и выработаны соответствующие организационные мероприятия по возможной локализации потенциально опасных для системы защиты угроз.

Общее правило выбора системы технической защиты информации

Обобщая все сказанное, отметим, что система технической защиты информации является ключевым звеном политики информационной безопасности, которая в своей основе базируется именно на механизмах технической защиты информации. Обоснованный выбор средства защиты для установки в корпоративной сети предприятия может быть осуществлен только на основе анализа на совпадения требуемого профиля защиты предприятия и профиля защиты, реализуемого системой зашиты, в частности добавочной.

Поэтому выбор системы защиты, равно как и разработка политики информационной безопасности, в общем случае является взаимосвязанной итерационной процедурой, состоящей из выполнения рассмотренных выше шагов.

Список литературы

1. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от НСД к информации. — Москва, 1992.
2. Гостехкомиссия России. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. — Москва, 1992.
3. Дейтел Г. Введение в операционные системы. Т.1. — М.: Мир, 1987.
4. Емелин И.В., Эльгиян Р.В. Обеспечение многоуровневой защиты в информационных и вычислительных системах. — М.: ВНИИМИ, 1979.
5. Коберниченко А.В. Недокументированные возможности Windows NT. — Москва, «Нолидж», 1998.
6. Кофман А., Крюон Р. Массовое обслуживание. Теория и приложения. — Москва, «Мир», 1965.
7. Люцарев В.С., Ермаков К.В., Рудный Е.Б., Ермаков И.В. Безопасность компьютерных сетей на основе Windows NT. — Москва, «Русская редакция». — 1998.
8. Мельников В. Защита информации в компьютерных системах — М.: Финансы и статистика; Электроинформ, 1997.
9. Оголюк А.А., Щеглов А.Ю. Технология и программный комплекс защиты рабочих станций и информационных серверов в Intranet-сетях//Информационные технологии. — №1. — 2000.
10. Оголюк А.А., Щеглов А.Ю. Технологии построения системы защиты сложных информационных систем//Экономика и производство. — №3. — 2001.
11. Павличенко И.П., Щеглов А.Ю. Обеспечение замкнутости программной среды путем разграничения прав доступа к файловой системе//Защита информации. Конфидент. — №4-5. — 2002.
12. Павличенко И.П., Щеглов А.Ю. Новые технологии защиты вычислительных систем. 1. Механизмы разграничения прав доступа к файловой системе и обеспечения замкнутости программной среды//Информационные технологии. — №12. - 2002.
13. Хофман Л.Дж. Современные методы защиты информации. — М.: Советское радио, 1980.
14. Хуотаринен А.В., Щеглов А.Ю. Технология физической защиты сетевых устройств//Экономика и производство. — №4. — 2002.
15. Хуотаринен А.В., Щеглов А.Ю. Комплексирование объектной и технической защиты вычислительной сети // Сборник научных статей «Современные технологии». Под. ред. С.А. Козлова и В.О. Никифорова, СПб, 2002.
16. Щеглов А.Ю.,Тарасюк М.В., Оголюк А.А. Технология защиты рабочих станций в сетевых архитектурах «клиент-сервер»//BYTE. Россия — №1 — 2000.
17. Щеглов А.Ю.Дарасюк М.В., Оголюк А.А. Технология оптимизации ядра открытых операционных систем по требованиям информационной безопасности//Информационные технологии. — №4. — 2000.

18. Щеглов А.Ю. Проблемы и принципы проектирования системы защиты информации от несанкционированного доступа. Часть 2. Системный подход к построению системы защиты//Экономика и производство. — №10-12. — 1999.
19. Щеглов А.Ю., Павличенко И.П. Технологии защиты рабочих станций и серверов корпоративной сети//ЭКСПРЕСС ЭЛЕКТРОНИКА. - №5/2002.
20. Щеглов А.Ю. Принципы обслуживания заявок в вычислительных системах с динамическими относительными приоритетами//Информационные технологии. – №8. – 1997.
21. Щеглов А.Ю. Исследование эффективности обслуживания заявок в ЛВС реального времени по приоритетным расписаниям//Информационные технологии. – №12. – 1998.
22. Щеглов А.Ю. Метод синтеза расписаний обслуживания заявок для распределенных вычислительных систем и ЛВС реального времени//Информационные технологии. — №10. — 1997.
23. Щеглов А.Ю. Основы теории надежности СЗИ. Назначение средств добавочной защиты // Защита информации. Кофидент. — №4. — 2003.
24. Щеглов А.Ю. Система защиты рабочих станций, информационных и функциональных серверов вычислительных систем и сетей с динамическими списками санкционированных событий. Патент №2166792.
25. Щеглов А.Ю. Система защиты рабочих станций, информационных и функциональных серверов вычислительных систем и сетей. Патент №2169941.
26. Щеглов А.Ю. Распределенная система защиты вычислительных систем и сетей. Патент №2169942.
27. Щеглов А.Ю. Система оптимизации структуры ядра открытых операционных систем по требованиям информационной безопасности. Патент №2174254.
28. Щеглов А.Ю. Система обеспечения целостности обработки информации вычислительных систем. Патент №2180135.
29. Щеглов А.Ю. Система защиты и контроля целостности резидентной программы. Патент №2185657.
30. Щеглов А.Ю. Система контроля доступа к запускаемым процессам (программам). Патент №2202122.
31. Щеглов А.Ю. Система контроля доступа к информационным ресурсам. Патент № 2207618.
32. Щеглов А.Ю. Система разграничения доступа к ресурсам. Патент № 2207619.
33. Щеглов А.Ю., Оголюк А.А., Павличенко И.П. и др. Комплексная система защиты информации «Панцирь» для ОС Windows 95/98/NT/2000 (КСЗИ «Панцирь»). Свидетельство об официальной регистрации программы для ЭВМ № 2002611971 от 22.11.2002. Правообладатель ЗАО «НПП «Информационные технологии в бизнесе».
34. www.rootshell.com
35. www.securityfocus.com
36. www.sysinternals.com