АНАЛИЗ И ПРОЕКТИРОВАНИЕ
ЗАЩИТЫ ИНФОРМАЦИОННЫХ СИСТЕМ

Контроль доступа
к компьютерным ресурсам

МЕТОДЫ, МОДЕЛИ, ТЕХНИЧЕСКИЕ РЕШЕНИЯ

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT
TEKNOLOGIALARI UNIVERSEITI
384 1/1
AXBOROT-RESURS MARKAZI

Профессиональная Литература
Санкт-Петербург
2017
СОДЕРЖАНИЕ

ВВЕДЕНИЕ.................................................................................. 11

ГЛАВА 1. ЗАДАЧИ И ОСНОВОПОЛАГАЮЩИЕ МЕТОДЫ
ЗАЩИТЫ ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА .......... 16

1.1. ОБЩАЯ КЛАССИФИКАЦИЯ ЗАДАЧ И МЕТОДОВ ЗАЩИТЫ
ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА .......... 16

1.2. ОБОБЩЕННАЯ СХЕМА КОНТРОЛЯ ДОСТУПА.......................... 19

1.3. КОНТРОЛЬ ДОСТУПА НА ОСНОВЕ МАТРИЦЫ ДОСТУПА ............ 21

1.4. КОНТРОЛЬ ДОСТУПА НА ОСНОВЕ МЕТОК БЕЗОПАСНОСТИ
(МАНДАТОВ) ........................................................................... 29

1.4.1. Защита от хищения (понижения категории) категорированной
информации ................................................................. 30

1.4.2. Защита от нарушения целостности информации .................. 43

1.5. ЗАДАЧИ ЗАЩИТЫ ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО
ДОСТУПА ............................................................................. 44

1.5.1. Формирование режимов обработки информации
субъектами доступа ....................................................... 44

1.5.2. Ролевая модель и метод контроля доступа ....................... 48

1.5.3. Сессионная модель и метод контроля доступа .................. 53

1.6. ЗАДАЧА НУЛЕВОЙ УРОВНЯ ТЕХНОЛОГИЧЕСКИХ УЯЗВИМОСТЕЙ
— ЗАДАЧА ЗАЩИТЫ ОТ АКТУАЛЬНЫХ УГРОЗ АТАК .................. 60

1.7. ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ ................................ 72

ГЛАВА 2. СУБЪЕКТЫ И ОБЪЕКТЫ ДОСТУПА .................... 75

2.1. СУБЪЕКТЫ ДОСТУПА ....................................................... 75
2.3.4. Контроль прямого доступа к дискам — к устройствам хранения данных ........................................ 144

2.4. ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ ........................................ 147

ГЛАВА 3. ПРИНЦИПЫ И МЕТОДЫ КОНТРОЛЯ ДОСУПА К СТАТИЧЕСКИМ ОБЪЕКТАМ ........................................ 150

3.1. КЛАССИФИКАЦИЯ ОБЪЕКТОВ ДОСУПА ........................................ 150

3.2. ПРИНЦИПЫ КОНТРОЛЯ ДОСУПА К СТАТИЧЕСКИМ ОБЪЕКТАМ ........................................ 151

3.3. ТРЕБОВАНИЯ К ПОСТРОЕНИЮ БЕЗОПАСНОЙ СИСТЕМЫ ........................................ 153

3.3.1. Требования к правилам контроля доступа ........................................ 154

3.3.2. Модель и метод контроля доступа, реализуемого перенаправлением запросов доступа ............... 163

3.3.2.1. Проблема и метод контроля доступа к неразделяемой системой и приложениям объектам .............. 163

3.3.2.2. Модель и метод контроля доступа перенаправлением запросов доступа ........................................ 169

3.3.2.3. Технология виртуализации системных средств ........................................ 174

3.3.2.4. Оценка эффективности контроля доступа с виртуализацией ........................................ 179

Модель исходной (базовой) системы ........................................ 179
Модели системы с виртуализацией ........................................ 182
Полная виртуализация системного средства ........................................ 182
Частичная виртуализация системного средства ........................................ 186

3.3.3. Требования к правилам контроля доступа в части защиты системных файловых объектов ............... 190

3.3.4. Общие требования к созданию потенциально безопасных систем и приложений ........................................ 192

3.4. РЕАЛИЗАЦИЯ КОНТРОЛЯ ДОСУПА. ТЕХНИЧЕСКИЕ РЕШЕНИЯ ........................................ 193

3.5. ОБЕСПЕЧЕНИЕ ЗАМКНУТОСТИ ПРОГРАММНОЙ СРЕДЫ ........................................ 198

3.5.1. Метод обеспечения замкнутости программной среды заданием списка исполняемых файлов, разрешенных на выполнение ........................................ 199

3.5.2. Метод обеспечения замкнутости программной среды заданием папок с исполняемыми файлами, разрешенными на выполнение ........................................ 200
3.5.3. Техническое решение ........................................ 202
3.5.4. Расширение функциональных возможностей метода обеспечения аутентичности программной среды .................. 205

3.6. ПРИМЕРЫ РЕШЕНИЯ НЕКОТОРЫХ АКТУАЛЬНЫХ ЗАДАЧ ЗАЩИТЫ ... 206
3.6.1. Защита от вредоносных программ .......................... 206
3.6.2. Защита от сетевых атак на уязвимости ОС и приложений.
Реализация процессной модели контроля доступа ............. 213

3.7. ИЛЛЮСТРАЦИЯ ОБЩИХ ПРИНЦИПОВ КОНТРОЛЯ ДОСТУПА К СТАТИЧЕСКИМ ОБЪЕКТАМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ФОРМИРОВАНИЯ РЕЖИМОВ ОБРАБОТКИ ИНФОРМАЦИИ СУБЪЕКТАМИ ДОСТУПА ........................................ 226
3.7.1. Контроль доступа к принтерам. Техническое решение .... 228
3.7.2. Контроль доступа к сетевым объектам. Техническое решение ..229

3.8. ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ ИЛИЛИЗАЦИИ КОНТРОЛЯ ДОСТУПА
К СТАТИЧЕСКИМ ОБЪЕКТАМ НА ПРИМЕРЕ КОНТРОЛЯ ДОСТУПА К СЕРВИСАМ ОЛИЦЕТВОРЕНИЯ ........................................ 238

3.9. КОНТРОЛЬ САНКЦИИРОВАНИЯ СВЕРШИВШИХСЯ
СОБЫТИЙ ......................................................... 242
3.9.1. Контроль запускаемых в системе процессов .............. 245
3.9.1.1. Контроль запускаемых в системе процессов .............. 245
3.9.1.2. Контроль активности обязательных процессов ......... 246
3.9.1.3. Реализация регламента работы с приложениями ....... 248
3.9.2. Информация о запускаемых в системе процессах ......... 250

3.10. ВЫВОДЫ ПО ТРЕТЬЕЙ ГЛАВЕ ........................................ 254

ГЛАВА 4. ПРИНЦИПЫ И МЕТОДЫ КОНТРОЛЯ ДОСТУПА
К СОЗДАВАЕМЫМ ОБЪЕКТАМ. ТЕХНОЛОГИЯ ЗАЩИТЫ ДАННЫХ ................................. 257

4.1. ПРИНЦИПЫ КОНТРОЛЯ ДОСТУПА К СОЗДАВАЕМЫМ ОБЪЕКТАМ ... 257
4.2. МЕТОДЫ И МОДЕЛИ КОНТРОЛЯ ДОСТУПА .......... 261
4.2.1. Контроль доступа на основе матрицы доступа .......... 261
4.2.2. Контроль доступа на основе меток безопасности (мандатов) .. 268

4.3. ТРЕБОВАНИЯ К ПОСТРОЕНИЮ БЕЗОПАСНОЙ СИСТЕМЫ ............. 271
4.3.1. Базовые требования ........................................ 271
4.3.1.1. Требования к методу диспетчерского контроля доступа ...... 271
4.3.1.2. Требования к методу мандатного контроля доступа ....... 273
4.3.2. Дополнительные требования, в части защиты от атак со стороны приложений, надежно выполняемых вредоносными
функциями при прочтении "зараженных" файлов .......... 277
4.3.2.1. Модель диспетчерского контроля доступа ............... 278
4.3.2.2. Модель мандатного контроля доступа .................. 283

4.4. ПРИМЕРЫ РЕШЕНИЯ НЕКОТОРЫХ АКТУАЛЬНЫХ ЗАДАЧ ЗАЩИТЫ ... 287

4.5. МЕСТО (ОБЛАСТЬ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ) МЕТОДОВ
КОНТРОЛЯ ДОСТУПА К СОЗДАВАЕМЫМ ОБЪЕКТАМ ........... 294

4.6. ТЕХНИЧЕСКИЕ РЕШЕНИЯ ........................................... 296
4.6.1. КОНТРОЛЬ ДОСТУПА К СОЗДАВАЕМЫМ ОБЪЕКТАМ ............ 296
4.6.2. Модель управления доступом к создаваемым объектам .... 304
4.6.2.1. Контроль доступа к создаваемым объектам ............... 304
4.6.2.2. Контроль доступа к создаваемым объектам ......... 305
4.6.2.3. Контроль доступа к создаваемым объектам ............... 309

4.7. ТЕХНОЛОГИЯ ЗАЩИТЫ ДАННЫХ .................................. 312

4.8. ИЛЛЮСТРАЦИЯ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ МЕТОДОВ
КОНТРОЛЯ ДОСТУПА К СОЗДАВАЕМЫМ ОБЪЕКТАМ .......................... 315
4.8.1. Защита от атак на java-машины, используемые для работы
с локальными приложениями ........................................ 316
4.8.2. Защита от атак на java-машины, используемые для работы
с сетевыми приложениями .......................................... 318

4.9. ПРИНЦИПЫ И МЕТОДЫ КОНТРОЛЯ ДОСТУПА С АВТОМАТИЧЕСКОЙ
РАЗМЕТКОЙ ОБЪЕКТОВ В ОБЩЕМ СЛУЧАЕ ............................. 319
4.9.1. Просвечивающие правила ....................................... 320
4.9.2. Модель контроля доступа ....................................... 322
4.9.3. Методы защиты ................................................ 325
4.9.3.1. Защита исполняемых файлов .................................. 325
4.9.3.2. Защита системных файлов .................................. 326
4.9.3.3. Защита статических файлов в общем случае .......... 327
4.9.3.4. Защита статических файлов в общем случае .......... 331
ВВЕДЕНИЕ

На сегодняшний день киберпреступность растет невиданными ранее темпами. Например, уже на 4-ом международном форуме по борьбе с киберпреступностью (ФИС 2010), засвидетельствовано, что доходы от киберпреступности в мире превысили доходы от незаконного оборота наркотиков. «Все это называется кибервойной, которой мы пытаемся противостоять» [63].

Особое место занимает кража информации. Например, эксперты компании Zecurion подсчитали, что уже за 2011 год в мире было зафиксировано 819 фактов кражи информации. Суммарный ущерб от них составил 20 млрд. долл. (по их оценке, российские убытки составили 1 млрд. долл. в год) [69]. Впрочем, в 2010 году ущерб от подобных действий составил 3,8 млрд. долл. [69], вот каков динамика! Сегодняшний же рост интенсивности киберугроз позволяет говорить об их промышленных масштабах [31]: «Интенсивность киберугроз из-за границы на британские правительственные организации и частные компании возросла до промышленных масштабов» [72].

Заметим, что в последние годы проблема вышла на новую веху – на межгосударственный. Вот небольшой отрывок из документа [70]: «Следствием уязвимости информационно-коммуникационных систем к радиоэлектронным и программно-аппаратным воздействиям в мире возникло и стало быстро распространяться информационное оружие, обладающее трансграничными поражающими факторами, резко возросла роль информационной войны. Российская Федерация, стремительно продвигающаяся по пути информатизации всех сфер жизнедеятельности общества, оказалась перед лицом новой серьезной угрозы, исходящей из глобального информационного пространства». Кстати говоря, в этом документе дано определение «Информационной войны»: информационная война — противоборство между двумя или более государствами в информационном пространстве с целью нанесения ущерба информационным системам, процессам и ресурсам, критически важным и другим структурам, подрыва политической, экономической и социальной систем, массированной психологической обработки населения для дестабилизации общества и государства, а также принуждения государства к принятию решений в интересах противоборствующей стороны.

На фоне всего этого, в печати постоянно «проскальзывают» высказывания специалистов, негативно характеризующих текущее состояние дел в области информационной безопасности: «о расширяющейся пропасти», «о технологическом тупике» и т.д. [66]. В частности, по мнению специалистов...
АНБ США (Agентство национальной безопасности США), современное положение в области безопасности компьютерных систем характеризуется как превалирующее [65].

Проанализируем, почему такое происходит. В основе происходящего находится объективная причина, состоящая в повсеместной компьютеризации в последние годы всех видов деятельности, в том числе, и финансовой. Именно это объясняет стремительный рост киберпреступности, причем, в первую очередь, той киберпреступности, которая направлена на получение финансовой выгода, что, в свою очередь, объясняет и рост уровня профессионализма злоумышленников, который естественно растет пропорционально росту их доходов, и появление совершенно новых угроз.

Кто, например, еще несколько лет назад связывал возможность хищения конфиденциальной информации с действиями санкционированного пользователя (сотрудника предприятия)? Сегодня же данную угрозу многие эксперты рассматривают в качестве доминирующей [73], что оценивается без попытки возможностью тем или иным образом получения финансовой выгоды в результате хищения конфиденциальной информации, но уже для сотрудника предприятия, обрабатывающего корпоративную информацию. А защита от вновь появляющихся угроз требует новых технических решений.

Итог — необходимость решения задачи защиты информации от внешних и внутренних угроз в комплексе. Как видим, угрозы не только модифицируются со временем, но и значительно расширяются в своем понимании, что, естественно, существенно усложняет и так непростую задачу реализации эффективной защиты информационных систем. Множество (если не подавляющая часть) атак сегодня осуществляется с использованием выявляемых уязвимостей (в первую очередь, ошибок программирования) в системных и программных средствах, которые могут устраиваться разработчиками этих средств месяцы. И от атак на такие уязвимости каким-то образом необходимо обеспечивать защиту.

Множество потенциальных угроз несут в себе штатные возможности современных приложений, позволяющие называть их вредоносными свойствами в результате работы с командными файлами. Перечислить потенциальные возможные, в том числе, появившиеся в последнее время, угрозы информационной безопасности можно долго.

Однако, существуют и субъективные причины. Исторически, еще в эпоху отсутствия каких-либо серьезных разговоров о киберпреступности и финансовых потерях, в основу построения систем защиты информации закладывались простейшие технологии, как правило, основанные на использовании методов контроля на соответствие каким-либо отнесенным к запрещенным с позиций безопасности событиям, например, сигнатурный анализ при реализации антивирусной защиты. В эпоху отсутствия киберпреступности никто особенно не задумывался об эффективности подобных решений, которая априори не может быть высокой.

Однако и сегодня, когда принципиально изменились требования к задачам и к эффективности защиты, принципиально изменились угрозы, именно эти технологии предполагают активно развиваться, например, методы контроля продолжают использоваться в антивирусных решениях, используемых в системах защиты, направленных на решение возникающих из практики эксплуатации современных информационных систем, крайне важных задач защиты — в системах защиты от утечек (хищения) конфиденциальной информации (DLP-решения [61,74]), в системах обнаружения (обнаружения и предотвращения) вторжений (IDS- системы [81]) и т.d.

Следует отметить, что реализуемые на основе этих технологий решения характеризуются большой маркетинговой привлекательностью, в частности, они позволяют наглядно показать потребителю возможный эффект от практического применения. Установить на свой компьютер любой антивирус и он сразу отыщет вирусы, возможно десятки, что убедит вас в целесообразности такой защиты. Но эффективность защиты в данном случае определяется тем, сколько новых вирусов он не обнаружит. А это уже вопрос специалистов, поэтому они и «бьют тревогу» [66].

Вместе с тем, все эти частные задачи защиты направлены на решение одной общей задачи — задачи защиты информации от несанкционированного доступа, который, следуя [1], может быть определен, как результат атаки (реализация угроз информационной безопасности), реализуемой с соответствующей целью — с целью раскрытия конфиденциальности, нарушения целостности или доступности (по отдельности, либо в любой их совокупности) информации. При этом в [1] под защитой информации от несанкционированного доступа понимается защита информации, направленная на предотвращение получения защищаемой информации заинтересованными субъектами с нарушением установленных нормативными и правовыми документами (актами) или обладателями информации прав или правил разграничения доступа к защищаемой информации с нарушением разграничительной политики доступа.

Как видим, основу реализации защиты информации от несанкционированного доступа, как следствие, решения в соответствующих частных задач за- щиты, следуя [1], должно составлять использование методов контроля и разграничения прав доступа, реализующих разграничительную политику доступа.

Таким образом, не смотря на то, что системы защиты информации от несанкционированного доступа, следуя их определению, априори предназначены...
для защиты от актуальных угроз атак, по средством реализации разграничительной политики доступа к защищаемой информации, совершенно не понятно как решать эти задачи. Ведь современные информационные системы подвержены множеству разнообразных угроз атак, которые имеют различную природу возникновения. В чем состоят единые подходы к защите от них, которые могли бы быть реализованы при построении систем защиты информации от несанкционированного доступа?

Если обратиться к существующим методам контроля и разграничения прав доступа, основанным на реализации абстрактных моделей контроля доступа, предложенных еще в [59,60], которые регламентируются к применению при построении систем защиты нормативным документом [2], и которые в конечном счете реализуются современными системами защиты информации от несанкционированного доступа, то можно сделать вывод, что они для решения рассматриваемых задач не предназначены. По крайней мере, нигде не описаны их возможности в части защиты от актуальных угроз атак. Не существует и возможности оценки их эффективности, поскольку требования к данным средствам в части защиты от актуальных угроз атак нигде не сформулированы.

Все это обусловливает актуальность разработки принципиально новых подходов к реализации защиты информации от несанкционированного доступа, включая разработку соответствующих методов контроля и разграничения прав доступа (и требований к их реализации), направленных на решение задачи защиты от современных актуальных угроз атак, по средством реализации разграничительной политики доступа. А главное, что позволит проводить соответствующие исследования и разрабатывать новые методы защиты, должны быть сформулированы задачи, которые должны решаться методами контроля и разграничения прав доступа.

В монографии системно излагаются ответы на все рассмотренные выше вопросы, начиная с определения задач, которые должны решаться современными системами защиты информации от несанкционированного доступа с целью реализации эффективной защиты от современных актуальных угроз атак, исследования недостатков существующих, широко используемых в современных системах защиты информации методов контроля и разграничения прав доступа, реализующих разграничительную политику доступа, заканчивая изложением и обоснованием предлагаемых методов защиты и реализованных технических решений, формулированием требований к их реализации, выполнение которых направлено на построение безопасных систем. Результаты проведенных исследований иллюстрируют необходимость использования принципиально новых подходов к построению систем защиты информации от несанкционированного доступа в современных условиях.

Отличает данный материал то, что в нем приводится не некие абстрактные рассуждения авторов на тему: «Как можно сделать». Все описанные в книге методы и технические решения реализованы и апробированы в рамках разработки программного средства защиты информации «Комплексная система защиты информации "Панцирь +" для ОС Microsoft Windows» [23] (разработчик и производитель Научно-производственное предприятие «Информационные технологии в бизнесе», www.mpp-itb.spb.ru), большая часть из которых авторами запатентована. Для иллюстрации материала в книге используются примеры механизмов данной системы защиты информации от несанкционированного доступа, для иллюстрации решения некоторых наиболее актуальных современных задач защиты информации — защиты информации от наиболее актуальных современных угроз атак, приводятся примеры реализации разграничительных политик доступа. Все это позволяет читателю уже по ходу прочтения рассматривать излагаемый в книге теоретический материал с точки зрения практической реализации.

"Панцирь +" — это новейшая разработка компании, пришедшая на смену хорошо известной и зарекомендовавшей себя КСЗИ "Панцирь-К". Практическое использование КСЗИ "Панцирь-К" позволяет прогнозировать и область применения новой разработки — это крупные информационные системы различного функционального назначения предприятий различных форм собственности. Почему крупные — все просто, в них есть, что защищать, как следствие, они подвержены угрозам целевых атак, соответственно нуждаются в эффективной защите. Эффективная защита, что, в том числе, мы увидим в данной книге, не может быть простой, как в реализации, так и в администрировании — требует профессиональных навыков при эксплуатации. Естественно, что предприятие, в информационной системе которого обрабатывается критичная к хищению, несанкционированному удалению, либо модификации информация, должно обеспечить профессиональную эксплуатацию сложного, но эффективного средства защиты. Все это и обусловливает область практического использования рассматриваемых в книге технических решений.

Отметим, что подавляющая часть разработанных и представленных в работе методов защиты информации, в том или ином виде, может применяться для различных типов ОС. Излагаю же мы их применительно к реализации защиты информации в среде ОС Microsoft Windows исключительно, исходя из того, что для этой платформы разработаны и апробированы технические решения, реализующие рассматриваемые методы защиты. Что подтверждает практическую значимость рассматриваемых подходов к защите информации от несанкционированного доступа.
ГЛАВА 1. ЗАДАЧИ И ОСНОВОПОЛАГАЮЩИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

1.1. Общая классификация задач и методов защиты информации от несанкционированного доступа

Прежде всего, приведем некоторые важные понятия, которые далее будем использовать [1].

Под безопасностью информации понимается состояние защищенности информации (данных), при котором обеспечена ее (их) конфиденциальность, доступность и целостность.

Под угрозой информационной безопасности понимается совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения состояния защищенности информации.

Под защитой информации понимается деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

Под несанкционированным доступом к информации понимается доступ к информации или к ресурсам информационной системы, осуществляемый с нарушением установленных прав и (или) правил доступа, образующих разграниченную политику доступа. Несанкционированный доступ – это результат атаки (реализация угрозы информационной безопасности), реализуемой с соответствующей целью – с целью раскрытия ее конфиденциальности, нарушения целостности или доступности (по отдельности, либо в любой их совокупности).

Под защитой информации от несанкционированного доступа понимается защита информации, направленная на предотвращение получения защищаемой информации заинтересованными субъектами с нарушением установленных нормативных и правовых документов (актами) или обязательствами информации прав или правил разграничения доступа к защищаемой информации с нарушением разграничительной политики доступа.

При создании систем защиты информации используются подход, основанный на представлении обработки информации в виде абстрактной вычислительной среды, в которой работают множество «субъектов» (пользователей и процессов) со множеством «объектов» (ресурсов и наборов данных). При этом построение системы защиты заключается в создании защитной среды в виде некоторого множества ограничений и процедур, способных под управлением ядра безопасности запретить несанкционированный и вместе с тем обеспечить санкционированный доступ субъектов к объектам, реализовав тем самым защиту последних от множества преднамеренных и случайных внешних и внутренних угроз.

Предполагают реализацию данного подхода и отечественные нормативные документы в области защиты информации [2, 4]. В указанных документах (с некоторыми различиями) приводится совокупность требований к набору и содержанию защитных функций, выполнение которых должно обеспечить защищенность информационных систем с учетом критичности обрабатываемой в них информации (области применения СЭИ НСД).

В общем случае следует выделять две основные группы методов защиты информации, которыми решаются принципиально различные даже собственно в своей постановке задачи защиты – это методы защиты информации от несанкционированного доступа (средства защиты обозначают СЭИ НСД) и методы криптографической защиты информации (средства защиты обозначают СЭИ КЗИ).

Как отмечали, к основным задачам защиты информации от несанкционированного доступа (НСД) принято относить:

- защиту от нарушения конфиденциальности обрабатываемой информации;
- защиту от несанкционированной модификации обрабатываемой информации;
- обеспечение доступности обрабатываемой информации.

Отметим, что в общем случае обеспечение доступности обрабатываемой информации обеспечивается не только защитой от ее несанкционированного удаления, но и несанкционированным выводом из эксплуатации вы
Контроль доступа к компьютерным ресурсам

• Контроль (разграничение) доступа к ресурсам системы — проверка (авторизация) и обеспечение разрешенного (санкционированного) доступа субъектов к объектам, отказ в несанкционированном доступе. Авторизация — это проверка прав доступа субъектов к объектам при попытке выполнения тех или иных операций.

• Контроль целостности ресурсов системы и обрабатываемой информации необходим для своевременного обнаружения несанкционированной модификации ресурсов системы и информации, что позволяет обеспечить правильность функционирования системы защиты и целостность обрабатываемой информации.

• Регистрация и анализ (аудит) событий, происходящих в системе — получение и анализ информации о состоянии ресурсов системы с помощью специальных средств контроля, а также регистрация потенциально опасных для безопасности системы действий. Анализ собранной информации позволяет выявить средства и определить, как далеко зашло нарушение, подсказать метод его расследования и способы исправления ситуации.

Эти универсальные методы защиты в том или ином виде могут применяться в различных вариациях и совокупностях в конкретных СЗИ НСД. Наибольший эффект достигается при использовании всех этих методов в комплексе. Однако, говоря об эффекте — об эффективности системы защиты, в первую очередь, требуется определиться с тем, какие задачи защиты ею должны решаться, и с тем, что представляет собой корректное решение той или иной задачи защиты информации от несанкционированного доступа.

1.2. Обобщенная схема контроля доступа

Контроль (разграничение) доступа осуществляется централизованным элементом системы защиты — диспетчером доступа, идентифицирующим субъекты, объекты и параметры запрашиваемого доступа субъектом к объекту, при запросе доступа к ресурсу, предоставляя субъекту запрашиваемый (санкционированный) доступ, либо отказывая субъекту в доступе к объекту (при несанкционированном доступе).

Диспетчер доступа перехватывает все запросы субъектов к объектам, поэтому должен реализовываться на уровне ядра ОС.
Идентифицирующую информацию субъектов и объектов доступа принято называть учетной информацией (или учетными данными); правила, на основании которых диспетчер доступа принимает решение о предоставлении (либо отказе) субъекту доступа к объекту, называют правилами разграничения доступа (или правилами доступа), которые в совокупности для множества субъектов и объектов информационной системы образуют разграничительную политику доступа.

Обобщенная схема контроля доступа, представленная на рис. 1.1.

Методы контроля доступа различаются способами идентификации субъекта и объекта доступа, способами задания и хранения правил разграничения доступа.

Рассмотрим и проанализируем на абстрактных моделях основополагающие (базовые), широко сегодня используемые на практике, методы контроля доступа, определив какие задачи защиты от несанкционированного доступа и решаются. При этом определим с тем, в чем состоят особенности решения задач контроля доступа в современных условиях, и какие это накладывает требования к корректности реализации соответствующих методов контроля доступа.

Замечание. Не смотря на дату издания (1992 год), требования к корректности реализации методов контроля доступа, сформулированные и соответствующий действующему сегодня нормативному документу [2], актуальны (с определенными оговорками) и по сей день. Поэтому далее, формулируя требования к корректности реализации соответствующих методов контроля доступа, будем соотносить их с соответствующими требованиями, регламентируемыми в [2].

**1.3. Контроль доступа на основе матрицы доступа**

Правила разграничения доступа в общем случае формируются матрицей доступа, в которой указывается каким субъектам (группам субъектов) к каким объектам (группам объектов) какие права доступа (чтение, запись, исполнение и т.д.) разрешены, либо, соответственно, запрещены.

Метод контроля доступа на основе матрицы доступа принято называть дискретным.

Замечание. Далее нам, по ряду причин, потребуется уточнить, что понимается в работе под дискретным контролем доступа [11].

Для оценки безопасности системы, основанной на реализации дискретного контроля доступа, на практике используется модель "Харрисона-Руза-Ульмана" [66], состояющая в следующем. Если считать, что множество $C = \{C_1, ..., C_i\}$ и $O = \{O_1, ..., O_k\}$ соответственно линейно упорядоченные множества субъектов и объектов доступа, а $R = \{w, r, x, d\}$ конечное множество прав доступа (чтение, запись, удаление, исполнение), то разграничительная политика доступа субъектов к объектам описывается матрицей доступа $M$, где $M[C, O] - ячейка матрицы, содержит набор прав доступа субъекта из множества $C = \{C_1, ..., C_i\}$ к объекту из множества $O = \{O_1, ..., O_k\}$. В любой момент времени система описывается своим текущим состоянием $Q = (C, O, M)$.

$$
M = \begin{bmatrix}
O_1 & O_2 & \cdots & O_k \\
C_1 & r & w & d & \cdots & 0 \\
C_2 & r & r & w & d & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
C_{i-1} & 0 & 0 & \cdots & r \\
C_i & 0 & w & \cdots & r & w & d \\
\end{bmatrix}
$$
Требование к безопасности системы в рассматриваемом случае может быть сформулировано следующим образом: «Для заданной системы состояние \( Q_0 = (C_0, O_0, M_0) \) следует считать безопасным относительно некоторого права \( R \) если не существует применимой к \( Q_0 \) последовательности действий, в результате выполнения которых субъектом \( C_0 \) приобретается право \( R \) доступа к объекту \( O_0 \), исходно отсутствующее в ячейке матрицы \( M_0[C_0, O_0] \). Если же право \( R \), отсутствующее в ячейке матрицы \( M_0[C_0, O_0] \), приобретается субъектом \( C_0 \) то следует говорить, что произошла утечка права \( R \), а система небезопасна относительно права \( R \).

**Замечание.** Определенный интерес представляет трансформация матрицы \( M_i \) в результате которого получаем матрицу \( M_{i+1} \):

\[
M_i = \begin{bmatrix}
C_1 & C_2 & \cdots & C_{i-1} & C_i \\
O_1 & r & w & d & r & \cdots & 0 & 0 \\
O_2 & \vdots & w & r & w & d & \cdots & 0 & w \\
O_i & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & r & w & d \\
O_k & 0 & 0 & \cdots & r & w & d
\end{bmatrix}
\]

В трансформированной матрице \( M_{i+1} \) отображаются права доступа к объектам субъектов (в исходной – права доступа субъектов к объектам).

**Замечание.** Поскольку последовательность команд (действий) генерирует субъект доступа, пользователь (действия администратора не имеет смысла рассматривать при анализе изменений состояния системы, т.к. администратор генерирует и корректирует в процессе работы системы матрицу доступа), то о безопасности системы можно говорить исключительно в предположении, что действия пользователя не приведут к утечке права \( R \).

Прежде всего, оценим уровень абстракции данной модели. Очевидно, что она универсальна в том смысле, что может быть применена для любых типов объектов доступа (ресурсов), будь то файловые объекты, объекты реестра ОС, принтеры, сетевые объекты и т.д. Отличия в модели для разнородных объектов будут состоять в используемом конечном множестве прав доступа \( R \).

Принципиальным для метода дискретного контроля доступа является реализуемый в нем способ управления потоками информации (информационными потоками). Потоки информации различаются действиями, выполняемыми субъектом над объектом. Рассмотрим на примере файловых объектов.

**Определение.** Под информационным потоком записи (w) понимается перенос информации от C к O, под информационным потоком чтения (r) — перенос информации от О к C, удаления (d) — понимается удаление субъектом C объекта O, под информационным потоком исполнения (x) понимается выполнение субъектом С исполняемого объекта O.

**Определение.** Под управлением информационными потоками понимается предоставление субъекту права генерирования потока из множества \( R = \{w, r, x, d\} \).

При управлении информационными потоками для субъекта C может быть, как произвольным, так и принудительным [11].

**Определение.** Под принудительным управлением информационными потоками для субъекта C понимается управление без предоставления субъекту C привилегированным пользователем – администратором, права предоставлять право генерирования потока из множества \( R = \{w, r, x, d\} \) другим субъектам.

Таким образом, при произвольном управлении информационными потоками субъект C включается в схему администрирования, на практике, как правило, в отношении созданных им объектов.

**Определение.** Под принудительным управлением информационными потоками для субъекта C понимается управление без предоставления субъекту C привилегированным пользователем – администратором, права предоставлять право генерирования потока из множества \( R = \{w, r, x, d\} \) другим субъектам.

При принудительном управлении информационными потоками обеспечивается полное исключение субъекта C из схемы администрирования (все задачи по созданию и модификации разграничительной политики доступа, в том числе, и в процессе функционирования системы) решаются исключительно привилегированным пользователем – администратором.

**Замечание.** Произвольное управление информационными потоками сегодня широко применяется на практике, в частности, состоит оно в реализации сущности «Владения», включенной в схему контроля доступа в современных универсальных ОС. При этом именно пользователь, создавший объект (как его «Владелец»), наделяет правом доступа \( R \) к этому объекту других пользователей.

О безопасности системы для метода дискретного контроля доступа с произвольным управлением информационными потоками, поскольку...
субъект доступа включен в схему администрирования, можно говорить исключительно в предположении, что субъект доступа (пользователь) не несет в себе угрозы генерирования утечки права $R$, с целью хищения, несанкционированной модификации или удаления обрабатываемой им на предприятии информации.

Однако, в проведенном исследовании [73], проиллюстрированном на рис. 1.2 а), сделан вывод о том, что наибольшую опасность для компании сегодня представляют именно собственные сотрудники (иссайдеры).

a) Наиболее актуальные ИБ угрозы

- 23% Потери важной конф. инф., персональных данных
- 1% Другие
- 19% Вирусы, черви
- 14% Атаки из Интернет
- 9% Шпионские программы
- 9% Спам
- 25% ИСД собственных информаций
- 1% Конф. информация

b) Психологический портрет инсайдера

Цели и мотивы инсайдеров (данные в %)

- Повышение финансового выигрыша
- Следует политической линии или действий руководства
- Несовпадение в интересах и заинтересованностях
- Политика скрытное, секретное, или информационные системы
- Время конфиденциальной информации
- Чем больше информации, тем больше связанная информация

На взгляд авторов в современных условиях уже не актуально говорить о профессиональных качествах потенциального злоумышленника, все сводится к одному параметру – стоимости защищаемой информации. Профессиональные же качества злоумышленника с учетом сказанного следует рассматривать, как функцию от стоимости защищаемой информации.

Вывод. Первое важнейшее и принципиальное изменение требований к построению защиты в современных условиях состоит в том, что защита должна строиться в предположении, что пользователь, обрабатывает информацию на защищаемом компьютере, должен рассматриваться в качестве вероятного потенциального злоумышленника, что принципиально меняет подход к реализации защиты информации, в частности, контроля доступа к защищаемым ресурсам.
Замечание. Вообще говоря, само понятие «Владелец» (не в технологическом смысле) не корректно для корпоративных приложений. Ведь пользователи на предприятии обрабатывают не собственную информацию, а информацию, принадлежащую предприятию, которое является собственностью совсем иных лиц, о каком же тогда владении может идти речь? Отсюда и потенциальный интерес к ее хищению пользователем.

Вывод. Метод дискретационного контроля доступа, основанный на использовании сущности «Владения» — позволяет построить безопасную систему в современных условиях, а построенная подобным образом система не безопасна относительно прав $R = \{w, r, x, d\}$.

Следствие. В безопасной системе контроля доступа должно осуществляться принудительное для субъекта управление потоками информации (или информационными потоками), сущность же «Владения» должна быть исключена из схемы контроля доступа как таковая, пользователь должен быть исключен из схемы администрирования.

Задача обеспечения принудительного управления потоками информации возлагается на диспетчера доступа, разграничивающего права, предоставлять право генерирования потока из множества $R = \{w, r, x, d\}$ субъектам доступа, предоставляя данные право исключительно привилегированным пользователям — администраторам.

Приведем требования к дискретационному контролю доступа (к корректности реализации), регламентируемые отечественным нормативным документом в области информационной безопасности [1]:

- Комплекс средств защиты (КСЗ) должен контролировать доступ наименованием субъектов (пользователей) к наименованным объектам (файлов, 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).
- КСЗ должен содержать механизм, претворяющий в жизнь дискретационные правила разграничения доступа.
- Контроль доступа должен быть применен к каждому объекту и каждому субъекту (индивида или группе равнopravnых индивидов).

- Механизм, реализующий дискретационный принцип контроля доступа, должен предусматривать возможности санкционированного изменения правил разграничения доступа (ПРД), в том числе возможность санкционированного изменения списка пользователей СВТ и списка защитных объектов.
- Право изменять ПРД должно предоставляться выделенным субъектам (администраторам, службе безопасности и т.д.).
- Должны быть предусмотрены средства управления, ограничивающие распространения прав на доступ.

Обратим внимание на следующее интересующее нас здесь требование: «Право изменять ПРД должно предоставляться выделенным субъектам (администраторам, службе безопасности и т.д.)». Как видим, в соответствии с данными требованиями, должно реализовываться именно принудительное для субъекта С управление потоками информации (или информационными потоками), сущность «Владения» должна быть исключена из схемы контроля доступа как таковая.

Остановимся на рассмотрении принципиального противоречия в использовании сегодня термина «Дискретционный контроль доступа». Изначально под дискретционным контролем доступа понимался контроль, основанный на предоставлении пользователю права назначать правила доступа к создаваемым им объектам, т.е. с произвольным управлением в системе информационными потоками. К слову сказать, именно это составляло принципиальное отличие метода дискретционного контроля доступа от метода мандатного контроля доступа, приори предполагающего принудительное управление информационными потоками, который рассмотрим далее.

Однако, как видим, со временем термин «Дискретционный контроль доступа» остался, но принципиально изменились требования к его реализации. Корректно в данном случае определить подобный метод контроля доступа, как метод избирательного контроля доступа с принудительным управлением информационными потоками, такое определение дано нами в [11]. Избирательного, поскольку отсутствует какая-либо формализация правил разграничения доступа — в отношении любого объекта администратор избирательно (произвольным образом) может назначить необходимые права доступа к нему субъектов.

Вместе с тем, исходя из того, что термин «Дискретционный контроль доступа» является достаточно устоявшимся, по крайней мере, в части использования матрицы доступа для задания разграничительной политики, будем далее использовать именно его, понимая при этом, что речь идет об избирав-
тельном контроле доступа с принудительным управлением информационными потоками.

Крайне важным является требование [2]: «Контроль доступа должен быть применен к каждому объекту и каждому субъекту (индивида или группе равноправных индивидов)». Данное требование в современных условиях, с учетом множества известных угроз атак, как на системные объекты, так и с получением системной учетной записи [51], имеет смысл понимать буквально, где под «каждым» нужно понимать, в том числе, и системные объекты и субъекты доступа, в отношении них также должна реализовываться разграничительная политика доступа.

В двух словах, собственно, о разграничительной политике доступа. Она может быть либо разрешительной, либо запретительной.

Определение. Под разрешительной разграничительной политикой доступа понимается политика, предполагающая реализацию следующего правила назначения прав доступа субъектам к объектам: «Все, что не разрешено (явно не указано), то запрещено».

Определение. Под запретительной разграничительной политикой доступа понимается политика, предполагающая реализацию следующего правила назначения прав доступа субъектам к объектам: «Все, что не разрешено (явно не указано), то запрещено».

Очевидно, что безопасная система должна строиться с использованием разрешительной разграничительной политики доступа [11], что, кстати говоря, диктуется и нормативным документом [2]: «...должно быть задано явное и не двусмысленное перечисление допустимых типов доступа (читать, писать и т.d.)».

В порядке замечания отметим, что данным нормативным документом формулируются требования к защите от утечки права R («Должны быть предусмотрены средства управления, ограничивающие распространения прав на доступ») и к корректности реализации («Контроль доступа должен быть применен к каждому объекту и каждому субъекту (индивида или группе равноправных индивидов)»). Однако здесь мы лишь акцентируем на этом внимание читателя, сами же эти вопросы мы рассмотрим позже.

Теперь, в двух словах, о достоинствах и недостатках метода дискретного контроля доступа. Как достоинства, так и недостатки определяются возможностью точного (избирательного) задания правил разграничения доступа.

Достоинства. Определяются возможностью точного задания разграничительной политики доступа, поскольку могут устанавливаться конкретные права доступа на множество R между каждой парой субъект-объект.

Основное достоинство – это универсальность – возможность использования для реализации контроля доступа к разнородным объектам с использованием необходимых для этого прав доступа. К недостаткам можно отнести сложность администрирования системы защиты, реализующей данный метод контроля доступа. Как следствие, задачу упрощения администрирования следует рассматривать в качестве одной из ключевых задач, решаемых при разработке методов дискретного контроля доступа (о других задачах будем говорить далее).

1.4. Контроль доступа на основе меток безопасности (мандатов)

Контроль доступа на основе меток безопасности (мандатов) называется методом мандатного контроля доступа. Если его сравнить с описанным выше методом дискретного контроля доступа с принудительным управлением потоками информации, метод мандатного контроля доступа призван упростить задачу администрирования, причем, как при запуске системы в эксплуатацию (при первоначальной настройке разграничительной политики доступа), так и в процессе последующей эксплуатации.

Замечание. Изначально принципиальное отличие метода мандатного от метода дискретного контроля доступа позиционировалось в реализации мандатным контролем принудительного управления информационными потоками. Однако здесь и далее мы сравниваем эти методы в предположении, что методом дискретного контроля доступа также реализуется принудительное управление информационными потоками.

Под мандатным контролем доступа понимается способ обработки запросов диспетчером доступа, основанный на формальном сравнении диспетчером в соответствии с заданным правилом меток безопасности (мандатов), назначаемых объектов и объектам доступа (в общем случае группам субъектов и объектов). Метки безопасности, как правило, являются элементами линейного упорядоченного множества М = {M1, ..., Mn} и служат для формализованного представления каких-либо свойств субъектов и объектов.

Разграничение доступа диспетчером реализуется на основе задаваемого правила, определяющего отношение линейного порядка на множестве M.
Контроль доступа к компьютерным ресурсам

где для любой пары элементов из множества $M$, задается один из типов отношения: $>$, $<$, $=\neq$ (на практике реализуется выбор подмножества $M$, изоморфного конечному подмножеству натуральных чисел – такой выбор делает естественным арифметическое сравнение меток безопасности). Правило сравнения меток также назначается из каких-либо свойств субъектов и объектов, применительно к решаемой задаче защиты информации.

1.4.1. Защита от хищения (понижения категории) категорированной информации

Реализация защиты

Наиболее широкое практическое использование мандатного метода нашло применение практики секретного делопроизводства в компьютерной обработке информации. Данная модель контроля доступа получила название «модели Белла-Ла Падулы» [39]. Основу реализации обработки категорированной информации составляет классификация информации по уровням конфиденциальности. Метки безопасности объектов отражают категорию конфиденциальности информации, которая может быть сохранена в соответствующих объектах. Метки безопасности субъектов отображают полномочия (по аналогии с формой допуска – уровень доступа) субъектов, в части допуска к информации различных уровней конфиденциальности.

Рассмотрим соответствующую модель мандатного контроля доступа.

Будем считать, что чем выше полномочия субъекта и уровень конфиденциальности объекта, тем меньше их порядковый номер в линейно полномочно упорядоченных множествах субъектов и объектов – $C = \{C_1, \ldots, C_t\}$ и $O = \{O_1, \ldots, O_l\}$, и тем меньшее значение метки безопасности $M_i$, $i = 1, \ldots, l$ им присваивается, т.е.: $M_1 < M_2 < M_3 < \ldots < M_l$.

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 $M$.

Замечание. В общем случае метка присваивается группе равноправных (имеющих одинаковые полномочия) субъектов и группе объектов одного уровня конфиденциальности.

Используем следующие обозначения:

$M_c$ – метка безопасности субъекта (группы субъектов) доступа;

$M_o$ – метка безопасности объекта (группы объектов) доступа.

Модель Белла-Ла Падулы применяется с целью защиты от нарушения конфиденциальности информации. что обеспечивается реализацией следующих формализованных правил, направленных на защиту от понижения категории обрабатываемой информации:

1. Субъект $C$ имеет доступ к объекту $O$ в режиме «Чтение» в случае, если выполняется условие: $M_c < M_o$.

2. Субъект $C$ имеет доступ к объекту $O$ в режиме «Запись» в случае, если выполняется условие: $M_c = M_o$.

Иногда также рассматривается возможность записи и при условии $M_c > M_o$.

Замечание. Большинство современных приложений редко использует открытие файла только на запись, как правило, одновременно на запись и чтение. Как следствие, возможность записи при условии $M_c = M_o$, сегодня следует рассматривать, скорее, как теоретическую возможность.

Таким образом, при запросе доступа субъекта к объекту, диспетчер доступа перехватывает запрос, определяет метку субъекта, запрошенного доступ, $M_c$, метку объекта, к которому запрошен доступ, $M_o$, и тип запрашиваемого доступа (запись, чтение). Сравнивает между собой метки $M_c$ и $M_o$, и на основании формализованных правил контроля доступа разрешает субъекту запрошенный им тип доступа к объекту, либо отказывает в нем.

Для иллюстрации формализованной рассматриваемой модели правил контроля доступа может использоваться матрица доступа $M_m$ (расширенные правила представлены матрицей доступа $M_{m\text{р}}$):

$O_1(M) \quad O_2(M) \quad \cdots \quad O_l(M)$

$C_1(M) \quad C_2(M) \quad \cdots \quad C_t(M)$

$M_m = \begin{bmatrix}
C_1(M_c) & 0 & r, w, d & \ldots & r \\
C_2(M_c) & 0 & r, w, d & \ldots & r \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{t-1}(M_c) & 0 & 0 & \ldots & r \\
C_t(M_c) & 0 & 0 & \ldots & r, w, d \\
\end{bmatrix}$

$O_1(M) \quad O_2(M) \quad \cdots \quad O_l(M)$

$C_1(M) \quad C_2(M) \quad \cdots \quad C_t(M)$

$M_{m\text{р}} = \begin{bmatrix}
C_1(M_c) & 0 & r, w, d & \ldots & r \\
C_2(M_c) & 0 & r, w, d & \ldots & r \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{t-1}(M_{c\text{р}}) & w & w & \ldots & r \\
C_t(M_{c\text{р}}) & w & w & \ldots & r, w, d \\
\end{bmatrix}$
Глава 1. Задачи и основополагающие методы защиты информации от НСД

Данные правила проиллюстрированы на рис. 1.3.

Таким образом, мандатный контроль доступа реализует защиту от понижения категории обрабатываемой информации с целью изменения режима ее обработки.

Рис. 1.3. Иллюстрация правил контроля доступа, направленных на защиту от понижения категории обрабатываемой информации

Преимущества по сравнению с дискреционным методом

Что же собою представляет метод мандатного контроля доступа по сравнению с дискреционным в предположении о том, что обоими этими методами контроль доступа реализуется принудительное управление информационными потоками? Методом мандатного контроля доступа решается частная задача защиты информации от несанкционированного доступа – задача защиты от понижения уровня конфиденциальности обрабатываемой информации. При этом одинаковая разграничительная политика доступа может быть реализована и методом дискреционного, что иллюстрирует представленная выше матрица доступа $M_n$, и методом мандатного контроля доступа, см. рис. 1.3. Как следствие, метод мандатного контроля доступа следует рассматривать как возможный метод решения частной задачи защиты, предполагающей собственно в своей постановке возможность формализации правил разграничения доступа, за счет возможности задания нерярхии субъектов и объектов доступа. Именно возможность формализации правил позволяет задавать права доступа не избирательно, а назначением меток безопасности субъектам и объектам, исходя из соответствующих иерархической признаком.

В результате этого меняется, как собственно задача администрирования реализующего подобный метод контроля доступа средства защиты, так и обработка запроса диспетчером доступа – реализуется не выбор правила из соответствующей ячейки матрицы доступа, а арифметическое сравнение меток безопасности.

Естественно, что с учетом сказанного, метод мандатного контроля доступа может рассматриваться в качестве одного из возможных решений рассматриваемой частной задачи защиты информации – защиты от понижения уровня ее конфиденциальности, реализация которого призвана упростить задачу администрирования средств защиты, реализующего контроль доступа, за счет возможности формализации правил разграничения доступа (иное какое-либо назначение данного метода автором определить не представляется возможным).

На первый взгляд, при реализации мандатного контроля доступа, задача администрации принципиальной упрощается – сводится лишь к назначению меток безопасности субъектам и объектам доступа (правила разграничения доступа формализованы, исходно задачи и настроек не требуют). Но это лишь в теории. На практике не все так просто. При подобной формализации задача разграничения доступа сразу же возникает масса вопросов: а что делать с правом исполнения (х), как размещать системы объекты, например, папки Windows, Program Files и др., как размещать системных субъектов доступа, например, System и т.д.? Ведь под подобную формализацию (категорирование информации по уровням конфиденциальности) подпадают лишь объекты, предназначенные для хранения непосредственно данных.

Проблема неразделяемых системных объектов

Особо здесь стоит остановиться на системных объектах, не разделяемых системой и приложениями, и принципиальной невозможности их разметки – назначения им меток безопасности.

Проиллюстрируем сказанное примерами. Поставим на инструментальный аудит (о задачах и методах аудита подробно поговорим далее) запросы к файловым объектам Интернет-браузера, запущенного с правами пользователя. Журнал аудита запрещаемого доступа к файловым объектам Ин-
тернет-браузера Mozilla Firefox при его запуске приведен на рис. 1.4. Интернет-браузера Internet Explorer — на рис. 1.5.

на рис. 1.4. Журнал аудита запрашиваемого доступа к файловым объектам Интернет-браузера Mozilla Firefox при его запуске.

Теперь предположим, что пользователю мы установили метку безопасности (мандат). Посмотрим на системные объекты, к которым осуществляется обращение от учетной записи пользователя при запуске браузера, см. рис. 1.4, рис. 1.5.

Следуя идеи мандатного контроля доступа, всем этим объектам должны быть присвоены соответствующие метки, иначе доступ к ним приложению будет запрещен. Но какие метки им присваивать, исходя из каких соображений, ведь это системные объекты, никак не связанные с обрабатываемой на компьютере категорированной информацией, они не подпадают под иное формализованную, основанную на классификации уровней конфиденциальности? И как выявить те файловые объекты, которые в обязательном порядке должны быть размещены? Ведь подобные обращения к системным ресурсам происходят практически при работе с любым приложением и не только с приложением.

Массовые обращения к незапертым системным объектам с правами пользователя осуществляются и собственно системой. Достаточно посмотреть журнал аудита — подобные обращения при входе пользователя в систему идут к файлам с расширениями ini, ink и др.

на рис. 1.5. Журнал аудита запрашиваемого доступа к файловым объектам Интернет-браузера Internet Explorer при его запуске.

Еще раз отметим, все эти обращения осуществляются с правами пользователя, т.е. в подобные объекты может осуществлять запись и собственное пользователь. Как следствие, возникает еще одна, куда более существенная проблема — проблема принципиальной невозможности разметки подобных объектов. Дело в том, что объекту можно присвоить лишь одну метку, при этом запись в подобный объект будет разрешена субъектом только размеченным какой меткой, остальные субъекты не получат требуемой возможности записи в соответствующие системные объекты со всеми вытекающими из этого последствиями.

Остаётся только одна возможность сделать системные ресурсы доступными всем пользователям — не устанавливать на них метки безопасности. Но в этом случае все субъекты, вне зависимости от назначенных им меток безопасности, получат полный доступ к одним и тем же объектам, что в корне противоречит самой идее мандатного контроля доступа! При этом отметим, что запишь осуществляется пользователем (не системой) процессом приложения (не системным процессом), см. рис. 1.4, рис. 1.5.
Вывод. Практическая реализация современных систем и приложений предполагает штатный множественный коллективный доступ пользователей к неразделяемым ими системным объектам, что, в том числе, может использоваться для несанкционированного обмена информацией между учетными записями, как следствие, штатно присутствует соответствующая утечка прав доступа.

Вывод. По возможности, средством защиты, реализующим принудительное управление потоками информации, должна предотвращаться утечка прав доступа, вызываемая коллективным доступом пользователей к неразделяемой системной и приложением системным объектам. Однако это может привести к нарушению штатного режима функционирования системы и приложений, и, как следствие, к выбору для конечного потребителя средства защиты между корректно работающей системой (приложением) и безопасной системой (приложением).

Вывод. В общем случае говорить о возможности практической реализации, и уж только при этом о какой-либо потенциальной возможности упрощения задачи администратора, за счет реализации метода мандатного контроля доступа, можно только в предположении, что из схемы мандатного контроля будут исключены системные объекты, т.е. мандатный контроль доступа в жестко формализованными правилами, не позволяет реализовывать разграниченную политику доступа к системным объектам.

Вывод. Разграниченная политика доступа к системным объектам, реализуемая с целью обеспечения корректности принудительного управления потоками информации, за счет защиты от утечки прав доступа, применительно к системным объектам, должна строиться с применением метода дискретного контроля доступа.

Следствие. Наиболее критичными в современных условиях, с учетом коллективного доступа пользователей к неразделяемой системной и приложением системным объектам, является именно предоставление возможности обработки на одном компьютере одним сотрудником предприятия информации различных уровней конфиденциальности, что, как правило, и требуется на практике.

Другая принципиальная проблема задания разграниченной политики при реализации мандатного контроля доступа состоит в том, что одна и та
жекетка безопасности в общем случае должна назначаться не отдельным пользователям, а группе пользователей, имеющих одинаковый уровень доступа к конфиденциальной информации. Для разграничения же прав доступа между различными пользователями, которым присвоена одинаковая метка безопасности, в дополнение к мандатному должен использоваться дискретный контроль доступа.

Опять вернемся к матрице доступа, описывающей мандатный контроль доступа:

\[
M_m = \begin{pmatrix}
    C_1(M_1) & 0 & \cdots & 0 \\
    C_2(M_2) & 0 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    C_t(M_t) & 0 & \cdots & 0 \\
\end{pmatrix}
\]

Из неё видим, что прочтенными пользователям, имеющим доступ к конфиденциальной информации, открыты данные, он может сохранять уже только, как конфиденциальные. Наверное, подобное решение трудно признать разумным!

Иное возможное решение, снимающее данное противоречие (можно рассматривать, как альтернативным, поскольку реализуются иные принципы задания правил доступа на основе меток безопасности) состоит в реализации правила ненерархической обработки иерархических меток безопасности. Введем несколько определений. Заметим, что мы говорим об иерархических метках безопасности, назначение которых осуществляется в соответствии с определенными категориями иерархическими признаками, например, по уровням конфиденциальности, и о различных правилах их обработки при анализе запроса доступа.

Определение. Под иерархической обработкой иерархических меток безопасности будем понимать такую обработку, при которой правилом, формирующим режимы доступа, учитывается иерархия меток, позволяющая использовать арифметическое сравнение, исходя из типов отношений: \(>\), \(<\), \(=\).

Определение. Под ненерархической обработкой иерархических меток безопасности будем понимать такую обработку, при которой правилом, формирующим режимы доступа, не учитывается иерархия меток, арифметическое сравнение осуществляется, исходя из одного типа отношений: \(=\).
получает возможность сохранения (записи) обработанной информации, в режиме, соответствующей ее категории, а не формы допуска пользователя (в частности, открытая информация будет сохранена, как открытая, а не как конфиденциальная, что соответствует форме (уровню) допуска пользователя). В общем случае метки безопасности могут быть как иерархическими (расмотрели выше), так и ненерархическими.

Определение. Под ненерархической меткой безопасности будем понимать метку, назначаемую субъектам и объектам доступа, не предполагающим возможности иерархического категорирования их соответствующих признаков.

Естественно, что применительно к ненерархическим меткам безопасности должна реализовываться нерерархическая обработка меток.

Требования к мандатному контролю доступа

Рассмотрим требования к мандатному контролю доступа, регламентируемые отечественным нормативным документом в области информационной безопасности [2]:

- Для реализации мандатного принципа должны сопоставляться классификационные метки каждого субъекта и каждого объекта, отражающие их место в соответствующей иерархии. Посредством этих меток субъектам и объектам должны назначаться классификационные уровни (уровни уязвимости, категории секретности и т.п.), являющиеся комбинациями иерархических и ненерархических категорий. Данные метки должны служить основой мандатного принципа разграничения доступа.

- КСЗ при вводе новых данных в систему должен запрашивать и получать от санкционированного пользователя соответствующие 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 КСЗ).

- КСЗ должен 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, и иерархические категории в классификационном уровне объекта включают 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, и все иерархические категории в классификационном уровне объекта включаются в нерерархические категории и 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, осуществляющее искажение всех обращений субъектов к объектам, а также разграничение доступа в соответствии с заданным принципом разграничения доступа. При этом решение о санкционированности запроса на доступ должно приниматься только при одновременном разрешении его и дискретционными, и мандатными ПРД. Таким образом, должны контролироваться не только единичный акт доступа, но и поток информации.

Как видим, регламентируется одновременное использование и мандатного, и дискретционного методов контроля доступа. Требование же «реализовать мандатный принцип контроля доступа применительно ко всем объектам» не дает ответ на вопрос, как включать в схему мандатного контроля доступа системные объекты.

Серьезные вопросы вызывает следующее требование: «КСЗ при вводе новых данных в систему должен запрашивать и получать от санкционированного пользователя классификационные метки этих данных». Если пользователь самостоятельно может назначать метку созданного им файла объекта, то о каком принудительном управлении информационными потоками идет речь? Отметим, что назначение метки объекту при мандатном методе, это, по сути, назначение прав доступа к этому объекту, другими словами, реализация сущности «Владение» для метода мандатного контроля доступа. О противоречиях использования сущности «Владение» мы уже ранее говорили.

Определенный интерес представляет то, как сформулированы требования к дискретционному контролю, при реализации его совместно с мандатным. Как видим [2], эти требования не зависят от использования дискретционного метода эксплицитно (требования приведены ранее), либо совместно с мандатным. Данные требования содержат в себе серьезные противоречия. В частности, требование «Для каждой пары (субъект – объект) в средстве
вычислительной техники (СВТ) должно быть задано явное и недопустимое перечисление допустимых типов доступа (читать, писать и т.д.), т.е. тех типов доступа, которые являются санкционированными для данного субъекта (индивиду или группы индивиду) к данному ресурсу СВТ (объекту). Определяет необходимость реализации в данном случае разрешительной разграничительной политики доступа. Что это означает? А означает это то, что правда доступа, например, матрицы \( M^* \), должны задаваться дважды, и мандатным, и дискретным механизмами контроля доступа. Это уже не упрощение, а кардинальное усложнение задачи администрирования!

Заметим, что разрешительная разграничительная политика задается собственности мандатным контролем доступа, в рамках которой уже необходимо дать разграничения (между субъектами одной группы) дискретным контролем. Как следствие, в данном случае следует говорить об использовании запретительной, а не разрешительной политики доступа, требование же в этом случае будет выглядеть следующим образом: «Для каждой неры (субъект — объект) в среде вычислительной техники (СВТ) должно быть задано явное и недопустимое перечисление недопустимых типов доступа (читать, писать и т.д.), т.е. тех типов доступа, которые являются несанкционированными для данного субъекта (индивида или группы индивиду) к данному ресурсу СВТ (объекту). В противном случае, применение мандатного контроля доступа становится просто бессмысленным, при условии, что дискретный контроль доступа используется в дополнение к мандатному для реализации разграничений прав доступа между субъектами, которые назначена одна и та же метка безопасности.

А теперь исследуем следующий важнейший вопрос, определяющий область применения рассмотренной модели мандатного контроля доступа — насколько абстрактна, в смысле универсальности (общности), рассмотренная модель контроля доступа. Ранее, применительно к дискретному контролю доступа мы сделали вывод об универсальности рассматриваемой модели, которая применима к реализации контроля доступа к разнородным объектам с соответствующей поправкой на то, какие права доступа могут задаваться субъектам к объектам с учетом его физических особенностей.

Мандатный контроль доступа (рассматриваем основную решаемую им задачу защиты от понижения категории обрабатываемой информации) применяется в отношении однозначно категорируемых объектов, это, в первую очередь, объекты, используемые для хранения данных — файлы. Если же рассматривать файловые объекты в общем случае, то под категорирование не подпадают системные файлы, к которым можно отнести исполняемые файлы и файлы конфигураций. Применительно к файловым объектам можно отметить и ограничение на контролируемое права доступа — реализуется управление потоками записи и чтения. Существуют собственно физические объекты, в отношении которых какое-либо категорирование не применимо, например, это объекты реестра ОС. Если же категорирование в отношении ряда объектов и применимо, например, внешний накопитель, принтер и т.д., то в общем случае оно не однозначно, что не позволяет формализовать правила контроля доступа к этим объектам. Например, один внешний накопитель может использоваться для хранения данных не выше категории «конфиденциально», другой, наоборот, не ниже этой категории, отдельный принтер может выделяться для печати открытых данных, и отдельный — только конфиденциальных.

Естественно, возникают вопросы, как о целесообразности практической реализации мандатного контроля доступа, так и о подходах к его реализации, к иным объектам, которые могут использоваться для обработки данных различных категорий (уровней) конфиденциальности.

1.4.2. Защита от нарушения целостности информации

Решение задачи защиты от нарушения целостности информации с использованием мандатного метода контроля доступа предлагается «моделью целостности Биба» [58], суть которой состоит в включении в систему неразрывного признака целостности, отображаемого в разграничительной политике доступа мандатом или меткой безопасности. В модели Биба вводятся уровни целостности, сопоставляемые с субъектами и объектами доступа, которые, в соответствии с заданным уровнем присваивается метка безопасности. Соответствующим образом изменяются и правила контроля доступа, которые здесь приобретают следующий вид:

1. Субъект \( C \) имеет доступ к объекту \( O \) в режиме «Чтения» в случае, если выполняется условие: \( M^*_C > M^*_O \).
2. Субъект С имеет доступ к объекту О в режиме "Записи" в случае, если выполняется условие: \( C < M_o \). В результате получаем полную противоположность (инверсию) модели Белла-Лападулы. Матрица доступа \( M_{a0} \) здесь имеет следующий вид:

\[
\begin{bmatrix}
O_1(M_1) & O_1(M_2) & \cdots & O_1(M_n) \\
O_2(M_1) & O_2(M_2) & \cdots & O_2(M_n) \\
\vdots & \vdots & \ddots & \vdots \\
C_{1-1}(M_{1-1}) & C_{1-1}(M_{2-1}) & \cdots & C_{1-1}(M_{n-1}) \\
C_{1}(M_1) & C_{1}(M_2) & \cdots & C_{1}(M_n)
\end{bmatrix}
\]

До сих пор данная модель больше рассматривается, как некая абстракция. В частности, модель Белла неоднократно специалисты критиковали за то, что она использует целостность как некую меру и ставил под сомнение то, что понятие "целостности" имеет какой-либо физический смысл. Их аргументом было то, что целостность субъектов и объектов следует рассматривать как двоичный атрибут, который или есть, или нет, что, вообще говоря, разумно.

Те. не понятен принципиальный вопрос, каким способом, и на основании чего, проводить классификацию субъектов и объектов по параметру "целостность".

В последующих разделах мы еще вернемся к этому важному вопросу, но уже несколько с иных позиций.

### 1.5. Задачи защиты информации от несанкционированного доступа

#### 1.5.1. Формирование режимов обработки информации субъектами доступа

В этом разделе мы рассмотрели базовые методы контроля доступа, проанализировали их с использованием соответствующих абстрактных моделей, определили достоинства и недостатки.

Рассмотрим ключевой вопрос — с какой целью могут на практике использоваться подобные методы защиты информации от несанкционированного доступа? И с какой целью и для решения каких задач ими должна реализовываться разграничительная политика доступа?

Ранее мы говорили о том, что под защитой информации от несанкционированного доступа в общем случае понимается защита информации, направленная на предотвращение получения защищаемой информации заинтересованными субъектами с нарушением установленных нормативных и правовых документов (актами) или обладательными информацией прав или правил разграничения доступа к защищаемой информации — с нарушением реализованной разграничительной политики доступа. Что может по-определениться в качестве функциональной задачи защиты информации от несанкционированного доступа. При этом ключевым элементом разграничительной политики является защищаемый объект доступа (в частности, файловый объект), к которому разграничиваются права доступа субъектов.

Однако, в такой постановке функциональная задача защиты информации от несанкционированного доступа выглядела в начале века. Сегодня она приобретает совсем иное значение.打造成 связано с переносом на компьютерную обработку множества различных задач решаемых с различными целями, что в том числе приводит к значительному увеличению номенклатуры устройств, как средств, условий, используемых при ее обработке с различными целями. Как следствие, подразумевается понятие режима обработки информации, различные режимы обработки информации пре- следуют соответственно различные цели, причем на одном и том же вычислительном среде может потребоваться обработка информации в различных режимах, в том числе, одним и тем же пользователем.

С учетом сказанного, функциональной задачей защиты информации в современных условиях, соответственно и задачей реализации разграничительной политики доступа является не решение нейкой локальной задачи — реализация контроля и разграничения прав доступа субъектов к каким-либо отдельно взятым объектам, а формирование режимов обработки информации для субъектов в информационной системе.

В рамках решения функциональной задачи защиты информации в подобной же постановке уже требуется не защищать доступ не к отдельным объектам, а организовать режимы обработки информации с различными целями, защищать эти режимы, а, при необходимости, их разделять — предотвращать обмен информацией между различными режимами. Это обусловливается тем, что различные режимы могут характеризоваться различными угрозами, а также, по-каким-либо причинам, существенно различающимися требованиями к безопасности обработки в них информации.

---

1 Например, локальная работа на компьютере и работа в сети
Как видим, в качестве субъекта доступа здесь уже выступает режим, для которого должны назначаться права доступа к объектам (а не наоборот, к объекту субъектам) — защищается не объект доступа, а режим обработки информации. Отличие моделей контроля доступа при этом, как отмечают, состоит в использовании прямой и транспонированной матриц доступа.

Все это принципиально меняет подходы к построению эффективных методов контроля доступа в современных условиях.

Определение. Под режимом обработки информации субъектом будем понимать выполнение им в информационной системе ограниченного какими-либо соображениями (целями формирования режима обработки) функционального набора задач обработки информации, включая использование необходимо для их решения набора ресурсов (объектов доступа) и прав доступа субъекта к этим объектам.

Определение. Функциональная задача защиты информации от несанкционированного доступа предполагает формирование режимов обработки субъектами информации, характеризуемых различными условиями и требованиями к ее обработке в информационной системе, с целью повышения безопасности обработки информации в информационной системе.

Замечание. Решение функциональной задачи защиты информации (реализации разграничительной политики доступа) позволяет реализовать защиту информации отдельно для каждого режима ее обработки, создавая защищенный режим обработки информации, что принципиально важно, в том числе, при различных требованиях к безопасности обработки информации в различных режимах.

Определение. Под защищенным режимом обработки информации субъектом будем понимать режим обработки информации, в котором реализована ее защита от несанкционированного доступа, по средству обеспечения контроля доступа субъектов ко всем объектам, используемым в данном режиме обработки информации.

Замечание. Говоря далее о формировании режимов обработки информации субъектами в информационной системе, будем понимать, что речь идет о формировании защищенных режимов обработки информации.

Важнейшим требованием к корректному формированию режима обработки информации субъектом доступа можно отнести следующее. Для возможности корректного формирования защищенного режима обработки информации субъектом, и информационной системе должна быть реализована возможность локализации набора объектов доступа для субъекта.

Здесь, в первую очередь, речь идет об устройствах. В отношении локальных устройств данная задача защиты должна решаться управлением разграничения (подключения отключения) к аппаратному устройству. При этом к системе в любой момент времени функционирования должны быть подключены (либо отключены) только устройства, необходимые для формируемого режима обработки информации субъектом.

Однако устройства могут быть не только локальными. Например, принтер может быть локальным, разделенным в сети (подключен к другому компьютеру), может быть сетевым объектом (подключен к сети с своим сетевым адресом).

Замечание. Сказать, что компьютер автономный, исходя из того, что к нему не подключен сетевой кабель, это ни о чем не сказать до тех пор, пока на этом компьютере не будут отмечены от системы все интерфейсы (сетевые адаптеры), позволяющие подключать к системе внешние устройства, а также использовать встроенные устройства доступа в сеть.

Следующее важное требование. Ко всем объектам доступа в информационной системе, в том числе, к подключенным и к разрешаемым для подключения субъектами устройствам, предполагающим обработку информации в формируемых защищенных режимах обработки информации, должен быть реализован контроль и разграничения прав доступа, т.к. в противном случае невозможно корректно сформировать защищенные режимы обработки информации субъектами.

Здесь важно остановиться на следующем. Многие локальные устройства, например, принтеры, сканеры и т.д. позволяют применять к ним при реализации разграничительной политики доступа только одно право доступа — решать или запрещать использование устройства субъекту. Как следствие, в отношении подобных устройств (объектов доступа), а их достаточно много, контроль доступа может реализовываться методом управления монтирования устройств по субъектам доступа (по пользователям) [57].

Данный метод универсален в том смысле, что одним средством защиты, которое в любом случае должно использоваться для управления монтированием устройств, может быть реализованный контроль доступа ко всем подобным устройствам, что, естественно, минимизирует загрузку вычислительного ресурса. При этом данный метод контроля доступа не предполагает

2 Соответственно, включая устройства, обеспечивающие возможность и корректность функционирования информационной системы
Давайте рассмотрим, что данный подход дает с точки зрения безопасности. Во-первых, за счет формирования режимов обработки информации соответствующим образом, для отдельно взятого режима локализуются и угрозы безопасности, которых для отдельно взятого режима обработки в данном случае становится меньше, чем для информационной системы в целом. Если же исходить из того, что в рамках роли субъекту предоставляются только необходимые для его работы ресурсы, то можно заключить, что набор угроз безопасности для режима обработки информации, сформированного для соответствующей роли, как следствие, и для информационной системы в целом, становится минимальным. Как следствие, реализацию ролевой модели контроля доступа можно позиционировать в качестве способа минимизации угроз безопасности информационной системы. Отметим, что создание соответствующего режима обработки информации, с целью минимизации угроз безопасности, имеет смысл и на вычислительном средстве, предназначенном для реализации на нем только одной роли. Во-вторых, ввиду локализации ресурсов, используемых каждой ролью, упрощается задача реализации контроля доступа с целью предотвращения несанкционированного обмена информацией между ролями — получения пользователем, работающим в одной роли, информации, обрабатываемой в иной роли, работа в которой не предусматривается его функциональными обязанностями.

Однако существует понятие и метода ролевого контроля доступа [82]. Рассмотрим, как может быть реализована ролевая модель, и при каких условиях можно говорить о методе ролевого контроля доступа, как о самостоятельном методе контроля доступа.

Например, в простейшем случае роль может соотноситься с учетной записью (учетная запись — это роль), сформированной для различных учетных записей соответствующие задаваемым им ролем режимы обработки информации субъектами. Все сотрудники, допущенные к работе в одной роли, будут входить в систему под одной и той же учетной записью, пользуясь одинаковым пароль для входа в систему. Одни и тот же сотрудник может работать в различных ролях (если ему предоставлен пароль для входа в систему под соответствующими учетными записями), выбор роли осуществляется входом под учетной записью соответствующей требуемой роли, смена роли реализуется сменой пользователя штатными средствами операционной системы. Естественно, что при таком решении задача администрирования системы защиты может быть достаточно сложна.

Задача упрощения администрирования решается в том случае, если с ролью сопоставить группу пользователей. Формирование режимов обработки информации субъектами доступа заключается в назначении правил
доступа к разнородным объектам для соответствующих групп пользователей. Для предоставления сотруднику работы в роли, для него заводится учетная запись, входящая в группу, соответствующую данной роли. Если сотруднику необходимо несколько ролей, для него должны быть занесены учетные записи по числу данных ролей, включаемые в соответствующие группы, для которых назначаются правила доступа. При входе в систему под определенной ролью сотрудник должен выбрать учетную запись, соответствующую данной роли, для смены роли штатными средствами операционной системы сменить учетную запись.

Отметим, что упрощение администрирования в данном случае не существенно, поскольку создание учетной записи — это задача на порядки ниже в своем решении, чем задача создания прав доступа к объектам, особенно к таким, как файловые объекты, объекты реестра ОС и т.д. Вместе с тем, кардиальным упрощением задачи администрирования, на взгляд авторов, следует искать в реализации и использовании базовых методов контроля доступа, а не в способах задания субъекта доступа при формировании режимов обработки информации субъектами. Далее мы это и будем рассматривать.

Отличительной особенностью метода ролевого контроля доступа является включение в схему контроля доступа в качестве субъекта доступа сущности "роль", в таком смысле, что именно для этой сущности (не для учетных записей) назначаются права доступа к объектам при формировании режимов обработки информации субъектами (субъект доступа уже не пользователь, а собственно сущность "роль"). Вход в систему сотрудник в этом случае будет под заведомо для него единственной учетной записью, но при этом вместе с идентификацией и аутентификацией его при входе в систему, пользователю будет предложено выбрать роль. При выборе пользователем соответствующей роли (из разрешенного для него набора ролей), вступают в силу разграничения прав доступа к объектам, назначенные для выбранного субъекта доступа роли (еще раз отметим, не для учетной записи, для которой в данном случае правила доступа к объектам не назначаются). Для смены роли также не потребуется смены пользователя, необходимо будет запустить соответствующее приложение, позволяющее выбрать новую роль без смены учетной записи.

Упрощение задачи администрирования при этом может позиционироваться упрощением добавления/удаления пользователей в системе. Для этого потребуется лишь добавить/удалить соответствующую учетную запись в системе и включить ее в роль, соответственно, исключить из роли. Особенностью практической реализации метода ролевого контроля доступа (а в данном случае уже можно говорить именно о методе контроля доступа) является то, что для одного пользователя заводится только одна учетная запись, под которой пользователь может работать в различных ролях (в тех ролях, в которые включена данная учетная запись).

Если существенное упрощение задачи администрирования, по сравнению с реализацией ролевой модели при соединении роли с группой пользователей, отнюдь не очевидно, то потенциальная возможность внесения весьма критической угрозы безопасности при реализации разграничительной политики доступа для субъекта доступа "роль" весьма вероятна.

Рассмотрим источники подобной угрозы безопасности. Все разграничительные политики современных системных средств и приложений основаны на реализации разграничений прав доступа к объектам для учетных записей. А это ведь не только объекты, используемые для хранения обрабатываемых в информационной системе данных, и системные объекты, например, конфигурационные файлы, соответствующие объекты реестра ОС, причем данные разграничения не требуют в большинстве случаев каких-либо настроек — они установлены в системе по умолчанию. Напомним, что при ролевом контроле доступа сотрудник в различных ролях будет работать под одной и той же учетной записью. Т.е. при таком решении отвергается основа разграничительной политики доступа, реализуемой в современных ОС. Посмотрим на примере, что произойдет, если конфигурационные объекты не разделить между различными учетными записями.

Например, на компьютере при работе с сетью браузерами создаются, так называемые, Cookies ("куки") — это небольшая порция текстовой информации, которую веб-узел (сайт) при обращении к нему передает браузеру. Браузер будет хранить эту информацию и передавать ее обратно с каждым запросом к этому сайту. Одни значения cookies могут храниться только в течение одного сеанса (единовременного сеанса связи с сайтом), они удаляются после закрытия браузера. Другие, установленные на некоторый период времени, записываются в специальный файл и хранятся на компьютере. Обычно такой файл называется «cookies.txt» и находится в рабочей директории браузера. Само же приезжать cookies не могут выполнять ничего, это только лишь некоторая текстовая информация. Однако сайт, к которому запрашивается доступ, может считывать содержащиеся в cookies сведения и, на основании их анализа, совершать те или иные действия. Можно представить себе, какое это приведет, если между ролями, например, предполагающими открытый доступ во внешнюю сеть (незащищенный режим) и работу с корпоративными сетевыми ресурсами, подобные объекты не разграничивались. А подобных примеров можно привести много. Например, тот же буфер обмена.
1.5.3. Сессионная модель и метод контроля доступа

Как отмечалось выше, функциональной задачей реализации разграничительной политики доступа является формирование режимов обработки информации для субъектов в информационной системе. В данном случае, в отличие от ролевой модели, режимы формируются для обработки информации различных уровней конфиденциальности с целью предотвращения нарушения конфиденциальности (предотвращения просмотра) обрабатываемой в информационной системе информации. Как следует, рассматриваемая модель контроля доступа основывается на использовании метода сессионного контроля доступа. Обусловливается необходимость реализации рассматриваемой модели контроля доступа тем, что обработка информации различных уровней конфиденциальности — в различных режимах, требует реализации в различных режимах защиты информации, обрабатываемой в различных режимах. Как следует, требуется реализовать защиту от любого способа незаконного использования информации в режиме обработки информации меньшего уровня конфиденциальности, где становится более вероятным ее использование, либо утечка.

Например, при работе с сетью в режиме обработки открытой информации в общем случае невозможно локализовать изображений безопасности на бор сетевых объектов (сайтов), к которым будет иметь право обращаться субъект. Другое дело — это обработка конфиденциальной информации с предоставлением возможности доступа пользователей только к корпоративным сетевым объектам. Естественно, уровень защиты в этих режимах обработки информации будет кардинально различным.

Для решения задачи формирования режимов обработки информации различных уровней конфиденциальности на одном компьютере на практике может использоваться сессионная модель контроля доступа и реализующий ее метод сессионного контроля доступа [13,48] (был рассмотрен выше). В качестве субъекта доступа в разграничительной политике доступа в данном случае выступает сессия. Поскольку различные сессии предоставляют обработку информации соответствующего уровня конфиденциальности — метка безопасности привязывает сессию. Пользователь же, имеющий соответствующий уровень допуска к конфиденциальной информации должен включаться в соответствующую сессию (сессии).

Отметим, что сессионная модель контроля доступа ни в коем случае не должна рассматриваться в качестве альтернативы ролевой модели, поскольку здесь совершенно иные требования к реализации защиты.
В отличие от ролевой модели, в рамках сессионной модели режимы обработки информации различных уровней конфиденциальности (сессий) должны в обязательном порядке разделяться (будем это называть разделительной политикой доступа [13,48]) — должна реализовываться разделительная политика доступа между различными сессиями.

Определение. Под разделительной политикой доступа будем понимать разделяние (изолирование) различных сессий — сессий, создаваемых для обработки данных различных уровней конфиденциальности.

Построим соответствующую модель сессионного контроля доступа. Будем считать, что в системе существует $k$ категорий конфиденциальности обрабатываемой информации, причем чем выше полномочия субъекта и уровень конфиденциальности объекта, тем меньше их порядковый номер в линейно полномочию упорядоченных множествах сессий и объектов доступа — $S = \{S_1, ..., S_k\}$ и $O = \{O_1, ..., O_k\}$, и тем меньше значение метки безопасности $M_x$ i $M_y$ им присваивается, т.е.: $M_1 < M_2 < M_3 < ... < M_k$, причем:

- $M_1$ — метка безопасности сессии;
- $M_k$ — метка безопасности объекта (группы объектов) доступа.

то правила, реализация которых направлена на защиту от понижения категории обрабатываемой информации при реализации сессионного контроля доступа в общем случае могут быть сформулированы следующим образом:

1. Из сессии $S$ может быть осуществлен доступ к объекту $O$ в режиме "Чтения" в случае, если выполняется условие: $M_x \leq M_y$.
2. Из сессии $S$ может быть осуществлен доступ к объекту $O$ в режиме "Записи" в случае, если выполняется условие: $M_x = M_y$.

Замечание. Именно необходимостью выполнения данного требования принципиально различаются ролевая и сессионная модели контроля доступа, практическая реализация которых в информационной системе преследует решение различных задач защиты информации.

Субъект доступа, имеющий соответствующий уровень допуска к обработке категорируемой по уровню конфиденциальности информации, может включаться в сессию, предназначенные для обработки информации, не превышающие уровень конфиденциальности соответствующий уровню допуска субъекта. В предположении, что назначаются субъекту и сессии метки безопасности из одного и того же множества $M$, и, исходя из того, что чем выше уровень допуска субъекта и уровень конфиденциальности объекта, к которому может осуществляться доступ в сессии, тем меньше значение метки безопасности $M_x$ из $M$ им присваивается, т.е.: $M_1 < M_2 < M_3 < ... < M_k$, соответственно обозначим эти метки $M_x$, и $M_y$, то субъект доступа, которому присвоена метка $M_y$ может включаться в те сессии, для которых выполняется условие $M_x \leq M_y$.

При выполнении условия $M_x \leq M_y$, с учетом выполнения в системе сформулированного выше требования, реализуются основополагающие правила мандатного контроля доступа субъектов к объектам, направленные на защиту от понижения категории обрабатываемой информации:

1. Субъект $C$ имеет доступ к объекту $O$ в режиме "Чтения" в случае, если выполняется условие: $M_x \leq M_y$.
2. Субъект $C$ имеет доступ к объекту $O$ в режиме "Записи" в случае, если выполняется условие: $M_x = M_y$.

Таким образом, реализация сессионной модели контроля доступа предполагает возможность работы одного и того же пользователя в различных доступных для него, что определяется уровнем допуска пользователя к обработке категорированной информации, сессиях — режимах обработки информации, в пределении возможности выбора текущей сессии пользователем.

Замечание. Поскольку пользователь может работать в различных доступных для него сессиях, за счет возможности выбора текущей сессии, т.е. обрабатывать информацию различных уровней конфиденциальности в соответствующих режимах, целесообразно использование следующего правила сессионного контроля доступа:

1. Из сессии $S$ может быть осуществлен доступ к объекту $O$ в режиме "Записи" и «Чтения» в случае, если выполняется условие: $M_x = M_y$.
2. В противном случае из сессии $S$ не может быть осуществлен доступ к объекту $O$.

По полной аналогии с ролевой моделью контроля доступа, для сессионного контроля сессия может задаваться учетной записью, сопоставляющей с группой субъектов (пользователей), либо в качестве субъекта доступа должна использоваться некая дополнительная сущность «сессия». В последнем случае здесь также может быть сформировано соответствующее требование к корректности реализации разграничительной политики доступа.
Замечание. Есть некоторое отличие, по сравнению с методом дискретного контроля доступа, в части реализации сопоставления сессии с группой субъектов (пользователей). Применительно к методу мандатного контроля доступа пользователи аннотируются в группы по совпадению их уровня допуска к категорируемой информации, именно в группе субъектов уже присваивается метка безопасности. Как следствие, для реализации сессионного контроля доступа для каждого сотрудника должны быть заведены учетные записи, по числу сессий, в которых ему разрешена работа (с учетом выполнения требования, формулируемого приведенным выше утверждением), каждой из которых должна быть присвоена соответствующая метка безопасности, чем и осуществляется включение учетной записи в соответствующую группу субъектов.

Естественно, для возможности корректной реализации в информационной системе метода сессионного контроля доступа с субъектом доступа «сессия», все системные средства и приложения, используемые в информационной системе, в отношении всех объектов доступа, используемых при формировании режимов обработки информации субъектами доступа, в том числе и конфигурационных, должны обеспечивать реализацию разграничителенной политики доступа для субъекта доступа «сессия».

Причем, в отличие от ролевой модели, в данном случае это должно быть в обязательном порядке.

Замечание. Исходя из проведенного анализа применительно к методу ролевого контроля доступа, очевидно, что в современных условиях модель сессионного контроля доступа корректно может быть реализована системой защиты только при реализации сопоставления сессии с субъектом — с учетной записью (с группой субъектов). Использование в разграничителенной политике доступа сущности "сессия" недопустимо. [48].

Как и для ролевого контроля доступа, для сессионного контроля доступа справедливо следующее требование «Ко всем объектам доступа в информационной системе, в том числе, к подключенным и к разрешаемым для подключения субъектами устройствам, предполагающим обработку информации в формируемых защищенных режимах обработки информации, должен быть реализован контроль и разграничения прав доступа, т.к. в противном случае невозможно корректно сформировать защищенные режимы обработки информации субъектами».

Глава 1. Задачи и основополагающие методы защиты информации от НСД

Как ранее отмечалось, однозначное категорирование объектов, к которым разграничиваются права доступа, в соответствии с уровнями конфиденциальности обрабатываемой информации, в общем случае возможно только в отношении файловых объектов, и только в той их части, которые используются в системе для хранения обрабатываемых в информационной системе данных. Естественно, возникает вопрос, как использовать метки безопасности (мандаты), на которых основана реализация сессионного контроля доступа, при реализации контроля доступа к иным объектам.

Универсальное решение рассматриваемой проблемы реализации сессионного контроля доступа состоит в следующем [57]. Для объекта (не файлового объекта), к которому разграничиваются права доступа субъектов, правила доступа наносятся заданием для него множества меток безопасности $M_j$, характеризующего набор субъектов, которые имеют право доступа к этому объекту.

Правило доступа, анализируемое при контроле доступа субъекта к объекту, при этом имеет следующий вид:

Субъект доступа, которому присвоена метка безопасности $M_j$, имеет доступ к объекту при выполнении условия: $M_j \in M$.

Остановимся с учетом сказанного на принципиальном противоречии метода мандатного контроля доступа к объектам, не являющимся файловыми, в отношении которых корректна модель Белла-Ла-Падулы. Дело в том, что пользователь, которому назначена некая метка безопасности $M_i$ при реализации модели Белла-Ла-Падулы сможет прочитать данные не только со значением той же метки, но и более низкого уровня конфиденциальности.

Доступ же к объекту (например, к некоторому устройству, которому назначено множество меток безопасности $M_j$) в этом случае пользователю будет разрешен при условии, что $M_j \in M$, при этом пользователь сможет выдать на это устройство и данные более низких уровней конфиденциальности (к которым он имеет право на чтение, как к соответствующим файловым объектам).

Как следствие, при реализации модели Белла-Ла-Падулы в отношении файловых объектов, в отношении объектов, не являющихся файловыми, корректным будет являться только одно формализованное правило доступа к этим объектам для субъекта, которому назначена метка безопасности $M_j$. 

56-57
Если при появлении данной технологии, реализующие ее системы позиционировались для использования в части защиты от инсайдерских атак (от ненадежных, несанкционированных воздействий на систему легальных пользователей, осуществляющих с целью хищения конфиденциальной информации), то в последние годы отношение к подобным системам кардинально изменилось — сегодня DLP системы позиционируются, как решения защиты от случайных, осуществляемых по халатности, действий легальных пользователей, потенциально приводящих к утечке конфиденциальной информации.

При принципиальном отличии постановки задачи защиты в том случае состоит в том, что контролируемая системой защиты конфиденциальность информацией умышленно не маскируется (не преобразуется) каким-либо образом пользователям перед ее отправкой за пределы информационной системы, что, естественно, принципиально упрощает задачу контроля и позволяет в этом случае говорить о какой-либо эффективности защиты. Активность же подобной задачи защиты обновляется разработчиками и поставщиками тем, что более 70% утечек связано именно с халатностью сотрудников.

Однако с этим никак нельзя согласиться, см. рис. 1.1.

Данная технология обладает еще одним важнейшим преимуществом. Сегодня многими приложениями, в том числе, сетевыми (в частности, приложениями электронной почты) информация шифруется перед ее отправкой во внешнюю среду. Естественно, что в отношении зашифрованной информации подобный анализ не применим.

Эффективно же данная задача защиты может решаться методом сессионного контроля доступа. Вообще говоря, отчасти именно для решения этой задачи защиты информации (естественно, при этом решается и задача защиты от сетевых атак, но об этом далее) он и разрабатывался [48].

Как отмечалось, основу реализации метода сессионного контроля доступа составляет формирование и изолирование (разделительная политика доступа) режимов обработки субъектами информации различных уровней конфиденциальности. Поскольку корректно сессионный контроль доступа может в современных условиях реализовываться исключительно при задании уровня «сессия» учетными записями, которым присваиваются метки безопасности, методом именно сложности администрирования метода сессионного контроля доступа сегодня ограничивается применение сессионного контроля доступа конфиденциальной информации на практике.

Однако далее мы рассмотрим метод мандатного, в том числе, сессионного контроля доступа к создаваемым объектам (к создаваемым файлам), сложность администрирования которого минимальна. Именно данный метод, на взгляд авторов, должен использоваться в основе реализации метода сессионного контроля доступа, что позволит эффективно его использовать на практике при решении одной из актуальных сегодня задач защиты — утечки от угроз со стороны легальных пользователей, направленных на хищение конфиденциальной информации.
1.6. Задача нивелирования угроз технологических уязвимостей — задача защиты от актуальных угроз атак

Как отмечали выше, функциональной задачей систем защиты информации от несанкционированного доступа, решаемой современными СЗИ НСД, является формирование с какой-либо целью (ролевой, либо сессионной модели контроля доступа) режимов обработки информации субъектами доступа (пользователями) в информационной системе.

При этом должны реализовываться требования к полноте реализации разграничительной политики доступа субъектов к объектам — к набору реализуемых методов контроля доступа, и к корректности реализации каждого метода в отдельности, что требует формирования соответствующих требований к корректности реализации (к построению безопасной системы) при разработке соответствующих методов контроля доступа. Не выполнение подобных требований создает угрозу безопасности, связанную с возможностью преодоления защиты злоумышленником.

Здесь будем говорить о технологической задаче реализации разграничительной политики доступа, соответственно, системы защиты информации от несанкционированного доступа, определяемой нами, как задачи защиты от актуальных угроз атак. Естественно, что данная задача должна решаться отдельно применительно к каждому сформированному режиму обработки информации, что обусловливает необходимость первоначального их формирования, т.к. различные режимы (например, локальная обработка информации и работа в сети) характеризуются различными угрозами атак (как следствие, решением различных задач защиты) и требованиями к безопасности. Отметим, что это ключевой момент в построении защиты от актуальных угроз атак.

С целью определения технологической задачи защиты информации — защиты информации от актуальных угроз атак, введем классификацию угроз уязвимостей, создающих угрозу атак.

Определение. Под угрозой технологической уязвимости информационной системы будем понимать технологические недостатки ее построения, включая отсутствие требуемых функций защиты информации, либо некорректность их реализации, не позволяющие в полном объеме реализовать защиту от несанкционированного доступа — реализовать необходимые права и (или) правила санкционированного доступа к информации, или (и) предотвратить доступ с нарушением установленных прав и (или) правил доступа.

Предположим, что все подобные сформулированные требования реализованы. Однако, и при этих условиях в той или иной мере существует угроза безопасности информационной системы, но уже по иным причинам. В этом случае злоумышленник уже может воспользоваться угрозами, связанными с ошибками реализации (например, ошибками программирования, подобные ошибки в том числе, позволяют осуществить SQL-инъекцию) системных средств и приложений. Пример отчета по количеству выявленных уязвимостей за год с отнесением их к уровню критичности, наглядно иллюстрирующему современное положение дел в информационной безопасности в рассматриваемой ее части, приведен на рис. 1.6 [84]. Отметим, что в соответствии с используемой классификацией угроз критичности уязвимости на уровне безопасности информационной системы существенно влияют уязвимости, характеризующиеся критической, высокой и средней степенями опасности (например, следующая классификация уязвимостей [84], к уязвимостям средней степени опасности относятся уязвимости, которые позволяют реализовать удаленный откат в обслуживании, неавторизованный доступ к данным или выполнение произвольного кода), см. рис. 1.6.

Рис. 1.6. Статистика обнаруженных уязвимостей в ОС и в браузерах

Понятно, что при подобной статистике обнаружения уязвимостей в системе и в приложениях, см. рис. 1.6, говорить о какой-либо безопасности современных информационных систем не приходится, соответствующий анализ достигаемого уровня информационной безопасности в этих условиях проведен в [35]. К слову сказать, подобные уязвимости (в частности, ошибки программирования) могут устраняться разработчиками системных средств и приложений месяцы. Все это время информационная система уязвима к атакам несанкционированного доступа к обрабатываемой в ней информации. Естественно, что защита от подобных атак в современных условиях...
стали доминирующей задачей защиты информации, причем, заметим, от несанкционированного к ней доступа! А ведь осуществить несанкционированный доступ к обрабатываемой информации невозможно не только с использованием ошибок реализации программных средств, но и по средствам внедрения и запуска на компьютере вредоносных программ, за счет нарушения вредоносными свойствами легальных приложений, предполагающих, например, штатную возможность работы приложений со скриптами и ActiveX-компонентами.

В порядке иллюстрации рассмотрим некоторые примеры атак, использующих соответствующие угрозы уязвимостей. Например, атака на повышение привилегий [51] связана с возможностью выполнения на компьютере созданного интерактивным пользователем файла, но уже с системными правами. Естественно, что в штатном режиме функционирования (без возникновения соответствующих условий) — эта штатная возможность не может рассматриваться в качестве технологической уязвимости — какой-либо ошибкой, либо некорректностью реализации защиты, однако, при выявлении (при условии) соответствующих ошибок программирования в системных средствах (и соответствующих компонентах системы [51]), эта возможность уже может рассматриваться в качестве угрозы технологической уязвимости.

Другой пример. В современных системах отсутствует возможность задания различных прав доступа к объектам для различных процессов, запускаемых из под одной и той же учетной записи — все запускаемые из под одной учетной записи процессы наследуют права доступа этой учетной записи. Опять же, в штатном режиме функционирования (без возникновения соответствующих условий) — эта штатная возможность не может рассматриваться в качестве технологической уязвимости — в качестве какой-либо ошибки, либо некорректности реализации защиты, однако, при выявлении (при условии) соответствующих ошибок программирования в данном случае, уже в приложениях [35], эта возможность уже может рассматриваться в качестве угрозы технологической уязвимости.

Подобных примеров можно привести много, однако общим для них является то, что некое свойство системы, либо приложения, не являющееся технологической уязвимостью при условии штатного функционирования системы и приложений, становится подобной уязвимостью при возникновении некоторых условий, в том числе, при выявлении (при условии) соответствующих ошибок программирования в системных средствах и в приложениях. Крайне важным при этом является то, что, как видим, подобные технологические уязвимости могут рассматриваться в качестве недостатков реализации (неполного, либо некорректного) разграничительной

С учетом сказанного введем следующие определения.

**Определение.** Под угрозой безусловной технологической уязвимости будем понимать угрозу технологической уязвимости, присущей в системе без возникновения каких-либо дополнительных условий.

Именно с целью предотвращения возникновения угроз безусловных технологических уязвимостей при создании защищенных информационных систем и должны формироваться требования к корректности реализации методов защиты информации (требования к построению безопасной системы).

**Определение.** Под угрозой условной технологической уязвимости будем понимать угрозу технологической уязвимости, которая возникает в системе лишь при возникновении каких-либо дополнительных условий, без которых соответствующая штатная возможность системы не несет в себе угрозу безопасности.

Отметим, что к условным технологическим уязвимостям (не являющимся уязвимостями без возникновения соответствующих условий — возникновения угроз уязвимостей реализации) могут быть отнесены широко используемые на практике штатные возможности современных приложений, позволяющие расширять их функционал с использованием макросов, скриптов и т.д. Угроза данных уязвимостей связана с возможностью нарушения соответствующего приложения вредоносным кодом — а это уже угроза уязвимости реализации.

**Определение.** Под угрозой реализации будем понимать ошибки реализации в используемых в информационной системе средствах, хотя возможны уязвимости системы и приложений — угрозу, при которой создается (возникает) угроза условной технологической уязвимости.
С учетом всего сказанного можем в общем случае сформулировать задачу защиты информации от несанкционированного доступа.

**Определение.** Технологическая задача защиты информации от несанкционированного доступа в общем случае предполагает выявление и нивелирование системой угроз безусловных и условных технологических уязвимостей.

Обратим внимание на это ключевое определение, следующее из введенной выше классификации угроз уязвимостей. По сути, это формулировка в общем виде задачи разработки систем защиты информации от несанкционированного доступа в части защиты от актуальных угроз атак, определяющая их назначение, очерчивающая круг решаемых ими функциональных задач защиты.

**Определение.** Задача защиты информации от несанкционированного доступа в общем виде включает в себя функциональную и технологическую задачи реализации разграничительной политики доступа субъектов к объектам.

Как отмечали, безусловная технологическая уязвимость может создаваться недостаточностью набора реализуемых методов защиты, не предоставляющей возможности в требуемом объеме (в соответствии с воспринимаемыми в организаторе целями защиты информации) реализовать права или привилегии разграничения доступа к защищаемой информации (режимы обработки информации субъектами доступа), и/или некорректностью реализации отдельных методов защиты, создающих условия обхода требуемых прав или правил разграничения доступа к защищаемой информации, как следствие, угрозы атаки с целью реализации несанкционированного доступа.

Следовательно, это определяет, что задача реализации защиты информационной системы — задача построения системы защиты от несанкционированного доступа, предполагает формирование и использование информации о защищаемой системе. В предложном общем случае задача реализации включает в себя выявление и нивелирование системой угроз безусловных и условных технологических уязвимостей.

В приведенной выше постановке задача построения системы защиты от несанкционированного доступа принципиально расширина тем, что она предполагает выявление и нивелирование системой угроз как безусловных, так и условных технологических уязвимостей, что, в свою очередь, предполагает выявление угроз уязвимостей реализации. Задача выявления и нивелирования угроз условных технологических уязвимостей принципиально иная собственной в своей постановке задача, решаемая при...
Глава 1. Задачи и основополагающие методы защиты информации от НСД

Говоря же об актуальности угроз атак, оценка которой необходима при проектировании системы защиты информации от несанкционированного доступа, необходимо понимать, что количественная мера актуальности применена к угрозам условных технологических уязвимостей, где она определяется вероятностью возникновения в системе угрозы уязвимости реализации соответствующей условной технологической уязвимости. Какая-либо оценка актуальности угрозы безусловной технологической уязвимости не имеет никакого смысла, поскольку наличие в системе подобной технологической уязвимости позволяет говорить о незащищенности информационной системы.

Естественным будет предположить, что решение задачи защиты информации от несанкционированного доступа в приведенной постановке, в том числе, с целью защиты от актуальных угроз атак, приведет к существенному пересмотру существующих подходов к построению методов контроля доступа к защищаемым ресурсам (объектам), соответствующих к необходимости формирования дополнительных требований к корректности их реализации (к построению безопасной системы).

Замечание. Угроза атаки на информационную систему в общем случае создается совокупностью угроз уязвимостей – угроз безусловных технологических уязвимостей, либо угроз условных технологических уязвимостей в совокупности с соответствующими угрозами уязвимостей реализации, наличие (выявление) которых в системе необходимо злоумышленнику для осуществления данной атаки. В результате этого может быть сформулирована и решаться задача проектирования системы защиты информации от несанкционированного доступа, в том числе, с целью определения оптимального набора задач защиты по нивелированию актуальных угроз безусловных технологических уязвимостей (методов защиты), которые должны решаться при построении защищенной информационной системы [27,50].

Однако рассмотрение задач и методов формального проектирования систем защиты информационных систем выходит за рамки данной работы, в которой мы сконцентрировались на вопросах именно построения систем защиты информации от несанкционированного доступа в современных условиях, на исследовании возможностей использования данных методов для защиты от актуальных угроз атак. Приведем здесь лишь один пример, иллюстрирующий необходимость решения подобных задач проектирования, основанных на математическом моделировании угроз информационной безопасности.

---

Контроль доступа к компьютерным ресурсам

построении систем защиты от несанкционированного доступа, поскольку решается она не с целью реализации прав или правил ограничения доступа к защищаемой информации, а с целью защиты от актуальных угроз атак, создаваемых в системе при выявлении соответствующих угроз уязвимостей реализации (реализации условных технологических уязвимостей).

Все сказанное выше позволяет определить следующий общий подход к построению системы защиты информации от несанкционированного доступа в части реализации защиты от актуальных угроз атак:

- Проведение анализа архитектурных особенностей системных и программных средств, используемых в защищаемой информационной системе (применительно к каждому режиму обработки информации), с целью выявления безусловных и потенциально возможных условных технологических уязвимостей. Потенциально возможные условные технологические уязвимости – это свойства системы и приложений, которые потенциально (при определенных условиях – в данном случае не важно, при каких условиях) могут быть использованы злоумышленником для реализации атаки на защищаемую систему. Разработка методов и средств контроля доступа, направленных на нивелирование выявленных технологических уязвимостей;

- Проведение анализа выявленных угроз уязвимостей реализации, использованных при реализации успешных атак несанкционированного доступа (c этой целью может быть использована соответствующая статистика, непрерывно ведущаяся в отношении выявляемых угроз уязвимостей, например, приведенная в [68]), с целью определения условных технологических уязвимостей, которые были в данном случае (при условии выявленных уязвимостей реализации) использованы злоумышленником для осуществления успешной атаки на информационную систему. Разработка методов и средств контроля доступа, направленных на нивелирование выявленных условных технологических уязвимостей.

Замечание. Проведение анализа выявленных угроз уязвимостей реализации, использованных при реализации успешных атак несанкционированного доступа, с целью определения условных технологических уязвимостей, которые необходимо нивелировать системой защиты информации от несанкционированного доступа, должно осуществляться разработчиками систем защиты информации непрерывно, с соответствующими, при необходимостями, доработками систем защиты информации.
Контроль доступа к компьютерным ресурсам

Отметим, что на практике сегодня уровень актуальности угрозы атаки, как следствие, и актуальности угрозы соответствующей условной технологической уязвимости, оценивается экспертным путем.

В качестве примера количественной оценки актуальности угрозы уязвимости рассмотрим угрозу атаки на повышение привилегий. Для данного исследования она интересна тем, что экспертная оценка ее актуальности при проектировании системы защиты совсем не очевидна. Подобная угроза атаки предполагает внедрение на компьютер вредоносной программы, что можно рассматривать как безусловную технологическую уязвимость, использование выявленной уязвимости реализации (выявленной программной ошибки) в компоненте (программе) ядра ОС, запущенного с системными правами, для исполнения внедренного на компьютер в процессе работы исполняемого вредоносного файла, что можно рассматривать уже в качестве условной технологической уязвимости системы (возможно при выявлении соответствующей ошибки в системном средстве), с системными правами. Если возможность загрузки на компьютер исполненного файла и возможность исполнения созданного в процессе работы системы файла, как отмечали, можно отнести к уязвимостям технологического характера (они всегда присутствуют в незащищенной системе), то интересующую нас угрозу уязвимости реализации – уязвимости, возникающей в результате выявления программных ошибок в компонентах (программах) ядра ОС – можно промоделировать с использованием соответствующей статистики [68].

Чтобы попытаться дать экспертную оценку актуальности угрозы данной атаки, можно, например, воспользоваться исследованиями, представленными в [75], результаты которых приведены на рис. 1.7 и на рис. 1.8.

Рассмотрим статистику, представленную на рис. 1.7 и на рис. 1.8, невольно задаем вопрос: как дать экспертную (хотя бы качественную, но говоря уже о количественной) оценку актуальности угрозы рассматриваемой атаки, исходя из подобной статистики, а уж тем более, как ее обосновать? На сколько адекватной будет подобная оценка, полученная экспертным путем?

Используем математическую модель исследуемой угрозы уязвимости реализации [50], для задания входных параметров модели обратимся к соответствующей статистике, приведенной в [68].

Уязвимостей, позволяющих осуществить атаку на повышение привилегий (Elevation of Privilege), за 2014 год было выявлено достаточно много. Однако уязвимости уровня ядра ОС сложны в практическом использовании для реализации соответствующей атаки — требуют разработки соответствующего эксплойта, уязвимости, для которых были созданы и использованы эксплойты [68]: CVE-2014-6324, CVE-2014-4113, CVE-2014-0318 (с учетом этого примем для расчетов значение интенсивности выявления угрозы интересующей нас уязвимости — 3 в год).

Исходя из того, что среднее время устранения выявленных уязвимостей в ОС Windows составляет порядка месяца [71], примем значение интенсивности устранения выявленной уязвимости — 12 в год. Используя соответствующую математическую модель [50], получаем количественную оценку актуальности угрозы рассматриваемой уязвимости реализации — вероятность готовности информационной системы к безопасной эксплуатации в отношении рассматриваемой угрозы атаки: \( P_\theta = 0.78 \), используя которую, можно утверждать, что доля времени, в течение которого в системе реализна угроза уязвимости реализации, позволяющая реализовать атаку на повышение привилегий, составляет 0.22. Таким же значением будет характеризоваться и уровень актуальности угрозы атаки на повышение привилегий, поскольку для остальных угроз уязвимостей — угроз технологических уязвимостей.
вимостей (которые реальны – присутствуют в системе), создающих рассматривающую угрозу атаки, имеем \( P_{утр} = 0 \).

Для практической оценки актуальности угрозы уязвимости более наглядно и на практике целесообразно использование иной характеристики безопасности – а именно: среднего времени наработки на отказ безопасности, либо между отказами безопасности. При этом мы рассматриваем систему с отказами и восстановлениями безопасности, то есть уязвимости выявляются и устраняются.

Используя данную характеристику безопасности, можем рассчитать, что в среднем рассматриваемая угроза уязвимости, а соответственно и угроза атаки на повышение привилегий, будет реальной каждые три месяца, причем нарушитель сможет ею воспользоваться на протяжении месяца.

Исходя из сказанного, применительно к рассматриваемому примеру, можно утверждать, что без использования каких-либо средств защиты, инвазирующих соответствующую угрозу технологической уязвимости, при условии соответствующей заинтересованности нарушителя в осуществлении несанкционированного доступа к обрабатываемой в системе информации (в предположении, что он реализует первую же реальную угрозу данной уязвимости), в среднем, соответствующая успешная атака будет осуществлена в течение 3 месяцев эксплуатации системы. А с учетом того, что речь идет о возможности запуска в системе произвольной вредоносной программы с системными правами, то эта атака может быть осуществлена нарушителем с любой целью (нарушение конфиденциальности, целостности и доступности информации).

Вот и вся реальная количественная оценка актуальности угрозы уязвимости, полученная которой не потребовало проведения каких-либо экспертных оценок. Имея подобную оценку уже не сложно делать объективный вывод о том, стоит ли инвазировать соответствующую угрозу технологической уязвимости системой защиты, актуальна ли она.

Приведенный пример достаточно показателен еще и для двух позиций. Во-первых, он иллюстрирует актуальность сформулированной и решаемой в работе задачи защиты, которую предлагается решать системами защиты информации от несанкционированного доступа (мы ведь на примере осознанно рассмотрели отнюдь не самую актуальную угрозу атаки, а полученный результат говорит сам за себя, судите сами, без решения соответствующей задачи защиты, в среднем через 2 месяца, только в отношении данной угрозы атаки, информационная система в среднем в течение месяца будет полностью незащищенной), во-вторых, он иллюстрирует возможность решения задачи защиты, расширением разграничительной политики доступа различными способами, как следствие, видим присутствие оптимизационной задачи. В частности, для решения рассматриваемой задачи можно предложить установку на компьютер вредоносной программы, можно разрешить системному пользователю использование только санкционированно установленных в системе программ (обеспечение замкнутости программной среды), можно запретить системному пользователю исполнение всех файлов, созданных в процессе работы интерактивными пользователями. Все эти меры защиты мы рассмотрим далее.

Несколько обобщенно (напрямую не связаны с решением задачи проектирования системы защиты информации) находятся угрозы уязвимости администрирования системы защиты. Кроме очевидного требования к необходимости решения задачи разработки методов контроля доступа к защищаемым ресурсам (объектам), направленных на упрощение задачи администрирования, что позволит снизить подобную угрозу, стоит еще и задача формирования требований к корректности назначенных прав доступа. Когда речь заходит о настройке системы защиты, реализующей метод дискрипционного контроля доступа, говорит о возможности избирательного задания прав доступа субъектов к объектам администратором. Но решение возникает вопрос, а существуют ли какие-либо ограничения (требования) – собственно правила задания правил контроля доступа? Для метода мандатного контроля доступа эти правила формализованы, но корректности ли они в общем случае, как это оценивать, если нет требований к корректности задания правил контроля доступа?

Подобная же многократная в этой главе, очередь круг основных вопросов, которые нами далее будут исследоваться в работе:

- разработка требований к построению безопасной информационной системы в части реализации методов контроля доступа;
- разработка требований к построению безопасной информационной системы в части назначения правил контроля доступа;
- разработка методов контроля доступа, применительно к реализации ролевой и сессионной моделей контроля доступа;
- разработка методов и моделей контроля доступа, реализация которых направлена на упрощение задачи администрирования реализующих их систем защиты информации от несанкционированного доступа;
- выявление актуальных угроз уязвимостей реализации, создающих угрозы условий технологических уязвимостей, исследование возможностей их инвазирования методами контроля доступа;
1.7. Основные результаты и выводы

1. Рассмотрены базовые методы контроля доступа к защищаемым ресурсам (к объектам доступа) — дисcretionary и мандатный, использование которых составляет основу защиты информации от несанкционированного доступа, определены их достоинства и недостатки, сформулированы базовые требования к реализации. Сделан вывод о целесообразности разработки новых методов контроля доступа, направленных на упрощение задачи администрирования реализующих их систем защиты информации.

2. Задачи реализации разграничительной политики доступа субъектов к объектам разделены на функциональную, решаемую с целью формирования (при необходимости, изоляции) режимов обработки информации субъектами доступа в информационной системе, и технологическую, решаемую для защиты от актуальных угроз атак. Сделан вывод о том, что, ввиду общем случае различия угроз атак для различных режимов обработки информации и различия требований к безопасности различных режимов обработки информации, защита от актуальных угроз атак должна решаться не применимо к информационной системе в целом, а отдельно в отношении каждого сформированного режима обработки информации.

3. Рассмотрены ролевая и сессионная модели контроля доступа, используемые для решения функциональной задачи защиты информации от несанкционированного доступа — задачи санктирования (при необходимости, изоляции) защищенных режимов обработки информации осуществляемых с различными целями. Сформулированы базовые требования к реализации ролевой и сессионной моделей контроля доступа.

4. Введена классификация угроз уязвимостей, которые подразделены на технологические - безусловные и условные, и угрозы уязвимостей реализации (реализации условных технологических уязвимостей). Задача защиты от актуальных угроз атак в общем виде, при этом, состоит в нивелировании в комплексе угроз безусловных и условных технологических уязвимостей, за счет реализации соответствующей разграничительной политики доступа. Применительно к защите от угроз условных технологических уязвимостей задача состоит в расширении разграничительной политики доступа, позволяющего предотвратить несанкционированный доступ к информации и при условии выявления соответствующих уязвимостей реализации. Как следствие, задача построения систем защиты информации от актуальных угроз атак в общем виде в том числе, предполагает выявление угроз технологических уязвимостей, создающих актуальными угрозами уязвимостей реализации, и их нивелирование методами контроля доступа к ресурсам (к объектам).

5. Сформулированы задачи исследования и очередной круг вопросов, которые должны решаться с целью реализации эффективной защиты информации, применительно к предложенной постановке задачи защиты информации от несанкционированного доступа в общем виде. Задача эта состоит, как в формировании (на необходимости, изоляции) защищенных режимов обработки информации, так и в нейтрализации безусловных и условных технологических уязвимостей.
ГЛАВА 2. СУБЪЕКТЫ И ОБЪЕКТЫ ДОСУПА.

2.1. Субъекты доступа

2.1.1. Субъект доступа пользователь (учетная запись)

2.1.1.1. Задача идентификации и аутентификации пользователей при доступе субъектов к объектам

Когда разговор заходит о реализации разграничительной политики доступа, мы сразу предполагаем, что разговор идет о контроле доступа пользователя (учетной записи) к объектам, в частности, к файловым объектам. Однако возникает вопрос — учетная запись, используемая администратором при задании разграничительной политики и учетная запись, которую получает диспетчер из запроса доступа, всегда совпадают?

Попытаемся ответить на этот вопрос [29]. Например, в широко распространенном OC семейства Windows для идентификации субъектов, выполняющих в системе различные действия, используются идентификаторы защиты (security identifiers, SID). SID имеются у пользователей, локальных и доменных групп локальных компьютеров, доменов и членов доменов. Все работающие в системе процессы и потоки выполняются в контексте защиты того пользователя, от имени которого они так или иначе были запущены, а для идентификации контекста защиты процесса или потока используется объект, называемый маркером доступа (access token). В процессе регистрации в системе создается начальный маркер, представляющий пользователя, который входит в систему; и сопоставляет его с процессом оболочки, применяемой для регистрации пользователя.

Маркер может быть основным (идентифицирует контекст защиты процесса) или олицетворяющим (применяется для временного заимствования потоком другого контекста защиты — обычно другого пользователя). Олицетворение (impersonation) — средство, часто используемое в модели защиты Windows, предоставляющее возможность отдельному потоку выполняться в контексте защиты отличном от контекста защиты процесса, т.е. действовать от лица другого пользователя. Олицетворение, в частности, исполь-
вления несанкционированного доступа к конфиденциальным данным в обход реализованной ракетной политики доступа.

Подобная возможность присутствует и в механизме идентификации Unix-подобных систем. В Unix существуют два типа идентификаторов пользователя: реальный и эффективный. Реальным идентификатором пользователя данного процесса является идентификатор пользователя, запустившего процесс. Эффективный идентификатор служит для определения прав доступа процесса к ресурсам системы (в первую очередь к ресурсам файловой системы). Обычно реальный и эффективный идентификаторы совпадают, т.е. процесс имеет в системе те же права, что и пользователь, запустивший его. Однако существует возможность задать процессу более широкие права, чем права пользователя, путем установки бита SUID, когда эффективному идентификатору присваивается значение идентификатора владельца выполняемого файла (например, пользователя root). Таким образом, процесс выполняется от лица пользователя - владельца исполняемого файла, причем запущение прав происходит прозрачно (не требуется дополнительной аутентификации) для пользователя, запустившего процесс.

Как видим, с помощью сервисов олицетворения, вполне легальных сервисов, предоставляемых современными ОС, можно получать право другого пользователя, с последующим обращением с его правами к файловым объектам в обход разграничительной политики доступа к ресурсам.

Из всего сказанного можем сделать вывод о том, что в общем случае, учетная запись, для которой назначаются права доступа, и учетная запись, от лица которой происходит обращение к ресурсам, это не одно и то же. Как следствие, возможна атака, направленная на смену учетной записи пользователя при обращении к объекту, как правило, реализуемая с целью повышения привилегий пользователя. Соответственно, актуальной становится задача аутентификации пользователя при запросах доступа к ресурсам (объектам).

Замечание. Напомним, что, как отмечали, под аутентификацией пользователя в общем случае понимается проверка подлинности идентификации пользователя.

Для понимания того, насколько актуальна задача контроля смены идентификатора при доступе к ресурсам рассмотрим статистику уязвимостей в компонентах ОС, опубликованную, например, в [76], см. рис. 2.2.
препятствовать доступу к зашифрованным ресурсам неидентифицированных пользователей и пользователей, подлинность идентификации которых при аутентификации не подтвердилась».

2.1.1.2. Метод и модель дискретного контроля смены имени пользователей при доступе субъектов к объектам

Как отмечалось, при реализации контроля доступа субъектов к объектам, мы имеем две сущности, характеризующие пользователя (субъект доступа) — «Первичное» и «Эффективное» имя пользователя. Первичное имя пользователя — это имя (учетная запись), от лица которого (с правами которого) запускается процесс. Запуск процесса должен контролироваться средством защиты (его компонентом — диспетчером доступа, перехватывающим и анализирующим все запросы к файловым объектам), а первичное имя для каждого запускаемого процесса — запоминается диспетчером.

Эффективное имя пользователя — это имя пользователя (учетная запись), от лица которого процесс (точнее, генерируемый им поток) непосредственно и обращается к объекту. Данное имя диспетчер доступа определяет из запроса доступа субъекта к объекту. Как видим, диспетчер доступа при любом запросе доступа к объектам имеет для анализа обе сущности (первичное и эффективное имя пользователя), как следствие, он может, во-первых, проникнуть в аутентификацию и аутентификацию пользователя при доступе к объектам (сравнивая первичное и эффективное имя, при совпадении — аутентификацию будет проведена, т.к. корректность запроса подтверждена), во-вторых, реализовать разграничительную политику доступа в части контроля смены имени пользователя при доступе субъектов к объектам.

Определение. Под контролем смены имени пользователей при доступе субъектов к объектам будем понимать контроль и разграничение прав на смену учетных записей пользователей при доступе к объектам.

Построим соответствующую модель контроля доступа [29]. Будем считать, что множество \( C = \{ C_1, ..., C_i \} \) — линейно упорядоченное множество пользователей (учетных записей), зарегистрированных в системе.

Метод дискретного контроля доступа предполагает реализацию разграничительной политики на основе матрицы контроля смены имен пользователей при доступе субъектов к объектам, позволяющей максимально точно задать соответствующие правила:
В любой момент времени система определяется своим текущим состоянием Q = (C, C, I); а I [C, C] - ячейка матрицы, содержит право смены имен пользователей: 41 - смена имени разрешена, 40 - запрещена. Условия строками матрицы обозначать первичные, а столбцами - эффективные имена пользователей. Будем обозначать C1(1)C1 разрешение изменения первичного имени пользователя C1 при доступе к объекту на эффективное имя Ci, i = 1, ..., k = 1, ..., l, i ≠ j. Запрет доступа будем обозначать следующим образом: C1(0)Ci. Следует заметить, что доступ к объекту осуществляется без смены имен пользователя в случае C1(1)C1, i = 1, ..., l.

Будем называть матрицу контроля смены имен пользователей при доступе субъектов к объектам, I канонической, если для нее справедливо: C1(1)C1, i = 1, ..., l, C1(0)Ci, i = 1, ..., k = 1, ..., l, i ≠ j. Это матрица, на главной диагонали которой расположены соответствующие разрешения 41, остальные ячейки матрицы содержат 40. При реализации в системе канонической матрицы, смены имен пользователей при доступе субъектов к объектам не предусмотрено.

Теперь определим с помощью правил (о требованиях к правилам) расширения канонической матрицы.

Для упрощения условий, что ключевая задача защиты, при контроле смены имен пользователей, является защита от повышения привилегий пользователя при доступе к объектам. Будем считать, что пользователи из множества C = (C1, ..., C1), зарегистрированные в системе, упорядочены следующим образом: чем меньше порядковый номер пользователя, тем выше его привилегия по доступу к объектам.

**Лемма 2.1.** Разрешение изменения первичного имени пользователя на эффективное имя при доступе к объекту: C1(1)C1, i ≠ j, i = 1, ..., h, j = 1, ..., l, корректно, при выполнении следующего условия: C1 < C1, C1.

**Доказательство.** При выполнении условия: C1 < C1 разрешается изменение первичного имени пользователя при доступе к объекту C1(1)C1 только на имя менее привилегированного пользователя. Лемма доказана.

Корректное правило изменения первичного имени пользователя при доступе субъектов к объектам описывается матрицей I:

\[
I = \begin{bmatrix}
C_1 & 1 & \cdots & 0 \\
C_2 & 0 & \cdots & 0 \\
C_{i-1} & 0 & \cdots & 0 \\
C_i & 0 & \cdots & 1 \\
\end{bmatrix}
\]

Обратим внимание на то, что запрещает смену идентификатора пользователя при обращении к объектам процесс, т.е. именно процесс в данном случае следует рассматривать в качестве субъекта доступа при задании правил смены учетных записей пользователя при доступе к ресурсам. В противном случае, один и те же правила контроля изменения первичного имени пользователя при доступе субъектов к объектам будут распространяться на все процессы, что не позволит корректно решить рассматриваемую задачу защиты (об этом далее).

Теперь определим с помощью правила привилегий и сформируем требование к корректности олицетворения в общем виде, применительно к матрице доступа M [52].

\[
M = \begin{bmatrix}
O_1 & O_2 & O_i \\
C_1 & r & w & d & 0 \\
C_2 & r & w & d & 0 \\
C_{i-1} & 0 & 0 & r \\
C_i & 0 & w & r & d \\
\end{bmatrix}
\]

При ее рассмотрении видим, что потенциальная возможность утечки права доступа R состоит в возможности олицетворения субъекта доступа C1 с субъектом доступа C1 (получения субъектом C1 прав доступа к объектам, заданным в матрице доступа M для субъекта доступа C1).

Рассмотрим, применительно к матрице M, два правила олицетворения C1(1)C1 и C1(1)C1. Как видим, правило C1(1)C1 не приводит к повышению привилегий, т.к. при таком олицетворении субъект доступа C1, лишь потерял права доступа к объекту O1 (w и d). Правило же C1(1)C1 наделит субъект доступа C1 дополнительными правами доступа (w и d) к объекту O1, т.е. его привилегии в отношении доступа к объекту O1 в этом случае будут повы-
шены, в результате чего имеет место утечка прав доступа \((\alpha, d)\) субъекта \(S_2\) к объекту \(O_r\). Как следствие, данное правило олицетворения некорректно.

**Определение:** В общем случае под повышением привилегий будем понимать приобретение в результате некоторой последовательности действий субъектом в отношении какого-либо объекта некоторого права доступа из множества \(R\) (т.е. действие, связанное с утечкой права \(R\)).

С учетом сказанного, переформулируем требование к корректности олицетворения, представленное Леммой 2.1. Данное требование сформулируем применительно к корректности расширения канонической матрицы олицетворения, определяемой следующим образом: \(C(1)\), \(C(0)\), \(C_j\) и т.д.

**Лемма 2.2.** Расширение канонической матрицы олицетворения правилом: \(C(1)\), \(C_j\), \(C_{ji}\) допустимо (корректно) в том случае, если множество прав доступа субъекта \(S_j\) к каждому объекту из множества \(O = \{O_1, \ldots, O_l\}\) совпадает или является подмножеством прав доступа субъекта \(S_j\) к каждому соответствующему объекту из множества \(O = \{O_1, \ldots, O_l\}\).

**Доказательство.** Только при выполнении данного условия, введение в каноническую матрицу олицетворения правила олицетворения: \(C(1)\), \(C_j\), \(C_{ji}\) не приведет к повышению привилегий субъекта \(S_j\), как следствие, в этом случае отсутствует утечка прав доступа из множества \(R\).

Таким образом, для реализации контроля и разграничения прав на смену идентификатора пользователя — контроля олицетворения \(C(1)\), \(C_j\), при доступе субъектов к объектам, — в разграничительной политике доступа к объектам должны присутствовать две сущности, идентифицирующие пользователя, как субъекта доступа — исходный (или первичный) идентификатор пользователя — идентификатор пользователя, запустившего процесс, и эффективный (или целевой) идентификатор пользователя. Правила же доступа субъекта к объекту должны предусматривать задание того, при каком олицетворении в эффективном пользователю этому пользователю какие права доступа разрешаются/запрещаются, поскольку за-рашивает доступ к объекту именно эффективный пользователь.

Теперь остановимся на следующем крайне важном требовании. В общем случае (попробуйте олицетворение субъекта доступа) недостаточно для решения рассматриваемой задачи контроля доступа, что связано с невозможностью корректного решения ключевой задачи защиты — предотвращения доступа к объектам в случае олицетворения интерактивного пользователя с системным. Это обусловлено двумя причинами, во-первых, все процессы, запускаемые одним и тем же пользователем, наследуют права доступа к объектам, заданные для этого пользователя, во-вторых, некоторые системные процессы штатно олицетворяют себя с правами интерактивного пользователя и наоборот. Приемлемо сказанное на примере. Системный процесс svchost при входе пользователя в систему сначала олицетворяет себя с правами этого пользователя, затем обратно с системными правами. Если запретить доступ к объектам при обращении олицетворения интерактивного пользователя в системном, то ввод в систему этому интерактивному пользователю будет не возможно.

Разрешение данного противоречия возможно в случае включения в схему контроля доступа еще одной сущности, идентифицирующей субъект доступа — сущности процесса, соответственно идентифицируемой полнопутевым именем исполняемого файла процесса. В этом случае для различных процессов, запускаемых одним и тем же пользователем, могут назначаться различные права доступа к объектам. Наследование права доступа к объектам процессом пользователя будет только в том случае, если они явно не указаны для процесса (процесс является более точным описателем субъекта доступа). Возвращаясь к рассмотренному ранее примеру, всем процессам можно запретить доступ к объектам при олицетворении интерактивного пользователя с системными правами, а для субъекта с более точным описателем — процесс в субъекте доступа задан исполняемым файлом процесса svchost, —разрешить. Система будет корректно работать.

2.1.1.3. Метод и модель мандатного контроля смены имени пользователей при доступе к файловым объектам

Напомним, что под мандатным контролем доступа понимается способ работы запросов диспетчером доступа, основанный на формальном сравнении диспетчеров в соответствии с заданным правилом меток безопасности (мандатов), называемых субъектом и объектом доступа. Метки безопасности, как правило, являются элементами линейно упорядоченного множества \(M = \{M_1, \ldots, M_p\}\) и служат для формализованного представления таких-либо свойств субъектов и объектов.

Разграничение доступа диспетчером реализуется на основе задаваемого правила, определяющего отношение линейного порядка на множестве \(M\), где для любой пары элементов из множества \(M\), задано один из типов отношения: \(\succ, \prec, =\) (на практике реализуется выбор подмножества \(M\), изоморфного конечному подмножеству натуральных чисел — такой выбор делает естественным арифметическое сравнение меток безопасности). Пра-
влиято сравнения меток также назначается из каких-либо свойств субъектов и объектов, применимо к решаемой задаче защиты информации.

Сформулируем соответствующее требование применительно к методу манда-тального контроля доступа к объектам [52]. В данном случае разрешаемое
правило олицетворения может быть представлено следующим образом: $C(M_1) \vdash C(M_2)$. Обратимся к матрице $M_{nip}$:

$$
M_{nip} = \begin{bmatrix}
C(M_1) & O_1(M_2) & \ldots & O_n(M_2) \\
C(M_2) & C(M_2) & \ldots & C(M_2) \\
C(M_1) & C(M_1) & \ldots & C(M_1)
\end{bmatrix}
$$

Видим, что любое подобное олицетворение $C(M_1) \vdash C(M_2)$ при $j \neq i$ приво-
дит к утрате прав пользователя из множества $R$. Как следствие, можем сфор-
мулировать следующее требование к заданию правил олицетворения для
метода мандатного контроля доступа.

Должны контролироваться и разграничаться права на смену идентифи-
катора пользователя — контролироваться олицетворение $C(M_1) \vdash C(M_2)$,
при доступе субъектов к объектам. Олицетворение, расширяющее канони-
ческую матрицу олицетворения: $C(M_1) \vdash C(M_2)$, $j \neq i$, не допустимо.

Корректное правило олицетворения субъектов доступа для метода мандат-
ного контроля доступа задается канонической (диагональной) матрицей $I$, имеющей следующий вид:

$$
I = \begin{bmatrix}
C(M_1) & C(M_2) & \ldots & C(M_2) \\
C(M_1) & C(M_1) & \ldots & C(M_1) \\
C(M_1) & C(M_1) & \ldots & C(M_1)
\end{bmatrix}
$$

С учетом сказанного, корректное правило олицетворения, как и правила до-
ступа для метода мандатного контроля доступа, может быть формализова-
но — задаваться на «умолчании», следующим образом: метка безопасности пользователя, запустившего процесс (первичного) и метка безопасности пользователя, от лица которого (с правами которого) этот процесс запра-
Теперь построим математическую модель системы массового обслуживания. Будем рассматривать следующую гипотетическую ситуацию — любая обнаруженная уязвимость градуало направляется на обслуживание (устранение). Никакой очереди неустранимых уязвимостей не образуется, т.е. будем рассматривать систему с бесконечным числом обслуживаемых приборов.

Замечание. Данное допущение позволяет утверждать, что модель будет описываться гипотетически идеальной (недостигаемой) ситуацией, т.е. расчетные значения будут не хуже реальных (оценяем верхнюю границу). Дело в том, что на практике ситуация одновременного (полностью, либо частичного) исправления разработчиком нескольких уязвимостей встречается крайне редко.

В данных предположениях, расчетная формула вероятности того, что в системе находится ровно \( n \) требований (или присутствует \( n \) неустранимых уязвимостей) выглядит следующим образом [9]:

\[
P_n = \frac{\lambda^n}{n!} \left( \frac{\lambda}{\mu} \right)^n
\]

С учетом же того, что:

\[
\sum_{n=0}^{\infty} P_n = 1
\]

т.е. в каком-то состоянии система всегда должна находиться, можем определять интересующий нас параметр — вероятность того, что в системе отсутствуют требования: \( n = 0 \), т.е. отсутствуют неустранимые уязвимости, или вероятность того, что система находится в безопасном состоянии:

\[
P_n = \frac{1}{1 + \sum_{n=1}^{\infty} \frac{1}{n!} \left( \frac{\lambda}{\mu} \right)^n}
\]

Итак, модель мы построили, теперь с ее помощью проведем исследование. Соответствующую статистику возьмем из исследований [71,85], в которых утверждается, что быстрее всех исправление выпускала компания Microsoft, которой требовалось в среднем, по разным оценкам, 21—29 дней для закрытия уязвимости, обнаруженных в количестве 39 за полгода. Сразу оговоримся, что нас точные данные особо здесь не интересуют, важен порядок!

Используем нашу модель и произведем соответствующие расчеты. Результаты исследований представлены на рис. 2.4 и рис. 2.5.
Отметим, что пунктирной линией на рис. 2.4 и рис. 2.5 отображено состояние, при котором система с равной вероятностью безопасна, либо нет. Если же продлить кривую на рис. 2.5 (а это лишь более 20 обнаруженных уязвимостей за год), то картина выглядит совсем печально. Заметим, что, следуя нашим расчетам, о какой-либо безопасности системы говорить не приходится уже при единицах обнаруживаемых подобных уязвимостях в год. На самом деле, дела обстоят куда хуже. Уязвимости ведь систематически обнаруживаются не только в ОС, а на самом деле, не столько в ОС. Пример отчета по уязвимостям [75] проиллюстрирован на рис. 2.6.

Изменения в уязвимостях, приведенные на рис. 2.5, могут быть связаны с изменением темпа выявления уязвимостей и уровнем их обнаружения в системе. В рамках исследуемого периода эти изменения происходят в соответствии с показанными на рисунках трендами.

Как видим, уязвимости ОС, которые, следя рис. 2.5, сами по себе приводят к катастрофическим последствиям, составляют лишь единицы процентов, подавляющую часть уязвимостей содержат приложения. К сожалению, данные уязвимости могут месяцами не исправляться, причем, как следует из рис. 2.6, в большом количестве. О какой же безопасности системы при этом может идти речь?

Получаем тупиковую ситуацию. С одной стороны, разрабатываются механизмы и средства защиты. С другой стороны, уязвимостей из года в год становится все больше, устраивается разработчиком они все дольше, что обусловливается, в том числе, ростом сложности ПО. Это свидетельствует на нет достижения в развитии систем безопасности. Конечно, рекомендации, данная в [77]: «Для того чтобы предотвратить использование злоумышленниками уязвимостей, организации должны сосредоточить свое внимание на сокращении периода времени между выявлением уязвимости и установкой «программной заплатки» (или «патча» - patch), устраняющей эту ошибку», бесспорно, правильно. Но она полностью нивелируется другим утверждением этого же отчета: «К концу 2010 года почти половина всех уязвимостей (44%) не имела таких официальных «заплаток» от разработчика соответствующего ПО».
Глава 2. Субъекты и объекты доступа

Расширение возможностей защиты

Итак, что же мы имеем в итоге, к чему сводится постановка задачи повышения уровня безопасности системы в рассматриваемой ситуации:

1. Угрозу несет в себе процесс (системный процесс, либо приложение). Любая система не может быть охвачена заинтересованными пользователями. В ней утверждение уязвимости реализации (для известной и неисправной уязвимости, ее значение можно изменить, равной 1). Значения этой вероятности можно определить, исходя из существующей статистики угроз.

2. Поскольку не представляется возможным в общем случае предсказать, что за уязвимость будет обнаружена в процессе (приложении), с целью повышения уровня безопасности системы, следует минимизировать последствия атак на уязвимость, представляющую собой "черный ящик", т.е. следует предполагать, что в результате атаки на
Процесс, процесс может быть несанкционированно наделен совершенно произвольным функционалом.

3. А вот цели атак, в отличие от способов, конечны и вполне предсказуемые. Как следствие, именно в противодействии целям атак и может состоять защита в общем случае.

Таким образом, чтобы вести разговор о минимизации последствий от атак на уязвимости системных процессов и приложений, нужно разобраться с целями подобных атак. Например, классификация целей наиболее распространенных сетевых атак рассматриваемой группы приведена на рис. 2.7 [83].

Рис. 2.7. Классификация целей сетевых атак

Как следует из рис. 2.7, к основным целям атак на уязвимости приложений можно отнести: установку и запуск вредоносной программы (28%), кражу информации (всего же можно отнести кражу денег, в совокупности, 37%), деинформацию, либо несанкционированную модификацию информации (19%). Те, подобные последствия атак следует минимизировать в первую очередь. Крайне важно отметить, что эти атаки, связанные с несанкционированным запуском и чтением информации, которые, практически, равно-вероятны.

При постановке задачи защиты в данном случае, в предположении, что угроза несанкционированного доступа потенциально несет в себе собственный процесс, объектом доступа должен только процессы (системный или приложение), идентифицируемый полномочием именем своего исполняемого файла. Как следствие, защита должна рассматриваться контролем доступа (разграничением прав доступа) процессов (приложений) к ресурсам (объектам).

Следствие. Существует «Процесс» следует рассматривать как содержательное, самостоятельного субъекта доступа, для которого должны устанавливаться права доступа к объектам.

Следствие. Как самостоятельный субъект доступа, процесс требует идентификации и аутентификации при доступе к объектам.

2.1.2. Вероятностные методы и модели контроля доступа

Сделав вывод о необходимости для реализации в современных условиях эффективной защиты, основанной на применении разграничительной политики доступа, рассмотрения процесса в качестве самостоятельного субъекта доступа, исследуем, за счет чего в данном случае может быть достигнут соответствующий эффект, в результате чего сформулируем требования к реализации разграничительной политики доступа для субъекта процесс. С этой целью построим и проанализируем соответствующие модели контроля доступа.

Контроль доступа с учетом оценки вероятности наделения субъекта доступа вредоносными свойствами в результате выявления уязвимостей в приложении

Сначала построим модель, учитывающую уровень безопасности субъекта доступа, характеризующий вероятность его наделения вредоносными свойствами в результате выявления уязвимостей в приложении [35].

Будем считать, что множество $C = \{C_1, ..., C_i\}$ — линейно упорядоченное множество субъектов доступа — процессов, идентифицируемых полномочными именами их исполняемых файлов, $R = \{R_1, ..., R_i\}$ конечное множество прав доступа (чтение (ч), запись (з), удаление (д), исполнение (и), отсутствие прав доступа (о)) субъекта $C_i$ к объекту $O_j$ из множества $O = \{O_1, ..., O_j\}$. Разрешение (санкционированное) право доступа субъекта $C_i$ к объекту $O_j$ будем обозначать: $C_i(R)O_j$ (соответственно, $C_i(R)O_j$ разрешено чтение, $C_i(O)O_j$ — запрет и т.д., $C_i(O)O_j$ — доступ не разрешен). Разграничительная политика доступа субъектов к объектам описывается матрицей...
цей доступа $M$, где $M(C, O)$ — ячейка матрицы, содержит набор прав доступа субъекта из множества $C = \{C_1, ..., C_l\}$ к объекту из множества $O = \{O_1, ..., O_l\}$. В любой момент времени система описывается своим текущим состоянием $Q = (Q, O, M)$.

Замечание. Далее везде, кроме случаев, где это будет оговорено особо, будем считать, что для множеств $C = \{C_1, ..., C_l\}$ и $O = \{O_1, ..., O_l\}$ соответственно линейно упорядоченных множеств субъектов и объектов доступа, число субъектов и объектов доступа совпадает.

Аксиома. При назначении разграниченной политики «по умолчанию» должны быть установлены права доступа: $C_i(r), r, d)O_i, i = 1, ..., l$. Данное правило обобщает задание диагональной "канонической" [111] матрицы доступа, характеризуемой условием: $C_i(r, r, d)O_i; C_i(O)O_i, i \neq j, i = 1, ..., l, j = 1, ..., l$.

$$
M = \begin{bmatrix}
O_1 & O_2 & \cdots & O_l \\
C_1 & r, w, d & 0 & \cdots & 0 \\
C_2 & 0 & r, w, d & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{l-1} & 0 & 0 & \cdots & r, w, d \\
C_l & 0 & 0 & \cdots & r, w, d
\end{bmatrix}
$$

Следствие. Любая разграниченная политика на практике реализуется добавлением в каноническую матрицу прав доступа $C_i(r)O_i, i \neq j, i = 1, ..., l, j = 1, ..., l$.

Введем следующие обозначения. Пусть множество $P = \{P_1, ..., P_l\}$ — линейно упорядоченное множество вероятностей обнаружения уязвимостей процессов (приложений) из множества субъектов доступа — процессов $C = \{C_1, ..., C_l\}$. С учетом того, что эти уязвимости приводят к несанкционированной записи (включая сюда и удаление) $w$, либо к несанкционированному чтению $r$, будем считать, что вероятность несанкционированной записи субъектом $C_i$ в объект $O_j$ (если ему разрешены запись и чтение) составляет $P_i(r, w)$, несанкционированного чтения — $P_i(r)$, при этом $P_i(r) = P_i(r, w) + P_i(r)$, $i = 1, ..., l$ [9,13].

В общем случае запись в один и тот же объект $O_j$ может быть разрешена нескольким субъектам (для чтения аналогично), в пределе все субъекты $C = \{C_1, ..., C_l\}$, поэтому здесь следует говорить о суммарных вероятностях несанкционированной записи в объект $O_j$: $P_{sc}(r, w)$, соответственно, чтения: из объекта $P_{sc}(r)$, определяемых следующим образом:

$$
P_{sc}(r) = 1 - \prod_{i=1}^l (1 - P_i(r)),
$$

где:

$P_{sc}(r) = P_i(r)$, если $C_i(r \neq 0)O_j, j \neq i$;

$P_{sc}(r) = P_i(r)$, если $j = i$;

$P_{sc}(r) = 0$, если $C_i(r = 0)O_j$.

соответственно:

$$
P_{sc}(w) = 1 - \prod_{i=1}^l (1 - P_i(w)),
$$

где:

$P_{sc}(w) = P_i(w)$, если $C_i(w \neq 0)O_j, j \neq i$;

$P_{sc}(w) = P_i(w)$, если $j = i$;

$P_{sc}(w) = 0$, если $C_i(w = 0)O_j$.

Вероятностная модель контроля доступа к объектам, определяющая вероятности несанкционированного доступа (записи или чтения) к объектам, за счет уязвимостей субъектов из множества $C = \{C_1, ..., C_l\}$, описывается матрицей доступа, ячейка которой $M(C, O)$ содержит суммарные вероятности несанкционированного чтения $P_{sc}(r)$ и записи $P_{sc}(w)$ объекта, осуществляющего за счет уязвимости субъектов $C = \{C_1, ..., C_l\}$.

Каноническая матрица доступа вероятностной модели контроля доступа, $M_{pk}$ имеет следующий вид:

$$
M_{pk} = \begin{bmatrix}
O_1 & O_2 & \cdots & O_l \\
C_1 & P_1(r), P_1(w) & 0 & \cdots & 0 \\
C_2 & 0 & P_2(r), P_2(w) & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{l-1} & 0 & 0 & \cdots & 0 \\
C_l & 0 & 0 & \cdots & P_{pk}(r, P_{pk}(w))
\end{bmatrix}
$$

Следуя (2.1) и (2.2), в матрице доступа $M_{pk}$: $P_{sc}(r) = P_i(r), P_{sc}(w) = P_i(w)$.

Лемма 2.3. При реализации разграниченной политики, основанной на полной изоморфии обработки процессами (приложениями) информации, что описывается канонической матрицей доступа $M_{pk}$ вероятностной модели, последствия от атак на уязвимости процессов (приложений) минимальны и состоят в: $P_i(r, P_i(w), i = 1, ..., l$. 

---

24

---

195
Доказательство. Для канонической матрицы доступа $M_c$ имеет место:
$P_c(r) = P_r(r), P_c(w) = P_r(w)$. Значения $P_c(r)$ и $P_r(r)$ нельзя изменить без запрета соответствующих прав доступа $C(r)O_i$, $C(w)O_i$ (т.е. без запрета необходимых прав доступа к соответствующим объектам) не представляется возможным. Включение в матрицу любого дополнительного права доступа $C(r)O_i$, соответственно, $C(w)O_i$, следуя (2.1) и (2.2), приводит к увеличению значений $P_c(r)$ и $P_r(r)$. Лемма доказана.

Следствие. Реализация разграничительной политики, основанной на полной изоляции обработки процессами (приложениями) информации, что описывается канонической матрицей доступа $M_c$, обеспечивает максимальный уровень безопасности.

Ввиду того, что уровень безопасности, характеризуемый величиной последствий от атак на уязвимости приложений, может быть понижен в результате добавления права $r$ или $w$ к канонической матрице доступа, далее сформулируем требование к безопасному расширению канонической матрицы доступа правами $r$ или $w$.

Важным для последующих наших рассуждений является то, что в общем случае значения вероятностей уязвимости различных процессов (приложений) на практике, по различным причинам, существенно различаются.

Лемма 2.4. Безопасным является включение в каноническую матрицу доступа $C(r)O_i$, $C(w)O_i$, соответственно, $C(r)O_i$, $i \neq j$, при выполнении условия: $P_j(w) << P_j(w)$, соответственно, $P_j(r) << P_j(r)$.

Доказательство. Обратимся к (2.1), (2.2). При включении в каноническую матрицу доступа права доступа $C(r)O_i$, соответственно, $C(w)O_i$, $i \neq j$, получаем:

$P_c(r) = 1 - (1 - P_r(r))(1 - P_r(r))$
$P_c(w) = 1 - (1 - P_r(w))(1 - P_r(w)),$

при выполнении условия: $P_j(w) << P_j(w)$, соответственно, $P_j(r) << P_j(r)$:

$P_c(r) = P_r(r)$
$P_c(w) = P_r(w).$

Лемма доказана.

Лемма 2.5. Максимально опасным является включение в каноническую матрицу доступа (присутствует право: $C(r)O_i$, $C(w)O_i$, соответственно, $C(r)O_i$, $i \neq j$, при выполнении условия: $P_j(w) >> P_j(w),$ соответственно, $P_j(r) >> P_j(r)$.

Доказательство. Обратимся к (2.1), (2.2). При включении в каноническую матрицу доступа права доступа $C(r)O_i$, соответственно, $C(w)O_i$, $i \neq j$, получаем:

$P_c(r) = 1 - (1 - P_r(r))(1 - P_r(r))$
$P_c(w) = 1 - (1 - P_r(w))(1 - P_r(w)),$

при выполнении условия: $P_j(w) >> P_j(w)$, соответственно, $P_j(r) >> P_j(r)$:

$P_c(r) = P_r(r)$
$P_c(w) = P_r(w).$

Лемма доказана.

Следствие. В современных условиях нельзя говорить о построении безопасной системы при отсутствии возможности реализации изолированной обработки информации процессами (приложениями), т.е. без реализации контроля доступа процессов (приложений) к объектам.

Следствие. Поскольку в современных ОС запускаемый процесс (приложение) наследует права доступа (маркер безопасности) запускающего его пользователя (все процессы (приложения), запущенные одним пользователем, имеют одинаковые права доступа к объектам), в безопасной системе в обязательном порядке должен быть реализован метод контроля доступа процессов (приложений) к объектам, в котором именно процесс (приложение) выступает в качестве самостоятельного субъекта доступа, для которого разграничиваются права доступа. В противном случае уровень безопасности системы будет определяться максимально опасным включением в каноническую матрицу доступа прав доступа, уровнем безопасности подобной системы на порядки ниже, чем уровень безопасности системы, описываемой канонической матрицей доступа.
Теорема. Максимально безопасной относительно атак на уязвимость приложе-
ний, является разграничительная политика доступа к файловым объ-
ектам, реализующая каноническую матрицу доступа вероятностной
модели контроля доступа. Допустимым ее расширением (в канони-
ческой матрице присутствует право: $C(r,w)O_i, i=1,...,l$) является допол-
нение матрицы правами доступа $C(r)O_j$. Соответственно, $C(w)O_i, i\neq j,$
$i=1,...,l, j=1,...,l$, при выполнении условия: $P(w) < P(r)$, соответственно,
$P(r) < P(w)$. Последовательность от атак на уязвимость процессов (приложе-
ний) в этом случае минимальны и составляют: $P(r), P(w), i=1,...,l$.

Доказательство. Теорема доказывается доказательством Лемм 2.3 - 2.5.

Теперь рассмотрим следующее условие. Пусть на множестве $P = \{P_1, ..., P_l\}$ —
линейно упорядоченное множество вероятностей обнаружения уязвимо-
стей процессов (приложений) из множества субъектов доступа — процесс-
сов $C = \{C_1, ..., C_l\}$, задано следующее отношение: $P_1 < P_2 < ... < P_l$. Вос-
пользуемся Леммами 2.3-2.5 и получим матрицу доступа $M_{se}$, описываю-
щую максимально безопасную разграничительную политику доступа субъ-
ектов к объектам, относительно атак на уязвимости приложений:

$$
\begin{bmatrix}
O_1 & O_2 & ... & O_l \\
C_1 & | & | & | & r, w, d & | & | & | & r, w \\
C_2 & 0 & | & | & r, w, d & | & | & | & r, w \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C_{l-1} & 0 & 0 & | & r, w & | & | & | & r, w, d \\
C_l & 0 & 0 & | & r, w & | & | & | & r, w, d \\
\end{bmatrix}
$$

Из матрицы $M_{se}$ видим (присутствует соответствующая иерархия отноше-
ний) возможность реализации в данном случае манипулятивного метода контро-
ля доступа процессов (приложений) к объектам.

Будем считать, что множество $C = \{C_1, ..., C_l\}$ — линейно упорядоченное моно-
жество субъектов доступа — процессов, идентифицируемыми полноравне-
выми именами их исполняемых файлов, $O = \{O_1, ..., O_l\}$ — множество объ-
ектов доступа, а множество $P = \{P_1, ..., P_l\}$ — линейно упорядоченное моно-
жество вероятностей обнаружения уязвимостей процессов (приложе-
ний) из множества субъектов доступа — процессов $C = \{C_1, ..., C_l\}$, причем
$P_1 < P_2 < ... < P_l$. Метки безопасности являются элементами линейно
упорядоченного множества $M = \{M_1, ..., M_l\}$. Метки безопасности, назна-
чаются субъектам (процессам, приложениям) и служат для формализованно-
го представления отношения значений вероятностей обнаружения их уяз-
вимостей на множестве $P = \{P_1, ..., P_l\}$. Условимся считать, что чем меньше
значение $P$, тем меньше значение метки $M$, т.e.: $M_1 < M_2 < M_3 < ... < M_l$.

Пусть:

- $M_s$ — метка безопасности субъекта (группы субъектов) доступа;
- $M_o$ — метка безопасности объекта (группы объектов) доступа.

Правила контроля доступа:

1. Субъект $C$ имеет доступ к объекту $O$ в режиме $R = \{w, r\}$ в случае, если
выполняется условие: $M_s > M_o$.
2. Субъект $C$ имеет доступ к объекту $O$ в режиме $R = \{w, r, d\}$, в случае,
если выполняется условие: $M_s = M_o$.
3. В случае $M_s < M_o$, доступ субъекта $C$ к объекту $O$ запрещен.

При запросе доступа к объекту, диспетчер доступа по заданному правилу
сравнивает метку безопасности субъекта, запрошенного доступа, $M_s$, с мет-
кой безопасности объекта, к которому запрошены доступ, $M_o$, в результате
чего, в соответствии с правилом контроля доступа, диспетчер либо разре-
шает запрошенный субъектом доступ, либо отклоняет в нем.

Контроль доступа с учетом оценки вероятности наделения субъекта
доступа вредоносными свойствами в результате внедрения
на компьютер и прочтения субъектом вредоносного
командного файла

Теперь построим модель, учитывающую уровень безопасности субъекта до-
ступа, характеризующий вероятность его наделения вредоносными свой-
ствами в результате внедрения на компьютер командного файла, при прочтении которого процессом выполняются присутствующие в этом файле команды (макросы, апплеты, скрипты и т.д.) [25].

С этой целью введем в вероятностную модель контроля доступахарактери-
стику $P_i$, но уже интерпретируемую иным образом: множество $P = \{P_1, ..., P_l\}$ —
линейно упорядоченное множество вероятностей внедрения вредоносно-
го файла (наделяемого приложением при его прочтении вредоносными свой-
ствами) в объекты из множества объектов доступа $O = \{O_1, ..., O_l\}$. Опять же
введем следующее упорядочивание характеристики $P_i, P_j < P_i, P_j < ... < P_i$. Отметим, что, естественно, в качестве субъектов доступа рассматриваем
приложения, которые могут быть наделены вредоносными свойствами в результае прочтения вредоносного файла. 
Естественно, нас будет интересовать решение задачи защиты информации в комплексе — защита от нарушения конфиденциальности информации, обеспечение ее целостности и доступности. В результате принятия вредоносного файла зараженное приложение (субъект доступа) может осуществить несанкционированный доступ на чтение к объектам, к которым ему разрешено его чтение, соответственно на запись/удаление к тем объектам, к которым ему разрешены соответствующие права доступа. Будем считать, что вероятность несанкционированной записи субъектом (для простоты, включая удаление) \( C_i \) в объект \( O_j \) (куда ему разрешены запись и удаление) составляет \( P_i(\omega) \), несанкционированного чтения — \( P_i(\tau) \).

С учетом введенных обозначений матрицу вероятностей внедрения вредоносных файлов, которые могут наделять субъектов вредоносными свойствами, можно представить следующим образом, \( M_w \):

\[
M_w = \begin{bmatrix}
O_1 & O_2 & \cdots & O_t \\
C_1 & P_1 & P_2 & \cdots & P_t \\
C_2 & P_1 & P_2 & \cdots & P_t \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{t-1} & P_1 & P_2 & \cdots & P_t \\
C_t & P_1 & P_2 & \cdots & P_t \\
\end{bmatrix}
\]

для которой матрица вероятностей несанкционированного доступа \( M_{sk} \) имеет вид, аналогичный приведенному ранее (естественно, что с другим определением характеристики вероятности):

\[
M_{sk} = \begin{bmatrix}
O_1 & O_2 & \cdots & O_t \\
C_1 & P_1(\omega), P_2(\tau) & 0 & \cdots & 0 \\
C_2 & 0 & P_2(\tau), P_2(\omega) & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{t-1} & 0 & 0 & \cdots & P_t(\tau), P_t(\omega) \\
C_t & 0 & 0 & \cdots & P_t(\tau), P_t(\omega) \\
\end{bmatrix}
\]

Расширим диагональную матрицу \( M_{sk} \) следующим правилом: \( C_i(\omega)O_j \). В результате этого при заданном условии категорирования: \( P_i << P_j \), матрица вероятностей несанкционированного доступа \( M_{sk} \) не изменяется, поскольку при добавлении подобного правила не изменяется вероятность наделения субъекта вредоносными свойствами (файл субъектом не читается, а может лишь быть записан (при условии \( P_i << P_j \) (величиной \( P_i \) можно пренебречь))

вероятность несанкционированного доступа к объекту \( O_j \) определяется вероятностью прочтения вредоносного файла субъектом \( C_j \).

Совсем иной результат получим, если расширим диагональную матрицу \( M_{sk} \) правилом: \( C_i(\tau)O_j \). Исходя из этого, что в этих предположениях вероятность \( P_i \) можно пренебречь, получим следующую матрицу \( M_{sk} \):

\[
M_{sk} = \begin{bmatrix}
O_1 & O_2 & \cdots & O_t \\
C_1 & P_1(\tau), P_1(\omega) & 0 & \cdots & 0 \\
C_2 & 0 & P_2(\tau), P_2(\omega) & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{t-1} & 0 & 0 & \cdots & P_t(\tau), P_t(\omega) \\
C_t & 0 & 0 & \cdots & P_t(\tau), P_t(\omega) \\
\end{bmatrix}
\]

Как видим, вероятность несанкционированного доступа к объекту \( O_j \) при этом возрастает.

Теперь при тех же условиях поочередно расширим диагональную матрицу \( M_{sk} \) следующими правилами: \( C_i(\omega)O_j \), \( C_i(\tau)O_j \). Несложно показать, что при условии \( P_i << P_j \) величиной \( P_i \) можно пренебречь, исходная матрица \( M_{sk} \) не изменяется.

Таким образом, в результате проведенного исследования можно выделить только один тип включаемых в диагональную матрицу прав доступа, при водящих к возрастанию вероятности несанкционированного доступа, который в общем случае может быть сформулирован следующим образом: \( C_i(\tau)O_j \) при наличии: \( P_i << P_j \) и, т.к. в противном случае повышается вероятность несанкционированного доступа к объекту \( O_j \).

С учетом полученного результата сформулируем требование к назначению прав доступа, применительно к построению безопасной системы в части защиты от атак на приложения, предполагающих наделение их вредоносными свойствами в результате внедрения на компьютер вредоносного файла, при прочтении которого приложением выполняются присутствующие в этом файле команды (макросы, апплеты, скрипты и т.д.).

При категорировании субъектов доступа по уровням безопасности, определяемым вероятностями их наделения вредоносными свойствами в результате внедрения на компьютер вредоносного файла, при прочтении которого процессом выполняются присутствующие в этом файле команды, не должны разрешаться права доступа, определяемые следующим условием: \( C_i(\tau)O_j \) при условии: \( P_i << P_j \).
2.1.2.3. Непротиворечивое правило мандатного контроля доступа

Как ранее отмечали, мандаты — метки безопасности должны присваиваться субъектам доступа — пользователям, т.е. данная схема управления доступа не предполагает использование в качестве субъекта доступа субъекты процесса, поскольку процессы в общем случае не могут категорироваться на уровень конфиденциальности обрабатываемой ими информации.

Однако обратимся к вопросу практического использования мандатного контроля доступа — к реализации модели сессионного контроля доступа, о которой говорили в предыдущей главе, при этом увидим, что в реальной системе требуется обеспечивать не только конфиденциальность обрабатываемой информации (защиту от ее киберкриминала), а также направлены реализация мандатного контроля доступа, но и ее целостность и целостность обрабатываемой информации, что иллюстрируется рис. 2.7, из которого видно, что для защиты данных от киберкриминала информации сопоставима с вероятностью атак на нарушение ее целостности (дезинформации).

В реальной системе для защиты конфиденциальной информации требуется не только разграничение прав доступа к файловым объектам, задача защиты на этом шире — состоит в формировании режимов обработки информации субъектами доступа (это вопросы мы рассмотрели применительно к модели сессионного контроля доступа), и состоит в том, чтобы различные изолированных между собой режимов обработки информации различных категорий информации различных субъектами доступа. А режимы эти принципиально различаются по обеспечиваемому уровню защиты информации на всех этапах ее обработки: ввод/вывод, хранение, передача и т.д. (само собой с этой целью они и создаются).

В таком (в общем виде) интерпретации мандатного контроля доступа, от изменения категории обрабатываемой информации, формализуемая для файловых объектов моделью Белла-ЛаПауэллы, состоит в предотвращении подмены информации более высокого уровня конфиденциальности в режим обработки информации более низкого уровня информации. Например, применительно к задаче защиты от сетевых атак (того которых проиллюстрированы на рис. 2.7) режим обработки открытой информации зашифрована минимально, поскольку не представляется возможным шифрование "канала" (необходимо обращаться к сетевым объектам, которыми не будет поддерживать такого шифрования), по той же причине невозможно разграничение прав доступа к сетевым объектам и т.д.

При обработке конфиденциальной информации требования и возможности реализации защиты принципиально меняются — обмен данными реализуется с корпоративными объектами (должен предотвращаться доступ к иным объектам), для защиты передаваемой информации может использоваться VPN и т.д.

Из сказанного может сделать достаточно важный вывод [42,46].

Таким образом, характеристика "уровень конфиденциальности обрабатываемой информации", задаваемая в модели Белла-ЛаПауэллы меткой безопасности, а характеристика "вероятность нарушения приложений вредоносными свойствами", используемая в вероятностной модели контроля доступа, зависимости — задаваемыми метками безопасности субъектов и объектов доступа, определяющими режим обработки категоризированной информации, тем самым категорируется и характеристика "вероятность нарушения приложений вредоносными свойствами", причем имеем прямую зависимость — чем меньше значение метки безопасности, тем меньше значение вероятности нарушения приложений вредоносными свойствами.

Построим и проанализируем соответствующие модели мандатного вероятностного контроля доступа [42,46].

Сначала рассмотрим контроль доступа с учетом оценки вероятности нарушения субъекта доступа вредоносными свойствами в результате выявления уязвимостей в приложении. Будем считать, что в системе существует $l$ категорий безопасности субъектов доступа, для которых выполняется следующее условие: $P_1 << P_2 << ... << P_l$ на множестве $P = \{P_1, ..., P_l\}$, т.е. чем выше уровень безопасности субъекта доступа, тем меньше его порядковый номер в линейно полномочном упорядочении множества субъектов $C = \{C_1, ..., C_l\}$, для которых реализуется разграничительная политика доступа к объектам на множестве, $O = \{O_1, ..., O_l\}$, и тем меньше значение метки безопасности $M_{i_1}, i = 1, ..., l$ им присваивается, т.е. $M_1 < M_2 < M_3 < ... < M_l$. 

Вывод. Чем выше категория конфиденциальности информации, т.е., тем меньше значение метки безопасности присваивается обрабатывающей в соответствующем режиме обработки информации объектам, тем ниже вероятность осуществления успешной атаки на используемые в данном режиме процессы (приложения). Причем вероятность нарушения их вредоносными свойствами, как следствие, и вероятность успешной атаки, на одно и то же приложение, используемое в различных режимах обработки информации, будет принципиально различаться.
С учетом сформулированного выше требования к построению безопасной системы в отношении дискреционного вероятностного контроля доступа, сформулируем правила мандатного контроля доступа, реализации которых позволит построить безопасную систему, при этом описать все используемые следующим образом:

- $M_C$ - метка безопасности субъекта (группы субъектов) доступа;
- $M_O$ - метка безопасности объекта (группы объектов) доступа.

Данные правила могут быть сформулированы следующим образом:

1. Субъект $C$ имеет доступ к объекту $O$ в режиме "чтения", "записи", "удаление" в случае, если выполняется условие: $M_C \leq M_O$.
2. Если $M_C > M_O$, субъект $C$ не имеет доступа к объекту $O$.
3. Этим правилам соответствуют следующая матрица доступа $M_{\text{упр}}$:

\[
\begin{bmatrix}
O_1(M_C) & O_2(M_C) & \cdots & O_I(M_C) \\
C_1(M_C) & r, w, d & r, w, d & \cdots & r, w, d \\
C_2(M_C) & 0 & r, w, d & \cdots & r, w, d \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_I(M_C) & 0 & 0 & \cdots & r, w, d \\
C_{I-1}(M_{I-1}) & 0 & 0 & \cdots & r, w, d \\
C_I(M_{I-1}) & 0 & 0 & \cdots & r, w, d \\
C_I(M_{I-2}) & 0 & 0 & \cdots & r, w, d \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_1(M_{I-2}) & r, w, d & r, w, d & \cdots & r, w, d \\
\end{bmatrix}
\]

Ранее мы говорили о том, что мандатный контроль доступа используется для обработки категории информации, а формально задаваемые в общем случае правила контроля доступа (в предположении, что в системе $l$ категорий конфиденциальности обрабатываемой информации, причем, чем выше полномочия субъекта и уровень конфиденциальности объекта, тем меньше их порядковый номер в линейно полноценно упорядоченных множествах субъектов и объектов доступа) могут быть представлены матрицей доступа $M_{\text{пр}}$:

\[
\begin{bmatrix}
O_1(M_C) & O_2(M_C) & \cdots & O_I(M_C) \\
C_1(M_C) & r, w, d & r & \cdots & r \\
C_2(M_C) & w & r, w, d & \cdots & r \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_I(M_{I-1}) & w & w & \cdots & r \\
C_I(M_0) & w & w & \cdots & r, w, d \\
\end{bmatrix}
\]

С учетом проведенного исследования сформулируем следующее требование к построению безопасной системы.

При реализации мандатного метода контроля доступа с целью формирования различных режимов обработки информации различных уровней конфиденциальности, не должно использоваться дополнительное формализованное правило: субъект $C$ имеет доступ к объекту $O$ в режиме "записи" в случае, если выполняется условие: $M_C > M_O$. Выполнение данного требования обеспечивает одновременную защиту и от понижения категории обрабатываемой информации, и от атак, направленных на нарушение конфиденциальности, доступности и целостности обрабатываемых информации, предполагающих нарушение приложений вредоносными свойствами в результате выявления в них уязвимостей.

Теперь рассмотрим контроль доступа с учетом оценки вероятности нарушения субъекта доступа вредоносными свойствами в результате внедрения на компьютер и прочтения им вредоносного командного файла, для чего построим соответствующую вероятностную модель мандатного контроля доступа [46].

Будем считать, что в системе существует $l$ категорий безопасности субъектов доступа, для которых выполняется следующее условие: $P_1 << P_2 << \cdots << P_l$ на множестве $P = \{P_1, \ldots, P_l\}$, т.е. чем выше уровень безопасности субъекта доступа, тем меньше его порядковый номер в линейно полноценно упорядоченном множестве субъектов $C = \{C_1, \ldots, C_l\}$, для которых реали-
иеза разграничивать политике доступа к объектам на множестве,
\( O = \{O_1, ..., O_i\} \), и тем меньше значение метки безопасности \( M_i, \), \( i = 1, ..., l \) присваивается, т.е.: \( M_l < M_{l-1} < ... < M_l \).

С учетом сформулированного выше требования к построению безопасной системы в отношении дискретного контроля доступа, сформулируем правила мандатного контроля доступа, реализация которых позволит построить безопасную систему:

1. Субъект \( C \) имеет доступ к объекту \( O \) в режиме "Чтение", "Записи", "Удаления" в случае, если выполняется условие: \( M_l > M_{l-1} \).

2. При \( M_l > M_{l-1} \) субъект \( C \) имеет доступ к объекту \( O \) в режиме "Записи", "Удаления".

Этим правилам соответствует следующая матрица доступа \( M_{vip} \):

\[
M_{vip} = \begin{bmatrix}
C_1(M_1) & O_1(M_1) & \ldots & O_l(M_1) \\
C_2(M_2) & O_1(M_2) & \ldots & O_l(M_2) \\
\vdots & \vdots & \ddots & \vdots \\
C_{l-1}(M_{l-1}) & O_1(M_{l-1}) & \ldots & O_l(M_{l-1}) \\
C_l(M_l) & O_1(M_l) & \ldots & O_l(M_l)
\end{bmatrix}
\]

Исходя из тех же рассуждений, что были приведены выше, в отношении того, что чем выше уровень конфиденциальности обрабатываемой информации (менее метка безопасности), тем меньше вероятность, в данном случае, внедрения в систему в созданных для данного уровня конфиденциальности режиме обработки информации вредоносного командного файла, при прочтении которого субъект доступа (приложение) наслаждается вредоносными свойствами, можем провести корректное сравнение (опять же в отличии от сравнения правил доступа, задаваемых моделями Белла-Ла Падуля и Бива, основанных на различных принципах категорирования) правил доступа, формализующих моделью Белла-Ла Падуля и построенной моделью мандатного вест.нутного контроля доступа.

Замечание. В отличие от сравнения с моделью Бива сравнение модели Белла-Ла Падуля с данной моделью корректно, поскольку в них используется один и тот же категорирующий признак — категорирование по уровню конфиденциальности обрабатываемой информации.

С этой целью сравнив на непротиворечивость правил матрицы доступа \( M_{vip} \), приведенной ранее, и полученную здесь матрицу \( M_{vip} \). В результате

сравнения получаем следующую матрицу доступа \( M \), удовлетворяющую всем соответствующим требованиям к правилам доступа:

\[
M = \begin{bmatrix}
C_1(M_1) & O_1(M_1) & \ldots & O_l(M_1) \\
C_2(M_2) & O_1(M_2) & \ldots & O_l(M_2) \\
\vdots & \vdots & \ddots & \vdots \\
C_{l-1}(M_{l-1}) & O_1(M_{l-1}) & \ldots & O_l(M_{l-1}) \\
C_l(M_l) & O_1(M_l) & \ldots & O_l(M_l)
\end{bmatrix}
\]

С учетом проведенного исследования сформулируем следующее требование к построению безопасной системы.

При реализации мандатного метода контроля доступа с целью формирования различных режимов обработки информации различных уровней конфиденциальности, не должно использоваться дополнительное формализованное правило: субъект \( C \) имеет доступ к объекту \( O \) в режиме "Чтение" в случае, если выполняется условие: \( M_l < M_{l-1} \). Выполнение данного требования обеспечивает одновременную защиту и от понижения категории обрабатываемой информации, и от атак, направленных на нарушение конфиденциальности, доступности и целостности обрабатываемой информации, предполагающих нарушение приложений вредоносными свойствами в результате внедрения в компьютерное программный команда программ, при прочтении которого приложении выполняются присутствующие в этом файле команды (макросы, апилены, скрипты и т.д.),

А теперь получим корректные формализованные правила мандатного контроля доступа при формировании им различных режимов обработки информации различных уровней конфиденциальности с защитой от атак, направленных на нарушение конфиденциальности, доступности и целостности обрабатываемой информации в общем случае (при различных возможных способах нарушения приложений вредоносными свойствами).

С этой целью сравнив на непротиворечивость правил соответствующих матрицы доступа \( M \), построенных для альтернативных способов нарушения приложений вредоносными свойствами, в результате чего получаем матрицу доступа \( M \), удовлетворяющую всем сформулированным выше требованиям к правилам доступа для мандатного контроля доступа при формировании им различных режимов обработки информации различных уровней конфиденциальности:
С учетом полученных результатов сформулируем правила мандатного контроля доступа, реализация которых позволит построить безопасную систему в общем случае.

Данные правила могут быть сформулированы следующим образом:

1. Субъект $C$ имеет доступ к объекту $O$ в режиме "Чтения", "Записи", "Удаления" в случае, если выполняется условие: $M_r = M_o$.

2. При $M_r \neq M_o$ субъект $C$ не имеет доступа к объекту $O$.

Данные правила доступа, реализующие каноническую (диагональную) матрицу доступа, определены нами в [46], как "непротиворечивые правила мандатного контроля доступа". Ими предполагается неравнорешательная обработка в неравненных (присвоенных на основе категорирований информации по уровням конфиденциальности) меток безопасности, поскольку метки сравниваются исключительно на равенство/неравенство.

Отметим, что именно непротиворечивое правило мандатного контроля доступа имеет смысл использовать, о чем мы говорили в первой главе, при реализации метода сессионного контроля доступа [48], как наиболее полно удовлетворяюще требованию к построению безопасной системы.

2.1.2.4. Идентификация и аутентификация субъекта доступа «Процесс»

Ранее мы рассматривали постановку и решение задачи идентификации и аутентификации пользователя (учетной записи) при доступе к ресурсам. Теперь, что касается субъекта доступа «Процесс».

Мы должны знать, что обращаюсь к какому-нибудь файловому объекту, например, процессом текстового редактора Word, действительно это этот процесс, а не иной запущен под его именем, что этот процесс (его исполняемый файл) не модифицирован, и что реализует этот процесс задекларированные (описанные в документации) функции. По сути именно в этом и состоит задача корректной идентификации и аутентификации субъекта доступа «процесс».

Идентификатор процесса является полноупотребным и его исполнимого файла. Таким образом, для корректной идентификации субъекта доступа «Процесс» необходимо предотвратить любую возможность запуска процессов под иными именами, кроме тех, которые разрешены для исполнения, и предотвратить любую возможность модификации исполняемых файлов, полноупотребные имена которых разрешены для выполнения. В результате этих действий будет решена задача аутентификации субъекта доступа «Процесс» — обеспечена подлинность идентификации процесса при обращении к ресурсу [29].

Для решения рассматриваемой задачи может быть использован метод обеспечения замкнутости (локализации) программной среды [11], который будет нами рассмотрен далее.

2.1.2.5. Процессная модель контроля доступа

В первой главе были рассмотрены базовые модели контроля доступа, используемые сегодня для формирования режимов обработки информации субъектами доступа — модели ролевого и сессионного контроля доступа (для сессионного контроля доступа, в дополнение к формированию, в обязательном порядке должна решаться задача разделения режимов — сессий, обработки информации субъектами доступа). В качестве субъекта доступа в данных моделях выступает пользователь.

Процессная модель контроля доступа предполагает использование в качестве субъекта доступа сущности процесс.

Сущность «Процесс» может использоваться в качестве доступа как с целью кардинального расширения ролевой модели контроля доступа, так и в качестве основы реализации защиты от актуальных угроз атак.

В первом случае в рамках формирования режимов обработки информации появляется возможность не только задавать то, какие права доступа к объектам имеет роль (пользователь в данной роли), но и то какие при этом могут использоваться приложения (процессы) при доступе к объектам. Во втором случае реализуется процессная модель контроля доступа, которая основана на предоставлении процессу только необходимых ему прав доступа к ресурсам (к объектам). То есть в этом случае модель предполагает формирование режимов обработки информации процессами.
В этом случае, в первую очередь, практический интерес представляет частный случай – задача формирования режимов обработки информации процессами (частная процессная модель контроля доступа), основанного на реализации вероятностной модели контроля доступа, рассмотренной выше. Данный способ формирования режимов обработки информации процессами предполагает объединение в различные группы процессы, в том числе, системные, характеризуемые различными значениями иерархии их на- деления (по той или иной причине, что рассмотрено выше) вредоносными свойствами, с последующим формированием и с обязательным разделением режимов обработки информации различными группами процессов. В этом случае решается задача защиты от наиболее актуальных сегодня угроз атак, связанных с нарушением процессов вредоносными свойствами, в том числе, в результате обнаружения уязвимостей (ошибок) их реализации.

Интерес представляет одновременная реализация и ролевой, либо сессионной, и процессной моделей контроля доступа – задачи защиты информации от несанкционированного доступа должны решаться в комплексе (и формирование режимов обработки информации пользователями, и защиты от актуальных угроз атак, в данном случае на процессы). Без этого комплексирования соответствующих решений в системе защиты информации эффективной защиты современной информационной системы не обеспечить.

При реализации ролевой модели, основной на применении метода дис- креционного контроля доступа, субъект доступа должен определяться двумя сущностями – пользователь, процесс (разграничивается права того, какому пользователю, каким процессом, к каким объектам, какой тип доступа разрешается) [14,18]. С целью же возможностью контроля смены пользователя (счетной записи) сущность «пользователь» в разграничительной политике доступа должна быть расширена, в результате чего субъект доступа должен в разграничительной политике идентифицироваться тремя сущностями – первичный идентификатор пользователя, полноправное имя процесса, эффективный идентификатор пользователя [14,19], что позволяет задавать различные права доступа к объектам при различных вариантах смены имени пользователя при запросах доступа. При реализации сессионной модели, основанной на применении метода сессионного контроля доступа, должно реализовываться непротиворечивое правило мандатного контроля доступа.

2.1.3. Создание субъекта доступа в системе защиты. Техническое решение

Технические реализации методов контроля доступа здесь и далее, если это отдельно не оговаривается, будут рассматриваться на примере построения средств защиты «Комплексная система защиты «Панцирь-+ для OC Microsoft Windows» (для иллюстрации будем использовать интерфейсы, реализованные в данном средстве защиты) – для наглядной иллюстрации практической реализации и применения рассматриваемых нами принципов [23]. Они могут быть реализованы иным образом, но это не меняет сути описываемых методов и подходов.

Замечание. Комплексной данной СЗИ НСД мы назвали, исходя из тех соображений, что она позволяет решать задачи защиты информаций от несанкционированного доступа в комплексе, т.е. в постановке задачи защиты информации от несанкционированного доступа к общем виде, состоящей из типов и в формирование режимов обработки информации, и в защите от актуальных угроз атак. Необходимость комплексного решения данных задач защиты, как указывало выше, во многом обусловливает изглажденные далее и подходы к построению систем защиты информации.

Отметим, что не излагаемые в книге технические решения реализованы и апробированы, наиболее важные из них запатентованы авторами [15-22].

Замечание. Будем рассматривать реализацию соответствующих методов защиты информации от несанкционированного доступа на примере построения системы защиты информации для OC Microsoft Windows, поскольку эти решения реализованы и апробированы, что не означает, что данные методы в большинстве своем в той или иной мере не могут использоваться и применительно к иным OC.

Как было отмечено, с целью реализации комплексного подхода к защите субъекта доступа в разграничительной политике в общем случае определяются три сущности:

- первичный идентификатор пользователя,
- полноправное имя процесса,
- эффективный идентификатор пользователя.

Интерфейс задания субъекта доступа в средствах защиты приведен на рис. 2.8а. Отображаются в интерфейссе заданные в разграничительной политике субъекты доступа в виде, представленном на рис. 2.8б.

Пользователь может задавать конкретной учетной записью, а так же группой пользователей, см. рис. 2.8в), либо маской «Все». Важно, чтобы в средство защиты пользователь идентифицироваться не текстовой записью (тестовая интерпретация целесообразна в интерфейсе), а своим SID (о чем мы упоминали ранее). В этом случае, см. рис. 2.8в), как отдельный пользователь, так и группа пользователей, будут однозначно идентифицированы.
Контроль доступа к компьютерным ресурсам

ны при запросах доступа к ресурсам. Имя процесса, может задаваться либо полноточным именем его исполняемого файла, либо дзеткой (возможны использование переменных среды окружения). Например, дзеткой C:\ Program File\ покрываются все исполняемые файлы из данного каталога.

а) Интерфейс задания субъекта доступа

б) Задание субъекта «пользователь»

в) Отображение в интерфейсе заданных субъектов доступа с возможностью просмотра их SID

г) Отображение «весов» субъектов доступа в разграничителной политике

д) Меню изменения «веса» субъектов доступа в разграничителной политике

Рис. 2.8. Задание и отображение субъектов доступа

Примеры задания субъекта доступа в матрице доступа приведены в табл. 2.1.
Таблица 2.1. Примеры задания субъекта доступа

<table>
<thead>
<tr>
<th>Пример</th>
<th>Субъект доступа</th>
<th>Имя (идентификатор) процесса</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System System</td>
<td>C:\Windows\system32*</td>
</tr>
<tr>
<td>2</td>
<td>System System</td>
<td>%WinDir%\system32\ss.exe</td>
</tr>
<tr>
<td>3</td>
<td>User1 User1</td>
<td>C:\Program Files\Internet\Explorer\iexplore.exe</td>
</tr>
<tr>
<td>4</td>
<td>User1 System</td>
<td>*</td>
</tr>
<tr>
<td>5</td>
<td>User1 Administrator</td>
<td>*</td>
</tr>
</tbody>
</table>

Как видим из табл. 2.1, у администратора появляются весьма широкие возможности по настройке разграничительной политики доступа к ресурсам при данном способе задания субъекта доступа. Другое дело, как это применить и использовать на практике — это мы рассмотрим в последующих главах.

Замечание. Создание субъекта доступа в системе защиты информации, реализующей метод мандатного (сессионного) контроля доступа, где субъект — это пользователь, которому должна быть назначена метка безопасности, рассмотрим далее.

Реализованная универсальность задания в разграничительной политике доступа субъекта доступа приводит к тому, что в общем случае под одну и ту же маску может подпадать (покрываться этой маской) несколько реальных субъектов (с этой целью маски и используются).

Прояснутствуем сказанное. Пусть с запросом доступа к какому-либо файловому объекту обращается процесс с полным путем именем исполняемого файла E:\\Program Files\\Internet Explorer\\iexplore.exe. Разграничительная же политика задана (заданы разные правила) для следующих субъектов, идентифицируемых масками «*», «E:\\Program Files\\IE*», «*\\iexplore.exe», «*\\iexplore.exe». Всеми представленными масками покрывается рассматриваемый реальный субъект доступа (процесс, запрошивший доступ) E:\\Program Files\\Internet Explorer\\iexplore.exe. Возникает вопрос — какое правило (для какого субъекта) и, исходя из каких соображений, выбрать из разграничительной политики (из таблицы правил) при анализе корректности (непротиворечивости заданной разграничительной политики доступа) запроса доступа.

Рассмотрим, например, какая из двух масок «E:\\ProgramFiles\\*\\iexplore.exe» или «*\\iexplore.exe» более точно описывает реальный субъект (или объект) доступа E:\\Program Files\\Internet Explorer\\iexplore.exe? Естественно, что в общем случае — это маска «*\\iexplore.exe». Таким образом, можно задать правила, позволяющие системе защиты автоматически ранжировать между собою маски задаваемых в разграничительной политике субъектов (объектов) доступа по точности описания реального субъекта (объекта). Например, маска «E:\\ProgramFiles\\*\\iexplore.exe» точно описывает субъект (объект), чем маска «*\\iexplore.exe», а маска «*\\iexplore.exe» чем маска «*\\iexplore.exe».

Сказанное позволяет сформулировать следующее важное требование к реализации метода контроля доступа [14].

Выбор требуемого правила для анализа корректности запрашиваемого доступа следует осуществлять по наиболее точному описанию (описательно) в правиле записанной в самом высоком субъекта доступа (субъекта, запрашивающего доступ). Рассмотрим соответствующее запатентованное авторами технические решения, обеспечивающее корректное выполнение данного требования.

Созданные субъекты доступа автоматически ранжируются системой защиты по точности описания (по заданным в системе правилам ранжирования субъектов, например, точное указание в субъекте имени процесса приоритетнее, чем имена пользователя, имя файла в субъекте «Процесс приоритетнее, чем имя каталога в этом субъекте и тд.») и отображаются в интерфейсе в соответствующем порядке (либо сверху вниз, либо снизу вверх), см. рис. 2.8 г.

Однако далеко не во всех случаях можно требуемым образом при создании соответствующей разграничительной политики доступа автоматически ранжировать субъекты доступа. Естественно, что при решении некоторых задач защиты администратору потребуется изменить автоматически устанавливаемые системой ранг субъектов. Например, применительно для двух рассматриваемых ранее масок субъектов «E:\\ProgramFiles\\*\\iexplore.exe» или «*\\iexplore.exe» ему может быть важнее именно то, в какой папке (на каком диске) находится исполняемый файл приложения (субъект доступа), а не то, какое имя у исполняемого файла.

Как видим, при создании субъектов доступа администратору должна предоставляться возможность изменения задаваемого ему автоматически системой защиты веса.

В рассматриваемой технической реализации при необходимости администратор имеет возможность принудительно изменять автоматически заданный системой ранг («вес») субъекта доступа, см. рис. 2.8 д), при этом в упорядочении системой отображении созданных субъектов, субъект, которому администратором изменяется ранг, будет соответствующим образом смещаться в списке заданных субъектов вниз, либо вверх.
При создании субъектов доступа им назначаются имена, которые затем используются при задании правил доступа. Данная реализованная схема единичный может быть применима (при создании субъектов из единого интерфейса) и при реализации расширенной (субъектом доступа «Процесс») ролевой модели, и при реализации процессной модели контроля доступа.

Важно здесь отметить то, что каких либо ограничений по созданию субъектов доступа нет, в том числе, это могут быть как интерактивные, так и системные пользователи, как процессные приложения, так и системные процессы, в отношении любого субъекта доступа может быть реализована разграничительная политика.

2.1.4. Субъект доступа «Профиль»

Как ранее отмечалось, основной проблемой практического использования дискретного метода контроля доступа является сложность настройки системы защиты (задания разграничительной политики доступа) при ее запуске в эксплуатацию и при последующей эксплуатации. Здесь рассматриваем вопрос задания большого количества субъектов доступа и включение в разграничительную политику новых субъектов впоследствии (соответственно, при исключении субъектов).

Для упрощения задачи администрирования используется сущность «Профиль субъектов доступа» (в частном случае данная сущность может иметь физический смысл ролей пользователей, определяемых решаемыми пользователями задачами на защищаемом компьютере).

Определение. Под профилем субъектов доступа будем понимать дополнительную сущность, включаемую в разграничительную политику с целью объединения субъектов с одинаковыми правами доступа в одну группу с последующим назначением прав доступа к ресурсам для всей этой группы (профиля).

Следствие. При реализации разграничительной политики доступа к ресурсам с использованием сущности «Профиль», в качестве субъекта доступа используется сущность «Профиль».

При создании субъекта доступа он сразу же включается в профиль, см. рис. 2.8, либо же в существующий, либо может быть создан новый профиль. Отображение созданных профилей в интерфейсе представлено на рис. 2.9 а). При создании нового профиля системой будет задан вопрос, создать ли новый профиль на основе уже существующего, см. рис. 2.9 б). При создании нового профиля системой будет задан вопрос, создать ли новый профиль на основе уже существующего. При положительном ответе на этот вопрос, будет предложено выбрать один из уже существующих профилей, см. рис. 2.9 б). В результате сделанного выбора, будет создан новый профиль, в который будут вложены все настройки из ранее созданного выбранного профиля. Далее настройки нового профиля уже можно корректировать. Это значительно упрощает настройку разграничительной политики доступа, особенно при настройке иерархических ролей (одна роль является расширенной/суженной возможностей другой) при реализации ролевой модели контроля доступа.

Права доступа назначаются не для каждого субъекта по отдельности, а сразу для группы равноправных субъектов. Включение нового субъекта позволяет минимизировать настройки его прав доступа, за счет включения его в соответствующий профиль. Новый профиль может создаваться на основе существующего, что предполагает не задание всех настроек, а модернизацию настроек одного из существующих профилей. Для исключения же субъекта из разграничительной политики доступа в этом случае достаточно исключить его из соответствующего профиля (разграничения устанавливаются для профилей).

а) Отображение созданных профилей в интерфейсе
2.2. Объекты доступа

2.2.1. Альтернативные способы реализации метода дисcretionного контроля доступа

В общем случае, как отмечалось ранее, возможны два способа реализации метода дисcretionного контроля доступа — метод, основанный на назначении атрибутов доступа объектам и метод, основанный на задании матрицы доступа субъектов [11].

Замечание. По понятным причинам подобных альтернативных возможностей не возникает при реализации мандатного и сессионного контроля доступа, где метки безопасности должны назначаться и субъектам, и объектам доступа, для объектов доступа они должны храниться в атрибутах объектов.

Данные методы дисcretionного контроля доступа, реализуемые различными способами хранения правил доступа, принципиально различаются своими возможностями, как следствие, областью эффективного использования, и потому могут рассматриваться в качестве альтернативных решений.

В обоих случаях контроль (разграничение) доступа осуществляется центральным элементом системы защиты — диспетчером доступа, перехватывающим все запросы от субъектов к объектам, идентифицируемым субъекты, объекты доступа и типы запрашиваемого доступа субъект к объекту (чтение, запись, исполнение и т.д.), предоставляемые на основании иначе установленных правил доступа, либо отказывающий в нем.

Идентифицирующую информацию субъектов и объектов доступа принимают в виде учетные записи (или в виде ключей на основании которых диспетчер допускает или отказывает в доступе к объекту), используя правила разграничения (или контроля) доступа.

Принципиальные отличия альтернативных способов реализации методов контроля доступа проиллюстрируем с использованием соответствующих схем, см. рис. 2.10, рис. 2.11.

В первом случае для задания правил доступа каждому объекту назначаются атрибуты доступа, указывающие на то, каким субъектам и какой (каких типов) разрешается/запрещается к ним доступ. При запросе доступа диспетчер, получая необходимую учетную информацию из запроса, считывает атрибуты, принадлежащие объекту, к которому запрошены доступ, и основа...
Альтернативная реализация метода контроля доступа радикально отличается тем, что правила доступа формируются уже не для объекта, а для субъекта доступа — называются субъектами. Именно для субъектов доступа задается, к каким объектам и каких типов им разрешается доступ. Естественно, что формируемая подобным образом матрица доступа уже не является принадлежностью какого-либо объекта, а образует самостоятельную сущность, как следствие, матрица доступа представляет собой отдельный системный объект (как правило, хранится в отдельном файле). При запросе доступа диспетчер обращается к объекту «матрица доступа» и считывает из него заданные правила доступа субъекта к объекту, фигурирующих в запросе, на основании которых анализирует корректность запроса.

На самом деле, при данных альтернативных реализациях метода контроля доступа решаются концептуально различные задачи защиты. При назначении прав доступа к объектам субъектам решается задача защиты критически важных объектов, при назначении же прав доступа субъектов к объектам решается задача защиты от критически опасных субъектов при реализации и защиты режимов обработки информации.

Исторически сложилось так, что схема контроля доступа на основе атрибутов доступа широко используется в современных системных средствах, в частности, в современных универсальных ОС. Данный метод в полной мере позволяет реализовать сущность «Владения» объектом, суть которой сводится к тому, что пользователь, создавший объект («Владелец») наделяется полномочиями разграничивать права доступа к этому объекту для иных пользователей. Однако, как мы отмечали ранее, подобный
современных условиях невозможна эффективная защита информации, поскольку именно процесс несет в себе максимальную угрозу, либо расширенной субъектом «Процесс» ролевой модели контроля доступа.

Например, при подобной реализации контроля доступа появляется принципиально новая достаточно важная возможность контроля доступа к типам объектов, в частности, к типам файлов, которую мы рассмотрим в пятой главе, как следствие, и новые возможности защиты от актуальных угроз.

Важнейшим достоинством данного способа реализации контроля доступа является его универсальность. Не будем забывать о том, что методами контроля доступа должны разграничиваться не только права доступа к файловым объектам, для которых можно установить атрибуты доступа, а в общем случае должны формироваться режимы обработки информации субъектами доступа, предполагающие реализацию разграничительной политики доступа к множество разнородных объектов — к сетевым объектам, к принтерам и т.д.

С учетом всего сказанного, и понимая, что эффективная разграничительная политика доступа к ресурсам, направленная на решение рассматриваемых задач защиты информации от несанкционированного доступа, априори не может быть простой. Как мы отмечали, современными системами защиты информации от несанкционированного доступа должна реализовываться процессная модель контроля доступа, как следствие, проблема упрощения администрирования средства защиты становится одной из ключевых задач.

Отсюда можем сделать вывод о том, что для решения рассматриваемых задач защиты информации предпочтительнее метод дискретного контроля доступа, основанный на заданиях и хранении прав доступа в виде матрицы доступа в отдельном объекте. Именно его реализацию мы далее будем рассматривать.

2.2.2. Вопросы корректности идентификации объектов доступа на примере файловых объектов

Файловой объект однозначно идентифицируется своим полупотерным именем. Однозначность и корректность идентификации достигается собственной реализацией разграничительной политики доступа к ресурсам, предотвращающей несанкционированную его модификацию.

Рассмотрим, в чем состоит проблема [40].

В NTFS файловый объект может быть идентифицирован различными способами:

- Файловые объекты, задаваемые длинными именами, характеризуются той отличительной особенностью, что к ним можно обращаться, как по длинному, так и по короткому имени, например к каталогу "\Program files\", можно обратиться по короткому имени "\ Progra-1\":

- Файловые объекты, задаваемые 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 "\Documents and Settings\USER1\Главное меню" выглядит как "C:\Documents-1\ USER1\5D29-1\". К этим объектам также можно обратиться, как по длинному, так и по короткому имени;

- Файловой объект идентифицируется не только именем, но и своим идентификатором (ID) — индекс объекта в таблице MFT, при чем некоторые программы обращаются к файловым объектам не по имени, а именно по ID.

Можно обратиться к одному и тому же объекту, используя различные способы идентификации объекта в запросе доступа.

Пусть реализованная система защиты не перехватывает и не анализирует лишь один подобный способ обращения к файловому объекту, что не позволяет реализовать требования к корректности (к однозначности) идентификации файлового объекта (он может быть несанкционированно модифицирован в обход разграничительной политики доступа к ресурсам). Заметим, что в данном случае мы уже имеем безусловную технологическую уязвимость.

Следствие. Любой объект доступа, в частности, файловый объект должен однозначно идентифицироваться при любом допустимом способе обращения к нему (при любом способе его идентификации приложением), как следствие, средством защиты должны перехватываться любые способы обращения к объекту, а объект должен однозначно идентифицироваться при любом способе обращения к нему.

Как отмечали, в части корректности идентификации файлового объекта принципиальным является вопрос, каким образом хранятся права доступа субъекта к объекту [40]. Если они хранятся в виде атрибутов файлового объекта, т.е. непосредственно физически привязаны к конкретному файлу, то не важно, каким образом (по какому имени) к нему обратиться, при ана-
лиза запроса доступа будут использоваться корректные в отношении дан- но гого объекта правила доступа.

Другое дело, если правила доступа физически не привязываются к объекту - матрица доступа хранится отдельно, например, в отдельном файле. Вот в этом случае выполнение рассмотренного требования к идентификации файлового объекта становится крайне критичным.

Если же говорить об информации, хранищейся на компьютере, в широком смысле, то далеко не все данные образуют файлы. Есть еще, так называемая, остаточная информация. Дело в том, что при удалении файла итоговые метаданные ОС, собственно данные не удаляются, осуществляется переработка MFT-таблицы. Другими словами, на жестком диске и внешних накопителях всегда присутствует так называемая остаточная информация, которую невозможно прочитать, обратившись к файлу (эта информация не образует файла), но достаточно просто это сделать с использованием сторонних программ поиска доступа к диску.

С учетом сказанного может быть сформулировано следующее требование к построению безопасной системы в отношении удаления файлов. Поскольку остаточная информация не образует какого-либо объекта, подлежащего идентификации, она не должна создаваться при удалении файла. Это реализуется отдельным механизмом гарантированного удаления файлов (остаточной информации), состоящим в следующем. Запрос на удаление файла перехватывается средством защиты, после чего им осуществляется преобразование содержимого освобождаемого дискового пространства (как правило, заданное число раз записывается какая-либо информация, например, все «0», либо иная, задаваемая из соответствующего шаблона, последовательность символов), затем управление передается системе для удаления файла из массива данных с помощью средствами. В результате этого в остаточную информацию попадает принудительно записанная в файл перед его удалением последовательность символов.

Следствие. Задачи идентификации объекта доступа файловой объект предполагает решение задачи гарантированного удаления файлов, соответствующих механизмом защиты, в противном случае возникает угроза технологическая уязвимость, позволяющая получить доступ к информации в обход правил и прав контроля доступа, реализуемых системой защиты.

Исходя из особенностей функционирования системы, применимо к решению данной задачи защиты информации может быть сформулировано следующее дополнительное требование [11]. В NTFS все данные, хранящиеся на томе, содержатся в файлах. Главная таблица файлов (MFT) занимает центральное место в структуре NTFS-тогда. MFT реализована, как массив записей о файлах, где каждая запись представляет собой совокупность пар атрибутов и их значений. Размер каждой записи фиксирован и равен 1 Кб. Если размер файла достаточно мал, чтобы помещиться в теле записи, то данные такого файла хранятся непосредственно в MFT.

В процессе работы системы, NTFS ведет запись в файл метаданных — файл журнала с именем $LogFile. NTFS использует его для регистрации всех операций, влияющих на структуру тома NTFS, как то: создание файла, удаление файла, расширение файла, урезание файла, установка файловой информации, перезапись файла и изменение прав доступа к файлу. Информация, описывающая подобные транзакции, включает в себя координаты файла из MFT и в дальнейшем используется для повтора или отмены изменений. Соответственно, если данные файла содержатся в записи MFT, то при каждом изменении, данные файла будут (в числе прочего) скопированы в файл журнала.

Во избежание такого многократного дублирования данных небольших файлов, система защиты должна при создании файлов принудительно выделять пространство на томе вне таблицы MFT размером 1 Кб, что обеспечивает гарантированную очистку данных даже небольшого файла при его удалении.

Аналогичная задача гарантированного удаления остаточной информации актуальна и применима к оперативной памяти, т.к. при освобождении области оперативной памяти, выделенной процессу, в ней остается остаточная информация.

Замечание. В общем случае, применимо к любому объекту доступа, приведенные требования могут быть сформулированы следующим образом — при удалении данных из объекта — они должны удаляться гарантировано, не должны образовывать остаточной информации, к которой может быть осуществлен несанкционированный доступ.

Естественно, что если данное требование не выполнено, в системе будет присутствовать технологическая уязвимость.
Если вернуться к нормативному документу [2], то требования к этим механизмам защиты регламентируются в нем следующим образом: "При первоначальном назначении или при переопределении внешней памяти КСЗ распределения оперативной памяти КСЗ должен осуществлять её обработку.

Неидентифицируемый (в части идентификации файловых объектов) до- ступ к хранящейся на компьютере информации, причем, не только в виде значимые для этих целей программы, так называемые программы "примо- дой доступа к диску".

Опять же обратимся к нормативному документу [2], соответствующему требованию к защитам в нем сформулировано следующим образом: "КСЗ (компьютерные программные средства защиты) должен содержать механизм, обеспечивающий доступ к файловым объектам, как для ядерных, обеспечивающих тем самым защиту ПЗУ. Под "ярким" здесь понимается действия, осуществляющиеся с помощью высокого уровня и т. д., a под "скрытым" — иное действием в этом.

Замечания. В общем случае, применительно к любому объекту доступа, в общем случае включается в рассмотрение, соответствующую значимость системного уровня, а также системную реализацию соответствующую значимость системы."
Примеры отображения созданных объектов доступа (файловых объектов) в системе защиты информации представлены на рис. 2.13.

2.2.4. Идентификация внешних файловых накопителей

Казалось бы, на первый взгляд, это весьма локальный вопрос, но на него следует обратить внимание. Интересен и показателен он для нас тем, что иллюстрирует создание в системе технологической уязвимости при малейшей непродуманности технического решения.

Современные ОС, относясь в большинстве своем к универсальным, имеют множество встроенных устройств и широкие возможности по подключению внешних устройств.

Для последующего изложения воспользуемся иерархией задания (идентификации) устройств, принятой в OC Windows: класс устройств, модель устройства, устройство (конкретное устройство идентифицируется серийным номером). Отображение устройств в интерфейсе системы защиты продемонстрировано на рис. 2.14.
Заметим, что на практике иногда (для ряда устройств) бывает достаточным оперировать с такими понятиями, как класс устройств, модель устройств. Но есть ряд ключевых задач защиты, в первую очередь, это относится к файловым устройствам — внешним накопителям, например, к flash-устройствам, где для идентификации устройства необходимо использовать его серийный номер, т.е. использовать понятие конкретное устройство. В противном случае становится невозможным отличить одно устройство от другого, т.е. задача идентификации решается не корректно.

Приведем пример. Пусть на предприятии разрешена защищенная (с использованием организационных мер) обработка информации с использованием внешних накопителей. Следуя нормативному руководящему документу в области защиты информации [4], такая обработка предполагает реализацию следующей организационной меры защиты: "Учет защищаемых носителей должен проводиться в журнале (картовке) с регистрацией их выдачи (приема)".

Как реализовать это требование, если носители не различимы между собой? Выдача одного устройства контролируется, при этом другое может бесконтрольно использоваться на компьютере.

Следствие. В общем случае определение (идентификации) устройства в соотвествии с иерархическим признаком: класс устройств, модель устройств, устройство, для реализации разграничительной политики доступа не коректно. Идентификация устройств должна осуществляться на основе их уникальных идентификаторов (серийных номеров).

Замечание. Задача идентификации объекта доступа «устройство» решается отдельным методом защиты – управлением монтированием к системе устройств по их серийным номерам, который и требования к которому мы далее рассмотрим.

В файловой системе внешний накопитель (файловый накопитель) отображается буквой диска, той буквой, к которой монтируется накопитель, как файловый объект. Однако буква диска не является однозначным идентификатором конкретного устройства, т.к. буква диска, к которой будет монтироваться устройство, может быть изменена, что может привести к несанкционированному доступу к файловому накопителю.

Замечание. На сегодняшний день ряд смартфонов известных производителей вообще не монтируются к букве диска, к ним невозможно обращение как к файловым объектам, они монтируются именно к системе как устройства, а для обмена с ОС данными имеют свой собственный драйвер.

При контроле доступа к файловым накопителям, накопитель должен идентифицироваться диспетчером доступа аналогично тому, как он идентифицируется драйвером устройств, см. рис. 2.14, рис. 2.15 а).

С учетом иерархии устройств в системе, в качестве объекта доступа устройство может задаваться (с использованием соответствующей функции обозра), как моделью устройства, так и конкретным устройством, с учетом его серийного номера, см. рис. 2.15 б). Серийный номер задается в последнем поле идентификатора накопителя.

Объектом доступа может выступать не только целым файловым устройством, но и конкретный объект (папка, файл) на устройстве, идентифицируемый соответствующим образом, см. рис. 2.15 в).
2.3. Технические решения, направленные на обеспечение корректности реализации разграничительной политики доступа

Прежде чем перейти к рассмотрению вопросов реализации непосредственно методов контроля доступа субъектов к объектам, остановимся на рассмотрении соответствующих технических решений, использование которых, о чем мы говорили выше, необходимо для предотвращения несанкционированного доступа к защищаемой информации в обход разграничительной политики доступа к ресурсам, т.е. направленных на обеспечение корректности реализации разграничительной политики доступа субъектам – на инвилерование соответствующих безусловных и условных технологических уязвимостей.

2.3.1. Управление мониторированием устройств

Управление мониторированием устройств крайне важная самостоятельная задача защиты, т.к. именно набором подключаемых устройств формируется объект защиты (формируются режимы обработки информации). Естественно, что только после того, как объект сформирован (определен, возможно ли подключение внешних накопителей и каких, подключение разнообразных сетевых устройств, принтеров и т.д.), можно вести какие-либо разговоры о формировании режимов обработки информации. Например, заявление о том, что компьютер автономный, ничего не стоит ровно до тех пор, пока не будут реализованы действия по отключению встроенных сетевых устройств и предотвращены возможности подключения внешних сетевых устройств.
Локализация набора устройств является ключевой задачей при формировании режимов обработки информации субъектами доступа в рамках реализации ролевой или сессионной моделей контроля доступа.

Отметим, что важнейшей задачей защиты здесь является не только управление монтированием внешних накопителей, но и системных устройств, к которым относятся сетевые адаптеры, громкоговорители, камеры и др. сетевые устройства, которые могут быть отнесены к критическим с точки зрения реализации защиты информации.

В современных ОС Windows реализовано управление подключением устройств к системе, т.е. те устройства, которые администратором будет разрешено подключать (монтировать), разрешаются подключать всем пользователям. Естественно, подобное решение крайне ограничивает возможности одной из ключевых задач защиты информации от несанкционированного доступа, состоящей в формировании режимов обработки информации субъектами доступа в рамках реализации ролевой или сессионной моделей контроля доступа.

Как следствие, для решения этой задачи интерес представляют динамическое подключение устройств, позволяющие разрешать/запрещать (а также принудительно отключать) подключение в общем случае различных устройств для различных пользователей в процессе функционирования системы.

Заметим, что это следует не только из соответствующей постановки задачи защиты, состоящей в формировании режимов обработки информации субъектами доступа, но и из требований соответствующего нормативного документа [2], в котором сказано: «КСЗ должен обеспечивать вывод информации на запрошенное пользователем устройство как для произвольно используемых устройств, так для идентифицированных (при совпадении маркировки). Идентифицированный КСЗ должен включать в себя механизм, посредством которого санкционированный пользователь надежно сопоставляется с выделенным ему конкретным устройством». Как видим, данное требование предполагает выделение (монтирование) устройств не к системе в целом, а выделение (сопоставление) устройств конкретным пользователям.

Важным является и тот аспект, что для многих устройств, например, локальный принтер, сканер и т.д., при реализации разграничительной политики может задаваться только один тип прав доступа — разрешено использовать устройство, либо нет (это не файловый объект, где можно для различных пользователей задавать различные права доступа — чтение, запись, исполнение, удаление, переименование и т.д.). Все это обусловливает важность и целостообразность реализации разграничительной политики прав доступа пользователей к локальным устройствам управлением монтированием устройств к системе по пользователям.

Замечание. С точки зрения рассмотрения метода управления монтированием устройств к системе по пользователям, как метода контроля доступа к соответствующим устройствам, отметим его ограничение — данный метод не позволяет регистрировать (вести аудит) доступ к устройствам пользователей. Аудит может вестись только в отношении промонтирования/отмонтирования устройства к/от системы, см. рис. 2.15.

Особенность реализации разграничительной политики доступа к устройствам в данном случае состоит в том, что не перехватываются и не анализируются непосредственно запросы доступа пользователей к устройствам (в противном случае для каждого типа устройств пришлось бы делать соответствующий механизм защиты, решающий данные задачи). Для каждого пользователя задаются устройства, с которыми он может, либо, наоборот, не должен работать. При входе пользователя в систему монтируются и ему разрешается монтирование (например, внешние накопители) определенные в разграничительной политике устройства.

С учетом же того, что ОС многопользовательская (в системе одновременно может присутствовать несколько интерактивных пользователей), решается задача динамического промонтирования/отмонтирования устройств. При этом если в системе будет одновременно зарегистрировано несколько пользователей, то промонтированные к ней будут и будут разрешаться для промонтирования пользователями (например, внешние накопители) только те устройства, которые разрешено использовать всем этим зарегистрированным пользователям (остальные устройства будут автоматически отмонтированы от системы, либо их будет запрещено промонтировать).

Рассмотрим соответствующее техническое решение [37].

Интерфейс механизма управления монтированием устройств приведен на рис. 2.16.
Если говорить о контроле доступа, то динамическое мониторинг устройств может быть описано следующей моделью. Если считать, что множества $C = \{C_1, \ldots, C_l\}$ и $U = \{U_1, \ldots, U_k\}$ – соответственно линейно упорядоченные множества субъектов доступа (пользователей) и устройств, а $M = \{M, 0\}$ – конечное множество прав мониторинга устройств ($M$ – мониторинг разрешено, $0$ – мониторинг запрещено), то разграничительная политика управления мониторингом устройств описывается матрицей доступа $F$, где $F[\{C, U\}]$ – ячейка матрицы, содержит право доступа субъекта из множества $C = \{C_1, \ldots, C_l\}$ на мониторинг устройства из множества $U = \{U_1, \ldots, U_k\}$.

В любой момент времени система описывается своим текущим состоянием $Q = (C, U, M)$.

\[
F = \begin{bmatrix}
C_1 & M & 0 & \cdots & 0 \\
C_2 & M & M & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{l-1} & 0 & 0 & \cdots & M \\
C_l & 0 & M & \cdots & M \\
\end{bmatrix}
\]

Требование к безопасности системы в рассматриваемом случае может быть сформулировано следующим образом: «Для заданной системы состояние $Q_0 = (C_0, U_0, M_0)$ следует считать безопасным относительно права $M$, если не существует применимой к $Q_0$ последовательности действий, в результате выполнения которых субъектом (пользователем) $C_0$ приобретается право $M$ к устройству $M_0$, исходно отсутствующее в ячейке матрицы $M_0[C_0, U_0]$. Если же право $M$, отсутствующее в ячейке матрицы $M_0[C_0, M_0]$, приобретается субъектом $C_0$, то следует говорить, что произошла утечка права $M$, а система небезопасна относительно права $M$.

Одна из очевидных причин утечки права $M$ состоит в использовании устройств, которые не могут быть однозначно идентифицированы, в частности, внешних устройств без возможности их идентификации по серийным номерам.

Теперь о реализации мандатной схемы управления мониторированием устройств.

Еще раз приведем соответствующее требование нормативного документа [2]: «КСЗ должен различать каждое устройство ввода-вывода и каждый канал связи как производно используемые или идентифицированные (помеченные). При вводе с «помеченного» устройства (выводе на «помеченное» устройство) КСЗ должен обеспечивать соответствие между меткой
вводимого (выводимого) объекта (классификационным уровнем) и метки устройства. Такое же соответствие должно обеспечиваться при работе с «помеченным» каналом связи».

Проблема назначения меток безопасности на файловым объектом была исследована в первой главе. На основании этого исследования был сделан вывод о корректности формирования режимов обработки категорированной информации пользователями на основании метода сессионного контроля доступа (иерархическая обработка) и нерациональных методах безопасности при задании следующего правила контроля доступа.

Для объекта (не файлового объекта), к которому ранее применялись права доступа субъектов, правила доступа назначаются заданием для него множества меток безопасности \( M \), характеризующего набор субъектов, которые имеют право доступа к этому объекту.

Правило доступа, анализируемое при контроле доступа субъекта к объекту, при этом имеет следующий вид:

- субъект доступа, которому присвоена метка безопасности \( M \), имеет доступ к объекту при выполнении условия: \( M \in M \).

Рассмотрим реализацию данного технического решения [57].

При этом будем исходить из того, что как ранее говорилось, в общем случае мандатный (сессионный) контроль доступа должен использоваться совместно с дисcretionным (мандатный позволяет разграничивать доступ только между категориями пользователей, определяемых их метками безопасности, а не между отдельными пользователями). Применительно к управлению монтированием устройств использование двух интерфейсов строи (для мандатного и дисcretionного механизмов), в данной постановке задачи излишне.

Интерфейс механизма управления монтированием устройств с учетом меток безопасности (мандатов), точнее их смысловой трансформации, приведен на рис. 2.18.

![Интерфейс механизма управления монтированием устройств при назначении пользователем меток безопасности](image)

Как видим из рис. 2.18, в данном случае управление монтированием устройств осуществляется не только применительно к пользователю, но и с учетом присвоенных им меток безопасности (в правиле сразу задаются конкретные пользователи соответствующей категории). Для любого устройства при таком решении можно не только назначать множество меток безопасности \( M \), характеризующего набор субъектов, которые имеют право доступа к этому объекту, но и конкретных субъектов (пользователей), реализован тем самым одновременно и схему мандатного (сессионного), и схему дисcretionного контроля доступа.

### 2.3.2. Гарантированное удаление файлов

При проектировании механизма гарантированного удаления файловых объектов следует учитывать, что это весьма ресурсозатратный механизм защиты. Дело в том, что он предполагает выполнение следующих последовательности действий: перехват запроса на удаление/модификацию файлового объекта, запись в этот файловый объект средств защиты собственной информации с заданным числом проходов (реализация нескольких циклов перезаписи приводит к повышению гарантированности результата – защи-
Контроль доступа к компьютерным ресурсам

ты от последующего восстановления), возврат управления ОС для реализации ёю удаления/модификации файлового объекта встроенными в ОС средствами.

Исходя из сказанного, могут быть сформулированы следующие требования к реализации данного метода защиты:

• гарантированное удаление должно применяться выборочно, только в отношении критичных объектов, предназначенных для хранения защищаемой информации;

• с учетом уровня конфиденциальности информации, сохраняемой в файле, в отношении которого реализуется гарантированное удаление, должна быть возможность задания шаблона, записываемой информации, и количества проходов записи (циклов перезаписи).

Интерфейс создания шаблона записи проиллюстрирован на рис. 2.19.

Рис. 2.19. Интерфейс создания шаблона записи

Интерфейс создания (редактирования) правила гарантированного удаления, включающий создание объекта доступа и сопоставления ему шаблона записи, приведен на рис. 2.20.

Рис. 2.20. Интерфейс создания (редактирования) правила гарантированного удаления

Интерфейс выбора шаблона из списка, с заданием количества проходов (циклов перезаписи) – на рис. 2.21.

Рис. 2.21. Интерфейс выбора шаблона из списка, с заданием количества проходов

Отображение заданных в системе правил гарантированного удаления приведено на рис. 2.22.
2.3.3. Гарантированное удаление информации в освобождающихся областях оперативной памяти

Ранее отмечали, что угрозу может нести в себе, как субъект доступа «пользователь», так и субъект доступа «процесс». Следуя требованию руководящего документа [2], в части того, что очистку памяти средство защиты должно осуществлять при перераспределении внешней памяти, будем рассматривать следующие варианты запуска процедуры очистки:

• при начале сеанса контролируемого пользователя;
• при завершении сеанса контролируемого пользователя;
• при старте контролируемого процесса;
• при завершении контролируемого процесса.

Рассмотрим техническое решение. Интерфейс задания события запуска процедуры по пользователям приведен на рис. 2.23 а.

В данном интерфейсе можно выбрать из списка созданных в системе защиты контролируемых пользователей, и задать применительно для них режим запуска процедуры очистки оперативной памяти — при начале сеанса пользователя или/и при его завершении.

Замечание. При реализации мандатного (сессионного) контроля доступа к этим, по аналогии с тем, как это сделано применительно к управлению мониторингом устройств, могут учитываться метки безопасности (уровни доступа) назначенные в системе для пользователей, см. рис. 2.23 а.

Интерфейс задания событий запуска процедуры по процессам приведен на рис. 2.23 б. В данном интерфейсе можно выбрать полноту имена исполняемых файлов контролируемых процессов, при старте или завершении которых производится очистка оперативной памяти (остаточной в оперативной памяти информации при запуске процесса, соответственно, сохраняемой в виде остаточной информации в выделенной системой процессу области оперативной памяти, освобождаемой процессом).

Естественно, по завершении процесса зачищать память следует для процессов, используемых для обработки критически важной информации, по запуску — для критических процессов, с позицией возможного наделения их вредоносными свойствами, что соответствует реализации процессной модели контроля доступа.
2.3.4. Контроль прямого доступа к дискам — к устройствам хранения данных

На сегодняшний день существует достаточное количество специализированных утилит прямого доступа к диску, кроме того, многие редакторы реализуют данный функционал при установке соответствующих плагинов. Да и вообще утверждать об отсутствии у программы подобной возможности нельзя, кроме того, данной возможностью приложение может быть наделено в результате реализации на него атак, предполагающей наделение приложения вредоносными свойствами. Другими словами, защиту следует строить в предположении, что данная возможность на компьютере присутствует.

Особенностью прямого доступа к диску является то, что объектом доступа уже является не файл, а непосредственно физическое дисковое пространство, как следствие, доступ к нему осуществляется в обход контрола доступа к файловым объектам. Другими словами, если прямой доступ к диску не контролируется, то существует прямая угроза несанкционированного доступа к хранящейся на диске информации в обход реализованной на компьютере разграничительной политики доступа к файловым объектам, причем штатными средствами.

В качестве дисков (объектов доступа), к которым должен контролироваться прямой доступ, естественно, следует рассматривать, как жесткий диск (диски), подключенный к компьютеру, так и разрешенные для мониторинга в системе внешние файловые накопители, в том числе, flash-устройства.

В руководимом документе [1] по этому поводу сказано следующее:

– КСЗ должен содержать механизм, претворяющий в жизнь дисcretionные ПРД как для яных действий пользователя, так и для скрытых, обеспечивающий тем самым защиту объектов от НСД (т.е., от доступа, недопустимого с точки зрения заданного ПРД). Под «яными» здесь подразумеваются действия, осуществляемые с использованием системных средств — системных макрокоманд, инструкций языков высокого уровня и т.д., а под «скрытыми» — иные действия, в том числе с использованием собственных программ работы с устройствами.

Таким образом, диск (жесткий диск, внешний накопитель) является объектом доступа, прямой доступ субъектов к которому должен контролироваться и разграничиваться.

Рассмотрим реализацию соответствующего технического решения. Объект доступа создается из интерфейса, представленного на рис. 2.25, он выбирается из списка накопителей, разрешенных для мониторинга в системе, в интерфейсе, приведенном на рис. 2.26.

Права прямого доступа субъектов (в общем случае в разграничительной политике субъектом выступает профиль, об этом говорили ранее) к объектам назначаются из интерфейса, приведенного на рис. 2.27. Применяя разграничительную политику прямого доступа к дискам для субъекта (для профиля) отображается в интерфейсе, представленном на рис. 2.28.

Замечание. Мандатная схема контроля прямого доступа к дискам какого-либо смысла не имеет (такую возможность необходимо предотвращать для всех пользователей). А вот использование субъекта доступа профиля, в предположении о том, что различные права прямого доступа к дискам могут назначаться различным процессам, необходимо. Это обусловливается тем, что некоторые приложения штатно используют именно такое обращение к данным, как следствие, подобную возможность для обеспечения работоспособности таких приложений необходимо разрешать.

Замечание. Подробно задачу создания правил и назначения прав доступа субъектов к объектам мы рассмотрим в следующей главе.
Глава 2. Субъекты и объекты доступа

Рис. 2.28. Интерфейс отображения созданных для субъекта "профиль" правил прямого доступа к диску

Как видим, корректная, в части нивелирования безусловных и условных технологических уязвимостей, реализация разграничительной политики доступа, требует решения весьма внушительного набора дополнительных задач защиты, применительно к решению каждой из которых могут быть сформулированы свои требования.

2.4. Основные результаты и выводы

1. Исследованы задачи идентификации субъектов и объектов доступа в разграничительной политике, направленной, как на формирование режимов обработки информации субъектами доступа, так и на нивелирование угроз условных и безусловных технологических уязвимостей.

2. Предложена процессная модель контроля доступа, которая может реализовываться наряду с ролевой или с сессионной моделями контроля доступа, реализация которой предполагает защиту от наиболее актуальной сегодня угрозы атак на системные процессы и приложения, в том числе, атак, использующих угрозы уязвимости реализации системных процессов и приложений. В результате исследования вопросов реализации процессной модели контроля доступа сформулированы требования к идентификации субъекта доступа в современной разграничительной политике – сделан вывод о том, что он должен идентифицироваться тремя сущностями: первичный идентификатор пользователя, полноту частное имя процесса, эффективный идентификатор пользователя.

3. Разработаны вероятностные модели и методы контроля доступа, с использованием которых сформулированы требования к назначению прав
Контроль доступа к компьютерным ресурсам

доступа к объектам при реализации процессной модели контроля доступа с учетом различных значений вероятностей неделения различными способами процессов вредоносными свойствами.

4. С использованием вероятностной модели мандатного контроля доступа, в основу которой положено категорирование объектов доступа по уровням конфиденциальности информации, сформулировано «непротиворечивое правило мандатного контроля доступа», которое должно составить основу реализации метода сессионного контроля доступа, как следствие, использовать в качестве основы в сессионной модели контроля доступа.

5. Исследованы альтернативные способы хранения прав доступа при реализации метода дискретного контроля доступа. Сделан вывод о том, что, ввиду универсальности, достигаемого существенного упрощения администрирования реализующей данный метод системы защиты информации, особенно в части реализации процессной модели контроля доступа, появляющихся новых важных возможностей защиты, правила доступа – матрицу доступа, целесообразно хранить в виде отдельного объекта, например, файла, а не задавать в качестве атрибутов объектам доступа.

6. Рассмотрены способы и технические решения идентификации субъектов и объектов доступа в разграничительной политике с использованием масок, появляющиеся при этом противоречия и подходы к их разрешению.

7. Сформулированы требования к корректности идентификации устройств и объектов доступа на примере файловых объектов.

8. Разработан метод и техническое решение по управлению монтированием устройств по пользователям, без использования которого в общем случае невозможно корректно реализовать ни рольевую, ни сессионную (при назначении пользователям меток безопасности) модели контроля доступа. Показано, что данный метод может позиционироваться и в качестве самостоятельного метода контроля доступа к устройствам, характеризуемым возможностью задания в отношении них только одного права доступа – разрешено использовать субъекту устройство или нет.

9. Сформулированы задачи, разработаны методы и технические решения, направленные на нивелирование технологических уязвимостей, связанных с корректностью идентификации обрабатываемых в информационной системе данных как объектов доступа – метод гарантированного удаления файлов, метод гарантированной очистки оперативной памяти, метод контроля прямого доступа субъектов к дискам (к устройствам хранения данных).
Глава 3. Принципы и методы контроля доступа к статическим объектам

3.1. Классификация объектов доступа

Введем принципиально важную классификацию объектов доступа, применительно к рассматриваемым вопросам реализации контроля доступа [30-32]. Будем их подразделять на статические и создаваемые в процессе работы системы.

**Определение.** Под статическими объектами доступа будем понимать объекты, присутствующие в системе на момент реализации администратором разграничительной политики доступа субъектов к объектам.

**Определение.** Под создаваемыми в процессе функционирования системы (или далее, под создаваемыми) объектами доступа будем понимать объекты, отсутствующие в системе на момент реализации администратором разграничительной политики доступа субъектов к объектам, и создаваемые пользователями впоследствии, уже в процессе функционирования системы.

Если задуматься, то между объектами доступа данных классов колоссальная разница. Различия рассмотрим на примере файловых объектов. Статические файловые объекты — это, прежде всего, системные объекты — исполняемые файлы и файлы настроек ОС и приложений. Создаваемые же файловые объекты — это объекты создаваемые пользователями непосредственно во время их работы на компьютере, и предназначенные, в первую очередь, для хранения обрабатываемой информации, в том числе, конфиденциальной.

Как следует, в эффективной защите, в том числе, в реализации контроля доступа, если мы рассматриваем задачу защиты информации от несанкционированного доступа, в первую очередь, нуждаются именно создаваемые файловые объекты, к ним, в первую очередь, и должна реализовываться разграничитительная политика доступа. Однако, на момент задания разграничитительной политики доступа администратором, этих объектов еще попросту нет. Возникает резонный вопрос — как назначить к ним правила доступа?

3.2. Принципы контроля доступа к статическим объектам

Основополагающим принципом реализации контроля доступа к статическим объектам является наличие двух равноправных сущностей — субъект и объект доступа в разграничительной политике доступа. То, предполагается, что объекты доступа присутствуют в системе, к ним могут быть разграниченны права доступа субъектов [14].

Рассмотренные в первой главе базовые методы контроля доступа, в соответствии с основополагающим принципом реализации контроля доступа реализуют контроль доступа именно к статическим объектам. Напомним, исходный посыл, который мы делали ранее при построении соответствующих моделей: «Если считать, что множество $C = \{C_1, \ldots, C_n\}$ и $O = \{O_1, \ldots, O_m\}$ — соответственно линейно упорядоченные множества субъектов и объектов доступа, а $R = \{w, r, x, d\}$ конечное множество прав доступа (чтение, запись, удаление, исполнение и т.д.), то разграничительная политика доступа субъектов к объектам описывается...». Т.e. все рассмотренные нами ранее методы и модели контроля доступа предполагают наличие объектов на момент задания разграничительной политики доступа.

Итак, с учетом сказанного выше сформулируем основополагающие принципы контроля доступа к статическим объектам, позволяющие реализовать эффективный контроль доступа:

1. Контроль доступа к статическим объектам предполагает наличие подобных объектов на момент реализации (настройки администратором) разграничительной политики доступа, т.е. до ввода защищенного компьютера в эксплуатацию.
2. В разграничительной политике доступа используются две равноправные сущности — субъект и объект доступа.
3. Разграничительная политика доступа предполагает назначение прав доступа субъектов к объектам (а не, наоборот, к объектам субъектам).
для созданных наноконтейнеров пользователи "принуждаются" администратором создавать свои файлы только в определенных папках. Созданные файлы наделены разграничением, установленное администратором для папок. По средством идеализации разграничителей политики доступа к папкам-контейнерам для пользователей разграничивается и последующий доступ к уже созданным в процессе функционирования системы файлам.

Таким образом, контроль доступа к создаваемым объектам реализуем через разграничение прав доступа по их созданию субъектами в соответствующих объектах-контейнерах, с последующим разграничение прав доступа субъектов уже к объектам-контейнерам [32]. Естественно, что подобный подход (а именно он сегодня и используется на практике) не только весьма не логичен (если речь не идет о системных объектах), но и обусловливает принципиальное усложнение реализации разграничителей политики доступа к файловым объектам, а в ряде случаев, и невозможность ее корректной реализации, о чем поговорим далее. Достаточно задуматься о том, какие действия потребуется выполнить администратору, чтобы, например, изолировать откушку информации десятком приложений при реализации процессной модели контроля доступа [14].

Однако принципы, модели и методы контроля доступа к создаваемым объектам (а это создаваемые файлы и данные, помещаемые процессами в буфер обмена для временного хранения) мы рассмотрим в следующей главе, здесь же будем рассматривать контроль доступа к статичным объектам, понимая при этом, что в состав информационной системы входит множество разнородных статичных (присутствующих на момент реализации разграничителей политики доступа) объектов, к которым должны разграничиваться права доступа субъектов с целью формирования режимов обработки информации субъектами доступа.

3.3. Требования к построению безопасной системы

Вновь обратимся к модели дискреционного контроля доступа [60]. Если считать, что множество \( C = \{C_1, ..., C_l, C_j, ..., C_k\} \) и \( O = \{O_1, ..., O_j, ..., O_k\} \) - соответственно линейно упорядоченные множества субъектов и объектов доступа, а \( R = \{w, r, x, d\} \) конечное множество прав доступа (чтение (r), запись (w), удаление (d)), исполнение (x), отсутствие прав доступа (0)) субъекта \( C_i \) к объекту \( O_j; i = 1, ..., l; j = 1, ..., k \), то матрица доступа \( M_x \) используемая для
реализации разграничительной политики методом контроля доступа с при- нуждительным управлением потоками информации имеет следующий вид:

\[
\begin{bmatrix}
O_1 & O_2 & \cdots & O_k \\
C_1 & r & w & d & w & \cdots & 0 \\
C_2 & r & w & d & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
C_i & 0 & 0 & \cdots & r \\
C_j & 0 & w & \cdots & r & w & d
\end{bmatrix}
\]

В любой момент времени система описывается своим текущим состоянием \(Q = (C, O, M[C, O])\) – ячейка матрицы, содержит набор прав доступа. Будем обозначать \(C_i(R)O_j\) разрешением права доступа субъекту \(C_i\) к объекту \(O_j\), где \(R = \{r, w, d\}\): исполнение \((x)\), запись \((w)\), чтение \((r)\), удаление \((d)\).

### 3.3.1. Требования к правилам контроля доступа

В первой главе мы отмечали, что серьезную проблему информационной безопасности составляет угроза уязвимости администрирования системы защиты. Для ее нивелирования должны быть сформулированы требования к правилам контроля доступа (разграничения прав доступа субъектов к объектам), выполнение которых обеспечивает построение безопасной системы (отсутствует возможность утечки прав доступа), которым должен следовать администратор при настройке системы защиты информации.

Таким образом, применительно к приведенной выше модели, сформулируем правила контроля доступа, при реализации которых в системе отсутствует утечка права \(R = \{r, w, d\}\), что не приводит к несанкционированным утечкам информации между субъектами, который будем обозначать \(C_i(P)C_j, i \neq j\) (субъект \(C_i\) получает посредством запроса на информацию, обрабатываемой субъектом \(C_j\), т.е. реализуется безопасная система [11].

Сформулируем данные требования на примере реализации контроля доступа к файловым объектам, исходя из их общности в части назначаемых для них типов доступа, понимая при этом, что подобные требования в отношении соответствующих типов доступа должны выполняться и в отношении иных объектов доступа. Сначала рассмотрим метод дискреционного контроля доступа.
Лемма 3.3. При расширении канонической матрицы доступа правом читения (r): C(r)O, при уже разрешенном в матрице праве читения (r): C(r)O, одновременно с этим должно разрешаться право читения (r): C(r)O, где i ≠ j ≠ k, i = 1, ..., k = 1, ..., l, что противоречит возможности несанкционированного обмена информацией между субъектами C[P]C, из-за образующей при этом утечки права чтения (r): C(r)O.

Доказательство. Обратимся к рис. 3.1, где проиллюстрирована утечка права чтения (r), происходящая при невыполнении требования, формулируемого Леммой 3.3.

![Diagram of access violation](image)

Рис. 3.1. Иллюстрация утечки права чтения (r)

Из рис. 3.1 видно, что в случае разрешения права C(r)O, при исходно заданном праве C(r)O, с учетом того, что C априори обладает правом записи: C(w)O, организуется информационный поток чтения от O в C, путем реализации следующей последовательности действий: C(r)O, C(w)O, C(r)O. Если право C(r)O не разрешено, то присутствует утечка права чтения C(r)O, как следствие, возможность несанкционированного обмена информацией между субъектами C[P]C. Из рис. 3.1 также видно, что сформулированное Леммой 3.3 правило распространяется на все случаи, задаваемые условием: i ≠ j ≠ k, i = 1, ..., k = 1, ..., l, k = 1, ..., l. Лемма доказана.

Лемма 3.4. При расширении канонической матрицы доступа правом записи (w): C(w)O, при уже разрешенном в матрице праве записи (w): C(w)O, одновременно с этим должно разрешаться право записи (w): C(w)O, где i ≠ j ≠ k, i = 1, ..., k = 1, ..., l, что противоречит возможности несанкционированного обмена информацией между субъектами C[P]C, из-за образующей при этом утечки права записи (w): C(w)O.

Доказательство. Обратимся к рис. 3.2, где проиллюстрирована утечка права записи (w), происходящая при невыполнении требования, формуллируемого Леммой 3.4.

![Diagram of access violation](image)

Рис. 3.2. Иллюстрация утечки права записи (w)

Из рис. 3.2 видим, что в случае разрешения права C(w)O, при исходно заданном праве C(w)O, с учетом того, что C априори обладает правом записи: C(r)O, организуется информационный поток записи от C в O, путем реализации следующей последовательности действий: C(w)O, C(r)O, C(w)O. Если право C(w)O не разрешено, то присутствует утечка права записи C(w)O, как следствие, возможность несанкционированного обмена информацией между субъектами C[P]C. Из рис. 3.2 также видно, что сформулированное Леммой 3.4 правило распространяется на все случаи, задаваемые условием: i ≠ j ≠ k, i = 1, ..., l, j = 1, ..., k = 1, ..., l, k = 1, ..., l. Лемма доказана.

Относительно удаления. Данное право (d) тесно связано с правами записи и модификации (естественно, что в моделях нами рассматривается каждый усеченный набор базовых прав доступа). Можно рассмотреть два способа записи: «запись» — право, позволяющее, кроме создания нового, модифицировать и удалять существующие файлы, и «действие» — право,
позволяющее осуществлять запись новых файлов без возможности модификации и удаления ранее созданных файлов. Естественно, что по своему функциональному назначению и возможностям применения – это совершенно различные права доступа.

Замечание. Ранее под правом записи (w) мы рассматривали право модификации существующих файлов и записи новых, без возможности удаления существующих (в моделях ранее записи и право удаления нами были разделены). Исходя из этого, сформулируем соответствующее требование к построению безопасной системы. Здесь опять же возникает вопрос, а модификацию следует рассматривать, как удаление (весь модификация не приводит к удалению объекта, но может привести к нарушению целостности информации, хранящейся в данном объекте).

Если рассматривать право записи (w), как право модификации существующих файлов и записи новых, без возможности удаления существующих, а модификацию существующих файлов, не рассматривать как удаление, то утечки права удаления (d) не происходит, дополнительных требований к построению безопасной системы формулировать не требуется. Если возможность модификации файла, входящей в право записи (w), рассматривать как удаление (искажение информации в определенном смысле можно приравнять к ее удалению), то дополнительно может быть сформулировано следующее требование, направленное на предотвращение утечек права удаления (d). При расширении канонической матрицы доступа правом записи (w): $C_w(v)O_v$, дополнительно должно включаться право удаления $C_d(v)O_v$.

Рассмотрим на примере реализации правил доступа, формулируемых Леммой 3.3 и Леммой 3.4, при построении разграничительной политики доступа к файловым объектам – рассмотрим, насколько меняет разграничительную политику доступа выполнение этих требований.

Пусть исходная матрица доступа $M_{in}$ имеет следующий вид:

$$M_{in} = C_1 = \begin{bmatrix} O_1 & O_2 & O_3 & O_4 & O_5 \\ r, w, d & 0 & 0 & 0 & 0 \\ C_2 & 0 & r, w, d & r & 0 \\ C_3 & w & 0 & r, w, d & 0 \\ C_4 & 0 & 0 & 0 & r, w, d \\ C_5 & 0 & 0 & 0 & 0 \\ \end{bmatrix}$$

Пусть администратору требуется внести в матрицу доступа $M_{in}$ (расширить соответствующую матрицу $M_{ir}$) правило $C_{r,v}O_v$. При этом, выполняя требования Леммы 3.3 и Леммы 3.4, администратору также потребуется внести два дополнительных правила: $C_{r,v}O_v$ и $C_{r,v}O_v$. Получаем следующую матрицу доступа $M_{ir}$, выполняющую сформулированные требования к построению безопасной системы:

$$M_{ir} = \begin{bmatrix} O_1 & O_2 & O_3 & O_4 & O_5 \\ r, w, d & 0 & 0 & 0 & 0 \\ C_2 & 0 & r, w, d & r & 0 \\ C_3 & w & 0 & r, w, d & 0 \\ C_4 & 0 & 0 & 0 & r, w, d \\ C_5 & 0 & 0 & 0 & 0 \\ \end{bmatrix}$$

Теперь о методах сессионного и мандатного контроля доступа, для которых правила формализованы (не задаются администратором).

Поскольку методом сессионного контроля доступа реализуется каноническая матрица доступа, то в соответствии с Акссиомой 3.1 может утверждать, что данными формализованными правилами выполняются требования к построению безопасной системы (утечка права доступа отсутствует).

Теперь рассмотрим формализованные правила, используемые в методе мандатного контроля доступа, направленные на защиту от появления категорий обрабатываемой информации, имеют следующий вид:

1. Субъект $C$ имеет доступ к объекту $O$ в режиме «Чтения» в случае, если выполняется условие: $M_{ir} < - M_{aco}$

2. Субъект $C$ имеет доступ к объекту $O$ в режиме «Записи» и «Удаления» в случае, если выполняется условие: $M_{ir} = M_{aco}$

Рассмотрим матрицу контроля доступа $M_{ir}$, реализующую эти правила:

$$M_{ir} = \begin{bmatrix} O_1(M) & O_2(M) & \ldots & O_5(M) \\ C_{r,v}(M_1) & 0 & \ldots & r \\ C_{r,v}(M_2) & 0 & \ldots & \ldots & r \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ C_{r,v}(M_{ir}) & 0 & \ldots & \ldots & r \\ \end{bmatrix}$$

Оценим корректность правил мандатного контроля доступа, в части возможности построения безопасной системы.
Лемма 3.5. Система, реализующая метод мандатного контроля доступа, безопасна относительно права записи (w).

Доказательство. Право записи в мандатном контроле доступа к создающему файловым объектам задается следующим образом: \( C_i(w)O_j, i \neq j, i = 1, ..., k; C_i(w = 0)O_j, i = 1, ..., l, j = 1, ..., l; \) что априори не может привести к утечке права записи (w). Лемма доказана.

Лемма 3.6. Система, реализующая метод мандатного контроля доступа, безопасна относительно права чтеания (r).

Доказательство. Обратимся к Лемме 3.3, в которой сформулировано корректное правило управления чтением: «При расширении канонической матрицы доступа правом чтения (r): \( C_i(r)O_j, \) при уже разрешенном в матрице праве чтения (r): \( C_i(r)O_j, \) одновременно с этим должно разрешаться право чтения (r): \( C_i(r)O_j, \) где \( i \neq j, i = 1, ..., k; j = 1, ..., l; \) что противоречит возможностям несанкционированного обмена информацией между субъектами \( C_i[P]C_j, \) из-за образующейся при этом утечки права чтения (r): \( C_i(r)C_j. \) Как следует из представленной выше матрицы \( M_{ac}, \) аргументы, сформулированные Леммой 3.3, для метода мандатного контроля доступа выполняются. Доказательство Леммы 3.6 сводится к доказательству Леммы 3.3. Лемма доказана.

Говоря о методе мандатного контроля доступа нельзя не затронуть проблемы разметки включающих объектов. Несмотря на то, что созданные для сохранения пользователем информации различных категорий конфиденциальности папки-контейнеры, размещаемые администратором, располагаются в некоторых включающих объектах (диск, каталог). Включающие объекты также следует как-то разметить (присваивать им метки безопасности), для разрешения к ним доступа, вопрос: как?

Лемма 3.7. Иерархическому объекту (папке), включающему в себя размещаемые включающие объекты, при использовании в системе меток безопасности \( M_i, i = 1, ..., k, \) должна присваиваться метка безопасности \( M_{ac}, \) исходя из следующего правила: \( M_{ac} = M_{max} > \min(M_i, i = 1, ..., k) \), при условии, что меньшее значение метки, тем выше уровень конфиденциальности. Метка \( M_{ac} \) должна присваиваться всем иерархическим включающим файловым объектам (папкам), в которых не требуется создания файлов для хранения конфиденциальной информации.

Доказательство. Иллюстрация корректного правила назначения метки безопасности иерархически включающим файловым объектам представлена на рис. 3.3.

![Рис. 3.3. Иллюстрация корректного правила назначения метки безопасности иерархически включающим файловым объектам](attachment:image)

Так как при реализации правила: \( M_{ac} = \min(M_i, i = 1, ..., k), \) субъект имеет право только чтения иерархического включающего файлового объекта (может осуществлять «обзор» размещенных папок, созданных для хранения файлов с категорированной информацией), и не имеет возможности создавать в нем новые объекты — может создавать файлы исключительно в созданных для этой цели включающих объектах, то при реализации данного правила назначения меток безопасности иерархически включающим файловым объектам, реализуется основополагающее правило контроля доступа, корректность которого в части утечек прав записи и чтения доказывается Леммой 3.5 и Леммой 3.6. Лемма доказана.

Следствие. При реализации мандатного контроля доступа, основанного на наследовании метки безопасности от включающего файлового объекта, для реализации корректного правила назначения метки безопасности иерархически включающим файловым объектам, целесообразно присваивать иерархическому объекту «обозра включающих объектов» \( M_{ac}, \) исходя из следующего правила: \( M_{ac} = M_{max} > \min(M_i, i = 1, ..., k) \), при условии, что чем меньше значение метки, тем выше уровень конфиденциальности. Метка \( M_{ac} \) должна присваиваться всем иерархическим включающим файловым объектам (папкам), в которых не требуется создания файлов для хранения конфиденциальной информации.
Глава 3. Принципы и методы контроля доступа к статическим объектам

Лемма 3.8. Мандатный контроль доступа, основанный на наследовании метки безопасности от включающего файлового объекта, корректно реализует принцип контроля удаления.

Доказательство. Иллюстрация возможных правил контроля удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от включающего файлового объекта, проиллюстрирована на рис. 3.4.

Рис. 3.4. Иллюстрация возможных правил удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от включающего файлового объекта

Любой субъект C имеет доступ к объекту O в режиме «Удаления» только в случае, если выполняется условие: \( M_C \geq M_O \). Это справедливо и в части включающего объекта (папки), который размещается, и в части вложенных объектов (файлов), наследующих метку включающего объекта. Иллюстрация корректного правила удаления для случая, если иерархически-

му включающему файловому объекту назначить метку, исходя из условия \( M_{max} > \min\{M_i, i = 1, ..., k\} \). Приведена на рис. 3.5.

Рис. 3.5. Иллюстрация корректного правила удаления при иерархии размещений файловых объектов

Поскольку ни один субъект не может удалить иерархический объект, ввиду выполнения условия: \( M_{max} > \min\{M_i, i = 1, ..., k\} \), правило контроля удаления в системе с иерархическими включающими объектами, представленное на рис. 3.5, сводится к корректному правилу удаления, проиллюстрированному на рис. 3.4. Лемма доказана.

3.3.2. Модель и метод контроля доступа, реализуемого перенаправлением запросов доступа

3.3.2.1. Проблема и метод контроля доступа к неразделяемой системе и приложениями объектам

Важнейшим является требование Аксиомы 3.2: Контроль доступа в безопасной системе реализован корректно только в том случае, если настройками диспетчера доступа может быть реализована каноническая матрица доступа. Без его реализации о корректности реализации контроля доступа в целом, как следствие, о построении безопасной системы и об эффективности защиты, говорить не приходится.

Рассмотрим проблему контроля доступа к неразделяемой системе и приложениями объектам на примере файловых объектов.

В современных ОС и приложениях, как правило, присутствуют папки (каталоги), не предназначенные для их разделения между субъектами доступа,
например, это каталоги, используемые системой и приложениями для временного хранения файлов, «Temp». Чтобы обеспечить нормальную работу системы и таких приложений, папки, в которые помещаются временные файлы, необходимо разрешить для доступа всем субъектам. И это не единственный пример неразделяемых файловых объектов, называемых объектами общего (или коллективного) доступа.

Заметим, что каждый доступ должен быть предоставлен различным субъектам (пользователям) и к ряду системных файлов. Этот вопрос в полной мере был нами проиллюстрирован в первой главе.

Таким образом, существуют папки, для которых невозможно разграничить доступ субъектам, в частности, пользователям (при реализации же разграничения, либо пользователи, которым запрещен доступ к рассматриваемым папкам, не смогут работать с приложением, либо им не сможет предоставляться необходимый сервис). Действительно, если обозначить группу объектов файловой системы (каталогов), не разделяемых системой между субъектами, как $O^*_n$, то каноническая матрица доступа $M_k$ примет следующий вид:

$$
M_k = \begin{bmatrix}
O_1 & O_2 & \cdots & O_n & \cdots & O_i \\
C_1 & 0 & \cdots & 0 & \cdots & 0 \\
C_2 & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_{n-1} & 0 & \cdots & 0 & \cdots & 0 \\
C_i & 0 & \cdots & 0 & \cdots & 0
\end{bmatrix}
$$

Рассмотрим метод мандатного контроля доступа при данных условиях реализации разграничительной политики. Пусть между субъектами, которым соответственно присвоены метки $M_1$ и $M_2$ (обрабатывают информацию различных уровней конфиденциальности), какая-либо папка не может быть разделена (в этой папке требуется обеспечить право доступа для обоих субъектов).

Поскольку одному объекту (папке) может быть присвоена только одна метка безопасности (в нашем случае $M_1$ или $M_2$), соответственно получаем возможность реализации одного из двух правил контроля запись/чтение, проиллюстрированных на рис. 3.6.

### Замечание

Разработчики средств защиты пытаются различными способами решить данную проблему. Например, широко используется следующее решение – разрешить всем субъектам доступ на запись/чтение в неразделяемые папки (чтобы обеспечить корректную работу системы и приложений), а по окончании текущего сеанса работы все объекты из этой папки удалить. Однако, во-первых, данное решение априори не является корректным в общем случае, т.к. современные системы многопользовательские и в одном сеансе может находиться несколько пользователей с различными полномочиями, во-вторых, оно никак не относится собственной к контролю доступа, в том числе, в части корректности его реализации.

Решением рассматриваемой проблемы в общем виде является реализация метода переадресации запросов доступа к объектам файловой системы (папкам), не разделяемым системой и приложениями между субъектами доступа, который состоит в следующем [11]. Для каждого субъекта доступа (напри-
мер, пользователя, в общем случае, как отмечали, субъект доступа определяется тремя сущностями для неразделяемого объекта реализуется соответствующий субъектный объект, например, для каталога «Общий ресурс», ведется каталог «Общий ресурс 1» для первого пользователя, «Общий ресурс 2» для второго пользователя и т.д. При записи информации системой или приложением в неразделяемый каталог (соответственно, чтении из каталога), диспетчер доступа перенаправляет запрос доступа в (из) соответствующий каталог текущего пользователя. Например, если текущим пользователем является первый пользователь, то при сохранении информации в каталоге «Общий ресурс» данная информация будет перенаправлена диспетчером доступа и сохранена в каталоге «Общий ресурс 1».

Замечание. В общем случае в рассматриваемом примере можно сделать перенаправление запросов доступа только для одного субъекта, предоставив возможность доступа к исходному объекту для другого субъекта.

При этом механизм перенаправления запросов к неразделяемым системе объектам должен обрабатывать запрос перед механизмом контроля (разграничения) доступа, который, в свою очередь, может реализовывать контроль доступа на основе матрицы (дисперсионной) или меток безопасности (манипуляций). Средствами механизма контроля доступа к объектам файловой системы разграничиваются права доступа к каталогам, в которые пере- направляется информация, например, доступ к каталогу «Общий ресурс 1» следует разрешить только первому пользователю, остальным – запретить. В результате реализации рассмотренного метода доступа позволяет обеспечить отсутствие общих ресурсов файловой системы для пользователей.

Замечание. Собственно в исходных неразделяемых системе пакетах в рассматриваемом примере, запрос доступа к которым переадресуется в специально созданные для этого пакеты, информация сохраняется не будет, т.е. данные пакеты становятся в системе виртуальными (к ним нет необходимости разграничивать доступ, он будет невозможен из-за переадресации запросов).

Лемма 3.9. При реализации метода переадресации запросов доступа к объектам файловой системы (пакетам), не разделяемым системой и приложениями между субъектами доступа, настройками диспетчера доступа может быть реализована каноническая матрица доступа.

Доказательство. При реализации метода переадресации запросов доступа к объектам файловой системы (пакетам), не разделяемым системой и приложениями между субъектами доступа, в системе не остается неразделяемых системой и приложениями файловых объектов. К любому файловому объ-
Глава 3. Принципы и методы контроля доступа к статическим объектам

3.3.2.2. Модель и метод контроля доступа перенаправлением запросов доступа

Контроль доступа к неразделяемым объектам формально описывается матрицей перенаправлений запросов доступа. В ячейке матрицы перенаправлений запросов доступа, реализуемых для разделения объектов между субъектами, может задаваться одно из следующих двух правил: $C(O_n)O_i$, что означает перенаправление запросов доступа от субъекта $C_i$ к объекту $O_n$ в объект $O_i$, отсутствие перенаправления — $C(O_n)O_i$.

Матрица перенаправлений запросов доступа $M_{rp}$ для матрицы доступа $M_a$ имеет следующий вид:

$$
M_{rp} = \begin{bmatrix}
O_1 & O_2 & \ldots & O_{n-1} & \ldots & O_m & \ldots & O_n & \ldots & O_i \\
C_1 & r, w, d & 0 & \ldots & r, w, d & 0 & \ldots & 0 & \ldots & 0 \\
C_2 & 0 & r, w, d & 0 & \ldots & 0 & \ldots & 0 & \ldots & 0 \\
& \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_{n-1} & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & \ldots & 0 \\
C_i & 0 & 0 & \ldots & 0 & r, w, d & \ldots & r, w, d & \ldots & 0 \\
\end{bmatrix}
$$

Контроль доступа, реализуемый перенаправлением (переадресацией) запросов доступа, требует совместного его использования с дисcretionным контролем доступа, который применяется для разграничения прав доступа к объектам, в которые перенаправляются запросы доступа.

Контроль доступа, реализуемый перенаправлением (переадресацией) запросов может использоваться и в схеме мандатного (сессионного) контроля доступа. В этом случае создаваемым объектам доступа, в которые перенаправляются запросы доступа как к неразделяемой системой и приложениям объектам, присваиваются метки безопасности. При этом подобная...
ных объектов создается по числу назначенных меток безопасности пользователем. Перенаправление запросов должно осуществляться в объект, которому назначена та же метка, что и пользователю, запрашивающему доступ к объекту, разделяемому системой защиты.

Рассмотрим возможность применения данного метода контроля доступа с целью реализации полноценного метода контроля доступа. Положенного в этом смысле, что средствами перенаправления запросов доступа требуется выполнить следующее условие: \( C_i (R=0)O_j \neq n, j \neq i, i = 1, ..., l, j = 1, ..., l. C \) Этот целию зададим правила перенаправления запросов доступа, описывающие следующей матрицей \( M_n \):

\[
M_n = \begin{bmatrix}
O_{11} & O_{12} & \cdots & O_{1d} & \cdots & O_{1l} \\
O_{21} & 0 & \cdots & O_{2d} & \cdots & O_{2l} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
O_{(l-1)d} & O_{(l-1)(l-1)} & \cdots & 0 & \cdots & 0 \\
O_{ld} & O_{ld} & \cdots & O_{ld} & \cdots & 0
\end{bmatrix}
\]

Видим, что реализацией подобных правил перенаправления запросов доступа может быть выполнено требуемое условие: \( C_i (R=0)O_j \neq n, j \neq i, i = 1, ..., l, j = 1, ..., l \). Исходный же разделаемый объект доступа он при этом становится виртуальным – для него справедливо: \( C_i (R=0)O_{i} = 1, ..., l \).

Естественно, если рассматривать перенаправление (передачу) запросов доступа, в качестве самостоятельного полноценного метода контроля и разграничения прав доступа (в качестве альтернативы дискретному методу контроля доступа субъектов к объектам), а не только применимо к разделению между субъектами доступа к разделяемым системой и приложениям объектов доступа, то в этом случае следует говорить о том, что перенаправление запросов доступа может задаваться администратором применительно к любым объектам доступа.

При этом в ячейке матрицы перенаправлений запросов доступа, применительно к любому объекту может задаваться одно из следующих правил перенаправлений: \( C_i (O_j)O_k \), что означает перенаправление запросов доступа от субъекта \( C_i \) к объекту \( O_j \) в объект \( O_k \), либо отсутствие подобного перенаправления \( C_i (O)O_k \).

Построим матрицу перенаправлений запросов доступа \( M_n \), реализующую диагональную матрицу доступа:

\[
D = \begin{bmatrix}
O_{i1} & 0 & \cdots & 0 \\
0 & O_{i2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & O_{il}
\end{bmatrix}
\]

Данный пример иллюстрирует то, что разграничительная политика доступа на рассматриваемом методом контроля доступа может быть реализована в полном объеме, причем корректно.

Принципиальным отличием данного метода контроля доступа от метода дисперсионного контроля доступа является то, что запрос доступа субъекта к объекту, в том случае, когда субъекту необходимо запретить запись в объект (документировать объект), не запрещается, что не всегда можно реализовать применимо к системным объектам без нарушения корректности работы системы и приложений, а перенаправляется (право доступа на запись содержится, но доступ при этом реализуется уже к иному объекту).

Данное отличие можно позиционировать как принципальное преимущество данного метода контроля доступа, применимо к защиту системных объектов, как в части защиты системных объектов одного объекта от атак со стороны другого объекта, так и в части создания различных конфигураций для субъекта, исходя создаваемых единой для системы в целом. Его же практическое использование позволяет решать принципиально новые задачи защиты, например, реализовать виртуализацию системных средств [5], что рассмотрим далее.

Построим соответствующую модель. Если считать, что множества \( C = \{C_1, ..., C_l\} \) и \( O = \{O_1, ..., O_l\} \) – соответственно линейно упорядоченные множества субъектов и объектов доступа, а право перенаправления запросов доступа \( P \) задается, как \( C_i (O_j)O_k \) познания перенаправление запросов доступа субъекта \( C_i \) к объекту \( O_j \) в объект \( O_k \), \( C_i (O)O_k \) – отсутствие перенаправления), то разграничительная политика доступа субъектов к объектам описывается матрицей перенаправлений запросов доступа \( M \), где \( M[C, O] \) – ячейка матрицы, которая содержит право перенаправлений запросов доступа субъекта из множества \( C = \{C_1, ..., C_l\} \) к объекту из множества \( O = \{O_1, ..., O_l\} \).

В любой момент времени система контроля доступа описывается своим текущим состоянием \( Q = (C, O, M) \).
При назначении правил перенаправления запросов доступа для субъектов, а не для объектов (прямая, а не транспонированная матрица по аналогии с методом дискретного контроля доступа), объекты доступа в правилах могут назначаться масками. Маски при этом задаются и применяются следующим образом. В общем случае маска при задании вправе объекта доступа имеет следующий вид $O^*$. Например, если в правилах перенаправлений запросов доступа задать исходный объект маской $^*x$com, а объект, в который перенаправляется доступ маской $^*x$е, то любое обращение к файлу с расширением $^*x$е будет перенаправляться к файлу с тем же полным именем, но уже с расширением $^*x$ом и в имени объекта в запросе доступа будет подставляться расширение $^*x$е (пример защите от замещающих вирусов).

Ограничения данной модели состоят в том, что рассмотренным методом контроля доступа невозможно разграничить отдельные права доступа субъекта к объекту (чтение, запись, удаление, исполнение); при любом виде запроса к объекту, как на чтение, так и на запись, этот запрос будет перенаправлен в соответствии с заданными правилами. Расширить возможности рассматриваемого метода контроля доступа введением следующих правил перенаправления запросов $C(O,w)O$ означает перенаправление запросов доступа на запись (в) от субъекта $C_i$ к объекту $O_j$ в объект $O_k, C(O,w)O$ – перенаправление запросов доступа на чтение (r) от субъекта $C_i$ к объекту $O_j$ в объект $O_k, C(O,0)O$ – отсутствие перенаправления.

**Замечание.** На практике, как правило, вместо права записи обязательно разрешается и право чтения, т.е. разрешается либо только $r$, либо одновременно $r$ и $w$, поскольку для модификации файла его сначала нужно прочитать.

При использовании данных правил уже анализируется не только объект, к которому запросен доступ, но и тип запрашиваемого доступа (чтение или запись, в общем случае могут анализироваться и иные типы доступа). Соответственно, с учетом типа запрашиваемого доступа перенаправления, либо нет, запросы доступа.

Модель контроля доступа приобретает в данном случае (уже в общем случае для данного метода контроля доступа) следующий вид. Если считать, что множества $C = (C_1, ..., C_i)$ и $O = (O_1, ..., O_j)$ – соответственно линейно упорядоченные множества субъектов и объектов доступа, а $R$ конечное множество прав доступа $R = \{r, w, d\}$ (чтение, запись, удаление), право перенаправления запросов доступа $P$ задается, как $C(O,R)O$ – означает перенаправление запросов доступа, затребованных соответствующим правом доступа из множества $R$ от субъекта $C_i$ к объекту $O_j$ в объект $O_k, C(O,0)O$ – отсутствие перенаправления), то разграничительная политика доступа субъектов к объектам описывается матрицей перенаправлений запросов доступа $M$, где $M[C,O]$ – ячейка матрицы, содержит право перенаправлений запросов доступа субъекта из множества $C_i = (C_1, ..., C_i)$ к объекту из множества $O_i = (O_1, ..., O_j)$. В любой момент времени система контроля доступа описывается своим текущим состоянием $Q = (C, O, M)$.

Примечанием построение разграничительной политики доступа с использованием рассматриваемого метода контроля доступа на примере. Рассмотрим произвольную матрицу $M$ (с заданными правилами доступа методом дискретного контроля доступа):

\[
M = \begin{bmatrix}
C_1 & 0 & \cdots & 0 \\
C_2 & 0 & \cdots & 0 \\
C_i & 0 & \cdots & 0 \\
C_j & 0 & \cdots & 0 \\
\end{bmatrix}
\]

\[
O_i & O_j & \cdots & O_i \\
O_i & O_j & \cdots & O_i \\
O_i & O_j & \cdots & O_i \\
O_i & O_j & \cdots & O_i \\
\end{bmatrix}
\]

Реализуем эту же разграничительную политику доступа с использованием метода перенаправлений запросов доступа. Получим матрицу перенаправлений запросов доступа $M^*$:

\[
M^* = \begin{bmatrix}
C_1 & 0 & \cdots & 0 \\
C_2 & 0 & \cdots & 0 \\
C_i & 0 & \cdots & 0 \\
C_j & 0 & \cdots & 0 \\
\end{bmatrix}
\]

\[
O_i & O_j & \cdots & O_i \\
O_i & O_j & \cdots & O_i \\
O_i & O_j & \cdots & O_i \\
O_i & O_j & \cdots & O_i \\
\end{bmatrix}
\]

Данный пример наглядно иллюстрирует то, что данным методом контроля доступа с учетом в правилах перенаправлений типа запрашиваемого доступа может быть реализована любая разграничительная политика доступа субъектов к объектам. Это позволяет сделать вывод о том, что рассмотренный метод контроля доступа перенаправлением запросов доступа является полноценным самостоятельным методом контроля доступа субъектов к объектам и может рассматриваться не только, как метод контроля доступа, привносящий новый способ реализацию защиты системных объектов доступа, но и в общем случае, качестве альтернативы методу дискретного контроля доступа.
3.3.2.3. Технология виртуализации системных средств

Особую сложность на практике представляет задача администрирования в части реализации разграничительной политики доступа к системным объектам — к исполняемым объектам, конфигурационным объектам системы и приложениям, поскольку подобных объектов в системе достаточно много, в общем случае всех их необходимо выявить и разграничить к ним права доступа субъектов. Как выше отмечалось, именно применительно к решению этих вопросов может эффективно использоваться метод контроля доступа перенаправлением запросов доступа. Проиллюстрируем сказанного примером его практического использования.

Рассмотрим технологию виртуализации системных средств [5] — метод защиты, основанный на практическом использовании метода перенаправления запросов доступа. При этом будем исходить из того, что между субъектами доступа с использованием метода перенаправления запросов доступа могут разграничиваться права доступа субъектов к любым объектам, в том числе, к системным объектам.

Определение. Под виртуальной системой — будем понимать систему, состоящую из копий системных объектов — исполняемых файлов, и файлов конфигурации, создаваемых для субъекта доступа, работа которого изолируется в системе, на которую перенаправляются запросы доступа субъекта к оригинальной (к исходной) системе.

Замечание. Данный метод виртуализации не имеет ничего общего с технологией виртуальных машин, реализуемой с целью более эффективного использования избыточной мощности ресурсов вычислительной системы, за счет предоставления возможности запуска на одном компьютере одновременно нескольких виртуальных операционных систем.

Рассмотрим идею технологии виртуализации системных средств [5]. Пусть исходно на системном диске — на диске C, установлена ОС и приложения, и пусть в системе заведены два интерактивных пользователя User 1 и User 2, кроме того присутствует системный пользователь, работу которых требуется разделить (изолировать по доступу к системным объектам) в системе.

Создадим конн системного диска C: на дисках D: и E: — скопируем в них соответствующие файловые объекты (в том числе, скрытые файлы), чтобы достаточно просто сделать. С учетом выполненной процедуры копирования, далее будем говорить, что на диске C: установлена базовая система, на дисках D: и E: — созданные виртуальные системы.

Заметим, что виртуальные системы не обязательно создавать на отдельных дисках, можно их создать в отдельных каталогах того же диска, где установлена базовая система.

Теперь зададим правила перенаправлений запросов доступа к соответствующим файловым объектам. Для пользователя User 1 это будет правило перенаправления доступа к диску C: на диск D:, для User 2 — правило перенаправления доступа к диску C: на диск E:. Реализуя разграничительную политику доступа к файловым объектам, запретим пользователю User 1 доступ к диску E:, а пользователю User 2 — к диску D:. В результате этого получим схему виртуализации системы, приведенную на рис. 3.9.

Рис. 3.9. Схема виртуализации системы

Рассмотрим, как будет работать подобная система с виртуализацией. С базовой системой может взаимодействовать только системный пользователь, запросы к базовой системе интерактивных пользователей User 1 и User 2 перенаправляются к соответствующим созданным для них виртуальным системам.

...
Контроль доступа к компьютерным ресурсам

системам. Т.е., если пользователь User 1 запускает с базовой системы какое-либо приложение, например, браузер, реально оно будет запущено с диска D. При последующей работе приложения любое его обращение с правами пользователя User 1 к базовой системе будет перенаправляться к соответствующей виртуальной системе. При этом приложение, инсталлированное на диск C, будет работать корректно, поскольку все перенаправления его запросов доступа «прозрачны» для приложения, в частности, корректно будут сохраняться временные данные, cookies, настройки, конфигурационная информация и т.д.

Заметим, что если пользователь User 2 запускает с базовой системы какое-либо приложение, например, тот же браузер, реально оно будет запущено уже с диска E. При этом опять же это приложение, исходно инсталлированное на диск C, будет работать корректно, поскольку все перенаправления его запросов доступа «прозрачны» для приложения, но в этом случае они уже будут сохраняться на диске E.

Таким образом, получаем полную изолированность системного средства для субъектов доступа, успешная атака, осуществленная на приложение, запущенное каким-либо интерактивным пользователем, не приведет к несанкционированной модификации (или удалению элементов) как базовой системы, так и виртуальных систем иных пользователей. И что очень важно, в результате виртуализации системы мы всегда имеем доверенную операционную систему (базовую систему), с которой осуществляется корректная загрузка, и с которой может быть восстановлена виртуальная система, подвергшаяся успешной атаке, поскольку с правами интерактивного пользователя, под которыми запускаются приложения, доступ к ней не осуществляется. При этом, как видим, сложность реализации технологии виртуализации системных средств, включая настройку соответствующих разграничительных политик доступа к защищаемым объектам, минимальны.

Иллюстрация разграничительной политики доступа при виртуализации системных средств, приведена на рис. 3.10.

Замечание. В общем случае рассматриваемый метод контроля доступа применим и в рамках реализации метода сессионного контроля доступа, используемого при реализации сессионной модели контроля доступа. В этом случае различные виртуальные системные средства должны создаваться для различных режимов (сессий) обработки информации субъектами доступа (по числу этих сессий), т.е. к различным виртуальным системным средствам должны иметь доступ пользователи, которым назначаются различные метки безопасности — уровни доступа.

Каждому виртуальному системному средству (кроме исходной базовой системы) назначается метка безопасности (метка сессии). Каждому интерактивному пользователю (учетной записи) — системному, назначается метка безопасности, т.е. вся метка безопасности, т.е. метка сессии, в которой может работать пользователь.

При запросе доступа пользователя к базовой системе, выбор диспетчером доступа виртуального системного средства, в которое перенаправляется этот запрос доступа, осуществляется по совпадению метки безопасности, присвоенной пользователю, с меткой безопасности, присвоенной виртуальному системному средству. При запросе системного пользователя (у пользователя отсутствует метка безопасности), этот запрос не перенаправляется — запрос доступа осуществляется к базовой системе.
Замечание. Эта технология применима для метода сессионного контроля доступа и не применима для метода мандатного контроля доступа, характеризуемого иерархической обработкой (сравнением) иерархических меток безопасности. Это обусловливается необходимостью однозначного выбора виртуального системного средства, к которому будет перенаправляться запрос доступа по метке безопасности, назначенной интерактивному пользователю, что возможно при реализации сессионного контроля доступа, в рамках которого осуществляется ненерархическая обработка иерархических меток безопасности (метка безопасности пользователя сравнивается с совпадение с меткой безопасности объекта доступа).

Напомним, что в общем случае обработка информации в различных сессиях имеет принципиально различный уровень безопасности (что подтверждается построенной ранее молью вероятностного контроля доступа), как следствие, принципиально различается и уязвимость системных объектов — базовой и виртуальной, используемых в различных сессиях, систем.

Отметим, что в общем случае подобным образом могут разделяться между субъектами не только системные средства, но и объекты (папки), используемые для хранения обрабатываемых данных, т.е. вся разграничительная политика может быть реализована соответствующим методом контроля доступа, как дискретционным, так и сессионным (как ранее показали, мандатная схема контроля доступа в данном случае не применима).

Возможный подход к виртуализации системы состоит в ее виртуализации для субъекта доступа "процесс" в рамках реализации процессной модели контроля доступа (субъект доступа в общем случае идентифицируется трением соответствующих сущностями). Принципиальное значение это на примере, для чего рассмотрим, как, используя данный подход, изолировать работу браузера — работу с сетью. Пусть опять же на диске С: установлена базовая система, сделаем ее копию на диск D: Теперь зададим разграничительную политику — для субъекта, определяющего, как исполняемый файл интересующего нас браузера зададим правила перенаправления запросов к файловым объектам к диску C: на диск D: кроме того, запретим браузеру доступ к диску С.

В результате этого работа пользователя с различными приложениями ставится полностью "прозрачной" — под одной и той же учетной записью. Примечание базовая OS защищена от потенциальной возможных атак на соответствующих браузер: браузер может повредить или модифицировать лишь копию системы.

3.3.2.4. Оценка эффективности контроля доступа с виртуализацией

Глава 3. Принципы и методы контроля доступа к статическим объектам

В этом случае реализуется процессная модель контроля доступа, в частности, может быть реализован ее частный случай — вероятностная модель контроля доступа. Критические (с учетом вероятности наделения их вредностями свойствами) приложения могут объединяться в одну группу (в соответствующие группы), для которой (для которых) создается виртуальное системное средство. В этом случае атаки на данные приложения, вне зависимости от способа наделения критического приложения вредностями свойствами, не позволяют получить доступ как к базовой системе, так и к виртуальному системным средством, созданным для менее критических приложений.

Аналогичным образом можно решить задачу виртуализации и в отношении системного объекта доступа "реестр операционной системы".

3.3.2.4.1. Модель исходной (базовой) системы

Построим и исследуем модели контроля доступа с виртуализацией системных средств, позволяющие оценить эффективность данного метода при различных способах виртуализации (полная и частичная) системных средств [5]. При этом эффективность метода будет оцениваться применительно к различным возможным сценариям атак.

Построим модель исходной системы без использования виртуализации системных средств. Пусть операционная система OS имеется следующие субъекты (Subjects):

- sp — процесс системы;
- adm — администратор;
- u₁ — первый пользователь;
- u₂ — второй пользователь.

Замечание. В качестве субъектов доступа рассматриваем пользователей (учетные записи).

Это можно записать следующим образом: Subjects = \{sp, adm, u₁, u₂\}.

Также на OS установлены следующие объекты (Objects):

- S — системные файлы, которые содержат:
  - SE — исполняемые файлы операционной системы;
  - CF — конфигурационные файлы (различные текстовые и бинарные файлы).
Контроль доступа к компьютерным ресурсам

- $P$ - пользовательские приложения.
- $UF$ - пользовательские файлы.

$\text{Objects} = S \cup P \cup UF$, где $S = SE \cup CF$, а $P = \{p_1, \ldots, p_k\}$, где $p_i$ - установленное приложение и $i \in 1..N$, $N$ - количество приложений, установленных в системе. $SE = \{se_1, \ldots, se_j\}$, где $se_j$ - установленное системное приложение и $j \in 1..M$, $CF = \{cf_1, \ldots, cf_k\}$, где $cf_k$ - конфигурационный файл, и $i \in 1..K$, $UF = \{uf_1, \ldots, uf_l\}$, где $uf_l$ - пользовательский файл, и $l \in 1..J$.

При работе с операционной системой OS пользователь $u_i$ имеет доступ к:

- $CONF_i$ - конфигурацию первого пользователю
  - $P_i$ - подмножество программ $P$, доступное для $u_i$,
  - $CF_i$ - подмножество конфигурационных файлов $CF$, доступное для $u_i$,
  - $UF_i$ - подмножество $UF$-файлов, хранящие пользовательские настройки пользователя $u_i$.

Аналогично пользователь $u_j$ имеет доступ к:

- $CONF_j$ - конфигурацию второго пользователя.
  - $P_j$ - подмножество $P$, доступное для $u_j$,
  - $CF_j$ - подмножество $CF$, доступное для $u_j$,
  - $UF_j$ - файлы, хранящие пользовательские настройки пользователя $u_j$.

$CONF_i = P_i \cup CF_i \cup UF_i$,
$CONF_j = P_j \cup CF_j \cup UF_j$

Следует отметить, что в общем случае множества $P_i \cap P_j$, $CF_i \cap CF_j$, $UF_i \cap UF_j$, могут быть не пустыми: $P_i \cap P_j \neq \emptyset$, $CF_i \cap CF_j \neq \emptyset$, $UF_i \cap UF_j \neq \emptyset$ (используются те же программы), $CF_i \cap CF_j \neq \emptyset$ (для учетных записей). Операции над множествами выполняются тех же настройки, например, для приложений).

С учетом сказанного можем записать:

Обозначим символом $x \rightarrow A$ отношение того, что для любого объекта из $A$ субъект $x$ может читать этот объект и, если это исполняемый файл, то его можно запустить, где $x \rightarrow A$ означает, что $u \in A$, и $x \rightarrow A$ может прочесть (исполнять) $a$, где $x \in Subjects$, $A \subseteq Objects$.

Глава 3. Принципы и методы контроля доступа к статическим объектам

Обозначим символом $x \rightarrow A$ отношение того, что для любого объекта из $A$ субъект $x$ может читать и изменять этот объект, где $x \rightarrow A$ означает, что $u \in A$, $x$ может прочесть (и исполнить) $a$, где $x \in Subjects$, $A \subseteq Objects$.

Из данных определений следует, что если $x \rightarrow A$, то $x \rightarrow A$, обратное неверно.

Тогда можно записать правила доступа всех субъектов следующим образом:

```plaintext
sp \rightarrow Objects
adm \rightarrow Objects
u_i \rightarrow CONF_i \cup u_i \rightarrow SE \cup (P \setminus P_i)
```

Для каждого пользователя доступна для записи схема конфигурации и доступна для чтения и запуска статические и пользовательские приложения, установленные в системе.

Рассмотрим различные сценарии атак.

Сценарий 1. Пользователь $u_i$, является злоумышленником.

Пользователь $u_i$, может модифицировать $CONF_i$, которая состоит из $P_i$, $UF_i$, $CF_i$, так как $P_i \cap P_j \neq \emptyset$, $CF_i \cap CF_j \neq \emptyset$, $UF_i \cap UF_j \neq \emptyset$, то есть могут существовать программы, конфигурационные файлы и пользовательские файлы, общие для нескольких пользователей, в результате пользователь $u_i$ может изменить настройки пользователя $u_j$.

Сценарий 2. Происходит заражение системного исполняемого файла определенной системы.

Пусть заражён некоторый системный исполняемый файл $se \in SE$. Так как $u_i \rightarrow SE$ и $u_i \rightarrow SE$, то $u_i \rightarrow se$, и $u_i \rightarrow se$, В результате оба пользователя оказываются под воздействием заражения и используют заражённый файл $se$.

Сценарий 3. Происходит заражение пользовательского исполняемого файла.

Пусть заражён некоторый пользовательский исполняемый файл $p \in P$. Если только один пользователь имеет доступ к этому файлу, то есть $p \in P \setminus P_i$, или $p \in P \cap P_i$, тогда только один из пользователей $u_i$ и $u_j$, соответственно, оказываются под влиянием заражённого файла. Если же исполняемый файл доступен обоим пользователям, то есть $p \in P \cap P_i$, то оба пользователя оказываются под воздействием заражённого файла.
Сценарий 4. Кража информации из конфигурационных файлов.

Пусть пользователь $u_1$ запускает исполняемый файл $sc_1 \in SE\cup P_1$, который является зараженным. Тогда приложение $sc_1$ выполняется с правами пользователя $u_1$ и имеет доступ к следующим файлам: $se, \Rightarrow CONF, (где CONF = P, \cap CF, \cup UF)$ и $e_1 \Rightarrow SE \cup (P \setminus P_1)$. Так как $P \cap P_1, CF \cap CF, UF_1 \cap UF_2$ могут быть не пустыми, может произойти кража данных не только одного пользователя, но и другого пользователя, не запускавшего зараженный файл.

Сценарий 5. Оказывается заражен исполняемый файл, выполняемый системным процессом $sp$ или администратором.

Пусть файл $f \in SE$ является зараженным, далее он запускается системным процессом или администратором. Так как и администратор и системный процесс имеют полный доступ к системе (sp $\Rightarrow$ Objects, adm $\Rightarrow$ Objects), то вся система оказывается доступна для процесса, запущенного из файла $f$.

3.3.2.4.2. Модели системы с виртуализацией

Полная виртуализация системного средства

При реализации метода виртуализации системного средства исходная модель системы изменяется следующим образом [5].

Субъекты системы остаются теми же. Subject = {sp, adm, $u_1$, $u_2$}. Однако так как виртуализация системы предполагает создание копий на которые переправляются запросы пользователя, то множество объектов операционной системы меняется. Для каждого пользователя создается своя копия системы:

Objects$_{u_1}$ = Objects $\cup$ Objects$_2$, $u_2$ = множества всех объектов, установленных на операционной системе.

Objects = S $\cup$ P $\cup$ UF = множество объектов, изначально установленных в системе. В случае полного копирования файлов операционной системы, копии этих файлов попадают во множество файлов, доступных для каждого из пользователей $u_1$ и $u_2$. В случае частичного копирования, только некоторое подмножество этих файлов попадает во множество доступных файлов для пользователей $u_1$ и $u_2$. Здесь подмножества определяются таким же образом, как и без использования виртуальных систем, то есть $S = SE \cup CF$, а $P = \{p_1, p_2\}$, где $p_1$ - установленное приложение i $\in 1.N$, $N$ = количество приложений, установленных в системе. $SE = se_1, se_2$, где se_1 = установленное системное приложение i $\in 1.M$, CF = cf_1, cf_2, где cf_1 = конфигурационный файл, i $\in 1.K$, $UF = uf_1, uf_2$, где $uf_1$ = пользовательский файл, i $\in 1.J$.

Глава 3. Принципы и методы контроля доступа к статичным объектам

Рассматриваем случай полного копирования файлов системных файлов (полная виртуализация системного средства). При этом множество объектов, доступных для пользователей $u_1$ и $u_2$, примут следующий вид.

Обозначим = отношение того, что один объект является копией другого. Следуя за образом: пусть объект $o$, был создан операцией копирования объекта $o_1$, тогда $o = o_1$.

$Object_{1} = S \cup P \cup UF$, где $S = S, S = P, P = UF, P = UF$.

$Object_{2} = S \cup P \cup UF$, где $S = S, S = P, P = UF, P = UF$.

Здесь $S = SE \cup CF, t \in \{1,2\}$ - индекс пользователя (для $u_1: t = 1$, для $u_2: t = 2$), а $P = \{p_1, p_2\}$, где $p_1$ - установленное приложение i $\in 1.N$, N = количество приложений, установленных в виртуальной системе для пользователя t. $SE = se_1, se_2$, где se_1 = установленное системное приложение i $\in 1.M$, CF = cf_1, cf_2, где cf_1 = конфигурационный файл, i $\in 1.K$, $UF = uf_1, uf_2$, где $uf_1$ = пользовательский файл, i $\in 1.J$.

Следующим этапом организации работы в виртуальной системе являются задание правил перенаправления (будем обозначать =). Пусть выполняется перенаправления всех операций чтения i/или записи файлов из множества O, на их копию O$_1$ | O$_2$ = O, для пользователя $u_1$, тогда это записывается следующим образом:

$O_1 \Rightarrow O_2$, где $L = \{r\}$ или $L = \{w\}$, или $L = \{r,w\}$ - множество типов операций, $r$ - операция чтения, $w$ - операция записи.

Тогда можно записать все правила перенаправления для исходной системы в виртуальные системы:

$Objects = [r,w] \Rightarrow Objects_1$.

$Objects = [r,w] \Rightarrow Objects_2$.

Для каждого из пользователей $u_1$, $u_2$, выделяется своя копия системы, и все запросы пользователя на чтение и запись перенаправляются на неё.

Учитывая, что перенаправление выполняется до того, как выполняется проверка прав доступа к объекту, для пользователей $u_1$, $u_2$ можно назначить следующие права:

$u_1 \Rightarrow Objects_2$, $u_2 \Rightarrow Objects_2$.

Рассмотрим сценарии атак применительно к виртуальным системам с полной виртуализацией.
Сценарий 1. Пользователь u, является злоумышленником.
Пользователь u, может модифицировать множество файлов Objects, которое является копией множества Objects. Однако конфигурация пользователя u, остаётся недоступной для пользователя u, так как Objects ∩ Objects = ∅. В связи с обычным использованием системы применение виртуального систем позволяет обнаружить пользователей системы от злоумышленной модификации одним из пользователей чужих конфигураций.

Сценарий 2. Происходит заражение системного исполняемого файла операционной системы.
Пусть заражён некоторый системный исполняемый файл se1 ∈ SE, SE ⊂ Objects. Види того, что u1,∈ Objects, a u2,∈ Objects, и ни один из пользователей не имеет доступа к Objects, то для каждого из пользователей u1, и u2, имеется своя копия файла se1, а именно: se1, = se, и se2, = se, где se,∈ SE, a se2,∈ SE, a SE ⊂ Objects, SE ⊂ Objects. И так как правила перенаправления настроены таким образом, что при обращении обоих пользователей к любому файлу из Objects фактически доступ достигается к файлу из копии, специально созданной для каждого из них, то каждый пользователь обращается к файлам se1, и se2, которые не были подтверждены заражению. Каждый из пользователей имеет доступ только к своей копии файла se; u1,→ se1, и u2,→ se2,.

В результате ни один из пользователей не оказывается под воздействием заражения, так как они используют копии файла se. В сравнении с использованием оригинальной системы, пользователи оказываются защищенным от данного типа угроз.

Сценарий 3. Происходит заражение пользовательского исполняемого файла.
Пусть заражён некоторый пользовательский исполняемый файл p,∈ P. Так как P ⊂ Objects, то ни один из пользователей u1, u2, не имеет доступа к файлу p,. Види включенного перенаправления, каждый из них использует свою копию этого файла p, = p, = p,;

Сценарий 4. Кража информации из конфигурационных файлов.
Пусть пользователь u, запускает исполняемый файл, который является зараженным, e,∈ SE, U P, SE ⊂ Objects, P ⊂ Objects. Тогда приложение e, выполняется с правами пользователя u, и имеет доступ к следующим файлам: e, = Objects. Таким образом, под воздействием оказываются только файлы пользователя u, файлы остальных пользователей, например пользователь u2, недоступны, так как они хранятся в Objects, к которому у пользователя u, нет доступа. Таким образом, виртуальные системы устраняют возможность кражи чужой конфигурационной информации.

Сценарий 5. Оказывается заражён исполняемый файл, выполняемый системным процессом sp или администратором.
Пусть файл f ∈ SE является заражённым. Он запускается системным процессом или администратором. Как и администратор и системный процесс имеют полный доступ к системе (sp⇒ Objects, adm⇒ Objects), то вся система оказывается доступна для процесса, запущенного из файла f.

Из сказанного можем сделать вывод о целесообразности перенаправления и запросов доступа системного пользователя при его обращении к виртуальным системам, создаваемым для интерактивных пользователей.
Частичная виртуализация системного средства

Частичная виртуализация предполагает создание коней с последующим перенаправлением к ним запросов доступа субъектов не всего системного средства, а некоторой его части.

При реализации частичной виртуализации модель виртуальной системы изменяется следующим образом [5].

Субъекты системы остаются теми же: Subjects = {sp. adm, user}. Однако так как виртуализация системы предполагает создание коней части системы, на которые перенаправляются запросы доступа пользователей, то множество виртуализируемых объектов системного средства меняется. Для каждого пользователя создается своя копия части системы. Пусть:

```
Objects_{sp} = Objects \cup Copy_{1} \cup Copy_{2} \cup \ldots
```

Множество объектов, изначально установленных в системе, определяется аналогично предыдущим случаям: Objects = S \cup P \cup UF. Так как выполняется частичное копирование (виртуализация), то только часть файлов попадают в копию для каждого из пользователей с и с'. Какие файлы переносят в копию, а какие - нет, определяется требованиями к изоляции системных объектов для пользователей.

Copy_{1}, Copy_{2}, ..., Copy_{n} - все копии, созданные всеми пользователями.

Опишем подмножества файлов от изначальной системы, которые копируются для пользователей u и u', соответственно, следующим образом:

- Copy_{u} = S_{u} \cup P_{u} \cup UF_{u}.
- Copy_{u'} = S_{u'} \cup P_{u'} \cup UF_{u'}.

Также стоит отметить, что копируемые файлы могут быть как совершенно различными, так и совпадать для разных пользователей, то есть Copy_{u} \cap Copy_{u'} \nsubseteq \emptyset.

Тогда частичные копии для пользователей u и u' можно описать так:

```
Copy_{u} = S_{u} \cup P_{u} \cup UF_{u}, \text{ где } S_{u} = S_{u} \cap P_{u} = P_{u} \cup UF_{u} = UF_{u}.
```

Также для использования виртуальных систем следует назначить правила перенаправления. Перенаправление осуществляется таким образом, что все операции пользователя к подмножеству скопированных файлов оригинальной системы фактически адресуются к копии, созданной для этого пользователя. Правила перенаправления для этого случая записем следующим образом:

```
Copy_{u} \rightarrow Copy_{u}.
```

Учитывая, что перенаправление выполняется до того, как выполняется проверка прав доступа к объекту, для пользователей u1 и u2 назначаются следующие права:

- u1 \rightarrow Copy_{1} \cup CF_{1} \cup P_{1} \cup UF_{1} \cup UF_{2} \cup SE_{1} \cup SE_{2} \cup (P_{1} \setminus P_{2}) \cup P_{2} \cup SE_{2} \cup SE_{1} \cup (P_{2} \setminus P_{1}) \cup P_{1}
- u2 \rightarrow Copy_{2} \cup CF_{2} \cup P_{2} \cup UF_{2} \cup SE_{2} \cup SE_{1} \cup (P_{2} \setminus P_{1}) \cup P_{1}

Оригинальные права пользователя сохраняются с тем лишь отличием, что доступ к скопированным файлам осуществляется к копии и соответственно, оригинальные файлы становятся недоступны для пользователя. Тем не менее, пользователю добавляются права на полный доступ к специально созданной для него частичной копии.

Рассмотрим сценарий атак применительно к виртуальным системам с частичной виртуализацией.

**Сценарий 1. Пользователь u, является злоумышленником.**

Пользователь u, может модифицировать множество файлов Copy_{u}, которое является копией множества Objects. Вся конфигурация пользователя u, скопированная для виртуальной системы, остаётся недоступной для пользователя u, так как Copy_{u} \cap Copy_{u'} = \emptyset, даже если были скопированы однаковые файлы. Тем не менее, если файлы, которые не были скопированы пользователем u', совпадают с файлами, которые не присутствуют в виртуальной системе пользователя u, то есть (CF_{1} \setminus CF_{u}) \cap (CF_{2} \setminus CF_{u'}) = \emptyset или (P_{1} \setminus P_{u}) \cap (P_{2} \setminus P_{u'}) = \emptyset или (UF_{1} \setminus UF_{u}) \cap (UF_{2} \setminus UF_{u'}) = \emptyset, то пользователь u, сможет модифицировать файлы из пересечения, т.е. частичная копия позволяет устранить пересекающиеся части конфигурации пользователей, остальная часть системы остаётся незащищенной от такой угрозы.
Сценарий 2. Происходит заражение системного исполняемого файла операционной системы.

Пусть заражён некоторый системный исполняемый файл $sc \in SE$, $SE \subseteq Objects$. Если $sc \in Objects \setminus Copy_u$, или $sc \in Objects \setminus Copy_r$, то есть этот файл не содержится во множестве файлов, скопированых для одного или обоих пользователей, то соответственно один или оба пользователя используют заражённый файл. Однако если файл $sc$ был скопирован для виртуальной системы пользователя, то он уже использует оригинальный, не подвергнутое заражению копию этого файла. Таким образом, если пользователи используют виртуальные системы с частичной конфигурацией и оригинальная версия файла сохранена в этой копии, то заражение исходного файла никак не влияет на работу пользователей. Однако если файл не был скопирован, то пользователи будут использовать заражённый файл.

Сценарий 3. Происходит заражение пользовательского исполняемого файла.

Пусть заражён некоторый пользовательский исполняемый файл $p_r \in P$. Расмотрим данный сценарий для пользователя $u_r$. Для пользователя $u_r$ можно записать аналогичные утверждения с изменением рассматриваемых множеств доступных файлов, заменив индексы в вырождениях. Аналогично предыдущему сценарию, необходимо определить, принадлежит ли файл $p_r$ множеству скопированных файлов пользователя из оригинальной системы $P_{cr}$. Если $p_r \in P_{cr}$, то существует файл $p_{cr}$, такой, что $p_r = p_{cr}$, принадлежит частичной копии пользователя $u_r \equiv Copy_u$, и, в частности, множеству пользовательских приложений $P_{cr}$ и выполняется $p_{cr} = p_r$. Ввиду включенного перенаправления пользователя $u_r$ использует файл $p_{cr}$ при обращении к $p_r$.

Следовательно, пользователь не оказывается под влиянием заражения.

Если же $p_r \notin P_{cr}$, то пользователь $u_r$ использует заражённый файл, так как перенаправление в виртуальную систему не выполняется для этого файла. Таким образом, пользователь не подвержен риску атаки, если зараженные файлы были ранее скопированы в виртуальную копию.

В другом варианте этого сценария зараженный файл находит в одной из частичных копий системы. Пусть заражён пользовательский исполняемый файл $p_r$ из копии $Copy_r$, одного из пользователей $u_r$; $p_{cr} \in P_{cr}$, $P_{cr} \subseteq Copy_r$, $p_{cr} = p_r$, где $p_r \in P$, $P \subseteq Objects$. Единственный пользователь, который имеет доступ к этому файлу — это $u_r \rightarrow P_{cr}$. Так как пользователь $u_r$ не имеет доступа к файлу $p_r$ и либо использует оригинальный файл $p_r$, либо использует свою копию $P_{cr}$, $p_{cr} = p_r$, то он не имеет доступа к зараженному файлу $p_r$. Таким образом, виртуальные системы локализуют заражение и при частичном копировании, предотвращая распространение заражения на других пользователей системы.

Сценарий 4. Кража информации из конфигурационных файлов.

Пусть пользователь $u$, запускает исполняемый файл, который является зараженным. Возможны два случая, когда $f \in SE$, $U \in P$, $SE \subseteq Objects$. Если $f \in Objects \setminus Copy_r$, то есть $f$ запускается из копии системы, либо когда $f \in SE \setminus U \in P$, $SE \subseteq Objects$, $P \subseteq Objects$. В любом из этих случаев приложение $f$ выполняется с правами пользователя $u$ и имеет доступ к следующим файлам: $f \rightarrow Copy_u \cup CF_r \setminus CF_u \cup P \setminus P_r \cup UF_r \setminus UF_u$ и $u \rightarrow SE \setminus SE_u \cup (P \setminus P_r)$. В результате этого возможны следующие сценарии:

Сценарий 5. Оказывается заражён исполняемый файл, выполняемый системным процессом $sr$ или администратором.

Пусть файл $f \in SE$ является зараженным. Он запускается системным процессом или администратором. Как и администратор и системный процесс, $sr$ и $P \subseteq Objects$ имеют полный доступ к системе ($sr \in Objects \setminus adm \equiv Objects \cup$), и все системы оказываются доступны для процессов, запущенных из файла $f$. Таким образом, возможны следующие сценарии реализации перенаправления запросов доступа и для системного пользователя, осуществляемых в виртуальных копиях частей систем, создаваемых для интерактивных пользователей.

Как видим, реализация частичной виртуализации системных средств позволяет найти компромисс между объемом создаваемых копий виртуализируемой системы и ее защищенностью системных объектов и пользовательских конфигураций. В качестве рекомендации по реализации частичной виртуализации можно рассмотреть перенос в виртуальную систему все системные объекты и ключевые пользовательские приложения, а также конфигурационные файлы, которые находятся в общем доступе для нескольких пользователей.

Замечание. Приведены модели виртуализации системных средств применительно к системному доступу пользователя, позволяющие оценивать эффективность защита применительно к различным сценариям...
3.3.3. Требования к правилам контроля доступа в части защиты системных файловых объектов

Как ранее говорили, системные файловые объекты, относящиеся к статичным, представляют собой исполняемые файлы системы и приложений и файлы настройки (конфигурации) системы и приложений. Кроме того, приложения могут создавать служебные файлы (временные, для хранения все возможной служебной информации, например, идентификаторов и паролей и т.д.).

Сформулируем требования к правилам контроля доступа, выполнение которых позволит построить безопасную систему. В данном случае, в первую очередь, речь следует вести о безопасности системы относительно права исполнения (x), поскольку системные объекты атрибуты не предназначены для сохранения в них обрабатываемой на компьютере информации.

Лемма 3.10. Не допустимо разрешение пользователям модифицировать (и удалять) исполняемые файлы системы и приложений, не допустимо разрешение пользователям создавать разрешенные для запуска новые исполняемые файлы системы и приложений. При реализации данного требования система безопасна относительно права исполнения (x).

Доказательство. Не допустимо одновременно предоставить право на запись (удаление) и исполнение (w (d), x). Поскольку исполняемые файлы системы и приложений могут быть разрешены в системе исполнить, к ним должно быть запрещено право записи, что предотвращает утечку права исполнения (x) и запрещено право удаления, что также предотвращает утечку права исполнения (x). Лемма 3.10 доказана.

Лемма 3.11. Не допустимо разрешение пользователям модифицировать (удалять) файлы настройки системы и приложений. При реализации данного требования система безопасна относительно права исполнения (x).

Доказательство. При модификации (удалении) файлов настройки системы и приложений, системные процессы и приложения могут либо не запускаться, либо приобретать иные свойства, что представляет собою угрозу безопасности системы (x). Следовательно, пользователем модифицировать (удалять) файлы настройки системы и приложений нецелесообразно. Лемма 3.11 доказана.

Лемма 3.12. Не допустимо разрешение доступа одним пользователем к служебным файлам, создаваемым приложением при работе с ним другим пользователем. При реализации данного требования система безопасна относительно права чтения (r) и записи (w).

Доказательство. При разрешении пользователям к служебным файлам, создаваемым приложением при работе с ним другим пользователем, открывает возможность системного объекта файла нарушить все ограничения безопасности, что также приведет к утечке прав записи (w) и/или чтения (r).

Теорема 3.3. Метод контроля доступа с принудительным управлением потоками информации позволяет корректно реаланизовать контроль доступа в части защиты системных файловых объектов в безопасной системе при условии реализации диспетчером доступа правил управления потоками информации (информационными потоками), сформулированных в Леммах 3.10 – 3.12.

Доказательство. Теорема 3.3 доказывается доказательством Лемма 3.10 – 3.12, в которых сформулированы требования, реализация которых обеспечивает предотвращение утечки права исполнения (x). Теорема доказана.
А вот здесь возникает крайне важный вопрос. Что будет, если системой зашита информации выполнить требования Теоремы 3.3. для приложения в полном объеме, к чему это приведет?

Из опыта. Часть приложений сохраняет при этом свою работоспособность (вполне, отключаясь отдельные их возможности). А ряд приложений не сможет корректно функционировать. Как следствие, здесь встаёт выбор перед потребителем, либо отказаться от подобного небезопасного приложения, заменив его каким-либо аналогом, либо, оставить данное приложение в эксплуатации, понимая при этом, что система не будет безопасна.

3.3.4. Общие требования к созданию потенциально безопасных систем и приложений

Вообще были сформулированы требования, реализация которых обеспечивает корректное и эффективное решение задачи защиты в современных условиях (на их проявлении мы останавливались в первой главе).

Однако, еще в первой главе, а затем, в предыдущем параграфе, мы останавливались на проблемах, ограничивающих возможность реализации эффективной разграничительной политики доступа к ресурсам, связанных с построением и функционированием современных ОС и приложений, как следствие, на ограничениях в построении безопасной системы.

Опираясь, на сформулированные требования к построению безопасной системы, реализация которых обеспечивает корректное и эффективное решение задачи защиты в современных условиях, сформулируем требования к созданию потенциально безопасных систем и приложений.

Определение. Под потенциально безопасными системами и приложениями будем понимать соответствующие программные средства, позволяющие реализовать для них корректную разграничительную политику доступа к ресурсам, т.е. обеспечивающие корректность своего функционирования, при реализации средством защиты сформулированных требований к построению разграниченной политики доступа к ресурсам.
Глава 3. Принципы и методы контроля доступа к статическим объектам

Внимание, см. рис. 3.12, поскольку информативно подобное представление создаваемой разграничительной политики доступа субъектов к объектам. Выбрав необходимого субъекта (профиля), см. рис. 3.12, мы сразу (в одном окне) увидим все заданные для него правила доступа – к каким объектам, какие права доступа разрешены/запрещены субъекту.

Теперь, что касается реализации разрешительной разграничительной политики. Поскольку в общем случае может понадобиться и запретительная политика, например, при совместной работе с мандатным контролем доступа (о нем будем говорить в следующей главе, применимо к решению задачи защиты обрабатываемых данных), разрешительная политика должна настраиваться. Реализуется это следующим образом. Создается субъект доступа «Любой», см. рис. 3.13, и объект доступа «Все файловые объекты», см. рис. 3.14.

Рис. 3.11. Интерфейс назначения прав доступа субъекта к файловому объекту

Рис. 3.12. Интерфейс, отображающий заданные правила доступа субъекта к файловым объектам

Рис. 3.13. Создание субъекта доступа «Любой»

Рис. 3.14. Создание объекта доступа «Все файловые объекты»
Настраивается соответствующая разграничивительная политика — всем субъектам доступа запрещается любой доступ ко всем файловым объектам, что проиллюстрировано на рис. 3.15, которая отображается в интерфейсе, представленном на рис. 3.16.

Рис. 3.15. Иллюстрация задания разрешительной разграничительной политики

Рис. 3.16. Отображение разграничительной политики для субъекта «Любой» — «Все пользователи»

Как видим из рис. 3.16, последняя запись в интерфейсе (наименее точный указатель) задает запрет любого доступа любого субъекта к любому объекту. Если запрос доступа подпадает под правило, располагаемое выше, будет действовать это правило, если запрос доступа не подпадает ни под одно правило, с более точным заданием субъектов и/или объектов доступа, в запросе доступа будет отказано.

Отметим, и это крайне важно. В данном случае (при данном способе задания и хранения, в качестве отдельного объекта доступа, правил доступа — правила не присваиваются в качестве атрибутов объектам, а задаются для субъектов доступа) реализуется разрешительная разграничительная поли-

Глава 3. Принципы и методы контроля доступа к статическим объектам

Замечание. Поскольку методы контроля доступа к статическим объектам для разнородных объектов реализуются по одним и тем же изложенными ранее принципам, в частности, могут использоваться маски для создания в системе защиты субъектов и объектов доступа, изложенный подход к реализации разрешительной разграничительной политики доступа, может и должен быть реализован в отношении любого объекта, к которому контролируются и разграничиваются права доступа субъектов.

Настройки OC и приложений (системные объекты) хранятся не только в файле, но и в реестре OC. Для контроля (разграничения) доступа к объектам реестра используются те же принципы, что и рассмотренные ранее, отличие составляет лишь объекты доступа (соответственно, ветвь реестра, ключ реестра или масса) и набор назначаемых прав доступа.

Интерфейс задания прав доступа субъекта (профиля) к объекту реестра OC представлен на рис. 3.17, интерфейс, отображающий заданные правила доступа субъекта (профиля) к объектам, на рис. 3.18.

Рис. 3.17. Интерфейс назначения прав доступа субъекта к объекту реестра OC
3.5. Обеспечение замкнутости программной среды

Ранее, говоря об идентификации и аутентификации субъекта доступа «процесс», мы упоминали об обеспечении замкнутости программной среды, как основного метода идентификации и аутентификации субъекта доступа «процесс», и об обязательности решения соответствующей задачи защиты в современных средствах защиты.

Определение. Обеспечение замкнутости программной среды — это отдельная, самостоятельная задача защиты, решение которой направлено на локализацию среды исполнения на компьютере и/или для отдельных субъектов доступа (пользователей и/или процессов).

Определение. Локализация среды исполнения — это реализация возможности запуска на защищенном компьютере и/или отдельными субъектами (пользователями и/или процессами) только определенных санкционированных программ с предотвращением любых способов несанкционированной модификации или удаления их исполняемых файлов.


3.5.1. Метод обеспечения замкнутости программной среды заданием списка исполняемых файлов, разрешенных на выполнение

Данный метод сегодня наиболее часто используется на практике и состоит в том, что на компьютере в целом и/или для каждого пользователя (в общем случае субъекта) отдельно списков процессов (исполняемых файлов), которые ему разрешено запускать. Соответственно объект доступа в матрице доступа к исполняемым файлам представляет собой список исполняемых файлов.

Как утверждается в Лемме 3.1, недопустимым является установление для одного и того же файлового объекта разрешения на исполнение и разрешение на запись/модификацию — возможно либо исполнение, либо запись/модификация файлового объекта. Это обусловливается тем, что, если для одного и того же файлового объекта одновременно разрешено разрешение на исполнение и разрешение на запись/модификацию, то в этом случае можно записать собственный исполняемый файл, а после чего его запустить. Как следствие, замкнутость программной среды корректную идентификацию и аутентификацию субъекта доступа «процесс» обеспечить не сможет. При этом система небезопасна, так как происходит утечка права исполнения (x).
Следствие. Если некоторое приложение требует для корректного своего функционирования для одного и того же файлового объекта разрешения на исполнение и разрешение на запись/модификацию, целесообразно отказаться от использования подобного приложения, т.к. его использование несет в себе крайне серьезную угрозу технологической безопасности.

Корректность реализации данного метода защиты связана с предотвращением явных и скрытых каналов модификации списков санкционированных исполняемых файлов, а также запуска исполняемых файлов, не входящих в список.

Требования к корректности реализации:

- Исполнимый файл должен быть задан с указанием его полного пути, что предотвращает возможность запуска несанкционированного процесса с таким же именем из другого места. При этом пользователю должен быть запрещен запуск программ из тех объектов, для которых невозможно гарантированно предотвратить модификацию (удаление) исполняемых файлов; с внешних накопителей, из сети и т.д.;

- К исполняемым файлам должен быть запрещен доступ пользователей «на запись» и «на модификацию», что предотвращает возможность заменить вместо легальных исполняемых файлов, исполняемые файлы несанкционированной программы;

- К исполняемым файлам, разрешенным на запуск в системе, должны быть отнесены не только исполняемые файлы соответствующих приложений, но и исполняемые файлы системных процессов, запуск которых необходим для корректной работы пользователя с системой и приложениями.

Очевидный недостаток данного метода обеспечения замкнутости программной среды — это сложность настройки и последующей эксплуатации (изменение настроек при изменении среды исполнения — санкционированной установки и удаления программ) средства защиты.

3.5.2. Метод обеспечения замкнутости программной среды заданием папок с исполняемыми файлами, разрешенными на выполнение

Для метода обеспечения замкнутости программной среды, опирающегося на разграничения, задаваемые списками разрешенных к запуску процессов, необходимо сопоставлять с системой, либо с каждым пользователем (в обще

щем случае и для системы, и для пользователей) свой собственный список разрешенных к запуску программ. Такой подход интуитивно понятен, но имеет существенные недостатки. Главным из них является необходимость перечислять в этих списках все процессы (в том числе, и системные), разрешенные на запуск, в том числе и эти процессы, запускаемые уже разрешенными процессами. В результате этого списки разрешенных процессов становятся громоздкими. Сильно усложняется администрирование подобной системы защиты, особенно при установке новых программ и удалении существующих, что связано с необходимостью модификации настроек.

Рассмотрим альтернативный подход к решению задачи защиты [8].

Обеспечить замкнутость программной среды можно не непосредственно заданием списков разрешенных к запуску процессов (чтобы можно запускать), а областью дискового пространства (папками), откуда можно запускать процессы. Другими словами, для системы в целом, а при необходимости, и для каждого пользователя (в общем случае, субъекта), должны быть выделены папки (каталоги), откуда им разрешается запуск программ — для этих каталогов должно быть разрешено только «Исполнение». Причем в данные каталоги должен быть запрещен доступ «на запись» и «на модификацию» (включая удаление), с целью предотвращения возможности внесения в данные каталоги несанкционированных — не разрешенных к запуску, исполняемых файлов, а также с целью предотвращения возможности несанкционированной модификации и удаления расположенных в этих каталогах исполняемых файлов. При этом список разрешенных к запуску исполняемых файлов определяется набором программ, инсталлированных администратором в каталоги, откуда разрешен их запуск.

Например, к подобным каталогам для OC Windows обязательно должны быть отнесены Windows (где располагаются исполняемые файлы системных процессов) и Program Files (либо некая его часть).

Таким образом, могут быть сформулированы следующие требования к корректности реализации:

- Исполнимые файлы должны быть доступами полного пути каталогов, из которых разрешен их запуск, что предотвращает возможность запуска несанкционированного процесса с таким же именем исполняемого файла из другого места. При этом пользователю должен быть запрещен запуск программ из тех объектов, для которых невозможно гарантированно предотвратить модификацию исполняемых файлов; с внешних накопителей, из сети и т.д.;
Контроль доступа к компьютерным ресурсам

- К каталогам с исполняемыми файлами, разрешенными на запуск, должен быть запрещен доступ «на запись» и «на модификацию» (удаление), что предотвращает возможность разместить в них исполняемый файл несанкционированной программы;
- К каталогам с исполняемыми файлами, разрешенными на запуск, должны быть отнесены не только каталоги с исполняемыми файлами соответствующих приложений (в частности, Program Files), но и каталог с исполняемыми файлами системных процессов – системный диск (Windows), запуск которых необходим для корректной работы пользователя с системой и приложениями.

Итак, достоинством данного метода является серьезное упрощение администрирования средства защиты. При этом, по сравнению с ранее рассмотренным методом обеспечения замкнутости программной среды, в данном случае реализуется защита от модификации соответствующих системных файлов, а не только исполняемых. Другими словами, реализуются куда более широкие возможности защиты.

Однако у этого метода присущи и недостаток, состоящий в том, что при реализации соответствующей разграничительной политики доступа возможны ситуации, когда некоторые приложения не смогут функционировать (корректно функционировать).

3.5.3. Техническое решение

Техническое решение проиллюстрируем, применимко к методу обеспечения замкнутости программной среды созданием папок с исполняемыми файлами, разрешенными на запуск. Практическая реализация метода обеспечения замкнутости программной среды созданием списков исполняемых файлов аналогична, соответственно другим способом при этом создаются объекты доступа, к которым разрешается право доступа на исполнение.

Настройки замкнутости программной среды для пользователей (в профиль «Все пользователи» включены пользователи, созданные в системе защиты, в том числе и системные), приведены на рис. 3.19. Настройки замкнутости программной среды для системных пользователей (в первую очередь, это пользователь System), на рис. 3.20.

Как видим из рис. 3.19, всем пользователям мы разрешили запуск процессов только из двух каталогов, заданных с использованием переменных среды окружения, что позволит легко тиражировать данные настройки, либо установить их в средстве защиты по умолчанию.

Системным же субъектам мы, наоборот, разрешили полный доступ к этим каталогам, но одновременно предотвратили доступ к каким-либо иным файловым объектам.

С учетом того, что при таких настройках крайне критичным становится получение системных прав, предоставим возможность доступа к защищенным каталогам при несанкционированном получении пользователем системных прав. С этой целью заведем субъект «Полный запрет», см. рис. 3.21 а) (характеризуемый сменой любого первичного идентификатора на эффективный идентификатор «System» для любого процесса – маска ":*"), и запретим ему (для профиля «Запрос доступа», в который включен данный субъект) какой либо доступ к какому-либо объекту файловой системы, см. рис. 3.21 б).

а) Заведение пользователя «Полный запрет»
Глава 3. Принципы и методы контроля доступа к статическим объектам

Ради справедливости отметим, что данное решение (приведенная разграниченная политика доступа) апробировано на практике. Отказы в доступе к системным файловым объектам от различных приложений следуют в массовом порядке, но некорректной работы системы и приложений нами отмечено не было. Однако не будем утверждать, что это справедливо в общем случае.

Рис. 3.22. Отказы в доступе при запуске Internet Explorer

Замечание. Здесь и далее мы приводим лишь некоторые примеры настроек, лишь иллюстрирующих потенциальные возможности рассматриваемых методов защиты и технических решений.

3.5.4. Расширение функциональных возможностей метода обеспечения замкнутости программной среды

Расширение функциональных возможностей данного метода защиты состоит в том, что с включением в схему контроля доступа субъекта доступа «процесс», и в качестве субъекта доступа, и в качестве объекта доступа при
настройке разграничительной политики доступа может рассматриваться сущность «процесс» (используемый файл). В этом случае можно разграничивать права доступа для процессов на запуск процессов, т.е. можно обеспечивать замкнутость программной среды не только на уровне контроля запуска санкционированных процессов в системе или пользователями, а уже на уровне контроля последовательностей запусков процессов, т.е. управлять потоками исполнения объектов.

3.6. Примеры решения некоторых актуальных задач защиты

Предложенной постановкой задачи защиты информации от несанкционированного доступа в общем, которая приведена в первой главе, определяется то, что методы защиты информации от несанкционированного доступа должны использоваться не только для формирования режимов обработки информации субъектами доступа (ролевая и сессионная модели контроля доступа), но и должны реализовывать защиту от актуальных угроз атак, по средством нивелирования угроз безусловных и условных технологических уязвимостей.

Рассмотрим примеры использования с этой целью методов контроля доступа к статическим объектам (существенно, что по ним не приходятся речь пойдет о динамическом контроле доступа).

Замечание. Данная работа – это не инструкция по настройке механизмов защиты под решение конкретных задач. Здесь мы не будем рассматривать все множество актуальных на сегодня задач защиты, связанных с контролем доступа к системным файловым объектам, ограничившись лишь рассмотрением некоторых примеров, иллюстрирующих потенциальные возможности, а также некоторые недостатки реализации защиты на основе контроля доступа к системным файловым объектам.

3.6.1. Защита от вредоносных программ

На сегодняшний день трудно представить себе более актуальную задачу защиты. Наверное, мало кто из наших читателей, не столкнулся в жизни с этой проблемой.

Актуальность задачи защиты от вредоносных программ возрастает из года в год. В [66] вместе с попыткой выявления причин сложившейся ситуации с защитой от вредоносного ПО, приведен прогноз роста вредоносного ПО на ближайшие годы. Несмотря на то, что приводимое исследование и прогнозы были проведены несколько лет назад и несколько потеряли актуальность, нам, в рамках данной книги, будет интересна методика, по которой проводился анализ. Поэтому мы остановимся на этом исследовании подробнее. По мнению автора [66], в 2013 году уже будет выпущено порядка 50 млн новых вредоносных программ, а в среднем 136 тысяч ежедневно, а в 2015 году количество новых вредоносных программ может превысить 200 млн, см. рис. 3.23 а).

Замечание. Интересно отметить соответствие сделанных прогнозов с реальной ситуацией. Так, согласно отчета компании Panda Security[87], уже во втором квартале 2015 года интенсивность создания новых вредоносных программ составила 230 тысяч ежедневно, с ростом в 43% по сравнению с аналогичным периодом годом ранее.

Критичность ситуации илюстрируется следующим образом [66]. Как известно ни один антивирус не дает 100% защиты. Предположим, что эффективность лучших антивирусов составляет 99% (очень высокий показатель, нереальный). Тогда в 2008 году такой антивирус не знает 1% от 1,6 млн = 16 тыс. вредоносных программ. В 2013 году независимо всего лишь 1% означает недетектирование 500 тыс. вредоносных программ! А в 2015 году уже 2 млн!

Интересно, что в данных исследованиях авторы приняли, что эффективность антивирусов (возможность детектирования новых вредоносных программ) составляет 99%. Это крайне завышенная оценка. Так, например, в исследованиях [80] утверждается, что уровень детектирования эвристических методов составляет в среднем 50-70%, соответственно, 30-50% вновь появляющихся угроз не детектируются (и подобная оценка не единственна). Увеличьте результаты, приведенные на рис. 3.23 а), хотя бы на порядок!

На основании своего исследования [80] автор делает следующий неутешительный вывод: «Если ничего не менять, то со старыми реактивными (сигнатураными) и старыми прогрессивными технологиями (расширенные сигнатуры, эвристика, поведенческий анализ и т.п.) антивирус с каждым годом будет пропускать все больше и больше, и в какой-то момент использовать их станет вовсе бессмысленно. Вам нужна будет защита, пропускающая на компьютер каждую вторую вредоносную программу (50%) или того меньше?».

Заметим, что сложившуюся ситуацию автор характеризует, как технологический тупик.

Однако, разовым это исследование, и посмотрим на проблему с иной стороны. Из того же исследования [80]: «По данным исследования, проведенного во втором квартале 2010 года компанией NSS Labs, время, необхо-
димое антивирусным компаниям для блокирования web-угроз, составляет от 4,62 до 92,48 часа. Это касается только одной угрозы. Несколько возникает вопрос, а какая же доля обнаруженного вредоносного ПО может быть обработана в разумные сроки, и какой для этого потребуется ресурс?

Ранее при оценке уровня защищенности современных ОС и приложений мы использовали аппарат теории массового обслуживания, применим его и здесь. В данном случае, допустим, что обнаружение новой вредоносной программы описывается пуассоновским входящим потоком (описывает наиболее случайные события, что и имеет место на практике), суммарную интенсивность которого обозначим, через:

\[ \lambda \]

Примем также, что время, необходимое антивирусной компании для блокирования угрозы, связанной с обнаружением новой вредоносной программы, имеет экспоненциальное распределение с интенсивностью:

\[ \mu \]

В теории массового обслуживания есть очень важный параметр, загрузка — для одноканальной системы массового обслуживания (М/М/1) это отношение \( \lambda / \mu \), и понятие стационарности системы, определяемое следующим условием [9]:

\[ \lambda / \mu < 1 \]

Суть данного неравенства сводится к следующему — если оно для системы выполняется, то система стационарна, в противном случае — нет. Стационарная система характеризуется тем, что все поступающие в систему заявки, в конечном счете — не говорим о среднем времени нахождения заявки в системе, это другой вопрос, будут обработаны, нестационарная характеризуется бесконечным увеличением очереди на обслуживание (все заявки никогда не будут обслужены).

Попробуем грубо оценить, сколько же потребуется вирусных анализиков в компании для обеспечения стационарности системы. Для этого рассмотрим уже систему массового обслуживания (М/М/1), т. е. систему с обслуживаемыми приборами, для которой условие стационарности определяется следующим образом: \( \lambda / C \mu < 1 \). Числом обслуживаемых приборов будем задавать число вирусных анализиков, работающих в компании. Будем считать, что работа вирусных анализиков не пересекается (каждый работает свою угрозу вируса), а среднее время необходимое аналитику для полного устранения угрозы, составляет, пусть, 24 часа (ранее говорили, о необходимости от 4,62 до 92,48 часа). Примем значение загрузки, равное 1 (с учетом того, что суммарная интенсивность обслуживания в С-канальной системе в 4 раза выше, чем в одноканальной), для задания интенсивности входного потока заявок на обслуживания воспользуемся статистикой, представленной на рис. 3.23 а), и рассчитаем число вирусных анализиков, требуемых в антивирусной компании для обеспечения стационарности системы. Их должно быть более расчетного значения С, т. к. расчеты мы делаем для загрузки, равной 1.

Результаты расчетов представлены на рис. 3.23 б). Как видим, в 2013 году вирусных анализиков в компании должно быть более 1000, а в 2015 году уже более 5000, и это в одной компании! Что же это должно быть за компания, да и где взять такое количество специалистов? А ведь выполнение данного требования обеспечивает лишь одно — стационарность системы, или условие того, что каждая заявка в конечном счете будет обслужена. За какое время (при загрузке системы стремящейся к 1, это иной вопрос). А, если, как ранее отмечали, входной поток увеличить на порядок (исходно мы говорили о 99% детектирования новых вредоносных программ, в то время как статистика указывает на 50-70%), получим десятки тысяч специалистов в одной компании!

К слову сказать, 50%-70%, это оценка экспертов. А какова сегодня оценка доли детектирования антивирусными средствами защиты новых вредоносных программ самих разработчиков средств защиты? Например, соответствующая оценка представлена в [84]. Утверждается следующее: Развитие новых технологий детектирования, опирающихся на возможности KSN(технология Kaspersky Security Network, основанная на создании антивирусного облака), позволило увеличить с 60% до 75% доля угроз, обнаруживаемых эвристическими методами без обновления классических антивирусных баз. Другими словами, не детектированными остаются не 1%, а 25% (и это при использовании новых технологий). Из представленных расчетов следует делать следующий вывод. Когда речь заходит о том, что время, необходимое антивирусным компаниям для блокирования web-угроз, составляет от 4,62 до 92,48 часа, это справедливо, относительно угроз, принятых на обслуживание (в отношении которых решается задача нивелирования). А как с теми угрозами, которые находятся в очереди на обслуживание, а, если эта очередь бесконечна аппаратно?

Наверное, куда более корректной будет характеристика: среднее время пребывания заявки на обслуживание в системе, либо в очереди на обслуживание, отображающая не только среднюю интенсивность обслуживания одной, отдельно взятой заявки, но также интенсивность поступления заявок в систему и число обслуживаемых приборов в системе (число вирусных анализиков в компании, устраивающих угрозы с соответствующей интенсивностью), что, в конечном счете, и определяет длину очереди заявок на обслуживание.
Глава 3. Принципы и методы контроля доступа к статичным объектам

Другая проблема применения современных технологий антивирусной защиты, которая также превращается в критическую с подобным ростом объема создаваемого вредоносного ПО – это влияние средства защиты на загрузку вычислительного ресурса защищаемого компьютера. Когда базы сигнатуры составляют миллионы образцов, ставится под вопрос целесообразность использования существующих технологий антивирусной защиты и по этой причине.

В исследовании [80] утверждается, что увеличение объема обновлений, уже подошло к той границе, когда загрузка обновлений начинает вызывать заметные неудобства и недовольство пользователей. Рост размера антивирусных обновлений проиллюстрирован на рис. 3.23 с) [80]. К слову, это не единственная проблема оперативности обновлений, если мы говорим о корпоративных приложениях. Какая ни одна оперативность обновлений может достигаться при их реализации через сеть Интернет (практически в автоматическом режиме), но многие корпоративные компьютеры (прочем с наиболее критической информацией, требующей защиты) априори не должны подключаться к внешней сети, т.к. это приводит к катастрофическому разрастанию поля угроз. Как тогда обеспечить оперативность обновлений?!

Теперь о том, какой тип вредоносного ПО доминирует на сегодняшний день.

Вновь обратимся к современной статистике. По данным ежеквартального отчета компании PandaLabs о вирусной активности за второй квартал 2015 года трояничные программы в очередной раз стали самой опасной угрозой, составив 76,25% от всех новых вредоносных программ. Далее следуют шпионьи и рекламное ПО (5,43%) и черви (2,63%), а "обычные" вирусы занимают 1.53% [87]. По статистике 2011 года от тот же компания статистика была несколько иной: трояничные программы также были самой опасной угрозой, составив почти 70% от всех новых вредоносных программ. Далее следуют классические вирусы (16%) и черви (11,6%) [83].

Таким образом, основную угрозу сегодня составляет внедрение на компьютер и последующее выполнение программы, наделенной вредоносными функциями (вредоносной программой).

Для того, чтобы определиться со способами защиты от подобных атак, в первую очередь, следует определиться с объектом защиты. Если мы говорим о корпоративных приложениях, то здесь следует говорить об обязательном использовании доверенных программ. Используются программы, функционал которых заранее определен, как правило, это коммерческое ПО, которое легитимно поставляется. В крупных компаниях, кроме того, перед внедрением ПО в эксплуатацию оно предварительно тестируется на стенде на совместимость с используемыми программами.
Контроль доступа к компьютерным ресурсам

В данном случае под вредоносным ПО следует рассматривать несанкционированные программы, случайно, но недосмотру, либо умышленно устанавливаемые на защищаемые компьютеры.

Другое дело, личные (домашние) компьютеры. Здесь, к сожалению, пользователи не столь ответственно относятся к выбору устанавливаемых программ, к их функционалу, способам приобретения.

Однако мы рассматриваем именно корпоративные приложения, для которых, в первую очередь актуальны, все рассматриваемые нами в работе методы защиты, где, как правило, присутствует администратор, имеющий соответствующую подготовку и квалификацию.

В этих приложениях, как отмечали, защита от вредоносного ПО сводится к защите от внедрения и запуска (либо только внедрения, либо только запуска) на компьютеры сторонних программ, в том числе, санкционированными пользователями. А для решения данной задачи защиты может эффективно использоваться метод обеспечения замкнутости программной среды, причем при любом способе его реализации, как заданием списка разрешенных к запуску процессов (исполняемых файлов), так и по средству задания папок с исполняемыми файлами, разрешенными на запуск.

Напомним, что суть обеспечения замкнутости программной среды состоит в локализации среды исполнения – это реализация возможности запуска на защищаемом компьютере и/или отдельными субъектами (пользователями или/и процессами) только определенных санкционированных программ с предотвращением любых способов несанкционированной модификации или удаления их исполняемых файлов.

Заметим, что данный метод защиты не убережет компьютер от внедрения на его стороне (несанкционированной программы), но при этом не позволит ее запустить (выполнить). Т.е. весь компьютер может быть заражен троянами, но выполнить их не удастся. Если использовать политику проприактивированную на рис. 3.20, то предотвращается возможность запуска несанкционированной программы и с внешних накопителей, и из сети.

Однако и эти разграничения можно усилить. Можно разрешить полный доступ к санкционированным исполняемым объектам не пользователю System (см. рис. 3.21), а лишь отдельным системным процессам. К таким процессам относятся: winlogon.exe, lsass.exe, crs2.exe, svchost.exe, services.exe и некоторые другие (всего не более десятка). В этом случае даже несанкционированное получение системных прав не позволит запустить несанкционированную программу.

Глава 3. Принципы и методы контроля доступа к статическим объектам

Отметим, что в случае обеспечения замкнутости программной среды, по средством задания папок с исполняемыми файлами, разрешенными на запуск, обеспечивается защита системных ресурсов (папок, содержащих исполняемые файлы и файлы настройки ОС и приложений).

3.6.2. Защита от сетевых атак на уязвимости ОС и приложений. Реализация процессной модели контроля доступа

В общем случае сетевые атаки весьма разнообразны, мы же здесь будем рассматривать атаки, направленные на использование уязвимостей системы и приложений, т.е. те атаки, в результате которых может быть осуществлен несанкционированный доступ к защищаемым ресурсам, в первую очередь к объектам файловой системы.

Для защиты от подобных сетевых атак сегодня широко используются, так называемые, системы обнаружения вторжений.

Системы обнаружения вторжений (COB) применяются для обнаружения различных типов вредоносной активности, которая может нарушить безопасность компьютерной системы [7]. К такой активности относятся сетевые атаки против уязвимых сервисов, атаки, направленные на повышение привилегий, неавторизованный доступ к важным файлам, а также действия вредоносного программного обеспечения (компьютерных вирусов, троянов и червей).

В пассивной COB (Intorsion Detection System, IDS) при обнаружении нарушения безопасности, информация о нарушении записывается в лог приложения, а также сигналы опасности отправляются на консоль и/или администратору системы по определенному каналу связи. В активной системе, также известной как система Предотвращения вторжений (Intrusion Prevention system, IPS), COB ведет ответные действия по нарушению. Ответные действия могут проводиться автоматически либо по команде оператора.

Обнаружение нарушения безопасности проводится обычно с использованием эвристических правил и анализа сигнатуры известных компьютерных атак.

Уже в 1984 Фред Коз (которым 4 ноября 1983 года был создан первый компьютерный вирус) сделал заявление о том, что каждое вторжение обнаружить невозможно и ресурсы, необходимые для обнаружения вторжений, будут расти вместе со степенью использования компьютерных технологий.

Как видим, и здесь используются те же технологии, что и в решении задачи защиты от вредоносного ПО – эвристические правила и анализ сигнатуры, т.е., как и при антивирусной защите, все основано на применении тех-
Глава 3. Принципы и методы контроля доступа к статическим объектам

Итак, большинство сетевых атак происходит через эксплуатацию уязвимостей в интернет-браузерах. В частности, к ним относятся выполнение вредоносного кода JavaScript, запуск вредоносных элементов ActiveX и Flash-объектов.

На втором месте находятся атаки через вложения в электронной почте. Злоумышленники используют различные социальные технологии, основанные на знании психологией людей, чтобы заставить получателя электронной почты запустить файл, вложенный в письмо, или перейти по указанной в письме ссылке.

На третьем месте располагаются различные уязвимости в операционных системах (их статистика была нами приведена на рис. 2.2).

Определим процессы сетевых служб Windows, критичных к защите от сетевых атак. Для этого сведем в табл. 3.1 следующие данные:

- имя сетевой службы,
- выполняемый процесс,

по основным сетевым службам операционных систем семейства Microsoft Windows (в частности Windows 7/8/10).

Таблица 3.1. Основные системы службы, поддерживаемые сетевым атакам

<table>
<thead>
<tr>
<th>Имя сетевой службы</th>
<th>Процесс</th>
</tr>
</thead>
<tbody>
<tr>
<td>Служба шлюза уровня приложения</td>
<td>%WinDir%\System32\alg.exe</td>
</tr>
<tr>
<td>Диспетчер учетных записей безопасности</td>
<td>%WinDir%\System32\sas.exe</td>
</tr>
<tr>
<td>Поставщик поддержки безопасности NTLM</td>
<td>%WinDir%\System32\sas.exe</td>
</tr>
<tr>
<td>Сетевой вход в систему</td>
<td>%WinDir%\System32\sas.exe</td>
</tr>
<tr>
<td>Службы IPSEC</td>
<td>%WinDir%\System32\sas.exe</td>
</tr>
<tr>
<td>Маршрутизация и удаленный доступ</td>
<td>%WinDir%\System32\svchost.exe</td>
</tr>
<tr>
<td>Обозреватель компьютеров</td>
<td>%WinDir%\System32\svchost.exe</td>
</tr>
<tr>
<td>Общий доступ к подключению Интернета/брандмауэра подключения к Интернету</td>
<td>%WinDir%\System32\svchost.exe</td>
</tr>
<tr>
<td>Сервер</td>
<td>%WinDir%\System32\svchost.exe</td>
</tr>
<tr>
<td>Служба времени Windows</td>
<td>%WinDir%\System32\svchost.exe</td>
</tr>
<tr>
<td>Удаленный вызов процедур (RPC)</td>
<td>%WinDir%\System32\svchost.exe</td>
</tr>
</tbody>
</table>

Из табл. 3.1 видим, что, в первую очередь, следует учитывать уязвимости следующих сетевых служб (на примере ОС семейства Windows):

- процесс %Windir%\System32\svchost.exe;
- процесс %Windir%\System32\sas.exe;
- процесс %Windir%\System32\alg.exe.
Здесь следует отметить, что сетевые атаки, основанные на использовании уязвимостей системных процессов, менее вероятны по сравнению с уязвимостями процессов приложений, но, вместе с тем, они более критичны, т.к. системные процессы функционируют с системными правами.

Четвертое место занимают заражения через загруженные файлы. При этом учитывается не только файлы, сознательно загруженные пользователем, но и файлы, загруженные из сети без ведома пользователя (такой тип загрузки называется "drive-by download"). Часто вредоносные программы используют уязвимости в интернет-браузерах или другие методы для скрытой зараженности и запуска других вредоносных программ. Например, при выборе ссылки на web-странице, может не только произойти открытие запрошенной web-страницы, но и начаться сокрытая загрузка вредоносного файла.

Итак, подытожим результаты нашего исследования:

1. **Угроза сетевой атаки несет в себе процесс.** Причем это может быть как процесс приложения, так и системный процесс. Именно уязвимость процесса используется злоумышленником, причем эксплуатация подобной уязвимости позволяет злоумышленнику заставить выполнять процессу по своему сценарию.
2. **Сетевая атака, как правило, направлена на внедрение вредоносной программы, хищение и нарушение целостности (модификация) конфиденциальной информации** (обработываемых данных).

Рассмотрим реализацию защиты от сетевых атак на критичные к сетевым атакам системные процессы [28].

Системные процессы, выполняемые с самыми высокими привилегиями, являются излюбленной мишенью для атак злоумышленников. Этому можно выделить две причины:
- скомпрометированная система служеб может позволить злоумышленнику исполнить произвольный код с административными полномочиями;
- скомпрометированная система служеб может позволить злоумышленнику заразить компьютер без совершения его владельцем каких-либо дополнительных действий.

Например, нет необходимости ждать пока пользователь посетит зараженную web-сайт — вредоносная программа в состоянии сама активироваться через уязвимую сетевую службу и дальше продолжить самостоятельное распространение по сети. В итоге, использование критических уязвимостей в сетевых службах позволяет злоумышленникам распространять вредоносные программы значительно быстрее, чем с помощью других активных

используемых средств. Поэтому защита от атак на сетевые службы является критически важной составляющей защиты операционной системы.

Сформируем соответствующую разграничительную политику доступа [28], для этого, прежде всего, рассмотрим категории сетевых атак, использующих уязвимости в сетевых службах Windows:

- **Атаки, приводящие к переполнению буфера.** Данные сетевые атаки направлены на вызов отказа в работе службы или на выполнение процессом службы произвольного кода. Отказ работы системной службы может привести к полной или частичной неработоспособности системы, например, к отказу в доступе к сети или появлеению "сигнала экрана" (BSOD).

- **Атаки, приводящие к открытой записи вредоносных программ с последующим запуском.** Таким образом распространяются вредоносные черви. Так как системные службы выполняются от имени учетной записи системы с самыми высокими привилегиями, то злоумышленники стараются произвести открытую запись вредоносного модуля злоцелевого приложения в системный каталог Windows.

Сформулируем требования к разграничительной политике доступа к файловым объектам для рассматриваемых системных процессов, с целью противодействия рассмотренным сетевым атакам:

- **Запрещение запуска и выполнения в контексте безопасности учетной записи пользователя.** Продесссы системных служб запускаются исключительно в контексте безопасности системной учетной записи.

- **Запрещение обращения по выполнению к любым объектам операционной системы, кроме объектов с расширениями исполняемых файлов из системного каталога (%WinDir%).**

- **Запрещение обращения по записи к объектам с расширениями исполняемых файлов в любые каталоги операционной системы.** Для этого для обращений по записи следует разрешить только объекты в каталоге системы (%WinDir%), каталоге приложений (%ProgramFiles%), а также в каталогах профилей учетных записей (Documents and Settings\*). Обращение по записи к любым другим объектам операционной системы должно быть запрещено.

- **Запрещение обращения по чтению к объектам каталогов, сохраняющих важную для пользователя информацию.** Например, подобные сведения могут храниться в папке "Мои документы" в каталоге профиля пользователя. При этом для обращений по чтению следует раз-
решить только системный каталог (%WinDir%), каталог приложения (%ProgramFiles%), а также каталоги профилей учетных записей (Documents and Settings\*). Обращение к любым другим объектам операционной системы должно быть запрещено.

- Запрещение субъекту обращения по удалению или переименованию ко всем объектам операционной системы.

На рис. 3.25 проиллюстрирована разграничительная политика доступа, реализующая сформулированные требования, в части защиты от сетевых атак, для следующих системных субъектов («первичный идентификатор пользователя, полнопотоковое имя процесса, эффективный идентификатор пользователя») – эти субъекты доступа образуют один профиль, для которого назначаются соответствующие разграничения:

- LOCAL SERVICE,%WinDir%\System32\ALG.EXE, LOCAL SERVICE;
- SYSTEM, %WinDir%\System32\ALG.EXE, LOCAL SERVICE;
- SYSTEM, %WinDir%\System32\ALG.EXE, SYSTEM;
- SYSTEM, %WinDir%\System32\LSASS.EXE, SYSTEM;
- LOCAL SERVICE, %WinDir%\System32\svchost.exe, LOCAL SERVICE;
- NETWORK SERVICE, %WinDir%\System32\svchost.exe, NETWORK SERVICE;
- SYSTEM, %WinDir%\System32\svchost.exe, SYSTEM.

Замечание. Дополнительно настраивается следующее правило: Если указанные процессы запускаются или выполняются от имени другой учетной записи (не корректна (не соответствует заданию субъектов в разграничительной политике доступа) пара «Первичный идентификатор пользователя» – «Эффективный идентификатор пользователя»), то обращения считаются запрещенными – доступ процессов ко всем объектам операционной системы запрещается правилом, приведенным на рис. 3.26.

<table>
<thead>
<tr>
<th>Тип</th>
<th>Объект файловой системы</th>
<th>Режим доступа</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ProgramFiles%</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemDrive%</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemDrive%\Documents and Settings</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemDrive%\RECYCLER</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemDrive%\System Volume Information</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.BAT*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.CMD*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.COM*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.DLL*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.EXE*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.JS*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.VBS(ESI)*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%*.WS([H]*)</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%\System32\TSP</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%\WindowsShell*</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
<tr>
<td>%SystemRoot%\System32*DRV</td>
<td>+4+5-8-Y-Y</td>
<td></td>
</tr>
</tbody>
</table>

Рис. 3.25. Разграничительная политика доступа для критических с сетевым атакам системных процессов

Рис. 3.26. Правило доступа для запрещенных обращений

Теперь рассмотрим категории атак, использующих уязвимости в интернет-браузерах [28].

a. Атаки на уязвимости в программном обеспечении, приводящие к переполнению буфера. Наличие таких уязвимостей на данный момент выявлено во множестве приложений, не исключая интернет-браузеры. Переполнение буфера может вызывать аварийное завершение или за- висание программы, ведущее к отказу обслуживания. Отдельные виды
переполнений позволяют злоумышленнику загрузить и выполнить произвольный машинный код от имени программы и с правами учетной записи, от имени которой она выполняется, что означает, в общем случае, получение злоумышленником доступа к конфиденциальной информации пользователя.

1. Выполнение команд пользователем, когда в результате обмана или принуджения пользователь запускает исполняемое содержимое, созданное злоумышленником. Существует несколько видов атак этого типа.

2. Исполняемое содержимое HTML-страницы. Было обнаружено немало уязвимостей в интернет-браузерах, которые при открытии web-сайта, сформированного специальным образом, приводили к автоматическому запуску приложений на компьютере пользователя. В частности, к таким ошибкам относятся неправильная обработка интернет-браузером тегов IFRAME и MIME.

6. Ошибки в технологиях активного содержания, которые позволяют запускать недопустимый код. К ним относятся ошибки в элементах управления ActiveX и ошибки в виртуальной Java-машинах. Кстати сказать, в исследовании [41] утверждается следующее: В 2011 году резко увеличилась популярность эксплоитов к Java-машина, в итоге уязвимости в Java заняли второе место в рейтинге уязвимостей — с ними было связано четверть всех инцидентов, при этом современные наборы эксплоитов наполовину состоят именно из Java-эксплоитов.

2. Запись локальных файлов, обычно в каталогах, для которых разрешено выполнение (чаще встречаются по причине неправильного размещения временных каталогов или кэш-файлов). После того, как файл локально записан, только списки контроля доступа файловой системы и права учетной записи пользователя определяют допустимые для него действия. Злоумышленник, зная, что вредоносный файл загружен в локальный каталог, допускающий выполнение содержащихся в нем объектов, может с помощью специально созданной HTML-страницы инцировать выполнение этого файла.

3. Чтение локальных файлов, например, с использованием свойств HTML или с помощью тега IFRAME. Общий результат применения таких методов — просмотр файлов cookie интернет-браузера и получение паролей пользователя.

4. Открытие исходных клиентских соединений. При этом используются уязвимости в интернет-браузере для создания не запланированных пользователем соединений. Например, открытия web-страниц, содержащих вредоносное или рекламное содержимое.

Сформулируем требования к разграничительной политике доступа к файловым объектам для интернет-браузеров [12] для противодействия последствиям указанных сетевых атак:

а. Запрещение санкционированному субъекту (субъекту, к которому применяется разграничительная политика) выполнение в контексте безопасности учетной записи, отличной от учетной записи санкционированного субъекта, запущенной процессе.

б. Запрещение санкционированному субъекту обращения по исполнению к любым объектам операционной системы, кроме объектов с расширениями исполняемых файлов из системного каталога (%WinDir%) и каталога приложений (%ProgramFiles%)

в. Разрешение санкционированному субъекту обращения по записи к объектам каталога профиля учетной записи пользователя (%UserProfile%). При этом обращение по записи к каталогам, содержащим важные файлы и управляющие процессом, должно быть запрещено.

г. Разрешение санкционированному субъекту обращения по чтению к объектам системного каталога (%WinDir%), каталога приложений (%ProgramFiles%), а также каталога профиля учетной записи пользователя (%UserProfile%). Запрещение санкционированному субъекту обращения по чтению данных объектов, содержащих вредоносные файлы, должно быть запрещено.

д. Запрещение санкционированному субъекту обращения по удалению и перезаписи файлов, содержащих конфиденциальную информацию, содержащую вредоносные файлы.

Реализация первого правила позволит предотвратить последствия успешной атаки на повышение привилегий.

Реализация второго и третьего правил позволит предотвратить последствия скрытой загрузки вредоносных файлов: загруженный вредоносный файл не может быть запущен защищаемым процессом.
Реализация четвертого правила позволит предотвратить скрытое чтение конфиденциальных данных пользователя.

Отметим, что приведенные правила ограничения доступа не позволяют предотвратить сетевые атаки, направленные на отказ приложения, а также создания исходящих клиентских соединений.

Стоит учесть (и это мы отмечали не раз), что, не смотря на необходимость следования приведенным выше политикам безопасности, многим приложением для выполнения своих функций необходимо разрешить обращение к объектам, запрещенным политикой безопасности. Причиной этому является нежелание разработчиков программных продуктов проводить оценку безопасности использования своих разработок с позиций реализации контроля доступа. Поэтому для каждого конкретного приложения необходима корректировка правил разграничения доступа. При этом, к сожалению, подчас, придется делать выбор между работоспособностью и безопасностью приложения.

На рис. 3.27 проиллюстрирована разграничительная политика доступа, реализующая сформулированные требования, в части защиты от сетевых атак, для интернет-браузера Microsoft Internet Explorer (для соответствующего процесса), запускаемого и выполняемого от имени учетной записи пользователя.

Если процесс интернет-браузера запускается или выполняется от имени другой учетной записи, то действует правило, проиллюстрированное на рис. 3.26.

Теперь рассмотрим категории атак на почтовые клиенты и их пользователей [5].

а. Атаки на уязвимости в программном обеспечении, приводящие к переполнению буфера.

б. Выполнение команд пользователем, когда в результате обмана или принуждения пользователь запускает исполняемое содержимое, созданное злоумышленником. Существует несколько видов атак этого типа:
1) замаскированные или безобидные на вид вложения в электронную почту;
2) исполняемое содержимое HTML в электронной почте.
Сформулируем политику разграничительного доступа к ресурсам для почтовых клиентов для противодействия последствиям указанных сетевых атак:

а. Запрещение санкционированному субъекту выполнения в контексте безопасности учетной записи, отличной от учетной записи, запустившей процесс.

б. Запрещение санкционированному субъекту обращения по выполнению к любым объектам операционной системы, кроме объектов с расширениями исполняемых файлов из системного каталога (\%WinDir\%) и каталога приложений (\%ProgramFiles\%).

в. Разрешение санкционированному субъекту обращения по запросу к объектам каталога профиля учетной записи пользователя (\%UserProfile\%). При этом обращение по записи к каталогам, содержащим важную для пользователя информацию (например, \"Мой документы\" или \"Cookies\") должно быть запрещено. Для каталогов, предназначенных для хранения конфиденциальной информации, обращение по записи санкционированному субъекту должно быть запрещено. При этом необходимо настроить почтовый клиент так, чтобы файлы и каталоги электронной почты хранились в каталоге профиля учетной записи пользователя.

г. Разрешение санкционированному субъекту обращения по чтению к объектам системного каталога (\%WinDir\%), каталога приложений (\%ProgramFiles\%), а так же каталога профиля учетной записи пользователя (\%UserProfile\%). Запрещение санкционированному субъекту обращение по чтению к объектам каталогов, содержащих важную для пользователя информацию (например, каталоги \"Мой документы\" и \"Cookies\" в каталоге профиля пользователя). Для каталогов, предназначенных для хранения конфиденциальной информации, обращение по чтению необходимо запретить.

d. Запрещение санкционированному субъекту обращения по удалению или перемещению ко всем объектам операционной системы.

На рис. 3.28 проиллюстрирована разграничительная политика доступа, реализуемая сформулированные требования, в части защиты от сетевых атак, для почтового клиента Microsoft Outlook Express, запускаемого и выполняемого от имени учетной записи пользователя [12].

Из процесс почтового клиента запускается или выполняется от имени другой учетной записи, то действует правило, проиллюстрированное на рис. 3.24.
3.7. Иллюстрация общности сформулированных принципов контроля доступа к статичным объектам для решения задачи формирования режимов обработки информации субъектами доступа

Ранее мы рассматривали принципы контроля доступа к статичным объектам и вопросы их практической реализации, применительно к реализации контроля доступа к файловым объектам, включая файловые накопители, и к объектам реестра ОС. Однако рассмотренные принципы могут успешно использоваться и при реализации контроля доступа к иным статичным объектам.

Еще раз, вкратце, напомним, о каких принципах контроля доступа идет речь.

1. Контроль доступа к статичным объектам предполагает наличие подобных объектов на момент реализации (настройки администратором) разграничительной политики доступа, т.е. до ввода защитного компьютера в эксплуатацию.

2. В разграничительной политике доступа используются две равноправные сущности — субъект и объект доступа.

3. Разграничительная политика доступа предполагает назначение прав доступа субъектов к объектам (а не, наоборот, — к объектам субъектам).

4. Реализуется разрешительная разграничительная политика доступа: «Все что я не сделал (не разрешил), то запрещено».

5. Реализуется принудительное управление информационными потоками — пользователи в любом виде исключаются из схемы администрирования.

6. Субъект доступа определяется тремя сущностями: исходный (или первичный) идентификатор пользователя, полноправное имя процесса, эффективный идентификатор пользователя.

7. Субъекты и объекты доступа могут характеризоваться, как своими именами, так и масками, позволяющими «покрывать» в разграничителей политике группы субъектов и объектов.

8. При реализации контроля доступа (при выборе актуального правила из матрицы доступа), как субъекты, так и объекты характеризуются наиболее их точным описателем, применимо к идентификатору, определяемому диспетчером из запроса доступа.

9. Субъектом, для которого назначаются права доступа к статичным объектам, выступает сущность «Профиль», позволяющий объединить заведенных в разграничительной политике доступа субъектов в группы, использование которой в значительной мере упрощает задачу администрирования средства защиты, как при вводе защищенного компьютера в эксплуатацию, так и при последующей эксплуатации.

Рассмотрим далее на нескольких примерах практическую реализацию (технических решений) изложенных принципов контроля доступа к статичным объектам, не являющимся ни файловыми объектами, ни объектами реестра ОС. Естественно, вид объекта определяет способы его идентификации и назначаемые к нему права доступа (в том числе, типы доступа) субъектов, как следствие, не смотря на единство реализуемых принципов, собственно технические решения могут значительно различаться, в том числе и по слож-
нности реализации, что далее проиллюстрируем примерами. При этом показем, что реализация контроля доступа к некоторым типам объектов доступа (на примере сетевых объектов) требует проведения соответствующего исследования, с целью формирования в отношении данных методов требований к построению безопасной системы.

### 3.7.1. Контроль доступа к принтерам.
#### Техническое решение

В качестве объектов доступа выступают все физические (локальные и разделенные в сети) и виртуальные принтеры, к которым может быть реализован доступ с защитного компьютера. Объекты — принтеры, к которым разграничивается доступ, выбираются из раскрывающегося списка, как, например, показано на рис. 3.29.

Заданные правила доступа к принтерам отображаются в интерфейсе среды защиты в виде, представленном на рис. 3.30. Здесь также может использоваться маска «*» (все), для реализации разрешительной политики доступа.

![Рис. 3.29. Меню выбора объекта доступа "Принтер"](image)

### 3.7.2. Контроль доступа к сетевым объектам.
#### Техническое решение

Реализация разграничительной политики доступа к сетевым объектам принципиально сложнее реализации контроля доступа к принтерам, ввиду специфики объекта [56].

Прежде, чем перейти к рассмотрению данного вопроса, акцентируем внимание на том, что в части решения задач сетевой безопасности на практике широко используется межсетевое экранирование. Межсетевой экран — отдельное средство защиты, устанавливается на стыке подсете с целью разграничения прав доступа из одной подсети в другую. Для принятия решения о санкционированности запрашиваемого доступа межсетевым экраном анализируются сетевые пакеты (заголовки сетевых пакетов), из которых определяется то, каким компьютером с каким и с какой целью запрашивается взаимодействие. На основании заданных в межсетевом экране правил взаимодействие разрешается межсетевым экраном (анализируемые пакет пропускается в иную подсеть), либо нет. Поскольку межсетевой экран устанавливается на стыке подсетей, дополнительно к нему, как правило, решаются задачи маршрутизации и трансляция адресов.

Таким образом, основной реализации контроля доступа (к подсети) межсетевым экраном является фильтрация сетевых пакетов, соответственно возможности для подобной фильтрации определяются содержимым заголовков сетевых пакетов (для различного уровня иерархии стека сетевых протоколов).
Требования к набору функциональных задач защиты в части фильтрации пакетов, которые должны решаться межсетевым экраном (в зависимости от его класса, определяющего область практического использования), сформулированы в [3], в первую очередь к ним относится следующее. Межсетевой экран должен обеспечивать:

- фильтрацию на сетевом уровне. Решение по фильтрации может приниматься для каждого сетевого пакета исключительно на основе, по крайней мере, сетевых адресов отправителя и получателя или на основе других эквивалентных атрибутов;
- фильтрацию пакетов служебных протоколов, служащих для диагностики и управления работой сетевых устройств;
- фильтрацию с учетом входного и выходного сетевого интерфейса как средства проверки подлинности сетевых адресов;
- фильтрацию с учетом любых значимых полей сетевых пакетов;
- фильтрацию на транспортном уровне запросов на установление виртуальных соединений. При этом, по крайней мере, учитываются транспортные адреса отправителя и получателя;
- фильтрацию на прикладном уровне запросов к прикладным сервисам. При этом, по крайней мере, учитываются прикладные адреса отправителя и получателя;
- фильтрацию с учетом даты/времени.

Рассматриваемая задача контроля доступа к сетевым объектам существенно отличается. Система защиты, решающая подобную задачу, в общем случае устанавливается не между подсетями, а на окончном сетевом устройстве — на подключенном к сети компьютере. Как следствие, при этом не требуется решать задачи маршрутизации и транслации адресов.

Естественно, что система защиты, устанавливаемая на окончном сетевом устройстве, также доступны для фильтрации сетевые пакеты, исходящие с защищаемого компьютера, и поступающие на данный компьютер. Однако рассматриваемая задача состоит в разграничении прав доступа субъектов (в рассматриваемой реализации — профилей, в один профиль включаются равноправные, с точки зрения доступа к сетевым объектам субъекты, идентифицируемые тремя сущностями — исходный идентификатор пользователя, полнопутевое имя процесса, эффективный идентификатор пользователя) к объектам.

Естественно возникает ряд вопросов по реализации контроля доступа заданных подобным образом субъектов к сетевым объектам, в первую оче-
Для разрешенных подобным образом виртуальных каналов назначаются правила доступа (исходящего и входящего) — разрешаются/запрещаются соответствующие команды (установления входящего и/или исходящего соединения для протокола TCP, соответственно отправки и/или приема пакета для протокола UDP), см. рис. 3.34 а.
Замечание. Очень важной дополнительной возможностью решения рассматриваемой задачи защиты на практике является возможность регламентирования доступа к сетевым объектам по времени (по дням недели и часам/минутам) — контроль доступа к сетевым объектам по времени.

При этом для любого заданного интервала времени, см. рис. 3.34 б, могут быть заданы свои правила доступа к сетевым объектам, см. рис. 3.34 в.

Не смотря на то, что сегодня на практике, как правило, при доступе во внешнюю сеть используется стек протоколов ТСР/IP, в разграничительной политике доступа нельзя забывать и о возможности применения, в том числе, несанкционированного использования, иных транспортных протоколов, которые требуется запретить, либо разрешить, задав для них соответствующие правила доступа — разрешенные команды, см. рис. 3.35 — рис. 3.37. Кроме того, необходимо разграничивать команды протокола управления устройствами в IP-сетях SNMP, см. рис. 3.35 — рис. 3.37.

Рис. 3.35. Окно настройки дополнительных транспортных протоколов, команд протоколов ICMP и SNMP

Рис. 3.36. Окно создания правил ICMP, SNMP, транспортного протокола

Рис. 3.37. Интерфейс отображения дополнительных транспортных протоколов, команд протоколов ICMP и SNMP

Замечание. Должен быть разрешен протокол ARP для обеспечения корректной работы в локальной сети.
Важнейшим объектом при реализации разграничительной политики доступа к сетевым объектам является сетевая служба, т.к. в конечном счете, при реализации разграничительной политики доступа не менее важно, с каким удаленным компьютером по какому виртуальному каналу может взаимодействовать субъект доступа, используя какой транспорт, то, с какой целью осуществляется удаленный доступ (какие задачи решаются субъектом при доступе к этому объекту). Оно задания сетевых служб, которыми может (или не должен) пользоваться субъект при осуществлении доступа к сети (объекту), приведено на рис. 3.38. Именно заданием разрешенных служб субъект разрешается использовать тот или иной доступ: к электронной почте, web-сервисам и т.д.

Рис. 3.38. Оконо задания объекта доступа сетевая служба и интерфейс отображения заданых правил для служб

Замечание. Для упрощения задачи администрирования реализована функция анализатора, обеспечивающая отображение списка всех событий аудита в реальном времени: соединения по протоколу TCP, пакеты UDP, соединения служб, пакеты всех остальных транспортных соединений, см. рис. 3.39. В первую очередь, это необходимо применительно для настройки прав доступа к сетевым объектам системных пользователей и процессов.

Рис. 3.39. Оконо анализатора события в реальном времени

А теперь проанализируем получаемую архитектуру средства защиты и сформулируем соответствующее требование к построению контроля доступа к сетевым объектам.

В составе средства защиты для реализации контоля и разграничения прав доступа к сетевым объектам должны решаться задачи на различных уровнях иерархии архитектуры операционной системы. Первая из них предполагает перехват запроса доступа для идентификации субъекта доступа — должна решаться на уровне драйвера TCP/IP ОС [56] операционной системы, где доступны имена пользователя и процесса, запрашивающего доступ к сетевому объекту. Однако на этом уровне драйвер средства защиты не имеет возможности анализа заголовков сетевых пакетов (фильтрации сетевых пакетов), они обрабатываются системой на более низком уровне. А именно из заголовков сетевых пакетов можно корректно идентифицировать некоторые сетевые объекты доступа, необходимые при реализации разграничительной политики доступа, например, протоколы и службы, команды протокола ICMP и др. Для решения этой задачи — фильтрации сетевых пакетов, средство защиты должно иметь в своем составе драйвер, работающий на уровне драйвера NDIS (Network Driver Interface Specification) [56] операционной системы. На этом уровне иерархии архитектуры уже могут перехватываться сетевые пакеты (исходящие и входящие) для анализа их заголовков, но, в свою очередь, на этом уровне невозможно определение субъекта доступа — какой пользователь каким процессом запросил доступ к сетевому объекту (таких данных нет в сетевом пакете).

Замечание. Отметим, что идентификация сетевой службы не из соответствующего заголовка сетевого пакета, а по номеру порта (что теоретически позволяет обойтись без фильтрации сетевых пакетов),
не корректна – несет в себе технологическую уязвимость, поскольку определенные службы лишь рекомендуется подключать к определенным портам, не более того. В общем случае любая служба может быть подключена к любому порту, как следствие, номер порта и сетевую службу следует рассматривать в качестве взаимосвязанных, но самостоятельных в разграничительной политике объектов доступа.

Из сказанного сформулируем соответствующее требование к решению рассматриваемой задачи контроля доступа к объектам.

Реализацией контроля доступа к сетевым объектам в обязательном порядке должны совместно решаться и задача контроля доступа к объектам, и задача фильтрации сетевых пакетов, причем корректное их решение возможно лишь на различных уровнях иерархии архитектуры операционной системы.

3.8. Дополнительные задачи реализации контроля доступа к статическим объектам на примере контроля доступа к сервисам олицетворения

Решение задачи защиты информации от несанкционированного доступа в общем виде предполагает в том числе решение задач инвилегирования технологических уязвимостей. Применимо к реализации контроля доступа к статическим объектам это предполагает решение дополнительных (к формированию режимов обработки информации субъектами доступа) задач разграничения прав доступа к статическим объектам, в качестве которых уже выступают системные ресурсы.

Проявлюем сказанное примером реализации метода контроля доступа к системным ресурсам, направленного на инвилегирование технологической уязвимости. Ранее мы говорили о том, что весьма актуальна угроза атаки на повышение привилегий. Для инвилегирования последствий подобной атаки в общем случае (при любом способе смены исходного (первичного) идентификатора пользователя), в разграничительную политику доступа включены две сущности, идентифицирующие пользователя при доступе к объектам – исходный и эффективный идентификаторы пользователя.

Однако это предотвращает лишь собственно несанкционированный доступ субъекта к объекту, а не получение им прав другого пользователя, что само по себе крайне критично в случае получения интерактивным пользователем прав администратора, а уж тем более системных прав.

Ранее также отмечалось, что для смены прав пользователя может использоваться штатная возможность современных ОС – сервисы олицетворения. Поскольку эта функция штатная (можно говорить о безусловной технологической уязвимости), при решении задачи защиты от атак на повышение привилегий применительно к этой возможности ОС защита должна реализовывать в первую очередь.

Рассмотрим метод защиты и техническое решение [52].

В качестве субъекта доступа в данном случае выступает процесс, в разграничительной же политике доступа к сервисам олицетворения задается, для какого процесса из какого исходного пользователя (пользователя, запустившего этот процесс), в какого эффективного пользователя следует разрешить/запретить олицетворение.

Замечание. В качестве субъекта доступа при реализации контроля доступа к сервисам олицетворения следует использовать процесс, т.к. именно это позволяет снять ряд противоречий, возникающих при создании разграничительной политики доступа. Например, очевидно, что нельзя разрешать интерактивному пользователю олицетворять себя с системным пользователем. Однако если мы (на самом деле, очень мало) системных процессов, например, svchost, необходимо разрешить подобное олицетворение, иначе это сказывается на корректности работы системы. Процесс svchost при загрузке системы олицетворяет себя (запускается с системными правами) с регистрирующимися в системе интерактивным пользователем, затем, обратно. Использование сущности «Процесс» в качестве субъекта доступа к сервисам олицетворения позволяет учесть подобные противоречия при настройке соответствующей разграничительной политики доступа.

Меню создания правил доступа к сервисам олицетворения представлено на рис. 3.40, интерфейс отображения созданных правил олицетворения на рис. 3.41.

Из меню, представленного на рис. 3.40, можно задать для какого процесса (могут использоваться маски) для какого пользователя (выбирается в поле «Из пользователя») в какого пользователя (выбирается в поле «В пользователя») разрешается/запрещается олицетворение. Здесь опять же требуемое правило при анализе запроса доступа может выбираться по более точному описанию. Например, для любого процесса (маска «*») можно запретить олицетворение с системными пользователями всем интерактивным поль-
Контроль доступа к компьютерным ресурсам

эта учетной записи в систему. За корректность входа в систему отвечает системный процесс winlogon, использующий в своей работе сервисы олицетворения. Сначала этот процесс запускается под системной учетной записью, а после идентификации и аутентификации пользователя, олицетворяет себя с учетной записью авторизованного в системе пользователя, в результате чего загружается «рабочий стол» этого пользователя. В случае, если разрешить олицетворение системного пользователя для данного процесса только с определенными заданными в системе учетными записями, вход в систему под несанкционированный созданной учетной записью станет невозможным.

2. Рассмотрим пример угрозы, вызванной использованием сервисов олицетворения, вносимой при реализации сессионного контроля доступа. Если запустить приложение с правами другого пользователя с использованием утилиты runas (один из тех сотрудников обрабатывает информацию различных уровней конфиденциальности под различными учетными записями — пользователями), то между этим пользователем и исходным пользователем не будет разграничиваться буфер обмена, что не позволит корректно разделить режимы обработки информации различных уровней конфиденциальности в системе, как следствие, имеем безусловную технологическую уязвимость. Однако утилитой runas используются сервисы олицетворения, подобное олицетворение для субъекта доступа — процесса runas, должно соответствующим образом запрещаться.

Отметим, что в общем случае контроль доступа к сервисам олицетворения позволяет реализовывать дополнительные весьма важные возможности защиты. Проиллюстрируем сказанное примером. Системный процесс spool, ответственный за печать, запускается с правами системного пользователя и олицетворяет себя с тем пользователем, от лица которого (под учетной записью которого) осуществляется печать. Достаточно запретить подобное олицетворение для этого процесса, чтобы предотвратить возможность печати под соответствующей учетной записью.

Вообще надо сказать, что контроль доступа к сервисам олицетворения позволяет решать множество весьма актуальных современных задач защиты информации.

Рис. 3.40. Меню создания правил доступа к сервисам олицетворения

Рис. 3.41. Интерфейс отображения созданных правил олицетворения

Несколько важных замечаний по возможности (и необходимости) использования контроля доступа к сервисам олицетворения.

1. Как ранее отмечалось, ключевым требованием к построению безопасной системы является реализация разрешительной разграничительной политики доступа, предполагающей ее построение по принципу «Вся, что явно не разрешено, то запрещено». Это справедливо и при контроле доступа пользователей в систему. В системе создают-ся учетные записи (см. выше), под которыми разрешается вход. Но это гарантированно не ограничивает возможности несанкционированного создания новой учетной записи с последующим входом под
3.9. Контроль санкционированности свершившихся событий

В качестве одной из важных компонентов защиты информации от несанкционированного доступа рассматривается задача контроля целостности [11]. На практике контроль целостности, как правило, применяется в отношении файловых объектов (реже в отношении объектов реестра ОС). Цель подобного контроля состоит в периодическом (либо с использованием иных способов запуска процедуры контроля) контроле исполнимости объектов (существенно, что контроль целостности применяется исключительно в отношении объектов, идентифицируемых в процессе функционирования системы) с использованием соответствующих контрольных сумм, при обнаружении же несанкционированного изменения контролируемых объектов, они восстанавливаются из заранее созданных для них резервных копий.

Мы значительно расширим задачи механизмов контроля применительно к решению задачи контроля доступа к статическим объектам, в части реализации контроля санкционированности свершившихся событий. Данный подход к защите рассматривается именно в этой главе, поскольку любой контроль может осуществляться в отношении исключительно статических объектов — объектов, присутствующих в системе на момент создания администрацией системного архива свершившихся событий.

Ранее говорилось о контроле доступа субъектов к статическим объектам, мы подразумевали реализацию разграничительной политики доступа, предполагающей реализацию разграничений прав субъектов на получение доступа к защищаемым объектам. Данный подход реализован на предприятии с помощью создания для каждого субъекта списка доступа. Данный метод защиты направлен на предотвращение возможного несанкционированного доступа к защищаемым объектам. Данный метод позволяет реализовать реализацию разграничительной политики доступа, предполагающую реализацию разграничений прав субъектов на получение доступа к защищаемым объектам.

Однако, в общем случае, в рамках реализации контроля доступа к объектам, исходя из этого, можно сделать вывод о том, что в этом случае контроль доступа к объектам, исходя из этого, можно сделать вывод о том, что в этом случае контроль доступа к объектам является недостатком в системе несанкционированного доступа, поскольку в этом случае контроль доступа к объектам является недостатком в системе несанкционированного доступа.

Однако в общем случае, в рамках реализации контроля доступа к объектам, исходя из этого, можно сделать вывод о том, что в этом случае контроль доступа к объектам является недостатком в системе несанкционированного доступа, поскольку в этом случае контроль доступа к объектам является недостатком в системе несанкционированного доступа.

Отметим, что данные подходы к защите никак нельзя признать альтернативными, т.к. при достижении в принципе одноканальных целей, реализация, а, как следствие, и возможности защиты кардинально различаются. Так, при реализации разграничительной политики предотвращается сам факт появления несанкционированного события при контроле же, фиксируется уже произошедшее событие, т.е. в данном случае подобное событие может привести к определенным негативным последствиям. Величина последствий определяется, как критичность контролируемого события, так и оперативностью обнаружения несанкционированного события и формирования реакции на него.

Вместе с тем, при реализации контроля фактов свершившихся событий обеспечивается принципиально новое свойство и возможности защиты, в...
3.9.1. Контроль запускаемых в системе процессов

3.9.1.1. Контроль санкционированности запускаемых процессов

Суть контроля санкционированности запускаемых процессов состоит в следующем. Задаются полнотуемые имена исполняемых файлов системных и прикладных процессов, разрешенных к запуску. Также можно задавать папки, из которых разрешается запуск исполняемых файлов (можно использовать маски и переменные среды окружения). В процессе работы системы, процесса защиты с устанавливаемым временем интервалов контролирует какие-либо процессы (контролирует полнотуевые имена исполняемых файлов) запущены в системе на момент контроля, и завершает процесс, не подходящие под заданые правила запуска санкционированных процессов.

Как видим, данный метод защиты можно использовать в дополнение к обеспечению запускности программной среды, реализуемому соответствующим методом контроля доступа (рассмотрен ранее).

Например, можно разрешить в системе запуск исполняемых файлов только из каталогов Windows и Program Files (с таким способе защиты, применительно к реализации запускности программной среды, говорили ранее). В интерфейсе системы защиты заданные настройки санкционированных процессов отображаются в виде, представленном на рис. 3.42, задается санкционированный процесс (соответствующий исполняемый файл, каталог или маска), а также реакция на обнаружение несанкционированного события – завершение неразрешенного процесса, из интерфейса, представленного на рис. 3.43.

Рис. 3.42. Отображение заданных процессов, санкционированных для запуска в системе
Глава 3. Принципы и методы контроля доступа к статичным объектам

Суть контроля состоит в следующем. Задаются полномочия имен исполняемых файлов контролируемых процессов, которые обязательно должны быть запущены в системе. В процессе работы системы система защиты с устанавливаемым временным интервалом контролирует активность заданных обязательных процессов, и при обнаружении деактивации обязательного процесса, осуществляет его автоматический перезапуск (если такая реакция задана). Поскольку процессы исходно могут запускаться в системе с различными правами (системой, с правами различных пользователей), при администрировании средства защиты должно задаваться то, от лица какой учетной записи средств защиты автоматически перезапускается обязательный завершенный процесс.

В интерфейсе системы защиты заданные настройки обязательных процессов отображаются в интерфейсе, представленном на рис. 3.44, выбор учетной записи (в том числе, и системой), из под которой автоматически перезапускается завершенный обязательный процесс, иллюстрирован на рис. 3.46.

Рис. 3.44. Интерфейс отображения заданных обязательных процессов

Рис. 3.45. Интерфейс задания обязательных процессов

3.9.1.2. Контроль активности обязательных процессов

Разовьем мысль в части реализации защиты в отношении запуска санкционированных процессов. Для некоторых, разрешенных на запуск процессов, может быть критическим их санкционированное завершение злоумышленником. Например, это различные компоненты системы защиты, либо дополнительные средства защиты, не обеспечивающие собственной защиты от их санкционированной деактивации. Это и определенные системные процессы.
Замечание. Если же говорить об обеспечении активности собственного компонент системы защиты, то здесь может реализовываться следующий метод защиты [11]. Одна выделенная компонента системы защиты контролирует активность (при необходимости перезапускает) остальных компонент (обязательных процессов) системы защиты. При этом возникает угроза несанкционированной деактивации контролирующей компоненты. Ее активность должен осуществляться каким-либо внешним средством защиты. Техническое решение – метод аппаратной защиты, предотвращающий загрузку системы без системы защиты (в том числе, с любого внешнего источника) и работу системы при деактивации контролирующей программной компоненты системы защиты, рассмотрен в [11].

3.9.1.3. Реализация регламента работы с приложениями

Рассмотрим на примере дополнительные возможности, получаемые при реализации контроля санкционированности событий.

При работе с некоторыми приложениями, например, обеспечивающими доступ в сеть Интернет, на предприятии может вводиться регламент. Этот регламент может технически реализовываться соответствующим механизмом контроля санкционированности событий.

Суть контроля состоит в следующем. Задаются полнопутевые имена исполняемых файлов контролируемых процессов, работа с которыми регламентируется в системе. Регламентирование состоит в задании дней недели (либо дне месяца) и времени (либо продолжительности), в течение которого (с/пд, либо продолжительности) для определенных дней недели в системе может использоваться данное приложение.

В процессе работы системы, система защиты с устанавливаемым временем интервалом контролирует выполнение регламента работы с контролируемым приложением, разрешая запускать контролируемое приложение и работать с этим приложением только в отведенные часы определенных дней недели/месца, в течение заданного интервала времени. При выполнении регламента приложение, либо не будет запущено, либо автоматически завершится системой защиты.

Рассмотрим техническую реализацию данного решения. В интерфейсе системы защиты заданные настройки контролируемых процессов отображаются в виде, представленном на рис. 3.47, задается контролируемый процесс и регламент его использования из интерфейса, представленного на рис. 3.48.

Рис. 3.46. Вкладка задания учетной записи для автоматического запуска обязательного процесса

Рис. 3.47. Интерфейс отображения заданных процессов, использование которых регламентируется в системе, и регламент их использования

Рис. 3.48. Интерфейс задания процессов, использование которых регламентируется в системе, и регламент их использования
3.9.2. Исполнение задачи и реализация контроля целостности файловых объектов и объектов реестра

Аналогично тому, как может быть в системе в обход разграничительной политики доступа запущен несанкционированный процесс, либо удалён обязательный процесс, могут быть несанкционированно модифицированы файловые объекты, в том числе системные объекты — исполняемые файлы системы и приложений, файлы настройки системы и приложений, объекты реестра OS. Поэтому, в дополнение к разграничительной политике доступа целесообразно реализовывать контроль целостности (неизменности) соответствующих файловых объектов и объектов реестра.

Замечание. Контроль целостности файловых объектов и объектов реестра при рассматриваемой постановке задачи защиты информации от несанкционированного доступа в общем виде можно интерпретировать, как контроль санкционированности событий, но уже в общем случае — контролируется любая возможность, которая может привести к несанкционированной модификации/удалению этих объектов, т.е. контроль в этом случае реализуется не применительно к конкретному событию (например, исполнение санкционированных процессов), а применительно к любому событию, которое потенциально может привести к несанкционированной модификации/удалению контролируемых объектов. Т.е. контроль целостности файловых объектов и объектов реестра OS может интерпретироваться, как верхний (последний) уровень контроля санкционированности событий [11].

В качестве реакции на несанкционированную модификацию файла реализуется автоматическое восстановление содержимого модифицированного файла из резервной копии, создаваемой при постановке файла на контроль, то же можно сказать и применительно к соответствующему объекту реестра.

Глава 3. Принципы и методы контроля доступа к статическим объектам

Важнейшим вопросом технической реализации контроля целостности файловых объектов является выбор способа запуска процедуры контроля. Естественно, возможен синхронный (с заданным временным интервалом) и асинхронный контроль. Асинхронный контроль предполагает запуск процедуры в отношении каждого контролируемого файла при обращении к этому файлу.

Конечно, по聂се, целесообразен асинхронный контроль. Но на практике он мало применим. Дело в том, что подобный контроль должен быть реализован в драйвере (процедура контроля должна запускаться при перехвате диспетчером запроса доступа к контролируемому файловому объекту). Процедура контроля весьма ресурсоемкая, запуск ее при каждом обращении к контролируемому файлу (а это ведь исполняемые и системные файлы), причем на уровне драйвера, приведет к невозможности нормального (без существенного, заметного «на глаз» торможения) функционирования системы.

Остаётся остановить свой выбор на синхронной процедуре контроля, понимая при этом, что интервал контроля не может быть очень мал, что в свою очередь, скажется на оперативности реакции.

Суть контроля состоит в следующем. Задаются контролируемые файловые объекты, при этом (если задается режим автоматического восстановления) автоматически создается их резервная копия.

Рис. 3.49. Интерфейс задания контролируемых файловых объектов
Контроль доступа к компьютерным ресурсам

В процессе работы системы средства защиты с устанавливающимся временным интервалом контролирует целостность заданных объектов, автоматически восстанавливая несанкционированно модифицированные файлы из резервной копии.

Контролируемые файловые объекты (это может быть либо отдельный файл, либо папка, тогда контролируются все файлы из заданной папки) задаются в интерфейсе, приведенном на рис. 3.49. Заданные контролируемые файловые объекты и их контрольные суммы отображаются в интерфейсе, представленном на рис. 3.50. Интервал контроля целостности задается из интерфейса, приведенного на рис. 3.51.

Рис. 3.50. Интерфейс отображения контролируемых файловых объектов и значений их контрольных сумм

Замечание. При установке на контроль целостности файлового объекта "Папка", контролируется не только содержимое файлов, находящихся в этой папке, с возможностью автоматического их восстановления из резервных копий, но и собственно файлы, находящиеся в этой папке, с возможностью автоматического восстановления файла (создания файла) из резервной копии, при его несанкционированном удалении или переименовании.

Рис. 3.51. Интерфейс задания интервала контроля целостности

Контролируемые объекты реестра OC (это могут быть как ключи, так и ветви реестра), контроль целостности осуществляется по полной аналогии с контролем целостности файловых объектов, задаётся в интерфейсе, приведенном на рис. 3.52. Заданные контролируемые объекты реестра и их контрольные суммы отображаются в интерфейсе, представленном на рис. 3.53. Интервал контроля целостности задается из интерфейса, приведенного на рис. 3.51.

Рис. 3.52. Интерфейс задания контролируемых объектов реестра OC

Рис. 3.53. Интерфейс отображения контролируемых объектов реестра OC и значений их контрольных сумм
3.10. Выводы по третьей главе

1. Введена классификация объектов доступа, которые подразделены на статичные, присутствующие на момент задания администратором разграничительной политики доступа и создаваемые – создаваемые уже впоследствии, в процессе функционирования информационной системы.

2. Определены принципиальные противоречия и недостатки использования методов контроля доступа к статичным объектам с целью защиты от угроз безопасности информационной системы, распространенных на практике. В результате чего сделан вывод об актуальности задачи разработки принципиально новых подходов к реализации защиты и методов контроля доступа к создаваемым объектам.

3. Сформулированы принципы контроля доступа к статичным объектам, реализация которых направлена на построение безопасной системы, для обеспечения которой должны рассматриваться при реализации контроля доступа к различным (разнородным) объектам доступа.

4. Сформулированы требования к построению безопасной системы, в том числе, требования к назначению непротиворечивых правил доступа, которые должны выполняться при реализации контроля доступа к статичным объектам. В частности, эти требования должны выполняться при установке правил доступа пользователей к объектам администратором, что делает управление угрозами безопасности возможным.

5. Разработаны, запатентованы и апробированы технические решения, реализующие методы контроля доступа к статичным объектам.

6. Разработаны модели и метод контроля доступа, реализуемого перенаправлением запросов доступа, направленный на решение задачи обеспечения безопасности информационной системы и приложений объектов. Показано, что без реализации данного метода в разграничительной политике доступа невозможно построение корректной разграничительной политики доступа к системным объектам, поскольку в этом случае будет присутствовать угроза безусловной технологической уязвимости.

7. Разработано, запатентовано и апробировано техническое решение, реализующее метод контроля доступа, реализуемого перенаправлением запросов доступа.
13. Сформулирована задача и разработаны методы (примеры методов) контроля санкционированности событий, которые могут быть реализованы в системе защиты информации от несанкционированного доступа в дополнение к методам контроля доступа, направленные на обнаружение и нивелирование (уменьшение негативных последствий) неизвестных на момент создания системы защиты, но потенциально возможных угроз технологических уязвимостей. Сделан вывод о том, что разработка методов контроля санкционированности событий представляет собой отдельное достаточно важное направление реализации защиты информации от несанкционированного доступа в предлагаемой ее постановке в общем виде, соответственно, отдельную технологию защиты информации от несанкционированного доступа.

---

ГЛАВА 4. ПРИНЦИПЫ И МЕТОДЫ КОНТРОЛЯ ДОСУПА К СОЗДАВАЕМЫМ ОБЪЕКТАМ. ТЕХНОЛОГИЯ ЗАЩИТЫ ДАННЫХ

4.1. Принципы контроля доступа к создаваемым объектам

В предыдущей главе нами была введена очень важная классификация объектов, мы их разделили на статичные и создаваемые в процессе работы пользователями [30-32].

Там же мы обратили внимание читателя на то, что в чем состоит основное противоречие при реализации контроля доступа к создаваемым объектам, на примере методов контроля доступа субъектов к статичным файловым объектам. А основное противоречие состоит в том, что в разграничительной политике доступа присутствуют и сущность субъект, и сущность объект — разграничиваются права доступа субъектов к объектам. Однако на момент назначения администратором разграничений еще отсутствуют создаваемые в процессе работы пользователей файлы, доступ к которым должен контролироваться, в первую очередь, т.к. в них записывается обрабатываемая на компьютере информация, в том числе, конфиденциальная. К чему тогда разграничивать доступ субъектов, если объекты отсутствуют?

Рассмотрим и существующий подход к реализации контроля доступа к создаваемым объектам (на примере файлов), состоящий в следующем. Поскольку на момент задания разграничительной политики доступа создаваемых в процессе работы пользователей файлов еще не существует в системе, администратором заранее создаются хранилища (своего рода «контейнеры») для последующего хранения создаваемых в процессе работы пользователей файлов. Т.е. администратором создаются папки (контейнеры), к которым и разграничивается доступ. Объект доступа «файл» в общем случае при этом исключает из разграничительной политики доступа. Разграничительной политикой для созданных папок-контейнеров, пользователи «принуждаются» создавать свои файлы только в определенных папках.
Созданные файлы следуют разграничению, установленные для папок. По средству же реализации разграничительной политики доступа к папкам-конкейнерам для пользователей разграничения доступа и к созданным в процессе функционирования системы файлам.

Естественно, что подобный подход (а именно он сегодня и используется на практике) не только весьма не логичен (если речь не идет о системных объектах), но и обусловливает принципиальное усложнение реализации разграничительной политики доступа, в первую очередь, это проявляется при реализации процессной модели контроля доступа, а в ряде случаев, и к невозможности ее корректной реализации (обо всем этом мы говорили ранее). Достаточно задуматься о том, какие действия потребуется выполнять, чтобы обеспечивалась администратору, чтобы, например, изолировать объекты информации десятком приложений — для каждого субъекта потребуется создавать свою папку, далее разграничивать к созданным папкам права доступа. Соответственно придается разграничивать доступ и к иным папкам, в частности, к системным. А не разделяемые системой и приложениями папки — их потребуется разделить отдельным механизмом защиты, при этом опять же создать соответствующие дополнительные папки, и т. д. Все это приводит не только к многократному усложнению задачи администрирования, как следствие, к ошибкам администрирования (что представляет собой весьма актуальную в данном случае угрозу), но и к невозможности гарантированно построить корректную разграничительную политику доступа к файловым объектам в общем случае. Возможность же корректной реализации в данном случае (ввиду наличия системных объектов) мандатного контроля доступа вообще ставится под сомнение.

Определение. Под контролем доступа к создаваемым объектам понимается контроль доступа, основанный на исключении сущности «объект доступа» из разграничивительной политики доступа (объект доступа не используется при задании прав доступа), за счет реализации в автоматическом режиме атрибутов создаваемых объектов, используемой для последующего контроля к ним доступа.

Причины контроля доступа к создаваемым файловым объектам основаны на исключении сущности «объект доступа» из разграничительной политики доступа к файловым объектам, как таковой (ввиду ее отсутствия на момент задания прав доступа администратором), и состоят они в следующем [30-32]:

1. Сущность «объект» исключается из схемы контроля доступа, при реализации разграничительной политики используются две сущности: идентификатор (учетная информация) субъекта, создавшего объект, и идентификатор субъекта, запрашивающего доступ к созданному объекту.

2. Правила доступа устанавливаются между субъектами: «субъект доступа (учетная информация), запрашивающий доступ к объекту» и «субъект доступа (учетная информация), создавший этот объект».

3. При создании субъектом нового объекта, этим объектом наследуется учётная информация субъекта доступа, созданного этого объекта.

4. При запросе доступа к любому объекту, диспетчер доступа анализирует наличие, а при наличии, содержимое унаследованной объектом учетной информации созданного его субъекта доступа. При наличии, анализирует заданные правила доступа, в результате чего предоставляет запрошенный субъектом доступ, либо отказывает в нем. При отсутствии — анализирует правило доступа к неразмещенным (не унаследовавшим учетную информацию субъектом) объектам.

Замечание. В общем случае данные принципы контроля доступа, предполагающие автоматическую разметку объектов при их создании — наследование объектом учетной информации субъекта доступа, могут реализовываться не только в отношении создаваемых объектов. Автоматически размещаться по определенным правилам могут и статические объекты, например, системные файлы, что позволяет решать достаточно важные задачи защиты системных объектов [37, 44]. Например, автоматически размещаться исполняемые файлы (при их запуске) с предотвращением задаваемого правилами доступа возможности их удаления или модификации [26, 34]. Эти вопросы мы рассмотрим далее.

В двух словах остановимся на тему того, где хранить унаследованную файлом (в качестве объекта доступа рассматриваем файл) учетную информацию субъекта доступа, созданного этого файл. Возможны различные варианты, каждый из которых имеет свои ограничения по использованию. Вариант хранения подобной информации в отдельной таблице (в отдельном файле) не стоит рассматривать по двум причинам. Во-первых, подобная таблица очень быстро разрастается, и ее анализ диспетчером при каждом запросе доступа приведет к существенному влиянию на загрузку вычислительного ресурса. Во-вторых, при утере подобной таблицы, восстановление разграничительной политики доступа станет крайне трудоемкой работой.

Другой вариант — это хранение учетной информации в дополнительных (резервных) атрибутах. Подобную возможность предоставляет, например, файловая система NTFS — возможность использования альтернативных потоков. Однако, при сохранении подобным образом размещения файлов в файловой системе, не предоставляющей рассматриваемой возможности,
например, FAT, учетная информация, унаследованная файлом, не может быть сохранена.

Достаточно интересен вариант сохранения учетной информации непосредственно в файле. Интересен он своей общейностью — учетная информация будет передаваться (принадлежать файлу) при любом способе передачи файла (сохранение в любой файловой системе, по сети и т.д.). Это позволит реализовать разграничительную политику доступа на одном отдельном компьютере, а единую разграничительную политику доступа в рамках всей распределенной информационной системы, т.к. любым способом перенесенный файл с одного компьютера на другой, унаследует при переносе информации учетную информацию созданного его субъекта, как следствие, доступ к нему может быть разграничен на любом компьютере [55]. Особенно это интересно при реализации в информационной системе мандатного (сессионного) метода контроля доступа.

Однако данный способ имеет и существенный недостаток. В случае его реализации, файл при сохранении искается — модифицируется дискетчержером доступа, который записывает непосредственно в файл учетную информацию (в том числе, метку безопасности) созданного файл субъекта. Как следствие, на компьютере, на котором не установлено средство защиты, реализующее метод контроля доступа к создаваемым файловым объектам (например, при обработке открытой информации, файлы могут обрабатываться, как на защищенных, так и на незащищенных, в том числе, на домашних компьютерах), подобный файл будет прочитан в исключенном виде (модифицированный дискетчером доступа при записи). Таким образом, данный способ разметки файлов предполагает возможность их обработки только на защищенных компьютерах.

Заметим, что данный недостаток отсутствует при сохранении учетной информации в дополнительных (резервных) атрибутах, т.к. система, не использующая эти атрибуты, не будет их запрашивать.

Можем сделать следующий вывод. В случае, когда контроль доступа к создаваемым объектам реализуется в рамках построения процессной модели контроля доступа — для разделения прав доступа к создаваемым объектам на одном и том же компьютере между различными процессами (приложениями) без необходимости их дополнительной криптографической защиты, целесообразно хранение учетной информации субъекта доступа в дополнительных (резервных) атрибутах, в случае же, когда используется криптографическая защита создаваемых объектов, а, в первую очередь, в такой защите нуждаются файлы, создаваемые на внешних накопителях и/или передаваемые по сети, уже имеет смысл реализация хранения учетной информации субъекта доступа непосредственно в файле.

**Глава 4. Принципы и методы контроля доступа к создаваемым объектам**

Отметим, что существенным достоинством в обоих случаях является непосредственная привязка учетной информации субъекта, созданного объекта, к объекту. Учетная информация хранится либо в атрибутах объекта, либо непосредственно в объекте, что снимает проблему корректности идентификации правил доступа к объекту доступа при анализе запроса доступа [40] имеющую место для методов контроля доступа субъектов к статичным объектам при хранении прав доступа (матрицы доступа) в отдельном объекте (рассмотрена ранее).

**4.2. Методы и модели контроля доступа**

**4.2.1. Контроль доступа на основе матрицы доступа**

**Замечание**. Исследование будем проводить на примере файловых объектов, а далее расширим понятие создаваемых объектов, к которым следует контролировать доступ рассматривающими методами.

В качестве контролируемого объекта здесь рассматривается именно создаваемый файл (какого-либо контролируемого, равно как и их создания административинатором, не требуется), как объект, непосредственно содержащий защищаемые данные. При создании субъектом нового файла (либо при модификации контролируемым субъектом доступа не размеочечного ранее файла), файлом наследуется учетная информация субъекта доступа, созданного этого файла [30,36]. Учетная информация субъекта доступа, как отмечалось ранее, в общем случае определяется следующим тремя сущностями: первичный идентификатор пользователя (учетная запись, под которой осуществлен вход в систему); полномочное имя процесса, обращающегося к ресурсу; эффективный идентификатор пользователя (учетная запись, от которой осуществлен запрос доступа к ресурсу).

При запросе доступа к любому файлу, дискетчер доступа анализирует наличие, а при наличии, содержимое унаследованной файлом учетной информации субъекта доступа. При отсутствии подобной информации, дискетчер либо разрешает запрошенный доступ (если доступ к такому объекту им не разграничивается), либо реализует иное заданное правило контроля доступа к неразмещенным объектам, при наличии, анализирует матрицу доступа, в которой определено, какие права доступа к объекту (характеризуется учетной информацией, созданного субъекта) разрешены субъекту (определяется его учетной информацией), запрашивающему доступ к этому объекту.
Матрица доступа в данном случае приобретает совершенно иной вид, т.к. из разграничительной политики доступа исключена сущность «объект» [30,36].

Если считать, что множество $C = \{C_1, ..., C_n\}$ линейно упорядоченное множество субъектов доступа, а $R = \{R_1, ..., R_m\}$ конечное множество прав доступа (чтение ($r$), запись ($w$), удаление ($d$), исполнение ($x$) и т.д., отсутствие прав доступа ($0$)) субъекта $C_i$ к объекту, созданному субъектом $C_j; i = 1, ..., l, j = 1, ..., l$, то матрица доступа $M$, используемая для реализации разграничительной политики методом контроля доступа с принудительным управлением потоками информации имеет следующий вид (условился в строках матрицы указывать учетную информацию субъектов, запрашивающих доступ к объектам, а в столбцах — учетную информацию субъектов, унаследованную созданными объектами):

$$M = \begin{bmatrix}
C_1 & C_2 & \cdots & C_n \\
C_1 & r & w & d & w & \cdots & 0 \\
C_2 & r & w & d & w & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
C_{l-1} & 0 & 0 & \cdots & 0 & r \\
C_l & 0 & w & \cdots & r & w & d \\
\end{bmatrix}$$

В любой момент времени система описывается своим текущим состоянием $Q = \{C, C, M\}$, $M[C, C]$ — ячейка матрицы, содержит набор прав доступа. Будем обозначать $C_i(R)C_j$ разрешением права доступа субъекту $C_i$ к объекту, созданному субъектом $C_j; i = 1, ..., l, j = 1, ..., l$, где $R = \{x, w, r, d\}$: исполнение ($x$), запись ($w$), чтение ($r$), удаление ($d$).

Рассмотрим принципиальные отличия метода контроля доступа к создаваемым файловым объектам:

1. Поскольку сущность «Владение» априори исключена из схемы контроля доступа, дискретный метод контроля доступа к файловым объектам является методом контроля доступа с принудительным управлением потоками информации, следствием чего, его использование, при выполнении соответствующих требований, которые будут сформулированы далее, позволяет построить безопасную систему.

2. Поскольку объект доступа не фигурирует в разграничительной политике доступа к ресурсам, не контролируется сам факт создания файла (контролируются последующие обращения к созданному файлу), не разграничивается, на каком диске, на каком накопителе, в какой папке может создаваться файл субъектом доступа.

3. Поскольку создание новых файлов не контролируется (контроль осуществляется к созданным файлам), метод не реализует разграничение доступа к объектам, в части создания в них новых файлов субъектами доступа.

4. Поскольку каждый созданный файл однозначно идентифицируется учетной информацией создавшего его субъекта доступа, контроль доступа к создаваемым файлам может быть реализован корректно в общем случае, вне зависимости от того, в каких объектах сохраняются файлы, в том числе, это относится и к объектам, не разделяемым системой и приложениями.

Очевидным преимуществом данного метода, применительно к контролю доступа к создаваемым файлам, по сравнению с методом контроля доступа к статическим файловым объектам, является простота назначения разграничительной политики (администрирования). В этом случае не требуется дополнительного создания каких-либо папок контейнеров, разделения каких-либо общих папок и т.д.

Крайне эффективно применение рассматриваемого метода защиты для изолирования режимов обработки информации при реализации ролевой модели контроля доступа и для изолирования работы процессов в рамках реализации процессной модели контроля доступа.

Проявляют пример. В табл. 4.1 приведен пример разграничительной политики, позволяющей изолировать работу интернет-браузера Internet Explorer [49, 54]. Первым правилом в разграничительной политике задается то, что файлы, созданные интернет-браузером могут быть только прочитаны или/и записаны интернет-браузером (исключена возможность выполнения). Вторым правилом задается то, что любым иным процессом (приложением) доступ к этим файлам запрещен, вне зависимости от того, где (в каких папках) были созданы эти файлы. Видите, как все просто!

Замечание. Рассматриваем реализацию разрешительной разграничительной политики доступа к ресурсам: «Все, что явно не разрешено — не прописано в правилах, то запрещено». Субъект доступа может задаваться, в том числе, с использованием масок и переменных среды окружения.

Более сложный пример разграничительной политики проиллюстрирован в табл. 4.2.
Таблица 4.1. Пример разграничительной политики доступа к создаваемым файловым объектам

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, создающего контролируемый файловый объект</th>
<th>Учетная информация субъекта, запрашивающего доступ к контролируемому файловому объекту, и разрешенные права доступа субъекта к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>; C:\Program Files\Internet Explorer\explore.exe,</em></td>
<td><em>; C:\Program Files; Internet Explorer\explore.exe,</em> (Зн/Чт)</td>
</tr>
<tr>
<td>2</td>
<td><em>; C:\Program Files\Internet Explorer\explore.exe,</em></td>
<td>*; * (−)</td>
</tr>
</tbody>
</table>

Таблица 4.2. Пример разграничительной политики доступа к создаваемым файловым объектам

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, создающего объекты, доступ к которым разграничивается</th>
<th>Учетная информация субъекта, имеющего право доступа к контролируемым созданным объектам, и права доступа субъекта к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User1,*; User1</td>
<td>User1,*; User1 (Зн/Чт)</td>
</tr>
<tr>
<td>2</td>
<td>User1,*; User1</td>
<td>*; * (−)</td>
</tr>
<tr>
<td>3</td>
<td>User1; C:\Program Files; Internet Explorer\explore.exe, User1</td>
<td>User1; C:\Program Files; Internet Explorer\explore.exe, User1 (Зн/Чт)</td>
</tr>
<tr>
<td>4</td>
<td>User1; C:\Program Files; Internet Explorer\explore.exe, User1</td>
<td>User1,*; User1 (−)</td>
</tr>
<tr>
<td>5</td>
<td>−</td>
<td>*; * (Зн/Чт/Чт)</td>
</tr>
</tbody>
</table>

Переднее включение определяет, что файлы, созданные пользователем User1, при обращении к файлу под учетной записью User1 (не произошло смены учетной записи при доступе к файлу — не было однотворчества) любым процессом (процесс в субъекте доступа задачой не выполняется), могут быть, как прочитаны, так и модифицированы (если не исполнены — это в части защиты от взлома вредоносных программ) только пользователем User1, причем опять же при отсутствии однотворчества (это в части защиты от изменения только в привилегий пользователя при доступе к файловым объектам), любым процессом. Разрешительная разграничительная политика при этом задается вторым правилом — любой иной пользователь (используется маска "*") не сможет получить доступ к файлам, созданным пользователем User1 (применяемое снятием доступа правило выбирается по более точным опи-
Пример предполагающий добавление права чтения пользователем User2 файловых объектов, созданных пользователем User1, приведен в табл. 4.4.

Таблица 4.4. Пример разграничительной политики доступа к создаваемым файловым объектам

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, имеющего право доступа к создаваемым объектам, и права доступа субъекта к объекту</th>
<th>Учетная информация субъекта, созданного объекта, доступ к которому разграничивается</th>
<th>Учетная информация субъекта, созданного объекта, доступ к которому разграничивается</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Administrator,* , Administrator</td>
<td>User2,* , User2(Чт/И)</td>
<td>User2,* , User2(Чт/И)</td>
</tr>
<tr>
<td>7</td>
<td>* *</td>
<td>System,* (Зн/Чт/И)</td>
<td>System,* (Зн/Чт/И)</td>
</tr>
<tr>
<td>8</td>
<td>User1,* , User1</td>
<td>System,* (→)</td>
<td>System,* (→)</td>
</tr>
<tr>
<td>9</td>
<td>User2,* , User2</td>
<td>System,* (→)</td>
<td>System,* (→)</td>
</tr>
<tr>
<td>10</td>
<td>* *</td>
<td>Administrator,* , Administrator (Зн/Чт/И)</td>
<td>Administrator,* , Administrator (Зн/Чт/И)</td>
</tr>
<tr>
<td>11</td>
<td>User1,* , User1</td>
<td>Administrator,* , Administrator (→)</td>
<td>Administrator,* , Administrator (→)</td>
</tr>
<tr>
<td>12</td>
<td>User2,* , User2</td>
<td>Administrator,* , Administrator (→)</td>
<td>Administrator,* , Administrator (→)</td>
</tr>
<tr>
<td>13</td>
<td>System,* , System</td>
<td>Administrator,* , Administrator (Чт/И)</td>
<td>Administrator,* , Administrator (Чт/И)</td>
</tr>
<tr>
<td>14</td>
<td>* *</td>
<td>Administrator,* , Administrator (→)</td>
<td>Administrator,* , Administrator (→)</td>
</tr>
</tbody>
</table>

Теперь, что касается системных объектов [37]. Те объекты, которые модифицируются системой в процессе работы (заметим, что при создании/установке системы контроль доступа еще не действует, поэтому изначально системные объекты окажутся неразграниченными), размечаются - им присваивается учетная информация субъекта, пользователем которой выступает System. Т.е. можно реализовать разграничительную политику и к системным объектам. Приложения в системе устанавливаются пользователем Administrator, следовательно, наследуют его учетную информацию - к исполняемым файлам приложений так же может разграничиваться доступ. Пример соответствующей разграничительной политики приведен в табл. 4.5.

Таблица 4.5. Пример разграничительной политики с контролем доступа к системным файловым объектам

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, имеющего право доступа к создаваемым объектам, и права доступа субъекта к объекту</th>
<th>Учетная информация субъекта, созданного объекта, доступ к которому разграничивается</th>
<th>Учетная информация субъекта, созданного объекта, доступ к которому разграничивается</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User1,* , User1</td>
<td>User1,* , User1 (Зн/Чт/И)</td>
<td>User1,* , User1 (Зн/Чт/И)</td>
</tr>
<tr>
<td>2</td>
<td>User2,* , User2</td>
<td>User2,* , User2 (Зн/Чт/И)</td>
<td>User2,* , User2 (Зн/Чт/И)</td>
</tr>
<tr>
<td>3</td>
<td>System,* , System</td>
<td>User1,* , User1 (→)</td>
<td>User1,* , User1 (→)</td>
</tr>
<tr>
<td>4</td>
<td>System,* , System</td>
<td>User2,* , User2 (Чт/И)</td>
<td>User2,* , User2 (Чт/И)</td>
</tr>
<tr>
<td>5</td>
<td>Administrator,* , Administrator</td>
<td>User1,* , User1 (Чт/И)</td>
<td>User1,* , User1 (Чт/И)</td>
</tr>
</tbody>
</table>

Вместе с тем, ряд системных объектов может продолжительное время быть неразмеченными, к ним также необходимо контролировать доступ, поскольку права доступа к системным объектам для пользователей и системы должны кардинально различаться. Пример соответствующей разграничительной политики приведен в табл. 4.6.

Замечание. При необходимости, пользователю Administrator можно добавить право записи в неразграниченные объекты.

Таблица 4.6. Пример разграничительной политики с контролем доступа к неразграниченным файловым объектам

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, имеющего право доступа к создаваемым объектам, и права доступа субъекта к объекту</th>
<th>Учетная информация субъекта, созданного объекта, доступ к которому разграничивается</th>
<th>Учетная информация субъекта, созданного объекта, доступ к которому разграничивается</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>–</td>
<td>User1,* , User1 (Чт/И)</td>
<td>User1,* , User1 (Чт/И)</td>
</tr>
<tr>
<td>2</td>
<td>–</td>
<td>Administrator,* , Administrator (Чт/И)</td>
<td>Administrator,* , Administrator (Чт/И)</td>
</tr>
<tr>
<td>3</td>
<td>–</td>
<td>System,* (Зн/Чт/И)</td>
<td>System,* (Зн/Чт/И)</td>
</tr>
</tbody>
</table>
4.2.2. Контроль доступа на основе меток безопасности (мандатов)

В части мандатного (сессионного) метода контроля доступа к создаваемым файлам можем сразу отметить, что именно данный метод может быть крайне эффективен виду собственного своего предназначения — обеспечивать контроль доступа к категорированный информации, которая в свою очередь сохраняется именно в создаваемых в процессе функционирования системы файлах [31].

Напомним, что под мандатным контролем доступа к статическим объектам понимается способ обработки запросов диспетчером доступа, основанный на формальном соответствии диспетчеру с заданными правилом меток безопасности (мандатов), называемых субъектам и объектам доступа. Метки безопасности, как правило, являются элементами линейно упорядоченного множества $M = \{M_1, \ldots, M_I\}$ и служат для формализованного представления каких-либо свойств субъектов и объектов.

Разграничение же доступа диспетчером реализуется на основе задаваемого правила, определяющего отношение линейного порядка на множестве $M$, где для любой пары элементов из множества $M$, всегда один из типов отношения: $>, =, \leq$ (на практике реализуется выбор подмножества $M$ изоморфного конечному подмножеству натуральных чисел — такой выбор делает естественным арифметическое сравнение меток безопасности). Правило сравнения меток также назначается из каких-либо свойств субъектов и объектов, применительно к решаемой задаче защиты информации.

Наиболее широкое использование мандатного метода нашло применение практики секретного делопроизводства в компьютерной обработке информации. Глава 4. Принципы и методы контроля доступа к создаваемым объектам

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 $M$.

Используем следующие обозначения:

- $M_s$ – метка безопасности субъекта (группы субъектов) доступа;
- $M_o$ – метка безопасности объекта (группы объектов) доступа.

Модель Беля-Лападулы применяется с целью защиты от нарушения конфиденциальности информации, что обеспечивается реализацией следующих правил, направленных на защиту от понижения категории обрабатываемой информации:

1. Субъект $C$ имеет доступ к объекту $O\inuzzer$ в режиме «Чтения» в случае, если выполняется условие: $M_C \leq M_O$.
2. Субъект $C$ имеет доступ к объекту $O\inuzzer$ в режиме «Записи» в случае, если выполняется условие: $M_C = M_O$.

Другими словами, записывать информацию субъект может только категории своего уровня допуска, читать же, не превышающего своего уровня допуска.

Это касается правил иерархической обработки меток безопасности (см. первую главу). В случае реализации иерархической обработки меток безопасности, вступают в силу следующие правила:

1. Субъект $C$ имеет доступ к объекту $O\inuzzer$ в режиме «Чтения» и «Записи» в случае, если выполняется условие: $M_C = M_O$.
2. Субъект $C$ не имеет доступ к объекту $O\inuzzer$ в случае, если выполняется условие: $M_C \leq M_O$.

Теперь о методе мандатного контроля доступа к создаваемым файловым объектам [31]. В качестве контролируемого объекта здесь рассматривается создаваемый файл (какого-либо контроля папок, равно как и их создания администратором, не требуется), как объект, непосредственно содержащий защитляемую информацию. Задание разграничительной политики доступа состоит исключительно в назначении меток безопасности субъектам $M_C$ (что кардинально упрощает задачу администрирования). При создании файлом нового файла, файлом следует учета информация субъекта доступа — его метка безопасности $M_C$ (обозначим унаследованную метку $M_{co}$ при этом $M_{CO} = M_C$).

При запросе доступа к любому файлу, диспетчер доступа анализирует наличие, а при наличии, собственно значение метки безопасности $M_{co}$ уна-
слеованной данным файлом. При наличии метки у файла — $M_{фвф}$, диспетчер сравнивает эту метку с меткой субъекта, запрашивющего доступ к файлу. $M_{ф}$ — анализирует выполнение заданных правил контроля доступа. В результате анализа этой информации, с учетом реализуемых правил контроля доступа, диспетчер либо разрешает запрошенный субъектом доступ к файлу, либо отказывает в нем.

Основополагающие правила, направленные на защиту от нежелательного категория обрабатываемой информации, в данном случае имеет тот же вид, что и для метода контроля доступа к статическим файловым объектам:

1. Субъект $C$ имеет доступ к объекту $O$ в режиме «Чтения» в случае, если выполняется условие: $M_{ф} < M_{фф}$.

2. Субъект $C$ имеет доступ к объекту $O$ в режиме «Записи» в случае, если выполняется условие: $M_{ф} = M_{фф}$.

Иллюстрирующая базовые правила мандатного метода контроля доступа к создаваемым файлам матрица доступа $M_{ф}$ уже имеет принципиально иной вид:

$$
M_{ф} = \begin{bmatrix}
C_{1}(M_{ф}) & C_{2}(M_{ф}) & \ldots & C_{n}(M_{ф}) \\
C_{1}(M_{ф}) & C_{2}(M_{ф}) & \ldots & C_{n}(M_{ф}) \\
\vdots & \vdots & \ddots & \vdots \\
C_{1}(M_{ф}) & C_{2}(M_{ф}) & \ldots & C_{n}(M_{ф})
\end{bmatrix}
$$

Как видим, в разграничительной политике доступа присутствуют только субъекты и их метки безопасности.

Аналогичным образом задается матрица доступа (диагональная матрица) и для метода сессионного контроля доступа.

Вдумайтесь в то, насколько простой становится задача администрирования — требуется лишь назначить метки безопасности субъектам (для мандатного метода, в общем случае, пользователям) и все! Никакие объекты разделять, а уж тем более разделять между субъектами доступа, не требуется! При этом заметим, что метод мандатного контроля доступа исходно предназначен исключительно для контроля доступа к обрабатываемой информации, т.e. к создаваемым объектам (для решения каких-либо системных задач защиты — для реализации контроля доступа к системным объектам, он априори не приспособлен).

4.3. Требования к построению безопасной системы

4.3.1. Базовые требования

4.3.1.1. Требования к методу дисcretionного контроля доступа

Базовые требования во многом совпадают с аналогичными требованиями для метода контроля доступа к статическим файловым объектам, с поправкой на то, что в разграничительной политике доступа отсутствует сущность «объект» доступа (т.e. матрица доступа имеет иной вид). Вместе с тем, ввиду специфики контроля (не контролируется доступ на запись к статическим объектам — например, в которых могут создаваться файлы различных подсхем) появляются дополнительные требования [38].

Сначала без копирования (доказательства представлены в третьей главе) приведем требования к методу дисcretionного контроля доступа, совпадающие (с соответствующей поправкой) с требованиями, сформулированными на этапе для метода дисcretionного контроля доступа к статическим файловым объектам:

1. Не допустимо разрешение пользователям права исполнения (r) созданного в процессе функционирования системы файла. При реализации данного требования система безопасна относительно права исполнения (r). Заметим, что на реализации именно этого требования основана эффективная защита от вредоносных программ.

**Замечание.** Как следствие, к разрешенным правам доступ к создаваемым файловым объектам относятся: $R = \{w, r, d\}$.

2. При назначении разграничительной политики «по умолчанию» должны быть установлены права доступа: $C_{i}(w, r, d)C_{i} = 1, \ldots, l$. Данное правило обусловливает задание диагональной («канонической») матрицы доступа, характеризуемой условием: $C_{i}(w, r, d)C_{j} = 1, \ldots, l; C_{i}(0)C_{j} = 1, \ldots, l, j = 1, \ldots, l$.

**Замечание.** Реализовывая каноническую матрицу доступа система безопасна относительно права записи (w) и относительно права чтения (r).
3. При расширенной канонической матрице доступа разрешенным правом записи (w): \( C(w)C_i, i \neq j \), при исходно разрешенном праве записи (w): \( C(w)C_i, i \neq k \), должно быть разрешено право записи (w): \( C(w)C_j, j \neq k \), что предотвратит утечку права записи (w): \( C(w)C_i \).

Замечание. Утечка права записи (w): \( C(w)C_i \) проиллюстрирована на рис. 4.1.

![Иллюстрация утечка права записи (w)](image)

Рис. 4.1. Иллюстрация утечки права записи (w)

4. При расширенной канонической матрице контроля доступа разрешенным правом чтения (r): \( C(r)C_i, i \neq j \), при исходно разрешенном праве чтения (r): \( C(r)C_i, i \neq k \), должно быть разрешено право чтения (r): \( C(r)C_j, j \neq k \), что предотвратит утечку права чтения (r): \( C(r)C_i \).

Замечание. Утечка права чтения (r): \( C(r)C_i \) проиллюстрирована на рис. 4.2.

![Иллюстрация утечка права чтения (r)](image)

Рис. 4.2. Иллюстрация утечки права чтения (r)

Теперь о дополнительном требовании.

---

**Где-4. Принципы и методы контроля доступа к создаваемым объектам**

**Лемма 4.1.** Система безопасна относительно права удаления (d) при реализации следующего правила управления: любой включающий объект (папка) может быть удален любым субъектом, при условии отсутствия включенных в него объектов (в первую очередь файлов).

Доказательство. Метод дискретного контроля доступа к создаваемым файловым объектам не предполагает реализации разграничительной политики доступа к статическим объектам – к папкам, в которые включаются (записываются) создаваемые файлы. Как следствие, любой субъект, потенциально может удалить подобную папку, соответствующим образом не включенные в нее файлы. Что может привести к утечке права удаления (d). При запрете удаления папки, при наличии включенных в нее файлов, с учетом реализации очевидного права: \( C_i (d)C_j \) (право \( C_i (d)C_j \), \( i \neq j \) не имеет практического смысла), утечка права удаления (d) становится невозможной. Лемма доказана.

Замечание. Правило удаления, сформулированное в Лемме, справедливо и при реализации канонической матрицы доступа.

**Теорема 4.1.** Метод дискретного контроля доступа к создаваемым файловым объектам позволяет корректно реализовать контроль доступа в безопасной системе при условии реализации дискретного управления потоками информации (информационными потоками), сформулированных в требованиях 1-4 и в Лемме 4.1.

Доказательство. Теорема доказывается доказательством Теоремы 3.1 и Леммы 4.1.

**4.3.1.2. Требования к методу мандатного контроля доступа**

Леммами 3.5 и 3.6 доказывается в общем случае (и применительно к контролю доступа к создаваемым файловым объектам), что система, реализующая метод мандатного контроля доступа, безопасна относительно права записи (w) и права чтения (r) [33, 38].

Особенностью контроля доступа к создаваемым файлам является то, что размещаются (назначаются метки безопасности) именно файлы, причем это осуществляется автоматически, при создании, по средствам наследования файлом метки от созданного его субъекта доступа. Включающие элементы (папки) никаким образом не размещаются – им метка безопасности не приписывается.
Глава 4. Принципы и методы контроля доступа к создаваемым объектам

Основополагающее правило, реализуемое мандатным методом контроля доступа, используемое для защиты от хищения конфиденциальной информации, проиллюстрировано на рис. 4.3.

Рис. 4.3. Иллюстрация основополагающего правила мандатного метода контроля доступа

При иерархической структуре файловых объектов (а она всегда присутствует на практике, по крайней мере, есть включающий элемент логический диск) схема мандатного контроля доступа имеет вид, приведенный на рис. 4.4.

Рис. 4.4. Схема мандатного контроля доступа при иерархической структуре файловых объектов

Лемма 4.3. Мандатный контроль доступа, основанный на наследовании метки безопасности от субъекта доступа, не корректно реализует основополагающее правило мандатного контроля доступа.

Доказательство. Иллюстрация возможных правил контроля удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от субъекта доступа, создающего объект, проиллюстрирована на рис. 4.6.

Рис. 4.6. Иллюстрация возможных правил контроля удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от субъекта доступа, создающего объект

Лемма 4.2. Мандатный контроль доступа к создаваемым файловым объектам, основанный на наследовании метки безопасности от субъекта доступа, создающего объект, корректно реализует основополагающее правило мандатного контроля доступа.

Доказательство. С учетом того, что включающий объект (папка) ни каким образом не является сущностью, используемой при задании разграничительной политики (наличие/отсутствие включающего объекта, в том числе и объекта, не разделяемого системой и приложениями не сказывается на корректности разграничительной политики), правило, проиллюстрированное на рис. 4.4 сводится к правилу, проиллюстрированному на рис. 4.3, как следствие, основополагающее правило мандатного метода контроля доступа при иерархической структуре файловых объектов реализуется корректно. Лемма доказана.

Теперь, что касается удаления. Корректное правило удаление проиллюстрировано на рис. 4.5.

Рис. 4.5. Иллюстрация корректного правила удаления
Поскольку доступ к включающим объектам никак не ограничивается, любой субъект доступа, вне зависимости от его метки безопасности, потенциально может удалить (и переименовать) любой включающий объект, при этом удаляются все файлы, включенные в этот объект, вне зависимости от наследованных ими меток безопасности. Лемма доказана.

**Лемма 4.4.** Корректность реализации удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от субъекта доступа, обеспечивается при реализации следующего правила удаления — включающий объект может быть удален (и переименован) любым субъектом доступа, вне зависимости от его метки безопасности, только при условии отсутствия в нем включаемых объектов.

**Доказательство.** При реализации данного правила удаления, правила, приведенные на рис. 4.6, сводятся к правилам, приведенным на рис. 4.7.

а) Правило удаления включаемых объектов

![Diagram 1]

б) Правило удаления включающего объекта

![Diagram 2]

**Рис. 4.7.** Иллюстрация корректных правил удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от субъекта доступа, создающего объект

Как видим (см. рис. 4.7а), правило удаления включаемых объектов — это не что иное, как корректное правило удаления, см. рис. 4.4 (субъект может удалить объект только при выполнении условия $M_i = M_s$), корректность же реализации этого правила обеспечивается включением правила удаления включающего объекта, см. рис. 4.7б — возможность удаления любым пользователем при отсутствии в нем включаемых объектов. Лемма доказана.

**Теорема 4.2.** Метод мандатного контроля доступа к создаваемым файловым объектам позволяет корректно реализовать контроль доступа в защищенной системе при условии реализации диспетчером доступа правила управления потоками информации (информационными потоками), сформулированного в Лемме 4.4.

Доказательство. Теорема доказывается доказательством Теоремы 3.2, Лемм 4.2 и 4.4.

**4.3.2. Дополнительные требования, в части защиты от атак со стороны приложений, населяемых вредоносными функциями при прочтении «зараженных» файлов**

Априори контроль доступа к создаваемым файловым объектам предполагает реализацию разграничительной политики доступа к созданным в процессе функционирования системы файлам. Среди подобных файлов могут быть вредоносные. Защита от запуска на защищаемом компьютере ненанесенной программы вредоносных программ (вредоносных исполняемых файлов) обеспечивается реализацией механизма обеспечения замкнутости программной среды (рассмотрен ранее).

Однако существует весьма большая класс угроз, связанных с атаками со стороны приложений, приобретающих вредоносные функции в процессе их работы, т.е. атаки со стороны санкционированно установленных и используемых на компьютере приложений. Это приложения, населяемые соответствующими функциями в результате процитания вредоносного командного файла (не являющегося исполняемым), записанного на компьютер в процессе работы пользователя. К подобным приложениям, например, относятся офисные приложения, которые могут приобрести вредоносные функции...
в результате прочтения документа, наделенного макро-вирусом, всевозможные командные интерпретаторы, наделяемые дополнительным функционалом, в результате прочтения ими «активного» содержимого, в частности, скриптов и ActiveX-компонентов. Другими словами, достаточно широкий круг современных санкционированных приложений может быть наделен вредоносными функциями, в результате прочтения ими специально созданного и далее внедренного на защищаемый компьютер с этой целью вредоносного файла.

Как в случае решения задач защиты от вредоносного ПО и от сетевых атак, на практике и в данном случае широко применяют антивирусные средства защиты. О всех недостатках реализуемых ими технологий защиты мы уже говорили не раз, не будем повторяться.

А что же контроль доступа к файловым объектам, может он быть применим и для решения данной задачи защиты?

Естественно, что задачу противодействия данному классу угроз можно попытаться возложить на контроль доступа к создаваемым файлам (именно к создаваемым файлам в процессе работы пользователей им контролируется доступ). Поэтому решение рассматриваем в данной главе. Однако это не означает, что рассматриваемый здесь подход к защите не может быть применен при реализации контроля доступа к статическим файловым объектам.

Заметим, что контроль доступа не предполагает какого-либо анализа (в первую очередь, сигнального) файла, т.е. защита может строиться, по средством минимизации ущерба от атаки, в предположении о том, что вредоносный файл приложением с определенной вероятностью может быть прочтен.

4.3.2.1. Модель дисперсионного контроля доступа

Напомним модель дисперсионного контроля доступа к создаваемым файловым объектам. Если считать, что множество $C = \{C_1, ..., C_i\}$ — линейно упорядоченное множество субъектов доступа, а $R = \{R_{i,j}, ..., R_{n,m}\}$ — конечное множество прав доступа (в этой работе нас интересуют права доступа чтение ($r$), запись ($w$), соответственно, отсутствие права доступа (0)), субъект $C_i$ к объекту, созданному субъектом $C_j\ (i = 1, ..., l, j = 1, ..., l, l$), то матрица контроля доступа $M$, используемая для реализации разграничительной политики методом контроля доступа с принудительным управлением потоками информации (сущность «Владение» исключена, как таковая) имеет следующий вид (условными в строках матрицы указывают учетную информацию субъектов, запрашивающих доступ к объектам, а в столбцах — учетную информацию субъектов, унаследованную созданными ими объектами):

$$
M = \begin{bmatrix}
C_1 & C_2 & \cdots & C_l \\
C_{i_1} & r & w & \cdots & 0 \\
C_{i_2} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{i_{l-1}} & 0 & 0 & \cdots & r \\
C_{i_l} & 0 & w & \cdots & r \\
\end{bmatrix}
$$

В любой момент времени система описывается своим текущим состоянием $Q = (C, C, M)$, $M\{C, C\}$ — ячейка матрицы, содержит набор прав доступа. Будем обозначать $C_i(r)C_j$ разрешением права доступа субъекту $C_j$ к объекту, созданному субъектом $C_i\ (i = 1, ..., l, j = 1, ..., l, l$), где $R = \{w, r\}$: запись ($w$), чтение ($r$) (нас интересует запись вредоносного файла и последующее его прочтение).

Итак, после прочтения вредоносного файла, приложение наделяется функциями, запрограммированными в этом файле [9, 15]. В данном случае нас интересует то, что после прочтения вредоносного файла пользователем, запустившим приложение, приложение получает право записи и право чтения во все/из всех объектов, в которые разрешены соответствующие права доступа (запись, чтение) пользователем, запустившим приложение (предполагаем, что разграничительная политика реализуется для учетной записи).

Будем считать, что вероятность записи вредоносного файла сделать $C_i\ (i = 1, ..., l)$. Моделируем нашу матрицу доступа $M$, поместив в нее параметр $P(w)$, $i = 1, ..., l$ (представим в ячейках матрицы право записи ($w$) на параметр $P(w)$, право чтения пока не рассматриваем), получим матрицу вероятностей создания в объектах вредоносных файлов: $M(P(w))$:

$$
M(P(w)) = \begin{bmatrix}
C_1 & C_2 & \cdots & C_l \\
P_1(w) & P_2(w) & \cdots & 0 \\
P_1(w) & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
P_1(w) & 0 & 0 & \cdots \\
P_1(w) & 0 & P_1(w) & \cdots \\
\end{bmatrix}
$$

Не сложно показать, что с учетом возможностей разрешения записи в один объект одновременно несколькими субъектами, в пределе всеми субъекта $C = \{C_1, ..., C_l\}$, суммарная вероятность записи вредоносного файла в объект $C_j\ (i = 1, ..., l)$, определяется следующим образом:
Контроль доступа к компьютерным ресурсам

\[ P_c(w) = 1 - \prod_{i=1}^{l} (1 - P_i(w)), \]  
где:

\[ P_i(w) = P_i(w), \text{если } C_i(w \neq 0), \quad i \neq i; \]

\[ P_i(w) = P_i(w), \text{если } i = i; \]

\[ P_i(w) = 0, \text{если } C_i(w = 0). \]

Теперь по поводу осуществления атаки. После прочтения приложением записанного вредоносного файла, приложение приобретает вредоносный функционал, будем рассматривать атаки, направленные на несанкционированную запись (w) и чтение (r) информации, для простоты (и на основании существующей статистики угроз, см. рис. 2.7) их можно принять равновероятными. Будем обозначать вероятность атаки на запись в объект \( C_i \), в результате прочтения вредоносного файла, как \( P_i a(w) \), на чтение, соответственно, как \( P_i a(r) \). При этом данные атаки полностью определяются вероятностью занесения на компьютер вредоносного файла \( P_c(w) \), определяемой в соответствии с (4.1).

Подставив в матрицу доступа \( M \), в ячейки с разрешением записи w (вместо w) вероятности \( P_i a(w) \), a в ячейки с разрешением чтения r (вместо r) вероятности \( P_i a(r) \), получим матрицу вероятностей атак на файловые объекты в результате внедрения на компьютер и последующего прочтения вредоносного файла \( M(Pa(w,r)) \) [15]:

\[
M(Pa(w,r)) = \begin{bmatrix}
C_1 & C_2 & \cdots & C_l \\
C_1 & P_1 a(r), P_1 a(w) & P_2 a(w) & \cdots & 0 \\
C_2 & \vdots & P_2 a(r), P_3 a(w) & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{l-1} & 0 & 0 & \cdots & P_i a(r) \\
C_l & 0 & P_a(w) & \cdots & P_i a(r), P_i a(w)
\end{bmatrix}
\]

Теперь сформулируем требования к безопасной работе с вредоносными файлами, реализации которых может принципиально понизить ущерб от рассматриваемой группы атак [15].

Лемма 4.5. Последствия от атак минимальны при реализации полностью изолированной обработки субъектами созданных ими файловых объектов. При этом реализуется каноническая матрица доступа \( M_k \):

Доказательство. Для канонической матрицы доступа \( M_k \) минимальны вероятности занесения в систему вредоносного файла, определяемые следующим образом \( P_i c(w) = P_i(w), i = 1, \ldots, l \). Лемма доказана.

Следствие. Реализация разграничительной политики, основанной на полной изолированности обработки субъектами (пользователями) информации, что описывается канонической матрицей доступа, обеспечивает максимальный уровень безопасности системы, в части защиты от атак со стороны приложений, наделяемых вредоносными свойствами.

Ввиду того, что уровень безопасности может быть повышен в результате добавления права w или r в канонической матрице доступа, далее сформулируем требование к безопасному расширению канонической матрицы доступа правами r и/или w.

Важным для последующих наших рассуждений является то, что в общем случае на практике значения вероятностей записи вредоносного файла различными субъектами \( C_i, P_i(w), i = 1, \ldots, l \), сильно различаются, иногда на порядки.

Лемма 4.6. Безопасным является добавление в каноническую матрицу \( M_k \) разрешения права записи \( C_i(w) \), \( i \neq j \), при условии \( P_i(w) < P_j(w) \).

Доказательство. При разрешение права записи \( C_i(w) \) \( i \neq j \), при условии \( P_i(w) < P_j(w) \), имеем:

\[
P_c(w) - P_j(w) \quad P_i(w) - P_i(w),
\]

как следствие, при дополнении права записи \( C_i(w) \) \( P_i(r), P_a(w) \), соответственно, \( P_i(a(r), P_a(w) \), по сравнению с канонической матрицей, не изменяются. Лемма доказана.
Замечание. Максимально опасным является добавление в каноническую матрицу $M_c$ разрешения права чтения $C_iC_j$, $i \neq j$, при условии $P_i(w) >> P_j(w)$. Это объясняется тем, что в данном случае значение $P_i(c(w))$ увеличивается до значения $P_i(w)$, максимальным образом возрастая $P_i(a(r), P_i(a(w))$.

Лемма 4.7. Безопасным является добавление в каноническую матрицу $M_c$ разрешения права чтения $C_iC_j$, $i \neq j$, при условии $P_i(r) >> P_j(r)$.

Доказательство. При разрешении права чтения $C_iC_j$, $i \neq j$, при условии $P_i(r) >> P_j(r)$, имеем:

$$P_i(c(w)) = P_i(w)$$
$$P_i(c(w)) = P_i(w),$$

как следствие, при дополнении права чтения $C_iC_j$, $P_i(a(r), P_i(a(w)))$, соответственно, $P_i(a(r), P_i(a(w)))$, по сравнению с канонической матрицей, не изменяются. Лемма доказана.

Замечание. Максимально опасным является добавление в каноническую матрицу $M_c$ разрешения права чтения $C_iC_j$, $i \neq j$, при условии $P_i(w) << P_j(w)$. Это объясняется тем, что в данном случае значение $P_i(c(w))$ увеличивается до значения $P_i(w)$, максимальным образом возрастая $P_i(a(r), P_i(a(w))$.

Следствие. Безопасной, при добавлении в каноническую матрицу $M_c$ разрешения права записи, при условии $P_i(w) << P_j(w) << ... P_i(w)$, в общем случае является матрица $M_i(w)$ следующего вида:

$$M_i(w) = \begin{bmatrix} C_1 & C_2 & \cdots & C_i \\ C_i & r, w & 0 & \cdots & 0 \\ C_2 & 0 & r, w & \cdots & w \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{i-1} & 0 & 0 & \cdots & r, w \\ C_i & 0 & 0 & \cdots & r, w \end{bmatrix}$$

Следствие. Безопасной, при добавлении в каноническую матрицу $M_c$ разрешения прав записи и чтения, при условии $P_i(w) << P_j(w) << ... P_i(w)$, в общем случае является матрица $M_i(w, r)$ следующего вида:

$$M_i(w, r) = \begin{bmatrix} C_1 & C_2 & \cdots & C_i \\ C_2 & r, w & 0 & \cdots & 0 \\ C_3 & 0 & r, w & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{i-1} & 0 & 0 & \cdots & r, w \\ C_i & 0 & 0 & \cdots & r, w \end{bmatrix}$$

Теорема 4.3. Безопасной, в части защиты от атак со стороны приложений, надежных вредоносными функциями, в результате прочтения приложением вредоносного файла, является полностью изолированная обработка субъектами созданных ими файловых объектов, при этом реализуется каноническая матрица доступа, которая может быть расширена добавлением разрешения права записи $C_iC_j$, $i \neq j$, при условии $P_i(w) << P_j(w)$, и добавлением разрешения права чтения $C_iC_i$, $i \neq j$, при условии $P_i(r) >> P_j(r)$.

Доказательство. Теорема доказывается доказательством Лемм 4.5 - 4.7. Теорема доказана.

4.3.2.2. Модель мандатного контроля доступа

Напомним, что метки безопасности (мандаты) назначаются субъектам (группам субъектов), и служат для формализованного представления соот-
Контроль доступа к компьютерным ресурсам

вственно их уровней полномочий и конфиденциальности. Заметим, что мандатный контроль доступа к создаваемым файловым объектам отличается (от соответствующего метода контроля доступа к статичным объектам) тем, что объекты доступа не размечиваются (им не испытываются на наличие конфиденциальности) задание разграничительной политики доступа состоит исключительно в назначении меток безопасности субъектам \( M_i \) (что кардинально упрощает задачу администрирования). При создании субъектом нового файла, файлу наследуется учетная информация субъекта доступа – его метка безопасности \( M_c \) (обозначим унаследованную метку \( M_{co} \), при этом \( M_{co} = M_c \).

При запросе же доступа к любому файлу, диспетчер доступа анализирует наличие, а при наличии, собственно значение метки безопасности \( M_{co} \) в данном файле. При наличии метки в файле – \( M_{co} \), диспетчер сравнивает эту метку с меткой субъекта, запрашивающего доступ к файлу, 

\[ M_c \] – анализирует выполнение заданного правила контроля доступа. В результате анализа данной информации, с учетом реализуемого правила доступа, диспетчер либо отклоняет запрос субъектом доступа к файлу, либо отказывает в нем.

Будем считать, что чем выше полномочия (имеет право обрабатывать информацию формально более высокого уровня конфиденциальности) субъекта из множества \( C = \{ C_1, ..., C_n \} \), тем меньше значение метки безопасности \( M_i, i = 1, ..., n \) ему присваивается, т.е. \( M_1 < M_2 < M_3 < ... < M_n \).

Наиболее широко используемые правила, направленные на применение практики секретного делопроизводства и компьютерной обработки информации, обеспечивающие защиту от понижения категории обрабатываемой информации, с целью защиты от нарушения ее конфиденциальности, имеют следующий вид:

1. Субъект \( C \) имеет доступ к объекту \( O \) в режиме «Чтения» в случае,

   если выполняется условие: \( M_c < M_{co} \).

2. Субъект \( C \) имеет доступ к объекту \( O \) в режиме «Записи» в случае,

   если выполняется условие: \( M_c = M_{co} \).

Замечание. В общем случае не противоречивой, в части защиты от понижения категории объекта, является следующая формулировка второго правила: Субъект \( C \) имеет доступ к объекту \( O \) в режиме «Записи» в случае, если выполняется условие: \( M_c \geq M_{co} \).

Как мы неоднократно отмечали, любой метод контроля доступа с принудительным управлением информационными потоками может быть представим в виде матрицы доступа. Рассмотрим матрицу доступа для мандатного метода в общем случае. \( M_m \):

\[
\begin{bmatrix}
C_1(M_1) & C_1(M_2) & \cdots & C_1(M_n) \\
C_2(M_1) & C_2(M_2) & \cdots & C_2(M_n) \\
\vdots & \vdots & \ddots & \vdots \\
C_n(M_1) & C_n(M_2) & \cdots & C_n(M_n)
\end{bmatrix}
\]

\( M_m = \begin{bmatrix}
 r & w & r & \cdots & r \\
 w & r & w & \cdots & r \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 w & w & \cdots & r & w
\end{bmatrix}
\]

Как мы говорили ранее, применительно к методам контроля доступа субъектов к статичным объектам, чем выше уровень конфиденциальности обрабатываемого документа (файла), тем более «жесткие» устанавливаются условия его обработки, тем меньше значение вероятности записи вредоносного файла \( P_i(w) \) в данный объект.

Как следствие, между отношениями для меток безопасности \( M_i, i = 1, ..., k \), \( M_1 < M_2 < M_3 < ... < M_k \), существуют прямая зависимость. Как следствие, матрицу \( M_k \) мы можем представить в следующем виде, \( M_k(p) \):

\[
\begin{bmatrix}
C_1(P_{1}(w)) & C_1(P_{2}(w)) & \cdots & C_1(P_{m}(w)) \\
C_2(P_{1}(w)) & C_2(P_{2}(w)) & \cdots & C_2(P_{m}(w)) \\
\vdots & \vdots & \ddots & \vdots \\
C_m(P_{1}(w)) & C_m(P_{2}(w)) & \cdots & C_m(P_{m}(w))
\end{bmatrix}
\]

\[
M_k(p) = \begin{bmatrix}
 r & w & r & \cdots & r \\
 w & r & w & \cdots & r \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 w & w & \cdots & r & w
\end{bmatrix}
\]

Сравнивая матрицу \( M_k(p) \) с матрицей \( M_k(w, r) \), представленной ранее. При этом напомним, что матрица \( M_k(w, r) \) представляет собой безопасную схему контроля доступа, в части защиты от атак со стороны приложений, наделяемых вредоносными функциями, в результате проникновения приложения вредоносного файла, при условии: \( P_k(w) << P_k(w) << ... < P_k(w) \). Как видим, матрицы \( M_k(p) \) и \( M_k(w, r) \) полностью исключают друг друга: Общего у них остается только одно – главная диагональ.
Глава 4. Принципы и методы контроля доступа к создаваемым объектам

Следствие. Наиболее критичным во всех отношениях является обработка на одном компьютере и открытой, и конфиденциальной информации. Следуя всему сказанному ранее, обработка открытой информации должна полностью изолироваться в обязательном порядке. В результате, считая, что субъектом \( C_i \) обрабатывается открытая информация, получаем обязательное непротиворечивое решение, описываемое следующей матрицей доступа, \( M_{oc} \):

\[
M_{oc}(o) = \begin{bmatrix}
C_1(M_1) & C_2(M_2) & \cdots & C_i(M_i) \\
C_1(M_1) & C_2(M_2) & \cdots & C_i(M_i) \\
\vdots & \vdots & \ddots & \vdots \\
C_{i-1}(M_{i-1}) & \vdots & \ddots & \vdots \\
C_i(M_i) & C_1(M_1) & \cdots & C_{i-1}(M_{i-1}) \\
\end{bmatrix}
\]

Правила, реализуемые данной матрицей доступа, на взгляд авторов, должны в обязательном порядке реализовываться в системах, предназначенных для одновременной обработки открытой и конфиденциальной информации.

4.4. Примеры решения некоторых актуальных задач защиты

Чтобы далее иметь возможность сравнить между собою методы контроля доступа к статическим и к создаваемым файловым объектам, определиться с областями их эффективного использования, рассмотрим, применительно к методу контроля доступа к создаваемым файловым объектам, возможность решения задач защиты от вредоносных программ и от сетевых атак на уязвимости ОС и приложений (т.е. тех же задач, которые мы ранее рассматривали для метода контроля доступа к статическим файловым объектам). По понятным причинам, будем рассматривать именно дискретный (а не манипулятивный, предназначенный для решения иных задач) метод контроля доступа к создаваемым файловым объектам.

Идея предлагаемого подхода к защите от вредоносных программ состоит в том, чтобы запретить исполнять создаваемые в процессе эксплуатации системы, файлы, т.к. это могут быть выполняемые файлы вредоносных программ, либо исполняемые файлы несанкционированно установленных пользователями программ [26]. Сначала, рассмотрим защиту от запуска вредоносных программ приложением (заметим, что рассматриваемые нами методы не противоводействуют несанкционированному внедрению/установ-
Контроль доступа к компьютерным ресурсам

не несанкционированной программы на защищаемом компьютере, о решении этой задачи речь пойдет далее).

Пусть доступ в сеть на защищаемом компьютере реализуется Internet-браузером Internet Explorer (параллельно будем рассматривать и защиту от сетевых атак. Пример разграничительной политики доступа к создаваемым контролируемым приложением (браузером) файловым объектам, в части защиты от вредоносного ПО, представлен в табл. 4.7.

Таблица 4.7. Пример разграничительной политики доступа к создаваемым контролируемым приложением файловым объектам

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, создающего объекты, доступ к которым разграничивается</th>
<th>Учетная информация субъекта, имеющего право доступа к контролируемым созданным объектам, и права доступа субъекта к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>; C:\Program Files\Internet Explorer\iexplore.exe,</em></td>
<td>*; *; (Зн/Чт)</td>
</tr>
</tbody>
</table>

Рассмотрим этот пример разграничительной политики доступа, включающей в себя лишь одно правило! Рассмотрим, что контролируемыми являются все файловые объекты, создаваемые приложением Internet Explorer. Поскольку исходный и эффективный идентификаторы в субъекте задачи маской "*" (+" Все"), не важно, под какой учетной записью, в том числе, при какой смене учетной записи при доступе к объекту, браузером создается объект — он будет контролироваться. Правилом определяется то, что любой файловый объект, созданный браузером Internet Explorer разрешено любому пользователю (маска "*"), причем любым процессом (имя процесса в субъекте определено маской "*"), в том числе, и браузером, только читать и заисывать. Исполнять нельзя! Т.е. одной строкой в разграничительной политике доступа всем запрещается исполнение всех файлов, созданных браузером.

Если говорить о защите от сетевых атак, то напомним основные цели сетевых атак — среди них доминируют установка вредоносного ПО (рассмотрели), а так же хищение и несанкционированная модификация (нарушение целостности) обрабатываемой на компьютере конфиденциальной информации.

Поскольку противодействие всем этим ключевым атакам напрямую связана с защитой создаваемых в процессе работы системы файлов (отличие составляет лишь назначение создаваемого файла — в нем может сохраняться обрабатываемая на компьютере информация, которая может быть похищена, либо несанкционированно модифицирована, либо удалена, или в

В табл. 4.8, мы прослеживаем разграничительную политику доступа, приведенную в табл. 4.7, двумя правилами. Правило 2 задает то, что программе Internet Explorer запрещен доступ ко всем файлам, создаваемым пользователем User (в том числе, и при открытии его с любой иной учетной записью). Но, поскольку под действие правила 2 подпадает и субъект доступа "User", C:\Program Files\Internet Explorer\iexplore.exe,* (+), для этого субъекта включено самостоятельное правило — правило 3, запрещающее пользователю User исполнять создаваемые им с использованием программы Internet Explorer файлы (в данном случае правило, анализируемое диспетчером доступа, выбирается по наиболее точному описанию идентификатора субъекта).

На этом примере разграничительной политики доступа (а это лишь пример), видим, как просто с использованием дискретного метода контроля доступа к создаваемым файлам решать на защищаемом компьютере в комплексе сложнейшие задачи защиты от наиболее актуальных сетевых атак.

Теперь рассмотрим защиту от запуска вредоносных программ пользователя. Это более общий случай разграничительной политики доступа к созда
Таблица 4.12. Пример разграничительной политики доступа к создаваемым файловым объектам в общем случае

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, создающего объекты, доступ к которому разграничивается</th>
<th>Учетная информация субъекта, имеющего право доступа к контролируемым созданным объектам, и права доступа субъекту к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.<em>.</em></td>
<td>.<em>.</em> (зп/чт)</td>
</tr>
</tbody>
</table>

Как видим из табл. 4.12, любой создаваемый файловый объект размещается дискретчиком доступа (учетную информацию субъекта, созданного файла, ему достаточно помеcат в виде ".*.*.*"), чем по сути, вносится различие между статическими и созданными в процессе функционирования файловыми объектами.

Данная разграничительная политика доступа очень проста, но на практике может привести к конфликтам системы. Это обусловливается тем, что созданные разграничения распространяются и на пользователя System (сystema), который не сможет в этом случае заполнить привилегированную ему исполняемые файлы (при условии, что контролируется и доступ к файлам, созданным с системными правами). Для обеспечения корректности функционирования системы следует включить дополнительный контроль доступа к создаваемым файлам для пользователя System, см. табл. 4.13. В таблицу дополнительно внесены два правила. Правило 1 задает то, что все, что создано системой, пользователям можно только читать и ипиривать. Правило 2 разрешает система неограниченный доступ к созданным ею же системным файлам.

Таблица 4.13. Пример разграничительной политики доступа к создаваемым файловым объектам в общем случае с контролем создаваемых файлов для пользователя System

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, создающего объекты, доступ к которому разграничивается</th>
<th>Учетная информация субъекта, имеющего право доступа к контролируемым созданным объектам, и права доступа субъекту к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.<em>.</em></td>
<td>.<em>.</em> (зп/чт)</td>
</tr>
<tr>
<td>2</td>
<td>System.*, System</td>
<td>.<em>.</em> (чт/чт)</td>
</tr>
<tr>
<td>3</td>
<td>System.*, System</td>
<td>System.*, System (зп/чт/чт)</td>
</tr>
<tr>
<td>4</td>
<td>System.*, System</td>
<td>.<em>.</em> (чт/чт)</td>
</tr>
<tr>
<td>5</td>
<td>System.*, System</td>
<td>System.*, System (зп/чт/чт)</td>
</tr>
</tbody>
</table>

Замечание. По аналогии с тем, как мы рассматривали ранее для механизма обеспечения замкнутости программной среды, эти настройки так же можно усилить, исключив третье правило, введя вместо него правило, позволяющее получить полный доступ к системным объектам только необходимым системным процессам с правами пользователя System.

С точки зрения защиты от модификации системных ресурсов может быть реализована разграничительная политика доступа, представленная в табл. 4.14.

Таблица 4.14. Пример разграничительной политики доступа, реализуемой для защиты системных ресурсов от несанкционированной модификации и удаления

<table>
<thead>
<tr>
<th>№ правила в разграничительной политике</th>
<th>Учетная информация субъекта, создающего объекты, доступ к которому разграничивается</th>
<th>Учетная информация субъекта, имеющего право доступа к контролируемым созданным объектам, и права доступа субъекту к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.<em>.</em></td>
<td>.<em>.</em> (зп/чт)</td>
</tr>
<tr>
<td>2</td>
<td>System.*, System</td>
<td>.<em>.</em> (чт/чт)</td>
</tr>
<tr>
<td>3</td>
<td>System.*, System</td>
<td>System.*, System (зп/чт/чт)</td>
</tr>
<tr>
<td>4</td>
<td>System.*, System</td>
<td>.<em>.</em> (чт/чт)</td>
</tr>
<tr>
<td>5</td>
<td>System.*, System</td>
<td>System.*, System (зп/чт/чт)</td>
</tr>
</tbody>
</table>

Видим, что интерактивные пользователи не имеют возможности модификации системных объектов. Альтернативное решение задачи защиты состоит во включении в разграничительную политику правила наследования учетной информации создаваемыми файлами от системных объектов. Например, можно запретить диспетчеру размещать файловые объекты, создаваемые субъектом «System.*, System». Особенностью реализации разграничительной политики доступа в этом случае обусловливается тем, что системные файлы в любом случае (статичные они, или создаваемые) оставляются инерзенными. В разграничительной политике, представленной в табл. 4.14, в этом случае потребуется отнять только правила 1, 4, 5.

Как видим из проведенного небольшого исследования, возможность контроля доступа к создаваемым файловым объектам, в части защиты от актуальных угроз, весьма интересна. А главное, что отличает метод дискретного контроля доступа к создаваемым файловым объектам, применительно к рассматриваемым задачам защиты, это простота администрирования! Однако важно отметить, что во всех файлах, создаваемых одним и тем же субъектом, права доступа одинаковы, что, как видим, полностью идентифицируется в концепцию решения задачи защиты от критически опасных субъектов, не от критически важных объектов, о чем говорили ранее. Если говорить о контроле доступа к системным файловым объектам, то в отличие от
метода контроля доступа к статическим файловым объектам, рассматриваемый метод не представляет возможности задания разграничительной политики к отдельным системным файлам, что существенно ограничивает его возможности, в части реализации разграничительной политики доступа для системных субъектов к системным объектам, что необходимо, например, при защите от атак на системные процессы.

4.5. Место (область эффективного использования) методов контроля доступа к создаваемым объектам

После рассмотрения возможностей методов контроля доступа к создаваемым объектам (на примере файловых объектов), уместно провести их сравнение с методами контроля доступа к статическим объектам, с целью определения их областей эффективного использования, целесообразности применения в комплексе, решаемых при этом задач защиты информации от несанкционированного доступа различными методами.

Области эффективного использования данных методов определяются собственностью их назначением, непосредственно отраженным в названиях. Методы контроля доступа к создаваемым файловым объектам, предназначенные для контроля доступа к файлам, создаваемым для хранения обрабатываемой информации, безусловно, эффективнее для решения этих задач. Высокая эффективность применения данных методов защиты здесь определяется простотой администрирования средств защиты и возможностью реализации корректной разграничительной политики доступа в общем случае, в том числе, и при наличии в системе неразделяемых файловых объектов. Поскольку именно защита обрабатываемой информации является ключевой задачей защиты, решаемой реализацией разграничительной политики доступа, то именно эти методы (как дискретционный, так и мандатный) могут позиционироваться в качестве основных методов контроля доступа, направленных на реализацию защиты обрабатываемых в информационной системе данных.

В свою очередь, данные методы обладают двумя принципиальными недостатками, ограничивающими их самостоятельное применение на практике. Первый, наиболее важный, состоит в том, что методами контроля доступа к создаваемым объектам не может разграничиваться доступ по созданию новых объектов (не контролируется создание нового файла — контролируется последующий доступ к созданному файлу). В частности, мы не можем разрешить, запретить создание файлов на определенных дисках, в определенных папках, в том числе, в разделенных в сети объектах, невозможно запретить/разрешить создание файлов на внешних файловых накопителях, в системных каталогах. Это ужесточение статичные объекты, и для решения задач контроля доступа к ним должны использоваться методы контроля доступа к статическим файловым объектам.

Второй недостаток связан с тем, что одинаковое право доступа будет автоматически установлено ко всем объектам, созданным одним и тем же субъектом. Если говорить о системных статичных файлах, то даже после их переименований, к этим объектам невозможно будет разграничить доступ. С этой целью (для создания тонкой разграничительной политики доступа к системным файлам, т.е. опять же к статическим файловым объектам), должен использоваться метод контроля доступа к статическим объектам.

Таким образом, в общем случае это не альтернативные, а дополняющие друг друга (эффективные при решении различных задач защиты) методы контроля доступа. В частности же, отдельные задачи могут быть решены и теми, и другими группами методов.

Механизмы защиты, реализующие контроль доступа к статическим и к создаваемым объектам должны работать в системе одновременно, обеспечивая последовательную обработку запроса доступа к объекту, при этом, по понятным причинам, первым запрошен определенно обрабатывать механизмы защиты, реализующим контроль доступа к статическим объектам.

Отметим, что с точки зрения защиты обрабатываемых данных (создаваемых объектов) при совместном использовании с методами контроля доступа к создаваемым объектам, методы контроля доступа к статическим объектам могут позиционироваться в качестве методов, разграничивающих права доступа к созданным объектам — разграничивают право того, в каких объектах (в том числе, на каких внешних накопителях) могут сохранять данные субъекты доступа [14].

Вывод. Из сказанного следует, что контроль доступа к создаваемым объектам имеет смысл использовать совместно с контролем доступа к создаваемым объектам, в части защиты обрабатываемых данных, последний же в этом случае может позиционироваться, как контроль доступа к системным объектам и контроль доступа к создаваемым объектам.
А вот мандатный контроль доступа целесообразно реализовывать именно методом контроля доступа к создаваемым файловым объектам. С его использованием, в первую очередь, должна решаться задача защиты от понимания уровня конфиденциальности информации. Но этот метод не будет контролировать (разграничивать) возможность записи на диски и в папки, в частности, на внешние файловые накопители. С этой целью в дополнение следует использовать, либо контроль доступа к устройствам (где размещаются накопители — им присваивается метка, метод контроля был нам рассмотрен ранее), если разграничивать запись следует только на внешние накопители, либо контроль доступа к статичным файловым объектам, где в качестве статичных объектов уже выступают статичные диски и папки.

### 4.6. Технические решения

#### 4.6.1. Мандатный контроль доступа к создаваемым файлам

Для оценки простоты администрирования, при реализации метода мандатного контроля доступа к создаваемым файлам, полностью приведем процедуру настройки [23,39,43].

Замечание. Рассматриваемые далее технические решения по реализации методов контроля доступа к создаваемым объектам авторами запатентованы [15-17].

Сначала требуется создать пользователей в среде защиты/в системе. Пользователи либо создаются из интерфейса средства защиты, приведен-ного на рис. 4.8, либо импортируются в средство защиты из системы, после чего для них устанавливается пароль на вход в систему, включая задание возможности входа в систему в безопасном режиме, см. рис. 4.8, что крайне важно, с точки зрения построения безопасной системы (возможность входа в систему в безопасном режиме любым пользователем можно рассматривать в качестве безусловной технологической уязвимости). Естественно, для системных пользователей пароль установить нельзя.

![Рис. 4.8. Интерфейс создания нового пользователя в среде защиты](image)

![Рис. 4.9. Интерфейс отображения созданных в системе пользователей](image)

В результате, созданные в системе и в среде защиты пользователи отображаются в окне интерфейса средства защиты, в виде, приведенном на рис. 4.9. Системные пользователи отображаются черным цветом, пользователи с установленными паролями для входа в систему — зеленым, пользователи, пароль которого не установлен — их вход в систему невозможен, красным.

После этого из меню, приведенного на рис. 4.10, задаются мандатные уровни. Уровни задаются как количественным значением метки безопасности, так и ее смысловой транскрипцией. В результате, заведенные уровни доступа отображаются в виде, представленном на рис. 4.11.
Глава 4. Принципы и методы контроля доступа к создаваемым объектам

Рассмотрим, что будет происходить при назначении пользователя уровня доступа (меток безопасности). При сохранении файла контролируемым пользователем (пользователем с присвоенным уровнем доступа — меткой безопасности), файлом будет наследоваться уровень доступа создавшего его пользователя. Наследование будет происходить и в том случае, если контролируемый пользователь модифицирует неразмещенный ранее файл.

При последующем обращении к контролируемому файлу (к файлу, созданному контролируемым пользователем), доступ к нему будет разрешаться/запрещаться, в соответствии с заданными правилами мандатного доступа. При этом доступ неразмещенного пользователя к размещенному файлу будет запрещен, а размещенному пользователя к неразмещеному файлу, разрешен.

Настройка правил мандатного доступа осуществляется из интерфейса, приведенного на рис. 4.13. Правилами, заданными на рис. 4.14, реализуется полная изоляционность обработки пользователями информации — задается «непротиворечивое» правило мандатного доступа.

Вот и все настройки. Больше ничего не требуется — не требуется каким-либо образом размещать файловые объекты. Вся разметка создаваемых в процессе работы файлов производится автоматически.

Все отмеченные преимущества метода мандатного (сессионного) контроля доступа к создаваемым файлам позволяют сделать вывод о целесообразности именно этого метода защиты информации в качестве основы реализации сессионной модели контроля доступа.

Далее для тех пользователей, доступ к создаваемым файлам которых должен контролироваться, сопоставляются метки безопасности — им присваивается уровень доступа, уровень доступа назначается из созданного до этого списка, см. рис. 4.11.

Пользователи, с заданными для них уровнями доступа отображаются в виде, приведенном на рис. 4.12.
Глава 4. Принципы и методы контроля доступа к создаваемым объектам

Заметим, что, поскольку могут использоваться, как иерархические, так и не- иерархические метки безопасности, методом мандатного контроля доступа (с использованием меток безопасности) может реализовываться как сессионная, так и ролевая модель контроля доступа. Пример задания уровня доступа для ролевой модели контроля доступа проиллюстрирован на рис. 4.15. При этом реализуются правила мандатного доступа, приведенные на рис. 4.14.

Рис. 4.13. Интерфейс настройки правила мандатного доступа

Как видим, подобная реализация мандатного контроля доступа принципиально упрощает администрирование системы защиты, позволяя при этом реализовать корректную разграничительную политику доступа в общем случае – не требуется назначать метки безопасности папкам, и тому же, не разделяя системой и приложениями. Это позволяет рекомендовать данный метод контроля доступа в качестве основы реализации сессионной модели контроля доступа (при использовании «непротиворечивого правила мандатного контроля доступа», см. рис. 4.14), которую, как говорили, сегодня можно рассматривать в качестве эффективной альтернативы, так называемым, DLP-решениям.

Настройка правил мандатного доступа

<table>
<thead>
<tr>
<th>Название доступа пользователя</th>
<th>Название доступа файла</th>
</tr>
</thead>
<tbody>
<tr>
<td>Запретить чтение</td>
<td>Запретить чтение</td>
</tr>
<tr>
<td>Запретить запись</td>
<td>Запретить запись</td>
</tr>
<tr>
<td>Разрешать чтение</td>
<td>Разрешать чтение</td>
</tr>
<tr>
<td>Разрешать запись</td>
<td>Разрешать запись</td>
</tr>
<tr>
<td>Запретить чтение</td>
<td>Запретить чтение</td>
</tr>
<tr>
<td>Запретить запись</td>
<td>Запретить запись</td>
</tr>
</tbody>
</table>

Рис. 4.14. Задание «непротиворечивого» правила мандатного доступа

Рис. 4.15. Пример задания уровней доступа для ролевой модели контроля доступа

Для возможности обзора произведенной системой в процессе работы разметки файлов, создаваемых контролируемыми субъектами, в состав среды защиты включена специальная утилита обзора мандатных меток созданных файлов. С ее использованием, выбрав соответствующий файловый объект, см. рис. 4.16, администратор может посмотреть, какие права устанавливаются файлом, при его создании (отображаются имя пользователя, создавшего файл, и его уровень доступа), см. рис. 4.17.

Рис. 4.16. Обзор файловых объектов утилитой, отображающей мандатные метки созданных файлов

Рис. 4.17. Отображение разметки созданных контролируемыми (размещеными) пользователями файлов
Важной возможностью утилиты, проиллюстрированной на рис. 4.16, является возможность удалить произведенную разметку созданных файлов. Это можно сделать для конкретного файла, выбрав его в проводнике, см. рис. 4.18, либо из всех файлов выбранной папки (диска), см. рис. 4.19, задав при этом, созданная каким пользователем (пользователями) разметка должна удалиться.

![Рис. 4.18. Удаление разметки утилитой из созданного файла](image)

Рис. 4.18. Удаление разметки утилитой из созданного файла

Не менее важной возможностью рассматриваемой утилиты, является возможность осуществления администратором вручную разметки уже созданных файлов – файлов, созданных до внедрения системы защиты. Для этого администратору потребуется выбрать размещаемый файловый объект, и задать параметры разметки (учетную информацию создателя файла), это может быть как файл, см. рис. 4.20, так и папка – будут соответствующим образом размещаться все файлы в выбранной папке, см. рис. 4.21.

![Рис. 4.19. Удаление разметки созданных файлов утилитой из выбранной папки](image)

Рис. 4.19. Удаление разметки созданных файлов утилитой из выбранной папки

Ручная разметка администратором файлов позволяет использовать метод контроля доступа как к создаваемым, так и к статичным объектам, в последнем случае администратор должен соответствующим образом размечать статичные файловые объекты. Это позволяет говорить о самостоятельном методе контроля доступа с ручной и с автоматической разметкой файлов [47]. Данные техническое решение авторами запатентовано [16].

С учетом того, что, как отмечали, методы контроля доступа к статичным и к создаваемым объектам в общем случае должны использоваться в комплексе, при этом задачи по защите системных объектов могут возлагаться на метод контроля доступа к статичным объектам, может потребоваться задание статичных объектов, в отношении которых не должен использоваться метод контроля доступа к создаваемым объектам (т.e. должна существовать возможность исключения статичных объектов из схемы контроля доступа к создаваемым объектам).

1 Хотя разметить файл можно и однократно его пересохранить пользователем.
4.6.2. Дискреционный контроль доступа к создаваемым объектам

Говоря ранее о дискреционном контроле доступа, мы говорили о том, что субъект доступа при реализации дискреционного контроля в эффективном средстве защиты информации в общем случае должен определяться тремя сущностями: первичный идентификатор пользователя, полнопопутое имя процесса, эффективный идентификатор пользователя, что, в том числе, позволяет реализовать процессную модель контроля доступа. При этом в качестве субъекта доступа должны рассматриваться, как интерактивные, так и системные пользователи, как прикладные, так и системные процессы.

Как отмечали, особенностю реализации контроля доступа к создаваемым объектам, является следующее:

- Правила доступа устанавливаются между сущностями: субъект доступа, запрашивающий доступ к объекту и субъект доступа, создавший этот объект.
- При создании субъектом нового объекта, объектом наследуется учетная информация субъекта доступа, созданного этом файл;
- При запросях доступа к контролируемому объекту, диспетчер доступа анализирует содержимое унаследованной объекта учетной информации созданного его субъекта доступа и заданные правила доступа в результате чего предоставляет запрошенный субъектов доступ, либо отказывает в нем.

4.6.2.1. Контроль доступа к создаваемым файлам

Таким образом, при реализации дискреционного контроля доступа к создаваемым файловым объектам, разметка создаваемого файла состоит в запоминании для созданного файла учетной информации созданного его файлового объекта. При этом субъект описывают три сущности: первичный идентификатор пользователя, полнотопутое имя процесса, эффективный идентификатор пользователя.

Рассмотрим реализацию [36] запатентованного авторами технического решения [15].

Субъекты доступа создаются из интерфейса, представленного на рис. 4.23.

![](image.png)

Рис. 4.23. Интерфейс создания субъектов доступа.

Субъекты доступа создаются по аналогии с системе [23] из того же интерфейса с реализацией контроля доступа к статическим объектам, опять же могут использоваться маски. С использованием маски «Любой» (во всех
полях субъект доступа задается маской "*" реализуется разрешительная разграниченная политика доступа.

А вот объекты доступа в данном случае уже создавать не требуется – объект исключен из разграниченной политики доступа.

Исходя из сформулированных ранее требований к назначению правил доступа, по умолчанию для контролируемых файлов, реализовано следующее правило доступа:

Субъект, создавший файл, имеет полный доступ (за исключением исполнения), к созданному им файлу (чем реализуется каноническая матрица доступа), т.е. по умолчанию реализована каноническая матрица доступа.

Поэтому, если выбрать в правом и в левом поле интерфейса одного и того же субъекта доступа, для него можно будет настроить только право исполнения и режимы аудита доступа.

Разграничительной же политикой задается, каким еще субъектам, кроме субъекта, создавшего файл, и какие права доступа разрешены к этому файлу, т.е. назначение правил доступа заключается в расширении исходно заданной канонической матрицы доступа.

Субъекты доступа создаются из интерфейса, приведенного на рис. 4.23 (при создании субъектов доступа присваиваются имена, которые используются при создании правил доступа), и отображаются в интерфейсе в виде, проиллюстрированном на рис. 4.24. Правила доступа создаются из интерфейса, приведенного на рис. 4.25, и отображаются в интерфейсе, приведенном на рис. 4.26.

![Интерфейс назначения правил доступа](image1)

![Интерфейс отображения назначенных правил доступа](image2)

Правила доступа к создаваемым файлам задаются из интерфейса, представленного на рис. 4.25, в котором созданные субъекты доступа отображаются присвоенными им при создании именами. Для задания правила в правом поле «Выберите субъектов создателей» выбирается субъект доступа к создаваемым файлам, которым будет контролироваться доступ. Если какой-либо субъект доступа в этом поле не будет выбран, то к создаваемым им файлам доступ контролироваться не будет.

Для выбранного в правом поле субъекта доступа в левом поле «Выберите субъектов, осуществляющих доступ» поочередно выбираются субъекты (либо одновременно несколько субъектов, если их права доступа совпадают), которые будут иметь права доступа к рассматриваемым создаваемым объектам, и им назначаются требуемые права доступа, а также правила аудита.
Контроль доступа к компьютерным ресурсам

Видим, насколько упрощается задача администрирования, по сравнению с реализацией контроля доступа к статическим объектам. Это позволяет говорить о том, что метод дисперсионного контроля доступа может рассматриваться в качестве основы реализации процессной модели контроля доступа, в части решения основной решаемой в рамках ее построение задачи защиты информации – реализация разграничения и изоляции обработки данных приложениями.

Пример разграничительной политики доступа к создаваемым файлам приведен на рис. 4.27.

![Рис. 4.27. Пример разграничительной политики доступа к создаваемым файлам](image)

Ранее, говоря о технической реализации мандатного контроля доступа к создаваемым файловым объектам, мы отмечали, что для возможности обзора произведенной системой разметки файлов, создаваемых контролируемыми субъектами, в состав средства защиты включена специальная утилита обзора мандатных меток созданных файлов. Аналогичным образом с использованием этой утилиты можно просмотреть (удалить, создать новую) разметку созданных файлов и при дисперсионном методе контроля доступа к создаваемым файлам, см. рис. 4.28.

![Рис. 4.28. Отображение разметки созданных файлов](image)

Приведенные иллюстрации наглядно показывают, насколько упрощается задача администрирования при исключении из разграничительной политики доступа сущности объект доступа. Данное техническое решение запатентовано [15]. Заданием всего лишь нескольких правил доступа можно полностью, причем гарантированно (корректно) изолировать работу критических субъектов доступа, в том числе, критических приложений. Отметим, что именно возможность корректного изоляирования работы отдельных субъектов, причем, достигаемая простейшими настройками разграничительной политики доступа, можно отнести к основным достоинствам методов контроля доступа к создаваемым объектам.

4.6.2.2. Контроль доступа к создаваемым данным в буфере обмена

К создаваемым объектам также следует отнести данные, создаваемые и временно хранящиеся в буфере обмена, который используется в информационной системе обмена данными между приложениями. Это связано с тем, что при решении задачи изолированной обработки данных в рамках реализации процессной модели контроля доступа по каким-либо соображениям критичными субъектами, для них необходимо изолировать все возможности обмена данными с остальными субъектами обмена [53,54].

Поскольку контроль и разграничение прав доступа к данным, создаваемым в буфере обмена, также следует рассматривать, как контроль доступа к создаваемым объектам, здесь применимы принципы реализации разграничительной политики доступа, изложенные ранее. Аналогичным образом, создаваемые данные автоматически размещаются: диспетчером доступа запоминается, каким субъектом записаны данные в буфер обмена. Последующий же доступ к сохраненным в буфере обмена данным разграничивается между субъектами по заданным администратором правилам доступа. Опять же объект доступа исключен из разграничительной политики, разграничение прав доступа задаются исключительно между субъектами. Не разграничиваются права доступа по записи (созданию) данных в буфер обмена — разграничиваются последующий доступ к созданным подобным образом данным.

Замечание. Реализация контроля доступа для процессов к буферу обмена, как к статическому объекту, не имеет какого-либо смысла, поскольку в этом случае может разрешаться, либо нет, использование буфера обмена, как такого, в том числе и для собственного его использования приложением, что бессмысленно.

При назначении правил доступа к буферу обмена используются те же субъекты доступа, что и при назначении правил доступа к создаваемым файлам, созданные из интерфейса, приведенного на рис. 4.23.

Правила доступа к данным, создаваемым в буфере обмена, назначаются из интерфейса, представленного на рис. 4.29, и отображаются в интерфейсе, представленном на рис. 4.30.
Для задания правила в правом поле «Выберите субъектов, поместивших информацию» (в буфере обмена) выбирается субъект доступа, к данным, размещаемым в буфере обмена будет контролироваться доступ.

Для выбранного в правом поле субъекта доступа в левом поле «Выберите субъектов, берущих информацию» (из буфера обмена) поочередно выбираются субъекты (либо одновременно, если их права доступа совпадают), которые будут иметь права доступа к рассматриваемым данным, размещаемым в буфере обмена, и им назначаются требуемые права доступа, а также правила аудита.

Замечание. Настройка разграничительной политики доступа в данном случае также строится по принципу расширения дополнительными правилами канонической (диагональной) матрицы доступа. Поэтому, если выбрать в правом и в левом поле интерфейса одно и того же субъекта доступа, для него можно будет настроить только режимы аудита доступа. По умолчанию субъект доступа может взаимодействовать с собою через буфер обмена.

Рассмотрим, насколько просто реализуется разграничительная политика доступа к обрабатываемым в системе данным критическими приложениями. На примере изолирования работы в системе по обрабатываемым данным приложения Internet Explorer [54]. С этой целью создадим двух субъектов доступа — "Все" и "IE", см. рис. 4.31, и назначим для них правила доступа, представленные на рис. 4.32. Отметим, что всего две правила мы полностью изолировали работу критичного приложения (браузера) на доступу к создаваемым файлам.

Соответствующим образом, используя интерфейс, представленный на рис. 4.29, следует запретить субъекту доступа "IE" доступ к буферу обмена для субъекта, поместившего в буфер обмена данные "Все". Вот и все настройки разграничительной политики доступа, решающей одну из актуальных современных задач защиты информации! Причем при реализации подобной защиты не важно, каким образом будет задано критическое приложение встроенным свойствами и каким, в том числе, предполагаемым автоматизированное процесса с другой учетной записью, включая системную, это приложение не получит доступ к данным, обрабатываемым иными приложениями в информационной системе.
4.7. Технология защиты данных

Как ранее отмечали, методы контроля доступа к создаваемым объектам предназначены для защиты данных (включая данные, сохраняемые в конфигурационных файлах), поскольку именно в создаваемых объектах хранится обрабатываемая информация. Как следствие, именно эти методы должны реализовываться в информационной системе в первую очередь.

Данные методы защиты принципиально меняют собственную технологию защиты данных. Это обусловливается тем, что методами защиты, предполагающими реализацию разграничителей политики доступа субъектов к статичным объектам, вся защита построена на реализации защиты конкретных объектов, в которых администратором разрешается сохранение субъектами данных, соответственно, конфиденциальных, требующих защиты. Таким образом, объектом защиты является объект доступа, в который разрешено сохранять определенные данные.

При реализации контроля доступа к создаваемым объектам, наоборот, создание данных каким образом не контролируется – субъект доступа их может создавать где угодно. Объектом защиты является не конкретный объект, а любой объект, в котором созданы данные определенным субъектом доступа. Т. е. технология защиты принципиально меняется, что требует и реализации принципиально иного подхода к реализации соответствующих методов защиты данных. Принципиально различных подходов на примере реализации гарантированного удаления файлов [45].

Как отмечали (эту задачу защиты информации рассматривали во второй главе), гарантированное удаление файлов предполагает перехват системой защиты запроса на удаление файла, а в случае, если к этому файлу предъявляется требования к гарантированному удалению, системой защиты осуществляется задание администратору число раз записи в файл администратора шаблона (некого набора символов). После этого системой защиты передается управление операционной системе, которой уже осуществляется удаление файла.

Сравним реализацию гарантированного удаления файлов для различных методов контроля доступа.

Окно создания правила гарантированного удаления в системе защиты при использовании методов контроля доступа субъектов к статичным объектам представлено на рис. 4.33 а, при использовании методов контроля доступа к создаваемым объектам – на рис. 4.33 б, соответствующие интерфейсы отображения правил гарантированного удаления приведены на рис. 4.34.
Принципиально важной особенностью рассматриваемой технологии защиты данных, при реализации криптографической защиты обрабатываемой информации, является возможность реализации принудительного управления шифрованием данных. Дело в том, что в этом случае можно задавать правила шифрования, при реализации которых, где бы субъектом доступа (например, пользователем или группой пользователей, для которых создается один ключ шифрования) и какой бы файл, в том числе, конфигурационный, не создавался, он будет принудительно для субъекта шифроваться.

При этом учетные данные субъекта (группы субъектов), созданного файл, целесообразно автоматически помещать непосредственно в файл, когда также целесообразно записывать значение вектора инициализации2, т.к. в этом случае можно реализовать криптографическую защиту файлов в распределенной информационной системе – файл созданный соответствующим субъектом доступа будет автоматически шифроваться при любом общем его переноса с одного компьютера информационной системы на другой, где к нему может быть получен доступ и он сможет быть расшифрован только определенным субъектом (в соответствии с заданными правилами).

Для разграничения прав доступа и для расшифрования файла при этом вполне достаточно такой служебной информации, которая помещается в файл при его создании – учетной информации субъекта, либо группы субъектов (при методе мандатного контроля доступа – это учетная запись и метка безопасности), необходимых для выбора ключа шифрования, и значения вектора инициализации. Данное техническое решение авторами запатентовано [17].

Замечание. Вопросы криптографической защиты в данной книге не рассматриваются по причине того, что это самостоятельное направление защиты, которое требует собственного серьезного исследования.

4.8. Иллюстрация возможности использования методов контроля доступа в комплексе

Ранее отмечали, что для эффективного решения современных актуальных задач защиты информации от несанкционированного доступа, методы контроля доступа к статичным и к создаваемым объектам должны реализовываться в комплексе. Помимо этого, также необходимо учитывать и другие аспекты защиты информации, такие как выделение областей доступа, контроль за предоставлением доступа, защита информации от несанкционированного доступа к субъекту, а также учетные данные субъекта, созданных файлов, и другие аспекты, которые необходимо учитывать при реализации комплексных систем защиты информации. При этом учетные данные субъекта (группы субъектов), созданного файл, целесообразно автоматически помещать непосредственно в файл, когда также целесообразно записывать значение вектора инициализации, т.к. в этом случае можно реализовать криптографическую защиту файлов в распределенной информационной системе – файл созданный соответствующим субъектом доступа будет автоматически шифроваться при любом общем его переноса с одного компьютера информационной системы на другой, где к нему может быть получен доступ и он сможет быть расшифрован только определенным субъектом (в соответствии с заданными правилами).

Векторов инициализации. В случае, если задаются ограничения на объем данных, шифруемых одним сессионным ключом, позволяет осуществлять шифрование одного большого по объему файла с различными — по частям, сессионными ключами.
вать в комплексе. Рассмотрим практический пример применения данного подхода к защите информации [41].

Одной из наиболее актуальных сегодня задач защиты является защита от угроз атак на Java-машины. В общем случае следует рассмотреть защиту от атак на Java-машины в двух принципиально различных условиях — Java-машина используемая на компьютере при работе только с локальными приложениями, либо Java-машина используется при работе с сетевыми приложениями, например, с интернет-браузером.

Рассмотрение этих альтернативных (по области использования) случаев вызвано следующими причинами: во-первых, они широко распространены на практике, во-вторых, для них принципиально различные задачи, как следствие, реализация защиты [41].

4.8.1. Защита от атак на Java-машины, используемые для работы с локальными приложениями

Очевидно, что подавляющая часть внедрения вредоносного кода, равно как и атак на уязвимости приложений, осуществляется из внешней сети. Как следствие, в предположении, что Java-машина должна использоваться на защищенном компьютере, подключенном к внешней сети, для работы только с локальными приложениями, может быть сформулирована задача защиты, состоящая в следующем:

- предотвратить возможность доступа Java-машины к коду, внедренному из внешней сети;
- предотвратить возможность запуска Java-машины сетевыми приложениями (для предотвращения использования уязвимостей Java-машины при атаках из сети).

Рассмотрим решение сформулированной задачи защиты.

Предположим, что в качестве сетевого приложения на защищаемом компьютере используется стандартный браузер операционной системы Windows (процесс iexplore.exe), а в качестве Java-машины java.exe JVM от Oracle (процесс java.exe). Соответственно, это наши субъекты доступа, для которых необходимо реализовать разграничительную политику.

Зададим эти субъекты доступа в среде защиты, см. рис. 4.35.

Глава 4. Принципы и методы контроля доступа к создаваемым объектам

Для того, чтобы предотвратить возможность доступа Java-машины к внедренному на компьютер коду из сети, среди которого велика вероятность вредоносного кода, используются контроллер доступа к создаваемым файловым объектам и предотвращающим заданием соответствующего правила доступа к создаваемым файлам возможность доступа Java-машины к любым файлам, создаваемым браузером, см. рис. 4.36.

Рис. 4.36. Правило контроля доступа к создаваемым файлам

Рассмотрим, что мы получим от реализации подобной простейшей разграничительной политики доступа. Все файлы, загружаемые на компьютер браузером, будут автоматически размещаться в системе защищенными, а к ним будет контролироваться доступ. При этом, следуя созданному правилу, Java-машина не сможет получить доступ к подобным файлам.

Следующим шагом реализации разграничительной политики доступа будет использование контроля доступа к статическим объектам, с целью блокировки доступа браузера к объектам Java-машины, см. рис. 4.37. Такая мера обеспечивает от извлечения вредоносного кода на языке Java, по существу, вызова его с помощью уязвимостей браузера. Так же следует отметить, что в данном случае теряется возможность запуска любого Java кода браузером, в том числе и Applet-ов.

Замечание. В качестве статического файлового объекта, доступ к которому контролируется, задана вся директория JVM, см. рис. 4.37.

Рис. 4.37. Правила контроля доступа к статическим файловым объектам

Как видим, сформулированная задача защиты решена в полном объёме, причем решена она при минимальной трудоемкости создания разграничительной политики доступа к файловым объектам.
4.8.2. Защита от атак на java-машины, используемые для работы с сетевыми приложениями

В данном случае защита должна строиться в предположении, что java-машина будет атакована из сети, причем будет атакована успешно. Как следствие, защита здесь должна строиться с целью минимизации последствий от реализованной сетевой атаки на java-машину.

Сформулируем задачу защиты:

- предотвратить возможность доступа java-машины к обрабатываемой на защищаемом компьютере информации;
- предотвратить возможность доступа java-машины к системным файлам, кроме тех, которые ей необходимы для корректного функционирования.

По сути, задача защиты здесь сводится к изолированию работы java-машины в системе.

Создадим правила контроля доступа к статическим объектам. Решение задачи защиты обеспечивается простейшими настройками разграничительной политики доступа, представленными на рис. 4.38. Данной разграничительной политикой виртуальной машины разрешен доступ только к собственным объектам, причем только на чтение и исполнение, ко всем остальным объектам файловой системы (объект задан маской "*" — "Bce"), какой-либо доступ ей запрещен.

![Рис. 4.38. Разграничительная политика доступа к статическим файловым объектам](image)

Видим, что сформулированная задача защиты решена в полном объеме. Однако для обеспечения корректности работы java-машины в рассматриваемых приложениях, настройки, представленные на рис. 4.38, необходимо несколько расширить. Например, дополнительные файлы здесь скармливается уже не браузер, а сама java-машина. Для обеспечения корректной работы системы в рассматриваемых условиях необходимо реализовать следующие дополнительные требования: обеспечить возможность доступа в сеть java-машины и доступа ее к временным папкам, в которых хранятся скопированнные файлы Applet-ов, созданные браузером. Корректная разграничительная политика доступа java-машины к файловым объектам представлена на рис. 4.39.

![Рис. 4.39. Корректная разграничительная политика доступа к статическим файловым объектам](image)

Из рис. 4.39 видим, что обеспечение корректности работы java-машины с сетевыми приложениями не привело к снижению эффективности защиты, сформулированные задачи защиты решены в полном объеме, причем трудоемкость их решения, в части задания разграничительной политики доступа к файловым объектам, минимальна.

Подобное исследование в полной мере иллюстрирует, что на практике имеется смысл использование одновременно методов контроля доступа и к статическим, и к создаваемым объектам. При этом эффективная защита может достигаться за счет предоставления субъектам права доступа только к необходимым им для выполнения решаемых задач при обеспечении корректности работы системы и приложений прав доступа к объектам.

4.9. Принципы и методы контроля доступа с автоматической разметкой объектов в общем случае

Выше мы рассмотрели метод контроля доступа к создаваемым объектам, в частности, к создаваемым файлам. Основу реализации данного метода составляет наследование (автоматическая разметка при создании) созданным объектом, в частности файлом, устлеров данных создавшего этот объект субъекта доступа (идентификатор субъекта, определяющего соответствующими тремя сущностями), либо его метка безопасности, которые могут со-
храняться в атрибутах файла (например, в альтернативных потоках), либо непосредственно в файле, выделенной для этой цели фиксированной области создаваемых файлов.

Используем возможности автоматической разметки файлов в общем случае, т.е. и применимо к реализации контроля доступа к статическим объектам, с целью упрощения задачи администрирования, на этот здесь будут интересоваться системные файловые объекты.

С учетом введенной выше классификации файловых объектов, их можно подразделить на статические и создаваемые. К статическим относятся объекты, нас здесь интересуют файлы, присутствующие на момент создания разграничительной политики доступа администратором файла – это системные файлы, в отношении которых мы здесь и рассматриваем решение задачи контроля доступа, создаваемые файлы – это файлы, создаваемые пользователями впоследствии, в процессе работы на компьютере.

4.9.1. Принципы контроля доступа

Итак, применимо к создаваемым файловым объектам, при реализации контроля доступа требуется обнаруживать факты создания (операция "запись") новых файлов (либо модификации уже существующих, но неразмеченных файлов), и автоматически "размечать" создаваемые/модифицируемые (не размеченные ранее) файлы – записывать в качестве атрибута, либо непосредственно в "теле" файла, в зависимости от реализации, учетную информацию субъекта, создавшего/модифицировавшего этот файл.

Для статических файлов ситуация с разметкой принципиально иная, применимо к ним удаление и модификация (операции "удаление", "переменование" и "запись") являются несанкционированными действиями (если подобные действия в отношении файла разрешены, то подобный файл уже следует отнести к создаваемым, реализовав для него соответствующие принципы контроля доступа).

Как следствие, размечать статические файлы необходимо совсем иначе. Следует записывать в качестве атрибута, либо непосредственно в "теле" файла, в зависимости от реализации, учетную информацию субъекта, прочитавшего/исполнявшего Этот файл, и выполненное им действие в отношении статического файла (чтение или исполнение) [37, 44].

Принципы контроля доступа к статическим файлам с их автоматической разметкой состоят в следующем [37]:

1. Сущность "объект" должна быть исключена из схемы реализации разграничительной политики доступа к статическим объектам.

2. Статический объект, в данном случае рассматриваем файл, однозначно характеризуется субъектом, использующим этот объект и действием (чтение исполнение), производимым данным субъектом в отношении объекта.

3. При реализации разграничительной политики доступа (назначении правил доступа) в общем случае должны использоваться три сущности: идентифицируемый субъект, использующая статический объект, способ использования субъектом объекта (чтение или исполнение – характеризует вид статического объекта – тип файла) и идентифицируемый субъект, запрашивающий последующий доступ к размещенному статичному объекту. Субъект доступа, как отмечалось ранее, в общем случае должен идентифицироваться тремя сущностями: исходный идентификатор пользователя, эффективный идентификатор пользователя, полнотуправление имя процесса, запрашивающего доступ к объекту.

4. Статический файловый объект (файл) должен автоматически размещаться – наделяться при обращении к нему (чтение или исполнение) системой защиты, в результате чего, включать в себя (в качестве атрибута, либо непосредственно в "теле" файла, в зависимости от реализации), учетную информацию субъекта, использующего этот файл, и способ использования файла субъектом (был прочитан, либо исполнен).

5. При запросе доступа к любому объекту система защиты анализирует наличие, а при наличии – содержимое ункодированной объектом учетной информации использующего его субъекта доступа. При наличии анализирует заданные правила доступа, в результате чего предоставляет запрошенный субъектом доступ либо отказывает в нем. При отсутствии аналитирует правило, открывает доступ к не разрешенным (не унаследованным схему учетной информации субъекта) объектам.

Определение. Под контролем доступа с автоматической разметкой статических объектов понимается контроль доступа, основанный на идентификации субъекта «объект доступа» из разграничивательной политики доступа (объект доступа не используется при запретных правила доступа), за счет автоматической разметки иначе используемых статических объектов, применяемой для последующего контроля к ним доступа.

320

321
Лемма 4.8. Метод контроля доступа с автоматической разметкой статичных объектов на практике корректно может реализовываться только совместно с методом контроля доступа к создаваемым объектам.

Доказательство. Докажем от обратного - пусть это не так. Тогда любой созданный объект после его прочтения будет размещен системой защиты, как статичный объект, соответственно, применимо к этому созданному объекту будут применяться правила, как к статичному объекту, который в общем случае может противоречить правилам доступа к создаваемым объектам.

При этом любой файловый объект на самом деле, относится либо к статичным, либо к создаваемым. Если он будет создан в системе, то в отношении него будут применяться правила контроля доступа к создаваемым объектам, если изначально к нему будет запрос доступа на чтение или исполнение, то к статическим. Т.е. имеем две разграничительные политики доступа реализуемых одновременно (описываемых двумя различными матрицами доступа - к создаваемым и к статическим объектам), при этом любой объект будет подпадать либо под одну, либо под другую разграничительные политики доступа.

4.9.2. Модель контроля доступа
Построим модель контроля доступа [37]. Если считать, что множество  
\[ C = \{ C_1, ..., C_i \} \]  
линейно упорядоченное множество субъектов доступа, и  
\[ R = \{ R_1, ..., R_k \} \]  
конечное множество прав доступа (чтение (r), запись (w), удаление (d), исполнение (x) и т.д., отсутствие прав доступа (0)) субъекта  
\[ C_i \]  
к статичному объекту, использованному субъектом  
\[ C_i \]  
и  
\[ C_j \]  
то правила доступа задаются в данном случае двумя матрицами доступа  
\[ M \]  
используемыми для реализации разграничительной политики доступа (условимся в строках матрицы указывать учетную информацию субъектов, запрашивающих доступ к объектам, а в столбцах - учетную информацию субъектов, унаследованную используемыми ими объектами) - матрицей доступа к объектам, используемым для чтения  
\[ M_{rw} \]  
и матрицей доступа к исполненным объектам  
\[ M_{mx} \]  
которые в общем случае принципиально различаются, поскольку различаются требования к правилам доступа к подобным статическим объектам.

Соответственно в любой момент времени система описывается своим текущим состоянием  
\[ Q_{mx} = (C,M_{mx}) \]  
и  
\[ Q_{mx} = (C,M_{mx}) \]  
т.е. матрицы  
\[ M_{mx} \]  
и  
\[ M_{mx} \]  
- ячейки соответствующих матриц, содержащие набор прав доступа к соответствующим статическим объектам.

Замечание. В общем случае может контролироваться доступ только к объектам, используемым отдельными субъектами, например, системным пользователем. При этом к остальным объектам права доступа разграничиваться не будут.

Теперь о требованиях к реализации данного метода контроля доступа.

Сначала рассмотрим статичные объекты, размещенные, как исполняемые.

Требование. К статическим объектам, размещаемым как исполняемые, должен предотвращаться доступ на запись/модификацию и удаление.

Замечание. В первую очередь это относится к запрету подобного права доступа для интерактивных пользователей.

В результате выполнения данного требования получаем следующую матрицу доступа к исполняемым объектам  
\[ M_{mx} \]:

\[
M_{mx} = \\
\begin{bmatrix}
C_1 & C_2 & \ldots & C_i \\
\end{bmatrix}
\]

Выполнение данного требования направлено на защиту системных исполняемых объектов от их несанкционированной модификации или удаления. Модификация предполагает возможность нападения легальных программ вредоносными свойствами, удаление – потенциальную возможность нарушения доступности обрабатываемой информации, за счет отказа в обслуживании.

Реализация рассмотренного требования приводит к проблеме (а в общем случае выявляет проблему, что рассмотрим далее более подробно) автоматических обновлений исполняемых файлов системы и приложений, поскольку для всех субъектов реализуется запрет модификации исполняемых файлов. Для решения этой проблемы (проблемы автоматических обновлений), естественно, с соответствующим снижением уровня безопасности, матрица  
\[ M_{mx} \]  
может быть расширена – может быть реализован компромиссное решение – объект доступа может модифицироваться и/или удаляться только исполняемые им же объекты, в пределе получим следующую матрицу доступа  
\[ M_{mx} \]:

\[
M_{mx} = \\
\begin{bmatrix}
C_1 & C_2 & \ldots & C_i \\
\end{bmatrix}
\]

При этом проблема внедрения на компьютер и последующий запуск вредоносной программы, по средством модификации исполняемого файла, в общем случае не решается, но предотвращается возможность влияние одних субъектов доступа на системные файлы, исполняемые другими субъектами доступа.

Теперь обратимся к статическим объектам, размеченным, как прочтенные.

**Требование.** Правила доступа к статическим объектам, размеченным как прочтенные, должны задаваться канонической матрицей доступа $M_w$ следующего вида:

$$
M_w = \begin{bmatrix}
C_1 & C_2 & \cdots & C_i \\
C_1 & r, x, w & r, x & \cdots & r, x \\
C_2 & r, x & r, x, w & \cdots & r, x \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_{i-1} & r, x & r, x & \cdots & r, x \\
C_i & r, x & r, x & \cdots & r, x, w
\end{bmatrix}
$$

В этом случае системные объекты, не являющиеся исполняемыми – в первую очередь, это файлы конфигурации, используемые одними субъектами доступа, не смогут быть несанкционированно модифицированы или удалены другими субъектами доступа.

Естественно, что возможность расширения подобной канонической матрицы доступа $M_w$ связана исключительно с вопросами корректности функционирования системных средств и приложений, поскольку речь здесь идет не об обрабатываемой информации, а о конфигурационных данных.

Таким образом, как видим, метод контроля доступа к статическим объектам, основанный на автоматической разметке объектов, позволяющий существенно упростить задачу администрирования системы защиты, за счет исключения сущности объект доступа из разграничительной политики доступа – все разграничения задаются исключительно между субъектами, может быть использован не только при реализации контроля доступа к создаваемым объектам, но и при реализации контроля доступа к статическим объектам – к системным файлам. Это позволяет рассматривать метод контроля доступа, основанный на автоматической разметке объектов, в случае комплексного решения задач контроля доступа и к создаваемым, и к статическим объектам, в качестве альтернативного решения методу контроля доступа субъектов к статическим объектам.

### 4.9.3. Методы защиты

Рассмотрим примеры методов защиты, реализующих контроль доступа с автоматической разметкой объектов, основанных на реализации запатентованных авторами технических решений.

#### 4.9.3.1. Защита исполняемых файлов

Для реализации данного метода защиты размечать следует статические объекты – исполняемые файлы, и в качестве разметки статического файла достаточно указывать признак файла – исполняемый. Диспетчер доступа при исполнении файлов, автоматически их помечает, присваивая им признак исполняемого. Заметим, что фиксировать нужно именно исполнение файла, а не открытие файла на исполнение, в противном случае будет много ложных срабатываний. Правила доступа в данном случае простейшие: файлу, помеченному, как исполняемому, запрещается доступ любым субъектам на удаление, модификацию, переназначение.

Рассмотрим, что даст подобное решение. Через некоторое время работы системы все (по крайней мере, основные, зависят от времени работы системы в режиме расширения, можно также принудительно запустить важнейшие для пользователя приложения, для разметки исполняемых файлов) используемые на компьютере исполняемые файлы (как системы, так и приложений) будут размещены. Их станет невозможно ни удалить, ни модифицировать, ни переназначить. При этом, как видим, не требуется практически никаких настроек – все настройки осуществляются автоматически, за счет автоматической разметки статических файлов.

К слову сказать, о защите от вредоносных программ. Ранее мы рассмотрели метод защиты от запуска создаваемых файлов и его техническую реализацию, применение которого может эффективно использоваться для защиты от запуска вредоносных программ. При реализации подобного метода, никакая, никаким способом установленная на компьютер вредоносная программа не может быть запущена. Однако при этом остается угроза подмены вредоносной программой легально установленного в системе исполняемого файла (контроль осуществлялся в отношении только создаваемых файлов). Да, в этом случае он не смог бы быть исполнен (т.к. при модификации
Глава 4. Принципы и методы контроля доступа к создаваемым объектам

Рассмотрим, что делает подобное решение. Через некоторое время работы системы в целом, основные, зависят от времени работы системы в режиме отладки) используются системой на компьютере статические файлы. Файлы будут размещены. Их становится невозможно несанкционированно удалять, модифицировать, переименовывать. К слову сказать, попытка несанкционированного создания нового системного файла (под видом системного), уже подпадает под разграниченную политику доступа к создаваемым файлам.

4.9.3.3. Защита статических файлов в общем случае

По аналогии с тем, как это было описано выше, применимо к защите системных файлов OC, можно налаживать использование статических файлов различными субъектами (как указано, в общем случае субъект определяется тремя сущностями: исходное имя пользователя (учетная запись, от лица которой запущен процесс), полномочное имя исполняемого файла процесса, эффективное имя пользователя (учетная запись, от лица которой запущен процесс), предотвратив, тем самым, возможность их несанкционированного удаления, модификации, переименования другими субъектами.

Для реализации данного метода размещают статические файлы, используемые контролируемым субъектом (для которого решается задача защиты).

Диспетчер доступа при доступе к статическим файлам (чтение, исполнение), контролируемыми субъектами, автоматически их помечается — размещать (в качестве атрибута, либо непосредственно в "теле" файла, в зависимости от реализации), учетную информацию контролируемого субъекта, используемого этот файл.

Последующий доступ к размещенному файлу, разрешается/запрещается в соответствии с правилами, заданными в матрице доступа. Пример матрицы доступа, реализующей контроль доступа к статическим файловым объектам для интернет-браузера, приведен в табл.4.16. Это лишь пример. В нем мы разрешаем доступ к статическим файлам, используемым браузером, только непосредственно браузеру и системе.

4.9.3.2. Защита системных файлов

Для реализации данного метода размещают статические файлы, используемые OC (пользователем System), соответственно, в качестве разметки используемого OC статического файла следует использовать учетную запись системы.

Диспетчер доступа при доступе к статическим файлам (пользователем System) статическим файлам на чтение, исполнение, автоматически их размещает — размещает (в качестве атрибута либо непосредственно в "теле" файла, в зависимости от реализации), учетную информацию системного субъекта [37].

Правила доступа в данном случае опять же простейшие: к файлу, размещенному, как системный — используется OC (пользователь System, процессом System), — запрашивается доступ любым иным субъектам на удаление, модификацию, переименование. Правило задается несколькими строками в матрице доступа, табл. 4.15.

Первым правилом (см. табл. 4.15) задается право полного доступа системе к используемой ею (соответствующим образом, размеченной) статическим файлам. Вторым правилом (субъект определяется как " sistem" — любой, т.е. более точный описатель субъекта) предотвращается любой доступ сто роны любого иного субъекта к системным файлам, используемым OC.

Поясню, что подобное правило можно задать для системы защиты "по умолчанию", т.к. оно универсально, и вообще не выносить в интерфейс.

Таблица 4.15. Матрица доступа к статическим системным файловым объектам

<table>
<thead>
<tr>
<th>№ правила в разграниченной политике</th>
<th>Учетная информация субъекта, использующего статический файловый объект в создаваемом файловом объекте</th>
<th>Учетная информация субъекта, запрашивающего доступ к размеченному файловому объекту, и права доступа субъекта к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>System, System, System</td>
<td>&quot;*, <em>, (</em>)&quot;</td>
</tr>
</tbody>
</table>

будет размещено, как создаваемый), но сама возможность подобной подмени может привести к нарушению работоспособности системы (ввиду того, что системный исполняемый файл, либо исполняемый файл приложении не может быть выполнен после разметки), в конечном счете, к нарушению доступности информации. При использовании же данного метода защиты подобная угроза снимается.
Однако, что подобную разграниченную политику доступа можно реализовать в отношении любого (любых) субъекта доступа, как пользователя, так и процесса. В частности, в пределе можно обеспечить доступ всем приложениям только к необходимым им для работы статическим файлам.

Как видим, методы контроля доступа к статическим файловым объектам на основе их автоматической разметки позволяют решать важнейшие задачи защиты системных ресурсов. И что важно, сложность администрирования средства защиты, при решении подобных сложнейших задач, опять же мнимальна!

Таким образом, в общем случае следует говорить о принципе контроля доступа на основе автоматической разметки файловых объектов, объединяя в одном средстве методы контроля доступа и к создаваемым, и к статическим файловым объектам. Реализация данного подхода позволяет в общем случае корректно решать сложнейшие задачи защиты, добиваясь при этом максимального упрощения задачи администрирования средства защиты.

Проясшественное сказанное примером. Реализуем защищенную работу интернет-браузера.

Технология защиты сводится к изолированнию работы интернет-браузера на компьютере, в предположении, что, в результате осуществленной на данном приложении атаки, оно может быть подключено в общем случае произвольными вредоносными функциями.

Реализация контроля доступа к создаваемым файловым объектам здесь состоит в следующем. Каждый создаваемый интернет-браузер определён автоматически размещается — наделяется при создании учетной информации субъекта доступа (в качестве процесса в системе) в качестве процесса интернет-браузера).

К размеченным подобным образом созданным файлам метрией доступа разрешается запись/чтение (запрашивается исполнение) интернет-браузера, запрещается доступ, в том числе, исполнение любым иным субъектом (за или внутренним процессом P0, внедренным на компьютере браузера).

Интернет-браузер разрешается доступ чтение/запись только к созданным им же файлам — размеченным, к неразмеченным файлам какой-либо доступ запрещается. Также разрешается соответствующий доступ к размеченным (используемым браузером) статическим объектам, см. ниже.

Замечание. Опыт же средства защиты, с целью минимизации числа ложных срабатываний, должно различить открытие файла на запись, исполнение, и собственно выполнение программной записи, исполнения, что собственно и должно контролироваться.
Реализация контроля доступа к системным файловым объектам состоит в следующем. Любой файл, доступ к которому запрещается браузером для чтения или исполнения, т.е. используется браузером, размещается — находится учетной информацией субъекта доступа (в качестве процесса в субъекте устанавливается процесс интернет-браузера), используемого файл, и признак использованного файла (открывался на чтение или исполнение). Для файлов, открывавшихся на исполнение, в матрицу доступа субъект запрашивает запись, модификация и удаление, для файлов, открывавшихся на чтение, разрешается чтение и запись (в случае записи в них, назначается в матрицу доступа, реализуемой для создаваемых (модифицируемых) файлов), запрашивается исполнение.

Интернет-браузеру разрешается соответствующий доступ только к используемым им статическим файлам — размещенным, с неразмеченными файлам доступ запрещается. Также разрешается соответствующий доступ к размещенному браузером создаваемым файловым объектам, см. выше.

Интеграция разграничительных политик доступа к создаваемым и к системным файлам в единую осуществляется по принципу логического «или»: доступ должен быть разрешен либо доступ к создаваемым файловым объектам, либо контролю доступа к системным файловым объектам (в обоих случаях реализуется разрешительная политика доступа к ресурсам).

Вы видите, что в данной схеме контроля доступа присутствует противоречие. Матрица доступа к создаваемым файловым объектам вступает в противоречие с матрицей доступа к статическим объектам, которые не создаются в процессе работы пользователями — к ним будет невозможен доступ субъекту. Для устранения данного противоречия необходим режим «обучения» системы защиты. В данном режиме должен действовать только контроль доступа к системным (статическим) объектам, к создаваемым, на «время обучения», контроль доступа должен отключаться, с целью возможности разметки используемыми браузером системных объектов.

В процессе «обучения», который, естественно, предполагает работу пользователя с доверенными сетевыми объектами, размещаются системные объекты, используемые браузером. При переходе в «боевой» режим будет обеспечен доступ к создаваемым и к статическим (уже размеченным) объектам.

Естественно, что в процессе «обучения» не все системные объекты (в общем случае, используемые не во всех режимах работы браузера) могут быть размещены. Для обеспечения корректной работы браузера в общем случае должны быть предусмотрена следующая возможность разметки пользователем системных файлов в процессе работы. При обращении браузером к неразмещеному файлу на чтение или исполнение, доступ к такому файлу средстом защиты блокируется. Пользователя же предлагается решить, санкционирован ли тот файл для чтения, исполнения браузером. Если да, то пользователю предлагается разметить данный системный файл (соответственно, указывает признак системного файла — должен открываться браузером на чтение или исполнение), нарезая тем самым файл в разряд санкционированных для исполнения интернет-браузером, либо удалить этот файл.

Рассмотрим, что даст подобное решение. Через некоторое время работы системы в режиме «обучения», все (по крайней мере, основные, используемые интернет-браузером статические файлы будут размещены, возможно, переезжая в боевой режим работы. В боевом режиме работы интернет-браузера будет предоставлена возможность работать только с создаваемыми им же файлами (к любым иным файлам, как следствие, к информации, обращающейся на компьютере, любой доступ приложению, как в части хищения, так и в части модификации или удаления будет невозможен). Предоставленная большая возможность запуска любым пользователем (в том числе, системным) любых приложений установленной на компьютере интернет-браузером варианты программы. Защита системных файлов (в том числе, используемых приложения в иных приложений) — браузер будет предоставлена возможность доступа только к необходимым ему для работы системным файлам. Полная изолированность работы интернет-браузера!

4.9.3.4. Пример практической реализации метода защиты с автоматической разметкой объектов

При разработке методов защиты необходимо учитывать их потенциального потребителя. Сложность существующих методов контроля доступа, эффективность которых доступа информации в современных условиях, как ранее показано, может быть достаточно высокой, существенно ограничивает их практическое использование при реализации защиты домашних компьютеров, поскольку их внедрение, администрирование и эксплуатация требуют определенной квалификации в области информационной безопасности. Кроме того, целевые атаки на подобные вычислительные средства не осуществляются, поскольку обрабатываются на них информация в общем случае мало кого интересует. Основную угрозу безопасности для таких компьютеров составляют вредоносные программы. В результате этого, в данных приложениях нашли широкое применение антивирусные системы защиты, об эффективности которых ранее было сказано.

Выше мы показали, насколько просто в администрировании методы контроля доступа с автоматической разметкой объектов (как создаваемых, так
и статичных), что позволяет сделать предположение о возможности их применения и для защиты личных компьютеров.

Рассмотрим техническое решение, реализованное в "Средстве защиты (СЗ) "Панцирь" [24] (далее СЗ), направленное на защиту от предпосылочных программ, иллюстрирующее подобную возможность.

Идея реализуемого метода защиты состоит в следующем. Любой создаваемый на компьютере в процессе работы файл средством защиты автоматически размещается, как создаваемый (то же происходит и при модификации не размещенного ранее системой защиты файла), а любой исполненный — как легальный исполнимый.

Это обусловливается тем, что вредоносная программа может быть создана, как по средством несанкционированного внесения на компьютер нового исполняемого файла, так и по средством модификации легального исполнимого файла, используемого на компьютере.

Соответствующие признаки размещаемых файлов (создаваемый, либо исполняемый) средством защиты автоматически помещаются в атрибуты (альтернативные потоки) соответствующих файлов (при создании/модификации, соответственно, при исполнении). При любом запросе доступа к файлу, средством защиты перехватывается запрос и анализируется наличие разметки файла, к которому запрашивается доступ. Если разметка файла нет, доступ разрешается (при модификации файла или при его исполнении, соответствующий файл соответствующим образом автоматически размещается). Если разметка есть, то она анализируется на непротиворечивость запрашиваемого доступа. Если запрашивается исполнение создаваемого (размещаемого) файла, такой запрос доступа средством защиты отклоняется, отключается он и в том случае, если запрашивается модификация/удаление/переназначение исполнимого (размещаемого) файла [34].

После установки СЗ и перезагрузки компьютера средство защиты автоматически запускается (автоматически далее запускается при каждом включении/перезагрузке) и полностью готово к работе, никаких настроек не требуется. В процессе эксплуатации также можно обойтись без таких-либо взаимодействий с СЗ, которая в реальном времени информирует пользователя о выявленных попытках несанкционированного доступа и ведет соответствующий журнал регистрируемых событий (аудита). При желании (необходимости) пользователь может проанализировать этот журнал.

Меню управления СЗ представлено на рис. 4.40, форма уведомления пользователя о выявленной попытке несанкционированного доступа на рис. 4.41.

Можно проигнорировать это уведомление и работать далее (попытка запуска несанкционированного создаваемого файла, либо несанкционированной модификации исполняемого файла была предотвращена), а можно проанализировать нештатную ситуацию. Для этого требуется открыть меню, рис. 4.40, и выбрать в нем закладку «Журнал», откроется журнал событий (аудита), см. рис. 4.42.

Рис. 4.41. Уведомление пользователя о попытке несанкционированного доступа к размещенному файлу

Рис. 4.42. Журнал событий (аудита)

Журнал событий имеет следующую структуру, см. рис. 4.42:
- «Дата/время» (временные характеристики регистрации события);
- «Файл» (полноопоптевое имя размещенного файла, несанкционированная попытка доступа к которому была предотвращена С3).
Контроль доступа к компьютерным ресурсам:

- Процесс, осуществивший доступ (полномоченное имя исполняемого файла процесса, совершенніе несанкционированную попытку исполнения размеченного созданного файла, либо модификацию/удаление/переименование размеченного исполняемого файла);
- Пользователь, осуществивший доступ (имя пользователя, учетной записи), который (от лица которого) совершена попытка несанкционированного доступа к размеченному файлу);
- Процесс, создавший файл, или Процесс, осуществляющий запуск файла (полномоченное имя исполняемого файла процесса, создавшего файл, к которому запрашивается несанкционированный доступ на исполнение, или полномоченное имя исполняемого файла процесса, исполняющего файл, к которому запрашивается несанкционированный доступ на модификацию/удаление/переименование). Пиктограммой иллюстрируется тип зарегистрированного события: запуск созданного или модификации/удаления/переименование исполняемого файла;
- Пользователь, создавший файл, или Пользователь, осуществлявший запуск файла (имя пользователя, учетной записи), который (от лица которого) создан файл, к которому запрашивается несанкционированный доступ на исполнение, или имя пользователя (учетной записи), исполняющего файл, к которому запрашивается несанкционированный доступ на модификацию/удаление/переименование). Пиктограммой иллюстрируется тип зарегистрированного события: запуск созданного или модификации/удаления/переименование исполняемого файла.

Выбрав файл в поле «Файл» журнала событий, см. рис. 4.42, можно открыть программу СЗ обзора файловой системы, рис. 4.43, для выполнения необходимых действий с размеченным выбранным файлом, либо с файлом.

Глава 4. Принципы и методы контроля доступа к созданным объектам

В обзоре отображаются выбранный файл и его разметкой. Здесь используются соответствующие пиктограммы для указания типа файла. На рис. 4.43а отображен размеченный СЗ созданный файл, на рис. 4.43б - размеченный СЗ исполняемый файл.

Выбрав требуемый файл в обзоре, можно открыть меню, предоставляющее возможность либо удаления файла, либо удаления его разметки, рис. 4.43б. При удалении разметки файла, впоследствии, его можно будет благополучно использовать либо соответственно модифицировать/удалить/переименовать, до очередной его разметки СЗ.

Для этого требуется выбрать в меню закладку «Обзор», см. рис. 4.40, открыть проводник СЗ обзора файловой системы, см. рис. 4.44.
Проблема автоматического обновления программных средств состоит в следующем. Нужно понимать, что в случае разрешения некоторому процессу особого режима, в системе модифицировать исполняемые файлы и/или создавать файлы, которые далее будет разрешено выполнять, тем самым создается угроза уязвимости реализации, которая может быть использована злоумышленником для запуска вредоносных программ. С этой целью может быть использована уязвимость (ошибка программирования) подобного особого режима, подмена виртуального канала связи, используемого разработчиком соответствующего программного обеспечения для автоматических обновлений, и т.д. Как следствие, получаем следующее противоречие: с одной стороны, для повышения уровня безопасности целесообразно максимально оперативно устранять выявленные уязвимости программных средств, что может достигаться автоматическим обновлением их компонент из внешней сети, с другой же стороны, при практической реализации подобного автоматического обновления вносятся дополнительные угрозы безопасности, связанные с потенциальной возможностью внедрения и использования вредоносных программ критическими процессами, ответственными за обновление программных средств.

Естественно, что для обеспечения безопасности автоматического обновления программного средства из сети полностью ответственен разработчик этого программного средства, что является, на взгляд авторов, самостоятельной и крайне важной современной задачей защиты информации. Однако, поскольку далеко не всегда подобная важнейшая современная задача решается (либо решается достаточно эффективно), разрешение автоматического обновления имеет смысл только для тех программных средств, разработчик которых гарантирует безопасность обновлений и им реализованы соответствующие меры защиты. Прежде чем разрешить автоматическое обновление того или иного программного средства, следует поинтересоваться, как разработчик обеспечивает безопасность такого обновления. Если никак, то лучше отказаться от подобной возможности, а все необходимые обновления осуществлять локально, при отключенном C3.

В случае же, если существует уверенность в безопасности проводимых разработчиком программного средства автоматических обновлениях (либо с целью достижения компромиссного решения между достижением уровнем защиты и удобством эксплуатации средства защиты), процесс, ответственный за обновление того или иного программного средства (полупустое имя его исполняемого файла легко узнать из журнала событий (аудита), см. рис. 4.42) может быть отнесен к категории "особых" в C3, для которых могут назначаться соответствующие правила исключений. Интерфейс задания особых процессов и исключающих правил для них приведен на рис. 4.43.
Рис. 4.45. Интерфейс задания особых процессов и исключающих правил для них

Для заданного (полупустым именем исполняемого файла) особого процесса существует возможность задания следующих исключающих правил (если задано одновременно несколько правил, то они будут действовать одновременно), см. рис. 4.45:

- «Разрешить запись в файлы, отмеченные как исполняемые». Процессу разрешается запись в исполняемые (размещенные СЗ, как исполняемые) файлы — их модификация, удаление, переназначение. При модификации исполняемого файла его размер не меняется;
- «Разрешить исполнение файлов, отмеченных как созданные». Процессу разрешается исполнять создаваемые (размещенные СЗ, как создаваемые) файлы;
- «Маркировать файлы при создании или записи». При создании процессом нового файла, этот файл не размещается, если модифицируется файл, размещенный СЗ, как создаваемый, его исходная разметка не изменяется.

Для того, чтобы процесс, ответственному за автоматическое обновление какого-либо программного средства, разрешить выполнить обновление в полном объеме, для этого процесса требуется задать два правила: установить флаг «Разрешить запись в файлы, отмеченные как исполняемые» и убрать флаг «Маркировать файлы при создании или записи».

Отметим, что данное решение ни в коей мере не снимает противоречия, связанных с автоматическим обновлением программных средств. Оно лишь позволяет реализовывать подобные обновления при соответствующем снижении уровня безопасности.

Возможны и иные технические решения для рассматриваемых приложений средств защиты. Например, не сложно изолировать работу приложений в системе, которым пользователь не доверяет — которые получены не из проверенных источников (или наоборот, которым доверяет) [54]. Для этого достаточно создать пакет, в которую пользователь должен будет инсталлировать подобные приложения. По умолчанию в средстве защиты можно задать разграничительную политику доступа, где в качестве субъекта доступа задать "*" * Пакет* "*", и этому субъекту разрешить доступ (без права исполнения) к объектам, только созданным им же (как к объектам файловой системы, так и к буферу обмена).

Как видно на примере реализации защиты от вредоносных программ, методы контроля с автоматической разметкой объектов могут эффективно, как в части достижимого результата, так и в части простоты администрирования и эксплуатации реализованных их средств защиты, применяться для реализации защиты личного (домашнего) компьютера. С учетом сказанного, в общем случае можно говорить о соответствующем направлении развития средств защиты информации от несанкционированного доступа — реализация защиты личных (домашних) компьютеров с использованием методов контроля доступа с автоматической разметкой объектов.

4.10. Выводы по четвертой главе

1. Предложены принципы и разработаны методы контроля доступа к создаваемым объектам, позволяющие исключить сущность «объект доступа» из разграничительной политики, что позволяют упростить администрирование системы защиты, снижая угрозу уязвимости администрирования, и позволяет реализовывать корректную разграничительную политику доступа в обтеком случае.

2. Разработаны модели контроля доступа, на которых сформулированы и обоснованы требования к построению безопасной системы, в том числе к назначению правил доступа к создаваемым объектам, которые в обязательном порядке должны выполняться при реализации контроля доступа.

3. Разработаны, запатентованы и апробированы технические решения, реализующие методы контроля доступа к создаваемым объектам — к создаваемым файлам и к данным, временно помещаемым в буфер обмена. Сделан вывод о том, что в отношении контроля доступа к буферу обмена важнейшим объекту доступа, используемому для хранения данных, реализация контроля доступа к создаваемым объектам является именно обоснованным методом защиты.
4. Показано, что методы контроля доступа к статичным (системным) и к создаваемым объектам основаны на реализации кардинально различающихся принципов контроля доступа и ими решаются различные задачи защиты информации, как следствие, сделан вывод о целесообразности их совместного использования при реализации разграниченной политики доступа. При этом методы контроля доступа к создаваемым объектам должны использоваться для защиты обрабатываемых в информационной системе данных, а методы контроля доступа к субъектов к статичным объектам - для защиты системных объектов, в том числе, для формирования режимов обработки информации субъектами доступа, и для контроля доступа по созданию объектов.

5. Ввиду того, что именно в создаваемых объектах хранится обрабатываемая на компьютере информация, сделан вывод о том, что практический интерес представляет реализация мандатного и сессионного контроля доступа именно к создаваемым объектам. Именно метод сессионного контроля доступа к создаваемым объектам, ввиду своих преимуществ, может служить основой реализации сессионной модели контроля доступа.

6. С учетом того, что применение методов контроля доступа к создаваемыми объектам во многом меняет собственно подход к построению защиты, в которой ключевым элементом разграниченной политики доступа становится не объект, а субъект доступа - объект из нее исключается, в частности, принципиально меняются подходы к решению многих задач защиты данных - гарантированного удаления файлов, шифрования файлов и др., сделан вывод о том, что методы контроля доступа к создаваемым объектам образуют новую технологию защиты данных. При этом сохранение учетной информации субъектов при автоматическом размещении создаваемых файлов непосредственно в файле позволяет реализовывать технологию защиты данных в распределенной информационной системе.

7. На примерах показано, что метод диспетчерского контроля доступа к создаваемым объектам может эффективно использоваться для решения задач защиты от наиболее актуальных современных угроз, в частности, для решения задач защиты от вредоносных программ и от угроз сетевых атак на уязвимости ОС и приложений.

8. Предложены принципы, разработаны методы и модель контроля доступа к статичным объектам с использованием их автоматической разметки, которая применяется при реализации контроля доступа к создаваемым объектам. Показано, что использование данных методов позволяет кардинально упростить задачу администрирования системы защиты при реализации контроля доступа к статичным объектам, что промиллистировано примером построения системы защиты от вредоносных программ.

9. Сделан вывод о том, что практическая реализация методов контроля доступа к объектам с использованием их автоматической разметки - контроля доступа и к создаваемым, и к статичным объектам, позволяющих минимизировать сложность администрирования систем защиты информации при реализации эффективной защиты информации, образует новое направление построения и применения систем защиты информации от незаконного доступа - защиту личных (домашних) компьютеров.
Глава 5. МЕТОД КОНТРОЛЯ ДОСУПА К ТИПАМ ОБЪЕКТОВ

5.1. Задачи и метод контроля доступа к типам объектов

Есть несколько причин, по которым целесообразен контроль доступа к типам объектов, будем рассматривать в качестве таких объектов файлы.

Замечание. Контроль к типам объектов в общем случае возможен и в отношении иных объектов. Например, разрешив удаленный доступ к какому-либо вычислительному средству по его адресу, можно дополнительно осуществить контроль доступа к типу объекта - к сетевым службам.

Рассмотрим причины, по которым целесообразно реализовывать контроль доступа к типам файлов.

Во-первых, при расширении канонической матрицы доступа, разрешён доступ одного субъекта к файлам, созданным другим субъектом, как правило, требуется разрешать подобный доступ не ко всем созданным пользователем файлам, а лишь к определенным их типам. Отчасти эту задачу можно решить реализацией соответствующей разграничительной политики для процессов (приложений), но, поскольку, в общем случае, одним и тем же приложением могут создаваться файлы различных типов, разграничить доступ к файлам конкретных типов подобным способом не удастся.

Замечание. При реализации контроля доступа к типам файлов уже можно говорить о реализации управления потоками информации определенных типов.

Во-вторых, это создание файлов пользователями (приложениями). Подчас, им нельзя разрешать создавать исполняемые файлы, скрипты и ActiveX-компоненты (командные файлы). А это все отдельные типы файлов.

На практике, может потребоваться разрешить работу пользователя только с определенными данными (типами файлов) и т.д. Определенные данные (типы файлов) может потребоваться разрешить передавать по сети, сохранять на внешних накопителях и т.д., тому много примеров.
Если говорить об первой причине, то для решения задачи контроля доступа к конкретным файлам (в том числе и к типам файлов) может использоваться сущность «Владение» (о ней мы говорили ранее). При этом пользователь может распространить доступ к созданному им файлу не только по его типу, но и по содержимому — к конкретному файлу, в котором сохранена конкретная информация. Однако для корпоративных приложений, в которых санкционированный пользователь сегодня рассматривается, чуть ли не главным источником угрозы хищения информации, реализация подобной возможности недопустима. Поэтому мы в работе рассматриваем исключительно методы контроля доступа с принудительным управлением потоками информации. В этом случае тип файла уже должен каким-то образом определяться администратором.

Другое дело защита личного компьютера, где пользователь обрабатывает собственную информацию и непосредственно заинтересован в защите ее от хищения, в этом случае можно говорить об использовании сущности «Владение».

Таким образом, задачей, решаемой методом контроля доступа к типам объектов (рассматриваем применительно к файлам), является расширение функциональных возможностей разграничительной политики доступа, реализуемой методами контроля доступа к файлам от статических объектов и контроля доступа к создаваемым объектам. Расширение осуществляется за счет возможности реализации разграничений прав доступа по типам файлов, т.е. за счет расширения (возможности уточнения) сущности "файловый объект доступа" при назначении правил доступа.

Прежде чем перейти к рассмотрению метода контроля доступа, остановимся на вопросе, каким образом идентифицировать тип файла в разграничительной политике доступа.

Каждый тип файлов имеет определенный уникальный формат, соответствующий его спецификации. Таким образом, при доступе к файлу существует возможность анализа формата этого файла. Однако существует две причины, ограничивающие его использование при выполнении возможности. Во-первых, это достаточно существенная дополнительная загрузка вычислительных ресурсов (т.е. требуется анализировать форматы всех файлов, включая системные файлы, к которым разграничиваются права доступа), во-вторых, спецификация далеко не всех типов файлов доступны, некоторые разработчики приложений относят эту информацию к коммерческой тайне.

Однако самой ОС необходимо иметь информацию о типе файла для определения режима его обработки, например, каким приложением открывать тот или иной файл. В ОС Microsoft Windows с этой целью используется специальное поле имени файла — расширение файла. Каждому типу файлов однозначно соответствует некоторое расширение. Это в значительной мере упрощает задачу определения типа файла — его идентификация по расширению.

Воспользуемся этой возможностью при реализации контроля доступа к типам файлов. При этом будем предполагать, что расширение файлов используются для упрощения задачи определения системной тип файла, что не предполагает решения каких-либо задач защиты информации. Пользователь может легко переименовать файл, в том числе переименовывать его расширение. С точки зрения реализации соответствующего контроля доступа к типам файлов это означает изменение объекта доступа, что, естественно, недопустимо. Как следствие, формирование требования именно к реализации права «переименование» файла в части переименования расширения в имени файла, при реализации контроля доступа к типам файлов, идентифицируемых их расширениями, с позиций построения безопасной системы становится ключевым.

Таким образом, будем исходить из того, что диспетчер доступа при анализе запроса доступа будет идентифицировать тип файла по его расширению.

Контроль доступа реализуется на основе матрицы доступа (метки безопасности здесь малоприменимы, ввиду неприменимости в общем случае категорирования файловых объектов по их типам — расширениям). Субъект доступа определяется, как и в других методах контроля, тремя существенными: первичный идентификатор пользователя, полноправное имя процесса, эффективный идентификатор пользователя; объект доступа — расширение файла.

Данный метод контроля доступа может рассматриваться в качестве вполне допустимого метода, реализуемого, наряду с методом контроля доступа к статическим, либо к создаваемым файловым объектам, с целью реализации управления потоками информации определенных типов.

Замечание. Вместе с тем, этот метод может использоваться и как самостоятельный метод защиты, применимый к решению отдельных актуальных задач защиты, привнося при этом новые возможности, что будет продемонстрировано далее.

В зависимости от того, какой метод контроля доступа расширяет контролем к типам файлов, идентификация объекта доступа может быть нужна — проведена детализация типов объектов. Например, при использовании совместно с методом контроля доступа к создаваемым файловым объектам, можно учитывать расширения создаваемых файлов, устанавливая различные права доступа к различным типам файлов [30], см. табл.5.1.
В частности, в разграничительной политике, приведенной в табл. 5.1, интернет-браузером могут загружаться (в том числе, создаваться) и читаться только файлы с расширением «html».

Как видим, достаточно просто (используя дополнительный параметр идентификации объекта доступа — расширение файла, см. табл. 5.1) реализовать расширение канонической матрицы доступа, задавая то, какие субъекты, и какую могут получить доступ к каким объектам, и каким образом созданным иным субъектам доступа.

Рассмотрим другой пример разграничительной политики доступа, представленный в табл. 5.2.

### Таблица 5.2. Матрица доступа к создаваемым файлам

<table>
<thead>
<tr>
<th>№ правила</th>
<th>Разрешение файла</th>
<th>Учетная информация субъекта, создающего объекты доступа к которым разграничена</th>
<th>Учетная информация субъекта, имеющего право доступа к контролируемым созданным объектам, и права доступа субъекта к объекту</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>doc</td>
<td>User1.1, User1</td>
<td>User2, User2(-/4)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>User1.1, User1</td>
<td>User2, User2(-/4)</td>
</tr>
</tbody>
</table>

Как видим в табл. 2, в данной разграничительной политике мы разрешаем пользователю User2 только чтение файлов только с расширением «doc», созданных пользователем User1.

Для статических файловых объектов реализация метода состоит в возможности задания объектов доступа масками, позволяющими контролировать доступ к файловым объектам по их расширениям (см. третью главу), например: 
```
<.*exe>, D:\User1\*.exe и т.д.
```

С помощью табл. 5.2 для данной разграничителейной политики можно представить матрицу доступа к файловым объектам, используемую для реализации разграничительной политики методом контроля доступа с принудительным управлением, как показано ниже: 

![Матрица доступа](image)

В любой момент времени система описывается своим текущим состоянием 

\[ Q = (C, T, O, M, M[C, T]) \]

где матрица содержит наборы прав доступа \( Q \) для каждого субъекта \( C \), объекта \( O \), типа \( T \), а также известное, 'известное' (I), 'неизвестное' (O), удаленье (D).

### 5.2. Модель контроля доступа

#### Требования к построению безопасной системы

Сформулируем требования к построению безопасной системы, для чего рассмотрим матрицу доступа.

Если считать, что множество \( C = \{C_1, \ldots, C_n\} \) и \( T = \{T_1, \ldots, T_m\} \) соответственно множеству линеарно упорядоченных множеств субъектов и типы объектов доступа, \( R = \{p, n, d\} \) множество трех состояний доступа (там, удаление, неизвестно) запись \((c, t, d)\) — идентификатор разрешения для субъекта \( C_i \), типа объекта \( T_j \), и права доступа \( M_{i,j} \). то матрица доступа \( M \) вида 

\[
C_i \rightarrow T_j \rightarrow d:
\]

В которой мостоять доля доступа для субъекта \( C_i \) к типу \( T_j \), и права доступа \( d \). Если \( D \) известно, 'известное' (I), 'неизвестное' (O), удаленье (D).
Сформулируем требования к построению безопасной системы исключительно для рассматриваемых методов, позиционируя их как расширение методов контроля доступа к статичным, либо к энкапсулированным файловым объектам (для которых сформулированы свои требования к построению безопасной системы).

При формулировке требований будем учитывать то, что объект идентифицируется своим расширением, и к файлам, обладающим различными расширениями, могут устанавливаться различные права доступа к объектам.

Лемма 5.1. Система безопасна, при условии, что файл характеризуется своим расширением в системе, в том случае, если локализован перечень расширений, файлы с которыми могут создаваться в системе.

Доказательство. Если данное требование не выполняется, в матрице $M_c$ могут появиться новые ячейки $M[c, TO]$ (при создании пользователем файла с новым по отношению к матрице расширением), для которых не определены права доступа к объекту. Лемма доказана.

Лемма 5.2.1. Система безопасна относительно утечки права R при условии, что предотвращена возможность переименования тех файлов, которые используются в разграничительной политике доступа.

Доказательство. При использовании данного метода контроля доступа априори предполагается назначение различных прав доступа к файлам с различными расширениями (имея в виду внедрение разграничения). Поэтому смена объектом расширения файла может привести к утечке права доступа, например, при переименовании расширения с «doc» на «exe», файлом будет утерян право записи ($w$) и приобретено право исполнения ($x$), если оба этих расширения используются в разграничительной политике доступа, либо утеряно соответствующее право, если переименование произведено в расширение, которое не используется в разграничительной политике доступа (соответственно при разрешительной разграничительной политике доступа). Лемма доказана.

Лемма 5.2.2. Система безопасна относительно утечки права R при условии, что предотвращена возможность переименования расширений в тех расширениях, которые используются в разграничительной политике доступа, из тех расширений, которые в ней не используются.

Доказательство. Смена объектом расширения файла может привести к утечке права доступа, например, при переименовании расширения с «exe», которое не используется в разграничительной политике доступа, на «exe», которое используется в разграничительной политике доступа, файлом с переименованным расширением будет приобретено право исполнения ($x$). Лемма доказана.

Вывод. При реализации контроля доступа по типам файлов, идентифицируемых их расширением, при включении расширения в разграничительную политику доступа, не должна предоставляться возможность как переименование этого расширения в иное, так и, наоборот, любого иного в это расширение.

Как отмечалось, при контроле доступа файл характеризуется не только расширением. Для метода контроля доступа к статичным файловым объектам он также характеризуется папкой, в которой размещается файл, для метода контроля доступа к создаваемым файловым объектам — идентификатором (учетной записью) субъекта доступа, созданного файла. С учетом этого могут быть сформулированы следующие две Леммы (применительно к соответствующему методу контроля доступа).

Лемма 5.3.1. Система безопасна, при условии, что файл характеризуется не только своим расширением в системе, но и папками, в которых данный тип файлов может быть сохранен (использование метода контроля доступа к статичным файловым объектам), в том случае, если для каждой папки в которой могут сохраняться файлы, локализованы перечень расширений, файлы с которыми могут быть сохранены.

Доказательство. В данном случае модель контроля доступа $M_{st}$ принимает следующий вид:

$O_1(\cdot)$, $O_2(\cdot)$, ..., $O_n(\cdot)$

$M_{st} = C_1 \begin{bmatrix}
    r, w, d \\
    r \\
    ... \\
    0 \\
    ... \\
    0 \\
    0 \\
    w \\
    ... \\
    r, w, d
\end{bmatrix}$

где $C_i$ — логическая матрица, представляющая разграничительную политику доступа для каждой папки, в которой могут сохраняться файлы.
Лемма 5.3.2. Система безопасности, при условии, что файл характеризуется не только своим расширением в системе, но и субъектами, которыми данный тип файлов может быть сохранен (используя метод контроля доступа к создаваемым файловым объектам), в том случае, если для каждого субъекта локализован перечень расширений, файлы с которыми им могут быть сохранены.

Доказательство. В данном случае модель контроля доступа $M_n^r$ принимает следующий вид:

$$
M_n^r = 
\begin{bmatrix}
C_1 & C_2 & \cdots & C_{n-1} & C_n \\
C_{TO_1} & TO_2 & \cdots & \cdots & \cdots \\
r & w & d & w & \cdots & \cdots & \cdots \\
C_{r-1} & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & w & \cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix}
$$

При этом в любой момент времени система описывается своим текущим состоянием $Q = (C, O(TO), M), M[C, O(TO)]$ – ячейки матрицы, содержит набор прав доступа, где $O(TO)$ – тип объекта, включающего объекты (папка).

Если данное требование не выполняется, то в матрице $M_n^r$, применительно к объекту $O$, могут появиться новые ячейки $M[C, O(TO)]$ (при создании объекта файла с новым по отношению к нему расширением), для которых не определены права доступа к объекту. Лемма доказана.

Теперь, что касается исполняемых файлов. Ранее была сформулирована и доказана Лемма 3.1, состоящая в следующем: не допустимо разрешение...
5.3. Техническое решение. Примеры реализации защиты от актуальных угроз

Естественно, что контроль доступа к типам файлов следует комплексировать с контролем доступа к статическим файловым объектам с целью получения универсального технического решения. Реализуется метод контроля доступа к типам файлов методом дискретного контроля доступа субъектов к статическим объектам (контролю доступа по созданию файлов и к системным файлам), что обеспечивается возможностью использования масок при создания объектов доступа [14], либо методом контроля доступа к создаваемым объектам (контроль доступа к создаваемым файлам), реализацию которого в данном случае требует включения в разграничительную политику доступа сущности "тип объекта". При этом уже будут разграничиваться права доступа, какие субъекты к созданным каким субъектам в процессе их работы типом объектов (характеризуемых расширениями файлов), какое право доступа имеют.

Рассмотрим далее практическое использование метода контроля доступа к статическим объектам, в части реализации контроля доступа по созданию объектов определенных типов, для защиты от актуальных угроз атак.

Для этого, прежде всего, рассмотрим примеры задания объектов доступа с использованием их расширений. Объект задается маской "*.*", что означает файл с любым расширением из любой папки, маской "*.*exe" — файл с расширением "exe" из любой папки, маской "%WinDir%*.exe" — файл с расширением "exe" из папки, заданной переменной среды окружения %WinDir% (папки Windows загруженной системы) и т.д.

Видим, что возможности достаточно широки — можно разрешать отдельным субъектам доступ (в том числе, создание) к файлам с локализованным набором расширений, в том числе с локализованным набором расширений файлов в отдельных папках.

С учетом задания в разграничительной политике субъекта доступа тремя сущностями, данными методом может быть локализована работа с отдельными типами файлов, как для пользователей, так и для приложений. А это уже позволяет принципиально расширить возможности формирования ре-
5.3.1. Обеспечение замкнутости программной среды

Как отмечалось ранее, основу обеспечения замкнутости программной среды, используемой, в том числе, для защиты от вредоносных программ, составляет локализация разрешенных для исполнения в системе объектов (файлов) с предотвращением любой возможности их несанкционированного удаления и/или модификации.

Рассмотрим, как может решаться эта задача защиты с использованием метода контроля доступа к типам файлов [12, 26].

Сразу оговоримся, что типы расширений файлов известны, например, основные типы исполняемых файлов [67] сведены в табл. 5.3, поэтому задача необходимых при формировании разграничительной политики доступа типов расширений файлов сложности не вызывает.

Таблица 5.3. Основные типы расширений исполняемых файлов

<table>
<thead>
<tr>
<th>Расширение</th>
<th>Описание</th>
</tr>
</thead>
<tbody>
<tr>
<td>.exe</td>
<td>То же, что и .exe, но с вирусом F-Secure</td>
</tr>
<tr>
<td>.acp</td>
<td>Приложение Authorware 6</td>
</tr>
<tr>
<td>.acr</td>
<td>Скрипт ACRobot</td>
</tr>
<tr>
<td>.acro</td>
<td>Максим AutoCAD</td>
</tr>
<tr>
<td>.ahk</td>
<td>Сценарий AutoHotkey</td>
</tr>
<tr>
<td>.air</td>
<td>Установочный пакет Adobe AIR</td>
</tr>
<tr>
<td>.apk</td>
<td>Пакет приложения Android</td>
</tr>
<tr>
<td>.app</td>
<td>Приложение Symbian OS</td>
</tr>
<tr>
<td>.awk</td>
<td>Скрипт AWK</td>
</tr>
<tr>
<td>.bat</td>
<td>Пакетный файл MS-DOS</td>
</tr>
<tr>
<td>.bin</td>
<td>Исполняемый файл Unix</td>
</tr>
<tr>
<td>.cgi</td>
<td>Электронная папка CGI</td>
</tr>
<tr>
<td>.cmd</td>
<td>Пакетный файл Windows</td>
</tr>
<tr>
<td>.com</td>
<td>Приложение MS-DOS</td>
</tr>
<tr>
<td>.csh</td>
<td>Скрипт C Shell</td>
</tr>
<tr>
<td>.cw</td>
<td>Файл Rbot::CWWorm</td>
</tr>
<tr>
<td>.dek</td>
<td>Пакетный файл Eavesdropper</td>
</tr>
<tr>
<td>.dll</td>
<td>Скомпилированная программа EdLog</td>
</tr>
<tr>
<td>.dmg</td>
<td>Сценарий Medical Manager</td>
</tr>
<tr>
<td>.ds</td>
<td>Источник данных TWAIN</td>
</tr>
<tr>
<td>.ebm</td>
<td>Основной макрос EXTRA!</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Расширение</th>
<th>Описание</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ect</td>
<td>Файл компонента SageCRM</td>
</tr>
<tr>
<td>.elf</td>
<td>Исполняемый файл Playstation, игры Nintendo Wii</td>
</tr>
<tr>
<td>.es</td>
<td>Файл сценария SageCRM</td>
</tr>
<tr>
<td>.esh</td>
<td>Расширенный пакетный файл DOS</td>
</tr>
<tr>
<td>.ex</td>
<td>Сжатый исполняемый файл, переименованный исполняемый файл Windows</td>
</tr>
<tr>
<td>.exe4</td>
<td>Скомпилированная программа MetaTrader</td>
</tr>
<tr>
<td>.exe</td>
<td>Приложение Windows</td>
</tr>
<tr>
<td>.exopc</td>
<td>Приложение ExOPC</td>
</tr>
<tr>
<td>.ezs</td>
<td>Протокол пакетного сценария EZ-R</td>
</tr>
<tr>
<td>.ezt</td>
<td>Вредоносный червь EZT</td>
</tr>
<tr>
<td>.fas</td>
<td>Скомпилированный исходный код Virtual ISP</td>
</tr>
<tr>
<td>.fky</td>
<td>Макрос Visual FoxPro</td>
</tr>
<tr>
<td>.fpi</td>
<td>Сценарий EPS Creator</td>
</tr>
<tr>
<td>.frs</td>
<td>Сценарий Flash Renamer</td>
</tr>
<tr>
<td>.fpx</td>
<td>Компиляторный код FoxPro</td>
</tr>
<tr>
<td>.gadget</td>
<td>Гаджет Windows</td>
</tr>
<tr>
<td>.gpe</td>
<td>Видеоигра GP2X</td>
</tr>
<tr>
<td>.hms</td>
<td>Скрипт HostMonitor</td>
</tr>
<tr>
<td>.hta</td>
<td>Исполненный HTML-документ</td>
</tr>
<tr>
<td>.icd</td>
<td>Защищенная программа SafeDisc</td>
</tr>
<tr>
<td>.im</td>
<td>Макрос iMacro</td>
</tr>
<tr>
<td>.inx</td>
<td>Скомпилированный скрипт</td>
</tr>
<tr>
<td>.ipa</td>
<td>Приложение iPhone или iPod Touch</td>
</tr>
<tr>
<td>.ipt</td>
<td>Скрипт установки System Management Server</td>
</tr>
<tr>
<td>.isu</td>
<td>Сценарий удаления программы (InstallShield)</td>
</tr>
<tr>
<td>.jar</td>
<td>Архив Java</td>
</tr>
<tr>
<td>.js</td>
<td>Исполняемый скрипт JScript</td>
</tr>
<tr>
<td>.jse</td>
<td>Закодированный скрипт JScript</td>
</tr>
<tr>
<td>.jsx</td>
<td>Исходный код ExtendScript</td>
</tr>
<tr>
<td>.kdk</td>
<td>Сценарий KxKontakt</td>
</tr>
<tr>
<td>.lo</td>
<td>Компиляторный исходный код Interleaf</td>
</tr>
<tr>
<td>.m3g</td>
<td>Приложение Mobile 3D Graphics</td>
</tr>
<tr>
<td>.mcr</td>
<td>Сценарий 3ds Max</td>
</tr>
<tr>
<td>.mem</td>
<td>Файл макроса</td>
</tr>
<tr>
<td>.mio</td>
<td>Приложение MioEngine</td>
</tr>
</tbody>
</table>
### Расширение | Описание
--- | ---
.mpx | Компилярированная программа-меню FoxPro
.mrc | Сценарий miRC
.ms | Скрипт 3ds Max
.mst | Сценарий модификации Windows SDK
.mxe | Макрос Macro Express
.obs | Сценарий ObjectScript
.paf | Файл-установщик портативных приложений
.paf.exe | Программа PortableApps.com
.pex | Исполняемый файл ProBoard
.pif | Информация о приложении Windows
.plsc | Скрипт Messenger Plus! Live
.pix | Исполняемый файл Perl
.prc | Файл ресурсов Palm
.prg | Исполняемая программа FoxPro
.ps1 | Скрипт Windows PowerShell
.pvd | Скрипт InstallIt!
.pwc | Установочный пакет PictureTaker
.py | Компилярированный файл Python
.pyo | Оптимизированный код Python
.qit | Вирус QIT (backdoor:QIT)
.qpx | Компилярированный запрос к базе данных FoxPro
.rbx | Скомпилированный скрипт Rembo-C
.rgs | Сценарий операций с реестром Windows
.rox | Исполнимый файл отчета Actuate
.rpm | Файл пакетного процесса Real Pac
.rxe | Исполнемая программа Lego Mindstorms NXT
.scm | Сценарий SCAR
.scb | Сценарий Scala
.scr | Скрипт
.script | Скрипт
.sct | Скрипт компонента
.shb | Ярлык документа Windows
.shs | Обработчик объектов-фрагментов Windows
.spr | Файл данных FoxPro
.top | Скомпилированное приложение (Tally)
.tlb | Библиотека OLE

### Расширение | Описание
--- | ---
.tms | Сценарий Telemate
.udf | Пользовательские функции Microsoft Excel
.vb | Скрипт VBScript
.vbe | Кодированный исходный код VBScript
.vbs | Скрипт VBScript
.vbscript | Скрипт Visual Basic
.vdo | Файл с возможным вирусом Heathen
.exe | Исполнимый файл, содержащий вирус
.vlx | Скомпилированный сценарий AutoLISP
.wcm | Макрос WordPerfect
.widget | Виджет Yahoo!
.wiz | Файл мастера Windows (Microsoft Wizard)
.wp | Макрос WordPerfect
.ws | Исполнимый сценарий Windows
.wsf | Файл сценария Windows
.xap | Файл данных Microsoft Silverlight
.xqt | Макрос SuperCalc
.xsys | Сценарий Xplorer


Метод защиты, следуя Леммам 5.4 и 5.5, состоит в следующем [26]:
- для субъекта доступа «Все пользователи» (задается маской "*"+) разрешается читать и исполнить необходимый перечень исполняемых файлов, определяемых их расширениями;
- исполнимые файлы из заданного списка запрещаются модифицировать, переименовывать (назвать), создавать с заданными расширениями новые файлы.

По сути этим формируется список типов файлов, исполнение которых разрешается на компьютере, и предотвращается любая возможность их модификации и исполнения иных файлов.

Пример соответствующей разграничительной политики доступа к исполняемым файлам представлен в табл. 5.4.
### Таблица 5.4. Пример разграничительной политики доступа

<table>
<thead>
<tr>
<th>Объекты файловой системы</th>
<th>Режим доступа</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.exe</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*.config</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*.dll</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*.manifest</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*.drv</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*.fon</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*.tif</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*.sys</td>
<td>+ч-з-и-у-п</td>
</tr>
<tr>
<td>*</td>
<td>+ч-з-и-у-п</td>
</tr>
</tbody>
</table>

Заметим, что последнее правило в табл. 5.4 задает запрет на исполнение любого файла, с расширениями, отличными от перечисленных в таблице выше (используется способ более точно описателя при идентификации объектов) – реализуется разрешительная разграничительная политика доступа.

Еще раз уточним, что крайне важно, устанавливаемый в разграничительной политике доступа запрет на переименование объекта ("-П", см. табл. 5.4) для объекта доступа, задаваемого маской "*" расширение", предполагает, как запрет переименования расширения объекта, для которого задан "-П", см. табл. 5.4, так и, наоборот, запрет переименования в это расширение любого иного расширения. Это ключевое требование к корректности реализации рассматриваемого метода контроля доступа.

Рассмотрим, как просто настроить данную разграничительную политику. Из интерфейса, приведенного на рис. 5.1, создается субъект доступа "*.*" (под который подпадает любой объект). Из интерфейса, приведенного на рис. 5.2, создается объект доступа (например, в соответствии с табл. 5.4). Из интерфейса, приведенного на рис. 5.3, задаются права доступа субъекта к объекту, заданные разграничения отображаются в интерфейсе, представленном на рис. 5.4.
Принципиальным отличием рассматриваемого метода является то, что данный метод позволяет не только предотвращать запуск на компьютере несанкционированных (а в рассматриваемом случае, вредоносных) программ, но и создание в системе подобных программ. Это реализуется следующими простыми настройками.

Вредоносные программы, могут иметь не только те расширения, которые мы разрешили на исполнение (при этом одновременно предотвратив возможность запуска на компьютер программ с данными расширениями), но и некоторые иные: *.com; *.bat; *.cmd; и т.д. Для предотвращения любой возможности установки/инсталляции на защищаемой системе подобных файлов, достаточно запретить возможность их создания (переназначения в них). Для этого для данных файлов следует установить права доступа, приведенные на рис. 5.5 (запретить любой доступ к файлу с запрещенным расширением), добавив к этим соответствующие правила в исходную разграничительную политику доступа, представленную в табл. 5.4.

Как видим, рассмотренный метод контроля доступа к типам файлов имеет принципиально иное по сравнению с рассмотренными ранее методами контроля доступа свойство.

### Рис. 5.4. Отображение заданных правил доступа субъекта к объектам

В результате приведенной настроек, на компьютере станет невозможно выполнить несанкционированную программу.

#### Замечание.
В общем случае для различных субъектов доступа (задаваемых соответствующими тремя сущностями) могут задаваться разные разграничительные политики доступа. В частности, можно предотвратить выполнение объектов при некорректном описании цели пользователя и т.д.

Теперь оценим принципиальное отличие в реализации рассматриваемой задачи защиты методом контроля доступа к типам файлов от ее реализации методом контроля доступа к статическим объектам и методом контроля доступа к создаваемым объектам. Напомним, что двумя последними методами эта задача решается, соответственно, локализацией папок, из которых возможно использование объектов, с предотвращением возможности несанкционированного удаления и/или модификации в них объектов; соответственно, запретом использования создаваемых в процессе работы пользователя файлов, с расширением этой возможности, в части предотвращения несанкционированного удаления и/или модификации статичных файлов, автоматически размещаемых, как исполняемые объекты.

### Рис. 5.5. Иллюстрация запрета создания в системе файла с заданным расширением

Отметим, что разграничительная политика доступа, приведенная в табл. 5.4 имеет существенный недостаток. Для этого обратим внимание на следующее. Еще в 2009 году было опубликовано исследование, в котором McAfee сообщало следующее: «...за 30 дней самозапускающиеся вредоносные ПО заразило более 27 млн файлов. Данное ПО использует особенности...
Windows, позволяющие запускать приложения автоматически, не требуют от пользователя даже клика мышкой для активации программы. Оно наиболее часто распространяется через USB-брелоки и другие устройства для внешнего хранения информации. Количество «накрученных» новостей ПО превышало даже показатели весьма известной черты Conflict on 100%: что делает самозапускающеся вредоносное ПО одним из самых известных в мире [276].

Если обратиться к разграничительной политике доступа, проиллюстрированной в табл. 5.4, то можно сделать вывод о том, что предлагаемая разграничительная политика доступа не защищает от запуска вредоносной программы с внешнего накопителя. Данный задача защиты может быть решена в том случае, если на защищаемом компьютере предоставить возможность запуска программ только с жесткого диска, т.е. в этом случае предотвращается возможность модификации их исполняемых файлов (что невозможно обеспечить на внешних накопителях). Модифицируя разграничительную политику доступа, проиллюстрированную в табл.3.4, в предположении, что исполняемые файлы системы и приложений можно будет запускать только из папок Windows и Program Files загруженной ОС [26]. Данный разграничительный политик доступа приведен в табл. 5.5.

Таблица 5.5. Пример разграничительной политики доступа с предотвращением возможности запуска программ с внешних накопителей

<table>
<thead>
<tr>
<th>Объект файловой системы</th>
<th>Режим доступа</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ProgramFiles%\config</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%ProgramFiles%\dll</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%ProgramFiles%\exe</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%ProgramFiles%\manifest</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%ProgramFiles%\sys</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%windir%\config</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%windir%\dll</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%windir%\exe</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%windir%\font</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%windir%\manifest</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%windir%\sys</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td>%windir%\tff</td>
<td>+ч-з-и-у-л</td>
</tr>
<tr>
<td><em>.</em></td>
<td>+ч-з-и-у-л</td>
</tr>
</tbody>
</table>

Замечание. Пример создания объекта доступа в интерфейсе системы защиты представлен на рис. 5.6.

5.3.2. Защита от нарушения приложений вредоносными свойствами

Ранее мы отмечали, что существует весьма большой класс угроз, связанных с атаками со стороны приложений, приобретающих вредоносные функции в процессе их работы, т.е. атак со стороны как пользователем установленных и используемых на компьютере приложений, в первую очередь, это относится к сетевым приложениям. При этом приложения могут выделять соответствующие функциями в результате прочтения вредоносного командного файла (не являющегося исполняемым — здесь определения файла, а не исполняемого), записанного на компьютер в процессе работы пользователя. К подобным вредоносным файлам, в частности, относятся скрипты и ActiveX-компонеты.

Из исследования [83]: «Из всех вредоносных программ, участие которых в интернет-атаках на пользователей, мы выделили 20 наиболее активных. На них пришлось 87,5% атак в интернете...На втором месте — вредоносные...»
Контроль доступа к компьютерным ресурсам

скрытны, внедряемые злоумышленниками в код вломанных легитимных сайтов с помощью специальных программ...».

Обратимся к табл. 5.3. Видим, что подобные файлы, которые могут характеризоваться вредоносными функциями, одинаково характеризуются своими расширениями, т.к. к ним может быть разграничен доступ субъектов. Используем эту возможность.

В простейшем случае предотвратим возможность создания (модификации) подобных файлов при работе в сети, например, с использованием браузера Internet Explorer. Это решение рассмотрено в [10].

Создание субъекта доступа проиллюстрировано на рис. 5.7, разграничительная политика доступа для созданного субъекта к критичным файлам, характеризуемым своими расширениями, на рис. 5.8.

Рис. 5.7. Иллюстрация создания субъекта доступа

Как видим из рис. 5.7 и рис. 5.8, при заданной разграничительной политике доступа приложение Internet Explorer при работе в сети не сможет (примечание с любыми правами – «первичный» и «эффективный» пользователи должны маской «*») создать новые критичные (потенциально опасные части возможности налогообложения приложения вредоносными свойствами) файла и модифицировать легальные подобные файлы, присутствующие на компьютере.

К слову сказать, это ограничивает пользователя и от ненужной ему баннерной рекламы.

Замечание. Дело предотвращения любой возможности создания опасных файлов, определяемым их расширениями, приведенными на рис. 5.8, представленные на рис. 5.8 правила доступа целесообразно применять к любому субъекту доступа (в этом случае субъект должен идентифицироваться следующим образом «*,*,*»).

Теперь рассмотрим решение задачи защиты от рассматриваемых сетевых атак в общем случае. С этой целью создадим три профиля (используем метод контроля доступа к статичным объектам), см. рис. 5.9 (доступ которых далее будем разграничивать к файловым объектам), для каждого из которых создадим трех субъектов доступа, см. рис. 5.10.

Разграничительную политику будет задавать исходя из того, что доступ в сеть реализуется с использованием приложения Internet Explorer пользователем ADMIN-PC\igor.

Сначала рассмотрим пользователя System (субъект «системный процесс», см. рис. 5.9, рис. 5.10). Данный субъект доступа предполагает запуск приложения с системными правами и обращения к системными же правами к
Глава 5. Метод контроля доступа к типам объектов

Рассмотрим, какие возможности защиты обеспечивает разграничительная политика, приведенная на рис. 5.13. Как видим, разрешается право только чтения и исполнения из трех системных каталогов. Запись, куда-либо защищена, установить вредоносную программу в системные объекты невозможно, модифицировать системные ресурсы невозможно, создать критичные файлы, с целью последующего их прочтения приложении для внесения в него вредоносных функций, невозможно, прочитать данные пользователей невозможно.

Реализация разрешительной разграничительной политики «Все, что не разрешено — явно не указано, то запрещено», при контроле доступа к сети приложения Internet Explorer, обеспечивается настройками правил доступа для профилей «Запрос доступа в сеть» (работа любого пользователя, не указанного явно, с этим приложением), см. рис. 5.14. Для всех субъектов, не указанных явно, какой-либо доступ к файловым объектам будет запрещен. Это защита от несанкционированного доступа к файловым объектам в обход разграничительной политики, за счет получения прав другого пользователя.

Теперь о контроле доступа в сеть с использованием приложения Internet Explorer пользователем ADMIN-PCIgor (санкционированный режим работы с сетью). Соответствующий субъект доступа «Интернет», см. рис. 5.10, выполнен в профиль «Доступ в сеть», рис. 5.9, для которого будем задавать разграничительную политику. При этом к файловым объектам могут обращаться только субъекты из профилей «Система» и «Доступ в сеть», другие субъекты, использующие приложение Internet Explorer, доступ соответствующего права доступа получить не должны.

Рисунок 5.12. Результаты аудита доступа профиля «Система» к файловым объектам

Однако, воспользовавшись результатами аудита доступа профиля «Система» к файловым объектам, приведенного на рис. 5.12, можем уточнить (делать более "жесткой") разграничительную политику, пример которой приведен на рис. 5.13.

Замечание. Вопросы реализации аудита доступа к файловым объектам в общем случае будут рассмотрены в следующей главе.
Контроль доступа к компьютерным ресурсам

Рис. 5.14. Иллюстрация задания разграничительной политики доступа для профиля «Запрет доступа в сеть»

Разграничительная политика доступа для профиля «Доступ в сеть» представлена на рис. 5.15. Рассмотрим заданные разграничения по порядку. Записывать и читать (не исполнять) соответствующему пользователю смотримым приложением мы разрешили только файлы с расширением «html». Далее маской файлов заданы разграничения, обеспечивающие корректность работы приложения. Разграничениями для каталогов разрешен запуск только санкционированных программ и предотвращена возможность модификации системного диска. Последней записью устанавливается разрешительная разграничительная политика – все, что не подпадает под правила, установленные выше, будет запрещено.

Как видим, в результате установить пользователю вредоносную программу в системные обьекты невозможно, модифицировать системные ресурсы невозможно, создать критичные файлы невозможно, прочитать данные пользователем, созданных инными приложениями, невозможно.

Как и ранее, при рассмотрении других методов контроля доступа, следует акцентировать внимание читателя на том, что это лишь некоторый пример настройки разграничительной политики, иллюстрирующий возможности рассматриваемого метода контроля доступа к типам файлов. Конкретная разграничительная политика создается для решения конкретных задач защиты, другое дело, какие возможности предоставляет реализуемый метод контроля доступа.

Замечание. Как видим из рис. 5.15, на данном примере проиллюстрированы возможности метода контроля доступа к статическим файловым объектам, расширенного контролем доступа к типам файлов (ряд объектов определены не только их расширениями, но и местоположением).

Глава 5. Метод контроля доступа к типам объектов

Рис. 5.15. Иллюстрация задания разграничительной политики доступа для профиля «Доступ в сеть»

А теперь определим, применительно к решению рассмотренной задаче защиты от актуальных угроз, отличия возможностей метода контроля доступа к типам файлов, от рассмотренных ранее методов контроля доступа субъектов к статическим объектам и метода контроля доступа к создаваемым объектам.

Как видим, с одной стороны, последние два метода предполагают, решение задачи защиты в общем виде – защита и от атак с использованием выявляемых ошибок реализации сетевых приложений, и от атак со стороны приложений, наделенных вредоносными свойствами в результате прочтения ими вредоносных командных файлов, в то время, как метод контроля доступа к типам файлов не предполагает защиту от атак, эксплуатирующих выявленные ошибки реализации сетевых приложений.

С другой стороны, если последние два метода направлены на минимизацию риска потерь, в результате реализации рассматриваемых атак – собственно наделение приложения вредоносными свойствами ими не предотвращается, то метод контроля доступа к типам файлов, наоборот, предотвращает
Глава 5. Метод контроля доступа к типам файлов

3. Разработано и апробировано техническое решение, реализующее метод контроля доступа к типам файлов.

4. Показано, что реализация контроля доступа к типам файлов методом дисперсионного контроля доступа субъектам к статичным объектам при использовании в разграничительной политике доступа масок для задания объектов, в том числе, масок ".расширение файла", возможно только в том случае, если права доступа в разграничительной политике присваиваются субъектам доступа, а не, наоборот, в виде атрибутов объектов доступа.

5. На примерах показано, что метод контроля доступа к типам файлов может эффективно использоваться для решения задач защиты от наиболее актуальных современных угроз, в частности для решения задач защиты от вредоносных программ и от угроз сетевых атак, предполагающих наведение приложений вредоносными свойствами в результате прописания (не исполнения) ими командных файлов, что является угрозой возможностью многих современных приложений.

6. С учетом того, что при реализации защиты от актуальных угроз метод контроля доступа к типам файлов обеспечивается реализация принципиально иных, нежели, чем методами контроля доступа субъектов к статичным объектам и контролем доступа к создаваемым объектам, возможностях защиты, в частности им может предотвращаться внесение в систему вредоносных объектов, сказан вывод о том, что применительно к решению задач защиты от актуальных угроз данный метод контроля доступа может рассматриваться в качестве самостоятельного метода защиты информации от несанкционированного доступа.

5.4. Выводы по пятой главе

1. Разработан метод контроля доступа к типам файлов, использование которого позволяет принципиально расширить возможности разграничительной политики доступа, в том числе, реализуемой с целью формирования режимов обработки информации субъектами доступа, за счет возможности задания типов создаваемых, а в общем применяемых субъектами доступа файлов и типов файлов, которыми могут обмениваться субъекты доступа. В результате чего сделан вывод о том, что метод контроля доступа к типам файлов может позиционироваться, как метод, используемый в дополнение к методам контроля доступа субъектов к статичным и к создаваемым объектам, обеспечивая возможность принудительного управления потоками информации определенных типов.

2. Разработаны модели контроля доступа, на которых сформулированы и обоснованы требования к построению безопасной системы (в части предотвращения возникновения соответствующей угрозы безусловной технологической уязвимости), которые в обязательном порядке должны выполняться при реализации контроля доступа. В результате чего сформулировано качественно новое требование к реализации прав доступа «перемещение» применительно к методу контроля доступа к типам файлов, в части переименования расширенных файлов, заключающееся в том, что при запрете переименования расширения какого-либо файла должно запрещаться, как переименование расширения данного файла в иные, так и переименование любого иного расширения (расширение иного файла) в расширение данного файла.
ГЛАВА 6. АУДИТ ДОСТУПА СУБЪЕКТОВ К ОБЪЕКТАМ. ЗАДАЧИ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ И ЗАЩИТЫ ОТ ВТОРЖЕНИЙ

6.1. Классификация задач и методов аудита доступа субъектов к объектам

Аудит событий1 – крайне важная задача обеспечения информационной безопасности, решение которой позволяет протоколировать и анализировать важные с точки зрения безопасности события, происходящие в защищенной информационной системе, в том числе, и применительно к контролю доступа к субъектов к объектам.

Если обратиться к соответствующему нормативному документу то требования к реализации аудита доступа субъектов к объектам (на примере файловых объектов [2]) звучат следующим образом:

- должна осуществляться регистрация запуска (завершения) 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, запустившего программу (процесс, задание);
  - результат запуска (успешный, неуспешный – несанкционированный);
- должна осуществлять регистрация попыток доступа программных средств (программ, процессов, задач, заданий) к защищаемым файлам. В параметрах регистрации указывается:
Инструментальный аудит можно рассматривать в качестве необхо-
dимого компонента системы защиты информации от несанкциони-
рованного доступа при реализации процессной модели контроля до-
стуна (без реализации подобного типа аудита событий задать кор-
ректную разграничительную политику доступа будет крайне затруд-
nительно);  
Функциональный аудит, используется в процессе эксплуатации
защищенной информационной системы для полного протоколиро-
вания всех событий безопасности, применительно к решаемым за-
дачам защиты. Применительно к данному типу аудита имеет смысл
говорить о протоколировании событий, связанных с реализацией
режимов обработки информации субъектами доступа, т.е. с реализа-
цией ролевой и сессионной моделей контроля доступа, а также
процессной в режиме эксплуатации защищенной информационной
системы. Реализация данного типа аудита предполагает возможность прото-
колирования одновременно всех событий с использованием отдель-
ных журналов аудита к каждому типу объекта доступа (защищае-
мому ресурсу). Фильтрация протоколируемых событий должна осу-
ществляться, посредством задания правил аудита. Журнал (журналы) аудита должны предоставляться системой защи-
ты администратору по его запросу (локальному, либо удаленному –
с сервера безопасности), с целью анализа запротоколированных со-
бытий доступа для формирования разграничительной политики до-
ступа субъектов к объектам.

Контроль доступа к компьютерным ресурсам

- дата и время попытки доступа к защищаемому файлу с указанием ее
  результата (успешная, неуспешная, несанкционированная);
- идентификатор субъекта доступа;
- спецификация защищаемого файла.
К как видим, в требованиях не сформулировано собственно задачи аудита, просто перечислено, какие события должны протоколироваться, причем, как видим, должны протоколироваться, как успешные, так и не успешные события. Нет требований к тому, как строить нейдеструмент аудита событий, как организовать доступ администратору к журналам аудита (по запросу, в реальном времени), как реализовывать фильтрацию журналов аудита и т.д.
При этом необходимо понимать, что аудит событий — это очень ресурсоемкая процедура, реализация протоколирования больших для решения необходи-
мых задач защиты событий, может привести к существенному влия-
nию на вычислительный ресурс защищаемого компьютера.

В общем случае, аудит событий можно классифицировать по типам. Можн
о выделить три основных типа аудита событий, применяемых на практике
для решения совершенно различных задач протоколирования событий без-
опасности [11]:
- Инструментальный аудит, используется для формирования разграниченной политики доступа к файловым объектам с це́лью последующей настройки механизмов защиты. Применяется при вводе системы защиты в эксплуатацию. Используется в тех слу-
чаях, когда необходимо проанализировать работу субъектов досту-
па, в первую очередь, системных процессов и приложений, с целью формирования правил доступа субъектов к объектам, с сохранением корректности функционирования системы и приложений, что необхоло-
димо при реализации процессной модели контроля доступа.

Реализация данного типа аудита предполагает возможность прото-
колирования отдельных событий с использованием отдельного жур-
нал аудита любого субъекта к любому объекту доступа (защищае-
ному ресурсу). Фильтрация протоколируемых событий должна осу-
ществляться, посредством задания правил аудита. Требования к фильтрации собственно журналов аудита в данном случае не предъявляется.

Журнал (журналы) аудита должны предоставляться системой защи-
ты администратору по его запросу (локальному, либо удаленному –
с сервера безопасности), с целью анализа запротоколированных со-
бытий доступа для формирования разграничительной политики до-
ступа субъектов к объектам.

Глав 6. Аудит доступа субъектов к объектам

374

Предметом аудита являются события, связанные
6.2. Инструментальный аудит.
Техническое решение

Прежде всего, с учетом назначения инструментального аудита, формулируем основные принципы реализации инструментального аудита.

1. Инструментальный аудит должен обеспечивать возможность протоколирования, как санкционированных, так и несанкционированных (запрещенных разграничительной политикой доступа) событий.

2. Инструментальный аудит должен обеспечивать протоколирование доступа пользователей ко всем защищенным объектам, для возможности выявления штатных (санкционированных) запросов пользователей доступа, в первую очередь, системных процессов и приложений.

3. Инструментальный аудит должен обеспечивать детальное задание параметров протоколирования событий по объекту доступа, по типу контролируемого доступа (по любым комбинациям этих параметров протоколирования). При этом в журнал аудита должны откладываться только события, соответствующие исходным заданным параметрам протоколирования событий.

4. Субъект доступа при настройке параметров протоколирования должен быть идентифицирован, как и в разграничительной политике доступа должен быть уникальным идентификатор пользователя, полнотестовое имя процесса, эффективный идентификатор пользователя.

5. Субъекты и объекты доступа при настройке параметров протоколирования событий должны иметь возможность задаваться, как своими именами, так и масками, таким образом они должны задаваться также, как и при реализации разграничений прав доступа.

6. При протоколировании событий, как субъекты, так и объекты должны характеризоваться наиболее их точным описанием, применительно к идентификатору, определяемому диспетчером (функции аудита событий, сопоставлений реализуются диспетчером доступа) в запросах доступа.

7. Протоколирование событий должно быть применено ко всем объектам доступа, к которым в системе контролируется доступ субъектов, при этом должно обеспечивать возможность протоколирования событий в отношении любого аутентичного взаимообмена для формирования разграничительной политики доступа к этому объекту.

Рассмотрим соответствующее техническое решение (см. разд. 6.1) в рамках образца.
Контроль доступа к компьютерным ресурсам

об обеспечении корректности его функционирования. Подобное исследования позволяет выявить и критичные с точки зрения безопасности запросы доступа субъектов к объектам, которые, по мнению автора, существуют и необходимы для обеспечения безопасности. В обсуждаемых вопросах речь идет о том, как запросы могут быть превращены в программы, а затем в объекты, с которыми они могут взаимодействовать.

Для реализации первого способа истребования аудита следует задать субъект и объект доступа, разрешить все права доступа субъекта к объекту, и включить аудит доступа по каждому типу права доступа (можно анализировать отдельные права). Здесь также можно использовать маски и переменные среды окружения (например, можно задать к любому объекту, использующему файл с расширением exe: "*.*exe"), а также, как задаются субъекты и объекты доступа при реализации разграничительной политики (технические решения рассмотрены ранее).

Приведем пример анализа работы приложения. Проанализируем, каким ресурсам и с запросом каких прав доступа обращается приложение Internet Explorer, запущенное под интерактивным пользователем, к элементам каталога Windows.

С этой целью соответствующим образом создадим субъект доступа, см. рис. 6.1, и зададим для него правила доступа (разрешаем все права, при этом все обращение поставим на аудит) субъекта к каталогу Windows, см. рис. 6.2. Результаты настройки аудита событий (браузер включен в профиль «Доступ в сеть») отображаются в интерфейсе, см. рис. 6.3.

Рис. 6.4. Журнал аудита доступа к файловым объектам

Замечание. Аналогичные действия можно совершить и в отношении любого иного объекта доступа, пример аудита событий доступа к объектам реестра ОС, проводимого с целью анализа обращений приложения к ветвям и ключам реестра, приведен на рис. 6.5.
Теперь проведем аудит событий доступа к файловым объектам системного процесса, например, winlogon.exe (отвечает за локальный вход пользователя в систему). Он нам интересен тем, что, как отмечалось ранее, предполагает олицетворение системы с учетной записью зарегистрированного пользователя при обращении к файловым объектам. Посмотрим, с использованием аудита событий, как это происходит. Создадим соответствующий субъект доступа, см. рис. 6.6, объект задаем маской "*", см. рис. 6.7. Журнал аудита приведен на рис. 6.8.

Журнал, представленный на рис. 6.8, наглядно показывает, при обращении к каким файловым объектам процессом winlogon.exe происходит олицетворения первичного пользователя "система" с интерактивным пользователем. Подобные запросы доступа (с соответствующими олицетворениями пользователей) для корректной работы системы следует разрешить.

![Рис. 6.5. Журнал аудита доступа к объектам реестра ОС](image)

![Рис. 6.6. Создание субъекта доступа](image)

![Рис. 6.7. Результаты настройки аудита событий](image)

Однако вернемся к журналу, представленному на рис. 6.4. Видим, что браузер, запущенный под пользователем, одновременно открывает некоторые системные файлы из папки Windows на чтение и запись, а некоторые еще в дополнение и на выполнение. Очевидно, что подобную возможность работы приложения с точки зрения обеспечения безопасности следует запретить. При этом необходимо проанализировать то, к чему это может привести, с этой целью может быть использован второй способ инструментального аудита.
Второй способ инструментального аудита может применяться для анализа работы системных процессов и приложений с запретом запрашиваемых ими прав доступа.

Создадим интересующий нас объект доступа (кatalog Windows), для которого будем фиксировать откazy доступа от субъекта доступа «Интернет» (рис. 6.1), см. рис. 6.9.

Предыдущее изложение, что к каталогу Windows для приложений следует разрешать доступ только на чтение и исполнение (тогда они, в результате атаки на уязвимости приложения, не смогут модифицировать системные исполняемые файлы и системные настройки). Настройка соответствующих прав доступа проиллюстрирована на рис. 6.10.

Замечание. Настройка правил аудита событий субъекта к выбранному объекту может задаваться из интерфейса, представленного на рис. 6.10, в том случае, если интерес представляет протоколирование событий отдельных типов, либо из интерфейса, представленного на рис. 6.9, что нам и настроено, если интерес представляет регистрация всех отказов доступа для какого-либо субъекта.

После запуска пользователем браузера в журнале аудита будут фиксироваться отказы в запрашиваемом им доступе к каталогу Windows, см. рис. 6.11.

Заметим, что при таких разграничениях данное приложение корректно работает, однако это может вести к безопасному использованию. Более того, при отсутствии запретов (разрешены соответствующие права доступа), используя интерфейс, представленный на рис. 6.10, влияющие на необходимый нам функционал приложения, и их отмены. При этом, к сожалению, мы можем столкнуться с вопросом выбора – корректность работы приложения, либо безопасность работы с приложением.
возможности (целесообразности) использования данных правил с формируемой разграничительной политике доступа субъекта к объекту.

Отметим, что данная методика универсальна в том смысле, что может использоваться при формировании разграничительной политики доступа любого субъекта к любому объекту, причем, как к статичному, так и к создаваемому.

Замечание. Это подтверждается возможностями настройки правил аудита событий, иллюстрируемыми соответствующими интерфейсами системы защиты информации, используемыми при реализации контроля доступа субъектов к объектам, представленными в предыдущих главах.

6.3. Функциональный аудит. Техническое решение

Функциональный аудит предполагает непрерывное в процессе работы системы протоколирование задаваемых администратором событий, с возможностью их последующего анализа. Анализ (обработка администратором журналов аудита) осуществляется по его запросу, формируемому в рамках соответствующего регламента, либо при появлении соответствующих признаков, например, каким-либо образом выявлен факт утечки конфиденциальной информации (при расследовании инцидента).

Рассмотрим техническое решение [23].

Когда речь заходит о функциональном аудите, то, в первую очередь, актуальной становится задача фильтрации событий при последующем их анализе администратором — это ключевая задача при реализации функционального аудита. Это обусловливается тем, что, несмотря на то, что и в этом случае можно контролировать не все события, происходящие на защищаемом компьютере, и выборочно, применительно к отдельным субъектам и объектам доступа, причем как санкционированные запросы, так и отказы в доступе, что задается настройкой соответствующих правил аудита, например, см. рис. 6.2, рис. 6.3, в общем случае журналы аудита могут иметь большой размер.

Это уступается и тем, что, как правило, систему защиты информации одновременно решается множество задач контроля доступа и аудита событий к разнородным объектам (к различным ресурсам), в рамках решения другой из которой ведет свой аудит — формируется свой журнал аудита событий. Журналы аудита событий администратор может получить удаленно с сервера.
При желании администратор может выбрать иной способ отображения журналов аудита, см. рис. 6.12 б.

Замечание. В окне «Клиенты сервера безопасности», см. рис. 6.12 в, отображается иерархия (баз данных (справочная система) запущенных компьютеров) подключенных клиентов с возможностью контроля их текущего состояния, отображаемого соответствующей иконкой:

- компьютер включен, клиентская часть присутствует, служба запущена;
- компьютер включен, клиентская часть не может пройти аутентификацию на сервере;
- установлено управляемое соединение, запущен интерфейс с сервера безопасности или на локальной машине;
- компьютер включен, клиентская часть отсутствует или служба не запущена.
компьютер выключен.

Естественно, что, с точки зрения, задания протоколируемых событий к функциональному аудиту применимы все те же принципы реализации, что и к инструментальному аудиту.

С учетом назначения и особенностей задач функционального аудита, формируем отличительные принципы реализации функционального аудита.

1. Журналы аудита доступа к различным объектам должны иметь единый формат, что позволяет реализовать единий фильтр событий по общим для них полям (дата/время, пользователь, процесс и др.).

2. В системе должен быть реализован общий фильтр журналов аудита, позволяющий проводить одновременную фильтрацию всех журналов по интересующей совокупности событий (по общим полям в журналах аудита). После применения общего фильтра должны открываться только журналы, содержащие искомые события (подпадающие под заданные правила фильтрации), причем в журналах должны отображаться только искомые события. Это позволяет проводить соконечный "сквозной" анализ запросов доступа.

3. Для каждого отдельно взятого журнала событий должен быть реализован собственный фильтр, позволяющий осуществлять более точную фильтрацию доступа к конкретному объекту доступа, в первую очередь, с возможностью более точного описания при фильтрации событий объекта доступа. Данные фильтры должны отображать для последующего выбора администратором запротоколированные в журналах события по отдельным полям (имена пользователей, присутствующих в журнале аудита, полноту имени процессов, идентификаторы объектов и т.д.). Это позволяет автоматизировать действия администратора по анализу журналов аудита (за счет выбора событий из присутствующих в журнале), и в значительной мере упрощает задачу анализа зарегистрированных событий.

Теперь рассмотрим техническую реализацию. Для взаимного увязывания журналов, возможность их одновременного анализа и просмотра по интересующей совокупности событий, может использоваться общий фильтр зарегистрированных событий, задающий параметры открытия журналов, см. рис. 6.13.

Данный фильтр позволяет задать необходимые параметры фильтрации одновременно для всех журналов аудита при их открытии, примеры приведены на рис. 6.13 - рис. 6.15.
В результате применения данного общего фильтра одновременно открываеться только те журналы, в которых присутствуют записи по отфильтрованным параметрам, причем в журналах отображаются только отфильтрованные записи. В результате администратор получает возможность одновременного просмотра всей регистрируемой совокупности событий, например, все зарегистрированные действия какого-либо и/или процесса за данный промежуток времени.

Рис. 6.16. Фильтр журнала

Рис. 6.17. Отображение полнотекстовых имен процессов, фигурирующих в журнале

Для более точного анализа конкретных журналов, в них присутствуют собственные фильтры. Задача данных фильтров сформировать администратору детальные списки наборов событий (по отдельным параметрам), среди которых он уже сможет сделать выбор необходимых ему для анализа событий. Рассмотрим реализацию подобного фильтра на примере решения задачи аудита доступа к файловым объектам. Фильтр представлен на рис. 6.16. В данном фильтре, при открытии соответствующих вкладок, автоматически отображаются полнотекстовые имена процессов (все, которые присутствуют в анализируемом журнале), см. рис. 6.17, имена пользователей, фигурирующие в журнале, см. рис. 6.18, имена объектов, фигурирующих в журнале, см. рис. 6.19. Это очень упрощает задачу анализа журнала, которая в данном случае предполагает выбор событий из списков. При выборе соответствующих событий, журнал соответствующим образом будет отфильтрован, в нем останется только необходимая администратору информация.

Рис. 6.18. Отображение имен пользователей, фигурирующих в журнале

Рис. 6.19. Отображение имен объектов, фигурирующих в журнале
Как мы ранее отмечали, в рамках контроля доступа к статичным объектам может не только реализовываться разграничивительная политика доступа, но и может контролироваться санкционированность событий, например, каким-либо образом (в обход разграничивительной политики доступа) запущенным процессом.

В отношении этих событий также ведется аудит. Например, журнал аудита контроола запуска и завершения процессов (контроля разрешенных процессов) приведен на рис. 6.20.

Для журналов аудита санкционированных событий (впрочем, как и для журналов протоколирования остальных событий) реализованы фильтры, см. рис. 6.21 – рис. 6.23, по аналогии с тем, как было описано выше.

Рис. 6.20. Журнал аудита контроля запуска и завершения процессов

Рис. 6.21. Фильтр журнала

Рис. 6.22. Отображение полнотуевых имен процессов, фигурирующих в журнале

Рис. 6.23. Отображение имен пользователей, фигурирующих в журнале
6.4. Оперативный аудит. Техническое решение

Вне зависимости от того, насколько эффективно реализована фильтрация событий, функциональный аудит не дает возможности уведомления администратора о произошедших в системе критических событиях, на которые необходимо немедленно — в реальном времени, реагировать. Это обусловлено следующим.

Работа с журналами функционального аудита осуществляется администратором по запросу. Как следствие, в качестве причины обращения к журналам аудита может выступать либо регламент (например, просмотр журналов в день, раз в неделю и т.д.), либо появление каких-либо сторонних причин, потребовавших проведения подобного анализа, например, выявлен каким-либо образом факт хищения информации.

Оперативный же аудит предполагает предложение администратору соответствующей информации о зарегистрированных событиях в реальном времени. Именно это требуется при реализации второй процессной модели контроля доступа.

С учетом назначения и особенностей задач оперативного аудита, сформулируем отличительные принципы реализации оперативного аудита.

1. В сети должен присутствовать специально предназначенный для этих целей компонент системы защиты информации — сервер аудита (отдельная консоль администратора безопасности), на который в реальном времени передаются контролируемые события аудита с защищаемых компьютеров. Активным компонентом во взаимодействии с сервером становится клиентская часть системы защиты информации, устанавливаемая на защищаемом компьютере. Обнаружив соответствующее событие, она немедленно должна передать его на сервер аудита. На консоли администратора безопасности в реальном времени отображаются события, переданные на сервер аудита.

2. В первую очередь, оперативный аудит событий предназначен для протоколирования и уведомления администратора в реальном времени о событиях, связанных с нарушением разграничительной политики доступа, в особенности, как отмечают, это касается атак на угрозы технологических уязвимостей, нивелируемые системой защиты информации от несанкционированного доступа. Как следствие, в первую очередь в данном случае должны протоколироваться отказы в доступе, определяемые администратором при задании прав аудита событий, как критичные. Вместе с тем, в общем случае может возникнуть необходимость информирования администратора и в осуществлении санкционированного доступа к определенным критическим объектам (защитным ресурсам), поэтому настройкой правил аудита должна предоставлять возможность и оперативного аудита санкционированных событий.

3. Для протоколирования событий доступа к различным ресурсам на сервере аудита должен вестись единый журнал аудита.

4. Протоколируемые события к различным объектам (ресурсам) должны иметь единый формат аудита, для возможности отображения событий в едином журнале аудита на сервере аудита. При настройке правил доступа и правил аудита к любому объекту доступа, см. например, рис. 6.2, рис. 6.9, раздельно могут задаваться правила функционального (фиксировать в локальном журнале) и оперативного (фиксировать на сервере аудита) аудита.

Рассмотрим техническое решение по реализации оперативного аудита [23]. Интерфейс сервера аудита приведен на рис. 6.24.

В данном интерфейсе возможны два способа просмотра записей событий аудита, либо отдельно по клиентам, см. рис. 6.25 (вкладка «Клиенты», см. рис. 6.24), либо одновременно по всем клиентам, см. рис. 6.26 (вкладка «Все», см. рис. 6.24).
Контроль доступа к компьютерным ресурсам

Во вкладке «Клиенты» (под клиентом здесь понимается удаленный компьютер, на котором функционирует система защиты информации), см. рис. 6.25, для контролируемого компьютера в реальном времени отображается время события и сообщение, содержащее имя процесса, имя пользователя, режим доступа, объект доступа, пиктограммой отображается вид доступа — санкционированный, либо нет (отказ в доступе, см. рис. 6.27). Существует три состояния подключаемых клиентов:

- поступило новое сообщение от клиента о зарегистрированном событии, администратор ещё его не просмотрел;
- рассматривается журнал оперативного аудита клиента;
- отсутствуют новые (не просмотренные администратором) сообщения о зарегистрированных событиях от клиента.

Естественно, что возможность работы с сервером аудита распространяется на все механизмы защиты, реализующие контроль доступа пользователей к разнородным объектам, а также на механизмы контроля санкционированности событий. Отображение событий осуществляется в едином формате. Например, отображение событий доступа к принтерам на сервере аудита, приведено на рис. 6.28, события контроля запуска и завершения процессов (контроля разрешенных процессов) отображаются на сервере аудита в виде, представленном на рис. 6.29.

Рис. 6.28. Примеры сообщений, поступающих в реальном времени на сервер аудита при протоколировании доступа к принтерам

Рис. 6.29. Примеры сообщений, поступающих в реальном времени на сервер аудита при протоколировании санкционированности событий (разрешенных процессов)

При появлении сообщений об отказе в доступе, выбрав это событие, администратор может использовать соответствующее контекстное меню, используя которое можно в частности удаленно завершить процесс, осуществивший несанкционированный доступ, см. рис. 6.30.
6.5. Обнаружение (обнаружение и предотвращение) вторжений

Сегодня на практике активно развиваются, так называемые, системы обнаружения или обнаружения и предотвращения вторжений COB (IDS — Intrusion Detection Systems) [81], о которых мы вскоре упомянем ранее. При их создании очень используются механизмы контроля, со всеми вытекающими отсюда последствиями, в частности, соответствующим образом анализируются журналы аудита OC и приложений, что является основой построения систем обнаружения вторжений уровня хоста (при построении соответствующих систем уровня сети основу составляет анализ сетевых протоколов, с целью выявления сетевой атаки, заметим, в данном случае обнаруживается не собственно вторжение, а попытка вторжения из сети — атака).

Определение. Под вторжением в информационную систему будем понимать результат атаки, под воздействием которой изменяются какие-либо свойства безопасности защищенной информационной системы.

Определение. Под обнаружением вторжения в информационную систему будем понимать выявление изменений контролируемых системой защиты свойств безопасности защищенной информационной системы.

Система обнаружения вторжений реализуется следующим образом: задаются правила аудита событий для системных и прикладных программных

Средств, в первую очередь, аудита доступа к объектам (может представить себе трудоемкость подобной настройки и влияние аудита на загрузку вычислительного ресурса защищаемого подобным образом компьютера при отсутствии реализации в системе процессной модели контроля доступа, когда можно задать аудит доступа не конкретного процесса к объекту, а только пользователя, т.е. аудит в этом случае будет вестись в отношении всех запускаемых пользователем процессов), правила анализа зарегистрированных средств аудита событий, их задание еще сложнее (возможны поведенческий и эвристический методы анализа журналов аудита, в первом случае, выявляется подозрительное поведение процесса, во втором, предполагающем более глубокий анализ, реализуется попытка выявления атаки — анализируются соответствующие последовательности событий).

В процессе работы информационной системы соответствующими системными и программными средствами по заданым администратором правилам (правилам аудита для соответствующих системных средств и приложений) непрерывно ведется аудит событий, который периодически собирается, фильтруется и по заданным же администратором правилам (но уже применительно к системе защиты информации) анализируется системой защиты информации. При выявлении подозрительных с точки зрения шисты системы защиты событий (при изменении в системе контролируемых свойств безопасности), системный обнаружения вторжений об этом уведомляется администратор, системой обнаружения и предотвращения вторжений, кроме того, формируется некая реакция на предотвращение выявленного события, интерпретируемых системной защиты, как вторжение.

Определение. Под вторжением в информационную систему будем понимать восстановление системой защиты зафиксированных изменений контролируемых свойств безопасности защищенной информационной системы.

Замечание. Именно по причине того, что современные методы обнаружения вторжений базируются на обработке по заданным правилам журналов аудита, мы рассматриваем эту задачу защиты в данной главе.

Ранее мы говорили о том, что любые методы контроля могут использовать лишь в качествене вспомогательных решений — дополнительных методов защиты информации, что обусловливается двумя принципами, во-первых, регулируется уже факт свершившегося события — здесь уже произошедшее вторжение, во-вторых, эффективность защиты в данном случае определяется оперативностью реакции на зарегистрированное событие. Конеч
но же, при рассмотренной реализации системы обнаружения вторжений о какой-либо оперативности реакции на обнаруживаемое вторжение, а уж тем более, о решении задачи защиты в реальном масштабе времени (т.е. об эффективном решении рассматриваемой задачи защиты), говорить не приходится. Однако, не смотря на это, при отсутствии иных подходов к решению данной крайне актуальной современной задачи защиты информации, подобные системы достаточно широко используются на практике (данную актуальную задачу как-то надо решать).

6.6. Защита от вторжений и предотвращение вторжений

Как отмечали ранее, эффективность решения по-существу любой задачи защиты информации от несанкционированного доступа достигается при использовании для её решения методов контроля и разграничения прав доступа субъектов к объектам.

Рассмотрим возможность и оценку эффективности использования рассмотренных в книге методов защиты информации применительно к решению задачи обнаружения (обнаружения и предотвращения) вторжений.

В предложенной в первой главе постановке задачи защиты информации от несанкционированного доступа в общем виде предполагается наделение системой защиты информации от несанкционированного доступа возможностью нивелирования угроз безусловных и условных технологических уязвимостей с целью защиты от актуальных угроз атак. Но именно угрозы безусловных и условных технологических уязвимостей и позволяют реализовать атаку – осуществить вторжение, при отсутствии подобных уязвимостей в системе вторжение в неё невозможно.

Сказанное позволяет нам дать следующее определение.

Определение. Под вторжением в информационную систему будем понимать воздействие на систему, изменяющее какое-либо свойство безопасности защищенной информационной системы, в результате реализации атаки, исполняющей изъявлённую (и не устраненную) в системе безусловную или условную технологическую уязвимость.

На основании данного определения может сделать следующий вывод.

4 Задача технологической защиты информации от несанкционированного доступа.

5 В частности, при реализации процессной модели контроля доступа.
Рассмотрим еще один важный вопрос — постановку задачи предотвращения вторжений при использовании систем защиты информации от несанкционированного доступа с целью защиты от вторжений. Дело в том, что не все угрозы технологических уязвимостей, в первую очередь, речь идет о угрозах условных технологических уязвимостях, могут инвектироваться методами контроля доступа к объектам. Применительно к некоторым из них, например, нарушение приложения вредоносными свойствами в результате выявления ошибки программирования (это уже угроза уязвимости реализации, которая может устраняться только разработчиком соответствующего приложения) решается задача минимизации потерь, потенциально вызванных использованием подобной угрозы уязвимости при реализации атаки. В том числе, на решение данной задачи защиты направлена реализация процессной модели контроля доступа, реализация которой позволяет разграничивать права доступа для критичных процессов. Что позволяет решать рассматриваемую задачу защиты в общем виде в том смысле, что не смотря на способ наложения критического приложения вредоносными свойствами, доступ его к конфиденциальной информации будет запрещен.

Вместе с тем, факт вторжения присутствует — инвектироваются его последствия. В данном случае задача предотвращения вторжения уже становится актуальной, решается она в частиности может задавать в разграничительной политике доступа для заданных правил доступа дополнительной реакции (дополнительной к аудиту соответствующих событий) — завершение процесса, осуществившего вторжение.

Таким образом, при задании правил доступа в объекты для критического процесса, можно установить реакцию — завершение критического процесса, применительно к определенным нарушениям заданных правил доступа, например, если критический процесс попытается исполнить созданный интерактивным пользователем файл. В данном случае вторжение будет предотвращено, при этом системой защиты будет решена и задача защиты от вторжения, в части предотвращения возможности доступа критическому процессу, заданного в результате вторжения, вредоносными свойствами.

Данное техническое решение авторам запатентовано [22].

Всё это относится и к системным процессам. В данном случае для решения задачи защиты от вторжений не требуется анализа сетевых протоколов (что, как отмечали, мало относится к обнаружению вторжений – это скорее выявление сетевых атак). Права доступа системных процессов, в том числе, сетевых служб, могут разграничиваться в рамках реализации процессной модели контроля доступа, в том числе, и с целью защиты от атак со стороны этих процессов, осуществляемых с целью хищения обрабатываемой конфиденциальной информации. Что может реализовываться как методами контроля доступа к статическим, так и методами контроля доступа к создаваемым объектам (эти вопросы были исключены в предыдущих главах). Другое дело, предотвращение вторжения посредством завершения системного процесса. Прежде, чем установить подобную реакцию на нарушение какого-либо задаваемого правила доступа, нужно определить с тем, насколько фатальной для системы будет подобная реакция, сможет ли после этого система нормально запуститься.

Таким образом, как видим, наиболее актуальные задачи защиты, которые сегодня непосредственно формируются из практики эксплуатации современных информационных систем — защита от утечек конфиденциальной информации (от возможности ее хищения санкционированными — легальными, пользователями), защита от вредоносных программ, защита от вторжений, в том числе из внешней сети и т.д., могут эффективно решаться системами защиты информации от несанкционированного доступа, при построении их с использованием рассмотренных в книге решений. Эффективно, поскольку основу их построения составляет реализация разграничительной политики доступа субъектов к объектам, а не контроль на соответствие неким эталонным (соответственно, известным) значениям и списков.

6.7. Основные результаты и выводы

1. Сформулированы задачи аудита событий, которые должны решаться при построении системы защиты информации от несанкционированного доступа в рамках предложенной постановки задачи защиты информации от несанкционированного доступа в общем виде. В результате чего дано обоснование необходимости реализации в системе защиты информации кроме метода функционального аудита событий, используемого для протоколирования событий в рамках реализации ролевой или сессионной моделей контроля доступа (а также процессной в режиме эксплуатации защищенной информационной системы); метода инструментального аудита событий, применяемого в рамках реализации процессной модели контроля доступа с целью формирования корректных правил доступа процессов к объектам, и метода оперативного аудита событий (аудита событий в реальном времени), используемого для протоколирования атак, направленных на реализацию угроз безусловных и условных (в первую очередь) технологических уязвимостей. С учетом того, что рассматриваемыми методами аудита событий решаются совершенно разные задачи, сделан вывод о том, что в современной системе защиты информации от несанкционированного доступа данные методы аудита событий должны быть реализованы в комплексе.
2. Разработаны принципы реализации и технические решения, реализующие методы функционального, инструментального и оперативного аудио-такта событий.

3. Исследованы подходы к решению важнейшей современной задачи защиты информации — задачи обнаружения (обнаружения и предотвращения) вторжений. Обоснованы причины их низкой эффективности, обусловливающие тем, что основу реализации методов обнаружения вторжений составляет использование методов контроля, в данном случае контроля по заданным правилам журналов аудита событий, создаваемых системными и прикладными программными средствами, что сильно загружает вычислительный ресурс и не позволяет оперативно предотвращать обнаруживаемые вторжения в информационную систему.

4. Поскольку вторжения в информационную систему возможны только в результате использования атакой безусловных или условных технологических уязвимостей, сделан вывод о том, что задача нивелирования угроз безусловных и условных технологических уязвимостей, решающая системой защиты информации от несанкционированного доступа при ее постановке в общем виде, по сути, является задачей защиты от вторжений в информационную систему. Как следствие, при решении данной задачи защиты информации от несанкционированного доступа реализуется защита от вторжений в информационную систему, что представляет собой совсем иной уровень эффективности защиты, по сравнению с решением задачи обнаружения (обнаружения и предотвращения) вторжений.

5. Виду того, что некоторые угрозы технологических уязвимостей (в первую очередь, это относится к условным технологическим уязвимостям) не могут быть нивелированы системой защиты информации от несанкционированного доступа6 (системой защиты могут лишь минимизироваться последствия от атак, направленных на реализацию подобных угроз), сделан вывод о целесообразности решения задачи предотвращения вторжений с использованием подобных уязвимостей в рамках решения общей задачи защиты от вторжений в информационную систему. Разработан метод и техническое решение предотвращения вторжений, состоящее в завершении критического процесса, как реакции на обнаружение нарушения соответствующего правила контроля доступа, позволяющего предложить на наделеении критического процесса вредоносными свойствами.

6 Например, угроза наложения приложения вредоносными свойствами с использованием вредоносной в нем ошибки программирования.

ЗАКЛЮЧЕНИЕ

В монографии систематизированы вопросы построения современных систем защиты информационных систем. При этом сделан обоснованный вывод о том, что ключевые задачи защиты сегодня должны решаться системами защиты информации от несанкционированного доступа, приводя соответственно образу построенных систем из соответствующего определения их назначения, решаемых задач и целей построения.

Принимая к вопросам построения систем защиты информации от несанкционированного доступа в работе проведено всестороннее исследование, начиная с определения их задач, решаемых с целью реализации эффективной защиты от современных актуальных угроз атак, исследования недостатков существующих методов защиты, широко используемых в современных системах защиты информации от несанкционированного доступа, заключая изложением и обоснованием предлагаемых методов защиты и реализованных технических решений, формулированием требований к их реализации, выполнение которых направлено на построение безопасных систем.

Основным же полученным результатом авторы считают то, что в работе наконец-то более конкретно, а не в общем виде, как защита от несанкционированного доступа, определено назначение систем защиты информации от несанкционированного доступа при их использовании в современных условиях, как систем, используемых и с целью формирования (при необходимости, разделения) режимов обработки информации субъектами доступа, для чего могут реализовываться ролевая, либо сессионная модели управления доступа, в зависимости от решаемой задачи защиты информации, и с целью нивелирования угроз технологических уязвимостей1, что уже представляет собой принципиально новую постановку задачи защиты для систем защиты информации от несанкционированного доступа (этот задачи должны решаться в комплексе). Определены и подходы к решению этих задач.

В первом случае — это локализация (в первую очередь, по средством управления мониторингом устройств) для субъектов доступа объектов с реализацией разграничения прав доступа субъектов к каждому разрешению для использования в информационной системе объекту доступа. Во втором случае — это выявление и нивелирование технологических угроз уязвимостей, в том числе, условных, возникающих при выявлении в системе угроз уязвимостей реализации, в том числе, соответствующих ошибок программирования. А это уже задачи защиты от современных актуальных угроз атак, которые могут быть осуществлены в информационной системе только в первую очередь, условных.
с использованием выявленных в ней технологических уязвимостей. Вторая задача уже не может решаться без применения в системе процессный модуль контроля доступа, используемый для реализации (разделения) режимов обработки информации, но уже для системных и прикладных процессов. Решение этой задачи защиты — это, наверное, единственный на сегодняшний день осмысленный подход к защите от наиболее актуальных современных угроз атак на уязвимости (ошибки программирования) системных средств и приложений. Без решения тем или иным способом этой задачи защиты безопасную информационную систему не построить, а отсутствие таких решений приводит к появлению и развитию так называемых систем обнаружения вторжений, в то время, как на практике должна реализовываться защита от вторжений (что реализуется нивелированием условных технологических уязвимостей), а не их обнаружение и, в лучшем случае, последующее предотвращение.

Принципиальная важность данных результатов обусловливается тем, что только после четкого определения (формулирования) задач, решаемых системой защиты, становится возможным сформулировать и требования к корректности их решения, в части построения безопасной системы, причем, как собственно к необходимому набору механизмов защиты, реализующих соответствующие методы защиты информации, в составе системы защиты (как, например, можно сформировать режим обработки информации субъекта доступа без реализации в системе механизма управления мониторингом устройств по пользователям), так и к корректности реализации уже непосредственно механизмов защиты информации.

Только после определения задач защиты, решаемых системой защиты информации от несанкционированного доступа, и формулирования требований к корректности их решения, можно оценить эффективность той или иной существующей системы защиты, ее достоинства и недостатки, и вообще целесообразность ее практического использования для защиты современных информационных систем.

Результаты проведенных исследований также наглядно иллюстрируют и необходимость использования в современных условиях принципиально новых подходов к построению систем защиты информации от несанкционированного доступа, как основы решения задач обеспечения безопасности современных информационных систем.

В книге изложены апробированные инновационные (основные технические решения запатентованы) технологии эффективной защиты информации в современных условиях, причем, предназначенные для решения наиболее актуальных современных задач защиты информации. Естественно, что авторы волнуются вопросы их востребованности на практике.

На самом деле, это серьезная проблема, никак не связанная с качеством тех или иных решений. Связана она с недопониманием многих ответственных лиц насущной необходимости серьезного отношения к вопросам информационной безопасности и защиты информации. К сожалению, это не только мнение авторов, сложившееся с учетом их опыта продолжительной работы в этой области знаний. Так, например, уже достаточно давно в обиход вошел термин «Бумажная безопасность», в частности, в [64] отмечается: «В России главная проблема рынка информационной безопасности, равно как и главная тенденция, — бумажная безопасность. К такому выводу пришли участники ежегодной конференции IDC IT Security RoadShow 2011. Однако, разрабатывая инновационные технологии эффективной защиты информационных систем, будем надеяться на скорейшую смену подобных тенденций и взглядов, которая, по понятным причинам, неминуема.

применительно же к вопросам внедрения новых технологий защиты, нельзя не поговорить о квалификации в области информационной безопасности, которая, рано или поздно, будет востребована в полном объеме при решении задач информационной безопасности, и которая, к сожалению, одновременно не появится. Если же говорить о квалификации, то авторы были, как красно понимают то, что, познакомившись с данным материалом, возможно в одночасье стать специалистом в этой сложной области знаний, существует масса фундаментальных и сопутствующих вопросов, затронутых в материалах книги.

Вместе с тем, авторы очень надеются на то, что ознакомление с представленным материалом в любом случае позволит читателю, по крайней мере, понять всю актуальность, глубину и серьезность современных проблем в области информационной безопасности, оценить сложность проектирования и построения эффективных систем защиты информации, их внедрения и эксплуатации, и, что, возможно, самое главное, оценить современные требования к квалификации специалистов в области защиты информации.
СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ Р 50922-96. Защита информации. Основные термины и определения.
15. Щеглов А.Ю., Щеглов К.А. Система контроля доступа к файлам на основе их автоматической разметки // Патент на изобретение № 2524566.
16. Щеглов А.Ю., Щеглов К.А. система контроля доступа к файлам на основе их ручной и автоматической разметки // Патент на изобретение № 2543556.
17. Щеглов А.Ю., Щеглов К.А. Система контроля доступа к шифруемым создаваемым файлам // Патент на изобретение № 2530816.
18. Щеглов А.Ю., Щеглов К.А. Система контроля доступа к ресурсам компьютерной системы с субъектом доступа "пользователь, процесс" // Патент № 2534599.
19. Щеглов А.Ю., Щеглов К.А. Система контроля доступа к ресурсам компьютерной системы с субъектом "исходный пользователь, эффективный пользователь, процесс" // Патент № 2534488.
20. Щеглов А.Ю., Щеглов К.А. Система переформирования объекта в запросе доступа // Патент № 2538918.
21. Щеглов А.Ю., Щеглов К.А. Система разграничения доступа по расширяемым файлов // Патент на изобретение № 2572385.
22. Щеглов А.Ю., Щеглов К.А. Система обнаружения и предотвращения вторжений на основе контроля доступа к ресурсам // Патент № 2543564.
23. Щеглов А.Ю., Павличенко И.П., Корнетов С.В., Щеглов К.А. Комплексная система защиты информации "Панцирь+" для ОС Microsoft Windows // Свидетельство регистрации программы для ЭВМ № 2014660889.
24. Щеглов А.Ю., Павличенко И.П., Щеглов К.А. Система защиты "Панцирь-+" (СЗ "Панцирь-+" ) // Свидетельство регистрации программы для ЭВМ №2013616107.


42. Щеглов К.А., Щеглов А.Ю. Модели и правила мандатного контроля доступа // Вестник компьютерных и информационных технологий. — 2014. — № 5. — С. 44-49.


70. Концептуальные взгляды на деятельность Вооруженных Сил Российской Федерации в информационном пространстве [Электронный ресурс]/ URL: http://www.its2b.ru/blog/infowars/934.html.