МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

А. Ю. Щеглов

МОДЕЛИ, МЕТОДЫ И СРЕДСТВА КОНТРОЛЯ ДОСТУПА К РЕСУРСАМ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Учебное пособие

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург

2014

Щеглов А.Ю. Модели, методы и средства контроля доступа к ресурсам вычислительных систем. Учебное пособие.– СПб: Университет ИТМО, 2014. – 95с.

В учебном пособии исследуются абстрактные модели и методы контроля доступа к защищаемым ресурсам вычислительных систем, реализуемые с целью защиты информации от несанкционированного доступа. Анализируются ключевые недостатки широко используемых методов применительно к реализации контроля доступа в современных условиях. Формулируются и обосновываются требования к построению безопасной системы, выполнение которых позволяет реализовать корректную разграничительную политику доступа к ресурсам и обеспечить эффективную защиту от наиболее актуальных современных угроз. Реализация современных методов контроля доступа, изложенных в учебном пособии, проиллюстрирована примерами реализованных и апробированных технических решений. Материал пособия разбит на 7 разделов, введение и заключение.

Пособие может быть использовано при подготовке магистров по направлениям231000.68 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ», 231100.68 “ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА”, а также инженеров и аспирантов.

Рекомендовано к печати ученым советом факультета Компьютерных технологий и управления, протокол11 ноября 2014 г., протокол № 10.

Оглавление

1. Введение

В учебном пособии исследуютсяи систематизируются широко используемые сегодня на практике модели и методы контроля доступа к ресурсам вычислительных систем применительно к современным условиям. Формулируются и строго обосновываются требования к построению безопасной системы, которые должны быть реализованы в современной разграничительной политике доступа. На основании рассматриваемых моделей, учитывающих архитектурные особенности современных ОС и приложений, формулируются требования к заданию субъектов и правил доступа в современных системах, выявляются ключевые недостатки альтернативных способов задания объектов доступа в разграничительной политике и хранения правил доступа в системе. Рассматриваются современные принципы и методы контроля доступа, реализация которых не только позволяет эффективно решать наиболее актуальные задачи защиты информации, но и кардинально упрощает задачу администрирования и эксплуатации системы контроля доступа, позволяя построить безопасную систему в современных условиях.

Контроль (разграничение прав) доступа к ресурсам применяется для решения ключевых задач защиты информации от несанкционированного доступа в вычислительной системе [2], в том числе является ключевым элементом защиты, направленной на повышение уровня эксплуатационной информационной безопасности современных информационных систем [17].В учебном пособии рассматривается наиболее общий случай реализации разграничительной политики доступа к ресурсам: контроль доступа к файловым объектам. Однако, все излагаемые подходы к реализации контроля доступа к файловым объектам, в той или иной мере, применимы и при реализации контроля доступа к иным ресурсам вычислительной системы, естественно, с учетом специфики субъектов, объектов (ресурсов) и типов доступа. На примерах реализации новых технологий контроля доступа к файловым объектам иллюстрируются возможности построения разграничительных политик для решения наиболее актуальных современных задач защиты информации от несанкционированного доступа.

Широко используемые сегодня на практике модели контроля доступа были разработаны еще в 80-х годах прошлого века. За прошедшее время кардинально изменились как задачи защиты, причем собственно в своей постановке, так и ключевые требования к эффективности средств защиты, что естественно требует пересмотра основополагающих методов и моделей контроля доступа применительно к современным условиям. Именно на этих вопросах и акцентируется внимание в данном учебном пособии.

2. Обобщенная схема и основополагающие методы контроля доступа

Контроль (разграничение в соответствии с заданными правилами) доступа осуществляется центральным элементом системы защиты – диспетчером доступа, перехватывающим и анализирующим все запросы доступа субъектов к объектам. В результате проведенного анализа на соответствие исходно заданным (как правило, администратором) правилам доступа (разграничительной политике доступа)диспетчер предоставляет субъекту запрашиваемый (санкционированный) доступ, либо отказываетв доступе (называется несанкционированным доступом)к объекту.

Поскольку диспетчер доступа должен перехватывать все запросы субъектов к объектам, он, как правило, реализуется на уровне ядра ОС.

Идентифицирующую информацию субъектов и объектов доступа принято называть учетной информацией; правила, на основании которых диспетчер доступа принимает решение о предоставлении (либо отказе) субъекту доступа к объекту, называют правилами (или политикой) разграничения доступа (иногда разграничительной политикой доступа).

Обобщенная схема контроля доступа, представлена на рис.1 [2].

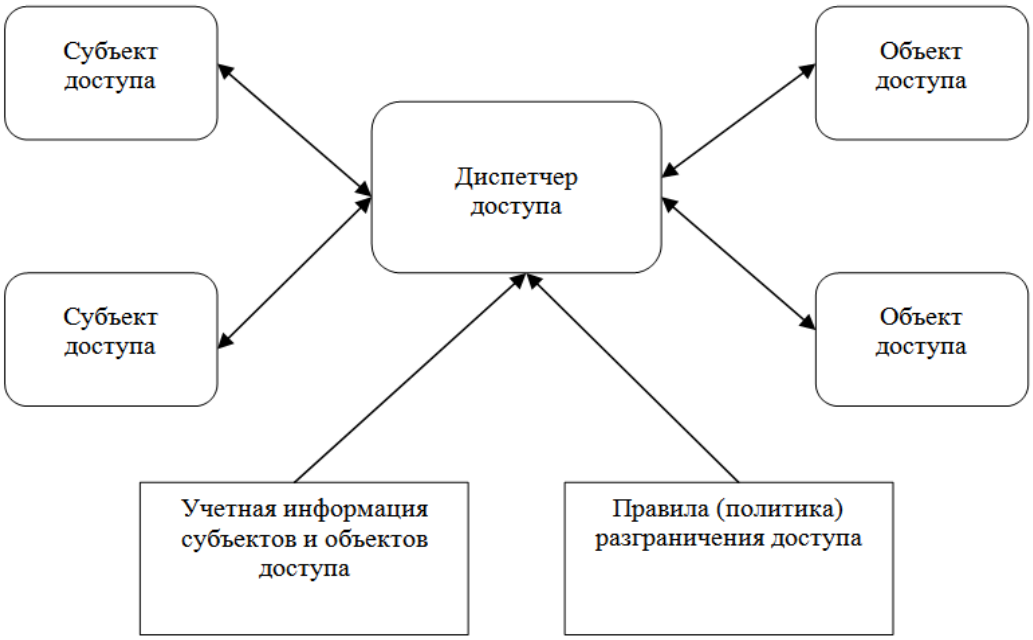


Рисунок 1. Обобщённая схема контроля доступа

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

Рассмотрим и проанализируем основополагающие, широко сегодня используемые на практике методы контроля доступа. Определимся с тем, в чем состоят особенности задач контроля доступа в современных условиях, и какие это накладывает требования к корректности реализации методов контроля доступа в части построения безопасной компьютерной системы.

2.1.Контроль доступа на основе матрицы доступа

Правила (политика) разграничения доступа в общем случае формируются матрицей доступа, в которой указывается, каким субъектам (группам субъектам), к

каким объектам (группам объектов), какие права доступа (чтение, запись, исполнение и т.д.) разрешены либо запрещены.

Метод контроля доступа на основе матрицы доступа еще называют дискреционным.

Для оценки безопасности системы, основанной на реализации дискреционного контроля, на практике используется модель «Харрисона-Руззо-Ульмана» [20]. Если считать, что множества C = {C1,…, Cl} и O = {O1,…, Ok} – 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, содержащая набор прав доступа субъекта из множества C = {C1,…, Cl} к объекту из множестваO = {O1,…, Ok}. В любой момент времени система описывается своим текущим состоянием Q = (C,O,M).

$$M = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & Ок \end{array} \\ \begin{bmatrix} r, w, d & w & & 0 \\ r, w, d & 0 & & \\ \dots & \dots & \dots & \dots \\ 0 & 0 & & r \\ 0 & w & & r, w, d \end{bmatrix} \end{array}$$

Требование к безопасности системы формулируется следующим образом: «Для заданной системы состояние Q0 = (C0,O0,M0) следует считать безопасным относительно некоторого права R, если не существует применимой к Q0 последовательности действий, в результате выполнения которой субъектом C0 приобретается право R доступа к объекту O0, исходно отсутствующее в ячейке матрицы M0[C0,O0]». Если же право R, отсутствующее в ячейке матрицы M0[C0,O0], приобретается субъектом C0, то следует говорить, что произошла утечка права R, а система небезопасна относительно права R.

Замечание. Под правом записи w здесь и далее рассматривается возможность создания нового файла и модификации существующих без возможности удаления. Также можно рассмотреть право добавления (предотвращает возможность модификации существующих файлов) и т.д. Право удаления (d) вынесено в самостоятельное право. На самом деле возможен достаточно высокий уровень детализации прав при моделировании. С целью же упрощения и повышения наглядности получаемых результатов здесь и далее рассматриваем основные права доступа, определяемые множеством R = {w,r,x,d}.

Замечание. Определенный интерес представляет транспонирование матрицы M, в результате которого получаем матрицу Mt:

$$Mt = \begin{array}{c} \\ O1 \\ O2 \\ \dots \\ Ок \end{array} \begin{array}{c} \begin{array}{cccc} C1 & C2 & \dots & C1\text{-}1 \quad C1 \end{array} \\ \begin{bmatrix} r, w, d & r & 0 & 0 \\ w & r, w, d & 0 & w \\ \dots & \dots & \dots & \dots \\ 0 & 0 & r & r, w, d \end{bmatrix} \end{array}$$

В транспонированной матрице Mt отображаются права доступа к объектам субъектов (в исходной матрице M– права доступа субъектов к объектам). Как видим, и об этом будем говорить далее, возможны альтернативные подходы к формированию

разграничительной политики доступа: назначение прав доступа субъектов к объектам либо назначение прав доступа к объектам субъектам (при технической реализации это альтернативные решения, имеющие свои достоинства и недостатки).

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 собственно матрицу доступа), то о безопасности системы можно говорить исключительно в предположении, что действия пользователя не приведут к утечке права R.

Принципиальным для дискреционного метода контроля доступа является реализуемый в нем способ управления потоками информации (или информационными потоками). Потоки информации различаются действиями, выполняемыми субъектом над объектом.

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

Определение.Под управлением информационными потокамипонимается предоставление субъекту права генерирования потока из множества R = {w,r,x,d}.

Право управления информационными потоками для субъекта С может быть как произвольным, так и принудительным.

Определение. Под произвольным управлением информационными потоками для субъекта Спонимается управление с предоставлением субъекту С 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 С включается в схему администрирования на практике, как правило, в отношении созданных им же объектов.

Определение. Под принудительным управлением информационными потоками для субъекта С понимается управление безпредоставления субъекту С 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 субъектаС 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 дискреционного 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 информации. Однако в современных условиях естественно, что в первую очередь речь идет о корпоративных

приложениях(обрабатывается информация, требующая защиты), где пользователь обрабатывает не собственную, а корпоративную информацию, и понятие «Владелец» (не в технологическом смысле) к нему мало применимо. Нельзя сделать подобное предположение даже с большими оговорками. В частности, в проведенном исследовании [21], проиллюстрированном на рис.2, сделан вывод о том, что наибольшую опасность для компаний сегодня представляют именно собственные сотрудники (санкционированные пользователи), которых еще называют инсайдерами.

Рисунок 2. Наиболее актуальные ИБ угрозы

Вывод. Первое важнейшее и принципиальное изменение требований к построению защиты в современных условиях состоит в том, что защита должна строиться в предположении, что пользователь, обрабатывающий информацию на защищаемом компьютере, должен рассматриваться в качестве вероятного потенциального злоумышленника, что принципиально меняет подходы к реализации защиты информации, в частности контроля доступа к защищаемым ресурсам.

Замечание. Вообще говоря, само понятие «Владелец» (не в технологическом смысле) некорректно для корпоративных приложений, ведь пользователи на предприятии обрабатывают не собственную информацию, а информацию, принадлежащую предприятию, подчас являющуюся собственностью совсем иных лиц. О каком же тогда владении может идти речь? Отсюда и потенциальный интерес к ее хищению непосредственно обрабатывающим информацию пользователем.

Вывод. Метод дискреционного контроля доступа, основанный на использовании сущности «Владения», не позволяет построить безопасную систему в современных условиях. А построенная подобным образом система небезопасна относительно права R = {w,r,x,d}.

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

Замечание. Далее, говоря о дискреционном контроле доступа, будем подразумевать метод дискреционного контроля доступа с принудительным управлением потоками информации.

Задача обеспечения принудительного управления потоками информации возлагается надиспетчера доступа, разграничивающего права предоставлять право генерирования потока из множества R = {w,r,x,d} субъектам доступа, предоставляя данное право исключительно привилегированным пользователям (администраторам).

В двух словах 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 субъектов 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 субъектов к объектам: «Все, что не запрещено (явно не указано), то разрешено».

Очевидно, что безопасная система должна строиться с использованием разрешительной разграничительной политики доступа, что обеспечивает разграничение прав доступа в отношении всех объектов вычислительной системы, чем предотвращаются каналы несанкционированного обмена (объекты, к которым не разграничиваются права доступа) между субъектами доступа.

Теперь в двух словах о достоинствах и недостатках метода дискреционного контроля доступа. Его достоинства, равно как и недостатки, определяются возможностью точного задания разграничительной политики доступа, поскольку при этом могут устанавливаться конкретные права доступа из множества R между каждой парой субъект-объект.

К существенным недостаткам можно отнести сложность администрирования, причем не только на этапе запуска системы в эксплуатацию, но и при последующей эксплуатации (а сложность администрирования – это также вопросы безопасности системы). В первую очередьэто связано с включением нового либо исключением существующего субъекта или объекта доступа, что может привести к значительному изменению матрицы доступа. Далее более внимательно рассмотрим эту проблему и пути снижения сложности задачи администрирования – одной из актуальнейших задач, решаемых при разработке новых методов контроля доступа.

2.2.Контроль доступа на основе меток безопасности (мандатов)

Контроль доступа на основе меток безопасности (мандатов) называется методом мандатного контроля доступа. Использование метода мандатного контроля доступа призвано упростить задачу администрирования.

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

Разграничение доступа диспетчером реализуется на основе задаваемого правила, определяющего отношение линейного порядка на множестве M, где для любой пары элементов из множества M задается один из типов отношения: {>,<,=}. 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.

2.2.1. Защита от хищения категорированной информации

Наиболее широкое практическое использование мандатного метода нашло применение практики секретного делопроизводства в компьютерной обработке информации. Данная модель контроля доступа получила название «модели Белла-ЛаПадулы» [18]. Основу реализации обработки категорированной информации составляет классификация (категорирование) 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) субъектов в части допуска к информации различных уровней конфиденциальности (категорий).

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 = {C1,…, Cl} и O = {O1,…, Ol}, и тем меньшее значение метки безопасности Mi, i = 1, …, k им присваивается, т.е.M1 <M2 <M3<…<Ml.

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:

- Mc – метка безопасности субъекта (группы субъектов) доступа;
- Mo – метка безопасности объекта (группы объектов) доступа.

Модель Белла-ЛаПадулы, как отмечали, применяется с целью защиты от нарушения конфиденциальности информациииза счет предотвращения возможности понижения ее категории при обработке в вычислительной системе, что обеспечивается реализацией следующих правил:

1. Субъект C имеет доступ к объекту O в режиме «Чтения» в случае, если выполняется условие:Mc<, = Mo.
2. Субъект C имеет доступ к объекту O в режиме «Записи» в случае, если выполняется условие:Mc = Mo.

Иногда также рассматривается возможность записи и при условии Mc>Mo. Однако большинство современных приложений редко используютоткрытие файла только на запись: как правило, при открытии на запись объект одновременно открывается на запись и чтение. Как следствие, возможность записи при условии Mc>Mo сегодня следует рассматривать скорее как теоретическую возможность, понимая, что включение подобной возможности в реальную разграничительную политику доступа приведет к некорректности работы многих приложений.

Таким образом, при запросе доступа субъекта к объекту диспетчер доступа перехватывает запрос субъекта к объекту, из которого определяет метку субъекта, запросившего доступ (Mc), метку объекта, к которому запрошен доступ (Mo), и запрашиваемый тип доступа (запись, чтение). Далее диспетчер доступа сравнивает между собою метки Mc и Mo и на основании рассмотренных правил доступа разрешает субъекту запрошенный им тип доступа к объекту либо отказывает в нем.

Как видим, при данном методе контроля доступа не только предполагается использование дополнительной учетной информации (меток безопасности), но и

принципиально меняется механизм анализа корректности запроса, который в данном случае состоит не в выборе правила из соответствующей ячейки матрицы доступа, а в формальном сравнении числовых значений меток безопасности субъекта и объекта доступа.

Очевидно, что для иллюстрации правил доступа может использоваться матрица доступа Mm (расширенные правила представлены матрицей доступа Mm(p)):

$$M = \begin{array}{c} \\ C1\ (M1) \\ C2\ (M2) \\ \dots \\ C1\text{-}1\ (M1\text{-}1) \\ C1\ (M1) \end{array} \begin{array}{c} \begin{array}{cccc} O1\ (M1) & O2\ (M2) & \dots & O1\ (M1) \end{array} \\ \begin{bmatrix} r, w, d & r & & r \\ 0 & r, w, d & & r \\ \dots & \dots & \dots & \dots \\ 0 & 0 & & r \\ 0 & 0 & & r, w, d \end{bmatrix} \end{array}$$

$$Mm\ (p) = \begin{array}{c} \\ C1\ (M1) \\ C2\ (M2) \\ \dots \\ C1\text{-}1\ (M1\text{-}1) \\ C1\ (M1) \end{array} \begin{array}{c} \begin{array}{cccc} O1\ (M1) & O2\ (M2) & \dots & O1\ (M1) \end{array} \\ \begin{bmatrix} r, w, d & r & & r \\ w & r, w, d & & r \\ \dots & \dots & \dots & \dots \\ w & w & & r \\ w & w & & r, w, d \end{bmatrix} \end{array}$$

Данные правила проиллюстрированы на рис.3.

Таким образом, мандатный контроль доступа позволяет защищать информацию от нарушения конфиденциальности посредством защиты от несанкционированного понижения категории обрабатываемой информации, осуществляемого с целью изменения режима ее обработки.

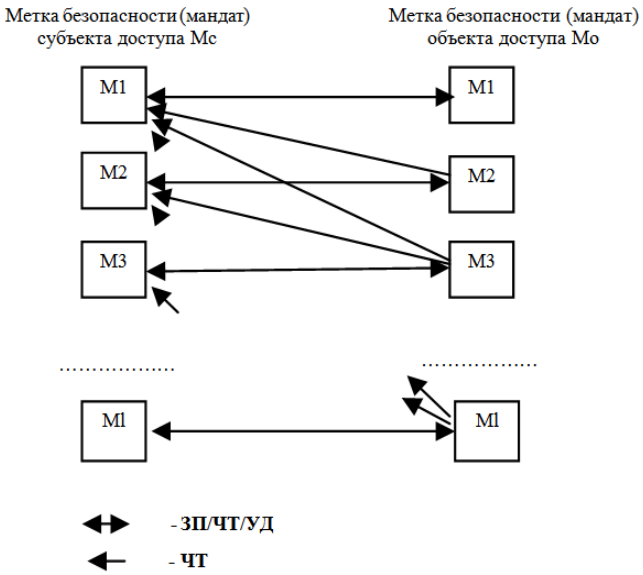


Рисунок 3. Иллюстрация правил доступа, направленных на защиту от понижения категории обрабатываемой информации

Уточним, с какой целью решается задача защиты от понижения категории обрабатываемой информации на практике. Информация различных категорий конфиденциальности обрабатывается совершенно в различных режимах, с использованием различных устройств и объектов. Чем меньше уровень конфиденциальности информации, тем шире предоставляются пользователю возможности по ее обработки: использование неконтролируемых внешних накопителей, неконтролируемых принтеров, неконтролируемых сетевых ресурсов и т.д., и как следствие, тем меньше требования к ее защите и реализуемый уровень безопасности ее обработки.

Так что же собою представляет мандатный метод контроля доступа по сравнению сдискреционным? По сравнению с дискреционным методом с принудительным управлением потоками информации мандатный метод можно рассматривать исключительно как интерфейсное решение [2]. Действительно, что в полной мере иллюстрируют представленные выше матрицы доступа, данные разграничения могут быть реализованы дискреционным методом, причем возможности дискреционного метода куда шире (мандатный можно рассматривать лишь как частный случай). При этом оба метода реализуют принудительное управление потоками информации.

По сравнению же с дискреционным методом с произвольным управлением потоками информации мандатный контроль доступа, конечно, отличается принципиально, а именно способом управления потоками информации. Но подобный метод контроля доступа мы условились далее не рассматривать как не обеспечивающий построение безопасной системы в современных условиях.

Интерфейсное же решение состоит в реализации формализованного способа задания разграничительной политики доступа путем назначения меток безопасности субъектам и объектам при исходно заданных правилах контроля доступа. Диспетчер доступа при этом анализирует права доступа не посредством выбора права R из искомой ячейки матрицы доступа, а путем сравнения по заданным правилам меток безопасности субъекта и объекта.

На первый взгляд, при реализации мандатного контроля доступа задача администрирования принципиально упрощается. Но это лишь в теории. На практике не все так просто. При подобной формализации задания правил разграничительной политики доступа сразу же возникает масса вопросов.А что делать с правом исполнения (x)? Как размечать системные объекты, например, папки Windows, ProgramFiles и др.?Как размечать системные субъекты доступа, например, System и т.д.? Под подобную формализацию (классификацию информации по уровням конфиденциальности) подпадают лишь объекты, предназначенные для хранения непосредственно обрабатываемой в вычислительной системе информации. Эту крайне важную особенность мандатного контроля доступа мы используем далее, когда будем рассматривать методы контроля доступа к создаваемым файлам.

Другим принципиальным недостатком метода мандатного контроля доступа является то, что он позволяет разграничивать права доступа к объектам не между отдельными пользователями, а между группами пользователей, где пользователи группы характеризуются одинаковым классификационным признаком (уровнем доступа к информации), им назначается одинаковое значение метки безопасности. Это приводит к необходимости использования на практике одновременно и мандатного, и дискреционного (для разграничения прав доступа к объектам субъектов, которым присвоена одинаковая метка безопасности)методов контроля доступа.

2.2.2. Защита от нарушения целостности информации

В общем случае защита информации от несанкционированного доступа состоит в обеспечении ее конфиденциальности (защиты от хищения), доступности и целостности (защиты от несанкционированного удаления и модификации информации).

Решение задачи защиты от нарушения целостности информации с использованием мандатного метода контроля доступа предлагается «моделью целостности Биба» [19], суть которой состоит во включении в систему иерархического признака «целостность», отображаемого мандатом или меткой безопасности. В модели Биба вводятся уровни целостности, сопоставляемые с субъектами и объектами доступа, которым в соответствии с заданным уровнем присваивается метка безопасности. Соответствующим образом изменяются и правила контроля доступа, которые здесь приобретают следующий вид:

1. Субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс>, = Мо.
2. Субъект С имеет доступ к объекту О в режиме «Записи» в случае, если выполняется условие: Мс<, = Мо.

В результате получаем полную противоположность (инверсию) «модели Белла-ЛаПадулы». Матрица доступа Mm(w) при этом имеет следующий вид:

$$Mm(w) = \begin{array}{c} \\ C1\ (M1) \\ C2\ (M2) \\ \dots \\ C1\text{-}1\ (M1\text{-}1) \\ C1\ (M1) \end{array} \begin{array}{c} \begin{array}{cccc} O1\ (M1) & O2\ (M2) & \dots & O1\ (M1) \end{array} \\ \begin{bmatrix} r, w, d & w & & w \\ r & r, w, d & & w \\ \dots & \dots & \dots & \dots \\ r & r & & w \\ r & r & & r, w, d \end{bmatrix} \end{array}$$

Сам по себе данный результат (полная противоположность (инверсия) «модели Белла-ЛаПадулы») крайне любопытен, поэтому далее к этому вопросу мы еще вернемся.

Вместе с тем до сих пор данная модель больше рассматривается как некая абстракция. В частности, модель Бибаспециалисты неоднократно критиковали за то, что она использует целостность как некую меру, и ставили под сомнение то, что понятие «большей 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 противоречит возможности категорирования по этому параметру, что, вообще говоря, разумно.

Т.е. не понятен принципиальный вопрос, каким способом и на основании чего проводить классификацию субъектов и объектов по параметру «целостность», что ограничивает практическую ценность данной модели контроли доступа.

3. Правила доступа. Требования к заданию в современных условиях

Ранее нами был сделан крайне важный вывод о том, что в безопасной системе контроля доступа должно осуществляться принудительное для субъекта управление потоками информации (или информационными потоками). Это связано с тем, что пользователь в современных условиях должен рассматриваться в качестве потенциального злоумышленника, который несет в себе угрозу несанкционированного доступа к информации. Это накладывает и соответствующие дополнительные

требования к реализации разграничительной политики (задания правил доступа) для построения безопасной системы.

Сформулируем основополагающие требования к правилам доступа (правилам разграничения доступа субъектов к объектам), реализация которых обеспечивает построение безопасности системы.

Для этого сформулируем основополагающие правила принудительного управления потоками информации, при реализации которых в системе отсутствует утечка права R = {x,w,r,d}, что не приводит к несанкционированному обмену информацией между субъектами (который будем обозначать Cj[P]Cj, j≠j; субъект Ci получает несанкционированный доступ к информации, обрабатываемой субъектом Cj), т.е. реализуется безопасная система.

Сначала рассмотрим метод дискреционного контроля доступа. При этом еще раз отметим, что речь идет о контроле доступа с принудительным управлением потоками информации, где пользователи исключены из схемы администрирования.

Замечание. В данном случае будем считать, что для множеств C = {C1,…, Cl} и O = {O1,…, Ol}, соответственно линейно упорядоченных множеств субъектов и объектов доступа, число субъектов и объектов доступа совпадает.

Лемма 1.Не допустимо разрешение пользователям права исполнения (x) из папок (каталогов), созданных администратором для сохранения пользователями файлов в процессе функционирования системы. При реализации данного требования система безопасна относительно права исполнения (x).

Доказательство.Не допустимо одновременно для одного и того же объекта(папки) предоставлять право на запись, исполнение (w,x) [6,7]. Это обусловлено тем, что пользователь сможет записать и выполнить несанкционированную программу, что приводит к утечке права исполнения (x), предполагающего при принудительном управлении потоками создание программ исключительно администратором. Лемма доказана.

Следствие.К разрешенным правам доступа к создаваемым в процессе работы системы файловым объектам (к папкам, предназначенным для хранения обрабатываемой в компьютерной системе информации) относятся R = {w,r,d}.

Аксиома 1.При назначении разграничительной политики «по умолчанию» должны быть установлены права доступа: Ci(w,r,d)Oj, i=1,…,l, j=1,…,k. Данное правило обусловливает задание диагональной «канонической» матрицы доступа, характеризуемой условием: Ci(w,r,d)Cj, i=1,…,l, j=1,…,k; Ci(0)Cj, j≠j, i=1,…,l, j=1,…,k [8].

Лемма 2.Реализующая каноническую матрицу доступа система безопасна относительно права записи (w) и относительно права чтения (r).

Доказательство.В системе, реализующей каноническую матрицу доступа, невозможно генерирование иных потоков, нежели чем Ci(w,r,d)Ci, i=1,…,l, априори не вызывающих утечку права R = {w,r,d}, т.к. обработка информации субъектами полностью изолирована. Лемма доказана.

Аксиома 2. Контроль доступа в безопасной системе реализован корректно только в том случае, если настройками диспетчера доступа может быть реализована каноническая матрица доступа.

Следствие. Ввиду того, что утечка права R может быть вызвана в результате добавления права R к канонической матрице доступа, далее будем формулировать требования к корректности расширения канонической матрицы доступа правом R.

Лемма 3.При расширении канонической матрицы доступа правом чтения (r): Ci(r)Oj, при уже разрешенном в матрице праве чтения (r): Ck(r)Oi, одновременно с этим должно разрешаться право чтения (r): Ck(r)Oj, где i≠j≠k, i=1,…,l; j=1,…,l; k=1,…,l, что

предотвращает возможность несанкционированного обмена информацией между субъектами Ck[P]Cj из-за образующейся при этом утечки права чтения (r): Ck(r)Oj.

Доказательство.Обратимся к рис.4, где проиллюстрирована утечка права чтения (r), происходящая при невыполнения требования, формулируемого Леммой 3.

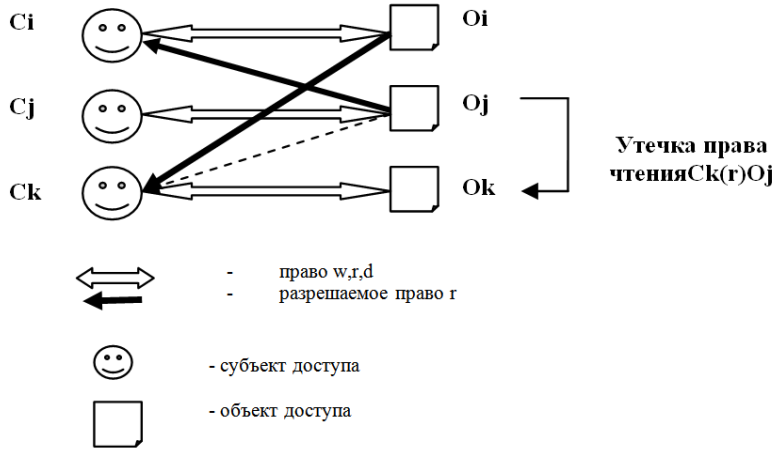


Рисунок 4. Иллюстрация утечки права чтения (r)

Из рис.4 видим, что в случае разрешения права Ci(r)Oj при исходно заданном праве Ck(r)Oic учетом того, что Ci априори обладает правом записиCi(w)Oi, организуется информационный поток чтения от Oj в Ck путем реализации следующей последовательности действий: Ci(r)Oj, Ci(w)Oi, Ck(r)Oi. Если право Ck(r)Oj не разрешено, то присутствует утечка права чтения Ck(r)Oj, как следствие, возможность несанкционированного обмена информацией между субъектами Ck[P]Cj. Из рис.4 также видно, что сформулированное Леммой 3 правило распространяется на все случаи, задаваемые условием: i≠j≠k, i=1,…,l; j=1,…,l; k=1,…,l. Лемма доказана.

Лемма 4. При расширении канонической матрицы доступа правом записи (w): Ci(w)Oj, при уже разрешенном в матрице праве записи (w): Cj(w)Ok, одновременно с этим должно разрешаться право записи (w): Ci(w)Ok, где i≠j≠k, i=1,…,l; j=1,…,l; k=1,…,l, что предотвращает возможность несанкционированного обмена информацией между субъектами Ci[P]Ck из-за образующейся при этом утечки права записи (w): Ci(w)Ok.

Доказательство.Обратимся к рис.5, где проиллюстрирована утечка права записи (w), происходящая при невыполнении требования, формулируемого Леммой 4.

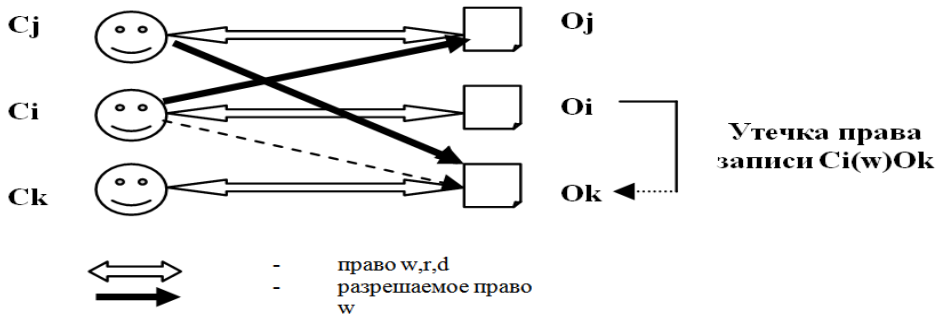


Рисунок 5. Иллюстрация утечки права записи (w)

Из рис.5 видим, что в случае разрешения права Ci(w)Oj при исходно заданном праве Cj(w)Ok с учетом того, что Cj априори обладает правом чтения: Cj(r)Oj, организуется информационный поток записи от Ci в Ok путем реализации следующей последовательности действий: Ci(w)Oj, Cj(r)Oj, Cj(w)Ok. Если право Ci(w)Ok не разрешено, то присутствует утечка права записи Ci(w)Ok, как следствие, возможность несанкционированного обмена информацией между субъектами Ci[P]Ck. Из рис.5 также видно, что сформулированное Леммой 4 правило распространяется на все случаи, задаваемые условием: i≠j≠k, i=1,…,l; j=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 некий усеченный набор базовых прав доступа). Можно рассмотреть два способа записи: «запись» как право, позволяющее, кроме создания нового, модифицировать и удалять существующие файлы; можно рассмотреть право «добавление» – право, позволяющее осуществлять запись новых файлов без возможности модификации и удаления ранее созданных файлов. Естественно, что по своему функциональному назначению и возможностям применения – это совершено различные права доступа.

Замечание. Ранее под правом записи (w) мы рассматривали право модификации существующих файлов и записи новых без возможности удаления существующих (в моделях право записи и право удаления нами были разделены). Исходя из этого, сформулируем требование к безопасной системе. Здесь опять же возникает вопрос: а модификацию следует рассматривать как удаление? Ведь модификация не приводит к удалению объекта, но может привести к нарушению доступности к 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 утечки права удаления (d): при расширении канонической матрицы доступа правом записи (w): Ci(w)Oj, дополнительно должно включаться право удаления (d):Ci(d)Oj.

Рассмотрим пример выполнения требований к корректности правил доступа субъектов к объектам в разграничительной политике, формулируемых Леммой 3 и Леммой 4.

Пусть исходная матрица доступа Мд1 имеет следующий вид:

$$Мд1 = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ C4 \\ C5 \end{array} \begin{array}{c} \begin{array}{ccccc} O1 & O2 & O3 & O4 & O5 \end{array} \\ \begin{bmatrix} r, w, d & 0 & 0 & 0 & 0 \\ 0 & r, w, d & r & 0 & 0 \\ w & 0 & r, w, d & 0 & 0 \\ 0 & 0 & 0 & r, w, d & 0 \\ 0 & 0 & 0 & 0 & r, w, d \end{bmatrix} \end{array}$$

Пусть требуется внести в матрицу доступа Мд1 правило C4(r,w)O3. При этом, выполняя требования Леммы 3 и Леммы 4, получаем следующую матрицу доступа (с учетом внесения дополнительных правил, требуемых данными Леммами), Мд2:

$$Мд2 = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ C4 \\ C5 \end{array} \begin{array}{c} \begin{array}{ccccc} O1 & O2 & O3 & O4 & O5 \end{array} \\ \begin{bmatrix} r, w, d & 0 & 0 & 0 & 0 \\ 0 & r, w, d & r & r & 0 \\ w & 0 & r, w, d & 0 & 0 \\ w & 0 & r, w & r, w, d & 0 \\ 0 & 0 & 0 & 0 & r, w, d \end{bmatrix} \end{array}$$

Выполнимость сформулированных требований необходима при построении разграничительной политики доступа: назначении правил доступа субъектов к объектам при использовании дискреционного метода контроля доступа.

Теперь о мандатном методе контроля доступа. Поскольку здесь правила доступа формализованы (исходно заданы), проверим в отношении данного метода выполнимость требований к корректности заданных правил доступа.

Известные правила, используемые на практике, направленные на защиту от понижения категории обрабатываемой информации, как отмечали, имеют следующий вид:

1. Субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс<, = Мсо.
2. Субъект С имеет доступ к объекту О в режиме «Записи» и «Удаления» в случае, если выполняется условие: Мс = Мсо.

Рассмотрим матрицу контроля доступа для мандатного метода. Правила сравнения меток безопасности, как ранее отмечалось, могут быть отображены в виде матрицы доступа Мм, рассматриваемые права доступа R = {r,w,d}: запись (w), чтение (r), удаление (d):

$$Мм = \begin{array}{c} \\ C1\ (M1) \\ C2\ (M2) \\ \dots \\ C1\text{-}1\ (M1\text{-}1) \\ C1\ (M1) \end{array} \begin{array}{c} \begin{array}{cccc} O1\ (M1) & O2\ (M2) & \dots & O1\ (M1) \end{array} \\ \begin{bmatrix} r, w, d & r & & r \\ 0 & r, w, d & & r \\ \dots & \dots & \dots & \dots \\ 0 & 0 & & r \\ 0 & 0 & & r, w, d \end{bmatrix} \end{array}$$

Оценим корректность правил мандатного контроля доступа, в части возможности построения безопасной системы.

Лемма 5.Система, реализующая мандатный метод контроля доступа, безопасна относительно права записи (w).

Доказательство.Право записи в мандатном контроле доступа к создаваемым файловым объектам задается следующим образом: Ci(w)Oi, i=1,…,l; Ci(w=0)Oj, j≠j, i=1,…,l, j=1,…,l, что априори не может привести к утечке права записи (w). Лемма доказана.

Лемма 6.Система, реализующая мандатный метод контроля доступа, безопасна относительно права чтения (r).

Доказательство. Обратимся к Лемме 3, в которой сформулировано корректное правило управления чтением: «При расширении канонической матрицы доступа правом чтения (r): Ci(r)Oj, при уже разрешенном в матрице праве чтения (r): Ck(r)Oi, одновременно с этим должно разрешаться право чтения (r): Ck(r)Oj, где i≠j≠k, i=1,…,l; j=1,…,l; k=1,…,l, что предотвращает возможность несанкционированного обмена

информацией между субъектами Ck[P]Cj из-за образующейся при этом утечки права чтения (r): Ck(r)Cj». Как следует из представленной выше матрицы Мм, требования, сформулированные Леммой 3, для метода мандатного контроля доступа выполняются. Доказательство Леммы 6 сводится к доказательству Леммы 3. Лемма доказана.

Говоря о мандатном методе контроля доступа, нельзя не затронуть проблемы разметки включающих объектов. Дело в том, что папки, созданные администратором для сохранения пользователями информации различных категорий конфиденциальности, располагаются в некоторых включающих объектах (диск, каталог). Включающие объекты также следует как-то размечать (присваивать им метки безопасности), вопрос – как?

Лемма 7. Иерархическому объекту (папке), включающему в себя размечаемые включающие объекты (папки), при использовании в системе меток безопасности (Mi, i = 1, …, k)должна присваиваться метка безопасности Мвкл, исходя из следующего правила: Мвкл>min{Mi, i = 1, …, k}, при условии, что чем меньше значение метки, тем выше уровень конфиденциальности. В этом случае отсутствует утечка прав записи и чтения.

Доказательство.Иллюстрация корректного правила назначения метки безопасности иерархическим включающим файловым объектам представлена на рис.6.

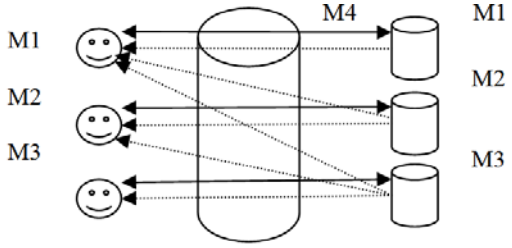


Рисунок 6. Иллюстрация корректного правила назначения метки безопасности иерархическим включающим файловым объектам

Так как при реализации правила:Мвкл>min{Mi, i = 1, …, k}, субъект имеет право только чтения иерархического включающего файлового объекта (может осуществлять «обзор» размеченных 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 иерархическим включающим файловым объектам реализуется основополагающее правило контроля доступа, корректность которого в части утечек прав записи и чтения доказывается Леммой 5 и Леммой 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 иерархическим включающим файловым объектам целесообразно включить иерархическую метку «обзора включающих объектов» Моб, исходя из следующего правила: Моб=Мвкл>min{Mi, i = 1, …, k}, при условии, что чем меньше значение метки, тем выше уровень конфиденциальности. Метка Моб должна присваиваться всем иерархическим включающим файловым объектам (папкам), в которых не требуется создания файлов для хранения конфиденциальной информации.

Следствие. При условии включения в схему контроля доступа иерархической метки «обзора включающих объектов» Моб должны быть реализованы следующие

правила наследование меток безопасности между включающими и включаемыми файловыми объектами (папка):

- «главные» иерархические включающие объекты (логические диски) должны размечаться меткой Моб в обязательном порядке;
- включаемый объект (папка) наследует метку безопасности от включающего объекта (папки) Мвкл=Моб в том случае, если на включаемый объект (папку) не установлена собственная метка безопасности, которая устанавливается из условия:Мвкл<Моб;
- если на включаемый объект (папка) установлена метка безопасности Мвкл, то включаемые в нее объекты (папки и файлы) наследуют данную метку, если на них не установлено собственных меток безопасности.

Лемма 8. Система, реализующая мандатный метод контроля доступа, основанный на наследовании метки безопасности от включающего файлового объекта, безопасна относительно права удаления (d).

Доказательство. Иллюстрация возможных правил контроля удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от включающего файлового объекта, проиллюстрирована на рис.7.

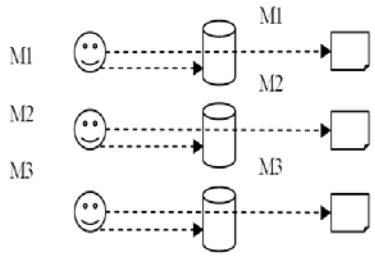


Рисунок 7.Иллюстрация возможных правил удаления для мандатного контроля доступа, основанного на наследовании метки безопасности от включающего файлового объекта

Любой субъект С имеет доступ к объекту О в режиме «Удаления» только в случае, если выполняется условие: Мс = Мо. Это справедливо и в части включающего объекта (папки), который размечается, и в части вложенных объектов (файлов), наследующих метку включающего объекта.

Иллюстрация корректного правила удаления для случая, если иерархическому включающему файловому объекту назначить метку, исходя из условия Мвкл>min{Mi, i = 1, …, k}, приведена на рис.8.

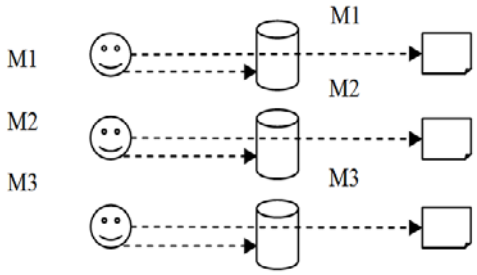


Рисунок 8.Иллюстрация корректного правила удаления при иерархии размеченных файловых объектов

Поскольку ни один субъект не может удалить иерархический объект, ввиду выполнения условия: Мвкл>min{Mi, i = 1, …, k}, правило контроля удаления в системе с иерархическими включающими объектами, представленное на рис.7, сводится к корректному правилу удаления, проиллюстрированному на рис.8. Лемма доказана.

Важнейшим требованием с точки зрения построения безопасной системы является требование, формулируемое Аксиомой 2 следующим образом: «Контроль доступа в безопасной системе реализован корректно только в том случае, если настройками диспетчера доступа может быть реализована каноническая матрица доступа». Иначе не могут быть разделены между пользователями все объекты или любой объект, как следствие, в общем случае не может быть реализована корректная разграничительная политика доступа, особенно это касается категорированных объектов.

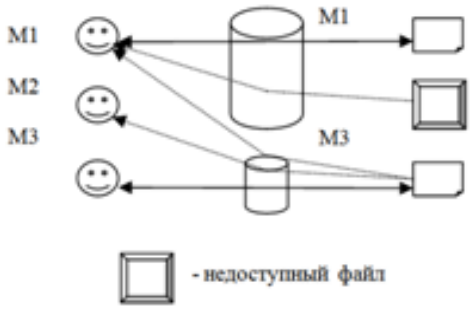
Поясним, с чем связаны проблемы с реализацией данного ключевого требования в современных вычислительных системах.

В современных ОС и приложениях, как правило, присутствуют папки, не предназначенные для их разделения между субъектами доступа.Ярким примером тому служат папки временного хранения файлов «Temp». Чтобы обеспечить нормальную работу системы и таких приложений, папки, в которые помещаются временные файлы, необходимо разрешить для доступа всем субъектам, вне зависимости от назначаемых им прав доступа. И это не единственный пример неразделяемых файловых объектов, называемых объектами общего доступа.

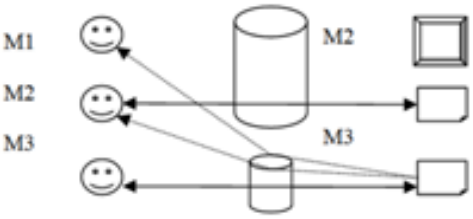
Таким образом, существуют папки, для которых невозможно разграничить доступ пользователям (при этом либо пользователи, которым запрещен доступ к рассматриваемым папкам, не смогут работать с приложением, либо приложение им не сможет предоставляться необходимый сервис). Проиллюстрируем это соответствующей моделью. Если обозначить группу объектов файловой системы (каталогов), не разделяемых системой либо приложением между пользователями, как Он, то каноническая матрица доступа Mk при этом не может быть корректно реализована и примет следующий вид:

$$Mk = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccccc} O1 & O2 & \dots & Oн & \dots & O1 \end{array} \\ \begin{bmatrix} r, w, d & 0 & & r, w, d & & 0 \\ 0 & r, w, d & & r, w, d & & 0 \\ \dots & \dots & \dots & & & \dots \\ 0 & 0 & & r, w, d & & 0 \\ 0 & 0 & & r, w, d & & r, w, d \end{bmatrix} \end{array}$$

Рассмотрим мандатный метод контроля доступа в данных предположениях, который априори должен разграничивать доступ к объектам различных уровней конфиденциальности. Пусть между субъектами, которым соответственно присвоены метки М1 и М2 (обрабатывают информацию различных уровней конфиденциальности), какая-либо папка не может быть разделена (к этой папке требуется обеспечить право записи/чтения для обоих субъектов). Поскольку одному объекту (папке) может быть присвоена только одна метка безопасности (в нашем случае М1 или М2), соответственно получаем возможность реализации одного из двух правил записи/чтения, проиллюстрированных на рис.9.



А) Неразделяемой папке присвоена метка М1



Б) Неразделяемой папке присвоена метка М

Рисунок 9.Иллюстрация возможных правил записи/чтения при наличии в системе неразделяемых объектов

Естественно, что наличие подобных файловых объектов общего или коллективного доступа не позволяет реализовать безопасную обработку в вычислительной системе (на одном компьютере) информации различных уровней (категорий) конфиденциальности.

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, которая состоит в следующем [2]. Для каждого субъекта доступа (к примеру, пользователя) средствами диспетчера доступа для неразделяемого объекта реализуется соответствующий собственный объект, например для каталога «Общий ресурс» заводятся каталоги: «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 каталог «Общий ресурс» данная информация будет перенаправлена диспетчером доступа и сохранена в каталоге «Общий ресурс 1».

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

Замечание.Собственно в исходных неразделяемых системой папках, запрос доступа к которым переадресуется в специально созданные для этого папки, информация сохраняться не будет, т.е. данные папки становятся в системе виртуальными: к ним нет необходимости разграничивать доступ, он будет невозможен из-за переадресации запросов.

Лемма 9.При реализации метода переадресации запросов доступа к объектам файловой системы (папкам), 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 (папкам), не разделяемым системой и приложениями между субъектами доступа, в системе не остается неразделяемых системой и приложениями файловых объектов. К любому файловому объекту, кроме исходно неразделяемых, может быть разграничен доступ между субъектами. Исходно неразделяемый же объект становится виртуальным, к нему не может быть применено ни одно право доступа R. Лемма доказана.

Проиллюстрируем данное решение моделью[2]. Если обозначить группу объектов файловой системы (каталогов), не разделяемых системой между пользователями, как Он, и введенное разделение объекта Он: для субъекта С1 создается объект Он1, С2 – Он2, ..., Cl – Онl, то каноническая матрица доступа Mkc разделенным объектом Он (Mkr) примет следующий вид:

Mkr =	O1	O2	...	Он1	...	Он2	...	Он1	…	O1
C1	r, w, d	0	...	r, w, d	...	0	...	0	...	0
C2	0	r, w, d	...	0	...	r, w, d	...	0	...	0
...	...	...	...	...	...	...	...	...	...	...
C1-1	0	0	...	r, w, d	...	0	...	0	...	0
C1	0	0	...	0	...	0	...	r, w, d	...	r, w, d

Вывод. При реализации в компьютерной системе метода мандатного контроля доступа безопасная система может быть построена только при реализации диспетчером доступа разделения между субъектами неразделяемых ОС и приложениями объектов.

4. Субъекты доступа. Требования к заданию в современных условиях

Очень важным при реализации контроля доступа в современных вычислительных системах является определение сущности субъекта доступа, для которой разграничиваются права доступа к объектам правилами, требования к которым мы сформулировали выше.

В современных системах контроля доступа в качестве субъекта доступа выступает исключительно сущность «пользователь» (корректным для идентификации пользователя в разграничительной политике является использование егоSID), однако, как покажем далее, в современных условиях этого недостаточно.

4.1. Требования к заданию субъекта доступа «пользователь»в разграничительной политике

С точки зрения построения безопасной системы в данном случае возникает вопрос: учетная запись (SID), используемая администратором при задании разграничительной политики, и учетная запись, которую получает диспетчер из запроса доступа, – это одно и то же?

Попытаемся ответить на этот вопрос [2,8].В широко распространенных ОС для идентификации субъектов, выполняющих в системе различные действия, используются идентификаторы защиты (securityidentifiers, 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 (accesstoken). 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 современных ОС, предоставляющее возможность отдельному потоку выполняться в контексте защиты отличном от контекста защиты процесса, т.е. действовать от лица другого пользователя. Олицетворение, в частности, используется в модели программирования «клиент-сервер». При заимствовании прав сервер временно принимает профиль защиты клиента, который обращается к ресурсу. Тогда сервер может работать с ресурсом от имени клиента, а система защиты проводить проверку его прав доступа. Приведенная схема обслуживания клиентского запроса проиллюстрирована на рис.10.

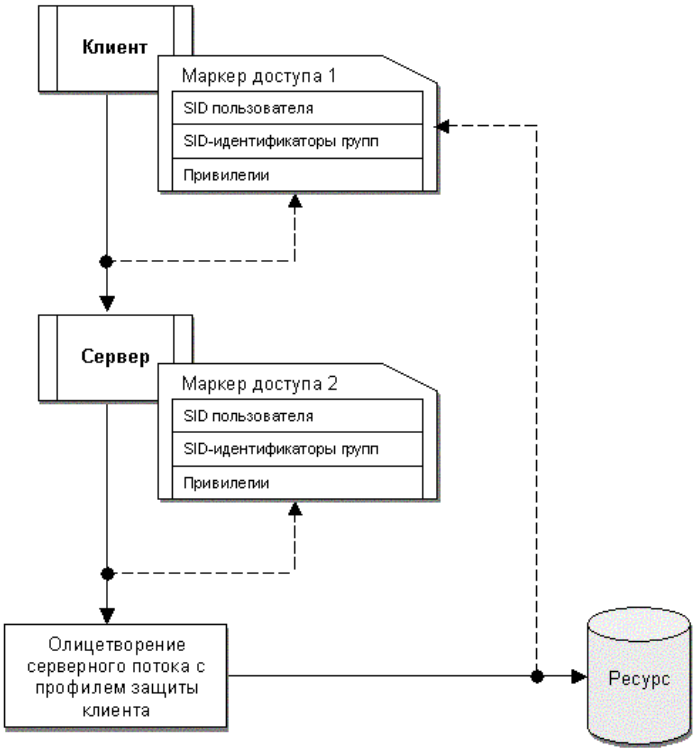


Рисунок 10.Обслуживание клиентского запроса на доступ к ресурсу с использованием олицетворения

Обычно серверу доступен более широкий круг ресурсов, чем клиенту, и при олицетворении сервер может терять часть исходных прав доступа. И, напротив, при

олицетворении сервер может получить дополнительные права. Таким образом, сервисы олицетворения потенциально опасны и могут быть использованы для расширения привилегий (расширения возможностей текущей учетной записи пользователя до возможностей более привилегированной учетной записи, например, суперпользователя, такой как учетная запись администратора или запись SYSTEM), достаточных для осуществления несанкционированного доступа к конфиденциальным данным в обход реализованной разграничительной политики доступа.

Подобная возможность присутствует и в ОС семейства Windows (рассмотрели выше), 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 заимствование прав происходит прозрачно (не требуется дополнительная аутентификация) для пользователя, запустившего процесс.

Как видим, и здесь с использованием сервисов олицетворения, вполне легальных сервисов, предоставляемых современными ОС, возможно получение прав другого пользователя с последующим обращением с его правами к 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.

Для понимания того, насколько актуальная задача контроля сервисов олицетворения (контроля смены идентификатора при доступе к ресурсам) рассмотрим статистику уязвимостей в компонентах ОС, опубликованную, например, в [22], см. рис.11.

Рисунок 11.Статистика уязвимостей ОС

Как видим из этого рисунка, с повышением привилегий связано почти 24% уязвимостей современных ОС, а повышение привилегий состоит в получении прав суперпользователя с последующим доступом к ресурсам в обход разграничительной политики доступа.

Таким образом, можем сформулировать задачу защиты как задачу идентификации и аутентификации пользователей при доступе к объектам, состоящую в обнаружении факта, а при обнаружении — в контроле корректности (на соответствие заданным правилам) смены реального (также, первичного или исходного) имени пользователя на эффективное имя (от лица которого формируется запрос) при запросах доступа к объектам. Рассмотрим решение этой задачи защиты для различных методов контроля доступа.

4.1.1. Дискреционный метод и модель контроля смены имен пользователей при доступе к файловым объектам

Как отмечали, при реализации контроля доступа к файловым объектам мы имеем две сущности, характеризующие пользователя (субъекта 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 диспетчером доступа, перехватывающим и анализирующим все запросы к объектам, а первичное имя для каждого запускаемого процесса – запоминаться диспетчером. Эффективное имя пользователя – это имя пользователя (учетная запись), от лица которого процесс (точнее, генерируемый им поток) непосредственно и обращается к объекту. Это имя диспетчер доступа определяет из запроса доступа к объектам. 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, проводить идентификацию и аутентификацию пользователя при доступе к объектам: сравнивать первичное и эффективное имя, где при совпадении аутентификация будет проведена, т.к. корректность запроса подтверждена; во-вторых, реализовать разграничительную политику доступа в части контроля смены имени пользователя при доступе к объектам.

Построим соответствующую модель [8].

Будем считать, что множество C = {C1,…, Cl}–линейно упорядоченное множество пользователей (учетных записей), зарегистрированных в системе.

Дискреционный методконтроля смены имен пользователей предполагает реализацию разграничительной политики на основе матрицы контроля смены имен пользователейпри доступе к файловым объектам(I), позволяющей максимально точно задать соответствующие правила:

$$I = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} C1 & C2 & \dots & C1 \end{array} \\ \begin{bmatrix} 1 & 1 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 1 \end{bmatrix} \end{array}$$

В любой момент времени система описывается своим текущим состоянием Q = (C,C,I); M[C,C] – ячейка матрицы, содержащая право смены имен пользователей: «1» –

смена имени разрешена, «0» – запрещена. Условимся строками матрицы обозначать первичные, а столбцами – эффективные имена пользователей. Будем обозначать Ci(1)Cjкак разрешение изменения первичного имени пользователя Ci при доступе к файловому объекту на эффективное имя Cj, i=1,…,l; j=1,…,l, i≠j. Запрет доступа будем обозначать следующим образом: Ci(0)Cj. Следует говорить, что доступ к файловому объекту осуществляется без смены имени пользователя в случае Ci(1)Ci, i=1,…,l.

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

Теперь о возможных правилах расширения канонической матрицы.

Ранее мы отмечали, что ключевой задачей защиты при контроле смены имен пользователей является защита от повышения привилегий пользователя при доступе к файловым объектам. Будем считать, что пользователи из множества C = {C1,…, Cl}, зарегистрированные в системе, упорядочены следующим образом: чем меньше порядковый номер пользователя, тем выше его привилегии по доступу к файловым объектам.

Лемма 10. Разрешение изменения первичного имени пользователя на эффективное имя при доступе к файловому объекту: Ci(1)Cj, i≠j, i=1,…,l; j=1,…,l, позволяет построить безопасную систему, при выполнении следующего условия: Ci<Cj.

Доказательство. При выполнении условияCi<Cj разрешается изменение первичного имени пользователя при доступе к файловому объекту Ci(1)Cj только на имя менее привилегированного пользователя. Лемма доказана.

Корректное правило изменения первичного имени пользователя при доступе к файловому объекту описывается матрицей Ik:

$$Ik = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} C1 & C2 & \dots & C1 \end{array} \\ \begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 1 \end{bmatrix} \end{array}$$

Обратим внимание на то, что запрашивает смену идентификатора пользователя при обращении к объектам процесс, т.е. именно процесс в данном случае следует рассматривать в качестве субъекта доступа к сервисам олицетворения. В противном случае, одни и те же правила контроля изменения первичного имени пользователя при доступе к объекту будут распространяться на все процессы, что априори не позволит реализовать эффективную разграничительную политику. Не будем забывать, что процессы, принципиально различающиеся по своим функциям, – это и процессы приложений, и системные процессы. Разграничивая доступ к сервисам олицетворения при доступе к объектам для системных процессов, можно получить достаточно интересные новые свойства защиты. Например, запретив олицетворение при доступе к файловым объектам процессу winlogon (отвечает за локальный вход в систему) из

пользователя System в пользователя User1, мы тем самым запретим вход в систему пользователю User1. Аналогично можно запретить печать для отдельных пользователей, запретив подобное олицетворение для процесса печати. И таких примеров много.

Таким образом, в общем случае дискреционный контроль доступа предполагает реализацию разграничительной политики на основе матрицы контроля смены имен пользователей (I) при доступе к объектам для процессов (права смены имен пользователей задаются соответствующей матрицей I для каждого контролируемого процесса, группы процессов).

4.1.2. Мандатный метод и модель контроля смены имен пользователей при доступе к файловым объектам

Говоря о дискреционном контроле, мы вводили условие, что пользователи из множества C = {C1,…, Cl}, зарегистрированные в системе, упорядочены следующим образом: чем меньше порядковый номер пользователя, тем выше его привилегии по доступу к объектам. При реализации же мандатного контроля доступа рассматривалось следующее условие: чем выше полномочия субъекта и уровень конфиденциальности объекта, тем меньше их порядковый номер в линейно полномочно упорядоченных множествах субъектов и объектов (C = {C1,…, Cl} и O = {O1,…, Ol}) и тем меньшее значение метки безопасности Mi, i = 1, …, l им присваивается (т.е.,M1 <M2 <M3<…<Ml). Естественно, что в данном случае следует защищать систему от атак на повышение привилегий – от возможности получения несанкционированного доступа к информации более высокого уровня конфиденциальности пользователем с меньшим уровнем допуска (за счет смены имени пользователя). Однако напомним, что мандатным контролем доступа реализуется защита от понижения категории обрабатываемой информации, как следствие, в качестве повышения привилегий (от таких атак также следует защищать систему) следует рассматривать и возможность записи в объект меньшего уровня конфиденциальности пользователем с высшим уровнем допуска, опять же, за счет смены имени пользователя.

Данные задачи защиты решаются при реализации следующих непротиворечивых правил мандатного контроля доступа с учетом возможности изменения первичного имени пользователя при доступе к файловому объекту [8]:

1. Субъект C не имеет доступ к объектам файловой системы при условии, что исходному имени субъекта C присвоена метка Mi, а при доступе к файловому объекту осуществлена смена его исходного имени на эффективное, которому присвоена метка Mj, i≠j, i = 1, …, l, j = 1, …, l, причем Mi>Mj.

2. Субъект C имеет следующее право доступа к объектам файловой системы при условии, что исходному имени субъекта C присвоена метка Mi, при доступе к файловому объекту осуществлена смена исходного имени на эффективное, которому присвоена метка Mj, i≠j, i = 1, …, l, j = 1, …, l, при этом Mi<Mj (где под Mc понимается эффективное имя Mj):

- Субъект C имеет доступ к объекту O в режиме «Чтения» в случае, если выполняется условие: Mc<, = Mo.
- Субъект C не имеет доступ к объекту O в режиме «Записи».
- Субъект C имеет следующее право доступа к объектам файловой системы при условии, что изменения имени пользователя не произошло, т.е. выполняется условие: Mi = Mj (где под Mc понимается эффективное имя Mj).

- Субъект С имеет доступ к объекту О в режиме «Чтения» в случае, если выполняется условие: Мс<, = Мо.
- Субъект С имеет доступ к объекту О в режиме «Записи» в случае, если выполняется условие: Мс = Мо.

Лемма 11. Мандатный контроль доступа с учетом возможности изменения первичного имени пользователя при доступе к объекту позволяет построить безопасную систему при условии, если будут реализованы сформулированные непротиворечивые правила для мандатного контроля доступа.

Доказательство.Первое правило предотвращает какой-либо доступ к объектам в случае, если несанкционированно повышен (за счет смены имени пользователя) уровень допуска пользователя (уменьшено значение метки безопасности при доступе к файловому объекту); второе же правило предотвращает доступ к объектам в случае, если несанкционированно понижен его уровень допуска (читать информацию соответствующей категории он при этом может, что не противоречит ключевому правилу мандатного контроля доступа). Ни утечки права записи, ни утечки права чтения не происходит. Лемма доказана.

Замечание. В данном случае получение прав менее привилегированного пользователя на самом деле связано с получением дополнительных привилегий по записи конфиденциальной информации в объект более низкой категории конфиденциальности, т.е. приводит к утечке права записи.

Теперь, как в данном случае должен быть реализован диспетчер доступа. Размечаются субъекты и объекты, им присваиваются метки безопасности. Диспетчером доступа контролируется запуск процесса: диспетчером запоминаются метки безопасности запустивших процессы пользователей Mi. Из запроса доступа диспетчер определяет имя процесса, обращающегося к объекту и эффективное имя пользователя. Эффективное имя пользователя сопоставляется с его меткой Mj. Диспетчер сравнивает значения Mi и Mj, в результате чего выбирает одно из трех корректных непротиворечивых правил (где под Мс понимается эффективное имя Mj). В результате же сравнения значений Мс и Мо,где Мо–определяемое диспетчером из запроса доступа значение метки безопасности объекта, диспетчер либо разрешает запрошенный доступ, либо отказывает в нем субъекту.

4.2.Требования к заданию субъекта доступа «процесс» в разграничительной политике

Использование в разграничительной политике доступа в качестве субъекта доступа исключительно сущности «пользователь» (что сегодня и реализуется на практике) приводит к тому, что все запускаемые пользователем процессы (приложения) имеют равные права доступа к объектам, наследуемые от пользователя. Как следствие, реализация подобной разграничительной политики доступа предполагает, что угрозу несанкционированного доступа к обрабатываемой информации несет в себе исключительно пользователь (права доступа к объектам разграничиваются между пользователями). Однако процесс (приложение) сам по себе уязвим, в том числе и в результате реализации удаленной атаки. Причем различные приложения по различным причинам характеризуются различной степенью уязвимости. Сегодня процесс несет в себе угрозу несанкционированного доступа к обрабатываемой информации ничуть не меньшую, если не большую, чем пользователь [10,13].

Вывод.Сущность «Процесс» следует рассматривать в качестве самостоятельного субъекта доступа, для которого должны устанавливаться права доступа к объектам.

4.2.1. Вероятностная модель контроля доступа процессов к объектам. Защита от атак на уязвимости приложений

Будем считать, что множество C = {C1,…, Cl} – линейно упорядоченное множество субъектов доступа (процессов), идентифицируемых полнопутевыми именами их исполняемых файлов;R = {R1,…, Rm} –конечное множество прав доступа (чтение (r), запись (w), удаление (d), исполнение (x), отсутствие прав доступа (0)) субъекта Ci к объекту Oj из множества O = {O1,…, Ol}. Разрешенное (санкционированное) право доступа субъекта Ci к объекту Oj будем обозначать: Ci(R)Oj (соответственно, Ci(r)Oj разрешено чтение, Ci(w)Oj –запись и т.д., Ci(0)Oj – доступ не разрешен). Разграничительная политика доступа субъектов к объектам описывается матрицей доступа M, где M[C,O] – ячейка матрицы, содержащая набор прав доступа субъекта из множества C = {C1,…, Cl} к объекту из множества O = {O1,…, Ol}. В любой момент времени система описывается своим текущим состоянием Q = (C,O,M).

Замечание.Далее везде, кроме случаев, где это будет оговорено особо, будем считать, что для множеств C = {C1,…, Cl} и O = {O1,…, Ol}, соответственно линейно упорядоченных множеств субъектов и объектов доступа, число субъектов и объектов доступа совпадает.

Аксиома 3.При назначении разграничительной политики «по умолчанию» должны быть заданы права доступа: Ci(w,r,d)Oi, i, i=1,…,l. Данное правило обусловливает задание диагональной «канонической» [2] матрицы доступа процессов к объектам Mk, характеризуемой условием: Ci(w,r,d)Oi; Ci(0)Oj, j≠j, i=1,…,l, j=1,…,l.

$$Mk = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & O1 \end{array} \\ \begin{bmatrix} r,w,d & 0 & \dots & 0 \\ 0 & r,w,d & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & r,w,d \end{bmatrix} \end{array}$$

Следствие. Любая разграничительная политика доступа процессов к объектам должна реализовываться на практике добавлением правил в каноническую матрицу прав доступа Ci(R)Oj, j≠j, i=1,…,l, j=1,…,l.

Введем следующие обозначения. Пусть множество P = {P1,…, Pl}– линейно упорядоченное множество вероятностей обнаружения уязвимостей процессов (приложений) из множества субъектов доступа (процессов) C = {C1,…, Cl}.

Замечание. Чтобы оценить, насколько уязвимы современные системные средства и приложения, достаточно обратиться к соответствующей статистике, например, [23], см. рис.12.

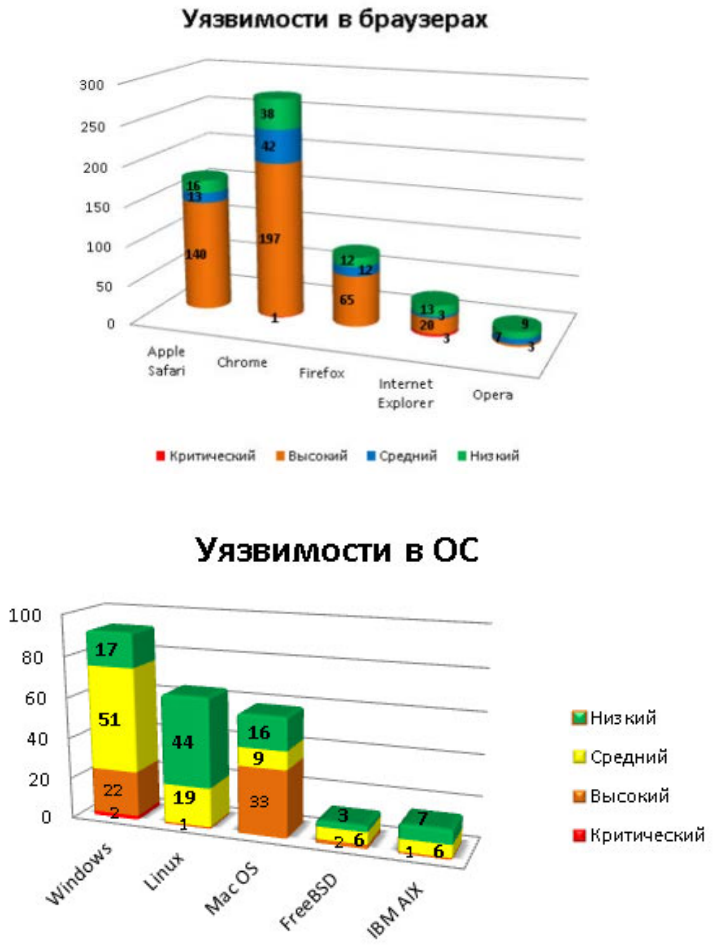


Рисунок 12.Статистика обнаруженных уязвимостей в ОС

и в браузерах за 2011 г.

С учетом того, что эти уязвимости приводят к несанкционированной записи (включим сюда и удаление) w либо к несанкционированному чтению r, будем считать, что вероятность несанкционированной записи субъектом Ci в объект Oi (куда ему разрешены запись и чтение) составляет $P_i(w)$, несанкционированного чтения $-P_i(r)$, при этом $P_i=P_i(w)+P_i(r)$, i=1,...,l [13].

В общем случае запись в один и тот же объект Oj может быть разрешена нескольким субъектам (для чтения аналогично), в пределе всем субъектам C = {C1,…, Cl}, поэтому здесь следует говорить о суммарных вероятностях несанкционированной записи в объект $OjP_jc(w)$, соответственно, чтения из объекта $P_jc(r)$, определяемых следующим образом:

$$P_j c(r) = 1 - \prod_{i=1}^{l} (1 - P_{j-j}(r)), \qquad (1)$$

где: $P_{i-j}(r) = P_i(r)$, еслиCi(r ≠ 0)Oj, j ≠ i;

$P_{i-j}(r) = P_i(r)$, еслиj=i;

$P_{i-j}(r) = 0$, еслиCi(r=0)Oj,

соответственно:

$$P_j c(w) = 1 - \prod_{i=1}^{l} (1 - P_{i-j}(w)), \quad (2)$$

где: $P_{i-j}(w) = P_i(w)$, если $Ci(w \neq 0)Oj$, $j \neq i$;

$P_{i-j}(w) = P_i(w)$, если $j=i$;

$P_{i-j}(w) = 0$, если $Ci(w=0)Oj$.

Вероятностная модель контроля доступа к файловым объектам, определяющая вероятности несанкционированного доступа (записи или чтения) к объектам за счет уязвимостей субъектов из множества C = {C1,…, Cl}, описывается матрицей доступа, ячейка которой Mp[C,O] содержит суммарные вероятности несанкционированного чтения $P_j c(r)$ и записи $P_j c(w)$ объекта, осуществляемого за счет атак на уязвимости субъектов C = {C1,…, Cl}.

Каноническая матрица доступа вероятностной модели контроля доступа, Мрк имеет следующий вид [13]:

$$Mpk = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & O1 \end{array} \\ \begin{bmatrix} P1(r), P1(w) & 0 & \dots & 0 \\ 0 & P2(r), P2(w) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & P1(r), P1(w) \end{bmatrix} \end{array}$$

Следуя (1) и (2), вматрицедоступаМрк: $P_j c(r) = P_i(r)$, $P_j c(w) = P_i(w)$.

Лемма 12. При реализации разграничительной политики, основанной на полной изолированности обработки процессами (приложениями) информации, что описывается канонической матрицей доступа Мрк вероятностной модели, последствия от атак на уязвимости процессов (приложений) минимальны и составляют: Pi(r), Pi(w), i=1,...,l.

Доказательство. Для канонической матрицы доступа Мрк имеет место: Pjc(r) = Pi(r), Pjc(w) = Pi(w). Значения Pjc(r) и Pjc(w) сделать меньше без запрета соответствующих прав доступа Ci(r)Oj, Ci(w)Oj (т.е. без запрета необходимых прав доступа к соответствующим объектам) не представляется возможным. Включение в матрицу любого дополнительного права доступа Ci(r)Oj, соответственно, Ci(w)Oj, следуя (1) и (2), приводит к увеличению значений Pjc(r) и Pjc(w). Лемма доказана.

Следствие. Реализация разграничительной политики, основанной на полной изолированности обработки процессами (приложениями) информации, что описывается канонической матрицей доступа Mk, 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 и/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 13. Безопасным является включение в каноническую матрицу доступа (присутствует право Ci(r,w)Oi) права доступа Cj(r)Oi, соответственно, Cj(w)Oi, i≠j, при выполнении условия: Pj(w) <<Pi(w), соответственно, Pj(r) <<Pi(r).

Доказательство. Обратимся к (1), (2). При включении в каноническую матрицу доступа права доступа Cj(r)Oi, соответственно, Cj(w)Oi, i≠j, получаем:

$$P_ic(r) = 1- (1-P_i(r))(1-P_j(r))$$

$$P_ic(w) = 1- (1-P_i(w))(1-P_j(w)),$$

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

$$P_ic(r) = P_i(r)$$

$$P_ic(w) = P_i(w).$$

Лемма доказана.

Лемма 14. Максимально опасным является включение в каноническую матрицу доступа (присутствует право Ci(r,w)Oi) права доступа Cj(r)Oi, соответственно, Cj(w)Oi, i≠j, при выполнении условия: Pj(w) >>Pi(w), соответственно, Pj(r) >>Pi(r).

Доказательство. Обратимся к (1), (2). При включении в каноническую матрицу доступа права доступа Cj(r)Oi, соответственно, Cj(w)Oi, i≠j, получаем:

$$P_ic(r) = 1- (1-P_i(r))(1-P_j(r))$$

$$P_ic(w) = 1- (1-P_i(w))(1-P_j(w)),$$

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

$$P_ic(r) = P_j(r)$$

$$P_ic(w) = P_j(w).$$

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 файловым 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 файловым объектам), в безопасной системе в обязательном порядке должен быть реализован метод контроля доступа процессов (приложений) к файловым объектам, в котором процесс (приложение) выступает в качестве самостоятельного субъекта доступа, для которого разграничиваются права доступа. В противном случае уровень безопасности системы будет определяться максимально опасными включениями в каноническую матрицу доступа прав доступа; уровень безопасности подобной системы на порядки ниже, чем уровень безопасности системы, описываемой канонической матрицей доступа.

Теперь рассмотрим следующее условие. Пусть на множестве P = {P1,…, Pl} – линейно упорядоченном множество вероятностей обнаружения уязвимостей процессов (приложений) из множества субъектов доступа (процессов) C = {C1,…, Cl}, задано следующее отношение: P1<<P2<<...<<Pl. Воспользуемся Леммами 12-14 и получим

матрицу доступа Мкб, описывающую максимально безопасную разграничительную политику доступа к файловым объектам относительно атак на уязвимости приложений:

$$\text{Мкб} = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & O1 \end{array} \\ \begin{bmatrix} r,w,d & r,w & \dots & r,w \\ 0 & r,w,d & \dots & r,w \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r,w \\ 0 & 0 & \dots & r,w,d \end{bmatrix} \end{array}$$

Из матрицы Мкб видим (присутствует соответствующая иерархия отношений) возможность реализации в данном случае мандатного метода контроля доступа процессов (приложений) к файловым объектам. Рассмотрим данный метод контроля доступа [13].

Будем считать, что множество C = {C1,…, Cl} – линейно упорядоченное множество субъектов доступа (процессов), идентифицируемых полнопутевыми именами их исполняемых файлов, O = {O1,…, Ol} – множество объектов доступа, а множество P = {P1,…, Pl} – линейно упорядоченное множество вероятностей обнаружения уязвимостей процессов (приложений) из множества субъектов доступа (процессов) C = {C1,…, Cl}, причем P1<<P2<<...<<Pl. Метки безопасности являются элементами линейно упорядоченного множества M = {M1,…,Ml}. Метки безопасности назначаются субъектам (процессам, приложениям) и служат для формализованного представления отношения значений вероятностей обнаружения их уязвимостей на множестве P = {P1,…, Pl}. Условимся считать, что чем меньше значение Pi субъекта, тем меньшее значение метки Mi, где i=1,...,l, ему присваивается, т.е.: M1 <M2 <M3<…<Ml.

Пусть:

- Mc – метка безопасности субъекта (группы субъектов) доступа;
- Mo – 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: Mc>Mo.
2. Субъект C имеет доступ к объекту O в режиме R = {w,r,d}, в случае, если выполняется условие: Mc = Mo.
3. В случае Mc<Mo доступ субъекта C к объекту O запрещен.

При запросе доступа к объекту диспетчер доступа по заданному правилу должен сравнивать метку безопасности субъекта, запросившего доступ (Mc), с меткой безопасности объекта, к которому запрошен доступ (M), в результате чего, в соответствии с правилом доступа, диспетчер либо разрешает запрошенный субъектом доступ либо отказывает в нем.

4.2.2. Вероятностная модель контроля доступа процессов к объектам. Защита от атак со стороны приложений, наделяемых вредоносными функциями при прочтении вредоносного файла

В части реализации защиты от атак со стороны приложений, приобретающих вредоносные функции в процессе их работы, следует рассмотреть наделение

приложения соответствующими функциями в результате прочтения им вредоносного файла, специально с этой целью созданного и загруженного в вычислительную систему. К подобным приложениям, к примеру, 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 приложений может быть наделен вредоносными функциями в результате прочтения ими специально созданного на компьютере с этой целью файла.

Построим соответствующую модель [10].

Будем считать, что вероятность записи (сохранения в процессе работы) вредоносного файла субъектом Ci, где i=1,…,l, составляет Pi(w), где i=1,…,l.

Рассмотрим некую произвольную матрицу доступа M:

$$M = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & O1 \end{array} \\ \begin{bmatrix} r, w & w & \dots & 0 \\ r & r, w & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r \\ 0 & w & \dots & r, w \end{bmatrix} \end{array}$$

Модифицируем матрицу доступа M, поместив в нее параметр Pi(w), где i=1,…,l, – вероятность записи вредоносного файла (заменим в ячейках матрицы право записи (w) на параметр Pi(w)) – получим матрицу вероятностей создания в объектах вредоносных файлов: M(Pi(w)):

$$M\,(Pi(w)) = \begin{array}{c} \\ C1P1(w) \\ C2P2(w) \\ \dots \\ C1\text{-}1P1\text{-}1(w) \\ C1P1(w) \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & O1 \end{array} \\ \begin{bmatrix} P1(w) & P1(w) & \dots & 0 \\ 0 & P2(w) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \\ 0 & P1(w) & \dots & P1(w) \end{bmatrix} \end{array}$$

Будем обозначать запись субъектом Ci в объект Oj какCi(w)Oj. Не сложно показать, что с учетом возможности разрешения записи в один объект одновременно нескольким субъектам(в пределе всем субъектам C = {C1,…, Cl}), суммарная вероятность записи вредоносного файла в объект CiP$_i$c(w) определяется следующим образом:

$$P_i c(w) = 1 - \prod_{j=1}^{l} (1 - P_j(w)), \qquad (3)$$

где:$P_j(w) = P_j(w)$, если Ci(w≠0)Oj, j ≠ i;

$P_j(w) = P_i(w)$, если j=i;

$P_j(w) = 0$, еслиCi(w=0)Oj

Теперь по поводу 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 (да и на основании существующей статистики угроз) их можно принять равновероятными. Будем обозначать вероятность атаки на запись в объект Ci 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 $P_i c(r)$, определяемой в соответствии с (3).

Подставив в матрицу доступа M в ячейки с разрешением записи w (вместо w) вероятности $P_i a(w)$, а 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)):

$$M(Pa(w,r)) = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & O1 \end{array} \\ \begin{bmatrix} P_1a(r), P_1a(w) & P_2a(w) & \dots & 0 \\ P_1a(r) & P_2a(r), P_2a(w) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & P_1a(r) \\ 0 & P_2a(w) & \dots & P_1a(r), P_1a(w) \end{bmatrix} \end{array}$$

Проанализируем матрицу M(Pa(w,r)) и сформулируем правила контроля доступа к вредоносным файлам, реализация которых может принципиально понизить ущерб от рассматриваемой группы атак.

Лемма 15. Последствия от атак, осуществляемых в результате прочтения приложением вредоносного файла, минимальны при реализации полностью изолированной обработки субъектами созданных ими файловых объектов, т.е. при реализации канонической матрицы доступа Mk:

$$Mk = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} O1 & O2 & \dots & O1 \end{array} \\ \begin{bmatrix} r,w & 0 & \dots & 0 \\ 0 & r,w & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & r,w \end{bmatrix} \end{array}$$

Доказательство. Для канонической матрицы доступа Mк минимальны вероятности занесения в систему вредоносного файла, определяемые следующим образом:$P_i c(w) = P_i(w)$, i=1,…,l. Лемма доказана.

Важным для последующих наших рассуждений является то, что в общем случае на практике значения вероятностей записи вредоносного файла различными субъектами Ci$P_i(w)$, гдеi=1,…,l, сильно различаются, иногда на порядки.

С учетом этого сформулируем правила безопасного расширения канонической матрицы доступа правами r и/или w.

Лемма 16. Безопасным является добавление в каноническую матрицу доступа Mk разрешения права записи Ci(w)Oj, гдеi≠j, при условии, чтоPi(w) <<Pj(w).

Доказательство. Приразрешении права записи Ci(w)Oj, где i≠j, при условии, что Pi(w) <<Pj(w), имеем:

$$P_j c(w) = P_j(w)$$

$$P_i c(w) = P_i(w),$$

как следствие, при дополнении права записи Ci(w)Oj, P_ja(r), P_ja(w), соответственно, P_ia(r), P_ia(w) по сравнению с канонической матрицей не изменяются.

Лемма доказана.

Замечание.Максимально опасным является добавление в каноническую матрицу Mk разрешения права записи Ci(w)Oj, где i≠j, при условии, чтоPi(w) >>Pj(w). Это объясняется тем, что в данном случае значение Pjc(w) увеличивается до значения Pi(w), максимальным образом возрастают Pja(r), Pja(w).

Лемма 17. Безопасным является добавление в каноническую матрицу Mk разрешения права чтения Ci(r)Oj, где i≠j, при условии, чтоPi(r) >>Pj(r).

Доказательство. При разрешении права чтения Ci(r)Oj, где i≠j, при условии, чтоPi(r) >>Pj(r), имеем:

$$P_jc(w) = P_j(w)$$

$$P_ic(w) = P_i(w),$$

как следствие, при дополнении права чтения Ci(r)Oj, P_ja(r), P_ja(w), соответственно, P_ia(r), P_ia(w) по сравнению с канонической матрицей не изменяются.

Лемма доказана.

Замечание.Максимально опасным является добавление в каноническую матрицу Mk разрешения права чтения Ci(r)Oj, где i≠j, при условии, чтоPi(w) <<Pj(w). Это объясняется тем, что в данном случае значение Pic(w) увеличивается до значения Pj(w), максимальным образом возрастают Pia(r), Pia(w).

4.3. Требование к заданию субъекта доступа в современной системе контроля доступа

Исходя из сказанного выше, субъект доступа в современной системе контроля доступа должен задаваться тремя сущностями: исходный идентификатор пользователя, эффективный идентификатор пользователя, процесс (полнопутевое имя исполняемого файла процесса) [8]. В общем случае при назначении правила, задаваемого для субъекта, субъект должен интерпретироваться следующим образом: какой процесс обращается к объекту, каким пользователем запущен этот процесс и от лица какого пользователя осуществляется обращение к объекту. Подобная система контроля доступа, с одной стороны, позволит решить массу наиболее актуальных современных задач защиты как в части защиты информации от несанкционированного к ней доступа пользователем, так и в части защиты от несанкционированного доступа к информации процессом, наделенным тем или иным способом вредоносными свойствами; с другой стороны, позволит решать частные задачи защиты, к примеру, посредством разграничения прав доступа только для пользователей либо только для процессов. Для этого при задании отдельных сущностей субъекта доступа должна использоваться маска «все» (например, «*»)для всех процессов и для всех пользователей, но не учитываться имя пользователя, запускающего процесс. Маски и переменные среды окружения для упрощения задания разграничительной политики целесообразно использовать и для задания субъекта доступа: например, маскойC:\Programfiles* можно указать все процессы, исполняемые из каталога C:\Programfiles [8].

4.4. Непротиворечивые модель и правила мандатного контроля доступа

С учетом построенных и рассмотренных выше вероятностных моделей контроля доступа вернемся к правилам мандатного контроля доступа, исходя из того, что ранее нами были рассмотрены альтернативные модели: модели защиты от хищения и нарушения целостности информации. Напомним, что модель целостности Биба предполагает включение в систему иерархического признака «целостность»,

отображаемого мандатом или меткой безопасности, в то время как в модели Белла-ЛаПадулы вводится классификация (категорирование) информации по уровням конфиденциальности: метки безопасности отражают категорию конфиденциальности информации.

Как отмечали, в вероятностной модели контроля доступа важным является то, что после прочтения вредоносного файла пользователем, запустившим приложение, приложение, наделяемое вредоносными функциями, получает право записи (а также модификации и удаления) во все объекты, в которые разрешены соответствующие права доступа этому пользователю.

Для построения соответствующей модели мандатного контроля доступа будем считать, что вероятность записи (сохранения в процессе работы) вредоносного файла субъектом Ci, где i=1,…,l, составляет Pi(w), где i=1,…,l. Также будем считать, что чем меньше значение Pi(w) для субъектов и, соответственно объектов, в который разрешена запись субъекту, тем меньше их порядковый номер в линейно полномочно упорядоченных множествах субъектов и объектовC = {C1,…, Cl} и O = {O1,…, Ol} и тем меньшее значение метки безопасности Mi, где i = 1, …, k, им присваивается, т.е.: M1 <M2 <M3<…<Ml, при условии, что $P_1(w)$ <<$P_2(w)$<<…<<$P_l(w)$.

Напомним, за счет чего обеспечивается защита от нарушения конфиденциальности информации при мандатном контроле доступа: в случае обработки одним и тем же пользователем информации различных уровней конфиденциальности информация различных категорий обрабатывается совершенно в различных режимах с использованием различных устройств и объектов. Чем меньше уровень конфиденциальности информации, тем шире предоставляемые возможности по ее обработке: использование неконтролируемых внешних накопителей, неконтролируемых принтеров, неконтролируемых сетевых ресурсов и т.д., что полностью обосновывает заданное нами условие: $P_1(w)$ <<$P_2(w)$<<…<<$P_l(w)$. Как следствие, несанкционированное понижение уровня конфиденциальности информации (например, за счет размещения ее пользователем в объекте, которому присвоено большее значение метки безопасности) существенно повышает риск ее хищения пользователем.

Но из этого следует и то, что чем меньше категория информации, соответственно более широкие предоставляются возможности по ее обработке с использованием неконтролируемых ресурсов, тем больше вероятность записи (сохранения в процессе работы) при ее обработке вредоносного файла.

Как следствие, между отношениями для меток безопасности Mi, где i = 1, …,l: M1 <M2 <M3<…<Ml, назначаемых при обработке категорированной информации, и для вероятностей записи вредоносного файла $P_i(w)$: $P_1(w)$ <<$P_2(w)$<<…<<$P_l(w)$, существует прямая связь.

С учетом сказанного построим модель мандатного контроля доступа, направленную на обеспечение целостности и доступности обрабатываемой информации [10]. С учетом введенных обозначений данной моделью регламентируются следующие правила доступа:

1. Субъект С имеет доступ к объекту О в режиме “Чтения” в случае, если выполняется условие: Mc>, = Mo.
2. Субъект С имеет доступ к объекту О в режиме “Записи” в случае, если выполняется условие: Mc<, = Mo.

Получаем следующую матрицу доступа Mm(ц,д):

$$Mm(ц,д) = \begin{array}{c} \\ C1(M1) \\ C2(M2) \\ ... \\ C1\text{-}1(M1\text{-}1) \\ C1(M1) \end{array} \begin{array}{c} \begin{array}{cccc} O1(M1) & O2(M2) & ... & O1(M1) \end{array} \\ \begin{bmatrix} r,w & w & ... & w \\ r & r,w & ... & w \\ ... & ... & ... & ... \\ r & r & ... & w \\ r & r & ... & r,w \end{bmatrix} \end{array}$$

Видим, что правила доступа, сформулированные для построенной нами модели и модели Биба, полностью идентичны, однако отличие этих моделей принципиально. Модель целостности Биба предполагает включение в систему иерархического признака «целостность», отображаемого мандатом или меткой безопасности (что, как отмечали ранее, и препятствует ее практическому использованию), в то время как в нашей модели, так же как и в модели Белла-ЛаПадулы, вводится классификация (категорирование) информации по уровням конфиденциальности: метки безопасности отражают категорию конфиденциальности информации.

Как следствие, именно приведенную выше модель и модель Белла-ЛаПадулы можно корректно сравнивать (рассматривать в качестве альтернативных решений), поскольку данные модели основаны на использовании одного и того же категорирующего признака. При этом, как видим, предложенная модель является полной инверсией модели Белла-ЛаПадулы.

Исходя из сказанного, можем сделать важнейший вывод о том, что безопасная обработка категорированной по уровням конфиденциальности информации в общем случае достигается при реализации следующих непротиворечивых правил мандатного контроля доступа [10]:

1. Субъект С имеет доступ к объекту О в режиме «Чтения» и «Записи» в случае, если выполняется условие: Мс = Мо.
2. Субъект С не имеет доступ к объекту О, если выполняется условие: Мс<, >Мо.

Данные правила отображаются канонической матрицей доступа Mm(к), имеющей следующий вид:

$$Mm(к) = \begin{array}{c} \\ C1(M1) \\ C2(M2) \\ ... \\ C1\text{-}1(M1\text{-}1) \\ C1(M1) \end{array} \begin{array}{c} \begin{array}{cccc} O1(M1) & O2(M2) & ... & O1(M1) \end{array} \\ \begin{bmatrix} r,w & 0 & ... & 0 \\ 0 & r,w & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & 0 \\ 0 & 0 & ... & r,w \end{bmatrix} \end{array}$$

Именно эта модель, на наш взгляд, имеет все основания быть позиционирована как «непротиворечивая модель мандатного контроля доступа», определяющая непротиворечивые правила доступа при обработке категорированной информации по уровням конфиденциальности, поскольку ее применение позволяет решать задачи защиты от нарушения конфиденциальности информации, обеспечения ее целостности и доступности в комплексе, т.е. позволяет построить безопасную систему.

Следствие.Корректная реализация мандатного контроля доступа, позволяющая решать задачи защиты от нарушения конфиденциальности информации, обеспечения

ее целостности и доступности в комплексе, т.е. позволяющая построить безопасную систему в общем случае, возможна при полной изолированности обработки информации различных категорий (уровней) конфиденциальности, что обеспечивается при реализации канонической матрицы доступа.

Замечание. Все сказанное относится и применительно к реализации дискреционного контроля доступа для решения им задачи защиты обработки категорированной информации.

5. Альтернативные способы задания объектов и хранения правил доступа

Сразу отметим, что здесь мы рассмотрим один из ключевых вопросов реализации механизма контроля доступа в вычислительной системе. Как отмечали ранее, возможны два подхода к реализации разграничительной политики: назначение правил доступа к объектам субъектов и назначение правил доступа субъектов к объектам.

Данные подходы принципиально различаются как по своим возможностям, так и способами реализации, в первую очередь в части задания и хранения правил доступа к объектам [16].

В обоих случаях контроль (разграничение) доступа осуществляется центральным элементом системы защиты – диспетчером доступа, перехватывающим все запросы от субъектов к объектам, идентифицирующим субъекты, объекты доступа и параметры запрашиваемого доступа субъекта к объекту (чтение, запись, исполнение и т.д.), предоставляя на основании изначально заданных правил субъекту запрашиваемый доступ либо отказывая в нем.

При этом в первом случае для задания правил доступа каждому объекту назначаются атрибуты доступа, хранящиеся вместе с объектом и указывающие на то, каким субъектам и какой доступразрешается/запрещается. При запросе доступа диспетчер, получая необходимую учетную информацию из запроса, считывает атрибуты, принадлежащие объекту, к которому запрошен доступ, на основании анализа которых определяет корректность обрабатываемого запроса. Т.е. в данном случае именно объекты являются первичной сущностью в разграничительной политике. Они не задаются в матрице, а им присваиваются правила доступа субъектов (реализуется транспонированная матрица доступа, см. выше).

Альтернативный подход к реализации контроля доступа радикально отличается тем, что правила доступа формируются уже не для объекта, а для субъекта доступа. Для субъектов доступа задается, к каким объектам и по каким правилам им разрешается доступ, т.е. правила доступа являются здесь уже принадлежностью не объекта, а субъекта доступа. Естественно, что формируемая подобным образом матрица доступа уже не является принадлежностью какого-либо объекта, а образует самостоятельную сущность, как следствие, матрица доступа представляет собою отдельный системный объект (как правило, файл). При запросе доступа диспетчер обращается к объекту «матрица доступа» и считывает из него разрешенные правила доступа субъекта к объекту, фигурирующие в запросе, на основании которых анализирует корректность запроса. В данном случае уже субъекты являются первичной сущностью в разграничительной политике, им присваиваются правила доступа к объектам, которые соответствующим образом назначаются в матрице доступа.

Заметим, что реализация мандатного метода контроля доступа предполагает назначение метки безопасности и субъекту и объекту доступа. Т.е. в данном случае вопрос о том, разграничиваются права доступа субъекта к объекту или наоборот, не актуален. При этом метки безопасности объектов доступа также могут храниться как в

качестве атрибутов доступа, так и в отдельном файле в зависимости от практической реализации.

Оценим достоинства и недостатки данных альтернативных подходов [16].

Как отмечали, принципиальное отличие рассмотренных подходов состоит в том, что при назначении правил доступа к объектам субъектов правила хранятся в атрибутах объекта, т.е. физически привязаны к конкретному объекту; приназначениже правил доступа субъектов к объектам, поскольку правила являются принадлежностью субъекта, а не объекта, они (матрица доступа) хранятся в отдельном объекте, например, в файле.

В современных ОС существует множество способов обращения к одному и тому же файловому объекту: по длинному или по короткому имени, с использованием различных кодировок, различных ссылок. Некоторые приложения вообще обращаются к файловым объектам только по ссылкам. Рассмотрим, к чему это приводит. Например, к каталогу “\Programfiles\” можно обратиться по короткому имени “\Progra~1\”. Следовательно, либо в разграничительной политике доступа должны быть указаны все возможные способы адресации к файловому объекту, что нельзя отнести к реализуемой на практике возможности (корректно задать подобную разграничительную политику будет невозможно), либо диспетчер доступа должен «уметь» преобразовывать все возможные способы идентификации объекта к одному виду (например, к длинному имени), который уже и будет использован в разграничительной политике. Не говоря уже о существенном усложнении технического решения со всеми вытекающими из этого последствиями, при таком подходе присутствует и ряд принципиальных вопросов, например, а что делать при обращении к объектам с использованием ссылок?

Как видим, хранение правил доступа в атрибутах объектов имеет неоспоримое преимущество в части построения безопасной системы.

Теперь о практической применимости альтернативных решений. Наверное, многие применяли на практике первый подход. Нужно для каждого (либо для критичных) файлового объекта задать права доступа к нему субъектов. Наверное, при реализации простейших разграничительных политик это и не так критично, но если, как мы отмечали, субъект доступа определяется тремя сущностями: исходный идентификатор пользователя, эффективный идентификатор пользователя, имя процесса (полнопутевое имя исполняемого файла процесса), а без этогореализовать эффективную разграничительную политику в современных условиях невозможно, то представьте себе сложность настройки подобной разграничительной политики.Необходимо разграничивать права доступа к объектам не только для пользователей (причем с учетом возможности смены их исходных идентификаторов), но и для процессов, причем как для прикладных, так и для системных (например, для сетевых служб, на которые часто осуществляются удаленные атаки[9]). Это уже становится крайне трудоемкой, если вообще решаемой на практике, задачей администрирования, а это также вопросы безопасности систем.

Реализация данного подхода связана не только со сложностью задания, что естественно сказывается на безопасности системы, но и со сложностью визуального восприятия заданной разграничительной политики доступа, как следствие, со сложностью последующей эксплуатацией защищенной системы. Ведь в конечном счете в вычислительных системах интерес представляет не то, какие права доступа заданы к объекту, а именно то, как права доступа распределены между субъектами, т.е. в интерфейсе средства защиты в одном окне целесообразно отображать все права доступа, разрешенные субъекту (а не к конкретному объекту), что будет наглядным и существенно упростит задачу администрирования и в результате существенно снизит вероятность ошибки администрирования.

Именно подобное решение и лежит в основе второго подхода, к важнейшим достоинствам которого также можно отнести возможность применения масок и переменных среды окружения для заданияобъектов доступа. Например, маска «*» покрывает все файловые объекты, маска «*.exe» покрывает все файлы с расширением «exe», %Windir% обозначает соответствующую системную папку загруженной ОС, а %Windir%*.exe – все файлы с расширением «exe» из этой папки, и т.д. Естественно, трудно себе представить возможность задания атрибутов для файлового объекта %Windir%*.exe (под эту маску подпадают десятки, если не сотни конкретных файлов).

Это не только кардинально упрощает реализацию сложных разграничительных политик доступа (без чего в современных условиях не обойтись), но и позволяет получить принципиально новые свойства защиты. Проиллюстрируем сказанное на примере реализации разграничительной политики доступа по типам файлов (по расширениям файлов) [1].

Важнейшей характеристикой файла в ОС MicrosoftWindows является его расширение. Именно расширением файла определяется его тип, например, исполняемый файл имеет расширения «.exe», «.com», «.bat», «.sys» и т.д., файлы-скрипты– «.vbs» и «.vbe» (VBScript) , «.js» и «.jse» (JavaScript), «.scpt» (AppleScript) и т.д. То же относится к тестовым и иным обрабатываемым в КС файлам.

Как следствие, расширение файла может быть использовано в качестве объекта доступа в разграничительной политике с целью реализации разграничения доступа к типам файлов.

Рассмотрим лишь несколько задач защиты и возможность их решения, посредством реализации разграничения доступа по расширениям файлов.

1) **Защита от загрузки в вычислительной системе и запуска вредоносных программ**

Решается данная задача следующим образом:

- Разрешается исполнять файлы только с конкретными (заданными) расширениями. Опытным путем определен список расширений исполняемых файлов, которые необходимо разрешить на исполнения для корректной работы операционной системы: «.exe», «.bat», «.config», «.dll», «.manifest», «.drv», «.fon», «.ttf», «.sys» (файлы с данными расширениями разрешаются на чтение и исполнение);
- Предотвращается возможность удаления и модификации разрешенных к исполнению файлов (файлов с заданными расширениями);
- Предотвращается возможность создания в новых разрешенных к исполнению файлов (файлов с заданными расширениями);
- Предотвращается возможность переименования разрешенных к исполнению файлов (файлов с заданными расширениями)
- Предотвращается возможность переименования иных файлов в файлы, разрешенные к исполнению (в файлы с заданными расширениями).

В результате в вычислительной системе можно будет исполнить только санкционированные исполняемые файлы; исполняемые файлы вредоносных программ (с заданными расширениями) не смогут быть установлены и затем исполнены.

Замечание. Подобные правила можно установить для всех пользователей, включая системных, можно же разрешить установку исполняемых файлов только администратору.

2) **Защита от несанкционированной модификации системных файлов (исполняемых файлов и файлов настройки ОС и приложений)**

Предотвращается возможность несанкционированного удаления/модификации, переименования файлов с соответствующими расширениями.

Замечание. Подобные правила можно установить для всех интерактивных пользователей, можно кже разрешить модификацию системных файлов только администратору.

Защита от загрузки в вычислительной системе вредоносных скриптовых файлов.

При прочтении подобного файла соответствующее приложение (например, интернет-браузер, Java-машина наделяются вредоносными свойствами.

Предотвращается возможность несанкционированного удаления/модификации, переименования установленных в системе скриптовых файлов (по их расширениям), создания новых скриптовых файлов, переименование созданного в системе файла в скриптовый файл.

Замечание. Подобные правила можно установить для всех интерактивных пользователей, можно же разрешить модификацию скриптовых файлов только администратору. Заданием разграничений для субъекта доступа (процесс) (в разграничительной политике задается полнопутевым именем его исполняемого файла) можно локализовать типы (по расширениям) скриптовых файлов, к которым будет разрешен доступ на чтение конкретному приложению.

3) **Ограничение работы пользователя только с определенными типами файлов (по расширениям), в том числе исполняемых**

Например, можно запретить пользователю работу с файлами, имеющими расширения «.http», «.avi» и т.д.

Замечание. В качестве субъекта доступа в разграничительной политике здесь выступает пользователь.

4) **Ограничение работы приложения только с определенными типами файлов (по расширениям), в том числе исполняемых**

Например, можно запретить Интернет-браузеру доступ к файлам, предназначенным для хранения в вычислительной системе конфиденциальной информации, например, к файлам, имеющим расширение «.doc» и т.д.

Замечание. В качестве субъекта доступа в разграничительной политике здесь выступает процесс.

Это лишь примеры решаемых задач, список которых может быть существенно расширен.

На данном примере видим, насколько широки возможности по реализации эффективных разграничительных политик, предоставляемые прireализации способа назначения правил доступа субъектов к объектам.

Подытожив сказанное, отметим, что оба из рассмотренных подходов характеризуются существенными недостатками, что обусловливает актуальность разработки принципиального нового подхода, устраняющего данные принципиальные недостатки. К этому вопросу мы вернемся в следующем разделе.

6. Исключение из разграничительной политики сущности «объект доступа». Модели и методы контроля доступа

Обобщим все сказанное ранее и определимся с тем, какие же задачи остаются наиболее актуальными для решения при разработке современных методов контроля доступа:

- необходимость упрощения задачи администрирования путем реализации разграничительной политики доступа, в которой субъект доступа задается тремя сущностями: исходный идентификатор пользователя, эффективный идентификатор пользователя, процесс (полнопутевое имя исполняемого файла процесса);
- необходимость реализации разграничительной политики с заданием разграничений прав доступа для субъектов (первичной сущностью в разграничительной политике должен быть «субъект доступа») при ее физической привязке (в качестве атрибутов) к объектам.

6.1.Принципы контроля доступа к создаваемым файлам

Прежде всего, в двух словах остановимся на основном противоречии существующих подходов к реализации разграничительной политики доступа субъектов к объектам [3,6]. Первичным при назначении разграничений прав доступа в известных методах контроля доступа является объект, что логично, т.к. именно объект и защищается от несанкционированного доступа. Однако файловые объекты принципиально различаются. Они могут быть подразделены на статичные (прежде всего системные) и создаваемые в процессе работы системы. Принципиальная разница между этими группами объектов в части задания разграничительной политики доступа огромна, и состоит она в том, что системные объекты присутствуют на компьютере на момент настройки администратором прав доступа субъектов к объектам, а создаваемых еще попросту нет. Резонно возникает вопрос: как же к ним разграничивать доступ, если их еще нет? А ведь это те объекты (прежде всего файлы), которые в первую очередь и нуждаются в защите от несанкционированного доступа, поскольку именно они и содержат защищаемую конфиденциальную информацию.

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 сохранения 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 исчезает из разграничительной политики доступа. Разграничительной политикой для созданных папок-контейнеров пользователи «принуждаются» создавать свои файлы только в определенных папках, в противном случае невозможно установить какие-либо разграничения. Созданные файлы наследуют разграничения, установленные для папок. Посредством же реализации разграничительной политики доступа к папкам-контейнерам для пользователей разграничивается доступ и к созданным в процессе функционирования системы файлам.

Как видим, задача реализации разграничительной политики доступа, состоящая в разграничении прав доступа субъектов к обрабатываемой в вычислительной системе информации, решается не напрямую, а опосредованно через объекты доступа.

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 к файловым объектам, а в ряде случае и к невозможности ее корректной реализации (обо всем этом мы говорили ранее). Достаточно задуматься о том, какие действия потребуется выполнить администратору, чтобы, например, изолировать обработку информации десятком приложений – для каждого субъекта потребуется создавать свою папку, далее разграничивать к созданным папкам права доступа. Соответственно придется разграничивать доступ и к иным папкам, в частностик системным. А не разделяемые системой и приложениями папки 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 собою весьма вероятную угрозу), но и к невозможности гарантированно построить корректную разграничительную политику доступа к файловым объектам в общем случае. Возможность же корректной реализации в данном случае мандатного контроля доступа вообще ставится под сомнение.

Принципы контроля доступа 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 [3-6]:

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 сущностями: «субъект доступа (учетная информация), запрашивающий доступ к объекту» и «субъект доступа (учетная информация), создавший этот объект».
3) При создании субъектом нового файла самим файлом наследуется учетная информация субъекта доступа, создавшего этот файл.
4) При запросе доступа к любому файлу диспетчер доступа анализирует наличие, а при наличии – содержимое унаследованной файлом учетной информации создавшего его субъекта доступа. При наличии анализирует заданные правила доступа, в результате чего предоставляет запрошенный субъектом доступ либо отказывает в нем. При отсутствии – анализирует правило контроля доступа к неразмеченным (не унаследовавшим учетную информацию субъекта) объектам.

6.2. Модель и метод дискреционного контроля доступа к создаваемым файлам

В качестве объекта доступа здесь рассматривается именно создаваемый файл (какого-либо контроля папок, равно как и их создания администратором, не требуется) как объект, непосредственно содержащий защищаемую информацию. При создании субъектом нового файла файлом наследуется учетная информация субъекта доступа, создавшего этот файл [4,11]. Наследуется в том смысле, что становится соответствующим атрибутом созданного файла. Учетная информация субъекта доступа, как отмечали ранее, в общем случае определяется следующими тремя сущностями: первичный идентификатор пользователя (учетная запись, под которой осуществлен вход в систему); эффективный идентификатор пользователя (учетная запись, от которой осуществлен запрос доступа к ресурсу), полнопутевое имя процесса, обращающегося к объекту.

При запросе доступа к любому файлу диспетчер доступа анализирует наличие, а при наличии–содержимое унаследованной файлом учетной информации субъекта доступа. При отсутствии подобной информации диспетчер либо разрешает запрошенный доступ (если доступ к такому объекту им не разграничивается), либо реализует иное заданное правило контроля доступа к неразмеченным объектам. При наличии– анализирует матрицу доступа, в которой определены, какие права доступа к объекту (характеризуется учетной информацией создавшего его 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 доступа».

Построим модель [4,11]. Если считать, что множество C = {C1,…, Cl}– линейно упорядоченное множество субъектов доступа, а R = {R1,…, Rm} –конечное множество прав доступа (чтение (r), запись (w), удаление (d), исполнение (x) и т.д., отсутствие прав доступа (0)) субъекта Ci к объекту, созданному субъектом Cj, где 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 (условимся в строках матрицы указывать учетную информацию субъектов, запрашивающих доступ к объектам, а в столбцах – учетную информацию субъектов, унаследованную созданными объектами):

$$M = \begin{array}{c} \\ C1 \\ C2 \\ \dots \\ C1\text{-}1 \\ C1 \end{array} \begin{array}{c} \begin{array}{cccc} C1 & C2 & \dots & C1 \end{array} \\ \begin{bmatrix} r,w,d & w & \dots & 0 \\ r & r,w,d & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r \\ 0 & w & \dots & r,w,d \end{bmatrix} \end{array}$$

В любой момент времени система описывается своим текущим состоянием Q = (C,C,M), где M[C,C] – ячейка матрицы, содержит набор прав доступа. Будем обозначать Ci(R)Cjкак разрешение права доступа субъекту Ci к объекту, созданным субъектом Cj, где i=1,…,l; j=1,…,l, aR = {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, дискреционный метод контроля доступа к создаваемым файловым объектам является методом контроля доступа с принудительным управлением потоками информации, вследствие 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), т.е. 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 разграничения 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.

6.3. Модель и метод мандатного контроля доступа к создаваемым файлам

В части мандатного метода контроля доступа к создаваемым файлам можем сразу отметить, что именно в данном случае (контроль доступа к создаваемым файлам) он может быть крайне эффективен ввиду собственно своего предназначения – обеспечивать контроль доступа к категорированной информации, которая в свою очередь сохраняется именно в создаваемых в процессе функционирования системы файлах [11].

Построим модель [5,15]. Будем считать, что чем выше полномочия субъекта и уровень конфиденциальности объекта, тем меньше их порядковый номер в линейно полномочно упорядоченных множествах субъектовС = {С1,…, Ск} и тем меньшее значение метки безопасности Mi, гдеi = 1, …, k, им присваивается, т.е.: М1 <М2 <М3<…<Mk.

В качестве контролируемого объекта здесь рассматривается создаваемый файл (какого-либо контроля папок, равно как и их создания администратором, не требуется) как объект, непосредственно содержащий защищаемую информацию. Задание разграничительной политики доступа состоит исключительно в назначении меток безопасности субъектам Мс (что кардинально упрощает задачу администрирования). При создании субъектом нового файла файлом наследуется учетная информация субъекта доступа – его метка безопасности Мс (обозначим унаследованную метку как Мсо, при этом Мсо= Мс).

При запросе доступа к любому файлу диспетчер доступа анализирует наличие, а при наличии– собственно значение метки безопасности Мсо, унаследованной данным файлом. При наличии метки Мсо у файла диспетчер сравнивает эту метку с меткой субъекта, запросившего доступ к файлу, Мс – анализирует выполнение заданных правил контроля доступа. В результате анализа данной информации с учетом реализуемых правил контроля доступа диспетчер либо разрешает запрошенный субъектом доступ к файлу, либо отказывает в нем.

Правила, направленные на защиту от понижения категории обрабатываемой информации (модель Белла-ЛаПадулы), в данном случае имеют тот же вид, что и для метода контроля доступа к статичным файловым объектам:

1) Субъект С имеет доступ к объекту О в режиме “Чтения” в случае, если выполняется условие: Мс<, = Мсо.
2) Субъект С имеет доступ к объекту О в режиме “Записи” в случае, если выполняется условие: Мс = Мсо.

Матрица доступа Mmздесь уже принимает принципиально иной вид:

$$Mm = \begin{array}{c} \\ C1(M1) \\ C2(M2) \\ \dots \\ C1\text{-}1(M1\text{-}1) \\ C1(M1) \end{array} \begin{array}{c} \begin{array}{cccc} C1(M1) & C2(M2) & \dots & C1(M1) \end{array} \\ \begin{bmatrix} w,r & r & \dots & r \\ 0 & r & \dots & r \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r \\ 0 & 0 & \dots & w,r \end{bmatrix} \end{array}$$

В случае реализации неиерархической обработки меток безопасности («непротиворечивая модель мандатного контроля доступа» вступают в силу следующие правила:

1) Субъект С имеет доступ к объекту О в режиме «Чтения» и «Записи» в случае, если выполняется условие: Мс = Мсо.
2) Субъект С не имеет доступ к объекту О в случае, если выполняется условие: Мс ≠ Мсо.

Как видим, в разграничительной политике доступа присутствуют только субъекты и их метки безопасности [5,15].

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 размечать, а уж тем более разделять между субъектами доступа, не требуется! При этом заметим, что мандатный метод предназначен исключительно для контроля доступа к обрабатываемой информации, т.е. к создаваемым файлам (для решения каких-либо системных задач защиты он априори не предназначен).

Аналогично тому, как это было описано ранее, в данном случае так же можно контролировать корректность сервисов олицетворения (смены первичного идентификатора на эффективный), что никак не скажется на сложности администрирования.

6.4.Правила доступа. Требования к построению безопасной системы

Требования в данном случае практически полностью совпадают с аналогичными требованиями, рассмотренными нами ранее, с поправкой на то, что в разграничительной политике доступа отсутствует сущность «объект» доступа (т.е. матрица доступа имеет иной вид). Вместе с тем появляются и дополнительные требования [6,11].

Сначала без доказательства (доказательства рассмотрены ранее) приведем требования к дискреционному методу контроля доступа:

1) **Не допустимо разрешение пользователям права исполнения (x) созданного в процессе функционирования системы файла.**

При реализации данного требования система безопасна относительно права исполнения (x). Заметим, что на реализации именно этого требования основана эффективная защита от вредоносных программ.

Замечание. Как следствие, к разрешенным правам доступа к создаваемым файловым объектам относятся: R = {w,r,d}.

Отметим, что реализация данного требования позволяет реализовать эффективную защиту от вредоносных программ [7].

Замечание. Простота настройки подобной разграничительной политики (вся настройка осуществляется автоматически средством защиты за счет автоматической разметки диспетчером доступа создаваемых файлов) позволила создать авторам эффективное средство защиты от запуска вредоносных программ для широкого применения. Минимальны требования к квалификации при эксплуатации данного средства защиты, а решается им одна из актуальнейших современных задач защиты информации [12].

2) **При назначении разграничительной политики «по умолчанию» должны быть установлены права доступа: Ci(w,r,d)Ci, i=1,…,l.**

Данное правило обусловливает задание диагональной («канонической») матрицы доступа, характеризуемой условием: Ci(w,r,d)Ci, i=1,…,l; Ci(0)Cj, j≠j, i=1,…,l, j=1,…,l.

Замечание. Реализующая каноническую модель доступа система безопасна относительно права записи (w) и относительно права чтения (r).

3) **При расширении канонической матрицы доступа разрешенным правом записи (w): Cj(w)Ci, i≠j, при исходно разрешенном праве записи (w): Ci(w)Ck, i≠k, должно быть разрешено право записи (w): Cj(w)Ck, j≠k, что предотвратит утечку права записи (w): Cj(w)Ck.**

4) **При расширении канонической матрицы контроля доступа разрешенным правом чтения (r): Cj(r)Ci, i≠j, при исходно разрешенном праве чтения (r): Ci(r)Ck, i≠k, должно быть разрешено право чтения (r): Cj(r)Ck, j≠k, что предотвратит утечку права чтения (r): Cj(r)Ck.**

Теперь о дополнительном требовании.

Лемма 18.Система безопасна относительно права удаления (d) при реализации следующего правила: любой включающий объект (папка) может быть удален любым субъектом при условии отсутствия включенных в него объектов (в первую очередь файлов).

Доказательство.Метод дискреционного контроля доступа к создаваемым файловым объектам не предполагает реализации разграничительной политики доступа к статичным объектам (к папкам), в которые включаются (записываются) создаваемые файлы. Как следствие, любой субъект может удалить подобную папку, соответственно и все включенные в нее файлы, что может привести к утечке права удаления (d). При запрете удаления папки, а также при наличии включенных в нее файлов с учетом реализации очевидного права: Ci(d)Ci (право Cj(d)Ci, i≠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.

Теперь проверим выполнимость соответствующих требований мандатным методом контроля доступа. Подобная необходимость вызвана тем, что метки безопасности в данном случае не задаются. Они наследуются от создавшего объект субъекта, причем наследуются исключительно файлами.

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.13.

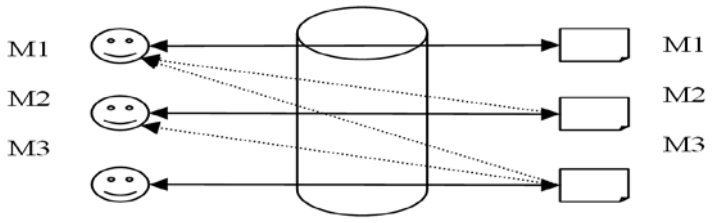


Рисунок 13.Схема мандатного контроля доступа при иерархической структуре файловых объектов

Лемма 19.Мандатный контроль доступа к создаваемым файловым объектам, основанный на наследовании метки безопасности от субъекта доступа, создающего объект, в общем случае корректно реализует правила мандатного контроля доступа.

Доказательство. С учетом того, что включающий объект (папка) никоим 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.13, сводится к правилу, проиллюстрированному на рис.14, как следствие, правила мандатного метода контроля доступа при иерархической структуре файловых объектов реализуется корректно.

Лемма доказана.

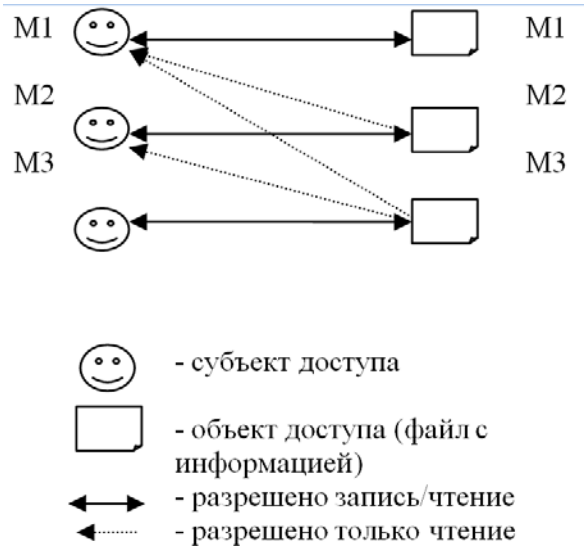


Рисунок 14.Иллюстрация основополагающего правила мандатного метода контроля доступа

Из сказанного можем сделать крайне важный вывод, иллюстрирующий преимущества метода контроля доступа к создаваемым файлам.

Вывод. Мандатный метод контроля доступа к создаваемым файлам позволяет в общем случае корректно реализовать разграничительную политику доступа на основе меток безопасности, в том числе и при наличии в системе не разделяемых файловых объектов [5].

Теперь, что касается удаления. Здесь имеем ту же ситуацию, что и для дискреционного метода контроля доступа.

Лемма 20.Система безопасна относительно права удаления (d) при реализации следующего правила: любой включающий объект (папка) может быть удален любым субъектом при условии отсутствия включенных в него объектов (в первую очередь файлов).

Доказательство.Метод мандатного контроля доступа к создаваемым файловым объектам не предполагает реализации разграничительной политики доступа к статичным объектам (к папкам), в которые включаются (записываются) создаваемые файлы. Папкам не присваиваются метки безопасности (доступ к ним не контролируется) – метки наследуются файлами. Как следствие, любой субъект может удалить подобную папку и все включенные в нее файлы соответственно, что может привести к утечке права удаления (d). При запрете удаления папки, при наличии включенных в нее файлов с учетом заданного права удаления созданного файла при условии:Мс = Мсо, утечка права удаления (d) становится невозможной. Лемма доказана.

Замечание. Естественно, что также как и при реализации контроля доступа к статичным файлам, при контроле доступа к создаваемым файлам мандатный и дискреционный методы контроля доступа могут применяться одновременно.

6.5. Преимущества методов контроля доступа к создаваемым файлам

Ранее мы акцентировали внимание на наиболее актуальных задачах, которые должны решаться при разработке современных методов контроля доступа:

- необходимость упрощения задачи администрирования путем реализации разграничительной политики доступа, в которой субъект доступа задается тремя сущностями: исходный идентификатор пользователя, эффективный идентификатор пользователя, процесс (полнопутевое имя исполняемого файла процесса);
- необходимость реализации разграничительной политики с заданием разграничений прав доступа для субъектов (первичной сущностью в разграничительной политике должен быть «субъект доступа») при ее физической привязке (в качестве атрибутов) к объектам.

Рассмотрим, как решены эти задачи [11].

Сначала об упрощении задачи администрирования путем задания разграничительной политики. В части мандатного метода контроля доступа какая-либо иллюстрация не требуется: вся задача администрирования сводится к назначению меток безопасности для субъектов [15]. Вся разметка объектов (создаваемых файлов) осуществляется диспетчером доступа автоматически.

Принципиальное упрощение задачи администрирования при использовании дискреционного метода контроля доступа к создаваемым файлам проиллюстрируем на нескольких примерах, см. табл.1 и табл.2.

Таблица 1. Пример разграничительной политики доступа к создаваемым файлам для пользователей

Правило доступа	**Субъект (пользователь, процесс), создавший файл (эти файлы автоматически размечаются)**	**Субъект (пользователь, процесс), имеющий право доступа к созданному (размеченному) файлу**	**Разрешенные права доступа к созданному (размеченному) файлу**
1	User1	User1	Чтение, запись
2	User1	*	Запрет доступа

В Табл.1 двумя правилами полностью изолирована работа пользователя User1. Только он сможет получить доступ к создаваемым им файлам. Никакой иной пользователь (что задается маской «*») не имеет права доступа к файлам, созданным пользователем User1. В правилах, представленных в табл.2, в качестве субъекта доступа уже выступает сущность процесс. Простейшей разграничительной политикой, представленной в табл.2, решена сложнейшая задача изолированной работы интернет-браузера: данное приложение не может получить доступ к файлам, создаваемым в

вычислительной системе иными приложениям, все что им сохранено на компьютере не может быть исполнено.Видим, насколько упрощает реализацию сложнейших разграничительным политик реализация разграничений прав доступа между субъектами, предполагающая исключение из разграничительной политики сущности «объект доступа».

Таблица 2.Пример разграничительной политики доступа к создаваемым файлам для процессов

Правило доступа	**Процесс, создавший файл (эти файлы автоматически размечаются)**	**Процесс, имеющий право доступа к созданному (размеченному) файлу**	**Разрешенные права доступа к созданному (размеченному) файлу**
1	E:\Program Files\Internet Explorer\iexplore.exe	E:\Program Files\Internet Explorer\iexplore.exe	Чтение, запись
2	*	E:\Program Files\Internet Explorer\iexplore.exe	Запрет доступа
3	E:\Program Files\Internet Explorer\iexplore.exe	*	Чтение, запись

Теперь о другой решаемой нами задачи –задачинеобходимости реализации разграничительной политики с заданием разграничений прав доступа для субъектов (первичной сущностью в разграничительной политике должен быть «субъект доступа») при ее физической привязке (в качестве атрибутов) к объектам.

Реализация контроля доступа базируется на следующих трех сущностях: матрица доступа M, хранящаяся в отдельном файле, учетная информация субъекта доступа, создавшего файл, хранящаяся в атрибутах этого файла, учетная информация субъекта доступа, запрашивающего доступ к файлу (получается диспетчером из запроса доступа). Как видим, особенность решения состоит в том, что при контроле доступа используются и матрица доступа, хранящаяся в отдельном файле, и атрибуты файла, в которых хранится учетная информация создавшего файл субъекта доступа [16].

Обратимся к матрице доступа. Как видим, в ней отсутствуют объекты доступа как таковые: все разграничения осуществляются между субъектами, при этом учетная информация субъекта, создавшего файл, к которому запрошен доступ, получается диспетчером из атрибутов данного файла. Таким образом, учетная информация субъекта, создавшего файл, однозначно физически связана с этим файлом (хранится в атрибутах файла). И именно она (а не имя файлового объекта) и используется в разграничительной политике доступа, определяя права доступа субъектов к объектам. Каким бы образом (по какому имени или ссылке) ни обратиться к файлу (идентифицировать файл), файлу однозначно будет соответствовать учетная информация именно создавшего этот файл субъекта (сохраненная в атрибутах файла), причем именно эта информация (а не имя файла) используется в заданных матрицей

доступа правилах доступа. Из сказанного можем сделать вывод о том, что и эта задача решается методами контроля доступа к создаваемым файлам в полном объеме: реализуется физическая привязка (в качестве атрибутов)правил к объектам доступа [16].

6.6. Реализация контроля доступа к статичным файлам, основанного на их ручной разметке

Выше мы показали, насколько серьезны преимущества методов контроля доступа к создаваемым файлам. Данные методы основаны на автоматической разметке диспетчером доступа создаваемых файлов: записи в атрибуты создаваемых файлов учетной информации субъекта доступа (метки безопасности субъекта при мандатном контроле), создающего файл, что позволяет их применять лишь в отношении реализации контроля доступа к создаваемым файлам. Однако на компьютере еще присутствуют и файлы, названные нами статичными, – это в первую очередь системные файлы.

Для практической реализации изложенных выше принципов контроля в данном случае потребуется ручная разметка статичных файлов. Данная разметка, только проводимая администратором вручную с использованием соответствующей утилиты, также предполагает запись в атрибуты статичных файлов учетной информации субъекта доступа создавшего файл. Здесь имеет смыл говорить о субъекте доступа – пользователь, администратор (что и имеет место на практике) [14].

Например, подобным образом вручную целесообразно разметить файлы в каталогах Windows и Programfiles. Разграничительной политикой доступа следует установить правила для всех интерактивных пользователей (кроме администратора), предотвращающие возможность ими удаления и модификации статичных (системных) файлов. Только администратор сможет удалить и модифицировать системный файл. Все же вновь создаваемые администратором системные файлы (например, установка новой программы) будут автоматически размечаться. Эти файлы будут наследовать в атрибутах учетную информацию администратора, в отношении них будут действовать соответствующие правила доступа.

Диспетчер доступа при этом должен функционировать точно так же, как это было описано применительно к контролю доступа к создаваемым файлам.

Таким образом, как видим, данный метод контроля доступа с ручной и автоматической разметкой файлов, предполагающий исключение из разграничительной политики сущности «объект доступа» [14], может рассматриваться в качестве полноценного и крайне эффективного метода, реализующего контроль доступакак к статичным (системным), так и к создаваемым в процессе эксплуатации вычислительной системы файлам.

7. Сессионный контроль доступа

Сессионный контроль доступа предназначен для реализации безопасной обработки информации различных уровней конфиденциальности (на практике, как правило, открытой и конфиденциальной) на одном и том же компьютере одним и тем же пользователем [2]. Сложность обеспечения защиты информации при реализации сессионного контроля доступа в современных условиях заключается в том, что пользователь должен рассматриваться в качестве вероятного потенциального злоумышленника. При этом ему должна предоставляться возможность работы на одном компьютере принципиально в различных режимах обработки информации, определяемых уровнем ее конфиденциальности. Это и есть возможность запуска пользователем тех или иных приложений, возможность подключения различных

устройств, работа с различными сетевыми ресурсами и т.д. Естественно, что возможности по обработке открытой информации априори куда шире, чем конфиденциальной. Как следствие, возникает задача защиты в первую очередь от утечек конфиденциальной информации за счет предоставления возможности пользователю обрабатывать на том же компьютере открытой информации с меньшими требованиями к безопасности ее обработки. Все сказанное справедливо и при реализации ролевой модели контроля доступа, когда на одном и том же компьютере один пользователь может выполнять несколько ролей при различных для них требованиях к безопасности обработки информации.

7.1.Альтернативные возможности задания сессии пользователя

Итак, разный уровень конфиденциальности информации (в общем случае различные роли) требует кардинально различающихся режимов ее обработки (используемых устройств, приложений и т.д.), а пользователь один. Таким образом, в данной постановке задачи защиты необходимо разграничивать уже режимы обработки информации на одном компьютере для одного и того же пользователя (а не доступ к ресурсам между различными пользователями).

Таким образом, в общем случае правильнее здесь уже говорить не о разграничительной, а о разделительной политике доступа к ресурсам [2].

Под *разделительной политикой доступа к ресурсам* будем понимать реализацию защищенной обработки информации в рамках различных ролей (в частности информации различных уровней конфиденциальности) одним и тем же пользователем на одном и том же компьютере, при этом задача разделительной политики доступа к ресурсам состоит в формировании и изолировании различных режимов обработки информации.

Поскольку в различных режимах обработку информации должен осуществлять один и тот же пользователь, введем понятие «Сессия пользователя».

Под *сессией*пользователя будем понимать работу пользователя в одном из разрешенных для него режиме обработки информации.

Под*сессионным контролем доступа*будем понимать реализацию разделительной политики доступа к ресурсам между сессиями одного и того же пользователя.

Следствие. Субъектом доступа при реализации разделительной политики доступа к ресурсам должна выступать сущность «Сессия».

Соответственно, возникает вопрос, а как задать сущность сессия для реализации разделительной политики доступа?

Так как мы говорим о контроле доступа к ресурсам, то в общем случае без включения в разграничительную политику каких-либо дополнительных сущностей мы имеем следующие возможности для задания сессии субъектом доступа (учетная запись пользователя, процесс (имя процесса)), объектом доступа.

Лемма 21.Сессия должна быть задана (выбрана) до начала обработки данных пользователем. Данные должны получаться пользователем из файлового объекта только после того, как определен режим их обработки и загрузки, т.е. сессия.

Доказательство. При невыполнении данного требования имеем следующее противоречие. Если сессия определяется тем, какие данные считаны (например, если считываются конфиденциальные данные, то вступает в силу «конфиденциальная» сессия – режим обработки конфиденциальных данных), то необходимо в начальный момент времени (сессия не определена) разрешить пользователю чтение данных всех категорий, к которым он допущен в рамках своей служебной деятельности. Однако при этом не могут быть корректно реализованы требования к заданию различных режимов обработки данных различных категорий: невозможно задать для данных различных категорий, каким процессом, с какими привилегиями пользователя, с какого

устройства, с какими правами доступа к системным ресурсам и т.д. они могут быть прочитаны. Следовательно, для реализации данного условия режимы обработки (в частности по доступу к данным различных категорий конфиденциальности) до выбора сессии должны совпадать, что противоречит требованию, сформулированному в Лемме 21.

Лемма доказана.

Сказанное проиллюстрировано на рис.15.

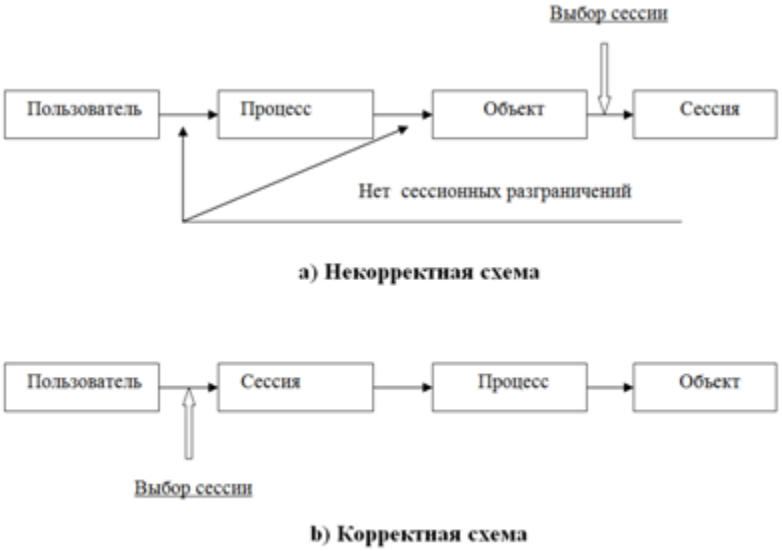


Рисунок 15.Альтернативные схемы выбора сессии

Таким образом, сущность «объект доступа» для задания сущности сессия отпадает. Сущность «процесс» не может с этой целью использоваться в общем случае по причине того, что в этом случае в различных сессиях должны использоваться различные процессы (приложения). Без включения в систему каких-либо дополнительных виртуальных (не используемых системой и приложениями) сущностей остается единственная возможность задания сессии: учетной записью пользователя.

7.2.Реализация контроля доступа с заданием сессий учетными записями

При реализации данного способа задания сущности сессия решение состоит в следующем. Поскольку один и тот же пользователь должен работать в различных сессиях, то для каждого пользователя должно быть создано несколько учетных записей: по одной для работы в каждой разрешенной для него сессии (соответственно для каждой роли). Пользователь для обработки данных в соответствующем режиме должен войти в систему под соответствующей учетной записью (при необходимости смены сессии осуществить штатными средствами смену учетной записи). При этом может быть обеспечена изолированность обработки данных в различных режимах (посредством реализации изолированной обработки данных для различных учетных записей), а сессия выбирается пользователем до получения доступа к данным, т.к. до прохождения процедуры идентификации и аутентификации пользователя (до выбора им сессии), доступ к какому-либо объекту невозможен.

Получаем корректную схему задания и выбора сессии, проиллюстрированную на рис.16.

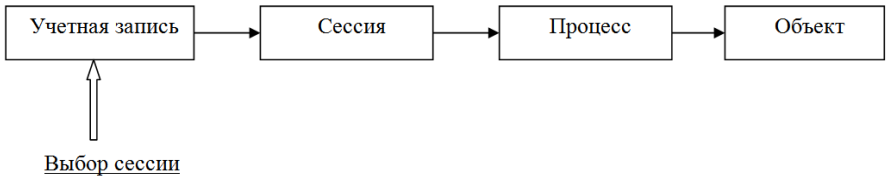


Рисунок 16.Схема выбора сессии при ее задании учетной записью

С целью реализации разделительной политики доступа в рамках сессионного контроля целесообразно использовать мандатный метод контроля доступа, основанный на применении меток безопасности (мандатов). В порядке замечания отметим, что ранее мы строго доказали, что корректно мандатный контроль доступа реализуется при условии использования неиерархических меток безопасности, а «непротиворечивое» правило мандатного контроля состоит в сравнении меток субъекта и объекта доступа исключительно на равенство/неравенство с предоставлением запрошенного субъектом доступа к объекту при совпадении значений их меток безопасности. Мандатный контроль доступа используется для изолирования обработки информации в различных сессиях.Различным пользователям (их соответствующим учетным записям) для работы в одной и той же сессии присваивается одна и та же метка – реализуется разделительная политика доступа.

В дополнение к мандатному методу для разграничения (уже разграничения, а не разделения) прав доступа между различными пользователями при их работе в одной и той же сессии (метки безопасности соответствующих учетных записей различных пользователей в данном случае совпадают) используется метод дискреционной контроля доступа.

Принципиальным достоинством данного решения является то, что вся разграничительная политика современных операционным систем основана на реализации разграничений прав доступа субъектов к объектам именно для учетных записей, причем многие из подобных разграничений установлены в системе «по умолчанию», например, разграничение доступа к буферу обмена. Это позволяет говорить о потенциальной возможности корректной реализации разделительной политики для сессионного контроля доступа при реализации рассматриваемого решения. Вместе с тем и в этом случае в системе могут присутствовать не разделяемые операционной системой и приложениями объекты в первую очередь файловые (так называемые объекты коллективного доступа). Например, это папки для временного хранения файлов. Наличие подобных неразделяемых объектов при реализации сессионного контроля доступа не допустимо (это аксиома, т.к. в противном случае теряется весь смысл реализации разделительной политики доступа), поскольку для корректного функционирования системы и приложений необходимо разрешить доступ всем учетным записям (что, к слову сказать, принципиально невозможно при использовании мандатного метода контроля доступа — какую метку безопасности присвоить подобному объекту?),— а это не что иное, как канал утечки конфиденциальной информации (в одном режиме информацию в подобном объекте пользователь может сохранить, в другом –прочитать).

Для обеспечения корректности реализации сессионного (в том числе и мандатного) контроля доступа должен использоваться рассмотренный ранее метод разделения файловых объектов, не разделяемых системой и приложениями.

Для каждого субъекта доступа (в данном случае для учетной записи) для неразделяемого объекта создается соответствующий собственный объект и реализуется перенаправление к нему запроса доступа. Проиллюстрируем сказанное. Для некоего каталога «Общий ресурс» (любой файловый объект, не разделяемый между учетными записями системой либо каким-либо приложением, каталог или файл) заводятся каталоги «Общий ресурс 1» для первого пользователя (в нашем случае учетной записи), «Общий ресурс 2» для второго пользователя и т.д. При записи информации в неразделяемый объект «Общий ресурс» (соответственно, чтении из данного объекта) средство защиты перенаправляет запрос доступа в (из) соответствующий объект того пользователя, который запросил соответствующий доступ к рассматриваемому объекту. Например, если текущим пользователем является первый пользователь, то при сохранении информации в объект «Общий ресурс», данная информация будет перенаправлена и сохранена в объекте «Общий ресурс 1».

Замечание. Механизм перенаправления запросов к неразделяемым объектам должен обрабатывать запрос перед механизмом контроля (разграничения) доступа, который, в свою очередь, уже должен реализовывать контроль доступа на основе матрицы (дискреционный) или меток безопасности (мандатный). Средствами механизма контроля доступа к объектам файловой системы разграничиваются права доступа к объектам, в том числе и к тем, в которые перенаправляется запрос доступа, например, доступ к каталогу «Общий ресурс 1» разрешается только первому пользователю. В результате применения данного механизма обеспечивается возможность полного разделения в системе объектов файловой системы для учетных записей (каталогов и файлов), как следствие, возможность корректно реализовать сессионный контроль доступа в общем случае.

Следствие. Обязательным требованием к корректной реализации сессионного контроля доступа, выполнение которого позволяет построить безопасную систему, является реализация в системе защиты метода разделения файловых объектов, не разделяемых системой и приложениями.

Принципиальным же недостатком рассмотренного решения является сложность администрирования, в том числе возрастающая с увеличением числа задаваемых учетных записей. Для каждого пользователя должно быть задано несколько учетных записей, число которых определяется числом сессий, в которых может работать этот пользователь.

Данный недостаток рассматриваемого способа реализации сессионного контроля доступа в полной мере устраняется при применении рассмотренных ранее методов контроля доступа к создаваемым файлам. Практическая реализация мандатного метода контроля доступа к создаваемым файлам описана в [15], дискреционного — в [11].Суть данных методов состоит в исключении сущности объект доступа из разграничительной (соответственно, и из разделительной) политики доступа. Например, метки безопасности при реализации разделительной политики доступа следует назначать только учетным записям (нет необходимости присваивать метки объектам). Создаваемый при этом файл наследует (метка записывается в качестве атрибута файла) метку безопасности учетной записи, под которой он создан. При последующем обращении к размеченному подобным образом файлу сравниваются метки безопасности учетной записи, запрашивающей доступ к размеченному файлу, и файла, к которому запрошен доступ [15]. Аналогично реализуется и дискреционный метод контроля доступа к создаваемым файлам, при этом созданным файлом наследуется

учетная информация пользователя (имя пользователя и имя процесса), создавшего файл [11].

7.3.Реализация контроля доступа с использованием виртуальной сущности «сессия»

Трудно сказать по какой причине, возможно, с целью решения задачи упрощения администрирования сессионного контроля доступа (как увидим далее, эта задача на деле не решается), на практике все чаще сессионный контроль доступа реализуется с включением в разграничительную (разделительную) политику виртуальной сущности «сессия».

Идея подобного подхода к реализации сессионного контроля доступа состоит в следующем. Для работы одного пользователя в различных сессиях используется одна и та же учетная запись. Вместе с тем вводится дополнительная сущность «сессия», для которой и реализуется разделительная (разграничительная) политика доступа к объектам. При входе в систему пользователю после прохождения им идентификации и аутентификации предлагается выбрать (при необходимости сменить) текущую сессию, что реализуется соответствующей программой выбора сессии (только она может быть запущена пользователем при входе в систему). После выбора сессии пользователю разрешается доступ к объектам с правами, заданными для выбранной им сессии.

Здесь также имеем корректную схему задания и выбора сессии, проиллюстрированную на рис.17.

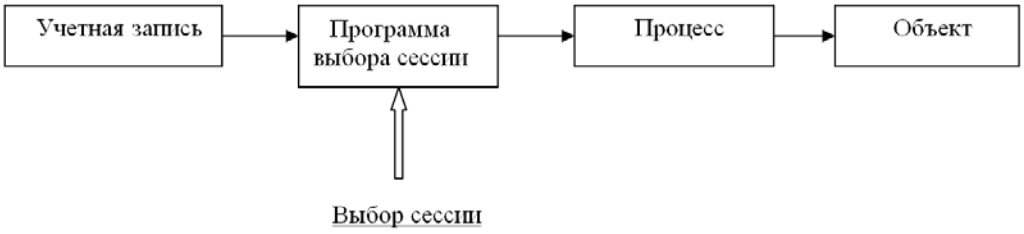


Рисунок 17.Схема выбора сессии с использованием программы выбора сессии

Лемма 22.При реализации сессионного контроля доступа с включением виртуальной сущности «сессия» субъект доступа в разделительной и разграничительной политиках должен задаваться двумя сущностями: сессия, учетная запись пользователя.

Доказательство. Если данное требование не выполняется, вся разделительная и разграничительная политики реализуются для субъекта доступа «сессия» (используется только одна сущность), то в любой момент времени защищаемый компьютер может эксплуатироваться только в одном режиме: в одной сессии. По сути, при этом современная многопользовательская система сводится к однопользовательской. Подобную реализацию защиты никак нельзя считать допустимой, поскольку это принципиально усекает функциональные возможности современного системного средства. Лемма доказана.

Итак, если систему защиты в данном случае строить корректно (не сводя систему к однопользовательской), субъектов доступа в разграничительной политике становится ничуть не меньше, чем в случае задания сессии учетной записью.

Остановимся на ключевых недостатках данного подхода к реализации метода сессионного контроля доступа.

1) **Механизмами защиты операционной системы и приложений (в том числе реализованных «по умолчанию» без настройки соответствующих механизмов защиты) разделительная политика доступа между сессиями не реализуется.**

К чему это приводит? К необходимости реализации в средстве защиты достаточно большого дополнительного набора механизмов защиты (например, контроль доступа к тому же буферу обмена) с возможностью их настройки: задания разграничений доступа к объектам для субъекта доступа сессия.

Замечание. Конечно, можно, как отмечали, свести систему к однопользовательской (одновременно разрешать на компьютере работу только в одной сессии), с реализацией ряда дополнительных процедур при смене сессии (очистка буфера обмена, освобождаемых областей оперативной памяти, принудительное завершение всех процессов и т.д., которые, кстати сказать, также надо реализовать в средстве защиты). Однако подобные решения, на наш взгляд, недопустимы (например, как в этом случае реализовать защиту сервера, на котором одновременно должны присутствовать различные режимы обработки, или опять же разрешить только одну сессию?).

2) **Основу модели безопасности современных операционных систем составляет не только реализация разграничительной политики доступа учетных записей к ресурсам, но и наделение пользователей (учетных записей) привилегиями (например, использование перекрестных ссылок и т.д.), круг которыхна самом деле достаточно широк.**

Естественно, что при реализации работы пользователя под одной учетной записью в различных сессиях различные привилегии пользователя при работе в различных сессиях устанавливаться не могут. На наш взгляд, это не допустимо, средства защиты должны усиливать безопасность операционных систем и приложений, а не наоборот!

3) **Главный недостаток же состоит в принципиальной невозможности корректной реализации разделительной политики доступа, т.е. сессионного контроля доступа в общем случае.**

Напомним, что для альтернативного решения, рассмотренного выше, обоснована необходимость использования метода разделения файловых объектов, не разделяемых между учетными записями системой и приложениями. Подобных объектов в системе не так уж и много, поэтому их разделение средством защиты достаточно просто решаемая на практике задача. В данном же случае речь идет о разделении между сессиями файлов, записываемых приложениями в процессе их работы под одной и той же учетной записью. Данная задача принципиально также может быть решена с использованием метода разделения файловых объектов с соответствующим заданием субъекта доступа (сессия, учетная запись), но вопрос в другом. Сколько и каких файлов создается и модифицируется в процессе работы приложениями, и какими приложениямипод учетной записью пользователя? А ведь каждый подобный файл –это «канал» утечки информации. Не должно быть в системе файлов, в которые разрешена запись приложениям, не разделенных между сессиями, что, как отмечали, является основополагающим требованием к корректности реализации сессионного контроля доступа, выполнение которого является обязательным при построении безопасной системы! Подобные объекты не то, чтобы разделить, а элементарно выявить в системе в полном объеме уже является большой проблемой!

Проиллюстрируем сказанное простым, но достаточно показательным примером. Идентификационные данные доменов хранятся в файле cookies. Данный файл по

умолчанию разделен между учетными записями. При работе в различных сессиях (открытая и конфиденциальная) одним и тем же пользователем (под одной учетной записью)Интернет-браузером будет использоваться один и тот же файл cookies, формируемый приложением как в открытой, так и в конфиденциальной сессиях. Как следствие, при работе в открытой сессии становится возможным получить доступ к идентификационным данным доменов, созданных в конфиденциальной сессии. И это далеко не единственный пример. Задумайтесь над принципиальной возможностью и трудоемкостью задачи элементарного выявления таких объектов с целью реализации корректной разделительной политики, а уж тем более их последующего разделения между сессиями.

Все сказанное позволяет нам сделать важнейший вывод.

Вывод.Сессионный контроль доступа может быть в общем случае корректно реализован, соответственно, построена безопасная система, только при задании сессий учетными записями пользователя. При этом сложность реализации разделительной и разграничительной политик доступа при таком решении намного ниже, чем в случае использования для задания сессий виртуальной сущности «сессия».

8. Примеры практической реализации методов контроля доступа и решаемых ими задач защиты

В данном разделе учебного пособия рассмотрим примеры практической реализации методов контроля доступа, позволяющей реализовать сформулированные требования в части построения безопасной системы; рассмотрим примеры использования методов контроля доступа для решения актуальных современных задач защиты информации. Отметим, что все рассматриваемые в учебном пособии технические решения реализованы и апробированы, в том числе приведенные разграничительные политики доступа в рамках построения комплексной системы защиты информации «Панцирь+» для ОС MicrosoftWindows.

8.1.Реализация методов контроля доступа к создаваемым файлам

8.1.1. Реализация мандатного метода контроля доступа

Как отмечали, важнейшим достоинством данного метода является простота администрирования, практически сводящая на нет вероятность ошибки администрирования средства защиты, что позволяет его использовать, в том числе для реализации разделительной политики доступа в рамках построения сессионного контроля доступа к файловым объектам.

Для иллюстрации кардинального упрощения задачи администрирования рассмотрим всю процедуру настройки мандатного контроля доступа к создаваемым файлам[15].

Прежде всего, требуется завести пользователей в средстве защиты/в системе. Пользователи либо заводятся из интерфейса средства защиты, приведенного на рис.18, 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.18, что крайне важно с точки зрения реализации эффективной защиты. Естественно, для системных пользователей пароль установить нельзя.

В результате заведенные в системе и в средстве защиты пользователи отображаются в окне интерфейса средства защиты, в виде, приведенном на рис.19. Системные пользователи отображаются черным цветом, пользователи с установленным

паролем для входа в систему – зеленым, пользователи, пароль которым не установлен – красным,их вход в систему невозможен.

После заведения пользователей из меню, приведенного на рис.20, задаются уровни доступа (мандатные уровни). Уровни задаются как количественным значением меток безопасности (мандатов), которые собственно и сравниваются диспетчером доступа, так и их смысловой транскрипцией.

В результате заведенные в системе уровни доступа отображаются в виде, представленном на рис.21.

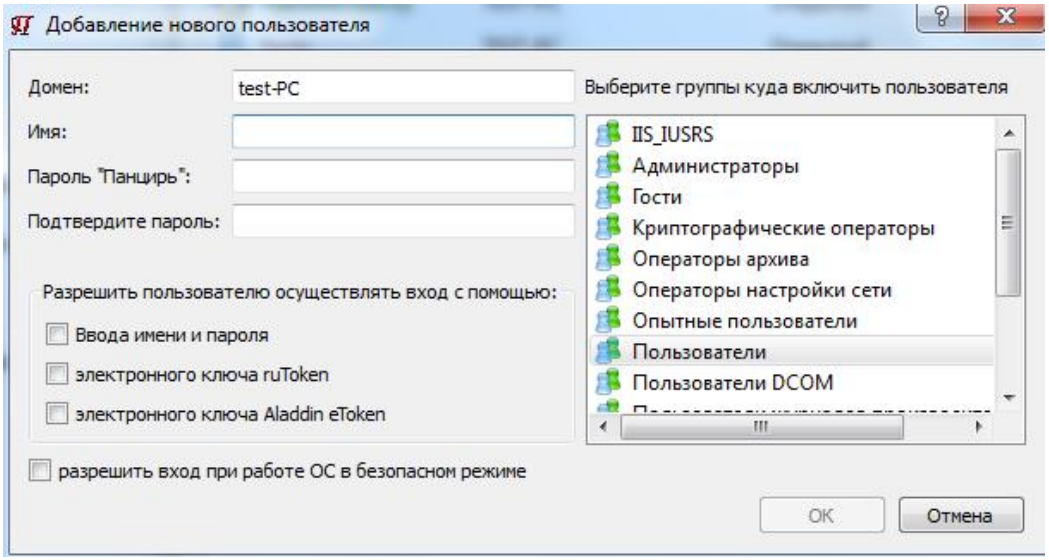


Рисунок 18.Интерфейс задания пользователей в средстве защиты

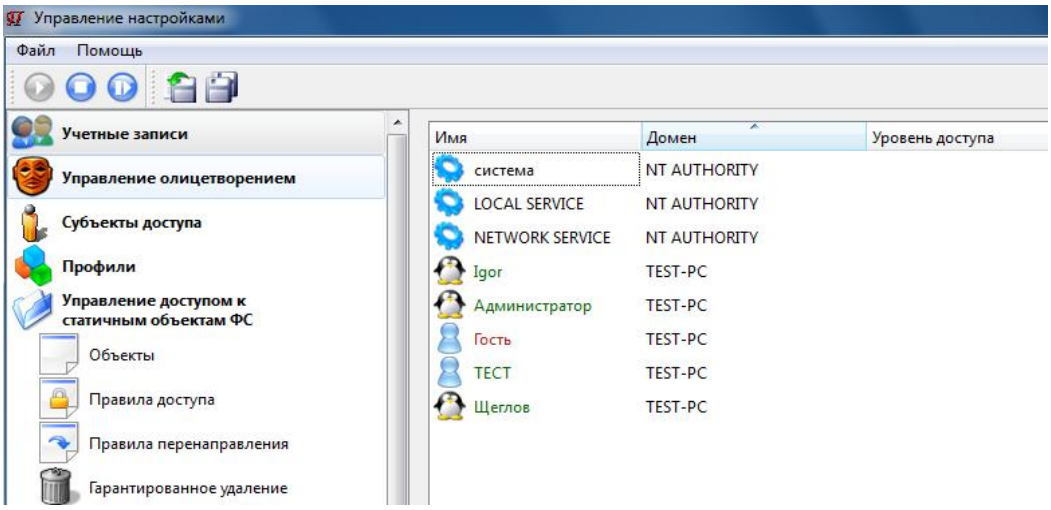


Рисунок 19.Отображение созданных в системе пользователей

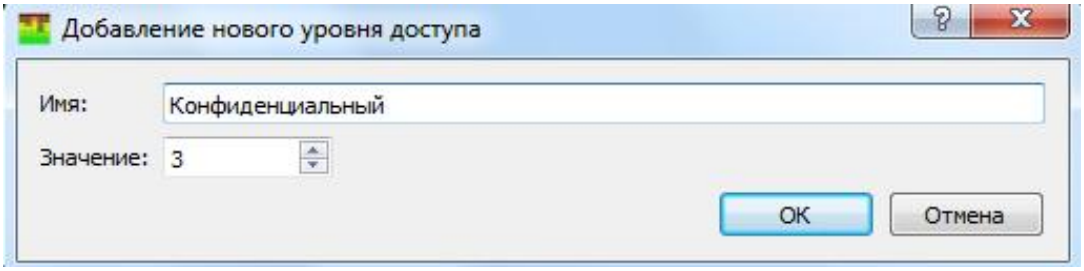


Рисунок 20.Меню задания уровней доступа (мандатных уровней)

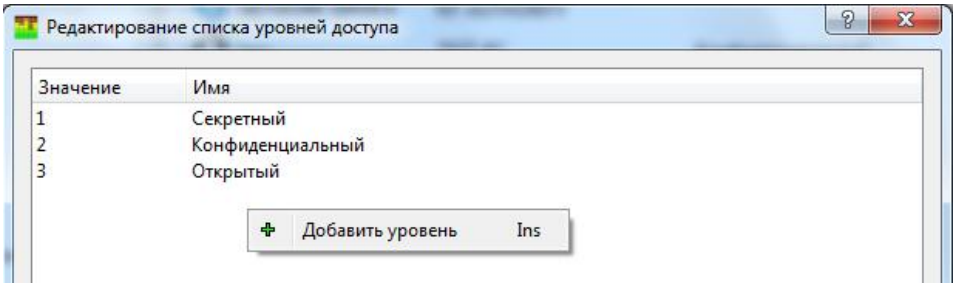
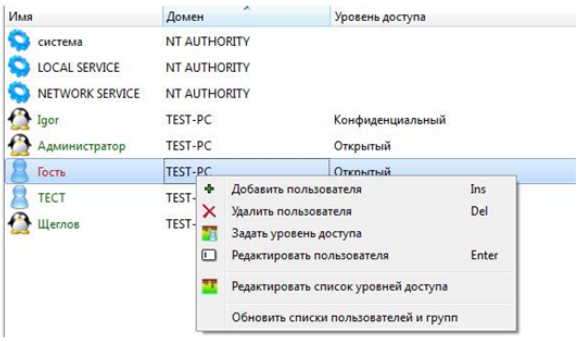


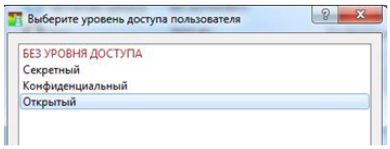
Рисунок 21.Отображение заданных в системе уровней доступа

(мандатных уровней)

Далее для тех пользователей, доступ к создаваемым файлам которыми должен контролироваться, сопоставляются метки безопасности. Им присваивается уровень доступа, см. рис.22 а), уровень доступа назначается из созданного до этого списка, см. рис.22 б).



а) Меню задания мандатного уровня для выбранного пользователя



б) Меню выбора мандатного уровня из заданного списка

Рисунок 22.Задание пользователям уровней доступа (мандатных уровней)

Пользователи с заданными для них уровнями доступа отображаются в интерфейсе в виде, приведенном на рис.23.

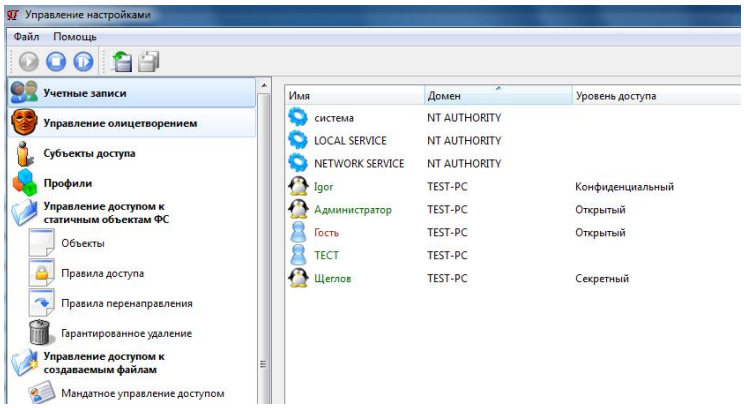


Рисунок 23.Отображение в интерфейсе заданных в системе пользователей с заданными для них уровнями доступа (мандатными уровнями)

Напомним, что будет происходить при назначении пользователям уровней доступа[15]. При создании файла контролируемым пользователем (пользователем, которому присвоен уровень доступа) создаваемым файлом автоматически будет наследоваться уровень доступа (метка безопасности) создавшего его пользователя. Файл будет автоматически размечаться диспетчером доступа: в атрибутах создаваемого файла им будетсохраняться мандатный уровень (метка безопасности)создавшего файл пользователя. Наследование будет происходить и в том случае, если контролируемый пользователь не создает новый, а модифицирует неразмеченный ранее файл.

При последующем обращении к контролируемому файлу (к размеченному файлу) доступ к нему будет контролироваться: разрешаться/запрещаться в соответствии с заданными правилами мандатного доступа. Доступ к контролируемому файлу от неконтролируемого пользователя, которому не присвоен уровень доступа, будет запрещен.

Настройка правил мандатного доступа осуществляется из интерфейса, приведенного на рис.24. В окне интерфейса, приведенного на рис.24, могут быть заданы следующие правила доступа: возможность чтения/записи пользователем файла одного с ним уровня и возможность чтения пользователем файла более низкого уровня доступа (категории).

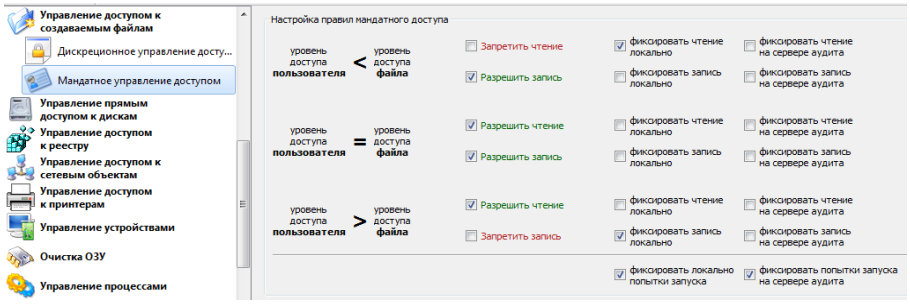


Рисунок 24.Интерфейс настройки правил мандатного доступа

Выбор (задание) правил, см. рис.24, позволяет использовать как иерархические, так и неиерархические метки безопасности.

Замечание. Ранее в пособии мы обосновали, что «непротиворечивая модель мандатного контроля доступа» предполагает реализацию неиерархических меток безопасности. Неиерархические метки безопасности должны назначаться и при построении метода сессионного контроля доступа в части реализации разделительной политики между сессиями, назначаемыми учетными записями пользователей. Также неиерархические метки безопасности с соответствующей их смысловой транскрипцией могут использоваться при реализации ролевой модели контроля доступа.

Как отмечали, правила мандатного контроля доступа задаются для двух основных типов доступа: чтение и запись (разрешение записи включает разрешение модификации/удаления/переименования).

Вот и все требуемые настройки! Больше ничего не нужно настраивать, а главное, не требуется каким-либо образом размечать файловые объекты (назначать им метки безопасности), создавать папки для последующего сохранения в них пользователями файлов. Вся разметка создаваемых в процессе работы файлов (т.е. тех файловых объектов, к которым контролируется доступ с использованием меток безопасности – мандатов) производится автоматически средством защиты.

Для возможности обзора произведенной в процессе работы системы разметки файлов, созданных контролируемыми субъектами (пользователями, которым присвоены метки безопасности), в состав средства защиты включена специальная утилита обзора разметки созданных файлов. С ее использованием, выбрав соответствующий файловый объект в программе-проводнике, администратор может посмотреть, какие права унаследованы файлом при его создании (отображается имя пользователя, создавшего файл, – учетная запись, его уровень доступа), см. рис.25. Используя данную утилиту, администратор может удалить разметку либо создать необходимую ему разметку вручную, включив тем самым файл (в том числе и системный) в мандатную схему контроля доступа.

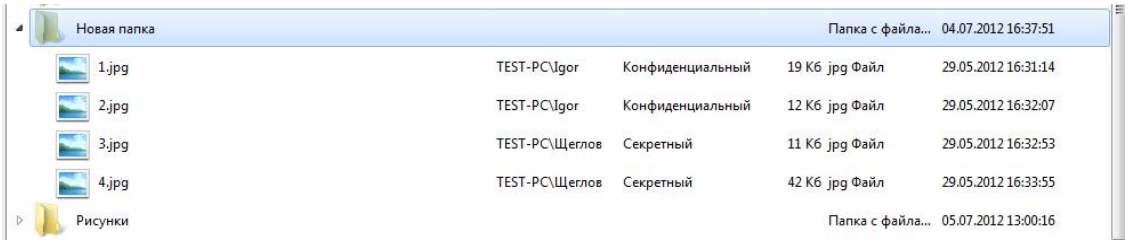


Рисунок 25.Отображение разметки созданных контролируемыми пользователями файлов при мандатном контроле доступа

8.2.Реализация дискреционного метода контроля доступа

Субъекты доступа в общем случае (что, как отмечали ранее, является одним из важнейших требований к реализации дискреционного контроля доступа в современных условиях) назначаются тремя сущностями: исходное имя (SID) пользователя, эффективное имя пользователя, имя процесса – полнопутевоеимя исполняемого файла процесса, из интерфейса, представленного на рис.26 [11]. При задании субъекта доступа могут использоваться маски и переменные среды окружения. Например, маска «*» (все или любой) позволяет не учитывать отдельные сущности субъекта доступа в разграничительной политике (например, если маску «*» указать в имени процесса субъектов доступа, то разграничительная политика будет применяться только в

отношении пользователей). Например, маской «E:\ProgramFiles*» можно задать все исполняемые файлы (процессы), запускаемые из этой папки, маской «*.exe» — все исполняемые файлы с расширением «exe»и т.д.

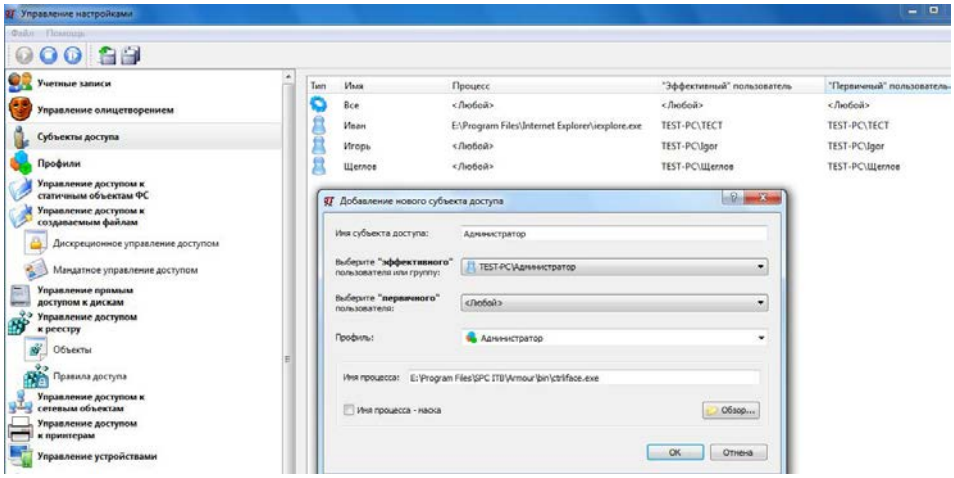


Рисунок 26.Задание и отображение в интерфейсе субъектов доступа

Основной принцип контроля доступа к создаваемым файлам, как отмечали, состоит в исключении сущности «объект доступа» из разграничительной политики как таковой (ввиду ее отсутствия на момент задания администратором правил доступа субъектов к объектам). Разграничение (контроль доступа) реализуется непосредственно между субъектами доступа к создаваемым ими файлам.

Реализуется контроль доступа следующим образом [11]. При создании субъектом нового файла средством контроля доступа (диспетчером доступа) создаваемый файл автоматически размечается –файлом наследуется учетная информация субъекта доступа (определяемая соответствующими тремя сущностями), создавшего этот файл. Данная информация размещается диспетчером доступа в атрибутах созданного файла.

При запросе же доступа к любому файлу диспетчер доступа анализирует наличие, а при наличии– содержимое унаследованной файлом учетной информации создавшего его субъекта доступа. Это осуществляется посредством считывания и анализа атрибутов файла, к которому запрошен доступ.

В соответствии с заданными администратором правилами разграничения доступа, которые назначаются исключительно между субъектами доступа из интерфейса, см. рис.27, диспетчер доступа предоставляет запрошенный субъектом доступ либо отказывает в нем, признавая тем самым запрос доступа несанкционированным.

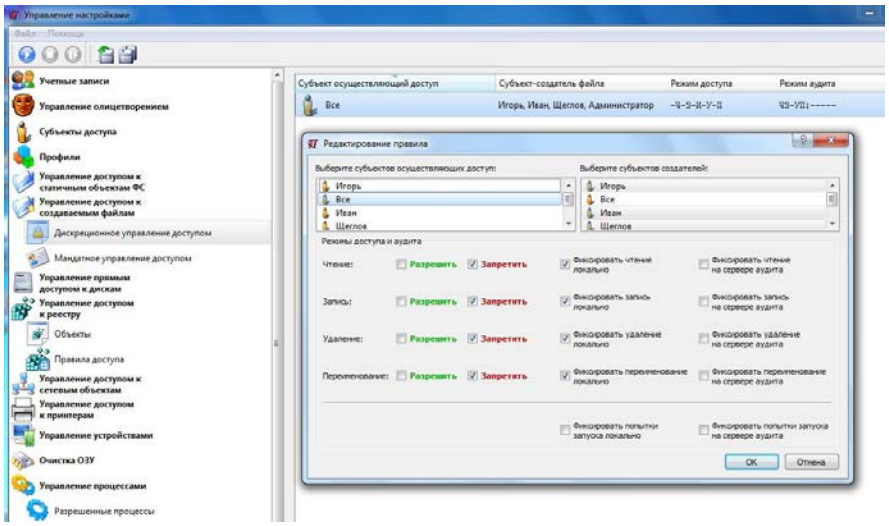


Рисунок 27.Задание и отображение в интерфейсе правил доступа к создаваемым файлам

В двух словах скажем о назначении правил доступа. В правом столбце интерфейса «Выберите субъектов создателей», см. рис.27, задаются те субъекты (из заведенных администратором ранее из интерфейса, представленного на рис.26), последующий доступ к файлам, создаваемым которыми, будет контролироваться (разграничиваться). Для каждого заданного в правом столбце интерфейса субъекта, в левом столбце интерфейса – «Выберите субъектов, осуществляющих доступ», задаются субъекты, которым разрешается доступ к файлам, создаваемым заданным субъектом-создателем, и назначаются права доступа к этим файлам (чтение, запись, переименование, удаление), а так же режимы аудита.

Замечания.

1) По умолчанию всем пользователям запрещено исполнение создаваемых контролируемым субъектом файлов (субъектом, определенным как субъектом создателем, см. рис.27), поэтому соответствующее правило доступа не выносится в интерфейс настройки разграничительной политики доступа.
2) По умолчанию разрешен полный доступ (чтение, запись, удаление, переименование) субъектов к создаваемым ими же файлам (этим реализуется каноническая матрица доступа, расширяемая заданием последующих правил). Поэтому при выборе одноименных субъекта-создателя и субъекта, осуществляющего доступ, см. рис.27, откроется интерфейс с усеченными возможностями настройки, предоставляющий возможность установить только режим аудита доступа.

Администратор с использованием соответствующей утилиты в данном случае также может посмотреть произведенную в процессе работы системы разметку файлов, созданных контролируемыми субъектами, — учетную информацию (имя пользователя с учетом возможного олицетворения и имя процесса, создавшего файл), см. рис.28.

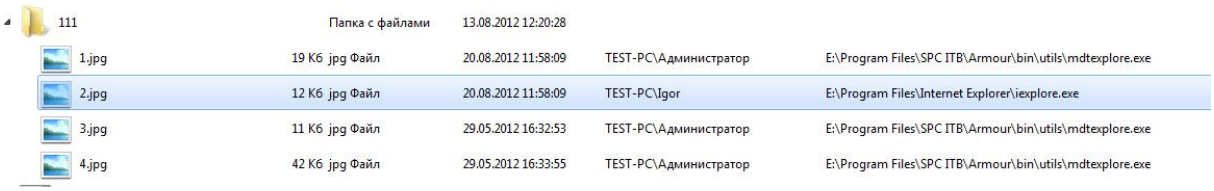


Рисунок 28.Отображение разметки созданных контролируемыми пользователями файлов при дискреционном контроле доступа

Для использования данного механизма защиты с целью реализации контроля доступа к статичным (системным) файлам предусмотрена возможность ручной разметки статичных файлов администратором. Эта задача администрирования решается при помощи специальной утилиты, интерфейс которой представлен на рис.29.

Для осуществления разметки администратору требуется выбрать в интерфейсе размечаемый им файловый объект (диск, каталог или файл). Как ранее говорили, имеет смысл размечать каталоги Windows и ProgramFiles, и задать учетную информацию субъекта (имя пользователя при необходимости и имя процесса – имя его исполняемого файла), которая будет унаследована размечаемыми администратором файлами (соответственно, всеми файлами, располагаемыми в выбранном каталоге либо на выбранном диске). В качестве имени пользователя, используемого при разметке статичных (системных) файлов, как отмечали, имеет смысл использовать имя учетной записи администратора (ведь именно им создаются на компьютере системные файлы), в качестве имени процесса при этом можно использовать маску «*» (любой).

Настройкой разграничительной политики доступа из интерфейса, см. рис.27, следует предоставить возможность только чтения и исполнения всеми интерактивными пользователями файлов, созданных администратором. Администратор, имеющий при этом полный доступ к системным объектам, сможет их модифицировать. При создании же администратором новых системных объектов (например, при установке новых приложений), их файлами будет наследоваться учетная информация администратора, создавшего эти файлы.

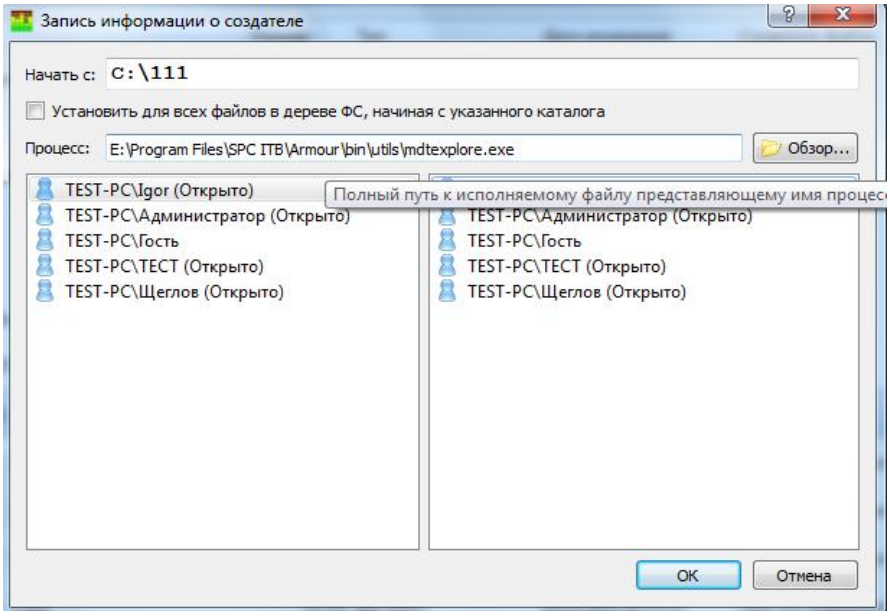


Рисунок 29.Интерфейс утилиты ручной разметки статичных файлов

В журнале аудита в зависимости от заданных правил аудита (фиксируются все события либо только отказы запрашиваемого доступа) запоминаются и отображаются соответствующие события. Пример журнала для мандатного контроля доступа с отображением в нем попыток несанкционированного доступа к файлам приведен на рис.30.

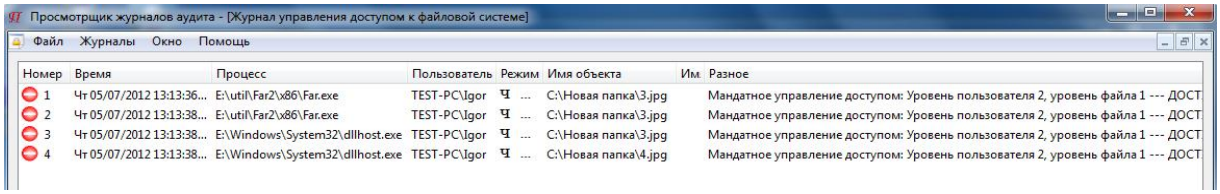


Рисунок 30.Отображение попыток несанкционированного доступа

в журнале аудита

При необходимости, используя интерфейс настройки механизма защиты, представленный на рис.31, можно исключить из разметки требуемые файловые объекты (каталоги или файлы), при записи в них автоматической разметки файлов производиться не будет. Например, это могут быть системные каталоги, см. рис.31.

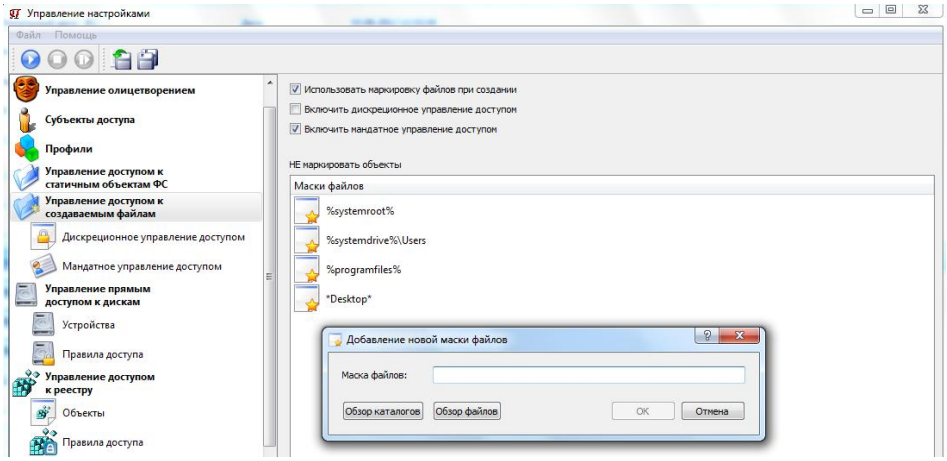


Рисунок 31. Интерфейс задания файловых объектов, исключаемых из схемы контроля доступа к создаваемым файлам

8.3.Примеры реализациираз граничительной политики доступа к создаваемым файлам

Рассмотрим несколько примеров реализации защиты (построения разграничительных политик доступа), основанной на применении метода контроля доступа к создаваемым файлам.

Прежде всего, рассмотрим реализацию защиты от запуска на компьютере вредоносных программ. При этом, естественно, угрозу внедрения на защищаемый компьютер и запуска вредоносных программ несут в себе именно создаваемые в процессе работы системы файлы.

Задача защиты от возможности запуска на компьютере создаваемых файлов решается при реализации простейшей разграничительной политики доступа к

создаваемым файлам: при простейших настройка механизма защиты, представленных на рис.32.

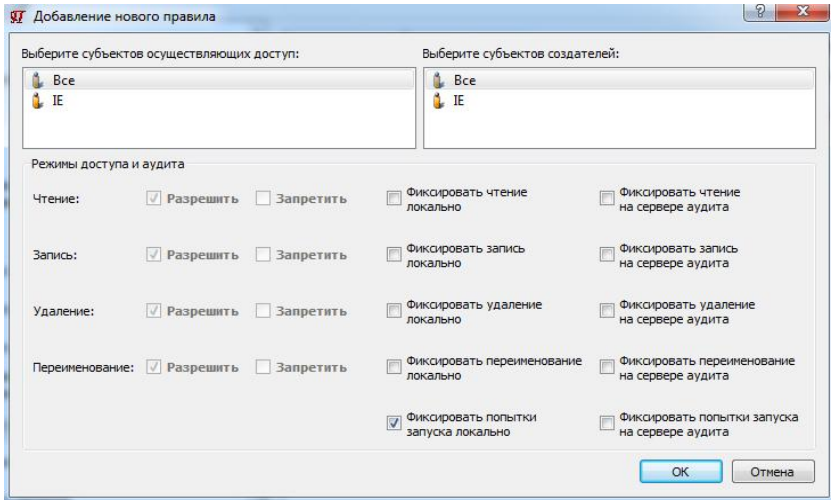


Рисунок 32. Реализация защиты от запуска вредоносных программ

Контролируется доступ к файлам, создаваемым всеми субъектами; со стороны всех субъектов для задания субъекта доступа «Все» используется маска «*». «По умолчанию» разрешается полный доступ всем субъектам ко всем создаваемым файлам, кроме исполнения создаваемых в процессе работы системы файлов,–реализуется каноническая матрица доступа (настроить в данном случае можно только правила аудита).

Замечание. Поскольку запрет на исполнение создаваемого файла установлен в средстве защиты «по умолчанию», данное правило не вынесено в интерфейс, см. рис.32.

Теперь рассмотрим реализацию защиты от атак на интернет-браузеры, эксплуатирующих уязвимости, обнаруживаемые в этих приложениях.

Актуальность этих угроз подтверждается соответствующей статистикой, что проиллюстрировали в предыдущих частях пособия.В общем случае для проведения успешной атаки с определенной конечной целью (как правило, это хищение либо модификация/удаление обрабатываемой в информационной системе информации) злоумышленнику нужен соответствующий инструмент: программа, реализующая соответствующие осуществляемой атаке возможности. Подобный инструмент может появиться у злоумышленника в двух случаях: либо посредством занесения на компьютер и последующего использования соответствующей вредоносной программы (программы, наделенной необходимыми для осуществления успешной атаки свойствами), либо посредством наделения санкционированно установленной и штатно используемой в информационной системе программы соответствующими вредоносными свойствами.

В обоих случаях мы имеем крайне актуальные угрозы.

Рассмотрим решение рассматриваемой задачи защиты посредством реализации разграничительной политики доступа к файловым объектам для потенциально уязвимого приложения, в качестве которого будем рассматривать интернет-браузер InternetExplorer (далее IE).

Отметим, что при построении защиты от атак на процессы (приложения) на основе контроля доступа (разграничительной политики доступа) здесь и далее (при решении различных задач защиты) реализуются одни и те же принципы защиты:

- – предоставить возможность доступа процесса (приложения), на который может быть осуществлена атака, только к необходимым ему ресурсам для корректного функционирования в системе;
- – обеспечить максимальное снижение последствий от успешной атаки в случае ее неминуемости на процесс (приложение) с учетом возможных (актуальных) целей потенциальных атак.

Зададим два субъекта доступа: «Все» и «IE», см. рис.33, и правила доступа, представленные на рис.34.

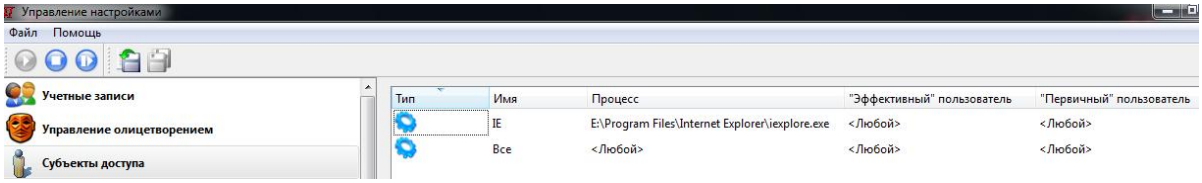


Рисунок 33.Отображение заданных субъектов доступа

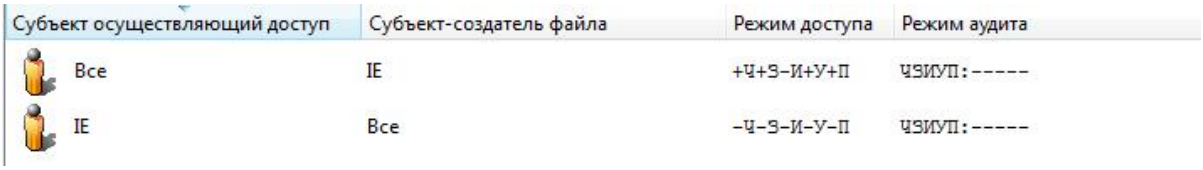


Рисунок 34. Отображение заданных правил доступа

Рассмотрим, что мы получим в результате реализации данной простейшей разграничительной политики. Интернет-браузер, вне зависимости от того, какими несанкционированными свойствами и каким образом он будет наделен, не получит доступ к конфиденциальной информации, обрабатываемой на компьютере: к создаваемым иными приложениями файлам (его работа в информационной системе в этой части изолирована), не сможет запустить созданный им файл. Предотвращается возможность нарушения конфиденциальности, целостности и доступности (в части защиты от удаления) обрабатываемой на компьютере информации в результате реализации любой известной и потенциально возможной атаки на интернет-браузер.

Иные же приложения при данной разграничительной политике имеют доступ (кроме исполнения) к файлам, создаваемым IE. На этом моменте следует акцентировать внимание. Уязвимый интернет-браузер может создать вредоносный файл (например, содержащий макро-вирус), при чтении которого иным приложением данное приложение будет наделено вредоносными свойствами. Из этого следует, что целесообразно не только запретить полный доступ браузеру к файлам, создаваемым иными приложениями, но и полный доступ иных приложений к файлам, создаваемым браузером, по крайней мере тех приложений, которые при прочтении вредоносного

файла могут быть наделены вредоносными свойствами (т.е. реализовать разделительную политику доступа в отношении потенциально уязвимого приложения).

Как видим, при решении этой актуальнейшей задачи защиты нами в полной мере реализован принцип «черного ящика». Поскольку невозможно реализовать эффективной защиты, противодействующей в общем случае наделению приложения вредоносными свойствами, реализуется разграничительная политика доступа для приложения в предположении, что оно наделено подобными свойствами, причем, исходя из того, что мы не знаем, как и каким вредоносным свойством наделено приложение, оно может быть любым. При этом в общем случае предотвращена возможность доступа этого приложения к обрабатываемой в информационной системе конфиденциальной информации.

8.4. Реализация методов контроля доступа к статичным файловым объектам

8.4.1. Реализация дискреционного метода контроля доступа к статичным файловым объектам

Как отмечали во второй главе, применительно к реализации контроля доступа (разграничительной политики доступа) к статичным файловым объектам (а это в первую очередь системные файловые объекты) имеет смысл говорить о реализации метода дискреционного контроля доступа; метод мандатного контроля доступа здесь мало применим, ввиду сложности использования меток безопасности (мандатов) для какого-либо категорирования системных объектов. Мандатный контроль доступа имеет смысл рассматривать и применять в отношении исключительно создаваемых файлов. Реализация данного метода контроля доступа была нами рассмотрена выше.

Субъекты доступа задаются из интерфейса, представленного на рис.47. Заметим, что в системе защиты создается и ведется единый список субъектов доступа для всех механизмов контроля доступа к защищаемым ресурсам, входящих в состав системы защиты. Это обусловливается тем, что доступ одного и того же субъекта может разграничиваться к различным ресурсам вычислительной системы.

Как отмечали ранее, субъекты доступа в общем случае назначаются тремя сущностями: исходное имя (SID) пользователя, эффективное имя пользователя, имя процесса – полнопутевоеимя исполняемого файла процесса, из интерфейса, представленного на рис.26 [8].Объекты доступа задаются из интерфейса, представленного на рис.35.

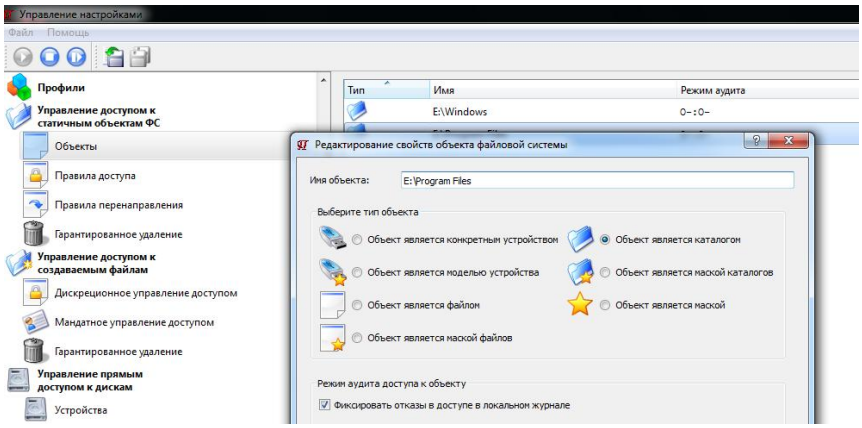


Рисунок 35. Задание и отображение в интерфейсе объектов доступа

Объект доступа в разграничительной политике может задаваться (из интерфейса, см. рис.35) как своим полнопутевым именем, так и маской, а также переменными среды окружения. Для однозначной идентификации объекта, к которому запрашивается доступ, в разграничительной политике (с целью выбора наиболее точного описателя объекта) для каждого задаваемого объекта должен быть определен его признак: файл, маска файла, каталог, маска каталога, маска, см. рис.35. Это обусловлено тем, что правила доступа субъектов к объектам хранятся не в качестве атрибутов, а в отдельном файле в виде матрицы доступа. Как следствие, при обработке запроса доступа к конкретному файлу диспетчером из матрицы доступа должен быть выбран объект, наиболее точно описывающий объект в запросе доступа, с учетом того, что при использовании масок один и тот же объект в запросе одновременно может подпадать под несколько масок файловых объектов, заданных в разграничительной политике. Это обусловливает реализацию следующего алгоритма выбора наиболее точного соответствия описателя объекта, к которому запрошен доступ, объекту, заданному в разграничительной политике. Сначала объект из запроса доступа сравнивается с объектами в разграничительной политике, определенными как «файл», см. рис.35. Если полное совпадение найдено, выбираются правила доступа, заданные в разграничительной политике для этого объекта (файла). Если полного совпадения не найдено, анализируется полное совпадение с объектами, определенными как «маска файла», затем при соответствующих условиях «каталог», затем «маска каталога», затем «маска».

Замечание. К статичным файловым объектам также относятся и файловые накопители (устройства), см.рис.35. Ключевым требованием к реализации разграничительной политики доступа к файловым накопителям является их задание в правилах доступа идентификаторами устройств, включая их серийные номера (для идентификации не модели, а конкретного устройства), и ни в коем случае не «буквой диска», к которой устройство монтируется к системе. При задании объекта доступа здесь также могут использоваться маски.

Правила доступа субъектов к объектам задаются из интерфейса, представленного на рис.36.

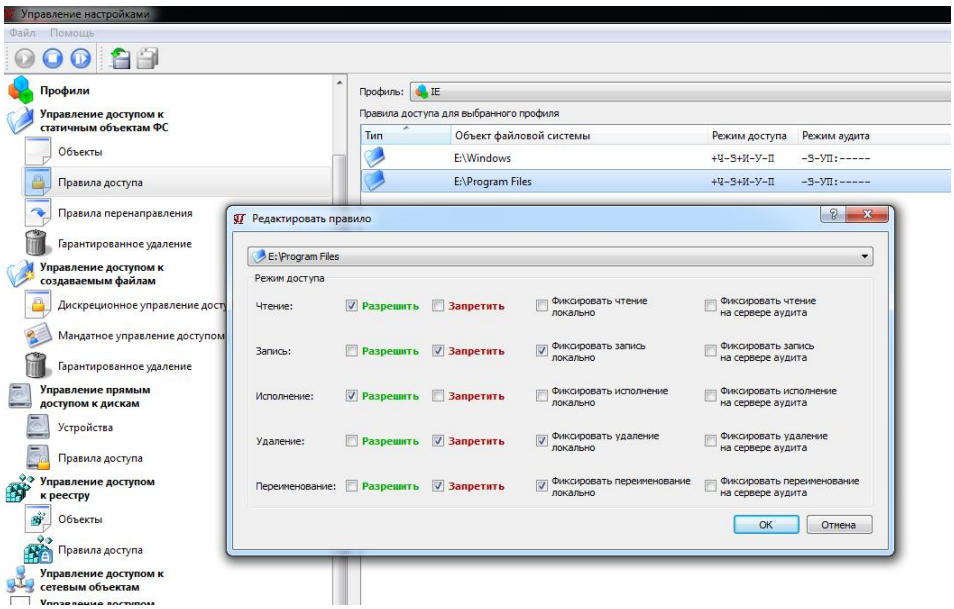


Рисунок 36. Задание и отображение в интерфейсе правил доступа субъектов к статичным файловым объектам

В двух словах собственно о разграничительной политике. Как отмечалось во второй главе, она может быть запретительной либо разрешительной. По умолчанию (без дополнительных настроек) реализована запретительная политика. Для задания разрешительной политики достаточно для любого субъекта, определяемого маской «*», запретить какой-либо доступ к какому-либо объекту; объект также при этом задается маской «*» и определяется как «маска», см. рис.35 (наименее точный описатель объекта доступа). После чего уже следует задавать разрешенные правила доступа. Все запросы доступа, не попадающие под эти разрешенные правила, будут отклоняться диспетчером доступа –реализуется общее правило разрешительной разграничительной политики доступа: все, что явно не разрешено, то запрещено.

Разрешительная политика может быть установлена применительно к какой-то отдельной разграничительной политике (не для системы в целом), например, при реализации контроля доступа к файловым объектам для какого-либо приложения. В этом случае для ее реализации сначала приложению следует запретить доступ ко всем объектам, объект доступа при этом задается маской «*», определяется как «маска», затем задать необходимые разрешения доступа к файловым объектам для данного приложения.

Естественно, что при реализации разрешительной разграничительной политики доступа возникает вопрос, каким образом определить те правила доступа, которые необходимо разрешить субъекту (в первую очередь приложению) для его корректного функционирования в системе. С этой целью используется инструментальный аудит. Для этого задается анализируемый субъект доступа, например, приложение, с учетом либо нет работающего с ним пользователя (обычный пользователь, администратор, System). В качестве объекта доступа задается тот файловый объект, доступ к которому анализируется. Если необходимо проанализировать все возможные обращения субъекта к файловой системе, объект доступа задается маской «*», определяется как «маска», см. рис.35. В интерфейсе, см. рис.36, разрешаются все права доступа заданного субъекта к объекту; для анализируемых же прав доступа, например, только запись либо только исполнение, в пределе для всех устанавливается режим аудита. При заданных настройках требуется осуществить работу приложения в требуемых режимах и соответствующим образом проанализировать результаты аудита, представленные в соответствующем журнале, см. рис.37. На основании проведенного анализа может быть реализована корректная разграничительная политика доступа, что уже не составит особого труда.

Как отмечали во второй главе, важнейшим методом, обеспечивающим корректность реализации разграничительной политики доступа к файловым объектам в общем случае, как следствие, возможность построения безопасной системы, является метод разделения между субъектами не разделяемых системой и приложениями файловых объектов за счет перенаправления диспетчером доступа запроса к неразделяемому объекту, к объекту, предварительно созданному для соответствующего субъекта администратором, чем может быть физически разделен между субъектами любой файловый объект.

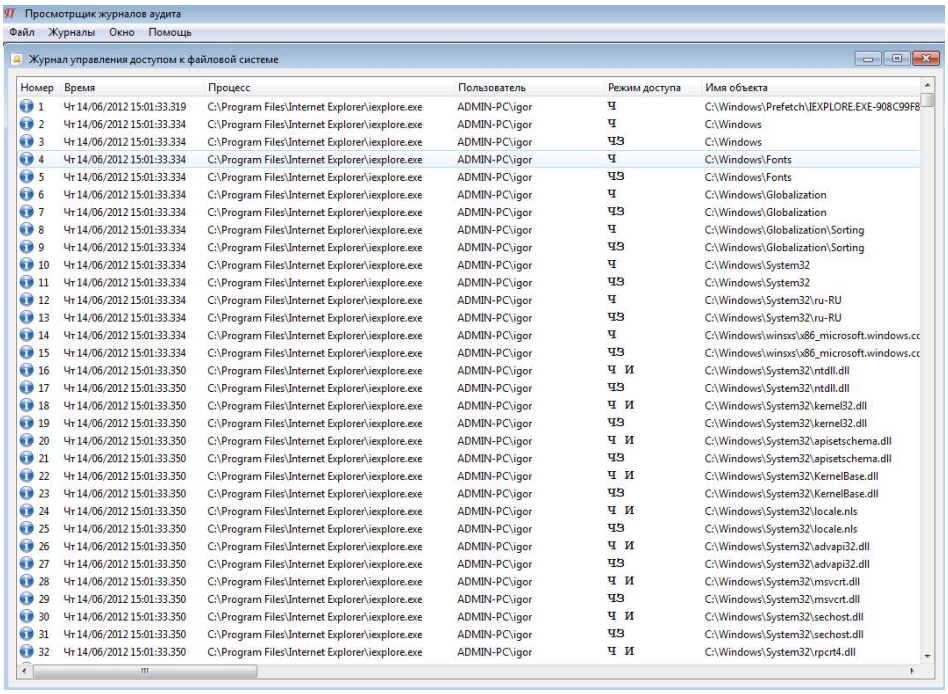


Рисунок 37.Журнал аудита запрашиваемого доступа к файловым объектам приложением InternetExplorer при его запуске

Интерфейс настройки механизма защиты, реализующего данный метод контроля доступа, представлен на рис.38.

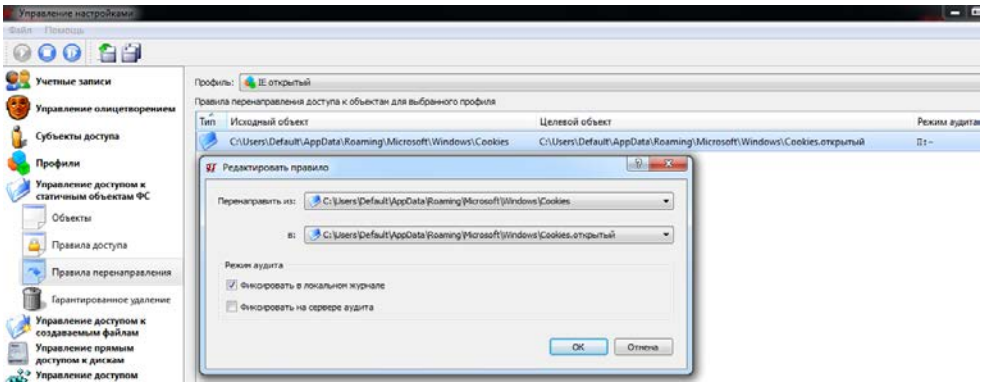


Рисунок 38. Интерфейс настройки перенаправления запроса доступа к неразделяемым файловым объектам

Здесь опять же используется субъект доступа, задаваемый соответствующими тремя сущностями из интерфейса, представленного на рис.26, как следствие, разделять файловые объекты можно как между пользователями (в том числе с учетом из возможного олицетворения), так и между процессами. В строке «Перенаправить из» интерфейса, см. рис.38, задается файловый объект, который требуется разделить между субъектами; в строке «Перенаправить в» созданный администратором для данного субъекта объект, в который будет перенаправлен соответствующий запрос доступа. Естественно, что перед настройкой механизма защиты данные объекты должны быть заданы в качестве объектов разграничительной политики доступа из интерфейса,

представленного на рис.35. К объекту, в который перенаправляется соответствующий доступ, из интерфейса, приведенного на рис.38, задаются правила доступа (перенаправление доступа срабатывает ранее, чем разграничение прав доступа). К исходному файловому объекту коллективного доступа устанавливать каких-либо правил доступа не требуется, поскольку запрос доступа до него не доходит (перенаправляется средством защиты).

8.4.2. Примеры реализации разграничительной политики доступа к статичным файловым объектам

Рассмотрим следующую задачу защиты. Пусть требуется защитить системные файловые объекты от атак со стороны потенциально уязвимого приложения, например, интернет-браузераIE.С целью реализации защиты зададим из соответствующего интерфейса в качестве объектов доступа системные каталоги и назначим к ним для субъекта доступа IE правила доступа, проиллюстрированные на рис.39.

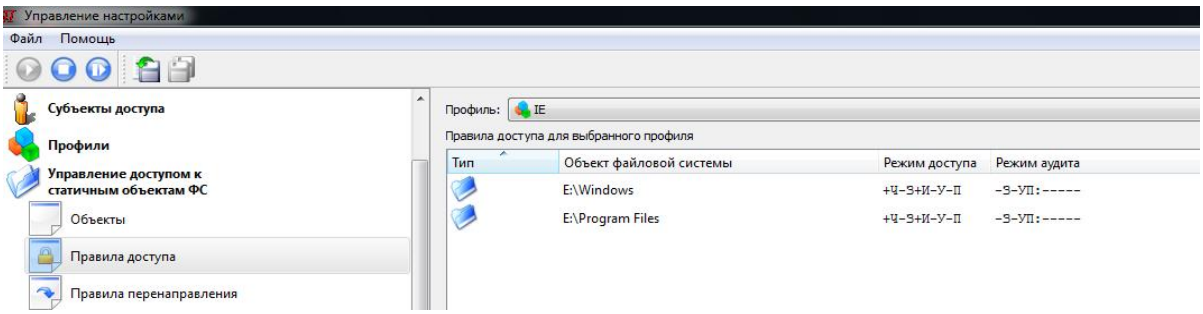


Рисунок 39. Заданные правила доступа к статичным (системным) файловым объектам

В результате подобной настройки механизма защиты (заметим, задаваемой лишь двумя записями в разграничительной политике доступа) потенциальное уязвимое приложение не сможет ни удалить, ни модифицировать, ни переименовать как собственно системные каталоги, так и находящиеся в них папки и файлы. Задача защиты решена.

Замечание. Подобным образом можно защитить системные каталоги не только от атак со стороны отдельных приложений, но и в общем случае, например, со стороны всех интерактивных пользователей, задав при этом соответствующим образом субъект доступа (субъект доступа, например, в этом случае может задаваться маской «*», а для пользователя System (при необходимости и Администратора) следует задать собственные правила доступа), для которого следует реализовать правила доступа, приведенные на рис.39.

Введем в разграничительную политику доступа к файловым объектам, приведенную на рис.39, еще одно правило. Субъекту доступа IE (либо всем интерактивным пользователям), в соответствии с рассмотренными выше примерами, запретим право «исполнение»; все остальные права доступа разрешим из всех объектов доступа (для этого сначала создадим соответствующий объект доступа, обозначив его маской «*», определив при этом, что этот объект является «маской»,– наименее точный описатель объекта). В результате задания одного этого правила мы предотвратим возможность запуска на компьютере несанкционированно установленной программы, в том числе вредоносной, поскольку исполнять файлы становится возможным только из заданных системных каталогов, см. рис.39, запись в которые запрещена.

Видим, насколько сложные задачи защиты, какими простейшими настройками соответствующего механизма защиты могут быть эффективно решены. При этом опять же реализуется принцип «черного ящика», т.к. не важны причины (используемые выявленные угрозы) осуществления атаки на системные файловые объекты, они защищены.

Следующий пример иллюстрирует возможность реализации контроля доступа по типам файлов (по их расширениям).

Замечание. Для реализации корректной разграничительной политики доступа по типам файлов (по расширениям файлов), см. вторую главу, объекты доступа, задаваемые в разграничительной политике расширениями (например «*.exe»), средством защиты запрещается модифицировать, переименовывать, удалять, а также, что крайне важно – что является основой реализации корректной разграничительной политики – запрещается создавать в системе новые файлы с заданными в разграничительной политике доступа расширениями файловых объектов.

Рассмотрим атаки типа «drive-by загрузки» для ОС семейства Windows. В случае реализации данного типа атак происходит несанкционированная загрузка эксплойта или заражение доверенных скриптов. Реализация защиты от подобных атак при реализации разграничительной политики доступа по типам файлов заключается в следующем: при помощи масок назначаются объекты файловой системы, которые потенциально несут в себе соответствующую угрозу, запрещается их модификация и создание новых подобных объектов, в том числе переименованием. В данном случае к подобным объектам могут быть отнесены файлы с расширениями *.js, *.vbs, *.php и др., которые являются файлами, написанными на скриптовых языках программирования.

В качестве объектов доступа задаются файлы-скрипты с указанными расширениями: *.vbs и *.vbe (VBScript) задается как «*.vb*»; *.js и *.jse (JavaScript) задается как «*.js *»; *.wsf; *.wsh задается как «*.ws*»; *.scpt (AppleScript); *.php и *.asp., *.cgi, определяемые как «маска файла», см. рис. 40.

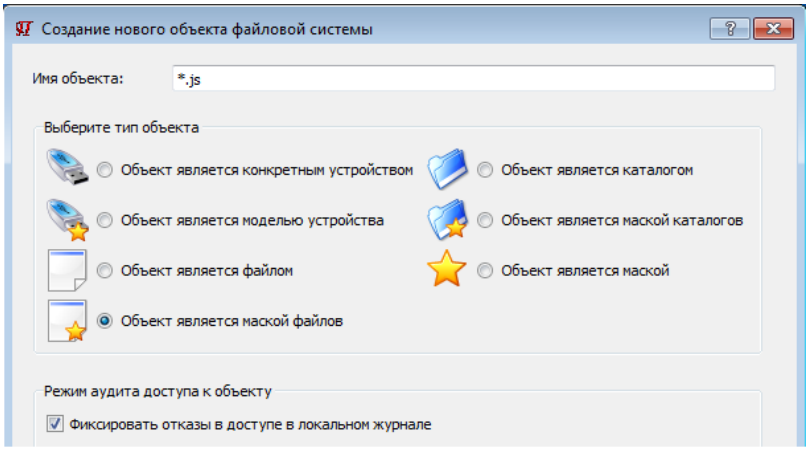


Рисунок 40. Задание объектов доступа

Для субъекта доступа, определяемого маской «*» (для любого субъекта), задаются правила доступа к файловым объектам, заданным данными расширениями. Данные правила предполагают разрешение лишь чтения подобных файловых объектов (запрещается запись, удаление, переименование).

В результате в системе смогут запускаться только санкционированно установленные скрипты, причем их удаление/модификация предотвращены.

Предотвращена также любая (несанкционированная) возможность установки на компьютер скриптового файла (это правило для подобного способа задания объекта доступа –по расширениям файлов – задается запретом переименования файла соответствующего типа с соответствующим расширением).

Как видим, задача защиты решена в полном объеме, причем опять же при реализации простейшей разграничительной политики доступа.

Отметим, что особенностью реализуемого подхода к защите является то, что не сохраняются и не запускаются и легальные скрипты с веб-сайтов. Большинство из этих скриптов, как правило, предназначено для авто заполнения форм, рекламного баннера, загрузки дополнительной страницы с рекламой. Как следствие, данное ограничение следует рассматривать скорее как достоинство, а не как недостаток рассмотренного подхода к защите.

Аналогичным образом может быть решена задача защиты от загрузки на защищаемый компьютер и запуска вредоносных программ (отметим, что описанными ранее примерами решалась задача защиты от запуска несанкционированных программ, при этом не предотвращать возможность их установки на компьютер).

Пример соответствующей простейшей разграничительной политики доступа к исполняемым файлам, обеспечивающей корректное функционирование системы, представлен в табл.3[1].

Таблица 3.Пример разграничительной политики доступа

Объекты файловой системы	**Режим доступа**
*.exe	+Ч-З+И-У-П
*.config	+Ч-З+И-У-П
*.dll	+Ч-З+И-У-П
*.manifest	+Ч-З+И-У-П
*.drv	+Ч-З+И-У-П
*.fon	+Ч-З+И-У-П
*.ttf	+Ч-З+И-У-П
*.sys	+Ч-З+И-У-П
*	+Ч+З-И+У+П

Заметим, что последнее правило, где объект определяется как «маска» (остальные объекты как «маска файла» более приоритетны), точнее, описатель объекта, задает запрет на исполнение любого файла с расширениями, отличными от перечисленных в таблице (используется способ более точно описателя при идентификации объектов). Запрет переименования предполагает предотвращение возможности изменения субъектом (в качестве субъекта доступа, для которого назначаются данные правила, может выступать критичное приложение, все интерактивные пользователи и т.д. в зависимости от решаемой задачи защиты) файла, разрешенного к исполнению, и

предотвращение возможности создания подобного файла (об этом говорили выше). Дополнительно в разграничительную политику могут вноситься объекты, создание которых следует запрещать, но вместе с тем не следует и разрешать их исполнение, например, объект «*.com». К подобным объектам следует полностью запрещать доступ.

Отметим, что при подобной разграничительной политике остается угроза запуска несанкционированной программы, в том числе вредоносной с внешнего накопителя, для которого невозможно проконтролировать создание исполняемого файла на другом компьютере. Для предотвращения подобной возможности необходимо разрешить исполнение файлов только с жесткого диска, задав объекты доступа соответствующей маской, например, «C:*.exe», определив их как «маску файла».

Достаточно важной возможностью построения разграничительной политики доступа к статичным файловым объектам является возможность контроля доступа к файловым объектам системных субъектов (пользователей и процессов). Например, достаточно серьезную угрозу несут в себе уязвимости в сетевых службах, работающих с системными правами. Некоторые примеры разграничительных политик для системных процессов приведены в [9].

8.5. Пример реализации контроля доступа к создаваемым файлам и к статичным файловым объектам в комплексе

В качестве вредоносной программы ранее мы рассматривали исполняемый файл, несанкционированно внедряемый на защищаемый компьютер. Исполняемыми файлами в Windows считаются бинарные файлы, которые могут напрямую вызываться операционной системой и исполняться в оперативной памяти. Однако в общем случае задача защиты от вредоносных программ должна рассматриваться с учетом того, что вредоносная активность связана не только с запуском несанкционированных исполняемых файлов, но и с вредоносным кодом (неисполняемые файлы), исполняемым компандными интерпретаторами, например, виртуальными машинами (санкционированными процессами). При прочтении виртуальной машиной вредоносного кода ее санкционированный процесс наделяется вредоносными свойствами.

Рассмотрим решение задачи защиты от атак на виртуальную машину с использованием вредоносного кода. При этом воспользуемся методами контроля доступа к статичным и создаваемым файловым объектам в комплексе. В обоих случаях субъект доступа определяется тремя сущностями: исходный идентификатор пользователя (который запускает процесс); эффективный идентификатор пользователя (от лица которого процесс запрашивает доступ к файловому объекту); полнопутевое имя процесса (имя исполняемого файла процесса), запрашивающего доступ.

Рассмотрим два варианта использования виртуальной машины:идля работы только с локальными приложениями и для работы с сетевыми приложениями. Разграничительные политики доступа для данных условий использования приложения будут отличаться.

Сначала рассмотрим случай использования виртуальной машины только для работы с локальными приложениями, т.е. предположим, что виртуальная машина должна использоваться на защищаемом компьютере, подключенном к внешней сети, для работы только с локальными приложениями.

Сформулируем задачу защиты в этом случае:

- предотвратить возможность доступа виртуальной машине к коду, внедренному из внешней сети;
- предотвратить возможность запуска виртуальной машины сетевыми приложениями.

Предположим, что в качестве сетевого приложения на защищаемом компьютере используется стандартный браузер операционной системы Windows –процесс iexplore.exe, а в качестве виртуальной машины JVM (JavaVirtualMachine)от Oracle – процесс java.exe. Соотвественно, эти процессы будут выступать в качестве субъектов доступа, для которых следует реализовать разграничительную политику.

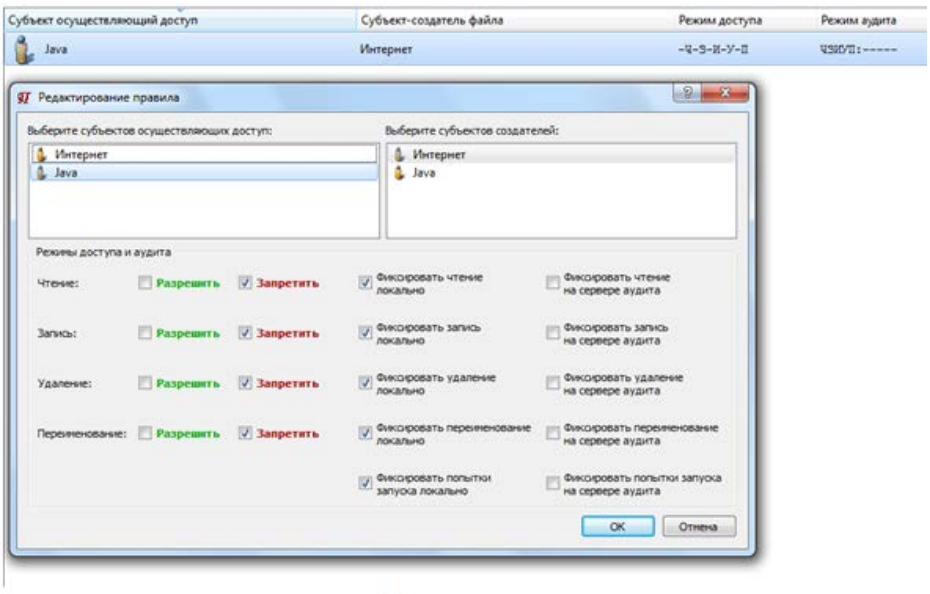
Зададим данные субъекты доступа (см. рис. 41) в средстве защиты.

Тип	Имя	Процесс	"Эффективный" пользователь	"Первичный" пользователь
	интернет	E:\Program Files\Internet Explorer\iexplore.exe	<Любой>	<Любой>
	java	E:\Program Files\Java\jre7\bin*	<Любой>	<Любой>

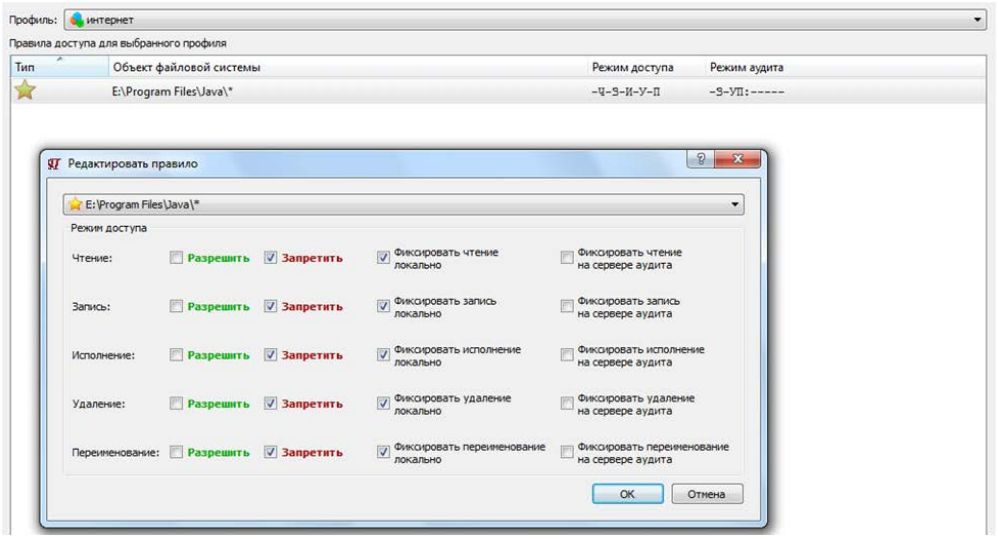
Рисунок 41. Заданные субъекты доступа

Замечание. Для задания субъекта виртуальной машины используем соответствующую маску.

Для того чтобы предотвратить возможность доступа виртуальной машине к внедряемому на компьютер коду из сети, среди которого велика вероятность вредоносного кода, воспользуемся контролем доступа к создаваемым файловым объектам, которым предотвратим возможность доступа виртуальной машине к любым файлам, создаваемым браузером. На рис.42, *а*, показан интерфейс настройки контроля доступа к создаваемым файлам и заданные правила доступа. Рассмотрим результат от реализации подобной простейшей разграничительной политики доступа. Все файлы, загружаемые на компьютер браузером, автоматически размечаются средством защиты при создании, т.е. к ним применяется контроль доступа. При этом, следуя заданному правилу, виртуальная машина не сможет получить доступ к подобным файлам. Заметим, что данная сложнейшая задача защиты решается одним правилом в разграничительной политике!



а)



б)

Рисунок 42. Интерфейсы настройки контроля доступа к создаваемым файлам (а) и статичным файловым объектам (б) и заданные правила контроля доступа

Следующим шагом является использование контроля доступа к статичным объектам в целях блокировки доступа браузера к объектам виртуальной машины. На рис. 42, *б*показан интерфейс настройки контроля доступа к статичным файловым объектам и заданные правила доступа. Такая мера обезопасит виртуальную машину от исполнения вредоносного кода на языке Java посредством вызова его с помощью уязвимостей браузера. Также следует отметить, что в данном случае теряется возможность запуска любого Java-кода браузером, в том числе и java.applet.

Замечание. В качестве статичного файлового объекта, доступ к которому контролируется, задана вся директория Java (см. рис.42, *б*).

Таким образом сформулированная задача защиты решена в полном объеме, при этомтрудоемкость задания разграничительной политики доступа к файловым объектам минимальна.

Теперь рассмотрим случай использования виртуальной машины для работы с сетевыми приложениями.

Сформулируем задачу защиты:

- предотвратить возможность доступа виртуальной машине к обрабатываемой на защищаемом компьютере информации;
- предотвратить возможность доступа виртуальной машине к системным файлам, кроме тех, которые ей необходимы для корректного функционирования.

По сути, задача защиты сводится к изолированию работы виртуальной машины в системе.

Для решения данной задачи защиты реализуем контроль доступа к статичным объектам. Решение задачи защиты опять же обеспечивается простейшими настройками разграничительной политики доступа, представленными на рис.43, *а*. Данной разграничительной политикой виртуальной машине разрешен доступтолько к

собственным объектам, причем только на чтение и исполнение, ко всем остальным объектам файловой системы (объект задан маской «*») какой-либо доступей запрещен.

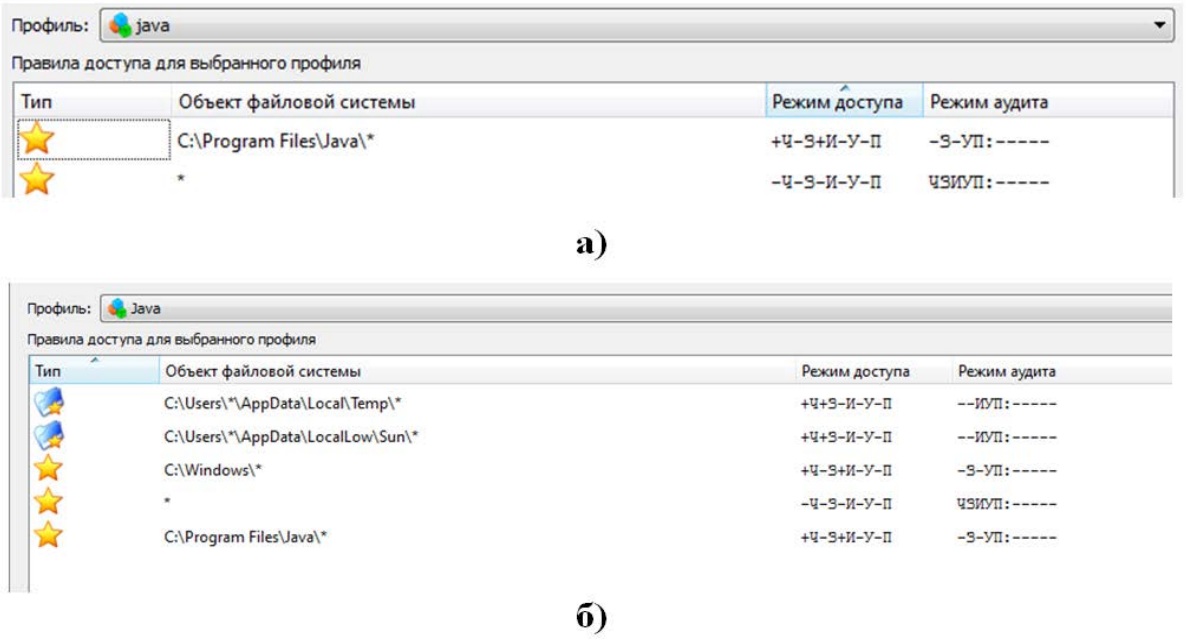


Рисунок 43. Функциональная (а) и корректная (б) разграничительные политики доступа к файловым объектам для виртуальной машины

Однако по результатам практической эксплуатации средства защиты можно определить, что для обеспечения корректности работы виртуальной машины в рассматриваемых приложениях настройки, представленные на рис.43, *а*, необходимо расширить. Например, дополнительные файлы скачивает в данном случае уже не браузер, а сама виртуальная машина. Для обеспечения корректной работы приложений в рассматриваемых условиях необходимо реализоватьследующие дополнительные требования: обеспечить возможность доступа в сеть виртуальной машине и доступа ее к временным папкам, в которых хранятся скомпилированные файлы java.applet, созданные браузером. Корректная разграничительная политика доступа виртуальной машины к файловым объектам представлена на рис.43, *б*, из которого видно, что обеспечение корректности работы виртуальной машины с сетевыми приложениями не приводит к снижению эффективности защиты. Как видим, сформулированнаязадача защиты решена в полном объеме, причем трудоемкость ее решения в части задания разграничительной политики доступа к файловым объектам минимальна!

Выше мы рассмотрели постановку и решение задачи упрощения администрирования средств защиты, в частности, в рамках этой задачи были предложены методы контроля доступа к создаваемым файлам, практическое использование которых позволяет решатьнаиболее актуальные задачи защиты информации. Возникает вопрос, а можно ли применить данные результаты при построении средств защиты личных компьютеров. Отметим, что задачи проектирования средств защиты корпоративных вычислительных средств и личных компьютеров принципиально различаются собственно в своей постановке. Если в первом случае предполагается реализация защиты конфиденциальной информации, причем, как правило, обрабатываемой на предприятии в значительных объемах, как следствие, потенциальная заинтересованность злоумышленников в ее хищении, в

результате необходимость защиты от потенциально возможных целевых атак. В данном случае предполагается использование сложных и эффективных средств защиты от разнообразных целевых атак, включение в штат сотрудников предприятия администратора безопасности — квалифицированного сотрудника, способного администрировать и эксплуатировать сложные средства защиты (это его работа), а пользователь должен рассматриваться в качестве потенциального злоумышленника, поскольку им в вычислительной системе обрабатывается не собственная, а корпоративная информация.

Во втором случае, все наоборот и намного проще.Здесь в общем случае уже не приходится говорить о целевых атаках на обрабатываемую в компьютере информацию, о квалификации пользователя в области информационной безопасности (администратора безопасности нет), о недоверии к пользователю, т.к. им обрабатывается личная информация. В данном случае мало применимы сложные средства защиты.

Большинство атак в данных приложениях сводится к внедрению на компьютер вредоносной программы с последующей ее эксплуатацией злоумышленником с какой-либо целью.

Рассмотрим реализацию защиты для решения данных задач на примере системы защиты «Панцирь+», далее СЗ (это, в отличие от рассматриваемого ранее, средство защиты для личного использования).

Прежде всего, остановимся на рассмотрении реализуемой технологии защиты.

8.5.1. Реализуемая технология защиты

1) <u>Технология защиты от запуска вредоносного ПО.</u>

Задача защиты — не позволять несанкционированно (без ведома пользователей, без их осознанного решения) запускать (исполнять) на компьютере файлы, созданные пользователями (в том числе несанкционированно от их имени) в процессе эксплуатации системы.

Решение. Любой создаваемый интерактивным пользователем файл автоматически размечается СЗ, ему сопоставляется учетная информация создавшего файл субъекта доступа (имя учетной записи и процесса —полнопутевое имя исполняемого файла процесса). При обращении к любому файлу на исполнение (в том числе и системой) СЗ анализируется, был ли создан этот файл в процессе эксплуатации системы (размечен ли он). Если это так, то автоматическое исполнение (запуск) подобного файла блокируется, пользователю выдается соответствующее уведомление. При этом пользователю предлагается проанализировать причину подобного несанкционированного события по журналу событий и решить, санкционирован ли этот файл для последующего исполнения. Если нет, то пользователь сможет удалить этот файл из проводника СЗ; если да, то удалить его разметку, переведя тем самым файл в разряд санкционированных для исполнения, и впоследствии запустить его.Универсальность решения достигается тем, что проводимая процедура анализа никак не связана с типом файла, в том числе с типом его расширения. Перехватывается системный запрос на запись, соответственно на исполнение. Именно подобным образом идентифицируется исполняемый файл. Принципиальным является и то, что перехватываются не запросы на открытие файла для записи и исполнения, а непосредственно запись и исполнение, что сводит к минимуму ложные срабатывания средства защиты. Универсальность решения обеспечивается и тем, что при реализации данной технологии защиты не важен способ занесения (внедрения) вредоносной программы (исполняемого файла) на защищаемый компьютер: загрузка из

интернета, почтовое вложение (в архиве, либо нет), копирование с внешнего накопителя и т.п. — любым способом записанный за защищаемый компьютер файл будет автоматически размечен, и в отношении него будет действовать защита от несанкционированного исполнения.

В рамках реализации данной технологии [12] СЗ решаются следующие задачи защиты:

- Предотвращение запуска программ с внешних файловых накопителей (аудит подобных попыток запуска СЗ не ведется, о запрете запрошенного доступа на исполнение к внешнему накопителю пользователь уведомляется штатным сообщением ОС).

  Примечание. В общем случае отследить создание файла на внешнем накопителе невозможно (он может быть создан на незащищенном компьютере, как следствие не будет размечен). Поэтому исполнение программ с внешних накопителей СЗ блокируется;
- Контроль (разметка) создаваемых интерактивными пользователями на компьютере файлов в процессе работы системы, автоматическое предотвращение несанкционированного запуска программ, СЗ осуществляет аудит создания и попыток исполнения создаваемых файлов.

2) <u>Технология защиты от модификации санкционированных исполняемых файлов.</u>

Описанная выше технология обеспечивает невозможность запуска несанкционированно установленной на компьютер программы, при этом санкционировано установленные исполняемые объекты (файлы), в том числе исполняемые системные файлы, остаются уязвимы. Они могут быть модифицированы, удалены, переименованы. Все эти действия не приведут к запуску вредоносной программы, но могут сказаться на работоспособности (корректности работы) ОС и приложений. Задача защиты — не позволять несанкционированно (без ведома пользователей, без их осознанного решения) модифцировать исполняемые файлы ОС и приложений.

Для защиты исполняемых объектов от несанкционированной модификации, удаления, переименования в СЗ реализована следующая технология защиты, также основанная на автоматической разметке файлов, но в данном случае, уже не создаваемых в процессе работы пользователей, а установленных ранее – исполняемых (заметим, можно осуществить разметку соответствующих файлов вручную, но наша цель – максимальное упрощение задачи администрирования). Любой исполненный (не размеченный как созданный, в противном случае он не сможет быть исполнен) файл СЗ автоматически размечается: ему сопоставляется учетная информация исполнившего файл субъекта доступа (имя учетной записи и процесса – полнопутевое имя исполняемого файла процесса).

Примечание. Чтобы отделить создаваемый файл от исполняемого, в разметку файла включается тип файла.

При обращении к любому файлу интерактивным пользователем на модификацию/удаление/ переименование СЗ анализируется, был ли он размечен как исполняемый. Если это так, подобный доступ к исполняемому файла блокируется, пользователю предлагается решить, санкционирован ли этот файл для изменения. Если нет, то пользователь сможет проанализировать причину подобного несанкционированного события по журналу событий, приняв далее необходимые меры; если да, то удалить его разметку, переведя тем самым файл в разряд санкциониированных для модификации, удаления, переименования, и впоследствии

изменить его. Универсальность решения достигается тем, что проводимая процедура анализа никак не связана с типом файла, в том числе с типом его расширения. Перехватывается системный запрос на исполнение, соответственно на модификацию/удаление/переименование,– именно подобным образом идентифицируется исполняемый файл. Принципиальным является и то, что перехватываются не запросы на открытие файла для исполнения, модификацию/удаление/переименование, а непосредственно контролируемые действия, что сводит к минимуму ложные срабатывания средства защиты.

Примечание. Для автоматической разметки исполняемых объектов ОС и приложений рекомендуется после ввода СЗ в действие, по крайней мере, один раз запустить критичные к модификации приложения.

В рамках реализации данной технологии СЗ решаются следующие задачи защиты:

- Контроль (разметка) исполненных на компьютере файлов ОС и приложений в процессе работы системы,
- Автоматическое предотвращение несанкционированной модификации/удаления/переименование исполняемых файлов,
- СЗ осуществляет аудит исполнения и попыток модификации/удаления/переименование исполняемых файлов.

3) <u>Дополнительная защита:</u>

- Защита от обхода реализуемых СЗ правил доступа к файловым объектам за счет несанкционированного получения интерактивными пользователями системных прав (атака на повышение привилегий). Реализована следующая технология защиты. СЗ фиксирует, каким интерактивным пользователем осуществлен запуск каждого приложения. В случае, если приложение обращается к файловому объекту не под учетной записью запустившего его пользователя, а под системной учетной записью, любой доступ к любому файловому объекту данному приложению СЗ блокируется.
- Самозащита. Файлы СЗ защищены от несанкционированного доступа к ним с целью удаления и модификации.

Теперь рассмотрим настройку и различные режимы СЗ, что проиллюстрируем соответствующими интерфейсами.

8.5.2. Настройка системы защиты

1) <u>Органы управления</u>

При нажатии по значку СЗ правой кнопкой мыши откроется меню управления, рис.44. В данном меню можно выбрать одно из следующих действий: убрать защиту, открыть журнал аудита, открыть программу обзора разметки файлов на компьютере.

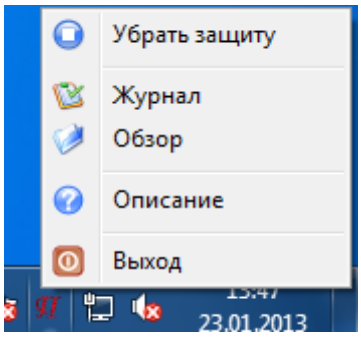


Рисунок 44. Меню управления СЗ

При выборе «Убрать защиту» значок СЗ на панели задач отобразится серым цветом. Для запуска СЗ следует нажать на ярлык СЗ правой кнопкой мыши и в открывшемся меню выбрать «Включить защиту».

2) Предварительная настройка, штатный режим эксплуатации

Никакой настройки СЗ производить не требуется. Запущенная программа в полном объеме автоматически решает свои задачи. До тех пор, пока не наступит нештатный режим, беспокоиться не о чем!

3) Нештатный режим, угроза несанкционированного доступа

Нештатный режим наступает при выявлении СЗ попытки несанкционированного доступа к размеченным файлам. При этом значок СЗ окрашивается в желтый цвет, пользователю выдается соответствующее уведомление, рис.45.

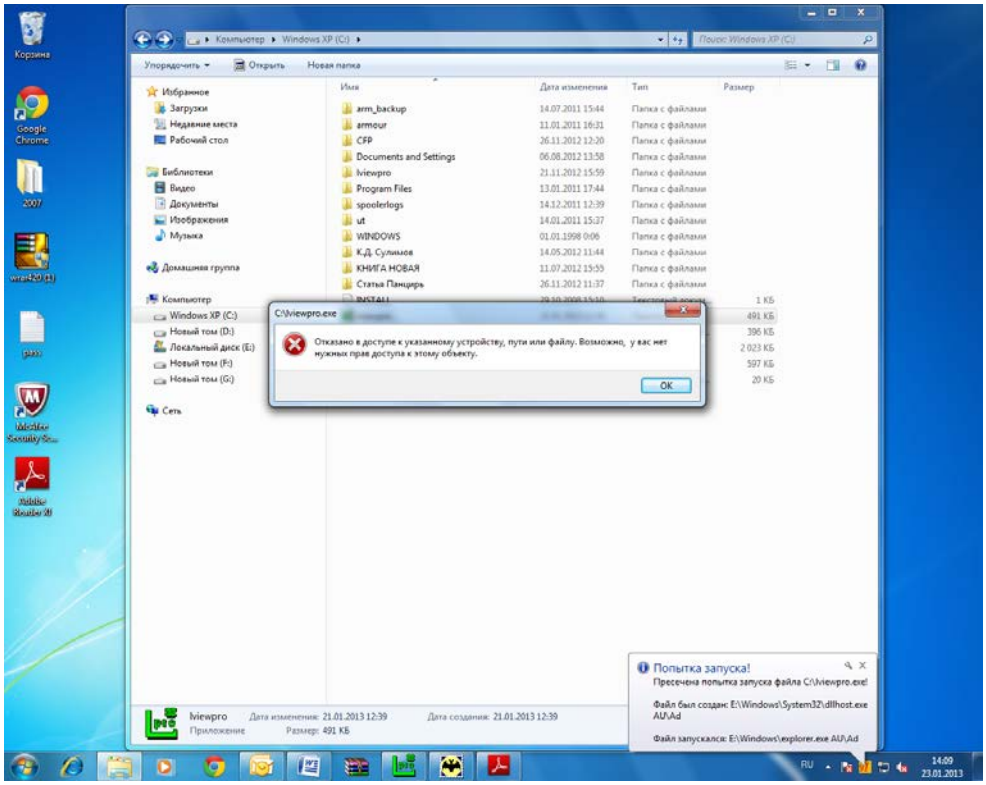


Рисунок 45. Уведомление пользователя о попытке несанкционированного доступа к размеченному файлу

Можно проигнорировать это уведомление и работать далее (СЗ свое дело сделала: попытка запуска несанкционированной программы либо несанкционированной модификации исполняемого файла была предотвращена), а можно проанализировать нештатную ситуацию. Для этого требуется открыть меню, рис.43, и выбрать в нем «Журнал», откроется журнал событий, рис.46.

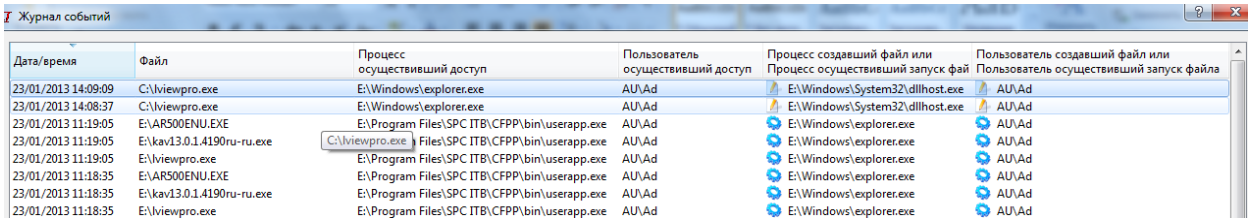


Рисунок 46. Журнал событий

Журнал событий имеет следующую структуру, см. рис.46:

- «Дата/время» (временные характеристики регистрации события);
- «Файл» (полнопутевое имя размеченного файла, несанкционированная попытка доступа к которому была предотвращена СЗ);
- «Процесс, осуществивший доступ» (полнопутевое имя исполняемого файла процесса, совершившего несанкционированную попытку исполнения размеченного созданного файла либо модификацию/удаление/переименование размеченного исполняемого файла);
- «Пользователь, осуществивший доступ» (имя пользователя (учетной записи), которым (от лица которого) совершена попытка несанкционированного доступа к размеченному файлу);
- «Процесс, создавший файл», или «Процесс, осуществивший запуск файла» (полнопутевое имя исполняемого файла процесса, создавшего файл, к которому запрашивается несанкционированный доступ на исполнение, или полнопутевое имя исполняемого файла процесса, исполнившего ранее файл, к которому запрашивается несанкционированный доступ на модификацию/удаление/переименование);
- «Пользователь, создавший файл», или «Пользователь, осуществивший запуск файла» (имя пользователя (учетной записи), которым (от лица которого) создан файл, к которому запрашивается несанкционированный доступ на исполнение, или имя пользователя (учетной записи), исполнившего ранее файл, к которому запрашивается несанкционированный доступ на модификацию/удаление/переименование).

Замечание. Тип размеченного файла, созданный в процессе работы пользователя либо исполняемый, несанкционированный доступ к которому предотвращается СЗ, отображается соответствующей пиктограммой в журнале событий, см. рис.46.

Каковы возможные дальнейшие действия?

a. **<u>В отношении размеченного файла, к которому осуществлена попытка несанкционированного доступа.</u>**

Дважды щелкнув левой кнопкой мыши по выбранному файлу в поле «файл» журнала событий, можно открыть программу СЗ обзора файловой системы, рис. 47.

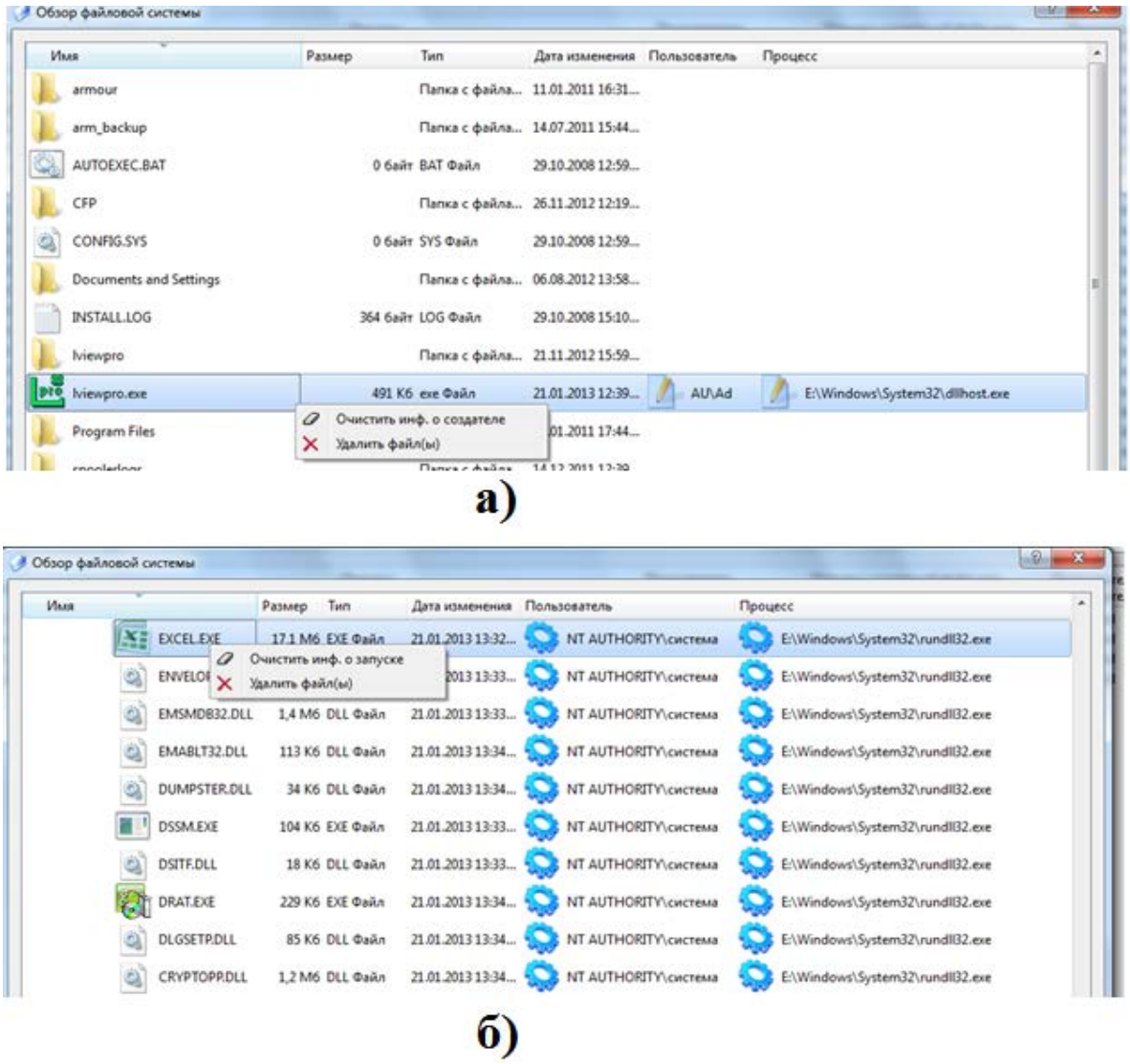


Рисунок 47. Программа СЗ обзора файловой системы. Отображение созданного (а) и размеченного исполняемого файла

В ней отобразится выбранный файл с его разметкой. Здесь также используется соответствующая пиктограмма для указания типа файла. На рис. 47, *а* отображен размеченный СЗ созданный файл, на рис.47, *б*–размеченный СЗ исполняемый файл.

Щелкнувправой кнопкой мыши на выбранный файл, можнооткрыть меню, предоставляющее возможность либо его удаления, либо удаления его разметки, рис.47. При удалении разметки файла (выбрать «Очистить инф. о запуске», см. рис.47) файл переводится в категорию санкционированных. Впоследствии не отключая СЗ, данный файл можно будет благополучно исполнять либо соответственно модифицировать/удалять/переименовывать (в зависимости от типа файла).

b. <u>В отношении процесса, которым осуществлена попытка несанкционированного доступа к размеченному файлу.</u>

Щелкнув же дважды левой кнопкой мыши на выбранный «Процесс, осуществивший доступ», либо на «Процесс, создавший файл», либо «Процесс, осуществивший запуск файла», см. рис.46, можно открыть программу обзора файловой системы, в которой будет отображен исполняемый файл выбранного процесса с его разметкой (поскольку он уже исполнялся), рис.47. Щелкнув по правой кнопке мыши, можно удалить исполняемый файл выбранного процесса либо удалить его разметку, рис.47,*б*.

В части дополнительных возможностей СЗ предоставляет возможность просмотра при (необходимости удаления) произведенной СЗ разметки файлов (в том числе всех файлов в выбранной папке), что реализуется из интерфейса, приведенного на рис.48.

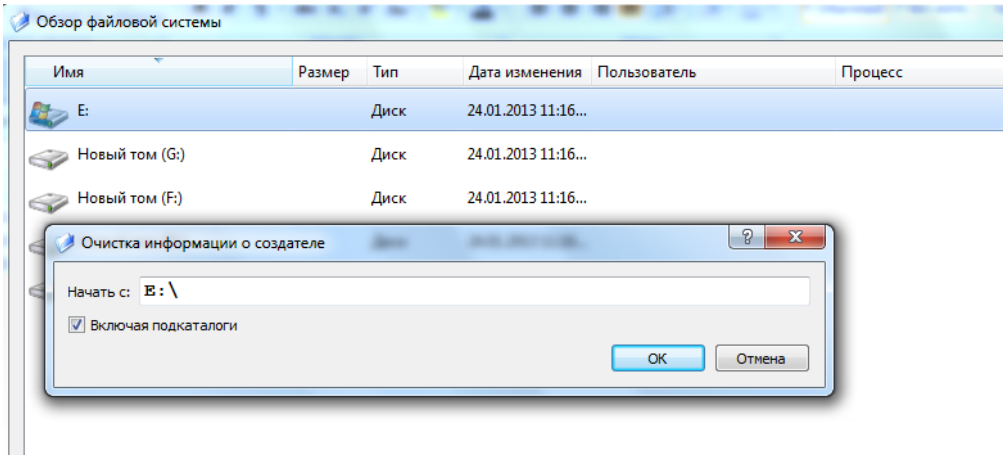


Рисунок 48. Задание параметров очистки разметки файлов в выбранном объекте

Как видим, нам удалось построить средство защиты от, наверное, самых актуальных сегодня угроз личному компьютеру, причем средство, которое вообще не требует какой-либо настройки, какого-либо администрирования.

Не сложно реализовать подобный подход, не требующий настройки разграничительных политик доступа к построению средств защиты личного компьютера и от иных актуальных угроз, например, изолировать работу на компьютере интернет-браузера.

Все это позволяет утверждать, что методы контроля доступа к ресурсам вычислительных систем могут применяться в различных приложениях средств защиты: для защиты как корпоративных, так и личных компьютеров, позволяя при этом эффективно решать наиболее актуальные на сегодняшний день задачи защиты информации от несанкционированного доступа.

Отметим, что это материалы учебного пособия, а не инструкции по защите от наиболее актуальных угроз, а уж тем более не инструкции по применению конкретных систем защиты информации от несанкционированного доступа. Целью приведенных примеров являлась лишь иилюстрация того, насколько приниципально реализация сформулированных в учебном пособии требований к реализации контроля доступа к ресурсам вычислительных систем в современных условиях меняет подходы к построению соответствующих механизмов защиты и насколько эффективно ими могут решаться наиболее актуальные современные задачи защиты информации от несанкционированного доступа, причем в различных приложениях средств защиты информации.

9. Заключение

Итак, в учебном пособии исследованы существующие широко используемые на практике модели и методы контроля доступа к защищаемым ресурсам вычислительных систем, реализуемого с целью защиты обрабатываемой информации от несанкционированного доступа. Выявлены их ключевые недостатки применительно к реализации контроля доступа в современных условиях. Сформулированы и обоснованы требования, реализация которых позволяет построить корректную разграничительную политику доступа к ресурсам и обеспечить эффективную защиту от наиболее актуальных современных угроз. Обоснование данных требований проведено на построенных и приведенных в учебном пособии моделях. Выполнение сформулированных требований приводит к кардинальному пересмотру самих принципов построения контроля доступа к защищаемым ресурсам. Реализация современных методов контроля доступа, изложенных в учебном пособии, проиллюстрирована практическими примерами апробированного технического решения. На примерах же рассмотрено построение разграничительных политик доступа, позволяющих решать наиболее актуальные задачи защиты информации.

В результате освоения изложенного в учебном пособии материала обучающийся овладеет принципами построения и оценивания эффективности, в том числе в части корректности реализации ключевых элементов защиты современных информационных систем — средств контроля доступа к защищаемым ресурсам вычислительных систем.

10.Литература

1) Шибаева Т.А., Щеглов А.Ю., Оголюк А.А. Защита от внедрения и запуска вредоносных программ // Вопросы защиты информации. Научно-практический журнал. Выпуск 2 (93), Москва, 2011. – С.26-35.
2) 2.Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. – СПб: Наука и техника, 2004, - 384с.
3) Щеглов К.А., Щеглов А.Ю. Принцип и методы контроля доступа к создаваемым файловым объектам // Вестник компьютерных и информационных технологий. - Москва, 2012. - № 7. - С. 43-47.
4) Щеглов К.А., Щеглов А.Ю. Принцип и метод дискреционного контроля доступа к создаваемым файловым объектам // Вопросы защиты информации. - Москва: ВИМИ, 2012. - Вып. 96. - № 1. - С. 30-38.
5) Щеглов К.А., Щеглов А.Ю. Принцип и метод мандатного контроля доступа к создаваемым файловым объектам // Вопросы защиты информации. - Москва: ВИМИ, 2012. - Вып. 96. - № 1. - С. 40-44.
6) Щеглов К.А., Щеглов А.Ю. Модель контроля доступа к создаваемым файловым объектам // Изв. ВУЗов. Приборостроение. - 2012. - Т. 55. - № 10. - С. 37-40.
7) Щеглов К.А., Щеглов А.Ю. Защита от вредоносных программ методом контроля доступа к создаваемым файловым объектам // Вестник компьютерных и информационных технологий. - Москва, 2012. - № 8. - С. 46-51.
8) Щеглов К.А., Щеглов А.Ю. Методы идентификации и аутентификации пользователя при доступе к файловым объектам // Вестник компьютерных и информационных технологий. - Москва, 2012. - № 10. - С. 47-51.
9) Щеглов К.А., Щеглов А.Ю. Контроль доступа к статичным файловым объектам // Вопросы защиты информации. - Москва: ВИМИ, 2012. - Вып. 97. - № 2. - С. 12-20.
10) Щеглов К.А., Щеглов А.Ю. Защита от атак со стороны приложений, наделяемых вредоносными функциями. Модели контроля доступа // Вопросы защиты информации. - Москва: ВИМИ, 2012. - Вып. 99. - № 4. - С. 31-36.
11) Щеглов К.А., Щеглов А.Ю. Практическая реализация дискреционного метода контроля доступа к создаваемым файловым объектам//Вестник компьютерных и информационных технологий, 2013. - № 4. - С. 43-49.
12) Щеглов К.А., Щеглов А.Ю. Система защиты от запуска вредоносных программ // Вестник компьютерных и информационных технологий. 2013. № 5. С. 38 – 43.
13) Щеглов К.А., Щеглов А.Ю. Защита от атак на уязвимости приложений. Модели контроля доступа // Вопросы защиты информации. - Москва: ВИМИ, 2013. - Вып. 101. - № 2. - С. 36-43.
14) Щеглов А.Ю., Щеглов К.А. Принципы и методы контроля доступа к статичным файловым объектам с исключением из разграничительной политики доступа сущности «объект доступа» // Вестник компьютерных и информационных технологий. - Москва, 2013. - № 8. - С. 53-59.
15) Щеглов К.А., Щеглов А.Ю. Реализация метода мандатного доступа к создаваемым файловым объектам системы // Вопросы защиты информации. - Москва: ВИМИ, 2013. - Вып. 103. - № 4. - С. 16-20.

16) Щеглов К.А., Щеглов А.Ю. Способ задания и хранения прав доступа субъектов к файловым объектам // Вестник компьютерных и информационных технологий. - Москва, 2013. - № 12. - С. 45-49.
17) Щеглов К.А., Щеглов А.Ю. Щеглов К.А., Щеглов А.Ю.Эксплуатационные характеристики риска нарушений безопасности информационной системы // Научно-технический вестник информационных технологий, механики и оптики. 2014. №1(89). С.129-139.
18) Bell D. E., LaPadula L. J. Security Computer Systems: Unified Exposition and MULTICS Interpretation, Revision 1, US Air Force ESD-TR-306, MITRE Corporation MTR-2997, Bedford MA, March 1976.
19) Biba K. J Integrity Consideration for Security Computer System. The MITRE Corp., Report MTR N3153 Revision 1, Electronic System Division, U.S. Air Force Systems Command, Technical Report ESD TR 76 372, Belford, Massachusetts, April 1977.
20) M. Harrison, W. Ruzzo, J. Ullman. Protection in operating systems. – Communication of ACM, 1976.
21) Опрос «Кода Безопасности» выявил наиболее актуальные ИБ угрозы [Электронный ресурс]//,URL:/ http://www.securitycode.ru/company/news/SC-analytic-2011.
22) Отчет по уязвимостям за второй квартал 2008 года[Электронный ресурс]// URL:/ http://www.securitylab.ru/analytics/358113.php
23) KasperskySecurityBulletin. Основная статистика за 2011 год [Электронный ресурс]// URL:/ http://www.securelist.com/ru/analysis/208050741/rss/analysis.

УНИВЕРСИТЕТ ИТМО

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

О кафедре

Кафедра вычислительной техники Университета ИТМО создана в 1937 году и является одной из старейших и авторитетнейших научно-педагогических школ России.

Первоначально кафедра называлась кафедрой математических и счетно-решающих приборов и устройств и занималась разработкой электромеханических вычислительных устройств и приборов управления. Свое нынешнее название кафедра получила в 1963 году.

Кафедра вычислительной техники является одной из крупнейших в университете, на которой работают высококвалифицированные специалисты, в том числе 7 профессоров и 14 доцентов.

Кафедра имеет 4 компьютерных класса, объединяющих более 70 компьютеров в локальную вычислительную сеть кафедры и обеспечивающих доступ студентов ко всем информационным ресурсам кафедры и выход в Интернет. Кроме того, на кафедре имеются учебные и научно-исследовательские лаборатории по вычислительной технике, в которых работают студенты кафедры.

Чему мы учим

Традиционно на кафедре вычислительной техники Университета ИТМО основной упор в подготовке специалистов делается на фундаментальную базовую подготовку в рамках общепрофессиональных и специальных дисциплин, охватывающих наиболее важные разделы вычислительной техники: функциональная схемотехника и микропроцессорная техника, алгоритмизация и программирование, информационные системы и базы данных, мультимедиатехнологии, вычислительные сети и средства телекоммуникации, защита информации и информационная безопасность. В то же время, кафедра предоставляет студентам старших курсов возможность специализироваться в более узких профессиональных областях в соответствии с их интересами.

Специализации на выбор

Кафедра вычислительной техники Университета ИТМО ведёт подготовку специалистов высшей квалификации в соответствии с Государственными образовательными стандартами 3-го поколения (ГОС-3) по двум направлениям:

09.04.01 «Информатика и вычислительная техника» (профиль подготовки «Вычислительные машины, комплексы, системы и сети»);

09.04.04 «Программная инженерия» (профиль подготовки «Разработка программно-информационных систем»);

с присвоением степени (квалификации) бакалавр (срок обучения – 4 года).

Прием абитуриентов на указанные направления подготовки бакалавров осуществляется в соответствии с общими Правилами приема в Университет ИТМО.

Студенты, успешно завершившие обучение и получившие дипломбакалавра, могут продолжить обучение в магистратуре кафедры (срок обучения – 2 года) по следующим магистерским программам:

- «Безопасность вычислительных систем и сетей» – руководитель д.т.н. профессор Щеглов Андрей Юрьевич;
- «Вычислительные системы и сети» - руководитель д.т.н. профессор Алиев ТауфикИзмайлович;
- «Информационно-вычислительные системы»- руководитель д.т.н. профессор Алиев ТауфикИзмайлович;
- «Интеллектуальные информационные системы» – руководитель д.т.н. профессор Тропченко Александр Ювенальевич;
- «Проектирование встроенных вычислительных систем» - руководитель д.т.н. профессор Платунов Алексей Евгеньевич;
- «Системотехника интегральных вычислителей. Системы на кристалле» – руководитель д.т.н. профессор Платунов Алексей Евгеньевич;
- «Сетевые встроенные системы» - руководитель д.т.н. профессор Платунов Алексей Евгеньевич;
- «Технологии компьютерной визуализации» (совместно с базовой кафедрой Института Прикладной математики им. М.В. Келдыша) – руководитель д.т.н. профессор Палташев Тимур Турсунович.

В магистратуру на конкурсной основе принимаются выпускники других вузов, имеющие диплом бакалавра.

На кафедре вычислительной техники Университета ИТМО в рамках аспирантуры и докторантуры осуществляется подготовка научных кадров по следующим специальностям:

- 05.13.05 – Элементы и устройства вычислительной техники и систем управления (технические науки);
- 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (технические науки);
- 05.13.12 – Системы автоматизации проектирования (приборостроение) (технические науки);
- 05.13.15 – Вычислительные машины, комплексы и компьютерные сети (технические науки);
- 05.13.17 – Теоретические основы информатики (технические науки);
- 05.13.18 – Математическое моделирование, численные методы и комплексы программ (технические науки);
- 05.13.19 – Методы и системы защиты информации, информационная безопасность (технические науки).

Щеглов Андрей Юрьевич

Модели, методы и средства контроля
доступа к ресурсам вычислительных систем

Учебное пособие

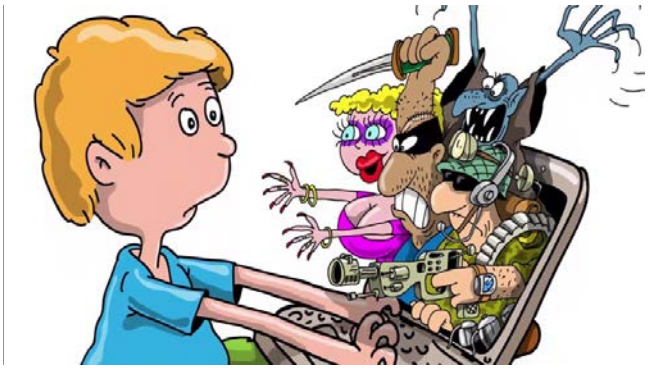
В авторской редакции
Редакционно-издательский отделУниверситет ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати …
Заказ №
Тираж
Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49

А. Ю. Щеглов

МОДЕЛИ, МЕТОДЫ И СРЕДСТВА КОНТРОЛЯ ДОСТУПА К РЕСУРСАМ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Учебное пособие



Санкт-Петербург
2014