РЕАЛИЗАЦИЯ КОНТРОЛЯ И РАЗГРАНИЧЕНИЯ ПРАВ ДОСТУПА К ТИПАМ ФАЙЛОВ ПО ИХ РАСШИРЕНИЯМ

Введение

В работе [1] были рассмотрены вопросы реализации контроля и разграничения прав доступа к статичным файловым объектам, при этом рассматривались вопросы включения в сущность субъект доступа процесса и возможности использования разграничительной политики доступа для решения задач защиты от вторжений. При реализации данного решения принципиально меняется собственно парадигма реализации контроля и разграничения прав доступа, поскольку целью создания разграничительной политики доступа становится не защита какого-либо критически важного файлового объекта, а защита от потенциальных негативных воздействий на систему какого-либо критичного, по каким-либо причинам [2,3] субъекта – процесса. На практике это приводит к тому, что правила доступа должны назначаться не объектам, в качестве их атрибутов, а должны присваиваться субъектам, в качестве их прав доступа к объектам. В [4,5] рассмотрены вопросы реализации контроля и разграничения прав доступа к создаваемым объектам, в частности, к создаваемым файлам. В данном случае объект доступа вообще исключается из разграничительной политики, права доступа к создаваемым ими объектам назначаются между субъектами, что образует принципиально новую технологий защиты информации от несанкционированного доступа [6]. Суть данного подхода состоит в автоматическом наследовании создаваемым объектом учетных данных, создавшего его объекта. На практике целесообразно совместное использование данных методов защиты, поскольку ими решаются различные задачи, что, например, проиллюстрировано в [7].

Существенным расширением функциональных возможностей разграничительной политики доступа субъектов к объектам является реализация контроля и разграничения прав доступа к типам файлов, при этом ввиду многообразия типов файлов, обрабатываемых в информационной системе – исполняемые, командные, файлы конфигурации системы и приложений, файлы, используемые для хранения обрабатываемых данных, в результате реализации

подобного контроля доступа может решаться множество важных задач защиты информации. Рассмотрим это в данной статье, рассмотрим реализацию контроля и разграничения прав доступа к типам файлов по их расширениям, а также сформулируем требование к корректности реализации реализующего данное решение механизма защиты информации.

Реализация контроля и разграничения прав доступа к статичным файлам

Каждый тип файлов имеет определенный уникальный формат, соответствующий его спецификации. Таким образом, при доступе к файлу существует возможность анализа формата этого файла. Однако существует две причины, ограничивающие практическое использование данной возможности. Во-первых, это достаточно существенная дополнительная загрузка вычислительных ресурсов (ведь придется анализировать форматы всех файлов, включая системные файлы, к которым могут разграничиваться права доступа), во-вторых, спецификации далеко не всех типов файлов доступны, некоторые разработчики приложений относят эту информацию к коммерческой тайне.

Однако самой ОС необходимо иметь информацию о типе файла для определения режима его обработки, например, каким приложением открывать тот или иной файл. В ОС Microsoft Windows с этой целью используется специальное поле имени файла – расширение файла. Каждому типу файлов однозначно соответствует некоторое расширение. Это в значительной мере упрощает задачу определения типа файла – его идентификация по расширению.

Воспользуемся этой возможностью при реализации контроля и разграничения прав доступа к типам файлов, при этом будем учитывать, что расширения файлов используются для упрощения задачи определения системой типа файла, не предполагая при этом решения каких-либо задач защиты информации.

Применительно к реализации контроля доступа к типам файлов статичных объектов, соответствующий механизм защиты во многом реализуется так, как это было описано в [1]. Рассмотрим реализацию данного запатентованного решения [8] в соответствующей коммерческой системе защиты информации [9], сертифицированной ФСТЭК России.

Субъекты доступа, которые идентифицируются в разграничительной политике доступа тремя сущностями – исходный идентификатор пользователя (от лица которого запущен процесс), эффективный идентификатор пользователя (от лица которого процесс обращается с запросом доступа к объекту, имя процесса (полнопутевое имя исполняемого файла процесса), который запрашивает доступ к объекту, создаются из меню, приведенного на рис.1. Обоснование подобного способа задания субъекта доступа в разграничительной политике представлено в [9].

Объекты доступа создаются из меню, приведенного на рис.2 и отображаются в интерфейсе, представленном на рис.3, правила доступа субъектов (профилей – субъекты, имеющие совпадающие права доступа к объектам, например, при реализации одной и той же роли в информационной системе, объединяются в профили, см. рис.1) к статичным объектам (разграничительная политика доступа) создаются из меню, представленного на рис.4, и отображаются в интерфейсе, приведенном на рис.5.

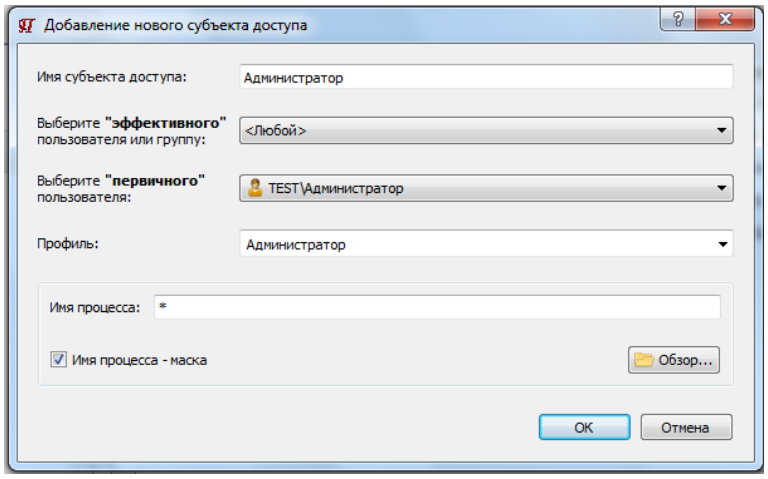


Рис.1. Меню создания субъекта доступа в разграничительной политике

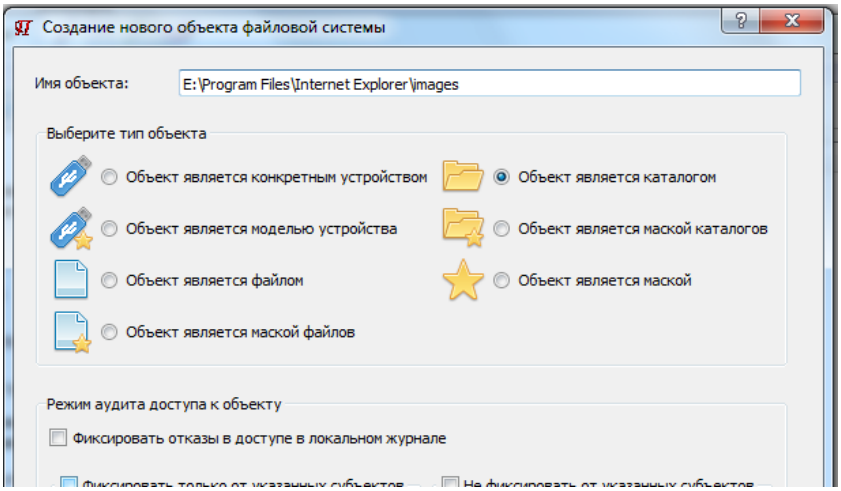


Рис.2. Меню создания объекта доступа в разграничительной политике

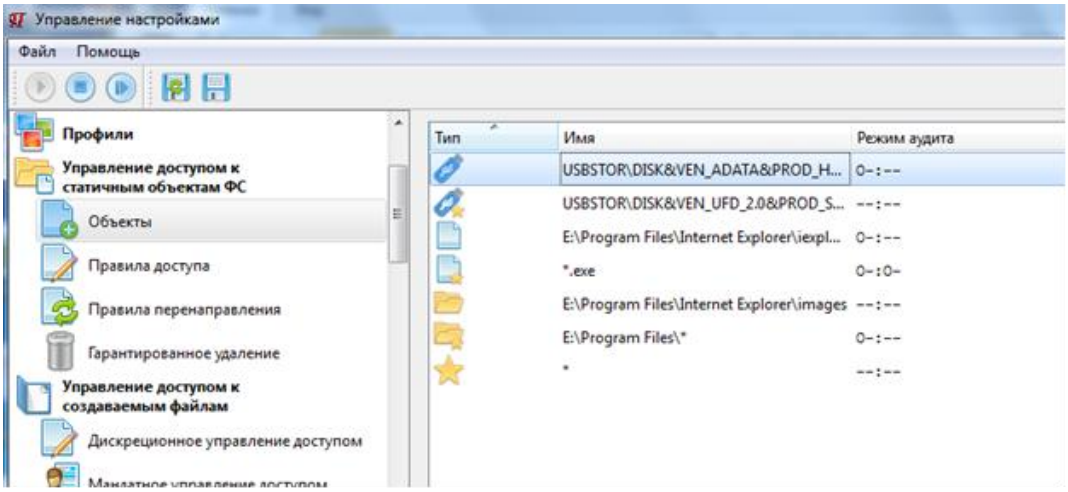


Рис.3. Интерфейс отображения созданных объектов доступа в разграничительной политике

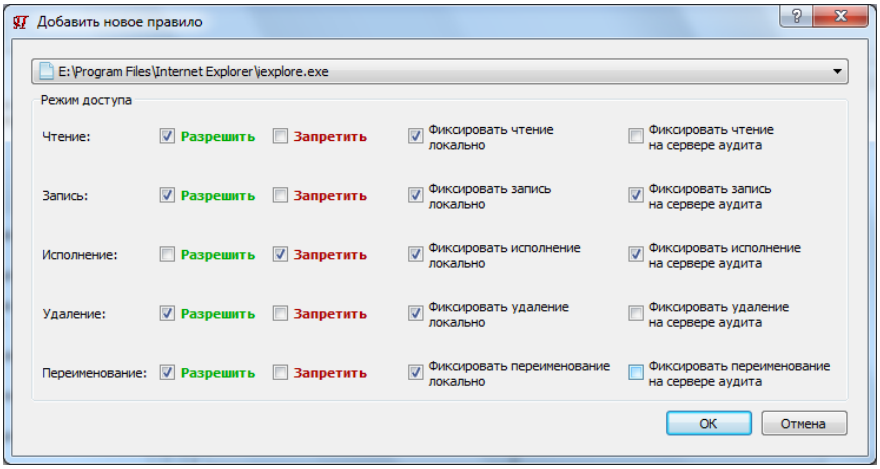


Рис.4. Меню задания правил доступа к статичным объектам

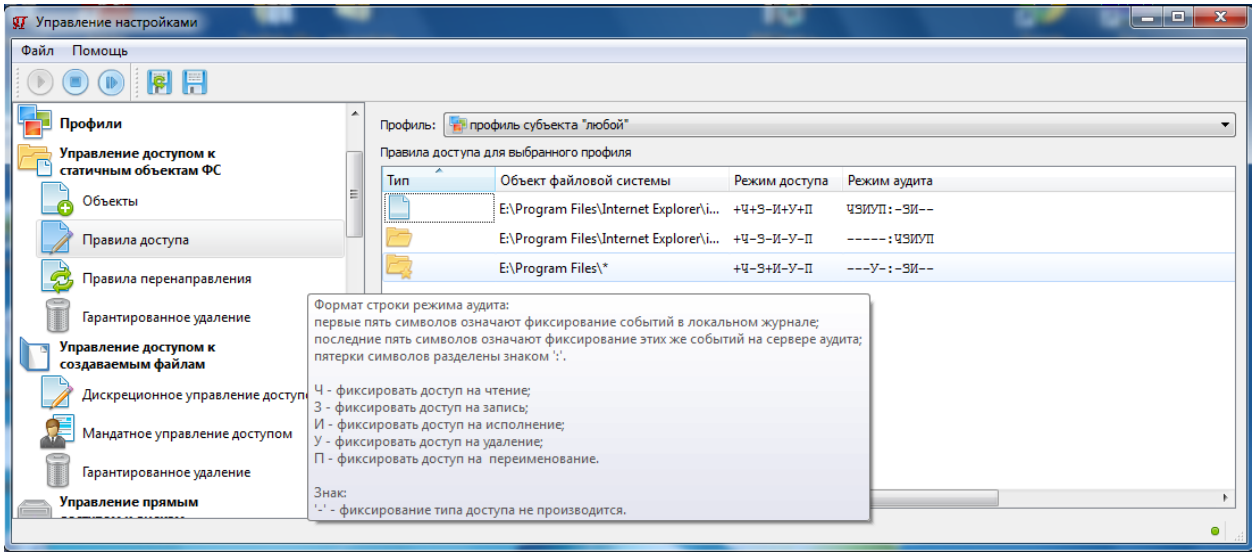


Рис.5. Интерфейс отображения созданных правил для субъекта доступа к статичным объектам

Правила доступа, см. рис.4, назначаются субъектам, что, во-первых, существенно упрощает реализацию разграничительной политики доступа, в том числе, за счет использования масок и переменных среды окружения, во-вторых, позволяет наглядно отображать правила, назначенные для критичных объектов, см. рис.5, в третьих появляется возможность тиражирования настроек между защищаемыми объектами крупномасштабной информационной системы, опять же за счет использования масок и переменных среды окружения (понятно, что тиражировать атрибуты файлов не представляется возможным, а в рассматриваемом случае правила доступа субъектов к объектам хранятся в отдельных файлах, при этом они могут унифицироваться, за счет применения масок). Применительно же к контролю и разграничению прав доступа к типам файлов альтернативный подход просто невозможен, поскольку файлового объекта без имени – только с расширением не существует – нет такого реального файлового объекта, которому могут присваиваться атрибуты.

Как видим, данный подход к реализации разграничительной политики доступа субъектов к объектам универсален, причем не только в отношении файловых объектов – применительно к любым объектам, защищаемым в информационной системе.

Объекты доступа могут задаваться с использованием масок, см. рис.3, применительно к заданию типов файлов, например, исполняемые файлы -маской «*.exe» - любой файл с расширением exe, можно задавать типы файлов и с учетом их расположения, например, «%Windir%*.dll» – любой файл с расширением «dll» из каталога загруженной ОС. Аналогичным образом при помощи масок могут назначаться объекты файловой системы - *.js, *.vbs, *.php и др., которые являются скриптовыми (командными) файлами, при прочтении которых приложение начинает их выполнять.

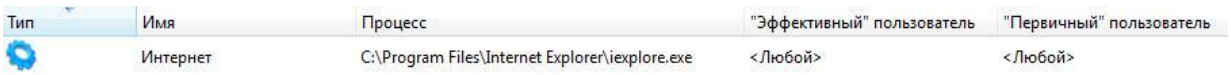
Задач защиты, предполагающих реализацию контроля и разграничения прав доступа к типам файлов, на практике очень много. Например – это локализация среды исполнения на защищаемом объекте. Пример разграничительной политики доступа приведен в табл.1

Таблица 1. 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

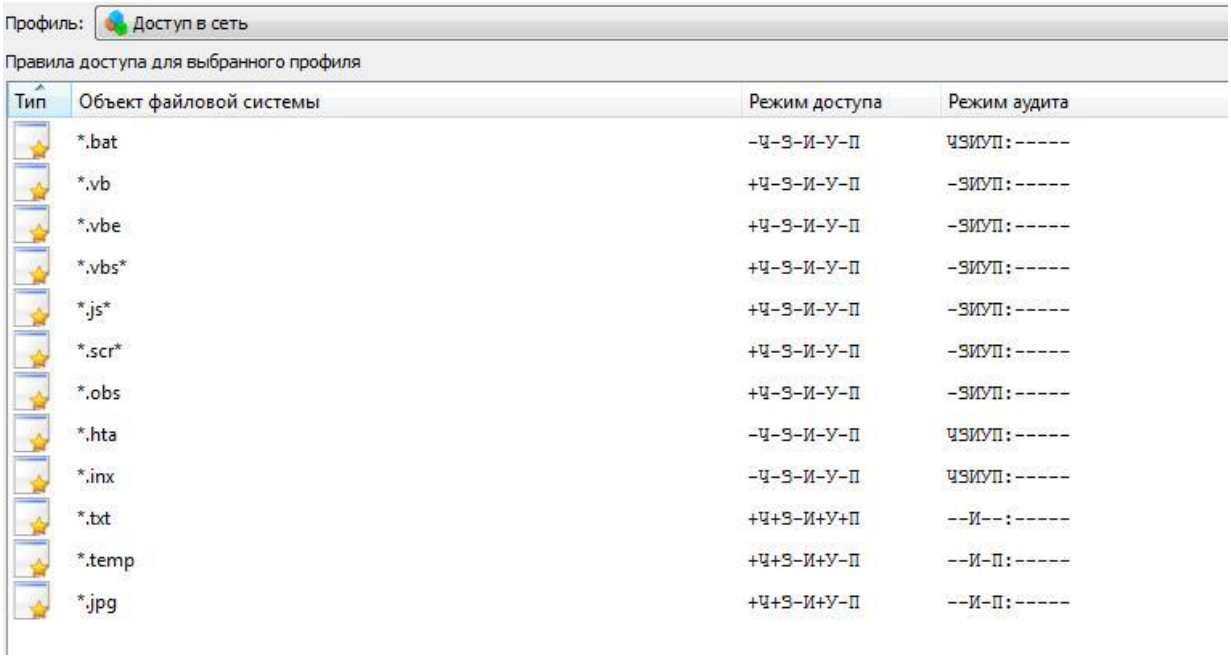
Данными простейшими настройками, где Ч- чтение, З – запись, И – исполнение, У – удаление, П – переименование, см. рис.4, а +/- соответственно разрешение/запрет; разрешается исполнять только перечисленные типы файлов (это реализуется последним правилом). А разрешенные на исполнение файлы запрещается удалять и модифицировать. Дополнительно можно внести правила, запрещающие внесение на компьютер критичных исполняемых файлов, например, «*.com», запретив запись подобных объектов. Чтобы запретить возможность запуска исполняемых файлов с внешних накопителей, объекты файловой системы в Табл.1 должны задаваться следующим образом: %ProgramFiles%*.exe.

Рассмотрим другую крайне важную задачу защиты – защиту от наделения приложений вредоносными свойствами, за счет прочтениями ими вредоносных командных файлов (скриптов) [3]. Защиту реализуем в отношении соответствующего интернет-браузера, для чего создадим соответствующий субъект доступа, см. рис.6, и назначим для него требуемые правила доступа, см. рис.7.



Тип	Имя	Процесс	"Эффективный" пользователь	"Первичный" пользователь
	Интернет	C:\Program Files\Internet Explorer\iexplore.exe	<Любой>	<Любой>

Рис.6. Иллюстрация создания субъекта доступа



Профиль: Доступ в сеть

Правила доступа для выбранного профиля

Тип	Объект файловой системы	Режим доступа	Режим аудита
	*.bat	-Ч-З-И-У-П	ЧЗИУП:-----
	*.vb	+Ч-З-И-У-П	-ЗИУП:-----
	*.vbe	+Ч-З-И-У-П	-ЗИУП:-----
	.vbs	+Ч-З-И-У-П	-ЗИУП:-----
	.js	+Ч-З-И-У-П	-ЗИУП:-----
	.scr	+Ч-З-И-У-П	-ЗИУП:-----
	*.obs	+Ч-З-И-У-П	-ЗИУП:-----
	*.hta	-Ч-З-И-У-П	ЧЗИУП:-----
	*.inx	-Ч-З-И-У-П	ЧЗИУП:-----
	*.txt	+Ч+З-И+У+П	--И--:-----
	*.temp	+Ч+З-И+У-П	--И-П:-----
	*.jpg	+Ч+З-И+У-П	--И-П:-----

Рис.7. Иллюстрация задания правил доступа

В результате реализации данной разграничительной политики доступа приложение Internet Explorer при работе в сети не сможет (причем с любыми правами – «первичный» и «эффективный» пользователи заданы маской «*» - любой, см. рис.6) создать новые критичные (потенциально опасные в части возможности наделения приложения вредоносными свойствами) файлы и модифицировать легальные подобные файлы, присутствующие на компьютере. Для чтения (исполнения) командного файла браузеру сначала его требуется сохранить в системе (что видно с использованием соответствующих средств аудита). К слову сказать, эта разграничительная политика ограничит пользователя и от ненужной ему баннерной рекламы при работе в сети.

В части же контроля доступа к файлам, используемым для хранения данных, также появляется много новых возможностей – можно разрешить пользователям обмениваться только определенными типами файлов, только определенные типы файлов разрешать сохранять на внешних накопителях, а с учетом того, что в субъект доступа включен процесс, можно разрешать отдельным процессам создание только определенных типов файлов, разрешить обмениваться некоторым процессам только

определенными типами файлов (естественно, при этом должны контролироваться и разграничиваться права доступа приложений к буферу обмена, но это уже иная задача защиты, решение которой рассмотрено, например, в [6]).

А теперь сформулируем важнейшее требование к корректности реализации контроля и разграничения прав доступа к типам файлов по их расширениям. При задании объекта доступа расширением файла и запрете его переименования, см. рис.4, должны запрещаться, как переименование файлов с заданным расширением, так и обратно - любых иных файлов в файлы с заданным расширением. Это принципиальное требование, без выполнения которого реализация контроля и разграничения прав доступа к типам файлов по их расширениям теряет смысл, поскольку на компьютере всегда может быть создан некий файл с расширением, с которым разрешено создание файлов, и далее переименован в файл с расширением, с которым создание файлов запрещено. Выполнение данного требование направлено на нивелирование безусловной технологической уязвимости [10].

Техническое решение, выполняющее данное требование нами запатентовано [11], реализовано и апробировано в [9].

Реализация контроля и разграничения прав доступа к создаваемым файлам

Данный метод контроля и разграничения прав доступа был разработан нами с целью принципиального упрощения задачи администрирования и обеспечения корректности реализации разграничительной политики доступа в общем случае, в том числе, в части реализации разграничительной политики доступа к каталогам, используемым для временного хранения файлов [4]. Техническое решение, реализующее данный метод, нами запатентовано [12], реализовано и апробировано в [9]. Суть данного метода контроля доступа состоит в том, что при создании файла, им наследуются (записывается либо непосредственно в файл, либо в альтернативный поток) учетные данные создавшего его субъекта доступа – имя пользователя и процесса (полнопутевое имя исполняемого файла). Это позволяет задавать не правила доступа субъектов к конкретным объектам, а правила доступа субъектов к созданными другими субъектами объектам. Субъекты доступа в данном случае создаются опять же из меню, представленного на рис.1, и отображаются в

интерфейсе, приведенном на рис.2. А вот правила доступа уже задаются из меню, приведенного на рис.8 и отображаются в интерфейсе, представленном на рис.9.

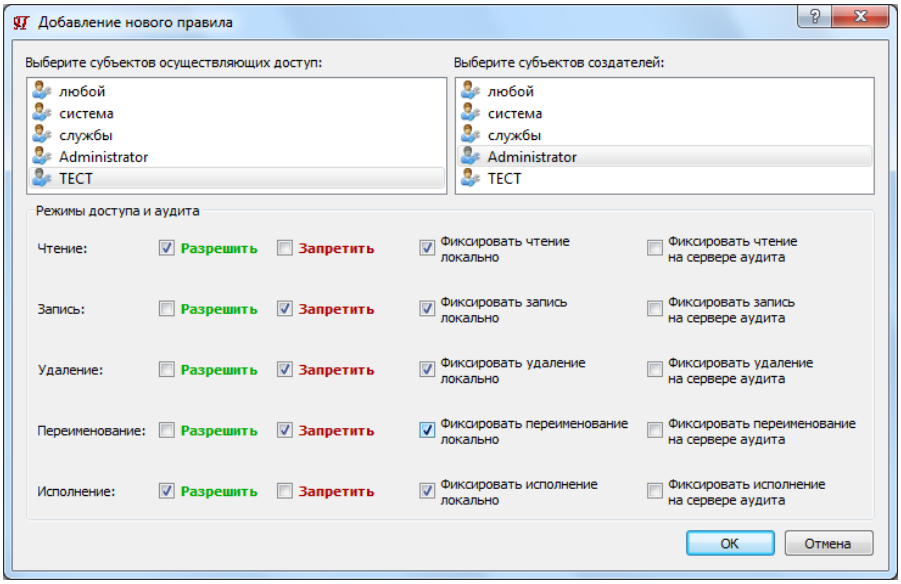


Рис.8. Меню задания правил доступа к создаваемым объектам

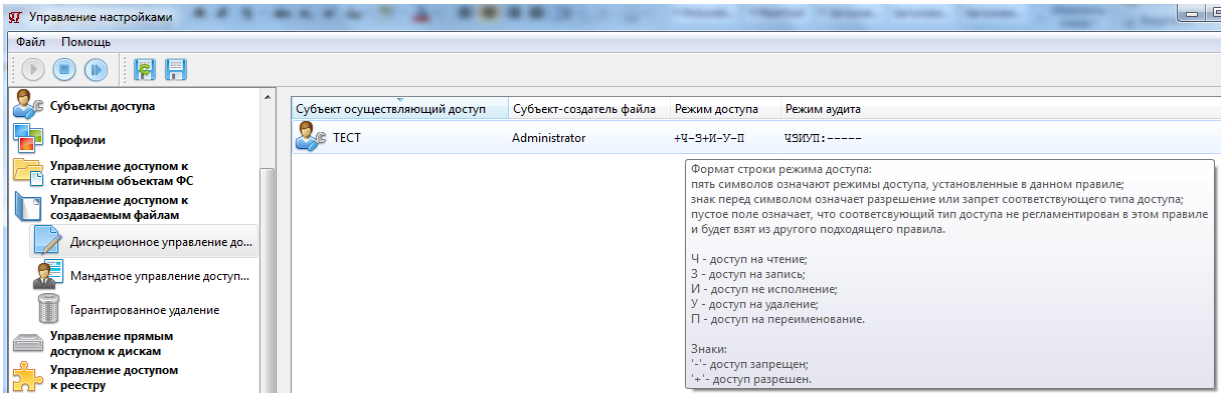


Рис.9. Интерфейс отображения созданных правил для субъекта доступа к создаваемым объектам

Расширение данного метода контроля и разграничения доступа так же возможно в части задания типов файлов их расширениями при задании правил обмена файлами (в данном случае типами файлов) субъектами доступа – пользователями и процессами, для чего в меню, приведенное на рис.8, вводится дополнительное поле для задания масками типов файлов, которыми могут (либо, наоборот, запрещено) обмениваться субъекты доступа, что делается по аналогии с тем, как было описано ранее. Остановимся на рассмотрении вопросов выполнения требования к корректности реализации метода защиты, сформулированного выше,

исходя из того, что реализация данного метода защиты направлена на упрощение задачи администрирования.

В данном случае вопрос с переименованием файлов (расширений файлов) решается следующим образом. При создании файла, им наследуются не только учетные данные создавшего файл субъекта, но и имя (в данном случае нас интересует расширение) созданного файла. Именно эта информация и будет анализироваться механизмом защиты при последующих запросах доступа к созданному (размеченному) подобным образом файлу, которая не будет изменена при любых переименованиях размеченных файлов.

Заключение

В заключение отметим, что рассмотренные в работе методы контроля и разграничения прав доступа, существенно расширяющие возможности реализации защиты информации, в полной мере укладываются в декларируемый нами подход к реализации разграничительной политики доступа - правила доступа должны назначаться не объектам, в качестве их атрибутов, а должны присваиваться субъектам, в качестве их прав доступа к объектам. Именно реализация данного подхода позволяет строить эффективные, универсальные, в части функциональных возможностей, системы защиты информации, соответствующие современным условиям их практического использования.

Литература

1. Щеглов К.А., Щеглов А.Ю. Реализация контроля и разграничения прав доступа к статичным объектам // Вестник компьютерных и информационных технологий - 2015. - № 11. - С. 52-60.
2. Щеглов К.А., Щеглов А.Ю. Защита от атак на уязвимости приложений. Модели контроля доступа // Вопросы защиты информации. - 2013. - Вып. 101. - № 2. - С. 36-43.
3. Щеглов К.А., Щеглов А.Ю. Защита от атак со стороны приложений, наделяемых вредоносными функциями. Модели контроля доступа // Вопросы защиты информации. - 2012. - Вып. 99. - № 4. - С. 31-36.

4. К.А. Щеглов, А.Ю. Щеглов. Принцип и методы контроля доступа к создаваемым файловым объектам // Вестник компьютерных и информационных технологий. - 2012. - № 7. - С. 43-47.

5. 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.

6. Щеглов К.А., Щеглов А.Ю. Технология изолированной обработки данных критичными приложениями // Вопросы защиты информации. - 2015. - Вып. 108. - № 1. - С. 15-22/

7. Щеглов К.А., Щеглов А.Ю. Защита на виртуальные машины от атак, использующих вредоносный код // Вестник компьютерных и информационных технологий - 2014. - № 2. - С. 45-51.

8. Щеглов А.Ю., Щеглов К.А. Система контроля доступа к ресурсам компьютерной системы с субъектом «исходный пользователь, эффективный пользователь, процесс». Патент на изобретение № 2534488. Приоритет изобретения 18.06.2013.

9. Щеглов А.Ю., Паличенко И.П., Корнетов С.В., Щеглов К.А.. Комплексная система защиты информации "Панцирь+" для ОС Microsoft Windows. Свидетельство о регистрации программы для ЭВМ №2014660889 от 17.10.2014.

10. Щеглов К.А. Постановка и подходы к решению задачи защиты информации от несанкционированного доступа в общем виде // Вестник компьютерных и информационных технологий. - 2016. - № 1. - С. 32-44.

11. Щеглов А.Ю., Щеглов К.А. Система разграничения доступа по расширениям файлов. Патент на изобретение №2572385. Приоритет изобретения 15.01.2014.

12. Щеглов А.Ю., Щеглов К.А. Система контроля доступа к файлам на основе их автоматической разметки. Патент на изобретение № 2524566. Приоритет изобретения 18.03.2013.