ТЕХНОЛОГИЯ ВИРТУАЛИЗАЦИИ СИСТЕМ

Введение.

Одной из ключевых задач защиты информационных систем является задача защиты от компрометации системы, в первую очередь, состоящая в защите от удаления и несанкционированной модификации, включая подмену, системных объектов, в том числе, исполняемых файлов системы и приложений, файлов пользовательских конфигураций.

Особенностью современного использования информационных систем является, во-первых, работа, как правило, одного и того же пользователя на одном компьютере в различных режимах обработки данных (сессиях [1]), как правило, характеризуемых различным уровнем конфиденциальности информации, обрабатываемых в различных режимах (например, либо только локально, либо с использованием сетевых ресурсов), как следствие, различными требованиями к защите информации обрабатываемой в различных сессиях, во-вторых, совершенно различным уровнем уязвимости современных приложений, вызванным различными причинами, например, наличием ошибок программирования, возможностью наделения вредоносными свойствами, в результате прочтения ими макросов или апплетов, и т.д. [2,3].

Задача защиты в данных предположениях состоит в изолировании сессий или/и работы приложений, предполагающем минимизацию риска компрометации системы (либо потерь от подобной компрометации) в результате реализации успешной атаки на критичную сессию, либо на критичное приложение. Естественно, что реализация подобной разграничительной политики доступа - настройка прав доступа к системным файлам и к файлам пользовательских конфигураций, представляет собою достаточно сложную задачу, что не может не сказываться на эффективности реализуемой защиты.

1. Возможность и недостатки использования виртуальных машин для решения рассматриваемой задачи защиты информационных систем.

Возможным вариантом решения рассматриваемой задачи защиты является использование виртуальной машины – программы, которая полностью эмулирует реальный (физический) компьютер со всеми его компонентами (жёсткий диск, сетевые адаптеры и т.д.). На каждый такой виртуальный компьютер можно установить операционную систему, драйверы, прикладные программы и т.д. Таким образом, на одном реальном компьютере можно сформировать несколько виртуальных компьютеров.

С точки зрения использования виртуальных машин для повышения уровня информационной безопасности, такая возможность позволяет решать рассматриваемую задачу защиты - реализовать разделенную (сессионную) обработку информации на компьютере между различными пользователями (либо различными учетными записями, создаваемыми для одного пользователя для работы в различных сессиях), в предположении, что для каждого пользователя (сессии) создается свой виртуальный компьютер. При этом системные объекты, в том числе, исполняемые файлы системы и приложений, файлы конфигурации, между виртуальными компьютерами (сессиями) полностью изолируются – не могут повреждаться (подменяться) для одной сессии, при успешной атаки на другую сессию. В простейшем случае примером может служить создание отдельного виртуального компьютера для работы с внешней сетью – вероятность успешной атаки на него будет значительно выше, чем вероятность успешной атаки на другой виртуальный компьютер, используемый для локальной обработки данных. Наиболее вероятная атака приведет в этом случае к компрометации системы только одного наиболее критичного виртуального компьютера, используемого для работы с сетью, остальные виртуальные компьютеры останутся работоспособными и смогут далее использоваться для решения соответствующих задач в защищенной информационной системе.

Однако использование для решения рассматриваемой задачи защиты виртуальной машины крайне избыточно и связано с существенными недостатками, к которым, в первую очередь, можно отнести следующее:

- высокая стоимость решения, т.к. нужно приобрести собственно виртуальную машину и несколько (по числу сессий) операционных систем, соответствующих приложений (которые используются несколькими сессиями),
- высокая загрузка вычислительного ресурса, т.к. одновременно запускается несколько (по числу сессий) операционных систем, повышенные требования к оборудованию (например, для каждого виртуального компьютера создается свой виртуальный жесткий диск),
- высокая трудоемкость администрирования, т.к. каждую операционную систему отдельно необходимо настроить по соответствующим требованиям информационной безопасности.

Все эти недостатки усугубляются в том случае, если задача защиты должна решаться дополнительными средствами защиты – для каждой операционной системы при этом потребуется установить и настроить средство защиты (число одновременно запускаемых на виртуальной машине средств защиты определяется числом организуемых виртуальных компьютеров).

Все это обусловливает актуальность поиска иных решений рассматриваемой важнейшей задачи защиты.

2. Метод перенаправления запросов доступа к неразделяемым файловым объектам.

Важнейшей проблемой корректности реализации разграничительной политики доступа к файловым объектам является наличие в системе неразделяемых между субъектами доступа объектов - папок (каталогов), например, каталогов, используемых для временного хранения файлов [4].

Решением рассматриваемой проблемы при построении системы защиты в общем виде является реализация метода переадресации запросов

доступа к объектам файловой системы (папкам), не разделяемым системой и приложениями между субъектами доступа, который состоит в следующем. Для каждого субъекта доступа (в простейшем случае, пользователя) для неразделяемого объекта в системе создается 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 каким-либо пользователем информации в неразделяемый каталог «Общий ресурс» (соответственно, при чтении из этого каталога), запрос доступа перенаправляется в соответствующий каталог пользователя, запросившего доступ. Например, если текущим пользователем является первый пользователь, то при записи в каталог «Общий ресурс», данный запрос доступа будет перенаправлен в каталог «Общий ресурс 1», если второй пользователь – то в каталог «Общий ресурс 2». Отметим, что при данной политике перенаправления запросов доступа каталог «Общий ресурс» становится виртуальным – в него пользователи не могут записать данные (создать файл), соответственно из него не могут и прочитать данные – любое обращение к этому каталогу пользователями соответствующим образом будет переадресовано.

Важным является то, что механизм перенаправления запросов к неразделяемым системой файловым объектам обрабатывает запрос до механизма контроля (разграничения) прав доступа к файловым объектам. Механизмом контроля доступа к файловым объектам уже разграничиваются права доступа к каталогам, в том числе и к тем, в которые перенаправляется запрос доступа, 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, как следствие, реализовать корректную разграничительную политику доступа к файловым объектам.

Техническое решение, реализующее рассмотренный метод перенаправления запросов доступа к неразделяемым файловым объектам,

авторами запатентовано [5], практически реализовано в КСЗИ «Панцирь+» для ОС Microsoft Windows (далее будем использовать для иллюстраций интерфейсы, разработанные для этой системы защиты) [6] и апробировано.

Интерфейс задания и отображения заданных правил перенаправления запросов доступа к файловым объектам, реализованные в данной системе защиты, проиллюстрирован на рис.1.

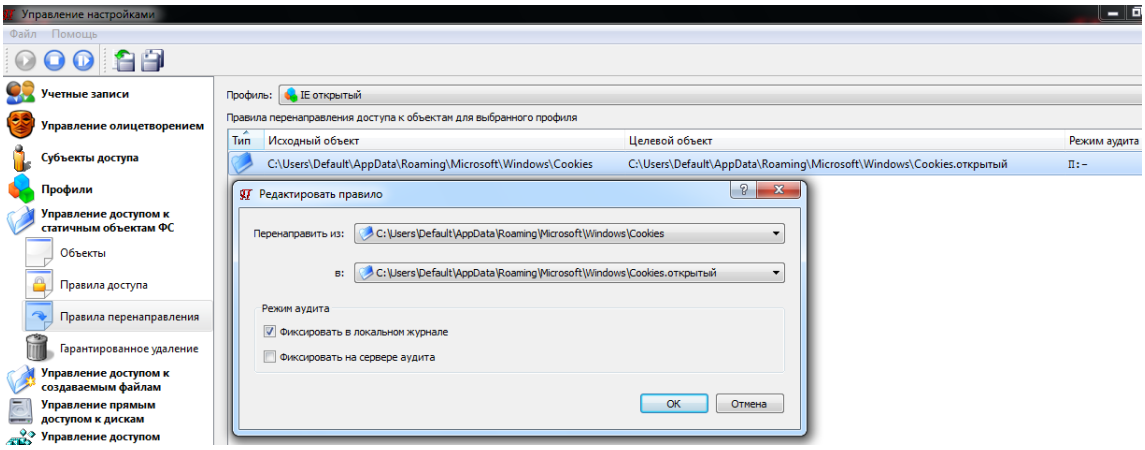

Рис.1. Интерфейс задания и отображения заданных правил перенаправления запросов доступа к файловым объектам

В качестве объектов доступа, в отношении которых будет применено заданное правило перенаправления запросов доступа, могут выступать файлы, каталоги, логические диски.

Теперь, что касается задания субъектов доступа. Для упрощения задачи администрирования в качестве субъекта доступа в правилах перенаправления используется сущность "профиль", см. рис.1. В один профиль объединяются субъекты, для которых назначаются одинаковые правила перенаправления к файловым объектам.

Интерфейс задания и отображения заданных субъектов доступа проиллюстрирован на рис.2.

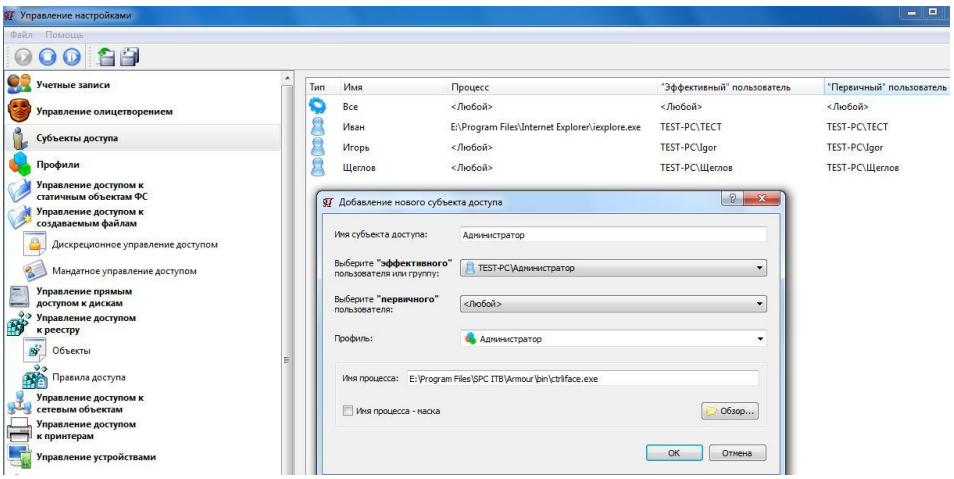

Рис.2. Интерфейс задания и отображения заданных правил перенаправления запросов доступа

С учетом того, что может быть актуальной задача (в зависимости от решаемой задачи защиты) разделения файловых объектов, как между пользователями (учетными записями – задаются своими SID), так и между процессами, субъект доступа задается сущностью «пользователь, процесс» (учитывается то, какой пользователь каким процессом запрашивает доступ к файловому объекту), см. рис.2. Субъект доступа процесс задается именем исполняемого файла процесса, можно задавать именем каталога, тогда задаваемые правила будут распространяться на все процессы, исполняемые из данного каталога. Можно использовать маски. Например, при использовании маски «Все» (*) при задании процесса, правила будут задаваться только для пользователей. Аналогично, при задании пользователя маской «Все» (*), правила будут задаваться для процессов.

Для защиты от обхода задаваемых правил, сущность «пользователь» в субъекте доступа расширена – пользователь задается сущностью «исходный, эффективный пользователь», см. рис.2. Исходный пользователь (учетная запись, идентифицируемая SID) – это пользователь, от лица которого запущен процесс (он запоминается средством защиты при запуске процесса), эффективный пользователь – это пользователь, от лица которого процесс уже непосредственно обращается к защищаемому ресурсу. При запросе доступа к

ресурсу средством защиты анализируются идентификаторы исходного и эффективного пользователей, для процесса, запросившего доступ, на соответствие заданным правилам смены пользователя при доступе к ресурсу. Данное решение также запатентовано [7]).

В порядке замечания отметим, что возможность смены пользователя запущенным процессом реализуется, в том числе, штатными возможностями современных ОС – сервисами олицетворения [8].

Рассмотренный метод защиты был разработан применительно к решению вполне конкретной задачи – для разделения между субъектами доступа файловых объектов для возможности корректного разделения между ними обрабатываемых в системе данных. Однако в общем случае рассмотренным техническим решением между субъектами доступа (пользователями и/процессами) могут разделять любые файловые объекты, в том числе, исполняемые файлы, различные системные и конфигурационные файлы. Это позволяет рассмотреть возможность использования рассмотренного метода для решения принципиально иной задачи – задачи виртуализации системы.

3. Технология виртуализации систем.

Под виртуальной системой – будем понимать систему, состоящую из копий системных файлов, создаваемую для субъекта доступа, сессия которого изолируется, на которую перенаправляются запросы доступа субъекта к оригинальной (к базовой) системе.

Сначала рассмотрим предлагаемую технологию в общем случае. Состоит она в следующем. Пусть исходно на системном диске - на диске C: установлена ОС и приложения, и пусть в системе заведены два интерактивных пользователя User 1 и User 2, сессии которых требуется разделить.

Создадим копии системного диска C: на дисках D: и E: - скопируем соответствующие системные и скрытые файлы, что достаточно просто сделать. С учетом выполненной процедуры копирования, далее будем

говорить, что на диске C: установлена базовая система, на дисках D: и E: - созданные нами виртуальные системы.

Заметим, что виртуальные системы не обязательно создавать на отдельных дисках, можно их создать в отдельных каталогах того же диска C:.

Теперь зададим правила перенаправления запросов к файловым объектам. Для пользователя User 1 это будет правило перенаправления доступа к диску C: на диск D:, для User 2 - правило перенаправления доступа к диску C: на диск E:. Реализуя разграничительную политику доступа к файловым объектам, запретим пользователю User 1 доступ к диску E:, а пользователю User 2 – к диску D:. В результате этого получим схему виртуализации системы, приведенную на рис.3.

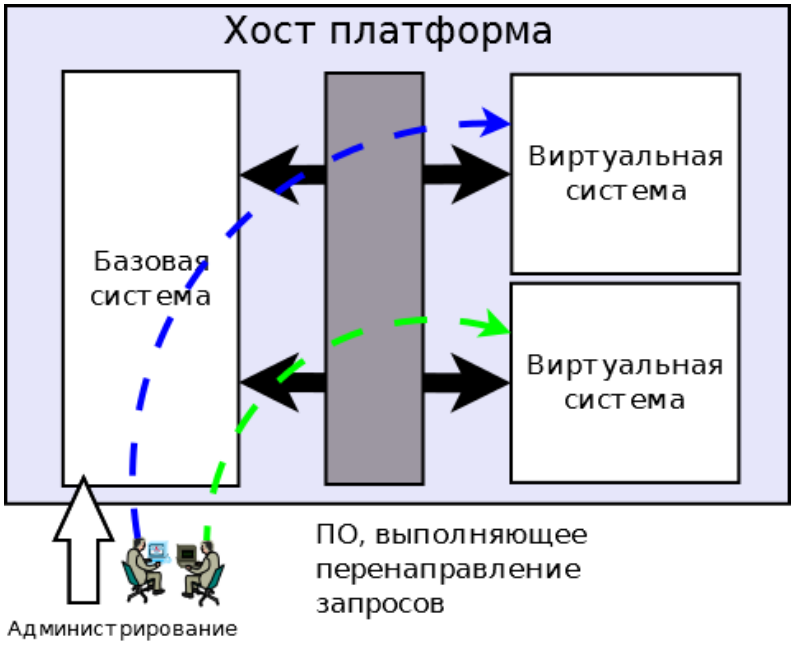

Рис.3. Схема виртуализации системы

Рассмотрим, как работает подобная система. С базовой системой может взаимодействовать только системный пользователь, запросы к базовой системе интерактивных пользователей User 1и User 2 перенаправляются к соответствующим виртуальным системам. Т.е., если пользователь User 1 запускает с базовой системы какое-либо приложением, например, браузер, реально оно будет запущено с диска D:. При последующей работе

приложения любое его обращение с правами пользователя User 1 к базовой системе будет перенаправляться к соответствующей виртуальной системе. При этом приложение, инсталлированное на диск C:, будет работать корректно, поскольку все перенаправления его запросов доступа «прозрачны» для приложения, в частности, корректно будут сохраняться временные данные, cookies, настройки, кэшированная информация и т.д.

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

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

Достоинства применения данной технологии для решения рассматриваемой задачи защиты достаточно очевидны и обусловлены они тем, что используется только одна операционная система (базовая система) и только одно средство защиты, реализующее перенаправление запросов доступа к файловым объектам, см. рис.3, что практически не приводит к

дополнительной загрузке вычислительных ресурсов и практически не усложняет задачу администрирования.

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

Частный случай виртуализации системы состоит в реализации ее виртуализации для субъекта доступа процесс. Проиллюстрируем это на примере, для чего рассмотрим, как, используя данный подход изолировать работу браузера - реализовать отдельную сессию работы с сетью. Пусть опять же на диске C: установлена базовая система, сделаем ее копию на диске D:. Теперь зададим разграничительную политику - для субъекта, определяемого, как исполняемый файл интересующего нас браузера зададим правило перенаправления запросов к файловым объектам к диску C: на диск D:, кроме того, запретим браузеру доступ к диску C: (соответственно и здесь в качестве субъекта доступа должна использоваться сущность "процесс").

В результате этого работа пользователя с различными приложениями становится полностью "прозрачной" - под одной и той же учетной записью, однако базовая ОС защищена от потенциально возможных атак на соответствующий браузер - браузер может повредить или модифицировать лишь копию системы.

Заключение.

В заключение еще раз акцентируем внимание на том, что в работе предложена не некая абстрактная технология, а реализованное на примере ОС Microsoft Windows решение, апробация которого позволяет утверждать о реализуемости предложенной технологии виртуализации систем и об эффективности ее использования для решения рассмотренной в работе задачи защиты информационных систем.

Литература.

1. Щеглов К.А., Щеглов А.Ю. Метод сессионного контроля доступа к файловым объектам. Вопросы практической реализации // Вестник компьютерных и информационных технологий. - 2014. - № 8. - С. 54-60.

2. 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.

3. 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.

4. Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. - 2004. - 384 с.

5. Щеглов А.Ю., Щеглов К.А. Система переформирования объекта в запросе доступа. Патент №2538918 от 24.11.2014.

6. Щеглов А.Ю. и др. Комплексная система защиты информации "Панцирь+" для ОС Microsoft Windows. Свидетельство о регистрации программы для ЭВМ №2014660889 от 17.10.2014.

7. Щеглов А.Ю., Щеглов К.А. Система контроля доступа к ресурсам компьютерной системы с субъектом "исходный пользователь, эффективный пользователь, процесс". Патент №2534488 от 03.10.2014.

8. Щеглов К.А. Щеглов А.Ю. Метод и средство контроля и разграничения прав доступа к сервисам олицетворения // Вестник компьютерных и информационных технологий. - 2015. - №3. - С.48-54.