# МЕТОД И СРЕДСТВО КОНТРОЛЯ И РАЗГРАНИЧЕНИЯ ПРАВ ДОСТУПА К СЕРВИСАМ ОЛИЦЕТВОРЕНИЯ

**Введение.**

Атаки класса повышения привилегий несут в себе очень серьезную угрозу безопасности для информационной системы. Связана подобная угроза с потенциальной возможностью обхода разграничительной политики доступа к ресурсам, назначаемой администратором, за счет получения процессом (либо потоком, порождаемым процессом) прав более привилегированного пользователя для последующего обращения с этими правами к защищаемым ресурсам. Одну из широко используемых на практике группу атак, связанных с возможностью запуска программы (вредоносной, либо программы, которая может наделяться вредоносными свойствами [1,2] с системными правами, за счет эксплуатации уязвимостей в системном процессе, либо драйвере, а также метод эффективной защиты от подобных атак мы рассмотрели в [3]. Однако для повышения привилегий совсем не обязательно использовать уязвимости, тем более в системных процессах, это можно сделать, используя штатные возможности современных универсальных ОС, в том числе, ОС семейства Microsoft Windows – возможности сервисов олицетворения – олицетворения потока, порождаемого процессом, запущенным одним пользователем, с правами другого пользователя, в результате чего потенциально может произойти изменение привилегий (порождаемый процессом поток получает права доступа иного пользователя), что несет в себе угрозу обхода заданной разграничительной политики доступа к защищаемым ресурсам. Поскольку это штатная возможность ОС, к ее использованию могут быть разграничены права доступа, т.е. может быть реализована разграничительная политика доступа, целью которой является предотвращение возможности повышения привилегий.

**1. Использование сервисов олицетворения современными ОС и приложениями.**

Проведем исследование на примере ОС семейства Windows. В широко распространенных ОС семейства Windows для идентификации субъектов, выполняющих в системе различные действия, используются не имена (которые могут и не быть уникальными), а идентификаторы защиты (security identifiers, SID). SID имеются у пользователей, локальных и доменных групп, локальных компьютеров, доменов и членов доменов. Все работающие в системе процессы и потоки выполняются в контексте защиты того пользователя, от имени которого они так или иначе были запущены, а для идентификации контекста защиты процесса или потока используется объект, называемый маркером доступа (access token). В процессе регистрации в системе создается начальный маркер, представляющий пользователя, который входит в систему, и сопоставляет его с процессом оболочки, применяемой для регистрации пользователя.

Маркер может быть основным (идентифицирует контекст защиты процесса) или олицетворяющим (применяется для временного заимствования потоком другого контекста защиты — обычно другого пользователя). Олицетворение (impersonation) — средство, используемое в модели защиты Windows, предоставляющее возможность отдельному потоку выполняться в контексте защиты отличном от контекста защиты процесса, т.е. действовать от лица другого пользователя [4].

Сервисы олицетворения на самом деле системой используются достаточно широко, например, они применяются 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.1 [4].

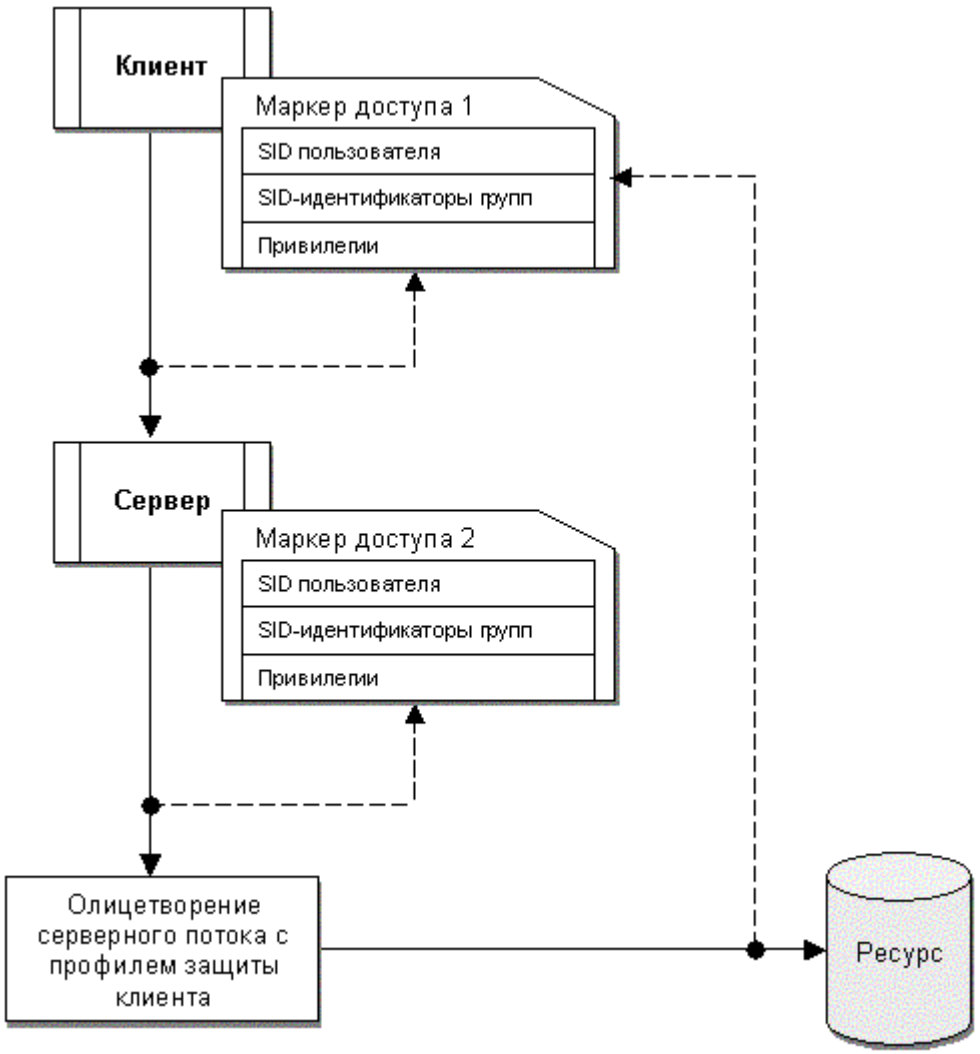


**Рис. 1. Обслуживание клиентского запроса на доступ к ресурсу с использованием олицетворения**

Обычно серверу доступен более широкий круг ресурсов, чем клиенту, и при олицетворении сервер может терять часть исходных прав доступа (штатный режим использования сервисов олицетворения). Вместе с тем, в результате олицетворения программа, в которой заложена соответствующая возможность использования сервисов олицетворения, 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, в том числе такой, как учетная запись администратора или системы), достаточного для осуществления несанкционированного доступа к

конфиденциальным данным в обход реализованной разграничительной политики доступа к ресурсам.

Рассмотрим еще несколько примеров использования сервисов олицетворения. Системный процесс winlogon, отвечающий за идентификацию и идентификацию пользователя при входе в систему, изначально запускается с системными правами (пользователь еще не определен), в результате идентификации этот процесс олицетворяет себя с соответствующим пользователем – оболочка уже запускается с соответствующими пользовательскими правами. Этот пример характерен для многих системных процессов, запускаемых при старте ОС, а далее используемых для работы с правами соответствующих пользователей. Другой пример – это запуск процесса (приложения) с правами другого пользователя, например, с использованием утилиты runas. Именно использование сервисов олицетворения позволяет запускать приложения под другой учетной записью без смены учетной записи и без перезагрузки компьютера. Можно рассмотреть и ряд других примеров. Все эти примеры иллюстрируют то, что сервисы олицетворения на практике широко используются, как собственно системой, так и приложениями (например, клиент-серверными), причем доступны для разработчиков ПО.

## 2. Реализация контроля и разграничения прав доступа к сервисам олицетворения.

Поскольку сервисы олицетворения широко используются на практике, как системным, так и прикладным ПО, то в общем случае запретить к ним доступ не представляется возможным (далее мы это проиллюстрируем примером) – необходимо реализовывать разграничение прав доступа. С учетом же того, что различными программами могут требоваться различные права доступа к смене учетной записи пользователей, в качестве субъекта доступа следует рассматривать сущность процесс. Правилами же доступа к сервисам олицетворения администратором задается из какой учетной записи в какую разрешено (разрешительная политика доступа), либо запрещено

(запретительная политика доступа) олицетворяться контролируемому субъекту доступа, для которого назначаются правила олицетворения, т.е. в качестве объектов доступа здесь выступают исходная для субъекта учетная запись пользователя (учетная запись, от лица которой запущен процесс), применительно к которой контролируемому субъекту разрешаются/запрещаются олицетворения, и целевая учетная запись пользователя, в которую контролируемому субъекту разрешаются/запрещаются олицетворения.

Реализация и апробация рассмотренного метода контроля и разграничения прав доступа к сервисам олицетворения в комплексной системе защиты информации «Панцирь+» для ОС Microsoft Windows (разработчик ЗАО «НПП «Информационные технологии в бизнесе») позволяют нам при изложении технической реализации данного метода защиты использовать интерфейсы разработанного средства.

Интерфейс настройки механизма защиты приведен на рис.2.

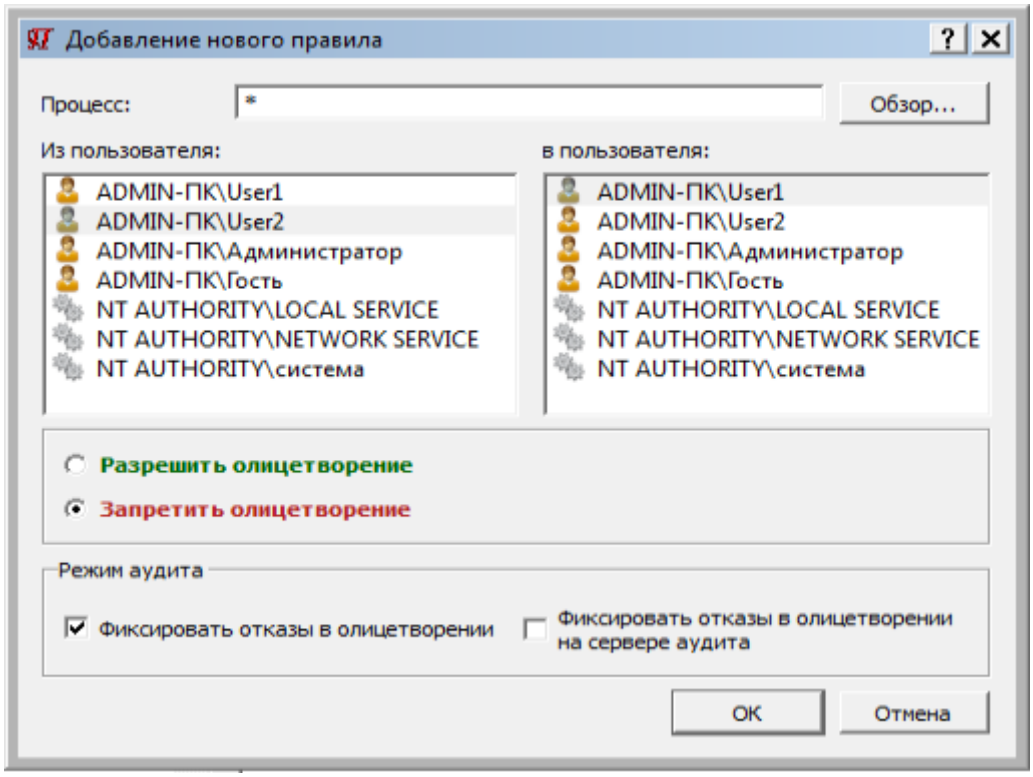


**Рис.2. Интерфейс настройки механизма защиты**

Субъект доступа задается в интерфейсе в поле «Процесс». Его возможно задать, как конкретным процессом - указанием его полнопутевого имени (полнопутевого имени исполняемого файла процесса), так и воспользоваться масками, задав определенную группу процессов, в том числе маской «*», обозначающей любой процесс. Если один и тот же субъект в правилах «порывается» несколькими способами, что возможно за счет применения масок, правило для анализа выбирается диспетчером по более точному описателю в нем субъекта, запрашивающего доступ к сервисам олицетворения. Далее для заданного субъекта доступа (процесса или маски) необходимо выбрать пользователей, между которыми необходимо разрешить, либо запретить олицетворение – задать правила доступа к сервисам олицетворения. В выпадающем списке пользователей присутствуют как интерактивные, заданные в средстве защиты пользователи, так и системные пользователи, присутствующие в системе.

Заданные правила олицетворения (разграничительная политика к сервисам олицетворения), будут отображены в интерфейсе, см. рис.3.

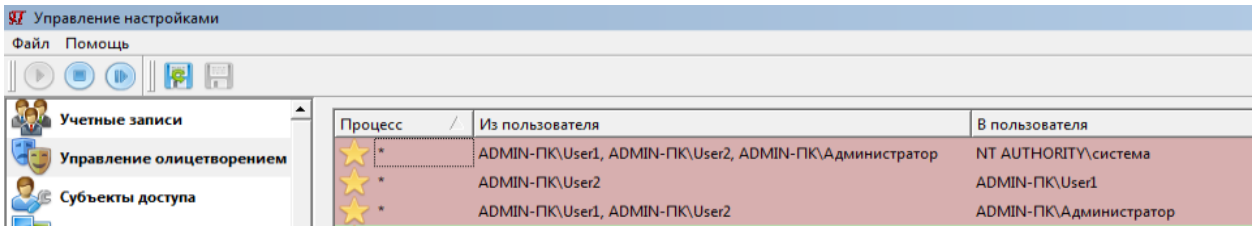


**Рис.3. Отображение в интерфейсе заданной разграничительной политики к сервисам олицетворения**

Субъектом доступа выступает любой процесс – задан маской «*». Первое правило в этой политике, см. рис.3, реализует запрет олицетворения интерактивных пользователей с системный пользователем. Второе правило – запрет олицетворения пользователя User2 с User1. Третье правило – запрет олицетворения интерактивных пользователей с учетной записью администратора.

Теперь рассмотрим обещанный ранее пример, наглядно иллюстрирующий необходимость использования в качестве субъекта

доступа в разграничительной политике сущности «процесс». Приведенная на рис.3 разграничительная политика доступа неработоспособна – компьютер будет не загрузить. Препятствует этому первое важнейшее правило, предотвращающее возможность олицетворения любого процесса, запущенного с правами интерактивного пользователя, с системными правами. Противоречие же состоит в том, что системному процессу svchost для корректной работы ОС требуется не только олицетворение системного пользователя с интерактивным, но и обратное олицетворение – интерактивного пользователя с системным. Как следствие, для обеспечения корректной работы ОС в разграничительную политику, приведеную на рис.3, требуется ввести отдельное правило для процесса svchost, проиллюстрированное на рис.4, разрешающее соответствующее обратное олицетворение.

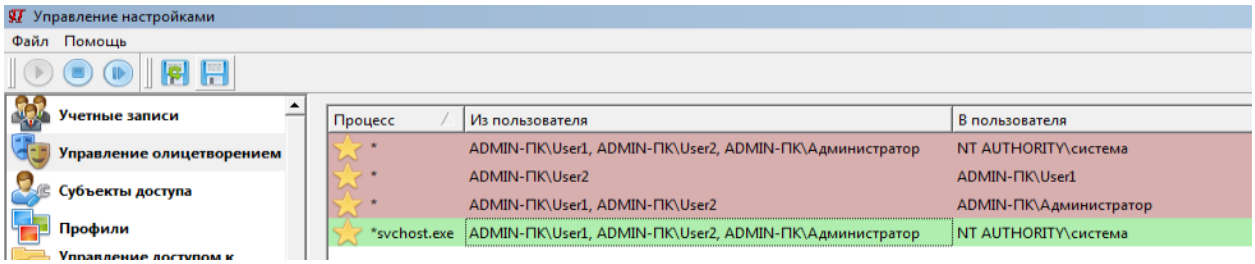


**Рис.4. Отображение в интерфейсе разграничительной политики с включением правила олицетворения для системного процесса svchost**

Акцентируем внимание на следующем моменте. Из разграничительной политики, приведенной на рис.4, видим, что процесс svchost критичен в отношении атак на повышение привилегий – для необходимо разрешить олицетворение интерактивного пользователя в системного, как следствие, исполняемый файл данного системного процесса должен быть защищен от модификации. В разграничительной политике, приведенной на рис.4, соответствующий субъект доступа нами задан маской *svchost.exe, что с рассматриваемой точки зрения (зашиты исполняемого файла) не корректно – данный субъект доступа должен задавать полнопутевым именем его исполняемого файла, например, C:\Windows\System32\ svchost.exe или с

использованием переменных среды окружения %Windows%\System32\ svchost.exe (то же задание, но применительно к запущенной ОС).

## 3. Требования к корректности задаваемых правил олицетворения.

Рассмотрим, в чем состоит повышение привилегий и сформулируем требования к корректности правил олицетворения применительно к защите от атак на повышение привилегий.

Интуитивно понятно, что недопустимо (при условии, что это не нарушает работоспособности ОС или приложения) разрешать олицетворение непривилегированного интерактивного пользователя с системными правами и правами администратора, причем для любого процесса. Сформулируем соответствующее правило в общем случае. При этом правила олицетворения, задаваемые для контролируемого субъекта доступа (для различных субъектов правила могут различаться) будем отображать в виде матрицы олицетворений I, имеющей следующий вид:

$$I = \begin{array}{c} \\ C1 \\ C2 \\ \vdots \\ Cl-1 \\ Cl \end{array} \begin{array}{c} \begin{array}{ccc} C1 & C2\ldots\ldots\ldots & Cl \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ & \ldots\ldots\ldots\ldots\ldots\ldots & \\ & \ldots\ldots\ldots\ldots\ldots\ldots & \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

Условимся строками матрицы обозначать исходные, а столбцами – целевые имена пользователей. Будем обозначать Cj(1)Ci разрешение изменения исходного имени пользователя Cj на целевое имя Ci, i=1,…,l; j=1,…,l, j≠i. Запрет соответствующего олицетворения соответственно будем обозначать следующим образом: Cj(0)Ci.

Задание правил олицетворения для субъекта доступа состоит в расширении канонической [5] матрицы олицетворений, задаваемой следующим образом: Cj(1)Ci, j=i, Cj(0)Ci, j≠i (только на главной диагонали матрицы присутствуют «1»). Требования к корректности правил,

реализующих расширение канонической матрицы олицетворений, определяемых условием Cj(1)Ci, j≠i, нам и требуется сформулировать.

Для оценки безопасности системы контроля и разграничения прав доступа к защищаемым ресурсам (будем рассматривать на примере файловых объектов – для различных ресурсов различаются назначаемые права доступа) – безопасности реализации разграничительной политики доступа, защищенной от ее обхода, на практике используется модель «Харрисона-Уззо-Ульмана» [6]. Если считать, что множества C = {C1,…, Cl} и O = {O1,…, Ok} – соответственно линейно упорядоченные множества субъектов и объектов доступа, а R = {R1,…, Rm} конечное множество прав доступа (чтение (r), запись (w), удаление (d), исполнение (x) и т.д.), то разграничительная политика доступа субъектов к объектам описывается матрицей доступа M, где M[C,O] –ячейка матрицы, содержит набор прав доступа субъекта из множества C = {C1,…, Cl} к объекту из множества O = {O1,…, Ok}:

$$M = \begin{array}{c} \\ C1 \\ C2 \\ . \\ . \\ . \\ Cl\text{-}1 \\ Cl \end{array} \begin{array}{c} \begin{array}{ccc} O1 & O2\ldots\ldots\ldots & Ok \end{array} \\ \begin{bmatrix} r,w,d & w,d & 0 \\ r & r,w,d & 0 \\ & \ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ & \ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ 0 & 0 & r \\ 0 & w & r,w,d \end{bmatrix} \end{array}$$

В любой момент времени система описывается своим текущим состоянием Q = (C,O,M). Требование к безопасности системы сформулируем следующим образом: «Для заданной системы состояние Q0 = (C0,O0,M0) считается безопасным относительно некоторого права доступа из множества R, если не существует применимой к Q0 последовательности действий, в результате выполнения которых субъектом C0 приобретается право доступа из множества R к объекту O0, исходно отсутствующее в ячейке матрицы M0[C0,O0]». Если же некое право доступа из множества R, отсутствующее в ячейке матрицы M0[C0,O0], в результате некоторой последовательности

действий может быть приобретено субъектом C0, то будем говорить, что возможна утечка соответствующего права доступа из множества R, в результате выполнения некоторой последовательности действий, а система небезопасна относительно этого права доступа из множества R.

С учетом всего сказанного, в общем случае под повышением привилегий будем понимать приобретение в результате некоторой последовательности действий субъектом в отношении какого-либо объекта некоторого права доступа из множества R.

В нашем случае в качестве некоторой последовательности действий следует рассматривать олицетворение субъекта доступа в соответствии с разрешенным правилом, расширяющим каноническую матрицу олицетворения: Cj(1)Ci, j≠i.

Замечание. Для иллюстрации обратимся к приведенной выше матрице доступа M. Рассмотрим с точки зрения построения безопасной системы – для простоты, в отношении реализации прав доступа только к объекту O1, включение в каноническую матрицу олицетворений правил олицетворения C1(1)C2 и C2(1)C1. Как видим, правило C1(1)C2 не приводит к повышению привилегий – при этом субъект доступа C1 лишь потеряет права доступа к объекту O1 (w и d). Правило же C2(1)C1 наделит субъект доступа C2 дополнительными правами доступа (w и d) к объекту O1, т.е. его привилегии в отношении доступа к объекту O1 в этом случае будут повышены, в результате чего имеет место утечка прав доступа (w и d) субъекта C2 к объекту O1. Как следствие, данное правило доступа некорректно.

**Лемма.** Правило олицетворения, расширяющее каноническую матрицу олицетворения: Cj(1)Ci, j≠i, корректно в том случае, если множество прав доступа субъекта Cj к любому объекту из множества O = {O1,…, Ok} совпадает или является подмножеством прав доступа субъекта Ci к соответствующему объекту из множества O = {O1,…, Ok}.

**Доказательство.** Только при выполнении данного условия, введение в каноническую матрицу олицетворения правила олицетворения: Cj(1)Ci, j≠i

не приведет к повышению привилегий субъекта Cj, т.к. при этом в отношении всех объектов из множества O = {O1,…, Ok} для субъекта Cj отсутствует утечка прав доступа из множества R. Лемма доказана.

Сформулированная лемма и определяет требования к корректности правил олицетворения применительно к защите от атак на повышение привилегий.

Включение в каноническую матрицу олицетворения правила, выполняющего данное требование, будем считать корректным и безопасным.

Замечания.

1. Сформулированные требования к корректности и безопасности правил олицетворения применительно к защите от атак на повышение привилегий актуальны при реализации дискреционного метода контроля к защищаемым ресурсам, причем, как к статичным, так и к создаваемым, в том числе, файловым объектам [7,8].

2. Применительно к реализации мандатного метода контроля доступа, в том числе, и к создаваемым объектам, прежде всего, файловым [9], корректным и безопасным будет разрешение права олицетворения субъектов (для мандатного метода – пользователей), которым назначена одна и та же метка безопасности (уровень доступа).

## 4. Дополнительные возможности механизма контроля и разграничения прав доступа к сервисам олицетворения.

Основной задачей механизма контроля и разграничения прав доступа к сервисам олицетворения является защита от атак на повышение привилегий с использованием штатной возможности современных универсальных ОС – сервисов олицетворения. Вкратце остановимся на рассмотрении двух достаточно важных дополнительных возможностей защиты, реализуемых данным механизмом.

Первая из них обусловливается возможностью разграничивать права доступа к сервисам олицетворения для системных процессов. Как отмечали ранее, олицетворение системных процессов широко используется в модели

безопасности Windows. Приведем пример практического использования этой возможности. Разрешим процессу winlogon олицетворение из системного пользователя, под которым он запускается, только в пользователя User1. При этом получаем следующее дополнительное свойство защиты. Под любой иной, кроме как User1 учетной записью, в том числе, и несанкционированно созданной в системе, вход в систему становится невозможен.

Другая возможность, которую мы рассмотрим в работе, связана с применением в компьютерной системе сессионного контроля доступа, требования к корректной реализации которого сформулированы в [10]. Сессионный контроль доступа предназначен для обеспечения возможности обработки одним и тем же пользователем на одном и том же компьютере различного рода информации – информации различных категорий конфиденциальности, различного назначения, обусловливаемого соответствующими ролями пользователя в информационной системе и т.д. При этом должна быть реализована разделительная политика доступа [10], позволяющая изолировать различные режимы обработки информации. В [10] обосновано, что сессионный контроль доступа корректно будет реализован только в том случае, если для работы пользователя в каждой сессии создается своя учетная запись, как следствие, должна решаться задача построения разделительной политики доступа между учетными записями к защищаемым ресурсам, предотвращающая несанкционированной доступ из одной сессии (режима обработки) к данным, обрабатываемым в другой сессии. При практической реализации данного требования достаточно важным является решение задачи корректной смены сессии – смены учетной записи. Проблема при этом состоит в том, что многие механизмы защиты, например, разграничение доступа к буферу обмена, реализуются только при полной перезагрузке системы, либо при смене пользователя. При запуске же приложения с правами иного пользователя, например, утилитой runas, основанного, как отмечали выше, на использовании сервисов олицетворения, данные механизмы защиты ОС не работают – соответствующих

разграничений прав доступа не осуществляется (можете провести соответствующий эксперимент на примере буфера обмена и убедиться в сказанном). Таким образом, использование сервисов олицетворения для смены сессии не должно использоваться при реализации сессионного контроля доступа. Эту задачу позволяет решать рассмотренный механизм защиты, настройка которого в этом случае состоит в предотвращении, в частном случае для субъекта доступа, определяемого утилитой runas, в общем случае для любого процесса, определяемого маской «*» (что является корректным решением), олицетворения интерактивных пользователей – учетных записей, задаваемых в системе для работы пользователя в различных сессиях.

**Заключение.**

В заключение отметим, что рассмотренные в работе метод и апробированное средство контроля и разграничения прав доступа к сервисам олицетворения предназначены для решения одной из ключевых задач защиты информации - защиты от атак, направленных на повышение привилегий, реализуемых с целью обхода разграничительной политики доступа к защищаемым ресурсам, являющейся основой защиты информации от несанкционированного доступа в современных информационых системах. Принципиально важным является то, что рассматриваемый в работе класс атак позволяет использовать предоставляемые современыми ОС и широко используемыми на практике ОС и приложениями сервисы олицетворения, применение которых является штатной, документированной возможностью ОС.

**Литература.**

1. 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.

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 на повышение привилегий // Вестник компьютерных и информационных технологий (принята к опубликованию, предположительно в № 2 2015 годf).

4. М. Руссинович, Д. Соломон. Внутреннее устройство Microsoft Windows. – СПб.: Питер, 2005.

5. Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. – СПб: Наука и техника, 2004.

6. M. Harrison, W. Ruzzo, J. Ullman. Protection in operating systems. – Communication of ACM, 1976.

7. Щеглов К.А., Щеглов А.Ю. Контроль доступа к статичным файловым объектам // Вопросы защиты информации. - 2012. - Вып. 97. - № 2. - С. 12-20.

8. 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.

9. Щеглов К.А., Щеглов А.Ю. Практическая реализация мандатного контроля доступа к создаваемым файлам // Вестник компьютерных и информационных технологий. - 2014. - № 6. - С. 50-54.

10. 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 – 61 (принята к опубликованию) .