ВОПРОСЫ ЭФФЕКТИВНОСТИ ЗАЩИТЫ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ. ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИЯ

А. Ю. Щеглов, *д.т.н, проф. ЗАО «НПП «Информационные технологии в бизнесе»*

Задача идентификации и аутентификации пользователей является ключевой задачей защиты компьютерной информации от несанкционированного доступа (НСД), решаемой практически любой системой защиты информации (СЗИ от НСД). Это обусловливается тем, что разграничительная политика в своей основе формируется заданием прав доступа пользователей к ресурсам. Следовательно, корректность реализации разграничительной политики доступа к ресурсам, а соответственно и эффективность СЗИ от НСД в целом, во многом зависят от корректности решения задачи идентификации и аутентификации пользователей в системе. Несмотря на сформулированные в соответствующем нормативном документе в области защиты информации требования к данному механизму защиты (на наш взгляд, весьма корректно и полно), на сегодняшний день в большинстве известных нам СЗИ от НСД решение данной задачи во многом сводится лишь к усилению процедуры аутентификации при входе пользователя в систему (в некоторых СЗИ от НСД это решается на аппаратном уровне еще до загрузки системы) посредством использования внешних устройств хранения и ввода парольных данных пользователя. На наш взгляд, реализация данного подхода не только не выполняет требований соответствующих нормативных документов, но и не обеспечивает эффективного решения ключевой задачи защиты компьютерной информации. В данной работе попытаемся разобраться с тем, в чем же состоит задача идентификации и аутентификации пользователей, и рассмотрим предлагаемый нами подход к ее решению. Заметим, что данный подход на сегодняшний день реализован и апробирован в КСЗИ «Панцирь-К» для ОС Windows 2000/XP/2003 (разработка ЗАО «НПП «Информационные технологии в бизнесе»), поэтому иллюстрировать его возможности мы будем с использованием интерфейсов соответствующего механизма защиты данной СЗИ от НСД.

Формализованные (заданные нормативными документами) требования к механизму идентификации и аутентификации

Прежде всего, напомним следующее. Идентификация — это процесс распознавания элемента системы, обычно с помощью заранее определенного идентификатора или другой уникальной информации; каждый субъект или объект системы должен быть однозначно идентифицируем. Аутентификация — это проверка подлинности идентификации пользователя, процесса, устройства или другого компонента системы (обычно осуществляется перед разрешением доступа).

Формализованные требования к механизму идентификации и аутентификации пользователей задаются действующим сегодня нормативным документом «Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от НСД к информации».

Для СЗИ от НСД, используемых для защиты конфиденциальной информации (5-й класс СВТ) требования к механизму идентификации и аутентификации состоят в следующем.

Комплекс средств защиты информации (КСЗ) должен обеспечивать идентификацию пользователей при запросах на доступ, должен проверять подлинность идентификатора субъекта — осуществлять аутентификацию. КСЗ должен располагать необходимыми данными для идентификации и аутентификации и препятствовать входу в СВТ неидентифицированного пользователя или пользователя, чья подлинность при аутентификации не подтвердилась.

Проанализировав данные требования, приходим к выводу, что по своей сути они состоят из двух взаимосвязанных частей:

1. КСЗ должен располагать необходимыми данными для идентификации и аутентификации и препятствовать входу в СВТ неидентифицированного пользователя или пользователя, чья подлинность при аутентификации не подтвердилась.
2. КСЗ должен обеспечивать идентификацию пользователей при запросах на доступ, должен проверять подлинность идентификатора субъекта — осуществлять аутентификацию.

Первая часть требования, которая как раз и выполняется большинством СЗИ от НСД, предполагает реализацию механизма идентификации и аутентификации (проверку подлинности идентификатора субъекта) при входе пользователя в систему (либо при загрузке системы). Данный вопрос сегодня достаточно хорошо проработан, и нами далее рассматриваться не будет.

Вторая же часть требований предполагает реализацию механизма идентификации и аутентификации (проверку подлинности идентификатора субъекта) при запросах доступа пользователя к ресурсам. Несомненно, это совсем иная задача защиты.

Резонно, что возникает вопрос: достаточно ли для эффективного решения задачи идентификации и аутентификации реализации только одного из регламентируемых нормативными документами требования? К чему может привести и по какой причине невыполнение СЗИ от НСД данных требований в полном объеме?

Чтобы ответить на эти вопросы, прежде всего на примере семейства ОС Windows, рассмотрим всю после-

довательность идентификации и аутентификации пользователя, реализуемую современными системными средствами.

Этапы идентификации и аутентификации пользователя, реализуемые ОС Windows

Этапы идентификации и аутентификации пользователя, реализуемые в системе (на примере ОС Windows), представлены на рис.1.

Первый шаг идентификации, поддерживаемый режимом аутентификации, реализуется при входе пользователя в систему. Здесь следует выделить возможность входа в штатном и в безопасном режиме (Safe Mode). В порядке замечания отметим, что принципиальным отличием безопасного режима является то, что при запуске системы в безопасном режиме можно отключить загрузку сторонних по отношению к системе драйверов и приложений. Поэтому, если в системе используется добавочная СЗИ от НСД, можно попытаться загрузить систему в безопасном режиме без компонент СЗИ от НСД, т.е. без средства защиты. С учетом же того, что загрузить систему в безопасном режиме может любой пользователь (в Unix-системах – только Root), то СЗИ от НСД должна обеспечивать возможность входа в систему в безопасном режиме (после идентификации и аутентификации) только под учетной записью администратора.

Второй шаг состоит в запуске пользователем процессов, которые уже, в свою очередь, порождают потоки (именно потоки в общем случае и осуществляют обращение к ресурсам). Все работающие в системе процессы и потоки выполняются в контексте защиты того пользователя, от имени которого они так или иначе были запущены. Для идентификации контекста защиты процесса или потока используется объект, называемый маркером доступа (access token). В контекст защиты входит информация, описывающая привилегии, учетные записи и группы, сопоставленные с процессом и потоком. При регистрации пользователя (первый шаг, см. Рис.1) в системе создается начальный маркер, представляющий пользователя, который входит в систему, и сопоставляет его с процессом оболочки, применяемой для регистрации пользователя. Все программы, запускаемые пользователем, наследуют копию этого маркера. Механизмы защиты в Windows используют маркер, определяя набор действий, разрешенных потоку или процессу.

Рис. 1. Этапы идентификации и аутентификации пользователя

В общем случае пользователь имеет возможность запуска процесса как с собственными правами, так и под учетной записью другого пользователя. Запуск пользователем процесса под другой учетной записью возможно только после выполнения процедуры аутентификации — пользователь должен ввести идентификатор и пароль, соответствующие той учетной записи, под которой им будет запущен процесс (например, подобную возможность в ОС Windows предоставляет утилита: runas.exe, но, начиная с ОС Windows XP, эта функция уже вынесена в проводник — ее можно реализовать, нажав правой кнопкой мыши на выбранном в проводнике исполняемом файле).

В порядке замечания отметим следующее. С одной стороны, это очень полезная опция, которая может быть использована в корпоративных приложениях, когда на одном компьютере требуется обрабатывать конфиденциальные и открытые данные. При этом предполагается, что для обработки данных различных категорий создаются различные учетные записи. Данная опция предполагает, что одновременно (без перезагрузки) можно обрабатывать данные различных категорий, например, под одной учетной записью обрабатывать необходимым приложением конфиденциальные данные, под другой учетной записью запустить Internet-приложение (у вас на мониторе может быть открыто одновременно два окна). Естественно, что реализация данной возможности выставляет и дополнительные требования к СЗИ от НСД (например, при подобном запуске приложения ОС Windows между пользователями не изолируется буфер обмена, который в ОС является «принадлежностью» рабочего стола). Это уже вопросы противодействия внутренним ИТ-угрозам (угрозам хищения информации инсайдерами – санкционированными пользователями), что является вопросом самостоятельного исследования. С другой стороны, рассматриваемая опция делает во многом неразрешимыми некоторые задачи защиты информации. В порядке иллюстрации рассмотрим вопрос реализации контроля доступа пользователей к устройствам. Проблема здесь состоит в том, что в большинстве случаев доступ к устройствам осуществляется драйвером, т.е. при фильтрации запроса на доступ на уровне драйвера мы увидим учетную запись System (естественно, что решать вопросы контроля доступа к ресурсам на прикладном уровне просто бессмысленно – будет найдено множество способов обхода такого механизма), вне зависимости от того, какой собственно пользователь запрашивает доступ к ресурсу. В случае однопользовательского использования ОС это не так страшно — в качестве имени пользователя (идентификатора) можно принять учетную запись, под которой осуществлен вход в систему. В рассматриваемом же случае имеет место уже многопользовательское использование ОС – одновременно активны две (а может быть, и более) учетных записей (та, под которой осуществлен вход в систему, и та, под которой запущен процесс). Возникает проблема однозначной идентификации пользователя, запросившего доступ к устройству. Ради интереса посмотрите, каким образом решена эта проблема в существующих средствах контроля доступа к устройствам. Сразу возникает вопрос к реализации политики безопасности: как совместить на одном компьютере обработку конфиденциальных и открытых данных, если ОС позволяет ее разделить только по учетным записям.

Третий шаг состоит в порождении процессом потоков, которые собственно и обращаются к ресурсам. Система предоставляет разработчикам приложений сервисы олицетворения. Сервис олицетворения (impersonation) предоставляет возможность отдельному потоку выполняться в контексте защиты, отличном от контекста защиты процесса, его запустившего, т.е. запросить оли-

цетворить себя с правами другого пользователя, в результате — действовать от лица другого пользователя. Как следствие, именно на этом этапе и возникают вопросы корректности идентификации и аутентификации пользователя при запросе доступа к ресурсам, а задача идентификации и аутентификации пользователей при запросах на доступ сводится к контролю корректности олицетворения.

Вывод. Требование: «КСЗ должен обеспечивать идентификацию пользователей при запросах на доступ, должен проверять подлинность идентификатора субъекта — осуществлять аутентификацию» актуально и должно реализовываться современными СЗИ от НСД. При этом задача защиты при выполнении этого требования сводится к контролю корректности олицетворения при запросах доступа к ресурсам, т.к. именно использование сервиса олицетворения может привести к неконтролируемой смене идентификатора.

В порядке замечания отметим, что аналогичная ситуация имеет место и в ОС семейства Unix, где существуют понятия идентификатора и эффективного идентификатора (под которым, собственно, и осуществляется запрос доступа к ресурсам).

Реализация механизма идентификации и аутентификации при запросах доступа к ресурсам (механизма контроля олицетворения)

В общем виде решение задачи должно состоять в следующем. При запросе доступа к ресурсу должны выявляться факты произошедшего олицетворения (соответственно, субъектом доступа здесь выступает процесс, для которого анализируется наличие олицетворяющего маркера доступа) и проверяться их корректность в соответствии с заданными разрешениями (запретами), что проиллюстрировано на рис.2. Очевидно, что проверка прав субъекта доступа к ресурсу должна осуществляться уже после проверки корректности его идентификации.

Таким образом, в качестве субъекта доступа выступает процесс (в том числе это обусловливается и тем, что различные процессы (приложения) могут затребовать и различных правил разрешенных (запрещенных) олицетворений, что невозможно

Рис. 2. Укрупненный алгоритм идентификации и аутентификации при запросе доступа к ресурсу

обеспечить, если в качестве субъекта доступа принять пользователя — учетную запись).

На рис.3 представлен интерфейс, реализованного в КСЗИ «Панцирь-К» для ОС Windows 2000/XP/2003, механизма «Проверка олицетворения субъектов доступа при запросах доступа к ресурсам».

Механизм защиты, реализован-

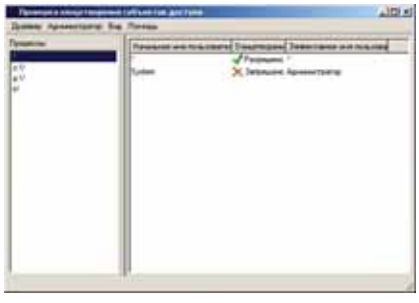

Рис. 3. Интерфейс механизма «Проверка олицетворения субъектов доступа при запросах доступа к ресурсам»

ный в КСЗИ «Панцирь-К», предполагает возможность назначить разрешительную либо запретительную политику смены первичного маркера для процессов, причем в качестве субъекта доступа здесь выступает процесс (может задаваться полнопутевым именем, именем папки – одинаковые разграничения действую для всех процессов, запускаемых из этой папки, маской), что обеспечивает максимальную гибкость применения механизма, в качестве объектов разграничений – пара: исходный меркер безопасности (исходное имя пользователя) и маркер безопасности, на который разрешается либо запрещается менять исходный маркер (эффективное имя пользователя), с которым осуществляется доступ к ресурсам. Может быть реализована разрешительная (основная) или запретительная политики смены первичного маркера.

Определим правило, задающее в общем случае условие корректности олицетворения. Для этого обозначим: I_{xy} — олицетворение потока с исходным идентификатором доступа x и олицетворяющим идентификатором доступа y. Представим I_{xy} в виде матрицы олицетворения I, отображающей варианты заимствования прав. Введем следующие обозначения. Пусть множество $C = \{C_1 \ldots Cn\}$ — линейно упорядоченное множество субъектов доступа. В качестве субъекта доступа C_k, $k = 1 \ldots n$ рассматривается как отдельный субъект, так и группа субъектов, обладающих одинаковыми правами доступа. Введем следующую иерархию субъектов доступа: чем меньше порядковый номер (идентификатор) k субъекта, тем большими полномочиями он обладает. Обозначим: X_k — исходный идентификатор субъекта доступа C_k, Y_k — олицетворяющий идентификатор субъекта доступа C_k.

При данных обозначениях модель управления олицетворением контекстов защиты формально может быть описана следующим образом: элемент (I_{ij}) матрицы олицетворения I назначается следующим образом: $I_{ij} = 1$, если $i \le j$; $I_{ij} = 0$, если $i > j$; где i – порядковый номер исходного идентификатора субъекта доступа (номер строки в матрице олицетворения), j – порядковый номер олицетворяющего идентификатора субъекта доступа (номер столбца в матрице олицетворения). Т.е. разрешенными (корректными) считаются олицетворения, не приводящие к повышению полномочий субъекта доступа.

Таким образом, основное правило корректности олицетворения состоит в следующем: смена идентификатора доступа считается корректной, если субъект доступа, описываемый исходным идентификатором доступа, обладает большими или равными полномочиями по отношению к субъекту доступа, описываемому олицетворяющим (целевым) идентификатором доступа $(I_{xy}, x \le y)$.

Рассмотрим дополнительные функциональные возможности рассматриваемого механизма защиты. С учетом того, что процессы, которые должны выступать в качестве субъектов доступа (для которых назначаются правила олицетворения) могут быть как прикладными, так и системными (системные – это процессы, запускаемые под системной учетной записью, например, System) возникает вопрос, с какой целью может быть использован предлагаемый подход при контроле олицетворений системных процессов (заметим, здесь речь не идет о приложениях, например, клиент-серверных, запускаемых с системными правами — для них все понятно, — речь идет собственно о процессах ОС).

Для того чтобы далее перейти к рассмотрению данного вопроса, рассмотрим упрощенную схему, иллюстрирующую возможные способы смены идентификатора для ОС семейства Windows.

Упрощенная схема, иллюстрирующая возможные способы смены идентификатора для ОС семейства Windows

Покажем, что рассмотренные выше второй и третий шаги имеют похожие принципы реализации в ОС, как следствие, в части построения системы защиты, здесь возможна реализация одного и того же подхода. С этой целью рассмотрим упрощенную схему запуска процесса с правами другого пользователя, реализуемую ОС Windows, которая приведена на рис. 4. Рассмотрим, как работает данная схема.

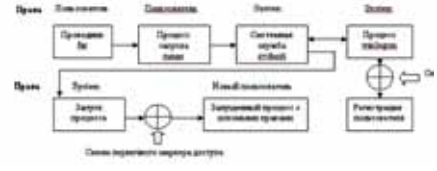

Рис.4. Упрощенная схема запуска процесса с правами другого пользователя, реализуемая ОС Windows

Для того чтобы запустить процесс (программу) с правами другого пользователя, пользователь со своей учетной записью запускает программу-проводник, позволяющую запускать соответствующую утилиту, например runas. В качестве параметров при запуске этой утилиты задается, какой процесс, с правами какого пользователя должен быть запущен, и пароль этого пользователя. Утилита запускается с правами текущего пользователя и взаимодействует с системной службой svchost, запущенной с правами System, которая, в свою очередь, взаимодействует с процессом winlogon, осуществляющим регистрацию входа нового пользователя в систему. При регистрации нового пользователя потоки, запускаемые процессом winlogon, олицетворяют себя с правами регистрируемого пользователя.

Далее системная служба svchost запускает процесс, запуск которого запросил пользователь, с правами System, после чего осуществляет смену первичного маркера доступа для запущенного процесса — смену маркера доступа System — на маркер доступа нового пользователя. В результате этого процесс начинает функционировать под учетной записью нового пользователя.

Таким образом, из рис.4 следует, что смена первичного маркера доступа осуществляется уже после запуска процесса, который изначально запускается с правами System. Видим, что если идентифицировать субъект доступа парой: идентификатор и эффективный идентификатор, то рассматриваемый процесс будет характеризоваться парой System — Новый пользователь. Следовательно, именно эта пара параметров для заданных процессов должна контролироваться при решении задачи контроля запуска процесса с правами другого пользователя. Возможность применения здесь данного механизма защиты обусловливается тем, что при смене первичного маркера процессы обращаются к объектам файловой системы (для доступа к необходимым динамическим библиотекам) и к объектам реестра ОС.

Вывод. Если запретить для процесса winlogon олицетворение пользователя System с каким-либо пользователем, то регистрация этого пользователя в системе станет невозможна.

Пример настройки механизма защиты представлен на рис.5.

При контроле олицетворения, как отмечали ранее, должна решаться аналогичная задача (при этом осуществляется олицетворение, а не смена первичного маркера, что в данном случае не так важно).

Рис. 5. Пример настройки механизма защиты

Из сказанного можем сделать вывод, что решение задач контроля олицетворения и контроля смены первичного маркера доступа базируется на одних и тех же принципах – на контроле изменения идентификатора субъекта доступа – пары: идентификатор и эффективный идентификатор пользователя. Следовательно, в обоих этих случаях может использоваться один и тот же механизм защиты, состоящий в том, что для процесса здесь можно разрешить либо запретить доступ к ресурсам применительно к конкретным вариантам изменения первичного маркера доступа.

Вывод. Если запретить для какого-либо процесса олицетворение пользователя System с каким-либо пользователем, то запуск данного процесса с правами данного пользователя в системе станет невозможен.

Использование данной возможности позволяет контролировать (управлять запуском) многопользовательский режим (в пределе может быть реализовано сведение работы современных ОС Windows к однопользовательскому (многозадачному) режиму обработки данных).

В заключение отметим, что описанные в работе (и практически апробированные) решения автор позиционирует не как некие дополнительные, а как важнейшие в составе средства защиты информации от несанкционированного доступа, т.к. невыполнение соответствующих сформулированных в нормативных документах требований к механизму идентификации и аутентификации не только не обеспечивает соответствия средства защиты классу защищенности (для которого сформулировано данное требование, а сформулировано оно уже для 5-го класса защищенности СВТ, т.е. должно быть реализовано в средствах защиты конфиденциальной информации), но и влечет за собой уязвимость средства защиты. А это уже вопросы оценки эффективности средства защиты информации.