Организация средств защиты информации для ОСРВ Багет

Н.Д. Байков[1], А.Н. Годунов[2], В.Н. Родионов[3]

[1]НИЦ «Курчатовский институт» – НИИСИ, Москва, Россия, nknikita@niisi.ras.ru;

[2]НИЦ «Курчатовский институт» – НИИСИ, Москва, Россия, nkag@niisi.ras.ru;

[3]НИЦ «Курчатовский институт» – НИИСИ, Москва, Россия, rodionov@niisi.msk.ru

Аннотация. Рассмотрены методы, применяемые при реализации средств защиты информации для операционной системы реального времени (ОСРВ) Багет. Модель управления доступом основана на управлении доступом пользователей к командам. Описаны методы идентификации и аутентификации пользователей. Также описаны пользовательский интерфейс выполнения команд, программный интерфейс регистрации обработчиков пользовательских команд и средства конфигурирования ОСРВ.

Ключевые слова: ОСРВ Багет, средства защиты информации, идентификация, аутентификация, управление доступом.

1. Введение

Формально управляющие системы можно разделить на два типа:

- автоматические,
- автоматизированные.

Принадлежность к одному из этих типов определяется наличием или отсутствием в системе оператора – лица, осуществляющего эксплуатацию управляющей системы, включая обработку хранящейся в ней информации. В случае автоматических систем такой оператор отсутствует: управляющая система функционирует самостоятельно по заранее определенному алгоритму работы, описывающему логику выполнения прикладной задачи. Это исключает возможность появления в автоматической системе внутреннего нарушителя в лице оператора, пытающегося получить несанкционированный доступ к данным или нарушить доступность системы путем эксплуатации возможностей предоставляемого ему интерфейса. Если при эксплуатации автоматической управляющей системы дополнительно возможно обеспечить замкнутость среды функционирования (неизменность выполняемого ПО) или даже полностью изолировать систему от внешнего мира, это также позволяет гарантировать отсутствие внешнего нарушителя, воздействующего на систему из-за ее пределов. Благодаря этому во многих автоматических системах встраивание средств защиты информации либо оказывается избыточным в силу условий эксплуатации системы, либо требуется в ограниченном объеме для защиты от угроз при начальной установке/обновлении ПО, а также для защиты от ошибок на уровне программной реализации алгоритма прикладной задачи.

В автоматизированной системе выполнение рабочего процесса по определению предполагает участие в нем оператора, функции которого могут быть самыми разнообразными. Например, в задачи оператора могут входить мониторинг состояния системы, сбор и обработка полученной информации, а также ручное управление рабочими процессами. Также перечисленные функции могут быть распределены между несколькими операторами, осуществляющими взаимодействие с системой параллельно. Как следствие, возникают требования по разграничению доступов пользователей к функциям системы в зависимости от той роли, которую пользователь выполняет в автоматизируемом рабочем процессе. Настоящая работа посвящена описанию механизмов управления доступом, предоставляемых разработчикам автоматизированных систем на базе операционной системы реального времени (ОСРВ) Багет.

ОСРВ Багет – это семейство операционных систем реального времени, разработанное в НИЦ «Курчатовский институт» – НИИСИ для ЭВМ серии «Багет» [1]. В отличие от операционных систем общего назначения (например, MS Windows, операционных систем семейства Linux), ОСРВ Багет преимущественно используется во встраиваемых системах в автоматическом режиме функционирования. В последние годы, в связи с деятельностью, направленной на импортозамещение иностранного программного обеспечения на объектах критически важной инфраструктуры, наблюдается интерес к функциональным средствам ОСРВ, которые были бы способны обеспечить ее функционирование в автоматизированном режиме. В том числе, к ним относится комплекс средств защиты информации (КСЗИ), с помощью которого в прикладной

системе можно было бы организовать управление доступом. Общему описанию интерфейса КСЗИ ОСРВ Багет посвящен раздел 2 настоящей статьи. Поскольку проверка разрешения на выполнение действия возможна только при условии того, что пользователь, от имени которого запрашивается выполнение действия, известен, при построении модели управления доступом возникают задачи идентификации пользователя и его аутентификации (верификации того, что пользователь действительно является тем, кем он себя идентифицирует в системе). Данные вопросы рассмотрены в разделе 3. Раздел 4 посвящен описанию политики управления доступом в ОСРВ Багет. Дополнительно в разделе 5 рассмотрены особенности реализации подсистемы протоколирования.

2. Общее описание интерфейса КСЗИ в составе ОСРВ Багет

Перед началом рассмотрения особенностей реализации КСЗИ ОРСВ Багет необходимо отметить, что применение ОСРВ Багет в прикладных системах основано на использовании технологии кросс-разработки. Ее суть сводится к использованию двух ЭВМ: инструментальной и целевой. Инструментальная ЭВМ должна использоваться разработчиками для написания прикладного программного обеспечения (ППО). Как правило, в роли инструментальной ЭВМ выступает персональный компьютер под управлением операционной системы общего назначения (например, ОС Astra Linux). В случае ОСРВ Багет ППО разрабатывается на инструментальной ЭВМ в виде исходных текстов на языке программирования Си. Далее выполняется компиляция исходных текстов ППО и их компоновка с библиотеками из установленного на инструментальную ЭВМ дистрибутива ОСРВ. В результате получаются исполняемые файлы ядра ОСРВ и прикладных процессов. Ответственным за их формирование является интегратор, которому на этапе сборки дополнительно доступна настройка конфигурационных параметров ОСРВ, определяющих режим ее функционирования (в том числе параметров, отвечающих за режим функционирования встроенных в ОСРВ средств защиты информации). Подготовленные файлы далее загружаются на целевую ЭВМ, где взаимодействие с ППО осуществляют операторы системы (пользователи). Часть из них назначаются администраторами системы (безопасности): наделяются полномочиями по управлению доступами всех остальных пользователей. Таким образом, безопасность системы в общем случае зависит от трех групп лиц:

- администраторы;
- интегратор;
- разработчики ППО.

Для каждой из перечисленных групп лиц определен свой тип интерфейса, посредством которого они оказывают влияние на конечное поведение системы. В случае разработчиков ППО это программный интерфейс, состоящий из набора разрешенных для вызова в ППО функций языка Си. Для работы интегратора предназначен конфигуратор ОСРВ, при помощи которого на инструментальной ЭВМ настраиваются конфигурационные параметры ОСРВ (см. Рис. 1). Взаимодействие с ОСРВ администраторов и других пользователей на целевой ЭВМ основано на вводе команд в режиме командного интерпретатора и/или режиме веб-сервера.

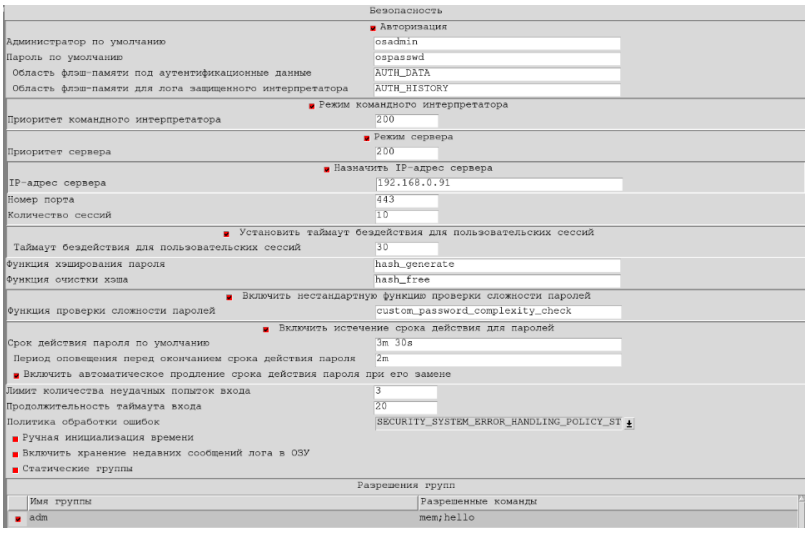

Рис. 1. Конфигурирование параметров безопасности ОСРВ на инструментальной ЭВМ

При конфигурировании ОСРВ интегратор должен указать, какие из этих двух способов ввода команд будут использоваться.

Если интегратором выбран способ ввода команд в режиме командного интерпретатора, при инициализации ОСРВ будет запущен специализированный поток выполнения, который будет заниматься обработкой поступающих в последовательный порт данных. При выборе этого режима дополнительно отключаются средства отладки ОСРВ, выполняющие чтение данных, поступающих в последовательный порт. В режиме интерпретатора команды обрабатываются последовательно. Пользователь вводит имя и пароль для входа в систему, после чего получает доступ к интерпретатору, в котором может последовательно вызывать команды. При наличии у пользователя достаточных прав будет вызван обработчик команды, и после его завершения управление будет возвращено пользователю для вызова следующей команды.

Если интегратором выбран способ отправки команд в режиме веб-сервера, при инициализации ОСРВ будет запущен поток выполнения, осуществляющий прием данных по сети. Обработка команд в этом случае осуществляется асинхронно. Веб-сервер выполняет роль диспетчера, который проверяет права доступа к командам и далее передает их потокам обработки команд. В начале пользователь посылает запрос на аутентификацию (по имени и паролю). В случае успеха в заголовках ответа будет передан идентификатор его сессии. Во всех последующих запросах на выполнение команд пользователю необходимо передавать назначенный ему идентификатор в заголовках запроса (куках).

Командный интерпретатор и веб-сервер могут быть включены одновременно.

Особенности, связанные с идентификаций и аутентификацией пользователей описаны в разделе 3. Правила управления доступом пользователей к командам описаны в разделе 4.

3. Идентификация и аутентификация пользователей

Идентификация и аутентификация пользователей с использованием КСЗИ ОСРВ Багет реализована при помощи имени и пароля. При каждом входе ОСРВ запрашивает у пользователя пароль и далее самостоятельно выполняет проверку этого пароля. В случае успеха пользователь считается аутентифицированным, т.е. подтвердившим, что он действительно является тем, кем он идентифицирует себя в системе. Ввиду отсутствия каких-либо средств делегирования, для верификации пароля пользователя требуется хранение в постоянной памяти целевой ЭВМ вспомогательных данных, связанных с учетными записями всех зарегистрированных в системе пользователей и достаточных для проведения процедуры верификации. Описанные данные называются аутентификационными данными. В простейшем случае в роли хранимых аутентификационных данных могли бы выступить сами пароли пользователей. Тогда для аутентификации пользователя было бы достаточно простого сравнения полученного от пользователя пароля с тем значением пароля, которое сохранено в памяти. Данный подход обладает недостатком: значения всех паролей пользователей в этом случае хранятся в памяти в открытом виде. Это автоматически означает, что при получении нарушителем доступа к чтению области памяти, в которой хранятся аутентификационные данные, все учетные записи мгновенно становятся скомпрометированными. По этой причине в большинстве систем, включая ОСРВ Багет, пароли напрямую не используются в качестве аутентификационных данных. Вместо этого используются производные от них хэши, получаемые в результате применения к исходному паролю пользователя специальной функции криптографического хэширования. Процедура аутентификации пользователя по паролю с использованием функции криптографического хэширования выглядит следующим образом:

- у пользователя запрашивается пароль;
- вычисляется хэш запрошенного пароля;
- вычисленное значение хэша сравнивается со значением хэша, хранящимся в постоянной памяти;
- в случае совпадения хэшей пользователь считается аутентифицированным.

Следует отметить, что формально в описанном алгоритме хэш-функция не обязана являться (и на практике, как правило, не является) инъективной. Т.е. алгоритм хэширования допускает существование коллизий – последовательностей символов, которым соответствует одинаковое значение хэша. Если одна из таких последовательностей используется в качестве пароля пользователя, все оставшиеся последовательности также могут быть использованы вместо пароля для его успешной аутентификации. С учетом этого для обеспечения требований безопасности необходимо, чтобы используемая при аутентификации хэш-функция обладала следующими свойствами:

- для одинаковых входных данных всегда должен получаться одинаковый результат вычисления;
- задача восстановления/подбора прообраза вычисленных значений должна быть избыточно трудоемкой – количество затрачиваемых на ее

решение ресурсов должно превышать потенциальную выгоду от получения несанкционированного доступа к системе в результате взлома пароля;

- задача поиска коллизий также должна быть избыточно трудоемкой.

Перечисленными свойствами обладает далеко не каждая хэш-функция. Разработка подобных функций является отдельным направлением исследований в информационной безопасности. Для использования на объектах критической инфраструктуры криптографическая хэш-функция должна пройти предварительную сертификацию на соответствие требованиям безопасности. По этой причине в ОСРВ Багет отсутствуют встроенные средства криптографического хэширования. Вместо этого предоставляется возможность подключения сертифицированных средств криптографического хэширования на этапе конфигурирования ОСРВ на инструментальной ЭВМ (см. Рис. 1). Интегратору необходимо указать имя функции, осуществляющей хэширование данных, и парную к ней функцию для освобождения ресурсов и очистки остаточного содержимого. Примером подходящей для заявленных целей функции может являться реализация алгоритма «Стрибог», описанного в межгосударственном криптографическом стандарте ГОСТ 34.11-2018 [2], разработанном на основе национального стандарта Российской Федерации ГОСТ Р 34.11.2012 [3].

Также следует отметить, что само по себе использование средств криптографического хэширования не гарантирует защиту паролей от их взлома. Уязвимость может содержаться в самом пароле. Многие пользователи используют в качестве паролей распространенные комбинации символов, что на порядки уменьшает для потенциального нарушителя область возможного поиска при переборе. В целях борьбы с этим в ОСРВ Багет предпринят ряд дополнительных мер безопасности:

- для защиты от взлома пароля путем перебора поддерживается возможность настройки таймаута при превышении лимита неудачных попыток входа в систему;

- неудачные попытки входа в систему протоколируются в журнале аудита, что позволяет администратору системы отследить попытки взлома путем перебора;

- предоставляется возможность задания в конфигурации ОСРВ пользовательской функции проверки сложности паролей, с помощью которой прикладная система может исключить использование слабых паролей, не отвечающих требованиям безопасности;

- поддерживается возможность ограничения сроков действия паролей пользователей по времени;

- хэши паролей хранятся и вычисляются с использованием «соли» (salt; см. ниже).

Понятие «соли» возникло как средство борьбы с использованием предварительно вычисленных таблиц хэшей от комбинаций символов, часто используемых пользователями различных информационных и управляющих систем в качестве паролей, т.к. с помощью таких таблиц нарушитель мог бы потенциально сократить время взлома учетных записей в случае утечки аутентификационных данных. Хотя утечка аутентификационных данных безусловно является угрозой безопасности системы, если удастся обнаружить ее до того, как нарушитель сможет вычленить из аутентификационных данных значения хэшей и восстановить их прообразы, это позволит избежать негативных последствий для системы. Будет достаточно выполнить замену паролей пользователей. Проблема заключается в том, что, если нарушителю известен используемый в системе алгоритм хэширования, он может заранее составить таблицу, состоящую из наиболее распространенных паролей, и предварительно вычислить все значения их хэшей. В том случае, если кем-то из пользователей системы используется пароль из таблицы, составленной нарушителем, время взлома такого пароля в случае утечки аутентификационных данных сократится практически до нуля, т.к. для этого будет достаточно выполнить поиск по таблице. Для предотвращения описанной угрозы необходимо сделать составление описанной таблицы нецелесообразным для нарушителя. Например, вынудить его увеличить объем таблицы до такой степени, чтобы на ее хранение и вычисление требовалось неоправданно большое количество ресурсов/времени. Для этого как минимум необходимо обязать всех пользователей системы использовать в качестве паролей достаточно сложные комбинации символов (не меньше установленной длины и задействующие как можно большее количество различных символов). Сделать это предлагается с помощью описанной выше настраиваемой функции проверки сложности паролей. Тем не менее, остается риск, связанный с тем, что нарушитель может приступить к составлению таблицы хэшей задолго до того, как в какой-либо из систем, использующих выбранный им алгоритм хэширования, возникнет утечка аутентификационных данных. Для того, чтобы сделать предварительное вычисление хэшей до утечки бессмысленным, в алгоритм аутентификации вводится соль. Под солью понимается произвольная последовательность символов, случайно формируемая в момент установки/изменения пароля пользователя.

Данная последовательность хранится в памяти в открытом виде и каждый раз добавляется к паролю (путем конкатенации) перед тем, как вычислить от него функцию криптографического хэширования (при этом хранящееся в аутентификационных данных значение хэша также должно быть получено с учетом соли). В таком случае нарушитель, который решил заранее вычислить таблицу хэшей распространенных паролей, столкнется с тем, что будет вынужден вычислять значения хэшей паролей с учетом всех возможных значений соли, поскольку на момент вычисления таблицы используемые в системе значения соли еще не известны нарушителю. Таким образом, увеличивая длину соли можно без усложнения требований к паролям пользователей добиться того, чтобы для хранения описанной таблицы у нарушителя оказалось недостаточно памяти.

4. Политика управления доступом

В случае ОСРВ объектами доступа являются команды. Команды делятся на встроенные и пользовательские. Все встроенные команды предопределены: они либо реализуют функции администрирования, либо относятся к общедоступным действиям. Например, к администрированию относится команда добавления нового пользователя `addUser`, а к общедоступным действиям относится команда выхода из системы в режиме командного интерпретатора `exit`. В отличие от встроенных, пользовательские команды предназначены для управления прикладной задачей. Список пользовательских команд варьируется от системы к системе. Добавление в систему пользовательских команд и назначение им функций-обработчиков выполняют разработчики ППО при помощи программного интерфейса. Для этого разработчику ППО необходимо реализовать алгоритм обработчика в виде функции языка Си и далее зарегистрировать обработчик, связав его с именем команды (например, при помощи предоставляемого ему для этого макроса `CUSTOM_CMD_DECL`). При этом ОСРВ обеспечивает передачу аргументов команды на вход функции-обработчику, а также предоставляет дополнительные функции для вывода результата выполнения команды: в последовательный порт в режиме командного интерпретатора и в тело ответа в режиме веб-сервера.

Политика управления доступом в ОСРВ Багет является ролевой. Роли пользователей реализуются через группы доступа. Описание каждой группы доступа состоит из уникального названия группы и списка разрешенных для ее членов команд. В список команд могут входить пользовательские команды и встроенные команды, для которых системой предусмотрена настройка доступа (например, возможность настройки предусмотрена для встроенной команды просмотра списка пользователей `users`). Для каждого пользователя в системе безопасности администратором задан атрибут, определяющий его принадлежность к группе доступа. Вызов команды доступен пользователю в следующих случаях:

- команда является встроенной и общедоступной;
- команда является пользовательской или встроенной с настраиваемым доступом, и пользователь состоит в группе, в списке разрешенных команд которой есть вызываемая команда;
- команда является встроенной командой администрирования, и пользователь является членом встроенной группы `adm` (зарезервирована для администраторов системы).

Согласно описанной политике управления доступом, решение о предоставлении аутентифицированному пользователю доступа к команде принимается только на основе проверки текущих значений атрибутов его учетной записи в системе. При этом результат вычисления не зависит от предыдущих действий этого и/или других пользователей системы (не считая изменений текущих значений атрибутов учетной записи пользователя в результате выполнения встроенных команд администратора). В частности, на уровне ОСРВ отсутствуют предопределенные ограничения на количество вызовов команды или доступность команды только в определенные интервалы времени и иные виды динамических ограничений. Модели доступа, обладающие описанными свойствами, принято относить к классу статических [4].

Дополнительной особенностью ОСРВ является наличие поддержки двух режимов работы с группами: статического и динамического. В динамическом режиме ответственным за создание групп и настройку доступных им команд являются администраторы системы на целевой ЭВМ. При выборе статического режима работы с группами действуют более жесткие ограничения на работу с группами. В этом случае администратору целевой ЭВМ доступно только включение пользователей в группы, а сам список групп и разрешенные им команды являются фиксированными и определяются интегратором на этапе конфигурирования ОСРВ на инструментальной ЭВМ. Предполагается, что режим статических групп может использоваться в системах с повышенными требованиями к разграничению информационных потоков для того, чтобы исключить потенциальное нарушения ролевой модели

прикладной системы в результате ошибочных действий администратора.

5. Подсистема протоколирования

Протоколирование событий безопасности, включая записи о действиях пользователей, является одним из общепринятых средств для выявления внутренних нарушителей. В ОСРВ Багет добавление записей в журнал событий работает в режиме кольцевого буфера. Его размер задается интегратором на этапе конфигурирования ОСРВ на инструментальной ЭВМ. Для хранения данных используется флэш-память целевой ЭВМ. При переходе к заполнению последнего из свободных секторов флэш-памяти система сигнализирует о приближающемся переполнении журнала. Команда очистки журнала доступна только администраторам, применяется к наиболее старым записям журнала и предотвращает его полное стирание. Таким образом, всегда известно, кто последним запрашивал очистку журнала (на случай, если внутренним нарушителем окажется администратор системы, который попытается скрыть следы своей деятельности).

Для целей аудита журнала предназначена встроенная команда `history`, позволяющая просматривать журнал и выполнять поиск по журналу с учетом примененных фильтров по времени и дате, номерам сессий пользователей, типам события, поисковой строке и другим дополнительным параметрам. Также ОСРВ поддерживает режим дублирования записей журнала в буфер оперативной памяти на случай, если требуется экспорт журнала аудита во внешние системы.

В журнале протоколируются все попытки вызова команд пользователями системы, результаты их выполнения, а также различные события безопасности. Например:

- начало и завершение каждой пользовательской сессии;
- неуспешные попытки аутентификации;
- предупреждения этапа инициализации системы безопасности ОСРВ (например, предупреждение о том, что не все пользовательские сессии были завершены при предыдущем запуске целевой ЭВМ, что может указывать на некорректное завершение работы).

Также программный интерфейс предусматривает наличие функции протоколирования, с помощью которой разработчики пользовательских команд могут добавлять собственные сообщения в журнал аудита.

6. Заключение

В работе рассмотрены особенности организации средств защиты информации, реализованных в составе ОСРВ Багет.

Публикация выполнена в рамках государственного задания по проведению фундаментальных исследований по теме «Создание и реализация доверенных систем искусственного интеллекта, основанных на новых математических и алгоритмических методах, моделях быстрых вычислений, реализуемых на отечественных вычислительных системах» (FNEF-2024-0001).

Security Features in RTOS Baget

N.D. Baykov, A.N. Godunov, V.N. Rodionov

Abstract. The methods used in implementing information security tools for the Baget real-time operating system (RTOS) are considered. The access control model is based on control of user access to commands. The methods for identifying and authenticating users are described. The user interface for executing commands, the software interface for registering user command handlers, and the RTOS configuration tools are also described.

Keywords: RTOS Baget, information security features, identification, authentication, access control

Литература

1. А.Н. Годунов, В.А. Солдатов. Операционные системы семейства Багет (сходства, отличия и перспективы) – «Программирование», Москва, 2014, № 5, 68–76.

2. ГОСТ 34.11-2018 «Информационная технология. Криптографическая защита информации. Функция хэширования». ФГУП «СТАНДАРТИНФОРМ». Москва. 2018.

3. ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования». ФГУП «СТАНДАРТИНФОРМ». Москва. 2013.

4. V.C. Hu, R. Kuhn, D. Yaga. Verification and Test Methods for Access Control Policies/Models. Special Publication (NIST SP). National Institute of Standards and Technology. 2017. https://doi.org/10.6028/NIST.SP.800-192.