УДК 004.056.5

И.С. Ястребов

УПРАВЛЕНИЕ ДОСТУПОМ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

***Ястребов Илья Сергеевич**, аспирант Ульяновского государственного университета, окончил факультет информационных технологий и телекоммуникаций УлГУ. Разработчик программного обеспечения в Европейском Институте Ядерных Исследований (ЦЕРН). Имеет статьи в области защиты информации в распределенных системах. [e-mail: Ilia.Yastrebov@cern.ch].*

Аннотация

В данной работе рассматриваются основные проблемы разработки и внедрения контроля доступа в распределенных системах. Приводится обзор существующих моделей контроля доступа, а также различных реализаций этих моделей для крупных производственных распределенных систем. В статье содержится анализ вышеупомянутых реализаций, а также определение области их применения.

Ключевые слова: управление доступом, защита информации, распределенные системы.

Abstract

The present article deals with main issues of development and implementation of access control in distributed systems. It gives an overview of access-control models as well as different implementation of these models for large-size production distributed systems. The article also contains an analysis of the above implementations as well as de-finition of their application domain.

Key words: access control, information security, distributed systems.

Введение

В рамках подготовки к запуску Большого адронного коллайдера (БАК) была поставлена задача внедрения контроля доступа для распределенной системы управления оборудованием. В настоящее время в Европейском Институте Ядерных Исследований разработано определенное множество активно используемых систем мониторинга и управления различными ускорителями и экспериментальными установками. Взаимодействие между большинством низкоуровневых процессов обеспечивается с использованием разработанного в ЦЕРН коммуникационного протокола CMW (Controls Middleware), в основе реализации которого использована общая архитектура брокера объектных запросов CORBA и JMS. Коммуникационный протокол CMW — это программная инфраструктура, позволяющая пользовательским приложениям контролировать удаленные устройства в распределенной системе управления (рис. 1).

Основной задачей системы контроля доступа является предотвращение неавторизованного доступа к управлению оборудованием, а также ведение журнала аудита.

Основными требованиями к системе управления доступом являются:

1. Децентрализованная система администрирования.
2. Распределенная система авторизации.

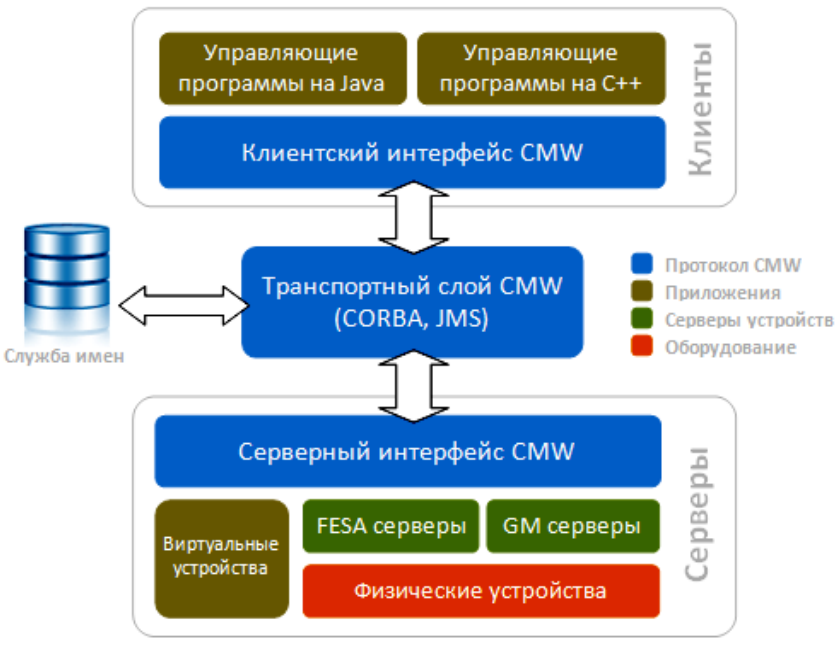

Рис. 1. Система управления устройствами

3. Динамическая авторизация.

4. Алгоритм авторизации должен учитывать контекст пользователя и оборудования.

5. Система должна быть гибкой к постоянным изменениям.

Для решения данной задачи требуется проанализировать существующие модели управления доступом, а также их реализации для распределенных систем. Целью данной работы является анализ существующих решений и определение их применимости для поставленной задачи.

1 Модели контроля доступа

Целью контроля доступа является ограничение действий или операций, которые могут выполнять в рамках системы авторизованные пользователи. Управление доступом работает совместно с другими службами безопасности в компьютерной системе, как показано на рисунке 2.

Ограничение доступа выполняется монитором обращений, который выступает посредником при каждой попытке доступа пользователя к ресурсам внутри системы. Монитор обращений отправляет запрос в базу данных, чтобы определить, имеет ли право пользователь выполнить определенную операцию с ресурсом. Монитор обращений протоколирует каждое изменение и хранит записи о соответствующей деятельности в системе [1].

Важно понимать, что контроль доступа не является полным решением проблемы обеспечения безопасности системы. Управление доступом должно быть реализовано в сочетании с системой аудита. Система аудита выполняет апостериорный анализ всех запросов и всей деятельности пользователей в системе, поэтому каждое обращение к защищенным ресурсам должно протоколироваться в журнале для последующего анализа [2].

В целом не существует политики безопасности, которая заведомо лучше, чем другие. Существуют политики, которые обеспечивают большую защиту, чем другие, однако не для всех систем требуется такая защита. Политика безопасности, подходящая для одной системы, может оказаться совершенно непригодной для другой. Например, политика очень строгого контроля доступа, необходимая для некоторых систем может оказаться неподходящей для другой системы. Выбор политики безопасности для ограничения доступа зависит от конкретных характеристик системы, которую требуется защитить [3].

В настоящее время существуют три основных модели управления доступом:

- Мандатное управление доступом;
- Избирательное управление доступом;
- Управление доступом на основе ролей.

Следует отметить, что эти модели не являются взаимоисключающими. Разные политики безопасности можно комбинировать для реализации оптимальной системы защиты. Рассмотрим подробно каждую модель управления доступом.

Рис. 2. Модель системы управления доступом

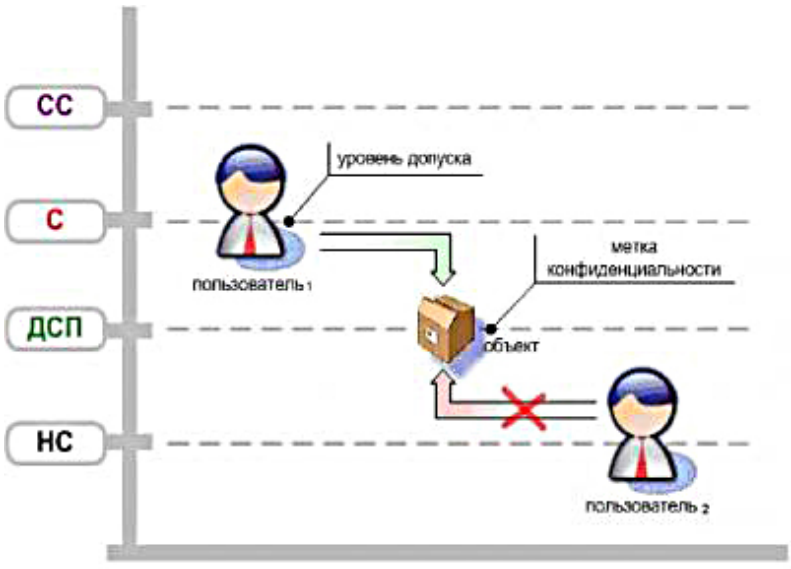

Рис. 3. Мандатное управление доступом

1.1 Мандатное управление доступом

Мандатное управление доступом — разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня [4]. Этот способ, сочетающий защиту и ограничение прав, применяется по отношению к компьютерным процессам, данным и системным устройствам и предназначен для предотвращения их нежелательного использования (рис. 3).

Самое важное достоинство этой модели заключается в том, что пользователь не может полностью управлять доступом к ресурсам, которые он создает. Такая система запрещает пользователю или процессу, обладающему определенным уровнем доверия, получать доступ к информации, процессам или устройствам более защищенного уровня [5].

1.2 Избирательное управление доступом

Избирательное управление доступом — управление доступом субъектов к объектам на основе списков управления доступом или матрицы доступа. Для каждой пары (субъект—объект) должно быть задано явное и недвусмысленное перечисление допустимых типов доступа (читать, писать и т. д.), то есть тех типов доступа, которые являются санкционированными для данного субъекта (индивида или группы индивидов) к данному ресурсу (объекту) [6] (рис. 4).

1.3 Управление доступом на основе ролей

Управление доступом на основе ролей RBAC (Role-Based Access Control) — развитие политики избирательного управления доступом, при котором права доступа субъектов системы на объекты группируются с учетом специфики их применения, образуя роли. Формирование ролей призвано определить четкие и понятные для пользователей правила разграничения доступа. Ролевое разграничение доступа позволяет реализовать гибкие, изменяющиеся динамически в процессе функционирования системы привилегии доступа. Как правило, данный подход применяется в системах защиты СУБД, а отдельные элементы реализуются в сетевых операционных системах. Ролевой подход часто используется в системах, для пользователей которых четко определен круг их должностных полномочий и обязанностей (рис. 5).

Использование RBAC для управления привилегиями пользователей широко используется во многих системах и признается лучшей моделью на сегодняшний день. Для больших систем, таких как Большой адронный коллайдер, с тысячами пользователей и миллионами разрешений, управление доступом и всеми взаимосвязями является сложной задачей, которую нереально выполнить малой группой администраторов безопасности. Привлекательной возможностью является использование RBAC для содействия децентрализованному управлению системой ограничения доступа. Эта особенность выгодно отличает RBAC от других моделей управления доступом.

Однако в системах управления оборудованием авторизация транзакций зачастую зависит не

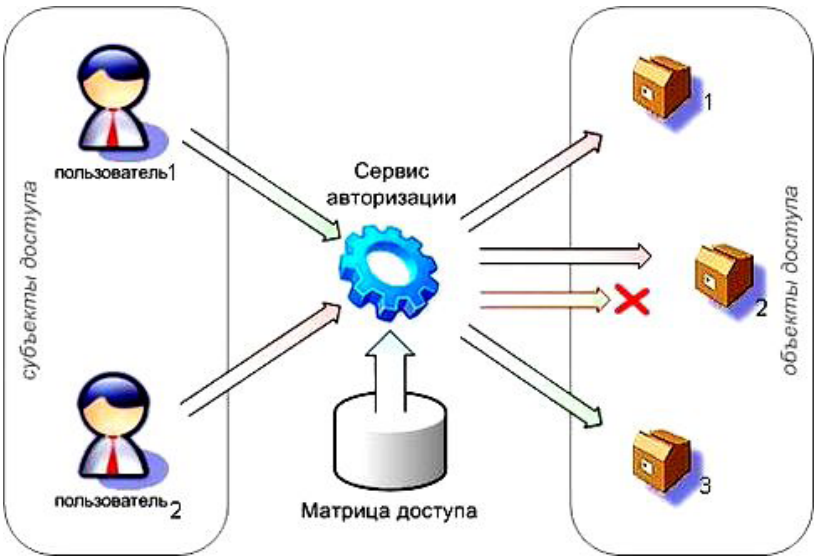

Рис. 4. Схема избирательной модели управления доступом

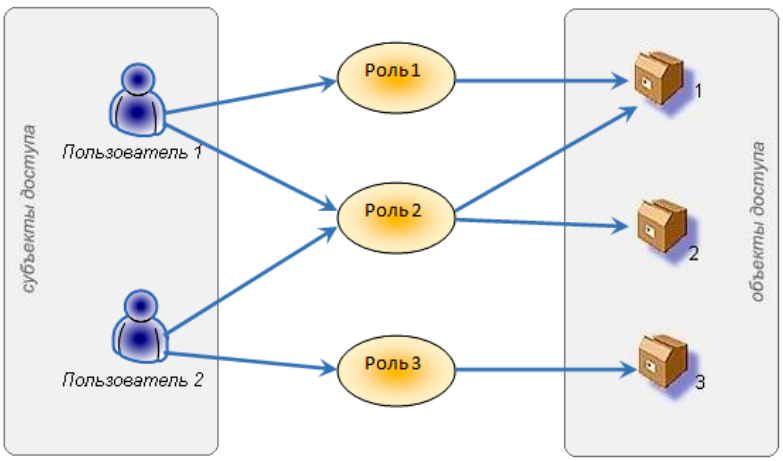

Рис. 5. Схема управления доступом на основе ролей

только от правил доступа, но также и от контекста пользователя и устройства. Классическая модель управления доступом на основе ролей не позволяет учитывать эти ограничения. Для решения этой проблемы требуется разработка гибридной модели контроля доступа на основе ролей и контекста, которая бы сочетала преимущества администрирования ролевой модели в совокупности с гибким управлением динамическими атрибутами объектов авторизации.

2 Проблемы контроля доступа в распределенных системах

Рассмотрим сложности, возникающие на практике при реализации перечисленных моделей контроля доступа в реальных распределенных системах.

Нужно отметить, что существующие решения для управления доступом сопряжены с двумя проблемами. Во-первых, доступ к защищенным ресурсам контролируется в нескольких точках: сетевой контроль (например, брандмауэры), промежуточный контроль (механизмы контроля доступа, выполняемые программным обеспечением промежуточного слоя, например, CORBA, EJB, DCE, DCOM), контроль со стороны баз данных и операционной системы. Осуществление всех этих мер контроля в масштабах предприятия является сложновыполнимой задачей, при которой необходимо согласовывать деятельность сотен приложений и вспомогательных систем.

Во-вторых, традиционные механизмы управления доступом предоставляют ограниченные возможности для обработки сложных стратегий и принятия решений по авторизации, основанных на факторах, характерных для прикладной области. Сложность политики управления доступом в некоторых областях применения, например в здравоохранении, требует более комплексного подхода и глубокой детализации, чем это предоставляется в области операционных систем, баз данных и служб безопасности в распределенных средах, таких как Java, DCOM, DCE, SESAME и CORBA.

3 Обзор существующих решений

В этом разделе мы рассмотрим основные существующие механизмы контроля доступа и обсудим, каким образом эти технологии могут быть использованы для распределенных прикладных систем, и каковы их ограничения.

3.1 JAAS

Платформа Java 2 представляет новую архитектуру безопасности, использующую политику безопасности для предоставления индивидуальных разрешений на доступ к запуску кода. Служба аутентификации и авторизации Java (JAAS) предлагает средства и стандартный интерфейс для аутентификации и присвоения привилегий пользователям [7]. В JAAS контроль доступа осуществляется с помощью посредников для системных ресурсов, таких как файлы, сокеты и т. д., в то время как объекты Java-приложений не защищены. Контроль доступа к этим объектам должен осуществляться самим приложением. Поскольку JAAS предоставляет гибкий механизм для определения специфичных для приложения ресурсов любого уровня детализации, приложение может использовать эти механизмы для определения разрешений с использованием JAAS логики принятия решений. Это позволяет использовать унифицированный интерфейс для авторизации. JAAS также предоставляет общую и расширяемую поддержку для различных аспектов авторизации. Привилегированные атрибуты в JAAS не обязательно предопределять.

3.2 DCE

В механизме контроля доступа DCE (Distributed Computing Environment), служба не контролирует доступ к приложениям и их ресурсам [8]. DCE-приложения сами обеспечивают соблюдение административного доступа и авторизацию. Для этого приложение должно само реализовывать функциональность контроля доступа, в том числе списки контроля доступа ACL (Access Control List), реализующие избирательное управление доступом.

В DCE списки контроля доступа поддерживают ограниченное число типов привилегий — сущностей пользователей, являющихся владельцами ресурсов, группой владельцев ресурсов и другими группами. Язык описания списков контроля доступа существенно ограничен и позволяет администраторам безопасности выдавать привилегии, основываясь только на сущности пользователя или атрибутах группы. Языку по-прежнему не хватает поддержки для реализации специфичных политик безопасности.

Администрирование этой системы не является масштабируемым из-за того, что любые изменения в политике безопасности должны быть отражены в базе данных ACL для каждого приложения.

3.3 GAA API

Обобщенный интерфейс аутентификации и контроля доступа GAA API (Generic Authorization and Access Control Application Programming Interface) пытается решить проблемы, связанные с недостатком стандартных механизмов авторизации для приложений [9]. Kerberos стал первой технологией обеспечения безопасности, реализующей функции GAA API. Это существенно повлияло на модель, лежащую в основе GAA API. Для сетевых приложений она предоставляет ограниченную поддержку разрешений управления доступом. Если клиент не имеет билета аутентификации на конкретный сервер сети, ему будет отказано в доступе.

Основным механизмом управления доступом в данной системе являются расширенные списки доступа EACL (частный случай избирательного управления доступом).

3.4 Microsoft DCOM

Технология Microsoft DCOM (Distributed Component Object Model) предоставляет два варианта для контроля доступа к приложениям и их ресурсам. С помощью декларативной безопасности DCOM реализует контроль доступа без какого-либо учета объекта и субъекта. Политика безопасности для приложения может быть настроена и реализована извне. Декларативные политики безопасности можно условно разделить на политики по умолчанию и специфические политики. Политика по умолчанию определяет запуск по умолчанию и настройки доступа для всех компонентов, работающих на локальной машине, которая не переопределяет эти настройки. Специфическая политика может быть использована для обеспечения безопасности конкретного компонента с помощью дополнительных настроек.

Программная безопасность DCOM предоставляет разработчикам инфраструктуру системы безопасности посредством программного интерфейса API2 с тем, чтобы и клиенты, и объекты могли защищать свои собственные приложения с помощью конкретной политики безопасности любой детализации и использовать любую информацию в качестве входных данных для принятия решений. Элементы программной безопасности могут быть использованы для переопределения политики безопасности по умолчанию, а также для настроек компонентов безопасности в системном реестре.

3.5 SESAME

SESAME (Secure European System for Applications in a Multi-vendor Environment) – европейский проект, который появился в конце 1980-х и частично финансируется Европейской Комиссией в рамках ее программы RACE [10]. Технология SESAME не является программным обеспечением промежуточного слоя. Это – архитектура для служб безопасности, реализующая управление доступом на основе ролей. Она не обеспечивает средства коммуникации, такие как ORB (Object Request Broker), RPC (Remote Procedure Call) слой в DCE или DCOM, поэтому не может контролировать события до или после вызова. Именно поэтому контроль доступа и другие функции безопасности должны явным образом вызываться из приложения. Это не дает системе возможности использовать посредников для контроля доступа. Вместо этого бизнес-логика предоставляется как служба авторизации. В этом ее главное отличие от DCE, где приложение должно само реализовывать хранение и управление списками доступа.

Еще одним недостатком контроля доступа в системе SESAME является отсутствие возможностей или, по крайней мере, простого способа применения одной политики безопасности для нескольких прикладных программ. Это означает, что политики безопасности должны быть настроены для различных приложений в индивидуальном порядке.

3.6 CORBA

Система безопасности в технологии CORBA (Common Object Request Broker Architecture), как и в DCOM, работает с использованием перехватчиков [11]. Сбор информации для авторизации и само принятие решения о доступе всегда срабатывает до вызова удаленного метода. Решения алгоритма авторизации основываются на значениях атрибутов субъекта, правилах доступа к объекту, а также на текущей политике управления доступом к объекту. Решения авторизации могут быть специфичными для каждого объекта, если объект находится в отдельном домене, или для большой группы объектов, связанных с политикой одного домена. Это означает, что модель масштабируется очень хорошо, не теряя детализации. В отличие от DCOM, в CORBA объекты, находящиеся на разных компьютерах, могут быть связаны в один домен.

К. Безносов показал, что механизм контроля доступа CORBA способен реализовать мандатное управление доступом [2]. Также было доказано, что систему безопасности CORBA можно настроить для моделирования управления доступом на основе ролей. Это означает, что избирательное управление доступом также можно реализовать посредством технологии CORBA.

4 Заключение

Современные технологии предоставляют ряд средств для контроля доступа к распределенным ресурсам. Существуют две группы технологий, используемых для обеспечения безопасности в распределенных системах. Первая группа предоставляет частичную аутентификацию, защиту коммуникации и независимость механизма контроля доступа от применяемой технологии связи. Эта группа включает в себя системы Kerberos, SESAME и GAA API. Это позволяет использовать любой коммуникационный протокол, но разработчики приложений вынуждены прилагать значительные усилия для интеграции технологий безопасности с основными механизмами коммуникации.

Вторая группа является программным обеспечением промежуточного слоя. Она включает в себя такие технологии, как CORBA, DCE, JAAS и DCOM, которые предоставляют коммуникационную инфраструктуру наряду с обеспечением безопасности системы. Эти системы намного проще интегрировать с пользовательскими приложениями. Кроме того, некоторые из них

позволяют управлять доступом без изменения прикладной программы, поскольку авторизация происходит до того, как удаленный вызов направляется на сервер приложений.

В больших распределенных системах довольно часто одновременно используются несколько коммуникационных протоколов, работающих на разных операционных системах. Примером такой системы может служить коммуникационная инфраструктура для управления физическим экспериментом. Это делает невозможным использование второй группы технологий контроля доступа. Другими характерными особенностями являются необходимость выполнения децентрализованной авторизации и использование контекстной информации в алгоритме авторизации. Эти требования обусловлены спецификой системы управления физическим экспериментом. В связи с этим приведенные решения не могут применяться для обеспечения безопасности системы управления физическим экспериментом. Целесообразным направлением исследований является разработка гибридной системы контроля доступа на основе ролей и контекста.

СПИСОК ЛИТЕРАТУРЫ

1. Sandhu R., Samarati P. Access Control: Principles and Practice // IEEE Communications — 1994. — Vol. 32, No. 9.

2. Beznosov K. Engineering Access Control for Distributed Enterprise Applications // PhD thesis, Florida International University. — Miami, FL, 2000.

3. Lampson B., Abadi M., Burrows M., Wobber E. Authentication in Distributed Systems: Theory and Practics // Proceedings of ACM Symposium on Operating Systems Principles. — USA, California, 1991. — pp. 165—182.

4. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. — М. : Стандартинформ, 2008.

5. Mandatory Access Control. Wikipedia. — Режим доступа: http://en.wikipedia.org/wiki/Mandatory_access_control.

6. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации : РД : утв. решением Гостехкомиссии при Президенте РФ 30.03.92. — М. : ГТК, 1992.

7. Lai C., Gong L., Koved L., Nadalin A., Schemers R. User Authentication And Authorization In The Java Platform // Proceedings of Annual Computer Security Applications Conference. — Phoenix, Arizona, USA, 1999. — pp. 285—290.

8. Kong M. M. DCE: An Environment for Secure Client/Server Computing // Hewlett-Packard Journal. — 1995. — Vol. 46, No. 6. — pp. 6—15.

9. Generic Authorization and Access Control API. — Режим доступа: http://gost.isi.edu/info/gaaapi/.

10. Secure European System for Applications in a Multi-vendor Environment. — Режим доступа: http://www.cosic.esat.kuleuven.be/sesame/.

11. Catalog of OMG CORBA services Specifications. — Режим доступа: http://www.omg.org/technology/documents/corbaservices_spec_catalog.htm.