Сергей Геннадьевич ЛЕПИХИН —
зам. директора Института
дистанционного образования

Иннокентий Николаевич ФИЛИПЕНКО —
программист отдела программного обеспечения
Института дистанционного образования

Тюменский государственный университет

МНОГОУРОВНЕВАЯ СХЕМА УПРАВЛЕНИЯ ДОСТУПОМ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ УПРАВЛЕНИЯ УЧЕБНОГО ПРОЦЕССА ИНСТИТУТА ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

УДК 004.056.5

АННОТАЦИЯ. В статье рассматривается механизм предложенной защиты схемы управления доступом информационной системы Института дистанционного образования, по которой приложение предоставляет набор сервисов, использующих единое хранилище данных о пользователях.

The essay is intended for describing security aspect of the Institut of distance education information system's access scheme. According to the scheme, the application gives an opportunity to use a set of services with a shared users database.

Введение. К одной из самых последних тенденций в области разработки информационных систем можно отнести предоставление различных сервисов и ресурсов пользователям. Под сервисом понимается единица прикладной логики, включающая методы реализации определенных операций, функций и преобразований. Таким образом, информационную систему можно рассматривать как совокупность работающих, распределенных и повторно используемых сервисов, решающих поставленные бизнес-задачи.

Зачастую множество сервисов информационной системы требует информации о множестве пользователей этой системы, при этом одновременно необходима и информация о возможности использования того или иного сервиса конкретным пользователем, а также степени использования связанного с этим сервисом информационного ресурса. Следовательно, необходимо единое хранилище данных информационной системы о пользователях и их правах доступа ко всем сервисам и ресурсам этой системы.

Механизмы управления доступом являются основой защиты ресурсов, обеспечивая решение задачи разграничения доступа субъектов к защищаемым информационным и техническим ресурсам — объектам. Под субъектом (в простейшем случае) понимается пользователь системы. Объектами же повышенного интереса информационной системы учебного заведения могут стать:

— образовательный контент (методические, контрольно-измерительные материалы);

— персональная информация юридических и физических лиц, сотрудничающих с учебным заведением;

— персональные данные процесса обучения;

— аппаратно-системное оборудование;

— возможности сетевого доступа и т.д.

Данная статья ориентирована на рассмотрение вопросов защиты схемы, по которой приложение предоставляет лишь набор сервисов, а не прямой доступ к хранилищам данных, и не касается аспектов безопасности физического уровня, уровня безопасности сетевых протоколов, а также доменной безопасности сервера и политики организации сети.

Основная часть. Информационная система поддержки управления учебного процесса Института дистанционного образования Тюменского государственного университета реализует трехуровневую схему управления доступом. В схеме заложено использование единого хранилища данных о пользователях и их правах доступа к ресурсам системы с целью дальнейшего использования в предоставлении набора конечных сервисов.

Первый уровень схемы служит целям хранения реквизитов пользователей (пароль, логин, роли) и реализован LDAP-каталогом. Такое решение обосновано тем, что одни и те же реквизиты пользователей зачастую необходимы сразу нескольким сервисам (например, для организации почтового сервера). LDAP-каталог легко интегрируется с различными сервисами и хорошо оптимизирован на большое количество запросов чтения данных.

Далее следует уровень приложения, который на основании введенных реквизитов проводит идентификацию пользователей системы и контролирует их доступ к ресурсам. Web-приложение написано на языке C# по технологии ASP.NET 2.0 — наиболее современной технологии разработки крупных web-приложений.

Третий уровень представлен СУБД, на котором осуществляется управление массивами данных в зависимости от роли текущего пользователя. В качестве СУБД выбрана постреляционная freeware PostgreSQL 8, близкая по функционалу к промышленным СУБД.

На рис. 1 приведена концептуальная трехуровневой схемы управления доступом, которая иллюстрирует взаимодействие уровней.

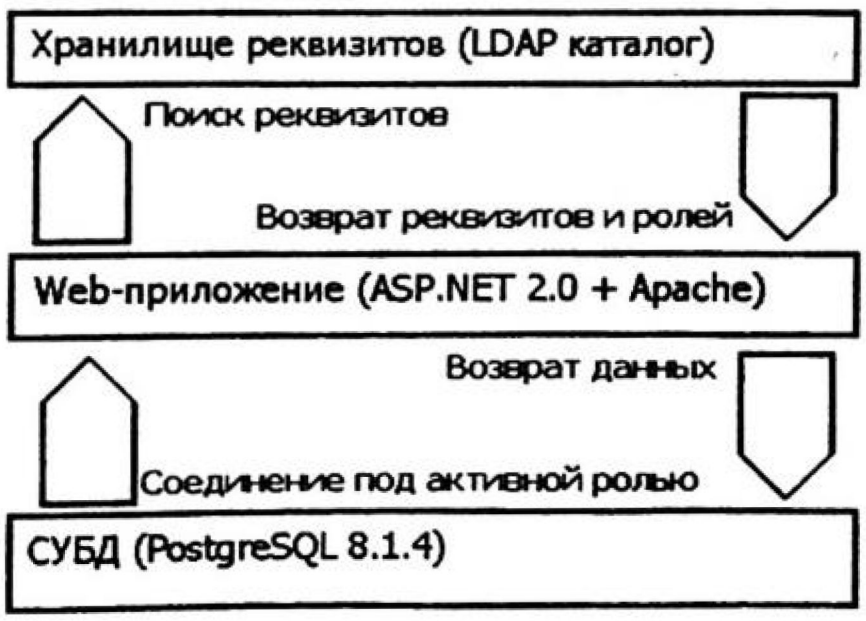

Рис. 1. Трехуровневая схема управления доступом

Основными понятиями при обеспечении безопасности информационной системы являются процессы аутентификации и авторизации.

Под аутентификацией понимается процесс выяснения и проверки личности пользователя с помощью получения от него пары логин/пароль (называемых реквизитами[1]) и сравнения их с каким-либо источником. За управление хранилищем реквизитов пользователей отвечает Membership API. Membership API является статическим классом, методы которого делегируют вызовы наследнику абстрактного класса MembershipProvider. Переопределяя абстрактные методы MembeshipProvider, класс LDAPMembershipProvider реализует всю логику хранения реквизитов в LDAP-каталоге как частном случае хранилища. Спецификация и настройка конкретного провайдера осуществляется с помощью конфигурационного файла. Замена хранилища реквизитов пользователей не затрагивает основную логику работы приложения, для этого достаточно лишь специфицировать другой провайдер.

Описанный выше механизм относится к гибкой модели провайдеров ASP.NET [2], позволяющей легко выполнить замену хранилища реквизитов, ролей, сменить формат карты сайта, переопределить механизм хранения состояния сеанса и т.д.

Механизм Membership API тесно связан с Roles API, предназначенным для работы с ролями пользователей (об этом речь пойдет ниже). Функционал классов LDAPMembeshipProvider и LDAPRolesProvider частично дублирует друг друга. Для устранения тесной связности и дублирования кода методы работы с записями реквизитов и списками ролей разнесены на два независимых класса — LDAPUserTree и LDAPRolesTree.

В информационной системе применяется аутентификация на основе форм [3]. При входе в систему пользователь попадает на страницу ввода реквизитов пользователя. После ввода реквизитов запускается проверка подлинности реквизитов методом Membership API ValidateUser, которому передаются введенные логин и пароль в неизменном виде (рис. 2). Данный метод перегружается в классе LDAPMembershipProvider и условно состоит из двух этапов.

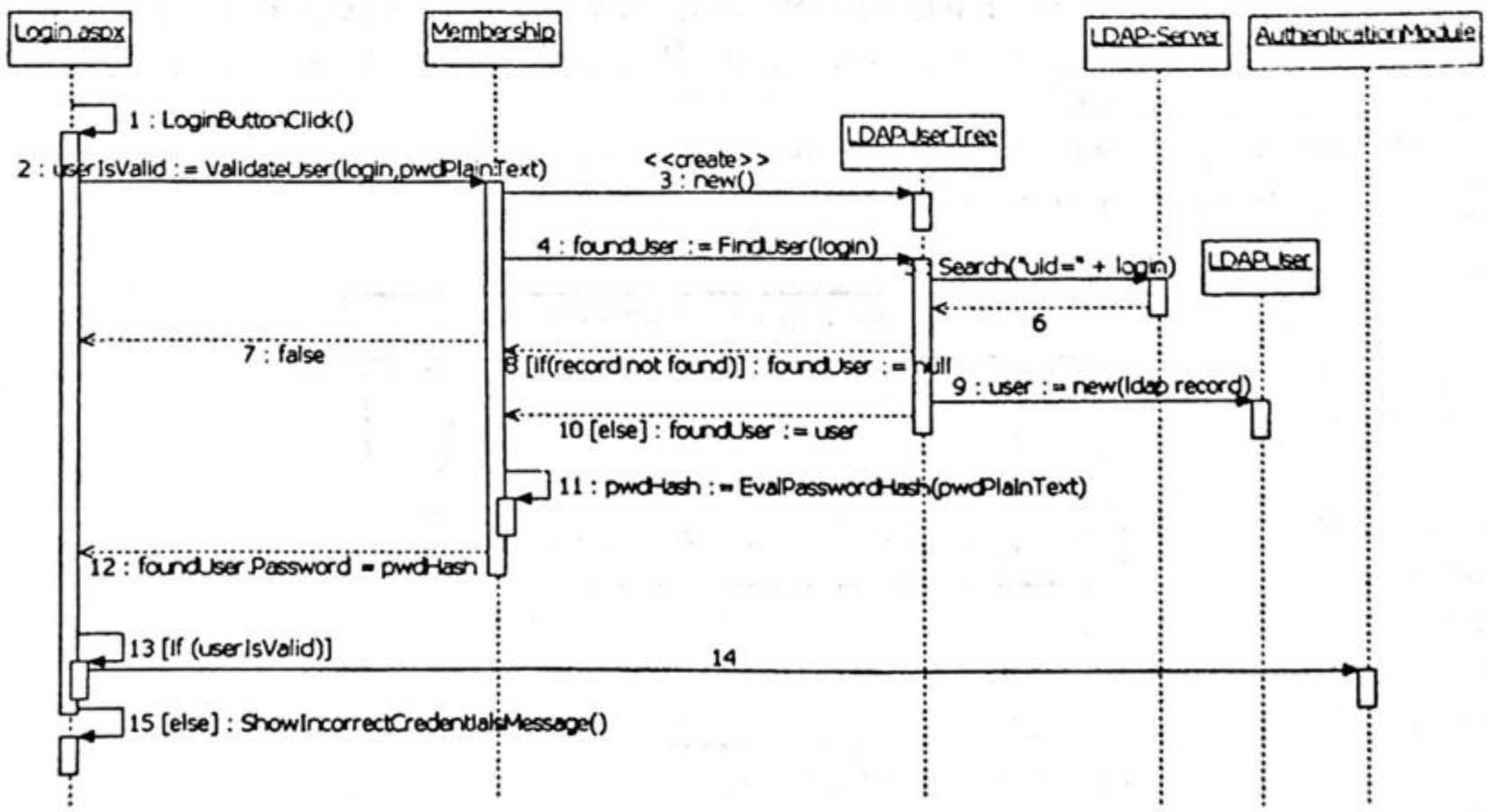

Рис. 2. Диаграмма последовательности процесса аутентификации в системе

На первом этапе производится поиск данных пользователя в LDAP-каталоге. Структура данной записи приведена в табл. 1. Выполнение запроса и обработка результатов поиска производятся с помощью метода FindUser(string login): LDAPUser класса LDAPUserTree. Данный метод возвращает указатель на экземпляр класса LDAPUser, если запись с указанным именем пользователя была обнаружена. В противном случае будет возвращена ссылка на null.

Таблица 1

Структура учетной записи пользователя в LDAP-каталоге

Атрибут	Описание
cn	Необходим для указания уникального пути к пользовательской записи. Генерируется по фамилии, имени и отчеству.
uin	Уникальный идентификатор пользователя в системе. Может быть изменен по желанию пользователя. По умолчанию равен cn.
userPassword	Хэш пароля для входа в систему. Имеет формат: {алгоритм хэширования}хэш.
roles	Список ролей для данного пользователя. Разделенный символами «|».
mail	e-mail адрес пользователя. По умолчанию генерируется как cn + @ + e-mail server
objectClass	InetOrgPerson (RFC2798) [4]

На втором этапе производится сравнение хэшей введенного и хранящегося в LDAP-каталоге паролей (свойство LDAPUser.PasswordHash). Для хранения пароля используется необратимое преобразование (Хэш-функция). Для реализации которого наиболее часто используется алгоритм хеширования MD5. Если хеши равны, реквизиты считаются верными и браузер пользователя перебрасывается на страницу по умолчанию (Default.aspx) — идентификация пользователя завершена. В противном случае пользователю предлагается ввести реквизиты заново.

Авторизация — определение и предоставление прав доступа к конкретным ресурсам или объектам [5]. Авторизация в системе заключается в разграничении прав доступа на ресурсы приложения, в том числе на проверку прав доступа к физическим таблицам базы данных.

За основу разграничения прав доступа к ресурсам системы была принята концепция ролей, каждая из которых представляет собой набор сервисов, как полезных бизнес-функций некоторой бизнес-единицы [6]. В схему авторизации системы введено понятие активной роли — роль согласно выбранной бизнес-единице. Активная роль определяет не только доступ к информации, но и интерфейс пользователя — у каждой роли свое меню.

За управление ролями отвечает Roles API. Все роли приложения хранятся в отдельном поддереве LDAP-каталога (табл. 2). Провайдером для данного API служит класс LDAPRolesProvider.

Таблица 2

Структура записи роли в каталоге LDAP

Атрибут	Описание
cn	Имя роли. Должно быть уникальным, так как используется в качестве dn
objectClass	organizationalRole (RFC 2256) [7]

Из табл. 1 и 2 видно, что список ролей каждого пользователя хранится в записи пользовательских атрибутов. Такая оптимизация обоснована жесткими требованиями по числу одновременных поисковых запросов к хранилищу реквизитов. Количество одновременных обращений к LDAP-серверу пользователей образовательной информационной системы может быть очень велико, и в случае альтернативной реализации списка ролей пользователя хранением атрибутов uin каждого пользователя в записи конкретной роли повышается условие целостности данных, что многократно увеличивает нагрузку на LDAP-сервер.

Управление доступом к страницам приложения осуществляется средствами ASP.NET 2.0. Соответствие страниц и ролей задается в конфигурационных файлах приложения. Для облегчения конфигурации страниц все приложение разбито на каталоги — отдельный для каждой роли. Например, все страницы, доступные только роли «методист», находятся в папке methodist.

Интерфейс каждой роли задается с помощью карты сайта, где каждой странице приложения соответствует перечень ролей и название данной страницы. При формировании меню активной роли из карты отбирается только тот функционал, для которого в списке ролей содержится указанная роль.

Доступ к базе данных осуществляется посредством создания синхронного со списком ролей приложения списка ролей базы данных и установления соединения под конкретной ролью. В предложенной схеме база данных физически отделена от интернет- или интранет-сетей и доступна только приложению. Каждая роль базы данных защищена паролем, который генерируется уровнем приложения. Через определенные промежутки времени приложение для всех ролей базы данных регенерирует пароли. Таким образом, защита уровня базы данных строится на защите учетной записи приложения, через которую оно соединяется с базой данных и управляет учетными записями для каждой из ролей.

Управление приведенной трехуровневой схемой безопасности возможно через рабочее место администратора. В функции данной подсистемы входит:

1) создание списка ролей,

2) редактирование конфигурационных файлов приложения для настройки доступа ролей к страницам приложения,

3) редактирование прав доступа ролей к таблицам базы данных (для этого администратор базы данных должен выдать учетной записи приложения соответствующие права),

4) создание учетных записей пользователей и назначение им списка доступных ролей.

Результаты построенной системы защиты информационной системы удобно раскрыть по классификации STRIDE [8]. Название классификации образуется по первым буквам английских названий категорий. Принципы классификации STRIDE придуманы, обоснованы и активно пропагандируются сотрудниками корпорации Microsoft Лоэном Конелдером, Преритом Гаргом, Джейсоном Гармсом. Далее приведены решения защиты в ответ на действия злоумышленника по STRIDE.

Подмена сетевых объектов (*Spoofing identity*). Данная атака нацелена на похищение реквизитов пользователей и основана на ненадежных методах аутентификации. В информационной системе применяется аутентификация на основе форм. Таким образом, процесс аутентификации полностью контролируется на сервере. После проверки пользователя на время открытия сеанса клиентскому компьютеру отправляется зашифрованный coockie-файл с меткой аутентификации.

Модификация данных (*Tampering with data*). Атака предполагает злонамеренную порчу данных, что тесно связано с повышением привилегий. Защищается встроенными механизмами ASP.NET и СУБД.

Отказ от авторства (*Repudiation*). В данном случае пользователь отрицает изменение данных. Для того, чтобы была возможность доказать «авторство» изменений данных, каждое действие пользователя, приводящее к модификации данных — создание отчетов, редактирование содержимого базы данных, правка контрольно-измерительных материалов — записывается в журнал (log-файл). Каждая запись в журнале состоит из нескольких атрибутов: дата и время действия; имя пользователя; уникальный идентификатор действия; уникальный идентификатор изменяемой сущности; данные до и после модификации. Для того, чтобы размер журнала не стал слишком большим, изменяемые данные записываются в минимальном объеме, для чего используются имеющиеся в системе выражения для каждой сущности, уникально идентифицирующие каждую сущность.

Разглашение информации (*Information disclosure*). Атака, направленная на разглашение запрещенных для данного пользователя данных. Механизм авторизации не позволит пользователю читать, изменять запрещенную для него информацию; авторизация доступа к базе данных не позволит выполнять SQL-инъекции на получение запрещенной информации, а управление отображаемым набором функций активной роли и доступом к страницам не позволит открыть запрещенную страницу.

Отказ в обслуживании (*Denial of service*). DoS-атака, направленная на лишение доступа к информационной системе правомочных пользователей; по большей части является задачей грамотной настройки сервера приложений (Apache). Приложение рассчитано на одновременную обработку большого количества запросов к сервисам системы, поэтому для успешного проведения данной атаки требуются большие финансовые и организационные затраты.

Повышение привилегий (*Elevation of privilege*). Атака заключается в получении непривилегированным пользователем привилегированного доступа к системе. В контексте системы повышение привилегий означает несанкционированное изменение списка ролей пользователями, что возможно только при похищении реквизитов административной учетной записи. Для этого администратор сервера приложения должен периодически обновлять реквизиты учетной данной записи. Кроме этого, в системе имеется возможность автоматически отслеживать срок жизни административной учетной записи, по истечении которого она признается недействительной и принудительно требуется ее обновление.

Результаты

1. Предложена схема управления доступом информационной системы.
2. Расширена схема авторизации до уровня базы данных;
3. Реализован механизм управления доступом на основе единого хранилища данных о пользователях, который обеспечивает работу следующих сервисов информационной системы Института дистанционного образования:

— организация обмена электронной почтой между пользователями информационной системы, реквизиты которых присутствуют в LDAP-каталоге;

— организация доставки голосовых сообщений;

— разграничение доступа к видеолекционному материалу пользователей, согласно текущему состоянию их учебного статуса;

— групповой обмен сообщениями в реальном времени между пользователями конкретных ролей информационной системы;

— групповой обмен сообщениями Offline между пользователями конкретных ролей информационной системы.

4. Рассмотрена безопасность работы реализованной схемы, согласно классификации STRIDE, одобренной корпорацией Microsoft.

СПИСОК ЛИТЕРАТУРЫ

1. Анализ требований и создание архитектуры решений на основе Microsoft.NET. Учебный курс MCSD/Пер. с англ. М.: ИТД Русская Редакция, 2004. 416 стр.
2. Schackow, S. Professional ASP.NET 2.0 Security, Membership, and Role Management. Wiley Publishing, Inc. 2006
3. Эспозито Д. Microsoft ASP.NET 2.0 Углубленное изучение. Серия Мастер класс. М.: ИТД Русская Редакция, 2002.
4. RFC 2798: Definition of the inetOrgPerson
5. Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. Санкт-Петербург: Наука и Техника,. 2004
6. RFC 2256. A Summary of the X.500(96) User Schema for use with LDAPv3
7. Уилсон С.Ф., Мэйплс Б., Лэндгрейв Т. Принципы проектирования и разработки программного обеспечения. Учебный курс MCSD/Пер. с англ. 2-е изд., испр. М: ИТД Русская Редакция, 2002. 736 стр.
8. Ховард М., Лебланк Д. Защищенный код: Пер. с англ, 2-е изд., испр. М.: ИТД Русская Редакция, 2004. 704 стр.
9. About Lightweight Directory Access Protocol. http://msdn2.microsoft.com/en-us/library/aa366075.aspx.
10. Using Lightweight Directory Access Protocol. http://msdn2.microsoft.com/en-us/library/aa367033.aspx.

Ирина Гелиевна ЗАХАРОВА —
директор Института математики
и компьютерных наук,
доктор педагогических наук, профессор

Яков Викторович ЛАНГ —
аспирант кафедры программного обеспечения

Елена Сергеевна ОХОТНИКОВА —
аспирант кафедры программного обеспечения
Тюменский государственный университет

УДК 519.7

МАТЕМАТИЧЕСКИЕ МОДЕЛИ ВАРИАТИВНЫХ ЭЛЕКТРОННЫХ УЧЕБНЫХ КУРСОВ

АННОТАЦИЯ. В работе предложены математические модели, позволяющие описать структуру и алгоритмы создания контента автоматизированной системы управления обучением, обеспечивающей реализацию индивидуальных траекторий изучения учебного курса.

The article viewed mathematical model to describe the structure and algorithms for automated content creation education management system that ensures the realization of individual course trajectories.

В условиях современного вуза у студентов появляются широкие возможности для использования современных компьютеров и средств коммуникации для поиска и получения информации, развития способностей, умения оперативно принимать решения в сложных ситуациях и т.д. Соответственно, предполагаются и новые качества систем управления обучением (Learning Management Systems — LMS). Они должны выступать как средство организации управляе-