УДК 004.75

ПРИМЕНЕНИЕ МОДЕЛИ БЕЗОПАСНОСТИ ASP.NET В ПРИКЛАДНОЙ СИСТЕМЕ

К. т. н. С. И. Гришин[1], И. Н. Лисицына[2]

[1]Одесский национальный политехнический университет,
[2]Одесский национальный университет им. И. И. Мечникова
Украина, г. Одесса
grishin_si@ukr.net

Рассмотрена организация безопасности в системе автоматизации работы туристического агентства. Если пользователю разрешается вход в систему, то для дальнейшей работы используется его личный идентификатор, который хранится в виде переменной сессии. Используя эту переменную, сценарии промежуточного уровня могут идентифицировать пользователя, от чьего имени был послан запрос.

Ключевые слова: аутентификация, страница регистрации, файл конфигурации.

Обеспечение информационной безопасности потребовалось при автоматизации деятельности туристического агентства с разветвленной сетью филиалов. Агентство занимается организацией как группового, так и индивидуального туризма. Более трудоемкой задачей является как раз организация индивидуального туризма за счет того, что схема отдыха формируется непосредственно менеджерами агентств, а не централизованно. Также было уделено внимание организации продажи билетов самим агентством. Для реализации поставленной задачи были выбраны СУБД Microsoft SQL Server для хранения и манипулирования данными, язык .NET C# и технология программирования Web-приложений ASP.NET.

Поскольку протокол HTTP, который используется для связи между браузером и Web-сервером, представляет собой протокол без сохранения состояния, при каждом запросе новой страницы браузер открывает новое сетевое соединение с Web-сервером. Web-сервер должен обладать механизмом, позволяющим идентифицировать пользователя из запроса.

Предполагается, что пользователями системы являются исключительно менеджеры туристического агентства. Поэтому для первоначальной аутентификации требуется ввести регистрационное имя и пароль пользователя. Если пользователь является менеджером (существует соответствующая строка в таблице базы данных Microsoft SQL Server), то ему разрешается вход в систему, а для дальнейшей работы используется его личный идентификатор, который хранится в виде переменной сессии. Соответствующая переменная уничтожается, когда пользователь выходит из системы или если он не обращается с запросами в течение заранее заданного промежутка времени.

Важным аспектом является то, что пользователь, который не зарегистрировался в системе, не должен получать доступ к сайту, т. е. к промежуточным страницам. Решений может быть несколько. Например, проверка переменной сессии: если она есть, то загружать страницу, если нет, то перенаправлять пользователя на страницу регистрации. Второй возможный способ – это использование модели безопасности ASP.NET.

Процесс запроса ASP.NET страницы выглядит следующим образом:

1. IIS пытается аутентифицировать пользователя. В большинстве случаев IIS разрешает запросы, поступающие от всех анонимных пользователей, и автоматически регистрирует их под учетной записью IUSR_[ServerName].

2. Если аутентификация пользователя успешно завершена, IIS передает запрос ASP.NET, сопровождая его дополнительной информацией об аутентифицированном пользователе. Затем ASP.NET может применить свои собственные службы безопасности в зависимости от настроек файла web.config и запрашиваемой страницы.

3. Если ASP.NET аутентифицирует пользователя, то ему разрешается запрос страницы aspx или Web-службы asmx. В коде страниц могут быть использованы другие параметры безопасности.

4. Когда код ASP.NET запрашивает внешние ресурсы (к примеру, пытается открыть файл или подсоединиться к БД), операционная система выполняет очередную проверку. Обычно код приложения ASP.NET исполняется с помощью специальной системной учетной записи, предоставляющей ему широкие возможности. Однако если разрешить использование персонификации, эти системные операции будут производиться посредством учетной записи аутентифицированного пользователя (или другой указанной записи).

Существует две основные стратегии обеспечения безопасности:

— разрешение анонимности, но применение модели аутентификации ASP.NET, основывающееся на формах, для защиты определенных частей сайта.

— запрещение анонимности и использование аутентификации IIS, чтобы принудить каждого пользователя к регистрации с использованием базовой, краткой или интегрированной аутентификации Windows.

В рассматриваемом Web-приложении использовался метод аутентификации через формы. Для этого прежде всего необходимо выполнить следующее.

1. Установить режим аутентификации в конфигурационном файле web.config

<authentication mode="Forms">

2. Запретить анонимным пользователям доступ к страницам сайта

<authorization>

<deny users="?" />

</authorization>

Изменения также вносятся в файл web.config

3. Создать страницу регистрации. В файле web.config необходимо указать на эту страницу (Password.aspx) для ввода пароля:

<forms loginUrl="Password.aspx">

</forms>

При попытке просмотра защищенной страницы ASP.NET проверяет, есть ли аутентификационный cookie (файл, создаваемый на жестком диске клиентского компьютера или в памяти браузера) в запросе. Если cookie нет, то запрос перенаправляется на страницу для регистрации, если есть — ASP.NET дешифрует cookie и извлекает из него регистрационную информацию.

В коде страницы регистрации добавлено внесение информации о регистрации при помощи специальной функции **SetAuthCookie().** Для получения данных о менеджере из таблицы Microsoft SQL Server использованы объекты классов Connection, Command, DataReader технологии доступа к данным ADO.NET.

Если в базе данных пользователь с введенным паролем и логином был найден, то добавляется элемент cookie и запускается начальная страница приложения. В функции **SetAuthCookie** присутствуют 2 параметра: имя пользователя и второй параметр, указывающий на то, является ли элемент cookie постоянным (true) или обычным (false). По умолчанию время жизни сеансового cookie, генерируемого ASP.NET, равно 30 минутам. Когда сеансовый cookie возвращается в следующих после регистрации запросах, он автоматически обновляется, если время жизни истекло больше чем наполовину.

Тестирование системы подтвердило работоспособность модуля безопасности.

S. I. Grishin, I. N. Lisitsina

Using ASP.NET security model in applied system.

The authors consider organization of security in the travel agency automation system. If the user is allowed to log in, then for further work his personal identifier is used. The identifier is stored as a session variable. Using this variable, middleware scenarios can identify the user on whose behalf the request is sent.

Keywords: *authentication, registration page, configuration file.*