электронное научно-техническое издание
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС 77 - 30569. Государственная регистрация №0421100025. ISSN 1994-0408

Методы аутентификации

77-30569/370630
04, апрель 2012
Большакова Д. О., Фенске А. В.
УДК 004.056.52

МГТУ им. Н.Э. Баумана
boldol@mail.ru
fenix.iu8@gmail.com

Введение

Методы аутентификации платформы *ASP.NET* разделяются на следующие основные группы: аутентификация *Windows*, аутентификация на основе файлов *cookie*, аутентификация на основе ролей. В данной работе рассмотрен каждый из этих методов. Платформа *ASP.NET* предоставляет услуги аутентификации и авторизации совместно со средствами аутентификации *IIS*. Она поддерживает также делегирование прав клиентов на уровне запросов. Система безопасности платформы *ASP.NET* на основе ролей позволяет настраивать содержимое выводимой страницы исходя из членства пользователя в ролях. Также приложение *ASP.NET* может предоставлять пользователю собственный интерфейс регистрации и выполнять проверку учетной информации, что упрощает процедуры аутентификации, применяемые в настоящее время в *Web*-сайтах [1].

Методы аутентификации для приложений в *SharePoint* разделяются на два основных типа: *Classic Mode* и *Claims Based*. При использовании *Classic Mode Authentication* пользователю будет доступен только один провайдер аутентификации *Windows*. В случае выбора *Claims Based Authentication* пользователю будут доступны все поддерживаемые провайдеры аутентификации.

Данная работа нацелена на предоставление информации об аутентификации в платформах *ASP.NET* и *Sharepoint*. Рассматриваются характеристики каждого метода аутентификации и детали реализации.

1. Безопасность в ASP.NET

Задать требования аутентификации клиентов приложения *ASP.NET* можно путем добавления соответствующих элементов в конфигурационный файл приложения

web.config. Тип аутентификации клиентов задается с помощью элемента *authentication* как показано на рисунке 1. Атрибут *mode* элемента *authentication* может принимать одно из четырех значений: *windows* (по умолчанию), *Forms*, *Passport* или *None* [1].

```
<!-- Файл: web.config -->
<configuration>
    <system.web>
        <!-- Можно также задать режим Forms,  Passport или None -->
        <!-- По умолчанию применяется Windows -->
        <authentication mode="Windows" />
    </system.web>
</configuration>
```

Рис. 1. Задание режима аутентификации в файле *web.config*.

1.1 Аутентификация Windows

Если задать аутентификацию *Windows*, то все задачи аутентификации перекладываются на *Web*-сервер (режим аутентификации клиентов можно выбрать в *IIS*).

Аутентификация выполняется для ограничения доступа клиентов ко всему приложению или к его частям с помощью элемента *authorization*. Для управления доступом к сайту в элемент *authorization* добавляются подэлементы *deny* и *allow*, задающие отказ в доступе или предоставление доступа определенным пользователям и ролям. Метасимвол * используется для представления всех пользователей, а ? — для представления анонимных пользователей. Например, чтобы отказать анонимным пользователям в праве доступа к сайту, необходимо задать элемент *authorization* с единственным под элементом *deny*, установленным в ?, как показано на рисунке 2 [3].

```
<!-- Файл: web.config -->
<configuration>
    <system.web>
        <authorization>
            <deny users="?" />
        </authorization>
    </system.web>
</configuration>

```

Рис. 2. Отказ в доступе анонимным пользователям.

Элементы *deny* и *allow* поддерживают три атрибута: *users*, *roles* и *verbs*. Значениями этих атрибутов могут быть разделенные запятыми списки пользователей, ролей и команд. Можно также сконструировать более сложные требования авторизации путем добавления нескольких элементов *deny* и *allow*. Проверяя элемент *authorization*,

ASP.NET проходит по спискам элементов *deny* и *allow* в последовательности объявления. Первый встретившийся элемент, удовлетворяющий учетным параметрам запроса, определяет, будет ли предоставлен доступ[2].

Для разных файлов и подкаталогов приложения можно задавать разные правила авторизации. Для изменения параметров авторизации для некоторого каталога приложения нужно добавить в него файл *web.config*, задающий другие параметры. Авторизация всегда применяется сначала согласно параметрам локального файла *web.config*. Дополнительные конфигурационные файлы верхних уровней просматриваются, только если не найдено соответствия в файлах нижних уровней. Следовательно, если в верхнем уровне сайта предоставлены права доступа анонимным пользователям, а в подкаталог добавлен файл *web.config*, отказывающий анонимным пользователям в праве доступа, то для получения доступа к страницам этого подкаталога пользователь должен пройти процедуру аутентификации.

Разные правила авторизации можно также задать для разных файлов и подкаталогов с помощью элемента *location* в файле верхнего уровня *web.config*. На рисунке 3 показан пример ограничения доступа к файлу верхнего уровня *secret.aspx* и к подкаталогу *secret*. При этом анонимным пользователям предоставляется право доступа к остальным страницам приложения[1].

```
<!-- Файл: web.config -->
<configuration>
<system.web>
<authorization>
<!-- По умолчанию доступ предоставляется всем -->
<allow users="*" />
</authorization>
</system.web>
<!-- Элемент location применяется для ограничения
прав доступа к определенному файлу -->
<location path="secret.aspx">
    <system.web>
        <authorization>
            <deny users="?"  />
        </authorization>
    </system.web>
</location>
</configuration>

```

Рис. 3. Тонкая настройка правил авторизации с помощью элемента *location.*

1.2 Аутентификация на основе файлов cookie

С помощью файла *web.config* можно задать применение одной из двух моделей аутентификации на основе файлов *cookie* (*Passport* или *Form*), предоставляющих гибкие

средства управления аутентификацией на уровне приложения. В режиме *Passport* применяется технология аутентификации на основе файлов *cookie* и паспортов *Microsoft*, в которой сайты и пользователи централизованно регистрируются "у *Microsoft*" в целях аутентификации клиентов. В режиме *Forms* используются файлы *cookie*, но подробности процедуры аутентификации определяет разработчик приложения [1].

Принцип действия аутентификации в режиме *Forms* показан на рисунке 4. Когда пользователь первый раз запрашивает ресурс, требующий аутентификации, сервер перенаправляет запрос в выделенную страницу регистрации. Регистрационная страница принимает имя пользователя, пароль, персональный идентификационный номер (*Personal Identification Number — PIN*) и др., а затем приложение аутентифицирует пользователя. Когда пользователь успешно зарегистрирован, сервер предоставляет ему аутентификационный файл *cookie* в зашифрованном виде. Затем пользователь перенаправляется в страницу, которую он запрашивал, однако теперь он предоставляет в запросе аутентификационный файл *cookie* и получает доступ к странице. Аутентификационный файл *cookie* действителен на протяжении сеанса. Следовательно, предоставив его, пользователь имеет право доступа ко всем страницам, определенным для него политикой авторизации. Если аутентификационный файл *cookie* сохранить на клиентском компьютере, то он будет использоваться в последующих сеансах[1, 2].

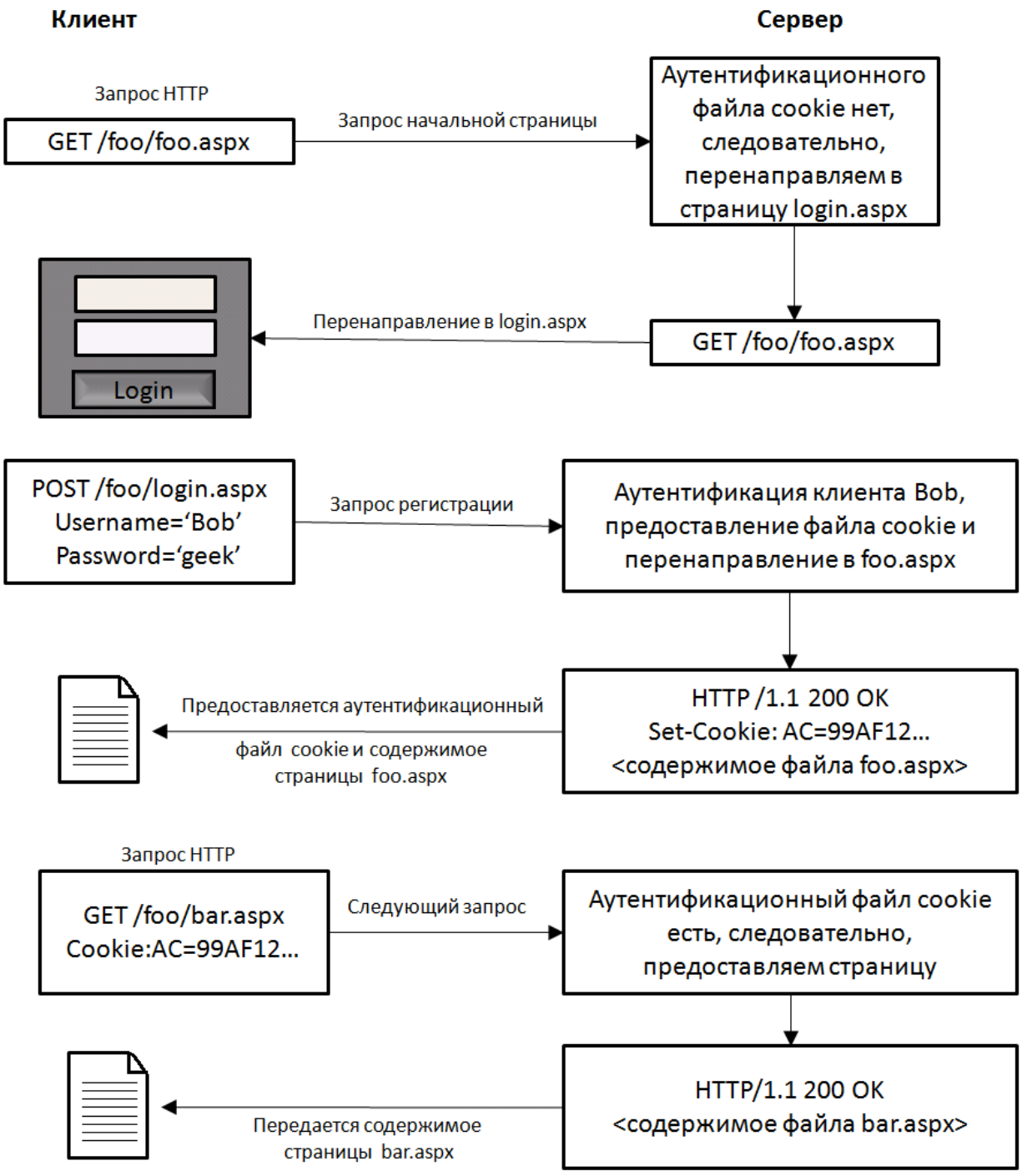

Рис. 4. Принцип действия аутентификации в режиме *Forms*.

Если в файле *web.config* задать аутентификацию *Forms*, то в элементе *forms* можно определить атрибут *loginUrl*, указывающий на страницу, в которую будут перенаправляться пользователи, пытающиеся выполнить аутентификацию и получить доступ к приложению. Затем можно явно отказать в праве доступа анонимным пользователям с помощью элемента *authorization*. Тогда платформа *ASP.NET* будет перенаправлять анонимных клиентов в выделенную страницу регистрации, где они, при

желании, смогут выполнить аутентификацию. Пример файла *web.config*, конфигурирующего аутентификацию в режиме *Forms*, и перенаправляющего анонимных пользователей на страницу *login.aspx* показан на рисунке 5 [2].

```
<!-- Файл: web.config -->
<configuration>
    <system.web>
        <authorization>
            <deny users="?" />
        </authorization>
        <authentication mode="Forms">
            <forms loginUrl="login.aspx" />
        </authentication>
    </system.web>
</configuration>
```

Рис. 5. Задание аутентификации в режиме *Forms*.

1.3 Аутентификация на основе ролей

Платформа *ASP.NET* поддерживает аутентификацию на основе ролей с помощью метода *IsInRole()* интерфейса *iPrincipal*. Если используется интегрированная аутентификация *Windows*, то метод *IsInRole()* проверяет принадлежность клиента к группе *Windows*. Если применяется аутентификация на основе файлов *cookie*, то для использования ролей необходимо определить их и задать отображение пользователей на роли [1, 2].

2 Методы аутентификации на портале в SharePoint 2010

При создании веб-приложения в *MSS 2010* необходимо указать способ аутентификации: *Classic Mode* и *Claims Based.*

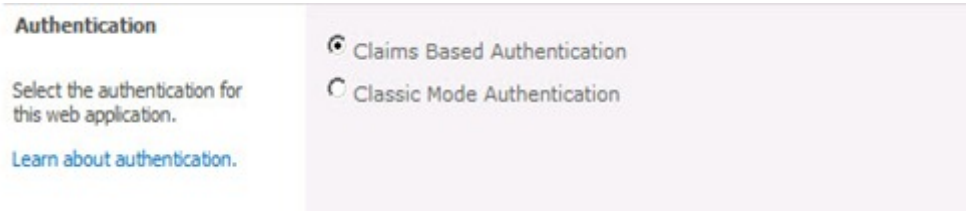

Рис. 6. Выбор способа аутентификации

Классическая модель использовалась в ранних версиях *SharePoint* и поддерживает только одного провайдера аутентификации *Windows*, в рамках которого можно задействовать следующие методы аутентификации [4]

- *Anonymous*

- *Basic*
- *Certificates*
- *NTLM*
- *Kerberos*

Anonymous – анонимная проверка подлинности позволяет пользователям получать доступ к порталу без предоставления учетных данных. Анонимная проверка подлинности позволяет любому человеку просмотреть содержимое портала, поэтому планировать анонимный доступ нужно крайне осторожно.

Basic – обмен сообщений между сервером и клиентом при аутентификации идет открытым текстом, поэтому использовать его без *SSL* крайне не рекомендуется.

Certificates – метод аутентификации, который позволяет настроить аутентификацию на основе пользовательских сертификатов. Данный способ не поддерживается *Microsoft*.

NTLM – данный метод при аутентификации передает по сети только хэш пароля, поэтому его можно считать безопасным. Среди недостатков можно выделить отсутствие возможности аутентифицироваться для доступа к внешним сервисам (отсутствие делегирования).

Kerberos – протокол решающий проблему делегирования, безопасней *NTLM*, быстрей *NTLM*, но требует дополнительной настройки, а именно прописывания *SPN* для учетных записей от которых работает веб-приложение.

Type	Provider	Methods
Classic	Windows	Anonymous, Basic, Digest, Certificates, NTLM, Negotiate (Kerberos or NTLM)
Claims-based	Windows	Anonymous, Basic, Digest, Certificates, NTLM, Negotiate (Kerberos or NTLM)
	FBA	LDAP, SQL database, Other DB, Custom
	SAML	ADFS 2.0, Windows Live ID, Third Party

Рис. 7. Методы аутентификации

Тип аутентификации *Claims-Based* поддерживает провайдера *Windows*, а также провайдеров *FBA(Forms-based authentication)* и *SAML(token-based authentication).*

FBA – аутентификация на основе форм, при которой учетные данные хранятся не в *Active Directory Domain Services*, а во внешних источниках. В качестве внешнего источника может выступать база *SQL, Novell eDirectory, Novell Directory Services (NDS), Sun ONE, AD LDS.* Данный провайдер удобен при интеграции со сторонними службами

каталогов, а также при необходимости аутентифицировать внешних посетителей без использования *AD DS* каталога для создания и хранения учетных записей.

SAML — проверка подлинности осуществляется на основе маркеров *SAML*. Возможна аутентификация с использованием данных *Windows Live Id,* аккаунтов *Facebook.*

Для новых реализаций *SharePoint 2010* необходимо использовать проверку подлинности типа *Claims Based*. При выборе этого режима веб-приложениям будут доступны все поддерживаемые провайдеры и методы.

Веб-приложение может использовать сразу несколько провайдеров аутентификации, например портал доступен по двум именам contoso.com и microsoft.com, по первому срабатывает провайдер *Windows* и *Kerberos,* по второму провайдер *FBA* и аутентификация на основе данных из базы *SQL*. Переключение веб-приложения между типами аутентификации, провайдерами и методами возможно и после создания веб-приложения.

Заключение

В работе приведен обзор методов аутентификации платформ ASP.NET и SharePoint.

Показано, что платформа *ASP.NET* избавляет от необходимости разработки собственных методов аутентификации, поскольку она предоставляет несколько способов ее релизации:

- Аутентификация *Windows* позволяет осуществить аутентификацию по учетной записи ОС;
- Инфраструктура средств аутентификации в режиме *Forms*;
- Предоставление авторизации на основе ролей выполняемых по-разному в зависимости от ролей пользователя;

Также продемонстрировано, что *ASP.NET* предоставляет гибкий программный интерфейс параметров безопасности приложений, посредством изменения нескольких параметров в конфигурационном файле *web.config* [1, 2, 3].

Касательно платформы *SharePoint* приведена информация о существующих способах аутентификации. Указано, какие методы применяются при разработке приложений в ранних и современных версиях *SharePoint.* Также описаны характеристики каждого метода аутентификации.

Литература

1. Оньон Фриц. Основы ASP.NET с примерами на C#. // М.: Издательский дом «Вильямс», 2003. – с. 275-294.

2. Мак-Дональд Мэтью, Фримен Адам, Шпушта Марио. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов , 4-е изд. : Пер. с англ. // М. : ООО “И.Д. Вильямс”, 2011. – с. 806-955.

3. Авторизация ASP.NET. // http://msdn.microsoft.com/ru-ru/hh801952 (дата обращения 18.03.12)

4. Развертывание SharePoint 2010: Шаг 3 — Создание портала // http://itband.ru/2011/03/sharepoint2010-step3/ (дата обращения 20.03.12)

electronic scientific and technical periodical
SCIENCE and EDUCATION
EL № FS 77 - 30569. №0421100025. ISSN 1994-0408

Authentication types

77-30569/370630

 # 04, April 2012

Bol'shakova D.O., Fenske A.V.

Bauman Moscow State Technical University
boldol@mail.ru
fenix.iu8@gmail.com

The article describes authentication types in ASP.NET. The operating principles and examples of implementation are given. Methods of authentication in SharePoint and features of their work are considered.

Publications with keywords: security, authentication methods, Sharepoint 2010
Publications with words: security, authentication methods, Sharepoint 2010

References

1. Fritz Onion. *Essential ASP.NET with Examples in C#*. 1st ed. Addison-Wesley Professional, 2003. 432 p. (Russ. ed.: On'on Frits. *Osnovy ASP.NET s primerami na C#.* Moscow, Izdatel'skii dom «Vil'iams», 2003. 304 p.).

2. Matthew MacDonald, Adam Freeman, Mario Szpuszta. *Pro ASP.NET 4.0 in C# 2010*. Fourth Edition. Apress, 2010. 1617 p. (Russ. ed.: Mak-Donal'd Met'iu, Frimen Adam, Shpushta Mario. *Microsoft ASP.NET 4 s primerami na C# 2010 dlia professionalov*. Moscow, Izdatel'skii dom “Vil'iams”, 2011. 1424 p.).

3. *Russian MSDN. Avtorizatsiia ASP.NET* [Authorization ASP.NET]. Available at: http://msdn.microsoft.com/ru-ru/hh801952, accessed 18.03.12.

4. *ITband.ru. Razvertyvanie SharePoint 2010: Shag 3 - Sozdanie portala* [Deploying SharePoint Server 2010: Step 3 - Create a Portal]. Available at: http://itband.ru/2011/03/sharepoint2010-step3/, accessed 20.03.12.