УДК 681.3
С.Н. Фисун, доцент, канд. техн. наук,
А.И. Копылов, магистрант
Севастопольский национальный технический университет
ул. Университетская д.33, г. Севастополь, Украина, 99053
E-mail: fserg48@mail.ru

АУТЕНТИФИКАЦИЯ И АВТОРИЗАЦИЯ В ПРИЛОЖЕНИЯХ ASP.NET

Рассматриваются способы обеспечения безопасности в Web-приложениях ASP.NET. Основное внимание уделено провайдерам аутентификации и авторизации платформы ASP.NET. С использованием этих провайдеров на языке программирования C# было разработано Web-приложение, которое реализует процесс регистрации и входа пользователей на Web-страницу.

***Ключевые слова**: защита информации, аутентификация, авторизация, платформа .NET.*

На сегодняшний день информация имеет очень большое значение, часто утечка даже самого незначительного её количества может оказаться огромной потерей. Во избежание подобных инцидентов нужно уметь её защищать. Для этого есть множество способов и средств, начиная от обычного шифрования информации методом Цезаря (перемещение каждого символа на 3 позиции) и заканчивая максимально безопасными, но в то же время более дорогостоящими программно-аппаратными средствами, такими как карты доступа и средства биосканирования. Наряду с такими дорогими и высокотехнологичными системами безопасности существуют технологии более низкого ранга, но всё же способные обеспечить безопасность на программном уровне [1]. Среди них можно выделить криптографические сервисы .NET и Java. Данные сервисы можно рассматривать как альтернативу интерфейсу CryptoAPI корпорации Microsoft, который подробно рассматривается в работе [1]. Они позволяют использовать различные криптографические алгоритмы для шифрования данных, такие как AES, DES, 3DES, MD5, RSA и многие другие [2]. Из всех вышеназванных алгоритмов шифрования наибольший интерес для данной работы представляет алгоритм MD5, поскольку именно данный алгоритм, в большинстве случаев, применяется для шифрования паролей в процессе аутентификации и авторизации пользователей. В данной статье рассматриваются средства, которые наиболее распространены в среде ASP.NET и платформе .NET Framework. Основное внимание будет уделено таким аспектам безопасности ASP.NET как аутентификация и авторизация пользователей.

Целью работы является разработка программного модуля, который будет выполнять регистрацию входа пользователей в систему (сайт). Для обеспечения аутентификации и авторизации необходимо использовать провайдеры платформы ASP.NET.

Для начала следует провести чёткую грань между понятиями аутентификации и авторизации. **Аутентификация** (идентификация) — процесс выяснения и проверки личности пользователя с помощью получения от него пары логин/пароль и сравнения их с каким-либо заслуживающим доверия источником. Наглядным примером аутентификации может служить вход в Windows. **Авторизация** — это проверка прав доступа к определённому ресурсу или проверка привилегий на выполнение определённых действий. Например, авторизация в Windows проходит каждый раз, когда вы пытаетесь изменить настройки системы, добавить или удалить пользователей, и, если у вас не окажется соответствующих прав для выполнения подобных действий, то операционная система вырабатывает сведения о соответствующей ошибке.

Проектируя приложение, необходимо понимать связь между аутентификацией Internet Information Services (IIS) и архитектурой защиты Microsoft® ASP.NET. Тогда вы сможете правильно аутентифицировать пользователей и получать корректный контекст защиты в приложении. Стоит обратить внимание на то, что параметры защиты ASP.NET-приложения и IIS никак не связаны между собой и могут использоваться как независимо друг от друга, так и в сочетании.

IIS хранит параметры конфигурации, связанные с защитой, в метабазе. В свою очередь ASP.NET хранит параметры защиты в конфигурационных XML-файлах. Хотя обычно это облегчает развертывание приложений с точки зрения безопасности, выбранная в приложении модель защиты потребует правильной настройки как метабазы IIS, так и вашего ASP.NET-приложения (через его конфигурационный файл Web.config).

Взаимосвязь между IIS и ASP.NET показана на рисунке 1:

В ASP.NET аутентификация осуществляется на основе провайдеров аутентификации, которые представляют собой модули кода, проверяющие удостоверения и реализующие прочие функции защиты, например, генерацию cookie. ASP.NET поддерживает следующие провайдеры аутентификации.

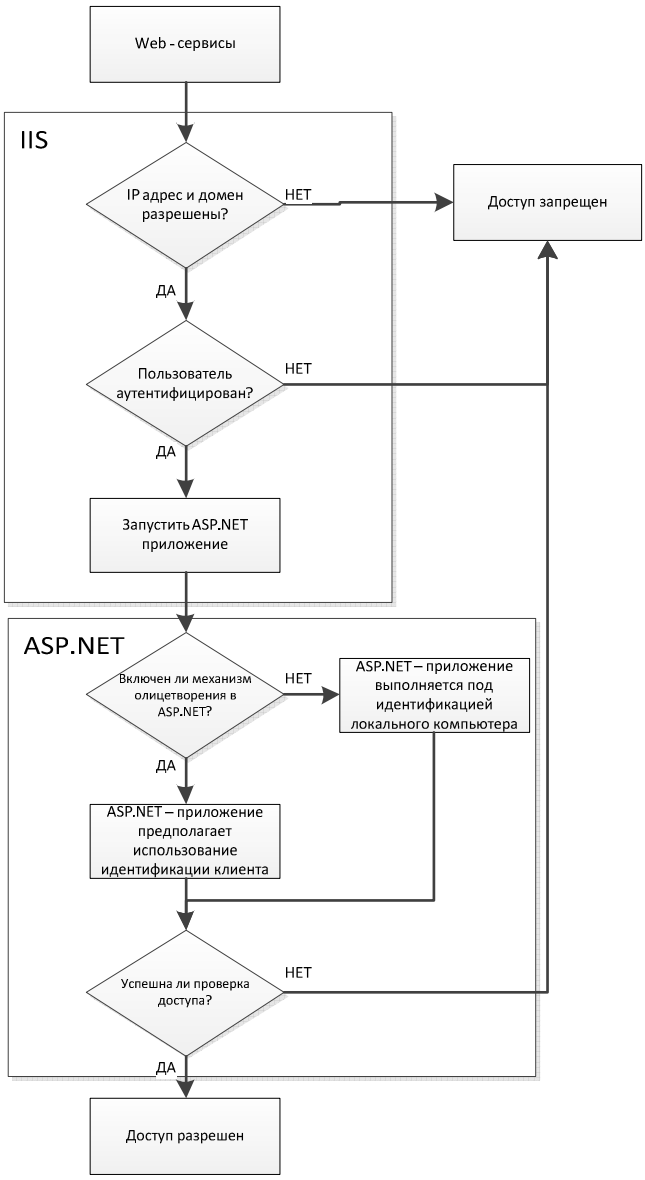

Рисунок 1 – Взаимосвязь средств защиты IIS и ASP.NET

Forms Authentication (аутентификация на основе форм). При применении этого провайдера все неаутентифицированные запросы перенаправляются на указанную HTML-форму путем клиентской переадресации. После этого пользователь может ввести свои учетные данные (удостоверения защиты) и отправить форму обратно на сервер. Если приложение аутентифицирует запрос (логика проверки специфична для конкретного приложения), ASP.NET генерирует cookie, содержащий удостоверения или ключ для повторного получения идентификации клиента. Последующие запросы содержат этот cookie в своем заголовке, поэтому никакая аутентификация больше не требуется.

Passport Authentication (аутентификация через Passport). Это централизованная служба аутентификации, предоставляемая Microsoft и предлагающая участвующим сайтам механизмы единой Michael Howard регистрации и подписки на сервисы. ASP.NET в сочетании с Microsoft Passport SDK предлагает пользователям Passport функциональность, аналогичную Forms Authentication;

Windows Authentication (аутентификация через Windows). Этот провайдер пользуется возможностями IIS. После того как IIS проводит аутентификацию, ASP.NET использует маркер аутентифицированной идентификации для авторизации доступа [3].

Рассмотрим пример аутентификации на основе форм. При использовании этой аутентификации запрос пользователя все равно должен пройти через IIS (и, таким образом, отработать аутентификацию Windows). Поэтому на уровне аутентификации Windows должен быть открыт доступ для анонимных пользователей. Далее аутентификация выполняется по следующему алгоритму.

1. Пользователь запрашивает доступ к защищенной Web-странице.

2. Сервер проверяет, включен ли в этот запрос authentification cookie. Если да, то происходит авторизация. При этом сравнивается имя пользователя, которое включено в cookie, с параметрами авторизации в web.config и на основании этого принимается решение — пускать пользователя или нет.

3. Если authentification cookie не встроен в запрос, то ASP.NET перенаправляет пользователя на logon page (путь к нему определяется в файле конфигурации приложения). На этой странице пользователь должен ввести имя и пароль.

4. Код приложения на logon page проверяет введенное пользователем имя и пароль, и, если информация корректна, то пользователя допускают к сеансу authentification cookie (п.2). Если нет — выдается сообщение о том, что доступ запрещен.

Описанный способ аутентификации может быть представлен в виде следующей диаграммы последовательностей (рисунок 2):

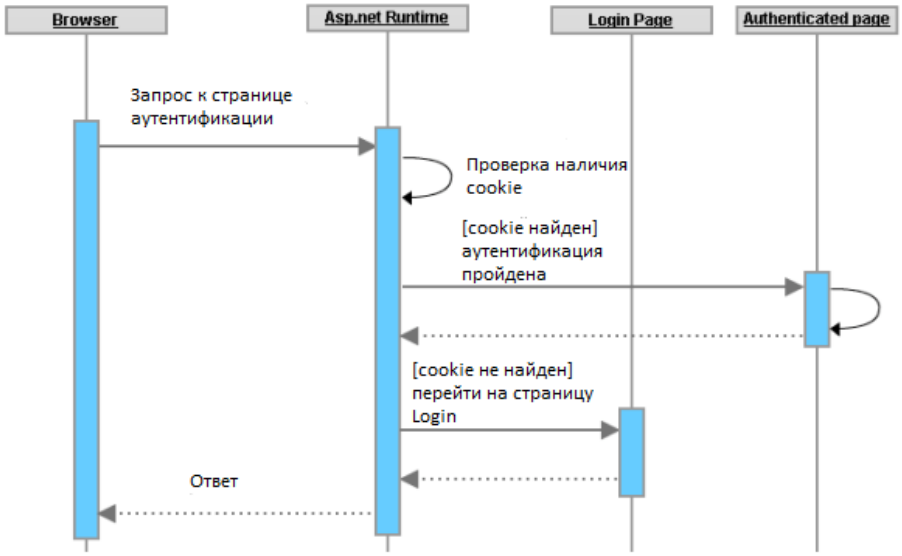

Рисунок 2 – Аутентификация с помощью форм

Именно этот способ аутентификации был использован в ходе написания приложения ASP.NET.

Для реализации вышесказанного на практике пропишем в файл web.config для приложения следующий код:

```
<system.web>
    <authentication mode="Forms">
        <forms name=".name" loginUrl="login.aspx" />
    </authentication>
</system.web>
```

То, что содержится в теге forms, относится к настройкам cookie. Атрибут name — это суффикс cookie, а loginURL — путь к странице аутентификации, на которую будут перенаправляться все неаутентифицировавшиеся запросы.

Теперь перейдем к рассмотрениям вопроса авторизации пользователя в приложениях ASP.NET. Для этого могут быть использованы следующие модули.

1. Модуль авторизации по URL – UrlAuthorizationModule (системный HTTP модуль) использует авторизационные правила из web.config (указанные в элементе < *authorization*>) для проверки прав доступа вызывающего к запрашиваемому файлу или директории;

2. Другой модуль Windows аутентификации – FileAuthorizationModule (так же системный HTTP модуль) проверяет полномочия вызывающего на доступ к требуемому ресурсу. Этот модуль задействуется только в случае конфигурирования Windows аутентификации для ASP.NET приложения. Полученный от IIS маркер доступа Windows проверяется с ACL который защищает ресурс;

3. Роли .NET так же могут быть использованы (декларативно или программно) для обеспечения вызывающему необходимого доступа к запрашиваемому ресурсу или операции.

Код приложения пользователя обращается к тем или иным локальным и удаленным ресурсам под правами конкретного пользователя (учетной записи Windows). По умолчанию, ASP.NET не подменяет (имперсонирует) пользователя и, следовательно, выполняется под выделенной ASP.NET учетной записью. Альтернативными опциями являются имперсонирование вызывающего пользователя (аутентифицированного IIS пользователя Windows) или конфигурирование выделенной учетной записи.

Основными этапами процесса авторизации пользователя в ASP.NET приложении являются следующие.

1. Уровень IIS.

При выключенной Anonymous аутентификации IIS разрешает доступ только пользователям, которых он может аутентифицировать в собственном домене или в доверяемом домене. Для статических файлов (файлы, для которых нет соответствия в ISAPI расширении – например, .jpg, .gif, .htm) IIS использует NTFS разрешения, ассоциированные с запрашиваемым файлом.

2. Уровень ASP.NET.

– **UrlAuthorizationModule**

Вы можете сконфигурировать <*authorization*> элемент в web.config и управлять через него, какие пользователи и группы имеют доступ к приложению. Можно конфигурировать на уровне отдельных файлов или целиком приложения. Авторизация основывается на **IPrincipal** объекте, который хранится в свойстве **HttpContext.Current.User**. Таким образом, авторизация на основе сопоставления имен пользователей и url запрашиваемых ресурсов работает для все типов аутентификации ASP.NET (т.е. не только для Windows, но и для Forms и для Pasport)

Примечание. Правила записи имен пользователей и групп могут зависеть от выбранной схемы аутентификации.

– **FileAuthorizationModule**

Для файлов, для которых существует соответствие в конфигурации IIS для ASP.NET ISAPI расширения (aspnet_isapi.dll), автоматически осуществляется проверка доступа по отношению к маркеру доступа WIndows (который может быть маркером доступа IUSR_MACHINE)

– **FileAuthorizationModule**.

Данный класс осуществляет проверки только для запрашиваемого в процессе web запроса файла. Он не проверяет доступ к файлам, к которым обращение происходит из кода ASPX страницы (или вызываемых из нее компонент). Например, при обращении в коде ASPX страницы к gif файлу. В ACL к ASPX странице запрещается доступ определенному пользователю и gif файл не может быть получен через обращение к ASPX странице этим пользователем. Но при обращении к Gif файл напрямую через web запрос проверка будет осуществляться только IIS и, если доступ к gif файлу разрешен, запрос будет обработан успешно.

– Запросы на доступ для Principal и явные проверки на принадлежность ролям. Этот механизм обеспечивает система безопасности доступа кода — Code Access Security (CAS). Вы можете использовать запросы доступа для Principal в коде декларативно и программно как дополнительный авторизационный механизм. Проверки на доступ для Principal позволяют контролировать доступ к классам, методам или индивидуальным фрагментам кода на основе идентификации пользователя и принадлежности его к группам, что определяется IPrincipal объектом, присоединенным к текущей нити процесса. Проверки на доступ для Principal приводят к генерации исключения в случае запрещения доступа. Этим обработка таких проверок отличается от IPrincipal.IsInRole метода, просто возвращающего булево значение.

При Windows аутентификации ASP.NET автоматически присоединяет **WindowsPrincipal** объект, который представляет аутентифицированного пользователя, к текущему web запросу (в свойстве HttpContext.User). При Forms и Passport аутентификации создается GenericPrincipal объект с соответствующим идентифицированным пользователем (без ролей – они должны быть добавлены кодом пользователя в событии AuthenticateRequest объекта Application) и записывается в HttpContext.User [3].

Рассмотрим простейший пример конфигурирования авторизации в ASP.NET приложении. Для этого в файле web.config нужно указать для какого файла или каталога кому будет открыт доступ. Для этой цели используется тег location:

```
<location path="ShoppingCart.aspx">
    <system.web>
        <authorization>
            <deny users="?" />
        </authorization>
    </system.web>
</location>
```

Если в атрибуте path будет указан каталог, то настройки распространятся на все файлы в этом каталоге и подкаталогах. Если будет указан конкретный файл (Web-форма), то настройки будут распространены только на эту Web-форму.

<*deny users="?"* /> означает, что запрещен доступ всем анонимным пользователям.

Чтобы разрешить доступ пользователю User, код может быть таким:

```
<authorization>
    <allow users="User" />
</authorization>
```

Если включить этот код в файл web.config для всего приложения, то параметры будут применены для всего Web-приложения.

Таким образом, можно сделать вывод, что платформа ASP.NET предоставляет ряд декларативных и программных механизмов авторизации, которые могут быть использованы совместно с различными схемами аутентификации. Это позволяет разрабатывать продуманную стратегию системы безопасности, которая может быть тонко сконфигурирована для настраиваемого доступа.

Библиографический список использованной литературы

1. Фисун С.Н. Использование криптодрайверов для защиты информации от несанкционированного доступа / С.Н. Фисун, О.И. Куржеевская / Зб. наук. праць Севастопольського військово-морського Ордена Червоної Зірки інст. ім. П.С. Нахімова. — Севастополь, 2008. — Вип. 2 (15) — С. 29–33.

2. Фисун С.Н. Использование криптографических сервисов .NET и Java для защиты информации от несанкционированного доступа / С.Н. Фисун, А.И. Копылов / Вестник СевНТУ. Серия Информатика, электроника, связь: сб. науч. тр. — Севастополь: СевНТУ, 2011. — Вып. 114/2011. — С. 81–86.

3. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0 / Э. Троелсен. — М.: Вильямс, 2011. — 1392 с.

Поступила в редакцию 13.03.2011 г.

Фісун С.М., Копилов А.І. Аутентифікація й авторизація в програмах ASP.NET

Розглядаються способи забезпечення безпеки в Web-програмах ASP.NET. Основна увага приділена провайдерам аутентифікації і авторизації платформи ASP.NET. З використанням цих провайдерів на мові програмування C# була розроблена Web-програма, яка реалізує процес реєстрації та входу користувачів на Web-сторінку.

Ключові слова: захист інформації, аутентифікація, авторизація, платформа .NET.

Fisun S.N., Kopylov A.I. Authentication and Authorization in ASP.NET applications

Consider ways of ensuring security in ASP.NET Web-based applications. The main attention is paid to use some ASP.NET providers of authentication and authorization. By using these providers and the C # programming language was developed Web application that implements a registration process and logon users to the Web-page.

Keywords: information security, authentication, authorization platform .NET.