Эссе

по курсу "Защита информации", кафедра радиотехники, Московский физико-технический институт (ГУ МФТИ), http://www.re.mipt.ru/infsec

Обеспечение безопасности web-приложений под управлением Microsoft IIS и ASP.NET

(Securing Web-applications under Microsoft IIS and ASP.NET)

Соколов Григорий, 215 группа.

Май 2006 года.

Введение

Обеспечение безопасности веб-приложений – одна из самых важных и актуальных проблем на сегодняшний день. И пусть она не столь животрепещуща для владельцев обыкновенных сайтов и собственных домашних страничек, но если необходимо сделать сайт электронной коммерции (или любой другой, имеющий дело с конфиденциальной информацией, в основном, конечно, финансовой) – она является одной из основополагающих!

В данной работе я рассмотрю процесс защиты доступа для интернет-приложений в системе ***Microsoft*** Windows (в данном случае – Windows 2003 Server или Windows XP) на основе IIS (Internet Information Services) и ASP.NET. И, хотя существует много альтернативных вариантов размещения своего сайта, я решил остановиться на этой конфигурации по следующим причинам:

- Большинство пользователей, так или иначе, имели дело с системами Windows, и, следовательно, им легче будет понять процесс защиты, так как он основан на безопасности самой Microsoft Windows.
- Будучи детищами той же корпорации, сервисы IIS и ASP.NET обеспечивают высокую интеграцию в эту систему.
- Платформа ASP.NET чрезвычайно проста и удобна для разработки сайтов, в которой реализована гибкая и мощная многоуровневая модель безопасности, облегчающая механизм защиты веб-приложений.
- И, наконец, эта именно та система, с которой я достаточно неплохо знаком, в отличие от остальных – а потому могу рассказать чуть больше.

Определение требований безопасности

Решив внедрить в приложение систему безопасности, сначала необходимо подумать, зачем, собственно, это нужно! Это может быть охрана частной информации, или, к примеру, служба подписки. А может такое случиться, что система безопасности, как таковая, Вам вообще не нужна, и Вы хотите просто создать систему регистрации для пользователей для последующей их персонализации. В зависимости от цели – и средства реализации будут отличаться.

К примеру, ограничивая доступ к определённой части сайта, необходимо быть уверенным (ой), что пользовательские данные хорошо защищены – возможно, хранятся в базе данных, пароль к которой знаете только Вы, да Господь Бог. И, ко всему прочему, надо гарантировать, что само приложение не будет отсылать пользователям конфиденциальную или секретную информацию (без какой бы то ни было защиты).

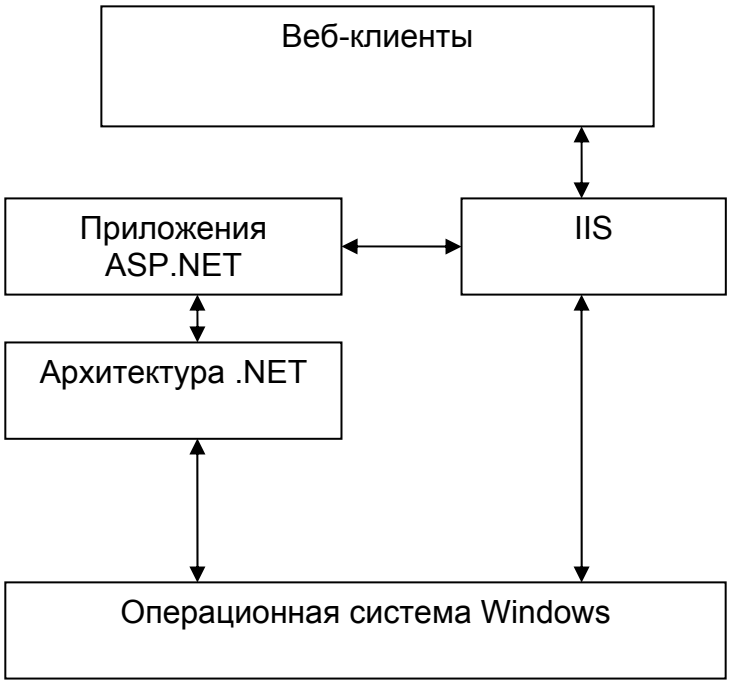
Модель безопасности ASP.NET

Для начала, введём краеугольные понятия системы безопасности!

- **Аутентификация** (Authentication) – процесс подтверждения личности пользователя. (Он тот, за кого себя выдаёт.)
- **Авторизация** (Authorization), следующая за аутентификацией, – процесс определения прав и полномочий пользователя на определённые действия. (Просмотра страницы, доступа к базе данных, etc.)
- **Имперсонация** (Impersonating) – состояние, при котором определённые операции исполняются под другим именем. (Анонимные пользователи веб-сайта аутентифицируются при помощи учётной записи IIS_ServerName, использующейся по умолчанию.)

Приложение ASP.NET реализовывает базовый уровень безопасности, автоматически блокируя запросы на определённые типы файлов (например, файлы с исходным кодом, или конфигурационные файлы).

Рассмотрим последовательность обработки запроса. Они обрабатываются сервером IIS, а затем передаются сервису ASP.NET, если тип файла зарегистрирован за ним. В противном случае IIS самостоятельно пытается отослать пользователю соответствующий файл (если учётная запись Windows это позволяет.)



1. IIS пытается аутентифицировать пользователя. Обычно, запросам анонимных пользователей присваивается учётная запись IIS_ServerName.
2. Если аутентификация прошла успешно, и файл зарегистрирован за ASP.NET, IIS передаёт запрос туда, сопровождая его информацией об аутентифицированном пользователе. Далее ASP.NET производит собственную аутентификацию, в зависимости от настроек, и, собственно, самой запрашиваемой страницы.
3. В случае успеха, ему разрешается запрос страницы aspx. Код самой страницы, естественно, может сам производить дополнительные проверки безопасности.
4. При взаимодействии ASP.NET с внешними ресурсами (любыми ресурсами в операционной системе или вовне) происходит очередная проверка – на этот раз самой Windows. Обычно код ASP.NET использует собственную учётную запись, обладающую достаточно широкими полномочиями. Но, при использовании имперсонации, и это можно изменить.
5. Если все вышеизложенные пункты пройдены – доступ разрешён!

Основные возможности обеспечения безопасности

Вообще говоря, их две.

- Разрешение анонимности IIS, но применение моделей аутентификации ASP.NET. (Обычно основанных на формах – это упрощает процесс управления регистрации пользователей на сайте и позволяет создавать собственный код для регистрации и аутентификации.)
- Запрещение анонимности в IIS. Это приводит к необходимости интегрированной аутентификации Windows. То есть для каждого пользователя необходимо создать свою учётную запись.

Вполне очевидно, что первый пункт используется в открытых веб-приложениях, где требуется возможность создания большого числа пользователей (всё те же электронные магазины); в то время как второй прекрасно подходит для intranet сайтов.

Аутентификацию windows подробно рассматривать не буду – если это делать подробно, то нужно писать, как минимум, ещё одно эссе, а если кратко, то всё просто: IIS возлагает все проблемы по аутентификации пользователей на плечи Windows.

Аутентификация через формы

Самым распространённым способом является использование *специальных элементов* cookie. В случае его отсутствия, пользователь перенаправляется на страницу регистрации, где этот cookie и создавался. (В случае успешной регистрации, естественно.)

Сразу возникает вопрос о безопасности: возможность подделки специальных элементов cookie. В ASP.NET этот вопрос решили программисты из корпорации Microsoft – аутентификационные элементы cookie автоматически защищаются с помощью сложного алгоритма шифрования и хеширования. Вообще говоря, все эти настройки можно менять в конфигурационном файле приложения:

- All – использовать и шифрование, и проверку целостности.
- None – использовать незащищённые cookies.
- **Encryption** – только шифрование. (Используются алгоритмы Triple DES или DES)
- **Validation** – проверка подлинности дошедшего cookie.

По умолчанию, используются все виды защиты аутентификационного cookie (аутентификационного билета). Так что создателю веб-приложения остаётся только реализовать страницу регистрации и позаботиться об основных принципах обеспечения безопасности – в частности, по возможности не передавать никаких секретных и конфиденциальных данных между пользователем и сервером. Ибо о безопасной передаче этой информации позаботились создатели ASP.NET.

Эта модель также ограничивать доступ к отдельным файлам или каталогам для различных пользователей без нарушения конфиденциальности и принципов безопасности. В качестве дополнения можно упомянуть, что нет необходимости держать списки пользователей в отдельной базе данных; их можно хранить в конфигурационном файле приложения вместе с хешированным паролем.

Указанные выше способы аутентификации построены на принципе «всё или ничего» – разрешая или, наоборот, запрещая доступ. Для выхода из этого положения предусмотрена система **ролей**. Она позволяет присваивать пользователям определённые роли (Administrator, User, etc) и на их основе реализовывать систему многоуровневого доступа.

Заключение

В этом эссе я вкратце рассмотрел основные принципы и возможности по обеспечению безопасности в современных веб-приложениях. Естественно, я многого не упоминал, но, вполне очевидно, что возможности этой системы чрезвычайно гибки и мощны, что сильно упрощает процесс создания и защиты интернет-сайта.

Используемые источники.

1. “ASP.NET” – Мэтью Макдональд (Matthew MacDonald); Санкт-Петербург «БХВ-Петербург» 2003
2. Microsoft IIS 6.0 Help
3. Microsoft Developer Network Library for Visual Studio 2005
4. “Программирование на платформе Microsoft® .NET Framework” – Джеффри Рихтер (Jeffrey Richter); Санкт-Петербург «Питер» 2005