УДК 004.056.53 : 371.315.7

БЕЗПЕЧНА АУТЕНТИФІКАЦІЯ КОРИСТУВАЧІВ WEB-ІНТЕРФЕЙСІВ СИСТЕМ ПІДТРИМКИ ДИСТАНЦІЙНОЇ ОСВІТИ

О.В. Слободянюк
Кам'янець-Подільський державний університет
e-mail: alexwizard@gmail.com

Питання забезпечення належного рівня безпеки та захисту різноманітних Інтернет-орієнтованих систем останнім часом стало особливо актуальним. Однією з ключових проблем інтернет-освіти залишається проблема аутентифікації користувача при перевірці знань. Більшість організацій, що пропонують послуги дистанційної освіти до цих пір надають перевагу очній формі контролю знань слухачів аніж онлайновій.

Існує багато різноманітних способів аутентифікації користувача, тобто визначення його достовірності та приналежності до відповідної групи безпеки. Серед них можна виокремити два великих напрями, кожен з яких містить в собі різні способи та методи. У першому напрямі передбачається використання апаратних засобів авторизації користувачів. До них відносяться:

- Біометричні сенсори (за відбитками пальців, формою сітківки ока, голосові сенсори, форма кисті рук).
- Смарт-карти (магнітна картка або картка із вбудованою мікросхемою).
- HASP-ключі (програмно-апаратні комплекси, які містять певний код, процедури або будь-які інші унікальні дані, за якими можна ідентифікувати легальність введених користувачем персональних даних).
- Магніто-оптичні носії інформації (використовуються фізичні особливості носіїв – компакт-дисків, дискет, флеш-дисків – система StarForce, SecuRom, SafeDisc).

Інший напрям стосується лише програмних способів та методів перевірки «особи» користувача. Це можуть бути різноманітні програмні комплекси та засоби, що здатні забезпечити надійний програмний «тунель» передачі даних за допомогою різноманітних криптосистем. До таких засобів слід віднести протокол віддаленої аутентифікації PAP (Password Access Protocol) – протокол доступу за паролем. Це найбільш розповсюджений спосіб передачі даних, але й, водночас, найбільш незахищений оскільки облікові дані користувачів пересилаються на сервер в незашифрованому вигляді. Протокол строгої аутентифікації типу «запит-відповідь» CHAP (Challenge-Handshake Authentication Protocol) навпаки характеризується достатнім рівнем безпеки. В ньому дані не передаються безпосередньо від «клієнта» до «сервера», а лише приймають участь у складних математичних обчисленнях.

Також існують гібридні системи, що поєднують різні підходи до реалізації процесу авторизації. До таких систем контролю доступу основаних як на програмних, так і на апаратних засобах захисту відноситься система **Kerberos**. В основі цієї системи авторизації лежать три компоненти: база даних (містить інформацію по всім мережевим ресурсам, користувачам, паролям, шифрувальним ключам та інше), авторизаційний сервер (опрацьовує всі запити користувачів на предмет отримання того чи іншого виду мережевих послуг, при чому паролі користувачів по мережі не передаються, що також підвищує степінь захисту інформації), сервер видачі дозволів (отримує від авторизаційного серверу "пропуск", що містить ім'я користувача, його мережеву адресу,

час запиту і ряд інших параметрів, а також унікальний сесійний ключ – пакет, що містить зашифрований за алгоритмом DES "пропуск", після отримання і розшифровки якого сервер видачі дозволів перевіряє запит і порівнює ключі а потім дає дозвіл на використання мережевої апаратури або програми).

Існуючі на даний момент системи підтримки процесу навчання на відстані як правило крім простого механізму листування та спілкування через веб-чати чи Інтернет-пейджери між студентами та тьюторами, включають в собі web-орієнтовані інтерфейси користувача. Даний елемент представляє собою програмний сервіс виконаний за клієнт-серверною архітектурою. І тому саме й виникає проблема розподілу користувачів за рівнями привілеїв та широтою прав доступу. Серед засобів що дозволяють розробляти подібні інтерфейси можна виділити п'ять найбільш розповсюджених:

1) CGI (Common Gateway Interface)
2) ASP.NET (Active Server Pages)
3) PHP (Personal Home Pages)
4) JSP (JavaServer Pages technology)
5) Micromedia ColdFusion

Відповідно кожен з цих засобів адаптований до роботи на певному веб-сервері. Як правило ASP.NET працює на IIS (Internet Information Services), PHP на Apache (також може бути і на IIS), JSP на Tomcat. Питання підтримки належного рівня безпеки в таких системах неможливо вирішити лише за рахунок коректного та грамотного програмного коду. Воно вирішується шляхом виконання комплексу дій та операцій із, в першу чергу, веб-сервером та інтерпретатором відповідної сервер-орієнтованої мови програмування.

Всі перераховані вище засоби розробки підтримують стандартні методи забезпечення конфіденційності даних, що передаються користувачами в процесі реєстрації в системі. Серед інших окремо стоїть технологія ASP.NET, що орієнтована і, відповідно, наділена всіма достоїнствами та перевагами платформи .NET Framework

В ASP.NET аутентифікація проводиться на основі провайдерів аутентифікації, які являють собою модулі коду, що реалізують різноманітні функції захисту, наприклад генерацію cookie. ASP.NET підтримує наступні провайдери аутентифікації.

- **Forms Authentication (аутентифікація на основі форм**). При використанні цього провайдера всі неаутентифіковані запити перенаправляються на вказану HTML-форму шляхом клієнтської переадресації. Після цього користувач може ввести свої облікові дані (посвідчення захисту) і відправити форму назад на сервер. Якщо додаток аутентифікує запит (логіка перевірки специфічна для кожного конкретного додатку), ASP.NET генерує cookie, що містить посвідчення або ключ для повторного отримання ідентифікації клієнта. Наступні запити містять цей cookie в своєму заголовку, тому ніяка аутентифікація більше не потрібна.
- **Passport Authentication (аутентифікація через Passport)**. Це централізована служба аутентифікації, що надається Microsoft і яка пропонує сайтам-учасникам механізми єдиної реєстрації і підписки на сервіси. ASP.NET в поєднанні з Microsoft Passport SDK пропонує користувачам Passport функціональність, аналогічну Forms Authentication.
- **Windows Authentication (аутентифікація через Windows)**. Цей провайдер користується можливостями IIS. Після того як IIS проводить аутентифікацію, ASP.NET використовує маркер аутентифікації ідентифікації для авторизації доступу [1].

Для того щоб задіяти той чи інший спосіб аутентифікації у файлі налаштувань віртуального каталогу сайту web.config необхідно прописати виклик відповідного провайдера аутентифікації. Це можна виконати вручну за допомогою будь-якого текстового редактора прописавши наступні рядки.

<authentication mode = "[*ProviderName*]">

</authentication>

Змінна [*ProviderName*] може приймати значень *Windows*, *Cookie*, *Passport* або *None*.

Якщо розробка ведеться за допомогою середовища VisualStudio 2005/2003 зручно користуватися майстром налаштувань ASP.NET. Для цього лише необхідно вибрати відповідний пункт меню і на закладці Security обрати потрібний тип аутентифікації. В тому разі коли сайт планується використовувати в межах однієї локальної або корпоративної мережі яка працює під управлінням контролером домену Windows NT Server із розгорнутою на ньому Active Directory, зручно користуватися аутентифікацію за допомогою облікових записів Windows. Для цього достатньо вибрати даний спосіб аутентифікації на оснастці IIS (закладка Directory Security). При цьому не потрібно писати ніякого специфічного коду для аутентифікації. Коли аутентифікація проводиться даним способом, ASP.NET створює об'єкт Windows Principal і зв'язує його з контекстом додатку для аутентифікованого користувача.

Для проведення базової аутентифікації в ASP.NET також може використовуватися класична схема. Спочатку користувач вводить свої облікові дані у веб-формі. Далі ці дані пересилаються на сервер де і проводиться звірка із існуючими обліковими записами. І якщо на етапі введення користувачем даних та перевірки їх на сервері можна досить просто захиститися від можливого перехоплення потоку даних програмами-шпигунами, то на етапі передачі необхідне надійне шифрування. IIS, що налаштований на базову аутентифікацію, вимагає від браузера передачі посвідчення користувача по HTTP. При цьому паролі й імена користувачів кодуються за основою Base64. Однак передача паролів та імен користувачів даним способом вважається не зовсім безпечним, оскільки він нестійкий до дешифрування. Тому при використанні даного способу аутентифікації можна використовувати власні процедури шифрування як імені користувача, так і паролю. Прикладом такої реалізації може являтися наступний код, в якому за допомогою стандартних засобів програмного інтерфейсу CryptoAPI реалізований алгоритм потокового симетричного шифрування Rijndael (AES):

```
Sub Cipher_Rijndael()
   Dim Rijndael1 As New RijndaelManaged
   Dim TextToCrypt As String = “Some text”
   Dim CryptedText As String
   Dim textConverter As New ASCIIEncoding()
   Dim Encrypted() As Byte
   Dim MessageForEncrypt() As Byte
   Dim Key() As Byte
   Dim IV() As Byte

'Створення ключа шифрування та вектора ініціалізації.
   Rijndael1.GenerateKey()
```

```
    Rijndael1.GenerateIV()
    Key = Rijndael1.Key
    IV = Rijndael1.IV
    Dim Encryptor As ICryptoTransform = Rijndael1.CreateEncryptor(Key, IV)

'Шифрування потокових даних.
    Dim msEncrypt As New MemoryStream()
    Dim csEncrypt As New CryptoStream(msEncrypt, Encryptor,    CryptoStreamMode.Write)

'Перетворення рядкової величини у масив байтів.
    MessageForEncrypt = textConverter.GetBytes(TextToCrypt)

'Запис даних в криптопотік і скидання буферу.
    csEncrypt.Write(MessageForEncrypt, 0, MessageForEncrypt.Length)
    csEncrypt.FlushFinalBlock()

'Переведення потокових зашифрованих даних у масив байтів
    encrypted = msEncrypt.ToArray()

'Перетворення масиву байтів у рядкову величину
    CryptedText = textConverter.GetString(encrypted)

'Виведення на екран оригінального та зашифрованих повідомлень
    Console.WriteLine("Original text -> ", TextToCrypt)
    Console.WriteLine("Crypted text -> ", CryptedText)
End Sub
```

За умови попередньої генерації та обміном секретними ключами між «клієнтом» та «сервером» необхідно просто із локального місця зберігання завантажити необхідний ключ. Даний алгоритм шифрування Rijndael дозволяє використовувати секретні ключі довжиною 256 байт, чого цілком достатньо для забезпечення надійності передачі облікових даних користувача та подальшого проведення процесу аутентифікації та авторизації.

Використані джерела

1. Керчер Джеф. Аутентификация в ASP.NET: руководство по защите в .NET. // Альманах программиста, том 2. Microsoft ASP.Net, Web-сервисы, Web-приложения. – 2003. – с.11-13.

2. Новичков А., Сардарян Р. Анализ рынка средств защиты от копирования и взлома программных средств. URL: http://citforum.ru/security/articles/analis

3. J.D. Meier, Alex Mackman, Michael Dunner, and Srinath Vasireddy. Building Secure ASP.NET Applications: Authentication, Authorization, and Secure Communication. URL: http://msdn.microsoft.com/library/en-us/dnnetsec/html/SecNetch03.asp