УДК 004.7; 004.056/057

В. И. Гайдаржи, Н. А. Смирнова
Национальный технический университет Украины «КПИ»
проспект Победы, 37, 03056 Киев, Украина

Защита информационных ресурсов

Рассмотрена классификация существующих угроз безопасности web-портала, проведен обзор существующих систем управления его содержимым (CMS) и их составляющих — систем безопасности. Основная часть статьи посвящена проектированию универсальной системы безопасности высокого уровня, которая призвана обеспечить защиту ресурса от большинства видов атак.

Ключевые слова: *система безопасности, безопасность, информация, информационные ресурсы, атака.*

Введение

В настоящее время в мире используется огромное количество информации, хранимой в виде информационных ресурсов сети Интернет. С развитием использования сетевых информационных технологий доступа к ресурсам растет и количество разнообразных атак на эти ресурсы. Поэтому в любой организации рано или поздно возникает проблема обеспечения безопасности ресурсов для предупреждения и нейтрализации последствий подобных атак. Решение данной проблемы предполагает установку соответствующей потребностям предприятия системы безопасности информационных ресурсов и организацию ее функционирования.

Говоря об информационной безопасности, все, в первую очередь, имеют в виду защиту от вирусов и «хакеров», но у специалистов по безопасности информационных ресурсов в большей степени озабоченность вызывают действия некомпетентных сотрудников. Это явно показали и ежегодный отчет ФБР «Computer Crime and Security Survey»[1], и отчет «Global Information Security Survey» компании Ernst&Young [2]. Так, по данным этих исследований, ущерб от неосторожных и неправомерных действий сотрудников в несколько раз превышает по объему причиненного вреда ущерб от действия вирусов и т.п. «хакерских» атак.

Построить защиту от «внутреннего врага» — сложная задача, требующая больших усилий. Она складывается из обеспечения безопасности программных приложений корпоративно-информационной системы предприятия и грамотного ее администрирования, которое, прежде всего, подразумевает под собой определение

четких привилегий сотрудников компании на доступ к ресурсам информационной системы (в сформулированном виде это — политика безопасности). Проведенный анализ существующих CMS показал, что ни одна из них не обеспечивает защиту содержимого web-портала от всех возможных угроз безопасности.

Возникает задача — спроектировать автоматизированную универсальную систему безопасности, которая обеспечит защиту как от «внешнего», так и от «внутреннего врага».

Система безопасности CMS

CMS (от англ. Content Management System) — это программная система, которая позволяет управлять текстовым и графическим наполнением web-портала, предоставляя пользователю удобные инструменты хранения и публикации информации [5].

На данный момент на рынке существующих систем CMS представлены продукты с очень слабыми системами безопасности, например, продукты, написанные на языке PHP (Joomla, Drupal, CMS Made Simple, XOOPS), которые в частых случаях уязвимы даже для таких внешних воздействий как SQL-инжекции, не говоря уже о серьезных нежелательных вмешательствах профессионалов. Рассмотрим пример.

Допустим, серверное ПО, получив входной параметр id, использует его для создания SQL-запроса. Рассмотрим следующий PHP-скрипт:

Предыдущий код скрипта...

$id = $_REQUEST['id']; # переменной $id присвоено значение переменной запроса $_REQUEST['id'];

$res = mysql_query("SELECT * FROM news WHERE id_news = $id"); # переменной $res присваивается результат выполнения SQL-запроса к базе данных;

Следующий код скрипта...

Если на сервер передан параметр id, равный 5 (например, так: http://example.org/script.php?id=5), то выполнится следующий SQL-запрос:

SELECT * FROM news WHERE id_news = 5

Но если злоумышленник передаст в качестве параметра id строку –1 OR 1=1 (например, так: — http://example.org/script.php?id=–1+OR+1=1), то выполнится запрос:

SELECT * FROM news WHERE id_news = -1 OR 1=1

Таким образом, изменение входных параметров путем добавления в них конструкций языка SQL вызывает изменение в логике выполнения SQL-запроса (в данном примере вместо новости с заданным идентификатором будут выбраны все имеющиеся в базе новости, поскольку выражение 1=1 всегда истинно) [3].

Как средство повышения безопасности все больше потребителей систем безопасности информационных ресурсов отдают предпочтения средствам разработки, основанным на платформах Java и .Net. Рассмотрим существующие CMS, написанные на Java.

Apache Lenya — система управления содержимым, написанная на Java с использованием стандартов XML и XSLT. Одной из основных компонент является модуль Apache Cocoon, включающий в себя контроль ревизий, механизм поиска,

журнал и WYSIWYG-редакторы.

Система безопасности этой CMS обеспечивает безопасность информационных ресурсов с помощью защищенного канала SSL (Secure Sockets Layer), а также реализацию механизма контроля доступа, который позволяет администратору ограничивать доступ к определенным частям сайта как группам, так и отдельным пользователям. Можно управлять такими функциями как просмотр, редактирование, утверждение и администрирование. Кроме того, используются механизм LDAP авторизации пользователей и контроль доступа по IP-адресу [6]. Но в данной системе отсутствуют элементы тестирования пользователя и самотестирования системы.

OpenCms — это система управления содержимым, которая дает возможность создавать web-сайты и управлять ими без знаний языка программирования и стандарта HTML. Встроенный WYSIWYG-редактор с пользовательским интерфейсом, подобным известным офисным приложениям, дает возможность пользователю создавать контент сайта, а шаблонный механизм формирует и отображает общую разметку сайта.

Функцию безопасности в OpenCms обеспечивает система распределения прав пользователей, а также возможность восстановления удаленных файлов и папок [7]. Недостатком системы является отсутствие таких элементов как создание безопасного канала, поддержка механизма тестирования пользователя, а также формирование журнала событий.

Magnolia — это первая CMS, которая была разработана с нуля для поддержки развивающегося стандарта API хранилища содержимого для Java (JCR) [8]. После первого выпуска в 2004 году Magnolia быстро стала популярным программным обеспечением для решения задачи управления содержимым на Java. Благодаря легкому в использовании интерфейсу и открытым стандартам, таким как JSR-170, она используется в нескольких тысячах инсталляций по всему миру.

Для обеспечения безопасности используется система управления резервными копиями и система контроля прав доступа пользователей и их групп [9]. Безопасный канал для передачи данных, тестирование пользователя и журнал событий в данной системе отсутствуют.

Классификация существующих угроз безопасности

Угрозу безопасности web-ресурсов создают внешние и внутренние атаки. К внешним атакам относятся:

— атаки, которые используют уязвимости в механизмах реализации аутентификации (процесса установления подлинности лица) web-серверов;

— атаки, направленные на авторизацию, т.е. на процесс, который использует web-сервером для предоставления определенному лицу доступа к некоторому содержимому или функциям приложения. С помощью различных техник злоумышленник может повысить свои права и получить доступ к защищенной информации;

— атаки на клиентов. Используя web-ресурс, пользователь, как правило, не ожидает атак со стороны сайта. Такое доверительное отношение злоумышленник может использовать в своих целях и различными методами проводить атаки на

клиентов сервера;

— атаки, приводящие к выполнению кода на web-сервере. При запросе пользователь передает серверу данные, которые применяются для формирования команд. Если во время разработки не учитывать требования безопасности, злоумышленник получает возможность модифицировать исполняемые команды;

— разглашение информации. Получая дополнительную информацию о web-приложении, такую как дистрибутивы программного обеспечения, номера версий клиента и сервера, расположение временных файлов и др., злоумышленник может использовать ее в своих целях;

— логические атаки. Атаки данного вида используют функции и логику самого приложения. Например, восстановление паролей, регистрация учетных записей, транзакции в системах электронной коммерции. Пользователю необходимо выполнить несколько последовательных действий, но злоумышленник может использовать эти механизмы в своих целях.

Внутренние атаки на web-ресурсы производятся инсайдерами (от англ. insider) — авторизованными пользователями, которые имеют доступ к web-сайту. Это может быть служащий, руководитель, бизнес-менеджер, консультант, подрядчик, бизнес-партнер и др. К инсадерским атакам относятся:

— невнимательность, халатность сотрудников;

— действия, приводящие к случайной утечке конфиденциальной информации. Данные могут быть утеряны по электронной почте, через чаты, форумы и другие службы Интернета, с помощью средств мгновенного обмена сообщениями, копирования информации на мобильные носители, а также посредством распечатки ее на принтере;

— неумышленное инфицирование компьютера. Сотрудник может действовать без злого умысла, создавая угрозу по собственной неосторожности. Например, в процессе сетевой игры или посещения того или иного сайта вирус или шпионская программа может инфицировать компьютер;

— умышленное инфицирование компьютера;

— неправомерные действия сотрудников, например, неправомерное отключение оборудования или изменение режимов работы устройств и программ;

— неквалифицированные действия сотрудников;

— намеренные нарушения политики информационной безопасности (уничтожение, блокирование, модификация, удаление информации);

— несанкционированное получение конфиденциальной информации;

— разглашение атрибутов разграничения доступа (паролей, ключей шифрования, идентификационных карточек, пропусков и т.п.). Например, неквалифицированные пользователи записывают процедуры входа-выхода при работе с компьютерной системой, а также пароли доступа, на отдельных бумажных листках;

— кража конфиденциальных данных;

— плохое качество программного обеспечения.

Последствия, которые влекут за собой как внешние, так и внутренние атаки, приводят к следующим угрозам:

— случайной или намеренной утечке конфиденциальной информации;

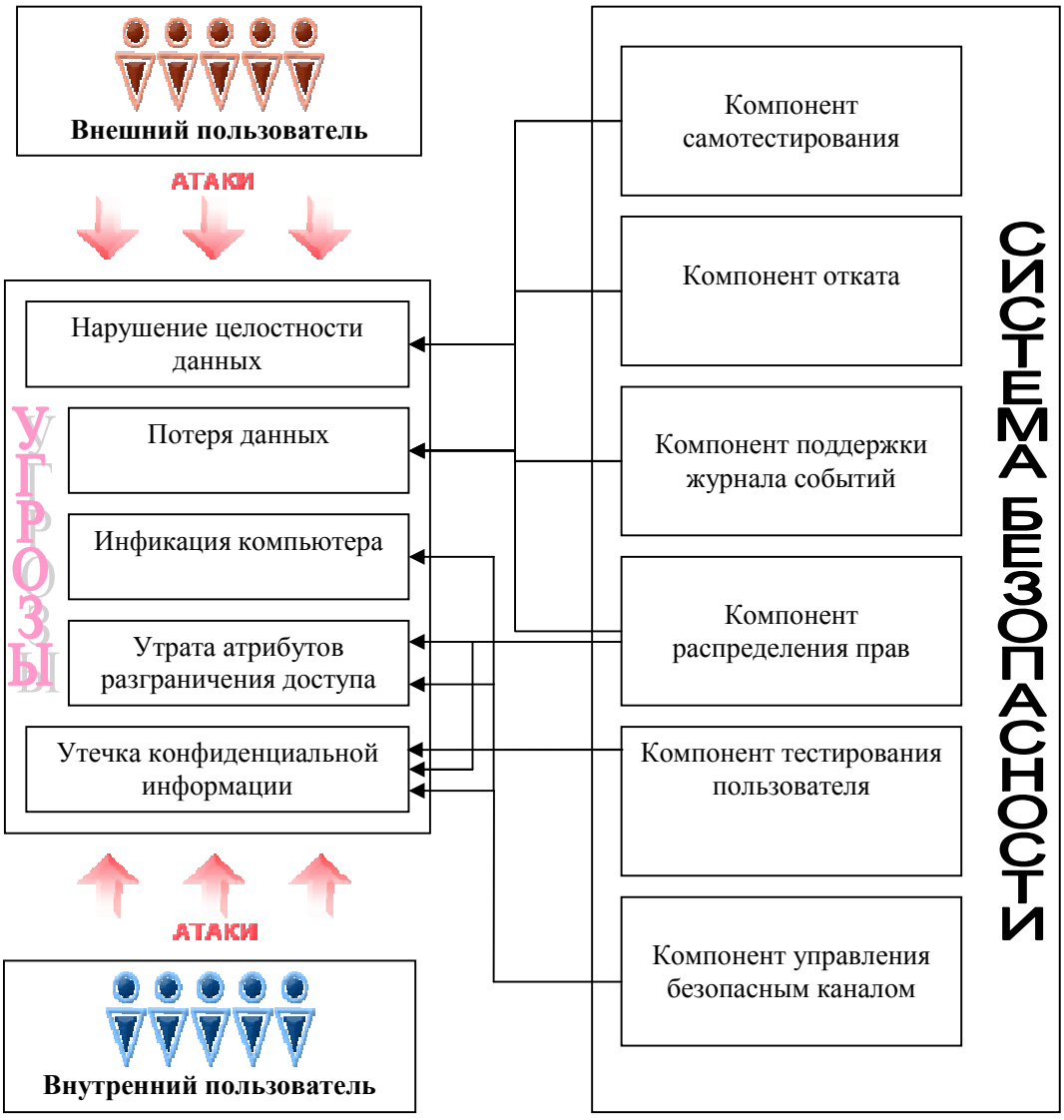
— нарушению целостности данных;

— потере данных (безвозвратной потере данных);

— инфикации компьютера;
— утрате атрибутов разграничения доступа.

Универсальная система безопасности

Чтобы разработать универсальную систему безопасности, необходимо учитывать все возможные угрозы. Система безопасности должна включать в себя компонент распределения прав, компонент самотестирования, компонент тестирования пользователя, компонент поддержки журнала событий, компонент отката, компонент управления безопасным каналом для передачи данных (см. рисунок).



Система безопасности

Компонент распределения прав. Права доступа (к информации) — совокупность правил, регламентирующих порядок и условия доступа субъекта к ин-

формации и ее носителям, установленных правовыми документами или собственником, владельцем информации.

Компонент распределения прав в проектируемой системе отвечает за распределения прав доступа между пользователями комплекса средств защиты. Он обеспечивает реализацию таких функций: создание и редактирование учетных записей пользователей; создание и редактирование групп пользователей; административное управление правами доступа групп или отдельных пользователей к защищенным объектам; административное управление правами доступа групп или пользователей к функциональным режимам работы с объектами. Пользователями системы могут быть: администраторы с неограниченными правами доступа; администраторы системы с ограниченными правами доступа, отвечающие за определенные составляющие системы управления web-порталом; контент-менеджеры и др.

Компонент тестирования пользователя. Для того, чтобы избежать автоматических регистраций и отправления сообщений программами-роботами обычно применяется CAPTCHA (от англ. «Completely Automated Public Turing test to tell Computers and Humans Apart» — полностью автоматизированный публичный тест Тьюринга) — компьютерный тест, используемый для того, чтобы определить, кем является пользователь системы: человеком или программой-роботом. Довольно часто CAPTCHA предлагает распознать искаженную или зашумленную надпись. Идея теста состоит в том, что задача, с которой легко справится человек, для компьютера невозможна или крайне сложна. Могут встречаться и другие задачи, например: определить, что находится на картинке, отметить картинки с живой природой, вычислить арифметическое действие или 3D CAPCHA, базирующейся на трехмерных объектах, которые могут вращаться в разных плоскостях.

Кроме того, возникают ситуации, когда необходимо защитить web-ресурс не от программы-робота, а от пользователей, для которых информация либо сервисы сайта по каким-либо причинам закрыты. Например, не допускать к просмотру содержимого лиц, не достигших определенного возраста, либо автоматически удалять резюме пользователей, которые не соответствуют определенным требованиям.

В этом случае, чтобы вычислить нежелательного пользователя необходимо провести более сложный интеллектуальный или психологический тест. Например, это могут быть вопросы либо загадки, требующие определенного уровня интеллекта, внимательности, логики или находчивости. В проектируемой универсальной системе безопасности спроектирован подобный компонент тестирования пользователя.

Компонент самотестирования. Так как при администрировании web-портала людской фактор всегда будет присутствовать, и полностью исключить его невозможно, с целью обеспечения безопасности информационных ресурсов необходимо предусмотреть систему обнаружения нарушений в работе web-приложения и быстрого восстановления его рабочей версии. Эту задачу решает компонент самотестирования, который осуществляет проверку на наличие нарушений в отдельных составляющих системы управления порталом, а также обеспечивает пользователям интерфейс для инициации процедур самотестирования в комплексе средств безопасности.

Компонент поддержки журнала событий. Все процессы и нарушения в CMS фиксируются в системном журнале событий, чтобы можно было своевременно обнаружить повреждение. Компонент журнала отвечает за отображение состояния системы на текущий момент, а также любой деятельности пользователей портала.

Компонент отката. При фиксации повреждения целостности web-приложения компонент отката системы безопасности запускает механизмы отката, обеспечивая его быстрое восстановление системы до рабочей версии.

Компонент управления безопасным каналом для передачи данных. Организация защищенного канала связи осуществляется поверх стандартных открытых каналов сети Интернет. Безопасность обмена данными обеспечивается средствами криптографии, такими как шифрование, аутоидентификация, инфраструктура публичных ключей и др. Для организации безопасного канала используются различные технологии и протоколы передачи данных. Наиболее широкое распространение сейчас имеет технология под названием SSL и ее современная версия TLS (Transport Layer Security). Компонент управления безопасным каналом отвечает за создание и конфигурирование безопасного канала для передачи данных на базе технологии SSL.

Таким образом, предлагаемый проект системы защиты способствует повышению уровня безопасности портала, при этом обеспечивается противодействие угрозам, возникающим при обнаружении как «внутренних», так и «внешних» атак на информационные ресурсы портала.

1. Computer Crime and Security Survey/ Computer Security Institute (CSI). — San Francisco Federal Bureau of Investigation's (FBI) Computer Intrusion Squad. — Режим доступа: http://www.usdoj.gov/criminal/cybercrime/CSI_FBI.htm

2. Global Information Security Survey/ Paul Van Cessel, Technology and Security Risk Services, Ernst&Young. — Режим доступа: http://www.ey.com/UA/en/Issues/Managing-risk/Information-security-and-privacy/Assurance---Advisory---Technology-and-Security-Risk---Global-Information-Security-Survey-2008

3. Секреты хакеров. Безопасность Web-приложений — готовые решения/ [Джоел Скембрей, Майк Шема, Йен-Минг Чен, Дэвид Вонг]. — М.: Вильямс, 2003. — 384 с.: ил.

4. *Валентин Цирлов.* Основы информационной безопасности / Валентин Цирлов. — М: Феникс, 2008. — 256 с.

5. *Горнаков Сергей.* Осваиваем популярные системы управления сайтом (CMS) / Горнаков С.Г. — М.: ДМК-Пресс, 2009. — 336 с.

6. Apache Lenya — Open Source Content Management (Java/XML) / The Apache Software Foundation. — The Apache Lenya Community. — Режим доступа: http://lenya.apache.org/

7. OpenCms, the Open Source Content Management System (CMS)/ Alkacon Software GmbH. — Режим доступа: http://www.opencms.org/en/

8. JCR — Java Code Reviewer / John Dickson. — Режим доступа: http://jcodereview.sourceforge.net/

9. Products — Magnolia — Simple Open Source Content Management / Magnolia International Ltd. — Режим доступа: http://www.magnolia-cms.com/home/products.html

Поступила в редакцию 11.09.2009