УДК 004.05

К.Г. Колодий

ПРОБЛЕМЫ РУЧНОГО ТЕСТИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ НА НАЛИЧИЕ УЯЗВИМОСТЕЙ БЕЗОПАСНОСТИ

Во многих организация web-приложения используются как критически важные системы, доступ к которым должен предоставляться постоянно. Однако наличие уязвимостей может привести к нарушению работоспособности web-приложения и прерыванию бизнес процессов в организации.

Наличие уязвимостей безопасности в web-приложениях может привести не только к простою организации, но и к крупным финансовым потерям в результате действий злоумышленника. После идентификации уязвимости злоумышленником, для осуществления атаки используется различные методы. Под методами использования уязвимости подразумевают классы атак. Многие из этих классов имеют распространенные названия, например: «внедрение кода SQL» (SQL Injection), и «межсайтовое выполнение сценариев» (Cross-site Scripting) [1].

Покажем проблемы, возникающие при ручном тестировании на примере двух web-приложений. Отделом разработки и сопровождения программного обеспечения Иркутского государственного университета путей сообщения было предоставлено web-приложение «Учет программного обеспечения» и отделом материально-технического обеспечения и связи web-приложение «Справочник ИрГУПС». Web-приложение «Справочник ИрГУПС» содержит контактные данные сотрудников университета, такие как номера внутреннего и внешнего телефонов, адрес электронной почты, номер IP телефона. Web-приложение «Учет программного обеспечения» содержит информацию об имеющихся у университета лицензий на использование программного обеспечения. Рассмотрим данные web-приложения на возможность существования в них наиболее распространенных уязвимостей к следующим видам атак: «внедрение кода SQL» (SQL Injection), «межсайтовое выполнение сценариев» (Cross-site Scripting) и «недостаточная аутентификация» (Insufficient Authentication).

Первый вид атаки, который приходит в голову при посещении главной страницы web-приложения — это поиск возможности для получения доступа к важной информации или функциям сервера без должной аутентификации, т.е. класс атак «недостаточная аутентификация». Web-приложение «Справочник ИрГУПС» — система общего пользования и для доступа к ней аутентификация не требуется, однако, как и большинство систем подобного рода, она может иметь страницу с административным доступом для редактирования данных в справочнике. Попытаемся определить адрес такой страницы. Для этого будем в строке адреса браузера указывать возможные варианты адресов, например: admin.html и т.п. Добавление к строке адреса /admin.php показал стандартную ошибку 404 «Объект не най-

ден», из которой видно, что в качестве web-сервера использован Apache 2. Добавление к адресу /admin/ браузер показал форму для входа в административную часть. Проверка возможности внедрения SQL кода, с помощью указания в качестве имени и пароля одинарной кавычки показало, что внедрить SQL код в данной форме не представляется возможным.

Изучим форму поиска сотрудников в справочнике на главной странице. Форма содержит одно текстовое поле для ввода строки поиска, введем в это поле символ одинарная кавычка. В результате вывелась строка «Вы искали \', по вашему запросу ничего не найдено». Видно, что одинарная кавычка экранируется. Введем в строку поиска часть SQL запроса «Иванов and 1=0» - результат поиска «Вы искали Иванов and 1=0, по вашему запросу ни чего не найдено» говорит о том что, строка поиска воспринимается как текс и на сам запрос ни как не влияет.

В меню web-приложения имеются ссылки для вывода сотрудников соответствующих отделов, в которых передается параметр i_dep. добавление одинарной кавычки результатов не дало и любые попытки внедрения SQL приводили к одному результату: «По вашему запросу ничего не найдено».

Страница /ip/index.php содержит информацию о настройках программного обеспечения для ip телефона, список номеров телефонов, а также на ней имеется страница для просмотра статистики по расчетному периоду. Для доступа к данной информации требуется ввести номер телефона и пароль. Возьмем любой номер из списка номеров. Ввод номера телефона без указания пароля выводит статистику, т.е. любой злоумышленник с любого компьютера внутри организации может посмотреть статистику любого сотрудника: когда и на какие номера совершались звонки и их продолжительность.

Ссылка с главной страницы «Справочник ИрГУПС» на страницу /ip/index.php содержит три переменных, в которых могут передаваться значения, а также в конце страницы показывается ошибка MySQL с частью запроса. Логично предположить, что параметры в адресе могут передаваться в SQL запрос. Исследуем это более подробно.

Строка адреса содержит переменные с именами *a*, *b* и *c*. Добавим к значению переменной *a* в адресе одинарную кавычку. В результате в конце страницы появилась новая ошибка, из которой видно, что одинарная кавычка в переменной *a* не экранируется, а также позволяет предположить, что запрос выбирает из таблицы записи, где одно из полей содержит значения между *a* и *b*.

Проверим, возможно, ли изменить SQL запрос, для этого к значению переменной *a* добавим « and 1=0» в результате запрос должен вернуть пустое множество, так оно и произошло. Проверка переменных *b* и *c* показала, что их можно также использовать для изменения запроса. Добавим к значению переменной *c* « union select null--», чтобы определить количество возвращаемых запросом столбцов. MySQL вернул ошибку о не корректном

использовании UNION и ORDER BY. Тогда добавим значение « union select null--» к значению переменной b. MySQL сообщает о различном количестве возвращаемых столбцов в объединяемых запросах. Теперь будем добавлять null до тех пор, пока запрос не сработает. Успешно отработал запрос с количеством столбцов равным девяти.

Определим название базы данных, имя пользователя под которым происходит обращение к базе данных, а также версию MySQL. Для этого будем заменять null на соответствующие функции MySQL. В последней строке html таблицы показываются интересующие нас данные. Попробуем обратиться к служебным базам данных mysql и information_schema. Доступ к базе данных mysql для пользователя оказался запрещен, а вот доступ к базе данных information_schema разрешен.

Теперь имеется возможность посмотреть, какие базы данных существуют на сервере, а также и их структуру. Дальнейшие действия ограничены только привилегиями пользователя MySQL, так как доступ к служебной базе mysql закрыт, то повысить привилегии не представляется возможным. Изучать операции, которые можно производить с базой данных нет необходимости, уязвимость обнаружена и требует устранения, так как данная уязвимость позволяет просматривать конфиденциальные данные, доступ к которым может разрешить только владелец. Результаты тестирования были переданы разработчикам для скорейшего исправления.

Также в web-приложении обнаружены скрипты, созданные разработчиками при отладке приложения и для обращения к ним пользователей не предназначены. Существует вероятность обнаружения уязвимости в данных скриптах, но поиск их затруднен отсутствием информации о наличии передаваемых в скрипты параметров, а формы ввода на данных страницах отсутствуют.

Перейдем к тестированию web-приложения «Учет программного обеспечения». Для доступа к системе требуется ввести имя и пароль. Введем одинарную кавычку в поля ввода имени и пароля. Сообщение системы: «короткое имя пользователя». Введем любое имя и кавычку в конце имени. Сообщение системы: «короткий пароль». Введем любой пароль с кавычкой в конце. Результат «Некорректный пароль / имя пользователя» Это значит, что происходит обработка одинарных кавычек.

В строке адреса видно, что сайт разработан на PHP. Попробуем ввести в адресе index.php — загрузилась страница входа. Логично предположить, что после входа загружается еще один из основных скриптов с часто используемым именем, например default.php или main.php. Добавление к концу адреса строки /main.php привело к отображению страницы с меню, однако к содержимому страницы доступа нет «У Вас нет доступа к этому ресурсу [/reestr/main.php], либо он временно отключен».

Исследуем все пункты меню. При выборе в меню пункта «Список лицензий» отобразился список лицензий на текущую дату — доступ к кон-

фиденциальным данным без должной аутентификации. Результаты тестирования переданы разработчикам.

Ручное тестирование заняло большое количество времени специалиста по тестированию. При этом приходилось производить подбор различных параметров, таких как адреса страниц, значения переменных и формирование различных запросов к базе данных с анализом ответной реакции web-приложения. Поэтому возникает необходимость автоматизированного тестирования безопасности. Однако тестирование безопасности сложно подается автоматизации, так как программная система автоматизированного тестирования должна не только знать возможные методы атак, но и уметь анализировать реакцию системы.

Проведенное тестирование web-приложений показало, что вопрос обеспечения качества и безопасности web-приложений является актуальным. Для повышения качества и безопасности web-приложений необходимо постоянно проводить мероприятия по выявлению ошибок в приложении и в первую очередь ошибок в системе безопасности. К таким мероприятиям относятся: тестирование, проверка исходного кода (инспектирование), а также организация различных атак на исследуемое приложение. Для повышения эффективности данных мероприятий, предлагается разрабатывать методику проведения тестирования, которая будет учитывать особенности используемых языков программирования, систем управления базами данных исследуемого web-приложения, комбинируя автоматизированные средства [2] для выявления проблемных мест и ручного тестирования для подробного изучения вероятности реализации угрозы через данную уязвимость.

БИБЛИОГРАФИЯ

1. Threat Classification. Официальный сайт Web Application Security Consortium [Электронный ресурс] – http://www.webappsec.org/projects/threat/

2. *К.Г. Колодий* Проблемы анализа и тестирования web-систем // Информационные технологии и проблемы математического моделирования сложных систем. – Иркутск: ИИТМ ИрГУПС, 2008. – Вып. 6 – с. 112 – 118