Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Поиск уязвимостей, позволяющих внедрить в код веб-приложений инъекции

АВТОРЕФЕРАТ

дипломной работы

студенки 6 курса 631 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Калинкиной Полины Алексеевны

Научный руководитель

доцент, к.п.н. _____ А.С. Гераськин

31.12.2016 г.

Заведующий кафедрой

профессор, к.ф.-м.н. _____ В.Н. Салий

31.12.2016 г.

Саратов 2017

ВВЕДЕНИЕ

Деятельность любой организации так или иначе связана с веб-технологиями. Широкое распространение получили веб-порталы различных услуг (в том числе государственных), интернет-магазины, торговые площадки, различные бизнес-приложения, системы дистанционного банковского обслуживания. Для того чтобы максимально использовать преимущества веб-технологий, необходимо обеспечить доступность ресурсов для целевой аудитории, например из сети Интернет. Но доступ, следовательно, могут получить и злоумышленники. Компрометация приложений может привести как к репутационным потерям, так и к финансовым, в том числе в виде упущенной прибыли.

Подавляющее большинство владельцев веб-ресурсов не следуют принципам обеспечения безопасности на всех этапах жизненного цикла приложений (secure software development lifecycle, SSDL), вследствие чего уязвимости не выявляются на ранних стадиях разработки, а остаются в приложениях даже после их приемки в эксплуатацию, что играет на руку злоумышленникам.

Разработка новых веб-приложений часто фокусируется на опыте заказчика и наиболее часто она сводится к обеспечению необходимой функциональности и качественного интерфейса для пользователей. Вопросами обеспечения безопасности в то же время часто пренебрегают или откладывают их решение на последнюю стадию – стадию тестирования приложения.

Целью работы является рассмотрение уязвимотей, которые приводят к инъекциям кода и разработка детектора уязвимостей веб-приложений.

Основными задачами данной работы являются: изучение уязвимостей веб-приложений, рассмотрение уже существующих средств их детектирования и сравнение этих средств, а так же более подробное рассмотрение такого типа

уязвимости как инъекции кода и написание собственного приложения для детектирования веб-уязвимостей.

Дипломная работа состоит из введения, 7 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы – 55 страниц, из них 42 страницы – основное содержание, включая 12 рисунков, список использованных источников из 18 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

В первом разделе дипломной работы представлены общие сведения архитектуры веб-приложений, для понимания потоков информации, которые проходят через веб-приложения и приведены общие определения, которые будут использоваться в дипломной работе.

Тестирование веб-приложений, которое описывается во втором разделе, имея много общего с тестированием классических приложений, содержит свои особенности, связанные прежде всего со средой функционирования. Имея компонентные, структурные и технологические особенности, веб-приложениям присущи особенности режимов работы, инсталляции, запуска, остановки и удаления, а также формирования интерфейсов. Описаны основные различия между тестированием классических и веб-приложений. Так же рассмотрена техника тестирования, которая использовалась при написании программного продукта – тестирование «чёрного ящика».

Поведение приложения определяется не только исходным кодом, но еще и множеством внешних факторов, таких как среда. В третьем разделе рассмотренны различные уровни обработки данных и потенциальные уязвимости, которые чаще всего игнорируются. Уровни обработки информации, которые принимаются во внимание: аппаратные средства, операционная система, браузер, сеть, веб-сервер, фреймворк, приложение, база данных, файловая система.

В четвертом разделе рассмотрены главные уязвимости веб-приложений исходя из классификации векторов атак и уязвимостей сообщества OWASP (Open Web Application Security Project). Рассмотрены следующие виды уязвимостей:

- инъекции (injections);
- недочеты системы аутентификации и хранения сессий (broken authentication and session management);
- межсайтовый скриптинг – XSS (cross site scripting);

- небезопасные прямые ссылки на объекты (insecure direct object references);
- незащищенность критичных данных (sensitive data exposure);
- отсутствие функций контроля доступа (missing function level access control);
- межсайтовая подделка запроса (cross-site request forgery, CSRF/XSRF);
- использование компонентов с известными уязвимостями (using components with known vulnerabilities);
- непроверенные переадресации и пересылки (unvalidated redirects and forwards).

В качестве подтверждения актуальности темы приведены исследования компании Positive Technologies, чьи отчеты представлены на рисунках 1.а, 1.б и 2.

В пятом разделе более подробно разобрана тема инъекций. В первом подпункте данного раздела рассмотрен самый распространенный вид иъекций – SQL-инъекции. Приведен пример и основные причины возникновения уязвимости: динамическое построение SQL-запросов, некорректная обработка исключений, некорректная обработка типов, некорректная обработка специальных символов и небезопасная конфигурация СУБД. Так же выделены следующие техники эксплуатации уязвимости:

- union SQL-инъекция;
- error-based SQL-инъекция;
- blind SQL-инъекция;
- time-based SQL-инъекция;
- out-bound SQL-инъекция.

Во втором и третьем подпунктах рассмотрены Xpath и LDAP инъекции, отличия их эксплуатации от SQL-инъекций.

В шестом разделе произведен сравнительный анализ приложений для сканирования веб-приложений на уязвимости. Сканеры безопасности веб-приложений – это комплексные решения, позволяющие производить поиск дефектов веб-приложений, приводящих к нарушению целостности системных или пользовательских данных, их кражи или получения контроля над системой в целом. В данном разделе проводится тестирование некоторых сканеров безопасности. Приложения, которые были рассмотренны в этом разделе: Acunetix WVS и QualysGuard WAS.

Есть несколько существующих веб-приложений для демонстрации общих уязвимостей веб-приложений, например серия «HacMe» и приложение «Webgoat». Код серии «HacMe» не находится в открытом доступе, что затрудняет оценку степени ложноположительных и ложноотрицательных результатов. Кроме того, эти приложения не реализуют все уязвимости из отчета OWASP TOP-10. Второе приложение используется в основном в учебных целях в связи с его сложной структурой. Из-за такой имплементации это приложение не максимально приближено к воспроизведению сценариев из реальной жизни. Поэтому для этой части дипломной работы было выбрано приложение MusicStore, которое было реализовано с учетом угроз из отчета OWASP TOP-10 и обычных сценариев использования приложений.

На рисунке 5 приведены результаты сравнения Acunetix WVS (A) и QualysGuard WAS (Q). Столбцы таблицы отражают: узвимость в тестовом приложении, тип уязвимости, общее число конкретного типа уявимостей в приложении, общее число уязвимостей, которые были обнаружены сканнерами, количество ложноположительных срабатываний и количество ложноотрицательных срабатываний.

В седьмом разделе описан реализованный программный продукт, который представляет из себя реализацию приложения по детектированию веб-уязвимостей.

Программа разработана на языке Python, принимает на вход ссылку или текстовый файл с набором URL для проверки на уязвимости: удаленное выполнение команд, Blind SQL-инъекции и XSS. На выходе: .xml файл с отчетом по каждой URL. Проверки на конкретный тип уязвимостей сделаны с помощью payloads (добавки) – строки или команды, которые эксплуатируют уязвимость. Для каждой ссылки производится анализ всех ссылок, которые ведут на тот же сайт (crawling).

Так же программа может определить, используется ли технология WebKnight WAF. WebKnight WAF – это межсетевой экран для IIS и других веб-серверов и выпускается под лицензией GNU General Public License. Это ISAPI фильтр, который защищает веб-сервер, блокируя определенные запросы.

В программе были использованы добавки, который не детектируются технологией XSS Auditor, которая используется в браузере Google Chrome (добавки не детектируются в Google Chrome v.50 и ниже). Добавки записаны в отдельных .txt файлах, что позволяет легко их конфигурировать.

Работа завершается следующими приложениями:

1) Приложение А. Листинг программы представляет собой реализацию проверки текущей ссылки на Blind SQL-инъекции;

2) Приложение Б. Листинг программы представляет собой реализацию проверки текущей ссылки на удаленное выполнение команд;

3) Приложение В. Листинг программы представляет собой реализацию проверки текущей ссылки на XSS;

4) Приложение Г. Листинг программы представляет собой реализацию вспомогательного функционала, реализующего генерацию отчета и перебор ссылок.

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены актуальные уязвимости веб-приложений, отчеты ведущих компаний по веб-безопасности, произведено сравнение двух распространенных средств детектирования уязвимостей, а так же более подробно рассмотрены уязвимости типа инъекций кода. Так же были рассмотрены точки входа в приложение недостоверных данных, которые могут привести к инъекциям кода. Была написана программа для детектирования удаленного выполнение команд, Blind SQL-инъекций и XSS, определения использованныя межсетевого экрана WebKnight WAF. Программа предоставляет механизм эксплоита указанных в конфигурационных файлах добавок, но сами добавки не привязанны к программе, что позволяет легко конфигурировать программу (добавлять инъекции и добавки, ориентированные на конкретные технологии).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Проблемы информационной безопасности [электронный ресурс] : научно-популярный, открытый доступ. URL: http://ieu.cfuv.ru/view-file/2372/Сборник трудов II междунар. конфер. Проблемы-информационной безопасности -2016.pdf (дата обращения: 15.09.2016) Загл. с экрана. Яз.рус.

2. OWASP Top 10 [электронный ресурс] : научно-популярный, открытый доступ. URL: https://www.owasp.org/images/0/0f/OWASP_T10_-_2010_rc1.pdf (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.

3. Web app architectures [электронный ресурс] : научно-популярный, открытый доступ. URL: http://www.cs.toronto.edu/~mashiyat/csc309/Lectures/Web%20App%20Architectures.pdf (дата обращения: 20.09.2016) Загл. с экрана. Яз.англ

4. Особенности тестирования web-приложений. [электронный ресурс] URL: http://qaevolution.ru/osobennosti-testirovaniya-web-prilozhenij/ (дата обращения: 25.09.2016) Загл. с экрана. Яз.рус.

5. Open to attack. Vulnerabilities of the Linux Random Number Generator [электронный ресурс] : научно-популярный, открытый доступ. URL: http://qaevolution.ru/osobennosti-testirovaniya-web-prilozhenij/ (дата обращения: 10.10.2016) Загл. с экрана. Яз.англ.

6. Hot or Not: Revealing Hidden Services by their Clock Skew [электронный ресурс] : научно-популярный, открытый доступ. URL: http://sec.cs.ucl.ac.uk/users/smurdoch/papers/ccs06hotornot.pdf (дата обращения: 15.10.2016) Загл. с экрана. Яз.англ

7. CVE-2013-1662 [электронный ресурс] : научно-популярный, открытый доступ. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1662 (дата обращения: 20.10.2016) Загл. с экрана. Яз.англ.

8. Vulnerabilities in data processing levels [электронный ресурс] : научно-популярный, открытый доступ. URL:

http://www.slideshare.net/beched/slides-34960189 (дата обращения: 24.10.2016) Загл. с экрана. Яз.англ.

9. Проблемы информационной безопасности [электронный ресурс] : научно-популярный, открытый доступ. URL: http://ieu.cfuv.ru/view-file/2372/Сборник трудов II междунар. конфер. Проблемы-информационной безопасности -2016.pdf (дата обращения: 15.09.2016) Загл. с экрана. Яз.рус.

10. 2015 Web Application Attack Report [электронный ресурс] : научно-популярный, открытый доступ. URL: https://www.owasp.org/images/0/0f/OWASP_T10_-_2010_rc1.pdf (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.

11. Уязвимости веб-приложений [электронный ресурс] : научно-популярный, открытый доступ. URL: http://www.ptsecurity.ru/upload/ptru/analytics/Web-Vulnerability-2016-rus.pdf (дата обращения: 15.09.2016) Загл. с экрана. Яз.рус.

12. XPATH Injection [электронный ресурс] : научно-популярный, открытый доступ. URL: https://www.owasp.org/index.php/XPATH_Injection (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.

13. Acunetix Web Vulnerability Scanner [электронный ресурс]. URL: http://itstream.net/products/element.php?ELEMENT_ID=391 (дата обращения: 15.09.2016) Загл. с экрана. Яз.рус.

14. The Next Generation Platform for end-to-end Web Application Scanning [электронный ресурс]. URL: https://www.qualys.com/suite/web-application-scanning/#features (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.

15. Hacme Bank v2.0 Released [электронный ресурс]. URL: http://www.mcafee.com/us/downloads/free-tools/hacme-bank.aspx (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.

16. Category:OWASP WebGoat Project [электронный ресурс] : научно-популярный, открытый доступ. URL:

https://www.owasp.org/index.php/Category:OWASP WebGoat_Project (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.

17. vulnerablewebapp [электронный ресурс] URL: https://code.google.com/archive/p/vulnerablewebapp/source (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.

18. Implementation of a Web Application for Evaluation of Web Application Security Scanners [электронный ресурс] : научно-популярный, открытый доступ. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.722.5926 &rep=rep1& type=pdf (дата обращения: 15.09.2016) Загл. с экрана. Яз.англ.