Бажанова С.А., Бобрышева Г.В. Безопасность веб-приложений. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XIX Междунар. научно-техн. конф. – Пенза: ПДЗ, 2019. – С. 070-074.

УДК 004.056.5

# БЕЗОПАСНОСТЬ ВЕБ-ПРИЛОЖЕНИЙ

С.А. Бажанова, Г.В. Бобрышева

# THE SECURITY OF WEB APPLICATIONS

S.A. Bazhanova, G.V. Bobrysheva

**Аннотация.** Статья посвящена вопросам безопасности веб-ресурсов. Рассматриваются виды уязвимостей веб-приложений, а также способы предотвращения атак на веб-ресурсы.

**Ключевые слова:** веб-ресурс, информационная безопасность, угрозы информационной безопасности, способы предотвращения угроз.

**Abstract**. The article is devoted to the security of web resources. We consider the types of vulnerabilities of web applications, as well as ways to prevent attacks on web resources.

**Keywords:** web resource, information security, information security threats, ways to prevent threats.

В настоящее время практически любая организация имеет свой официальный сайт или страничку, которые чаще всего размещают на бесплатном хостинге. Использование бесплатного хостинга расширяет возможности третьих лиц по получению несанкционированного доступа к информационным ресурсам организации, размещенным в веб-приложении, и тем самым существенно усложняет решение проблемы обеспечения его безопасности.

В качестве третьих лиц могут быть как законные пользователи веб-приложения (например, администраторы), так и внешние нарушители.

Внешние нарушители являются основным источником угроз информационной безопасности веб-приложений, в качестве которых часто выступают недобросовестные конкуренты, руководствующиеся преступными намерениями, например, хищение денежных средств, модификация или уничтожение информационных ресурсов или компрометация веб-приложения.

Следствием компрометации веб-приложения являются финансовые потери, в частности, в форме упущенной прибыли, которая может заключаться в потери важного клиента или срыва финансовой сделки. При этом предпосылки к компрометации веб-приложения могут возникнуть, начиная с момента возникновения идеи его создания и заканчивая выходом из употребления. Это объясняется тем, что разработчики не всегда уделяют достаточно внимания вопросам обеспечения информационной безопасности веб-приложения, сосредоточиваясь в первую очередь на его функциональности, а администратор веб-приложения часто недостаточно осведомлен в вопро-

сах защиты информации, в результате чего может совершать ошибки в рамках своих полномочий, которые существенно повышают уровень его уязвимости. Поэтому обеспечение безопасности веб-при-ложений является актуальной задачей на протяжении всего их жизненного цикла.

При разработке веб-приложений данная задача часто решается за счет следования концепции SSDL (Secure software development lifecycle) [5]. Данная концепция охватывает все этапы жизненного цикла веб-при-ложений и обеспечивает поддержание необходимого уровня их информационной безопасности путем нацеливания разработчиков и администраторов на необходимость первоочередного решения вопросов защиты информационных ресурсов, включая идентификацию рисков реализации угроз и управления ими.

На основе результатов аналитического анализа выделены потенциальные угрозы информационной безопасности веб-приложений иосуществлена их классификация, представленная на рисунке.


*Классификация угроз информационной безопасности*

Реализация потенциальных угроз информационной безопасности связана с такими факторами, как уязвимость веб-приложений или их компонентов, а также с использованием слабых механизмов аутентификации пользователей. Эффективность обеспечения информационной безопасности веб-приложений во многом определяется своевременностью и качеством анализа потенциальных угроз, и выбором способа их предотвращения.

В настоящее время многие разработчики и администраторы для анализа потенциальных угроз информационной безопасности используют список уязвимостей открытого проекта обеспечения безопасности веб-приложений «OWASP» (Openweb application security project) [3]. На основе проведенного аналитического анализа источников [1, 2, 4] и списка уязвимостей «OWASP» выявлены наиболее опасные угрозы информационной безопасности веб-приложений и возможные следствия от их проявления и предложены способы предотвращения (таблица).

*Квалификация потенциальных угроз информационной безопасности веб-приложений*

| Название угрозы | Следствия угрозы | Способы предотвращения угрозы |
|---|---|---|
| 1 | 2 | 3 |
| Injection | Выполнение непреднамеренных команд, например, неправомерные SQL, PHP, LDAP запросы и команды ОС | - Проверка данных пользователя на стороне сервера<br>- Использование безопасных API и параметризованных запросов |
| Broken authentication | Сломанная аутентификация | Использование многофакторной аутентификации, изоляции сессии, безопасных файлов cookie |
| Sensitive data exposure | Кража или модификация конфиденциальных данных | - Использование безопасных протоколов и алгоритмов<br>- Отключение кэширования ответов с конфиденциальными данными |
| XML External Entities (XXE) | Выполнение вредоносных задач | - Предотвращение сериализации конфиденциальных данных<br>- Предотвращение загрузки вредоносного XML путем использования подхода белого списка на стороне сервера<br>- Использование WAF для обнаружения и блокировки XXE |
| Broken Access control | Сломанный контроль доступа | - Преобразование токенов и куки-файлов в недействительные после выхода из системы<br>- Осуществление принудительного входа / выходы из системы после смены пароля<br>- Ограничение ресурсов на стороне сервера<br>- Ограничение доступа ко всем ресурсам на основе ролей |
| Security misconfigurations | Несоответствие конфигурации требованиям безопасности | - Изменение настроек по умолчанию<br>- Установка только необходимых функций из фреймворка<br>- Проверка безопасности конфигурации через фиксированные интервалы времени |
| Cross Site Scripting (XSS) | Возникает межсайтовый скриптинг | - Вывод кодировки и экранирование ненадежных символов<br>- Включение Content-Security-policy (CSP) |

Продолжение таблицы

| 1 | 2 | 3 |
|---|---|---|
| Insecure Deserialization | Приложения, которые зависят от клиента, для поддержания состояния, могут допускать вмешательство в сериализованные данные | - Шифрование сериализованных данных<br>- Использование десериализаторов для запуска с наименьшими привилегиями |
| Insufficient logging and monitoring | Неэффективное отслеживание злонамеренных намерений злоумышленников | - Непрерывный мониторинг трафика приложений и анализ логов<br>- Использование эффективных процедур безопасности и процедур реагирования |
| Using Components with known vulnerabilities | Нарушение безопасности или захват сервера | - Регулярное исправление нарушений<br>- Регулярное отслеживание информации по новым уязвимостям и методам их предотвращения / исправления<br>- Проверка устройств / программного обеспечения на уязвимость и их предотвращение |

Для того чтобы защититься от потенциальных угроз, необходимо соблюдать следующие рекомендации:

1) использовать протокол https (Hyper Text Transfer Protocol Secure), поддерживающий шифрование и защиту данных пользователей при передаче [6];

2) регулярно обновлять программное обеспечение;

3) использовать и обновлять инструменты для анализа безопасности приложения;

4) использовать максимально длинные пароли, построенные на основе разных алфавитов, и их шифрование;

5) осуществлять проверку данных, полученных со стороны клиента;

6) использовать механизмы распределения прав доступа к информационным ресурсам;

7) использовать файлы Cookie;

8) не хранить строки подключения к базам данных и ключи к сервисам в открытом виде.

Выделенные виды потенциальных угроз информационной безопасности могут быть использованы при разработке тестов для тестирования веб-приложений на уязвимость.

Библиографический список

1. Уязвимости веб-приложений. URL: https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/Web-Vulnerability-2016-rus.pdf (дата обращения: 01.09.19).

2. Список уязвимостей 2017 года // Официальный сайт проекта «OWASP». URL: https://www.owasp.org/index.php/Top_10-2017_Top_10 (дата обращения: 10.09.19).

3. HarpreetPassi. OWASP-Top 10 Vulnerabilities in web applications (updated for 2018). URL: https://www.greycampus.com/blog/information-security/owasp-top-vulnerabilities-in-web-applications (дата обращения: 07.04.19).

4. Поставщик сервисов для обеспечения безопасности приложений // Официальный сайт Positive Technologies. URL: https://www.ptsecurity.com/ww-en/services/sdl/ (дата обращения: 11.09.19).

5. Как защитить веб-приложение. URL: https://tproger.ru/translations/webapp-security/ (дата обращения: 07.09.19).

**Бажанова Светлана Андреевна**
Пензенский государственный университет, г. Пенза, Россия

**Bazhanova S.A.**
Penza State University, Penza, Russia

**Бобрышева Галина Владимировна**
Пензенский государственный университет, г. Пенза, Россия

**Bobrysheva G.V.**
Penza State University, Penza, Russia