УДК 159.937:004.51

Студ. А. А. Иванова, К. А. Крайнов

Науч. рук. проф. Урбанович П. П  
(кафедра информационных систем и технологий БГТУ)

**АНАЛИЗ ЦЕЛЕСООБРАЗНОСТИ И ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ В WEB-ПРИЛОЖЕНИЯХ**

Современный мир несет в себе тысячи угроз и потенциальных опасностей в каждый момент времени.  Мы считаем, что недостаточно знать теорию для понимания масштаба угрозы – ведь уязвимости любого уровня, могут нести катастрофические для бизнеса последствия. Классификацией атак и уязвимостей занимается сообщество [OWASP](https://www.owasp.org/). Эта организация создала список из 10-и самых опасных векторов атак на Web-приложения. Эти уязвимости будут рассмотрены в данной статье.

**Инъекции.** Все данные, хранятся в базах данных. При недостаточной проверке данных от пользователя, злоумышленник может внедрить в форму Web-интерфейса приложения специальный код, содержащий кусок SQL-запроса. Это уязвимость, позволяющая злоумышленнику получить доступ к данным, которые для него не предназначены. Эта уязвимость является следствием недостаточной проверки данных, поступающих от пользователя.

**Недочеты системы аутентификации и хранения сессий**. Для того, чтобы отличать одного пользователя от другого, web-приложение использует так называемые сессионные куки. В случае, если ваш идентификатор украдет злоумышленник, а в системе не были реализованы проверки IP-адреса сессии или проверки наличия более одного соединения в одной сессии, злоумышленник сможет получить доступ в систему с правами вашего аккаунта.

**Межсайтовый скриптинг – XSS**. Это еще одна ошибка валидации пользовательских данных, которая позволяет передать JavaScript код на исполнение в браузер пользователя. Во-первых, злоумышленник может украсть вашу сессионную cookie. Во-вторых, могут быть украдены конфиденциальные данные, вводимые в формы на зараженной странице. В-третьих, через JavaScript можно изменять данные, расположенные на странице.

**Небезопасные прямые ссылки на объекты.** Суть атаки заключается в том, что при выводе каких-либо конфиденциальных данных, для доступа к объекту используется идентификатор, который передается в открытом виде в адресной строке браузера. Перебирая число после "id=" можно будет читать чужие личные сообщения. Этим способом были подвержены достаточно крупные европейские платежные системы.

**Небезопасная конфигурация**. Безопасность Web-приложения требует наличия безопасной конфигурации всех компонентов инфраструктуры. Например, кража сессионной cookie через JavaScript при XSS-атаке становится возможна благодаря выключенной по умолчанию настройке cookie\_http only. Пренебрежение использования базовых правил безопасности серверов, позволяет злоумышленнику запросто читать и изменять данные. Кроме того, программное обеспечение должно быть в актуальном состоянии. Даже если приложение правильно написано и тщательно проверяет все входящие данные и имеет хорошую защиту, это не означает что в один прекрасный момент не найдется уязвимость в ОС или Web-сервере.

**Незащищенность критичных данных.** Многие веб-приложения не защищают конфиденциальные данные. Самый простой пример – передача данных по протоколу HTTP. Такие данные должны передаваться исключительно по протоколу HTTPS. Протокол HTTPS требует наличие SSL-сертификата, задача которого – подтвердить, что он выдан именно для данного сайта. Другой пример – отсутствие шифрования критичных данных. В случае, если данные зашифрованы, то даже в случае получения несанкционированного доступа на сервер, злоумышленник не сможет украсть критичные данные.

**Отсутствие функций контроля доступа**. Суть уязвимости заключается в отсутствии проверки наличия надлежащего доступа к запрашиваемому объекту. Приложения должны выполнять проверки контроля доступа на сервере при запросе любой функции.

**Межсайтовая подделка запроса (CSRF/XSRF).** Этотвектор атаки, позволяющий злоумышленнику выполнять от имени жертвы действия на сервере. Если жертва заходит на сайт, созданный злоумышленником, от её лица тайно отправляется запрос на вышеуказанную страницу платежной системы. В итоге деньги уйдут на счет злоумышленника. Предполагается, что жертва должна была предварительно пройти аутентификацию в платежной системе.

**Использование компонентов с известными уязвимостями**. Зачастую web-приложения написаны с использованием специальных библиотек или «фреймворков», которые поставляются сторонними компаниями. Уязвимости ищут в более низкоуровневых компонентах системы, таких как сервер базы данных, web-сервер, и наконец, компоненты операционной системы вплоть до ее ядра.

**Непроверенные переадресации и пересылки**.Web-приложения зачастую переадресуют пользователя с одной страницы на другую. Без соответствующих проверок, атакующий может использовать такие страницы для переадресации жертвы на подложный сайт, который может иметь очень схожий или неотличимый интерфейс, это может привести к краже конфиденциальных данных.

Мы рассмотрели основные виды уязвимостей. В ходе исследования мы выяснили что web-приложение может быть атаковано с любой стороны, что в свою очередь показывает, что даже самое простое web-приложение необходимо защищать хотя-бы самыми базовыми способами, что может минимизировать количество взломов и утечек данных из приложения.

ЛИТЕРАТУРА

1. Эндрю Х. Безопасность веб-приложений, - Москва, изд. Прогресс книга 2021. – [102 – 330] с.