АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ БЕЗОПАСНОСТИ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ

Хильчук А.С., Куликов С.С.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь, khilchuk.alexei@gmail.com, kulikov@bsuir.by

Abstract. The application testing of security, its main approaches, the most frequent types of attacks on Internet resources, the improvement of the process of security testing by introducing its automation are considered.

На данный момент в сфере образования используется множество программных продуктов. При создании подобных решений следует уделить особое внимание вопросу их защищённости от информационных атак. Для минимизации возможности появления таких уязвимостей внедряется процесс тестирования, в частности тестирования безопасности.

Тестирование безопасности – тестирование, направленное на проверку способности приложения противостоять злонамеренным попыткам получения доступа к данным или функциям, права на доступ к которым у злоумышленника нет [1]. Подход к такому виду тестирования базируется на принципах конфиденциальности, целостности и доступности.

Под конфиденциальностью подразумевается ограничение доступа к тому или иному ресурсу программного средства определённым пользователям ПО. Целостность означает ожидание того, что ресурс изменяется должным способом и ожидаемыми пользователями программного средства, а также определение критичности восстановления ресурса приложения при его повреждении. Под доступностью понимается ограничение доступа ресурса пользователям в зависимости от его критичности. Причинами появления уязвимостей может быть обновление компонентов системы, рефакторинг исходного кода, а также изменение инфраструктуры ключевых компонентов программного средства.

Одними из наиболее частых видов атак являются следующие.

1. Межсайтовый скриптинг (XSS) – внедрение вредоносного кода на сгенерированную сервером страницу. Например, код может быть заключен в HTML тэги «<script>» или «<object>», который осуществляет доступ к файлам cookie, либо производит перенаправление на сторонний сайт злоумышленника.

2. Межсайтовая подделка запроса (CSRF) – осуществление перенаправления пользователя на вредоносный сайт с помощью уязвимостей протокола HTTP. В основном для осуществления такого вида взлома используются HTML тэги «img», либо JavaScript объекты Image, источник в которых указан вредоносный сайт.

3. Инъекция кода – возможность запуска вредоносного исполняемого кода из приложения для доступа и/или повреждения ресурсов приложения. Если у приложения отсутствует валидация данных в полях ввода, то обращение к внутренним ресурсам приложение может быть выполнен введенным в данные поля SQL, PHP либо каким-нибудь другим вредоносным кодом.

4. Инъекции серверных команд (SSI) – уязвимости, в случае эксплуатации которых осуществляется выполнение команд на серверной стороне приложения путём их внедрения на клиентской стороне либо сразу на сервер. Команды зависят от операционной системы, на которой работает сервер. Например, для UNIX систем, команда «rm*.db» удаляет все файлы с расширением «db».

5. Обход авторизации – злоумышленный доступ к персональным данным других пользователей программного средства. Если идентификационные данные пользователя находятся в строке GET запроса, то доступ к такой учетной записи крайне прост.

Для проведения проверок на вышеупомянутые и другие виды уязвимостей вручную уходит существенное количество времени. Для снижения временных затрат на ручное выполнение проверок, а также для определения регрессии в уязвимых точках безопасности на проект внедряется процесс автоматизации тестирования безопасности.

С его помощью возможно осуществлять инъекции специально подготовленного вредоносного кода во все поля ввода приложения, встраивание скрипта-взломщика в код HTML страницы и прочие проверки безопасности в автоматическом режиме. Также к плюсам такого подхода можно отнести автоматическую генерацию отчёта о проверке. К минусам можно отнести трудозатраты по созданию соответствующего фреймворка, а также высокие требования к квалификации разработчиков.

Помимо фреймворков по автоматизированному тестированию, которые создают сами разработчики приложений, существуют решения сторонних компаний. Одними из наиболее популярных являются «SoapUI» компании «SmartBear Software», «XSpider» (компания разработчик «Positive Technologies»), «Acunetix Web Vulnerability Scanner» и другие. Для осуществления (полу)ручного тестирования безопасности используются приложения «Firebug», «Charles», «WinDump» и другие аналоги.

Таким образом, для образовательных ресурсов, будь это электронные библиотеки, системы дистанционного обучения или системы проведения онлайн-олимпиад, очень важна устойчивость к различным видам атак, так как возникновение регрессии в данных областях приложений является критическим для программ данных направлений.

Литература

1. Куликов, С.С. Тестирование программного обеспечения. Базовый курс. 2-е издание // С.С. Куликов // «Четыре четверти», Минск, 2017. – 312 с.