С.С. Долгачёв (ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **П.Л. Чечет,** канд. техн. наук, доцент

РЕАЛИЗАЦИЯ ЗАЩИТЫ ОТ МЕЖСАЙТОВОЙ ПОДДЕЛКИ ЗАПРОСА

Защита от межсайтовой подделки запроса была реализована с помощью таких средств, как Html, Java SE, Java EE и Framework Struts версии 1.3. В рамках реализации данной защиты был выбран метод защиты, который называется Synchronizer Token. Данный метод предполагает хранения защитного токена на стороне сервера. В процессе разработки защиты был создан JSP тег, а также валидационный метод. Логика тега достаточно проста. Когда пользователь первый раз проходит аутентификацию для своего профиля, тег проверяет объект HttpSession на наличие, там сгенерированного токена и, если токен не был найден, тогда тег генерирует защитный токен и помещает его в объект HttpSession. Далее тег достаёт сгенерированный защитный токен из объекта HttpSession и помещает значение защитного токена в скрытое поле Html. В случае, если значение уже было сгенерировано, тег просто достаёт сгенерированный защитный токен из объекта HttpSession и помещает значение защитного токена в скрытое поле Html. Тег обязательно должен быть описан в специальном файле с расширением *.tld, как этого требует спецификация Java EE. В случае, если у приложения одна точка входа в профиль пользователя, логику с генерацией защитного токена можно вынести в тот участок кода, где осуществляется аутентификация. Но в данном случае большой плюс, что тег ответственен за логику генерации защитного токена так, как приложение часто изменяются и функциональность растёт, то можно упустить из вида тот факт, что точка входа в профиль пользователя должна быть одна.

Валидационный метод интегрирован в функциональность Framework Struts. Он проверяет защитный токен полученный из http запроса и сравнивает его с защитным токеном, который хранится непосредственно на стороне сервера в объекте сессии пользователя, а именно в объекте HttpSession. Если токен из http запроса и токен из пользовательской сессии не совпадут, тогда пользователю будет возвращено сообщение об ошибке, в противном случае управление будет передано логике, ответственной за выполнения http запроса.

Существует много проектов написанных с использованием более ранних технологий, которые на тот момент не предоставляли подобной функциональности по умолчанию. Создание собственной защиты от межсайтовой подделки запроса для приложений подобного типа един-

ственный выход из данной ситуации. Конечно, можно рассмотреть вопрос миграции на более новые технологии но, когда проект существует более 5 лет и написано много кода, который привязан к конкретному фреймворку, ни один заказчик не согласиться платить за миграцию из-за причины отсутствия защиты от межсайтовой подделки запроса по умолчанию для данного фреймворка. Защита от межсайтовой подделки запроса, делает веб-приложение более надежным и об атаке подобного типа нужно помнить и в наши дни. Многие современные фреймворки (на пример Spring версии 4 или Django) предоставляют подобную защиту по умолчанию и это показывает важность и актуальность безопасности веб-приложений и самой защиты от межсайтовой подделки запроса в наши дни.

С.С. Долгачёв (ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **П.Л. Чечет,** канд. техн. наук, доцент

ЗАДАЧИ РАЗРАБОТКИ ПРОЕКТА ЗАЩИТЫ ОТ МЕЖСАЙТОВОЙ ПОДДЕЛКИ ЗАПРОСА ДЛЯ ИООО «ЭПАМ СИСТЕМЗ»

В представленной работе осуществляется разработка защиты от межсайтовой подделки запроса веб-сайта для ИООО "ЭПАМ СИСТЕМЗ".

Защита от межсайтовой подделки запроса защищает профиль пользователя и его конфиденциальную информацию от злонамеренных действий хакеров, которые используют данный тип атаки для изменения важных данных профиля пользователя в своих корыстных целях одна из которых, это получения полного контроля над профилем пользователя. Разработанная защита от межсайтовой подделки запроса в ходе выполнения данной работы является достаточно гибкой. Это означает, что её можно легко настроить и подключить или отключить при необходимости для любой веб-страницы в рамках одного веб-приложения. Так как ИООО "ЭПАМ СИСТЕМЗ" имеет много заказчиков в рамках одного веб-приложения так же реализована функциональность, которая позволяет включить или исключить возможность использования данной защитой, в случае неоплаты данной функциональности заказчиком. Данная защита от межсайтовой подделки запроса актуальна для более старых проектов, которые используют не самые новые технологии и даже в том случае, если более новый фреймворк предоставляет такую функциональность нет гарантий что она будет такой же гибкой и легко настраиваемой для отдельных частей веб-приложения.