Московский Государственный Университет имени М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Автоматизации Систем Вычислительных Комплексов

Чиботару Виктор Дорианович

Исследование способов эксплуатации недостатков веб-приложений, связанных с возможностью изменения области видимости переменных объектной модели веб-страницы

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
младший научный сотрудник
А.А. Петухов

Москва, 2016

Аннотация

В данной работе рассматривается метод автоматического анализа веб-приложений с целью выявления способов эксплуатации уязвимостей типа DOM Clobbering, связанных с возможностью изменения области видимости переменных объектной модели веб-страницы. Идея метода состоит в отслеживании и дальнейшем анализе потоков данных в коде веб-страницы, исполняемого на стороне клиента. В работе приводится обзор существующих методов анализа кода, написанного на языке JavaScript и предлагается использование метода тейнтирования (taint analysis). Предложенные идеи реализованы в виде дополнения к браузеру Mozilla Firefox. Для проверки корректности его работы было проведено тестирование как на синтетических наборах тестов, так и на реальных веб-приложениях.

Оглавление

1 Введение

В период зарождения всемирной паутины веб-страницы представляли собой статические документы, сверстанные на языке HMTL. Но, с течением времени, страницы становились все более и более динамическими. Сегодняшние веб-приложения обычно содержат большое количество кода, исполняемого на стороне клиента (в веб-браузере) и отвечающего за первичную валидацию данных, введенных пользователем, модификацию страницы без взаимодействия с веб-сервером и т. д. Стоит отметить, что несмотря на существование конкурентов, язык JavaScript используется в подавляющем большинстве веб-приложений для написания кода для клиентских частей. Об этом факте свидетельствует бурный рост популярности данного языка и технологий, которые его используют (например, NodeJS, AngularJS, AJAX и т. д.) Активное использование языка JavaScript привело к появлению новых угроз для веб-приложений. Самой распространенной из них является уязвимость межсайтового скриптинга (Cross Side Scripting или XSS). В рейтинге уязвимостей веб-приложений OWASP (Open Web Application Security Project) Top Ten XSS занимает третье место. Недостатки типа XSS делятся на три типа:

Отраженные (reflected XSS) Хранимые (stored XSS) DOM-based XSS

Общей чертой всех трех типов XSS является возможность исполнения злоумышленником произвольного кода в веб-браузере жертвы. С помощью такой уязвимости, преступник может, например, украсть конфиденциальные данные о банковском счете клиента. Первые два типа недостатков возникают из-за недостаточно тщательной фильтрации данных на стороне сервера. Например, веб-сервер генерирует ответ на запрос пользователя, вставляя в страницу необработанные данные. Уязвимости третьего типа отличаются тем, что вся работа производится только на стороне клиента. То есть, код веб-страницы сам обрабатывает пользовательские данные и модифицирует её содержимое. Этот тип уязвимостей был открыт позже остальных, но, несмотря на это, он достаточно хорошо исследован и для него были разработаны эффективные автоматические средства обнаружения. В виду некоторых особенностей языка JavaScript (слабая типизация, возможность динамического исполнения кода) статические методы анализа кода оказываются непригодными для решения подобных проблем. Поэтому, разработанные решения опираются на такие методы, как динамический тейнт анализ и фаззинг. Уязвимость, рассматриваемая в данной работе носит название DOM Clobbering (от англ. DOM - Document Object Model, Объектная Модель Документа и Clobber - перезаписывать). Суть DOM Clobbering заключается в возможности подмены объектов (переменных) веб-страницы, с помощью изменения их области видимости. Как и DOM-based XSS, DOM Clobbering - это уязвимость целиком на стороне клиента. Стоит отметить, что в реальных сайтах DOM Clobbering встречается редко, что и является причиной её недостаточной изученности. Однако, высокая степень опасности, которую таят в себе уязвимости данного типа, указывает на необходимость разработки автоматизированного средства их поиска.