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 приложения обычно содержат большое количество кода, исполняемого на стороне клиента (в веб-браузере). 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 новой парадигмы создания веб-приложений, называемой Single Page Application (Одностраничное приложение). Используя такие приложения, пользователь все время находится в рамках одной веб-страницы, в коде которой и реализована большая часть бизнес логики. Всё общение между клиентом и веб-сервером осуществляется с помощью AJAX запросов без необходимости перезагрузки или перехода на другие страницы.

Высокая популярность Single Page Application наглядно демонстрирует тенденцию к всё большему усложнению кода клиентской части. Однако, хорошо известно, что рост сложности системы часто ведёт к снижению её безопасности. В случае веб-приложений этот факт доказывается большим количеством недостатков и уязвимостей на стороне клиента. Вот самые опасные и распространенные из них:

1. DOM Based XSS
2. DOM Redirection
3. Некорректное использование механизмов Same Origin Policy
4. DOM Clobbering

Самой распространенной из них является уязвимость DOM Based XSS. В рейтинге уязвимостей веб-приложений OWASP (Open Web Application Security Project) Top Ten она занимает третье место. Она заключается в том, что код веб-страницы обрабатывает пользовательские данные и модифицирует её содержимое, позволяя злоумышленнику исполнять произвольные команды. Этот тип уязвимостей 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.