Классификация запросов к серверу в задаче автоматического обхода современных веб-приложений

Лапкина Анна Вадимовна

Студент

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: amiriya@seclab.cs.msu.ru

Научный руководитель *— Гамаюнов Денис Юрьевич, Петухов Андрей Александрович*

Задача классификации запросов от веб-клиента к веб-приложению и соотнесение классов с функциями веб-приложения наиболее часто возникает при анализе приложений методом черного ящика. В случае автоматизированного анализа веб-приложений на первом этапе обычно решается задача сбора информации о нем: о его структуре, функциях, входных параметрах и типах запросов [2]. Для сбора этой информации требуется решить задачу навигации по веб-интерфейсу — в автоматизированном режиме находить управляющие элементы (ссылки, формы и т.п.) и активировать их.

В традиционных многостраничных веб-приложениях пользователь взаимодействовал с серверной частью посредством перехода по ссылкам и отправки данных веб-форм. При этом функции приложения можно было однозначно идентифицировать по URL исходящих запросов, и их определение для каждой страницы сводилось к выделению ссылок и форм через разбор HTML-документа.

В современных веб-приложениях используется понятие маршрутизации запросов: при создании приложения разработчик задает таблицу с предикатами над HTTP-запросами и названиями функций, которые будут вызываться для обработки запроса при выполнении условия. При этом URL-часть запросов может совсем не участвовать в маршрутизации запросов, а выбор вызываемой функции осуществляться в зависимости от параметров запроса [1].

Таким образом на стороне веб-сервера классификация и маршрутизация запросов происходит на основе признаков, полученных из HTTP-запросов. Так как таблица маршрутизации запросов на стороне сервера неизвестна при анализе приложения методом черного ящика, актуальна задача по восстановлению этой таблицы на основе анализа свойств приложения.

В рамках работы было сделано предположение о возможности построения классификатора запросов к веб-приложению, возника-

ющих при работе с пользовательским интерфейсом, используя не только содержимое запросов, но и дополнительные признаки, полученные из контекста их выполнения в веб-клиенте. Контекстом возникновения или трассой запроса будем называть совокупность дополнительных идентификационных данных, которые возможно получить, наблюдая за исполнением JavaScript-кода на веб-странице или за состоянием объектной модели документа в результате активации элементов интерфейса. Для получения контекста исходящих запросов в рамках работы использовалась реализация протокола Chrome DevTools: данные о выполнении запроса собирались на уровне (англ. engine) браузера [3].

Отбор значимых признаков контекста возникновения HTTP-запросов для последующего включения в классификатор был проведен посредством ручной разметки элементов контекстов на выборке из 20-ти сайтов. На основе анализа частоты встречаемости элементов контекста и их типов были получены наборы признаков, в дальнейшем используемых для классификации запросов к серверу в дополнение к параметрам запроса.

В результате исследования на основе предложенного подхода был разработан инструмент, позволяющий выделять признаки из HTTP-запросов и браузерного контекста и производить их классификацию в ручном и автоматизированном режимах.

Анализ результатов экспериментального исследования применимости подхода и инструмента, использующего семь основных признаков для алгоритмической классификации запросов, на выборке из 100 сайтов подтвердил применимость предложенного метода для решения задачи классификации запросов к серверу при автоматическом обходе веб-приложений. При этом удалось достигнуть 100%-ой точности при классификации запросов к различным функциям и избежать рассмотрения незначащих признаков, используемых при классификации запросов без учета контекста.

Литература

1. Носеевич Г. М., Петухов А. А. Поиск входных точек для веб-приложений с динамическим пользовательским интерфейсом // Безопасность информационных технологий 2013. Т. 20
2. Pandikumar T., Eshetu T. Detecting web application vulnerability using dynamic analysis with penetration testing // International Research Journal of Engineering and Technology 2016, Т. 3, № 10.
3. Протокол инструментария разработчика Chrome DevTools : `https://chromedevtools.github.io/devtools-protocol`