Список литературы

1. Gallager R.G. Low Density Parity-Check Codes. MIT Press, Cambridge, MA, 1963.
2. Jian-Bing H.A.N., Chen H.E., He Yun H.E. Research on regular LDPC codes with better performance than turbo codes // Materials of International Conference on Information Engineering ICIE '09.

ИНВАРИАНТНОСТЬ ЦИФРОВОГО ОТПЕЧАТКА УСТРОЙСТВА ПОЛЬЗОВАТЕЛЯ, ИСПОЛЬЗУЕМОГО ДЛЯ ЕГО ДОСТУПА К ИНФОРМАЦИОННЫМ РЕСУРСАМ

О.А. Хожевец, Т.В. Борботько

Идентификация пользователя информационных ресурсов предполагает использование некоторой уникальной информации, которая известна непосредственно пользователю, а также хранится в информационной системе для выполнения процедуры его аутентификации. В качестве такой информации выступает, как правило, некоторые сведения (логин, пароль и т.д.) вводимые пользователем для проверки его подлинности.

Учитывая особенности программных средств, применяемых пользователями сети интернет для доступа к информационным ресурсам, можно выделить следующие признаки таких средств, позволяющих выполнить процедуру идентификации его устройства: javascript, user agent, IP, flash, ActiveX, содержание кэша браузера, cookie, supercookie, настройки используемого программного обеспечения. Формирование из вышеперечисленных признаков массива данных и вычисление его хэш суммы позволяет получить цифровой отпечаток (фингерпринт) устройства пользователя, который будет неизменным в течение достаточно длительного времени.

Необходимо отметить, что во всех существующих браузерах есть широкий спектр переменных с большой инвариантностью, которые позволяют, даже не подготовленному пользователю, изменять цифровой отпечаток своего устройства перед подключением к информационному ресурсу. Наиболее простыми способами, являются: использование нового браузера или новой версии, изменение масштаба окна браузера, изменение часового пояса устройства, изменение языка браузера. Экспериментально установлено, что варьируя значениями указанных переменных можно получить от 1000 до 230000 уникальных цифровых отпечатков устройства, в зависимости от используемого браузера. Таким образом, для снижения ошибок первого и второго рода необходимо формировать базу цифровых отпечатков устройства пользователя, которая, в дальнейшем, может быть использована для его идентификации.

БЕЗОПАСНОСТЬ REACT-ПРИЛОЖЕНИЙ

М.П. Хоронеко, Н.В. Харитонов, М.А. Медунецкий, В.Я. Анисимов

ReactJS – самая популярная JavaScript библиотека для построения пользовательских интерфейсов по данным Google Trends на 2019 год [1]. Приложения на ее основе используют большое количество js-кода, поэтому справедливо предположить что атаки типа XSS могут принести злоумышленникам определенный результат.

Практика использования ReactJS в мире показывает, что библиотека содержит большое число компонентов поддержания безопасности приложения. Например, спецсимволы, без которых невозможно осуществить XSS атаку, автоматически заменяются управляющей последовательностью при их использовании в строчных значениях в JSX (синтаксис, подобный HTML, в основе которого лежат теги, позволяющий использовать JS-код непосредственно при построении разметки) [2]. Тем не менее, проблемы, связанные с внедрением скриптов, могут быть результатом использования неподходящих практик программирования и не всегда являются очевидными. Среди них:

– создание React-компонентов из объектов, поставляемых пользователем;
– отображение ссылок с href-атрибутом, определяемым пользователем;
– явное задание свойства dangerouslySetInnerHTML у элемента;