МОДУЛЬ СБОРА ПОЛЬЗОВАТЕЛЬСКИХ СОБЫТИЙ ДЛЯ ИССЛЕДОВАНИЯ ПОВЕДЕНЧЕСКОЙ АВТОРИЗАЦИИ В ВЕБ-ПРИЛОЖЕНИЯХ

А.Т. Газизов
Томский политехнический университет
atg1@tpu.ru

Введение

С повсеместным развитием веб-приложений, оперирующих с ценной пользовательской информацией (например, приложения онлайн-банкинга), улучшение качества аутентификации становится все более актуальным. Для этого научным сообществом предлагается использование методов биометрической авторизации [1-2]. Поведенческая биометрическая авторизация, основанная на анализе данных с клавиатуры и мыши, является довольно многообещающей, поскольку не требует специального оборудования и может проводится неявно для пользователя [3-6]. В отличие от *непрерывной* поведенческой авторизации, *статичная* обладает лучшей точностью и не требует времени для срабатывания, но требует регистрации предопределенных пользовательских сценариев [8-10]. Обычно применение статичной поведенческой авторизации рассматривают на примере сценария заполнения формы авторизации, однако она может быть использована и после входа в систему, например, при заполнении формы отправки платежа или при переключении кнопок навигации в приложении онлайн-банкинга. В данной работе разрабатывается модуль сбора пользовательских событий для исследования статичной поведенческой авторизации в веб-приложениях.

Модуль сбора пользовательских событий

Модуль предназначен для сбора UI-событий в рамках определенных пользовательских сценариев. Данный модуль представляет собой JavaScript-библиотеку, подключаемую на HTML-страницу веб-приложения. Библиотека разработана с использованием возможностей стандарта ECMAScript 2015, позволяющего использовать синтаксис классов. Диаграмма классов библиотеки представлена на Рис. 1 (используется синтаксис TypeScript). Клиентской части веб-приложения предоставляется класс *ImplicitBioStaticRecorder,* объекты которого соответствуют определенному пользовательскому сценарию. В конструктор *ImplicitBioStaticRecorder* передается объект с полем *elements*, содержащим массив объектов-описаний элементов графического интерфейса, участвующих в сценарии: в данном случае (Листинг 1), это поля ввода для логина и пароля, а также кнопка отправки формы. В каждом объекте-описании указывается уникальное имя элемента, его тип (поле ввода или кнопка), а также CSS-идентификатор, позволяющий найти этот элемент на HTML-странице. По указанным объектам-описаниям объект *ImplicitBioStaticRecorder* инициализирует экземпляры *ImplicitBioElement Recorder* и агрегирует их в приватном поле *_elements.* После инициализации объекта производится сохранение UI-событий с указанных элементов, а также событий мыши. Когда пользовательский сценарий завершен (в данном случае при отправке формы), клиент-приложение может получить записанные события вызовом публичного метода *getRecordedEvents* у объекта, инициализированного ранее и передать их для регистрации или авторизации по поведенческим признакам на сервер.

Листиг 1

```
1.  this.recorder = new ImplicitBioStaticRe-
    corder({
2.    elements: [
3.      {
4.        name: 'login',
5.        type: 'input',
6.        cssId: 'login'
7.      },
8.      {
9.        name: 'password',
10.       type: 'input',
11.       cssId: 'password'
12.     },
13.     {
14.       name: 'submitButton',
15.       type: 'button',
16.       cssId: 'submit-button'
17.     },
18.   ],
19. })
```

Разработанный модуль используется в автоматизированной системе для исследования статичной мультимодальной поведенческой авторизации и опубликован в Интернете [11] для дальнейшего массового исследовательского эксперимента.

Заключение

В данной работе разработан модуль сбора пользовательских событий для исследования статичной поведенческой авторизации в веб-приложениях.
Следует отметить, что в перспективе разработанный модуль может быть легко использован не только для неявной поведенческой авторизации (т.е. с использованием уже существующих в клиент-приложении элементов пользовательского интерфейса), но и для явной – в случае создания отдельных UI-сценариев, поставляемых вместе с библиотекой, которые позволяют записывать более

уникальные профили пользователей, и, следовательно, дают лучшую точность.

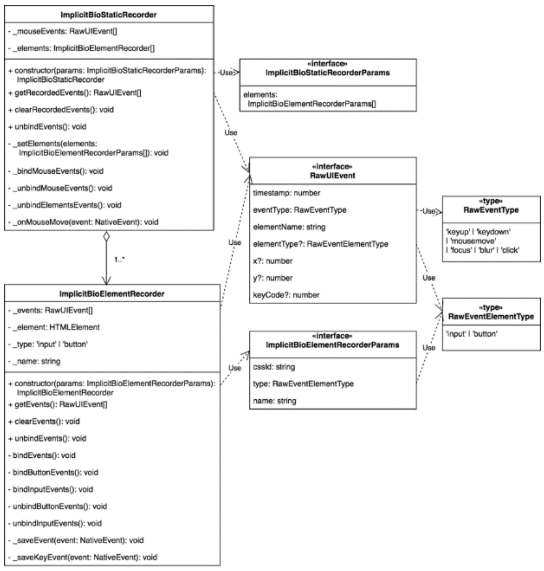

Рис. 1. Диаграмма классов модуля

Список использованных источников

1. V.M. Patel, R. Chellapa, D. Chandra and B. Barbell Continuous User Authentication on Mobile Devices // IEEE Signal processing magazine, vol. 33, no. 4, pp. 49–61, 2016.
2. N. L. Clarke, Transparent User Authentication: Biometrics, RFID and Behavioural Profiling. London: Springer, 2011.
3. H. Khan, U. Hengartner, and D. Vogel Usability and security perceptions of implicit authentication: Convenient, secure, sometimes annoying // 11th Symp. Usable Privacy and Security (SOUPS 2015), 2015, pp. 225–239.
4. Stewart JC, Monaco JV, Cha S-H, Tappert CC. An investigation of keystroke and stylometry traits for authenticating online test takers. Proceedings of the International Joint Conference on Biometrics (IJCB '11); October 2011; pp. 1–7.
5. Balagani KS, Phoha VV, Ray A, Phoha S. On the discriminability of keystroke feature vectors used in fixed text keystroke authentication. Pattern Recognition Letters. 2011;32(7):1070–1080.
6. A. Weiss, A. Ramapanicker, et. al User Authentication Through Mouse Dynamics // IEEE Trans. on Information Forensics and Security, Vol. 8, No. 1, January 2013.
7. Z. Cai, C. Shen, and X. Guan Mitigating Behavioral Variability for Mouse Dynamics: A Dimensionality-Reduction-Based Approach // IEEE Transactions on Human-Machine Systems, Vol. 44, No. 2, April 2014.
8. N. Zheng, A. Paloski, and H. Wang. 2011. An efficient user verification system via mouse movements. // Proceedings of ACM Conference on Computer and Communications Security (CCS'11). 139–150.
9. S. Mondal, P. Bours Combining keystroke and mouse dynamics for continuous user authentication and identification // 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA).
10. K. Bailey, J. Okolica, G. Peterson User identification and authentication using multi-modal behavioral biometrics // Computers & security, Vol. 43, June 2014, Pages 77-89.
11. ImplicitBio Demo [Electronic resource] // URL: https://agazizov.pro/projects/implicitbio/ (accessed 07.10.2018).