3. *Акушуев Р.Т.* Принцип работы VPN и его особенности // Modern Science. – 2020. – № 7. – С. 312-314.

4. *Хант К.* TCP/IP. Сетевое администрирование. – СПб.: Питер, 2007. – 816 с.

5. *Baka P, Schatten J.* SSL/TLS under lock and key: a guide to understanding SSL/TLS cryptography. – Keyko books, 2020. – 132 p.

6. *Шапошников И.В.* Web-сервисы Microsoft .NET. – СПб: БХВ-Петербург, 2002. – 336 с.

7. *Мак-Дональд М., Шпушта М.* Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. – М.: Вильямс, 2009. – 1408 с.

Саломатин А.А.

Методы противодействия отслеживанию браузерных отпечатков пользователей

Аннотация: Рассматриваются методы противодействия отслеживанию браузерных отпечатков пользователей. Проанализированы различные группы мер, позволяющие препятствовать корректному отслеживанию браузерных атрибутов и изменять их значения таким образом, что сформированный отпечаток браузера не сможет верно идентифицировать пользователя. Приведены примеры способов осуществления мер в каждой группе. Отдельное внимание уделено оценке эффективности применения данных методов на сущности, полученные с помощью инструмента «fingerprint3.js». На основе проведенного исследования становится возможным осуществление практического эксперимента по идентификации пользователя с учетом применения методов, препятствующих получению полного и верного идентификатора пользователя. Развитие методов противодействия отслеживания браузерных отпечатков особенно важно для сложных систем, обслуживающих большое количество субъектов доступа.

Ключевые слова: кибербезопасность, браузерный отпечаток, идентификатор, информационная безопасность, браузерные атрибуты

В эпоху информационных технологий все большее внимание приобретает проблема кибербезопасности. Пользователи интернет-сетей, не только обычные люди, но и критически важные государственные инфраструктуры, подвергают себя рискам потери конфиденциальности, связанной с тем, что большинство веб-серверов в наше время собирает информацию о пользователях, которые с ними взаимодействуют. В таком случае, несмотря на положительный аспект, связанный с тем, что обеспечивается безопасность самого веб-сервера, безопасность посетителей ставится под угрозу.

Нередко собираемая информация о пользователях преподносится в виде их цифрового следа, который представляет собой набор данных о статических и динамических поведенческих признаках пользователя в сети [1,2]. Более того, получение цифрового следа не всегда требует длительного периода времени. Такая ситуация происходит, например, если формируется браузерный отпечаток пользователя, который содержит сведения о браузерных атрибутах пользователя (версии браузера, операционной системе, языке и т.д.) [3].

Снятие отпечатков браузера происходит в два этапа [4,5]. Первый этап – получение пользовательских данных о признаках по различным каналам (например, с помощью JavaScript или плагинов). Второй этап – объединение всех значений признаков в одну строку и затем вычисление идентификатора – цифрового следа пользователя.

Рассмотрим, какие существуют меры противодействия сбору информации, необходимой для выполнения этой цели, и проведем классификацию используемых методов по группам с примерами для каждой из групп:

1) *Блокировка доступа к данным пользователя*. Ее можно осуществить с помощью отключения JavaScript, который является основным механизмом получения атрибутов пользователя в наше время. Многие веб-сайты запрещают доступ, если браузер не поддерживает JavaScript [6]. Таким образом, остановка JavaScript может быть неприемлемой для значительной части пользователей

Другим вариантом является использование Do Not Track HTTP заголовка, который принимает значение, указывающее на предпочтение пользователя в отслеживании и отправляемое с сообщением запроса. Однако, и этот метод порой тяжело реализовать, потому что большинство серверов игнорируют такие запросы на запрет, либо теряют функциональность при отключении отслеживания.

2) *Создание идентичных признаков*. Подразумевается, что пользователи будут иметь одинаковые значения браузерных атрибутов, что затруднит различие пользователей между собой. В качестве примера можно привести Tor браузер, концепция которого заключается в изменении и ограничении многих атрибутов (например, списка шрифтов, списка плагинов, User-Agent и т.д.).

3) *Снижение уникальности идентификатора без надстроек.* Оно происходит за счет изменения числа некоторых атрибутов. Например, за счет уменьшения числа используемых плагинов или увеличения числа используемых браузеров.

4) *Снижение уникальности идентификатора с помощью надстроек.* С одной стороны, возможно скрывать значимые атрибуты, оказывающие наибольшее влияние на идентификацию пользователя. В данном случае, например, возможно применить Privaricator, который генерирует случайным образом значимые признаки (например, списки плагинов и шрифтов) с помощью добавления случайного шума в значения атрибутов. Другой пример – User Agent Switcher, который может управлять HTTP-заголовком User-Agent [7]. Изменение лишь значимых атрибутов не всегда может быть достаточно, поэтому более эффективным может быть использование надстроек, влияющих на получение и других признаков. Например, RubberGlove может находить объекты навигатора и экрана внутри веб-страницы, а затем заменять их значения на нулевые. Объект навигатора содержит информацию о названии и версии браузера, поддерживаемых типах MIME и плагинах, платформе, на которой был скомпилирован браузер, поддерживаемом языке и операционной системе, в которой браузер запущен. Объект экрана, в свою очередь, хранит информацию об экране пользователя, например, информацию о разрешении (высоте и ширине) экрана, а также глубины цветов. Имеются также расширения, блокирующие Canvas, которые ограничивает веб-

сайты от получения данных о canvas параметре пользователя, получаемого с помощью JavaScript.

Для того, чтобы лучше понять, как влияет применение мер противодействия на формирование браузерного отпечатка, предлагается обратиться к эксперименту, в котором будут вычислены браузерные атрибуты без применения надстроек и вместе с ними.

Для вычисления браузерных отпечатков пользователей можно задействовать уже готовые ресурсы. С одной стороны, возможно использование специализирующихся сайтов. С другой стороны, можно применить методы самостоятельно, при этом создав свой сервер. Такой подход можно осуществить с помощью библиотеки «fingerprint3.js». Ее преимуществом является частое обновление и оптимизация кода с целью увеличения числа получаемых браузерных атрибутов и улучшения параметров работы самой программы.

Было выявлено, что всего для полученных 32 атрибутов с помощью надстроек, примеров для группы мер под номером 4, является возможным изменение таких параметров, как userAgent, fonts, screenFrame, screenResolution, colorDepth, languages, osCpu, plugins, platform, canvas, cpuClass, contrast. Изменение других браузерных атрибутов также может иметь место, но зависит от используемой надстройки в эксперименте.

Таким образом, в ходе выполнения данного исследования были рассмотрены четыре группы методов противодействия идентификации пользователей на основе браузерных отпечатков. В каждой были приведены примеры с описанием функциональности конкретных методов. Создались условия для проведения будущего эксперимента по применению надстроек с целью скрытия подлинного отпечатка пользователя и спрогнозировать часть его результатов. Эксперимент может проводиться для любого пользователя, поэтому его результаты могут оказаться эффективны в задаче обеспечения конфиденциальности информации о пользователях в сложных системах.

Исследование выполнено при частичной финансовой поддержке гранта Президента Российской Федерации в рамках научного проекта МК-3172.2021.1.6

Литература:

1. *Агафонов Ю.М.* Деанонимизация пользователей на основе цифровых отпечатков браузера / Безопасность информационного пространства – 2017: XVI Всероссийская научно-практическая конференция студентов, аспирантов, молодых ученых (12 декабря 2017 г. Екатеринбург). – Екатеринбург: Изд-во Урал. ун-та, 2018. – С. 3-5.

2. *Feher K.* Digital Identity and The Online-Self: Footprint Strategies – An Exploratory and Comparative Research Study // Journal of Information Science. – 2021. – Volume 47. Issue 2. – P. 192-205.

3. *Nair K., RoseLalson E.* The Unique Id's You Can't Delete: Browser Fingerprints / International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR) (11-13 July 2018 Ernakulam, India). – Ernakulam, 2018. – P. 1-5. – URL: https://ieeexplore.ieee.org/document/8529040 (дата обращения 10.10.2021).

4. *Bujlow T., Carela-Español V., Solé-Pareta J. and Barlet-Ros P.* A Survey on Web Tracking: Mechanisms, Implications, and Defenses // Proceedings of the IEEE. – 2017. – Vol. 105. № 8. – P. 1476-1510.

5. *Luangmaneerote S., Zaluska E., Carr L.* Inhibiting Browser Fingerprinting and Tracking / IEEE 3rd International Conference on Big Data Security on Cloud (26-28 May 2017 Beijing, China). – Beijing, 2017. – P. 63-68. – URL: https://ieeexplore.ieee.org/document/7980318 (дата обращения 10.10.2021).

6. *ElBanna A., Abdelbaki N.* Browsers Fingerprinting Motives, Methods, and Countermeasures / International Conference on Computer, Information and Telecommunication Systems (CITS) (11-13 July 2018 Alsace, Colmar, France). – Colmar, 2018. – P. 1-5. – URL: https://ieeexplore.ieee.org/document/8440163 (дата обращения 10.10.2021).

7. *Fiore U., Castiglione A., De Santis A. and Palmieri F.* Countering Browser Fingerprinting Techniques: Constructing a Fake Profile with Google Chrome / 17th International Conference on Network-Based Information Systems (NBiS 2014) (10-12 September Salerno, Italy). – Salerno, 2014. – P. 355-360. – URL: https://ieeexplore.ieee.org/document/7023976 (дата обращения 10.10.2021).