DOI 10.37882/2223–2966.2022.06.37

ДЕАНОНИМИЗАЦИЯ ПОЛЬЗОВАТЕЛЯ ВЕБ-РЕСУРСА С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ ФОРМИРОВАНИЯ ОТПЕЧАТКА БРАУЗЕРА

DEANONIMIZATION OF A WEB RESOURCE USER USING BROWSER FINGERPRINT FORMATION TECHNOLOGY

V. Sharmaev
E. Karpukhin
S. Sidorin
A. Zherdev

Summary. There are many approaches to collecting information about a user on the network, which differ both in the means used and in the end goal. The most common deanonymization methods are easily bypassed by attackers — the site's cookie processing is disabled, the IP address is hidden, and only the incognito session is used. One of the most flexible approaches is the browser fingerprint technology discussed in the article, the essence of which is to generate a unique user identifier based on a set of user parameters (information about the operating system, browser version, selected system language, time zone, screen resolution, color depth, and also many others). The paper highlights the main characteristics of the user that can be used in the formation of a browser fingerprint, their features are disclosed. User deanonymization can be used to create customized advertising campaigns tailored to the interests of the person, to improve content recommendation systems (for example, articles, videos, music), for secure authentication, collecting statistics about site visitors and analytics. The article also provides other possible scenarios for applying the technology. The methodology presents three possible scenarios: cross-browser solution, maximum amount of data, and high accuracy. For each of the scenarios, the most appropriate array of user characteristics used to form the fingerprint was selected, and examples of the JavaScript script were demonstrated. The disadvantage of the technology is the fact that when the value of one of the analyzed parameters changes, the entire output data block also changes. The solution to this problem is the choice of the optimal sensitivity threshold. The calculation of the optimal sensitivity threshold depending on the number of analyzed parameters is made, examples of its use to determine whether to consider the user of the web service as a repeat visitor or a new user are given. Conclusions are drawn about the applicability of the technology in practice, recommendations are given to improve the accuracy of the result.

Keywords: browser fingerprint, device fingerprint, user deanonymization, JavaScript.

Шармаев Вадим Игоревич
ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)»
vadssx@ya.ru
Карпухин Евгений Олегович
Ведущий инженер, ФГАО ВО «Национальный исследовательский университет ИТМО»
ret1987@ya.ru
Сидорин Сергей Юрьевич
ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)»
sarmatsid@yandex.ru
Жердев Александр Александрович
Вирусный аналитик, Group-IB
misterrio535@gmail.com

Аннотация. Существует множество подходов к сбору информации о пользователе в сети, которые различаются как используемыми средствами, так и конечной целью. Наиболее распространенные методики деанонимизации легко обходятся злоумышленниками — отключается обработка cookie сайта, скрывается IP-адрес, используется исключительно сеанс в режиме инкогнито. Одним из наиболее гибких подходов является рассмотренная в статье технология browser fingerprint, суть которой состоит в генерации уникального идентификатора пользователя на основе набора его параметров (сведений об операционной системе, о версии браузера, выбранном языке системы, часовом поясе, разрешении экрана, глубине цвета, а также множества других). В работе выделены основные характеристики пользователя, которые можно использовать при формировании отпечатка браузера, раскрыты их особенности. Деанонимизация пользователя может использоваться для создания индивидуальных рекламных компаний, подходящих под интересы человека, для усовершенствования систем рекомендации контента (например, статей, видеозаписей, музыки), для безопасной аутентификации, сбора статистики о посетителях сайта и аналитики. В статье также приведены другие возможные сценарии применения технологии. В методологии представлены три возможных сценария: кросс-браузерное решение, максимальное количество данных и высокая точность. Для каждого из сценариев выбран наиболее подходящий массив характеристик пользователя, используемый для формирования отпечатка, а также продемонстрированы примеры работы JavaScript-сценария. Недостатком технологии является тот факт, что при изменении значения одного из анализируемых параметров изменяется и весь выходной блок данных. Решением этой проблемы является выбор оптимального порога чувствительности. Произведен расчет оптимального порога чувствительности в зависимости от количества анализируемых параметров, приведены примеры его использования для определения, считать ли пользователя веб-сервиса повторным посетителем или новым пользователем. Сделаны выводы о применимости технологии на практике, даны рекомендации для повышения точности результата.

Ключевые слова: отпечаток браузера, отпечаток устройства, деанонимизация пользователя, JavaScript.

Технология формирования отпечатка браузера представляет собой метод, используемый онлайн-сервисами и сайтами для идентификации посетителей, суть которого заключается в присвоении каждому пользователю неповторимого идентификатора (отпечатка), например: *mhxkbwxa6mrxpz5g*. Этот идентификатор зависит от набора параметров пользователя, представляющих собой уникальный массив данных, например, совокупность сведений о разрешении экрана, установленных шрифтах и модели используемого устройства [1]. Полученный отпечаток останется постоянным, даже если пользователь перейдет в режим инкогнито или включит VPN.

Название такого метода определяет его ключевую особенность — полученный идентификатор уникален, как и реальные отпечатки пальцев. В силу своей уникальности полученный идентификатор называют также отпечатком устройства (device fingerprint).

Первоначальное применение данной технологии заключалось в оптимизации сайта для пользователя независимо от того, с какого устройства пользователь посетил онлайн-ресурс — с телефона, с планшета или с компьютера. Без лишних действий пользователь сможет увидеть привычную для себя ленту новостей на интересующие его темы, даже если он еще не выполнил авторизацию на сайте, сохранятся настройки пользователя, указанные предпочтения. Технология нашла свое применение и в рекламе. Так, сервер, собрав информацию о модели поведения пользователя и его характеристиках, сможет тонко настроить персональную (таргетированную) рекламную компанию. Такая реклама окажется более точной, чем реклама, настроенная на простом анализе IP-адреса пользователя. Могут использоваться и отдельные характеристики устройства — например, человек с низким разрешением экрана (1024x768) может стать потенциальным покупателем нового монитора в онлайн-магазине, а человек, посетивший страницу магазина в ближайшие дни после крупного релиза новой модели смартфона, которым он пользуется, может заинтересоваться в обновлении своего устройства.

Значимая роль уделяется отпечаткам устройств при модерации онлайн-ресурсов. Злоумышленник, сменивший IP-адрес и свою учетную запись, останется заблокированным, потому что помимо этих характеристик будет проанализировано множество дополнительных — его устройство, версия браузера, операционной системы и т.д. Такой подход позволит как минимизировать активность злоумышленников, так и отделить реальных посетителей сайта от ботов, совершающих вход с одного и того же устройства.

Зачастую отпечатки устройств используются и для аналитических целей — с их помощью легко собрать статистику посетителей сайта, например, узнать, есть ли среди них значительная доля пользователя с нестандартным разрешением экрана, для которого стоит разработать адаптивную версию сайта. Применяются отпечатки браузера и для слежения за состоянием сессии, для аутентификации пользователей.

Код, который производит вычисление идентификатора пользователя, описывается на языке JavaScript — язык позволяет связать функционал элементами HTML (также с Flash, Silverlight, прекратившими поддержку) и использовать их как вспомогательные средства.

Метод идентификации пользователя в Интернете по отпечатку устройства заменил следящие cookie [2]. Файлы cookie — это небольшие пакеты текстовых файлов, которые хранятся на компьютере пользователя и содержат данные, которые могут предоставить веб-сайтам информацию для улучшения пользовательского опыта. Такие файлы помогают как разработчику, так и пользователю онлайн-сервиса, например, сохраняя метки времени для просмотренного фильма, запоминая указанные пользователем настройки. Для разработчика cookie — это инструмент для сбора статистики, для оптимизации и совершенствования работы сайта.

Однако cookie, хранящиеся на персональном компьютере пользователя, могут быть удалены как с помощью настроек браузера, так и вручную, что является проблематичным при использовании их как уникального идентификатора пользователя. В отличие от них, отпечаток браузера пользователя хранится на самом сервере и не зависит от действий пользователя.

При формировании отпечатка браузера (устройства) могут использоваться различные сведения [3, 4]:

- User-agent (строка, которая включает в себя информацию как о самом браузере, так и о его версии, типе устройства, языковых настройках и т.д.);
- Часовой пояс (разность в минутах между временем Всемирное Координированным временем UTC и местным временем устройства пользователя);
- Разрешение экрана и глубину цвета (дополнительно разрешение экрана может сообщить информацию о том, поддерживает ли устройство поворот экрана);
- Установленные системные шрифты (в этом случае достаточно простого перебора путем добавления на страницу элемента со шрифтом из массива проверяемых шрифтов и проверки, поменялся ли размер символов — если размер символов изменился, значит, шрифт установлен в системе);

VR4Xu19CZgU1dn1ud0zAzMsw6bsu6AigiJqRKO4RI1kwXyJZnEhLohLXJKoUWPUaKLGJLhERNw
wav5okk/zKUaNEdx3ECGICLIzrAIyMHvX/5zquj3VNdXTVd1VPd09730enWH6rudW31PvehXyvB
gw+gM4EMD+APYBAMhAXwC9AAxOMf3VALYCqALA31cA+ATAIgIgW1XrcxYFRYfR8AgP+NBlAJoB
xAR+un/r2z1a4aQC2AGus//fsOAIsBLLF+cizWiRcj2HVANa8jz7dQpicICAJFioDKt3UZMEgUxwP4
KoAjAAwMeI5fAiAJkCB6BNx3ojuy1XOV2PbiKNR9eCQ67zgWXXE0AE1D2Q+8FsDbF4H8B8ox
SE9FWMqDE8Vi6ySmoW8e94BTOTrhQ3qV4sMdllOESOQF18oA8ZXAJwK4JuWpFGQkL8D4GkA
z1riTotFRAEcDJhEcox1dHQNbKkkEA79NJTiVFlWIZDAMM+0o24ALgNw+UEjO3Wr7Fxi9rOzuhE
fLdtNSfYZADcBWJXgFNJOEMgFAm1GIAYMqqCmADgXwH65WGwYY1BPNhvAQwCWZjAOs6
z2xFIee/X06FUI5oNpTiEpCIF4h/OAFscBGDulEm9u91w3jAM6UtBuLmsgqrF7DmbcNODa0gkV1

Рис. 1. Изображение, полученное с помощью тега Canvas, и соответствующий ему код

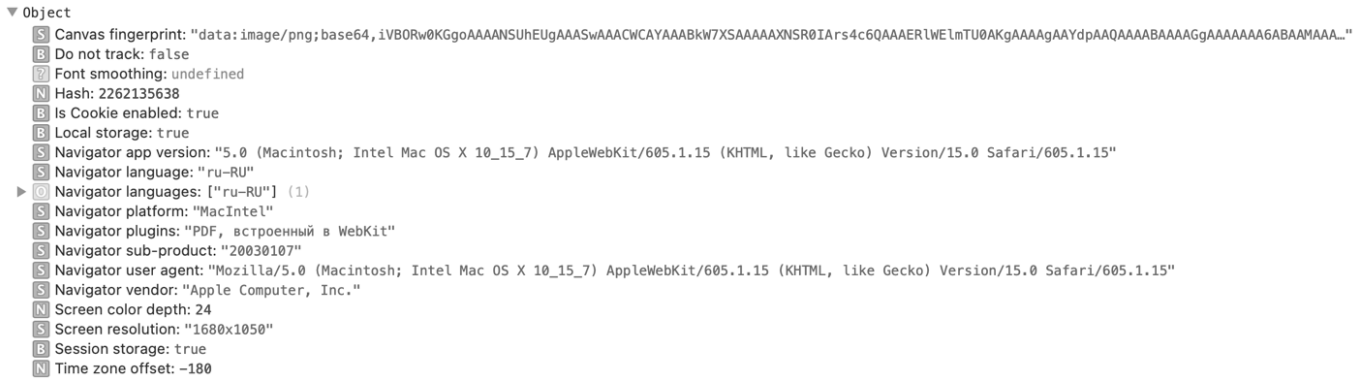

Рис. 2. Полученный массив значений

- Установленные плагины и их версии (несмотря на то, что современные браузеры не отдают весь список установленных плагинов, вновь достаточного простого перебора — в том случае, если плагин из массива проверяемых плагинов установлен у пользователя, браузер подтвердит этот запрос);
- Операционная система и множество других сведений, таких как запрет на определение геолокации, сглаживание шрифтов, тип соединения и т.д.

С развитием языка JavaScript и появлением новых возможностей браузеров этот список продолжает расширяться. Новым подходом для формирования отпечатка браузера стала технология Canvas [4].

Canvas (англ. холст) — это тег HTML5, предназначенный для создания растрового изображения при помощи скриптов, обычно на языке JavaScript. WebGL использует холст (canvas) HTML5 для визуализации 2D- и 3D- графики в браузере. Суть подхода состоит в том, что каждый компьютер рендерит (отрисовывает) изображение по-разному из-за особенности конфигурации компьютера, характеристик операционной системы, свойств самого браузера. Полученное изображение можно использовать как уникальный идентификационный код, превратив его в хэш.

Для того, чтобы изменить идентификатор, полученный с помощью Canvas, будет недостаточно изменить часовой пояс или поменять разрешение экрана. Потре-

Рис. 3. Проверка на наличие записи в базе данных

буется замена графического адаптера, драйвера графического адаптера, а при их отсутствии — всего процессора.

В рамках эксперимента был выполнен JavaScript-сценарий fingerprint.js, работа которого заключается в последовательном вызове функций, определяющих основные характеристики браузера и устройства пользователя. Все полученные значения записываются в массив, а затем проходят через хэш-функцию [5].

Дальнейшая логика работы скрипта заключается в проверке, имеется ли в базе данных полученный отпечаток браузера. Если отпечаток имеется — на экран выводится уведомление о том, что страница уже была посещена ранее. Если такой записи не обнаружено — она добавляется в базу данных.

Эксперимент проведен для трех возможных сценариев:

1. Кросс-браузерное решение. Анализируемые параметры: разрешение экрана, язык системы, часовой пояс, операционная система.

Сценарий используется для деанонимизации пользователя независимо от используемого им браузера. Считываемые параметры обладают высокой стойкостью и не меняются со временем, например, на них не окажет влияния обновление браузера. Такой подход можно использовать, например, для сохранения настроек пользователя, его предпочтений.

2. Максимальное количество данных о пользователе. Анализируемые параметры: разрешение экрана, отпечаток Canvas, включены ли cookie, включено ли Local Storage, включено ли Session Storage, операционная система, версия браузера, браузер, разработчик браузера, язык системы, список плагинов, часовой пояс, глубина цвета, запрет на определение геопозиции.

Результат работы скрипта при таком сценарии может регулярно меняться независимо от действий пользователя, например, после обновления одного из установленных плагинов. В различных браузерах результат также будет отличаться, но такой подход можно использовать для аналитических целей, например, с целью определения целевых групп сайта.

3. Высокая точность. Анализируемые параметры: хэш Canvas.

Результат работы скрипта при таком сценарии максимально стабилен и точен, но в пределах одного браузера, для его изменения может потребоваться обновление графического адаптера или всего процессора в целом. Такой подход может использоваться для контроля состояния сессии пользователя или для обеспечения безопас-

ности учетной записи. Например, при использовании в сервисах онлайн-банкинга в случае изменения отпечатка браузера пользователя система может высылать код-подтверждение на другое устройство пользователя.

Работа скрипта была проверена на компьютере с операционной системой Windows, на компьютере с операционной системой MacOS, на мобильном устройстве на платформе Android и на мобильном устройстве на платформе iOS в трех режимах: обычный режим, режим инкогнито и режим инкогнито с включенным VPN в стандартной конфигурации. Эксперимент повторялся для двух браузеров (Safari и Google Chrome, на Windows — Mozilla и Google Chrome), затем повторялся вновь на другой версии браузера.

В ходе эксперимента получены следующие результаты:

1. При кросс-браузерном решении полученный хэш оставался неизменным при включенном VPN, включенном режиме инкогнито, при смене браузера. Аналогичный результат был получен на компьютере с операционной системой MacOS и на мобильных устройствах. После повторения эксперимента на другой версии браузера все устройства получили такой же результат.
2. При сборе максимального количества данных хэш изменялся при смене браузера, но оставался неизменным при включенном режиме инкогнито и включенном VPN. Аналогичный результат был получен на компьютере с операционной системой MacOS и на мобильных устройствах. При повторении эксперимента на другой версии браузера результат поменялся на всех устройствах.
3. При решении с высокой точностью полученный хэш изменялся при смене браузера, но оставался неизменным при включенном режиме инкогнито и включенном VPN. Аналогичный результат был получен на компьютере с операционной системой MacOS и на мобильных устройствах. После повторении эксперимента на другой версии браузера все устройства получили такой же результат.

Рассмотрим вектор весов параметров пользователя $\eta = [\eta_1, \eta_2, \eta_3 \dots \eta_{17}]$ (сценарий максимального количества данных, при котором анализируется 17 различных параметров), причем $\eta_1 + \eta_2 + \dots + \eta_{17} = 1$. Оценим каждый из весов параметров.

Из 17 параметров выделим 7 строковых параметров как наиболее уникальные (это значения canvas η_1, платформы браузера η_6, массива языков η_8, платформа устройства η_9, список плагинов η_{10}, версия браузера η_{11} и производитель устройства η_{13}). Тогда этим семи параметров определим вес уникальности 0.7, а всем остальным параметрам — 0.3. Получившийся вектор весов с учетом их уникальности: $I = [0.1, 0.03, 0.03, 0.03, 0.03, 0.1 \dots 0.03]$ $(I_1 + I_2 + \dots + I_{17} = 1)$

Время получения и информативность каждого из параметров можно считать одинаковым, тогда вес каждого параметра с учетом его времени получения и информативности: $T = [0.0588, 0.0588, 0.0588, 0.0588 \dots 0.0588]$ ($T_1 + T_2 + \dots + T_{17} = 1$)

Будем считать, что оба этих критерия равнозначны, тогда умножим оба вектора на коэффициенты значимости критериев $\alpha = [0.5, 0.5]$.

Итоговый вес параметров:

$$\eta = [0.07941, 0.04441, 0.04441, 0.04441, 0.04441, 0.07941, \dots, 0.04441].$$

Будем считать допустимой погрешность, при которой изменился лишь один из наименее уникальных параметров (например, параметр «Не отслеживать» η_2), тогда вектор подобия $S = [1, 0, 1, 1, \dots 1]$, а вероятность верной идентификации пользователя:

$$P = S * \eta = 1 * 0.07941 + 0 * 0.04441 + + 1 * 0.04441 + \dots + 1 * 0.04441 = 1 - - 0.04441 = 0.95559$$

Тогда с учетом возможных округлений за порог чувствительности примем величину 0.955. Рассмотрим еще две ситуации:

1) Изменился один из наиболее уникальных параметров (например, параметр canvas η_1:

Вектор подобия $S = [0, 1, 1, 1, \dots 1]$. Вероятность верной идентификации пользователя:

$$P = S * \eta = 0 * 0.07941 + 1 * 0.04441 + + 1 * 0.04441 + \dots + 1 * 0.04441 = 1 - -0.07941 = 0.92059$$

Значение меньше порога чувствительности 0.955. Считаем, что веб-ресурс посетил новый пользователь.

2) Изменилось два из наименее уникальных параметров (например, параметр «Не отслеживать» η_2 и параметр «Сглаживание шрифтов» η_3):

Вектор подобия $S = [1, 0, 0, 1, \dots 1]$. Вероятность верной идентификации пользователя:

$$P = S * з = 1 * 0.07941 + 0 * 0.04441 +$$

$$+ 0 * 0.04441 + \cdots + 1 * 0.04441 = 1$$
$$- 0.04441 - 0.04441 = 0.91118$$

Значение меньше порога чувствительности 0.955. Считаем, что веб-ресурс посетил новый пользователь.

Теоретические значения получились близки к статистическим значениям, представленным исследованиями ресурса Panopticlick [6], согласно которому лишь 1 из 286777 браузеров дадут такой же отпечаток, как браузер другого пользователя. В среднем, точность идентификации пользователя при помощи отпечатка браузера составляет 99,24%. Изменение одного из параметров браузера снижает точность идентификации пользователя лишь на 0,3%.

Результаты эксперимента подтвердили, что данная технология обладает гибкостью в настройке, а при верном выборе анализируемых параметров обладает также и высокой точностью. Отметим, что с увеличением параметров увеличивается вероятность изменения результата, поэтому важным остается вопрос выбора оптимального порога чувствительности, который определяет — считать ли посетителя с изменившимся параметром тем же самым пользователем, либо ассоциировать его с новым пользователем.

Технология формирования отпечатка браузера отличается своей гибкостью и, при грамотном подходе к выбору анализируемых параметров, высокой точностью. При изменении любого анализируемого параметра изменяется конечный результат (изменение как минимум одного бита входных данных должно приводить к изменению значения всего выходного блока). Для усовершенствования технологии вводят «порог чувствительности» для каждого из параметров, и чем больше уникальность параметра, тем выше его чувствительность (такое хэширование называют фазихэшированием, или нечетким хэшированием). При этом сами значение порога чувствительности зависит от веб-ресурса, на котором он используется. Разработчик выбирает оптимальное значение, являющееся компромиссом между ситуацией излишнего ложного срабатывания и ситуацией чрезмерного реагирования системы.

Была проанализирована точность, с которой может быть идентифицирован пользователь при анализе 17 пользовательских параметров. Существуют готовые библиотеки с открытым кодом, позволяющие разработчикам генерировать и обрабатывать отпечатки браузеров пользователей, гарантируя точность свыше 99,5% без дополнительных затрат на разработку. Использование таких библиотек может предотвратить кражи учетных записей пользователей, мошенничество. В том случае, если уникальный идентификатор пользователя изменился, будет разумным если не блокировать пользователя, то хотя бы выслать подтверждающий код на почту или номер телефона, либо даже завершить текущую сессию.

Со стороны разработчиков, помимо улучшения интерфейса взаимодействия с пользователем, важнейшей задачей является обеспечение безопасности работы пользователя. Рассмотренная в статье технология в совокупности с другими средствами позволяет защитить учетную запись пользователя от кражи, предотвратить мошенничество в сфере банковских карт, защитить авторские права. Любые действия злоумышленника, совершенные с использованием данных пользователя-жертвы, но при этом обладающего отличающимся отпечатком устройства, могут быть остановлены системой, а затем дополнительно запрошены иные сведения — например, кодовое слово, указанное при регистрации на сайте. На изначальное устройство при этом направляется оповещение безопасности с рекомендацией проверить активность в своем аккаунте и при необходимости сменить пароль или обратиться в службу поддержки. Фактически, отпечаток может быть использован для оценки вероятности мошенничества или иных противоправных действий со стороны конкретного пользователя.

Как и любая современная технология, отпечатки браузера могут стать опасным инструментом в руках самих злоумышленников. Избежать или проконтролировать сбор параметров браузера практически не представляется возможным. Точность, с которой отпечаток браузера идентифицирует пользователя, в руках злоумышленника может стать опасным инструментом. Не исключена возможность дальнейшей передачи баз таких отпечатков или их продажи, а сами пользователи никак не могут вмешаться в этот процесс и повлиять на использование их данных.

Идентификация пользователя без его ведома и согласия может нарушить один из основных принципов — право на анонимность. Исключить возможность такого сбора параметров практически невозможно, единственный надежный способ отгородиться от него — это отказаться от пользования сетью Интернет. Если cookie в настоящее время регулируются в ряде стран, а сайты должны в обязательном порядке запрашивать согласие на их обработку, то получение отпечатка — совсем новая технология, пока еще не затронутая ни одним законом в сфере информационной безопасности. Следующим логичным шагом может стать создание регламента сбора и обработки характеристик пользователя посредством формирования отпечатков браузера, принятого на уровне законодательных актов стран.

ЛИТЕРАТУРА

1. Pugliese G., Riess C., Gassmann F., Benenson Z. Long-Term Observation on Browser Fingerprinting: Users' Trackability and Perspective. Proceedings on Privacy Enhancing Technologies, 2020 (2). p 558–577.
2. Papadogiannakis E., Papadopoulos P, Kourtellis N, Markatos E. User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users. Computers and Society, 2021. p 3
3. FaizKhademi A., Zulkernine M., Weldemariam K. FPGuard: Detection and Prevention of Browser Fingerprinting. Data and Applications Security and Privacy XXIX, 2019. Lecture Notes in Computer Science, vol 9149. p 310.
4. Vastel A., Rudametkin W., Rouvoy R., Blanc X. FP-Crawlers: Studying the Resilience of Browser Fingerprinting to Block Crawlers. NDSS Workshop on Measurements, Attacks, and Defenses for the Web, 2020. p 6
5. Bird S., Willoughby R., Mishra V., Zeber D., Lopatka M., Englehardt S., Rudametkin W. Actions speak louder than words: semi-supervised learning for browser fingerprinting detection. Cryptography and Security, 2020. p 5
6. Eckersley P. How Unique Is Your Web Browser? Electronic Frontier Foundation. URL: https://panopticlick.eff.org/static/browser-uniqueness.pdf

Московский авиационный институт