СЕКЦИЯ 1.
ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИОННЫХ СИСТЕМ

УДК 004.056

Ю. М. Агафонов
Научный руководитель: ст. преп. Т. И. Паюсова
Тюменский государственный университет, Тюмень

ДЕАНОНИМИЗАЦИЯ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ ЦИФРОВЫХ ОТПЕЧАТКОВ БРАУЗЕРА

Аннотация. В данной работе рассматривается наиболее актуальный и перспективный с точки зрения законодательства Российской Федерации способ идентификации пользователей сети Интернет. Рассмотрены также основные моменты, связанные с разработкой серверного приложения для деанонимизации пользователей.

Ключевые слова: анонимизация; деанонимизация; VPN; цфировой отпечаток браузера; JavaScript; параметры браузера; серверное приложение; TLSH.

Необходимым условием для совершения безнаказанного преступления в киберпространстве является анонимность. Нарушители активно используют методы анонимизации для сокрытия следов своих преступлений: несанкционированного доступа и кражи данных, подделки платежных реквизитов, нарушения авторских прав, атак, направленных на отказ в обслуживании и других правонарушений. Среди основных методов анонимизации можно выделить web-анонимайзеры, механизмы так называемой «луковой» маршрутизации TOR, сети I2P, VPN-туннели. Деанонимизация пользователя, в частности для

проведения следственных мероприятий или, например, для повышения качества сервиса, осуществима с помощью определения IP-адреса, MAC-адреса, цепочки DNS-серверов, GeoIP, cookie-файлов и прочих методов.

Современное отечественное законодательство стремится ограничить применение анонимайзеров для повышения общего уровня защищенности. В ноябре 2017 года вступил в силу закон о запрете обхода блокировок через VPN-тунели и web-анонимайзеры. Ответственными за выявление анонимайзеров и VPN-сервисов назначены Федеральная служба безопасности и Министерство внутренних дел Российской Федерации. Роскомнадзор ответственен за определение провайдеров, через которые работают анонимайзеры. Стоит отметить, что деанонимизация пользователя должна осуществляться в рамках действующего законодательства и не должна нарушать прав пользователя в отношении его персональных данных. А также необходимо учитывать то, что с учетом использования наиболее популярных механизмов сокрытия сетевого трафика отследить пользователя по сетевому адресу в большинстве случаев практически не представляется возможным.

Одним из возможных и вполне надежных способов деанонимизации пользователей является распознавание цифровых отпечатков браузера — уникальных значений, отражающих настройки веб-обозревателя пользователя. При деанонимизации с помощью цифровых отпечатков браузера можно будет решить довольно много проблем. Представится возможным предотвращать распространение так называемого «пиратского» программного обеспечения. Можно будет привлечь к ответственности конкретного пользователя, который распространял нелегальный цифровой продукт. Появится больше возможностей по предотвращению экстремистской деятельности. Ведь если представится возможным однозначно установить личность того, кто, к примеру, периодически обращается с использованием механизмов сокрытия сетевого трафика к различным экстремистским ресурсам, то, вероятно, удастся предотвратить не одно преступление.

Основная сложность, связанная с деанонимизацией пользователей с помощью цифровых отпечатков браузера, связана с разработкой алгоритма создания уникального значения, однозначно характеризующего конкретный «портрет» браузера. В алгоритме должны быть продуманы изучаемые параметры браузера, должна быть определена используемая хеш-функция, также должно быть выбрано время «забывания» цифрового отпечатка сервером с целью снижения нагрузки на систему и оптимизации процесса использования вычислительных ресурсов.

В разрабатываемом серверном приложении для деанонимизации пользователей применяются в основном JavaScript-сценарии, позволяющие получить информацию о некоторых параметрах браузера пользователя. Помимо основных параметров, используются также не вполне очевидные техники и методи-

ки для получения более подробного и информативного отпечатка. К числу наиболее специфичных техник относятся Canvas Fingerprinting, WebGL Clipping Planes, AudioContext, FontList (JS + CSS).

Все используемые для составления цифрового отпечатка характеристики и техники оцениваются с точки зрения энтропии, обнаруживаемости, доступности и постоянности. Этот квартет свойств позволяет определить, какие из полученных данных принесут наибольшую практическую пользу при формировании итогового отпечатка браузера [1].

Получив и оценив всю необходимую информацию о браузере пользователя, следующим шагом становится организация и хранение этой информации. В приложении используются свыше дюжины различных методик и параметров, половина из которых предоставляет очень большие объемы информации о браузере. Результирующая строка по итогу получается довольно большой, в среднем около тридцати тысяч символов для современных версий браузеров. Хранить такой объем данных в изначальном виде не имеет смысла. Поэтому наиболее разумно хешировать полученную строку данных. При этом обычные алгоритмы хеширования, вроде MD5 или SHA-1 не подойдут, так как при изменении хотя бы долей процента входных данных, в результате хеширования мы получим две совершенно разные строки. Поэтому было принято решение использовать один из алгоритмов нечеткого хеширования, которые нашли широкое применение при поиске похожих электронных писем и последующей блокировки спам-рассылок. Был выбран алгоритм TLSH (Trend Micro Locality Sensitive Hash). Результатом этого алгоритма является 70-значная шестнадцатеричная строка, первые 6 символов которой несут информацию о данных в целом (например, длина исходной строки), а остальные 64 символа уже используются для информации о содержимом данных [2].

Разработка приложения ведется на основании уже существующих проектов в области сбора идентификационной информации о пользователях сети Интернет. Среди них проекты Panopticlick, ClientJS, Am I Unique, BrowserLeaks и еще несколько менее известных проектов. При разработке учитываются их исследования в области энтропии и эффективности использования тех или иных методик и характеристик.

Список литературы

1. *Doty N.* Mitigating Browser Fingerprinting in Web Specifications // W3C: World Wide Web Consortium, 2017. URL: https://w3c.github.io/fingerprinting-guidance (дата обращения: 01.11.2017).

2. JavaScript port of TLSH (Trend Micro Locality Sensitive Hash) // GitHub: The world's leading software development platform, 2017. URL: https://github.com/idealista/tlsh-js (дата обращения: 01.11.2017).