Министерство образования и науки Российской Федерации

Санкт-Петербургский Политехнический Университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Информационная безопасность компьютерных систем»**

**КУРСОВОЙ ПРОЕКТ**

**ПО МОДЕЛЯМ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ**

**«Исследование методов деанонимизации пользователей VPN»**

Выполнил

студент гр. 33636/1 Малинко А.В.

Проверил

преподаватель Овасапян Т.Д.

Санкт-Петербург

2018

Оглавление

[Введение 3](#_Toc532939143)

[Цель работы 3](#_Toc532939144)

[Задачи 3](#_Toc532939145)

[1. Теоретические сведения о деанонимизации в сетях VPN. 4](#_Toc532939146)

[Деанонимизация 4](#_Toc532939147)

[VPN сеть (Virtual Private Network) 5](#_Toc532939148)

[Структура VPN 5](#_Toc532939149)

[IPSec 7](#_Toc532939150)

[Фаза Один и Фаза Два 9](#_Toc532939151)

[Популярные методы деанонимизации 10](#_Toc532939152)

[1. Деанонимизация пользователей VPN путем сопоставления соединений 10](#_Toc532939153)

[2. С помощью цифрового отпечатка 11](#_Toc532939154)

[3. С помощью DNS 12](#_Toc532939155)

[4. С помощью WebRTC 13](#_Toc532939156)

[Глава 2. Практическая часть. 16](#_Toc532939157)

[Простой пример скрипта, для составления цифрового отпечатка: 16](#_Toc532939158)

[Еще один пример: 17](#_Toc532939159)

[Подмена DNS-ответа; внедрение ложного DNS-сервера . 18](#_Toc532939160)

[Схема реализации атаки в случае, если есть доступ к трафику жертвы 18](#_Toc532939161)

[Схема реализации атаки в случае, если доступа к трафику жертвы нет 19](#_Toc532939162)

[Использование WebRTC 20](#_Toc532939163)

[Простой пример кода, который находится в открытом доступе, и позволяет использовать данную уязвимость на своем сайте . 21](#_Toc532939164)

[Список использованной литературы 24](#_Toc532939165)

# Введение

Понятие деанонимизации неотделимо от понятия анонимности. Анонимность — это возможность посещать сайты, совершать какие-либо активные действия на веб-ресурсах, например оставлять сообщения, без возможности связать ваши действия с вашей реальной личностью или местом вашего выхода в сеть.

В день миллионы людей посещают сотни web-страниц, совершая там различные действия.

Бывает так, что эти действия оказываются противоправными, и тогда работникам специальных органов необходимо установить реальную личность нарушителя. Или же наоборот, необходимо скрыть свою личную информацию от посторонних людей. Поэтому вопрос деанонимизации и методов защиты от нее стоит очень остро, и чем больше развивается интернет, тем более актуальной становится эта тема.

# Цель работы

Целью данной работы является описать методы деанонимизации в сетях VPN. Некоторые из них могут использоваться и в случае, когда сети VPN нет, а есть обычное подключение к интернету.

# Задачи

В данной работе передо мной стояли несколько задач:

- Углубиться в значение термина “деанонимизация”.

- Разобраться со структурой и способом работы сетей VPN.

- Найти теоретическую информацию о методах деанонимизации в сетях VPN

- Продемонстрировать работу нескольких методов на практике.

# Теоретические сведения о деанонимизации в сетях VPN.

Деанонимизация − процесс установления личности пользователя в сети либо подлинного места выхода в сеть.

Многие приравнивают анонимность к сокрытию подлинного IP-адреса, но это очень упрощенный подход. Во-первых, злоумышленник может провести JavaScript-атаку и, используя уязвимости в веб-браузере, получить доступ к вашему устройству. Дальше, получив контроль над устройством, постараться определить владельца на основе анализа посещаемых сайтов, реального IP-адреса, документов и информации в мессенджерах. Во-вторых, через сайт можно проверить наличие у вас аккаунтов в социальных сетях, где вы авторизованы. Если вы авторизованы в Facebook и там указаны ваши реальные данные, владелец сайта сможет получить их незаметно для вас.

Установление личности пользователя или точки выхода в сеть по IP-адресу или mac-адресу называется пассивной деанонимизации, когда против пользователя не применяется никаких атак. Атаки на пользователя, например через сайт с вредоносным JavaScript, или фишинг — это активная Деанонимизация.

Важно понимать отсутствие прямой связи между IP-адресом и каким-либо пользователем. IP-адрес принадлежит точке выхода в сеть. Например, если вы подключаетесь к Wi-Fi роутеру, то получаете IP-адрес данного роутера. Все остальные пользователи, подключившись к этому роутеру, получат этот же самый IP-адрес.

Проверка IP-адреса может указать только интернет-провайдера и общее место расположения, проведя сканирование, можно определить модель роутера. Провайдер интернет-связи, подключивший кабель интернета к данному роутеру, знает, на кого он оформлен и где располагается, при запросе от компетентных органов он обязан выдать эту информацию. Есть еще возможность проанализировать логи провайдера, логи роутера − обо всем этом мы расскажем в главе, посвященной деанонимизации.

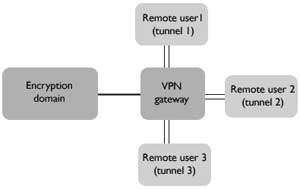
Злоумышленник может взломать Wi-Fi роутер и затем начать перехватывать трафик с целью найти в нем зацепки для установления личности, например аккаунты социальных сетей, либо пытаться заразить компьютер через подмену DNS или «склейку» пакетов. Все это − активная деанонимизация, о которой мы будем еще много говорить.

VPN сеть (Virtual Private Network) — это криптосистема, позволяющая защитить данные при передаче их по незащищенной сети, такой, как Интернет. Цель VPN - прозрачный доступ к ресурсам сети, где пользователь может делать все то, что он делает обычно независимо от того, насколько он удален. По этой причине VPN приобрел популярность среди дистанционных работников и офисов, которые нуждаются в совместном использовании ресурсов территориально разделенных сетей.

## Структура VPN [1]

VPN соединение всегда состоит из канала типа точка-точка, также известного под названием туннель. Туннель создается в незащищенной сети, в качестве которой чаще всего выступает Интернет. Соединение точка-точка подразумевает, что оно всегда устанавливается между двумя компьютерами, которые называются узлами или peers. Каждый peer отвечает за шифрование данных до того, как они попадут в туннель и расшифровку этих данных после того, как они туннель покинут.

Хотя VPN-туннель[2] всегда устанавливается между двумя точками, каждый peer может устанавливать дополнительные туннели с другими узлами. Для примера, когда трем удаленным станциям необходимо связаться с одним и тем же офисом, будет создано три отдельных VPN-туннеля к этому офису. Для всех туннелей peer на стороне офиса может быть одним и тем же. Это возможно благодаря тому, что узел может шифровать и расшифровывать данные от имени всей сети (Рисунок 1):



*Рисунок 1. Схема работы VPN-узла(шлюза).*

В этом случае VPN-узел называется VPN-шлюзом, а сеть за ним - доменом шифрования (encryption domain). Использование шлюзов удобно по нескольким причинам. Во-первых, все пользователи должны пройти через одно устройство, которое облегчает задачу управления политикой безопасности и контроля входящего и исходящего трафика сети. Во-вторых, персональные туннели к каждой рабочей станции, к которой пользователю надо получить доступ, очень быстро станут неуправляемыми (так как туннель — это канал типа точка-точка). При наличии шлюза, пользователь устанавливает соединение с ним, после чего пользователю открывается доступ к сети (домену шифрования).

Внутри домена шифрования самого шифрования не происходит. Причина в том, что эта часть сети считается безопасной и находящейся под непосредственным контролем в противоположность Интернет. Таким образом гарантируется шифрование только той информации, которая передается по небезопасному каналу между офисами.

Всякий раз, когда соединение сетей обслуживают два VPN-шлюза, они используют режим туннеля. Это означает, что шифруется весь пакет IP, после чего к нему добавляется новый IP-заголовок. Новый заголовок содержит IP-адреса двух VPN-шлюзов, которые и увидит пакетный сниффер при перехвате. Невозможно определить компьютер-источник в первом домене шифрования и компьютер-получатель во втором домене.

Существует много вариантов VPN-шлюзов и VPN-клиентов. Это может быть аппаратное устройство или программное обеспечение, которое устанавливается на маршрутизаторах или на ПК. Скажем, ОС FreeBSD поставляется вместе с ПО для создания VPN-шлюза и для настройки VPN-клиента. Свои VPN-решения существуют и для ПО компании Microsoft.

Независимо от используемого ПО, все VPN работают по следующим принципам:

1. Каждый из узлов идентифицирует друг друга перед созданием туннеля, чтобы удостовериться, что шифрованные данные будут отправлены на нужный узел.

2. Оба узла требуют заранее настроенной политики, указывающей, какие протоколы могут использоваться для шифрования и обеспечения целостности данных.

3. Узлы сверяют политики, чтобы договориться об используемых алгоритмах; если это не получается, то туннель не устанавливается.

4. Как только достигнуто соглашение по алгоритмам, создается ключ, который будет использован в симметричном алгоритме для шифрования/расшифровки данных.

Есть несколько стандартов, регламентирующих вышеописанное взаимодействие: L2TP, PPTP, и IPSec. IPSec - наиболее широко поддерживаемый стандарт.

### IPSec[2]

Стандарт IPSec был разработан для повышения безопасности IP-протокола. Это достигается за счет дополнительных протоколов, добавляющих к IP- пакету собственные заголовки. Т.к. IPSec - стандарт Интернет, то для него существуют RFC (Requests For Comments - рабочее предложение). Рассмотрим следующие RFC:

- RFC 2401 IPSec;

- RFC 2402 AH;

- RFC 2406 ESP;

- RFC 2409 IKE.

AH(Authentication Header) - протокол заголовка идентификации. Обеспечивает целостность путем проверки того, что ни один бит в защищаемой части пакета не был изменен во время передачи. Использование AH может вызвать проблемы, например, при прохождении пакета через NAT-устройство. NAT меняет IP-адрес пакета, чтобы, например, разрешить доступ в Интернет с закрытого локального адреса. Так как пакет в таком случае изменится, контрольная сумма AH станет неверной. Также стоит отметить, что AH разрабатывался только для обеспечения целостности. Он не гарантирует конфиденциальности путем шифрования содержимого пакета.

ESP(Encapsulating Security Protocol)- инкапсулирующий протокол безопасности, который обеспечивает и целостность, и конфиденциальность. В режиме транспорта ESP-заголовок находится между оригинальным IP-заголовком и заголовком TCP или UDP. В режиме туннеля заголовок ESP размещается между новым IP-заголовком и полностью зашифрованным оригинальным IP-пакетом.

Так как оба протокола - AH и ESP - добавляют собственные заголовки, они имеют свой ID протокола, по которому можно определить, что последует за заголовком IP. Каждый тип заголовка имеет собственный номер. Например, для TCP это 6, а для UDP - 17. При работе через firewall важно не забыть настроить фильтры, чтобы пропускать пакеты с ID AH-и/или ESP-протокола. Для AH номер ID - 51, а ESP имеет ID протокола равный 50.

Третий протокол, используемый IPSec — это IKE или Internet Key Exchange protocol. Как следует из названия, он предназначен для обмена ключами между двумя узлами VPN. Несмотря на то, что генерировать ключи можно вручную, лучшим и более масштабируемым вариантом будет автоматизация этого процесса с помощью IKE. Для использования IKE, необходимо настроить правило на файрволле для UDP-порта с номером 500, так как именно этот порт используется IKE.

SA (Security Association), что можно приближенно перевести как "связь или ассоциация безопасности" — это термин IPSec для обозначения соединения. При настроенном VPN, для каждого используемого протокола создается одна SA-пара (то есть одна для AH и одна для ESP). SA создаются парами, так как каждая SA — это однонаправленное соединение, а данные необходимо передавать в двух направлениях. Полученные SA-пары хранятся на каждом узле. Если ваш узел имеет SA, значит VPN-туннель был установлен успешно.

Так как каждый узел способен устанавливать несколько туннелей с другими узлами, каждый SA имеет уникальный номер, позволяющий определить, к какому узлу он относится. Это номер называется SPI (Security Parameter Index) или индекс параметра безопасности.

SA хранятся в базе данных c названием - SAD (Security Association Database) или БД ассоциаций безопасности.

Каждый узел IPSec также имеет вторую БД - SPD или Security Policy Database (БД политики безопасности). Она содержит настроенную политику узла. Большинство VPN-решений разрешают создание нескольких политик с комбинациями подходящих алгоритмов для каждого узла, с которым нужно установить соединение.

Какие настройки включает в себя политика?

1. Симметричные алгоритмы для шифрования/расшифровки данных.

2. Криптографические контрольные суммы для проверки целостности данных.

3. Способ идентификации узла. Самые распространённые способы - это предустановленные ключи (pre-shared secrets) или RSA-сертификаты.

4. Использовать ли режим туннеля или режим транспорта.

5. Какую использовать группу Diffie Hellman.

6. Как часто проводить переидентификацию узла.

7. Как часто менять ключ для шифрования данных.

8. Использовать ли PFS.

9. Использовать ли AH, ESP или оба вместе.

При создании политики, как правило, возможно создание упорядоченного списка алгоритмов и Diffie Hellman-групп. В таком случае будет использована первая совпавшая на обоих узлах позиция. Очень важно, чтобы все в политике безопасности позволяло добиться этого совпадения. Если за исключением одной части политики все остальное совпадает, узлы все равно не смогут установить VPN-соединение.

### Фаза Один и Фаза Два[2]

Установка и поддержка VPN-туннеля происходит в два этапа. На первом этапе (фазе) два узла договариваются о методе идентификации, алгоритме шифрования, хэш-алгоритме и группе Diffie Hellman. Они также идентифицируют друг друга. Все это может пройти в результате обмена тремя нешифрованными пакетами (так называемый агрессивный режим) или через обмен шестью нешифрованными пакетами (стандартный режим - main mode). При успешном завершении операции создается SA первой Фазы - Phase 1 SA (также называемый IKE SA) и процесс переходит к Фазе Два.

На втором этапе генерируются данные ключей, узлы договариваются насчет используемой политики. Этот режим, называемый быстрым режимом (quick mode), отличается от первой фазы тем, что может установиться только после первого этапа, когда все пакеты второй фазы шифруются. Такое положение дел усложняет решение проблем в случае неполадок на второй фазе при успешном завершении первой. Правильное завершение второй фазы приводит к появлению Phase 2 SA или IPSec SA, и на этом установка туннеля считается завершенной.

Сначала на узел прибывает пакет с адресом назначения в другом домене шифрования, и узел инициирует Фазу Один с тем узлом, который отвечает за другой домен. Допустим, туннель между узлами был успешно установлен и ожидает пакетов. Однако, узлам необходимо переидентифицировать друг друга и сравнить политику через определенное время. Это время известно как время жизни Phase 1 или IKE SA lifetime. Узлы также должны сменить ключ для шифрования данных через другой отрезок времени, который называется временем жизни Phase 2 или IPSec SA lifetime. Phase 2 lifetime короче, чем у первой фазы, так как ключ необходимо менять чаще. Типичное время жизни Phase 2 - 60 минут. Для Phase 1 оно равно 24 часам.

Задача заключается в том, чтобы сконфигурировать оба узла с одинаковыми параметрами времени жизни. Если этого не произойдет, то возможен вариант, когда изначально туннель будет установлен успешно, но по истечении первого несогласованного промежутка времени жизни связь прервется. Странные проблемы могут возникнуть и в том случае, когда время жизни Фазы Один меньше аналогичного параметра Фазы Два. Если настроенный ранее туннель виснет, то первая вещь, которая нуждается в проверке — это время жизни на обоих узлах.

При смене политики на одном из узлов, изменения вступят в силу только при следующем наступлении Phase 1. Чтобы изменения вступили в силу немедленно, надо убрать SA для этого туннеля из SAD. Это вызовет пересмотр соглашения между узлами с новыми настройками политики безопасности.

## Популярные методы деанонимизации

### Деанонимизация пользователей VPN путем сопоставления соединений

Сопоставление соединений – это один из самых эффективных путей деанонимизации пользователей VPN, применяемых спецслужбами и правоохранительными органами по всему миру.

Первым делом отправляется запрос владельцу VPN с просьбой выдать данные о том, кто использовал такой-то IP-адрес в такое-то время. Предположим, владелец VPN игнорирует ваш запрос.

Существует система ОРМ (оперативно-розыскных мероприятий, в России это СОРМ), которая сканирует весь трафик всех пользователей, а провайдеры, по закону сохраняют данные об активности пользователей. Расшифровать записанный VPN-трафик не получится, но это и не нужно. Необходимо лишь посмотреть, кто из жителей в интересующий промежуток времени устанавливал зашифрованное соединение с нужным вам VPN-сервером.

Если это популярный публичный VPN, таких пользователей может быть несколько, но их не может быть много. Установить искомого человека уже не представляется сложной задачей, для этого используются стандартные практики расследования, не имеющие прямого отношения к вопросам анонимности и безопасности в сети.

### С помощью цифрового отпечатка

При посещении любых сайтов, пользователи оставляют так называемый цифровой отпечаток. Концепция отпечатка устройства связана с практической ценностью отпечатков пальцев человека. В идеале все машины имеют разное значение отпечатка (различие) и это значение никогда не поменяется (стабильность). В таком случае можно было бы однозначно определять каждую машину в сети без согласия пользователя.

На практике ни различие, ни стабильность не могут быть достигнуты в полной мере. Улучшение одного из параметров влечёт за собой ухудшение другого.

**Различие** состоит в том, что нет двух машин с одинаковым отпечатком. Тем не менее большое количество устройств, вероятно, будут иметь одинаковые настройки и тем самым иметь одинаковые отпечатки. Это особенно актуально для установленных на заводе операционных систем. Одним из средств является использование скриптовых языков, которые будут собирать бо́льшее количество параметров. Однако это повлечёт за собой нестабильность системы, так как многие параметры поменяются со временем.

**Стабильность** состоит в том, что отпечаток никогда не поменяется. Однако по определению конфигурация браузера не является гарантией. Например, один из изменяемых параметров в Cookie отображается как «вкл»/«выкл». Данное изменение может изменить и отпечаток. Таким образом следует вывод, что собирать нужно только те параметры, которые с бо́льшей уверенностью не поменяются, что повлечёт за собой уменьшение используемых параметров для сбора информации.

Методы фингерпринта бывают скрытыми и активными.

**Скрытый** фингерпринт происходит скрытым запросом на клиентскую машину. Эти методы основываются на точной классификации таких параметров клиента, как конфигурация TCP/IP, цифровой отпечаток ОС, настройки IEEE 802.11 (WiFi) и смещение времени.

**Активный** фингерпринт основывается на том, что клиент будет позволять делать запросы. Наиболее распространённым методом является установка запускаемого кода прямо на клиентскую машину. Такой код будет иметь доступ к более скрытым параметрам, таким как MAC-адрес или уникальные серийные номера оборудования.

Сбор отпечатков с клиентов (при помощи браузера) может быть сделан при помощи JavaScript или других скриптовых языков для сбора достаточно большого количества параметров. Лишь два класса пользователей сети имеют серьёзные ограничения для слежки: мобильные устройства и программы с повышенной безопасностью.

Отдельной проблемой остаётся возможность пользователя иметь на одном устройстве несколько браузеров, а тем более несколько виртуальных хостов. Поскольку у каждой из сущностей может быть свой фингерпринт, менять его можно крайне быстро, если не используется новая технология кроссбраузерного фингерпринта.

### С помощью DNS [6]

DNS- Domain Name System, или система доменных имён. Именно эта система позволяет запоминать не IP-адрес нужного сервера, а его имя, состоящее из букв и имеющее некий смысл. Основные части системы DNS - это домены, поделённые на зоны, серверы и клиенты. Домены - это адреса вида kv.by, доменные зоны - это by, com, org, net и т.д., а IP-адреса - это цифровые адреса вида 127.0.0.1, использующиеся различными интернет-протоколами.

Часто в ответ на вопрос о том, как работает система DNS, "специалисты" говорят нечто вроде: "Браузер посылает запрос, DNS сервер ищет в базе данных соответствующий IP-адрес и передаёт его обратно". Такой вариант изложения работы системы DNS является, мягко говоря, несколько упрощённым. Дело в том, что нет какого-то одного-единственного DNS-сервера, к которому напрямую обращаются все мыслимые и немыслимые клиенты. Структура DNS иерархическая: если тот DNS-сервер, которому направлен запрос клиента, не имеет никакой информации об искомом домене, то он обратится к корневому (стоящему выше по иерархии) серверу. Так, по цепочке запрос клиента может дойти до сервера, являющегося в терминах DNS авторитетным для данной доменной зоны. По мере передачи ответа сервера сверху вниз по иерархии серверов адреса сохраняются в кэше, чтобы при повторном запросе не было необходимости обращения к серверам, имеющим высокий уровень в иерархии. Серверы, которые кэшируют ответы, - это рекурсивные серверы (т.к. они рекурсивно получают адреса и сохраняют их для клиентов). Серверы высоких уровней являются не рекурсивными - если они не находят адрес, нужный клиенту, они самостоятельно не проводят опрос других серверов, а предоставляют это делать клиентам (в данном случае обычно клиентами являются серверы низкого уровня).

Для того чтобы обеспечить функционирование системы в случае сбоя какого-либо из DNS-серверов, при регистрации домена требуется указывать, как минимум, два сервера, которые будут его обслуживать.

Представление о том, что каждому домену соответствует один-единственный IP-адрес, также является неверным. Для тех доменов, на которые ложится очень большая нагрузка, используется несколько отдельных физических серверов для её равномерного распределения. Аналогично, для одного IP-адреса может существовать несколько разных символьных имён.

Провайдер знает куда обращается пользователь, что позволяет отслеживать перемещение пользователя в интернете. Несмотря на использование VPN все запросы пользователя попадут к провайдеру, и он будет знать о всех доменах которые запрашивал компьютер пользователя. Провайдер может подменить DNS ответ на подконтрольный ему сервер и получить весь трафик пользователя. VPN в этом случае бессилен, так как он перехватывает трафик фактически у конечной точки вашего туннеля. Провайдер получает доступ к двум потокам данных, до шифрования и после шифрования, что даёт ему возможность как добавлять в трафик свои данные, так и дешифровать ключ и соответственно весь трафик пользователя в VPN туннеле. Использование «соли» в данном случае может усложнить взлом, но не исключит совсем. Для того, чтобы избежать перехвата и подмены DNS трафика необходимо обеспечить два условия.

* перекрыть любой выход DNS трафика (целевой порт 53 протоколы UDP и TCP) за пределы локальных интерфейсов компьютера где работает VPN клиент.
* настроить локальный DNS ресолвер так чтобы он запрашивал адреса исключительно через VPN туннель

В большинстве случаев вопрос может решиться если на компьютере с VPN в качестве имени сервера прописать адреса публичных NS серверов, например google: 8.8.8.8 или 8.8.4.4 или яндекса 77.88.8.8 или 77.88.8.8.

### С помощью WebRTC [4]

WebRTC — это бесплатный открытый проект, который предоставляет браузерам и мобильным приложениям функции связи в реальном времени (RTC) с помощью простых API.

Он включает в себя фундаментальные строительные блоки для высококачественной связи в Интернете, такие как сетевые, аудио и видео компоненты, используемые в приложениях голосового и видеочата. Эти компоненты, реализованные в браузере, могут быть доступны через JavaScript API, что позволяет разработчикам легко реализовать собственное веб-приложение RTC.

STUN / ICE

Компонент, позволяющий вызовам использовать механизмы STUN и ICE для установления соединений в различных типах сетей. Сервер STUN отправляет пинг обратно, содержащий IP-адрес и порт клиента

Эти серверы STUN (Session Traversal Utilities for NAT) используются VPN для преобразования локального домашнего IP-адреса в новый общедоступный IP-адрес и наоборот . Для этого во время подключения сервер STUN поддерживает таблицу как общедоступного IP-адреса на основе VPN, так и вашего локального («реального») IP-адреса (домашние маршрутизаторы реплицируют аналогичную функцию при преобразовании частных IP-адресов в общедоступные и обратно).

WebRTC позволяет отправлять запросы на серверы STUN, которые возвращают «скрытый» домашний IP-адрес, а также адреса локальной сети для системы, которую использует пользователь.

К результатам запросов можно получить доступ с помощью JavaScript, но поскольку они сделаны вне обычной процедуры запроса XML / HTTP, они не отображаются в консоли разработчика.

Единственное требование для этой техники деанонимизации - поддержка WebRTC и JavaScript из браузера.

Эта функциональность может также использоваться для деанонимизации и отслеживания пользователей за общими службами защиты конфиденциальности, такими как: VPN, SOCKS Proxy, HTTP Proxy и, в прошлом, пользователи TOR.

Даже если вы подключены к VPN, все пакеты отправляются вашему провайдеру, который затем перенаправляет их в VPN. Таким образом, даже когда вы подключены к VPN, ваш компьютер все еще общается с вашим провайдером.

Поскольку этот эксплойт работает на вашем компьютере (в браузере с JavaScript), он узнает ваш реальный IP-адрес, который вам дает ваш провайдер.

Браузеры, в которых WebRTC включен по умолчанию:

* Brave
* Edge
* Epiphany (Gnome)
* Mozilla Firefox
* Google Chrome
* Google Chrome on Android
* Internet (Samsung Browser)
* Opera
* Safari
* Vivaldi

# Глава 2. Практическая часть.

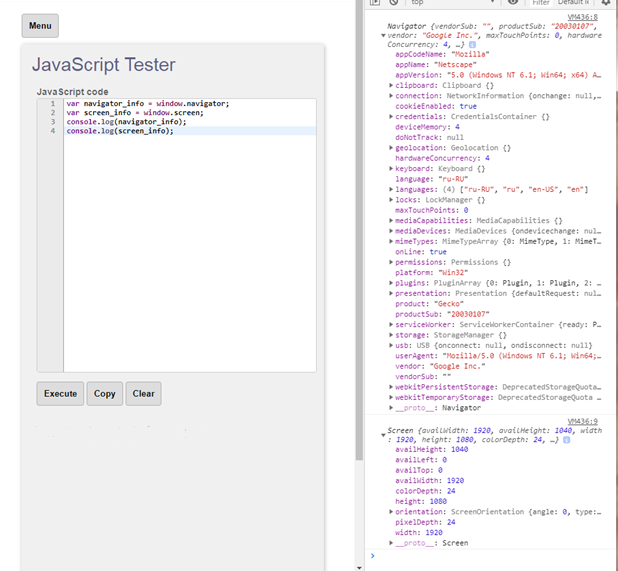
## Простой пример скрипта, для составления цифрового отпечатка:

var navigator\_info = window.navigator;

var screen\_info = window.screen;

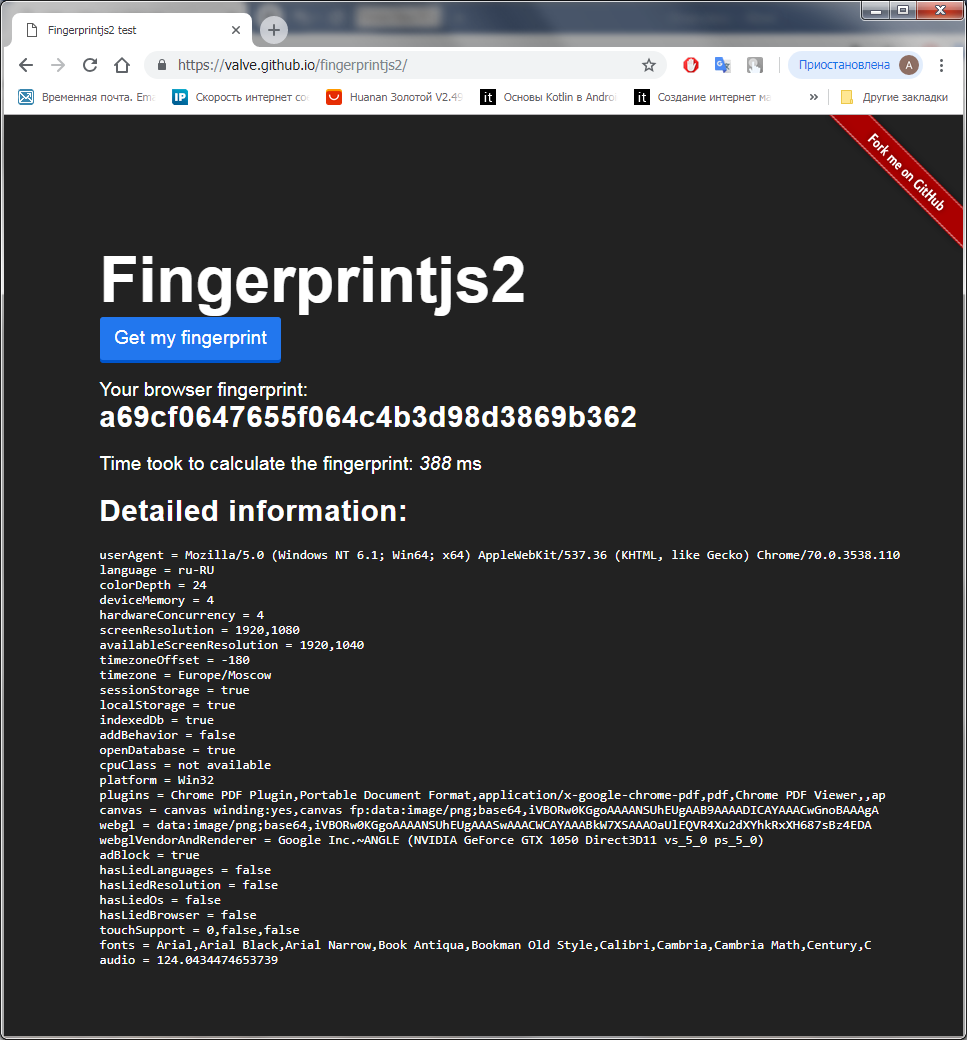
console.log(navigator\_info);

console.log(screen\_info);



*Рисунок 2. Демонстрация работы скрипта.*

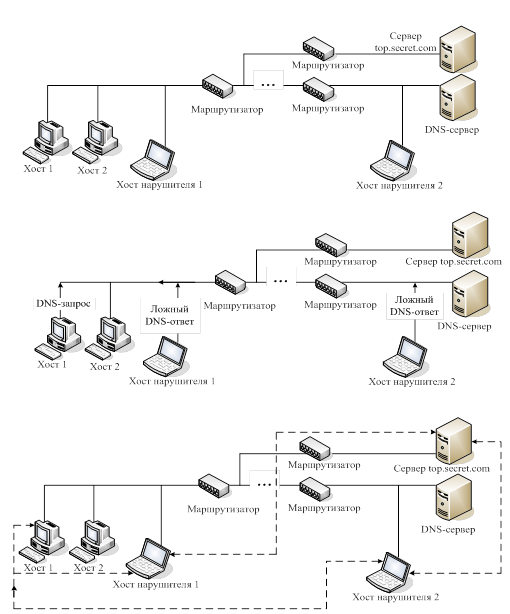
### Еще один пример:



*Рисунок 3.* *Результат работы одного из многих сайтов [3], составляющих уникальный цифровой отпечаток.*

## Подмена DNS-ответа; внедрение ложного DNS-сервера [6].

Схема реализации атаки в случае, если есть доступ к трафику жертвы (случай 1 на рисунке 4):



*Рисунок 4. Различные варианты реализации атаки.*

Атакующий ждёт DNS-запроса от жертвы на хосте 1 (атакующий находится либо на хосте нарушителя 1, либо на хосте нарушителя 2; но может быть и где-либо ещё, где есть доступ к трафику хоста 1 (жертвы)).

После передачи хостом 1 DNS-запроса, атакующий принимает запрос, в котором запоминает ID и порт. Далее, такакующий отправляет ложный DNS-ответ, в котором подменяет поле IP-адрес DNS-сервера на свой IP, делая свой компьютер для жертвы валидным DNS-сервером.

Хост 1 принимает ложный DNS-ответ, принимает IP-адрес хакера за подлинный DNS-сервер и отправляет все последующие запросы ему.

Атакующий после получения DNS-запросов пересылает их на настоящий DNS-сервер, получает правильный ответ и пересылает его назад – жертве. Существует лёгкая возможность подменить в DNS-ответе IP любого запрашиваемого DNS-имени.

Схема реализации атаки в случае, если доступа к трафику жертвы нет:

Атакующий не дожидается DNS-запроса (он его и не получит, ибо траффик жертвы через него не проходит), а отправляет массированный поток ложных DNS-ответов, подбирая на ходу нужные порт и ID запроса (обыкновенным bruteforce-ом). При этом в ложном ответе атакующий подменяет поле IP-адрес DNS-сервера на свой IP, делая свой компьютер для жертвы валидным DNS-сервером.

Хост 1, отправив запрос, принимает ложный DNS-ответ, в котором указан IP-адрес хакера, как IP подлинного DNS-сервера. В итоге, жертва отправляет все последующие DNS-запросы злоумышленнику.

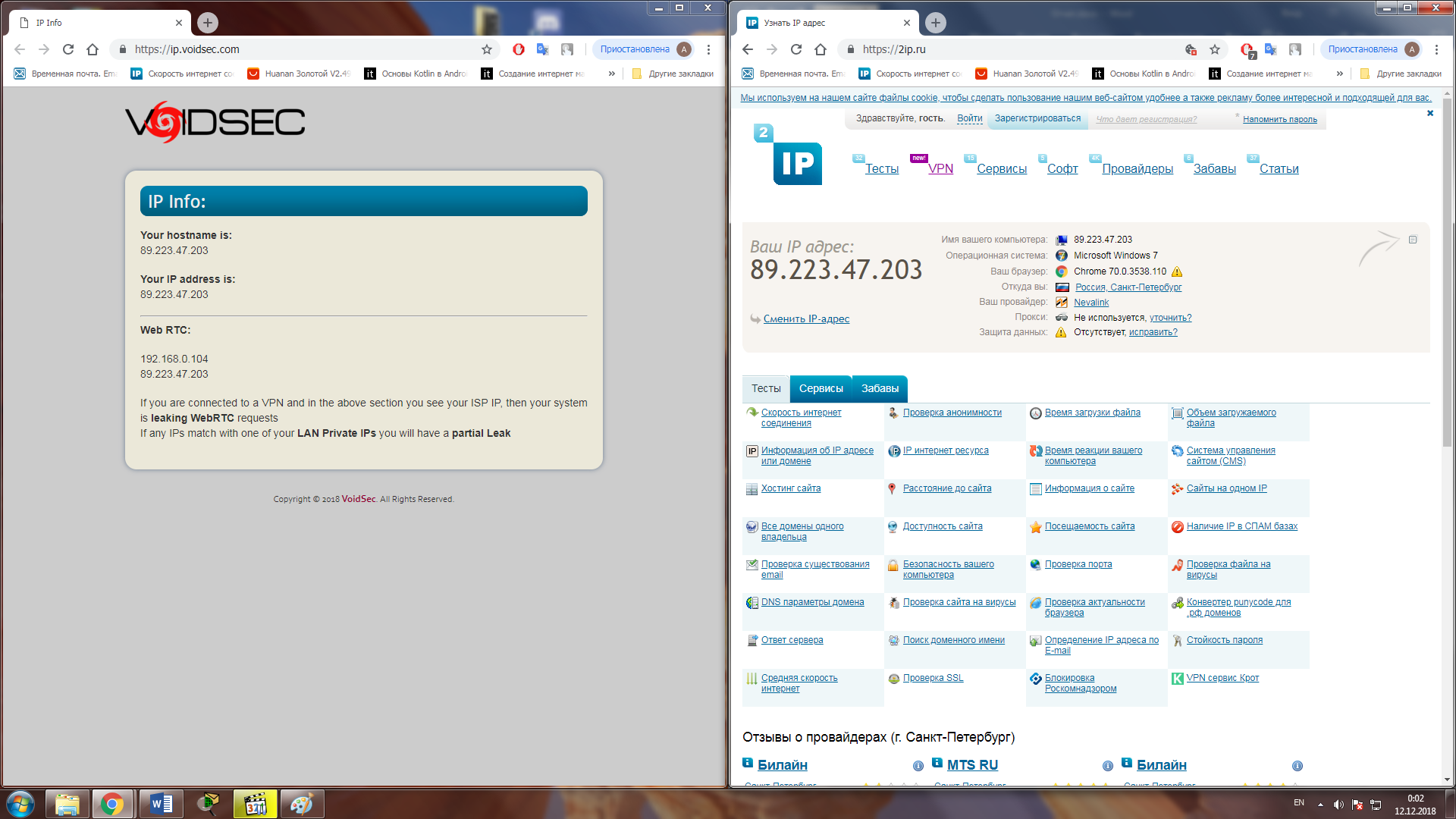
Атакующий после получения DNS-запросов пересылает их на настоящий DNS-сервер, получает правильный ответ и пересылает его назад – жертве. Существует лёгкая возможность подменить в DNS-ответе IP любого запрашиваемого DNS-имени.

В случае, если атакующий находится за отдельным маршрутизатором и не имеет доступа к трафику клиента, но находится в том же сегменте сети, что и DNS-сервер жертвы (сама жертва), схема остаётся почти тай же. Фаза 1 заменяется на фазу массированной отправки DNS-ответов, не дожидаясь запроса жертвы. В этом случае, жертва после запроса моментально получит ответ, один из которых окажется правильным.

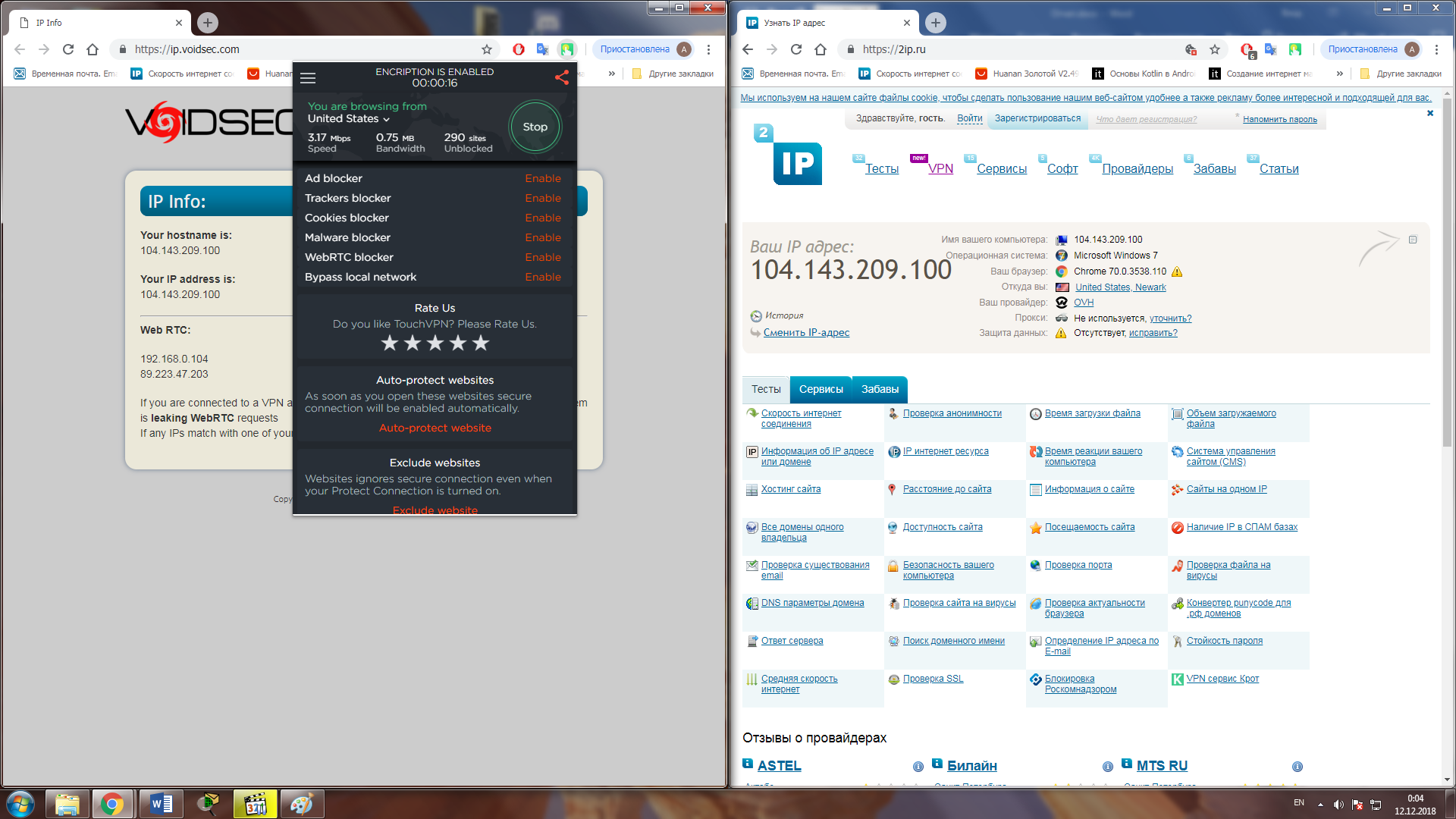
Стоит отметить, что данная угроза основана на использовании недостатков алгоритмов удаленного поиска. В случае, если объекты сети изначально не имеют адресной информации друг о друге, используются различные протоколы удаленного поиска (например, SAP в сетях Novell NetWare; ARP, DNS, WINS в сетях со стеком протоколов TCP/IP), заключающиеся в передаче по сети специальных запросов и получении на них ответов с искомой информацией.

## Использование WebRTC

Технология позволяет устанавливать соединения между двумя браузерам, в обход web-сервера. Если соединение с сервером осуществляется через VPN и сервер видит только внешний IP клиента [7], а не реальный, то WebRTC + немного кода на JavaScript позволяют узнать реальный адрес [5].



*Рисунок 5. Без включенного VPN.*



*Рисунок 6. При включенном VPN*

### Простой пример кода, который находится в открытом доступе, и позволяет использовать данную уязвимость на своем сайте [4].

// получаем IP-адреса, связанные с учетной записью

function getIPs(callback){

var ip\_dups = {};

// совместимость для Firefox и Chrome

var RTCPeerConnection = window.RTCPeerConnection

|| window.mozRTCPeerConnection

|| window.webkitRTCPeerConnection;

var useWebKit = !!window.webkitRTCPeerConnection;

// обходим блокировку webrtc с помощью iframe

if(!RTCPeerConnection){

//iframe должен быть на странице прямо над тегом script:

//<iframe id="iframe" sandbox="allow-same-origin" style="display: none"></iframe>

//<script>...getIPs called in here...

var win = iframe.contentWindow;

RTCPeerConnection = win.RTCPeerConnection

|| win.mozRTCPeerConnection

|| win.webkitRTCPeerConnection;

useWebKit = !!win.webkitRTCPeerConnection;

}

// минимальные требования для подключения к данным

var mediaConstraints = {

optional: [{RtpDataChannels: true}]

};

var servers = {iceServers: [{urls: "stun:stun.services.mozilla.com"}]};

// создаем новое соединение RTCPeerConnection

var pc = new RTCPeerConnection(servers, mediaConstraints);

function handleCandidate(candidate){

//match just the IP address

var ip\_regex = /([0-9]{1,3}(\.[0-9]{1,3}){3}|[a-f0-9]{1,4}(:[a-f0-9]{1,4}){7})/

var ip\_addr = ip\_regex.exec(candidate)[1];

// удаляем дубликаты

if(ip\_dups[ip\_addr] === undefined)

callback(ip\_addr);

ip\_dups[ip\_addr] = true;

}

// слушаем события

pc.onicecandidate = function(ice){

// пропускаем события, не являющиеся “кандидатами”

if(ice.candidate)

handleCandidate(ice.candidate.candidate);

};

// создаем фиктивный канал данных

pc.createDataChannel("");

//создаем offer sdp

pc.createOffer(function(result){

// вызываем запрос на stun сервера

pc.setLocalDescription(result, function(){}, function(){});

}, function(){});

// ждем некоторое время

setTimeout(function(){

//read candidate info from local description

var lines = pc.localDescription.sdp.split('\n');

lines.forEach(function(line){

if(line.indexOf('a=candidate:') === 0)

handleCandidate(line);

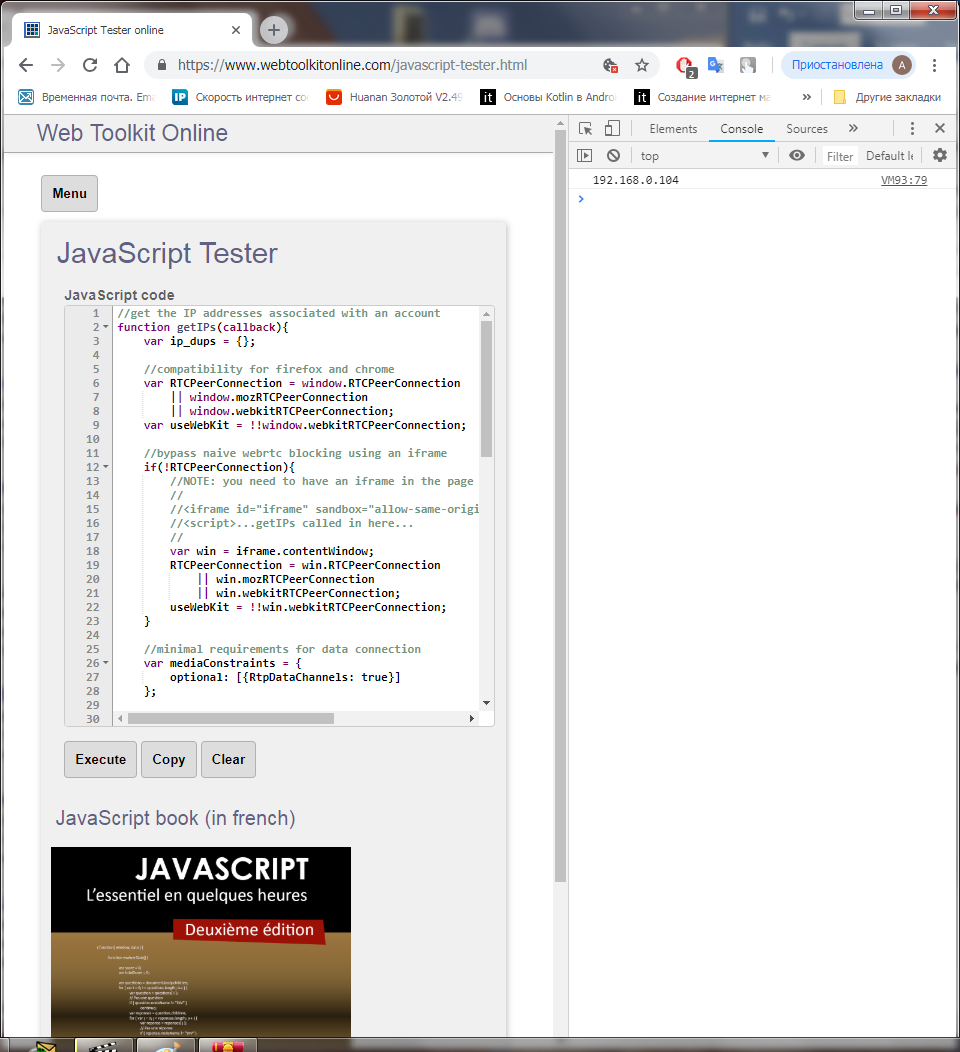
});

}, 1000);

}

// Выводим IP-адреса в консоль

getIPs(function(ip){console.log(ip);});



*Рисунок 7. Результат работы скрипта.*

# Заключение.

По результатам проделанной работы можно сделать выводы:

1. Нужно стараться соблюдать анонимность в сети.

Это необходимо не только в том случае, если пользователь старается скрыть свою активность, но также для того, чтобы как можно меньше рекламных компаний имели информацию о пользователе.

1. VPN сети помогают оставаться анонимным.

Сокрытие настоящего ip-адреса, а также частая смена ip-адресов затрудняет распознание пользователя.

1. VPN сети не гарантируют полную анонимность.

Если использовать один и тот же ip адрес, даже не являющийся настоящим, довольно просто опознать пользователя. Кроме того, существуют методы распознания реального ip- адреса.

1. Существует достаточно большое количество методов деанонимизации в сетях VPN и в интернете в целом.

Некоторые из них описаны в данно й работе. Однако существуют методы, информацию о которых мне не удалось найти.

Метод составления цифрового отпечатка реализуется достаточно просто, а значит его могут использовать многие сайты. Однако же минусом этого метода является то, что при стандартных настройках браузера, отсутствии расширений, использование популярной ОС и тд, цифровой отпечаток будет не очень точным. А, при использовании телефона, идентифицировать конкретного пользователя будет почти невозможно.

Второй несложный в реализации метод - использование WebRTC. Этот метод хорош тем, что позволяет определит реальный ip-адрес пользователя. И в отличие от цифрового отпечатка, реальный ip-адрес позволяет идентифицировать пользователей устройств любой конфигурации.

Но самый действенный, на мой взгляд, метод - подмены DNS-ответа, поскольку данным способом можно адресовать пакеты пользователя на любой другой сайт, который сможет собирать любую необходимую информацию. Однако он же и самый сложный в реализации.

# Список использованной литературы

1. VPN [Электронный ресурс]. -

<https://ru.wikipedia.org/wiki/VPN>

1. Парфинчук, Е.Б. VPN и IPSec на пальцах [Электронный ресурс]. -

<http://www.nestor.minsk.by/sr/2005/03/050315.html>

1. Васильев, В.А. Fingerprintjs2 [Электронный ресурс]. -

<https://valve.github.io/fingerprintjs2/>

1. Daniel Roesler. STUN IP Address requests for WebRTC [Электронный ресурс]. –

<http://diafygi.github.io/webrtc-ips>

1. Voidsec [Электронный ресурс]. –

<https://ip.voidsec.com/>

1. Лысяк, А.С. DNS-атаки: полный обзор по схемам атак [Электронный ресурс]. –

<http://inforsec.ru/technical-security/network-security/77-dns-attack>

1. Тест IP адреса [Электронный ресурс]. -

<https://2ip.ru/>