

ПОНЯТЬ ИНТЕРНЕТ РАЗ И НАВСЕГДА

МЯТНЫЙ ЧАЙ

ИЗДАНИЕ
ЖУРНАЛА
UNCERTAINTY

ОТ

ОБ АВТОРЕ

Мятный Чай – главный редактор журнала Uncertainty, который увлекается программированием и различными концепциями, связанными с децентрализацией и обеспечением приватного общения в интернете.

ОТ АВТОРА

На момент написания эта статья – мой magnum opus. Над ее идеей я думал много лет, изучал множество различных материалов и долго экспериментировал, чтобы наконец прийти до такого этапа, когда я могу объяснять интернет другим.

Если эта статья показалась вам полезной, пожалуйста, поделитесь ей в вашем предпочтительном канале связи с другими людьми.

Критика, замечания и другие комментарии по этой статье принимаются по контактам на сайте minttea17.github.io

Отправить пожертвование автору можно в криптовалюте.

Bitcoin:

bc1qpnmsa024ms9enff7ac3l08csn9c80fqpf9mhue

Monero:

89f35CGTtoMaLYWwyjXQ1XF8x5skTZvkpRUywmJrQpzU4721otgJ
foR3muP1BbHLet5zjGB3jn6EsXpv9eKTZqQe8NfwfbC

—НАЧАЛО—

В 1969 году на свет появилась сеть ARPANET, разработанная Агентством Министерства обороны США по перспективным исследованиям.

Немногим позже, в по-своему ироничном 1984 году появилась и сеть NSFNet, основанная на наработках предшественницы, разработанная Национальным научным фондом США. Именно ей и суждено было стать по-настоящему массовой.

Интернету не так много лет. Ещё меньше он находится в руках общества. Но уже сегодня дети чуть ли не рождаются с телефонами в руках, потребляя бесчисленное количество информации каждый день. Интернет – это мощный инструмент. Но в последнее время он не кажется таким приятным местом, каким был в свои ранние годы.

На Планете повсюду развит институт интернет-цензуры, государства как бы делят интернет границами. Сегодня очень сложно свободно использовать всемирную паутину, особенно, если, по мнению некоторых регионов, вы, мой дорогой читатель, паспортом не вышли.

В этой статье я хотел бы без лишних технических подробностей, но с упором на техническую базу объяснить обычным людям, как работает современный интернет, что с ним не так и призвать читателя бороться за своё право на приватность с помощью современных технологий. Но вместе с этим я также хочу рассказать о подводных камнях предложенных технологий, поделиться накопленным багажом знаний по теме.

Считаю себя в известной мере компетентным в этом вопросе, потому что уже много лет изучаю всё, что связано с распределёнными сетями и различными концепциями, которые предполагают децентрализацию.

Не будем утруждать дорогого читателя, переходим к сути.

КАК РАБОТАЕТ ИНТЕРНЕТ СЕГОДНЯ?

СКАЗОЧКА ОБ АССИМЕТРИЧНОМ ШИФРОВАНИИ

Жила была Алиса. У неё был друг Боб. У Алисы есть мама по имени Ева.

У Алисы дома была большая банка варенья. Но Алисе запрещалось давать ее кому-то еще, а Боб варенье очень любил.

Один раз Алиса все-таки осмелилась поделиться с Бобом вареньем и написала ему письмо, приглашая к себе домой, когда мама будет на работе. Отправив его через почтовый ящик, она начала томительно ждать.

Но к сожалению Алиса не учла, что мама умеет читать и иногда заглядывает в почтовый ящик. Она узнала о намерениях Алисы и запретила ей звать Боба кушать варенье.

Алиса подумала и решила, что раз ей нельзя звать Боба кушать варенье, то они с Бобом могут придумать специальный язык (который криптографы называют ключом), который позволит им общаться так, чтобы мама ни о чем не догадалась. Например, фраза "Я люблю ландыши" будет означать, что сегодня можно прийти за вареньем. Идея отличная. Так думали и древние криптографы. Они шифровали свои послания симметричными алгоритмами, то есть такими алгоритмами, которые предполагают, что будет использован один ключ как для шифрования, так и для дешифровки сообщений.



Проблема только в том, что этот ключ надо как-то передать. А мы помним, что мама Алисы читает письма в почтовом ящике. Нельзя просто написать: "Боб, вот это теперь будет нашим ключом", потому что в таком случае Ева сможет дешифровать любое написанное ребятами письмо.

Да, задачка... Но к счастью ребят, умные криптографы придумали алгоритмы ассиметричного шифрования, о которых Алиса читала в книжках.

Если объяснять на высоком уровне, то клиент с помощью тяжелой математики генерирует приватный ключ. Его, как следует из названия, он никому не передает.

На основе приватного ключа, он генерирует публичный ключ по общеизвестному алгоритму. Алгоритм должен быть таким, чтобы, **зная только публичный ключ, невозможно было получить приватный** за разумный срок.

Публичный ключ используется для шифрования сообщений. Приватный ключ может дешифровать сообщения, зашифрованные для соответствующего ему публичного ключа.

То есть, например, Алиса генерирует пару ключей и отправляет свой публичный ключ Бобу. Боб шифрует сообщение этим публичным ключом Алисы и отправляет ей шифротекст. Алиса дешифрует шифротекст Боба своим приватным ключом и читает посланное ей сообщение.



На практике это означает, что имеющимися у человечества ресурсами, невозможно получить приватный ключ из публичного. В одной из самых популярных криптосистем RSA для реализации такого алгоритма используется принцип чрезвычайной вычислительной сложности разложения на простые множители больших полупростых чисел.

Полупростыми числами называют такие числа, которые можно представить в виде произведения двух простых чисел (то есть чисел, отличных от одного, которые делятся только на 1 и на самих себя).

Пример: возьмем полупростое число 4, которое раскладывается как произведение двух простых чисел: двух и двух. А теперь попробуйте проделать то же с числом 1927, которое также является полупростым.

Что, не получается? То-то же!

А компьютеры работают с числами, значительно большими чем 4 или 1927, что и обеспечивает надежность шифрования.

После того, как Алиса и Боб сгенерировали пары ключей на своих компьютерах, они послали публичные их части их через почтовый ящик, а на следующий день Ева обнаружила, что банка варенья значительно опустела. Она была крайне удивлена, ведь думала, что Алиса и Боб обмениваются числами для проекта по математике. Но не тут-то было!

НЕМНОГО ПРО DNS

Итак, что происходит, когда вы заходите на ваш любимый сайт через браузер?

Например, вы заходите на `example.com`.

Сначала ваш браузер отправляет запрос к DNS-провайдеру.

Это такой орган, который занимается тем, что отдает `ip`-адрес сайта, понятный для вашего компьютера, принимая на вход понятный для человека домен.

Ip-адрес – это уникальный публичный идентификатор устройства в интернете. По нему любой компьютер может отправить вам некоторый запрос.

Главное отличие вашего `ip`-адреса, как клиента, и `ip`-адреса сервера сайта, которым вы желаете воспользоваться, заключается ровно в том, что адрес последнего чаще всего статичен.

Так происходит из-за того, что замена `ip`-адреса у DNS-провайдера занимает некоторое время, из-за особенностей работы всей схемы.

Здесь необходимо сделать некоторое лирическое отступление.

DNS – это не сервис. Это не компания и не правительственная организация.

DNS – это протокол. То есть буквально набор правил, на базе которого разработчики создают программное обеспечение, которое позволяет компьютерам общаться между собой с какой-то определенной целью. В данном случае она заключается в получении `ip`-адреса некоторого ресурса.

А также, да простят меня ещё не окрепшие умы (правда жестока), ip-адреса не совсем уникальны. Если быть точным, в большинстве случаев — совсем не уникальны. Потому что интернет не задумывался как место, к которому будет подключено всё: от смартфонов и телевизоров, до кофемолок и умных автомобилей.



Здесь я бы хотел также упомянуть, что в мире существует два стандарта протокола интернета. IPv4 (IP версии 4) и IPv6 (IP версии 6). Люди пока не договорились, какой из них следует использовать. Поэтому используются оба.

В шестой версии протокола диапазон возможных ip-адресов гораздо больше, чем в четвёртой. В шестой версии адресов хватит примерно на всю Солнечную Систему, а в четвертой их всего около 4 миллиардов. Поэтому в теории мы можем раздать всем подряд уникальные статичные ip-адреса, но на практике это не используется, что несколько удручает.

Именно этот нюанс работы сети сыграет большую роль в этой статье.

Здесь читатель может спросить: как тогда компьютеры общаются между собой без выделенных ip-адресов? Об этом мы поговорим чуть позже, в разделе о NAT.

Возвращаемся к DNS. Существует множество способов послать DNS-запрос. Традиционный крайне простой: мы посылаем незашифрованный запрос к DNS-серверу, а в ответ получаем такой же незашифрованный ответ. Это позволяет третьей стороне, прослушивающей трафик, легко узнать, на какие сайты вы хотите попасть.

Из ныне активно используемых также можно отметить DNS-over-TLS и DNS-over-https. Первый позволяет использовать шифрование по протоколу TLS (то есть то же шифрование, что обычно используется при подключении к сайтам в интернете), а второй в добавок к этому скрывает, что мы используем протокол DNS. Подключение выглядит так, будто пользователь подключается к обычному веб-сайту.

Если вы не шифруете ваш DNS-запрос, его можно подменить по дороге. Из-за современных мер по обеспечению безопасного подключения, если вы используете последние веб-стандарты, вы не сможете зайти на подменный сайт, который вам предлагают. Браузер выдаст страшное предупреждение, и вы закроете страницу.

Но это ничуть не мешает блокировать определенные сайты в интернете. Сейчас такой метод не является основным, но тем не менее, иногда продолжает использоваться.

Следует также отметить (не отказывая себе в удовольствии в очередной раз ломать детское представление о добром и прекрасном мире), что большая часть сайтов в интернете пользуется услугами различного рода компаний (например, Cloudflare), которые скрывают реальный ip-адрес ресурса, пропуская трафик через свои сервера, которых, разумеется, конечное количество. Запомните этот момент. Он пригодится нам в будущем.

NAT – ЭТО БОЛЬ

Так как невозможно выделить уникальный IPv4 адрес каждому компьютеру в мире, людям пришлось извернуться. Перейти на IPv6 – дорого, а делать что-то надо.

Поэтому придумали такой механизм, который объединяет множество компьютеров, выдавая им один ip-адрес. Узел NAT принимает подключения от компьютеров в локальной сети и отправляет пакеты в интернет, заменяя адрес для ответа на свой. А затем принимает ответы извне, отдавая нужные нужным пользователям по конкретному порту. В этом случае, IP-адрес – это улица и номер дома, а порт – это квартира конкретного абонента.

NAT не позволяет адекватно строить децентрализованные системы. Приходится искать компромиссы, о которых мы поговорим в главе 3.

HTTPS – БЭКДОР ИЛИ ДОБРОДЕТЕЛЬ?

С DNS разобрались. Теперь у нас на руках есть ip-адрес ресурса, к которому мы ходим обратиться.

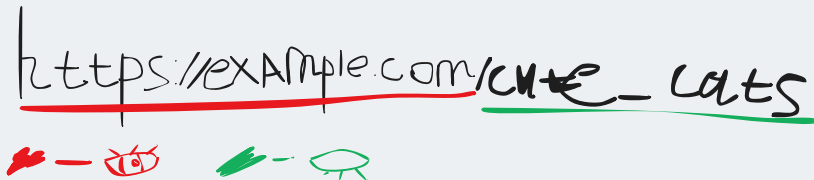
Раньше сайты использовали протокол http. Он работает так же просто, как и незашифрованный DNS, отправляя запрос до определенного веб-ресурса в нешифрованном виде, и получая такого же рода ответ. Это позволяет злоумышленнику перехватить ваш трафик и потом весело продавать данные вашей кредитки на злых и страшных торговых площадках. Или просто наблюдать, на какие сайты вы ходите и чем занимаетесь.

Это печально. С этим надо было что-то сделать, и в 1994 году Netscape (будущая Mozilla) разработает для своего одноименного браузера протокол https, использующий сначала шифрование SSL, а затем и TLS (новая версия протокола SSL с исправлением некоторых коренных уязвимостей).

Каждый раз, когда вы видите знаменитый замочек рядом с сайтом или надпись "подключение защищено", вы используете https.

TLS устанавливает подключение между пользователем и сервером с помощью уже знакомого нам асимметричного шифрования, а затем уже по зашифрованному каналу обменивается с сервером необходимыми данными для генерации симметричного ключа, который уже будет использован для шифрования данных (это делается из-за того, что симметричное шифрование работает гораздо быстрее и требует меньше вычислительных мощностей).

В такой системе сторонний наблюдатель сможет увидеть лишь адрес сайта, на который мы направляемся, но никак не содержимое, которое мы передаем. Включая информацию о том, на каких страницах сайта мы находимся и т.п.



A hand-drawn diagram illustrating the components of a URL. The text 'https://example.com' is written in black and underlined with a red line. To its right, 'chats' is written in black and underlined with a green line. Below the red underline, there is a red squiggle followed by a red eye icon. Below the green underline, there is a green squiggle followed by a green eye icon.

Но у нас есть одна проблема. Дело в том, что я не рассказал читателю об одном обстоятельстве, которое значительно усложняет всю процедуру. Давайте представим, что наблюдатель-злоумышленник не просто следит за нашим интернет-каналом, но и пытается активно вмешаться в трафик. В таком случае, при установлении безопасного канала связи через TLS, злоумышленник может подsunуть нам и нашему собеседнику свой публичный ключ.

Теперь, когда мы будем шифровать сообщение подменным ключом, злоумышленник будет дешифровать его своим приватным ключом, шифровать полученное сообщение публичным ключом получателя и отправлять полученный шифротекст ему.

Наш собеседник расшифрует шифротекст своим приватным ключом и не заметит подлога.

В итоге отправитель и получатель думают, что ведут зашифрованный сеанс связи, но это в корне не так. Человек посередине перехватывает и записывает их сообщения.

Более того, он даже может отправить свое собственное сообщение, вместо одного из клиентов.

Именно эта проблема становится ключевой в вопросах, связанных с осуществлением безопасной коммуникации. Где-то она обозначается как проблема TOFU (Trust Of First Use), где-то как то, что система уязвима перед вредоносным сервером, но суть проблемы одна: нам нужен надежный контрагент, который подтвердит, что ключ, который мы получили, действительно является подлинным.

Эту роль берут на себя Центры Сертификации (ЦС), которые расположены по всему миру и работают независимо друг от друга.

Они предлагают установить на ваше устройство их корневой сертификат безопасности. Обычно такие сертификаты устанавливают производители оборудования, но вы можете установить подобный сертификат самостоятельно и даже сгенерировать свой собственный, став ЦС для сайтов своих друзей.

ЦС не выполняет ровным счетом никаких полезных функций, кроме одной: подписывать публичные ключи веб-сайтов.

Владелец сайта обращается в ЦС, отправляя свой публичный ключ с просьбой на подпись. ЦС криптографически подписывает его ключ, используя как основу свой сертификат безопасности, копия которого есть на устройстве пользователя.

Затем ЦС сохраняет новый сертификат для конкретного сайта, позволяя вашему браузеру проверить, доверяет ли предпочтительный ему ЦС этому сайту.

Если да, подключение пройдет. Если нет, браузер выдаст ошибку.

На вопрос о том, кто решает, какие ЦС заслуживают доверия, а какие нет, напомним дорогому читателю, что на свободном рынке нет актива более ценного, чем репутация. Вы никогда не будете помогать с делами тому, кто не кажется вам достаточно порядочным, чтобы в дальнейшем честно оказать вам какую-то дружескую услугу.

Но есть нюанс. Представьте себе вредоносный центр сертификации. Например, контролируемый преступниками или правительством. Например, практику по обязательной установке корневого сертификата от властей практиковали в Казахстане и Беларуси (во время народных волнений). Россия тоже не отстает. Минцифры с 2022 года имеет свой ЦС, выпускающий сертификаты для российских компаний.

В случае с Казахстаном крупные компании, владеющие браузерами Google Chrome и Firefox, просто отказались доверять корневому сертификату от властей, даже если он отмечен в системе как доверенный. И правильно.

Допустим, вредоносный ЦС решит понаблюдать за трафиком пользователей, которые установили его корневой сертификат в свою систему. Дружественный такому ЦС провайдер может совершить атаку MITM (человек посередине), если ЦС подпишет его ключ, вместо подлинного, который предлагает сайт.

Пользователь в теории не заметит подлога, а злоумышленник сможет читать всё содержимое зашифрованных https страниц как открытую книгу. На практике в браузере, скорее всего, отобразится уведомление о том, что что-то идет не так.

Поэтому за https нужно тщательно следить и не забывать его истинную природу. Это протокол, который принципиально основан на доверии к мастодонтам.

А вы им доверяете?

2 ВВЕДЕНИЕ

Я волком бы выгрыз бюрократизм.

К мандатам почтения нету.

К любым чертям с матерями катись любая бумажка.

Но эту...

Я достану из широких штанин дубликатом бесценного груза.

*Читайте, завидуйте, я – гражданин Советского Союза.
(Владимир Маяковский – Стихи о советском паспорте)*

Представьте себе государство. Внушительная машина, которая вершит справедливый суд, издаёт законы, защищает себя и своих жителей от нападок врагов и пользуется немыслимым авторитетом.

Такая конструкция нуждается в системе сдержек и противовесов. Ведь с большой властью приходит и большая ответственность. Или, как говорил классик: "Власть развращает, абсолютная власть развращает абсолютно".

Государство принимает законы. Иногда законы ограничивают то, какие мысли общество может выражать. Так появляется институт цензуры. Это такой механизм, который применяет к людям методы легального государственного насилия для принуждения их к какому-то образу мысли. Когда вы запускаете такой механизм, который превращает осуждённых за слова людей в галочки, звездочки и доллары, это уже не остановить. И рано или поздно его начнут использовать против общества, даже если изначально цели были весьма благородными.

ГЛАВА ДАМОКЛОВ МЕЧ

Государство не умеет быть милосердным. Оно умеет быть эффективным, если ему дать для этого инструменты. Такие инструменты нужно держать в нескольких руках, а всю полноту власти не давать решительно никому.

А теперь вспомните, что мы говорим про интернет. Про глобальную распределенную сеть. Подумайте, что будет, если попытаться поделить ее границами?

ОНИ НАС РАЗДЕЛИЛИ.

В наше время, используя интернет, вы могли столкнуться с некоторыми трудностями. Некоторые веб-сайты заблокированы вашим государством, в связи с нарушением местного законодательства, некоторые заблокировали вас, в связи с тем, в каком государстве вы находитесь.

Не просто так в последние пару лет люди так полюбили различные решения, которые позволяют скрыть истинный ip-адрес. Именно они позволили им преодолеть такого рода ограничения.

Такие ограничения в большинстве своём не являются частной дискриминацией. Их не диктует рынок. Их диктуют сверху вниз.

Глобально, на физическом уровне, интернет – это кабель. Государства, занимающиеся интернет-цензурой, буквально запрещают людям отправлять электрические сигналы из точки А в точку В.

Это страшно. Принцип сетевого нейтралитета, который действовал так долго, окончательно разрушен. Интернет перестал быть глобальным. Люди всё больше замыкаются в своём информационном вакууме строго определенных источников и сюжетов.

О том, как именно государства это сделали, и как люди противодействуют цензуре, в следующей главе.

Мы не рассматриваем США и страны, придерживающиеся подобной модели интернет-цензуры. Чаще всего они не вводят списки запрещенных сайтов, а действуют юридическими методами. В этой статье мы рассматриваем исключительно техническую сторону вопроса там, где используется грубая сила.

Это не означает, что такие государства безгрешны. Не имеет значения, блокируете ли вы сайты Яндекса и ВКонтакте (Украина), запрещаете своим жителям видеть правду о работе вашего правительства (Wikileaks и США) или штрафуете людей за скачивание пиратских фильмов (Германия и другие страны Европы). Не бывает безгрешных государств. Но государства, не использующие технические средства блокировок, не входят в поле нашего интереса в рамках этой статьи. Этим должны заниматься юристы.

ГЛАВА 3 ЧТО СЛУЧИЛОСЬ?

Во имя исполнения положений местного законодательства, хочу отметить, что в этой статье мы обсуждаем технические реализации подобных инструментов, а автор не создает позитивного образа их использования. Мы лишь рассуждаем о том, что и как используют люди. Спасибо.

DPI, ECH, КГБ

Сначала различные государства взяли на вооружение простой принцип, который некоторые из них продолжают использовать до сих пор.

Интернет-сервисы чаще всего используют статические IP-адреса. Можно просто спустить на провайдеров обязанность блокировать подключения до различных IP-адресов. Ничего сложного.

Проблемы начинаются тогда, когда мы вспоминаем, что у нас повсеместно распространен протокол интернета версии 4, в котором преступно мал диапазон возможных IP-адресов. Сервисы в интернете размещаются на серверах, у которых есть такие адреса, но проблема в том, что на одном сервере может быть расположена инфраструктура множества сервисов, которые вы тоже случайно можете заблокировать.

Именно это мы наблюдали в 2018 году, когда Роскомнадзор пытался заблокировать Telegram. Тогда легли Google, YouTube, большие CDN и другие популярные платформы.

Более того, такая технология категорически неприменима для блокировки сайтов. Вы же еще помните, что многие сайты скрывают свой реальный IP-адрес, заменяя его на адрес от Cloudflare или других подобных служб? Так вот, если вы начнете блокировать Cloudflare, то упадет пол-интернета. Никому это не надо.

Поэтому в большинстве случаев такой метод больше не используется.

На его место пришел DPI. Это такая система анализа пакетов передаваемых данных, которая позволяет узнать больше информации о том, куда и какой идет запрос.

DPI позволяет разгадать тип протокола, используемого для передачи данных и даже получить домен из зашифрованного https трафика (потому что домен, как упоминалось выше, передается в открытом виде).

"А почему вообще домен сайта передается в открытом виде?" – может спросить интересующийся читатель. Отвечаю мнимому читателю на его мнимый вопрос. Для этого придется сделать очередное лирическое отступление на тему того, как работает протокол https.

Как мы все уже знаем, изначально сайты работали по протоколу http, который не использовал шифрование TLS для защиты пользовательских данных. Это было грустное время, но сейчас большая часть интернет трафика зашифрована.

Для установления безопасного соединения через TLS клиенту и серверу необходимо изначально обменяться некоторыми данными, договориться о том, какую версию протокола использовать и прочее. Этот процесс называется рукопожатием.

Проблема тут состоит в другом. Для того, чтобы изначально развернуть шифрование, необходимо было немного сыграть в сделку с совестью. Дело в том, что раньше ЦС выдавали сертификаты одному сайту на один ip-адрес. Никто и не думал, что количество сайтов на одном сервере будет множиться.

Но время пришло. В 2003 появляется спецификация SNI (Server Name Indication), браузеры внедряют поддержку в течение нескольких следующих лет. SNI отправляет адрес сайта на сервер в незашифрованном виде, еще до того, как произошел обмен ключами.

Комментарий CEO Cloudflare Мэттью Принса по этому поводу: *"В то время это казалось приемлемым компромиссом. Большинство интернет-трафика было не зашифровано. Добавление расширения TLS, которое упрощало поддержку шифрования, казалось отличной сделкой, даже если само расширение не было зашифровано"*.

Но в наши дни все начали понимать, что с этим надо что-то делать. Появляются первые идеи нового стандарта. В 2019 году опубликован первый черновик стандарта Encrypted Client Hello (ранее – ESNI), который позволяет шифровать имя сайта при рукопожатии.

ESN на данный момент поддерживается всеми популярными браузерами, но до массового внедрения ещё далеко. Необходимо дать интернету время на реформуляцию. По информации с форума ntc.party ECH открывает доступ к некоторым сайтам, но крайне избирательно, и эта тема пока никем досканально не изучалась.

Минцифры России еще в 2020 году предлагало запретить стандарт законодательно.

DPI позволяет анализировать паттерны различных протоколов и блокировать их. Так, например, делают в Китае, блокируя популярные протоколы VPN и прокси. Но устойчивые к цензуре решения вроде Shadowsocks продолжают работать.

Внимательный читатель может заметить, что раз ECH решает проблему блокировки сайтов по доменному имени, то почему бы все-таки не продолжить блокировать их по ip-адресам, если это имеет крайнюю необходимость? Пусть владельцы законных сайтов на том же сервере сами разбираются с хостингом.

Вариант замечательный. Но в случае с большими CDN-сетями он не работает. Cloudflare сделало крайне много для создания стандарта ECH именно потому, что они знают: власть в их руках. Никто не может заблокировать Cloudflare, не заблокировав половину интернета. Урон от блокировки ip-адресов Cloudflare слишком велик для того, чтобы заблокировать один несчастный ресурс. Поэтому ECH – это главное оружие обычных пользователей против цензуры.

НАСКОЛЬКО ГЛУБОКА КРОЛИЧЬЯ НОРА?

Вы наверняка слышали про VPN. Это такая технология, которая позволяет делать три основные вещи:

- Объединять устройства в локальную сеть через интернет, выдавая им понятные локальные ip-адреса.
- Выходить в интернет через главный объединяющий сервер, скрывая свой ip-адрес.
- Шифровать трафик по дороге.

VPN чаще всего обычный человек использует именно из-за второго и третьего пунктов. Люди используют VPN для обхода региональных ограничений и государственной цензуры.

Например, читатель может помнить сервис WARP от Cloudflare. Он использовал ноды компании для перенаправления вашего трафика с учетом оптимальной локации сервера. При этом WARP предоставлял полноценный обход блокировок с выводом трафика в более свободную страну. По сути WARP позиционирует себя уже не просто как VPN сервис. Cloudflare заявляют, что WARP помогает починить интернет. То, чего мы хотим добиться в конце этой статьи.

Cloudflare действительно сделали хороший продукт, с которым интернет было использовать намного приятнее. Но это не решает те проблемы, с которыми сталкивается интернет, до конца.

Видите ли, мы не можем просто взять и починить интернет, отдав, пускай даже и доверенной компании, как Cloudflare, обязанности по шифрованию и маршрутизации нашего трафика.

Это исправляет симптомы, но лишает нас контроля и не исправляет саму проблему.

Интернет невозможно исправить до конца. Но у нас есть способы минимизировать те проблемы, которые составляют его дурную природу. Придется работать с тем, что у нас есть, хотим мы того или нет. Самое главное, на что эта статья и направлена, так это на то, что необходимо трезво оценивать, какой уровень свободы тебе даёт та или иная технология.

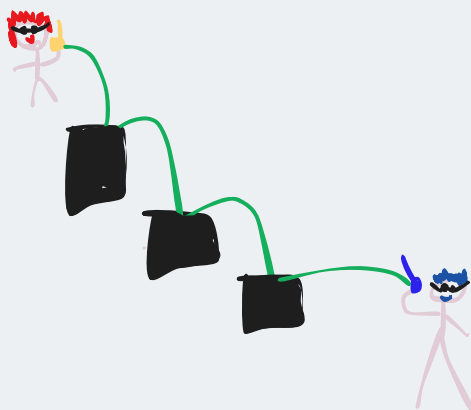
Интернет сломан до основания. И его пора (за)менять.

Дорогой читатель, если к этому моменту вы разочаровались в современном интернете, значит, я хорошо делаю свою работу. Я рад. Но это ещё не всё. Если вы верите в существование волшебных решений, которые могут всё изменить, то в следующем разделе я развенчаю этот опасный миф. На интернет нужно смотреть трезво. Этому я вас и учу.

ЛУКОВИЦЫ И СМЕЛЫЕ ЭКСПЕРИМЕНТЫ

Tor представляется многим как панацея от всех болезней интернета. Таких людей я делю на две основные категории:

- Люди, которые знают о нерешаемых проблемах, но считают, что это данность интернета и иначе нельзя.
- Люди, которые не до конца понимают, как работает сеть Tor.



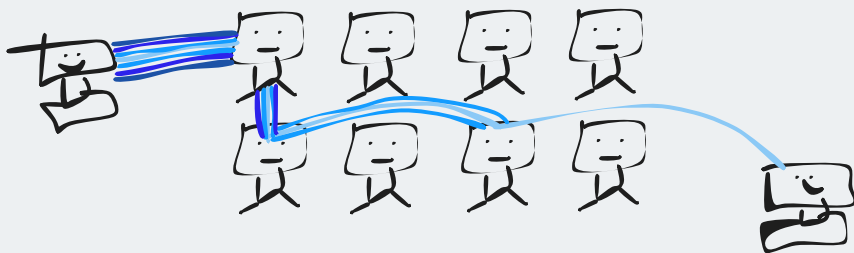
Сеть Tor разработана в 1995 году Исследовательской лабораторией Военно-морских сил США, совместно с управлением перспективных исследовательских проектов Министерства обороны США (DARPA). В общем, примерно те же люди, что в своё время создали сам интернет.

Tor работает на основе луковичной маршрутизации. Волонтеры поднимают ноды Tor – по сути своей компьютеры, обычно в дата-центрах, которые маршрутизируют через себя трафик пользователей.

Когда вы отправляете запрос к сайту, используя браузер Tor, вы шифруете его три раза и перенаправляете через три случайных ноды из списка доступных.

Первая (входная) нода Tor знает, кто вы (ваш ip-адрес), но не знает ваш трафик. Первый слой шифрования снимается и пакет отправляется второй (промежуточной) ноде. Она не знает, кто вы, но знает ip-адреса входной и выходной нод, трафик она тоже не знает.

Промежуточная нода снимает второй слой шифрования и отправляет пакет третьей (выходной) ноде. Она снимает последний слой шифрования и отправляет трафик в интернет.



Надо понимать, что нод в сети на текущий момент всего около 7 тысяч. Это связано с тем, что Tor Project не заставляют вас становиться ретранслятором при обычном сценарии использования клиента или браузера Tor. На самом деле, это скорее плохо, потому что если бы все (или хотя бы большая часть) устройств могла быть ретрансляторами, это бы увеличило приватность пользователей на порядок.

Для того, чтобы запустить ноду, вам необходим публичный IPv4 адрес. Желательно, чтобы он был статичным, но это не является обязательным требованием. Однако если ваш ip-адрес меняется более, чем раз в 3 часа, нет особого смысла запускать ноду – процесс дистрибуции ip-адресов занимает какое-то время и происходит раз в час.

Есть ещё один нюанс, который не позволяет запускать сколько угодно нод из под NAT. Разработчики Tor специально ограничивают количество нод на одном ip-адресе для защиты от атаки Сивиллы.

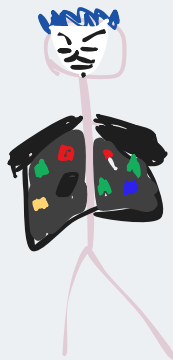
Если коротко: атака Сивиллы предполагает, что злоумышленник контролирует гигантское количество узлов сети, удерживая жертву в своеобразной "песочнице", из которой у него не получается выйти, потому что вредоносные узлы ссылаются только друг на друга. Таким образом, необходимо ограничивать злоумышленника от создания бесконечного числа нод, устанавливая определенные искусственные лимиты, но такие, чтобы легитимные пользователи могли продолжить использовать сеть с минимальными издержками.

В 2014 атака Сивиллы уже была запущена в сети Tor.

Немногие знают, почему нод именно три. Tor Project объясняет, что три – это оптимальное количество нод, которого достаточно для того, чтобы обеспечить своеобразную игру с неполной информации для злоумышленника. Если добавлять больше нод, то ваш цифровой след будет оставаться на большом количестве устройств, что негативно влияет на уровень вашей конфиденциальности.

Таким образом, каждая нода знает только то, что ей непосредственно необходимо для выполнения её роли в цепочке подключений. При этом ни одна нода не знает всей картины.

Если вы используете скрытый сервис Tor (для читателей, первый раз слышащих это название, поясню – это такие ресурсы, которые доступны только через Tor. Владелец такого ресурса может сохранять анонимность и скрывать местоположение своих серверов), то аналогичная игра в горячую картошку повторяется еще один раз. В итоге в передаче участвуют 6 нод: 3 для клиента, 3 для сервера.



СПУСКАЕМСЯ НИЖЕ

Tor с первого взгляда кажется неуязвимым. Но представьте себе такую картину: злоумышленник контролирует все три ноды Tor, через которые вы делаете запрос и может полностью отслеживать ваше подключение. Такое может произойти.

Или злоумышленник может контролировать две из трех нод, например, входную и выходную. И тогда он сможет получить некоторые метаданные о запросе и получить данные о том, кто использует Tor в данный момент.

Если наш злоумышленник является глобальным пассивным наблюдателем (то есть в реальном времени наблюдает за коммуникацией значительной части узлов сети Tor), то любые наши схемы анонимизации становятся просто-напросто бесполезными. На практике мы не видели подобных претендентов. Но в теории это возможно.

Справедливости ради, в threat-model Tor не входит глобальное пассивное наблюдение. Tor Project честно указывают это на своём сайте.

Tor также не скрывает некоторые метаданные передаваемых данных, которые можно использовать для проведения статистического анализа.

Также многие этого не знают, но в современном интернете невозможно построить по-настоящему децентрализованную сеть из компьютеров. Вам всегда нужна будет централизованная нода, которая позволит первый раз подключиться к сети и получить список других нод. В сети Tor она тоже есть.

Злоумышленник может попытаться также скомпрометировать ваш скрытый сервис: для этого необходимо поднять некоторое количество нод сети и начать отправлять на ваш сайт множество запросов до тех пор, пока на одной из ваших нод в тот же момент не появится новое подключение от этого самого ресурса. Подобную атаку можно проверить и с входной нодой сервера.

Некоторые проекты пытаются исправить несовершенства проекта Tor: в анонимной сети I2P по-умолчанию к каждому пакету добавляется какое-то число случайных данных, а для получения ответа от сервера используется другая связка нод, называемая "туннелем".

Туннели могут быть разной длины, но по-умолчанию она равна трём. При этом передающие ноды не знают, какой туннель длины.

В I2P устройства общаются между собой напрямую, даже через NAT, это работает при помощи так называемых "проводников" и техники UDP Hole Punch.

Объясню на примере.

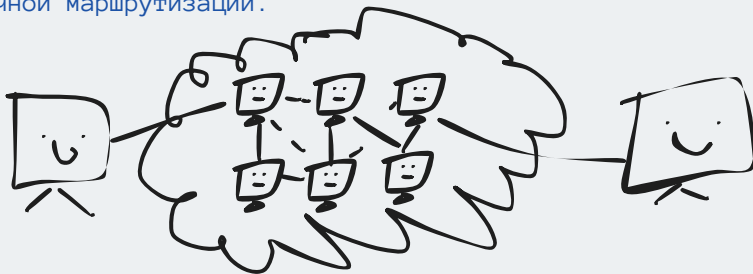
У нас есть внешний клиент I2P, который хочет соединиться с нашим.

Наш клиент сети каждый час выбирает нам проводника – узел сети с выделенным ip-адресом, который может принимать стандартные подключения извне без каких-либо проблем.

Внешний узел сети отправляет запрос проводнику, с просьбой о соединении. Проводник передает запрос на подключение нам, указывая ip-адрес и порт узла, который хочет установить соединение. Наш роутер отправляет пустой UDP-запрос к внешнему узлу и создает "окно" для подключения.

(credits to pureacetone's article:
<https://habr.com/ru/articles/564772>)

Но я хочу обратить внимание читателя на менее известную альтернативу вышеупомянутым проектам. Сети из семейства микснетов (например – микснет Нум, который сейчас активно развивается) позволяют скрыть источник передаваемых данных от глобального пассивного наблюдателя. Такие сети делают все пакеты одного размера, отправляют в сеть мусорные пакеты, но самое главное – перемешивают пакеты разных пользователей между собой и отправляют дальше **в случайном порядке, независимо друг от друга**. Этим они и отличаются от представителей луковичной маршрутизации.



В одной из многочисленных частных бесед с моим знакомым я предлагал и другую реализацию скрытой сети, на мой взгляд, являющуюся более подходящей для обмена анонимными сообщениями.

Она использует несколько другой подход. Вместо того, чтобы пытаться использовать какое-то оптимальное количество нод, моя сеть рассылает пакеты всем пользователям. Нечто подобное (изначального отправителя крайне сложно отследить, но упомянутая далее реализация анонимной сетью не является) можно увидеть в протоколе децентрализованного мессенджера Status – Waku. При передаче сообщений он использует сквозное шифрование, но вместе с этим позволяет довольно надежно скрыть изначальный источник сообщения (просто из-за большого количества нод).

Перескажу вам содержание.

Я думал над сетью, в которой невозможно будет эффективно проводить такие атаки, когда злоумышленник поднимает множество своих нод сети и может следить за трафиком, используя статистический анализ.

(Если через его входную и выходную ноды пройдет пакет (промежуточному узлу необязательно быть вредоносным), то он сможет с хорошей точностью понять, куда пытался зайти конкретный пользователь.)

У меня сразу появилось две идеи.

Во-первых, можно просто начать раздавать зашифрованные пакеты всем участникам сети, а выходные ноды, которым он предназначался, его расшифруют. Но проблема в том, что это все равно не спасает от глобального наблюдения, лишь расширяет круг поисков.

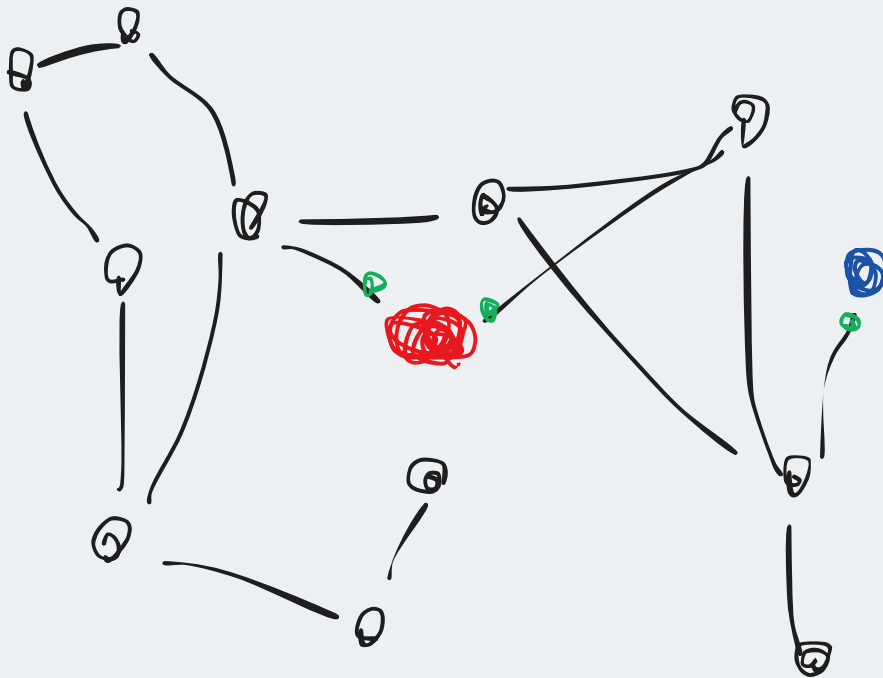
(Напомню, что глобальное пассивное наблюдение предполагает, что мы узнаем, какая именно из нод отправит пакет, который в дальнейшем распространится по сети)

Я понял, что пакеты должны выглядеть одинаково: иметь одинаковый размер и отправляться только в, например, четное время по `unixtime'y`.

Для читателей, незнакомых с понятием времени в операционных системах семейства `unix`, поясняю. Unix-время – это количество секунд с 1 января 1970 года. В контексте нашего текста, "четное" время означает четное количество секунд.

А также нужно посылать пакеты, которые ничего не значат, randomным нодам, чтобы запутать наблюдателя.

Для обеспечения такого уровня анонимности, который эта схема заслуживает, нодами должны стать абсолютное большинство участников сети. Не важно, как именно это будет реализовано: мы можем даже пропускать трафик нашей сети через любую другую оверлейную, главное, чтобы она выдавала нам статический адрес, по которому другие компьютеры смогут к нам обратиться.



Я считаю также крайне важным защищаться от тайминг-атак (о которых подробнее немного ниже), т.е. серверу нужно отвечать на запросы не мгновенно, а когда абсолютное большинство нод сети получит пакет.

Если этого не сделать, то тайминг-атаки потенциально могут деанонимизировать клиент и сервер.

Здесь тоже есть нюанс. Я говорю "абсолютное большинство", потому что у некоторых людей слабый интернет. А замедлять работу сети из-за них – кощунство.

Для адекватной работы этой схемы, необходимо, чтобы все ноды постоянно тестировали друг друга на скорость интернета. Если у какой-то из нод оно маленькое, её необходимо на время удалять из нашей модели консенсуса (она не будет защищена от тайминг-атак).

Отправка пакетов по сети будет выстроена следующим образом:

Вы знаете одни ноды, вы отправляете им пакеты (правило "чем больше, тем лучше" здесь не работает, нам нужно оптимальное количество нод, чтобы максимально расширить круг поисков, чтобы сложнее было найти изначального отправителя, но использовать только одну ноду тоже плохо, потому что пакет появится только в одной части сети), они получают пакет. Эти ноды знают другие ноды, они отправляют им твой пакет.

Если нода уже получала пакет, она не передает его дальше.

Так, пока все не получают пакет.

ПОДРОБНЕЕ ПРО АТАКИ ПО СТОРОННИМ КАНАЛАМ

В этом разделе я хочу еще немного поговорить про различные атаки по сторонним каналам. Это такой тип атак, который ищет уязвимости в вспомогательных компонентах криптосистемы, если последняя настолько хороша, что атака на нее чрезвычайно сложна.

Например, представим, что вы администрируете скрытый сервис в сети I2P, и вдруг в крупном дата-центре, где расположен ваш сервер, случается пожар. Ваш сайт падает.

Очевидно, вы себя выдали. Корреляция на лицо.

Теперь можно начать искать вас, как владельца ресурса, среди клиентов этого хостинг-провайдера...

От подобных недоразумений спасает multihoming (размещение ресурса сразу на нескольких независимых серверах).

Временные промежутки в системах с низкой задержкой – коварная вещь. С их помощью в теории возможно определить тип используемого вами приложения или получить корреляцию ваших запросов к интернет–

провайдеру с каким-нибудь сервером, участвующим в маршрутизации трафика скрытой сети.

Приведу также более простой пример: допустим, мы сделали сайт. На нем есть функция авторизации. Она требует ввести логин и пароль от пользователя, и в случае, если пароль оказывается неверен, или если такой пользователь не существует, не пускает его на сайт.

Функция хэширования пароля, которую используют все современные интернет-сервисы, в целях безопасного хранения паролей, занимает сравнительно долгий промежуток времени на исполнение.

Если в кратце, хэширование – однонаправленный процесс, который позволяет получить некоторое значение у функции $f(x)$ по аргументу x , чтобы по значению y было крайне сложно (на практике – невозможно) получить x .

Если мы проверим, что пользователя не существует и сразу, не затрачивая ресурсы нашего сервера на хэширование, отправим пользователю ошибку, то мы позволим ему определять, есть ли пользователь с определенным логином на нашем сервере или нет: атакующий сможет сравнить время, затрачиваемое сервером на ответ при вводе действительного и несуществующего аккаунтов.

В случае с сайтом мы можем проверять на соответствие какому-то предварительно хэшированному стандартному паролю тот пароль, что отправил нам пользователь, дабы занимать то же время на выполнение, что и в случае действительного аккаунта (когда хэш пароля действительно берется из базы данных).

Но что делать с timing-атаками в децентрализованных сетях? Если мы не хотим увеличивать время задержки (в сети Tor важна скорость передачи данных, это не электронная почта), то нам остается лишь один выход – увеличивать количество мусорных данных, которые придется обработать злоумышленнику, чтобы подобраться к сути.

Именно поэтому клиенты должны маршрутизировать трафик других клиентов. Это выгодно всем.

ГЛАВА 4 НЕУТЕШИТЕЛЬНЫЕ ВЫВОДЫ

У нас нет способа абсолютно безопасно выходить в интернет. Любой вариант, который мы можем придумать, подвержен разного рода теоретическим атакам (даже микснеты). Так происходит, потому что принципиально сложно установить приватную коммуникацию на расстоянии. У любого подхода всегда будут какие-то минусы.

Более того, не стоит забывать и о подводных камнях шифрования. К любому алгоритму шифрования необходимо относиться с определенной долей скепсиса. Помните, что "всё тайное становится явным", а квантовые компьютеры не за горами.

Интернет сломан и ничего лучше уже не будет. Будет только хуже. Общество глобальной слежки набирает обороты.

Мы должны сопротивляться такому курсу. Интернет будет оставаться свободным, только если люди будут к этому стремиться, разрабатывая новые протоколы и совершенствуя старые.

Но на самом деле, вместе с попытками починить интернет, мы должны не забывать об альтернативах, например, о Mesh-сетях.

И конечно, не нужно верить красивым обещаниям волшебных инструментов анонимизации. Волшебства не бывает.

Это не значит, что их не нужно использовать. Нужно! Но осторожно. Нужно понимать, что вы делаете и зачем.

КОНЕЦ!

ЗАНАВЕС!

ПУСКАЙТЕ ТИТРЫ!

СПАСИБО,

что прочитали.

Вы – меньшинство, получающее информацию из длинных текстов. Оставайтесь меньшинством. Получайте знания и распространяйте их по свету.

Мы советуем зайти на Википедию и найти там статьи по теме, почитать статьи pureacetone на Хабре и официальную документацию Tor. Всё это крайне полезно для изучения скрытых и оверлейных сетей.

Наш журнал делается в России полностью на частные деньги, и мы гордимся этим. Мы не получаем денег ни от одного из правительств мира.