Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
«Сыктывкарский государственный университет им. Питирима Сорокина»
Институт точных наук и информационных технологий
КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Допустить к защите
Зав. кафедрой информационной безопасности,
к. ф. -м. н., доцент
_____ Л. С. Носов
«____»_____ 2017 года

Курсовая работа
по дисциплине «Информационные технологии»

Tor

Научный руководитель,
к. ф. -м. н., доцент _____ Ю. В. Гольчевский
«____»_____ 2017 года

Исполнитель,
студент 123 группы _____ М. А. Виноградов
«____»_____ 2017 года

Сыктывкар, 2017 г.

Оглавление

Список использованных определений 3

Список использованных сокращений 5

Введение 6

1. Общие сведения о системе Tor 7

1.1. Вступление 7

1.2. Историческая справка 7

1.3. Аналогичные продукты 8

2. Система Tor 10

2.1. Принцип работы Tor 10

2.2. Угрозы безопасности 11

2.3. Луковая маршрутизация и её возможности 12

3. Краткое описание архитектуры Tor 15

3.1. Ячейки 15

3.2. Ячейки и потоки 16

3.3. Создание цепочки 17

3.4. Передающие ячейки 18

Заключение 20

Список использованных источников 21

Список использованных определений

В настоящей работе применяются следующие термины с соответствующими определениями:

- Трафик — объём информации, передаваемой через компьютерную сеть за определённый период времени.
- Прокси-сервер — сервер (комплекс программ) в компьютерных сетях, позволяющий клиентам выполнять косвенные запросы к другим сетевым службам. Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс (например, e-mail), расположенный на другом сервере. Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кэша (в случаях, если прокси имеет свой кэш). В некоторых случаях запрос клиента или ответ сервера может быть изменён прокси-сервером в определённых целях. Прокси-сервер позволяет защищать компьютер клиента от некоторых сетевых атак и помогает сохранять анонимность клиента.
- Клиент-сервер — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.
- Релей — узел, занимающийся получением/пересылкой сообщений
- Контрольная сумма — некоторое значение, рассчитанное по набору данных путём применения определённого алгоритма и используемое для проверки целостности данных при их передаче или хранении.
- Трассировка — это процедура получения информации о маршрутизаторах (узлах), через которые проходят пакеты данных к интересующему компьютеру (серверу). В случае проблем при доставке данных до какого-либо узла программа позволяет определить, на каком именно участке сети возникли неполадки.
- Маршрутизатор — специализированный сетевой компьютер, имеющий два или более сетевых интерфейсов и пересылающий пакеты данных между различными сегментами сети. Маршрутизатор может связывать разнородные сети различных архитектур.
- Буферизация — метод организации обмена, в частности, ввода и вывода данных в компьютерах и других вычислительных устройствах, который подразумевает использование буфера для временного хранения данных.
- Оверлейная сеть — общий случай логической сети, создаваемой поверх другой сети. Узлы оверлейной сети могут быть связаны либо физическим соединением, либо логическим, для которого в основной сети существуют один или несколько соответствующих маршрутов из физических соединений. Примерами оверлеев являются сети VPN и одноранговые сети,

которые работают на основе интернета и представляют собой «надстройки» над классическими сетевыми протоколами, предоставляя широкие возможности, изначально не предусмотренные разработчиками основных протоколов.

- Мультиплексирование — уплотнение канала, т. е. передача нескольких потоков (каналов) данных с меньшей скоростью (пропускной способностью) по одному каналу.
- Дескриптор — лексическая единица (слово, словосочетание) информационно-поискового языка, служащая для описания основного смыслового содержания документа или формулировки запроса при поиске документа (информации) в информационно-поисковой системе. Дескриптор однозначно ставится в соответствие группе ключевых слов естественного языка, отобранных из текста, относящегося к определённой области знаний.
- Постоянное HTTP-соединение, также называемые HTTP keep-alive или повторное использование соединений HTTP — использование одного TCP-соединения для отправки и получения многократных HTTP-запросов и ответов вместо открытия нового соединения для каждой пары запрос-ответ. Новый протокол HTTP/2 расширяет эту идею, позволяя одновременные многократные запросы/ответы в одном соединении. предназначенная для автоматизированного внесения определённых изменений в компьютерные файлы.

Список использованных сокращений

- IP — (англ. Internet Protocol) — это межсетевой протокол — маршрутизируемый протокол сетевого уровня стека TCP/IP.
- TCP — (англ. transmission control protocol) — протокол управления передачей — один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных.
- TCP/IP — набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет.
- SSL — (англ. secure sockets layer) — уровень защищённых сокетов — криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений. Протокол широко использовался для обмена мгновенными сообщениями и передачи голоса через IP.
- TLS — (англ. Transport Layer Security) — безопасность транспортного уровня, как и его предшественник SSL, криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети Интернет.
- AES — (англ. Advanced Encryption Standard) — симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES.

Введение

Сегодня интернет является очень важной частью нашей жизни. Раньше, его использование было проще и безопаснее, но теперь всё иначе. Отслеживать IP адреса стало легче и наблюдение за интернет активностью производится очень тщательно, чем когда-либо, а значит, сохранить анонимность в интернете стало труднее. Так, как множество вещей связано с использованием сети интернет, зачастую вы можете оставлять за собой большое количество следов и информации, которая может быть использована злоумышленниками для грабежа и взлома. Очень важно осуществлять мониторинг и вести наблюдение активности и в то же время нужно иметь возможности обходить такие наблюдения. Поэтому, пора пересмотреть свое беспечное отношение к пользованию Интернетом — научиться скрывать и шифровать свою деятельность в Сети.

В данной работе предлагается одно из решений проблемы безопасного использования Интернета — преодоления блокировки её ресурсов со стороны провайдеров, обеспечения анонимности при общении и при посещении сайтов, блогов, форумов и т. д. Это система Tor.

Tor — одно из лучших средств на сегодня для анонимной работы в Интернет. К сожалению, в настоящее время очень мало литературы по использованию системы Tor. Да и та, которая имеется, далеко не полная. На официальном сайте разработчиков Tor — на русский язык переведены всего несколько страниц.

Целью данной работы является попытка «заглянуть за кулисы» - попытка узнать что такое Tor, как с ним работать и собственно принцип его работы.

Постановка задачи

Для достижения поставленной цели необходимо решить следующие задачи:

- Обзор сети Tor и правила её использования.
- Более конкретный обзор «луковой маршрутизации».
- Описание слабостей и угроз безопасности данной сети.
- Обзор принципов его работы.
- Краткий обзор архитектуры Tor.
- Проверить работу Tor с помощью Tor-browser.

Выбор метода реализации задачи:

Главной задачей данной работы является глубокое понимание принципа работы сети Tor.

1. Общие сведения о системе Tor

1.1. Вступление

Tor (сокр. от англ. The Onion Router) — свободное и открытое программное обеспечение для реализации второго поколения «луковой маршрутизации». Он может пригодится тем, кто хочет сохранить анонимность в интернете, а также защитить трафик от третьей стороны.

Также, использование подобного программного обеспечения актуально для стран в которых есть интернет-цензура.

Луковая маршрутизация — это система прокси-серверов, позволяющая устанавливать анонимное сетевое соединение, защищённое от прослушивания. Рассматривается как анонимная сеть виртуальных туннелей, предоставляющая передачу данных в зашифрованном виде. Написана преимущественно на языках программирования C, C++ и Python.

1.2. Историческая справка

Ключевой принцип технологии Tor — «луковую маршрутизацию» разработали математик Пол Сиверсон (Paul Syverson) и программисты Майкл Рид (Michael Reed) и Дэвид Гольдшлаг (David Goldschlag) в 1995 году в исследовательской лаборатории Военно-морских сил США,в рамках проекта Free Haven по федеральному заказу, с целью защиты правительственной связи. Окончательно технология сформировалась в 1997 году в Агентстве перспективных исследовательских программ в области обороны (DARPA).

В 2002 эту разработку решили рассекретить, а исходные тексты были переданы независимым разработчикам, Пол Сиверсон совместно с бостонскими программистами Роджером Дингледайном (Roger Dingledine) и Ником Мэтьюсоном (Nick Mathewson) на базе технологии «луковой маршрутизации» разработали проект Tor. Проект стал некоммерческой организацией Tor Project Inc. Tor — акроним от The Onion Router (луковый маршрутизатор).Они создали клиент-серверное приложение и опубликовали его под свободной лицензией, чтобы все желающие могли провести проверку на отсутствие ошибок.

О поддержке проекта объявила правозащитная организация Electronic Frontier Foundation, которая начала активно пропагандировать новую систему и прилагать значительные усилия для максимального расширения сети. Но, при всем этом, большая часть источников финансирования приходится на правительство США [1,4].

По состоянию на конец 2014 года Tor имеет более 6500 узлов сети, разбросанных по всем континентам Земли, кроме Антарктиды, а число участников сети, включая ботов, превышает 2,5 миллиона. По данным Tor Metrics, в июле 2014 года Россия вошла в тройку стран, наиболее активно использующих Tor [1].

1.3. Аналогичные продукты

1.3.1. VPN

VPN — это общее название сети или соединения, которое создано внутри или поверх другой сети, например Интернет. Упрощенно, представляет собой туннель из VPN-клиента, установленного на устройстве пользователя, и VPN-сервера. Внутри этого туннеля происходит шифрование данных, которыми обмениваются пользователь и веб-ресурсы. Суть технологии VPN заключается в защите трафика любых информационных сетевых систем, аудио- и видеоконференций, систем электронной коммерции и т. д.

На сегодняшний день VPN является одним из самых надежных способов передачи данных благодаря тому, что в этой технологии реализован опыт двух серьезных компаний — Microsoft и Cisco. К примеру, совместная работа протокола PPTP (детище Microsoft) и GRE (продукт Cisco). А также еще более совершенный протокол L2TP и L2F- также являются разработками Microsoft и Cisco.

Конфиденциальность данных при VPN-соединении обеспечивается за счет того, что шифрование происходит на уровне отправителя, а расшифровка — только на уровне получателя. Содержание перехваченных пакетов, отправленных в такой сети, понятно только владельцам общего ключа шифрования, длина которого — важнейший параметр безопасности.

Ключ формируется на устройстве пользователя и сервере и доступен только им. Формирование происходит на основе случайных данных вроде случайного вопроса, ответа вашего компьютера, времени ответа, операционной системы и т. д. Этот набор факторов неповторим. Любой злоумышленник для подбора способа расшифровки должен будет повторить все эти случайные факторы, что практически невозможно, так как современные VPN-сервисы используют мощные алгоритмы шифрования на уровне финансовых организаций.

Таким образом, VPN защищает все исходящие и входящие данные на устройстве пользователя. Также пользователь получает IP-адрес VPN-сервера, который заменяет его собственный, при этом существует возможность выбора IP по локации. Допустим, вы хотите подключиться к какому-либо сервису как пользователь из США, тогда вам нужно выбрать IP американского сервера. Благодаря смене IP и шифрованию обеспечивается безопасность ваших данных от хакеров и других злоумышленников, а также полностью скрывается ваша активность в Интернете. Из недостатков VPN можно отметить снижение скорости трафика. Также вам, скорее всего, придется заплатить за использование хорошего VPN-сервиса, если защищенное соединение необходимо регулярно.

У некоторых VPN-провайдеров существует проблема с утечкой информации через IPv6 и/или при помощи подмены DNS, но полагаю, что теперь, когда на это обратила внимание общественность, улучшения в защите не заставят себя ждать [7].

2. Система Tor

2.1. Принцип работы Tor

Анонимная сеть Tor состоит из «нодов» (англ. nodes), так же для обозначения участников сети может применяться термин «релей» (англ. relays). Каждый релей — прокси-сервер, который способен принимать и отправлять данные. Любой пользователь, настроив Tor, может сделать свой ПК элементом цепочки сети (узлом), т. е. нодом. Пакет собранный клиентом Tor идёт от клиента к серверу не напрямую, а через цепочку, состоящую из трех случайно выбранных нодов, но некоторые участники сети могут изменить свою цепочку до одного нода для увеличении скорости работы клиента, что сказывается на падении уровня безопасности. Мы же в данной работе будем рассматривать цепь состоящую из трёх элементов сети. Примерный путь, который проходит каждый пакет в анонимной трёх элементной сети Tor схематически представлен на рис. 1:

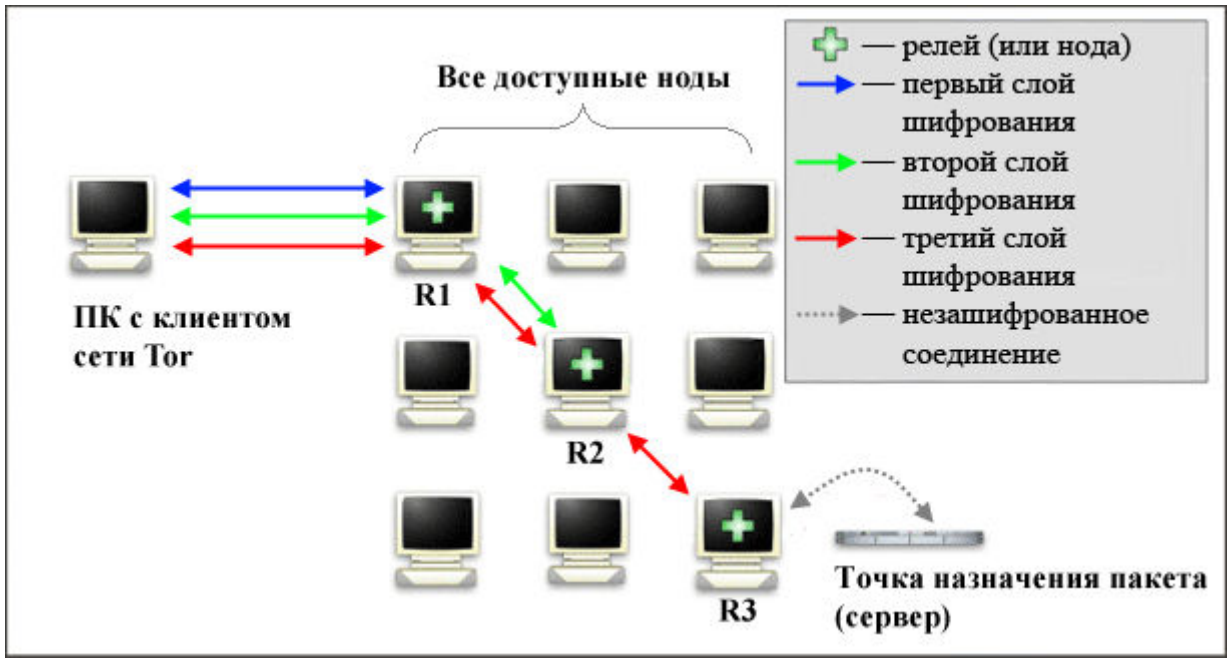

Рис. 1. — Сеть Tor

Когда пользователь запускает клиент анонимной сети Tor, последний подключается к серверам Tor и получает список всех доступных нодов. Из большого числа релеев случайным образом выбирается всего три. Дальнейшая передача данных осуществляется по этим трем нодам, выбранным случайно, причем она осуществляется поэтапно от более «верхнего» релея к более «нижнему». Перед тем, как отправить пакет первому релею в цепочке, на стороне клиента происходит поэтапное шифрование этого пакета: вначале для третьей ноды (красная стрелка), затем для

второй (зеленая стрелка) и, наконец, для первой (синяя стрелка). Когда первый релей (R1) принимает пакет, он производит расшифровку самого верхнего уровня (синяя стрелка). Тем самым релей получает данные о том, куда дальше требуется отправить пакет. Происходит ретрансляция пакета, но уже с двумя слоями шифрования вместо трех. Работа второго и третьего релея происходит аналогичным образом: каждая нода получает пакет, расшифровывает «свой» слой и отправляет пакет далее. Последний (третий, R3) релей в цепочке поставляет пакет в точку назначения (сервер) в незашифрованном виде. Ответ от сервера аналогично проходит по той же цепочке, но в обратном направлении. Такой подход обеспечивает больше гарантий для анонимности, чем традиционные анонимайзеры. Анонимность достигается за счет того, что первичный источник пакета скрывается. Важным является и то, что все ноды, участвующие в передаче, не получают информацию о содержании пакета, а лишь данные о том, откуда пришло зашифрованное сообщение и кому передать его дальше. Для обеспечения анонимности в сети Tor применяется как симметричное шифрование, так и ассиметричное. Каждый слой использует оба метода, что так же выгодно отличает Tor от других анонимайзеров [8].

2.2. Угрозы безопасности

Следует помнить, что один из узлов цепочки Tor вполне может оказаться уязвимым. Также по той или иной причине враждебные клиенту действия может осуществлять сайт — от попыток выяснить истинный адрес клиента до «отражение» его сообщения.

Просмотр и модификация сообщения.

На последнем узле цепочки Tor исходное сообщение от клиента окончательно расшифровывается для передачи его серверу в первоначальном виде.

Раскрытие отправителя.

При работе с сетью Tor к сообщениям пользователя может добавляться техническая информация, полностью либо частично раскрывающая отправителя.

Другие угрозы безопасности:

- На пути от последнего узла сети Tor до сервера назначения информация, идущая в открытом виде, может быть модифицирована, поэтому необходимо обязательно проверять целостность данных, например, при помощи контрольных сумм.
- На пути от последнего узла сети Tor к серверу назначения, существует возможность кражи пользовательских реквизитов доступа к серверу, например логина и пароля, cookie или сеанса связи.
- Сервер может отклонить сообщение с адресом отправителя узла сети Tor [1,3].

2.3. Луковая маршрутизация и её возможности

2.3.1. Алгоритм

Давайте более подробно рассмотрим алгоритм работы «луковой маршрутизации», Луковая маршрутизация — это технология анонимного обмена информацией через компьютерную сеть. Сообщения неоднократно шифруются и потом отсылаются через несколько сетевых узлов, называемых луковыми маршрутизаторами. Каждый маршрутизатор удаляет слой шифрования, чтобы открыть трассировочные инструкции и отослать сообщения на следующий маршрутизатор, где все повторяется. Таким образом, промежуточные узлы не знают источник, пункт назначения и содержание сообщения.

Идея «луковой маршрутизации» состоит в том, чтобы сохранить анонимность отправителя и получателя сообщения и обеспечить защиту содержимого сообщения во время его передачи по сети.Она работает в соответствии с принципом смешанных соединений Чаума: сообщения передаются из источника к месту назначения через последовательность прокси («луковых маршрутизаторов»), которые перенаправляют сообщение в непредсказуемом направлении.

Преимущество «луковой маршрутизации» (и смешанных соединений в целом) состоит в том, что отпадает необходимость доверия каждому участвующему маршрутизатору. Даже если один или несколько из них окажутся взломанными, анонимное соединение все равно сможет быть установлено. Это достигается за счёт того, что каждый маршрутизатор в сети из прокси принимает сообщения, шифрует их заново и передает их на другой луковый маршрутизатор. Злоумышленник, имеющий возможность проводить мониторинг всех луковых маршрутизаторов в сети, теоретически может проследить путь сообщения через сеть. Но задача сильно усложняется, даже если злоумышленник имеет доступ к одному или нескольким маршрутизаторам на пути сообщения.

2.3.2. Маршрутизация

Маршрутизатор в начале передачи выбирает случайное число промежуточных маршрутизаторов и генерирует сообщение для каждого, шифруя их симметричным ключом и указывая для каждого маршрутизатора, какой маршрутизатор будет следующим на пути. Для получения симметричного ключа («ключа сессии») с каждым из промежуточных маршрутизаторов производится начальное установление связи с использованием открытого ключа этого маршрутизатора, через маршрутизаторы, предшествующие ему в цепочке. В результате сообщения, передаваемые по цепочке, имеют «слоистую» структуру, в которой необходимо расшифровать внешние слои, чтобы получить доступ к внутреннему слою. Каждый маршрутизатор, получающий сообщение, «сдирает» слой с лука — расшифровывает своим ключом сессии содержимое сообщения:

предназначенные этому маршрутизатору инструкции по маршрутизации и зашифрованное сообщение для маршрутизаторов, расположенных дальше по цепочке. Последний маршрутизатор снимает последний слой шифрования и отправляет сообщение адресату. Установленная цепочка остается доступной для двусторонней передачи данных в течение некоторого периода времени. Иллюстра-

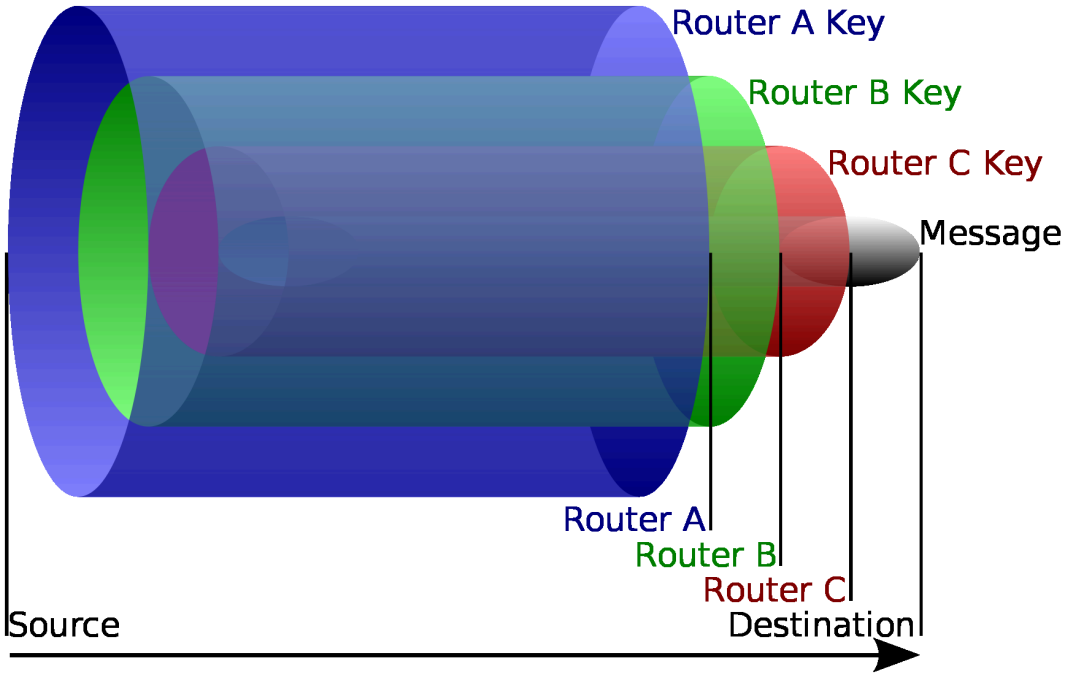

Рис. 2. — Слои шифрования у «луковой маршрутизации»

тивное представление слоев шифрования у «луковой маршрутизации» можно увидеть на рис. 2.

Получатель запроса может отправить ответ по той же цепочке без ущерба для анонимности каждой из сторон. При этом слои шифрования, наоборот, «наращиваются» на каждом маршрутизаторе, пока ответ не достигнет отправителя запроса. Отправитель владеет всеми ключами сессии, используемыми в цепочке, и поэтому сможет расшифровать все слои: от внешнего, зашифрованного ближайшим к отправителю маршрутизатором в цепочке, до внутреннего, зашифрованного маршрутизатором, ближайшим к получателю запроса.

2.3.3. Слабости

У «луковой маршрутизации» есть несколько слабостей. С одной стороны, она не обеспечивает защиту против анализа синхронизации. Если злоумышленник наблюдает за относительно слабо загруженным луковым маршрутизатором, он может соединять входящие/исходящие

сообщения путем просмотра того, как близко по времени они были получены и переправлены. Однако это может быть преодолено путем буферизации нескольких сообщений и передачи их с использованием псевдослучайного временного алгоритма. Сети «луковой маршрутизации» также уязвимы к перекрещивающимся и предшествующим атакам. Перекрещивающиеся атаки базируются на том, что луковые маршрутизаторы периодически прекращают работать или отключаются от сети, в то время как к сети подключаются новые маршрутизаторы. Любой путь передачи, который продолжает функционировать, не может проходить ни через отключённые маршрутизаторы, ни через маршрутизаторы, присоединившиеся к сети в последнее время. В предшествующей атаке злоумышленник, который контролирует луковый маршрутизатор, отслеживает сессии в то время, как они проходят через несколько перестроений пути. Если злоумышленник наблюдает, как изменяется путь в ходе нескольких перестроений, он сможет увидеть первый маршрутизатор в цепи более отчетливо. Луковая маршрутизация не в состоянии защитить данные, проходящие через выходные узлы, отдавая оператору полный доступ к передаваемому содержанию, и поэтому луковые сети не должны использоваться для передачи личной информации без использования конечной криптографии, такой как SSL [4,5]. Схематическое представление слабостей «луковой маршрутизации» можно увидеть на рис. 3 [1].

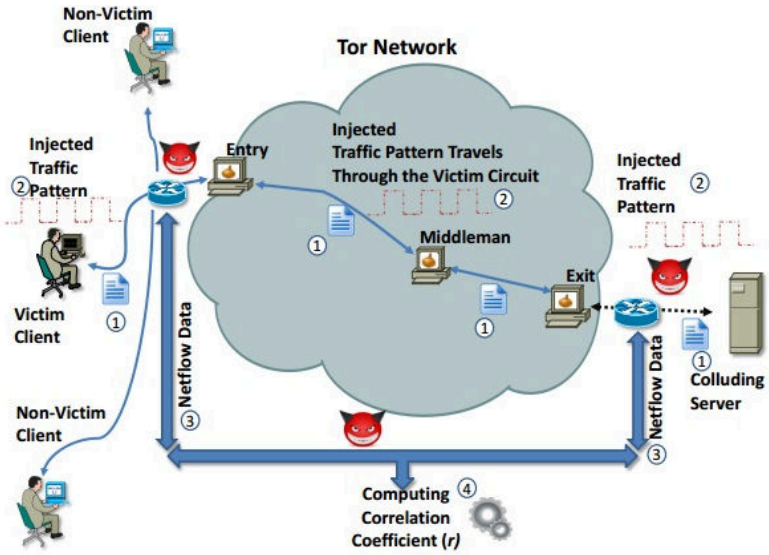

Рис. 3. — Слабости «луковой маршрутизации»

3. Краткое описание архитектуры Tor

Сеть Tor — это оверлейная сеть; каждый луковый маршрутизатор работает, как обычный процесс уровня пользователя без всяких специальных привилегий. Каждый луковый маршрутизатор поддерживает TLS соединение с каждым другим луковым маршрутизатором сети.

Каждый пользователь запускает на своём компьютере программу, называемую луковым прокси, для получения информации из серверов каталогов, установления цепочек в сети и обработки соединений от пользовательских приложений. Эти луковые прокси принимают TCP-потоки и мультиплексируют их по цепочкам. Луковый маршрутизатор на другом конце цепочки соединяет пользователя с адресом назначения и передаёт данные дальше.

Каждый луковый маршрутизатор поддерживает долгосрочный идентификационный ключ и краткосрочный луковый ключ. Идентификационный ключ используется для подписи TLS сертификатов, дескриптора лукового маршрутизатора (объединение его ключей, адресов, скорость передачи, политики точки выхода и т. п.), каталогов (используется серверами каталогов). Луковый ключ используется для расшифровки запросов от пользователей на установление цепочки и для согласования временных ключей. TLS протокол также устанавливает краткосрочный ключ соединения при общении между луковыми маршрутизаторами. Краткосрочные ключи меняются периодически и независимо друг от друга, для того чтобы уменьшить ущерб от компрометации ключа.

3.1. Ячейки

Луковые маршрутизаторы общаются друг с другом, а также с пользователями луковых прокси через TLS соединения, зашифрованными временными ключами. Использование TLS защищает данные соединения с совершенной прямой секретностью, не даёт возможности атакующему изменить данные или выдать себя за луковый маршрутизатор.

Трафик в этих соединениях передаётся ячейками фиксированного размера. Каждая ячейка имеет размер 512 байт и состоит из заголовка и полезной информации. Заголовок содержит идентификатор ячейки (circID), который определяет, на какую цепочку ссылается данная ячейка (в одном TLS соединении могут быть мультипликсированы много цепочек) и команда, которая определяет что делать с полезной информацией ячейки (идентификатор ячейки является заданным для соединения: каждая цепочка имеет различный circID на каждом соединении "луковый маршрутизатор — луковый прокси" или "луковый маршрутизатор — луковый маршрутизатор", которые она проходит). В зависимости от их команды ячейки могут быть управляющими или передающими. Управляющие интерпретируются узлами, которые их получают. Передающие несут в себе данные, которые нужно передать насквозь через сеть. Управляющие команды: выровнять (в данный момент используется для keepalive, может также быть использована для выравнивания

ссылок); создать или создано (служит для создания новой цепочки); уничтожить (завершение цепочки).

Передающие цепочки имеют ещё один заголовок (передающий заголовок), который находится перед полезной информации и содержит streamID (идентификатор потока: на одну цепочку могут быть мультиплексированы несколько потоков) и сквозную контрольную сумму для проверки целостности, длину передаваемой полезной информации и передаваемую команду. Всё содержимое передаваемого заголовка и передаваемой полезной информации ячейки шифруется и дешифруется вместе каждый раз, когда передающая ячейка движется по цепочке. Для этого используется 128-битный AES шифрователь, работающий в режиме счётчика для генерации шифрованного потока. Команды передачи: передать данные, начать передачу данных (для открытия потока), закончить передачу данных (для корректного закрытия потока), завершить передачу данных (для закрытия испорченного потока), передача данных начата (для оповещения луковых прокси, что начало передачи было успешным), расширить цепочку и цепочка расширена (для расширения цепочки за один переход и для получения подтверждения), сократить цепочку и цепочка сокращена (для того, чтобы завершить только часть цепочки, и чтобы послать подтверждение), передать самому себе (служит для контроля перегрузки) и прервать передачу данных (служит для создания пустышек на длинные дистанции). Сначала мы сделаем обзор структуры ячейки (обычной и передающей), а потом опишем все типы ячеек и их команды более подробно.

Пример структуры ячейки приведен на рис. 4.

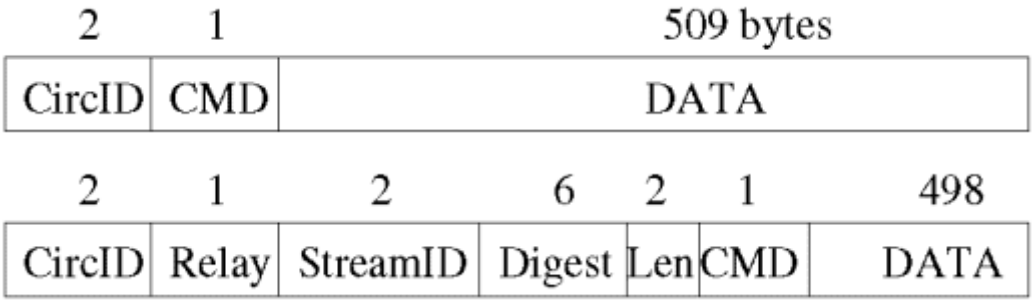

Рис. 4. — Структура ячейки

3.2. Ячейки и потоки

Луковая маршрутизация изначально использовала по отдельной цепочке на каждый TCP-поток. Поскольку создание каждой цепочки может занимать несколько десятых секунды (из-за

использования шифрования с открытым ключом и задержек сети), система в целом получается медленной для приложений, открывающих много TCP-потоков (например, web-браузеров).

В Tor каждая цепочка может быть использована несколькими TCP-потоками. Для того чтобы избежать задержек, пользователи создают цепочки заранее. Для того чтобы снизить связность между его потоками, луковый прокси пользователя периодически признаёт старые неиспользуемые цепочки устаревшими и создаёт взамен новые. Луковый прокси заменяет цепочки новыми раз в минуту: таким образом даже для «тяжёлых» (генерирующих много трафика) пользователей время на генерацию ничтожно, но количество цепочек, которые могут быть связаны друг с другом через выбранный узел выхода, ограничено. В дополнение к этому, так как цепочки генерируются в фоне, луковый прокси может восстановиться после ошибки при создании цепочки незаметно для пользователя.

3.3. Создание цепочки

Луковый прокси пользователя создаёт цепочку последовательно, согласуя симметричный ключ с каждым луковым маршрутизатором в цепочке, по одному переходу за раз. Для начала создания новой цепочки луковый прокси (назовём его Алёной) посылает ячейку с командой создать на первый узел (назовём его Боря) в выбранном ею пути. Она выбирает новый circID C_{AC}, который не был использован в текущем соединении её с Борей. Ячейка создать содержит в качестве полезной информации первую часть квитирования протоколом Диффи-Хеллмана (g^x), зашифрованного луковым ключом Бори. Боря отвечает ячейкой создано, которая содержит g^y вместе с хешем согласованного ключа:

$$K = g^{xy} \tag{3.1}$$

Сразу после того как ячейка была создана, Алёна и Боря могут отправлять друг другу передающие ячейки, зашифрованные с помощью согласованного ключа.

Для того чтобы расширить цепочку, Алёна шлёт ячейку расширить цепочку Боре, в которой указан адрес следующего лукового маршрутизатора (назовём его Костя) и зашифрованное значение g^{x2} для этого маршрутизатора. Боря копирует половину квитирования в ячейку создать и передаёт её дальше Костя для расширения цепочки. При этом Боря выбирает новый circID C_{BC}, который не был ещё использован в текущем соединении между ним и Костей. Алёне не нужно знать этот circID; только Боре нужно поддерживать связь между двумя circID: C_{AB} на соединении с Алёной и C_{BC} на соединении с Костей. Когда Костя отвечает ячейкой создано, Боря оборачивает полезную информацию в ячейку цепочка расширена и посылает её назад Алёне. Теперь цепочка расширена до Кости и Алёна и Костя используют общий ключ:

$$K_2 = g^{x2y2} \tag{3.2}$$

Для расширения цепочки до третьего узла и дальше Алёна продолжает делать то же самое, каждый раз сообщая последнему узлу в цепочке расширить её на один переход.

В следующем шаге Боря подтверждает, что именно он получил g^x и выбрал у. Мы используем шифрование с открытым ключом, поскольку единственная ячейка слишком мала, чтобы содержать сразу и открытый ключ, и подпись.

3.4. Передающие ячейки

Как только Алёна устанавливает цепочку (то есть обменивается ключами с каждым луковым маршрутизатором в цепочке), она может слать передающие ячейки.

Перед тем как принять передающую ячейку, луковый маршрутизатор находит соответствующую цепочку, расшифровывает заголовок и полезную информацию с помощью сессионного ключа этой цепочки. Если ячейка снабжена заголовком Алёна, то луковый маршрутизатор потом проверяет корректность свёртки расшифрованной ячейки. Процесс проверки можно сильно оптимизировать, если учесть, что первые два байта должны быть нулями. Таким образом, в большей части случаев вычислять хеш не требуется.

Если свёртка корректна, то он принимает передающую ячейку и совершает действия, описанные ниже. В противном случае, луковый маршрутизатор смотрит на circID и луковый маршрутизатор для следующего шага в цепочке, заменяет circID на подходящий и посылает расшифрованную передающую ячейку следующему луковому маршрутизатору. Если луковый маршрутизатор на конце цепочки получает передающую цепочку, которую он не может опознать, он генерирует ошибку и цепочка завершается.

Луковый прокси обрабатывает входящие передающие ячейки похожим образом: он последовательно раскрывает передающий заголовок и полезную информацию при помощи сессионных ключей, используемых совместно с каждым луковым маршрутизатором в цепочке, начиная от ближайшего, заканчивая дальним. Как только свёртка оказывается корректной, автором ячейки признаётся тот луковый маршрутизатор, зашифрованное сообщение которого было только что удалено.

Для того чтобы создать передающую ячейку для конкретного лукового маршрутизатора, Алёна находит соответствующую цепочку, расшифровывает заголовок и полезную информацию с помощью сессионного ключа этой цепочки. Так как на каждом шаге свёртка шифруется в различные значения, только луковый маршрутизатор, которому предназначалось сообщение, получит осознанное значение. Эта топология протекающей трубы позволяет потокам Алёны выходить через разные луковые маршрутизаторы с использованием единственной цепочки. Алёна может

выбирать различные точки выхода благодаря политикам точек выхода, а также сохранять в тайне то, что посланные ею потоки были созданы одним отправителем.

Когда луковый маршрутизатор отвечает Алёне передающей цепочкой, он шифрует заголовок передающей ячейки и полезную информацию единственным ключом (общим с Алёной) и посылает ячейку по цепочке назад Алёне. Последующие луковые маршрутизаторы при передаче ячейки назад Алёне добавляют свои слои шифрования.

Для того чтобы уничтожить цепочку, Алёна шлёт управляющую ячейку уничтожить. Каждый луковый маршрутизатор, который получает ячейку уничтожить, закрывает все потоки этой цепочки и передаёт ячейку уничтожить дальше. Нормальное завершение цепочек происходит так же, как и создание: инкрементально. Алёна посылает ячейку сократить цепочку единственному луковому маршрутизатору из цепочки. Этот луковый маршрутизатор посылает ячейку уничтожить дальше и подтверждает этот факт ячейкой цепочка сокращена. Алёна может расширить цепочку до различных узлов без оповещения промежуточных узлов о своих намерениях. Если один из узлов прекращает свою работу, соседний узел может послать ячейку цепочка сокращена назад Алёне [2,6].

На этом краткий обзор работы архитектуры можно считать завершённым.

Заключение

В данной работе были выполнены все поставленные задачи. Был проведён обзор всего процесса работы сети Tor. Был подробно разобран принцип её работы, а также была описана архитектура сети Tor, на основе которой она была создана. Так же были рассмотрены слабости и основные угрозы безопасности, с которыми может столкнутся любой пользователь данной сети.

Это замечательный инструмент для обеспечения анонимности и сохранности данных в сети Интернет, т. к. весь трафик передается через цепочку узлов исключительно в зашифрованном виде. Крайне затруднительно или, если полностью поверить разработчикам, то даже невозможно становится отследить источник отправки данных благодаря постоянно меняющимся цепочкам специальных узлов-посредников, через которые передаются данные. Минус на первый взгляд один — скорость работы. Каждый из нодов, входящих в цепочку, вносит серьезную задержку, как по времени отклика, так и банально по ширине канала. Но при всем этом данные которые передаются можно перехватить и сделать это не трудно.

Еще одним неожиданным результатом данной работы оказалось обнаружение того, что число нодов в системе фиксированное — три. Это было сделано разработчиками для оптимального сохранения безопасности и скорости передачи данных. Недавно была добавлена возможность уменьшить число нодов до одного для увеличения скорости соединения, но пользователя такого соединения легче деаномизировать.

Можно подвести итог: Tor не является панацеей используя Tor, всё-таки надо соблюдать несколько правил и разобраться в том, как он работает, что он умеет и чего он не может, чтобы не сводить на «нет» все его усилия. Но при всем этом он очень бурно развивается и количество его пользователей увеличивается.

Данная работа будет полезна тем, кто желает узнать что такое Tor, как он работает и как с ним работать.

Список использованных источников

1. АО Kasperky Lab [Электронный ресурс] URL: https://securelist.ru/analysis/obzor/25842/deanonimizator-gde-zakanchivaetsya- anonimnost-v-darknete/ (дата обращения 16.05.2016).
2. Dingledine R., Mathewson N., Syverson P.N. Tor: The Second-Generation Onion Router. — The Free Haven Project, 2014. — 17 с.

3 LibrePortal [Электронный ресурс] URL: http://libreportal.net/security/tor.html (дата обращения 16.05.2016).

4. Reed M.G., Syverson P.F., Goldschlag D.M. Anonymous Connections and Onion Routing. — USA: Naval Research Laboratory. — 15 с.
5. Syverson P., Tsudik G., Reed M., Landwehr C. Towards an Analysis of Onion Routing Security. — USA: Naval Research Laboratory. — 19 с.
6. Torproject [Электронный ресурс] URL: https://www.torproject.org (дата обращения 16.05.2016).

7 Trashbox.ru [Электронный ресурс] URL: https://trashbox.ru/topics/85930/tor-vs.-vpn-odno-drugoe-ili-vsyo-vmeste (дата обращения 16.05.2016).

8. КакПросто [Электронный ресурс] URL: http://www.kakprosto.ru/kak-883181-kak-rabotaet-tor (дата обращения 16.05.2016).