Анализ вариантов применения протоколов туннелирования при реализации вложенного туннелирования

Е. А. Мазина[1], А. К. Канаев[2]

Петербургский государственный университет путей сообщения Императора Александра I

[1]e07.mazina@gmail.com, [2]kanaevak@mail.ru

Аннотация. **В статье представлен обзор протоколов туннелирования, которые уже нашли применение в сетях VPN. Методы обеспечения безопасности передачи данных рассматриваются в рамках технологий виртуальных туннелей VPN. В качестве механизма, внедрение которого позволит повысить безопасность и доступность передачи информации исследуется технология вложенного туннелирования. Также в работе предлагается оценка этих протоколов при реализации вложенного туннелирования на предмет перспективного моделирования процессов передачи данных в технологических сетях связи.**

Ключевые слова: VPN; протоколы туннелирования; передача данных; телекоммуникации; сети связи; анализ данных

I. Введение

В наше время защита данных стала неотъемлемым аспектом телекоммуникационных систем. В соответствии с [1], для безопасной передачи данных между двумя пользователями сети требуется решение двух основных задач:

- защита локальных сетей и отдельных устройств, подключенных к общественным каналам связи, от несанкционированного доступа;
- защита информации при ее передаче по каналам связи.

Эти задачи могут быть решены с помощью протоколов туннелирования, или, другими словами, построения Virtual Private Network (VPN-сети). Под VPN понимают потоки данных одного предприятия, которые существуют в публичной сети с коммутацией пакетов и в достаточной степени защищены от влияния потоков данных других пользователей этой публичной сети [1].

Одной из главных причин использования VPN-сетей является шифрование и туннелирование данных при их передаче. Туннелирование – это метод создания сетей, при котором один сетевой протокол инкапсулируется в другой. В статье приведён анализ некоторой выборки протоколов туннелирования.

На сегодняшний день защита персональных данных требует не только обеспечения безопасности самой сети, но также и применения методик создания «вложенных» VPN, которые помогут построить дополнительную защиту, отделяющее зону обработки персональных данных от остальной части информационной системы [2].

Термин «вложенный» VPN может иметь разные значения, но обычно означает наличие географически разделенных по сети узлов для подключения и выхода в Интернет. Также многократная вложенность появляется из-за стремления к безопасности каждого участника процесса (клиент, оператор доступа, оператор транспортных услуг), которые пытаются внедрить свои средства защиты.

Рассмотрим основные пути при установлении VPN-соединений. При создании VPN между серверами клиент устанавливает VPN-соединение только с первым сервером (рис. 1*а*). Туннель между серверами может использовать иной протокол, отличный от того, который использует клиент. В этом режиме все промежуточные серверы видны в трассировке маршрута. У клиента нет способа проверить, как именно промежуточные серверы соединены друг с другом.

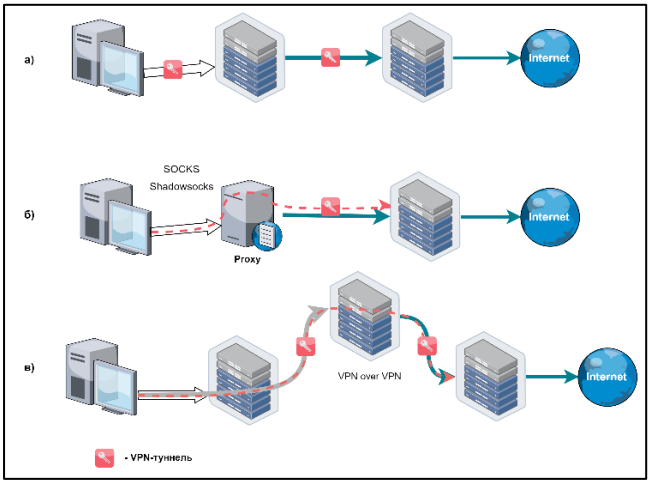

Рис. 1. Варианты построения туннеля в VPN-технологиях: а) VPN между серверами; б) VPN-соединения через прокси; в) Метод «вложенного» VPN

Установление VPN-соединения через прокси-сервер также довольно распространенный метод (рис. 1*б*). Он часто используется для скрытия VPN-трафика за другим протоколом. В этом режиме прокси-сервер не виден в трассировке маршрута.

Метод группового VPN (рис. 1*в*) требует сложной маршрутизации на стороне клиента и правильного порядка запуска всех VPN-соединений, однако при трассировке маршрута промежуточные серверы не видны [3].

Таким образом, при рассмотрении вариантов применения протоколов туннелирования при реализации многоуровневого туннелирования будем руководствоваться последней приведённой реализацией.

Однако, следует помнить, что использование вложенного туннелирования может замедлить скорость соединения и увеличить нагрузку на оборудование. Поэтому перед применением данной технологии необходимо оценить ее преимущества и недостатки в конкретной ситуации. Оценка протоколов при использовании технологии вложенного туннелирования позволит решить задачу защищенности передаваемых данных и канала связи в целом.

II. Обзор литературы

Вложенный туннель VPN является эффективным средством обеспечения безопасности и конфиденциальности данных в сети. Для более подробного изучения механизмов VPN была рассмотрена спецификация RFC4923[4]. В ней определены преимущества использования многоуровневого VPN, варианты резервирования и приоритетности, а также варианты рассмотренные реализации туннелей.

В работе [5] исследована зависимость производительности сети от мощности алгоритмов VPN на предмет влияния на накладные расходы. Эмпирические измерения показывают, что производительность критически зависит от характера данных и сжимаемости при различных условиях пропускной способности Интернета.

При передаче несжимаемых данных, таких как видео MP4, использование сжатия в VPN-алгоритме вызывает увеличение задержек. Если сжимать текстовый файл с помощью VPN-алгоритма при недостаточной загрузке полосы пропускания, это может привести к значительным задержкам. В связи с этим, стандартное использование сжатия по умолчанию в настройках VPN является неэффективным. Поэтому рекомендуется адаптировать конфигурацию VPN к типу данных, передаваемых через туннель, и учитывать доступную пропускную способность Интернета.

Также было выявлено, что влияние на производительность из-за вложенного VPN очень велико. Для повышения безопасности увеличение размера ключа шифрования может быть лучшим вариантом по сравнению с многоуровневым VPN. Однако, следует отметить, что шифрование информации требует больших вычислительных ресурсов и, как следствие, также приводит к появлению задержек. К весомым задержкам при рассмотрении высокоскоростной передачи (более 10 Мбит/с) можно отнести задержки при контроле целостности и шифровке/расшифровке сообщений, так как время передачи пакета по тракту будет превышать время на шифрование/дешифрование и контроль целостности пакета.

Также основным параметром оценки производительности VPN-сервисов являются задержки, возникающие при создании инкапсулируемого пакета, т. е. при добавлении дополнительных заголовков. Эти проблемы решаются увеличением быстродействия программных или аппаратных средств шифрования и использованием протоколов туннелирования, удовлетворяющих пропускной способности канала.

В статье [6] рассматривается использование вложенного туннеля как более защищенное соединение. В ней описывается возможность объединения преимуществ технологий VPN IPSec и SSL в качестве соединения site-to-site.

Также для более подробного изучения механизмов SSL-туннелирования была рассмотрена книга [7]. Преимущество SSL VPN в том, что он не зависит от платформы. Используя любой браузер, поддерживающий SSL, возможно получать доступ к ресурсам, не беспокоясь о базовой операционной системе. С помощью технологий удаленного доступа VPN-клиент инициирует прямое подключение к серверам, расположенным в защищенной сети. Так же не нужно устранять неполадки в стороннем VPN-клиенте, если соединение не работает должным образом.

III. Теоретический подход и методология

Как было сказано, вложенное туннелирование представляет собой процесс, при котором один VPN-туннель создается внутри другого VPN-туннеля (рис. 2). Это позволяет обеспечить дополнительную защиту данных, передаваемых через Интернет, поскольку они шифруются дважды. Рассмотрим основные варианты «вложенных» туннелей ниже.

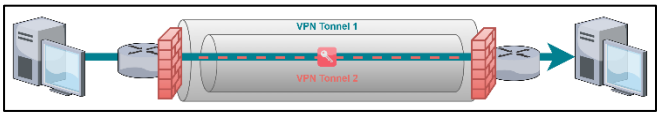

Рис. 2. Модель вложнного туннеля

Протокол IPsec может быть использован для создания туннеля, внутри которого будет передаваться трафик, защищенный SSL/TLS. В этом случае, SSL/TLS будет использоваться для защиты данных внутри туннеля, созданного IPsec.

Также протоколы L2TP и SSH могут быть использованы вместе для создания безопасного соединения между двумя точками в сети. L2TP обеспечивает безопасность и конфиденциальность данных, передаваемых внутри туннеля, а SSH обеспечивает безопасное удаленное управление компьютером.

Протокол «проксирования» SOCKS используется для создания туннеля, внутри которого будет передаваться трафик. SSH может быть использован для создания безопасного удаленного управления компьютером. Таким образом, SOCKS и SSH могут быть использованы вместе для создания безопасного соединения между двумя точками в сети. SOCKS используется для создания туннеля, внутри которого будет передаваться трафик, а SSH используется для создания безопасного удаленного управления компьютером.

В качестве примера можно привести применение туннеля SHADOWSOCKS для сохранения работоспособности территориально распределённой локальной сети, построенной на базе протокола OpenVPN [8]. Далее предлагается исследовать взаимодействие протоколов между собой и их «способность к вложенности». Основные виды протоколов туннелирования и их характеристики приведены в табл. 1.

Основываясь на результаты, полученные в [5], можно сделать вывод, что применение вложенности не является эффективным с точки зрения скорости передачи: было выявлено, что передача файлов задерживалась примерно в 1,83–4 раза для различных комбинаций, вложенных VPN, по сравнению с одиночным туннелем. сжатие существенно не влияет

на скорость передачи видеофайла, однако сжатие увеличивает скорость передачи текстового файла примерно в 1,4–1,8 раза для различных вложенных комбинаций VPN.

ТАБЛИЦА I. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ПРОТОКОЛОВ ТУННЕЛИРОВАНИЯ

Название протокола	**PPTP**	**L2TP**	**IPsec**	**OpenVPN**	**WireGuard**	**SOCKS**	**SSL**	**SSH**
Уровень OSI	Канальный	Канальный	Сетевой	Сетевой	Сетевой	Сеансовый	Сеансовый	Прикладной
Порты	TCP-порт 1723	UDP-порт 500 для первонач. обмена ключами и UDP-порт 1701 для начальной конфигурации L2TP	UDP-порт 500 для первоначального обмена ключами	Любой UDP- или TCP-порт	Любой UDP-порт	Стандартные порты 1080, 1081	Порт 443 для HTTPS, 465 для SMTPS (электронная почта), 636 для LDAPS, 563 для NNTPS, 994 для IRCS (чат), 995 для POP3S	TCP-порт 22
Доступность	Поддерживается большинством современных устройств	Большинство современных устройств поддерживают этот протокол	Устанавливается между двумя четко определенными наборами устройств защищенного доступа и/или клиентского программного обеспечения	OpenVPN не входит ни в один выпуск операционной системы и требует установки клиентского программного обеспечения	Wire Guard встроен в ядро Linux версии 5.6. Для других операционных систем, отличных от Linux, требуется установка клиентского приложения Wire Guard	Требуется установка дополнительного программного обеспечения, такого как клиент SOCKS или VPN-клиент, который поддерживает SOCKS	Доступ возможен из любого места с использованием стандартных браузеров	Чтобы обеспечить SSH доступ пользователю необходимы SSH-клиент и SSH-сервер. Каждая операционная система имеет свой набор программ, обеспечивающ их соединение. Так, для Linux это lsh (server и client), openssh (server и client). Для Mac OS зачастую используется NiftyTelnet SSH
Наличие механизмов шифрования	Использует Microsoft Point-to-Point Encryption (MPPE), который реализует RSA RC4 с максимум 128-битными сеансовыми ключами	3DES или AES	Реализует большое количество криптографически х алгоритмов, включая AES, Blowfish, Camellia	Использует библиотеку OpenSSL (реализует большинство популярных криптографичес ких стандартов)	Обмен ключами по 1-RTT, Curve25519 для ECDH, RFC7539 для ChaCha20 и Poly1305 для аутентификацион-ного шифрования, и BLAKE2s для хеширования	Отсутствует	Асимметричная криптография для аутентификации ключей обмена, симметричное шифрование	Симметричное шифрование, AES, Blowfish или 3DES. Целостность передачи данных проверяется с помощью CRC32 в SSH1 или HMAC-SHA1/HMAC-MD5 в SSH2.
Взаимодействие с другими протоколами (в т.ч. в части вложенности туннель в туннеле)	IPsec, L2TP и SSL	L2TP/IPsec считается безопасным и не имеет серьезных выявленных проблем. L2TP можно использовать с SSH, вложив его в SSH-туннель. Это позволяет L2TP-трафику проходить через SSH, что обеспечивает дополнитель-ную защиту и анонимность	Взаимодействует с IKE, ESP и AH. Можно организовать Site-to-site соединение с использованием традиционного туннеля IPSec и SSL-туннеля	Может туннелироваться через другие протоколы, такие как IPv4 или IPv6; взаимодейство-вать с протоколами аутентификации, такими как TLS, для обеспечения безопасности соединения. Трафик OpenVPN может быть вложен внутрь SSH-соединения [8]	IPsec, L2TP, SSL/TLS.Также может работать с другими протоколами маршрутизации, такими как BGP, для настройки оптимальных маршрутов.	SOCKS и SSH могут совместно использоваться для создания безопасного соединения между двумя узлами в сети	Он может взаимодействоват ь с протоколы HTTP, FTP, SMTP используют SSL как расширение для защиты данных. Также возможно образование многоуровневого соединения IPSec внутри SSL [6]	SSH может взаимодейство вать с SFTP для передачи файлов. Он также может использоваться для туннелировани я других протоколов, таких как HTTP
Слабые места в системе безопасности	Обладает серьезными уязвимостями. MSCHAP-v2 уязвим для атаки по словарю, а алгоритм RC4 подвергается атаке Bit-flipping	3DES уязвим для Meet in the Middle[1], но AES не имеет известных уязвимостей	Серьезных недостатков безопасности не было выявлено	Серьезных недостатков безопасности не было выявлено	Серьезных недостатков безопасности не было выявлено	Заголовки пакетов могут содержать персональные данные пользователей, которые не защищены из-за отсутствия шифрования	Уязвим для Meet-in-the-middle; POODLE (Padding Oracle On Downgraded Legacy Encryption): Эта уязвимость позволяет злоумышленнику расшифровать данные, передаваемые по SSL	SSH является надежным протоколом, но его уязвимости могут быть использованы злоумышленни ками для получения доступа к конфиденциаль ным данным. Поэтому важно регулярно обновлять программное обеспечение и следить за новыми уязвимостями.

[1]MITM (Meet in the Middle- это форма кибератаки, при которой для перехвата данных используются методы, позволяющие внедриться в существующее подключение или процесс связи) [9]

Рассмотрим размер пакета, который может быть передан с помощью приведённых способов вложенного туннелирования (табл. 2). Вид туннеля в столбце «Структура туннеля» представлены в виде схемы последовательных инкапсуляций используемых протоколов.

Известно, что максимальный блок передачи (MTU – maximum transmission unit) для Ethernet равен 1500 байт [10]. Размер служебных полей кадра Ethernet составляет 14 байт, к которым добавлены поля 802.1q VLAN (4 байта), 802.1ad QinQ (4 байта) и MACSec (32 байта). Таким образом, суммарный размер полей Ethernet равен 54 байтам, а размер кадра становится равным 1554 байтам. Допустим так же то, что механизмы сжатия не применялись.

В первой реализации к заголовкам Ethernet прибавляются служебные поля протокола IPsec (69 байт [11]), затем 32 байта отводится на SSL [12], к которым добавляются 20 байт заголовков IP. Во всех случаях последний является «открытым» заголовком, который используется для транспортировки по сети. Для внешних пакетов используются адреса пограничных маршрутизаторов, а внутренние адреса конечных точек содержатся во внутренних пакетах в зашифрованном виде [1].

Во втором варианте к Ethernet заголовкам добавлены 8 байт протокола L2TPv3[13], затем 29 байт протокола SSHv2[14].

ТАБЛИЦА II. РЕЗУЛЬТАТЫ РАСЧЁТОВ

№ п/п	VPN Tonnel 1	VPN Tonnel 2	Структура туннеля	Суммарный размер заголовков, байт	Размер поля данных, байт
1	SSL	Ipsec	IP[SSL[IPsec[Ethernet]]]	175	1325
2	SSH	L2TP	IP[SSH[L2TP[Ethernet]]]	111	1389
3	SSH	WireGuard	IP[SSH[WireGuard[Ethernet]]]	143	1357
4	SSH	OpenVPN	IP[SSH[OpenVPN[Ethernet]]]	172	1328

В третьем случае Ethernet инкапсулируется с помощью протокола WireGuard (40 байт [13]), далее, аналогично второму варианту, добавляются заголовки SSH и 20 байт заголовков IP.

Четвёртая реализация схожа с третьей, однако заголовки OpenVPN занимают 69 байт [14].

Увеличение размера MTU позволяет обеспечить заданную пропускную способность путем отправки меньшего числа пакетов. Большие пакеты подлежат фрагментации, тем самым снижая пропускную способность сети. Тем самым, при выборе оптимального варианта группового туннелирования необходимо руководствоваться не только надёжностью используемых протоколов, но и размером полезного блока с данными в одном пакете, так как это будет напрямую влиять на пропускную способность канала.

IV. ЗАКЛЮЧЕНИЕ И ДАЛЬНЕЙШИЕ ПЕРСПЕКТИВЫ ИССЛЕДОВАНИЯ

Использование двух разных уровней безопасности и разных алгоритмов шифрования могут усложнить использование MITM. В результате расчётов максимальный размер полезной нагрузки можно передать в реализации SSH-L2TP, однако такое соединение уязвимо для некоторых атак. Туннель SSH-WireGuard незначительно уступает предыдущему соединению в объеме передаваемых данных, но значительно выигрывает в надёжности. Сжатие вложенных заголовков может помочь в снижении задержек в том случае, если объем передаваемых превышает значение MTU.

Таким образом, рассмотренные варианты многоуровневого туннелирования могут быть применены для повышения защищённости открытых сетей и использованы в дальнейшем при моделировании процессов передачи данных. Также выбор количества и видов средств защиты должно основываться на анализе защищённости сетей связи [16].

СПИСОК ЛИТЕРАТУРЫ

[1] Запечников С.В., Милославская Н.Г., Толстой А.И. Основы построения виртуальных частных сетей. 2012.

[2] Рябко Е.И. Калейдоскоп VPN технологий // T-Comm-Телекоммуникации и Транспорт. 2009. №. S. С. 16-20.

[3] Жовнер П. Двойной VPN в один клик. Как легко разделить IP-адрес точки входа и выхода. URL: https://habr.com/ru/companies/vdsina/articles/469879

[4] Baker F., Bose P. Quality of Service (QoS) Signaling in a Nested Virtual Private Network. 2007. №. rfc4923.

[5] Taneja D., Tyagi S.S. Factors impacting the performance of data transferred via vpn // International Journal of Innovative Technology and Exploring Engineering. 2019. Т. 8. С. 2962-2966.

[6] Taneja D., Tyagi S.S. Nested Tunnel: Information Security in Hybrid Cloud Computing. 2018.

[7] Frahim J., Huang Q. Ssl remote access vpns (network security). Cisco Press, 2008.

[8] Shadowsocks-туннелирование корпоративного VPN. URL: https://habr.com/ru/companies/ruvds/articles/757848

[9] Розенкранц Л. Man-in-the-Middle: советы по обнаружению и предотвращению. URL: https://habr.com/ru/companies/varonis/articles/526632

[10] Arberg P. et al. Accommodating a Maximum Transit Unit/Maximum Receive Unit (MTU/MRU) Greater Than 1492 in the Point-to-Point Protocol over Ethernet (PPPoE). 2006. №. rfc4638.

[11] IPSec Overhead Calculator. URL: https://ipsec-overhead-calculator.netsec.us/

[12] Freier A., Karlton P., Kocher P. Rfc 6101: The secure sockets layer (SSL) protocol version 3.0. 2011.

[13] Визуальный калькулятор размера пакета. URL: https://www.baturin.org/tools/encapcalc/

[14] OmniSecu.com. URL: https://www.omnisecu.com/tcpip/ssh-packet-format.php

[15] Narkive. URL: https://openvpn-users.narkive.com/jzEE4wYX/overhead-added-to-each-packet-by-openvpn

[16] Коллинз М. Защита сетей. Подход на основе анализа данных. Москва: ДМК Пресс. 2020.