УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
сентябрь–октябрь 2020 Том 20 № 5 ISSN 2226-1494 http://ntv.itmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
September–October 2020 Vol. 20 No 5 ISSN 2226-1494 http://ntv.itmo.ru/en/
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

УДК 004.7 doi: 10.17586/2226-1494-2020-20-5-747-754

АНАЛИЗ АУТЕНТИЧНОСТИ ТРАФИКА НА ОСНОВАНИИ ДАННЫХ ЦИФРОВЫХ ОТПЕЧАТКОВ РЕАЛИЗАЦИЙ СЕТЕВЫХ ПРОТОКОЛОВ

С.М. Ишкуватов[a,b], И.И. Комаров[a]

[a] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
[b] АО «НИИ «Вектор», Санкт-Петербург, 197376, Российская Федерация
Адрес для переписки: sysroot0@gmail.com

Информация о статье
Поступила в редакцию 20.07.20, принята к печати 10.09.20
Язык статьи — русский

Ссылка для цитирования: Ишкуватов С.М., Комаров И.И. Анализ аутентичности трафика на основании данных цифровых отпечатков реализаций сетевых протоколов // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20. № 5. С. 747–754. doi: 10.17586/2226-1494-2020-20-5-747-754

Аннотация

Предмет исследования. Рассмотрена задача определения аутентичности трафика на основании данных цифровых отпечатков реализаций сетевых протоколов. Показаны способы описания цифровых отпечатков сетевых протоколов и характерные изменения исходных цифровых отпечатков в процессе передачи по различным каналам связи. Исследована возможность выявления использования средств анонимизации, обнаружения атак типа Man-in-the-Middle, вредоносных программ на основе анализа используемых цифровых отпечатков реализаций протоколов. Предложены способы совершенствования формата записи цифровых отпечатков для исключения коллизий отпечатков. **Метод.** Признаки каждой реализации существующего или потенциально возможного протокола передачи информации могут быть описаны цифровым отпечатком этой реализации и идентифицированы принимающей стороной. Оборудование связи на пути передачи информации может быть вынуждено менять некоторые из исходных параметров в силу своих внутренних ограничений или ограничений передающей среды. Принимающая сторона на основе предварительно подготовленных списков цифровых отпечатков идентифицирует текущую реализацию протокола передающей стороны с учетом допустимых характерных изменений узлами на пути следования передаваемых данных. Сравнивая исходный цифровой отпечаток с отпечатком, полученным сервером по определенным наборам параметров, принимающая сторона делает предположения о способах передачи данных, использовании клиентом средств анонимизации или стороннем вмешательстве в процесс передачи. На основе полученной в результате сопоставления цифровых отпечатков информации принимающая сторона принимает решение о возможности ведения сеансов связи с текущим отправителем. На протяжении всех сеансов связи с текущим отправителем получатель контролирует неизменность исходного цифрового отпечатка протокола активными и пассивными методами. **Основные результаты.** В ходе исследования продемонстрирована возможность определения способов сетевого подключения, средств анонимизации, подключения от потенциально опасной реализации на примере mitmproxy. **Практическая значимость.** Автоматизированный анализ цифровых отпечатков клиентских реализаций сетевых протоколов позволяет выявлять входящие соединения вредоносных приложений, сетевых роботов, факты использования клиентом средств анонимизации. Определение вредоносных реализаций по их цифровым отпечаткам является возможным не только на принимающей стороне, но и на всем участке сети по пути следования пакетов, что делает возможным блокировку таких соединений на границе сетей.

Ключевые слова
цифровой отпечаток, fingerprint, Man-in-the-Middle-атака, mitmproxy, анонимизация, сеть TOR

Благодарности
Работа подготовлена при финансовой поддержке Министерства науки и высшего образования Российской Федерации по соглашению № 075-15-2019-1707 от 22.11.2019 (идентификатор RFMEFI60519X0189, внутренний номер 05.605.21.0189).

doi: 10.17586/2226-1494-2020-20-5-747-754

TRAFFIC AUTHENTICITY ANALYSIS BASED ON DIGITAL FINGERPRINT DATA OF NETWORK PROTOCOL IMPLEMENTATIONS

S.M. Ishkuvatov[a,b], I.I. Komarov[a]

[a] ITMO University, Saint Petersburg, 197101, Russian Federation
[b] AO Vector Research Institute, Saint Petersburg, 197376, Russian Federation

Article info
Received 20.07.20, accepted 10.09.20
Article in Russian

For citation: Ishkuvatov S.M., Komarov I.I. Traffic authenticity analysis based on digital fingerprint data of network protocol implementations. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2020, vol. 20, no. 5, pp. 747–754 (in Russian). doi: 10.17586/2226-1494-2020-20-5-747-754

Abstract

Subject of Research. The problem of traffic authenticity determination based on digital fingerprint data of network protocol implementations is considered. Description methods for digital prints of network protocols and characteristic changes in the original digital prints during transmission over various communication channels are studied. The applicability of anonymization tools, detection of Man-in-the-Middle Attacks, and malware based on the digital fingerprint analysis of protocol implementations is researched. Ways of record format improvement for digital prints with the view to avoid collisions of prints are proposed. **Method.** Features of each implementation of an existing or potentially possible information transfer protocol can be described by a digital fingerprint of this implementation and identified by the receiving party. Communication equipment on the information transmission path may be forced to change some of the initial parameters due to its internal limitations or limitations of the transmitting environment. The receiving party identifies the current implementation of the transmitting party's protocol, based on pre-prepared lists of digital fingerprints, taking into account the permissible characteristic changes by nodes along the path of transmitted data. Comparing the original digital fingerprint with the fingerprint received by the server for certain sets of parameters, the receiving party makes assumptions about the methods of data transmission, the client's use of anonymization tools, or third-party intervention in the transmission process. Based on the information obtained as a result of comparing digital fingerprints, it takes a decision about the possibility of communication sessions with the current sender. Within all communication sessions with the current sender, the recipient controls the immutability of the original digital fingerprint of the protocol by active and passive methods. **Main Results.** In the course of the study, network connection methods, anonymization tools, and connection from a potentially dangerous implementation are determined on the example of mitmproxy. **Practical Relevance.** Digital fingerprint automated analysis of network protocol client implementations provides the detection of incoming connections of malicious applications, network robots, and confirmation facts about the client's applying of anonymization tools. Detection of malicious implementations by their digital fingerprints is possible not only on the receiving side, but on the entire network section along the path of packets, and therefore, blocks such connections at the network border.

Keywords
digital fingerprint, Man-in-the-Middle Attack, mitmproxy, anonymization, Tor network

Acknowledgements
The paper was prepared with the financial support of the Ministry of Science and Higher Education of the Russian Federation under the agreement No. 075-15-2019-1707 dated from 22.11.2019 (identifier RFMEFI60519X0189, internal number 05.605.21.0189).

Введение

Актуальность научной задачи выявления использования средств анонимизации и обнаружения атак типа Man-in-the-Middle (MITM) [1] определяется наличием объективного противоречия между практической потребностью проверки аутентичности устройства (пользователя) и недостаточным уровнем развития научно-методического аппарата автоматической идентификации аутентичного устройства (пользователя).

Идеология использования распределенных систем естественным образом предполагает высокую степень свободы при реализации протоколов взаимодействия на всех уровнях модели OSI (Open Systems Interconnection Basic Reference Model), что на практике приводит к модификации передаваемых данных и создает впечатление «стирания» уникальных признаков устройства. Вместе с тем, в качестве проверяемой гипотезы можно принять следующее утверждение «*практически невозможно полностью скрыть следы внешнего воздействия на передаваемое сообщение вне зависимости от уровня развития методов скрытия воздействия в канале передачи данных*». Предполагается, что решение упомянутой научной задачи может быть получено путем анализа цифрового отпечатка (ЦО) (Fingerprinting) [2] устройства, под которым понимается совокупность отличительных признаков конкретной реализации протокола передачи данных. Оценка энтропии и условий коллизий признаков, векторов признаков при реализации аутентичных протоколов взаимодействия позволит определить множество допустимых (аутентичных) преобразований над цифровыми отпечатками. Тогда задача определения аутентичности устройства сводится к задаче определения принадлежности исследуемого цифрового отпечатка к множеству допустимых.

Обсуждаемая научная задача осложнена группами факторов:

— высокой степенью свободы реализации большинства протоколов взаимодействия в модели OSI, что приво-

дит к естественной корректной модификации протокольной части передаваемых данных и ЦО в целом;
— отсутствием достоверной априорной информации об эталонном ЦО исследуемого устройства;
— широким использованием открытых точек доступа, предоставляющих условия для реализации различных атак, например [3];
— ограничениями на использование практически безопасной криптографии, связанными как с проблемой качества криптографических сертификатов [4], так и с ограниченностью вычислительных ресурсов, прежде всего, в технологии интернета вещей, например [5].

К решению научной задачи проверки аутентичности устройства предъявляются требования контролируемой ресурсоемкости и чувствительности для обеспечения возможности использования в широком спектре практических задач, который может быть разделен на две большие группы:

1) аутентификация для реализации бизнес-процессов информационной системы (например, таргетированная реклама [6]);
2) реализация задач обеспечения информационной безопасности системы, в том числе, но не ограничиваясь, противодействие MITM-атакам, противодействие бот-системам.

Обзор существующих решений

В настоящее время известны следующие открытые результаты в области частных решений обсуждаемой научно-технической задачи.

Проект *p0f* [7] анализирует отличительные особенности пакетов на основе собственной базы шаблонов и делает заключения об источнике кадра. Среди анализируемых признаков IP особая роль отводится значениям флагов IP и полей IP Identification, IP TOS. Для случая TCP рассматриваются SYN и SYN + ACK-пакеты, анализируются флаги, начальное значение размера TCP-окна, порядок следования и значения опциональных полей TCP. Вместе с тем проект имеет ряд особенностей, ограничивающих получение обобщенного решения задачи:

— игнорируются *значения* и даже *наличие* некоторых потенциально информативных не обязательных параметров пакетов TCP SYN (например, параметры TCP Fast Open и MPTCP [8], которые часто используются мобильными устройствами, имеющими несколько способов доступа в интернет);
— анализируется только отдельный TCP-пакет без рассмотрения динамики TCP-соединения (тогда как, например, есть возможность анализа следующих информативных пакетов сессии);
— не учитываются временные параметры взаимодействия (например, количество и времена отправки нескольких SYN-пакетов при неполучении ответа отправителем);
— при анализе ЦО используются шаблоны, содержащие константы, что приводит к ошибочной дифференциации пакетов от одного источника (например, смены параметров названий ОС с «Windows:7 or 8» на «Windows:NT kernel 6.x»).

Вызывают вопросы ряд правил анализа некоторых значений, например:

— правило обработки Maximum Transmission Unit (MTU) и Maximum Segment Size (MSS): на основе MSS p0f делает предположения о наличии на пути пакетов компрометирующих устройств, которые уменьшают его за счет собственных накладных расходов на передачу, например, VPN-туннель. Однако любое сетевое устройство может производить произвольную модификацию этих параметров, исходя из логики своей работы (например, установка флага IP ECN для информирования сторон соединения о своей предельной нагрузке), поэтому такие изменения могут считаться легитимными;
— правило определения компрометации на основании SYN-параметра: легитимное использование прокси-сервера подменит SYN-параметр легитимного отправителя, в отличии, например, от параметра TLS, который при использовании прокси измениться не может, а модифицируется лишь в случае MITM-атаки.

Проект *Satori* [9] является продолжением проекта p0f, снимающим ряд его ограничений. Так, например, вводится параметр веса признаков для разрешения возникающих коллизий, обеспечена возможность формирования информативных векторов из произвольного набора признаков, реализована возможность объединить в группу различные векторы признаков, ассоциированных с одним источником (листинг, строки 3 и 4).

Листинг. Пример записи группы ЦО ОС из БД *Satori*

```
1. <fingerprint name=”Linux … last_updated=”2014-06-23”>
2. <tcp_tests>
3. <test weight=”5” matchtype=”exact” tcpflag=”S” tcpsig=”29200:64:1:60:M1460,S,T,N,W5:.”/>
4. <test weight=”5” matchtype=”exact” tcpflag=”SA” tcpsig=”29200:64:1:52:M1460,N,N,S,N,W5:ZA”/>
5. </tcp_tests>
6. </fingerprint>
```

К числу практических ограничений этого проекта можно отнести:

— не введены мнемоники для обозначения значений редко используемых значений поля TCP Options;
— не предусмотрено задание диапазона возможных значений параметров ЦО, которые определены в проекте p0f.

Известно решение, применяемое сетевым инструментом Intercepter-NG [10] для пассивного определения ОС в одноранговой сети. Каждый отпечаток описывается с помощью первых трех байтов MAC-адреса, начального размера TCP-окна, начального TTL и мнемоник, описывающих порядок следования и значения полей TCP Options пакетов SYN и SYN + ACK.

Обсуждаемое решение имеет ряд принципиальных ограничений: используются только *статичные* шаблоны (векторы признаков); не обеспечивается полный охват диапазонов возможных значений признаков (например, согласно [11], под код производителя оборудования отводится 36 бит для MAC-адресов в диапазоне 00:50:C2); не поддерживается группировка шаблонов (векторов признаков); определен ограниченный набор операций над шаблонами (векторами признаков).

С точки зрения практического применения имеются сложности, связанные с ограниченным объемом БД шаблонов и отсутствием механизма анализа коллизий.

Известна работа [12] по получению и анализу ЦО реализаций протокола TLS. Предложено сопоставлять каждой конкретной реализации протокола набор поддерживаемых ей кривых CipherSuites, передаваемых в каждом приветственном пакете HelloClient каждой TLS-сессии. Несмотря на ряд достоинств, связанных с возможностью получения ЦО на любом устройстве канала связи без расшифровки трафика, практическая реализация показала неприемлемый уровень коллизий ЦО для разных групп реализаций протокола.

Для уменьшения количества коллизий в проекте JA3 [13] предложено: дополнить отпечаток CipherSuites значениями необязательных полей EllipticCurves и EllipticCurvesPointFormat; описать порядок следования передаваемых необязательных полей Extensions; формировать и вести обработку хеш, в том числе с учетом множественности хеш-отпечатка одной реализации.

Следует отметить неоднозначность использования хеш-функции для сокращения размера записи отпечатков – экспериментально подтверждено, что все отпечатки одной реализации имеют незначительные отличия, которые могут быть легко замечены и спрогнозированы в полной записи и полностью теряются в случае хеш-представления ЦО.

Кроме того, представляется избыточным хранение всех возможных хеш-значений при использовании защитного механизма GREASE for TLS [14], при котором один отпечаток будет записываться, например, 16 × 16 способами в формате JA3, тогда как в нехешированном виде он может быть записан всего одним шаблоном.

Проведенные эксперименты

Для изучения факторов и закономерностей, которые ведут к изменению цифровых отпечатков, было зарегистрировано доменное имя, на которое с помощью certbot получен действительный SSL-сертификат. В облачном сервисе Яндекс.Облако был развернут Ubuntu-сервер, на котором сконфигурирован Web-сервер Apache. Для записи всего принимаемого сервером трафика использована утилита tcpdump. Сервер получал запросы на загрузку определенного файла по ссылке, которая заранее не разглашалась. В один момент времени проводился только один запрос этого файла, что позволило сопоставить записанный сервером трафик исходному и исследовать их различия.

На *первом этапе* исследования различные устройства загружали файл с сервера без средств анонимизации, по разным каналам связи, с использованием стационарного соединения и с использованием передачи через 4G модем Yota. Далее эти же устройства запрашивали целевой файл с использованием разных средств анонимизации таких как TOR и анонимный VPN от NordVPN. Схема первого этапа представлена на рис. 1.

На втором этапе исследования на одно из Android-устройств был установлен небезопасный сертификат, устройство подключалось к сети Интернет через специально подготовленное устройство на базе Raspberry PI установленным программным обеспечением (ПО) mitmproxy [15]. На устройстве запрашивался один из интернет-сайтов, и записывался одновременно входящий и исходящий трафик согласно схеме на рис. 2. Изменения записей изучались с использованием программы Wireshark.

Результат сравнения исходного и полученного пакета первого этапа исследований представлен в табл. 1.

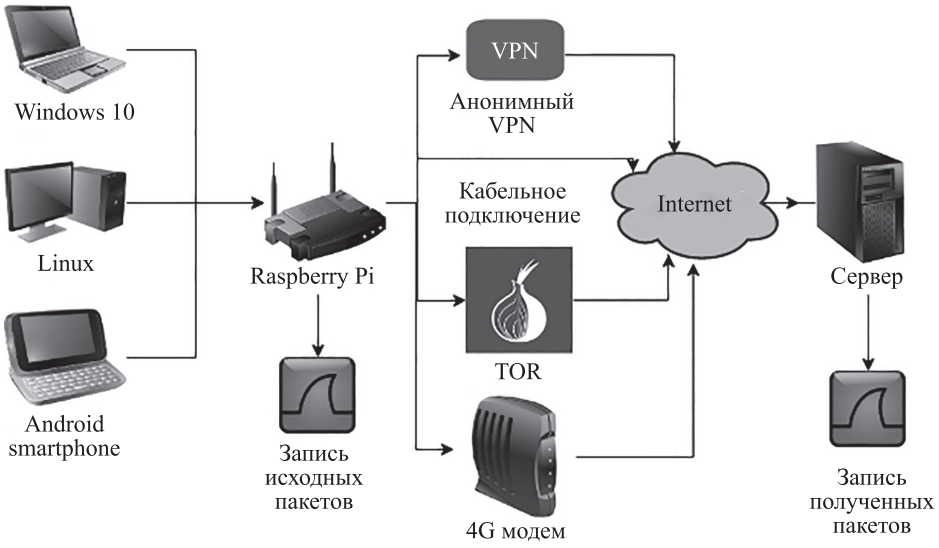

Рис. 1. Схема организации записи отправленного и полученного трафика

Рис. 2. Схема организации записи исходного и отправляемого программой mitmproxy трафика при MITM-атаке

Таблица 1. Полученные в ходе экспериментов отпечатки

Устройство и условия проведения исследования	Параметры пакета TCP SYN		Исходное значение поля IP identification не изменилось
	Значения TCP Options	Начальный размер TCP-окна	
Android-устройство, исходный пакет	MSS=1460, SACK, timestamp, nop, window_scale=8	65 535	—
Android-устройство, подключение по Wi-Fi к стационарной точке получено сервером	MSS=1460, SACK, timestamp, nop, window_scale=8	65 535	Да
Android-устройство, подключение через 4G модем, получено сервером	MSS=1360, SACK, nop, nop, nop, nop, nop, nop, nop, nop, nop, nop, nop, window_scale=8	65 535	Нет
Android-устройство, подключение через анонимный VPN, получено сервером	MSS=1460, SACK, timestamp, nop, window_scale=9	29 200	Нет
Linux, исходный пакет	MSS=1460, SACK, timestamp, nop, window_scale=7	64 240	—
Linux, подключение стационарно, получено сервером	MSS=1460, SACK, timestamp, nop, window_scale=7	64 240	Да
Linux, подключение через TOR, получено сервером	MSS=1460, nop, nop, SACK, nop, window_scale=11	42 340	Нет
Linux, подключение через анонимный VPN, получено сервером	MSS=1460, SACK, timestamp, nop, window_scale=9	29 200	Нет
Windows 10, исходный пакет	MSS=1460, nop, window_scale=8, nop, nop, SACK	64 240	—
Windows 10, подключение стационарно, получено сервером	MSS=1460, nop, window_scale=8, nop, nop, SACK	64 240	Да
Windows 10, подключение через 4G модем, получено сервером	MSS=1360, nop, window_scale=8, nop, nop, SACK	64 240	Да
Windows 10, подключение через TOR, получено сервером	MSS=1460, nop, nop, SACK, nop, window_scale=11	42 340	Нет
Windows 10, подключение через анонимный VPN, получено сервером	MSS=1460, SACK, timestamp, nop, window_scale=9	29 200	Нет

Во время проведения эксперимента с Windows 10 и 4G модемом получен интересный результат (рис. 3), когда сервер получил пакет с Time to life больше, чем устанавливал отправитель. В приведенном примере некоторое сетевое устройство на пути пакета вместо уменьшения значения TTL согласно RFC 791 увеличило его, в случае потери пакета ICMP-дейтаграмма от устройства, обнаружившего достижение TTL нулевого значения, вероятно, не дошла бы до отправителя пакета.

На основании полученных данных можно сделать вывод, что все проверенные средства анонимизации меняют исходный отпечаток пакета SYN определенным образом. Независимо от того, какая нода TOR будет являться последней, ее трафик может быть определен принимающей стороной по характерному цифровому SYN-отпечатку, при этом сам трафик и TLS-отпечаток останутся неизменными. При прохождении пакетов по сети с меньшим MTU (4G или IPSec туннель) исходный отпечаток также изменится, за счет уменьше-

а

```
Internet Protocol Version 4,
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0xba26 (47654)
  ▸ Flags: 0x4000, Don't fragment
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x9759 [validation disabled]
    [Header checksum status: Unverified]
    Source:
    Destination:
Transmission Control Protocol, Src Port: 61521, Dst Port: 443, Seq: 0, Len: 0
    Source Port: 61521
    Destination Port: 443
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x002 (SYN)
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0x2d7b [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, N
  ▸ [Timestamps]
```

б

```
Internet Protocol Version 4,
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x60 (DSCP: CS3, ECN: Not-ECT)
    Total Length: 52
    Identification: 0xba26 (47654)
  ▸ Flags: 0x4000, Don't fragment
    Time to live: 249
    Protocol: TCP (6)
    Header checksum: 0x79ed [validation disabled]
    [Header checksum status: Unverified]
    Source:
    Destination:
Transmission Control Protocol, Src Port: 41536, Dst Port: 443, Seq: 0, Len: 0
    Source Port: 41536
    Destination Port: 443
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x002 (SYN)
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0xd7e4 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, N
  ▸ [Timestamps]
```

Рис. 3. Зафиксированное при проведении эксперимента некорректное изменение значения поля IP TTL: заголовок исходного пакета (*а*) и полученного сервером (*б*)

ния значения параметра TCP Maximum Segment Size, но будут сохранены остальные исходные параметры TCP, их хронология следования и некоторые значения полей IP. При использовании анонимизирующих прокси или специальных платных анонимизирующих VPN цифровой отпечаток SYN будет определяться ПО такого сервиса, а исходная информация будет потеряна. Сервер, принимающий такие соединения, может определять соединения от анонимных источников и дополнительно применять к ним определенные политики, для предотвращения злоупотреблений со стороны таких клиентов.

Для демонстрации искажения исходного TLS-трафика браузера Android-смартфона на втором этапе эксперимента приведен результат сравнения полей TLS Client Handshake до и после MITM-устройства одного из интернет-ресурсов. Отличия параметров представлены в табл. 2.

При сравнении результатов видны различия в порядке следования параметров и значений полей паке-

Таблица 2. Различия исходных и отправленных mitmproxy параметров TLS

Исходная версия SSL	TLSv1.3
Отправленная mitmproxy версия SSL	TLSv1.2
Исходный список поддерживаемых клиентом ciphersuites	Reserved (GREASE), TLS_CHACHA20_POLY1305_SHA256, TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_3DES_EDE_CBC_SHA
Отправленный mitmproxy список поддерживаемых клиентом ciphersuites	TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256, TLS_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_EMPTY_RENEGOTIATION_INFO_SCSV
Исходный список присутствующих опциональных полей Extension	Reserved (GREASE), server_name, extended_master_secret, renegotiation_info, supported_groups, ec_point_formats, session_ticket, application_layer_protocol_negotiation, status_request, signature_algorithms, signed_certificate_timestamp, key_share, psk_key_exchange_modes, supported_versions, compress_certificate, Reserved (GREASE), padding
Отправленный mitmproxy список присутствующих опциональных полей Extension	server_name, ec_point_formats, supported_groups, session_ticket, application_layer_protocol_negotiation, encrypt_then_mac, extended_master_secret, signature_algorithms, supported_versions, psk_key_exchange_modes, key_share

тов. Различия обусловлены тем, что для полностью скрытной работы MITM-серверу нужно поддерживать все существующие ciphersuite, которые клиент может передать в своем пакете, корректно обрабатывать все возможные значения полей Extensions (в том числе не стандартизованные) и копировать их так, чтобы их корректность сохранилась независимо от вносимых им изменений, что на практике реализовать сложно по причине большой трудоемкости создания такого ПО и объема требований к аппаратной составляющей такого устройства. Также следует отметить, что TLS-отпечаток mitmproxy будет постоянным, и принимающий сервер, анализируя входящие отпечатки, способен его распознать и отправить предупреждающее сообщение своему клиенту.

Обсуждение результатов исследования

С точки зрения непосредственного практического применения результатов исследования можно выделить следующие рекомендации.

При использовании сети TOR сервер, принимающий соединение, может установить факт использования этой сети по цифровому отпечатку SYN-пакета выходных узлов; ведение списка адресов всех выходных узлов TOR в данном случае нецелесообразно. Для повышения уровня информационной безопасности администратору такого сервера целесообразно ограничить принятие соединений от TOR-сети, если они не связаны с общим доступом, например, следует запретить прием TOR-соединений для протоколов SSH, RDP.

Информативной совокупностью признаков, свидетельствующих о применении бот-механизмов, является использование клиентом сети TOR (идентифицированных по отпечатку SYN), браузера, не являющегося стандартным, входящим в состав Tor Browser Bundle, что будет устанавливается по TLS-отпечатку. Для противодействия использованию бот-механизмов рекомендуется вводить дополнительные проверки, например, проверка CAPTCHA, использование методов лингвистической идентификации [16].

Продуктивным механизмом противодействия вредоносному ПО является формирование БД TLS-отпечатков, которые были выявлены при использовании вредоносных и потенциально опасных программ (например, отпечатков TLS-программ загрузки сайтов целиком, такие как HTTrack или GetLeft).

Экспериментально подтверждено, что трафик всех приложений, использующих сервис mitmproxy, будет обязательно изменен. Это обеспечивает возможность принимающему серверу на основании ЦО, характерного для mitmproxy, выявить наличие между собой и клиентом MITM-сервера. Для сокрытия своего присутствия атакующему следует полностью транслировать исходные IP атрибуты ECN, DSCP, TTL; повторять параметры TCP SYN; на протяжении всей сессии следить, чтобы размер TCP-окна был меньше или равен размеру TCP-окна клиента. Однако выполнение таких требований возможно только в статических условиях. Для предотвращения потенциальной возможности сокрытия присутствия атакующего требуется разработка методов и средств проактивного противодействия.

Заключение

В ходе исследования подтверждена гипотеза о «практической невозможности полностью скрыть следы внешнего воздействия на передаваемое сообщение вне зависимости от уровня развития методов скрытия воздействия в канале передачи данных» на основе анализа цифровых отпечатков реализаций протоколов.

Выявлена ограниченность известных подходов к решению задачи идентификации реализации протоколов связи, допускающих коллизии признаков различных реализаций. Для снятия этих ограничений предлагается: разработка унифицированного формата векторов цифровых отпечатков, позволяющих отражать наличие и значения всех существующих и перспективных TCP-параметров; использование цифровых отпечатков TLS в развернутом формате без применения хеш-представления, снижающего информативность данных о реализации протокола; использование высоко информативных данных о динамике сессии, например, дополнение цифровых отпечатков признаками времени ретрансмиссии пакета.

Перспективными направлениями дальнейших исследований являются:

— определение информативности признаков, входящих в цифровые отпечатки;
— определение продуктивных алгоритмов обработки цифровых отпечатков с целью обнаружения компрометации сессии;
— разработка методов выявления родственных реализаций протоколов, имеющих различные цифровые отпечатки;
— разработка активных методов противодействия MITM-атак, когда клиент и сервер, меняя передаваемые атрибуты, сами провоцируют атакующего внести в трафик раскрывающие его присутствие изменения, например: имитация потери пакета и замер времени его ретрансмиссии с последующим сравнением его с эталонным значением времени ретрансмиссии; имитация переполнения буфера приема отправкой клиентом TCP-пакета с размером окна, равным нулю, и последующий контроль получения такого пакета сервером.

Литература

1. Man-in-the-middle attack [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Man-in-the-middle_attack (дата обращения: 19.07.2020).
2. Shu G., Lee D. Network protocol system fingerprinting - A formal approach // Proc. INFOCOM 2006: 25th IEEE International Conference on Computer Communications. 2006. P. 4146810. doi: 10.1109/INFOCOM.2006.157
3. Carnut M., Gondim J. ARP spoofing detection on switched Ethernet networks: A feasibility study // Proc. of the 5th Simposio Seguranca em Informatica. 2003.
4. Liu H., Zhang Y., Wang H., Yang W., Li J., Gu D. TagDroid: hybrid SSL certificate verification in android // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015. V. 8958. P. 120–131. doi: 10.1007/978-3-319-21966-0_9
5. Smith S. The Internet of Risky Things: Trusting the Devices That Surround Us. O'Reilly Media, Inc., 2017. 240 p.
6. Куликова О.М., Суворова С.Д. Таргетированная реклама как инструмент построения коммуникаций с целевой аудиторией // Экономика и бизнес: теория и практика. 2020. № 3-2(61). P. 98–102 [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/targetirovannaya-reklama-kak-instrument-postroeniya-kommunikatsiy-s-tselevoy-auditoriey (дата обращения: 10.09.2020). doi: 10.24411/2411-0450-2020-10218
7. Zalewski M. p0f v3 [Электронный ресурс]. URL: http://lcamtuf.coredump.cx/p0f3/ (дата обращения: 19.07.2020).
8. Transmission Control Protocol (TCP) Parameters [Электронный ресурс]. URL: https://www.iana.org/assignments/tcp-parameters/tcp-parameters.xhtml (дата обращения: 19.07.2020).
9. Satori [Электронный ресурс]. URL: https://github.com/xnih/satori/blob/master/fingerprints/tcp.xml (дата обращения: 19.07.2020).
10. Intercept the planet! [Электронный ресурс]. URL: http://intercepter-ng.blogspot.com/ (дата обращения: 19.07.2020).
11. Laurent D. Ethernet vendor codes, and well-known MAC addresses. The first single application for the entire DevOps lifecycle — GitLab. URL: https://gitlab.com/wireshark/wireshark/raw/master/manuf (дата обращения: 19.07.2020).
12. Husák M., Čermák M., Jirsík T., Čeleda P. HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting // EURASIP Journal on Information Security. 2016. V. 2016. N 1. P. 6. doi: 10.1186/s13635-016-0030-7
13. JA3 — A method for profiling SSL/TLS Clients [Электронный ресурс]. URL: https://github.com/salesforce/ja3 (дата обращения: 19.07.2020).
14. Chrome Platform Status. GREASE for TLS. Last updated on 2017-06-14 [Электронный ресурс]. URL: https://www.chromestatus.com/feature/6475903378915328 (дата обращения: 19.07.2020).
15. mitmproxy — an interactive HTTPS proxy [Электронный ресурс]. URL: https://mitmproxy.org/ (дата обращения: 19.07.2020).
16. Воробьева А.А. Отбор информативных признаков для идентификации Интернет-пользователей по коротким электронным сообщениям // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 1. С. 117–128. doi: 10.17586/2226-1494-2017-17-1-117-128

References

1. *Man-in-the-middle attack*. Available at: https://en.wikipedia.org/wiki/Man-in-the-middle_attack (accessed: 19.07.2020).
2. Shu G., Lee D. Network protocol system fingerprinting - A formal approach. *Proc. INFOCOM 2006: 25th IEEE International Conference on Computer Communications*, 2006, pp. 4146810. doi: 10.1109/INFOCOM.2006.157
3. Carnut M., Gondim J. ARP spoofing detection on switched Ethernet networks: A feasibility study. *Proc. of the 5th Simposio Seguranca em Informatica*, 2003.
4. Liu H., Zhang Y., Wang H., Yang W., Li J., Gu D. TagDroid: hybrid SSL certificate verification in android. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 8958, pp. 120–131. doi: 10.1007/978-3-319-21966-0_9
5. Smith S. *The Internet of Risky Things: Trusting the Devices That Surround Us*. O'Reilly Media, Inc., 2017, 240 p.
6. Kulikova O.M., Suvorova S.D. Targeted advertising as a tool for building communications with the target audience. *Economy and business: theory and practice*, 2020, no. 3-2(61), pp. 98–102. Available at: https://cyberleninka.ru/article/n/targetirovannaya-reklama-kak-instrument-postroeniya-kommunikatsiy-s-tselevoy-auditoriey (accessed: 10.09.2020). (in Russian). doi: 10.24411/2411-0450-2020-10218
7. Zalewski M. *p0f v3*. Available at: http://lcamtuf.coredump.cx/p0f3/ (accessed: 19.07.2020).
8. *Transmission Control Protocol (TCP) Parameters*. Available at: https://www.iana.org/assignments/tcp-parameters/tcp-parameters.xhtml (accessed: 19.07.2020).
9. *Satori*. Available at: https://github.com/xnih/satori/blob/master/fingerprints/tcp.xml (accessed: 19.07.2020).
10. *Intercept the planet!* Available at: http://intercepter-ng.blogspot.com/ (accessed: 19.07.2020).
11. Laurent D. *Ethernet vendor codes, and well-known MAC addresses. The first single application for the entire DevOps lifecycle — GitLab.* Available at: https://gitlab.com/wireshark/wireshark/raw/master/manuf (accessed: 19.07.2020).
12. Husák M., Čermák M., Jirsík T., Čeleda P. HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting. *EURASIP Journal on Information Security*, 2016, vol. 2016, no. 1, pp. 6. doi: 10.1186/s13635-016-0030-7
13. *JA3 — A method for profiling SSL/TLS Clients*. Available at: https://github.com/salesforce/ja3 (accessed: 19.07.2020).
14. *Chrome Platform Status. GREASE for TLS. Last updated on 2017-06-14*. Available at: https://www.chromestatus.com/feature/6475903378915328 (accessed: 19.07.2020).
15. *mitmproxy — an interactive HTTPS proxy*. Available at: https://mitmproxy.org/ (accessed: 19.07.2020).
16. Vorobeva A.A. Dynamic feature selection for web user identification on linguistic and stylistic features of online texts. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2017, vol. 17, no. 1, pp. 117–128. (in Russian). doi: 10.17586/2226-1494-2017-17-1-117-128

Авторы

Ишкуватов Сергей Маратович — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация; инженер-программист 1 категории, АО «НИИ «Вектор», Санкт-Петербург, 197376, Российская Федерация, ORCID ID: 0000-0002-4006-3693, sysroot0@gmail.com

Комаров Игорь Иванович — кандидат физико-математических наук, доцент, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 57194932349, ORCID ID: 0000-0002-6542-4950, I_krov@mail.ru

Authors

Sergei M. Ishkuvatov — Postgraduate, ITMO University, Saint Petersburg, 197101, Russian Federation; Software Engineer of the 1st category, AO Vector Research Institute, Saint Petersburg, 197376, Russian Federation, ORCID ID: 0000-0002-4006-3693, sysroot0@gmail.com

Igor I. Komarov — PhD, Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 57194932349, ORCID ID: 0000-0002-6542-4950, I_krov@mail.ru