ПОДХОД К СОКРЫТИЮ ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ МЕТОДОМ ОБФУСКАЦИИ СЕТЕВОГО ТРАФИКА

Д.И. Ударцева, А.В. Мансуров
Алтайский государственный университет

Защита передаваемой по сети информации, обеспечение ее конфиденциальности и целостности являются важными задачами информационной безопасности. На сегодняшний день существует множество различных под-ходов и способов по их решению, которые включают в себя применение специализированных протоколов транспортирования данных, осуществление шифрования передаваемой информации, либо же ее сокрытие путем вложения одного протокола в другой (туннелирование). Однако, подобные подходы, будучи подробно документированными, всегда легко обнаружить в процессе анализа сетевого трафика, что значительно облегчает работу для потенциальных злоумышленников. В этом случае перспективным направлением является применение специальных схем и методов «запутывания» и «обмана», которые позволяют растворить передаваемые данные во множестве сетевых транзакций. Методики имитации какого-то сетевого протокола для замаскированной передачи целевой информации активно развиваются [1-3,9-10], и на сегодняшний день уже есть достаточно большое количество реально работающих решений – таких, как аноннимная сеть Tor[4], методики StegoTorus[5] и другие.

Успешность реализованных подходов и неограниченная свобода действия дают возможность дорабатывать, адаптировать и комбинировать принципиальные моменты предлагаемых решений для получения своих собственных уникальные методик обфускации сетевого обмена и успешного сокрытия передаваемой информации при ее транспортировании по незащищенным открытым каналам сетевого обмена.

Под обфускацией сетевых протоколов понимается скрытие или маскировка истинного протокола передачи данных под имитированный протокол («обложку») с целью предотвращения обнаружения действительного протокола. Интуитивно, обфускация означает, что никой из анализаторов трафика не сможет распознать трафик, генерируемый системой обхода, а именно определить конечные точки, участвующие в обходе. Обфускационные сетевые методы обычно используются с целью защиты передаваемых данных от глубокого анализа истинной природы протокола.

Очень важным для обфускации является выбор протокола сокрытия или протокола «обложки». Трафик, используемый для обфускации, не должен блокироваться по политическим или экономическим причинам. Имитировать не популярный протокол бессмысленно, потому что будет блокироваться как сам протокол, так и его имитации[6].

Использование метода глубокого анализа сетевого трафика (DPI) с целью блокировки нежелательного трафика вдохновило исследователей и активистов на изучение различных контрмер, направленных на обфускацию сетевых пакетов. Среди всех подходов к обфускации сетевого трафика можно выделить четыре категории: шифрование, рандомизация, имитация, и туннелирование.

Имитация, как один из методов обфускации сетевого трафика, пытается сделать пакеты нагрузки похожими на другие известные протоколы, которые будут определены анализаторами как допустимые, а значит не будут заблокированы. Общим примером является создание полезных нагрузок похожими на HTTP протокол, который редко блокируется из-за его вездесущности в любом сетевом трафике. Обфускация сетевого трафика происходит с помощью HTTP-заголовков проекта с пакетами Tor [4], так же используется обфускация с применением метода стеганографии против конкретного типа противника, напимер, системы глубокого

анализа данных (DPI). Stegotorus [5] и SkypeMorph [7], примеры использования стегано-графических подходов к имитации выглядеть как HTTP траффик и Skype, соответственно. К счастью, у данных методов непомерно низкая производительность[8].

Минималистский, но гибкий подход к мимикрии можно найти в форматирующем шифровании (FTE). Как реализовано в Tor, uProxy и в других местах, FTE позволяет программистам использовать регулярные выражения для указания формата зашифрованных текстов. Поскольку регулярные выражения также используются в DPI методе, то зачастую можно неправильно классифицировать протокол[11].

Основная задача исследователей обфускации сделать пакетные нагрузки похожими на «обложки» протокола, но не использовать его реализацию по назначению. Это означает, что синтаксис и семантика сообщений, исходящие из систем скрытия протокола, могут часто значительно отличаться от сообщений, соответствующих протоколу обложки. В качестве одного примера, имитирующий обфускатор HTTP может генерировать сообщение, являющееся GET запросом, после того, как запрос был получен фомируется HTTP-ответ, тем самым не нарушая правильную HTTP-семантику.

В предлагаемом подходе весь трафик частями маскируется под обмен с имитированием некоторых стандартных сетевых протоколов прикладного уровня. Для связи клиента и сервера используется шаблон обфускации, в котором содержится структура для каждого метода обфускации. Причем первый из применяемых методов обфускации (имитация первого протокола) известна заранее и клиентской части и серверной, далее методы обфускации протоколов выбираются случайным образом.

В процессе реализации обфускации сетевого обмена, отправитель данных выступает в качестве сервера, а получатель – в качестве клиента. В соответствии с этим в качестве данных клиента выступают имитированные запросы, ответ на которые имитирует сервер и добавляет данные для клиента, встраиваемые после «поддельного» заголовка. Для реализации двухстороннего обмена каждая из сторон имеет клиентскую и серверную части.

На рисунках 1 и 2 представлена система сокрытия данных транспортируемой по сети информации под протоколы прикладного уровня. Данные передаются между двумя участниками с использованием двухстороннего обмена. Такой метод обеспечивает сокрытие сетевого обмена данными на основе обфускации протоколов высокого уровня между сторонами A и B

В качестве выходных данных выступает трафик сетевого обмена между сторонами A и B. Для осуществления метода обфускации его разбивают на некоторое количество кусков/порций общее количество которых достигает значения N. После этого данные в этом же порядке поступают на блок обфускатора №1, в котором встроен «Шаблон обфускации A». С помощью данного шаблона определяется структура встраивания имитированных прикладных протоколов.

Процесс имитации протокола для сокрытия действительно подразумевает под собой обфускацию сетевого трафика. На выходе стороны A формируется ряд данных со встроенными в них техническими заголовками (RFC–заголовками), которые необходимы для идентификации типа протокола. Следовательно, первая порция трафика маскируется под начальный протокол (Метод 1), имитируя запрос клиента и ответ сервера со встроенной порцией данных. Выбор Метода 2 для обфускации второй порции трафика осуществляется случайным образом, далее процесс продолжается пока все порции трафика не будут замаскированы под протоколы прикладного уровня. В конечном итоге вся эта цепочка протоколов обфускации формирует «уникальный ключ» (элемент обфускации), с помощью которого сторона B сможет восстановить данные в первоначальном виде.

На стороне В происходит обратный процесс сборки конечного результата. Трафик со встроенными заголовками и служебными данными поступает на обфускатор №2, в котором по «уникальному ключу» можно с легкостью выбрать необходимые данные, сборка полученного результата. Для осуществления деобфускации используется «Шаблон обфускации А'», который содержит в себе точно такую же структуру имитации, что и «Шаблон обфускации А». Содержимое данных шаблонов должно быть идентичным, иначе корректно восстановить данные будет невозможно. Шаблоны обфускации обоих сторон содержат копию «уникального ключа» – это главный элемент обфускации, без знания которого восстановить первоначальный вид передаваемых данных не получиться, т.к. не будет известен правильный порядок встраиваемых заголовков для осуществления обфускации.

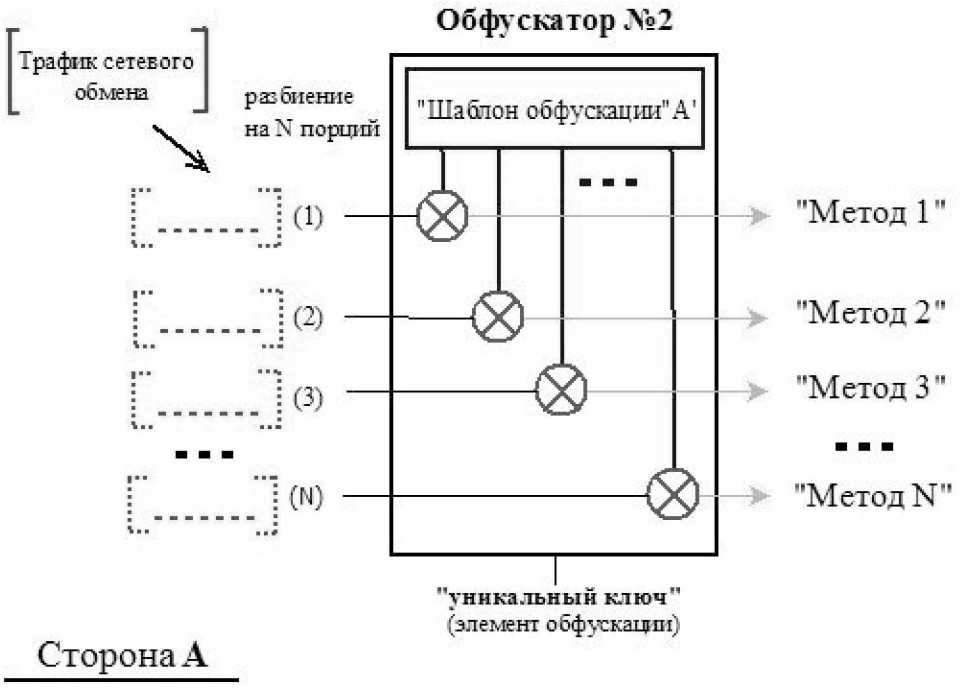

Рисунок 1. Методы имитации сетевого трафика с добавлением служебных данных, Сторона А.

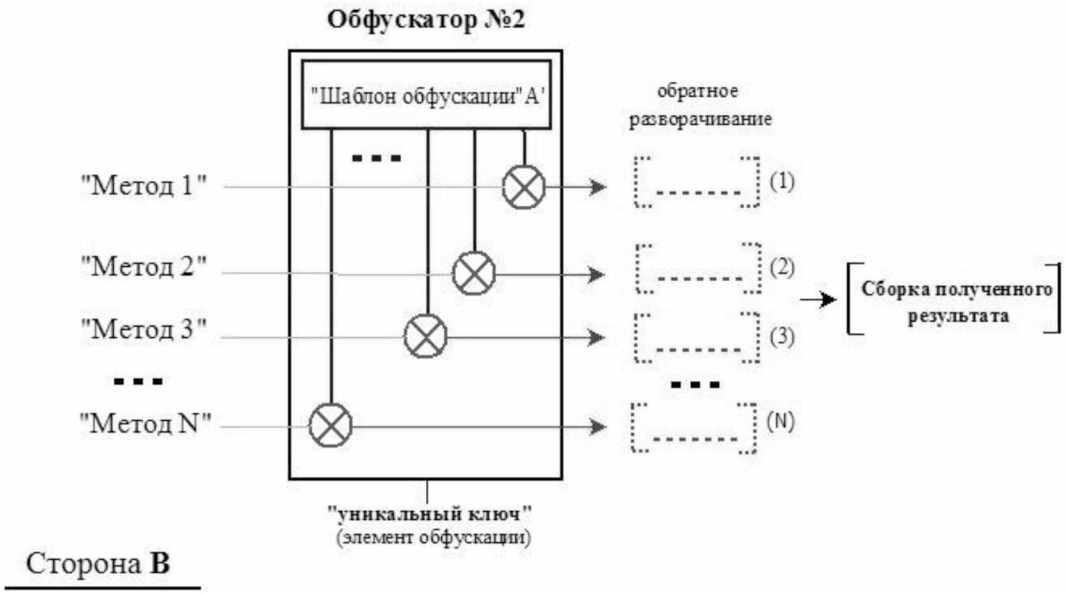

Рисунок 2. Методы имитации сетевого трафика с добавлением служебных данных, Сторона В.

Заключительным этапом данной концепции является формирование конечного результата, путем сборки полученных порций данных. Начальное соединение осуществляется на порту TCP сокета, который известен изначально сторонам сетевого обмена. Метод для первого соединения определяется на этапе инициализации данного метода и отправляется на сторону получателя в качестве служебных данных номер следующего порта, метода и размер передаваемого фрагмента сетевого трафика. Номер порта и метод обфускации при последующих запросах выбираются случайным образом. Запрос клиента к серверу формируется в соответствии с методом шаблона обфускации, какой метод применяется такой шаблон и выбирается.

При взаимодействии этих сторон имитируется незашифрованный трафик с помощью клиентского запроса и сгенерированного ответа сервера в соответствии с типом соединения. Сервер и клиент полагаются на предварительно записанную трассировку запросов и ответов. Общение между клиентом и сервером происходит по следующей схеме: клиент посылает запрос, ждет от сервера ответ, затем он может отправить еще один запрос, и так далее. Для параллельной отправки большого количества запросов клиенты открывают несколько соединений к одному серверу. Каждый запрос содержит «метод», который указывает на то, как серверу подготовить необходимый ответ.

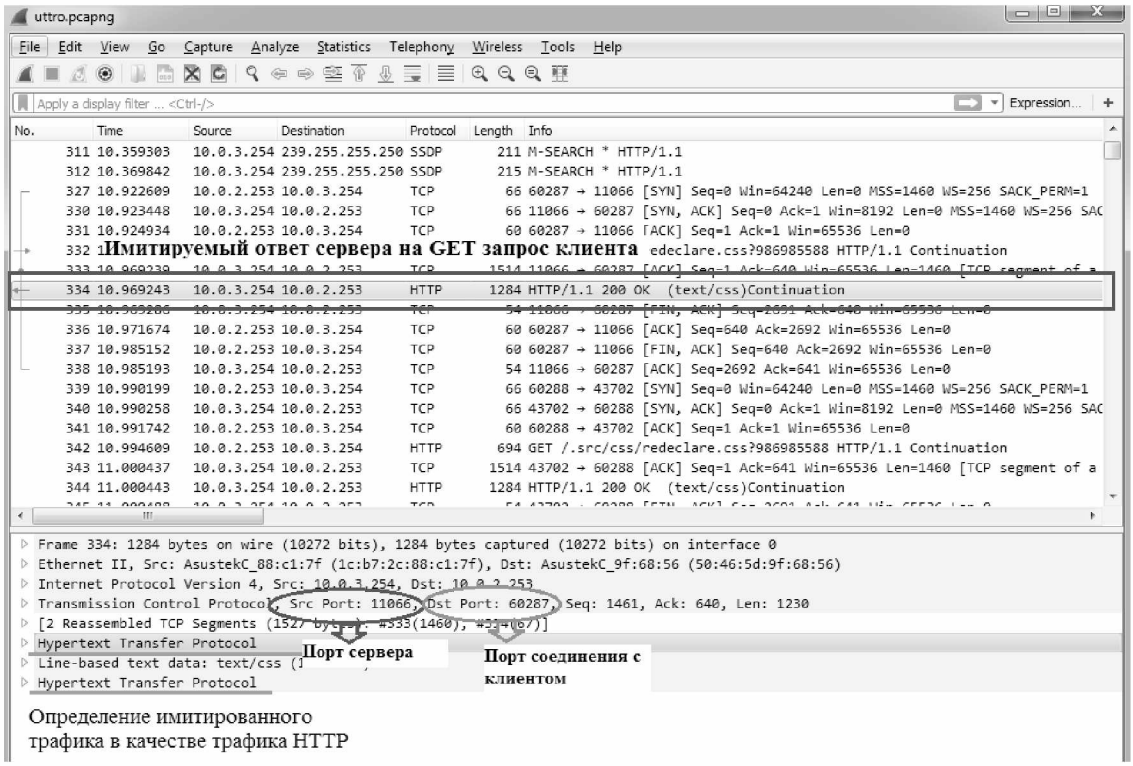

Рисунок 3. Применение шаблона обфускации трафика методом имитирования HTTP протокола. Имитация ответа сервера.

Определение номера протокола передачи данных необходимо для работы Шаблона обфускации A и Шаблона обфускации A’. Без знания номера протокола обфускации невозможно восстановить данные после их передачи. Следовательно, номер следующего протокола должен знать и клиент и сервер. При этом возникает проблема предусмотреть возможность сигнализирования о следующем выбранном методе из шаблона обфускации, в соответствии с типом соединения/протоколом, портом и другой служебной информацией, необходимой для корректной работы обфускаторов. В данном методе сетевого обмена данные передаются вместе со встроенными служебными данными, которые располагаются в виде заголовков в начале каждого фрагмента данных. Встраиваемые данные представляют собой

характерные заголовки, состоящие из служебных полей, параметры которых для каждого сетевого протокола индивидуальны. Большинство систем анализа трафика извлекают из порций трафика начальные биты, в которых и содержатся служебные данные для каждого из протоколов.

В качестве шаблонов для обфускации имплементированы (на данный момент) шаблоны служебной информации протокола HTTP (рис.3), протокола SMTP и протокола FTP. Увеличение числа шаблонов будет повышать надежность сокрытия сетевого обмена путем увеличения выбора имитируемых протоколов.

Применяя данную систему защиты, можно скрыть трафик сетевого обмена данными между сторонами A и B и «обмануть» стандартные анализаторы трафика, которые определяют имитированный протокол по служебным данным и полям сетевых протоколов в своем традиционном варианте использования.

Библиографический список.

1. Современное состояние исследований в области обфускации программ: определения стойкости обфускации[Электронный ресурс]/Н.П. Варновский [и др.]// сб. научн. тр. ИСПРАН/Институт проблем информационной безопасности/– М. - 2014. – Т.26. - №. 3. – URL: http://www.ispras.ru/proceedings/docs/2014/26/3/isp_26_2014_3_167.pdf.
2. Petitcolas F., Anderson R., Kuhn M. Information Hiding // A Survey. Proceedings of the IEEE, specialissue on protection of multimedia content – 1999 - C.1062 –1078.
3. Pfitzmann, A., Hansen M. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management [Электронный ресурс] // 2010. URL:http://dud.inf.tudresden.de/literatur/Anon_Terminology_v0.34.pdf
4. Официальный сайт Tor [Электронный ресурс] // URL: https://w.w.w.torproject.org.
5. Weinberg Z., Wang J., Yegneswaran V., Briesemeister L., Cheung S., Wang F., Boneh D., StegoTorus A Camouflage Proxy for the Tor Anonymity System // CCS '12 Proceedings of the 2012 ACM conference on Computer and communications security – 2012. – C. 1-7.
6. Houmansadr A., Brubaker C., Shmatikov V. The Parrot is Dead: Observing Unobservable Network Communications [Электронный ресурс] // In Proc. of: Symposium on Security & Privacy, 2013, IEEE. – URL: https://people.cs.umass.edu/~amir/papers/parrot.pdf.
7. Moghaddam H.M., Li, B., Derakhshani, M., и др. SkypeMorph: protocol obfuscation fot Tor bridges // In: Proceedings of the 2012. ACM Conference on Computer and Communications Security – 2012. - C. 2-6.
8. Кравцов К.Н. Передача данных в сетях с динамической рандомизацией адресного пространства // Труды XVII Международной конференции DAMDID/RCDL'2015 «Аналитика и управление данными в областях с интенсивным использованием данных - 2015 – С.273-277.
9. Wang. Q., Gong, X., Nguyen, G.T., Houmansadr, A., Borisov, N. Censor-Spoofer: asymmetric communication using web browsing // ACM Conference on Computer and Communications Security - 2012.
10. Wright C.V., Ballard L., Monrose F., Masson G.M. Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob // Proceedings of 16th USENIX Security Symposium - 2007.
11. Identifying and Measuring Internet Traffic: Techniques and Considerations [Электронный ресурс]. – URL: https://www.sandvine.com/hubfs/downloads/archive/whitepaper-internet-traffic-classification.pdf /Headquarters Sandvine Incorporated ULC Waterloo, Ontario Canada, 2015.