DOI 10.37882/2223–2966.2022.04.19

МЕТОД ОБНАРУЖЕНИЯ СЕТЕВОЙ СТЕГАНОГРАФИИ НА ОСНОВЕ СТАТИСТИЧЕСКОГО РАСПРЕДЕЛЕНИЯ ПОЛЕЙ СЕТЕВЫХ ПАКЕТОВ[1]

A METHOD FOR DETECTING NETWORK STEGANOGRAPHY BASED ON THE STATISTICAL DISTRIBUTION OF NETWORK PACKET FIELDS

A. Krasov

Summary. Classification of steganographic methods of information transformation is described in the basic model of threats to the security of personal data during their processing in personal data information systems, approved by the FSTEC of the Russian Federation in 2008. An analysis of existing programs for creating steganographic channels shows that most of the methods presented in it have not been in demand in practice, at the same time, new methods have appeared that are not included in this model. The article presents the results of work on the Grant-I B5/2020 project, proposals for improving the basic model.

Keywords: network packet, steganography, FSTEC, stegulation, steganalysis.

Красов Андрей Владимирович
К.т.н., доцент, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
krasov@inbox.ru

Аннотация. Классификация стеганографических методов преобразования информации описано в базовой модели угроз безопасности персональных данных при их обработке в информационных системах персональных данных, утверждённой ФСТЭК РФ в 2008 году. Анализ существующих программ для создания стеганографических каналов показывает, что большинство из представленных в нем методов так и остались не востребованы на практике, в то же время появились новые методы, не вошедшие в данную модель. В статье приводятся результаты работы по проекту Грант-ИБ 5/2020, предложения по совершенствованию базовой модели.

Ключевые слова: сетевой пакет, стеганография, ФСТЭК, стеговложение, стегоанализ.

Очевидно, что внедряемый цифровой водяной знак должен обладать некоторой степенью устойчивости, то есть противостоять удалению или модификации. Считается, что заранее можно предугадать, содержится ли водяной знак или нет. В настоящее время существуют методы, позволяющие защитить подобным образом от нелегального копирования различные цифровые объекты, такие как файлы мультимедиа и программ. В частности, ряд публикаций, например, посвящен методам внедрения скрытой информации в программы. В данной статье речь пойдет о методе, размещающем секретное сообщение в неиспользуемых местах секции исполняемых файлов формата Portable Executable (PE).

Portable Executable — формат исполняемых файлов, объектного кода и динамических библиотек, используемый в 32- и 64-битных версиях операционной системы Microsoft Windows. Формат PE представляет собой структуру данных, содержащую всю информацию, необходимую PE загрузчику для проецирования (отображения) файла в память. Исполняемый код включает в себя ссылки для связывания динамически загружаемых библиотек, таблицы экспорта и импорта API функций, данные для управления ресурсами и данные локальной памяти потока (TLS). В операционных системах семейства Windows NT формат PE используется для EXE, DLL, SYS (драйверов устройств), и других типов исполняемых файлов.

Файлы PE не содержат позиционно-независимого кода. Вместо этого они скомпилированы для предпочтительного базового адреса, и все адреса, генерируемые компилятором/компоновщиком, заранее фиксированы. Если PE файл не может быть загружен по своему предпочтительному адресу (потому что он уже занят чем-то ещё), операционная система будет перебазировать его. Это включает в себя перевычисление каждого абсолютного адреса и изменение кода для того, чтобы использовать новые значения. Загрузчик делает это, сравнивая предпочтительный и фактический адреса загрузки, и вычисляя значение разности. Тогда для получения нового

[1] Исследование выполнено при финансовой поддержке Минобрнауки России (грант ИБ) в рамках научного проекта № 5/2020.

Рис. 1. Взаимосвязь между устойчивостью стеганосистемы и объемом скрываемого сообщения при неизменном размере файла-контейнера

адреса ячейки памяти эта разность складывается с предпочтительным адресом. Базовые адреса перемещений хранятся в списке и при необходимости добавляются к существующей ячейке памяти. Полученный код является теперь отдельным по отношению к процессу и не является больше разделяемым, так что при таком способе теряются многие из преимуществ экономии памяти динамически загружаемых библиотек. Такой способ также значительно замедляет загрузку модуля. По этой причине следует избегать перебазирования везде, где это возможно; например, библиотеки, поставляемые Microsoft, имеют предварительно вычисленные неперекрывающиеся базовые адреса. В случае отсутствия необходимости перебазировании PE файлы имеют преимущество очень эффективного кода, но при наличии перебазирования издержки в использовании памяти могут быть значительными.

В стеганодетекторе определяется наличие в контейнере (возможно уже измененном) скрытых данных. Это изменение может быть обусловлено влиянием ошибок в канале связи, операций обработки сигнала, намеренных атак нарушителей. Различают стеганодетекторы, предназначенные только для обнаружения факта наличия встроенного сообщения, и устройства, предназначенные для выделения этого сообщения из контейнера — стеганодекодеры.

Итак, анализ типичной структуры стеганосистемы показал, что в такой системе происходит объединение двух типов информации таким образом, чтобы они по-разному воспринимались разными детекторами. В качестве одного из детекторов может выступать система выделения скрытого сообщения или человек.

Алгоритм встраивания сообщения в простейшем случае состоит из двух основных этапов:

1. встраивание в стеганокодере секретного сообщения в контейнер-оригинал.
2. обнаружение в стеганодетекторе скрытого зашифрованного сообщения из контейнера-результата.

Исходя из этого, процесс стеганографического преобразования описывается зависимостями: E: C×M⟶S; D: S⟶M, где S= $\{(c_1, m_1), (c_2, m_2), \ldots, (c_n, m_n), \ldots, (c_q, m_q)\} = \{s_1, s_2, \ldots, s_q\}$ — множество контейнеров-результатов (стеганограмм).

Первая зависимость описывает процесс скрытия информации, вторая — извлечение скрытой информации. Необходимым условием при этом является отсутствие «пересечения». Кроме того, необходимо, чтобы мощность множества [C] ≥[M]. При этом оба адресата (отправитель и получатель) должны знать алгоритм прямого (E) и обратного (D) стеганографического преобразования.

Итак, в общем случае стеганосистема — это совокупность ∑ = (C, M, S, E, D) контейнеров (оригиналов и результатов), сообщений и преобразований, которые их связывают.

Для большинства современных методов, которые используются для скрытия сообщений в файлах цифрового формата, имеет место зависимость надежности системы от объема встраиваемых данных, представленная на рисунке 1.

Из рисунка 1 видно, что увеличение объема встраиваемых данных значительно снижает надежность системы.

Таким образом, существует перспектива принятия оптимального решения при выборе между количеством

скрываемых данных и степенью устойчивости (скрытости) к возможной модификации (анализу) сигнала-контейнера. Путем ограничения степени ухудшения качеств контейнера, которые способен воспринимать человек, при стеганографической обработке контейнера можно достичь или высокого уровня (объема) встраиваемых данных, или высокой устойчивости к модификации (анализу), но никоим образом не обоих этих показателей одновременно, поскольку рост одного из них неизбежно приводит к уменьшению другого. Несмотря на то, что данное утверждение математически может быть продемонстрировано только для некоторых методов стеганографии (например, для скрытия путем расширения спектра), очевидно, что оно является справедливым и для других методов скрытия данных.

Реализация метода

Рассмотрим перечисленные некоторые перспективные варианты.

Класс очень простой, поэтому будет легко увидеть связь между исходным кодом и сгенерированным байт-кодом. Первым делом мы видим, что в байт-код-версии класса компилятор вызывает конструктор по умолчанию (как и написано в спецификациях JVM). Далее, изучая байт-кодовые инструкции (у нас это aload_0 и aload_1), мы видим, что некоторые из них имеют префиксы типа aload_0 и istore_2. Это относится к типу данных, с которыми оперирует инструкция. Префикс «a» обозначает, что опкод управляет ссылкой на объект. «i», соответственно, управляет integer. Интересный момент здесь заключается в том, что некоторые из инструкций оперируют странными операндами типа #1 и #2, что на самом деле относится к пулу констант класса.

Размер массива локальных переменных определяется во время компиляции в зависимости от количества и размера локальных переменных и параметров метода. Стек операндов — LIFO-стек для записи и удаления значений в стеке; размер также определяется во время компиляции. Некоторые опкоды добавляют значения в стек, другие берут из стека операнды, изменяют их состояние и возвращают в стек. Стек операндов также используется для получения значений, возвращаемых методом (return values).

Применение данных методик вложения в байт-код относятся к недокументированным возможностям Java машины, которые могут варьироваться в зависимости от версии и нуждаются в экспериментальной проверке.

Для выявления стеганографических каналов в сети использовался статистический метод. К сожалению, данный метод сложно реализуем в реальном времени, поэтому сравнение производилось постфактум, однако данное выявление стеганографических каналов позволяет обнаружить инсайдера. В [6] уже разбирались поля пригодные для построения стеганографических каналов, будем считать, что инсайдер использует поля изменение которых не влияет на прохождение пакета по сети кардинальным образом. Такими полями являются:

- DSCP (Differentiated Services Code Point) — используется для разделения трафика на классы обслуживания, отвечает за приоритет обработки пакетов. Ранее являлось частью поля ToS, в настоящее время определено в RFC2474 как «Differentiated Services». Размер данного поля 6 бит.
- ECN (Explicit Congestion Notification) — используется для предупреждения о перегрузке сети без потерь пакетов. Является необязательной функцией, используется только в том случае, если оба хоста её поддерживают. Ранее являлось частью поля To S. Размер данного поля составляет 2 бита.
- Identification — имеет размер 16 бит и используется в основном для идентификации фрагментов пакета. Изначальное значение генерируется псевдослучайным образом, а у фрагментированных пакетов значение в этом поле должно быть одинаковым.
- Source Address — используется для записи в него IP адреса источника. Длинна поля 32 бита. Поле Source Address может быть заменено если используется преобразование сетевых адресов (NAT, PAT). Данное поле может быть использовано для сокрытия информации, однако только в том случае если на пути нет NAT, PAT и нам не требуется отсылать обратно какие-либо данные.

Выявление стеганографических каналов было решено проводить на основе вероятности появления определенных значений в полях DSCP, ECN, Identification, Source Address. При постоянном или частом вложении данных в поля заголовков меняется вероятность появления определенного значения поля заголовка пакета, проще говоря изменяется закон распределения значения. Стоит отметить, что при передаче текстовой информации было замечено, что закон распределения изменяется и начинает походить на равномерное распределение, поэтому было решено использовать формулу меры информационной энтропии по Шенону (формула 1). Следовательно, информационная энтропия какой-либо системы — это сумма с противоположным знаком всех относительных частот появления состояния (в нашем случае значение поля заголовков пакетов) с номером i умноженного на их двоичные логарифмы.

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i \qquad (1)$$

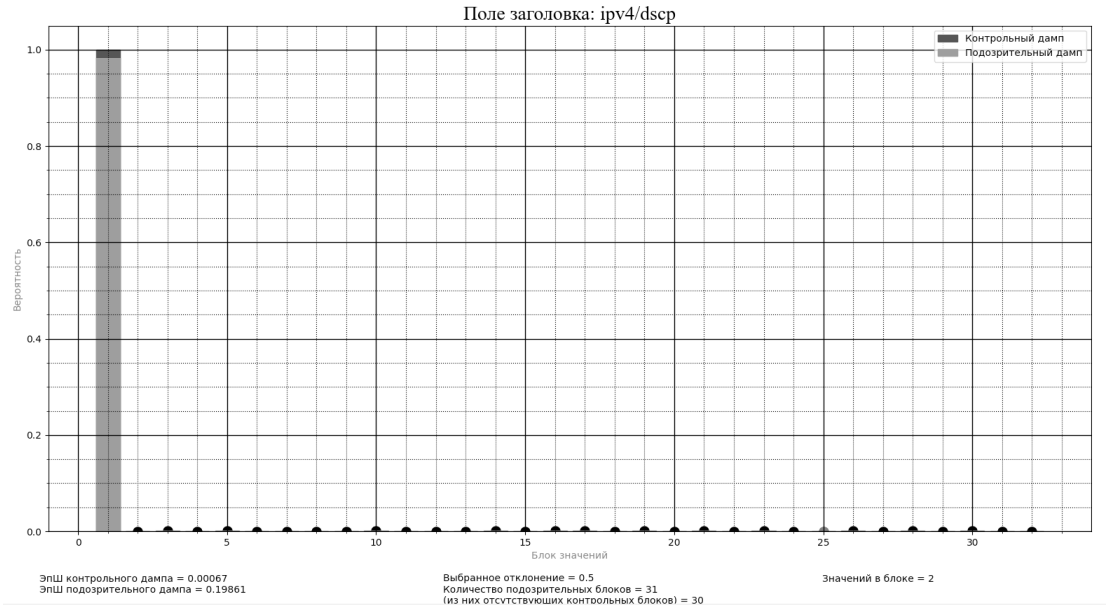

Рис. 2. Сравнение полей Differentiated Services Code Point в дампах контрольного и подозрительного трафика

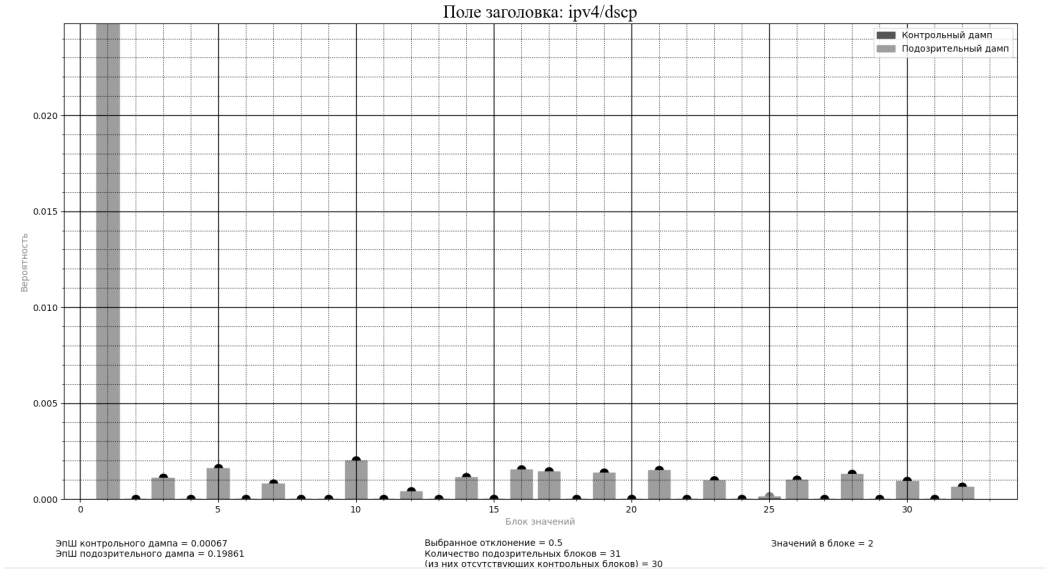

Рис. 3. Сравнение полей Differentiated Services Code Point в дампах контрольного и подозрительного трафика

Где $H(x)$ — мера информационной энтропии по Шеннону, n — количество значений поля, p_i — вероятность появление i-го значения.

Суть метода заключается в том, что изначально в компании записывается один или несколько контрольных дампов, вычисляется мера информационной энтропии всех значений определенного поля заголовка протокола IPv4, после этого происходит снятие дампа в сети, вычисление его меры информационной энтропии по Шеннону. В конце происходит сравнение двух значений. Как правило, при передаче текстовой информации мера информационной энтропии по Шеннону уменьшается.

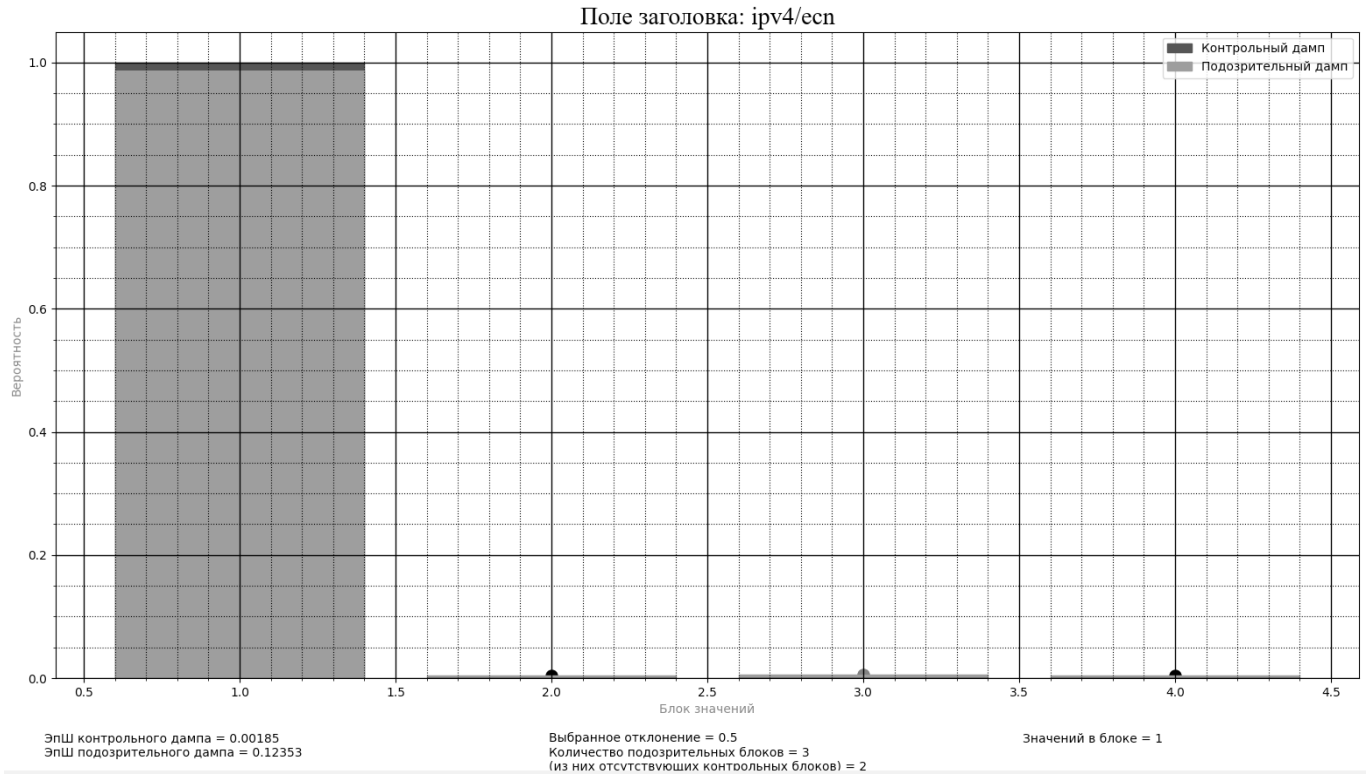

Рис. 4. Сравнение полей Explicit Congestion Notification в дампах контрольного и подозрительного трафика

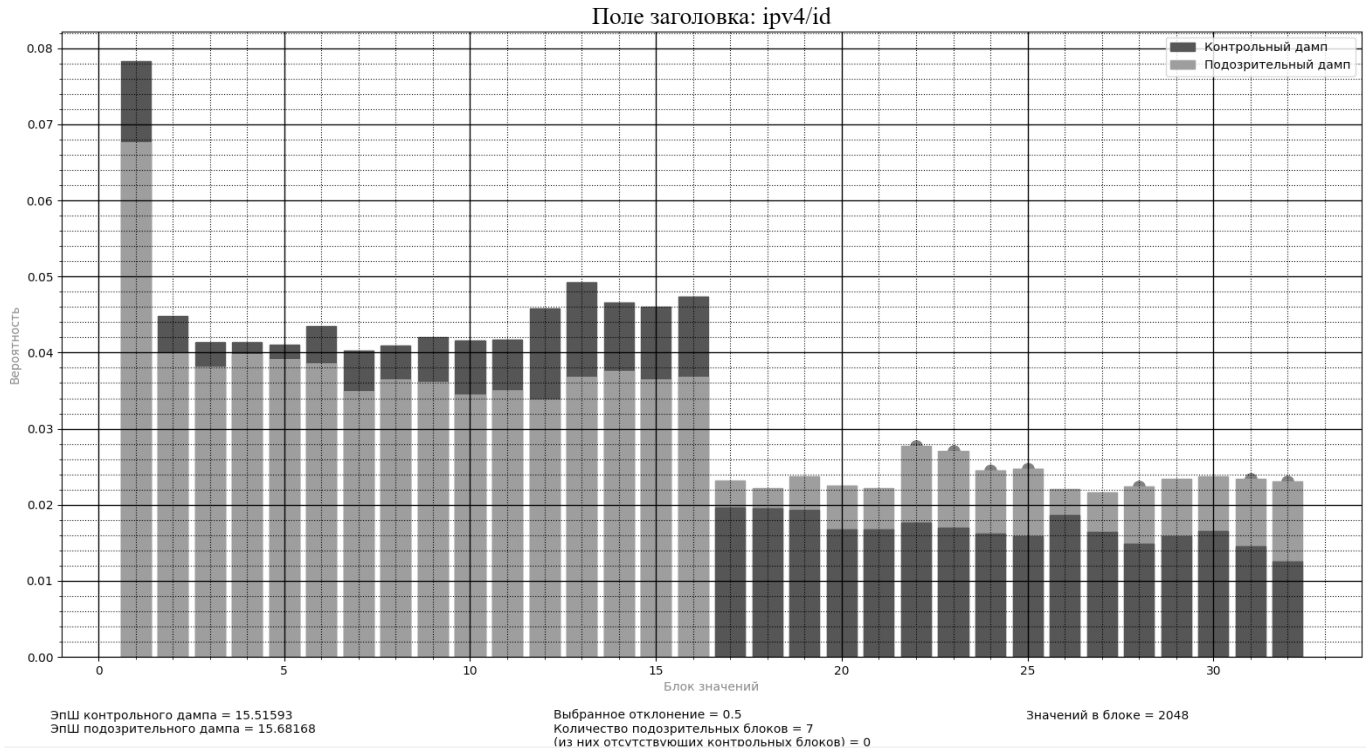

Рис. 5. Сравнение полей Identification в дампах контрольного и подозрительного трафика

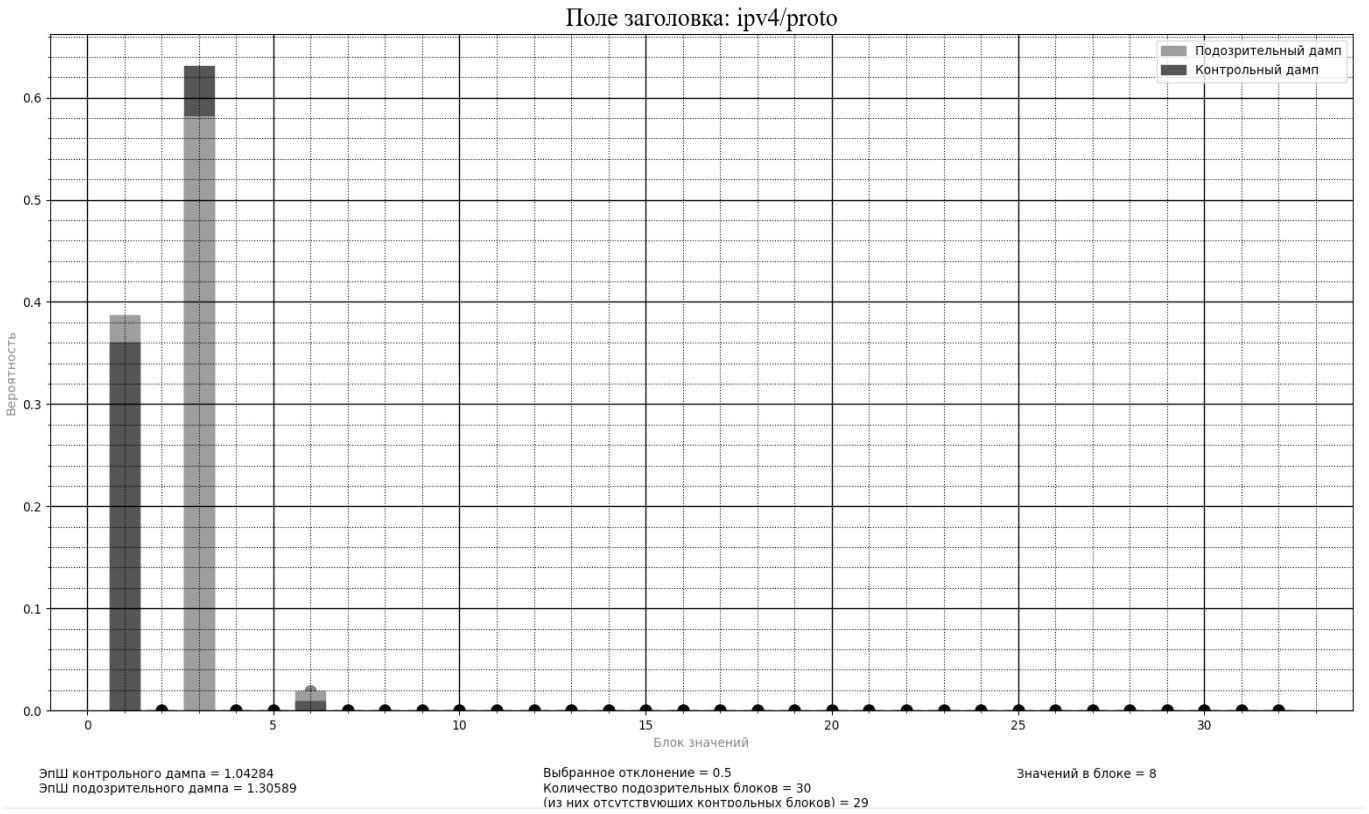

Рис. 6. Сравнение полей Protocol в дампах контрольного и подозрительного трафика

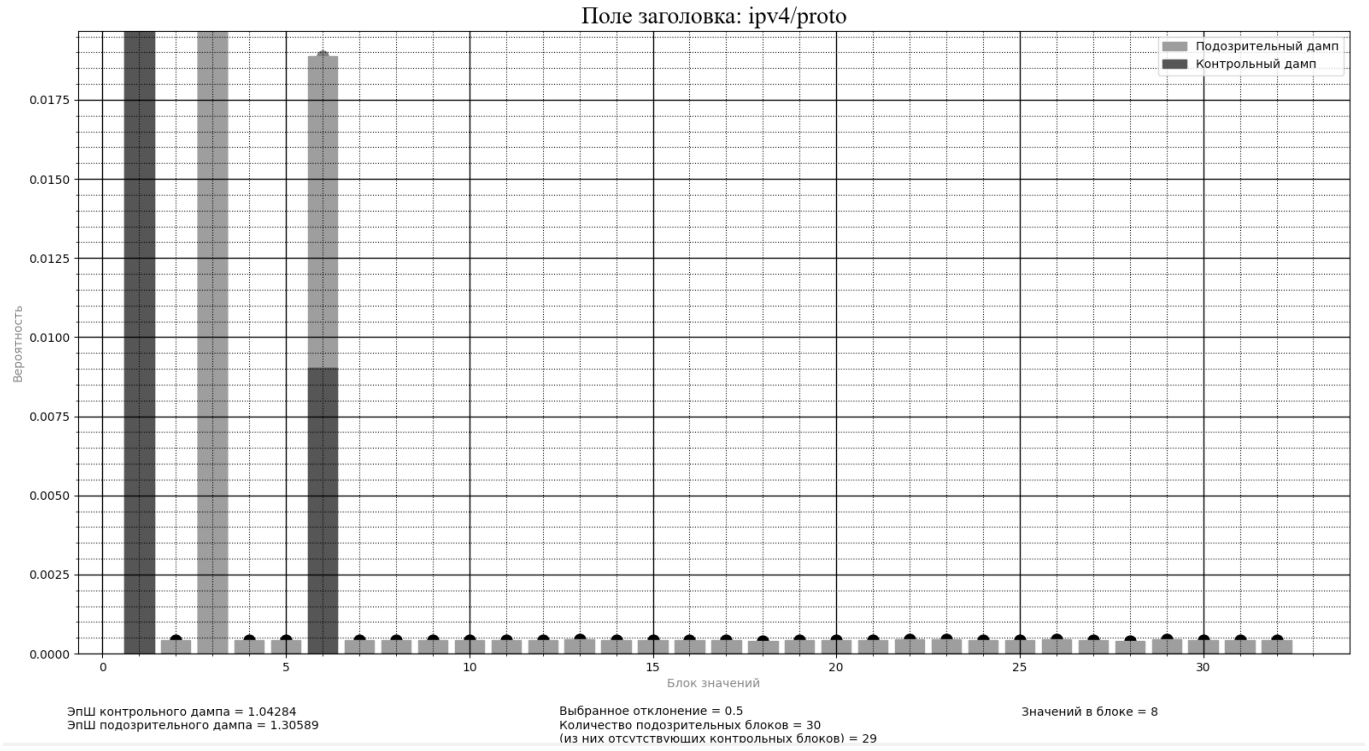

Рис. 7. Сравнение полей Protocol в дампах контрольного и подозрительного трафика

Метод информационной энтропии не всегда помогает явно определить присутствует ли в сети стеганографический канал, поэтому его стоит использовать в совокупности с другими методами. Таким методом может являться вычисление отклонений вероятности. За основу был взят критерий Колмогорова-Смирнова, который предназначен для проверки простых гипотез о принадлежности анализируемой выборки некоторому известному закону распределения.

Пусть X_n — выборка независимых одинаково распределенных случайных величин, $F_n(x)$ — эмпирическая функция распределения, $F(x)$ — некоторая «истинная» функция распределения с известными параметрами. Статистика критерия определяется выражением (формула 2):

$$D_n = sup|F_n(x) - F(x)| \quad (2)$$

Пакеты в сети чаще всего используют одни и те же значения полей в определенной сети. Отсюда формируется определенная модель поведения сети. Сравнение отклонения вероятностей с заранее выбранным порогом поможет определить наличие стеганографического канала связи. Сравнение отклонений происходи по формуле 3.

$$Z \geq \frac{|p_{i_c} - p_{i_p}|}{p_{i_p}} \quad (3)$$

Где Z — заданный порог отношения, p_{i_c} — вероятность появления i значения поля контрольного дампа, а p_{i_p} —вероятность появления i значения поля подозрительного дампа

Для выявления стеганографических каналов в сети на языке Python 3.9.4 была написана программа. Основной задачей программы является сравнение двух дампов трафика и преобразование большого количества значений в человекочитаемый вид и вывод данных на экран в виде графиков, а также сохранение графиков и данных в числовом виде в файл для дальнейшего анализа. Под дампом сетевого трафика понимаются записанные заголовки пакетов.

Стоит отметить, что стеганографический канал работал не на полную мощность (это значит, что при отправке пакетов делались небольшие перерывы, а также максимальная пропускная способность канала и не была загружена более чем на 50%), чтобы усложнить его выявление стандартными средствами на основе общей статистики трафика.

Изначально на графике поля Differentiated Services Code Point не видно каких-либо подозрительных отклонений (рисунок 2), информационная энтропия по Шеннону больше, чем у контрольного дампа, что свидетельствует об увеличении разнообразности значений в данном поле или о том, что распределение носит более случайных характер. После приближения графика (рисунок 3) отчетливо наблюдаем, что встречаются значения, которые не использовались ранее, а также то, что они распределены относительно равномерно. Что свидетельствует о появлении стеганографического канала связи.

Также стоит обратить внимание, что крайне сильно возрос двадцать пятый блок, у контрольного вероятность появления значений из данного блока составляла 0,0000420912557131085, а у подозрительного составляет 0,000129236768998063. На основании вышеперечисленных данных можно сделать вывод, что в данном поле присутствует стеганографический канал.

Аналогичную ситуацию мы наблюдаем и для поля Explicit Congestion Notification. Как и в поле Differentiated Services Code Point появляются ранее не встречающееся значения (рисунок 4), а вероятность появления третьего блока подозрительного дампа гораздо выше, чем контрольного и составляет 0,00521372142962748, в отличие от 0,000129079850853533 контрольного. Следовательно, мы можем утверждать о присутствии стеганографического канала в данном дампе.

Рассмотрим поле Identification, на графике (рисунок 5) сразу наблюдаем равномерный рост вероятности блоков подозрительного дампа в отличии от контрольного, разница вероятностей у семи блоков превышает заданный порог, что свидетельствует о наличии стеганографического канала в данном трафике.

Поле Protocol изначально было оценено как непригодное ввиду генерации таких протоколов, которые не могут использоваться в сети. Например, протокол OSPF (Open Shortest Path First) имеет значение 89, а IGRP (Interior Gateway Routing Protocol) — значение 88. Крайне подозрительным является появление данных протоколов в локальной сети. На графике (рисунок 6) большая вероятность у тех значений, которые постоянно встречаются в сети, например ICMP, UDP, TCP и другие. Однако при приближении (рисунок 7) мы видим, что используются и другие протоколы. Их распределение похоже на равномерное, стоит учесть тот факт, что в стеганографическом канале объем данных был небольшим поскольку передавалась текстовая информация, поэтому вероятность в некоторых блока небольшая, но не нулевая, как в контрольном дампе, что было видно из сохраненных данных.

Заключение

Данный метод позволяет достаточно эффективно выявлять вложения в заголовки протокола Ipv4, одна-

ко, одним из основных минусов рассмотренного метода является выявление канала после его построения и передачи части данных. Еще одним минусом является необходимость в работе оператора для своевременного выявления стеганографического канала. Но стоит отметить, что даже такой метод может помочь своевременно отследить и прервать передачу данных поскольку пропускная способность полей заголовка протокола IPv4 относительно мала.

Заметить скрытое вложение, которое было внедрено в файл, будет сложно из-за того, что изменения внутри файла не затронут ни размера, ни функциональности исполнимого кода. Предлагаемый метод прост в реализации и малозатратный. Скрытое вложение, реализуемое способами эквивалентных замен операторов, позволяет реализовать защиту авторских прав на техническом уровне при использовании цифровой подписи.

ЛИТЕРАТУРА

1. Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных (выписка). ФСТЭК России, 2008 год
2. ГОСТ Р 53113.1–2008 «Защита информационных технологий и автоматизированных систем от угроз безопасности, реализуемых с использованием скрытых каналов».
3. Штеренберг, С.И. Разработка методики построения доверенной среды на основе скрытого программного агента. Часть 1. исследование / С.И. Штеренберг, А.В. Красов // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки.— 2021.— № 2.— С. 14–20.— DOI 10.46418/2079–8199_2021_2_2.
4. Свидетельство о государственной регистрации программы для ЭВМ № 2020617876 Российская Федерация. Модель угроз и нарушителя: № 2020616749: заявл. 29.06.2020: опубл. 15.07.2020 / А.В. Красов, А.А. Миняев, А.И. Пешков; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ).
5. Израилов К.Е., Татарникова И.М., Подход к анализу безопасности программного кода с позиции его формы и содержания // В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). Сборник научных статей VIII Международной научно-технической и научно-методической конференцияи: в 4 т. 2019. С. 462–467.
6. Коржик В.И., Небаева К.А., Герлинг Е.Ю., Догиль П.С., Федянин И.А. Цифровая стеганография и цифровые водяные знаки. Часть 1. Цифровая стегагография. СПб., СПбГУТ, 2016, 226 с.
7. Израилов К.Е., Методика оценки эффективности средств алгоритмизации, используемых для поиска уязвимостей // Информатизация и связь. 2014. № 3. С. 44–47.
8. I. Kotenko, K. Izrailov, A. Krasov, I. Ushakov, An approach for stego-insider detection based on a hybrid nosql database // Journal of Sensor and Actuator Networks.— 2021.— Vol. 10.— No 2.— DOI 10.3390/jsan10020025.
9. P.I. Sharikov, A.V. Krasov, A.M. Gelfand, N.A. Kosov, Research of the Possibility of Hidden Embedding of a Digital Watermark Using Practical Methods of Channel Steganography // Intelligent Distributed Computing XIII, St.-Petersburg, 07–09 октября 2019 года.— St.-Petersburg: Springer Verlag, 2019.— P. 203–209.— DOI 10.1007/978–3–030–32258–8_24.
10. Ушаков И.А., Обнаружение инсайдеров в корпоративной компьютерной сети на основе технологий анализа Больших данных // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2019. № 4. С. 38–43.
11. Г.А. Орлов, А.В. Красов, А.М. Гельфанд, Применение Big Data при анализе больших данных в компьютерных сетях // Наукоемкие технологии в космических исследованиях Земли.— 2020.— Т. 12.— № 4.— С. 76–84.— DOI 10.36724/2409–5419–2020–12–4–76–84.
12. А.С. Салита, А.В. Красов, Создание стеганографического канала при помощи полей // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки.— 2021.— № 2.— С. 36–40.— DOI 10.46418/2079–8199_2021_2_6.
13. Simmons, G.J.: The prisoners' problem and the subliminal channel. In Chaum, D., ed.: Crypto '83. Advances in Cryptography, Plenum Press (1983) 51–67

Журнал «Современная наука: актуальные проблемы теории и практики»