УДК 004.056 Вестник УрФО № 1(35) / 2020, с. 48–57

**Домуховский Н. А., Синадский А. Н.**

DOI: 10.14529/secur200105

# *ИТЕРАТИВНЫЙ СТАТИСТИКО-ЭНТРОПИЙНЫЙ МЕТОД И АЛГОРИТМ АНАЛИЗА СЕТЕВОГО ТРАФИКА ПРИ ОТСУТСТВИИ АПРИОРНЫХ СВЕДЕНИЙ О ЕГО СТРУКТУРЕ*

*Статья посвящена анализу трафика при отсутствии априорных сведений о его структуре с целью выявления уязвимостей и проведения аудита. В результате объединения существующих энтропийного и статистического алгоритмов разработан статистико-энтропийный метод выделения сетевых узлов и значимых полей из трафика неизвестных протоколов. Энтропийный алгоритм, анализируя массив трафика, на основе энтропии отдельных байт и взаимной информации пар байт принимает решение о границах значимых полей. Статистический алгоритм для определения сетевых адресов использует оценку количества вхождений похожих на части сетевого пакета подстрок в ранее полученный массив сетевого трафика. На основе энтропийного алгоритма разработан итеративный алгоритм, решающий задачу анализа трафика, имеющего в своём составе более одного протокола. Математические модели каждого из алгоритмов реализованы программно, результатом работы программной реализации описанного статистико-энтропийного метода из сетевого трафика без априорных сведений об используемых в нём протоколах выделяются сетевые адреса и предлагается разделение на семантические поля.*

***Ключевые слова:*** *анализ сетевого трафика, реверс-инжиниринг, статистика.*

**Domukhovsky N.A., Sinadsky A.N.**

# *ITERATIVE STATISTICAL-ENTROPY METHOD FOR ZERO KNOWLEDGE NETWORK TRAFFIC ANALYSIS ALGORITHM IMPLEMENTATION*

*The article is devoted to traffic analysis with zero knowledge about its structure. As a result of combining existing entropy and statistical algorithms, a statistical-entropy method has been developed capable of distinguishing network nodes and significant fields from traffic with un-*

*known protocol. The decision about significant fields boundaries in the analyzed traffic sample made by the algorithm is based on the entropy of individual bytes and byte pairs mutual information. The statistical algorithm determines network addresses using estimate number of occurrences parts of a network packet similar (as a strings) to parts of a previously received array of network traffic. Based on the entropy algorithm, an iterative algorithm has been developed that solves the problem of traffic analysis, which includes more than one protocol. The mathematical models each of the algorithms are implemented as a module of the program that implements the statistical-entropy method. As a result of the software implementation of the described statistical-entropy method, network addresses are allocated from the network traffic with zero knowledge about the protocols used in it, and separation into semantic fields is proposed.*

***Keywords:*** *network traffic analysis, reverse engineering, statistics.*

В Требованиях по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации, утвержденных приказом ФСТЭК России от 25 декабря 2017 г. №239, в качестве меры АУД.5 указан «Контроль и анализ сетевого трафика». При проведении мониторинга в условиях проприетарных протоколов, распространённых как в сетях АСУ ТП, так и IoT, средство анализа сетевого трафика не может дать достаточно информации для обеспечения мер по защите сетевых ресурсов.

Задача состоит в выделении сетевых адресов и границ полей заголовков протоколов.

Разделение входного массива сетевого трафика на отдельные поля и идентификация сетевых адресов при отсутствии априорных знаний о протоколах является актуальной задачей. При этом предполагаются следующие предположения-эвристики:

– в каждом сетевом пакете присутствует адресная и семантическая части данных;

– адресная часть всегда расположена ближе к началу пакета, чем семантическая;

– адресная часть всегда содержит адреса отправителя и получателя;

– адресная часть меняется реже, чем семантическая.

Известные решения [1-7] предлагают варианты решения частных проблем (унифицированное описание сети, выделение полей из неизвестного трафика одного протокола, классификация трафика на протоколы), но не дают возможности выполнять все действия одновременно.

В [8] представлен способ использования информационной энтропии в качестве метода определения границ полей, позволяющий, используя сравнительно небольшие вычислительные ресурсы, по графикам изменения энтропии отдельных байтов и их взаимной информации делать предположения о структуре анализируемого сетевого протокола. Недостатком такого метода является невозможность его использования на массиве трафика, имеющем более одного протокола.

Предложенный статистико-энтропийный метод, применяет энтропийный модуль для определения границ полей протокола с помощью информационной энтропии и статистический модуль для выделения сетевых адресов на основе анализа статистики вхождения частей пакета в массив трафика.

## Статистико-энтропийный метод и его реализация

Для решения проблемы одновременного выделения сетевых адресов и границ семантических полей предлагается объединить два известных алгоритма – статистический и энтропийный (рис. 1). Статистический алгоритм использует оценку количества вхождений похожих на части сетевого пакета подстрок в ранее полученный массив сетевого трафика для выделения из сетевого трафика уровней адресации и конкретных адресов сетевых узлов, а энтропийный с помощью вычисления информационных характеристик осуществляет поддержку решения статистического и определяет границы полей в семантической части.

Входные данные для статистико-энтропийного алгоритма – набор из *lp* сетевых пакетов. Каждый сетевой пакет имеет номер *n* и содержит $lb_n$ байт *d*. Пакет – набор байт $D = (d_i)_{i=1}^{lb_n}$, $d_i$ – байт пакета, расположенный по смещению i от его начала, n – порядковый номер пакета. Набор сетевых пакетов определяется как $DS = (D_i)_{i=1}^{lp}$.

Выходные данные алгоритма – полученный из энтропийного алгоритма набор полей $F = (f_i)_{i=1}^{lf}$, где *lf* – количество выделенных полей, и сформированные из статистического алгоритма множества адресных $AS = (A_i)_{i=1}^{lp}$

и семантических $SS = (S_i)_i^{lp}$ частей сетевых пакетов и сетевых адресов $AddrS = (addr_i)_{i=1}^{lp}$.


Рис. 1. Статистико-энтропийный алгоритм

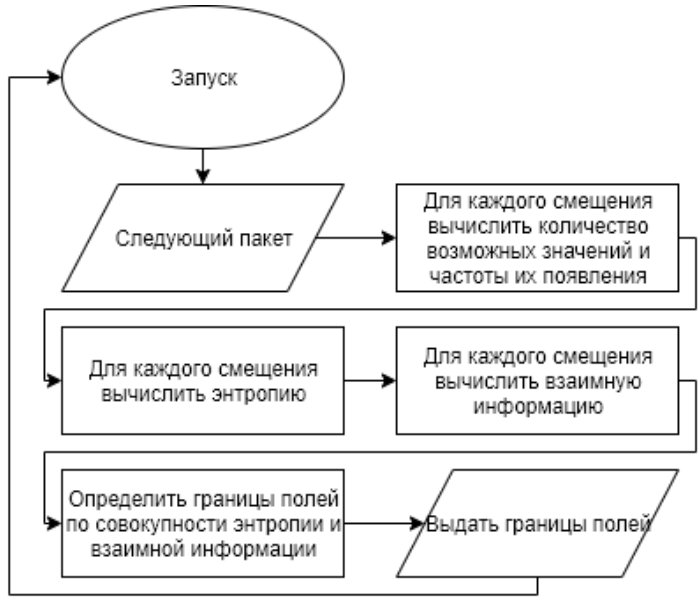


Рис. 2. Энтропийный алгоритм

На основе предложенного в статье [8] метода был разработан новый алгоритм (рис. 2) и предложена программная реализация.

В отличие от алгоритма [8], основанного на раздельном принятии решений, реализовано совместное использование энтропии и взаимной информации для принятия решения о границах полей вместо предложенного раздельного принятия решений.

Для каждого значения *i* оценивается количество вхождений $vh_{i,j}$ каждого значения байта $d_i$ и количество вхождений $vmi_{i,k}$ пар значений. Всего вариантов одиночных значений может быть $2^8 = 256$, поэтому $j \in \overline{(1,256)}$, а пар значений – $(2^8)^2 = 65536$, поэтому $k \in \overline{(1,65536)}$. Красным прямоугольником (рис. 3) показан выбор отдельных значений, зеленым – пары значений с использованием метода скользящего окна.

По формуле $Ph_{ij} = \frac{vh_{i,j}}{lp}$, где i – смещение байта от начала пакета, а j – значение $j \in \overline{(1,256)}$, определяется вероятность появления определённого значения в этой позиции для каждого байта пакета, то есть $i \in \overline{(1,lb)}$, $j \in \overline{(1,256)}$. Аналогично вычисляется $Pmi_{i,j} = \frac{vmi_{i,j}}{lp}$, $i \in \overline{(1,lb_i)}$, $j \in \overline{(1,65536)}$.

По формуле Шеннона рассчитывается 256-чная энтропия для каждого смещения $H_j = -\sum_{i=1}^{lp} Ph_{i,j} \log_{256} Ph_{i,j}$, где $j$ определена

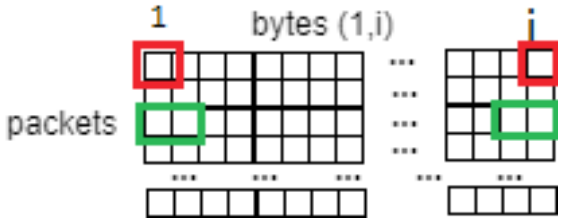

Рис. 3. Результат работы энтропийного алгоритма

так же, как в расчёте вероятности выше. В результате вычислений формируется матрица-строка *H* длиной *max (lb)*.

После расчёта энтропии для каждого смещения от минимального до максимального $j \in \overline{(1, \max(lb)-1)}$ рассчитывается взаимная информация $MI_j = -\sum_{i=1}^{lp} Pmi_{i,j} \log_{65536} \frac{Pmi_{i,j}}{Ph_{i,j} * Ph_{i,j+1}}$.

После того, как энтропия каждого байта и их взаимная информация рассчитаны, можно приступать к определению границ пакета. Создаются матрицы-строки *Hr, Mlr* и *Res* длиной $\max(lb)$, причём $Hr_1$='start', *Mlr1*='start', *Res1*='start'.

Для $j \in \overline{(2, \max(lb))}$ $Hr_j$ принимает значение *'start'*, если $Hr_{j-1}$='end', иначе *'field'*, если $H_{j-1}<H_j$, иначе *'end'*. Для $j \in \overline{(2, \max(lb)-1)}$ $Mlr_j$ принимает значение *'start'*, если $Mlr_{j-1}$='end', иначе *'field'*, если $Ml_{j-1}>Ml_j$, иначе *'end'*.

После определения границ полей пакета для энтропии и взаимной информации в отдельности для принятия совместного решения о расположении границ во входных данных задаётся порог расходимости *T*, который обозначает возможное отличие принятия решения по энтропии и по взаимной информации.

Для каждого $j \in \overline{(2, \max(lb))}$, если $Hr_j=Mlr_j$, то $Res_j=Hr_j$. Иначе если $\exists Hr_{k_1} = 'start'$, $\exists MIr_{k_2} = 'start'$, $|k_1 - j| < T$ и $|k_2 - j| < T$, то $Res_j = Hr_{k_1}$, $Hr_{k_1} = 'field'$, $Hr_j = 'start'$, $MIr_{k_2} = 'field'$, $MIr_j = 'start'$. Для значения $'end'$ аналогично.

Очищается набор полей $F = \emptyset$. Для каждого $j \in \overline{(2, \max(lb))}$, $Res_j = 'start'$, ищется минимальное *k*, такое что *k>j*, и ко множеству *F* добавляется элемент $f_i = \{j, k\}$, имеющий смысл координат начала и конца поля.

При получении на вход нового пакета $D_{n+1}$ для каждого значения *i* обновляется количество вхождений каждого возможного значения байта *di*. Затем по описанному выше алгоритму определяются границы полей, и выполняется переход к ожиданию следующего пакета.

В результате *F* содержит набор границ полей. Их можно использовать для разбора (и ускорения разбора) протокола после того, как будет выделена адресная часть.

Цель статистического алгоритма – выделить из входного потока сетевого трафика адресную и семантическую части, из адресной части – адреса отправителя и получателя.

Перед началом работы алгоритма задаются константы и начальные значения. В качестве констант задаются отношение частот вхождения адресной части к семантической *ADDR_TO_DATA_MULTIPLIER = 5*, значение относительного расстояния Хэмминга, при котором строки считаются похожими *HAMMING_MEASURE_OF_SIMILARITY = 0.1*, в качестве начального значения – средняя длина адресной части *average_border = 1*.

Обработка происходит итеративно по пакетам. Для *i* пакета решения принимаются на основе *i-1* пакетов, обработанных до него.

В целом алгоритм (рис. 4) выглядит так: для каждого пакета D определяются адресная $addr\_part = (d_i)_{i=1}^{la}$ и семантическая $data\_part = (d_i)_{i=la}^{lb}$ части, где *la* – длина адресной части, а *lb* – длина сетевого пакета. В соответствии с полученным значением *la* обновляется средняя длина адресной части *average_border*. Адресная часть $addr\_part$ снова обрабатывается алгоритмом так, будто это весь сетевой пакет *D*, и в этот раз на выходе алгоритма появляются две адресные части верхнего *max_level* и следующего за ним *max_level-1* уровня. Адресная часть верхнего уровня разделяется на две части, каждая из которых добавляется в список сетевых адресов *AS*. Адресная часть уровня *max_level-1* снова обрабатывается алгоритмом для выделения следующих уровней адресации и конкретных сетевых адресов. Так продолжается до тех пор, пока не будет разобран на части сетевой адрес самого низкого уровня.

Полученные адреса позже могут использоваться для построения карты сети.

В общем алгоритме была указана операция выделения адресной и семантической частей, а затем – адресов из адресной части. Для её реализации используется нижеописанный алгоритм (рис. 5).

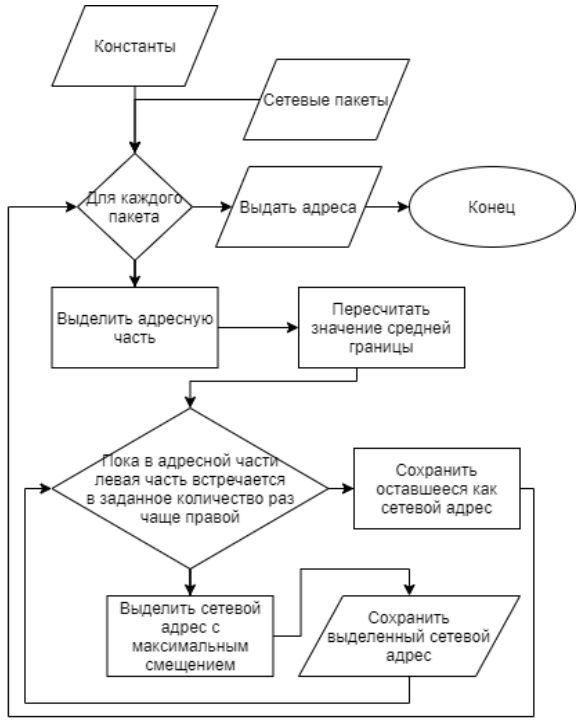


Рис. 4. Общий вид статистического алгоритма

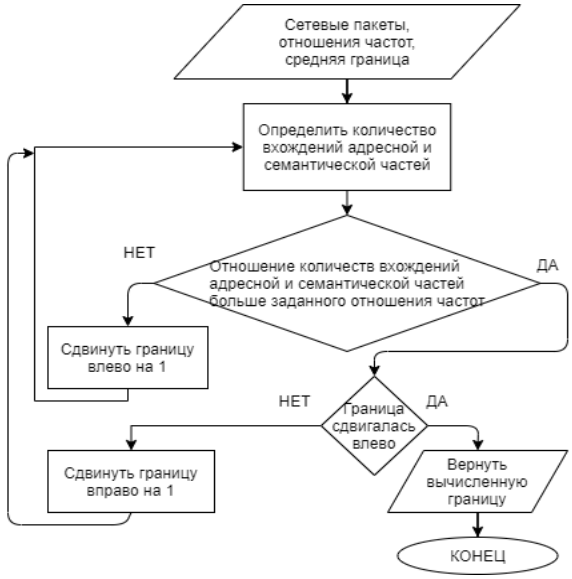


Рис. 5. Определение границы адресной части

Задаётся смещение *border = avarage_ border*, вычисляется количество вхождений $(d_i)_{i=1}^{border}$ и $(d_i)_{i=border}^{lp}$ в обработанный ранее массив трафика. Если количество вхождений первого в *ADDR_TO_DATA_MULTIPLIER* раз больше, чем второго, то *border* увеличивается на 1 и расчёт повторяется. В противном случае действия зависят от того, был ли уже выполнен расчёт для значения *border-1*. Если расчёта ещё не было, то выполняется итеративное уменьшение значения *border* на 1 и перерасчёт количества вхождений $(d_i)_{i=1}^{border}$ и $(d_i)_{i=border}^{lp}$ до тех пор, пока первое не станет встречаться в *ADDR_TO_DATA_MULTIPLIER* раз чаще второго. Если расчёт уже был выполнен, то значение *border* уменьшается на 1.

Если выполнялось выделение адресной и семантической частей из пакета, то по значению смещения border обновляется значение average_border, $(d_i)_{i=1}^{border}$ добавляется в $AS$, а $(d_i)_{i=border}^{lp}$ – в $SS$. В противном случае $(d_i)_{i=border}^{lp}$ сохраняется в $AddrS$, а $(d_i)_{i=1}^{border}$ обрабатывается повторно.

Под «количеством вхождений» в описа-

нии алгоритма выделения частей из трафика понимается количество вхождений похожих подстрок в массив строк, полученный из сетевых пакетов. Из-за особенностей сетевого трафика (поля-разделители, поля-идентификаторы) принято решение искать не абсолютно совпадающие строки, а похожие, причём в качестве алгоритма оценки похожести использовать модифицированный метод Хэмминга.

Модификация состоит в том, что расстояние Хэмминга считается не абсолютное, а относительно длины строки $hamming_{sim} = \frac{dist}{num\_symbols}$, где *dist* – расстояние Хэмминга, *num_symbols* – количество символов в сравниваемых подстроках. Таким образом, вместо термина «расстояние» уместнее использовать «коэффициент схожести».

Для оценки количества вхождений строка *D* по предполагаемой длине адресной части делится на адресную $(d_i)_{i=1}^{border}$ и семантическую $(d_i)_{i=border}^{lp}$ части. Для каждой части оценивается коэффициент схожести *sim_coef* с остальными обработанными пакетами. В случае, если *sim_coef* оказывается больше *HAMMING_MEASURE_OF_SIMILARITY*, то количество оцениваемой подстроки вхождений увеличивается.

Такой метод оценки количества вхождений подстрок вычислительно затратен. Поэтому предлагается альтернативный метод: строить дерево всех возможных комбинаций, и в листьях хранить количество появлений каждой из них.

Сетевой пакет представляет собой набор байт, каждый из которых может принимать 256 значений. Учитывая ограниченность реальных вычислительных ресурсов, предлагается использовать последовательности длиной 4 байта, в этом случае, с одной стороны, 8ГБ оперативной памяти достаточно для работы, и, с другой, последовательности не будут слишком короткими.

В таком случае скорость поиска количества вхождений заданной комбинации будет линейно зависеть от её длины, что даст возможность увеличивать размер окна при анализе больших объёмов трафика.

В результате обработки набора сетевых пакетов двумя разработанными алгоритмами имеются набор полей $F = (f_i)_{i=1}^{lf}$, множества адресных $AS = (A_i)_{i=1}^{lp}$ и семантических $SS = (S_i)_{i=1}^{lp}$ частей сетевых пакетов и дерево сетевых адресов $AddrS = (addr_i)_{i=1}^{lp}$.

Семантические части $SS = (S_i)_{i=1}^{lp}$ пакетов делятся на сегменты по известному набору полей *F*, формируя множество наборов семантических полей $FS = ((fs_i)_{i=1}^{lp})_{j=1}^{lf}$. Затем *FS* и *AddrS* передаются во внешнюю систему для построения графической топологии сети и отображения типов узлов на основе *FS* для каждого узла.

### Анализ трафика, содержащего более одного протокола

Используемый энтропийный алгоритм позволяет быстро обрабатывать трафик, в состав которого входят пакеты, сформированные исключительно по одному протоколу (или одной иерархической группе протоколов). Для обхода этого ограничения предлагается для входного трафика строить на основе длин энтропийных полей дерево протоколов, и на каждом сетевом уровне анализировать их раздельно (рис. 6).

Дерево протоколов представляет собой иерархический граф, корнем которого является родительский протокол (протокол самого нижнего уровня, например, Ethernet), а узлами – протоколы следующих уровней, причём кратчайшее расстояние от корня до узла определяет уровень узла.

Задаётся структура *Node*, содержащая порядковый номер родителя на предыдущем уровне *parent_id*, длину первого поля текущего узла *field_len*, номера пакетов, относящиеся к текущему узлу *num_packets*, суммарную длину первых полей до начала пакета *sum_len*.

Создаётся список уровней *nodes*, каждый из которых является списком узлов (структур *Node*). Таким образом, *nodes[level][id_on_level]* однозначно определят узел, где *level* – порядковый номер уровня в дереве протоколов, а *id_on_level* – порядковый номер ветви на уровне.

В списке структур *nodes* создаётся нулевой уровень с единственным узлом – корневым протоколом. Для него с помощью энтропийного алгоритма выбирается длина первого поля *field_len*, и из трафика выбираются все уникальные значения этого поля. Для каждого уникального значения на следующем уровне списка структур *nodes* создается новый узел.

Исходя из предположения о том, что адресная часть всегда находится ближе к началу пакета, чем часть данных, выбирается глубина анализа (количество байт от начала пакета, которое будет рассматриваться). Анализ продолжается до тех пор, пока не будет достигнут этот предел.

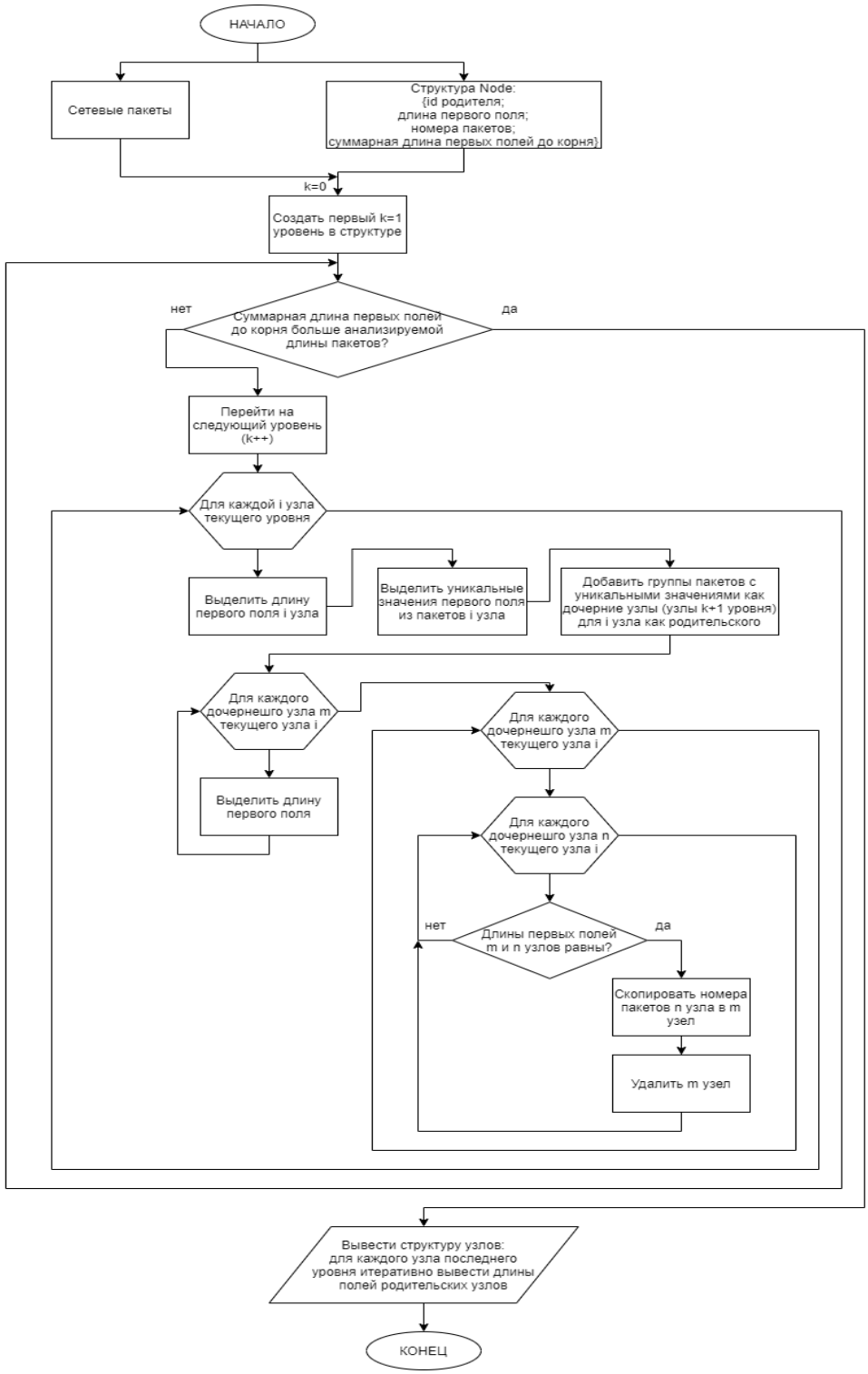


Рис. 6. Итеративный энтропийный алгоритм

Выполняется переход на следующий уровень структуры *nodes,* на котором уже существуют несколько узлов, соответствующих уникальным значениям первого поля родительского узла. Для каждого из узлов с помощью энтропийного алгоритма выбирается длина первого поля, из принадлежащих узлу пакетов выбираются все уникальные значения этого первого поля, на следующем уровне для каждого из них создаются узлы, значение *parent_id* которых соответствует порядковому номеру родителя (текущего узла), а в

список пакетов *num_packets* записываются номера таких пакетов родительского узла, первые несколько байт которых соответствуют выбранному уникальному значению. Затем для каждого из созданных узлов следующего уровня с помощью энтропийного алгоритма выбирается длина первого поля, и те узлы, у которых его значения совпали, объединяются в один узел.

После достижения заданного количества байт от начала пакета анализ завершается, и формируется дерево протоколов. При этом узлы, расположенные ближе к корневому, описывают более низкоуровневые протоколы, чем узлы, расположенные дальше от корневого.

**Результаты работы алгоритмов**

Программная реализация статистического метода позволяет получить массивы длин адресных частей и предполагаемые адреса. При наложении полученных данных на реальный сетевой трафик видно (рис. 7), что верно определяется длина заголовка Ethernet и входящие в него MAC-адреса (выделено зелёным), но при этом к адресу источника ошибочно добавляются служебное редко меняющееся поле type (выделено красным).

```
>>> print(src_addr[7], dst_addr[7])
  b'\xb4\xb5/t\x08\x9d' b'\xac\xc1\xd6b\xc2\x08\x00'
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000003E0   9B A5 AD 50 11 7E 12 B1 EE 00 00 8A 18 7A 5D 18
000003F0   A6 00 00 3C 00 00 00 3C 00 00 00 B4 B5 2F 74 08
00000400   9D F4 AC C1 D6 62 C2 08 00 45 00 00 28 04 03 40
00000410   00 7E 06 46 F9 C0 A8 0C 75 C0 A8 24 0E 32 C8 24
00000420   47 31 9B A5 AD 31 BF E0 69 50 10 80 00 3D 7F 00
00000430   00 00 00 00 00 00 00 8A 18 7A 5D 50 A7 00 00 3C
```

Рис. 7. Результат работы скрипта (статистический алгоритм)

В результате обработки сетевого трафика с помощью программы, реализующей энтропийный метод, построены графики энтропии и взаимной информации.

Анализируя график энтропии (рис. 8) по модели, описанной выше, можно видеть, что энтропия возрастает (или хотя бы не убывает) на длине всего поля и уменьшается в его конце. По этому графику можно с некоторой вероятностью выделить MAC-адреса, подтверждая полученные статистическим алгоритмом результаты.

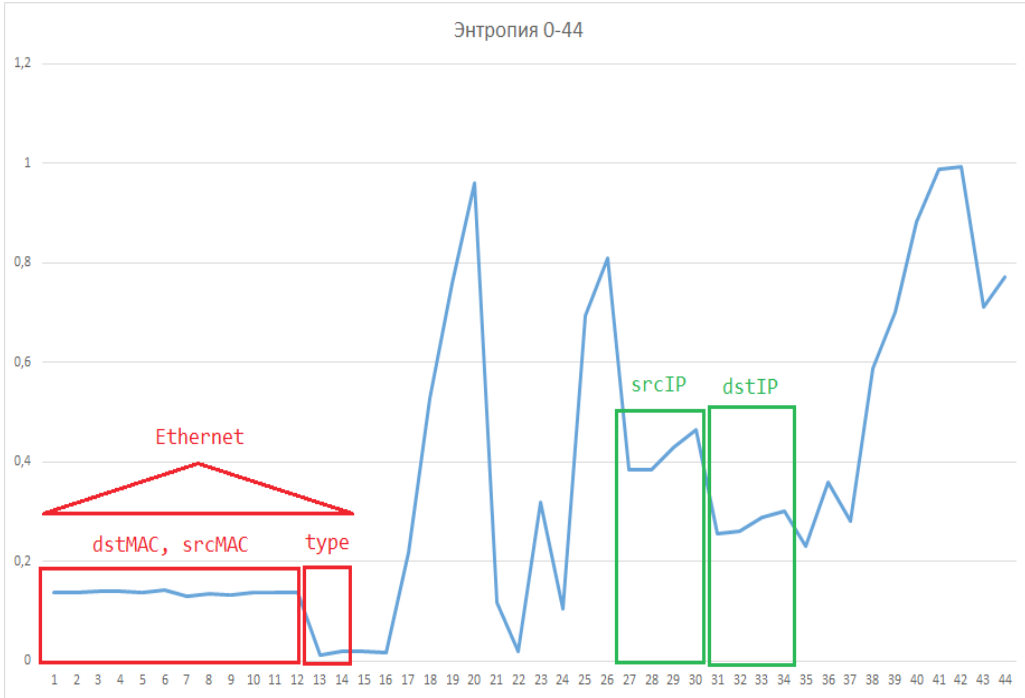


Рис. 8. Энтропия

График взаимной информации (рис. 9) показывает уменьшение значения в конце полей, описывающих IP-адреса источника и получателя, но в целом не поддаётся описанию, поэтому для разделения полей предлагается использовать энтропию и взаимную информацию совместно, принимая решение так, как описано в модели энтропийного алгоритма.

Объединение результатов в рамках статистико-энтропийного метода позволяет наложением полей из энтропийного алгоритма подтвердить полученную статистическим методом информацию о MAC-адресах. Дополнительная информация о разделении на поля может быть использована для классификации обнаруженных узлов сети.

Итеративный энтропийный алгоритм (Рисунок 6) в результате работы позволяет определить длины использованных протоколов даже в случае, если их было несколько.

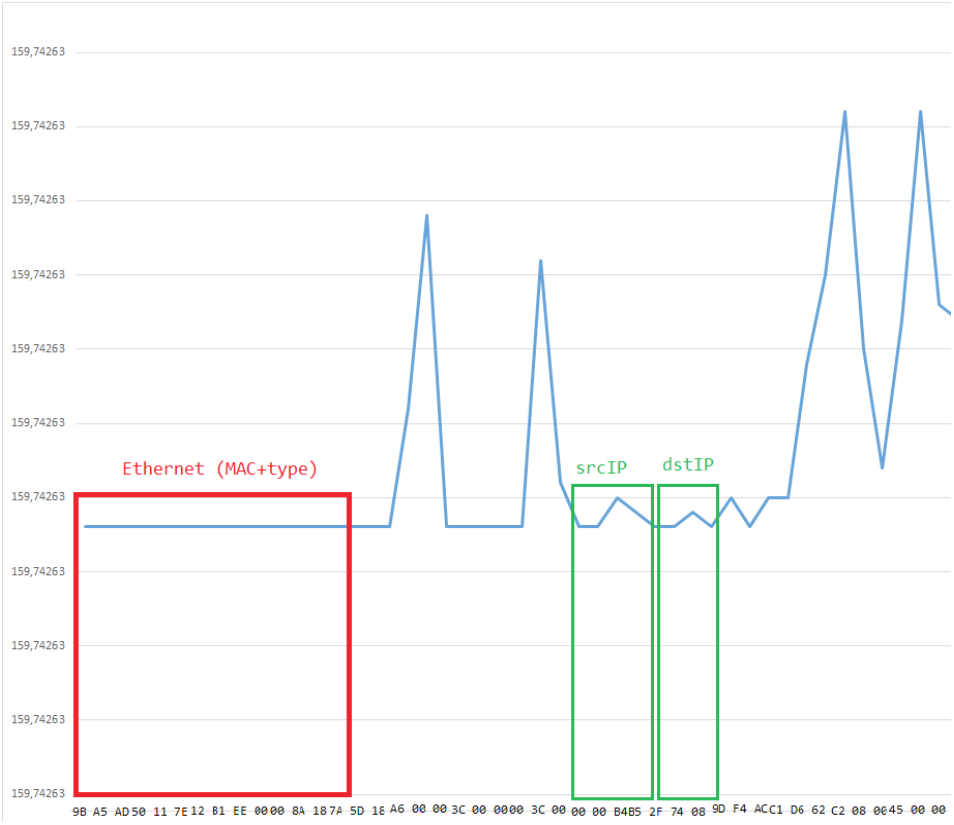


Рис. 9. Взаимная информация

## Выводы

Разработанные алгоритмы описывают метод анализа сетевого трафика при отсутствии априорных сведений о нём. Программная реализация решает поставленную задачу выделения из трафика адресов (статистический) и разделения трафика на поля (энтропийный), а их объединение позволяет получать информацию не только о наличии или отсутствии сетевых узлов, но и давать им характеристику. Разработанный итеративный энтропийный метод решает задачу анализа трафика, содержащего пакеты, относящиеся к различным протоколам.

Дальнейшее развитие проекта будет заключаться в оптимизации коэффициентов, что позволит получать более точные результаты, и в доработке итеративного энтропийного алгоритма для увеличения его производительности.

## Литература / References

1. Dmitri Bekerman, Bracha Shapira, Lior Rokach, Ariel Bar “Unknown Malware Detection Using Network Traffic Classification”, 2015 IEEE Conference on Communications and Network Security (CNS), Florence, Italy, 28-30 Sept. 2015, pp. 134-142, DOI: 10.1109/CNS.2015.7346821.

2. Rui Li, Xi Xiao, Shiguang Ni, Haitao Zheng, Shutao Xia “Byte Segment Neural Network for Network

Traffic Classification", 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4-6 June 2018, DOI: 10.1109/IWQoS.2018.8624128.

3. Antônio J.Pinheiro, Jeandrode M. Bezerra, Caio A.P.Burgardt, Divanilson R.Campelo "Identifying IoT devices and events based on packet length from encrypted traffic", Computer Communications, Volume 144, 15 August 2019, Pages 8-17, DOI: 10.1016/j.comcom.2019.05.012.

Аветисян А.И. , Гетьман А.И. Восстановление структуры бинарных данных по трассам программ. Труды Института системного программирования РАН, том 22, 2012, с. 95-118. DOI: 10.15514/ISPRAS-2012-22-7. [Avetisyan A.I., Get'man A.I. Vosstanovleniye struktury binarnykh dannykh po trassam programm. Trudy Instituta sistemnogo programmirovaniya RAN, tom 22, 2012, s. 95-118. DOI: 10.15514/ISPRAS-2012-22-7].

4. Shaun Voigt, Catherine Howard, Dean Philp and Christopher Penny "Representing and Reasoning about Logical Network Topologies" In book: Graph Structures for Knowledge Representation and Reasoning, 2018, DOI: 10.1007/978-3-319-78102-0_4.

5. Weidong Cui, Jayanthkumar Kannan, Helen J. Wang "Discoverer: Automatic protocol reverse engineering from network traces" Proceedings of 16th USENIX Security Symposium, 6-10 August 2007 Article No.: 14 Pages 1–14.

6. João Antunes Nuno Neves Paulo Verissimo "Reverse Engineering of Protocols from Network Traces", 2011 18th Working Conference on Reverse Engineering, Limerick, Ireland, 17-20 Oct. 2011, DOI: 10.1109/WCRE.2011.28.

7. Fanghui Sun Shen Wang, Chunrui Zhang, Hongli Zhang "Unsupervised field segmentation of unknown protocol messages", Computer Communications, Volume 146, 15 October 2019, Pages 121-130, DOI: 10.1016/j.comcom.2019.06.013.

---

**ДОМУХОВСКИЙ Николай Анатольевич**, заместитель генерального директора по научно-технической работе ООО «Уральский центр систем безопасности» (ООО «УЦСБ»). 620100, г. Екатеринбург, ул. Ткачей, 23. E-mail: ndomukhovsky@ussc.ru

**СИНАДСКИЙ Алексей Николаевич**, младший инженер ООО «Уральский центр систем безопасности» (ООО «УЦСБ»). 620100, г. Екатеринбург, ул. Ткачей, 23. E-mail: alexsin@e1.ru

**DOMUKHOVSKY Nikolay,** Deputy General Director for Scientific and Technical Work LLC Ural Center for Security Systems (LLC USSC). 620100, Yekaterinburg, Tkachey str., 23. E-mail: ndomukhovsky@ussc.ru

**SINADSKIY Alexey**, junior engineer LLC Ural Center for Security Systems (LLS USSC). 620100, Yekaterinburg, Tkachey str., 23; e-mail: alexsin@e1.ru