DOI: 10.51790/2712-9942-2023-4-2-09

ИСПОЛЬЗОВАНИЕ КОЭФФИЦИЕНТА ДОСТОВЕРНОСТИ ДАННЫХ ДЛЯ ОПРЕДЕЛЕНИЯ ДОСТОВЕРНОСТИ ПЕРЕДАВАЕМЫХ ДАННЫХ ПО СЕТИ

В. А. Жебель[1,а], **А. И. Солдатов**[1,2,б]

[1] *Сургутский государственный университет, г. Сургут, Российская Федерация*

[2] *Томский государственный университет систем управления и радиоэлектроники, г. Томск, Российская Федерация*

[а] *ORCID: http://orcid.org/0009-0003-9625-5250,* ✍ *vladzhebel@yandex.ru;* [б] *asoldatof@mail.ru*

Аннотация: в статье представлены результаты исследования способов защиты информации, передаваемой по вычислительным сетям, которые могут быть перехвачены и прочитаны, а также перехвачены и модифицированы. Показано, что существующие методы защищают информацию в сети с помощью криптографии (шифрования трафика), и только несколько методов проводят анализ поля TTL. В статье проведены исследования стека протоколов TCP/IP, сетевой модели OSI и полей кадра, установлены поля заголовков, которые будут использоваться в работе. Описаны инструменты, при помощи которых проводились исследования, отражается конфигурация пакетов. Предложен комплексный коэффициент достоверности для повышения достоверности передачи данных в сети предприятия за счет использования проверочных пакетов. Данный коэффициент является составным и включается в себя различные флаги и поля из сетевых протоколов стека TCP/IP, которые анализируются после приема. Предлагается гипотеза составления данного коэффициента достоверности и примерная шкала измерения. По результатам анализа делается вывод о достоверности принятых данных согласно данной шкалы измерения.

Ключевые слова: защита информации, сетевой трафик, коэффициент достоверности, поле кадра, протокол TCP/IP.

Для цитирования: Жебель В. А., Солдатов А. И. Использование коэффициента достоверности данных для определения достоверности передаваемых данных по сети. *Успехи кибернетики*. 2023;4(2):60–67. DOI: 10.51790/2712-9942-2023-4-2-09.

Поступила в редакцию: 24.04.2023. *В окончательном варианте: 24.04.2023.*

THE RELIABILITY COEFFICIENT FOR NETWORK DATA TRANSMISSIONS

V. A. Zhebel[1,a], **A. I. Soldatov**[1,2,b]

[1] *Surgut State University, Surgut, Russian Federation*

[2] *Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russian Federation*

[a] *ORCID: http://orcid.org/0009-0003-9625-5250,* ✍ *vladzhebel@yandex.ru;* [b] *asoldatof@mail.ru*

Abstract: this study considers the protection of information transmitted over computer networks that can be intercepted and decoded, or modified. It is shown that existing data protection methods use cryptography (traffic encryption). Just a few methods analyze the TTL field. We studied the TCP/IP protocol stack, the OSI network model, and the frame fields, and identified the header fields to be used. We presented the research tools and their configuration. We proposed a comprehensive reliability coefficient to assess the reliability of data transmission in corporate network through the use of verification packets. The composite coefficient includes various flags and fields from the TCP/IP network protocols, which are analyzed after reception. A hypothesis for building the reliability coefficient, and an approximate measurement scale are proposed. The analysis shows that the hypothesis and scale are correct.

Keywords: information security, network traffic, reliability coefficient, frame field, TCP/IP protocol.

Cite this article: Zhebel V. A., Soldatov A. I. The Reliability Coefficient for Network Data Transmissions. *Russian Journal of Cybernetics*. 2023;4(2):60–67. DOI: 10.51790/2712-9942-2023-4-2-09.

Original article submitted: 24.04.2023. *Revision submitted: 24.04.2023.*

Введение

В современном мире информационные данные предприятий и учреждений передаются по сети. Сети передачи данных делятся на локальные вычислительные сети и глобальные сети. Данные,

которые передаются, могут быть перехвачены и прочитаны или перехвачены и модифицированы. При таком исходе данные, которые пришли по локальной или глобальной сети, могут нести в себе как вредоносные программы, такие как вирусы, черви, троянские кони, а также модифицированные данные или просто неправильные данные. Эта проблема не нова, и разные ученые и компании решают её различными способами.

Первым и самым очевидным способом является шифрование [1]. Протоколы шифрования значительно увеличили безопасность передачи данных на некоторое время, однако с появлением мощных компьютеров, способных в реальном масштабе времени расшифровывать трафик, а также бурным развитием нейронных сетей проблема защиты данных обострилась [2]. Необходимо также указать на создание квантовых компьютеров, которые способны за довольно короткое время расшифровывать сетевой трафик. Данную проблему пытаются решить при помощи наращивания длины ключа, например, для протокола SSH длина ключа постепенно увеличивалась с 512 бит сначала до 1024 бит, далее до 2048 бит, а теперь производители оборудования рекомендуют ключ длиной в 4096 бит [3]. Это косвенно подтверждает развитие направления по расшифровке данных.

Вторым способом защиты данных являются сетевые брандмауэры, в английской терминологии firewall, или межсетевые экраны [4]. Данные устройства способны блокировать трафик извне и пропускают трафик только на выход. На выход пропускают только те данные, которые запрашивал пользователь. Реализация подобных устройств возможна как на аппаратном, так и на программном уровне [5]. Однако такие устройства не защищают от подменного трафика, если злоумышленник модифицирует трафик, который передается от запрашиваемого сервера. Кроме того, если злоумышленник находится между жертвой и сервером, он может получать ответ от сервера, переупаковывать трафик и посылать ответ машине жертвы.

Сетевых атак существует достаточно большое количество (может насчитываться несколько десятков), и они различаются по:

- характеру воздействия, оказываемого на сеть;
- цели оказываемого воздействия;
- наличию обратной связи с сетью, подвергнутой атаке;
- условию начала атаки;
- расположению субъекта по отношению к объекту атаки;
- уровню эталонной модели ISO.

Проблематикой защиты трафика от вредоносного воздействия или прослушивания занимаются многие ученые, среди которых можно отметить А. В. Черниговского, М. В. Кривова, А. Л. Истомина, В. А. Буковшина, Д. А. Чуб, И. Н. Колосок, А. Ч. Аманову. Каждый из них предлагает различные методы и способы решения указанных выше проблем на разных уровнях и для разных вычислительных сетей. Однако большинство из них предлагают для защиты трафика использовать криптографию [6] (шифрование трафика), и только пара методов обращаются к анализу поля TTL [7].

А. В. Черниговский, М. В. Кривов предлагают использовать способ анализа трафика с помощью сетевого анализатора и нейронной сети Кохонена [8]. В. А. Буковшин, Д. А. Чуб, И. Н. Колосок предлагают проводить анализ зашифрованного сетевого трафика на основе вычисления энтропии и применения нейросетевых классификаторов [9]. А. Ч. Аманова предлагает воспользоваться методом решения данной задачи при помощи применения помехоустойчивого кодирования, т.е. внесения избыточности в передаваемый по каналу цифровой сигнал для повышения достоверности передаваемой информации [10]. А. В. Неижмак предлагает использовать модель функционирования системы повышения достоверности с применением контрольных пакетов, в которой осуществляется защита от пассивного прослушивания трафика [11].

Авторы предлагают использовать коэффициент достоверности для выявления модифицированных данных, передаваемых по сети. В данной работе исследуется новый метод проверки достоверности данных, передаваемых по сети, путем передачи проверочных пакетов, которые будут иметь различные параметры на различных уровнях модели OSI, а также принятия этих пакетов на другой машине и расчета коэффициента достоверности данных, передаваемых по сети.

Предлагается ввести коэффициент, который состоит из различных параметров, где будут как качественные, так и количественные характеристики.

Материалы и методы

Предложенный нами новый метод проверки достоверности данных заключатся в расчете коэффициента достоверности информации. Метод предполагает добавление «маячков» в кадры, пакеты, сегменты трафика, которые передаются по сети и выявляются на конечной электронно-вычислительной машине. Пакеты передаются через определенное время и определенным размером. Инструменты для этого можно применять различные, как самые простые, такие как утилита ping в расширенном ее применении, так и специальное программное средство на примере Scapy (https://scapy.net/) [12]. Данная программа, по сути, является библиотекой к языку программирования Python, которая позволяет генерировать трафик с заданными критериями, а также перехватывать трафик в сети.

Проведя исследования стека протокола TCP/IP были выбраны PDU (Protocol Data Unit — обобщённое название фрагмента данных на разных уровнях стека протокола TCP/IP) различных уровней, такие как: кадр, пакет, сегмент данных. В процессе исследования кадра информативных признаков выявить не удалось, однако в данном PDU как количественную характеристику можно применить размер кадра.

Следующим рассмотрен IP-пакет данных. Исследование проводилось на пакете протокола IPv4 (рис.1). В данном пакете можно применить поля, такие как:

- DSCP (Differentiated Services Code Point), под которое выделено 6 бит. Это поле используется для разделения трафика на классы обслуживания;
- поле Flag (флаги). Для него выделено 3 бита, которые используются для контроля над фрагментацией пакетов. Нас будет интересовать второй бит;
- поле TTL (Time-to-Live) имеет размер 1 байт (8 бит) и называется «Время жизни пакета». Данное поле нужно, чтобы пакет не блуждал по сети до бесконечности в том случае, если конфигурация транзитных узлов некорректная и произошла петля маршрутизации. В данном поле можно выбрать произвольное значение для проверочного пакета и изменять его через какое-то время.

Рис. 1. *Фрагмент заголовка пакета сетевого протокола IPv4*

На сетевом уровне были рассмотрены два протокола: IPv4 и IPv6. Флаги заголовка IPv4 были описаны ранее. IPv6 в целом похож на IPv4 (рис.2).

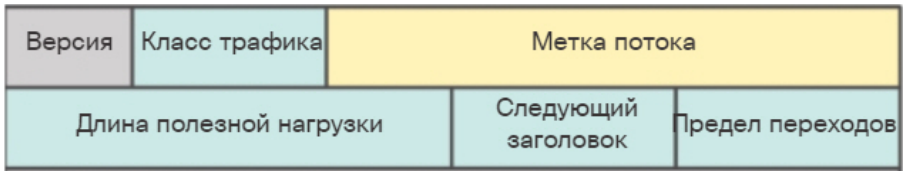

Рис. 2. *Фрагмент заголовка пакета сетевого протокола IPv6*

Существует небольшая разница в названии полей, в то же время поля «Класс трафика», где находятся флаги — DSCP (Differentiated Services Code Point), и Hop Limit (поле вместо TTL) остаются, что дает нам возможность использовать их.

Следующим является сегмент протокола управления передачей TCP (рис. 3). В данном PDU имеются:

- поле «Срочно» (Urgent), это 16-битное поле, используемое для указания срочности содержащихся данных;
- контрольные биты — 6 бит (Control bits);
- также есть гипотезы по использованию полей: опции (Options), Sequence Number и Acknolwledgement Number.

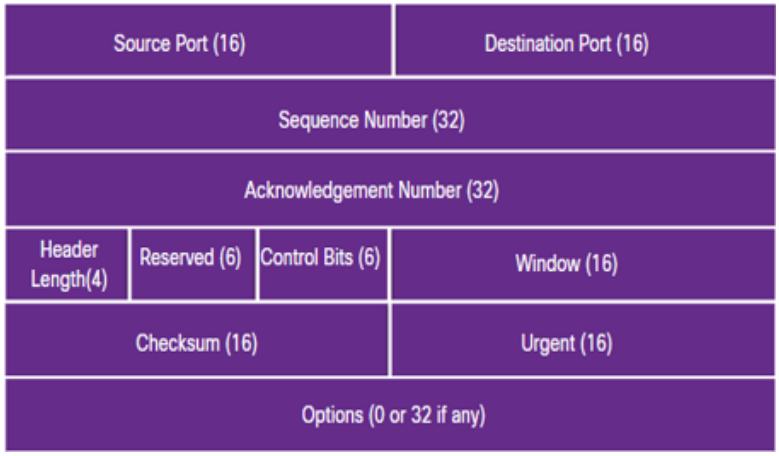

Рис. 3. *Поля сегмента сетевого протокола TCP*

Для шифрования трафика задаем интервалы времени, через которые будут отправляться проверочные пакеты. Например, пять или шесть секунд. Дополнительно рассматривается еще вариант внесения и кодирования информации, которая инкапсулируется с верхнего уровня (уровня приложения). После этого становятся известны все составляющие компоненты комплексного коэффициента, которые имеют буквенное обозначение: длина кадра — LF, длина пакета — LP, длина сегмента — LS, поле пакета DSCP — DSCP, поле пакета Flag (флаги) — F, поля пакета TTL для IPv4 и Hop Limit для IPv6 — T и H, поле сегмента Urgent протокол TCP — U, поле сегмента Control bits протокол TCP — CB, поле сегмента Options протокол TCP — O, поле сегмента Sequence Number протокол TCP — SN, поле сегмента Acknolwledgement Number — AN, информация с верхнего уровня (уровень приложения) — Data. Таким образом, комплексный коэффициент достоверности (ККД) для протокола IPv4 — см. формулу 1.

$$LF + LP + LS + DSCP + F + T + U + CB + O + SN + AN + Data = ККД. \quad (1)$$

Для протокола IPv6 ККД будет иметь вид (см. формулу 2):

$$LF + LP + LS + DSCP + H + U + CB + O + SN + AN + Data = ККД. \quad (2)$$

Знаки плюс в формулах (1) и (2) не предполагают математическое сложение, а указывают на объедение данных в одно целое. Какой математической операцией — сложения или умножения — будут объединяться компоненты, авторам предстоит еще определить по результатам будущих исследований.

Далее, стоит определиться, какие показатели будут количественными, какие качественными. Предположим, длина кадра, длина пакета, длина сегмента — это количественные показатели, остальные — качественные показатели. При отсылке определенного сообщения оно обрастает заголовками разных уровней, в которых мы помечаем определенные поля и задаем такой формат, чтобы знать длину кадра и пакета.

Результаты и их обсуждение

Для проведения экспериментальных исследований необходим набор инструментов, при помощи которых можно формировать ККД. Так, при помощи элемента библиотеки Scapy напрямую нет возможности устанавливать длину кадра (рис.4). Поэтому ее придется задавать косвенно.

При рассмотрении IP-пакета выявлены поля и определены возможности их изменения (рис.5).

```
>>> ls(Ether)
dst        : DestMACField                        = ('None')
src        : SourceMACField                      = ('None')
type       : XShortEnumField                     = ('36864')
>>>
```

Рис. 4. *Поля кадра Ethernet, которые можно задавать в Scapy*

```
>>> ls(IP)
version    : BitField  (4 bits)                  = ('4')
ihl        : BitField  (4 bits)                  = ('None')
tos        : XByteField                          = ('0')
len        : ShortField                          = ('None')
id         : ShortField                          = ('1')
flags      : FlagsField                          = ('<Flag 0 ()>')
frag       : BitField  (13 bits)                 = ('0')
ttl        : ByteField                           = ('64')
proto      : ByteEnumField                       = ('0')
chksum     : XShortField                         = ('None')
src        : SourceIPField                       = ('None')
dst        : DestIPField                         = ('None')
options    : PacketListField                     = ('[]')
>>>
```

Рис. 5. *Поля пакета IPv4, которые можно задавать в Scapy*

Как видно из рис. 5, есть программная возможно изменять поля пакета, а именно: длину пакета — LP, поле пакета DSCP — DSCP, поле пакета Flag (Флаги) — F, поля пакета TTL для IPv4.

В пакете протокола IPv6 имеются поля: «Класс трафика», где находятся флаги — DSCP и Hop Limit (поле вместо TTL), которые можно изменять (рис.6).

```
>>> ls(IPv6)
version    : BitField  (4 bits)                  = ('6')
tc         : BitField  (8 bits)                  = ('0')
fl         : BitField  (20 bits)                 = ('0')
plen       : ShortField                          = ('None')
nh         : ByteEnumField                       = ('59')
hlim       : ByteField                           = ('64')
src        : SourceIP6Field                      = ('None')
dst        : DestIP6Field                        = ('None')
>>>
```

Рис. 6. *Поля пакета IPv6, которые можно задавать в Scapy*

Результаты экспериментальных исследований по формированию пакета, его отправке по сети и перехвату при помощи программы WireShark приведены на рис. 7 и рис. 8.

Из рис. 8 видно, что пакет был отправлен на определенный IP-адрес, с определенной длиной и TTL. Однако в экспериментах было выяснено, что длина пакета, установленная вручную, не меняет длину кадра. По данному критерию нужно будет еще поработать по возможности изменения длины кадра или отказаться от манипуляций с ней.

Далее были исследованы поля протокола TCP, создана переменная tcp, и проведены эксперименты с полями данного протокола (рис. 9).

```
>>> ip = IP()
>>> ip.display
<bound method Packet.display of <IP  |>>
>>> ip.display()
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = ip
  chksum    = None
  src       = 127.0.0.1
  dst       = 127.0.0.1
  \options   \
>>> ip.ttl =100
>>> ip.src ="192.168.0.6"
>>> ip.dst ="192.168.0.3"
>>> ip.display()
>>> ip.display()
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = '100'
  proto     = ip
  chksum    = None
  src       = 192.168.0.6
  dst       = 192.168.0.3
  \options   \
>>> ip.ttl=100
>>> send (ip)
Sent 1 packets.
>>>
```

Рис. 7. *Формирование пакета в Scapy*

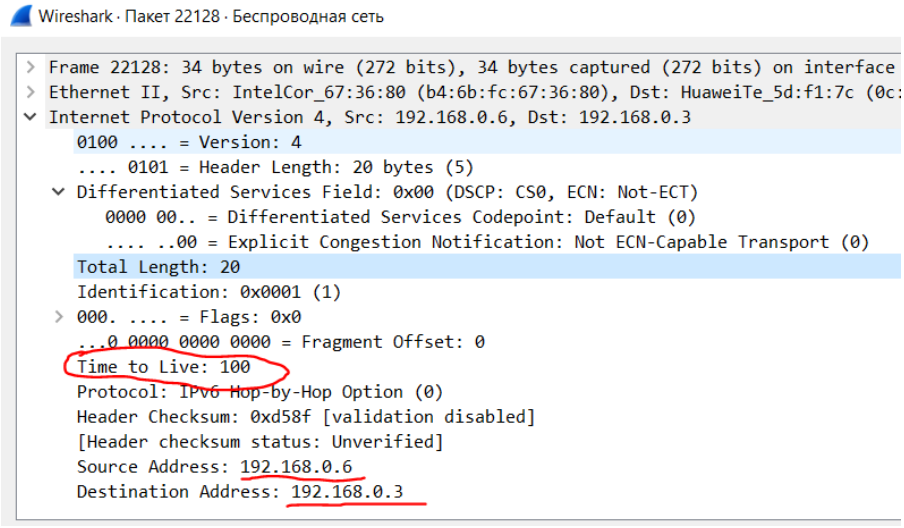

Рис. 8. *Формирование пакета в Scapy*

```
>>> tcp.display()
###[ TCP ]###
  sport     = ftp_data
  dport     = http
  seq       = 0
  ack       = 0
  dataofs   = None
  reserved  = 0
  flags     = S
  window    = 8192
  chksum    = None
  urgptr    = 0
  options   = ''
```

Рис. 9. *Поля сегмента TCP, которые можно задавать в Scapy*

Комплексный пакет, состоящий из данных «Hello», TCP-сегмента и IPv4-пакета, отправлен в сеть (рис. 10). Данные инкапсулируются в TCP-сегмент, он в свою очередь — в IPv4-пакет.

После формирования пакета проверяем его поля на адекватность и перехватываем данный пакет в программе WireShark (рис. 11).

```
>>> send(ip/TCP()/"Hello")
.
Sent 1 packets.
>>> ip/TCP()/"Hello"
<IP  frag=0 ttl=67 proto=tcp src=192.168.0.6 dst=192.168.0.3 |<TCP  |<Raw  load='Hello' |>>>
>>>
```

Рис. 10. *Отправка комплексного пакета с данными*

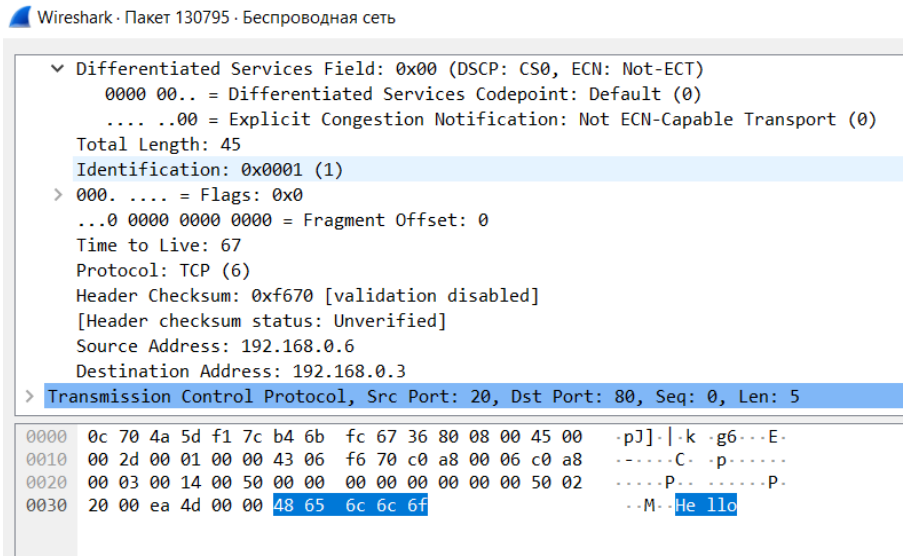

Рис. 11. *Перехват пакета в программе WireShark*

В данном пакете наблюдаем IP-пакет с заданными параметрами и сегмент TCP, в хвосте — данные «Hello».

Таким образом, в результате наблюдений и экспериментов был выявлен и утвержден инструмент, при помощи которого будет проводиться исследование, и апробированные данные и гипотезы по управлению полями в заголовках пакета и сегмента.

Заключение

В результате проведённого исследования предложен новый метод для повышения достоверности данных, передаваемых по сети, который основан на комплексном коэффициенте достоверности, состоящем из разнообразных критериев на различных уровнях стека протокола TCP/IP. Данный подход находится в стадии апробации, приготовлен экспериментальный стенд, проведены первичные эксперименты и произведен подбор инструментального программного обеспечения для его реализации. Далее необходимо написать программу на языке Питон, в которую будет импортироваться библиотека Scapy для более точной и удобной реализации программного продукта. Предложенный подход дополняет и расширяет метод поиска пассивного перехвата трафика.

В дальнейшем планируется провести исследования полей и флагов, выявить качественные и количественные критерии, а также определить их влияние на коэффициент достоверности. После этого будет сформирован комплексный коэффициент с определённой шкалой, например, от нуля до единицы. И данная шкала будет откалибрована с заданными значениями и вероятностями. Допустим, при отметке по шкале ниже 0.7 наши данные считаются недостоверными, а при отметке по шкале выше 0.8 — достоверными.

ЛИТЕРАТУРА

1. Запечников С. В. Криптографическая защита процессов обработки информации в недоверенной среде: достижения, проблемы, перспективы. *Вестник современных цифровых технологий*. 2019;1:4–16. EDN: JDKVZQ.

2. Бабенко Л. К. *Криптографическая защита информации: симметричное шифрование*: учеб. пос., 1-е изд. М.: Издательство Юрайт; 2019. 220 с. ISBN 978-5-9916-9244-1. EDN: KGIGHZ.
3. Ворожейкин Д. С. Исследование сетевого протокола прикладного уровня Secure Shell. *Инновации и инвестиции*. 2020;6:190–193. EDN: KPSSUQ.
4. Любухин А. С. Межсетевые экраны на страже информации. *Наукоемкие технологии и интеллектуальные системы в XXI веке*: сб. ст. Международ. науч.-практ. конф.: в 2-х частях. Пермь, 03 ноября 2017 года. Часть 2. Пермь: Общество с ограниченной ответственностью «ОМЕГА САЙНС»; 2017. С. 4–7. EDN: ZSEZRZ.
5. *Банк данных угроз безопасности информации*: *BDU:2023-00876: Уязвимость реализации технологии преобразования сетевых адресов Network Address Translation (NAT) виртуального сервера FastL4 межсетевых экранов BIG-IP Advanced Firewall Manager, позволяющая нарушителю вызвать отказ в обслуживании*. Режим доступа: https://bdu.fstec.ru/vul/2023-00876.
6. Багдасарян Р. Х., Осипян В. О., Литвинов К. И. и др. О технологии распределенной передачи данных и проблемах проверки достоверности информации по каналу связи. *Прикаспийский журнал: управление и высокие технологии*. 2021;4:48–57.
7. Бухарин В. В. Метод обнаружения сетевого перехвата информационного трафика информационно-телекоммуникационной сети. *Электронный журнал «Труды МАИ»*. 2012;57:1–9.
8. Черниговский А. В., Кривов М. В. Моделирование работы сетевого анализатора данных на основе SOM. *Современные технологии и научно-технический прогресс*. 2014;1:12. EDN: SHOTZV.
9. Буковшин В. А., Чуб П. А., Короченцев Д. А. и др. Анализ зашифрованного сетевого трафика на основе вычисления энтропии и применения нейросетевых классификаторов. *Известия ЮФУ. Технические науки*. 2020;6:117–128. DOI: 10.18522/2311-3103-2020-6-117-128. EDN: RFSDXK.
10. Аманова А. Ч. Обеспечение высокой достоверности передаваемых данных методом помехоустойчивого кодирования. *Потенциал современной науки*: мат. Международ. (заоч.) науч.-практ. конф. Прага, Чехия, 30 ноября 2020 г. Нефтекамск: Научно-издательский центр «Мир науки» (ИП Вострецов Александр Ильич); 2020. С. 32–36. EDN: MVKYFZ.
11. Неижмак А. В. Модель функционирования системы повышения достоверности с использованием контрольных пакетов. *Автоматизация процессов управления*. 2018;2:41–49.
12. *Официальный сайт Scapy*. Режим доступа: https://scapy.net/.