

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

DOI 10.54398/2074-1707_2022_1_9
УДК 004.4

ПАРСИНГ ТЕЛЕГРАМ-КАНАЛОВ КАК ЭЛЕМЕНТ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО АНАЛИЗА ИНФОРМАЦИИ, ПОЛУЧЕННОЙ ИЗ СЕТИ ИНТЕРНЕТ

Статья поступила в редакцию 17.01.2022, в окончательном варианте – 14.02.2022.

Карабак Илья Игоревич, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,
магистрант, ORCID: 0000-0002-5894-9273, e-mail: razvitienonedegradacia@gmail.com

Зорин Кирилл Андреевич, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,
ассистент кафедры информационной безопасности и цифровых технологий, ORCID: 0000-0003-1614-8168, e-mail: kirocan95@gmail.com

Ажмухамедов Искандар Маратович, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,
доктор технических наук, декан факультета цифровых технологий и кибербезопасности, профессор кафедры информационной безопасности, ORCID: 0000-0001-9058-123X, e-mail: aim_agtu@mail.ru

В данной статье были описаны специфика работы и протоколы одного из самых популярных мессенджеров – Telegram. Разработанное программное обеспечение позволяет автоматизировать сбор текстовых данных из публичных сообществ мессенджера Telegram. Для реализации сервисного взаимодействия был использован брокер сообщений RabbitMQ. Все полученные сообщения из мессенджера передаются в очередь брокера, к которой при наличии конфигурационных данных может подключиться любой сервис – анализатор и начать обрабатывать полученные данные. Реализованное программное обеспечение актуально для автоматизированной обработки информации, публикуемой в мессенджере.

Ключевые слова: анализ, Telegram, параметры, парсинг, короткие сообщения, сервис

PARSING OF TELEGRAM CHANNELS AS AN ELEMENT OF THE SYSTEM OF AUTOMATED ANALYSIS OF INFORMATION RECEIVED FROM THE INTERNET

The article was received by the editorial board on 17.01.2022, in the final version – 14.02.2022.

Karabak Ilya I., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation, undergraduate student, ORCID <https://orcid.org/0000-0002-5894-9273>, e-mail: razvitienonedegradacia@gmail.com

Zorin Kirill A., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation, Assistant of the Department of Information Security and Digital Technologies, e-mail: kirocan95@gmail.com

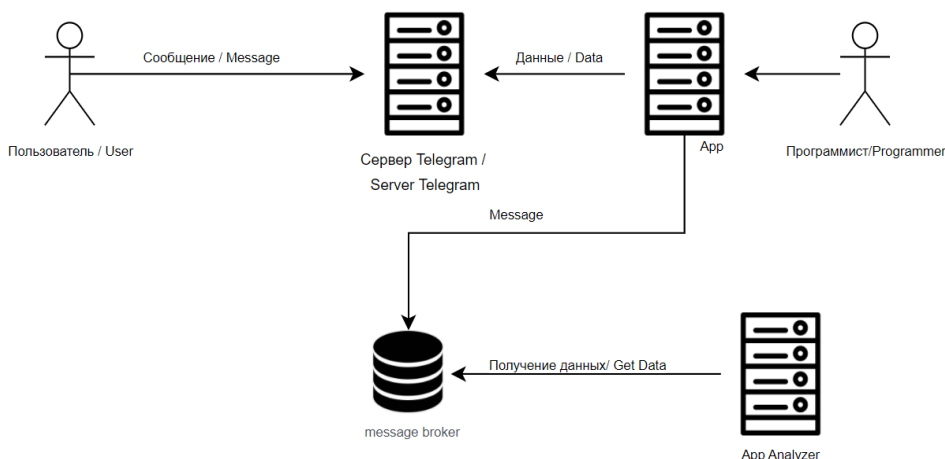
Azhmukhamedov Iskandar M., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation,

Doct. Sci. (Engineering), Dean of the Faculty of Digital Technologies and Cybersecurity, Professor of the Department of Information Security, e-mail: aim_agtu@mail.ru

This article described the specifics of the work and described the protocols of one of the most popular instant messengers – Telegram. The developed software allows you to automate the collection of text data from the public communities of the Telegram messenger. To implement the service interaction, a message broker, RabbitMQ, was used. All received messages from the messenger are transferred to the broker's queue, to which, if configuration data is available, any analyzer service can connect and start processing the received data. The implemented software is relevant for automated processing of information published in the messenger

Keywords: analysis, Telegram, parameters, parsing, short messages, service

Graphical annotation (Графическая аннотация)



Введение. На сегодняшний день интернет-пространство стало неотъемлемой частью жизни современного человека. Объемы информации, которую получает человек в современном мире, с каждым днём увеличиваются в геометрической прогрессии. В интернет-среде существуют различные источники информации, такие как онлайн-СМИ, социальные сети, мессенджеры, чаты, форумы. Информация в этих источниках может содержать мнения пользователей; общедоступные данные, размещаемые пользователями в социальных сетях; результаты поисковой выдачи и т. д.

Одним из наиболее распространённых источников распространения информации в интернет-пространстве является мессенджер Telegram, который в последние годы приобрел большую популярность в мире. Только за 4 года, с 2016 по 2020 г., аудитория данного мессенджера увеличилась более чем в 4 раза (рис. 1).

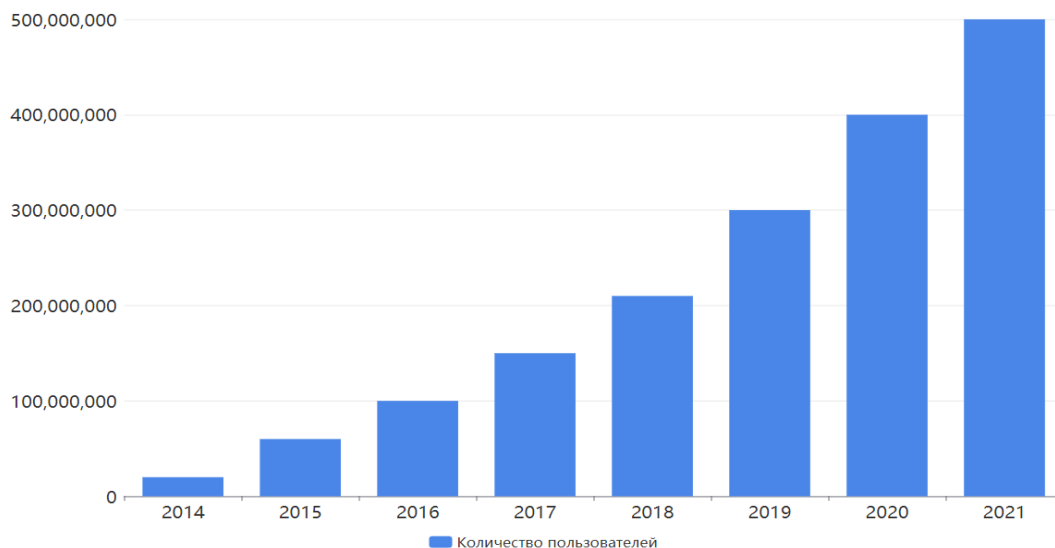


Рисунок 1 – Рост пользователей мессенджера Telegram по данным сайта telegram.org [1]

При этом среди пользователей мессенджера наиболее популярны Telegram-каналы. Существует два вида Telegram-каналов:

- публичные – общедоступный вид сообществ – такие каналы доступны любому пользователю мессенджера для чтения и подписки. Это самый распространенный формат сообществ в Telegram;
- частные – закрытый вид сообществ, доступ в которые можно получить только по ссылке, известной только администратору канала.

Пользователей Telegram привлекает его анонимность, безопасность, понятный интерфейс,

благодаря которым любой желающий может делиться с другими пользователями тем или иным контентом. Как следствие, это приводит к увеличению количества источников информации в мессенджере. Ввиду роста объема информации возникает потребность автоматизации ее анализа и систематизации, что может представлять интерес для различных организаций, например, для бизнеса или правоохранительных органов. К примеру, компании могут проводить анализ лояльности клиентов к их продукту или услуге, основываясь на комментариях пользователей в мессенджере, а правоохранительные органы – выявлять неправомерную информацию. При этом ручной сбор информации является весьма трудоемкой задачей ввиду больших объемов данных. Одним из возможных методов решения этой проблемы является автоматизация процесса сбора информации по заданным параметрам – парсинг.

В статье Т.Э. Вильданова, Н.С. Иванова «Анализ инструментов парсинга и веб-скрейпинга в рамках разработки арбитражной инвестиционной стратегии на рынке спортивных ставок» [2] приводится исследование популярных инструментов для парсинга публичных сайтов, а также выявлены наиболее быстрые инструменты и методы для быстрого извлечения и обработки информации в больших количествах. В статье Д.З. Цхошвили, Н.А. Иванова «Примеры использования технологии парсинга» [3] приводятся примеры использования технологии анализа данных – парсинга, которые представлены в современных научных работах. Однако методы и инструменты, описанные в данных статьях, не позволяют производить парсинг публичных Telegram-каналов, так как, в отличие от сайтов, мессенджер Telegram имеет свою сложную структуру хранения сообщений, а также использует отдельный протокол для работы с сообщениями. Таким образом, задача автоматизированного сбора информации, публикуемой в общедоступных группах, созданных в мессенджере Telegram, является не до конца решенной.

Постановка задачи. Исходя из этого, целью данной работы стало описание структуры разработанного программного обеспечения, которое позволит осуществлять сбор информации, публикуемой в общедоступных группах, созданных в мессенджере Telegram.

Программное обеспечение будет являться компонентом общей системы анализа текстовой информации, полученной из различных источников информации интернета. Общая схема взаимодействия компонентов такой системы, построенная в стандарте UML [2], представлена на рисунке 2.

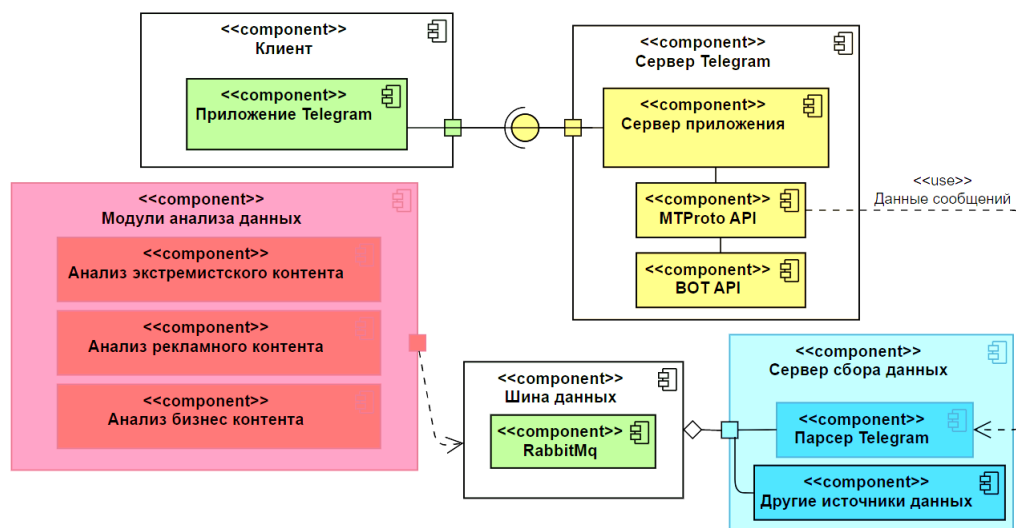


Рисунок 2 – Диаграмма развертывания системы анализа текстовой информации, полученной из различных источников интернета

Прежде чем начать разработку ПО, выполняющего функцию парсинга, требуется рассмотреть специфику мессенджера Telegram.

Специфика мессенджера Telegram. Контент, обрабатываемый в мессенджере, можно разделить на 3 группы:

- текстовый контент (сообщения, комментарии к сообщениям);
- медиаконтент (видео, аудио);
- прочие файлы (документы, архивы, файлы приложений).

В мессенджере Telegram существует два вида интернет-сообществ: чаты и каналы. Telegram-

канал представляет собой чат с односторонней связью, в котором только администратор канала может оставлять сообщения. Также, по желанию, администратор канала может предоставить доступ подписчика канала комментировать его публикации. Telegram-чат представляет из себя площадку, где все участники могут принимать участие в общении. Возможность комментировать сообщения участников отсутствует.

Для решения задачи автоматизации сбора данных, представляющих собой текстовую информацию, были проанализированы способы и методы, позволяющие взаимодействовать с программным обеспечением мессенджера Telegram. В частности, была исследована документация API-набора функций, методов, которые позволяют взаимодействовать с программным обеспечением Telegram строго в рамках политики компании [3]. Это позволило выбрать возможные пути доступа к сервисам мессенджера, которые могут быть использованы при разработке программного обеспечения для парсинга. Для того чтобы взаимодействовать с API, требуется заполнить специальную форму на сайте <https://my.telegram.org/>, с помощью которой создаётся собственное приложение Telegram API (рис. 3) [4].



Delete Account or Manage Apps

Log in here to **manage your apps** using Telegram API or **delete your account**. Enter your number and we will send you a confirmation code via Telegram (not SMS).

Your Phone Number

Please enter your number in **international format**

Next

Рисунок 3 – Заполнение формы Telegram API

Приложение хранит уникальные значения, идентифицирующие пользователя в системе Telegram API, – «api_id» и «api_hash». Значение «api_id» представляет собой последовательность цифр, а «api_hash» – буквенно-цифровую последовательность. Оба значения автоматически генерируются сервером Telegram на бессрочный период.

Идентификаторы API являются токенами приложения. Это означает, что при использовании приложения пользователю не нужно вводить собственные идентификаторы API, достаточно идентификатора разработчика.

Для взаимодействия с сервисами мессенджера пользователю предоставляется два вида API:

- MTProto API;
- Bot API.

Bot API был создан для пользователей, которые разрабатывают Telegram-боты. Боты – это специальные аккаунты, которые авторизуются с помощью токенов, а не телефонных номеров. Bot API построен на основе MTProto API, но использует промежуточный HTTP-сервер для обмена запросами с серверами Telegram. В качестве формата данных для взаимодействия с API используется формат JSON. Большинство методов MTProto API, созданных для работы с сообщениями, недоступны в Bot API.

Изначально протокол MTProto был создан командой разработчиков мессенджера как криптографический протокол для шифрования сообщений. Протокол основан на алгоритме шифрования AES [5], а также на криптографическом алгоритме Диффи – Хеллмана [6].

MTProto API является основным API Telegram, с помощью которого пользователи могут подключаться напрямую к сервисам Telegram (рис. 4). Данный протокол использует сериализацию данных в бинарный вид.

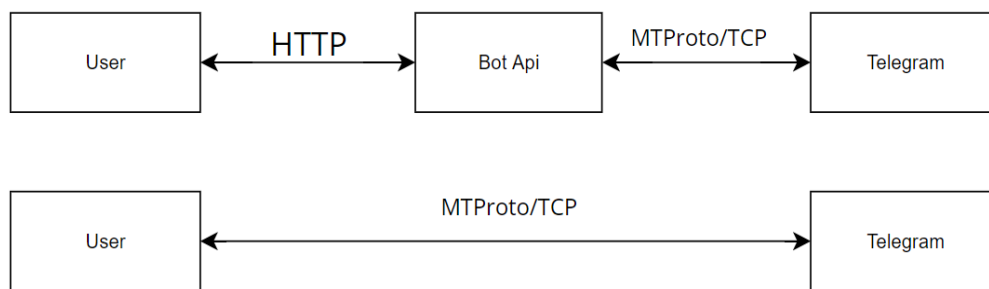


Рисунок 4 – Схема работы протоколов MTProto API и BotAPI

Bot API является упрощенным вариантом MTProto API и используется исключительно для создания ботов, снижая порог входа для обычных пользователей, тогда как MTProto API имеет большой функционал, предлагая пользователю широкие возможности работы с сервисами Telegram. Для задачи парсинга сообщений использовать Bot API невозможно, для этого предлагается использовать протокол MTProto.

Для парсинга сообщений в Telegram-сообществах мессенджер предоставляет различные методы API. Одним из таких является `messages.getHistory`, которому на вход могут подаваться следующие параметры:

- `peer` – сущность (идентификатор канала, идентификатор пользователя), от которой разработчик хочет получить сообщения; имеет тип данных `InputPeer`;
- `offset_id` – идентификатор сообщения; если параметр задан, будут возвращены сообщения, начиная с заданного идентификатора; имеет тип данных целочисленный;
- `offset_date` – дата отправки сообщения; если параметр задан, будут возвращены только те сообщения, которые были отправлены до указанной даты; имеет тип данных целочисленный;
- `add_offset` – данный параметр дополняет параметр `offset_id`; имеет целочисленный тип данных;
- `limit` – количество сообщений, которое нужно вернуть; если параметр имеет значение `None`, то метод возвратит всю историю сообщений; имеет тип данных целочисленный;
- `max_id` – если данный параметр задан, то метод вернёт все сообщения с меньшими идентификаторами; имеет тип данных целочисленный;
- `min_id` – если данный параметр задан, то метод вернёт все сообщения с большими идентификаторами; имеет тип данных целочисленный;
- `hash` – данный параметр позволяет хэшировать запрос; имеет целочисленный тип данных.

После того как запрос отправлен, метод `messages.getHistory` возвращает ответ в виде объекта типа `messages.Messages`, который содержит в себе список сообщений и информацию об этих сообщениях. Данный объект имеет четыре так называемых конструктора, которые формируют ответы на запрос пользователя в зависимости от значения параметров запроса:

- `messages.messages` – возвращает полный список сообщений и информацию об этих сообщениях;
- `messages.messagesSlice` – выдаёт список сообщений и информацию об этих сообщениях;
- `messages.channelMessages` – возвращает сообщения Telegram-канала;
- `messages.messagesNotModified` – возвращается в том случае, если сообщения не обнаружены.

В мессенджере Telegram также есть возможность комментировать публикации сообщества, если администратор сообщества даёт доступ к этой функции. Метод `messages.getReplies` позволяет получить такие комментарии в зависимости от заданных параметров. Основные параметры метода `messages.getReplies`:

- `msg_id` – идентификатор сообщения – комментария;
- `limit` – максимальное количество возвращаемых результатов.

Данный метод также возвращает объект типа `messages.Messages` в виде списка комментариев в соответствии с заданными параметрами.

Для глобального поиска сообщений мессенджера Telegram существует функция `messages.searchGlobal`, которая имеет следующие основные параметры:

- `folder_id` – идентификатор каталога, имеет тип данных `flags.0?int`;
- `q` – задаёт значение для глобального поиска, имеет строковый тип данных;
- `filter` – фильтр сообщений для глобального поиска, тип данных `MessagesFilter`.

Данный метод также возвращает объект типа `messages.Messages`.

На основе описанных выше методов было разработано программное обеспечение, которое

позволяет осуществлять сбор текстовой информации, публикуемой в общедоступных группах, созданных в мессенджере Telegram.

Описание ПО. Telegram MTProto API для языка программирования Python реализовано в виде библиотеки Telethon [7]. Данная библиотека содержит в себе множество функций и методов, позволяющих осуществлять работу с серверами мессенджера Telegram.

Для парсинга текстового контента целесообразно использовать функцию `iter_messages` библиотеки Telethon. С помощью данной функции программа отправляет запрос на сервера Telegram с заданными параметрами. Если параметры верны, программа получает ответ.

Функция `iter_messages` позволяет задать следующие параметры:

- `entity` – данный параметр определяет объект, от которого нужно получить сообщение; может быть ссылкой на канал или чат;
- `limit` – количество сообщений, которое требуется получить; если задано `None`, то параметр вернёт все сообщения объекта;
- `reply_to` – комментарии к сообщению, если известен идентификатор сообщения;
- `offset_date` – возвращает сообщения, предшествующие этой дате.

Взаимодействие программного обеспечения с Telegram API представлено на рисунке 5.

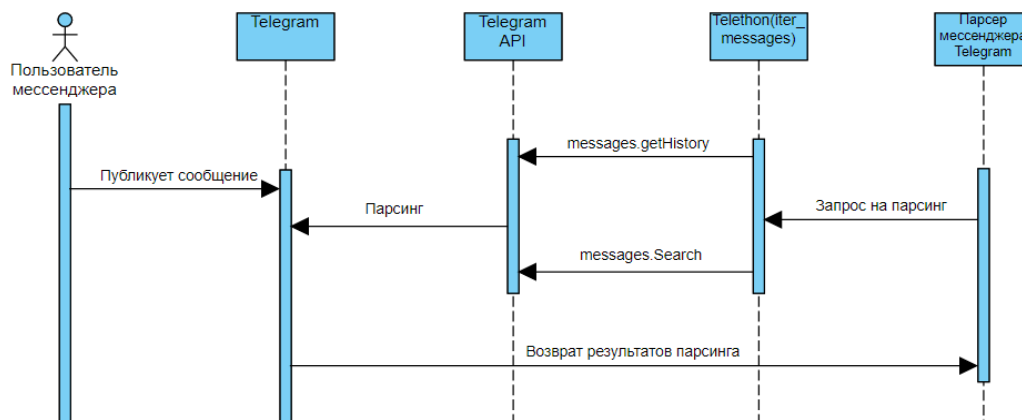


Рисунок 5 – Схема взаимодействия программного обеспечения с Telegram API

Программное обеспечение было реализовано с помощью языка программирования высокого уровня Python версии 3.7 [8]. Для нормального функционирования программе требуется постоянный доступ в интернет. Работа ведётся посредством использования командной строки. Данное программное обеспечение может собирать информацию из интернет-сообществ в мессенджере Telegram, таких как чаты и каналы. Для обоих видов сообществ можно задавать лимит выгружаемых сообщений; период, в течение которого сообщения были опубликованы; имя пользователя, который публиковал сообщения и т.д. После того как результаты парсинга были получены программой-парсером, они выгружаются в брокер сообщений.

Конечная структура ПО выглядит следующим образом (рис. 6).

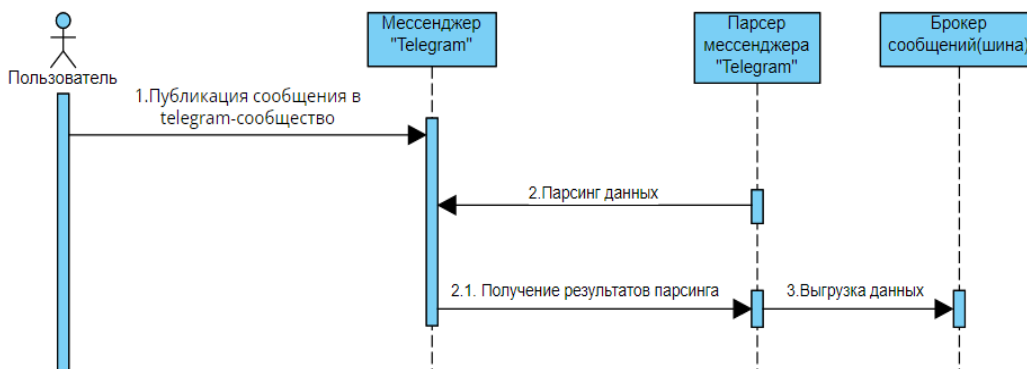


Рисунок 6 – Схема работы парсера

На данной диаграмме отображена схема работы парсера. После того как пользователь мессенджера «Telegram» публикует сообщение в телеграм сообществе, ПО автоматически производит парсинг данных о сообщении, после чего результаты парсинга отправляет в брокер сообщений RabbitMQ, тем самым создавая поток сообщений с возможностью подключения подписчиков.

Пример работы программного обеспечения. Работа с программой осуществляется через командную строку. Для корректного взаимодействия с приложением пользователю необходимо подать на вход программы ссылку на канал или чат из мессенджера Telegram (рис. 7).

Введите ссылку на канал или чат: |

Рисунок 7 – Стартовый экран программы

После того как пользователь ввёл ссылку программа начинает парсинг сообщений из канала или чата по ссылке в соответствии с заданными параметрами.

Все данные, полученные в ходе работы парсера, передаются в брокер сообщений RabbitMQ [9]. Брокер хранит в себе данные о публикациях, которые были получены в результате парсинга. Их можно использовать в дальнейшей работе для анализа информации. Для этого сервису-подписчику необходимо подключиться к брокеру, используя строку подключения, введя логин и пароль для доступа, после чего он может использовать передаваемый поток сообщений для проведения анализа и систематизации этих данных. Пример процесса отправки данных в брокер сообщений проиллюстрирован на рисунке 8.

```
[x] Отправлено Задержан вице-спикер заксобрания Красноярского края Натаров (бывший депутат Госдумы от ЛДПР), расска
По его словам, задержание связано с делом сына Натарова, которого задержали еще в 2019 по делу о мошенничестве.
https://t.me/rian\_ru/135984
[x] Отправлено По всем трем эпизодам с **истязанием заключенных в красноярских колониях, **видео с которыми появило
По его словам, первые два эпизода имеют криминальный характер и относятся к 2015 - "по ним уже проведены все мыслимы
По третьему видео, которое снято в 2020, проводилась долгая проверка - в СК пришли к заключению, что имел место суици
"Были наказаны сотрудники многие и даже те, кто не участвовал, а просто знал и не сообщил - они были уволены"
https://t.me/rian\_ru/135983
[x] Отправлено На этом видео из сгоревшей томской "Ленты" не поджигатель, как сообщали некоторые СМИ, а промоутер,
```

Рисунок 8 – Пример отправки сообщений в брокер сообщений RabbitMQ

Данные, полученные в ходе парсинга, передаются в брокер сообщений в формате данных JSON. Структура сообщения, хранящегося в RabbitMQ, представлена на рисунке 9.

```

RabbitMQ... RabbitMQ 3.9.11 Erlang 24.1.7 Refreshed 2022-01-03 16:16:30
Virtual host
Cluster rabbit@DESKTOP-1EEGBMT
User guest

Overview Connections Channels Exchanges Queues Admin

The server reported 0 messages remaining.
Exchange (AMQP default)
Routing parse_tg
Key
Redelivered
Properties
Payload
5193 bytes
Encoding: string
{
  "Sender": [
    {
      "Name": "Telegram",
      "link": "https://t.me/rian_ru"
    }
  ],
  "Messages": [
    {
      "Data": "Российская лыжница Наталья Непряева победила в масс-старте классическим стилем на этапе \"Тур де С
      \"link\": \"https://t.me/rian_ru/137581\"
    },
    {
      "Data": "В посольстве ФРГ в Москве тоже не подтвердили РИА Новости сообщения о возможной встрече канцлера Ф
      \"link\": \"https://t.me/rian_ru/137580\"
    }
  ]
}

```

Рисунок 9 – Структура хранения сообщений в RabbitMQ

Сообщение представляет из себя набор пар «ключ – значение», где в качестве значений выступают текстовый контент, полученный в результате парсинга, а также ссылки на полученные данные.

Представленное ПО используется как часть общей системы для анализа текстовой информации. Модуль зарегистрирован как программное обеспечение [12]. Для того чтобы воспользоваться возможностями данного модуля, необходимо получить логин и пароль для доступа к брокеру сообщений и считывать данные в любом режиме. Благодаря данному модулю пользователи смогут автоматизировать процесс сбора данных в различных направлениях, к примеру: сбор комментариев пользователей об услуге, товаре и т. д.

Заключение. При постоянно увеличивающемся потоке информации в интернете проблема автоматизированного сбора публичной информации в телеграм-каналах становится всё более актуальной. В данной статье была описана структура разработанного программного обеспечения, которое позволит осуществлять сбор информации, публикуемой в общедоступных группах, созданных в мессенджере Telegram. Предложенный программный продукт автоматически обрабатывает публикуемые сообщения в сообществах и группах, после чего передает данные в брокер сообщений RabbitMQ. К брокеру сообщений возможно подключить множество программ, которые будут использовать эти данные для различных целей. Основными функциями таких сервисов может быть: обнаружение экстремистских материалов, анализ рекламных блоков, анализ вовлеченности аудитории и т. д. В текущем виде предложенный модуль парсинга телеграм-каналов используется как компонент общей системы для анализа текстовой информации в интернете. Описанный в данной статье программный продукт позволяет экономить ресурсы и средства, а также осуществить в дальнейшем интеграцию модулей для анализа информации, поступающей из разных источников.

Библиографический список

1. 400 миллионов пользователей. – Режим доступа: <https://telegram.org/blog/400-million/ru>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 07.12.2021).
2. Вильданов, Т. Э. Анализ инструментов парсинга и веб-скрейпинга в рамках разработки арбитражной инвестиционной стратегии на рынке спортивных ставок / Т. Э. Вильданов, Н. С. Иванов // Скиф. Вопросы студенческой науки. – 2021. – № 5 (57). – С. 23–33.
3. Цхошвили, Д. З. Примеры использования технологии парсинга / Д. З. Цхошвили, Н. А. Иванова // Актуальные вопросы в науке и практике : сборник статей по материалам IV Международной научно-практической конференции : в 5 ч. Самара, 11 декабря 2017 года. – Самара : Общество с ограниченной ответственностью «Дендра», 2017. – С. 135–138.
4. Грейди, Буч. Язык UML. Руководство пользователя = The Unified Modeling Language user guide / Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. – 2-е изд. – Москва ; Санкт-Петербург : ДМК Пресс, Питер, 2004. – 432 с. – ISBN 5-94074-260-2.
5. Telegram APIs. – Режим доступа: <https://core.telegram.org/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 07.12.2021).
6. Delete Account or Manage Apps – Режим доступа: <https://my.telegram.org/auth>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 07.12.2021).
7. Баричев, С. Г. 2.4.2. Стандарт AES. Алгоритм Rijdael / С. Г. Баричев, В. В. Гончаров, Р. Е. Серов // Основы современной криптографии. – 3-е изд. – Москва : Диалог-МИФИ, 2011. – С. 30–35. – ISBN 978-5-9912-0182-7.
8. Черёмушкин, А. В. Криптографические протоколы: основные свойства и уязвимости / А. В. Черёмушкин // Прикладная дискретная математика. – 2009. – Вып. 2. – С. 115–150.
9. Telethon's Documentation. – Режим доступа: <https://docs.telethon.dev/en/latest/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 07.12.2021).
10. Python 3.7.0. – Режим доступа: <https://www.python.org/downloads/release/python-370/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 07.12.2021).
11. RabbitMQ. – Режим доступа: <https://www.rabbitmq.com/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 22.12.2021).
12. Крэг, Ларман. Применение UML 2.0 и шаблонов проектирования / Крэг Ларман. – Вильямс, 2019. – 736 с. – ISBN 978-5-907144-36-1.
13. Ричардсон Крис. Микросервисы. Паттерны разработки и рефакторинга. – Прогресс книга, 2021. – 544 с. – ISBN 978-5-4461-0996-8.
14. Карабак, И. И. Программа для парсинга сообщений в публичных телеграм-каналах : свидетельство о регистрации программы для ЭВМ RU 2021680437, 10.12.2021. – Заявка № 2021669987 от 30.11.2021 / И. И. Карабак, К. А. Зорин, И. М. Ажмухамедов.

References

1. 400 millionov polzovateley [400 million users]. Available at: <https://telegram.org/blog/400-million/ru> (accessed 07.12.2021).
2. Vildanov, T. E., Ivanov, N. S. Analiz instrumentov parsinga i veb-skreyepinga v ramkakh razrabotki arbitrazhnoy

investitsionnoy strategii na rynke sportivnykh stavok [Analysis of parsing and web-scraping tools as part of the development of an arbitrage investment strategy in the sports betting market]. *Skif. Voprosy studencheskoy nauki* [Scythian Issues of Student Science], 2021, no. 5 (57), pp. 23–33.

3. Tskhoshvili, D. Z., Ivanova, N. A. Primery ispolzovaniya tekhnologii parsinga [Examples of the use of parsing technology]. *Aktualnye voprosy v nauke i praktike : sbornik statey po materialam IV Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Topical issues in science and practice : collection of articles based on the materials of the IV International Scientific and Practical Conference], in 5 parts, Samara, December 11, 2017. Samara, Dendra Limited Liability Company, 2017, pp. 135–138.

4. Grady, Booch, James, Rumbaugh, Ivar, Jacobson. *Yazyk UML. Rukovodstvo polzovatelya* [The Unified Modeling Language. User guide]. 2nd ed. Moscow, St. Petersburg, DMK Press, Peter, 2004. 432 p. ISBN 5-94074-260-2.

5. *Telegram APIs*. Available at: <https://core.telegram.org/> (accessed 07.12.2021).

6. *Delete Account or Manage Apps*. Available at: <https://my.telegram.org/auth> (accessed 07.12.2021).

7. Barichev, S. G., Goncharov, V. V., Serov, R. E. 2.4.2. Standart AES. Algoritm Rijdael [AES standard. Rijdael algorithm]. *Osnovy sovremennoy kriptografii* [Fundamentals of modern cryptography]. 3rd ed. Moscow, Dialogue-MEPhI, 2011, pp. 30–35. ISBN 978-5-9912-0182-7.

8. Cheryomushkin, A. V. Kriptograficheskie protokoly: osnovnye svoystva i uyazvimosti [Cryptographic protocols: basic properties and vulnerabilities]. *Prikladnaya diskretnaya matematika* [Applied Discrete Mathematics], 2009, iss. 2, pp. 115–150.

9. *Telethon's Documentation*. Available at: <https://docs.telethon.dev/en/latest/> (accessed 07.12.2021).

10. *Python 3.7.0*. Available at: <https://www.python.org/downloads/release/python-370/> (accessed 07.12.2021).

11. *RabbitMQ*. Available at: <https://www.rabbitmq.com/> (accessed 12.22.2021).

12. Craig, Larman. *Primenenie UML 2.0 i shablonov proektirovaniya* [Applying UML 2.0 and Design Patterns]. Williams, 2019. 736 p. ISBN 978-5-907144-36-1.

13. Richardson, Chris. *Mikroservisy. Patterny razrabotki i refaktoringa* [Microservices. Patterns of development and refactoring]. Progress book, 2021. 544 p. ISBN 978-5-4461-0996-8.

14. Karabak, I. I., Zorin, K. A., Azhmukhamedov, I.M. *Programma dlya parsinga soobshcheniy v publichnykh telegram-kanalakh : svidetelstvo o registratsii programmy dlya EVM RU 2021680437* [Program for parsing messages in public telegram channels Registration certificate for the computer program RU 2021680437], 10.12.2021. Application No. 2021669987 dated 11.30.2021.

DOI 10.54398/2074-1707_2022_1_17

УДК 004.001

СОЗДАНИЕ КОРПОРАТИВНОЙ ПОЧТОВОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ

Статья поступила в редакцию 19.11.2021, в окончательном варианте – 13.01.2022.

Иванова Наталья Александровна, Брянский государственный университет им. академика И.Г. Петровского, 241036, Российская Федерация, г. Брянск, ул. Бежицкая, 14,

кандидат технических наук, доцент, ORCID: 0000-0001-5135-9310, e-mail: ivanova.na@brgu.ru

Кубанских Олеся Владимировна, Брянский государственный университет им. ак. И.Г. Петровского, 241036, Российская Федерация, г. Брянск, ул. Бежицкая, 14,

кандидат физико-математических наук, доцент, ORCID: 0000-0003-0320-1652, e-mail: netbay.ov@brgu.ru

В данной статье рассмотрен процесс развертывания корпоративного почтового сервера посредством системы управления конфигурациями. Проведен анализ наиболее распространенных СМ-систем с учетом используемой архитектуры, наличия web-интерфейса, используемого языка, поддерживаемых платформ, применяемых настроек на контролируемом хосте. В работе были определены технические требования, необходимые для корректного функционирования системы управления конфигурациями и корпоративного почтового сервера. Контроль и управление трафиком почтовой службы вуза, максимальная защита обмена деловой корреспонденции от массовых рассылок и внешних атак, организация комфортного взаимодействия коллег – всё это возможности реализованной корпоративной почтовой системы. Данный способ организации корпоративной почты с использованием системы управления конфигурациями упрощает работу по настройке и развертыванию почтового сервера, оптимизируя работу технического персонала и образовательного учреждения в целом.

Ключевые слова: почтовый сервер, корпоративная почта, система управления конфигурациями, iRedMail, Ansible