УДК 004.42

*И. В. Коротаев*[1*]

# Инструмент обеспечения безопасности информационных ресурсов при мгновенном обмене сообщениями

[1] Национальный исследовательский ядерный университет «МИФИ», г. Москва, Российская Федерация
* e-mail: ivkor01@mail.ru

**Аннотация.** В статье приводятся результаты исследования защищенности систем мгновенного обмена сообщениями на базе протокола *XMPP*. В случае некорректной конфигурации серверной части системы злоумышленник способен получить конфиденциальные данные пользователей, что позволит ему в будущем провести фишинговые кампании, атаки полным перебором, а также упростит закрепление в атакуемой инфраструктуре. С целью предотвращения подобных угроз спроектирован и разработан программный инструмент, в автоматизированном режиме разворачивающий и настраивающий корпоративную вычислительную сеть в соответствии с определенными в ходе исследования предметной области требованиями информационной безопасности. Полученное программное решение было успешно протестировано путем проведения как мануального, так и автоматизированного тестирования. Благодаря использованию автоматизации инструмент позволяет ускорить процессы администрирования сети в сравнении с ручным конфигурированием, а также гарантировать приведение системы в состояние, описанное в конфигурационных файлах за счет исключения влияния человеческого фактора и минимизации влияния сбоев программного обеспечения.

**Ключевые слова:** корпоративная вычислительная сеть, система обмена мгновенными сообщениями, *XMPP*, Ejabberd, Ansible, OpenVPN

*I. V. Korotaev*[1*]

# The software tool for ensuring security of information resources during instant messaging

[1] National Nuclear Research University "MEPhI", Moscow, Russian Federation
*e-mail: ivkor01@mail.ru

**Abstract.** The article presents the results of study of the security of instant messaging systems based on the *XMPP* protocol. In the case of an incorrect configuration of the server part of the system, an attacker is able to obtain confidential user data, which will allow him to conduct future phishing campaigns, brute-force attacks, and also make it easier to gain a foothold in the attacked infrastructure. In order to prevent such threats, a software tool has been designed and developed to automatically deploy and configure a corporate computer network in accordance with the information security requirements identified during the study of the subject area. The resulting software solution was successfully tested by both manual and automated testing. Thanks to the use of automation, the tool allows you to speed up network administration processes compared to manual configuration, as well as ensure that the system is brought to the state described in the configuration files by eliminating the influence of the human factor and minimizing the impact of software failures.

**Keywords:** corporate computer network, instant messaging system, XMPP, Ejabberd, Ansible, OpenVPN

## *Введение*

На сегодняшний день системы обмена мгновенными сообщениями предлагают широкий спектр полезных функций, в которые входят общение в режиме реального времени путем отправки текстовых сообщений и аудиосообщений, создание групповых чатов и видеоконференций, обмен файлами, возможность удаления и редактирования отправленных сообщений. Благодаря этому, а также бесплатной модели распространения многих мессенджеров (англ. *Messenger* – клиентская программа, необходимая для обмена мгновенными сообщениями), количество пользователей подобных сервисов стремительно растет. По статистике исследовательской компании *The Radicati Group*, в 2023 году в сети Интернет имеется 8,976 миллиардов аккаунтов в *IM*-системах (сокращенно от англ. *Instant Messaging*), к 2025 году ожидается, что это число станет равным 9,536 миллиардов (без учета аккаунтов, зарегистрированных в системах и приложениях для мобильных устройств) [1].

*IM*-сервис – одна из технологий, созданных для коммуникации между людьми, нашедшая применение во многих различных областях. В сфере профессиональной деятельности внедрение мессенджеров позволило сильнее сплотить коллектив и увеличить количество совместной работы, что позитивно сказалось на общей продуктивности компании [2]. Общение в реальном времени также может использоваться в обучении как дополнительный инструмент, позволяющий преподавателю чаще консультировать ученика и отправлять ему учебные материалы [3]. Системы обмена мгновенными сообщениями могут служить в качестве средства распространения информации для большой группы людей. Мессенджеры позволяют создавать ботов – программы, работающие в качестве помощника, которые посредством заранее заданного набора команд и удобного интерфейса позволяют выполнять ряд элементарных задач [4].

Системы обмена мгновенными сообщениями используют компьютерные сети для передачи отправленных сообщений. Это означает, что отправка данных в глобальную сеть в открытом виде без использования каких-либо мер безопасности с высокой вероятностью приведет к нежелательным сценариям, прямо нарушающим фундаментальные принципы информационной безопасности: конфиденциальность, целостность, доступность. Также факт того, что такие сервисы позволяют отправлять не только текстовые сообщения, но и файлы, говорит о том, что они могут использоваться злоумышленниками в качестве инструмента для распространения вредоносного программного обеспечения. Поэтому аспекту сетевой безопасности *IM*-сервисов уделено немало внимания со стороны технического и научного сообщества. Работа Iraklis Symeonidis и Gabriele Lenzini направлена на изучение угроз при обмене мгновенными сообщениями и их систематизацию [5]. В статье Christian Johansen, Aulon Mujaj, Hamed Arshad и Josef Noll в журнале «Hindawi» проводится исследование и сравнительный анализ существующих на рынке мессенджеров с точки зрения их безопасности [6]. Ряд работ Mauli Bayu Segoro, Prasetyo Adi Wibowo Putro, Muhammad Nursalman, P Rizky Rachman Judie, Ammar Ashshiddiqi рассматривают алгоритмы шифрования в мессенджерах как один из методов защиты информации [7, 8].

С целью обеспечения полноценной информационной безопасности пользователей и их информации в среде *IM*-систем необходимо рассматривать не только угрозы, возникающие во время непосредственного продвижения данных по открытым каналам связи в глобальной сети, но также и угрозы, исходящие от самого окружения, в котором эти системы разворачиваются. В качестве такой превентивной меры может выступать инструмент, в автоматическом режиме подготавливающий среду для коммуникации и гарантирующий ее защищенность. Автоматизация развертки и конфигурирования программного обеспечения является важным аспектом в задачах, связанных с информационной безопасностью. Инструмент, работающий в автоматическом режиме, позволяет исключить человеческий фактор из задач администрирования, а также уменьшить риски некорректной настройки по причине сбоев программ или операционных систем, возникших из-за непредвиденных обстоятельств или их несовершенства.

### *Постановка задачи*

Целью данной работы является разработка и программная реализация инструмента обеспечения безопасности информационных ресурсов при мгновенном обмене сообщениями по открытым каналам связи в вычислительной сети путем автоматизированной безопасной развертки и конфигурирования корпоративной инфраструктуры.

В ходе проведения исследовательской работы был выделен ряд задач:

– исследование и анализ предметной области;
– моделирование предметной области;
– проектирование алгоритма работы инструмента;
– программная реализация инструмента;
– комплексное тестирование разработанного решения.

### *Исследование предметной области*

Стремительное распространение *IM*-сервисов вызвало интерес у разных предприятий в сфере *IT*, многие из которых занимаются разработкой собственных систем обмена мгновенными сообщениями. Каждая компания пытается привнести в свое решение определенные особенности, выделяющие ее среди конкурентов, что также отражается на принципах работы и функциональных возможностях того или иного *IM*-сервиса.

Однако логика работы рассматриваемых систем всегда базируется на основе клиент-серверной модели подобно системам электронных писем, поэтому для их использования необходима клиентская программа, называемая мессенджером. Мессенджер выступает инструментом, с помощью которого пользователь может отправить сообщение *IM*-серверу, который в свою очередь выполняет функции маршрутизатора в *IM*-сети. В случае, когда серверы принадлежат только определенному лицу, мессенджер называется централизованным. Обратно, когда любой желающий имеет возможность организовать собственный *IM*-сервер, мессенджер называется децентрализованным.

Разработчики технологий обмена мгновенными сообщениями как и в случае с разработкой стека *TCP/IP* поначалу не задавались вопросами сетевой безопасности их будущих приложений, что со временем привело к обеспокоенности пользователей проблемами конфиденциальности их данных, так как обмен данными в глобальной сети Интернет всегда подвержен риску. Сегодня это привело к повсеместному внедрению в мессенджеры средств безопасности, в частности, метода сквозного шифрования, при котором зашифрованные данные могут расшифровать только отправитель и получатель, что оберегает информацию от вмешательства третьих лиц в сеанс общения.

С помощью сквозного шифрования возможно обеспечить принципы конфиденциальности, но оно никак не помогает сохранить анонимность в сети, поскольку в случае использования централизованного мессенджера все сообщения проходят через серверы провайдера и позволяют ему определить пользователя по его уникальному идентификатору в системе или же по его *IP*-адресу. По этой причине вырос спрос на *IM*-сервисы, основанные на децентрализованных *IM*-протоколах, решающих одновременно вопросы как конфиденциальности, так и анонимности, одним из которых является протокол *XMPP* (англ. *eXtensible Messaging and Presence Protocol*).

Расширяемый протокол обмена сообщениями и информацией о присутствии *XMPP*, также известный как *Jabber* – это открытый, свободный для использования протокол для обмена мгновенными сообщениями и информацией о присутствии в режиме реального времени. Начало разработки протокола началось в 1999 году Джереми Миллером с создания сервера *jabberd* [9]. Тогда при появлении первых версий протокола *Jabber* были сформированы основы для современного протокола *XMPP*, в последующем закрепленные в виде *RFC 3920* (англ. *Request for Comments*) в 2004 году [10].

Протокол нашел применение в разных сферах. Компании *Google* и *Apple* используют его для отправки push-уведомлений пользователям. Распределенная социальная сеть *Movim* построена на основе *XMPP*. Данный протокол используется для поддержки внутриигровых чатов в онлайн-играх. *XMPP* способствует развитию *Интернета Вещей* (англ. *Internet of Things*, сокр. *IoT*) [11]. Нередко его используют в качестве протокола, с помощью которого умные устройства обмениваются между собой информацией.

Среди главных особенностей рассматриваемого *IM*-протокола, позволивших ему стать столь распространенным, стоит отметить [12]:

– децентрализацию. В XMPP-архитектуре не существует единого управляющего сервера, которым обязаны пользоваться все клиенты – каждый имеет право организовать собственную IM-систему и пользоваться ей. Это отличает данное решение от проприетарных мессенджеров (WhatsApp, Telegram и т. д.);

– стандартизацию. Инженерный совет Интернета IETF (англ. Internet Engineering Task Force) формализовал данную технологию и опубликовал ряд спецификаций: RFC 3920, RFC 3921, RFC 6120, RFC 6121, RFC 6122, RFC 7590, RFC 7622. Это означает, что множество открытых реализаций клиентов и серве-

ров XMPP будут совместимы и будут поддерживать определенный в стандартах функционал;

– расширяемость и гибкость.

Благодаря функциям языка разметки *XML* (англ. *eXtensible Markup Language*) имеется возможность разработать собственное расширение для *XMPP* поверх существующих решений, что активно способствует развитию технологии в целом. Наиболее значимые расширения, созданные сообществом, стандартизируются и публикуются организацией *XMPP Standards Foundation* (сокр. *XSF*) в виде *XEP*-документов для обеспечения совместимости [13]. Расширения позволяют использовать *XMPP* не только как технологию для обмена текстовыми сообщениями. С помощью существующих расширений возможно использовать протокол для обмена файлами, реализации групповых чатов, определения геолокации и *IP*-телефонии.

Как уже упоминалось выше? многие организации и обычные граждане заинтересованы в применении *XMPP* либо в качестве технологии для *IM*-общения, либо в качестве технологии, лежащей в основе приложения или его части. Так как *XMPP* работает по клиент-серверной модели, то необходимо развернуть свой собственный сервер или подключиться к существующему. Однако при разворачивании и настройке сервера стоит отметить, насколько важна его правильная конфигурация, поскольку в обратном случае это открывает широкий спектр возможностей для злоумышленника. Проблема становится особенно значимой, если принять к сведению тот факт, что нередко *XMPP*-серверами пользуются медицинские центры, фабрики и заводы тяжелой промышленности, крупные *IT*-компании. Посредством неправильно настроенного сервера киберпреступник способен получить конфиденциальные данные клиентов или сотрудников фирмы и закрепиться в ее инфраструктуре, что поможет ему педантично подготовить будущие атаки.

Одним из существующих *XMPP*-расширений является “*XEP-0077: InBand Registration*”. Данное расширение реализует функцию регистрации через клиента (англ. *In-Band Registration*), разрешая любым пользователям самим создавать себе учетные записи на сервере без участия администраторов и сторонних приложений для регистрации. Другим расширением является “*XEP-0175: Best Practices for Use of SASL ANONYMOUS*”, описывающее рекомендации по использованию *SASL ANONYMOUS* механизмов из *RFC 4505* в *XMPP*. Это расширение позволяет реализовать функцию анонимного входа на *XMPP*-сервер. Именно с помощью этих двух функций злоумышленник имеет возможность легально проникнуть на сервер и впоследствии получить конфиденциальную информацию, в связи с чем при конфигурации частного закрытого сервера необходимо их отключать.

Получив доступ к учетной записи атакуемого сервера одним из описанных способов, киберпреступник может использовать следующие расширения в целях сбора данных:

– XEP-0030: Service Discovery. Данное расширение позволяет запросить у сервера информацию о существующих на нем XMPP-сущностях и их свойствах.

К примеру, такими сущностями могут быть групповые чаты на сервере, а также использующиеся на сервере XEP;

– XEP-0055: Jabber Search. Расширение описывает функции по поиску информации на XMPP-сервере. Чаще всего используется для поиска зарегистрированных пользователей. У найденных профилей возможно просмотреть всю указанную в них информацию, в том числе ФИО, адрес электронной почты, никнейм, иногда название компании сотрудника, его должность, номер телефона и прочее. Особо критична утечка подобной информации в случае, если сервером пользуется какая-либо организация. Наличие таких данных в руках злоумышленника означает возможные фишинговые кампании и атаки типа Password Spraying, заключающиеся в том, что к большому количеству известных логинов применяется один и тот же пароль в надежде, что одна из пар логин-пароль окажется верной и возможно будет получить доступ к атакуемому ресурсу;

– XEP-0045: Multi-User Chat.

Это расширение определяет функции многопользовательского чата в *XMPP* системе со всеми соответствующими функциями: приглашение пользователей и их удаление, роли администраторов комнаты, установка пароля для присоединения и так далее. С точки зрения потенциальной утечки информации особенно важна функция истории, то есть сохранения переписки. Некоторые чаты в *XMPP* предоставляют всю историю сообщений вновь присоединившимся пользователям, что означает, что злоумышленнику достаточно лишь на момент присоединиться к комнате, получить данные переписки и выйти, не дожидаясь начала общения.

Неправильная конфигурация описанных выше *XMPP*-расширений позволяет злоумышленнику провести сетевую разведку и кражу данных путем легального входа на атакуемый сервер. Однако утечка информации может произойти по другим причинам. После отправки *XMPP*-сообщений они маршрутизируются в глобальной сети и проходят по открытым каналам связи. Таким образом, при отсутствии шифрования отправленных данных высока вероятность прослушивания сетевого трафика клиентов *XMPP*-сети злоумышленником, что возможно в случае отсутствия корректной настройки стека протоколов *TCP/IP* на *XMPP*-сервере, так как он в своем базовом виде не поддерживает какие-либо механизмы обеспечения информационной безопасности пользователей [14]. Привнесение встроенного шифрования достижимо за счет использования современных криптографических протоколов *TLS* (англ. *Transport Layer Security – Протокол защиты транспортного уровня*) версии 1.2 и 1.3, так как версии ниже указанных, а также предшественник *TLS* - протокол *SSL* (англ. *Secure Sockets Layer*), на сегодняшний день считаются устаревшими и небезопасными. Важность применения алгоритмов шифрования при сетевом взаимодействии подробно освещена в работах как отечественных ученых Е. А. Басыни, Ж. Б. Ахаевой [15], так и зарубежных исследователей Gabriel Arquelau Pimenta Rodrigues, Gabriel De Oliveira Alves [16].

Действие криптографических протоколов при защите каналов связи между узлами *IM*-сети было продемонстрировано на экспериментальном стенде, состо-

ящем из двух виртуальных машин (*ОС – Ubuntu 20.04*) с предустановленными *XMPP*-клиентом и *XMPP*-сервером (*Spark 2.9.4* и *Openfire 4.7.1*). На компьютере-хосте (*ОС – Windows 10*) предустановлен анализатор сетевого трафика (*Wireshark 3.6.5*).

На рис. 1 продемонстрировано произвольное *XMPP*-сообщение, отправленное от одного клиенту другому при отключенном шифровании на порту 5222 сервера и перехваченное в анализаторе трафика. В данном случае присутствует возможность полностью извлечь текст сообщения.

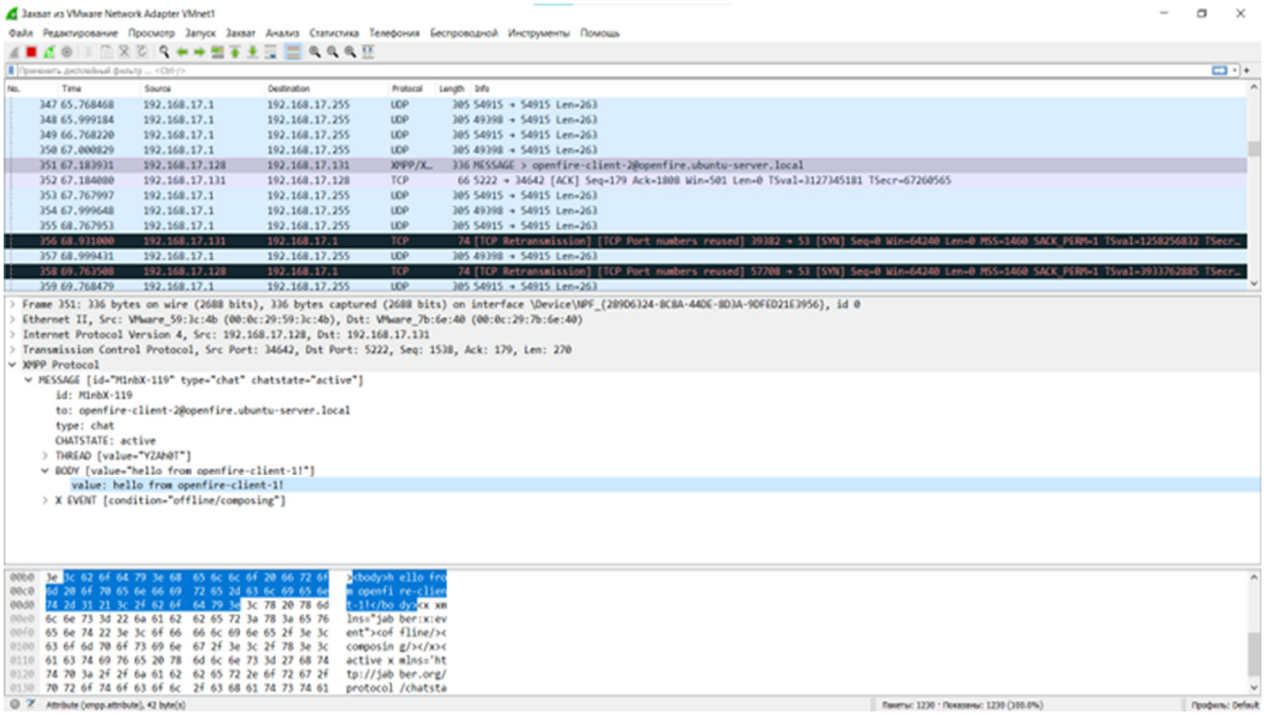


Рис. 1. Перехваченный пакет в *Wireshark* без *TLS*

На рис. 2 также продемонстрирована отправка произвольного сообщения, но с применением протокола *TLS* v1.2. Текст сообщения передается в зашифрованном виде, и не имеется возможности извлечь его содержимое без знания ключей, примененных в процессе шифрования.

Протокол *TLS* v1.2 использует асимметричное шифрование для обмена ключами и симметричное шифрование для защиты передачи данных. Процесс формирования общего ключа для применения симметричного шифрования называется *TLS*-рукопожатием, и стоит отметить, что во втором случае его также можно наблюдать в анализаторе трафика.

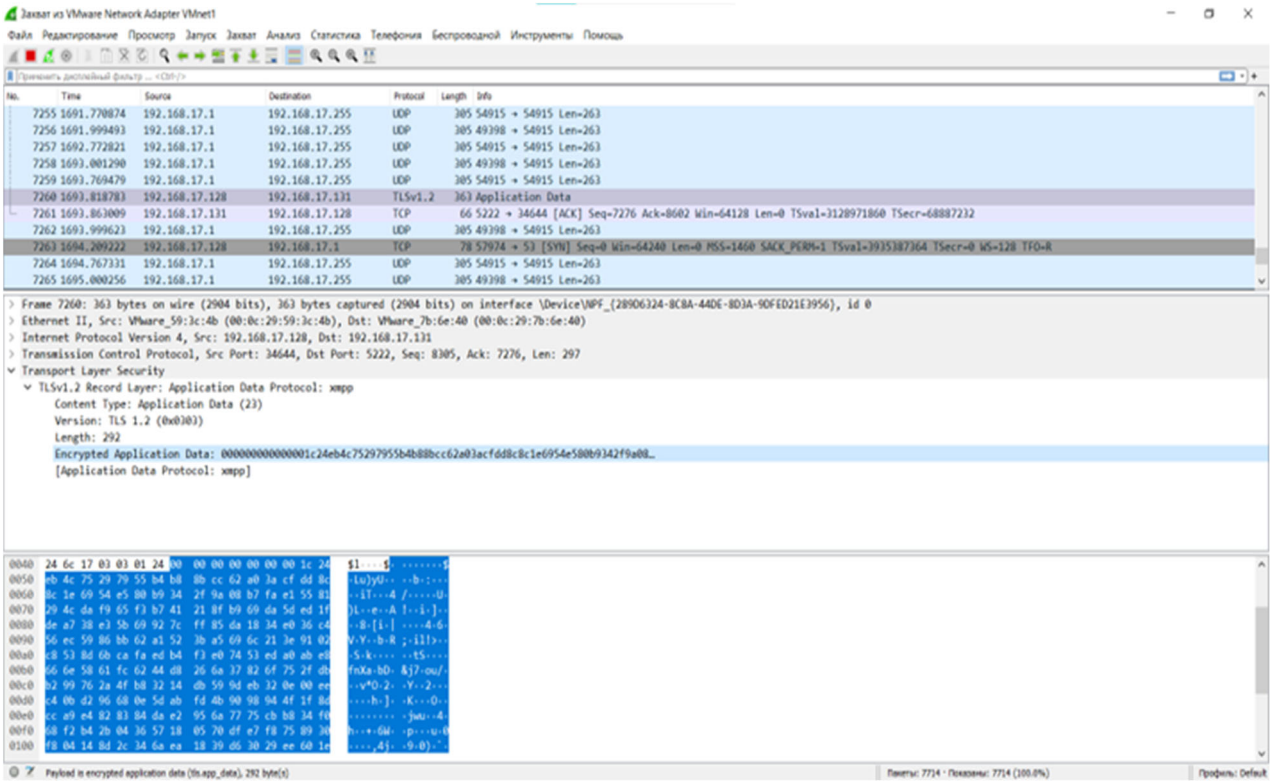

Рис. 2. Перехваченный пакет в *Wireshark* с *TLS*

### *Моделирование предметной области*

В результате исследования и анализа протокола обмена мгновенными сообщениями *XMPP* возможно построить *UML*-диаграмму прецедентов, отражающую возможные взаимодействия акторов с *IM*-системой. Данная диаграмма изображена на рис. 3.

На ней можно выделить три актора: администратор *XMPP*-сети, ее пользователь и злоумышленник, атакующий систему, а также их возможные взаимодействия с системой, то есть прецеденты. С помощью построенной диаграммы возможно точнее определить необходимые решения по обеспечению безопасности сервиса при программной реализации инструмента.

Таким образом, для обеспечения комплексной защиты *XMPP*-сети необходимо провести корректную и безопасную конфигурацию сервера в соответствии с требованиями сетевой безопасности, в первую очередь для упреждения возможных анонимных регистраций злоумышленника в сети. Также необходимо защитить каналы связи, по которым передаются сообщения клиентов, от вмешательства третьих лиц и прослушивания.

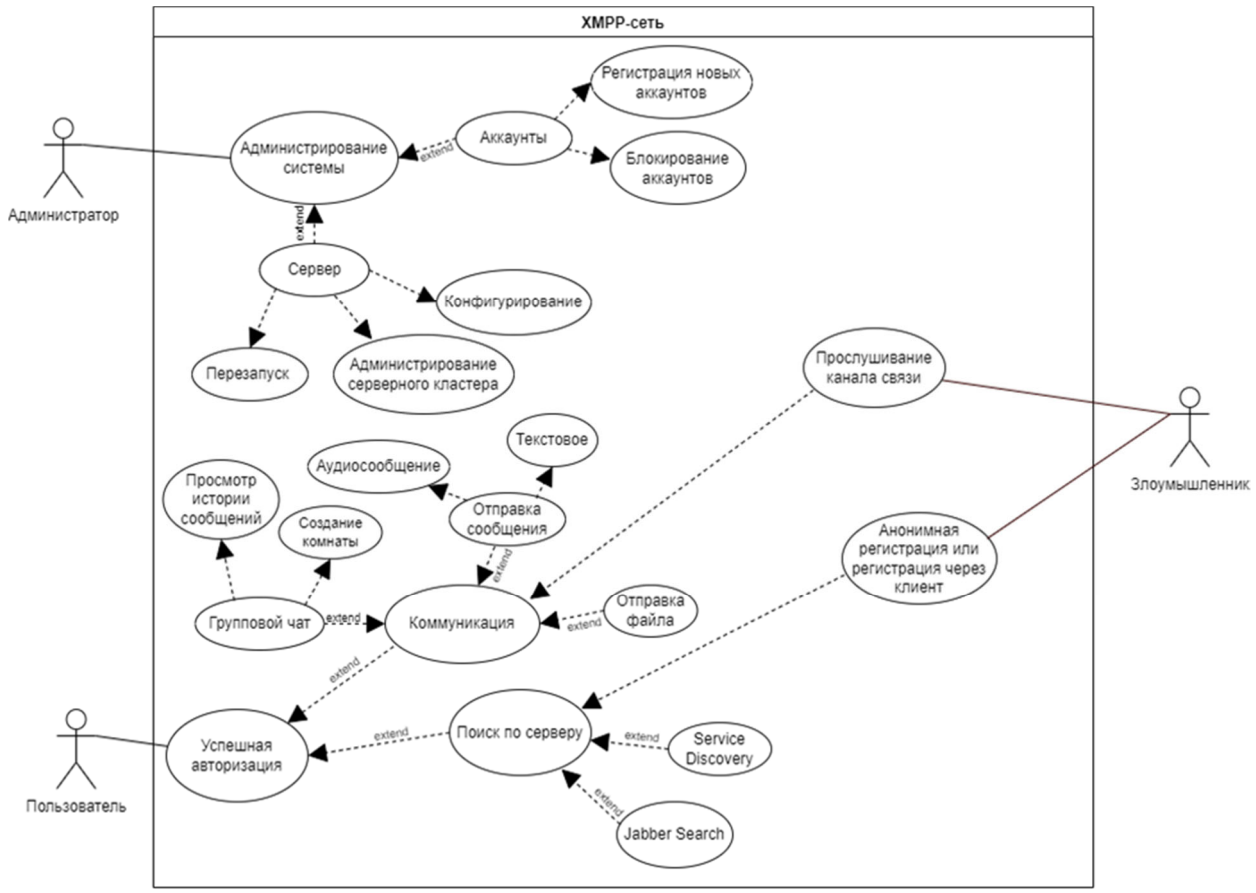


Рис. 3. *UML*-диаграмма прецедентов, моделирующая взаимодействие с *XMPP*-сетью

### *Предлагаемое решение*

При создании собственного инструмента по обеспечению безопасности информационных ресурсов при мгновенном обмене сообщениями прежде всего предлагается разработать собственные конфигурационные файлы для используемого *XMPP*-сервера, поскольку они в своем исходном виде не способны обеспечить требуемый уровень безопасности сервера и не устраняют уязвимости, описанные в предыдущих пунктах работы. В качестве *XMPP*-сервера был выбран *ejabberd* – это свободный, распределённый и устойчивый к отказам сервер, написанный преимущественно на языке программирования *Erlang*. Он поставляется с базовым файлом для настройки, в который были внесены следующие изменения:

– запрет на анонимную регистрацию и регистрацию собственных аккаунтов для упреждения проникновения злоумышленника на сервер;

– использование электронных сертификатов в принудительном порядке для обеспечения шифрования;

– использование ACL-правил для управления доступом к определенным возможностям сервера;

– смена портов по умолчанию для затруднения сетевой разведки злоумышленнику;

– обязательное использование криптографических протоколов при различных соединениях сервера и запрет на использование устаревших и небезопасных криптографических протоколов (всех версий протокола SSL, TLS v1.0 и TLS v1.1), использование ряда криптостойких способов шифрования;

– использование журналирования для хронологической записи событий на сервере;

– ограничение скорости передаваемых данных для обычных пользователей и количества возможных подключений к серверу для предотвращения атак типа «отказ в обслуживании».

Далее для большего повышения уровня защищенности корпоративной среды коммуникации предлагается развернуть инфраструктуру открытых ключей и *VPN*-инфраструктуру (англ. *Virtual Private Network – Виртуальная частная сеть*). Исходя из построенной *UML*-диаграммы прецедентов было решено привнести данные механизмы защиты, поскольку за счет установления зашифрованных каналов связи между устройствами, то есть *VPN*-туннелей, возможно защитить трафик от прослушивания и возможных вмешательств злоумышленником, а с помощью используемых в *VPN* механизмов идентификации, аутентификации и последующей авторизации – защитить всю сеть от несанкционированного доступа и скрыть ее топологию от атакующего. В качестве технологии *VPN* была выбрана *OpenVPN* – это одна из реализаций технологии виртуальной частной сети с открытым исходным кодом для создания зашифрованных каналов связи между устройствами в сети.

На основе выбранных решений по обеспечению защиты корпоративной инфраструктуры возможно выделить четыре типа устройств в ней:

– клиенты – устройства, которыми будут пользоваться обычные пользователи сети. Это означает, что на них будет запускаться клиентская часть системы мгновенного обмена сообщениями, то есть программа-мессенджер для подключения к серверу и последующего общения с другими участниками сети, а также клиентская часть системы технологии виртуальной частной сети для формирования частной закрытой сети предприятия и получения к ней доступа;

– XMPP-сервер – устройство, обеспечивающее работу всей XMPP-сети, на котором будет запущена серверная часть системы мгновенного обмена сообщениями;

– VPN-сервер – устройство, обеспечивающее создание и поддержание VPN-инфраструктуры, на котором будет запущена серверная часть системы виртуальной частной сети;

– CA-сервер (англ. Certification Authority – Центр Сертификации) – устройство, позволяющее нам реализовать PKI (англ. Public Key Infrastructure) в нашей корпоративной сети.

На основании такой классификации была построена обобщенная структурная схема взаимодействия узлов, изображенная на рис. 4.

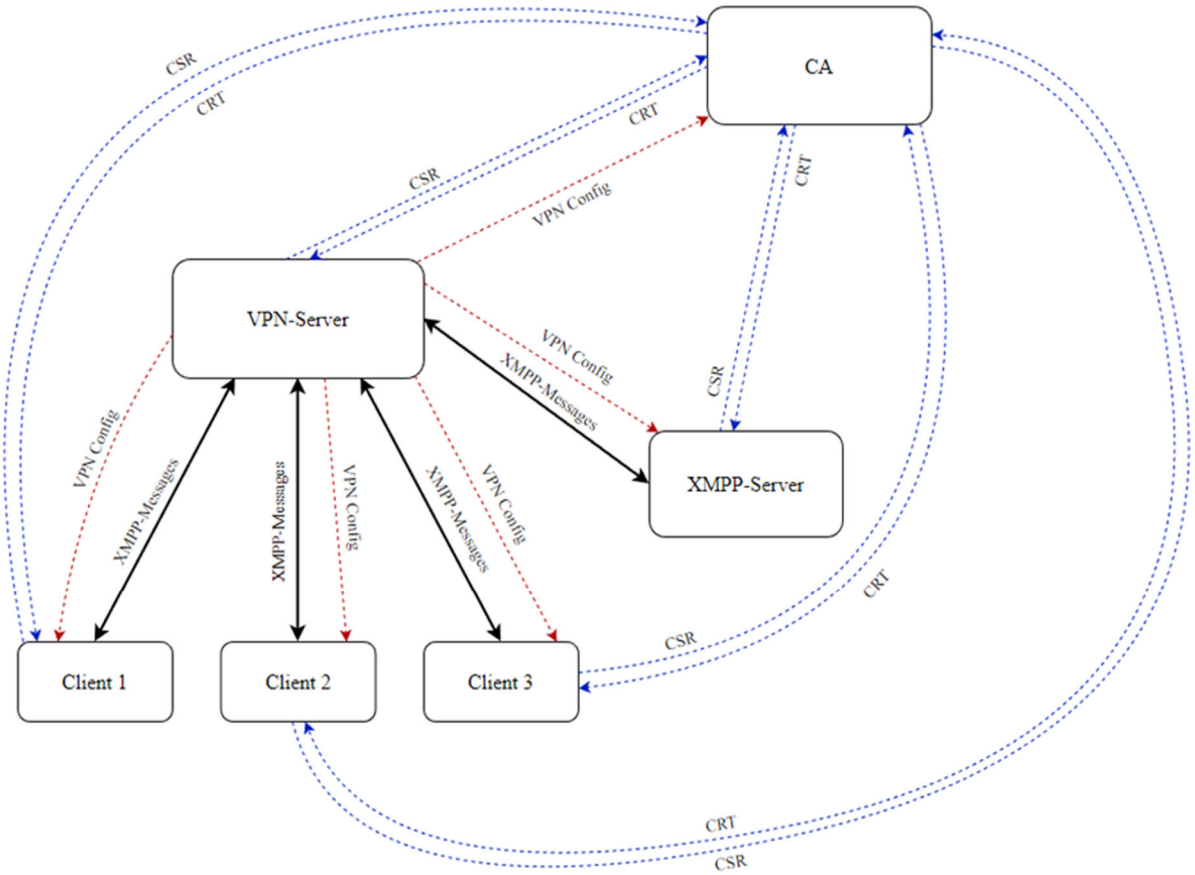


Рис. 4. Обобщенная схема взаимодействия устройств в спроектированной сети

Взаимодействие между узлами происходит в соответствии со следующим порядком:

1) узлы генерируют запросы на подпись сертификатов (сокр. *CSR*, англ. *Certificate Signing Request*), отправляют удостоверяющему центру, центр их подписывает и отправляет обратно сертификаты (сокр. англ. *CRT*) владельцам;

2) генерируются клиентские *VPN*-конфигурации для построения *VPN*-инфраструктуры и рассылаются клиентам;

3) сеть инициализирована, клиенты могут начинать общение между собой по протоколу *XMPP*.

В разработанном решении для организации спроектированной сети предлагается использовать в основе инструмента систему управления конфигурациями *Ansible* с целью осуществления инсталляции необходимого программного обеспечения и его конфигурирования в автоматизированном режиме. Подобный подход позволяет сократить время, затрачиваемое на развертывание инфраструктуры в сравнении с ручным подходом, а также гарантировать настройку сети по заданным безопасным конфигурационным файлам за счет использования декларативной модели в *Ansible*.

Алгоритм работы инструмента проиллюстрирован с помощью блок-схемы на рис. 5. Он последовательный и не имеет ветвлений. Это связано с тем, что весь

процесс заключается в последовательной настройке узлов сети. Каждый блок представлен функцией и реализован в виде *Ansible Playbook* или *Ansible Role*.

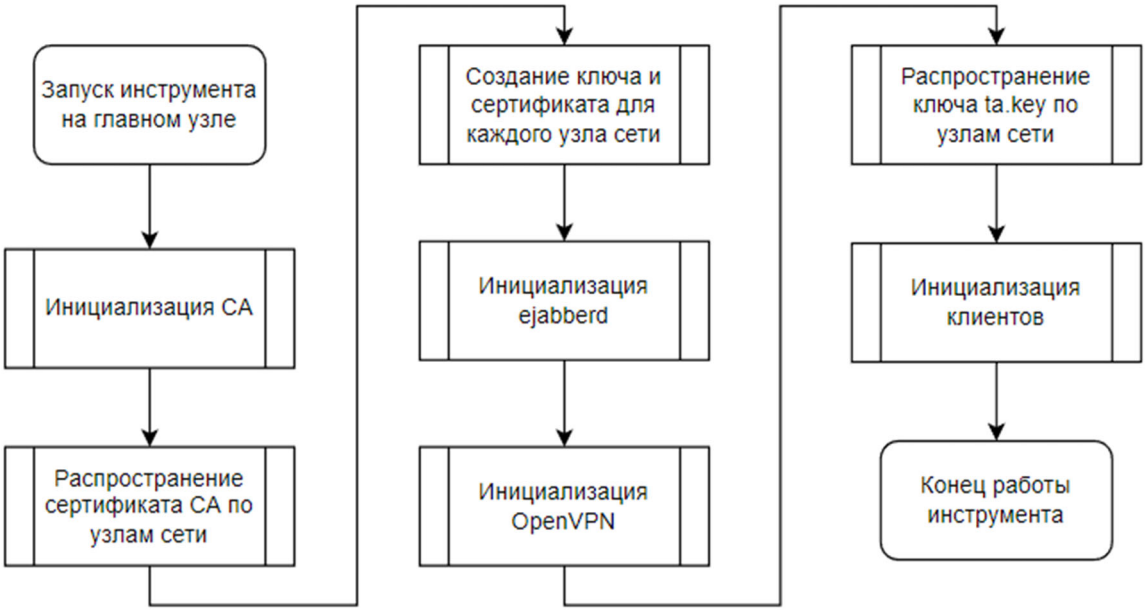


Рис. 5. Блок-схема алгоритма инструмента

В процессе работы инструмента также соблюдаются требования информационной безопасности: на все используемые директории и файлы были выставлены соответствующие правила доступа *Linux*, инструмент использует *sudo*-подход для повышения привилегий при своей работе.

### *Тестирование разработанного решения*

Комплексное тестирование разработанного программного решения включало в себя следующие тесты:

1) мануальное тестирование, представляющее собой ручную отладку продукта с целью нахождения и исправления различных ошибок, допущенных при разработке, а также проверку базовой работоспособности системы;

Состояние командной строки после успешного завершения работы инструмента представлено на рис. 6.

В результате работы инструмента на тестовом стенде была произведена развертка базовой инфраструктуры открытых ключей, *VPN*-инфраструктуры на основе технологии *OpenVPN*, инсталляция и конфигурация *XMPP*-сервера *ejabberd*, также установлены все необходимые клиентские программы для пользователей.

2) запуск автоматизированных тестов, разработанных с помощью фреймворка *Ansible Molecule*, для проверки функциональности *Ansible*-ролей (рис. 7).

*Molecule* – система тестирования для ролей и плейбуков *Ansible*, которая позволяет тестировать и проверять функциональность кода *Ansible* на различных платформах и в различных средах. *Molecule* предоставляет стандартизированный

способ тестирования кода инфраструктуры в чистой среде, изолированной от локальной машины. С помощью *Molecule* было проведено функциональное компонентное тестирование модулей инструмента.

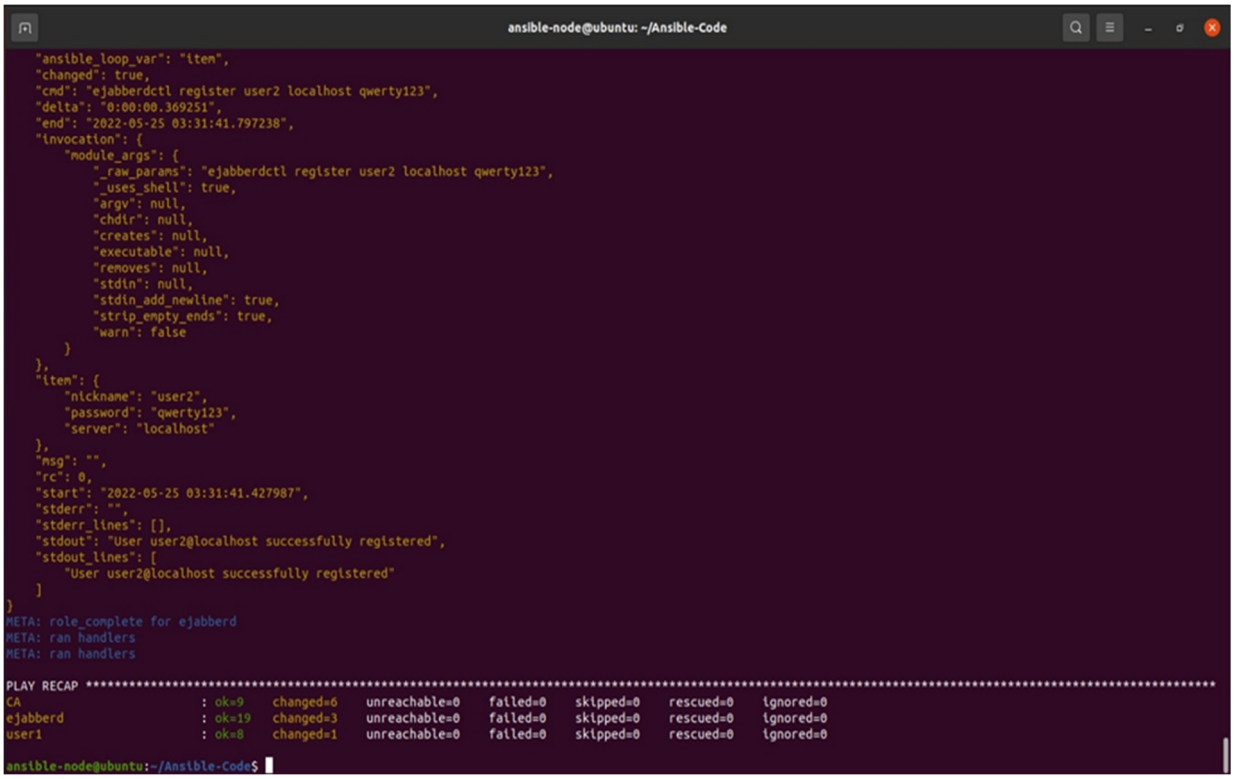


Рис. 6. Результат работы инструмента на тестовом стенде

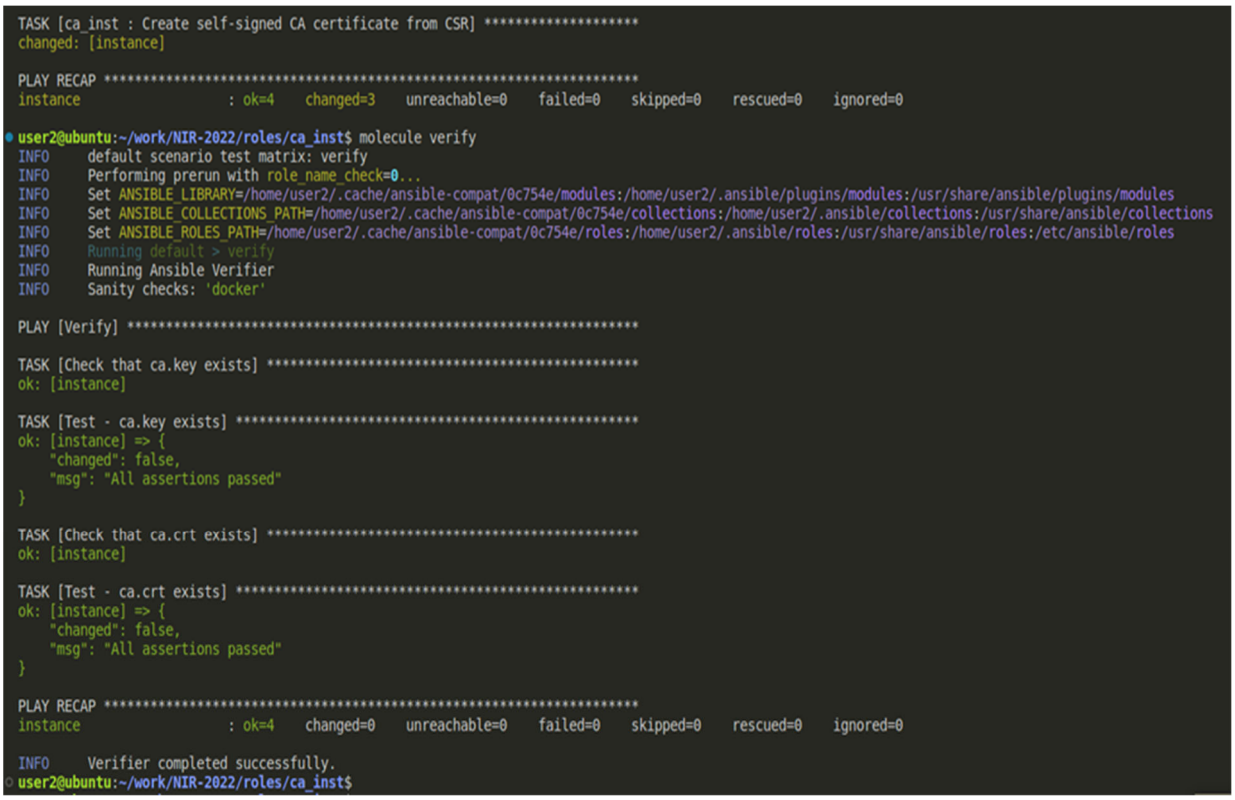


Рис. 7. Автоматизированное тестирование *Ansible*-роли

## *Заключение*

В рамках данной работы было проведено исследование потенциальных уязвимостей в серверной части системы мгновенного обмена сообщениями, работающей на базе протокола *XMPP*, а также выявлены возможные угрозы при передвижении отправленных *XMPP*-сообщений по каналам связи в вычислительной сети. На основе полученных результатов был спроектирован и разработан инструмент обеспечения безопасности информационных ресурсов при мгновенном обмене сообщениями. Полученное программное решение было успешно протестировано с помощью мануального системного тестирования и автоматизированных компонентных тестов.

Практическая значимость данной работы заключается в повышении надежности конфигурирования компьютерных систем при подготовке их к использованию для систем обмена мгновенными сообщениями, обеспечении соответствия конфигурации требованиям сетевой безопасности, ускорении процесса настройки в сравнении с ручным конфигурированием за счет использования инструмента, работающего в автоматическом режиме.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. The Radicati Group, Inc. Instant Messaging Statistics Report, 2021-2025. — URL: https://www.radicati.com/wp/wpcontent/uploads/2021/Instant_Messaging_Statistics_Report,_2021-2025_Executive_Summary.pdf (дата обращения 29.03.2023).

2. Bolton Anthony. Transforming the Workplace: Unified Communications & Collaboration Usage Patterns in a Large Automotive Manufacturer. / Bolton Anthony, Murray Meg, Fluker Joy // Hawaii International Conference on System Sciences — 2018.

3. Ying Tang. Is mobile instant messaging (MIM) useful in education? Examining its technological, pedagogical, and social affordances. / Ying Tang, Khe Foon Hew // Educational Research Review — 2018. — Vol. 21, P. 85-104.

4. Hari Setiaji. Design of Telegram Bots for Campus Information Sharing. / Hari Setiaji, Irving V Paputungan. // IOP Conference Series: Materials Science and Engineering — 2018.

5. Iraklis Symeonidis. Systematization of Threats and Requirements for Private Messaging with Untrusted Servers: The Case of e-Mailing and Instant Messaging. / Iraklis Symeonidis, Gabriele Lenzini// ICISSP — 2020.

6. Christian Johansen. The Snowden Phone: A Comparative Survey of Secure Instant Messaging Mobile Applications. / Christian Johansen, Aulon Mujaj, Hamed Arshad, Josef Noll // Security and Communication Networks — 2021. Vol. 2021, P. 30.

7. Mauli Bayu Segoro. Implementation of Two Factor Authentication (2FA) and Hybrid Encryption to Reduce the Impact of Account Theft on Android-Based Instant Messaging (IM) Applications. / Mauli Bayu Segoro, Prasetyo Adi Wibowo Putro. / International Workshop on Big Data and Information Security (IWBIS), Depok, Indonesia, 2020, P. 115-120.

8. Muhammad Nursalman. Implementation of AES and ECDSA for Encrypted Message in Instant Messaging Application. / Muhammad Nursalman, P Rizky Rachman Judie, Ammar Ashshiddiqi. // 6th International Conference on Science in Information Technology (ICSITech), Palu, Indonesia, 2020, P. 165-170.

9. History of XMPP. — URL: https://xmpp.org/about/history/ (дата обращения 07.04.2023).

10. RFC 3920. — URL: https://datatracker.ietf.org/doc/html/rfc3920 (дата обращения 07.04.2023).

11. Brian Russell, Drew Van Duren. Practical Internet of Things Security - Second Edition. — Packt Publishing, 2018. — P. 382.

12. XMPP Technology overview. — URL: https://wiki.xmpp.org/web/Technology_overview (дата обращения 11.04.2023).

13. XMPP Specifications. — URL: https://xmpp.org/extensions/ (дата обращения 11.04.2023).

14. Basinya E. A. Countermeasure method against unauthorized and anonymous information system data collection / E. A. Basinya, V. E. Khitsenko, A. A. Rudkovskiy // Dynamics of systems, mechanisms and machines (DYNAMICS) : proc., 13 intern. sci. and techn. conf., Omsk, 2019. – IEEE, 2019.

15. Basinya E. A. System of a self-organizing virtual secure communication channel based on stochastic multi-layer encryption and overlay technologies / E. A. Basinya, Z. B. Akhayeva, D. H. Omarkhanova, G. B. Tolegenova [et al.]. – Text : direct // Journal of Theoretical and Applied Information Technology. - 2022. – Vol. 100, iss. 16. – P. 4918-4927.

16. Gabriel Arquelau Pimenta Rodrigues. Securing Instant Messages with Hardware-Based Cryptography and Authentication in Browser Extension. / Gabriel Arquelau Pimenta Rodrigues; Robson De Oliveira Albuquerque; Gabriel De Oliveira Alves, Fábio Lúcio Lopes De Mendonça, William Ferreira Giozza, Rafael Timóteo De Sousa, Ana Lucila Sandoval Orozco. // IEEE Access — 2020. Vol. 8, P. 95137-95152.