ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
INFORMATION TECHNOLOGIES

УДК 004.75
https://doi.org/10.37661/1816-0301-2023-20-4-69-86

Оригинальная статья
Original Paper

Онтологический анализ в задачах моделирования угроз системам на основе контейнерных приложений

А. И. Бражук✉, Е. В. Олизарович

Гродненский государственный университет имени Янки Купалы,
ул. Ожешко, 22, Гродно, 230023, Беларусь
✉*E-mail: brazhuk@grsu.by*

Аннотация

Цели. Основной целью работы является экспериментальная верификация методики автоматического моделирования угроз на основе онтологического подхода на примере многокомпонентных контейнерных приложений, представленных в виде диаграмм потоков данных.

Методы. В работе применены методы онтологического моделирования и управления знаниями. Для представления знаний использован язык веб-онтологий, для моделирования угроз – функции автоматического логического вывода на основе дескрипционных (описательных) логик.

Результаты. Разработан машинно-читаемый набор (датасет) из 200 диаграмм потоков данных, каждая диаграмма получена из конфигурации реального контейнерного приложения и представлена в виде онтологии и графа знаний. Сформирована онтологическая двухуровневая предметно-ориентированная модель угроз контейнерных приложений. Проведен эксперимент по сравнению величины покрытия угрозами посредством общепринятого подхода и посредством предметно-ориентированных угроз для разработанного датасета. Для 95 % диаграмм предметно-ориентированная модель угроз показала величину покрытия, аналогичную или бóльшую в сравнении с общепринятым подходом.

Заключение. Результаты эксперимента доказывают пригодность и эффективность онтологического подхода для автоматического моделирования угроз. Разработанный датасет может быть использован для различных исследований в области автоматизации моделирования угроз.

Ключевые слова: компьютерные системы, контейнерные приложения, системный анализ, информационная безопасность, моделирование угроз, онтологии, автоматический логический вывод, дескрипционные (описательные) логики

Для цитирования. Бражук, А. И. Онтологический анализ в задачах моделирования угроз системам на основе контейнерных приложений / А. И. Бражук, Е. В. Олизарович // Информатика. – 2023. − Т. 20, № 4. – С. 69–86. https://doi.org/10.37661/1816-0301-2023-20-4-69-86

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 12.07.2023
Подписана в печать | Accepted 11.09.2023
Опубликована | Published 29.12.2023

Ontological analysis in the problems of container applications threat modelling

Andrei I. Brazhuk✉, Evgeny V. Olizarovich

Yanka Kupala State University of Grodno,
st. Ozheshko, 22, Grodno, 230023, Belarus
✉*E-mail: brazhuk@grsu.by*

Abstract
Objectives. The main purpose of the work is the experimental verification of the method of automatic threat modelling based on the ontological approach using the example of multicomponent container applications presented in the form of data flow diagrams.
Methods. Methods of ontological modelling and knowledge management are used in the work. The Web Ontology Language is used to represent knowledge; automatic reasoning based on description logics is used for threat modelling.
Results. A machine-readable set (dataset) of 200 data flow diagrams is developed; each diagram is obtained from the configuration of a real container application and is presented as an ontology and a knowledge graph. An ontological two-level domain-specific threat model of container applications is formed. An experiment is conducted to compare the coverage by threats using the common approach and using domain-specific threats for created dataset. For 95 % of the diagrams, the domain-specific threat model showed the coverage similar or greater than the common approach.
Conclusion. The results of the experiment prove the suitability and effectiveness of the ontological approach for automatic threat modelling. The created dataset can be used for various research in the field of automation of threat modelling.

Keywords: computer systems, container applications, system analysis, information security, threat modelling, ontologies, automatic reasoning, description logics

For citation. Brazhuk A. I., Olizarovich E. V. *Ontological analysis in the problems of container applications threat modelling*. Informatika [*Informatics*], 2023, vol. 20, no. 4, pp. 69–86 (In Russ.).
https://doi.org/10.37661/1816-0301-2023-20-4-69-86

Conflict of interest. The authors declare of no conflict of interest.

Введение. Моделирование угроз – это дисциплина, обеспечивающая анализ компьютерных систем на ранних стадиях их жизненного цикла (формирование требований, архитектурное проектирование), а также при аудите существующих систем с целью построения списков релевантных угроз, которые могут быть использованы для выбора мер и средств защиты [1, 2]. В течение долгого времени моделирование угроз рассматривалось как ручной, сложный, итеративный процесс, основанный на знаниях и взаимодействии экспертов [3]. В настоящее время ручной подход все чаще неприменим, поскольку широкое распространение получили быстрые методики разработки программного обеспечения, а также технологии и средства автоматического развертывания приложений [4], что требует существенного сокращения времени анализа защищенности, внедрения методик, работающих в условиях неполной формальной документации приложений, и автоматизации данного процесса [5].

Теоретическая постановка задачи автоматизации моделирования угроз указывает на недостатки существующих экспертных методов и предполагает широкий спектр возможных подходов, в частности машинное обучение, нейронные сети, логико-лингвистическое моделирование [6], системы нечеткого вывода [7], элементы науки о данных [8]. В этом направлении также активно исследуются технологии обработки естественных языков, трансформеры [9], цифровые двойники [10], а также интеллектуальный анализ текста и когнитивное моделирование [11].

Следует отметить, что практические вопросы моделирования угроз исследованы недостаточно [11]. Для представления различных артефактов при прикладном моделировании угроз используются различные нотации и графические форматы, такие как контрольные списки, диа-

граммы процессов, а также диаграммы потоков данных (Data Flow Diagram, DFD). Неформальные методики на основе популярных нотаций имеют целый спектр проблем, связанных с взаимодействием экспертов и разработчиков в процессе проектирования систем [12], а также определением оптимальной структуры системы защиты [13].

Настоящая работа посвящена практическим аспектам автоматизации моделирования угроз на основе диаграмм потоков данных посредством онтологий с использованием предметно-ориентированных моделей угроз. Переход к автоматизации требует исследований ее эффективности в сравнении с существующими подходами, а также тестовых наборов данных для подобных оценок. Для решения этих задач в работе исследуется авторская методика автоматического моделирования угроз на примере многокомпонентных контейнерных приложений.

1. Онтологическое моделирование угроз компьютерных систем. Диаграммы потоков данных являются наиболее известным способом представления структуры системы как набора процессов, хранилищ данных и внешних сущностей, связанных различными направленными потоками передачи информации [14, 15]. Этот подход хорошо отражает основную тенденцию в информационной безопасности, согласно которой большинство угроз реализуются посредством каналов взаимодействия, а также показывает важность защиты передаваемых данных.

Диаграмма потоков данных представляет собой веб-приложение, состоящее из двух программных контейнеров Docker (рис. 1). В данном варианте нотации окружности соответствуют процессам (process0, process1), прямоугольник – внешней сущности (user), а фигуры, состоящие из двух параллельных линий, – хранилищам данным (hostStorage, storage0).

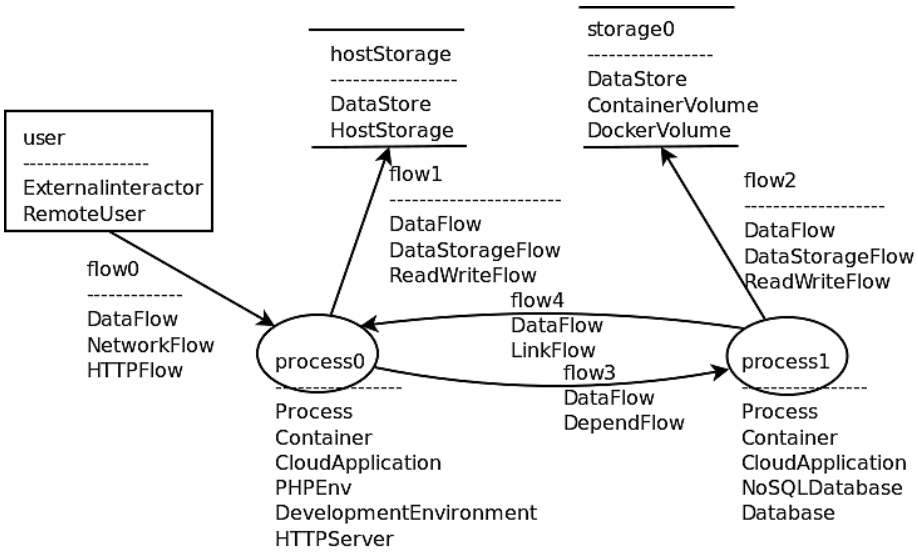

Рис. 1. Пример диаграммы потоков данных

Fig. 1. An example of Data Flow Diagram

Особенностью диаграммы, изображенной на рис. 1, является то, что она содержит предметно-ориентированные знания, т. е. информацию о функциях и реализациях компонентов и потоков. Так, flow0 представляет собой сетевой поток (NetworkFlow), в частности поток HTTP (HTTPFlow). Процесс process0 является средой исполнения (DevelopmentEnvironment) и включает интерпретатор языка программирования PHP (PHPEnv). В связи с тем что process0 является назначением потока HTTP, он может быть распознан как реализация сервера HTTP (HTTPServer).

С точки зрения моделирования угроз предметно-ориентированные знания позволяют рассматривать более конкретные угрозы компонентам или потокам. Так, с сетевым потоком можно ассоциировать некий высокоуровневый набор сетевых угроз, но уточнение о том, что поток ис-

пользует протокол HTTP, позволяет рассматривать угрозы, характерные для данного прикладного протокола, а также учитывать особенности организации обработки сообщений этого протокола клиентом (user) и сервером HTTP (process0).

Методика онтологического моделирования угроз. Для автоматизации моделирования угроз использована авторская методика, основанная на онтологическом подходе [16]. Методика позволяет, во-первых, внедрять технологии объектно-ориентированного проектирования в процесс моделирования угроз на основе концептов, объектных свойств и экземпляров, а во-вторых, автоматизировать анализ защищенности компьютерных систем на основе онтологий.

Онтологическое моделирование и графы знаний [17, 18] как средства построения интеллектуальных компьютерных систем [19, 20] активно используются для решения различных задач [21]. В частности, рассматриваемая методика реализует подход к прикладным онтологиям, наиболее близкий к сценарию семантического анализа [22], который подразумевает применение к знаниям функций автоматического логического вывода и семантической обработки, что позволяет создавать и использовать «новые» знания [23]. Отличительным признаком подхода онтологического моделирования, реализованного в методике, является использование экземпляров наравне с концептами не только в конечных моделях, но и в метамоделях.

Схема методики онтологического моделирования угроз показана на рис. 2. Основная идея методики заключается в том, что разработчик (архитектор) компьютерной системы описывает ее в виде диаграммы потоков данных. Программная система моделирования угроз способна семантически интерпретировать диаграмму (т. е. представить ее) в виде онтологии, а затем применить к данной семантической интерпретации процедуры автоматического логического вывода для нахождения списка релевантных угроз, соответствующих описанию системы.

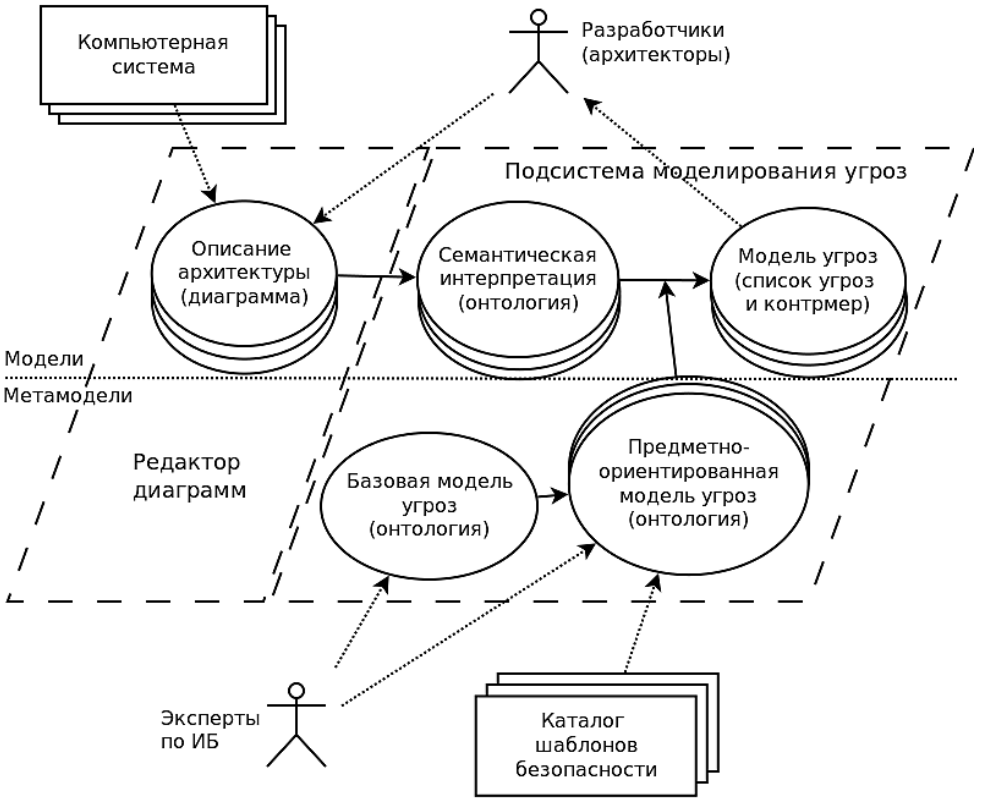

Рис. 2. Онтологическое моделирование угроз

Fig. 2. Ontological threat modelling

Для обеспечения автоматического логического вывода угроз в методике использованы два типа метамоделей (онтологий) в комбинации их с семантической интерпретацией диаграммы [16]:

– базовая онтологическая модель угроз, которая содержит концепты, объектные свойства и экземпляры, представляющие компоненты диаграмм, информационные потоки и угрозы;

– предметно-ориентированные модели угроз, каждая из которых описывает аспекты безопасности определенного типа компьютерных систем (облачных систем, систем интернета вещей, контейнерных систем и т. п.). Для построения предметно-ориентированной модели угроз необходимо, используя знания экспертов и различные источники информации, расширить базовую модель путем наследования ее концептов и создания необходимых подклассов архитектурных компонентов и экземпляров угроз, относящихся к данной предметной области.

Семантические интерпретации диаграмм (рис. 2), представленные как онтологии, а также описания систем и модели (перечни) угроз являются конечными моделями.

Метамодели и модели реализованы на языке веб-онтологий OWL (Web Ontology Language) [24, 25]. Язык OWL основан на дескрипционных логиках DL (Description Logics) [26, 27], используемых для управления знаниями; для OWL созданы редакторы онтологий (Protege), системы автоматического логического вывода (Fact++, Hermit, Pellet) и средства разработки (Java OWL API). Формат RDF (Resource Description Framework) может быть использован для представления знаний в виде графа знаний, состоящего из триплетов «объект – свойство – субъект».

Основная информация о методике и исходные файлы онтологий опубликованы в репозитории Github (URL: https://github.com/nets4geeks/OdTM).

Семантическая интерпретация диаграмм. Опишем компоненты диаграммы следующим образом: процессы как концепты Process, внешние сущности – ExternalInteractor, хранилища данных – DataStore. Взаимодействия компонентов описываются потоками данных (DataFlow) между компонентами. Потоки данных описываются через их источники и назначения, т. е. для некоторого потока можно утверждать, что он имеет источником (hasSource) и назначением (hasTarget) некоторый компонент. На рис. 3 показан пример простой диаграммы. Для формализации подобного примера может быть использован редактор онтологий Protege, при этом далее в тексте применяется смешанная терминология языка OWL и редактора Protege.

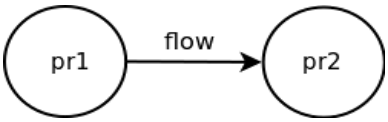

Рис. 3. Простая диаграмма потоков данных
Fig. 3. A simple data flow diagram

Используя концепты и объектные свойства, упомянутые выше, эту диаграмму можно описать на языке OWL набором утверждений, показанным в листинге 1.

Листинг 1.

```
Process (pr1)
Process (pr2)
DataFlow (flow)
hasSource (flow, pr1)
hasTarget (flow, pr2)
```

В листинге 1 отражается то, что считается исходными, явными (explicit) знаниями. Однако свойства базовой модели угроз и функции автоматического логического вывода позволяют получить дополнительные знания, называемые неявными (implicit). Например, модель содержит свойства «является источником» (isSourceOf) и «является назначением» (isTargetOf), которые описаны как обратные к свойствам hasSource и hasTarget соответственно. Это позволяет прави-

лам логического вывода автоматически установить, что процесс pr1 служит источником потока flow, а pr2 – его назначением (листинг 2).

Листинг 2.
isSourceOf (pr1, flow)
isTargetOf (pr2, flow)

Автоматический логический вывод угроз. Разработанная методика моделирования угроз предполагает использование автоматического логического вывода с включением предметно-ориентированных знаний для нахождения в онтологическом представлении диаграмм некоторых структурных шаблонов, которые могут быть ассоциированы с набором потенциальных угроз. В данной работе такие формальные конструкции называются семантическими шаблонами.

Например, пусть сервисы, которые обслуживают клиентов по протоколу HTTP, считаются небезопасными (так как HTTP передает данные в открытом виде). Семантический шаблон для этой угрозы неформально будет звучать как «процесс, который является назначением HTTP-потока». Формальное определение такого шаблона посредством утверждения OWL (эквивалентность – Equivalent to) будет иметь вид, показанный в листинге 3 (в листингах 3 и 4 используется нотация, применяемая для описания утверждений в Protege).

Листинг 3.
Process and (isTargetOf some HTTPFlow)

Если вернуться к диаграмме на рис. 3 и семантической интерпретации в листинге 1, то для соответствия процесса pr2 данному шаблону необходимо, чтобы поток flow был экземпляром концепта HTTPFlow. Это знание может быть передано системе явно или же следовать из соответствующей предметно-ориентированной модели угроз.

В листинге 4 показано выражение OWL (подкласс Subclass), которое позволяет сопоставить шаблон, приведенный в листинге 3, соответствующему экземпляру угрозы (insecureProcessThreat).

Листинг 4.
isAffectedBy value insecureProcessThreat

Описание определяемого класса (Defined Class), включающего два последних утверждения (листинги 3, 4), позволит посредством автоматического логического вывода обнаружить все процессы, которые небезопасно используют протокол HTTP, и сопоставить им экземпляр угрозы insecureProcessThreat.

Путем применения определяемых классов и необходимых экземпляров в разработанной базовой модели угроз реализован общепринятый подход к моделированию угроз STRIDE [1], который заключается в анализе каждого элемента диаграммы на возможность осуществления злоумышленником следующего набора действий: спуфинг (Spoofing), незаконное изменение (Tampering), отказуемость (Repudiation), раскрытие информации (Information Disclosure), отказ в обслуживании (Denial of Service), повышение привилегий (Elevation of Privilege). При этом считается [1], что процессы могут быть подвержены всем вышеперечисленным угрозам: хранилища данных – незаконному изменению, отказуемости, раскрытию информации и отказу в обслуживании; внешние сущности – спуфингу и отказу в обслуживании; потоки данных – раскрытию информации, незаконному изменению и отказу в обслуживании.

В табл. 1 показано, как подход STRIDE автоматизирован в базовой онтологической модели угроз. Столбцы ID и «Наименование» содержат идентификаторы и названия соответствующих угроз. В столбце «Семантический шаблон» перечислены элементы диаграммы, к которым могут быть применены угрозы.

Таблица 1
Подход STRIDE в базовой онтологической модели угроз
Table 1
The STRIDE approach in the Basic ontological threat model

ID	Наименование угрозы *Threat name*	Семантический шаблон *Semantic template*
Spoofing	Подмена (спуфинг)	Process, ExternalInteractor
Tampering	Незаконное изменение	Process, DataStore, DataFlow
Repudiation	Отказуемость	Process, DataStore
InformationDisclosure	Раскрытие информации	Process, DataStore, DataFlow
Denial of Service	Отказ в обслуживании	Process, DataStore, ExternalInteractor, DataFlow
Elevation of Privilege	Повышение привилегий	Process

Следует отметить, что для процессов, хранилищ данных и внешних сущностей ассоциация угрозы возможна при участии в каком-либо потоке. Семантические шаблоны подхода STRIDE реализованы в виде соответствующих определяемых классов.

2. Набор (датасет) семантических диаграмм контейнерных приложений. Одним из факторов, ограничивающих исследования по автоматизации моделирования угроз, является недостаток наборов данных (формальных описаний систем в виде диаграмм потоков данных), которые могли бы использоваться для оценки эффективности и корректности различных методик моделирования угроз [28]. Современные исследования оперируют всего лишь десятками диаграмм [28, 29].

В рамках настоящей работы для количественной оценки характеристик онтологического моделирования угроз был создан открытый набор (датасет) из 200 семантических диаграмм потоков данных на основе конфигураций реальных многоконтейнерных приложений. Диаграммы являются семантическими, потому что каждая из них представлена в машинно-читаемом виде как онтология OWL и граф знаний RDF.

В работе были использованы автоматические конфигурации Docker compose, которые являются декларативными описаниями в одном файле (docker-compose.yml) контейнерных приложений [30, 31], состоящих из нескольких сервисов (контейнеров). Соответствующие файлы были получены из публичных репозиториев (Github.com, Gitlab.com), а также из ряда корпоративных репозиториев. Для обеспечения приватности исходных данных созданный набор включает только «деперсонифицированные» артефакты – описания диаграмм потоков данных в виде онтологий и графов знаний. Также диаграммы представлены в формате YAML (YAML Ain't Markup Language),что может быть использовано для создания их графических представлений. Процесс создания диаграммы из файла docker-compose.yml показан на рис. 4.

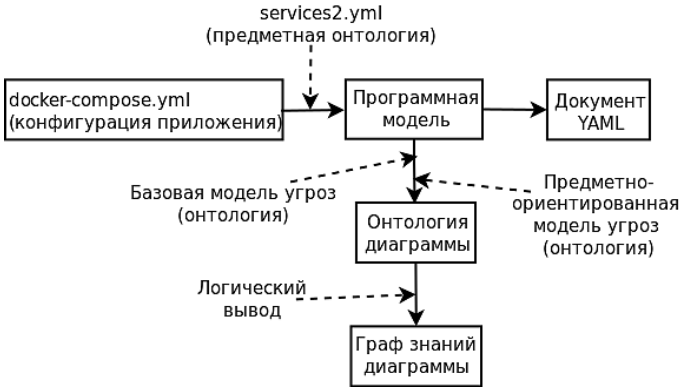

Рис. 4. Создание семантической диаграммы
Fig. 4. Creating a semantic diagram

Могут быть выделены следующие стадии данного процесса:

1. *Создание программной модели из файла docker-compose.yml*. Включает чтение исходного файла; создание экземпляров программных классов для сущностей, описанных в этом файле, а также классификацию сущностей в соответствии с предметной онтологией. Ниже показан пример файла docker-compose.yml.

Листинг 5.

```
services:
  web:
    image: php:8.0
    volumes:
      - ./app:/var/www/html
    depends_on:
      - mongodb
    ports:
      - 80:80
  mongodb:
    image: mongo:latest
    volumes:
      - dbdata:/data/db
    links:
      - web
```

В листинге 5 описаны два сервиса (web и mongodb), каждый из которых реализован как контейнер. Свойство image указывает на базовый образ контейнера (набор ПО, необходимый для работы сервиса): для первого контейнера это среда разработки PHP, для второго – нереляционная СУБД MongoDB. Описание каждого контейнера также содержит перечни внешних хранилищ (раздел volumes), опубликованных сетевых портов (ports) и взаимосвязи с другими контейнерами (depends_on и links). В системе Docker внешние данные могут быть сохранены как папки на хосте или в томе, обеспечиваемом контейнерной системой.

Программная модель подразумевает следующие трансформации сущностей для создания диаграмм:

– контейнеры считаются процессами (Process);

– внешние данные моделируются как хранилища данных (DataStore), причем папки на хосте принадлежат классу HostStorage, а тома – DockerVolume;

– для моделирования сетевых взаимодействий приложения с внешней средой создается сущность внешнего пользователя (RemoteUser).

Между сущностями создаются следующие потоки (DataFlow):

между контейнерами и хранилищами – потоки хранилищ данных (DataStorageFlow);

между удаленным пользователем и контейнерами (при наличии открытых портов) – сетевые потоки (NetworFlow);

между контейнерами (при наличии свойств depends_on и links) – потоки зависимости (DependFlow) и связи (LinkFlow).

На данном этапе ключевое значение имеет предметная онтология, которая используется для ассоциации сущностей с дополнительными классами и содержит следующие данные:

– категории используемых сервисов (image); например, mongodb – это нереляционная СУБД (NoSQLDatabase) и СУБД в общем (Database);

– типы используемых хранилищ; например, для mongodb данные хранятся в папке /data/db;

– типы опубликованных сервисов; например, если контейнер публикует порт 80, то он может быть распознан как сервер HTTP (HTTPServer), а поток к нему – как HTTP-поток (HTTPFlow).

Создание предметной онтологии является итеративным ручным процессом по мере добавления новых конфигураций. Предметная онтология для данного набора сохранена в формате YAML.

В листинге 6 представлен фрагмент предметной онтологии: поле images позволяет отнести контейнер к данной категории по базовому образу, а поля name и labels являются дополнительными классами, ассоциированными с процессом, представляющим контейнер на диаграмме.

Листинг 6.

```
 - name: NoSQLDatabase
   images:
     - mongodb
     - mongo
     - influxdb
     - zookeeper
     - couchdb
   labels:
     - Database
```

Концепты, описанные в предметной онтологии, могут быть использованы для формирования семантических шаблонов различных предметно-ориентированных моделей угроз.

Показанное ранее на рис. 1 графическое представление диаграммы получено из конфигурации Docker compose, приведенной в листинге 5, на основе предметной онтологии.

2. *Создание онтологии диаграммы*. Для этого необходимо сформировать новую онтологию и импортировать базовую онтологическую модель, затем добавить необходимые утверждения OWL относительно диаграммы, в частности:

– экземпляры процессов, хранилищ, а также экземпляр удаленного пользователя;

– ассоциации экземпляров и их концептов с использованием предметной онтологии;

– определения потоков через объектные свойства hasSource и hasTarget;

– ряд ассоциаций с концептами базовой модели угроз и предметно-ориентированных моделей угроз.

3. *Создание графа знаний диаграммы*. Онтология диаграммы, созданная на предыдущем этапе, содержит неполные (явные) знания о диаграмме. Два шага должны быть сделаны, чтобы расширить знания: во-первых, импортирована предметно-ориентированная модель угроз (см. разд. 3), во-вторых, выполнена процедура автоматического вывода для заполнения базы знаний дополнительными (неявными) знаниями. Заполненная онтология может быть сохранена в формате RDF, что позволяет использовать язык запросов SPARQL для получения необходимых фактов.

В рамках исследования был разработан набор утилит на языке программирования Java с помощью библиотеки OWL API, который использовался для обработки 200 файлов docker-compose.yml. В результате был получен датасет семантических диаграмм потоков данных, который опубликован в открытом доступе как репозиторий Github (URL: https://github.com/nets4geeks/DockerComposeDataset) в папке clear2, предметная онтология опубликована в файле services2.yml.

3. Построение предметно-ориентированных моделей для анализа защищенности. Для целей эксперимента была сформирована предметно-ориентированная модель угроз, которая включает две подмодели: общие угрозы облачных приложений и угрозы, специфичные для контейнерных приложений.

3.1. Модель угроз облачных компьютерных систем. В качестве перечня общих угроз облачным системам был использован авторский каталог шаблонов угроз облачных систем ACCTP (Academic Cloud Computing Threat Patterns), разработанный ранее [32]. Каталог агрегирует угрозы облачным системам, перечисленные в некоторых известных каталогах безопасности и научной литературе. Он реализован как онтология OWL и может быть автоматически конвертирован в соответствующую предметно-ориентированную модель угроз, в которой семантические шаблоны представлены в виде определяемых классов. Текстовая версия каталога доступна по ссылке URL: https://nets4geeks.github.io/acctp/catalog/.

Каталог ACCTP включает несколько профилей (архитектурный, операционный, инфраструктура как услуга) и ряд сценариев в рамках этих профилей [32]. Для эксперимента в настоящей работе использованы сценарии «Простое облачное приложение» и «Взаимодействие облачных приложений» архитектурного профиля ACCTP. В первом сценарии рассматривается взаимодействие удаленных пользователей с облачным приложением: в разрезе защищенности как пользователи могут влиять на облачное приложение, так и облачное приложение может влиять на пользователей. Во втором сценарии рассматривается взаимодействие облачных приложений друг с другом: в случае многокомпонентных приложений каждый компонент считается облачным приложением (например, веб-сервер взаимодействует с СУБД). Табл. 2 содержит перечень угроз, сформированный согласно этим сценариям. Столбцы ID и «Наименование угрозы» описывают идентификаторы и имена шаблонов угроз. Столбец «Семантический шаблон» содержит условное описание семантических шаблонов в виде потоков между концептами. В данном случае потоки являются направленными: первый концепт – клиент (источник), второй концепт – сервер (назначение), столбец «Цель» определяет цель угрозы (клиент или сервер).

Таблица 2
Шаблоны угроз облачным приложениям

Table 2
Threat patterns of cloud applications

ID	Наименование угрозы *Threat name*	Семантический шаблон *Semantic template*	Цель *Target*
AB01	Сбой облачного приложения	RemoteUser > CloudApplication CloudApplication > CloudApplication	Клиент
AB02	Потеря соединения с облачным приложением		
AC01	Вредоносный контент от облачного приложения	RemoteUser > CloudApplication	Клиент
AC02	Доступ к облачному приложению через незащищенную сеть		
AC04	Социальная инженерия против пользователя		
AD01	Ошибки аутентификации	RemoteUser > CloudApplication CloudApplication > CloudApplication	Сервер
AD02	Ошибки контроля доступа		
AD03	Утечка данных облачного приложения		
AD04	Потеря данных облачным приложением		
AD05	Потеря резервной копии облачного приложения		
AD06	Утечка резервной копии облачного приложения		
AD07	Потеря журнала событий облачного приложения		
AD08	Утечка журнала событий облачного приложения		
AE01	Отказ в обслуживании (DoS/DDoS)	RemoteUser > CloudApplication	Сервер
AE02	Экономический отказ в обслуживании (EDoS)		
AE03	Сетевые атаки на облачное приложение		
AE04	Неправомерное использование облачного приложения		
AE05	Блокировка разделяемых ресурсов облачного приложения		
AE06	Утечка учетных данных интерфейса администрирования		

Применение указанных шаблонов в эксперименте с контейнерными приложениями предполагает использование концепта удаленного пользователя (RemoteUser), а также трактовку всех процессов как облачных приложений (CloudApplication). Так как предметно-ориентированная модель угроз АССТР позволяет определять угрозы посредством автоматического логического вывода, то для ее использования достаточно при создании онтологии диаграммы ассоциировать экземпляры с соответствующими концептами (RemoteUser, CloudApplication) и импортировать онтологию модели угроз в онтологию диаграммы.

Ниже приведен пример реализации шаблона, который в онтологии OWL является определяемым классом, что позволяет автоматически классифицировать соответствующие экземпляры (листинг 7, выражение эквивалентности).

Листинг 7.

```
(CloudApplication) and (isTargetOf some (hasSource some RemoteUser))
```

Экземпляры угроз ассоциируются с компонентами диаграмм посредством специальных выражений, описывающих подклассы. Пример такой конструкции на языке OWL показан в листинге 8.

Листинг 8.

```
SubClassOf(:CloudApplicationAndIsTargetOfHasSourceRemoteUser ObjectHasValue
(<http://www.grsu.by/net/OdTMBaseThreatModel#isAffectedBy>
<http://www.grsu.by/net/ACCTP#threatAD01_BrokenAuthentication>))
```

3.2. Модель угроз контейнерных приложений. В настоящее время существует ряд неформальных моделей угроз и руководств по безопасности контейнеров, например методический документ ФСТЭК (URL: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/trebovaniya-po-bezopasnosti-informatsii-utverzhdeny-prikazom-fstek-rossii-ot-4-iyulya-2022-g-n-118) и соответствующая база данных угроз (URL: https://bdu.fstec.ru), ряд корпоративных и общественных матриц угроз, например Microsoft (URL: https://www.microsoft.com/en-us/security/blog/2021/03/23/secure-containerized-environments-with-updated-threat-matrix-for-kubernetes/) и ATT&CK (URL: https://attack.mitre.org/matrices/enterprise/containers/). На основе этих источников в рамках настоящей работы был создан дополнительный профиль шаблонов угроз многокомпонентных контейнерных приложений для каталога АССТР. Профиль ориентирован, во-первых, на угрозы, возникающие при взаимодействии компонентов, во-вторых, на угрозы, которые не перекрывают угрозы архитектурного профиля.

Для описания соответствующих онтологических шаблонов в модель были добавлены следующие концепты:

контейнер (Container) – каждый процесс диаграммы;

хранилище хоста (HostStorage) – способ хранения данных контейнера на файловой системе хоста (устаревший подход);

том контейнера (ContainerVolume) – способ хранения данных контейнера посредством абстракций, создаваемых системой управления контейнерами;

сокет контейнера (ContainerSocket) – канал для чтения (записи) команд для контейнерной системы.

Кроме того, в профиле применяется концепт удаленного пользователя (RemoteUser).

Табл. 3 содержит созданный перечень шаблонов угроз контейнерных приложений. Все цели в табл. 3 имеют роль сервера, шаблоны EC05-EC07 содержат угрозы для потоков. В связи с тем что профиль является частью АССТР, подход к реализации и использованию шаблонов такой же, как описан в разд. 3.1.

Таблица 3
Шаблоны угроз контейнерных приложений
Table 3
Threat patterns of container applications

ID	Наименование угрозы *Threat name*	Семантический шаблон *Semantic template*
EA01	Использование хранилища хоста	Container > HostStorage
EA02	Использование сокета системы управления	Container > ContainerSocket
EB01	Доступ к тому в режиме записи	Container > HostStorage Container > ContainerVolume
EB02	Утечка данных с тома	
EB03	Повреждение данных, хранимых томом	
EC01	Использование средств управления контейнерами (CLI, API)	Container > Container RemoteUser > Container
EC02	Выход за границы контейнера (escape)	
EC03	Повышение привилегий контейнера	
EC04	Скомпрометированные токены доступа	
EC05	Утечка данных при передаче	Container > Container (flow) RemoteUser > Container (flow)
EC06	Повреждение данных при передаче	
EC07	Недоступность канала взаимодействия	
ED01	Доступ к интерфейсу управления контейнера (WEB)	RemoteUser > Container
ED02	Уязвимости опубликованных приложений	
ED03	Развертывание контейнера с вредоносным ПО	
ED04	Обнаружение контейнеров	

4. Экспериментальная верификация методики онтологического моделирования угроз. Важной характеристикой результатов моделирования угроз является количество потенциальных угроз, сопоставленных с элементами диаграмм [28]. В рамках проведенного эксперимента было выполнено сравнение количества угроз, полученных посредством общепринятого подхода STRIDE (табл. 1), с количеством угроз, полученных посредством предметно-ориентированных моделей шаблонов угроз (табл. 2 и 3), для семантических диаграмм из созданного датасета.

В эксперименте использованы базовая модель угроз, которая содержит шаблоны угроз STRIDE (табл. 1), и предметно-ориентированная модель угроз, которая включает архитектурную подмодель (профиль) шаблонов угроз облачных систем (табл. 2) и профиль шаблонов угроз контейнерных приложений (табл. 3). Для каждой семантической диаграммы датасета с помощью автоматического логического вывода и запросов SPARQL сформированы списки релевантных угроз, затем скриптами командного интерпретатора подсчитано количество предметно-ориентированных угроз (ПОУ) и количество угроз STRIDE. Далее для каждой диаграммы рассчитано значение показателя эффективности путем деления количества полученных ПОУ на количество угроз STRIDE.

Полученная в результате выборка показателей эффективности для 200 объектов датасета (рис. 5) подтверждает, что в 95 % случаев набор угроз, сформированный с использованием рассматриваемой методики автоматического моделирования угроз на основе онтологического подхода, равен или больше количества угроз, полученных на основе модели STRIDE. Среднее зна-

чение показателя эффективности равно 159 %. Результаты эксперимента показывают пригодность и эффективность предложенной методики для использования в задачах автоматизации моделирования угроз.

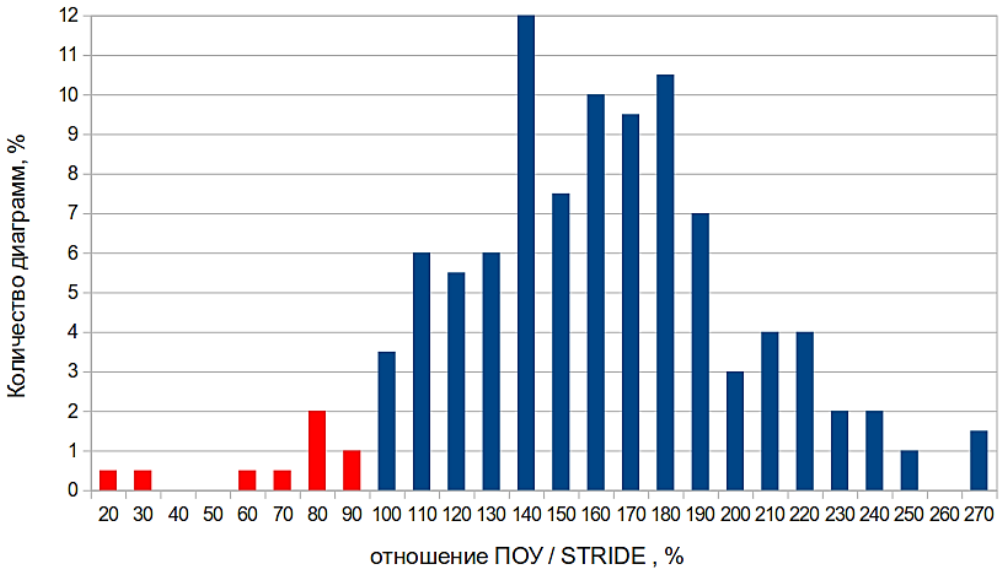

Рис. 5. Результаты эксперимента по анализу автоматического покрытия угрозами

Fig. 5. Experiment results of automatic threat covering

На рис. 5 изображена гистограмма распределения результатов эксперимента после их обработки. Ось X отображает отношения ПОУ к STRIDE в процентах (значения отношений предварительно округлены до одного знака после запятой), ось Y – количество диаграмм от общего числа диаграмм в процентах, соответствующих данным значениям отношения ПОУ/STRIDE.

Как преимущество предлагаемой методики следует рассматривать также то, что ПОУ, подобные использованным в данной работе, сразу ориентированы на анализ конкретных проблем защищенности, а подход STRIDE дает общие угрозы, которые сначала должны быть конкретизированы для данного приложения (т. е. для каждой категории STRIDE синтезированы новые частные угрозы) и только потом оценивается их применимость. Таким образом, использование ПОУ сокращает время анализа защищенности. Однако для получения численных оценок данного преимущества требуется проведение дополнительных исследований на массированных наборах данных с привлечением экспертного сообщества, в частности качественной оценки ПОУ в сравнении с угрозами STRIDE, что выходит за рамки данной работы.

Наличие 5 % случаев, в которых ПОУ меньше, чем угроз STRIDE, не может рассматриваться как недостаток методики, поскольку ПОУ могут применяться и детализироваться в соответствии с целями моделирования. Например, в данном эксперименте фактически были использованы два уровня абстракции: общие модели угроз облачных систем (архитектурный профиль ACCTP) и угрозы многокомпонентных приложений Docker (профиль контейнеров ACCTP). Очевидно, что добавление новых профилей (например, операционного профиля или профиля персональных данных) увеличило бы долю предметно-ориентированных моделей относительно STRIDE. В то же время, если цель моделирования угроз точно сформулирована, например защита персональных данных, то избыточные профили могут быть исключены, чтобы уменьшить количество угроз для рассмотрения.

Заключение. Данная работа посвящена экспериментальной верификации разработанной методики онтологического моделирования угроз на основе предметно-ориентированных моделей на примере контейнерных приложений. Для обеспечения эксперимента был разработан датасет из 200 семантических диаграмм на основе реальных облачных приложений, который может быть использован для различных исследований в области автоматизации моделирования угроз.

Каждая диаграмма была описана в терминах предметной онтологии и представлена в виде онтологии и графа знаний. Исходными данными набора являются реальные облачные многокомпонентные приложения (конфигурационные файлы Docker compose). Набор отличается от аналогов большим количеством элементов (сотни против десятков) и машинно-читаемым представлением.

Сформирована онтологическая предметно-ориентированная модель угроз, включающая соответствующие шаблоны угроз, что обеспечивает автоматический логический вывод угроз по онтологическим представлениям диаграмм. Предложенная модель угроз состоит из двух уровней: общие угрозы облачных приложений и угрозы, специфичные для контейнерных приложений.

Для разработанного датасета выполнен расчет показателей эффективности предложенной методики автоматического моделирования угроз и экспериментально показана ее адекватность, в частности то, что применение онтологической предметно-ориентированной модели угроз позволяет успешно выявлять угрозы, которые основаны на взаимодействии компонентов приложения. Для использованного датасета количество ПОУ для 95 % диаграмм в среднем на 59 % больше количества угроз, полученных посредством общепринятого подхода STRIDE.

Выявление новых критериев и методик оценки эффективности автоматизации анализа угроз является перспективным направлением научных исследований, требующих привлечения экспертного сообщества и представителей индустрии. Также научно-практический интерес представляет создание классификаций датасета, его расширение и сегментирование; разработка новых наборов данных для других типов приложений и их использование для исследований в области информационной безопасности, в том числе с применением методов машинного обучения.

Практическая значимость полученных результатов заключается в возможности их применения для автоматизации анализа защищенности компьютерных систем. В частности, предметно-ориентированные модели угроз могут быть научной основой для разработки средств автоматизации анализа состава, структуры и информационных потоков информационных систем в рамках процесса создания и аттестации систем защиты информационных систем.

Вклад авторов. *А. И. Бражук* сформировал датасет и необходимые онтологические модели, разработал программное обеспечение и выполнил расчет результатов эксперимента. *Е. В. Ализарович* участвовал в анализе и интерпретации эксперимента, осуществлял научное редактирование статьи.

Список использованных источников

1. Shostack, A. Experiences threat modeling at Microsoft / A. Shostack // MODSEC@ MoDELS. – 2008. – Vol. 2008. – 35 p.

2. A survey on threat-modeling techniques: protected objects and classification of threats / A. Konev [et al.] // Symmetry. – 2022. – Vol. 14, no. 3. – P. 549.

3. Макаревич, В. А. Анализ и моделирование угроз информационной безопасности предприятия на основе универсального шаблона / В. А. Макаревич, Е. А. Минюкович, К. С. Мулярчик // Журнал Белорусского государственного университета. Экономика. – 2021. – № 1. – С. 57–68.

4. Кочин, В. П. Проектирование и обеспечение безопасности интегрированных образовательных информационно-коммуникационных систем / В. П. Кочин, Ю. И. Воротницкий. – Минск : БГУ, 2022. – 168 с.

5. Verreydt, S. Expressive and systematic risk assessments with instance-centric threat models / S. Verreydt, D. Van Landuyt, W. Joosen // SAC'23: Proceedings of the 38th ACM/SIGAPP Symp. on Applied Computing, Tallinn, Estonia, 27–31 Mar. 2023. – Tallinn, 2023. – P. 1450–1457.

6. Язов, Ю. К. Логико-лингвистическое моделирование угроз безопасности информации в информационных системах / Ю. К. Язов, С. В. Соловьев, М. А. Тарелкин // Вопросы кибербезопасности. – 2022. – № 4(50). – С. 13–25.

7. Большаков, А. С. Управление информационной безопасностью персональных данных с использованием нечеткой логики / А. С. Большаков, А. И. Жила, А. В. Осин // Наукоемкие технологии в космических исследованиях Земли. – 2021. – Т. 13, № 4. – С. 37–47.

8. Миняев, А. А. Моделирование угроз безопасности информации в территориально-распределенных информационных системах / А. А. Миняев // Наукоемкие технологии в космических исследованиях Земли. – 2021. – Т. 13, № 2. – С. 52–65.

9. Васильев, В. И. Оценка актуальных угроз безопасности информации с помощью технологии трансформеров / В. И. Васильев, А. М. Вульфин, Н. В. Кучкарова // Вопросы кибербезопасности. – 2022. – № 2(48). – С. 27–38.

10. Массель, Л. В. Семантическое моделирование при построении цифровых двойников энергетических объектов и систем / Л. В. Массель, А. Г. Массель // Онтология проектирования. – 2023. – Т. 13, № 1(47). – С. 44–54.

11. Методика оценки актуальных угроз и уязвимостей на основе технологий когнитивного моделирования и Text Mining / В. И. Васильев [и др.] // Системы управления, связи и безопасности. – 2021. – № 3. – С. 110–134.

12. Kourbatski, A. Semantic aspects of the experts' communication problem in relation to the conceptual design of complex systems / A. Kourbatski, K. Mulyarchik // Open Semantic Technologies for Intelligent Systems: 11th Intern. Conf., OSTIS 2021. Communications in Computer and Information Science. – Cham : Springer, 2022. – Vol. 1625. – P. 77–88.

13. Касумов, В. А. Модель и метод определения оптимальной структуры системы обеспечения безопасности для критической информационной инфраструктуры / В. А. Касумов, Д. И. Мамедов // Доклады БГУИР. – 2023. – Т. 21, № 2. – С. 95–103.

14. Никитина, И. С. Использование диаграмм потоков данных для представления предметной области / И. С. Никитина // Вестник современных исследований. – 2018. – № 7.1. – С. 324–328.

15. Давлетшина, Л. А. Моделирование информационных потоков IT-компании на основе методологии диаграммы потоков данных / Л. А. Давлетшина, И. К. Будникова // Информационные технологии в строительных, социальных и экономических системах. – 2021. – № 1. – С. 87–91.

16. Brazhuk, A. Framework for ontology-driven threat modelling of modern computer system / A. Brazhuk, E. Olizarovich // Intern. J. of Open Information Technologies. – 2020. – Vol. 8, no. 2. – P. 14–20.

17. Гаврилова, Т. А. Онтологический инжиниринг от истории к практическому формированию / Т. А. Гаврилова // Когнитивные исследования ; под ред. В. Д. Соловьева. – 2022. – № 2. – С. 293–307.

18. Грибова, В. В. Онтологические инфраструктуры для решения интеллектуальных задач / В. В. Грибова, Е. А. Шалфеева // Интегрированные модели и мягкие вычисления в искусственном интеллекте (ИММВ-2021) : сб. науч. тр. X Междунар. науч.-техн. конф., Коломна, 17–20 мая 2021 г. – Смоленск : Универсум, 2021. – Т. 1. – С. 68–77.

19. Городецкий, В. И. Искусственный интеллект: метафора, наука и информационная технология / В. И. Городецкий, Р. М. Юсупов // Мехатроника, автоматизация, управление. – 2020. – Т. 21, № 5. – С. 282–294.

20. Голенков, В. В. Основные направления развития интеллектуальных компьютерных систем нового поколения и соответствующей им технологии / В. В. Голенков, Н. А. Гулякина, Д. В. Шункевич // Science and innovation. – 2023. – Т. 2, special iss. 3. – С. 267–280.

21. Интеллектуальный анализ данных и облачные вычисления / М. М. Татур [и др.] // Доклады БГУИР. – 2019. – № 6. – С. 62–71.

22. Гаврилова, Т. А. Инженерия знаний. Модели и методы / Т. А. Гаврилова, Д. В. Кудрявцев, Д. И. Муромцев. – СПб. : Лань, 2016. – 324 с.

23. Милько, Д. С. База знаний экспертной системы оценки угроз безопасности информации / Д. С. Милько, А. В. Данеев, А. Л. Горбылев // Докл. Томского гос. ун-та систем управления и радиоэлектроники. – 2022. – Т. 25, № 1. – С. 61–69.

24. Разин, В. В. Метод принятия решений на основе анализа ситуаций и семантических технологий / В. В. Разин, А. Ф. Тузовский // Изв. Томского политехн. ун-та. Инжиниринг георесурсов. – 2012. – Т. 321, № 5. – С. 188–193.

25. Буракова, Е. Е. Языки описания онтологий для технических предметных областей / Е. Е. Буракова, Н. М. Боргест, М. Д. Коровин // Вестник Самарского гос. аэрокосмического ун-та им. академика С. П. Королёва (Национального исследовательского ун-та). – 2014. – № 3(45). – С. 144–158.

26. Осипов, Г. Методы искусственного интеллекта / Г. Осипов. – М. : ФИЗМАТЛИТ, 2011. – 296 с.

27. Маторин, С. И. Системно-объектный детерминантный анализ. Построение таксономии предметной области / С. И. Маторин, В. В. Михелев // Искусственный интеллект и принятие решений. – 2021. – № 1. – С. 15–24.

28. Automating the early detection of security design flaws / K. Tuma [et al.] // Proceedings of the 23rd ACM/IEEE Intern. Conf. on Model Driven Engineering Languages and Systems, Virtual Event, Canada, 16–23 Oct. 2020. – Canada, 2020. – P. 332–342.

29. Adopting threat modelling in agile software development projects / K. Bernsmed [et al.] // J. of Systems and Software. – 2022. – Vol. 183. – P. 111090.

30. Architecture of cloud telecommunication network monitoring platform based on knowledge graphs / K. Krinkin [et al.] // 2021 30th Conf. of Open Innovations Association FRUCT 2021, Oulu, Finland, 27–29 Oct. 2021. – Oulu, 2021. – P. 107–114.

31. Забавский, В. В. Уязвимости в технологии контейнеризации docker / В. В. Забавский, Т. В. Борботько // Управление информационными ресурсами : материалы XVII Междунар. науч.-практ. конф., Минск, 12 марта 2021 г. – Минск : Академия управления при Президенте Республики Беларусь, 2021. – С. 208–209.

32. Brazhuk, A. Threat modeling of cloud systems with ontological security pattern catalog / A. Brazhuk // Intern. J. of Open Information Technologies. – 2021. – Vol. 9, no. 5. – P. 36–41.

References

1. Shostack A. Experiences threat modeling at Microsoft. *MODSEC@ MoDELS*, 2008, vol. 2008, 35 p.

2. Konev A., Shelupanov A., Kataev M., Ageeva V., Nabieva A. A survey on threat-modeling techniques: protected objects and classification of threats. *Symmetry*, 2022, vol. 14, no. 3, p. 549.

3. Makarevich V. A., Miniukovich K. A., Mulyarchik K. S. *Organisation's information security threat analysis and modelling based on a universal canvas.* Zhurnal Belorusskogo gosudarstvennogo universiteta. Ekonomika [*Journal of the Belarusian State University. Economics*], 2021, no. 1, pp. 57–68 (In Russ.).

4. Kochin V. P., Vorotnitsky U. I. Proektirovanie i obespechenie bezopasnosti integrirovannyh obrazovatel'nyh informacionno-kommunikacionnyh system. *Design and Security of Integrated Educational Information and Communication Systems*. Minsk, Belarusian State University, 2022, 168 p. (In Russ.).

5. Verreydt S., Van Landuyt D., Joosen W. Expressive and systematic risk assessments with instance-centric threat models. *SAC'23: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, Tallinn, Estonia, 27–31 March 2023*. Tallinn, 2023, pp. 1450–1457.

6. Yazov Yu. K., Soloviev S. V., Tarelkin M. A. *Logical-linguistic modeling of security threats information in information systems*. Voprosy kiberbezopasnosti [*Cybersecurity Issues*], 2022, no. 4(50), pp. 13–25 (In Russ.).

7. Bolshakov A. S., Zhila A. I., Osin A. V. *Fuzzy logic data protection management*. Naukoemkie tekhnologii v kosmicheskih issledovaniyah Zemli [*High Tech in Earth Space Research*], 2021, vol. 13, no. 4, pp. 37–47 (In Russ.).

8. Minyaev A. A. *Modeling information security threats in territorial-distributed information systems.* Naukoemkie tekhnologii v kosmicheskih issledovaniyah Zemli [*High Tech in Earth Space Research*], 2021, vol. 13, no. 2, pp. 52–65 (In Russ.).

9. Vasilyev V. I., Vulfin A. M., Kuchkarova N. V. *Assessment of current threats to information security using transformer technology*. Voprosy kiberbezopasnosti [*Cybersecurity Issues*], 2022, no. 2(48), pp. 27–38 (In Russ.).

10. Massel L. V., Massel A. G. *Semantic modeling in the construction of digital twins of energy objects and systems*. Ontologia proektirovania [*Ontology of Design*], 2023, vol. 13, no. 1(47), pp. 44–54 (In Russ.).

11. Vasilyev V. I., Vulfin A. M., Kirillova A. D., Kuchkarova N. V. *Methodology for assessing current threats and vulnerabilities based on cognitive modeling technologies and text minin*g. Sistemy upravlenia, sviazi i bezopasnosti [*Systems of Control, Communication and Security*], 2021, no. 3, pp. 110–134 (In Russ.).

12. Kourbatski A., Mulyarchik K. Semantic aspects of the experts' communication problem in relation to the conceptual design of complex systems. *Open Semantic Technologies for Intelligent Systems: 11th International Conference, OSTIS 2021. Communications in Computer and Information Science*. Cham, Springer, 2022, vol. 1625, pp. 77–88.

13. Kasumov V. A., Mamedov D. I. *Model and method for determining the optimal structure of the security system for critical information infrastructure*. Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioèlektroniki [*Reports of the Belarusian State University of Informatics and Radioelectronics*], 2023, no. 21(2), pp. 95–103 (In Russ.).

14. Nikitina I. S. *The use of data flow diagrams for representation of subject area*. Vestnik sovremennyx issledovanii [*Bulletin of Modern Research*], 2018, no 7.1, pp. 324–328 (In Russ.).

15. Davletshina L. A., Budnikova I. K. *Modeling of IT company information flows based on data flow diagrams methodology*. Informacionnye texnologii v stroitelnych, socialnych i ekonomicheskich sistemah [*Information Technologies in Construction, Social and Economic Systems*], 2021, no. 1, pp. 87–91 (In Russ.).

16. Brazhuk A., Olizarovich E. Framework for ontology-driven threat modelling of modern computer system. *International Journal of Open Information Technologies*, 2020, vol. 8, no. 2, pp. 14–20.

17. Gavrilova T. A. *Ontological engineering from history to practical use*. Kognitivnye issledovania [*Cognitive Research*]. In V. D. Soloviev (ed.), 2022, no. 2, pp. 293–307 (In Russ.).

18. Gribova V. V., Shalfeeva E. A. *Ontological infrastructures for solving intellectual tasks*. Integrirovannye modeli i mjagkie vychislenija v iskusstvennom intellekte (IMMV-2021) : sbornik nauchnyh trudov X Mezhdunarodnoj nauchno-tehnicheskoj konferencii, Kolomna, 17–20 maja 2021 g. [*Integrated Models and Soft Computing in Artificial Intelligence (IMMV-2021) : Collection of Scientific Papers of the X International Scientific and Technical Conference, Kolomna, 17–20 May 2021*]. Smolensk, Universum, 2021, vol. 1, pp. 68–77 (In Russ.).

19. Gorodetsky V. I., Yusupov R. M. *Artificial intelligence: metaphor, science and information technology*. Mekhatronika, avtomatizatsiya, upravlenie [*Mechatronics, Automation, Control*], 2020, no. 21(5), pp. 282–294 (In Russ.).

20. Golenkov V. V., Guliakina N. A., Shunkevich D. V. Main directions of development of intelligent computer systems of new generation and appropriate technology. *Science and Innovation*, 2023, vol. 2, special iss. 3, pp. 267–280 (In Russ.).

21. Tatur M. M., Lukashevich M. M., Pertsev D. Y., Iskra N. A. *Intelligent data analysis and cloud computing.* Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioèlektroniki [*Reports of the Belarusian State University of Informatics and Radioelectronics*], 2019, no. 6, pp. 62–71 (In Russ.).

22. Gavrilova T. A., Kudrjavcev D. V., Muromcev D. I. Inzhenerija znanij. Modeli i metody. *Knowledge Engineering. Models and Methods*, Saint Petersburg, Lan', 2016, 324 p. (In Russ.).

23. Milko D. S., Daneev A. V., Gorbylev A. L. *Knowledge base of the expert system for cyber security threat modeling*. Doklady Tomskogo gosudarstvennogo universiteta sistem upravlenija i radiojelektroniki [*Reports of Tomsk State University of Control Systems and Radioelectronics*], 2022, vol. 25, no. 1, pp. 61–69 (In Russ).

24. Razin V. V., Tuzovsky A. F. *Decision-making method based on situation analysis and semantic technologies*. Izvestija Tomskogo politehnicheskogo universiteta. Inzhiniring georesursov [*News of Tomsk Polytechnic University. Georesources Engineering*], 2012, vol. 321, no. 5, pp. 188–193 (In Russ.).

25. Burakova E. E., Borgest N. M., Korovin M. D. *Ontology description languages for high-tech fields of applied engineering.* Vestnik Samarskogo gosudarstvennogo ajerokosmicheskogo universiteta im. akademika S. P. Koroljova (Nacional'nogo issledovatel'skogo universiteta) [*Bulletin of Samara State Aerospace University named after Academician S. P. Korolev (National Research University)*], 2014, no. 3(45), pp. 144–158 (In Russ.).

26. Osipov G. Metody iskusstvennogo intellekta. *Artificial Intelligence Methods*. Moscow, FIZMATLIT, 2011, 296 p. (In Russ.).

27. Matorin S. I., Mikhelev V. V. *System-object determinant analysis. Partitive classification using the formal-semantic normative system*. Iskusstvennyj intellekt i prinjatie reshenij [*Artificial Intelligence and Decision Making*], 2021, no. 1, pp. 15–24 (In Russ).

28. Tuma K., Sion L., Scandariato R., Yskout K. Automating the early detection of security design flaws. *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Virtual Event, Canada, 16–23 October 2020*. Canada, 2020, pp. 332–342.

29. Bernsmed K., Cruzes D. S., Jaatun M. G., Iovan M. Adopting threat modelling in agile software development projects. *Journal of Systems and Software*, 2022, vol. 183, p. 111090.

30. Krinkin K., Kulikov I., Vodyaho A., Zhukova N. Architecture of cloud telecommunication network monitoring platform based on knowledge graphs. *2021 30th Conference of Open Innovations Association FRUCT 2021, Oulu, Finland, 27–29 October 2021*. Oulu, 2021, pp. 107–114.

31. Zabavskii V. V., Borbotko T. V. *Vulnerabilities in docker container technology*. Upravlenie informacionnymi resursami : materialy XVII Mezhdunarodnoj nauchno-prakticheskoj konferencii, Minsk, 12 marta 2021 g. [*Information Resource Management : Materials of the XVII International Scientific and Practical Conference, Minsk, 12 March 2021*]. Minsk, Akademija upravlenija pri Prezidente Respubliki Belarus', 2021, pp. 208–209 (In. Russ.).

32. Brazhuk A. Threat modeling of cloud systems with ontological security pattern catalog. *International Journal of Open Information Technologies*, 2021, vol. 9, no. 5, pp. 36–41.

Информация об авторах

Бражук Андрей Иосифович, магистр естественных наук, ведущий инженер-программист Информационно-аналитического центра Гродненского государственного университета имени Янки Купалы.
E-mail: brazhuk@grsu.by

Олизарович Евгений Владимирович, кандидат технических наук, доцент, начальник Информационно-аналитического центра Гродненского государственного университета имени Янки Купалы.
E-mail: e.olizarovich@grsu.by

Information about the authors

Andrei I. Brazhuk, M. Sc., Lead Software Engineer at the Information and Analytical Center, Yanka Kupala State University of Grodno.
E-mail: brazhuk@grsu.by

Evgeny V. Olizarovich, Ph. D. (Eng.), Assoc. Prof., Head of the Information and Analytical Center, Yanka Kupala State University of Grodno.
E-mail: e.olizarovich@grsu.by