Проектирование отказоустойчивых систем с микросервисной архитектурой

Т. М. Татарникова

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

tm-tatarn@yandex.ru

Е. Д. Архипцев

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

lokargenia@gmail.com

Аннотация. **Обсуждается проблема отказоустойчивости распределённых систем на микросервисной архитектуре. Рассматриваются шаблоны к проектированию систем для повышения надёжности. Описываются критерии, которым должны соответствовать микросервисы. На примерах показано, как осуществляется использование инфраструктуры для повышения отказоустойчивости. Обсуждается проблема распределенного хранения больших данных.**

Ключевые слова: микросервисы, отказоустойчивость микросервисов, пул запросов, распределённые ресурсы

I. Введение

Отказоустойчивость системы означает, что система способна сохранять свою работоспособность даже при отказе отдельных компонентов или связанных систем, и восстанавливать работоспособность после восстановления отказавших компонентов или связанных систем. Это достигается путем расчета обеспечения деградации работоспособности системы пропорционально серьезности отказа конкретных компонентов. То есть, система должна продолжать работать стабильно даже при отказе не критичных компонентов, минимизируя влияние на остальные функциональные возможности. Кроме того, стабильность системы предполагает ее способность автоматически восстанавливать работоспособность после сбоя как отдельных компонентов, так и всей системы в целом [1]. Например, при временном отключении сети стабильная система автоматически восстанавливает связь и продолжает работу без вмешательства со стороны операторов.

Рассмотрим пример построения сервера в монолитной архитектуре. Как видно из рис. 1 система состоит из 3 компонент: балансировка нагрузки, сервер и база данных.

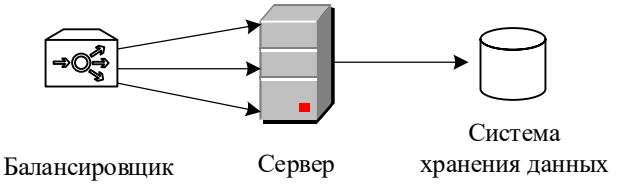


Рис. 1. Пример балансировки нагрузки в монолитной архитектуре

Монолитная архитектура является централизованной, поскольку все части приложения, включая пользовательский интерфейс, бизнес-логику и слой данных, развертываются и запускаются в одном процессе или на одном сервере.

При централизованном подходе один узел или группа узлов управляют передачей служебной информации, на основании которой происходит сбалансированное распределение нагрузки в системе. Недостатком такого подхода заключается в возможной перегрузке узла с централизованным принятием решения – наличие так называемой точки отказа [2].

В микросервисной архитектуре сервер и база данных не являются единым целым, а разбиваются на составляющие, объединенные областью применения (рис. 2). В такой архитектуре, например сервис по продаже компьютеров может быть разделен на следующие микросервисы: сервис-каталог, сервис-заказов, сервис-оплаты. Каждый из сервисов взаимодействует со своей базой данных, тем самым повышая надежность: при выходе из строя одного из сервисов. Архитектура приложения должна частично регрессировать, при этом сохранив оставшийся рабочий функционал [3].

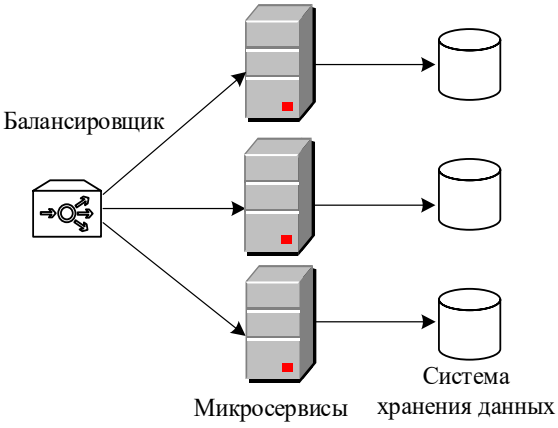


Рис. 2. Пример балансировки нагрузки в микросервисной архитектуре

Все участвующие в реализации функций балансировки нагрузки узлы обмениваются служебной информацией, на основании которой локально принимается решение с учетом собственных ресурсов узла.

На рис. 3 и рис. 4 приведены гистограммы среднего времени обработки запросов пользователей и длины очереди для монолитной и микросервисной архитектуры соответственно. Расчеты выполнены по моделям M|M|1 и M|M|3 при времени обслуживания сервера 0,1 с. Очевидно, что микросервисная архитектура является более эффективной с ростом нагрузки.

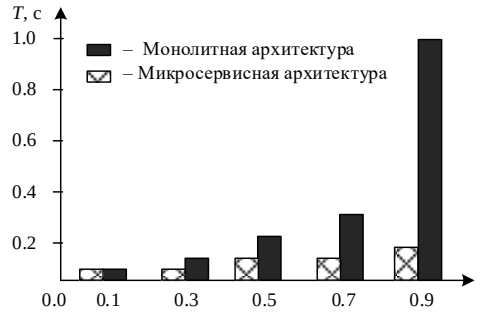


Рис. 3. Среднее время обработки запросов пользователей в монолитной и микросервисной архитектурах

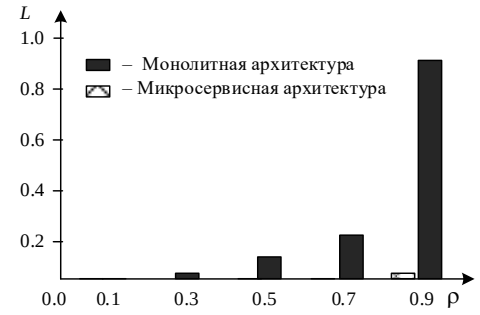


Рис. 4. Средняя длина очереди в монолитной и микросервисной архитектурах

Учитывая особенности архитектуры, можно сформулировать критерии, которым должна соответствовать система:

- надежность и отказ в каскаде (Cascading Failures) – один отказ не должен привести к другим отказам в системе;
- восстановление (Backup and Restore) – система должна иметь возможность к быстрому восстановлению.

Единственным способом создания системы, устойчивой к сбоям, является проведение стресс-тестов или хаос-инжиниринга. После анализа результатов тестов делаются выводы, которые влияют на изменения в архитектуре и подходах к написанию кода. Этот цикл повторяется до достижения нужного уровня отказоустойчивости. Количество успешно пройденных тестов становится метрикой для оценки степени отказоустойчивости как отдельных компонентов, так и всей системы в целом. Создание таких тестов представляет собой инженерную задачу, начиная с описания и автоматизации сценариев отказа, таких как потеря сети, нехватка места на диске, ошибки от внешних сервисов и т. д.

A. Отсутствие единой точки отказа

Отсутствие единой точки отказа (No SPoF) является важным принципом для обеспечения надежности и отказоустойчивости системы на всех уровнях, от отдельных компонентов до целых дата-центров. Для реализации этого принципа сервисы должны быть готовы к запуску в нескольких экземплярах и не зависеть от сохранения состояния (stateless).

Рассмотрим сценарий, когда сервис хранит состояние и не готов к горизонтальному масштабированию. Примером такого состояния может быть сессия пользователя, хранимая на конкретном микросервисе.

При горизонтальном масштабировании до нескольких экземпляров сервиса возникает проблема балансировки запросов, что может привести к тому, что запрос авторизованного пользователя попадет на микросервис, где отсутствует его сессия [3]. Для решения этой проблемы можно вынести сессии в отдельное хранилище, например, Redis, где данные хранятся в виде пар «ключ-значение» (рис. 5).

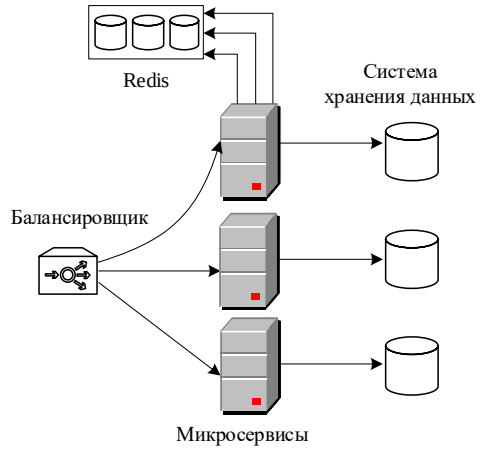


Рис. 5. Использование Redis в качестве хранилища общих данных (сессии пользователя)

Из рис. 2 и рис. 5 заметно, что для каждого микросервиса (группы эквивалентных микросервисов) используется именное хранилище. Данный подход соответствует шаблону проектирования «База данных на сервис» (Database Per Service).

Один из основных принципов при переходе на архитектуру микросервисов – обеспечить каждому сервису свою собственную область данных, чтобы избежать сильных зависимостей на уровне хранилища. Это означает логическое разделение данных, где микросервисы могут работать с общей физической базой данных, но используют отдельные схемы, коллекции или табл. [4].

Шаблон проектирования «Database Per Service» основан на этом принципе и направлен на увеличение автономности микросервисов и снижение связности между командами, разрабатывающими эти сервисы. Однако у этого подхода есть и недостатки: усложняется обмен данными между сервисами и обеспечение транзакционной согласованности ACID. Кроме того, этот шаблон не рекомендуется для небольших приложений, а скорее подходит для крупномасштабных

проектов с множеством микросервисов, где каждая команда должна иметь полный контроль над своими данными для ускорения разработки и лучшего масштабирования.

B. Создание пула соединений

В небольших проектах наиболее простым способом взаимодействия с микросервисами является прямое обращение от клиента к каждому сервису по отдельности. Однако в корпоративных приложениях с множеством микросервисов рекомендуется использовать паттерн API Gateway. API Gateway – это шлюз, расположенный между клиентским приложением и микросервисами, предоставляющий единую точку входа для клиента.

Таком образом схема приложения обретает следующий вид, приведенный на рис. 6.

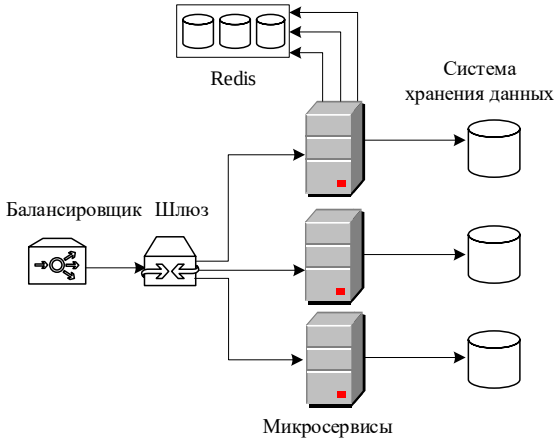


Рис. 6. Использование пула запросов

Использование пула http-соединений для обработки запросов могут вызвать следующую проблему: переполнение пула одним из сервисов. В случае, если один микросервис медленно (с задержкой) обрабатывает запросы, то это может переполнить пул соединений. Когда сервис полностью выходит из строя, запрос моментально отклоняется, что не способствует переполнению. Для решения данной проблемы необходимо завести отдельный пул на каждый из сервисов, как показано на рис. 7.

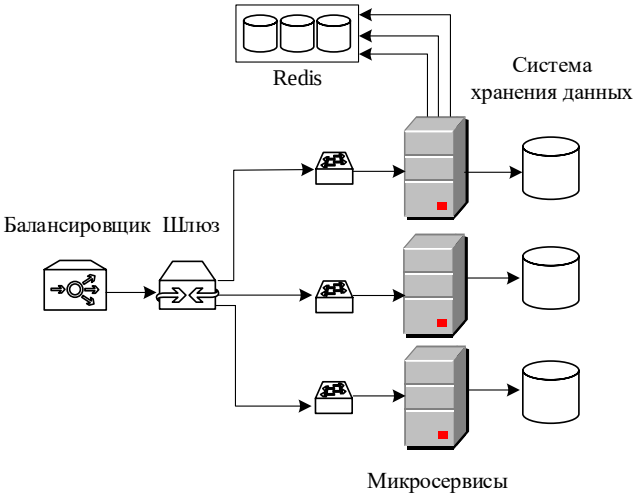


Рис. 7. Использование именной очереди для каждого сервиса

C. Повышение отказоустойчивости

В рассмотренном выше разделе для каждого микросервиса выделен отдельный пул соединений. Это решает проблему в случае с медленными и избыточными запросами, поскольку они не перегружают оставшиеся микросервисы. Однако даже в таком случае не учитываются ресурсы хоста, на котором были развёрнуты сервисы. Если один из сервисов будет полностью использовать ресурсы на машине, это приведёт к сбою и отказу системы.

Для решения данной проблемы в систему необходимо добавить контроль за ресурсами. Данный принцип описан в шаблоне «Переборка» (Bulkhead). Этот шаблон получил свое название благодаря принципам, используемым в судостроении для защиты кораблей от полного затопления в случае повреждения. Точно так же, в архитектуре приложений, он позволяет изолировать различные компоненты приложения, чтобы при сбое одного из них остальные продолжали функционировать. Этот шаблон позволяет эффективно управлять ресурсами, гарантируя, что использование ресурсов для одного сервиса не оказывает влияния на ресурсы, используемые для других сервисов.

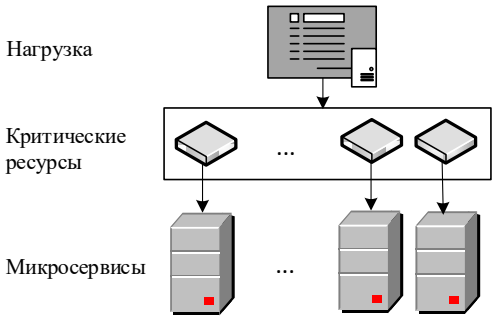


Рис. 8. Использование критических ресурсов на каждый микросервис

Как видно из рис. 8, для каждого сервиса выделено часть ресурсов хоста, с которыми может взаимодействовать лишь указанный микросервис.

В данном примере ограничения наложены на микросервисы. Такой же подход можно применить и к пользователям – назначение каждому клиенту своего собственного экземпляра сервиса. Таким образом, если один клиент генерирует слишком много запросов, перегружая свой экземпляр, это не повлияет на работу других клиентов, которые смогут продолжить свою работу независимо (рис. 9).

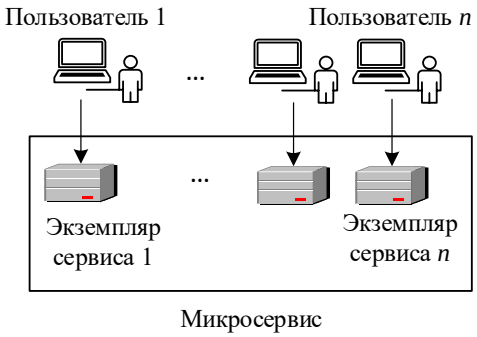


Рис. 9. Распределение ресурсов на каждого пользователя

Каждый пользователь получает свой собственный экземпляр сервиса, что уменьшает вероятность отказа системы из-за перегрузки отдельного пользователя. Это помогает изолировать проблемы, возникающие с одним пользователем, от влияния на других. Однако при использовании отдельных экземпляров могут возникать сложности с согласованием данных между ними, особенно если между ними требуется обмен информацией или синхронизация состояния. Это может потребовать дополнительных механизмов синхронизации и управления состоянием.

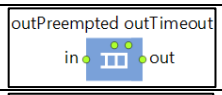
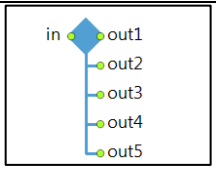
II. Сравнение микросервисных архитектур с разными найстройками надежности

Выше были рассмотрены подходы к повышению отказоустойчивости микросервисов. Произведём сравнение первоначальной настройки и архитектуры с повышенной надежностью. Под повышенной надежностью подразумевается следующее:

- Использование ApiGetway и отдельных пулов-соединений на каждом из сервисов, как на рис. 7.
- Распределение критических ресурсов на каждый микросервис.

Для выполнения сравнения микросервисной архитектуры с первоначальными настройками (сценарий 1) с микросервисной архитектурой с повышенной надежностью (сценарий 2) разработана имитационная модель в системе AnyLogic 8.7 [5]. В табл. I приведены компоненты имитационной модели.

ТАБЛИЦА I. Компоненты имитационной модели

Компонент имитационной модели	**Блок**	**Назначение**
Агент		Запрос пользователя
Блок «Приемник»		Терминал пользователя
Блок «Узел»		Микросервис
Блок «Очередь»		Очередь на обработку микросервисом
Блок «Отказ»		Блокировка потока агентов
Блок «Распределитель»		Распределение запросов по микросервисам в соответствии с вероятностями переходов
Блок «Пул ресурсов»		Моделирование модуля Redis

На рис. 10 приведен график зависимости отклоненных запросов, в % от общего количества запросов, сгенерированных в эксперименте.

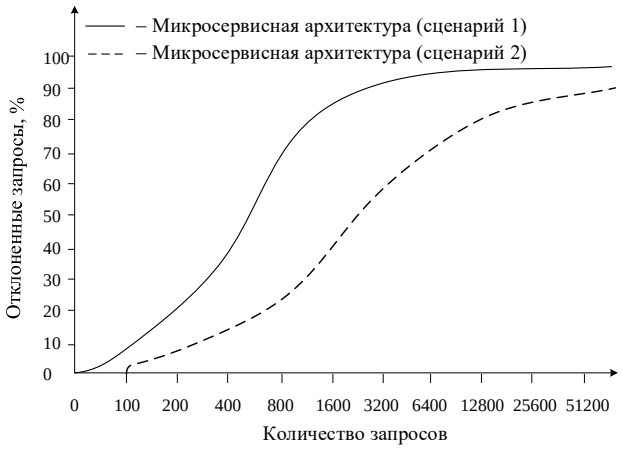


Рис. 10. Процентное соотношение отклоненный запросов

На рис. 10 видно, что сервис, организованный по сценарию 2, справляется лучше с нагрузкой, а число отклонённых запросов никогда не достигает 100 %. При таком подходе сервис не перестаёт полностью отвечать на запросы и сохраняет частичную работоспособность. В микросервисной архитектуре, организованной по сценарию 1, сервис полностью утрачивает возможность функционировать и требует восстановления.

III. Заключение

Основным критерием, которому должна соответствовать информационная система является надежность предоставляемого сервиса – обработка запросов пользователей. Надежность информационной системы обеспечивается отказоустойчивостью микросервисови, что определяет актуальность данного направления в развитии современных архитектурах информационных систем.

Рассмотрены известные способы повышения отказоустойчивости микросервисов. Приведены примеры шаблонов, который повышают отказоустойчивость микросервисов.

Эксперимент на имитационных моделях показывает, что реализация микросервисной архитектурой с повышенной надежностью позволяет сохранить частичную работоспособность информационной сиситемы.

Список литературы

[1] Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. СПб: Питер. 544 с.

[2] Советов Б.Я., Колбанёв М.О., Татарникова Т.М. Диалектика информационных процессов и технологий // Информация и космос. 2014. № 3. С. 96-104.

[3] Определение числа реплик распределенного хранения больших данных / Т.М. Татарникова, Е.Д. Архипцев // Международная конференция по мягким вычислениям и измерениям. 2023. Т. 1. С. 305-308.

[4] Татарникова Т.М., Архипцев Е.Д. Алгоритм контроллера нечеткой логики для размещения файлов в системе хранения данных // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23. № 6. С. 1171-1177. doi: 10.17586/2226-1494-2023-23-6-1171-1177

[5] Кутузов О.И., Татарникова Т.М. К анализу парадигм имитационного моделирования // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 3. С. 552-558.

Проектирование цифрового двойника в нефтегазовых технологиях: интеграция технологий

Н. А. Шатилова[1], П. А. Мальцев[2],
М. Е. Подкина [3]

Санкт-Петербургский горный университет императрицы Екатерины II

[1]n_a_shatilova@mail.ru, [2]maltcev-pave@mail.ru,
[3]m.losckaya@yandex.ru

С. Е. Абрамкин

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

seabramkin@etu.ru

Аннотация. **Концепция цифрового двойника стала наиболее востребована в последнее десятилетие в связи с развитием информационных технологий. Большие данные, искусственный интеллект, облачные вычисления, IoT и т. д. – это направления и технологии, объединение которых и представляет собой цифровой двойник. Однако в этом и заключается некоторая проблема, т.к. цифровой двойник зависит от составляющих, которые сами находятся в процессе становления. В данной работе производится исследование ключевых технологий, составляющих цифровой двойник, их место в нефтегазовой отрасли, а также анализируется структура цифрового двойника и место ключевых технологий в ней.**

Ключевые слова: цифровой двойник; информационные технологии; нефтегазовые технологии; цифровизация

I. Введение

Внедрение цифровых технологий в производственные процессы – актуальная тема для теоретических исследований и для практической реализации. Быстрое развитие информационных технологий способствует решению проблем, связанных с проектированием, изготовлением и безопасной эксплуатацией сложных современных технических систем [1].

Цифровой двойник (ЦД) как система объединенных технологий предлагает более эффективные средства для решения этих проблем, чем применение технологий раздельно. Направления, технологии и инструменты, образующие ЦД комплексно описывают объект, интегрируются между собой для обеспечения непрерывности взаимодействия виртуальной среды и физического объекта.

Интеграция является основой определения ЦД: не смотря на отсутствие согласованности по данному вопросу, начиная с зеркального отражения реальности Д. Гелернтера [2] она может быть определена как один из ключевых элементов. Действительно, что для идеи зеркального мира, что для создания любого из современных высокотехнологичных изделий, необходимы знания различных дисциплин. Благодаря интеграции различных технологий и реализуется взаимосвязь физического и виртуального.

Модификация определения ЦД сопровождается изменениями в наборе технологий, определяющих его функционал [3]. От этого и возникают проблемы с определением ЦД, а также со смежными понятиями цифровой тени и цифровой модели [4]. Рассмотрим ключевые технологии, которые обычно связывают с понятием ЦД.

II. Ключевые технологии цифрового двойника

Нефтегазовые компании в процессе своей ежедневной деятельности получают колоссальные объемы данных. Для выявления взаимосвязей, выполнения предсказаний результатов и поведения, достижения оптимальных результатов, используют Большие данные (Big data).

ТАБЛИЦА I. Эволюция ЦД и сопутствующих технологий в концепции Microsoft. Источник: [3]

1 этап	1995 – 2002 гг. 1. Появление термина ЦД; 2. Модель информационного зеркалирования; 3. Выделенные рабочие станции и серверы; 4. Развитие 3D – моделирования и компьютерного цифрового управления; 5. Роботы.
2 этап	2003 – 2013 гг. 1. Выделение цифрового имитационного моделирования, 3D-печати, цифрового моделирования, виртуальной сборки, предварительного имитационного моделирования; 2. Браузеры, веб-доступ; 3. Появление 3D-печати в массах.
3 этап	2014 – 2016 гг. 1. Реализация обмена данными между цифровым и физическим миром; 2. Широкое применение IoT и аналитики Больших данных; 3. Облака; 4. Быстрая обратная связь с объектом на этапах жизненного цикла; 5 Дополнение продуктов цифровыми сервисами.
4 этап	2017 г. – по н.в. 1. Голография; 2. Дополненная и виртуальная реальность; 3. Интеллектуальные сервисы; 4. Искусственный интеллект; 5. Новые форматы человеко-машинного взаимодействия; 6. Автономная работа; 7. Самовосстановление.

Эффективное применение Big data помогает сократить расходы и максимизировать прибыль благодаря комплексному применению прогностических методов.

Но следует учесть, что при использовании Big data закупка датчиков, IoT-платформ, сбор обширных объемов информации, не являются единственными сдерживающими факторами: высокая стоимость установки и обслуживания датчиков, их склонность к ошибкам и сбоям, возможность неправильных показаний, перегрузка данными, их зашумленность и т. д. Эти проблемы и ограничения могут создавать трудности при применении Big data в некоторых условиях и объектах.

Свойству искусственных интеллектуальных систем выполнять творческие функции – искусственному интеллекту (AI) в нефтегазовой отрасли есть несколько основных применений: Машинное обучение (Machine learning) и Наука о данных (Data science).

Машинное обучение необходимо для извлечения полезной информации из больших и разнородных наборов данных и их интерпретирования без участия человека с итеративным совершенствованием. Немаловажно, что модели, основанные на алгоритмах машинного обучения, могут обладать высокой точностью, от чего другие технологии ЦД могут извлечь выгоду [5].

Применение алгоритмов машинного обучения требует наличия большого количества данных, которые могут быть доступны не для каждого процесса в производственном цикле. На месторождениях, работающих в режиме падающей добычи с продолжительной историей эксплуатации исторической информации достаточно, однако на таком месторождении в случае прогнозирования добычи существует не так много параметров, на которые оператор может оказать влияние.

Для задач добычи нефти совмещение классических подходов с технологиями машинного обучения и искусственного интеллекта позволит повысить маржинальность процессов добычи углеводородного сырья, снизив стоимость технологического процесса посредством их оптимизации [6], отслеживать операции и быстро реагировать на проблемы. Machine learning также можно использовать для проведения симуляций, используя модели прогнозных данных для выявления закономерностей на основе разнородных входных данных [7].

Модели, основанные только на базе технологий Machine learning, уступают математическому моделированию физических процессов в сочетании с данными. Подобное сочетание является более перспективным для решения задач «что, если?» и может применяться в ситуациях, когда недостаточно данных для статистических подходов. Модели, основанные на данных, ограничены стадией эксплуатации изделия [3].

Data science – это наука, которая использует возможности искусственного интеллекта для анализа и обработки данных. Она позволяет извлекать ценную информацию из сложных и объемных наборов данных, используя нейронные сети для объединения фрагментов информации и создания полной картины. В контексте разведки и добычи нефти и газа применение data science имеет особую важность, так как оно делает эти данные более доступными и позволяет эффективно использовать уже существующую инфраструктуру.

Нефтяная промышленность потенциально опасная отрасль, поэтому является крайне полезным и важным применение AI с целью тестирования потенциальных воздействий новых разработок, оценки экологического риска [8].

AI используется с целью прогнозирования динамики добычи на месторождении, оптимизации плана разработки, идентификации остаточной нефти, трещин и повышение нефтеотдачи пластов и т. д.

В нефтегазовой отрасли применение AI не получило достаточного развития в связи с рядом причин [9]:

- Отсутствие стопроцентной точности при вычислении определенных переменных;
- Риск возникновения аварий и технических ошибок;
- Недостаточный уровень доверия к технологии;
- Риск нарушения приватности и конфиденциальности информации;
- Снижение количества рабочих мест;
- Уязвимость к кибератакам и взлому и др.

Все эти причины можно отнести и к ЦД, что также объясняет позднее внедрение ЦД в нефтегазовую промышленность. Однако возможности, которые предлагает ЦД толкают специалистов к поиску компромиссов и решению этих проблем.

Быстрая обработка и анализ данных изображений и видео – востребованное направление в данной отрасли. Машинное зрение широко применяется для задач обнаружения объектов, отслеживания объектов, виртуальных измерений, выступая с целью мониторинга, контроля и управления. Так, применение машинного зрения может повысить автономность интеллектуальных алгоритмов управления нефтебазой и обеспечить ее безопасную эксплуатацию [10].

Облачные вычисления необходимы для внедрения ЦД, дополненных возможностями внешних облачных серверов по требованию, обеспечивающих масштабируемость коммуникаций, хранения данных и вычислений [11]. Облачные архитектуры программного обеспечения характеризуются независимостью между уровнем служб и базами данных, лежащими в их основе, благодаря доступу к данным на основе API.

Предоставление облачных сервисов может быть как общедоступным, так и частным. Однако выбор в пользу удобства общедоступных облачных сервисов делают не все компании из-за возможной утечки данных. При этом данная модель выделяется более безопасной чем облачное частное хранилище по причине большего масштаба; они подлежат в большей степени тестированию безопасности, надежности и др. Также организации, предоставляющие общедоступные

облачные сервисы более расширяемые и направленные на улучшения оборудования для обеспечения безопасности сервисов. При использовании частного облака часто опускается вопрос безопасности при проникновении в сеть, защиты информации от новых методов кибератак. В любом из выбранных вариантов, ЦД объектов нефтегазовой отрасли должны быть развернуты с выполнением требований безопасности [12].

IoT-платформа представляет собой программно-аппаратное решение, разработанное для связи и управления датчиками, контроллерами и другими внешними устройствами сбора данных. Ее функционал также включает реализацию пост-процессинговых и аналитических возможностей, зависящих от специфики производственных задач, которые платформа предназначена решать.

IoT используется в нефтегазовой отрасли для сбора, передачи и анализа необработанных данных в режиме реального времени, чтобы получить четкое представление о процессах на объектах.

IoT и ЦД являются взаимосвязанными и тесно взаимодействующими технологиями, которые вместе способствуют взаимному развитию. Благодаря резкому увеличению числа устройств IoT и снижению их стоимости, использование ЦД активно расширяется. К тому же, прогресс в области разработки «умных» датчиков для IoT позволяет ЦД использовать все более компактные и эффективные решения.

Киберфизическая система (CPS) – это одна из тенденций в исследовательских работах, связанных с IoT. CPS можно определить как совокупность цифровых, аналоговых и физических компонентов, функционирующих посредством интегрированной физической технологии и логики [13], определяя тем самым ЦД как цифровую составляющую CPS. Все чаще системы и датчики коммуницируют друг с другом через Интернет, что приводит к взаимодействию физического и виртуального, образуя CPS. Подход CPS предъявляет к моделям идентичные требования как на стадиях проектирования и эксплуатации, так и при решении задач идентификации объектов в процедуре синтеза систем управления [14].

Предшественниками CPS являются беспроводные сенсорные сети, АСУ ТП [15, 16], системы реального времени, распределенные вычислительные системы. И CPS, и ЦД способствуют развитию Индустрии 4.0.

Роль человека в Индустрии 4.0 поднимается большим количеством исследователей. Используя интеллектуальные устройства, виртуальную и дополненную реальность (VR/AR), человек находится в цикле производственного процесса, использует преимущества принятия решений на основе искусственного интеллекта и машинного обучения.

Стоит отметить, что машинное зрение, способное распознавать изображения, является превосходным по сравнению с человеческим с возможностью визуализировать не только видимые в физическом мире объекты и их параметры.

Методы расширенной реальности необходимы по причине того, что большая часть информации, поступающая в человеческий мозг, является визуальной, так процессы мониторинга, обучения и др. становятся более естественными для человека.

IoT, Big data, AI и др. делают ЦД реальных объектов возможной и доступной реальностью.

Применение ЦД для нефтегазовой промышленности дает возможность [17]:

- снизить риски;
- создать исполняемые управляемые графики работы;
- выявить любые изменения в процессе и осуществить соответствующее реагирование;
- оптимизировать производство активов;
- дистанционного наблюдения;
- профилактического технического обслуживания;
- оптимизировать планируемые проверки;
- сократить капитал и операционные расходы;
- сократить время на оптимизацию тестирования;
- повысить производительность;
- повысить безопасность;
- сократить выбросы;
- предотвратить простои.

Ценность ЦД выделяется исследователями [18] как возможности для реализации непрерывного цикла выдвижения, проверки, коррекции гипотез благодаря взаимосвязи виртуальной и физической составляющей. Такая цепочка становится возможной благодаря технологиям, входящим в состав ЦД.

Взаимодействие технологий не может рассматриваться без архитектуры ЦД.

III. Архитектура цифрового двойника

В большинстве ранних работ, посвященных исследованию ЦД, не рассматривались конкретные архитектурные предложения. Выделяли три основные части ЦД: физическая система в реальном пространстве; виртуальная система в киберпространстве; и связь ними для передачи данных и информации.

Впоследствии была предложена четырехуровневая архитектура с физическим, цифровым уровнями, уровнями подключения и приложения. Прикладной уровень, использующий передовые технологии как AI, Data science, явно выделяется в указанном шаблоне для извлечения знаний в режиме реального времени.

Рэйлеану и др. [19] разработали четырехуровневую архитектуру для двунаправленного обмена данными между физическим и цифровым пространством. От уровня сбора и обработки данных, уровнем подключения к уровням, находящимся в облаке и отвечающим за обновление и агрегацию данных (третий) и анализ и принятие решений (четвертый).

В последнее время большинство архитектурных концепция ЦД характеризуется пятью или шестью

уровнями. Каждый из дополнительных уровней обычно специализируется некоторыми функциональными возможностями физического, цифрового или сервисного уровней.

В шестиуровневой архитектуре, предложенной Redelinghuys и др. [20], первые два уровня соответствуют физическому двойнику: к первому относятся различные физические устройства, такие как исполнительные механизмы и датчики, обменивающиеся сигналами с локальным контроллером или устройством сбора данных, которое расположено на втором уровне.

Третий уровень определен авторами как локальное хранилище данных, которое используется для получения значений датчиков от контроллеров предыдущего уровня.

Шлюз Интернета вещей, добавляется как уровень 4. Этот уровень добавляет контекст к данным, полученным с уровня 3, и/или обрабатывает данные в формы, более полезные для более высоких уровней. Уровень 5 состоит из облачных сервисов, которые хранят информацию, полученную с уровня 4.

Так, первые пять уровней обеспечивают необходимую инфраструктуру, на шестом добавляются интеллектуальные возможности. Он подключается к уровням 3, 4 и 5 позволяя обеспечить подключение пользователя к программной информации и исторической информации о физическом двойнике в реальном времени.

Данную архитектуру можно выделить, как одну из наиболее подходящих для последующей модернизации, с целью адаптации к сложным производственным объектам.

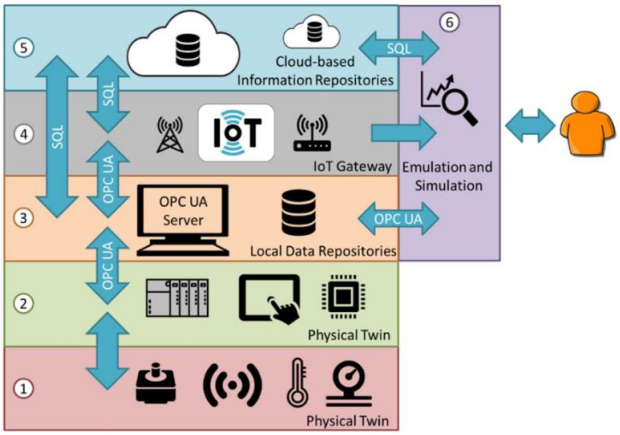


Рис. 1. Рис.1. Шестиуровневая архитектура ЦД. Источник: [20].

Другими исследователями вводятся дополнительные уровни безопасности или конфиденциальности к уровням физического двойника, данных, цифрового двойника и сервисного уровня [21].

Так и помимо указанных уровней Сингх и др. [22] добавили уровень доступа для управления взаимодействия человека и ЦД.

IV. Заключение

Применение ЦД позволяет проверять различные характеристики, поведение на очень большом числе виртуальных испытаний и является достаточным для обеспечения необходимых условий, не приводящих к избытку.

Однако, по вопросу архитектуры ЦД и применения различных технологий у исследователей также нет единого мнения, как и по многим аспектам ЦД. В контексте стандартизации это несёт негативное влияние, однако общая идея – одна: исследователи стремятся расширить известную модель «физический двойник – связь – цифровой двойник» в контексте той технологии, которой уделяется наибольшее внимание авторами. Интеллектуальные технологии применяются на одном из верхних уровней: это может быть как уровень ЦД, сервисный или приложения.

Правильно выполненная интеграция технологий позволит сделать ЦД именно таким, каким его видят в будущем, а для этого необходима работа с представленными технологиями, их интеграцией и архитектурой ЦД и CPS.

Список литературы

[1] Pershin I.M., Papush E.G., Kukharova T.V., Utkin V.A. Modeling of Distributed Control System for Network of Mineral Water Wells. Water 2023, 15, 2289. https://doi.org/10.3390/w15122289

[2] Gelernter D. Mirror Worlds: Or: The Day Software Puts the Universe in a Shoebox. How it Will Happen and What it Will Mean; Oxford University Press: Oxford, UK, 1993

[3] Прохоров А., Лысачев М. Цифровой двойник. Анализ, тренды, мировой опыт. Издание первое, исправленное и дополненное. М.: ООО «АльянсПринт», 2020. 401 стр., ил.

[4] Kritzinger W., Karner M., Traar G., Henjes J., & Sihn W. Digital Twin in manufacturing: A categorical literature review and classification // IFAC-PapersOnLine. 2018. № 11 (51). С. 1016–1022.

[5] Jiang, Yuchen & Yin, Shen & Li, Kuan & Luo, Hao & Kaynak, Okyay. (2021). Industrial applications of digital twins. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 379. 20200360. DOI: 10.1098/rsta.2020.0360.

[6] Shabonas Arturas & Timonov Alexey & Schmidt Sergey. (2021). Повышение эффективности разработки месторождений за счет применения методов машинного обучения для определения оптимальных режимов работы нагнетательных скважин (Field Development Optimization Using Machine Learning Methods to Identify the Optimal Water Flooding Regime).

[7] Zakizadeh Mahdieh & Zand Mazyar & Ir,. (2023). AI application in Oil & Gas Industries.

[8] Katysheva E. Analysis of the Interconnected Development Potential of the Oil, Gas and Transport Industries in the Russian Arctic. Energies 2023, 16, 3124. https://doi.org/10.3390/en16073124

[9] Тутыгин В. Искусственный интеллект в нефтегазовой индустрии как фактор развития производственной системы //Science and innovation. 2023. Т. 2. №. Special Issue 3. С. 168-171.

[10] Levin Maxim & Nagornov Stanislav & Levina Ekaterina & Levina Lyubov & Kovalenko Irina. (2022). The Concept Of "Smart Oil Storage" Based On Machine Vision Technologies. Science In The Central Russia. 94-101. DOI: 10.35887/2305-2538-2022-5-94-101.

[11] K.M. Alam and A. El Saddik, "C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems," in IEEE Access, vol. 5, pp. 2050-2062, 2017, DOI: 10.1109/ACCESS.2017.2657006.

[12] Knebel F.P., Trevisan R., do Nascimento G.S., Abel M., & Wickboldt J.A. A study on cloud and edge computing for the implementation of digital twins in the Oil & Gas industries // Computers & Industrial Engineering. 2023. (182). С. 109363.

[13] Современные технологии. Киберфизические системы: учебное пособие / Авт.- сост. Е.И. Громаков, А.А. Сидорова; Томский политехнический университет. Томск: Изд-во Томского политехнического университета, 2021. 166 с.

[14] Abramkin S.E., Dushin S.E., Pervukhin D.A. Problems of development of control systems of gas-producing complexes. Journal of Instrument Engineering. 2019. Vol. 62, N 8. P. 685—692 (in Russian). DOI: 10.17586/0021-3454-2019-62-8-685-692

[15] Asadulagi M.-A.M., Pershin I.M., Tsapleva V.V. Research on Hydrolithospheric Processes Using the Results of Groundwater Inflow Testing. Water 2024, 16, 487. https://doi.org/10.3390/w16030487

[16] Olga Afanaseva, Oleg Bezyukov, Dmitry Pervukhin, Dmitry Tukeev. Experimental Study Results Processing Method for the Marine Diesel Engines Vibration Activity Caused by the Cylinder-Piston Group Operations. Inventions 2023, 8(3), 71; https://doi.org/10.3390/inventions8030071

[17] Singh M., Srivastava R., Fuenmayor E., Kuts V., Qiao Y., Murray N., & Devine D. Applications of Digital Twin across Industries: A Review // Applied Sciences. 2022. (12). С. 5727.

[18] Jones David & Snider Chris & Nassehi, Aydin & Yon, Jason & Hicks, Ben. (2020). Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology. 29. DOI: 10.1016/j.cirpj.2020.02.002.

[19] Răileanu S., Borangiu T., Ivănescu N., Morariu O., & Anton T. (2020). Integrating the digital twin of a shop foor conveyor in the manufacturing control system. In T. Borangiu, D. Trentesaux, P. Leitão, A. Giret Boggino, & V. Botti (Eds.), Service oriented, holonic and multi-agent manufacturing systems for industry of the future. SOHOMA 2019. studies in computational intelligence (Vol. 853, pp. 134–145). Cham: Springer.

[20] Redelinghuys A.J.H., Basson A.H., & Kruger K. ‘‘A six-layer architecture for the digital twin:a manufacturing case study implementation,’’ Journal of IntelligentManufacturing, 2020.

[21] A. De Benedictis, N. Mazzocca, A. Somma, and C. Strigaro, “Digitaltwins in healthcare: an architectural proposal and its application in asocial distancing case study,” IEEE Journal of Biomedical and HealthInformatics, pp. 1–12, 2022

[22] S. Singh, M. Weeber, and K.-P. Birke, “Advancing digital twin implementation: a toolbox for modelling and simulation,” Procedia CIRP,vol. 99, pp. 567–572, 2021, 14th CIRP Conference on IntelligentComputation in Manufacturing Engineering, 15-17 July 2020.

Анализ сетевых киберпреступлений

Р. Р. Фаткиева[1], А. С. Судаков[1], Д. О. Дедов[2]

[1]*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)*
[2]*Санкт-Петербургский государственный университет аэрокосмического приборостроения*

rikki2@yandex.ru, asudakov.mail@gmail.com, dedovdanil3@yandex.ru

Аннотация. **Сетевые киберпреступления представляют собой серьезную угрозу для информационной безопасности в современном цифровом мире. В статье приведен анализ особенностей сетевых киберпреступлений и предложен метод для их анализа. К отличительным особенностям метода можно отнести построение цифрового следа атаки с использованием ориентированного графа, а также цифрового профиля нарушителя. Для этого осуществляется фиксация состояния киберфизической системы в нормальном режиме и выявление момента обнаружения атаки, а также сбор и анализ данных из журналов событий для выделения событий, приводящих к атаке. Построение графа атаки позволяет идентифицировать типичные сценарии атак и характеристики злоумышленников. Предложенный метод позволяет повысить эффективность обнаружения и противодействия сетевым киберпреступлениям, сократить время реакции на инциденты и улучшить уровень безопасности информационных систем.**

Ключевые слова: сетевые киберпреступления, цифровой след атаки, цифровой профиль нарушителя, моделирование инцидента, журналы событий, граф атаки

I. Введение

В современном информационном обществе сетевые технологии играют ключевую роль во многих аспектах нашей повседневной жизни. Однако вместе с их ростом и распространением возрастает и уровень угроз со стороны киберпреступников, которые используют сети для совершения различных преступлений и атак на цифровые системы. В связи с этим, анализ сетевых киберпреступлений становится критически важным для обеспечения безопасности информации и защиты от киберугроз [1]. К отличительным особенностям расследования киберпреступлений можно отнести [2–9]:

- необходимость анализа больших объемов криминалистически значимой информации;
- отсутствие цифрового следа для ряда операций или транзакций;
- необходимость построения цифрового профиля злоумышленника для идентификации образов и закономерностей в сценариях его поведения;
- высокую латентность со стороны пострадавшего;
- необходимость использования цифровой форензики;
- глобализм в источнике атаки и возможность анонимизации в распределенных элементах, участвующих в атаке;
- быстрая эволюция сетевых атак и автоматизация их проведения.

Анализ основных тенденций, методологических подходов, которые исследователи используют для расследования киберпреступлений, представлен в [2]. В [3] показано, что нарушители используют анонимность, препятствуют установлению связей между цифровыми учетными записями и реальными удостоверениями личности. Для решения этой проблемы в [4] затронута задача обнаружения нарушений в облачной инфраструктуре. Предложена интеллектуальная система, формирующая с заданной периодичностью снимок состояний виртуальных машин, расположенных в облачном сервисе, с отправкой их на сервер Trusted Center Server (TCS) для хранения. В случае обнаружения нарушения производится извлечение снимка и его анализ для извлечения доказательств и реконструкции сценария преступления. Предлагаемая система реализована в качестве примера, где криминалистика выступает как услуги (FaaS) для выполнения процесса цифрового расследования за счет использования огромных возможностей облачных вычислительных ресурсов, таких как обработка, вычисления и хранение. Однако такой подход не предусматривает автоматического построения правил обнаружения вторжений, поэтому в работах [5, 6] предложено использовать онтологию цифровых нарушений, что облегчает обнаружение аномалий и автоматизирует обработку следов цифровых доказательств.

Работы [7, 8] посвящены разработке архитектур систем предупреждения на ранней стадии для расследования преступлений с помощью различных процессов сбора и обработки данных и применения инструментов извлечения знаний.

Исследования характеристик сетевых кибернарушений и методов их обнаружения представлены в [9, 10]. Для выявления нарушений в [9] предложено использовать нейронные сети, а в [10] метод растущих пирамидальных сетей. К преимуществам метода можно отнести сочетание статистических и сигнатурных методов обнаружения атак и возможность реконфигурации сетевой инфраструктуры при изменении структурной и функциональной динамики, обусловленной перераспределением соединений между вершинами сети. К недостаткам использование вычислительных ресурсов и длительность анализа.

Методы построения профилей нарушителя с использованием методов кластерных вычислений

рассмотрены в работах [11, 12]. В [11] анализируются 25 атрибутов для обнаружения киберпреступника с помощью различных методов, таких как гауссовская кластеризация, K-средние, нечеткие C-средние и нечеткая кластеризация. В [12] дан систематический обзор литературы по профилированию в качестве основы для разработки методов киберповеденческого анализа.

Обобщая работы [1–12] можно отметить, что разработка систем анализа киберпреступлений является актуальной. В рамках этого целесообразно осуществить интеграцию различных методов обнаружения и анализа киберпреступлений, а также осуществить разработку более эффективных методов построения профилей нарушителей. Сочетание различных подходов значительно улучшит процессы предотвращения и расследования киберпреступлений.

II. Модель сетевых киберпреступлений

Модель инцидента можно представить кортежем

$$M =< M_{\text{КС}}, M_N, M_{\text{A}} >, \quad (1)$$

где $M_{\text{КС}}$ – модель атакуемой киберфизической системы; M_N – модель нарушителя; M_{A} – модель атаки.

В этом случае для выявления инцидентов нарушения безопасности необходимо осуществить анализ точек вхождения в киберфизическую систему, выявить цифровой след сетевой атаки и на основании этого сформировать цифровой профиль злоумышленника или группы нарушителей. Рассмотрим каждый из этапов более подробно.

2.1 Построение модели киберфизической системы и сбор журналов событий.

Для формирования цифрового следа атаки на первом шаге необходимо зафиксировать состояние исследуемой киберфизической системы в нормальном режиме функционирования и временной момент выявления атаки. Это позволит зафиксировать интервал времени, на котором осуществлялась атака, за счет выявления отклонений от нормального режима работы.

$$T_A =< t_{start}, t_{stop} >, \quad (2)$$

где t_{start}. – время начало атаки; t_{stop} – время окончания атаки.

На втором шаге, на интервале времени (2) целесообразно осуществить сбор и анализ данных из журналов событий. Пусть E представляет собой множество всех событий, содержащих информацию о действиях, происходящих в киберфизической системе. Тогда каждое событие e_i можно представить в виде кортежа $e_i =< t_i, s_i, a_i >$, где t_i – временная метка события; s_i – источник события (например, сервер, рабочая станция); a_i – действие или событие (например, попытка доступа к диску, изменение файлов).

Задача анализа заключается в выделении событий, которые могут привести к атаке: $E_A = \{e_i \in E | t_i \geq t_{start}, (\widetilde{s_i}, \widetilde{a_i})\}$, где $\widetilde{s_i}$ – источник события, приводящий к атаке; $\widetilde{a_i}$-событие или действие, приводящее к атаке.

2.2 Построение цифрового следа атаки

Для построения цифрового следа атаки возможно использовать так называемые графы атак и/или сценарии атак. Тогда каждому уникальному событию нарушения $\widetilde{a_i}$ из множества E_A ставится в соответствие вершина v_i в графе атак G. Далее из множества найденных вершин необходимо зафиксировать все возможные совместные события. Для этого для каждой пары события (a_i, a_j), где a_i и a_j имеют определенную зависимость (например, временную $t_i < t_j$) выделяются вершины (v_i, v_j) которые соединяются ребрами. При этом если выявлено, что a_i и a_j имеют временную зависимость $t_i < t_j$, то целесообразно строить ориентированный граф, который лучше отражает последовательность нарушений безопасности. Для отражения этого также возможно использовать вес ребра между этими событиями, отражающий степень их временной зависимостей. Для этого обозначим вес ребра w_{ij} между событиями e_i и e_j.

В качестве альтернативного подхода допускается определить вес исходя из вероятности дуги и/или важности каждого события для процесса атаки.

Процедура является интеграционной и повторяется до тех пор, пока не будут рассмотрены все возможные совместные события. Таким образом, введение весов к ребрам графа позволяет учитывать временные связи между событиями и их важность для анализа структуры атаки и выявить зависимости между различными компонентами атаки.

2.3 Построение цифрового профиля нарушителя:

Получение множества вершин в графе атаки G позволяет для каждой вершины v_i сформировать полный набор характеристик, идентифицирующих знания и умения нарушителя(ей), для проведения атаки, который можно представить в виде множества:

$$\mathrm{C} = (\mathrm{c}_{i1}, \mathrm{c}_{2i}, \ldots, \mathrm{c}_{ij}, \ldots, \mathrm{c}_{in}), \quad (3)$$

где c_{ij} – характеристика знаний и умений нарушителя, например такая, как умение эксплуатировать тип использованной уязвимости, метода атаки, уровень привилегий и т. д.

На этом этапе также целесообразно рассмотреть информацию о возможных используемых инструментах злоумышленника. Это можно представить в виде множества:

$$Tl = (tl_{i1}, tl_{2i}, \ldots tl_{ij} \ \ldots, tl_{iz}), \quad (4)$$

где tl_{ij} – множество инструментов, используемых нарушителем в вершине графа.

Таким образом, введение информации об инструментах злоумышленника позволяет учитывать использованные средства при формировании цифрового профиля нарушителя.

Выделение из множества (3) отдельных поведенческих подмножеств позволяет сформировать набор шаблонов действий, идентифицирующих нарушителя, который определяет типичные последовательности действий или паттерны атаки:

$$X_N = (x_1, x_1, \dots, x_l, \dots, x_k), \quad (4)$$

где x_l – последовательность действий, применяемых нарушителем для реализации сетевой атаки.

Выделение множеств (3–5) позволяет определить цифровой профиль нарушителя M_N как набор характеристик C и поведенческих шаблонов X, представленных в виде:

$$M_N = (C, X, Tl). \quad (5)$$

Модель (5) позволяет описать характеристики и поведение потенциального злоумышленника на основе анализа графа атаки и других источников информации о кибератаках, а также сформировать характеристики для отнесения тех или иных особенностей поведения к определенному кластеру. С другой стороны, это позволит также соотнести профиль нарушителя с конкретным кластером атак или кластером применяемых инструментов, что позволит лучше понять его характеристики и методы действий.

Рассмотрим возможность построения сценария атаки на примере [10] атаки навязывания ложного маршрута за счет несанкционированного изменения маршрутно-адресных данных как внутри сети, так и во внешних сетях. Возможность навязывания ложного маршрута обусловлена недостатками, присущими алгоритмам маршрутизации (в частности, из-за проблемы идентификации сетевых управляющих устройств).

К основному элементу выявления можно отнести изменение исходной маршрутизации на объекте, при которой новый маршрут проходит через ложный объект. Атака осуществляется путем несанкционированного изменения маршрутно-адресных данных как внутри сети, так и во внешних сетях. К характеристикам изменений сетевых настроек можно отнести: изменение IP-адресов устройств, входящих в сеть; используемые в сети протоколы маршрутизации; изменения маски сети; метрики маршрута; используемые на хостах порты; IP-адреса маршрутизаторов; времени передачи данных.

В общем виде при обнаружении нарушения возможно обнаружить и зафиксировать t_{stop} – время окончания атаки. Исходя из этого сформируем конечное событие как $e_n = < \widetilde{t_n}, \widetilde{s_n}, \widetilde{a_n} >$, где $t_n = t_{stop}$, $\widetilde{s_n}$ – хост, $\widetilde{a_n}$ – задержка в передаче данных по сети. Предыдущим по отношению к событию e_n является событие e_{n-1}, которое характеризуется кортежем $e_{n-1} = < \widetilde{t_{n-1}}, \widetilde{s_{n-1}}, \widetilde{a_{n-1}} >$, где s_{n-1} – маршрутизатор сети, через который должен протекать маршрут в нормальном режиме функционирования сети; a_{n-1} – нарушение маршрута. Далее итерационно рассматривая весь путь прохождения пакета по логам работы маршрутизаторов можно зафиксировать все нарушения прохождения сетевого трафика, и построить граф атак. Конечной точной такого графа будет событие, характеризующее начало атаки с временной отметкой. Анализируя все собранные логи и соединяя последовательно все вершины событий, получим ориентированный граф, представленный на рис. 1, характеризующийся t_{start}. – временем начало атаки и событиями e_n – обнаружение в задержке по времени передачи данных; e_{n-1} – анализ пропускной способности сети; e_{n-2} – обнаружение нарушения маршрутизации пакета; e_{n-3} – обнаружение отсутствие/нарушения маршрута в таблице маршрутизации; e_{n-4} – изменение таблиц маршрутизации; e_{n-5} – фиксация формирования сообщения об изменении таблицы маршрутизации; e_{n-6} – установление связи с маршрутизатором хоста с несанкционированным IP.

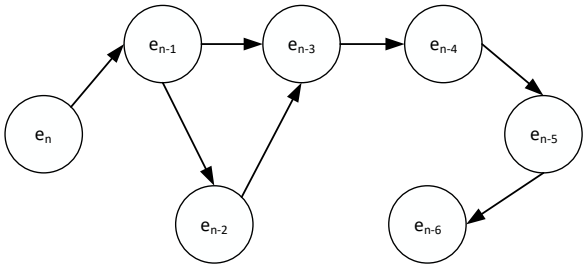


Рис. 1. Граф событий атаки навязывания ложного маршрутизатора

Выявление начального события позволяет зафиксировать время завершения атаки t_{stop} в вершине графа e_{n-6} и перейти к построению цифрового следа атаки. Для этого осуществим противоположное движение по полученному ранее ориентированному графу. Анализ события e_{n-6} позволяет предположить, что для возможности формирования связи с маршрутизатором злоумышленнику требуется как минимум выяснить IP адрес маршрутизатора, и установить с ним связь. В этом случае полученный на рисунке граф целесообразно дополнить вершинами (рис 2), где v_1 – установка программного обеспечения для осуществления атаки; v_2 – перехват трафика; v_3 – получение IP адресов сетевых устройств; v_4 – получение доступа к сетевому устройству (получение пароля маршрутизатора); v_5 – идентификация протоколов маршрутизации и сообщений об ошибках, используемых в сети; v_6 – изменение таблиц маршрутизации; v_7 – «замыкание трафика на себя»; v_8 – анализ трафика на устройстве нарушителя.

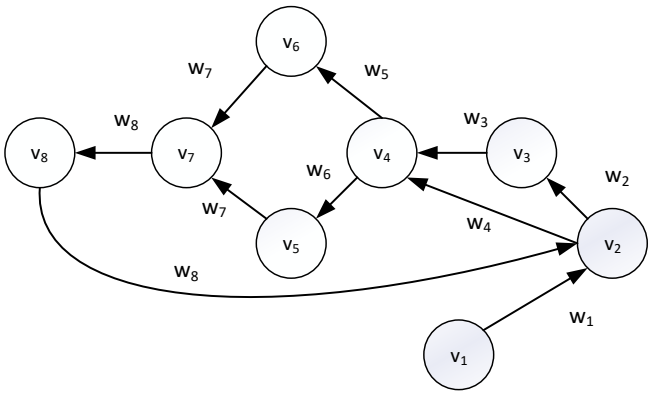


Рис. 2. Граф атаки навязывания ложного маршрутизатора

Оценка маршрутизации в графе (рис. 2) позволяет предположить цифровой след атаки в виде «множества путей передвижения на графе», в зависимости от аргумента ребра w_i и анализа событий зафиксированных в журналах логов и графе на рис 1. Например, в общем случае, анализируя граф атаки (рис. 2) возможно

выделить следующие варианты маршрутов исходя из ребер:

Вариант 1: w_1, w_2, w_3, w_5, w_7, w_8.

Вариант 2: w_1, w_4, w_5, w_7, w_8.

Вариант 3: w_1, w_2, w_3, w_6, w_7, w_8.

Тогда для каждого варианта можно выделить паттерны атаки, например для первого варианта: $X_N = (x_1(w_1), x_2(w_2), x_3(w_3), x_5(w_5), x_7(w_7), x_8(w_8))$.

Далее для каждого элемента из данного множества целесообразно выделить навыки и умения, которые требуются для данного паттерна атаки, что дает возможность получить множество (3). Выделение используемых инструментов позволяет перейти к формированию множества (4). Рассмотрения трех вариантов атаки, представленных ранее, дает возможность построить как минимум три варианта профиля нарушителя, для построения обобщенной базы профилей (5). Однако это не всегда позволяет идентифицировать личностные характеристики нарушителя. В этом случае целесообразно изучить отдельные личностные характеристики, сопоставляя события, происходящие в сети с реакцией нарушителя по логам, и построенным графам событий и атаки (рис. 1 и рис. 2). Например, выделение действий не входящих в (4), но выявленных по логам событий дает возможность выделить индивидуальные характеристики нарушителя, а в некоторых случаях идентифицировать наличие группы нарушителей.

Таким образом, полученная модель позволяет эффективно анализировать сетевые атаки и строить цифровые профили нарушителей, а предложенный подход обеспечивает комплексный анализ киберпреступлений и создание подробных цифровых профилей нарушителей, что способствует повышению уровня безопасности информационных систем.

III. Заключение

Построение цифрового следа атаки и цифрового профиля нарушителя позволяет идентифицировать типичные сценарии атак, выявлять характеристики и поведенческие шаблоны злоумышленников, а также эффективно реагировать на киберпреступления. К дальнейшим направлениям исследования целесообразно отнести разработку более сложных моделей анализа и предотвращения сетевых киберпреступлений, а также проведение эмпирических исследований на реальных кибератаках для оценки эффективности предложенных подходов.

Список литературы

[1] W.A. Al-Khater, S. Al-Maadeed, A.A. Ahmed, A.S. Sadiq and M.K. Khan, "Comprehensive Review of Cybercrime Detection Techniques," in IEEE Access, vol. 8, pp. 137293-137311, 2020, doi: 10.1109/ACCESS.2020.3011259.

[2] Jack Hughes, Sergio Pastrana, Alice Hutchings, Sadia Afroz, Sagar Samtani, Weifeng Li, and Ericsson Santana Marin. 2024. The Art of Cybercrime Community Research. ACM Comput. Surv. 56, 6, Article 155 (June 2024), 26 pages. https://doi.org/10.1145/3639362.

[3] Chetry A., Sharma U. Anonymity in decentralized apps: Study of implications for cybercrime investigations. International Journal of Experimental Research and Review (2023), 32, 195-205. https://doi.org/10.52756/ijerr.2023.v32.017.

[4] Hemdan E.ED., Manjaiah D. An efficient digital forensic model for cybercrimes investigation in cloud computing. Multimed Tools Appl 80, 14255–14282 (2021). https://doi.org/10.1007/s11042-020-10358-x.

[5] Sikos L.F. Artificial intelligence in Digital forensics: Ontology engineering for cybercrime investigation. WIREs Forensic Sci. 2021; 3: e1394. https://doi.org/10.1002/wfs2.1394.

[6] Ayo F.E., Awotunde J.B., Ogundele L.A. et al. Ontology-Based Layered Rule-Based Network Intrusion Detection System for Cybercrimes Detection. Knowl Inf Syst (2024). https://doi.org/10.1007/s10115-024-02068-9.

[7] Fernandez-Basso C., Gutiérrez-Batista K., Gómez-Romero J., Ruiz M.D., Martin-Bautista M.J. An AI knowledge-based system for police assistance in crime investigation. Expert Systems, (2024). e13524. https://doi.org/10.1111/exsy.13524.

[8] Baror Stacey, Adeyemi Ikuesan, Venter Hein. Functional Architectural Design of a Digital Forensic Readiness Cybercrime Language as a Service. European Conference on Cyber Warfare and Security. (2023). 22. 73-82. 10.34190/eccws.22.1.1240.

[9] Shichkina Y.A., Fatkieva R.R, Puzako I.A. Information Threat Recognition Method Using a Neural Network. 2022 III International Conference on Neural Networks and Neurotechnologies (NeuroNT), Saint Petersburg, Russian Federation, 2022, pp. 42-46, doi: 10.1109/NeuroNT55429.2022.9805531.

[10] Y.A. Shichkina, R.R. Fatkieva, "Detection of network attacks using of growing pyramid networks," 2021 10th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2021, pp. 1-4, doi: 10.1109/MECO52532.2021.9460188.

[11] Meena K., Veena K. Performance evaluation of cybercriminal detection through cluster computing techniques. J Ambient Intell Human Comput (2019). https://doi.org/10.1007/s12652-019-01605-7

[12] Martineau, M., Spiridon, E., Aiken, M. A Comprehensive Framework for Cyber Behavioral Analysis Based on a Systematic Review of Cyber Profiling Literature. Forensic Sci. 2023, 3, 452-477. https://doi.org/10.3390/forensicsci3030032

Анализ и моделирование vampire-атак в самоорганизующихся беспроводных сенсорных сетях

В. А. Десницкий

Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук»

desnitsky@comsec.spb.ru

И. В. Котенко

Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук»

ivkote@comsec.spb.ru

Аннотация. **Работа включает исследование вопросов информационной безопасности самоорганизующихся беспроводных сенсорных сетей (БСС). Объектом исследования являются vampire-атаки, присущие БСС и направленные на истощение энергоресурсов узлов, функционирующих от ограниченных автономных источников энергопитания. Такая атака базируется на эксплуатации злоумышленником одного или нескольких скомпрометированных узлов, к которым у нарушителя имеется доступ. Атака включает нарушение процессов маршрутизации в самоорганизующейся БСС со злонамеренным изменением фактических маршрутов следования пакетов данных. Тем самым нарушитель заставляет узлы расходовать значительно больше энергоресурса. Это в свою очередь приводит к значительно более быстрому разряду батарей, в результате чего сеть перестает полноценно функционировать, и предоставляемые ею прикладные сервисы оказываются недоступными конечным пользователям. В частности, такие последствия становятся особенно критичными в случае сетей, функционирующих в рамках важных инфраструктурных объектов, производств непрерывного цикла, на транспорте. В работе предлагается обобщенная имитационная модель vampire-атак, основанная на правилах и пригодная при имплементации широкого спектра БСС. Модель оценивается на примере фрагмента прототипа самоорганизующейся беспроводной сенсорной сети с узлами Digi XBee серии 2. По результатам моделирования сформулированы основные выводы.**

Ключевые слова: беспроводная сенсорная сеть; безопасность; vampire-атака; анализ; моделирование

I. Введение

В настоящее время беспроводные сенсорные сети получают все большее распространение в различных прикладных областях, таких как сети соединенных автомобилей, дронов, коммуникационные системы для поддержки операционных процессов на производствах, в торговых и складских помещениях, на территориях морских портов и пр. Ввиду критически важного характера таких систем в условиях работы в потенциально ненадежном и не доверенном окружении, особого внимания заслуживают вопросы информационной безопасности таких систем. В частности, скомпрометировав один или несколько узлов БСС, атакующий может осуществить различные атаки, направленные на нарушение аутентичности циркулирующих по сети данных и нарушение доступности устройств.

Наличие в таких системах автономных модулей, формирующих мобильные и перемещаемые в пространстве узлы БСС, обуславливает подверженность таких устройств атакам истощения энергоресурсов. При помощи таких атак злоумышленник воздействует на доступные ему проводные и беспроводные коммуникационные интерфейсы узлов сети, другие аппаратные модули БСС, на хранимые и передаваемые по сети данные, на среду окружения сети, на пользователей для единовременного, скачкообразного или постепенного снижения заряда батарей атакуемых узлов. Помимо прямого воздействия, такие атаки, называемые атака типа Denial-of-Sleep, могут осуществляться опосредованно, например, путем периодической отправки на узел-жертву запросов, нарушающих процесс периодического нахождения узла в режиме экономного энергопотребления [1]. Сложность идентификации такого воздействия связана с тем, что оно осуществляется не напрямую, а с вовлечением других легитимных узлов сети. При этом трафик, исходящий от каждого такого узла, сам по себе сложно идентифицируем в качестве атакующего, тогда как группа таких узлов формирует атаку в совокупности.

Еще в качестве одного характерного примера разновидности атак истощения энергоресурсов можно выделить vampire-атаки, которые направлены на постепенное исчерпание заряда группы узлов путем эксплуатации уязвимостей протоколов маршрутизации [3]. Модификация полей пакета в процессе его ретрансляции на скомпрометированном промежуточном узле позволяет потенциально неограниченное число раз перенаправлять этот пакет по тем же маршрутам повторно, тем самым истощая энергозаряд автономно функционирующих узлов, располагающихся по данному маршруту. И в общем случае выявление таких атакующих пакетов представляется сложно осуществимой задачей ввиду изменчивости таких пакетов и их маскировки под доброкачественный трафик штатных прикладных сервисов сети. Таким образом, для

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00486, https://rscf.ru/project/24-21-00486/.

решения задач обнаружения vampire-атак и повышения защищенности от них необходимы дополнительные исследования данного вида атак, включающее анализ их существующих и возможных особенностей и разновидностей, а также их моделирование, что необходимо для повышения их детектируемости.

В работе проведен анализ существующих моделей и решений по моделированию vampire-атак в БСС. Предлагается обобщенная имитационная модель vampire-атак, основанная на правилах и пригодная для имплементации для широкого класса беспроводных сенсорных сетей, поддерживающих программно определяемые процессы маршрутизации с использованием современных сетевых протоколов БСС, таких как ZigBee, Wi-Fi, LPWAM-протоколы, в том числе Sigfox, LoRa и др. Модель апробирована на примере фрагмента прототипа самоорганизующейся беспроводной сенсорной сети с узлами Digi XBee серии 2. По результатам моделирования сформулированы основные выводы. К отличительным элементам новизны предлагаемой модели относится ее универсальный характер, предполагающий ее применимость для широкого класса беспроводных БСС, способных функционировать в различных условиях и окружениях, а также позволяющий моделировать различные вариации атаки, в том числе, эксплуатирующие уязвимости используемых протоколов сетевого уровня и функций маршрутизации, в частности.

II. Анализ предметной области

Vampire-атаки, проводимые на сетевом уровне взаимодействия, являются в достаточной степени универсальными и применимы к множеству различных протоколов функционирования БСС [3]. Такие атаки, как правило, базируются на недостаточной защищенности протокола на сетевом уровне и эксплуатируют уязвимости протокола, связанные в том числе с нарушением целостности адресов, выполненных хопов пакета, меток времени и других данных, содержащихся в служебных полях сетевого протокола. При этом ряд исследований подтверждает достаточно высокую эффективность таких атак, выражаемую при помощи показателей скорости истощения батареи атакуемого узла, в количестве таких одновременно атакуемых узлов, а также в скрытности такой атаки [4], [5].

В рамках существующих или проектируемых БСС принципиально важной оказывается необходимость оценки потенциального ущерба от применения vampire-атак, учитывающей как особенности конкретного используемого беспроводного протокола, так и характеристик самой сети. В зависимости от этого должны выбираться и применяться, как контрмеры, так и средства обнаружения, позволяющие выявить необходимость применения какой-либо контрмеры [6].

Однако стоит обращать внимание на комплексный характер vampire-атак, в процесс выполнения которых может вовлекаться различное число уязвимых узлов, которые нарушитель способен эксплуатировать. Кроме того, атака может учитывать такие характеристики как ширину коммуникационного канала, разветвленность и динамизм сетевой топологии, величину средней передаваемой полезной нагрузки, состав аппаратной части узлов, энергоресурсы узлов, а также свойства самоорганизации и децентрализации БСС. Поэтому оценка эффективности vampire-атак в конкретной БСС с заданным сценарием функционирования зачастую оказывается проблематичной и практически невыполнимой при помощи исключительно лишь средств аналитического исследования.

В результате этого возникает потребность в проведении натурных исследований таких атак, что в свою очередь, как правило, затруднено из-за технической сложности натурного моделирования такой атаки с необходимостью одновременного скоординированного воздействия на ряд узлов, а также воспроизведением репрезентативных экспериментальных условий нормального функционирования БСС с заданной сетевой нагрузкой и проведением измерений. Также часто оказывается, в особенности в случае работы БСС в рамках критически важных инфраструктур, нет ни юридической, ни фактической возможности натурного тестирования моделей атак на имеющейся аппаратной инфраструктуре, штатная работа которой не может быть нарушена или приостановлена. Все это определяет потребность в проведении имитационного моделирования vampire-атак, которое бы являлось, во-первых, настраиваемым под различные виды сетей, сценариев, различный состав сетей, различные виды беспроводных технологий, во-вторых, адекватно отражало процессы функционирования сети под атакой и, в-третьих, было бы практически выполнимым с точки зрения технических ограничений и компетенций, а также затрачиваемых на него ресурсов [7]. Поэтому, все это позволяет сделать вывод о высокой актуальности и важности решения задач настоящего исследования.

III. Моделирование БСС и атак

Для имитационного моделирования атаки в первую очередь необходимо провести моделирование нормального поведения БСС. Не умаляя общности процесса моделирования, рассматриваем следующую сетевую топологию БСС, состоящую из пяти узлов и представляемую при помощи графа. В каждой вершине графа располагается некоторый узел сети, а линии между вершинами обозначают имеющиеся физические беспроводные двунаправленные каналы связи. Каждый канал связи предполагает ровно двух абонентов.

Нормальное сетевое поведение узлов БСС представляет собой регулярную отправку показаний сенсоров в сеть по заданному адресу, отправку и получение служебных сетевых команд, подтверждающих живучесть узла и формирующих другие системные характеристики. В рамках проводимого моделирования такое поведение узлов задается при помощи генератора псевдослучайных чисел на основе случайной величины с нормальным распределением путем задания математического ожидания *mu*, описывающего усредненное значение промежутка времени, через которое узел будет отправлять очередное сообщение в сеть и *si* – среднеквадратичное отклонение. Такое распределение позволяет смоделировать в первом приближении наиболее характерное усредненный временной интервал

с учетом рандомизации фактического формирования сообщений в заданных временных рамках.

В рамках проводимого моделирования для пяти узлов используются следующий вектор значений

$$\{(mu_t, si_t)\}_{t=1..5} = \{(A_{mu}(1), A_{si}(0.1)), (B_{mu}(2), B_{si}(0.1)), (C_{mu}(-), C_{si}(-)), (D_{mu}(2), D_{si}(0.2)), (E_{mu}(-), E_{si}(-))\},$$

где узлы помечены буквами алфавита от *A* до *E*, а числовые значения заданы в секундах. При обозначении математического ожидания и среднеквадратичного отклонения дефис обозначает отсутствие событий генерации сообщений на данном узле. Возможные нулевые значения случайной величины игнорируются в процессе моделирования, тогда как отрицательные значения берутся по модулю. При этом для каждого узла *X*, генерирующего сообщения конечный адресат каждого сообщения *Y* определяется равновероятно из оставшихся четырех узлов сети с вероятностью $P(dest(X) = Y) = 0.25$. Старт процесса имитационного моделирования БСС производится от нулевого момента времени с естественной скоростью течения времени. В качестве полезной нагрузки нормальных пакетов данных используются произвольные данные фиксированной длины, что как-либо ощутимо не влияет на процесс моделирования. Для моделирования vampire-атаки узлы *C* и *D* моделируются как контролируемые атакующим. Предполагается, что атакующий имеет полный доступ к их таблицам маршрутизации и способен создавать и отправлять любые пакеты данных, перехватывать и модифицировать любые системные поля и полезную нагрузку любые сообщений, проходящих через данный узлы *C* и *D*.

Для реализации модели vampire-атаки нарушитель злонамеренно модифицирует правила маршрутизации на этих двух узлах следующим образом. Во всех ZigBee-пакетах, у которых в качестве конечного адреса указан адрес узда *C*, на узле *C* осуществляется подмена, адреса на узел *E* для того, чтобы пакет продолжил дальнейшее движение по сети на узел *E* через узел *D*. В зависимости от реализации протокола, для этого могут также корректироваться такие поля заголовков как sequence number пакета, хэш-сумма, метка времени и пр., чтобы пакет не было отброшен автоматически из-за того, что он превысил число хопов. Аналогичным образом, любой пакет пришедший на узел E с узла C подвергается замене адреса снова на адрес C и отправляется в качестве промежуточного узла на узел *A*. На рис. 1 схематично показана моделируемая сеть для случая БСС на основе узлов Digi XBee серии 2, причем узел *B* функционирует от автономного, ограниченного источника питания, и является узлом-жертвой vampire-атаки.

Происходит фактическое зацикливание пакета данных, и он способен пройти по кругу $C \rightarrow D \rightarrow E \rightarrow A \rightarrow B \rightarrow C$ подряд потенциально неограниченное число раз. Однако, для злоумышленника, чтобы снизить вероятность обнаружения нелегитимного трафика в сети, количество таких циклом может быть ограничено до нескольких штук в зависимости от условий и целей vampire-атаки. Кроме того, в целях скрытности атаки в качестве подобных циклически пересылаемых пакетов потенциальному нарушителю целесообразно использовать уже существующие легитимные пакеты, не создавая новых искусственно, чтобы простейшие механизмы безопасности, основанные на шаблонах разрешенных потоков данных, не выявили бы такой аномальный пакет.

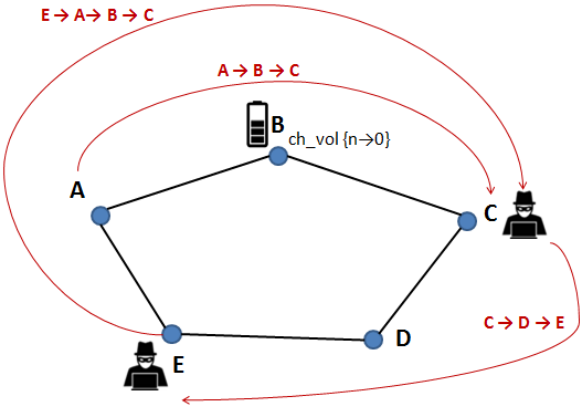


Рис. 1. Моделирование vampire-атаки в самоорганизующейся беспроводной сенсорной сети

Моделирование расхода заряда батареи автономно работающего узла *B* производится с использованием целочисленной переменной *ch_vol*, которой задается некоторое начальное значение $n \leq 1$ перед стартом процесса имитационного моделирования. Процесс изменения данной переменной с течением времени задается в виде правил вида

$$ch\_vol(e) \text{ -= } d,$$

где *e* определяет экземпляр события определенного вида, d задает усредненную величину изменения нормированного заряда батареи узла, тогда как символ '-=' обозначает бинарную операцию изменения левого операнда на величину значения правого операнда. Кроме того, независимо от происходящих в сети событий, в которые вовлечен узел *B*, применяется следующее правило

$$ch\_vol(t \mid t=1,2,...) \text{ -= } \Delta,$$

где *t* определяет дискретный счетчик времени, исчисляемый в секундах, инициализируемая значением 0 в момент запуска имитационной модели и увеличиваемый на 1 по прошествии каждой очередной секунды функционирования модели. Значение Δ задает константную усредненную величину уменьшения заряда в условиях фонового функционирования узла *B*. Отметим, что для запуска модели значения *d* и Δ устанавливаются экспертно.

IV. Реализация и дискуссия

Предложенная имитационная модель реализована с использованием языка Python, и в ее основу заложены модель состояния узлов сети, задающая узлы БСС, коммуникационные каналы, характеристики узлов, в том числе величину заряда, а также генерация и движение пакетов данных по сети. Реализованный комбинаторный алгоритм при помощи генераторов псевдослучайных чисел генерирует события, как нормального функционирования, так и события, инициируемые атакующим. Выходными данными разработанного

Python-скрипта, является лог, содержащий последовательности моделируемых событий, снабженных метками относительно времени, в том числе лог, получаемый в результате моделирования приведенной в разделе 3 vampire-атака с использованием скомпрометированных узлов *C* и *E*.

Результаты экспериментов по моделированию vampire-атак в условиях отличающейся интенсивности атаки приведены на рис. 2. В зависимости от значения показателя скрытности атаки *L*, варьирующего в диапазоне (0, 1) в зависимости от интенсивности воздействий, в каждой из итерации эксперимента вычислялась эффективность атаки. Максимальная скрытность атаки, соответствует значению 1, тогда как минимальное значение – 0. В последнем случае атака считается хорошо наблюдаемой, и, как предполагается, она высоковероятно может быть обнаружена. Эффективность выражается в виде отношения снижения величины заряда узла-жертвы, находящегося под атакой к величине снижения заряда данного узла при нормальной работе сети. Величины изменения заряда делимого и делителя рассматриваются за одинаковый промежуток времени, по умолчанию равный 1 сек. К особенностям осуществленного моделирования можно отнести довольно низкие затраты на вычислительные ресурсы, позволяющие его проводить на типовом пользовательском персональном компьютере.

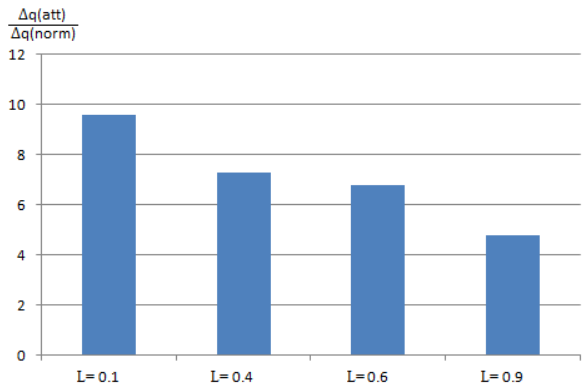


Рис. 2. Оценка эффективности vampire-атаки

Проведенные эксперименты подтверждают выполнимость имитационного моделирования vampire-атак без необходимости вовлечения каких-либо значительных вычислительных мощностей. Вместе с тем моделирование подтверждает потенциально высокую эффективность таких атак при различных величинах скрытности атаки. Поэтому, в частности, такую атаку целесообразно учитывать в рамках комплексных моделей атак в самоорганизующихся децентрализованных БСС. Возможность динамического перераспределения функций сбора, обработки данных, а также функций управления сетью открывают широкие возможности для атакующего злонамеренно эксплуатировать автономно работающие узлы БСС для значительно более быстрого исчерпания энергоресурса.

В целом vampire-атакам в особенности оказываются подверженными и наиболее критичными БСС в областях, где киберфизические устройства, такие как БПЛА, оказываются в тесном сопряжении с информационно-вычислительными процесса сети. К значимым последствиям успешной vampire-атаки можно отнести не только внезапная приостановка пользовательских сервисов, которые такой БПЛА предоставляет, но также и физическое крушение самого дрона, его разрушение и возможное причинение вреда окружающей инфраструктуре сети [8]. Ввиду того, что vampire-атака проявляется на сетевом уровне сетевого взаимодействия, то к перспективным способам предотвращения таких атак, не требующего для этого значительных ресурсов узлов, или, как минимум, к способам повышения сложности осуществления таких атак, можно отнести повышение защищенности протоколов маршрутизации в БСС. В частности, возможно встраивание в протокол проверок неизменности маршрута пакета данных на протяжении его передачи. В частности, для этого может использоваться технология блокчейна, но требования к ресурсам БСС должны быть дополнительно изучены.

V. Заключение

В рамках дальнейшей работы по данному направлению планируется расширение экспериментальной части моделирования vampire-атак с целью получения расширенного набора выходных логов, которые будут обладать высокой вариативностью, а также репрезентативностью для задач построения классификаторов, как средств программного обнаружения vampire-атак.

Список литературы

[1] Balueva A., Desnitsky V., Ushakov I. Approach to detection of denial-of-sleep attacks in wireless sensor networks on the base of machine learning // Studies in Computational Intelligence. 2020. Т. 868. С. 350-355.

[2] Hatfield J.W., Kominers S.D. A Simple Theory of Vampire Attacks // SSRN-Elsevier, 2023. https://ssrn.com/abstract=4377561. DOI: 10.2139/ssrn.4377561.

[3] Channawar P.M., Chavan Y.V. Vampire Attack: Energy Efficient Trust Based Solution // International Journal of Science and Research (IJSR). 2012. Т. 3, Вып. 12. С. 314-317.

[4] Juneja V., Dinkar S.K. An Approach against Vampire Attack for Successful Transmission in Wireless Sensor Network // 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT). 2023. С. 1-7.

[5] Alkwai L., Aledaily A., Almansour S., Alotaibi S., Yadav K., Lingamuthu V. Vampire attack mitigation and network performance improvement using probabilistic fuzzy chain set with authentication routing protocol and hybrid clustering-based optimization in wireless sensor network // Hindawi, Mathematical Problems in Engineering. 2022. № 4948190. С. 1-11.

[6] Srikaanth P.B., Nagarajan V. Fuzzy rough set derived probabilistic variable precision-based mitigation technique for vampire attack in MANETs // Wireless Personal Communications. Springer. 2021. Т. 121. С. 1085–1101.

[7] Verma V., Jha V.K. Detection and prevention of vampire attack for MANET // Nanoelectronics, Circuits and Communication Systems. Lecture Notes in Electrical Engineering, Springer. 2021. Т. 692. С. 81–90.

[8] Desnitsky V., Kotenko I. Simulation and assessment of battery depletion attacks on unmanned aerial vehicles for crisis management infrastructures // Simulation Modelling Practice and Theory. 2021. № 102244. С. 107.

Исследование скорости определения пожара в производственном помещении с применением системы «Умное производство» на основе технологии Интернета вещей

Е. Д. Григорьева[1], В. А. Ушаков[2]

Санкт-Петербургский государственный университет аэрокосмического приборостроения

[1]lizarapne@gmail.com, [2]ushakov@guap.ru

Аннотация. **Для снижения затрат на содержание производственного помещения была разработана система «Умное производство», содержащая в себе компонент пожарной безопасности. В статье рассматривается скорость обнаружения пожара разработанной системой на основе результатов исследования различных сценариев пожара.**

Ключевые слова: Интернет вещей; риск пожара; производственные помещения; скорость определения пожара; «Умное производство»

I. Введение

Интернет вещей (Internet of Things, IoT) [1–3] представляет собой объединенную сеть, к которой посредством коммуникационной и информационной инфраструктуры подключено множество объектов. Технология интернета вещей значительно расширяет возможности сбора и анализа информации, а применение методов кибернетики – науки об общих закономерностях получения, хранения, преобразования и передачи информации в сложных управляющих системах [4] – позволяют повысить эффективность применения технологии и открыть новые возможности.

Интернет вещей применяется в различных областях, в том числе и для систем охранно-пожарной сигнализации. Технология позволяет повысить эффективность и скорость реагирования при пожаре, оперативно выявляя возможную опасность [5–6].

Для снижения затрат на содержание производственного помещения была разработана система автоматизированного управления «Умное производство» основанная на технологии Интернета вещей [7–9]. Система обеспечивает улучшение организации труда и сокращение экономических затрат. Производственное помещение, для которого разрабатывается система, относится к категории умеренной пожароопасности (Г) [10], где установка систем пожарной безопасности не является обязательной. Однако статистика возгораний в производственных помещениях высока, а использование датчиков дыма не оптимально: из-за специфики работ высокое количество ложных срабатываний сигнализации и значительное число выходов из строя ее компонентов [11–12]. Применение технологии Интернета вещей в сочетании с интеллектуальной динамичной системной архитектурой, которая гарантирует непрерывный сбор и анализ данных, позволяет системе «Умное производство» определять пожары на начальном этапе возгорания без использования датчиков дыма, защищая персонал и снижая общие организационные риски.

II. Исследование скорости определения пожара в производственном помещении

Для достижения поставленной цели был проведен ряд экспериментов, направленных на выявление закономерностей изменения показаний различных датчиков при пожаре и его возникновении, и определения оптимальных параметров работы системы. Для расчета скорости определения пожара системой и вероятности эвакуации сотрудников было проведено исследование сценариев возникновения пожара в производственном помещении [13].

Исследование было произведено в программе RiskManager, позволяющей выполнить оценку пожарного риска с учетом динамического изменения опасных факторов пожара и времени эвакуации людей. Время блокировки путей эвакуации определяется по зонной модели расчета динамики опасных факторов пожара в начальной стадии [14–15].

План производственного помещения, пути эвакуации и расчетные точки представлены на рис. 1. Местоположение расчетных точек было выбрано исходя из расположения рабочих мест сотрудников (по два рабочих места для двух сотрудников в каждом помещении).

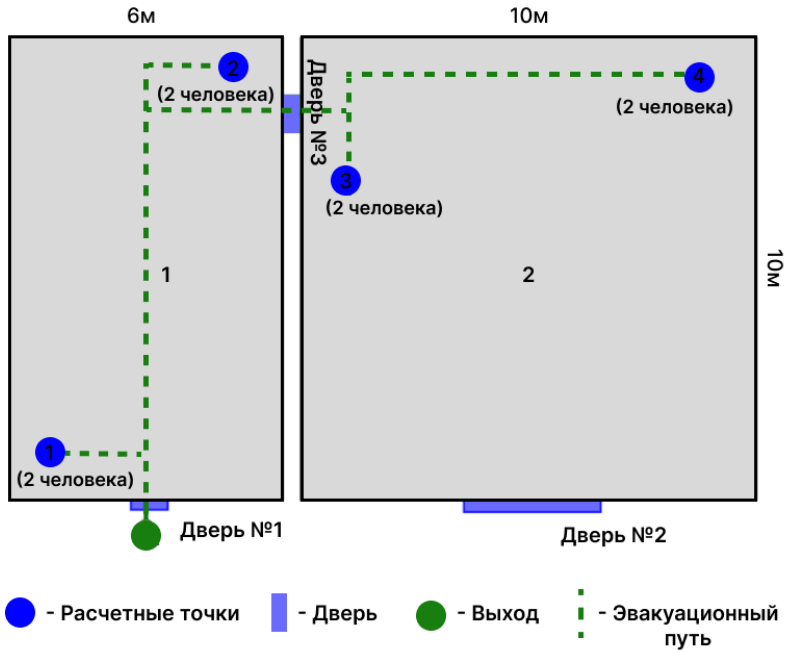


Рис. 1. План производственного помещения, пути эвакуации и расчетные точки

Было исследовано несколько сценариев пожара с возгораниями в разных помещениях при различных условиях доступности дверей. Время реагирования системы на пожар в среднем составляло 33 секунды. Из-за ограниченности размера статьи приведем результаты только двух сценариев.

При выполнении первого сценария «Возгорание в помещении 2 с открытыми дверьми» время блокирования эвакуационных путей составило 65 секунд, время срабатывания звукового сигнала, предупреждающего сотрудников об опасности, – 33 секунды, а время, требующееся на эвакуацию сотрудников – 25 секунд. В данном сценарии разница во времени необходимом на эвакуацию и временем блокирования путей было максимальным среди всех сценариев.

Рассмотрим подробнее результат выполнения второго сценария «Возгорание в помещении “1” с закрытой дверью №2», при котором время блокирования эвакуационных путей было наименьшим, а время срабатывания звукового сигнала – наибольшим. В табл. 1 представлены результаты выполнения сценария: время достижения критических значений основными факторами, опасными для здоровья человека и влияющими на процесс эвакуации. Из табл. 1 можно сделать вывод, что эвакуационный путь проходящий через помещение 1 (помещение с очагом пожара) наиболее опасен, так как время его блокирования составляет 58 секунд, а время потери видимости в помещении – 41 секунду. Время достижения очагом пожара размеров помещения 1 составило примерно 92 секунды. Динамика развития площади очага представлена на рис. 2. Исходя из полученных данных время эвакуации сотрудников из всех расчетных точек с учетом реагирования должно составлять меньше 92 секунд. За расчетное время эвакуации принимается максимальное время пути от расчетной точки до выхода. В данном сценарии общее расчетное время эвакуации при возгорании составило 31 секунду. Таким образом, время реагирования системы не должно превышать 61 секунду – разницу между временем достижения очагом пожара размеров помещения и временем на эвакуацию всех групп сотрудников [16–17].

ТАБЛИЦА I. РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ СЦЕНАРИЯ «ВОЗГОРАНИЕ В ПОМЕЩЕНИИ “1” С ЗАКРЫТОЙ ДВЕРЬЮ №2»

Контр. точка № / Помещение № / Высота раб. зоны	Время блокирования, сек.	По температуре, сек.	По потере видимости, сек.	По недостатку кислорода, сек.	По содержанию углекислого газа, сек.	По содержанию угарного газа, сек.	По содержанию хлороводорода, сек.	По тепловому потоку, сек.
1 / 1 / 1,7м (очаг)	58	65	41	93	200	200	56	6
2 / 1 / 1,7м (очаг)	58	65	41	93	200	200	56	6
3 / 2 / 1,7м	122	128	123	200	200	200	200	200
4 / 2 / 1,7м	122	128	123	200	200	200	200	200

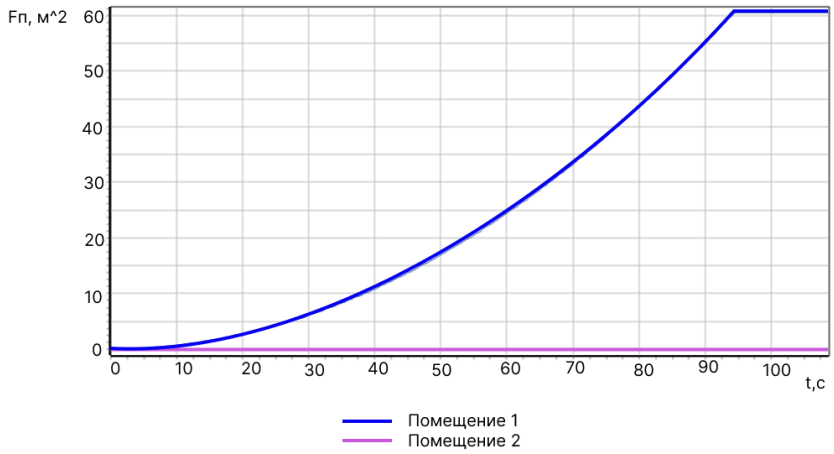


Рис. 2. Динамика развития площади очага пожара

В разработанной системе «Умное производство» пожарная сигнализация срабатывает при выполнении совокупности условий:

$$\begin{cases} n \geq 0{,}005\% \\ T > 30\ ℃ \\ L \leq 250\ лк \end{cases},$$

где n – концентрация угарного газа в воздухе, T – температура воздуха, L – освещенность.

Из рис. 3 можно сделать вывод, что задымленная зона достигнет уровня датчика угарного газа, расположенного на высоте 8 метров в помещении 1, спустя 10 секунд с момента начала пожара, датчика света (высота 3 метра) – спустя 35 секунд, а датчика температуры (высота 4 метра) – спустя 12 секунд. Температура газовой среды, в которой окажется датчик, превысит значение в 30℃ спустя 18 секунд с момента наступления пожара, что следует из рис. 4.

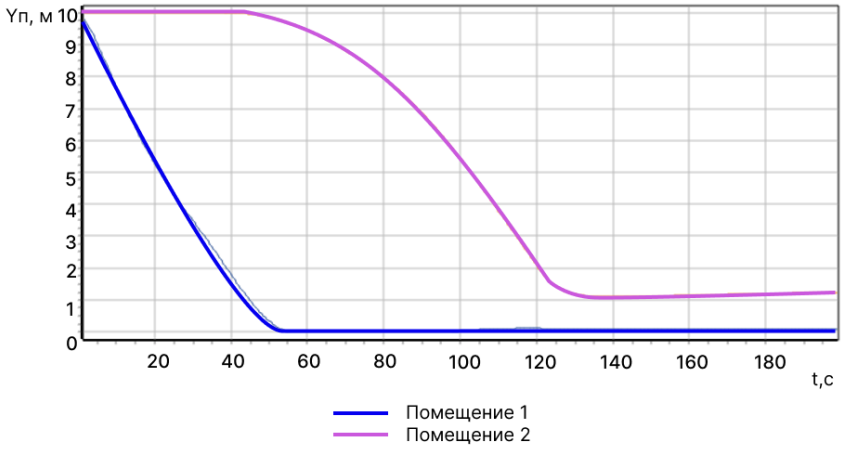


Рис. 3. График зависимости координаты задымленной зоны от времени

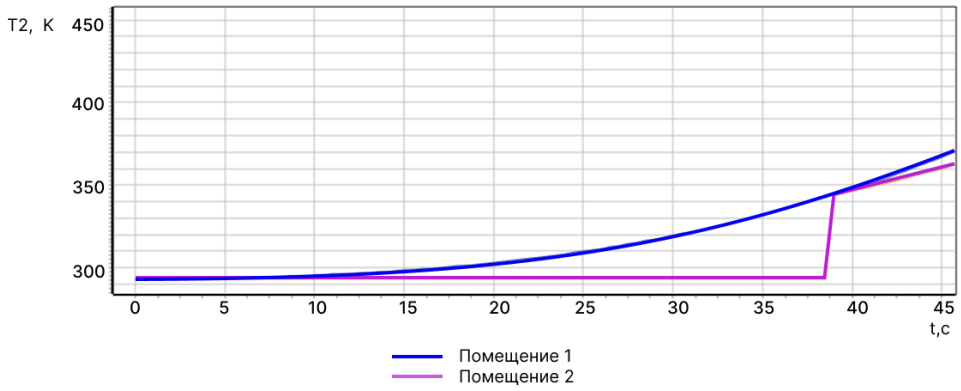


Рис. 4. График зависимости температуры газовой среды в задымленной зоне от времени

Таким образом, в выполненном сценарии условия срабатывания разработанной системы реагирования звуковой сигнал сирены уведомит сотрудников о начале возгорания спустя 35 секунд с момента начала пожара.

Вероятность эвакуации сотрудников рассчитывается по формуле:

$$P_m = \begin{cases} \frac{\tau_{бл} - t_p}{\tau_m}, \text{если } t_p < \tau_{бл} < (t_p + \tau_m) \\ 0{,}999, \text{если } t_p < \tau_m \leq \tau_{бл} \\ 0, \text{если } t_p \geq \tau_{бл} \end{cases},$$

где τ_m – время от момента обнаружения пожара до запуска звукового сигнала, t_p – время эвакуации в минутах, $\tau_{бл}$ – время блокирования путей эвакуации [18].

При расчете для групп сотрудников с минимальным временем блокирования эвакуационных путей выполняются условия $t_p < \tau_{бл} < (t_p + \tau_m)$. Вероятность эвакуации составила 81,5 %.

III. Заключение

По результатам исследования время определения пожара разработанной системой «Умное производство» при выполнении сценария с наименьшим временем блокирования эвакуационных путей и наибольшим временем реагирования системы составляет 35 секунд. Расчетная вероятность эвакуации сотрудников равна 81,5 % и позволяет говорить о применимости указанной системы для определения пожара в производственном помещении. Однако, необходимо выполнить пересмотр условий срабатывания пожарной сигнализации, который позволит приблизить вероятность эвакуации сотрудников к максимальному значению.

Список литературы

[1] P. R. Gunjal, S. R. Jondhale, J. Lloret Mauri, and K. Agrawal, Internet of things: Theory to practice. Boca Raton: CRC Press, 2024, doi: 10.1201/9781003282945.

[2] M. Taneja, “A mobility analytics framework for Internet of Things,” in 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015, doi: 10.1109/ICGCIoT.2015.7380440.

[3] B. Mostafa, “Monitoring internet of things networks,” in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, doi: 10.1109/WF-IoT.2019.8767203.

[4] Новиков Д.А. Кибернетика: Навигатор. История кибернетики, современное состояние, перспективы развития. М.: ЛЕНАНД, 2016. 160 с.

[5] P. Guo, “The application of Internet of Things technology in intelligent fire protection,” Applied and Computational Engineering, vol. 47, no. 1, pp. 159–163, 2024, doi: 10.54254/2755-2721/47/20241278.

[6] T. Peng and W. Ke, “Urban fire emergency management based on big data intelligent processing system and Internet of Things,” Optik (Stuttg.), vol. 273, no. 170433, p. 170433, 2023, doi: 10.1016/j.ijleo.2022.170433.

[7] B. Sokolov, V. Ushakov, and V. Zakharov, “Optimal planning and scheduling of information processes during interaction among mobile objects,” Int. J. Prod. Res., pp. 1–20, 2024, doi: 10.1080/00207543.2024.2302388.

[8] Ушаков В.А. Модели и алгоритмы управления информационными процессами при взаимодействии подвижных объектов // Морские интеллектуальные технологии, 2022, № 3-1 (57), с. 235-247. DOI: 10.37220/MIT.2022.57.3.031.

[9] D.D. Savelyeva and T.M. Tatarnikova, “Internet of things traffic consumption control system,” // 2022 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), 2022, doi: 10.1109/WECONF55058.2022.9803422.

[10] Федеральный закон от 22.07.2008 N 123-ФЗ (ред. от 25.12.2023) “Технический регламент о требованиях пожарной безопасности” [Электронный ресурс]. URL:https://legalacts.ru/doc/FZ-Teh-reglament-o-trebovanijah-pozharnoj-bezopasnosti/ (Дата обращения: 04.04.2024)

[11] Society of Fire Protection Engineers, “Risk, fire risk, and fire risk assessment,” in SFPE Guide to Fire Risk Assessment, Cham: Springer International Publishing, 2023, pp. 5–9, doi: 10.1007/978-3-031-17700-2_2.

[12] Society of Fire Protection Engineers, “Overview of the fire risk assessment process,” in SFPE Guide to Fire Risk Assessment, Cham: Springer International Publishing, 2023, pp. 11–17, doi: 10.1007/978-3-031-17700-2_3.

[13] D. Vasilyev and I. Ozden, “Improvement of fire risk calculation method for linear part of main pipeline,” in VII International conference “Safety problems of civil engineering critical infrastructures” (SPCECI2021), 2023, doi: 10.1063/5.0125407.

[14] Пузач С.В. Математическое моделирование тепломассообмена при решении задач пожаровзрывобезопасности: монография. М.: Академия ГПС МЧС России, 2003. 150 с.

[15] Пузач С.В. Методы расчета тепломассообмена при пожаре в помещении и их применение при решении практических задач пожаровзрывобезопасности: монография. М.: Академия ГПС МЧС России, 2005. 336 с.

[16] S. I. Marakkaparambil, R. Rameshkumar, M. P. Dinesh, A. Aslam, and M. S. Ansari, “FireNet-micro: Compact fire detection model with high recall,” in Advances in Intelligent Systems and Computing, Cham: Springer Nature Switzerland, 2024, pp. 65–78, doi: 10.1007/978-3-031-47508-5_6.

[17] X. Zhou and C. Wang, “Research and implementation of forest fire detection algorithm improvement,” Int. J. Adv. Netw. Monit. Controls, vol. 8, no. 4, pp. 90–102, 2023, doi: 10.2478/ijanmc-2023-0080.

[18] ТСН 31-304-95 г. Москвы (МГСН 4.04-94) Многофункциональные здания и комплексы (С Изменением N 1) Официальное издание ГУП "НИАЦ", 1994 год [Электронный ресурс]. URL:https://docs.cntd.ru/document/1200000270?ysclid=luzmgqg4qo964593766 (Дата обращения: 04.04.2024)