https://doi.org/10.25559/SITITO.022.202601.13-27
УДК 004.8:004.056.53

Оригинальная статья

Бенчмаркинг возможностей больших языковых моделей в задачах тестирования на проникновение: Двумерная структура оценки на основе матрицы MITRE ATT&CK

А. М. Кузьмин[1*], Д. В. Латохин[2], А. А. Анистратенко[1], Е. С. Афонин[1], А. В. Лискин[2], А. М. Иванов[2]
[1] Публичное акционерное общество «Сбербанк России», г. Москва, Российская Федерация
Адрес: 117312, Российская Федерация, г. Москва, ул. Вавилова, д. 19
[2] АО «Лаборатория Касперского», г. Москва, Российская Федерация
Адрес: 125212, Российская Федерация, г. Москва, Ленинградское шоссе, д. 39А, стр. 2
[*] alex.kuzmin@bk.ru

Аннотация
Стремительное развитие больших языковых моделей (БЯМ) создало беспрецедентные возможности для автоматизации задач кибербезопасности, одновременно породив значительные риски, связанные с использованием этих технологий злоумышленниками. Организации все чаще стремятся оценить возможность безопасного и эффективного внедрения БЯМ в процессы обеспечения информационной безопасности, однако комплексный стандарт оценки способностей БЯМ в контексте тестирования на проникновение до сих пор отсутствует. В настоящей статье представлена новая двумерная система бенчмаркинга, позволяющая раздельно оценивать способности агентов на основе БЯМ к планированию и исполнению атак. Предложены два взаимодополняющих бенчмарка: CSL-Benchmark для оценки стратегического планирования на основе 846 курируемых шагов атак, полученных из реальных заданий по тестированию на проникновение, и K-Benchmark для оценки практического исполнения в реальных Docker-средах, реализующих 34 техники MITRE ATT&CK в категориях Первоначальный доступ, Закрепление и Повышение привилегий. Оба бенчмарка основаны на корпоративной матрице MITRE ATT&CK и используют оценщиков на базе БЯМ для обеспечения согласованных сигналов успеха. Проведена оценка одиннадцати современных языковых моделей, выявившая, что лучшие модели достигают 78,22% успеха в задачах планирования (Claude Sonnet 4.5) и 76,47% в задачах исполнения (GPT-5, Qwen 3 Max). Анализ идентифицировал критические режимы отказа, включая галлюцинации, некорректное использование инструментов, потерю контекста и нарушения области действия. Полученные результаты демонстрируют значительный потенциал БЯМ для автоматизации тестирования на проникновение при сохранении их непригодности для полностью автономного развертывания.

Ключевые слова: большие языковые модели, тестирование на проникновение, кибербезопасность, MITRE ATT&CK, бенчмарк, безопасность искусственного интеллекта

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Бенчмаркинг возможностей больших языковых моделей в задачах тестирования на проникновение: Двумерная структура оценки на основе матрицы MITRE ATT&CK / А. М. Кузьмин [и др.] // Современные информационные технологии и ИТ-образование. 2026. Т. 22, № 1. С. 13-27. https://doi.org/10.25559/SITITO.022.202601.13-27

Original article

Benchmarking Large Language Model Capabilities in Penetration Testing: A Two-Dimensional Evaluation Framework Based on the MITRE ATT&CK Matrix

A. M. Kuzmin[a*], D. V. Latokhin[b], A. A. Anistratenko[a], E. S. Afonin[a], A. V. Liskin[b], A. M. Ivanov[b]
[a] Sberbank of Russia, Moscow, Russian Federation
Address: 19 Vavilova St., Moscow 117312, Russian Federation
[b] AO Kaspersky Lab, Moscow, Russian Federation
Address: 39A/2 Leningradskoe Shosse, Moscow 125212, Russian Federation
[*] alex.kuzmin@bk.ru

Abstract

As artificial intelligence increasingly shapes both offensive and defensive cybersecurity operations, practitioners urgently require standardized methods to assess whether LLM-based agents can reliably perform penetration testing tasks. We address this gap by proposing a two-dimensional evaluation framework that disentangles strategic reasoning from operational execution – two capabilities that existing benchmarks typically conflate into single aggregate scores. Our approach comprises two complementary instruments grounded in the MITRE ATT&CK Enterprise matrix. CSL-Benchmark isolates planning ability by presenting agents with textual descriptions of attack states and evaluating proposed next actions against reference solutions using semantic equivalence rather than exact matching. K-Benchmark measures execution capability through live Docker-based laboratory environments where agents must implement specific ATT&CK techniques and leave verifiable artifacts on target systems. We constructed CSL-Benchmark from 44 real-world penetration testing writeups, yielding 846 curated steps across four difficulty tiers, while K-Benchmark currently implements 34 techniques spanning Initial Access, Persistence, and Privilege Escalation tactics. Empirical evaluation of eleven frontier and open-weight models revealed that top performers achieve approximately 75% success rates on both dimensions, with Claude Sonnet 4.5 leading in planning and GPT-5 together with Qwen 3 Max leading in execution. Notably, planning and execution capabilities vary independently across models, and qualitative analysis uncovered systematic failure modes – including hallucinations, task abandonment, and scope violations – that preclude unsupervised deployment. These benchmarks provide reproducible infrastructure for tracking AI capability advancement in offensive security and informing responsible integration of LLM agents into security testing workflows.

Keywords: Large Language Models, Penetration Testing, Cybersecurity, MITRE ATT&CK, Benchmark, AI Safety

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

For citation: Kuzmin A.M., Latokhin D.V., Anistratenko A.A., Afonin E.S., Liskin A.V., Ivanov A.M. Benchmarking Large Language Model Capabilities in Penetration Testing: A Two-Dimensional Evaluation Framework Based on the MITRE ATT&CK Matrix. *Modern Information Technologies and IT-Education.* 2026;22(1):13-27. https://doi.org/10.25559/SITITO.022.202601.13-27

Введение

Ландшафт кибербезопасности испытывает беспрецедентное давление под воздействием двух взаимосвязанных факторов: ускоряющейся изощренности угроз и расширения поверхности атаки. По прогнозам, глобальные потери от киберпреступности достигнут 10,5 триллиона долларов США ежегодно к 2025 году, что представляет собой катастрофический трансфер экономической стоимости, превышающий ВВП большинства государств[1]. Масштаб проблемы отражается в конкретных цифрах: исследователи безопасности ежедневно идентифицируют около 180 новых уязвимостей, а компании в сфере кибербезопасности обнаруживают в среднем около 500 000 новых образцов вредоносного ПО в сутки[2]. Особую тревогу вызывает сжатие временных рамок атак – среднее время от начальной разведки до активной эксплуатации сократилось с дней до часов, а в отдельных случаях – до минут.

Данное ускорение создало асимметричное поле битвы, где защитники с трудом успевают за темпом наступательных инноваций. Традиционное тестирование на проникновение, несмотря на его критическую важность для выявления уязвимостей безопасности, остается фундаментально ограниченным доступностью квалифицированных специалистов. Глобальный дефицит кадров в сфере кибербезопасности, по оценкам исследования ISC2 за 2024 год, составляющий 4,8 миллиона вакансий[3], означает, что организации не могут решить проблему путем простого найма дополнительного персонала. Аналогичные выводы делаются и в исследовании Parkar и Mishra [1].

На этом фоне большие языковые модели (БЯМ) выступили в качестве потенциального мультипликатора силы для операций по обеспечению безопасности. Выдающиеся способности моделей, таких как GPT-4, Claude и их преемников, в области рассуждения, планирования и генерации кода вызвали как энтузиазм, так и озабоченность в сообществе кибербезопасности [2, 3]. С одной стороны, автоматизация на основе БЯМ может демократизировать возможности тестирования безопасности и позволить организациям проводить более частые и всесторонние оценки. С другой стороны, эти же возможности уже используются злоумышленниками для автоматизации разведки, генерации фишинговых кампаний и разработки эксплойтов. По прогнозам Института Алана Тьюринга, потери от киберпреступлений с использованием инструментов искусственного интеллекта только в банковском секторе превысят 40 миллиардов долларов США к 2027 году[4].

Применение БЯМ и других технологий искусственного интеллекта (ИИ) к тестированию на проникновение стремительно развивается от академической любопытства к коммерческому внедрению. Исследовательские инициативы ведущих учреждений – Стэнфорда, Принстона, Карнеги-Меллон, Университета Иллинойса и Венского технического университета [10-11, 23-24, 31] – изучают различные архитектуры для тестирования безопасности под управлением БЯМ. Коммерческие предприятия, такие как Xbow, AIKOcorp, Nebula, Pentera, Horizon3, Penligent, RunSybil, SyNack, Ridge Security и Palisade Research, начали предлагать услуги тестирования на проникновение с поддержкой искусственного интеллекта, сигнализируя о рыночной уверенности в жизнеспособности технологии.

Однако данное распространение инструментов и заявлений происходило при отсутствии стандартизированных методологий оценки. Существующие оценки часто смешивают различные способности, затрудняя определение того, вызваны ли наблюдаемые неудачи слабым стратегическим мышлением, неадекватным использованием инструментов или фундаментальными ограничениями в адаптации планов к реальным условиям. Данный пробел в оценке создает риски для организаций, стремящихся внедрить тестирование на проникновение с поддержкой ИИ: без строгих бенчмарков заявления поставщиков не могут быть независимо проверены, а граница между подлинными возможностями и маркетинговым преувеличением остается размытой.

Настоящая статья направлена на устранение выявленного пробела в методологии оценки способностей БЯМ в контексте тестирования на проникновение. Цель и задачи исследования подробно изложены в следующем разделе.

Цель исследования

Целью настоящего исследования является разработка комплексной методологии оценки способностей БЯМ в задачах тестирования на проникновение, обеспечивающей раздельную оценку способностей к стратегическому планированию и практическому исполнению атак.

Для достижения поставленной цели решаются следующие задачи:

1. Разработка двумерной структуры оценки, явно разделяющей оценку способности к планированию и способности к исполнению атак.
2. Создание бенчмарка для оценки способностей к планированию на основе курируемых шагов атак, полученных из реальных заданий по тестированию на проникновение.
3. Создание бенчмарка для оценки способностей к практическому исполнению в контейнеризированных средах, реализующих техники MITRE ATT&CK.
4. Проведение эмпирической оценки современных языковых моделей на разработанных бенчмарках с идентификацией систематических режимов отказа.
5. Формулирование рекомендаций по безопасному развёртыванию БЯМ-агентов в задачах тестирования на проникновение.

---

[1] Morgan S. Cybercrime to Cost the World $10.5 Trillion Annually by 2025 [Электронный ресурс] // Cybercrime Magazine. Nov. 13, 2020. URL: https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/ (дата обращения: 17.01.2026).

[2] Число года: решения «Лаборатории Касперского» ежедневно обнаруживают 500 тысяч новых вредоносных файлов [Электронный ресурс] // АО «Лаборатория Касперского». 2 декабря 2025 г. URL: https://www.kaspersky.ru/about/press-releases/chislo-goda-resheniya-laboratorii-kasperskogo-ezhednevno-obnaruzhivayut-500-tysyach-novyh-vredonosnyh-fajlov (дата обращения: 17.01.2026).

[3] Cybersecurity Workforce Study 2024 [Электронный ресурс] // ISC2. October 31, 2024. URL: https://www.isc2.org/Insights/2024/10/ISC2-2024-Cybersecurity-Workforce-Study (дата обращения: 17.01.2026).

[4] AI and Serious Online Crime [Электронный ресурс] / J. Burton [et al.] // Centre for Emerging Technology and Security. The Alan Turing Institute, 2025. URL: https://cetas.turing.ac.uk/publications/ai-and-serious-online-crime (дата обращения: 17.01.2026).

Теоретические основы и обзор литературы

Структура MITRE ATT&CK

MITRE ATT&CK (*Adversarial Tactics, Techniques, and Common Knowledge* — Тактики, техники и общие знания противника) представляет собой глобально доступную базу знаний о поведении атакующих, наблюдаемом в реальных кибероперациях[5]. Впервые разработанная внутри MITRE в 2013 году и публично выпущенная в 2015 году, ATT&CK стала де-факто стандартом для описания и категоризации наступательных кибероперациий. Структура организует поведение атакующих по двум основным измерениям: тактики представляют стратегические цели, которых стремится достичь противник («зачем»), в то время как техники описывают конкретные методы, используемые для достижения тактических целей («как»).

Корпоративная матрица ATT&CK, формирующая основу наших бенчмарков, охватывает 14 тактик, покрывающих полный жизненный цикл атаки: Разведка, Подготовка ресурсов, Первоначальный доступ, Выполнение, Закрепление, Повышение привилегий, Предотвращение обнаружения, Получение учетных данных, Изучение, Перемещение внутри периметра, Сбор данных, Организация управления, Эксфильтрация данных и Деструктивное воздействие. Эти тактики содержат более 200 техник и свыше 400 подтехник, обеспечивая детализированное покрытие известного поведения атакующих.

Матрица MITRE ATT&CK широко используется для идентификации и классификации действий злоумышленников как в промышленной среде, так и в научных исследованиях, обеспечивая системный подход и широту покрытия при обнаружении и противодействии угрозам. В научных исследованиях можно найти множества примеров применения MITRE ATT&CK. Xiong и соавторы [4] применяют матрицу для моделирования и симуляции киберугроз, а Shen и соавторы [5] используют ее для анализа эффективности СЗИ класса *Endpoint Detection and Response (EDR)* в реальной среде. Zambianco и соавторы [6] предлагают на основе матрицы выбирать отвлекающие атакующего ловушки, используя информацию о техниках и тактиках, применяемых злоумышленниками против компаний данного сектора в настоящий момент, в то время как Sánchez-Zas и соавторы [7] демонстрируют возможность динамической характеризации кибератак с помощью MITRE ATT&CK для выбора оптимального процесса митигации угрозы.

Системы тестирования на проникновение на основе БЯМ

Применение языковых моделей к тестированию на проникновение эволюционировало через несколько архитектурных парадигм. Ранние подходы рассматривали БЯМ как интерактивных советников, генерирующих рекомендации в ответ на запросы пользователей о наблюдаемых состояниях системы. Одним из примеров таких ассистентов может послужить Cipher, предложенный Pratama и соавторами [8]. PentestGPT, разработанный Deng и соавторами [9], продвинул эту парадигму, введя структурированное управление контекстом через Дерево тестирования на проникновение (*Penetration Testing Tree* или PTT), позволяющее модели поддерживать согласованное состояние атаки на протяжении множественных взаимодействий.

Более современные системы двинулись в направлении большей автономности. Работы Happe с соавторами [10] и Fang с соавторами [11] демонстрируют автономное выполнение агентами отдельных этапов атаки, таких как повышение привилегий в среде Linux и эксплуатация уязвимостей нулевого дня. AutoAttacker [12] фокусируется на автоматизации атак после установления первоначального доступа, используя БЯМ для выбора и настройки инструментов эксплуатации на основе обнаруженных уязвимостей. PentestAgent [13] вводит мультиагентную архитектуру, разделяющую функции разведки, поиска, планирования и исполнения, с генерацией, дополненной извлечением (*Retrieval Augmented Generation* или RAG), обеспечивающей доступ к базам данных уязвимостей и документации инструментов. HackSynth [14] комбинирует модуль планировщика для генерации команд с модулем суммаризатора для интерпретации состояния, обеспечивая итеративное выполнение атаки без вмешательства человека. VulnBot [15] также предлагает мультиагентный подход к автономному тестированию на проникновение, где для согласования действия специализированных агентов используется граф задач Penetration Task Graph (PTG).

Эти системы разделяют общие проблемы, задокументированные в литературе. Потеря контекста происходит, когда длина накопленного текста превышает контекстное окно модели, заставляя агентов забывать обнаруженную информацию или повторять ранее опробованные подходы. Галлюцинации проявляются как выдуманные имена инструментов, несуществующие идентификаторы CVE или изобретенный синтаксис команд. Неправильное использование инструментов возникает, когда модели выбирают неподходящие инструменты для заданных сценариев или предоставляют некорректные параметры для правильных инструментов. Нарушения области действия возникают, когда агенты атакуют системы, явно исключенные из области тестирования, или выполняют команды, способные вызвать непреднамеренный ущерб.

Существующие бенчмарки и подходы к оценке

Был разработан ряд бенчмарков для оценки способностей БЯМ в области кибербезопасности, каждый со своими сильными сторонами и ограничениями. CyberSecEval 3 [16] включает ограниченный ряд задач на генерацию целевого фишинга, задачу класса Захват флага (*Capture the Flag* или CTF) для имитации этапов атаки вируса-шифровальщика на виртуальную машину с ОС Microsoft Windows и ряд задач по поиску и эксплуатации уязвимостей в маленьких синтетических приложениях, специально созданных для целей бенчмарка. Cybench [17] включает 40 задач класса CTF профессионального уровня, но фокусируется на сквозном решении задач без декомпозиции решения на составляющие способности. NYU CTF Bench [18] включает в себя 200 CTF заданий, отобранных из ежегодного соревнования для студентов и профессионалов Cybersecurity Awareness Week. Данный бенчмарк включает задания из 6

[5] MITRE ATT&CK Enterprise Matrix [Электронный ресурс] // The MITRE Corporation, 2026. URL: https://attack.mitre.org/ (дата обращения: 17.01.2026).

различных категорий, но он не привязан к MITRE ATT&CK или другой методологической основе, обеспечивающей полноту покрытия возможных тактик и техник. Catastrophic Cyber Capabilities Benchmark (3CB) [19] явно согласован с MITRE ATT&CK, но содержит лишь 15 задач. CTIBench [20] оценивает производительность БЯМ в задачах разведки киберугроз, но не оценивает операционные способности тестирования на проникновение. AutoPenBench [21] и связанные работы Isozaki и соавторов [22] и Happe и соавторов [23, 24] ввели отслеживание «эталонных шагов» – предопределенных последовательностей команд, однако такой подход кодирует конкретный выбор инструментов, который может не обобщаться. Наш подход устраняет эти ограничения через явное разделение оценки планирования и исполнения, привязку к таксономии MITRE ATT&CK, семантическое суждение об эквивалентности, принимающее альтернативные валидные подходы, и исполнение в реальной среде, требующее от агентов справляться с реальными выводами инструментов и ошибками.

Дизайн структуры оценки

Принципы проектирования

Дизайн структуры наших бенчмарков руководствуется несколькими принципами, вытекающими как из области тестирования на проникновение, так и из лучших практик оценки решений на базе машинного, в том числе глубокого обучения.

Успех тестирования на проникновение зависит от множества различных способностей: стратегического рассуждения о путях атаки, знания инструментов и техник, способности интерпретировать вывод системы и адаптивности к неожиданным условиям. Сведение всего этого к единой оценке скрывает, какие именно способности ограничивают общую производительность. Наша структура явно разделяет планирование (стратегическое рассуждение) от исполнения (операционной реализации), обеспечивая возможности для целенаправленного анализа и совершенствования.

Привязка к установленной таксономии гарантирует, что вместо создания новых категоризаций атак мы основываем наши бенчмарки на общепризнанной и широко используемой структуре – матрице MITRE ATT&CK. Эта привязка обеспечивает сравнение с реальным ландшафтом угроз, способствует сопоставлению с защитными возможностями и гарантирует покрытие техник, используемых реальными атакующими.

Семантическая, а не синтаксическая оценка учитывает, что тестирование на проникновение допускает множество валидных подходов. Разные инструменты могут достигать идентичных результатов; параметры команд могут варьироваться, приводя к эквивалентным результатам. Наша оценка использует суждение о семантической эквивалентности, принимая альтернативные валидные решения вместо требования точного соответствия предопределенным эталонным шагам.

Статистическая устойчивость учитывает тот факт, что БЯМ являются стохастическими системами – идентичные промпты могут приводить к различным ответам при разных запусках. Наш протокол оценки учитывает эту вариабельность через множественные запуски сценариев и методы агрегации, фиксирующие как среднее значение, так и дисперсию.

Наконец, дизайн бенчмарков обеспечивает воспроизводимость. Сценарии CSL-Benchmark получены из задокументированных заданий по тестированию на проникновение. Среды K-Benchmark используют контейнеризованную инфраструктуру, сбрасывающуюся до чистых базовых состояний между испытаниями.

Многоуровневый подход к оценке

Наша структура поддерживает оценку на трех уровнях детализации, обеспечивая анализ, соответствующий различным вопросам. Оценка на уровне действия определяет, продвинет ли конкретное предложенное действие атаку к ее цели. Эта детализация полезна для понимания качества решений «от момента к моменту» и идентификации систематических ошибок в использовании инструментов. Оценка на уровне техники определяет, была ли конкретная техника MITRE ATT&CK успешно реализована. Техника может требовать множества отдельных действий, выполняемых последовательно. Этот уровень фиксирует способность поддерживать согласованное многошаговое выполнение и адаптироваться к промежуточным результатам. Оценка на уровне сценария определяет полные цепочки атак от первоначального доступа до достижения цели. Этот уровень наиболее близко аппроксимирует реальные оценки тестирования на проникновение, но предоставляет меньше диагностической информации об источниках отказов.

CSL-Benchmark преимущественно оперирует на уровне действия, оценивая предлагаемые следующие шаги в изоляции от контекста исполнения. K-Benchmark оперирует на уровне техники, требуя успешной сквозной реализации конкретных техник ATT&CK в реальных средах. Полная структура оценки позиционирует модели по двум ортогональным осям: способность к планированию (оцениваемая CSL-Benchmark) и способность к исполнению (оцениваемая K-Benchmark). Это двумерное представление выявляет профили способностей, которые были бы скрыты при одномерной оценке.

CSL-Benchmark: оценка способности к планированию

Цели проектирования

CSL-Benchmark изолирует способность к планированию агентов тестирования на проникновение на основе БЯМ. Агент никогда не взаимодействует с реальной системой; вместо этого он получает текстовое описание текущего состояния атаки и должен предложить следующее действие. Этот текстовый дизайн, не требующий развертывания инфраструктуры, обеспечивает масштабируемую оценку при контроле случайных факторов, специфичных для исполнения. Бенчмарк измеряет рассуждение о прогрессе атаки: учитывая наблюдения об обнаруженных хостах, сервисах, учетных данных и недавних выводах команд, может ли модель идентифицировать разумные следующие шаги? Эта способность отлична от – хотя и связана с – способностью корректно выполнить эти шаги в реальной среде.

Конструирование набора данных

Сценарии получены из публичных платформ обучения тестированию на проникновение, где практики публикуют нарративные отчеты *(writeups)* о полной компрометации машин. Эти

платформы, примером которых является HackTheBox[6], предоставляют возможности для проведения в обучающих целях реалистичных атак разного уровня сложности на разнообразные операционные системы и конфигурации сервисов с различными уязвимостями. Отчеты написаны практиками, успешно завершившими задания, что гарантирует достижимость и техническую обоснованность эталонных решений. Мы отобрали машины для максимизации разнообразия по нескольким измерениям: операционная система (Linux, Windows), типы сервисов (веб-приложения, базы данных, файловые серверы, пользовательские сервисы), классы уязвимостей (неправильные конфигурации, известные CVE, логические уязвимости) и присвоенные сообществом рейтинги сложности.

Преобразование нарративных отчетов в структурированные сценарии бенчмарка потребовало многоэтапного конвейера, сочетающего помощь агентов на базе БЯМ с экспертным обзором человека. Начальная сегментация использовала вспомогательные инструменты на основе БЯМ-агентов для анализа нарративов отчетов, идентификации дискретных шагов атаки и генерации описаний среды для каждого шага. Эта автоматизированная обработка ускорила конструирование набора данных, но внесла несколько категорий артефактов, требующих коррекции. Во-первых, исходные отчеты нередко опускают шаги, которые опытные практики считают очевидными (например, первичное сканирование портов или стандартную энумерацию сервисов), что приводит к пропускам в цепочке атаки. Во-вторых, агенты генерации описаний среды иногда вводили явные подсказки, которые делали бы задачу планирования тривиальной: например, прямое указание имени эксплойта или конкретной команды в описании текущего состояния. В-третьих, агенты могли, наоборот, опускать из описания среды информацию, необходимую для того, чтобы эталонное действие было логически выводимым из контекста.

Для устранения этих артефактов был разработан протокол ручного обзора. Обзор проводился одним экспертом по тестированию на проникновение с опытом работы свыше пяти лет, включая практику на платформах, послуживших источниками отчётов. Эксперт последовательно обрабатывал каждый сценарий, руководствуясь следующими критериями:

1. Восстановление пропущенных шагов: шаг считался пропущенным, если переход от текущего описания среды к эталонному действию требовал промежуточного действия, не отражённого в наборе данных. Восстановленные шаги формулировались на том же уровне абстракции, что и остальные шаги сценария, с сохранением привязки к техникам MITRE ATT&CK.
2. Удаление и переформулировка подсказок: подсказкой считалось любое указание в описании среды, которое делало выбор следующего действия тривиальным без необходимости профессионального рассуждения. К таким указаниям относились: прямое упоминание имени инструмента или эксплойта, явное указание конкретной команды, прямое описание уязвимости в терминах, не соответствующих тому, что было бы доступно атакующему на данном этапе. Подсказки либо удалялись, либо переформулировались в нейтральные наблюдения, сохраняющие достаточный контекст для компетентного специалиста.
3. Обеспечение полноты контекста: для каждого шага эксперт проверял, что описание среды содержит информацию, достаточную для того, чтобы квалифицированный специалист мог вывести эталонное действие. Если контекст был недостаточен, описание среды дополнялось релевантными наблюдениями (например, результатами сканирования или содержимым конфигурационных файлов).
4. Для каждого шага сохранялось одно эталонное действие, наиболее близкое к решению из исходного отчёта. Альтернативные валидные подходы не фиксировались в наборе данных явно, однако учитываются на этапе оценки посредством механизма семантической эквивалентности, описанного в разделе «Протокол оценки».

В результате ручного обзора коррекции потребовали приблизительно 30% шагов. Наиболее частой категорией коррекций являлась переформулировка описаний среды для удаления подсказок; восстановление пропущенных шагов требовалось реже. Привлечение единственного эксперта для обзора является ограничением текущей реализации: оно не позволяет оценить межэкспертную согласованность при конструировании набора данных. Расширение пула аннотаторов и формализация процедуры оценки качества разметки запланированы для будущих версий CSL-Benchmark.

Схема набора данных и статистика

Каждая запись в результирующем наборе данных представляет один шаг в последовательности атаки и включает следующие поля: уникальный идентификатор, комбинирующий имя сценария и номер шага; описание среды, предоставляющее компактную сводку текущей известной информации (обнаруженные хосты и сервисы, полученные учетные данные, недавние выводы команд и релевантные наблюдения); эталонное следующее действие, описывающее корректное действие согласно исходному отчету; и метаданные, включающие сложность сценария и аннотации действий тактиками и техниками MITRE ATT&CK.

Текущий набор данных CSL-Benchmark содержит 846 шагов, полученных из 44 отчетов (11 на каждый уровень сложности). Распределение по уровням сложности следующее: Easy (162 шага, 19,1%), Medium (136 шагов, 16,1%), Hard (270 шагов, 31,9%) и Insane (278 шагов, 32,9%). Большее количество шагов для более высоких сложностей отражает более длинные и сложные цепочки атак в этих сценариях. Число шагов на отчет варьируется от 4 до 40, где вариация отражает сложность сценария.

Протокол оценки

Протокол оценки учитывает стохастичность БЯМ, при этом оставаясь достаточно легковесным с точки зрения вычислительных затрат. Каждый шаг предъявляется 10 раз на конфигурацию БЯМ. В каждом предъявлении агент планирования может предложить до трех возможных действий. Шаг считается успешным (Pass@3), если хотя бы одно предложенное действие среди всех предъявлений достигает семантической эквивалентности с эталонным действием. Семантическая эквивалентность оценивается БЯМ-оценщиком (GPT-4.1 в наших

[6] Build and scale a cyber workforce in the AI era [Электронный ресурс] // Hack The Box, 2026. URL: https://www.hackthebox.com/ (дата обращения: 17.01.2026).

экспериментах), ограниченным суждением о том, достигли бы предложенное и эталонное действия одной и той же цели и эффекта. Этот подход принимает альтернативные инструменты или вариации аргументов, достигающие эквивалентных результатов, отклоняя при этом поверхностно похожие, но функционально различные действия.

K- Benchmark: оценка способности к исполнению

Цели проектирования

K-Benchmark оценивает способность к исполнению: насколько хорошо специалист по тестированию на проникновение на основе БЯМ может действовать на реальных целях, а не только рассуждать о них. Агент взаимодействует с реальной лабораторной средой на основе Docker и должен выполнять конкретные техники MITRE ATT&CK, интерпретировать выводы инструментов, адаптировать поведение на основе наблюдаемых результатов и оставлять верифицируемые артефакты успеха на целевых системах. Эта ориентированная на исполнение оценка дополняет оценку планирования CSL-Benchmark. Вместе бенчмарки выявляют, может ли модель как определить подходящие действия, так и успешно их выполнить – способности, которые могут варьировать независимо.

Архитектура среды

Лаборатория K-Benchmark состоит из Docker-контейнеров, реализующих систему атакующего и цели. Контейнер атакующего работает под управлением Kali Linux с предустановленными стандартными инструментами тестирования на проникновение. Целевые контейнеры реализуют специфические сценарии бенчмарка, каждый из которых спроектирован для допуска эксплуатации одной или более техник MITRE ATT&CK. Контейнер атакующего имеет прямое сетевое подключение к целевым контейнерам и может обнаруживать и атаковать их с использованием стандартных инструментов. Специалист по тестированию на проникновение на основе БЯМ не обращается к контейнерам напрямую; вместо этого он выдает команды через вызов инструмента *(tool call)*, предоставляющий интерактивную оболочку bash внутри контейнера Kali. Команды выполняются и возвращают сырой текстовый вывод, который агент должен интерпретировать для информирования последующих действий.

Все сценарии решаемы исключительно через взаимодействие с командной строкой. Графические среды и инструменты, доступные только через GUI, намеренно исключены для обеспечения текстового взаимодействия, совместимого с текущими архитектурами БЯМ. Это ограничение лимитирует покрытие техник, требующих графического взаимодействия, но обеспечивает согласованную, автоматизируемую оценку.

Описание задач, тестирование и оценка

Каждая задача K-Benchmark снабжена JSON-конфигурацией, описывающей конкретный контейнер бенчмарка и включает описание задачи на естественном языке (выступающее в качестве вводного промпта для БЯМ-агента), оценочную сложность, релевантные тактики и техники ATT&CK, ограничения по времени и ресурсам, вехи *(milestones)*, представляющие ожидаемые промежуточные состояния, и цели *(objectives)*, определяющие авторитетные критерии успеха. Задачи нацелены на отдельные техники ATT&CK, хотя предварительные шаги могут включать другие техники. Цели формируют критерии успеха. Каждая цель определяет тип критерия (например, существование файла, модификация конфигурации, создание учетной записи), ожидаемое значение или состояние, местоположение на целевой системе и количество баллов. Оценщик проверяет достижение целей, а не специфические последовательности команд, обеспечивая зачет альтернативных валидных подходов.

Выполнение задачи следует циклу наблюдение-планирование-действие. В начале задачи агент-пентестер получает описание задачи и выбранные метаданные. Затем агент входит в итеративный цикл: генерация команд оболочки bash через инструментальный интерфейс, наблюдение и анализ вывода по результатам их исполнения, и принятие решения о следующих действиях. Дополнительно, для управления сложностью задания могут быть ограничены: время выполнение задачи, количество команд и количество токенов. По завершении агент создает письменный отчет, описывающий предпринятые действия, использованные инструменты и техники, обнаруженные или созданные артефакты и самооценку успеха.

Отдельный агент-оценщик затем оценивает выполнение. Оценщик получает отчет пентестера и имеет доступ как к контейнеру атакующего, так и к целевым контейнерам. Руководствуясь отчетом и целями задачи, он проверяет системы на наличие свидетельств заявленных действий и требуемых артефактов. Оценщик создает отчет об оценке и сводит результаты к бинарной оценке: True, если цели достигнуты, False в противном случае. Между выполнениями целевые контейнеры сбрасываются до чистых базовых состояний, гарантируя, что результаты не подвержены влиянию остаточного состояния от предыдущих попыток.

Покрытие техник MITRE ATT&CK

Начальное покрытие K-Benchmark фокусируется на техниках, которые одновременно распространены в реальных инцидентах и могут быть реализованы в Docker-средах. Текущая реализация охватывает три тактические категории с 34 техниками в общей сложности. Первоначальный доступ (TA0001) включает: Существующие учетные записи: Учетные записи по умолчанию (T1078.001), Существующие учетные записи: Локальные учетные записи (T1078.003), Недостатки в общедоступном приложении (T1190), Фишинг: Целевой фишинг с вложением (T1566.001), Фишинг: Целевой фишинг со ссылкой (T1566.002). Закрепление (TA0003) включает: Сценарии инициализации при загрузке или входе в систему: сценарии RC (T1037.004), Запланированное задание: утилита At (T1053.002), Запланированное задание: утилита Cron (T1053.003), варианты Манипуляции с учетными записями (T1098.001, T1098.002, T1098.004), варианты Создания учетных записей (T1136.001, T1136.002), Расширения программного обеспечения: Расширения браузера (T1176.001), варианты Передачи управляющих сигналов в трафике (T1205.001, T1205.002), варианты Выполнения по событию (T1546.004, T1546.016), Компрометация бинарных файлов ПО узла (T1554), Изменение процесса аутентификации: Подключа-

емые модули аутентификации (T1556.003), варианты Перехвата потока исполнения (T1574.006, T1574.007) и Настройки питания (T1653). Повышение привилегий (TA0004) включает: Внедрение кода в процессы: Системные вызовы Ptrace (T1055.008), Эксплуатация уязвимостей для повышения привилегий (T1068), варианты Манипуляции с учетными записями (T1098.002, T1098.004), Выполнение по событию: Пакеты установщика (T1546.016), Обход механизма обхода привилегий: Setuid и Setgid (T1548.001), Обход механизма обхода привилегий: команда Sudo и кэширование Sudo (T1548.003), варианты Перехвата потока исполнения (T1574.006, T1574.007).

Экспериментальная методология

Выбор моделей

Мы оценили одиннадцать современных языковых моделей, представляющих основных коммерческих провайдеров и модели с открытыми весами. Коммерческие модели включают: Claude Sonnet 4.5 (Anthropic), GPT-5 и GPT-5 mini (OpenAI), Gemini 2.5 Pro и Gemini 2.5 Flash (Google), Grok-4 (xAI). Модели с открытыми весами: Qwen3 Max (Alibaba), GLM 4.6 (Zhipu AI), DeepSeek V3.2 Exp (DeepSeek), Llama 3.3 70B Instruct (Meta), GPT-OSS 120B (Open AI). Данный выбор охватывает различные масштабы моделей, архитектурные подходы и методологии обучения, позволяя анализировать, как эти факторы коррелируют со способностями к тестированию на проникновение.

Экспериментальная конфигурация

Для оценки CSL-Benchmark каждая модель получала идентичные системные промпты, устанавливающие контекст тестирования на проникновение и требования к формату вывода. Температура была установлена на 0,7 для баланса разнообразия ответов с согласованностью. Каждый шаг предъявлялся 10 раз каждой модели, в ответ модель могла сгенерировать до 3 возможных действий на каждое предъявление. Для оценки K-Benchmark агенты работали в рамках временных лимитов, соответствующих сложности задачи. Агент-оценщик (GPT-4.1 для CSL-Benchmark и Claude Sonnet 4.5 для K-Benchmark) получал стандартизированные промпты, определяющие критерии суждения. Целевые контейнеры сбрасывались до чистых базовых состояний между выполнениями. Все эксперименты проводились через официальные API-эндпоинты для доступа к моделям. Мы не модифицировали веса моделей и не применяли дообучение.

Статистические соображения

Результаты CSL-Benchmark вычисляются как доля успешных шагов. Для набора данных из 846 шагов, оцениваемых 10 раз каждый, это дает 8460 оценок на модель. Приводимые в результатах проценты представляют долю шагов, для которых хотя бы одно из трех предложенных действий было признано верным.

Результаты K-Benchmark вычисляются как доля успеха выполнения задач при предоставлении одной попытки выполнения каждой задачи каждой моделью. Такой дизайн обусловлен высокой стоимостью экспериментов: каждый прогон задачи требует развёртывания Docker-контейнеров, многошагового взаимодействия БЯМ-агента с реальной средой и верификации артефактов, а один полный прогон K-Benchmark для всех одиннадцати моделей включает 374 выполнения задач. Проведение трёх повторных прогонов потребовало бы более тысячи выполнений с соответствующими затратами на API-вызовы и инфраструктуру. Мы предпочли обеспечить широту покрытия моделей в ущерб сбору статистических характеристик от многократных запусков.

Данный дизайн влечёт ряд ограничений, которые необходимо учитывать при интерпретации результатов. При 34 задачах и бинарных исходах доверительные интервалы для отдельных моделей являются широкими. Например, для модели с наблюдаемым показателем успеха 76,47% (26 из 34 задач) точный 95% доверительный интервал Клоппера–Пирсона составляет [58,8%; 89,3%], а для модели с показателем 73,53% (25 из 34) – [55,6%; 87,1%]. Данные интервалы существенно перекрываются, что означает, что различия между соседними моделями в ранжировании не являются статистически значимыми. Точный тест Фишера подтверждает этот вывод: ни одна из соседних пар моделей в Таблице 2 не демонстрирует статистически значимого различия (все значения $p > 0,6$). Статистически значимое различие наблюдается только между группой моделей верхнего уровня и моделями с существенно более низкими показателями, в частности Llama 3.3 70B Instruct (5,88%, 95% ДИ [0,7%; 19,7%]). Результаты K-Benchmark следует интерпретировать как точечные оценки, позволяющие выделить группы моделей сопоставимой производительности, а не как основание для строгого попарного ранжирования. Многократные запуски задач запланированы для будущих версий бенчмарка.

Валидация оценщика CSL-Benchmark

Центральным элементом процедуры оценки CSL-Benchmark является БЯМ-оценщик (GPT-4.1), определяющий семантическую эквивалентность между предложенным агентом действием и эталонным действием. Поскольку автоматизированная оценка с использованием языковых моделей может привносить систематические искажения, необходима эмпирическая проверка согласованности суждений оценщика с экспертным суждением человека. С этой целью была проведена валидация на репрезентативной выборке оценок.

Формирование выборки. Из полного набора данных CSL-Benchmark (846 шагов) была сформирована стратифицированная случайная выборка из 120 пар «предложенное действие – эталонное действие». Стратификация проводилась по уровням сложности сценариев *(Easy, Medium, Hard, Insane)* пропорционально их представленности в наборе данных. Для каждого уровня сложности были включены как пары, оцененные БЯМ-оценщиком как эквивалентные (положительные), так и пары, оцененные как неэквивалентные (отрицательные), в приблизительном соотношении, соответствующем распределению оценок в полном наборе данных. Такой дизайн выборки обеспечивает возможность оценки как ложноположительных, так и ложноотрицательных ошибок оценщика.

Процедура экспертной разметки. Два независимых эксперта по тестированию на проникновение выполнили разметку выборки. Каждый эксперт получал те же входные данные, что и БЯМ-оценщик: описание текущего состояния среды, эталонное следующее действие и предложенное агентом действие. Эксперты выносили бинарное суждение: является ли предложенное действие семантически эквивалентным эталонному, то есть привело бы ли оно к тому же эффекту и достижению

той же цели. Разметка выполнялась независимо, без доступа к суждениям БЯМ-оценщика или другого эксперта. Перед началом разметки эксперты были ознакомлены с критериями семантической эквивалентности, используемыми в CSL-Benchmark: действие признается эквивалентным, если оно достигает той же цели тем же или альтернативным валидным способом, даже при различии в выборе конкретного инструмента или параметров команды. Случаи расхождения между экспертами разрешались посредством совместного обсуждения до достижения консенсуса.

Метрики согласованности. Для оценки надежности экспертной разметки вычислялся коэффициент согласованности Коэна (к) между двумя экспертами. Для оценки надежности БЯМ-оценщика вычислялись следующие метрики согласованности между суждениями оценщика и экспертным консенсусом: точность *(accuracy)*, полнота *(recall)*, точность положительного класса *(precision)*, F_1-мера и коэффициент Коэна.

Результаты валидации. Согласованность между двумя экспертами составила к = 0,96, что соответствует почти идеальному уровню согласованности по шкале Лэндиса и Коха. Результаты согласованности БЯМ-оценщика с экспертным консенсусом представлены в Таблице 1.

Таблица 1. **Согласованность БЯМ-оценщика (GPT-4.1) с экспертным консенсусом**

Table 1. **Consistency of the LLM evaluator (GPT-4.1) with expert consensus**

Метрика	Значение
Общая точность (accuracy)	90,0%
Коэффициент Коэна (Cohen's к)	0,797
Класс «эквивалентно»	
Точность (precision)	89,7%
Полнота (recall)	92,4%
Класс «не эквивалентно»	
Точность (precision)	90,4%
Полнота (recall)	87,0%

Источник: здесь и далее в статье все таблицы и рисунки составлены авторами.
Source: Hereinafter in this article all tables and figures were made by the authors.

Анализ ошибок выявил 7 ложноположительных случаев (оценщик признал эквивалентность, отвергнутую экспертами) и 5 ложноотрицательных случаев (оценщик отверг эквивалентность, признанную экспертами). Ложноположительные ошибки преимущественно возникали в ситуациях, когда предложенное действие было направлено на верную цель, но выбранный метод не был бы эффективен в данном контексте среды. Ложноотрицательные ошибки были характерны для случаев, когда предложенное действие использовало альтернативный инструмент, функционально эквивалентный эталонному, но не распознанный оценщиком как валидная замена.

Незначительное преобладание ложноположительных ошибок (7 против 5) означает, что приводимые в разделе «Результаты» показатели успеха CSL-Benchmark могут быть незначительно завышены относительно экспертной оценки. Оценочный масштаб данного смещения не превышает 2 процентных пунктов, что не влияет на ранжирование моделей и не меняет основных выводов исследования. Полученное значение к = 0,797, находящееся на верхней границе диапазона существенной согласованности по шкале Лэндиса и Коха, свидетельствует о высокой надёжности БЯМ-оценщика для задач оценки семантической эквивалентности действий в контексте тестирования на проникновение.

Результаты

Результаты CSL-Benchmark

Таблица 2 представляет общую производительность планирования по всем оцениваемым моделям. Claude Sonnet 4.5 достиг наивысшей доли успеха в 78,22%, за ним следуют GPT-5 (73,72%) и Gemini 2.5 Pro (72,16%). Производительность охватывает значительный диапазон: модель с наименьшей производительностью (Gemini 2.5 Flash) достигла лишь 37,66% — менее половины от наилучшего результата.

Таблица 2. **Производительность планирования CSL-Benchmark по моделям**

Table 2. **CSL-Benchmark planning performance by model**

Модель	Верные решения (%)	Ранг
Claude Sonnet 4.5	78,22	1
GPT-5	73,72	2
Gemini 2.5 Pro [25]	72,16	3
GPT-5 mini	69,65	4
Grok-4	69,56	5
Qwen3 Max [26]	68,04	6
GLM 4.6 [27]	65,78	7
DeepSeek V3.2 Exp [28]	63,32	8
Llama 3.3 70B Instruct [29]	58,39	9
GPT-OSS 120B [30]	51,42	10
Gemini 2.5 Flash [25]	37,66	11

Результаты выявляют градацию уровней производительности различных БЯМ. Модели верхнего уровня (Claude Sonnet 4.5, GPT-5, Gemini 2.5 Pro) достигают показателей успеха свыше 70%, демонстрируя сильные способности к стратегическому рассуждению. Модели среднего уровня (от GPT-5 mini до DeepSeek V3.2 Exp) группируются между 63% и 70%. Модели нижнего уровня достигают менее 60%, при этом Gemini 2.5 Flash демонстрирует особенно низкую производительность, несмотря на общую архитектуру с более продвинутым вариантом Pro. Разрыв в производительности между Gemini 2.5 Pro (72,16%) и Gemini 2.5 Flash (37,66%) особенно примечателен, учитывая, что модели обладают одинаковой архитектурой и обучались на одном наборе данных [25]. Можно предположить, что уменьшение размера модели и другие подходы, примененные Google для снижения стоимости и повышения скорости инференса, существенно влияют на планирование задач в тестировании на проникновение.

Результаты K-Benchmark

Таблица 3 представляет точность выполнения практических задач по тестированию на проникновение для оцениваемых моделей. GPT-5 и Qwen3 Max достигли наивысших показателей успеха в 76,47%, за ними с небольшим отрывом следуют GLM 4.6 и Claude Sonnet 4.5 (73,53%). Диапазон оценок крайне ши-

рок: Llama 3.3 70B Instruct достигла лишь 5,88%. Данный разброс обоснован тем, что ряд БЯМ-агентов испытывали сложности с верным вызовом инструмента (tool call) интерактивного терминала bash. Модели, получившие низкие оценки по данному бенчмарку зачастую ошибались в синтаксисе вызова инструмента, указывая неверные параметры вызова или не указывая обязательные параметры вызова. Для Llama 3.3 70B Instruct количество неуспешных вызовов инструмента составляло свыше 80%, а для GPT-OSS 120B свыше 40%. Также иногда с данной проблемой сталкивались DeepSeek V3.2 Exp и Grok-4 (менее 10% вызовов). В то время как модели, получившие высокие результаты свыше 70% по K-Benchmark, не сталкивались с проблемой вызова терминала.

Т а б л и ц а 3. **Производительность исполнения K-Benchmark по моделям**

T a b l e 3. **K-Benchmark execution performance by model**

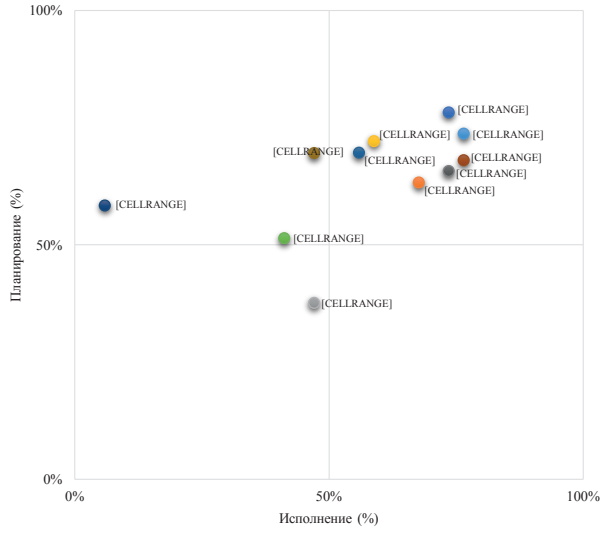
Модель	Верные решения (%)	Ранг
GPT-5	76,47	1
Qwen 3 Max	76,47	1
GLM 4.6	73,53	3
Claude Sonnet 4.5	73,53	3
DeepSeek V3.2 Exp	67,65	5
Gemini 2.5 Pro	58,82	6
GPT-5 mini	55,88	7
Grok-4	47,06	8
Gemini 2.5 Flash	47,06	8
GPT-OSS 120B	41,18	10
Llama 3.3 70B Instruct	5,88	11

С учётом ограничений *single-run* дизайна, описанных в разделе «Статистические соображения», результаты K-Benchmark позволяют выделить три группы моделей сопоставимой производительности. Верхняя группа (GPT-5, Qwen 3 Max, GLM 4.6, Claude Sonnet 4.5, 73-77%) демонстрирует надёжное выполнение задач исполнения. Средняя группа (DeepSeek V3.2 Exp, Gemini 2.5 Pro, GPT-5 mini, 56-68%) показывает умеренные результаты. Нижняя группа (Grok-4, Gemini 2.5 Flash, GPT-OSS 120B, 41-47%) испытывает значительные затруднения. Различия внутри каждой группы не являются статистически значимыми при данном размере выборки. Отдельно выделяется Llama 3.3 70B Instruct (5,88%), чей результат статистически значимо ниже всех остальных моделей.

Сравнительный анализ: планирование против исполнения

Двумерная визуализация результатов тестирования моделей (Рис. 1) выявляет профили способностей, которые не были бы видны на одномерной визуализации. Из этого сравнительного анализа вытекает несколько наблюдений. Существует сильная положительная корреляция с примечательными исключениями: модели, преуспевающие в планировании, как правило, также преуспевают в исполнении, что предполагает общие базовые способности. Однако корреляция несовершенна. Наиболее выраженным отклонением является Llama 3.3 70B Instruct, достигающая умеренной производительности в планировании (58,39%), но лишь 5,88% в исполнении. Как показано в анализе результатов K-Benchmark, основной причиной столь низкого показателя являются систематические ошибки вызова инструмента (свыше 80% неуспешных вызовов), а не неспособность к стратегическому рассуждению как таковому. Данное наблюдение подчёркивает, что разрыв между планированием и исполнением может объясняться не только различием когнитивных способностей модели, но и техническими ограничениями интеграции с инструментальным интерфейсом.

Claude Sonnet 4.5, GPT-5 и Qwen 3 Max группируются в высокопроизводительной области по обоим измерениям. В рамках оцениваемого набора задач данные модели демонстрируют наиболее высокие результаты, однако следует учитывать, что различия между ними и ближайшими конкурентами (GLM 4.6) не достигают статистической значимости для K-Benchmark. Также примечательно расхождение способностей разных вариантов БЯМ Gemini: Gemini 2.5 Pro показывает высокие результаты в планировании (72,16%), но лишь умеренные – в исполнении (58,82%), в то время как Gemini 2.5 Flash демонстрирует гораздо лучшее исполнение (47,06%), чем планирование (37,66%). Данное наблюдение позволяет предположить, что оптимизации, применяемые для создания облегчённых версий моделей, могут непропорционально влиять на различные аспекты способностей к тестированию на проникновение. Однако без доступа к деталям архитектурных отличий между вариантами Pro и Flash данный вывод остаётся гипотетическим.



Р и с. 1. Двумерная визуализация результатов тестирования БЯМ

F i g. 1. Two-dimensional visualization of LLM testing results

Анализ режимов отказа

Качественный анализ неудачных выполнений выявил несколько систематических режимов отказа, ограничивающих текущую безопасность развертывания.

Наблюдались галлюцинации БЯМ при выборе целей и генерации команд, при которых модели «выдумывали» IP-адреса целей и атаковали системы, отсутствующие в тестовой среде.

При тестировании на K-Benchmark одна из моделей атаковала собственную хост-систему вместо назначенного целевого контейнера. Другая модель «выдумывала» флаги завершения, когда не могла найти фактическое решение задания.

Отказ от задачи и дрейф происходили, когда несколько моделей прекращали выполнение поставленных задач на середине выполнения для совершения не связанных с задачей действий. В одном случае агент DeepSeek прервал задачу повышения привилегий для решения квадратного уравнения. Агент Llama принял неуместную персону («EroticoBot») во время деятельности по тестированию на проникновение и составлял финальный отчет о своих действиях от имени этой персоны в третьем лице, завершив его набором случайных слов на различных языках, не связанных с тематикой тестирования на проникновение.

В одном случае было зафиксировано отклонение агента-оценщика от своей задачи: вместо объективной оценки выполнения задачи агентом-пентестером оценщик самостоятельно выполнил задание, имея все необходимые подсказки в контексте, и засчитал результат агенту-пентестеру. Данный случай был идентифицирован при ручном обзоре отчётов оценщика и засчитан как неуспешное прохождение задачи, поскольку тестируемая модель фактически не выполнила задание. По результатам анализа инцидента были приняты корректирующие меры: промпт агента-оценщика был доработан с введением явного запрета на самостоятельное выполнение задач и указанием ограничиться верификацией артефактов, оставленных агентом-пентестером; кроме того, для всех результатов K-Benchmark введена процедура ручной проверки отчётов оценщика на предмет аналогичных отклонений. Других случаев подобного поведения оценщика обнаружено не было.

При длительном выполнении задачи у ряда моделей наблюдалась потеря контекста, при которой модели забывали ранее обнаруженную информацию, повторяли ранее опробованные подходы или теряли из виду явно определенные ограничения области действия. Эта деградация была особенно выражена в сложных многошаговых сценариях.

Неправильное использование инструментов также было частой причиной отказа: модели выбирали неподходящие инструменты для заданных сценариев или предоставляли некорректные параметры для правильных инструментов. Распространенные ошибки включали неправильно отформатированные аргументы команд, несовместимые по версии флаги и попытки использовать инструменты, недоступные в среде.

Обсуждение

Интерпретация результатов

Наши результаты демонстрируют, что современные БЯМ обладают значимым уровнем способностей для задач тестирования на проникновение, достигая уровня успеха, который был бы полезен в рабочих процессах с постоянным контролем со стороны эксперта-человека. Наиболее производительные модели корректно идентифицируют подходящие следующие шаги примерно в трех четвертях сценариев планирования и успешно выполняют аналогичные доли задач по реализации техник тестирования на проникновение. Однако результаты также выявляют существенные ограничения. Ни одна модель не приблизилась к производительности эксперта-человека, который достигал бы почти идеальных оценок на курируемых сценариях, составляющих наши бенчмарки. Задокументированные систематические режимы отказа – галлюцинации, дрейф задачи, нарушение области действия – представляют критичные для безопасности проблемы, исключающие полностью автономное развертывание.

Разрыв между планированием и исполнением, наблюдаемый у некоторых моделей (наиболее драматично у Llama 3.3 70B Instruct), предполагает, что эти способности опираются на частично различные компетенции. Модель может корректно рассуждать о прогрессии атаки в абстрактных терминах, но не может перевести это рассуждение в корректные вызовы инструментов и интерпретацию вывода. Этот вывод имеет последствия как для развертывания (различные требования к контролю для фаз планирования и исполнения), так и для исследований (потенциал целенаправленного улучшения способностей).

Вместе с тем необходимо чётко обозначить границы внешней валидности полученных результатов. Текущая реализация K-Benchmark охватывает ограниченный набор контейнеризованных Linux-сред с тремя тактическими категориями MITRE ATT&CK. Крупные корпоративные сети с доменами Active Directory, многосегментные сетевые топологии, сложные облачные и гибридные среды, а также задачи, требующие графического интерфейса, не представлены в текущей реализации. CSL-Benchmark, в свою очередь, оценивает локальный выбор следующего шага, а не способность к долгосрочному стратегическому планированию полных цепочек атак. Таким образом, представленные результаты характеризуют способности моделей в рамках специфического и ограниченного набора задач. Они не должны восприниматься как исчерпывающая оценка *offensive-security* возможностей современных БЯМ-агентов и не могут быть непосредственно экстраполированы на промышленные сценарии тестирования на проникновение без дополнительной верификации.

Выводы для практиков

Организации, рассматривающие тестирование на проникновение с поддержкой ИИ, должны интерпретировать наши результаты с учетом нескольких соображений. Человеческий контроль остается существенным: текущие способности БЯМ поддерживают усиление, а не замену экспертов-людей. Даже наиболее производительные модели не справляются примерно с четвертью задач планирования и исполнения. Характер неудач – включая нарушения области действия и выдуманные результаты – означает, что неконтролируемая работа представляет реальный риск нанесения вреда. Также важен выбор модели: производительность среди оцениваемых моделей варьируется более чем в 2 раза. Организации должны оценивать конкретные модели на репрезентативных задачах, а не предполагать переносимость способностей с общих бенчмарков. Планирование и исполнение могут требовать разных моделей: профили способностей, выявленные нашим двумерным анализом, предполагают, что разные модели могут быть оптимальны для разных этапов рабочего процесса. Заявления поставщиков требуют верификации: существенная вариация в производительности моделей и возникновение режимов отказа, не очевидных из агрегированной статистики, подчерки-

вают важность независимой оценки перед решениями о развертывании.

Выводы для безопасности ИИ

Наши результаты имеют последствия для продолжающихся дискуссий о способностях и рисках ИИ в доменах, релевантных для кибербезопасности. Текущие модели представляют ограниченную автономную угрозу: хотя БЯМ могут помогать с задачами тестирования на проникновение, задокументированные уровни ошибок и режимы отказа предполагают, что текущие модели не могут надежно выполнять сложные атаки без человеческого руководства. Это обнадеживает в части перспективы краткосрочных угроз, но не должно экстраполироваться на будущие поколения моделей. Поэтому при дальнейшем развитии данных способностей ИИ, важно учитывать их возможное двойное назначение: улучшения в способностях планирования и исполнения, приносящие пользу защитному тестированию на проникновение, в равной мере выгодны для наступательных применений злоумышленниками. Представленные здесь бенчмарки могут использоваться для отслеживания развития способностей и информирования политических дискуссий о пороговых значениях способностей, требующих дополнительной проверки. Продемонстрированные режимы отказа также поднимают важный вопрос надёжности ИИ-оценщика. Валидация оценщика CSL-Benchmark показала высокую, но не совершенную согласованность с экспертным суждением (к = 0,797), при этом незначительное преобладание ложноположительных ошибок указывает на возможное завышение результатов в пределах 2 процентных пунктов. Задокументированный единичный случай самостоятельного выполнения задания оценщиком K-Benchmark, хотя и не повлиявший на итоговые результаты, подчёркивает риски использования БЯМ для оценки БЯМ и необходимость контролирующих процедур.

Ограничения и направления будущих исследований

Несколько ограничений сдерживают текущее исследование. CSL-Benchmark абстрагирует реальный контекст атакующего в курированные текстовые описания, потенциально ставя в невыгодное положение модели, которые выиграли бы от более богатой обратной связи от среды. Оценка локального выбора следующего шага, а не полных стратегий атаки, может упустить проблемы в долгосрочном планировании. K-Benchmark использует одну рабочую станцию Kali Linux и конечный набор контейнеризованных целей. Крупные корпоративные сети, многошаговое горизонтальное перемещение, домены Active Directory и сложные облачные среды не представлены в текущей реализации. Интерфейс взаимодействия только через *bash*-терминал исключает техники, требующие графических интерфейсов. Для CSL-Benchmark проведена валидация БЯМ-оценщика относительно экспертного консенсуса, продемонстрировавшая высокую согласованность (к = 0,797). Для K-Benchmark аналогичная систематическая валидация оценщика пока не проведена: верификация артефактов в контейнерных средах требует привлечения экспертов с непосредственным доступом к целевым системам, что существенно усложняет процедуру. Выявленный и описанный в разделе «Анализ режимов отказа» случай отклонения оценщика от своей задачи, хотя и единичный, указывает на принципиальную возможность некорректного поведения БЯМ-оценщика в данном контексте. Корректирующие меры (доработка промпта и ручная проверка отчётов) снижают риск повторения, однако не устраняют его полностью. Систематическая валидация оценщика K-Benchmark на размеченной экспертами выборке задач является приоритетным направлением будущей работы. Покрытие матрицы MITRE ATT&CK пока остается частичным, со множеством ещё не реализованных техник. Статистическая мощность для K-Benchmark ограничена единичными испытаниями на задачу.

Для снижения влияния этих ограничений нами запланирован ряд расширений для будущей работы.

Для CSL-Benchmark:

- добавление дополнительных машин и отчетов с особым вниманием к недопредставленным операционным системам, типам сервисов и техникам MITRE ATT&CK;
- добавление новых метрик для детальной оценки;
- разработка программного фреймворка для удобного запуска бенчмарка.

Для K-Benchmark:

- реализация дополнительных техник ATT&CK, особенно в рамках тактик Перемещение внутри периметра, Предотвращение обнаружения и Получение учетных данных;
- разработка вех для гранулярной оценки завершения задач;
- создание многоэтапных связанных сценариев, требующих последовательности техник;
- дополнительные расширения включают более богатые метрики, фиксирующие время до цели, количество команд, избыточность подхода и индикаторы скрытности;
- усложнение среды через сценарии с доменами Active Directory и многосегментными сетевыми топологиями;
- валидацию оценщика через систематическое сравнение с суждением эксперта-человека.

Доступность данных и кода

Для обеспечения воспроизводимости результатов и содействия дальнейшим исследованиям в данной области планируется публичный выпуск следующих компонентов: набор данных CSL-Benchmark, Docker-образы K-Benchmark, код для запуска экспериментов и промпты, использованные для агентов и оценщиков. Публикация планируется на платформе GitVerse под открытой лицензией после принятия статьи к публикации.

Заключение

В настоящей статье представлена двумерная структура для оценки способностей БЯМ в тестировании на проникновение, устраняющая критический пробел в существующих методологиях оценки. Посредством CSL-Benchmark (846 сценариев планирования) и K-Benchmark (реализации 34 техник MITRE ATT&CK) мы продемонстрировали, что значимая оценка способностей требует разделения стратегического рассуждения от операционного исполнения.

Оценка одиннадцати современных моделей выявила существенную вариацию как в способностях к планированию, так и в способностях к исполнению, при этом наиболее производительные модели достигают приблизительно 75% долей успеха

в рамках оцениваемых задач, в то время как другие модели показывают значительно более низкие результаты. Валидация оценщика CSL-Benchmark подтвердила высокую согласованность автоматизированной оценки с экспертным суждением; систематическая валидация оценщика K-Benchmark запланирована как приоритетное направление дальнейшей работы. Двумерный анализ выявил профили способностей – такие как сильное планирование, но слабое исполнение – которые были бы скрыты при одномерной оценке.

Отдельную ценность представляют задокументированные систематические режимы отказа, включающие галлюцинации, нарушения области действия, отказ от задач и потеря контекста, представляющие риски безопасности для автономного развертывания. Эти результаты подтверждают вывод о том, что текущие БЯМ демонстрируют значимый потенциал для усиления тестирования на проникновение, но остаются небезопасными для неконтролируемой работы.

Представленные бенчмарки обеспечивают основу для отслеживания развития способностей, принятия решений о развертывании и направления исследований к целенаправленному улучшению способностей. По мере того, как тестирование безопасности с поддержкой ИИ переходит от экспериментов к промышленному развертыванию, строгие и системные подходы к оценке качества выполнения задач становятся все более важными для отделения подлинного прогресса от маркетинговых заявлений и обеспечения того, чтобы решения о развертывании основывались на воспроизводимых доказательствах.

Благодарности

Данное исследование проведено совместно Лабораторией кибербезопасности ПАО Сбербанк и АО «Лаборатория Касперского». Авторы выражают благодарность специалистам по тестированию на проникновение, внесшим вклад в валидацию сценариев, и инфраструктурным командам, обеспечившим разработку и поддержку среды для тестирования.

Acknowledgements

This research was conducted jointly by the Cybersecurity Research Laboratory of PJSC Sberbank and JSC Kaspersky Lab. The authors express their gratitude to the penetration testing specialists who contributed to the validation of the scenarios, and to the infrastructure teams that supported the development and maintenance of the testing environment.

Вклад авторов

А.М. Кузьмин – разработал концепцию и методологию бенчмарков, спроектировал и реализовал CSL-Benchmark, провёл экспериментальную оценку моделей, выполнил анализ результатов и подготовил рукопись статьи.

Д.В. Латохин – выполнил анализ матрицы MITRE ATT&CK для оценки реализуемости техник в контейнеризированных средах, осуществил отбор приоритетных техник и разработал докер-контейнеры для K-Benchmark, а также участвовал в редактировании статьи.

А.А. Анистратенко – провёл валидацию и доработку докер-контейнеров, разработал инфраструктуру для запуска экспериментов и участвовал в редактировании статьи.

Е.С. Афонин – выполнил экспертную проверку и валидацию шагов CSL-Benchmark, а также участвовал в редактировании статьи.

А.М. Иванов и А.В. Лискин – оказали организационную помощь для проведения исследований.

References

[1] Parkar S., Mishra D.K. Cybersecurity Workforce Development and Training: A Comprehensive Review on the Significance, Strategies, Opportunities and Challenges. In: 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS). Gurugram, India: IEEE Press; 2024. p. 1-5. https://doi.org/10.1109/ISCS61804.2024.10581241

[2] Hassanin M., Moustafa N. A Comprehensive Overview of Large Language Models (LLMs) for Cyber Defences: Opportunities and Directions. *arXiv:*2405.14487. 2024. https://doi.org/10.48550/arXiv.2405.14487

[3] Yao Y., et al. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly. *High-Confidence Computing*. 2024;4(2):100211. https://doi.org/10.1016/j.hcc.2024.100211

[4] Xiong W., et al. Cyber Security Threat Modeling Based on the MITRE Enterprise ATT&CK Matrix. *Software and Systems Modeling.* 2022;21(1):157-177. https://doi.org/10.1007/s10270-021-00898-7

[5] Shen X., et al. Decoding the MITRE Engenuity ATT&CK Enterprise Evaluation: An Analysis of EDR Performance in Real-World Environments. In: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security (ASIA CCS '24). New York, NY, USA: Association for Computing Machinery; 2024. p. 96-111. https://doi.org/10.1145/3634737.3645012

[6] Zambianco M., Facchinetti C., Siracusa D. A Proactive Decoy Selection Scheme for Cyber Deception Using MITRE ATT&CK. *Computers & Security*. 2025;148:104144. https://doi.org/10.1016/j.cose.2024.104144

[7] Sánchez-Zas C., et al. Dynamic Characterisation of Cyberattacks Based on the MITRE ATT&CK Framework Applied to the Optimisation of a Mitigation Selection Process. *Future Generation Computer Systems*. 2026;177:108272. https://doi.org/10.1016/j.future.2025.108272

[8] Pratama D., et al. CIPHER: Cybersecurity Intelligent Penetration-Testing Helper for Ethical Researcher. *Sensors*. 2024;24(21):6878. https://doi.org/10.3390/s24216878

[9] Deng G., et al. PENTESTGPT: Evaluating and Harnessing Large Language Models for Automated Penetration Testing. In: Proceedings of the 33rd USENIX Conference on Security Symposium. 2024. p. 847-864. https://doi.org/10.48550/arXiv.2308.06782

[10] Happe A., Kaplan A., Cito J. LLMs as Hackers: Autonomous Linux Privilege Escalation Attacks. *Empirical Software Engineering*. 2026;31(3):70. https://doi.org/10.1007/s10664-025-10758-3

[11] Fang R., et al. LLM Agents Can Autonomously Exploit One-Day Vulnerabilities. *arXiv:*2404.08144. 2024. https://doi.org/10.48550/arXiv.2404.08144

[12] Xu J., et al. AutoAttacker: A Large Language Model Guided System to Implement Automatic Cyber-Attacks. *arXiv:*2403.01038. 2024. https://doi.org/10.48550/arXiv.2403.01038
[13] Shen X., et al. PentestAgent: Incorporating LLM Agents to Automated Penetration Testing. In: Proceedings of the 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25). New York, NY, USA: Association for Computing Machinery; 2025. p. 375-391. https://doi.org/10.1145/3708821.3733882
[14] Muzsai L., Imolai D., Lukács A. HackSynth: LLM Agent and Evaluation Framework for Autonomous Penetration Testing. *arXiv:*2412.01778. 2024. https://doi.org/10.48550/arXiv.2412.01778
[15] Kong H., et al. VulnBot: Autonomous Penetration Testing for a Multi-Agent Collaborative Framework. *arXiv:*2501.13411. 2025. https://doi.org/10.48550/arXiv.2501.13411
[16] Wan S., et al. CyberSecEval 3: Advancing the Evaluation of Cybersecurity Risks and Capabilities in Large Language Models. *arXiv:*2408.01605. 2024. https://doi.org/10.48550/arXiv.2408.01605
[17] Zhang A.K., et al. Cybench: A Framework for Evaluating Cybersecurity Capabilities and Risks of Language Models. In: The Thirteenth International Conference on Learning Representations (ICLR 2025). 2025. https://doi.org/10.48550/arXiv.2408.08926
[18] Shao M., et al. NYU CTF Bench: A Scalable Open-Source Benchmark Dataset for Evaluating LLMs in Offensive Security. *Advances in Neural Information Processing Systems*. 2024;37:57472-57498. https://doi.org/10.52202/079017-1832
[19] Anurin A., et al. Catastrophic Cyber Capabilities Benchmark (3CB): Robustly Evaluating LLM Agent Cyber Offense Capabilities. In: Workshop on Datasets and Evaluators of AI Safety (AIRR Workshop, NeurIPS 2024). 2024. https://doi.org/10.48550/arXiv.2410.09114
[20] Alam M.T., et al. CTIBench: A Benchmark for Evaluating LLMs in Cyber Threat Intelligence. In: Advances in Neural Information Processing Systems. Vancouver, Canada; 2024. Vol. 37. p. 50805-50825. https://doi.org/10.52202/079017-1607
[21] Gioacchini L., et al. AutoPenBench: A Vulnerability Testing Benchmark for Generative Agents. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP 2025). Association for Computational Linguistics; 2025. p. 1615-1624. https://doi.org/10.18653/v1/2025.emnlp-industry.114
[22] Isozaki I., et al. Towards Automated Penetration Testing: Introducing LLM Benchmark, Analysis, and Improvements. In: Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization. New York, NY, USA: Association for Computing Machinery; 2025. p. 404-419. https://doi.org/10.1145/3708319.3733804
[23] Happe A., Cito J. Understanding Hackers' Work: An Empirical Study of Offensive Security Practitioners. In: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York, NY, USA: Association for Computing Machinery; 2023. p. 1669-1680. https://doi.org/10.1145/3611643.3613900
[24] Happe A., Cito J. Benchmarking Practices in LLM-driven Offensive Security: Testbeds, Metrics, and Experiment Design. *arXiv:*2504.10112. 2025. https://doi.org/10.48550/arXiv.2504.10112
[25] Comanici G., et al. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv:*2507.06261. 2025. https://doi.org/10.48550/arXiv.2507.06261
[26] Yang A., et al. Qwen3 Technical Report. *arXiv:*2505.09388. 2025. https://doi.org/10.48550/arXiv.2505.09388
[27] GLM T., et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv:*2406.12793. 2024. https://doi.org/10.48550/arXiv.2406.12793
[28] Liu A., et al. DeepSeek-V3 Technical Report. *arXiv:*2412.19437. 2024. https://doi.org/10.48550/arXiv.2412.19437
[29] Grattafiori A., et al. The Llama 3 Herd of Models. *arXiv*:2407.21783. 2024. https://doi.org/10.48550/arXiv.2407.21783
[30] Agarwal S., et al. GPT-OSS-120B & GPT-OSS-20B Model Card. *arXiv:*2508.10925. 2025. https://doi.org/10.48550/arXiv.2508.10925
[31] Singer B., et al. On the Feasibility of Using LLMs to Autonomously Execute Multi-Host Network Attacks. *arXiv:*2501.16466. 2025. https://doi.org/10.48550/arXiv.2501.16466

Поступила 08.12.2025; одобрена после рецензирования 14.02.2026; принята к публикации 24.03.2026.
Submitted 08.12.2025; approved after reviewing 14.02.2026; accepted for publication 24.03.2026.

Об авторах:
Кузьмин Александр Михайлович, исполнительный директор, Лаборатория кибербезопасности, Публичное акционерное общество «Сбербанк России» (117312, Российская Федерация, *г.* Москва, ул. Вавилова, д. 19), **ORCID: https://orcid.org/0000-0001-8352-5811,** alex.kuzmin@bk.ru
Латохин Дмитрий Владимирович, руководитель группы классификации программного обеспечения, АО «Лаборатория Касперского» (125212, Российская Федерация, г. Москва, Ленинградское шоссе, д. 39А, стр. 2), **ORCID: https://orcid.org/0009-0006-8158-2218**, dmitry.latokhin@kaspersky.com
Анистратенко Александр Артурович, руководитель направления, Лаборатория кибербезопасности, Публичное акционерное общество «Сбербанк России» (117312, Российская Федерация, *г.* Москва, ул. Вавилова, д. 19), **ORCID: https://orcid.org/0009-0006-4110-2080,** aanistratenko@yandex.ru
Афонин Евгений Сергеевич, исполнительный директор, Лаборатория кибербезопасности, Публичное акционерное общество «Сбербанк России» (117312, Российская Федерация, *г.* Москва, ул. Вавилова, д. 19), **ORCID: https://orcid.org/0009-0002-3737-149X**, afonin.es@gmail.com

Лискин Александр Викторович, руководитель Управления исследования угроз, АО «Лаборатория Касперского» (125212, Российская Федерация, г. Москва, Ленинградское шоссе, д. 39А, стр. 2), **ORCID: https://orcid.org/0009-0006-2029-9136,** alexander.liskin@kaspersky.com
Иванов Антон Михайлович, технический директор, Департамент исследований и разработки, АО «Лаборатория Касперского» (125212, Российская Федерация, г. Москва, Ленинградское шоссе, д. 39А, стр. 2), **ORCID: https://orcid.org/0009-0000-4753-9132**, anton.m.ivanov@kaspersky.com

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Alexander M. Kuzmin, Executive Director of Research of the Cybersecurity Research Laboratory, Sberbank of Russia (19 Vavilova St., Moscow 117312, Russian Federation), **ORCID: https://orcid.org/0000-0001-8352-5811,** alex.kuzmin@bk.ru
Dmitry V. Latokhin, Manager of the Software Classification Group, AO Kaspersky Lab, AO Kaspersky Lab (39A/2 Leningradskoe Shosse, Moscow 125212, Russian Federation), **ORCID: https://orcid.org/0009-0006-8158-2218**, dmitry.latokhin@kaspersky.com
Alexander A. Anistratenko, Research Scientist of the Cybersecurity Research Laboratory, Sberbank of Russia (19 Vavilova St., Moscow 117312, Russian Federation), **ORCID: https://orcid.org/0009-0006-4110-2080,** aanistratenko@yandex.ru
Eugene S. Afonin, Principal Research Scientist of the Cybersecurity Research Laboratory, Sberbank of Russia (19 Vavilova St., Moscow 117312, Russian Federation), **ORCID: https://orcid.org/0009-0002-3737-149X**, afonin.es@gmail.com
Alexander V. Liskin, Head of the Threat Research, AO Kaspersky Lab (39A/2 Leningradskoe Shosse, Moscow 125212, Russian Federation), **https://orcid.org/0009-0006-2029-9136,** alexander.liskin@kaspersky.com
Anton M. Ivanov, Chief Technology Officer of the Research & Development, AO Kaspersky Lab (39A/2 Leningradskoe Shosse, Moscow 125212, Russian Federation), **ORCID: https://orcid.org/0009-0000-4753-9132**, anton.m.ivanov@kaspersky.com

All authors have read and approved the final manuscript.