УДК 004.896

Галимов Р.А.
Студент/магистрант
НИЯУ МИФИ, г. Москва
Костюк М.С.
Студент/магистрант
НИЯУ МИФИ, г. Москва

МОДЕЛЬ УПРАВЛЕНИЯ ИТ-РИСКАМИ ПРИ ВНЕДРЕНИИ РЕШЕНИЙ INFRASTRUCTURE AS CODE С КОМПОНЕНТАМИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

***Аннотация**: В статье рассмотрены актуальные вызовы управления информационными рисками при комбинированном применении технологий Infrastructure as Code (IaC) и искусственного интеллекта в корпоративных информационных системах. Предлагается двухуровневая модель идентификации и оценки рисков, учитывающая специфику автоматизированной генерации инфраструктурных конфигураций. Разработана матрица рисков с категоризацией по технической, процессной и организационной составляющим. В работе обоснована необходимость специализированных механизмов контроля качества кода, генерируемого нейросетевыми моделями, и предложены практические рекомендации для внедрения в DevSecOps-конвейерах российских организаций.*

***Ключевые слова**: Infrastructure as Code (IaC), управление ИТ-рисками, искусственный интеллект, DevOps, информационная безопасность, конфигурационный контроль, автоматизация, генеративные модели, таксономия рисков, цифровая трансформация.*

Трансформация моделей управления ИТ-инфраструктурой предприятий характеризуется переходом от традиционных подходов ручного администрирования к парадигме Infrastructure as Code (IaC), позволяющей

описывать и воспроизводить конфигурации через программный код. Одновременно развитие технологий машинного обучения и генеративных моделей открывает возможности для автоматизации процесса создания самих IaC-шаблонов, что существенно ускоряет время вывода решений на рынок [1]. Однако внедрение подобных систем порождает новый спектр информационных рисков, которые традиционные методологии управления рисками (COBIT, ISO 27001) не охватывают в полной мере.

Специфика рисков, связанных с IaC+AI-интеграцией, заключается в том, что ошибки в инфраструктурных конфигурациях немедленно тиражируются на все развернутые окружения, а недоконтролируемые AI-системы могут генерировать код, несущий скрытые уязвимости [2]. Российские организации, осуществляющие цифровую трансформацию в условиях импортозамещения технологий, особо нуждаются в локализированных методиках оценки и снижения подобных рисков.

Целью настоящего исследования является разработка комплексной модели управления ИТ-рисками, адаптированной к контексту применения IaC и генеративного ИИ, включающей инструментарий для идентификации, оценки, обработки и мониторинга рисков на всех этапах жизненного цикла инфраструктурных решений.

Misконфигурация является наиболее распространенной причиной инцидентов безопасности в облачных окружениях, развернутых посредством IaC. Поскольку IaC-код описывает всю архитектуру окружения в единой системе версий, ошибка в одном шаблоне может привести к одновременному развертыванию уязвимой конфигурации на множество ресурсов [3]. Типичные примеры включают развертывание баз данных с публичным доступом, открытие портов SSH в инфраструктуре, назначение чрезмерно широких IAM-прав для сервисных аккаунтов.

При применении AI-систем для генерации IaC-кода риск misконфигурации усложняется непредсказуемостью поведения нейросетевых моделей. Генеративные

модели, обученные на исторических данных кодовых репозиториев, могут воспроизводить типичные ошибки безопасности, присутствующие в обучающем наборе данных, не имея явного понимания контекста безопасности [4]. Следовательно, требуется дополнительный уровень автоматизированной проверки сгенерированного кода перед его применением к production-окружениям.

Хранение конфиденциальной информации непосредственно в IaC-шаблонах (hardcoded secrets) остается актуальной проблемой, несмотря на наличие специализированных систем управления секретами [3]. Разработчики часто размещают API-ключи, пароли к базам данных, сертификаты и токены доступа в виде переменных окружения или строк конфигурации, допуская закоммичивание таких данных в системы контроля версий.

В контексте использования AI-инструментов для генерации конфигураций угроза усугубляется возможностью обучения нейросетевых моделей на репозиториях, содержащих реальные учетные данные. Модель может непреднамеренно воспроизводить паттерны, включающие элементы конфиденциальной информации, которая была видима ей во время обучения [5]. Кроме того, если промежуточные результаты работы AI-системы (embedding-векторы, логи генерации) передаются к внешним сервисам обработки, это создает дополнительный канал утечки.

Внедрение IaC требует качественно новых подходов к организации разработки и развертывания инфраструктуры. Операционные риски включают недостаточную экспертизу команды в области современных практик DevSecOps, отсутствие формализованных процессов code review для инфраструктурного кода, слабое разграничение доступа к критическим элементам CI/CD-конвейеров [2]. Использование AI-систем добавляет дополнительное измерение сложности: команда должна понимать не только синтаксис и семантику генерируемого кода, но и ограничения и потенциальные уязвимости самих моделей ИИ.

Процессные риски также связаны с управлением версионированием инфраструктурного кода, откатом конфигураций при возникновении инцидентов, координацией между разработкой, DevOps и командой безопасности [6]. При внедрении автоматизированной генерации через ИИ возникает дополнительный вопрос: как организовать валидацию, утверждение и развертывание AI-генерированного кода, сохраняя скорость delivery и учитывая повышенные требования по контролю качества.

Сами системы искусственного интеллекта, используемые для генерации IaC, являются потенциальными источниками рисков [4]. К этой категории относятся: отравление обучающих данных (data poisoning), неправомерный доступ к моделям и их параметрам, утечка чувствительной информации через анализ выходов моделей (membership inference attacks), несоответствие поведения модели ожиданиям разработчиков [5].

Особую опасность представляет использование облачных AI-сервисов (например, OpenAI API, YandexGPT), где исходные промпты и сгенерированный код могут быть логированы и потенциально переиспользованы в целях обучения модели или анализа поведения клиента [6]. Для критичных инфраструктурных решений это может привести к раскрытию архитектурной информации или конфиденциальных деталей развертывания.

Первый уровень модели предусматривает систематическую идентификацию потенциальных рисков на всех этапах жизненного цикла IaC+AI-решения: планирование, разработка конфигурационных шаблонов, генерация кода через AI, валидация и тестирование, развертывание, мониторинг и поддержка [1].

Для каждого этапа выделены типовые категории рисков:

Этап планирования и проектирования: отсутствие требований безопасности в спецификации, недостаточное определение области применения AI-инструментов, непроработанность механизмов контроля качества.

Этап разработки: использование устаревших или небезопасных шаблонов IaC, интеграция с неудостоверенными AI-сервисами, отсутствие инструментов статического анализа кода.

Этап валидации: неполнота или неправильность набора тестов безопасности, отсутствие peer review для критичного кода, недостаточное тестирование поведения AI-модели в edge-cases.

Этап развертывания: несоответствие deployed-конфигурации утвержденной версии в репозитории, отсутствие механизмов отката, недостаточное разграничение доступа.

Этап мониторинга: отсутствие логирования изменений конфигурации, недостаточная видимость в процессы работы AI-систем, задержка в обнаружении инцидентов.

На втором уровне происходит количественная оценка выявленных рисков на основе вероятности возникновения и величины потенциального ущерба. Используется матрица рисков, в которой по горизонтали отложена вероятность (низкая, средняя, высокая), а по вертикали — влияние на бизнес-процессы (низкое, среднее, высокое, критичное) [2].

Для каждого идентифицированного риска определяется стратегия обработки: избежание (отказ от использования определенных AI-инструментов или конфигураций), снижение (внедрение дополнительных средств контроля), перенос (страхование, использование облачных сервисов с гарантиями SLA), принятие (осознанное решение жить с риском при условии его мониторинга) [3].

Принципиально важным является мониторинг остаточного риска после внедрения мер контроля. Это предполагает регулярный аудит инфраструктурных конфигураций, анализ логов доступа к системам управления секретами, отслеживание обновлений и патчей как для самих IaC-инструментов (Terraform, Ansible), так и для используемых AI-компонентов [4].

Рынок предлагает специализированные инструменты для статического анализа IaC-кода, выявляющие типичные ошибки и отклонения от best practices безопасности [6]. К числу наиболее распространенных относятся Checkov (поддерживает Terraform, CloudFormation, Kubernetes), Terrascan (универсальный сканер для множества IaC-платформ), TFsec (специализирован на анализе Terraform-кода), Kics (анализирует качество и безопасность инфраструктурных конфигураций).

Интеграция этих инструментов в CI/CD-конвейер обеспечивает автоматическую проверку каждого коммита, предотвращая развертывание небезопасных конфигураций в production. Однако следует отметить, что существующие сканеры оптимизированы для выявления типовых паттернов ошибок, и могут пропускать нетривиальные уязвимости, особенно те, которые появляются при комбинировании нескольких IaC-элементов [5].

Для обеспечения безопасности кода, сгенерированного AI-системами, предлагается многоуровневый подход [6]:

Первый уровень: задание строгих guardrails при генерации через промпт-инжиниринг, явное указание требований безопасности и соответствия стандартам (ISO 27001, NIST Cybersecurity Framework).

Второй уровень: применение статических анализаторов (упомянутые выше Checkov, TFsec), но с усиленным набором правил, специфичных для AI-генерированного кода.

Третий уровень: code review специалистом по безопасности, выполняющим проверку не только синтаксиса, но и логики конфигурации, обоснованности назначения ресурсов, соответствия архитектурным принципам.

Четвертый уровень: динамическое тестирование конфигурации в изолированной тестовой среде, выявление нежелательных побочных эффектов, проверка соответствия ожиданиям по перформансу и надежности.

Необходимо разделение хранилищ для конфиденциальной информации и кода конфигурации. Современные подходы предусматривают использование специализированных систем управления секретами (HashiCorp Vault, AWS Secrets Manager, YandexLockbox для российских организаций), откуда IaC-инструменты получают необходимые учетные данные во время выполнения [3].

При использовании AI-систем для генерации конфигураций критично обеспечить, чтобы промежуточные данные и результаты работы моделей не содержали реальных секретов. Это требует предварительной обработки входных данных, использования mock-данных в примерах и шаблонах, из которых обучается или которые используются для prompt-инжиниринга генеративных моделей [5].

Организациям, планирующим комбинированное применение IaC и AI-технологий, рекомендуется следующий набор мер:

На уровне планирования: явно определить требования к безопасности инфраструктурных решений, провести оценку соответствия выбранных AI-инструментов политикам безопасности организации, разработать стандарты и шаблоны для IaC-кода, включающие принципы наименьших привилегий, явного определения ресурсов и их назначений [7].

На уровне процессов: внедрить формализованный процесс code review для всех инфраструктурных изменений, обучить команду best practices DevSecOps, установить четкие процедуры утверждения и развертывания конфигураций, включив дополнительные проверки для AI-генерированного кода [8].

На уровне инструментов: интегрировать сканеры безопасности в CI/CD-конвейеры, настроить логирование всех доступов к системам управления секретами и CI/CD-платформам, настроить мониторинг отклонений развернутых конфигураций от утвержденных версий, рассмотреть использование локальных или приватных AI-моделей вместо облачных сервисов для критичных операций [9].

На уровне организации: назначить ответственное лицо за управление рисками IaC+AI-решений, проводить регулярные аудиты и penetration testing

инфраструктуры, инвестировать в повышение квалификации команды, создать knowledge base типичных ошибок и способов их предотвращения [10].

Предложенная двухуровневая модель управления ИТ-рисками обеспечивает систематический подход к идентификации, оценке и контролю рисков, специфичных для интеграции Infrastructure as Code и искусственного интеллекта. Модель охватывает технические аспекты (сканирование конфигураций, контроль качества code), процессные элементы (code review, утверждение изменений) и организационные составляющие (обучение, распределение ответственности) [11].

Применение предложенного инструментария позволяет организациям снизить вероятность инцидентов безопасности, связанных с мисконфигурацией, утечкой конфиденциальных данных и неконтролируемым поведением AI-систем. Особую актуальность модель приобретает для российских компаний, осуществляющих трансформацию ИТ-функций в условиях импортозамещения технологий и повышенных требований по информационной безопасности.

Направления дальнейших исследований включают разработку специализированных метрик для оценки эффективности мер по управлению рисками, анализ реальных инцидентов и их классификацию, адаптацию моделей управления рисками под специфику различных отраслей (финансовые услуги, критичная инфраструктура, здравоохранение).

Библиографический список

1. Ломакина, Е. Г. Модели управления ИТ-инфраструктурой предприятия / Е. Г. Ломакина, С. А. Лукасевич // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2009. № 2. С. 81–89. URL: https://cyberleninka.ru/article/n/modeli-upravleniya-it-infrastrukturoy-predpriyatiya
2. Бубакар, И. Онтологическое обеспечение управления рисками информационной безопасности / И. Бубакар // Вестник Дагестанского государственного технического университета. Технические науки. 2021. Т. 48, № 2. С. 71–81. URL-адрес:https://cyberleninka.ru/article/n/ontologicheskoe-obespechenie-upravleniya-riskami-informatsionnoy-bezopasnosti
3. Сагидова, М. Л. Методы исследования оценки риска информационной безопасности / М. Л. Сагидова // Инновации. Наука. Образование. 2024. № 88. С. 1453–1461. URL-адрес:https://cyberleninka.ru/article/n/issledovanie-metodov-otsenki-riskov-informatsionnoy-bezopasnosti
4. Особенности организации управленческих и технологических стандартов в условиях цифровой трансформации / А. В. Косов [и др.] // Вопросы управления. 2025. № 2. С. 45–58. URL: https://cyberleninka.ru/article/n/osobennosti-integratsii-upravlencheskih-i-tehnologicheskih-standartov-v-usloviyah-tsifrovoy-transformatsii
5. Хади, М. А. Универсальная система управления информационной безопасностью: организационно-правовые принципы / М. А. Хади // Вестник

Санкт-Петербургского университета МВД России. 2025. № 1 (109). С. 156–163. URL: https://cyberleninka.ru/article/n/universalnaya-sistema-upravleniya-informatsionnoy-bezopasnostyu-organizatsionno-pravovye-printsipy

6. Интеграция искусственного интеллекта в системы кибербезопасности / А. С. Иванов [и др.] // Научный лидер. 2024. № 7 (162). С. 12–18. URL: https://scilead.ru/article/5919-integratsiya-iskusstvennogo-intellekta-v-sist
7. Роль искусственного интеллекта в кибербезопасности / Н. В. Петрова // Универсум: техническая наука. 2023. № 11-3 (116). С. 42–46. URL: https://7universum.com/ru/tech/archive/item/16310
8. Внедрение инфраструктуры как кода (IaC) в России: актуальные проблемы и перспективы [Электронный ресурс] // SecurityLab. 2024. URL:https://www.securitylab.ru/blog/personal/paragraph/354484.php
9. Инфраструктура безопасности как код: как защитить облака от мисконфигураций [Электронный ресурс] // SecurityMedia. 2025. URL:https://securitymedia.org/info/bezopasnost-infrastructure-as-code-kak-zashchitit-oblaka-ot-miskonfiguratsiy.html
10. DevSecOps: как встроить безопасность в процессы разработки [Электронный ресурс] // АВГ. 2025. URL:https://www.awg.ru/news/devsecops-kak-vstroit-bezopasnost-v-protsessy-razrabotki/
11. Как автоматизировать безопасность с помощью DevSecOps и искусственного интеллекта [Электронный ресурс] // Tproger. 2024. URL:https://tproger.ru/articles/kak-avtomatizirovat-bezopasnost-s-pomoshhyu-devsecops-i-iskusstvennogo-intellekta