

УДК 004.056.53

Динамический анализ модельных сервисов

Юрьев Артемий Сергеевич¹

forhhpurpose@yandex.ru

SPIN-код: 7517-6977, ORCID: 0009-0003-0369-0422

Дацук Никита Александрович^{2(*)}

dazuk07@yandex.ru

SPIN-код: 4612-8029, ORCID: 0009-0006-5403-0960

¹ ИСП РАН, Москва, Россия² МГТУ им. Н.Э. Баумана, Москва, Россия

Аннотация. Рассмотрены современные подходы к динамическому анализу безопасности (DAST) для систем машинного обучения. Подчеркнута недостаточность классических методов DAST и необходимость внедрения специализированных проверок, характерных для моделей, таких как инъекции промптов или jailbreak-атаки.

Ключевые слова: динамический анализ, DAST, MLSecOps, GenAI

Введение. Стремительная интеграция искусственного интеллекта (ИИ) и машинного обучения (ML) в процессы организаций приводит к появлению нового класса приложений — модельных сервисов. Эти сервисы, часто развернутые с помощью средств виртуализации или контейнеров (например, Docker-образов [1]), представляют собой сложную и многогранную структуру, которая требует комплексного анализа безопасности.

Методы и материалы; результаты. Для систематизации особенностей и угроз модельных сервисов следует рассмотреть такие методологии, как OWASP Top 10 for LLM Applications [3], OWASP Machine Learning Security Top Ten, а также MITRE. Анализ источников, показывает, что для ML-систем наибольшее внимание следует уделять следующим угрозам:

- prompt-инъекции: манипулирование LLM для выполнения несанкционированных команд через входные данные;
- небезопасные данные для обучения: внедрение закладок или утечка данных (отравление данных);
- jailbreak модели: обход семантической фильтрации;
- небезопасный доступ: отсутствие ролевой модели разграничения прав доступа к модельному сервису;
- атаки на конфиденциальность (inference attacks, membership inference, model inversion);
- атаки на доступность (например, длинные запросы к модели);
- атаки на цепочки поставок (supply chain attacks, вредоносные библиотеки).

Поверхность атак на современный модельный сервис в общем ключе можно описать в виде следующего набора:

- атаки на протоколы взаимодействия с сервисом (например, REST (Representational State Transfer) или gRPC, а также функции API [4]);

- традиционные уязвимости (инъекции, обход авторизации, несанкционированное выполнение команд на сервере (RCE), XSS, CSRF);
- уязвимые компоненты (библиотеки, устаревшие версии ПО);
- небезопасные конфигурации (открытые порты, отсутствие шифрования);
- уязвимости инфраструктуры и средств виртуализации (например, небезопасные настройки Docker или Kubernetes).
- уязвимости кода применения (обвязки модели, функции преобразования данных, микросервис, пред-/постобработки данных (небезопасная десериализация);
- уязвимости в коде обучения модели (отравления данных, небезопасные источники данных);

Классический DAST сканер не может проанализировать логический вывод модели на предмет небезопасного контента (утечка персональных данных, вредоносный код), он ищет конкретные, заранее определенные, сигнатуры, что провоцирует неспособность обнаружить уязвимости в бизнес-логике.



Динамический анализ модельных сервисов

На основании вышесказанного, для комбинированного анализа (см. рисунок) предлагается обогатить текущую DAST практику новыми подходами.

Заключение. Динамический анализ безопасности модельных сервисов требует комбинированного подхода. Многослойная поверхность атаки, сочетающая традиционные WEB-уязвимости и специфические угрозы для моделей или ИИ, требует доработок динамического анализа для более полного покрытия. Необходимо интегрировать в процесс тестирования методы, способные к целенаправленным атакам на модели. Требуется DAST-анализ с акцентом на языковые специфики (например, небезопасные функции Python). Внедрение систем мониторинга позволит обогатить логическим и семантическим анализом вывода (инференса) моделей. Динамический анализ в этом процессе является не просто опцией, а строгой необходимостью, методология которого должна непрерывно развиваться и адаптироваться под новые векторы атак.

Список источников

- [1] Аскарров Т. Е. Архитектура docker и технологии оркестровки: docker swarm и kubernetes. *Научный Лидер*, 2024, № 20 (170), с. 9–12. EDN: JXOSVI.
- [2] Сидоров Е.С. 2.5. SAST и DAST: автоматизация тестирования безопасности. В кн. *Цифровая трансформация и проблемы информационной безопасности*. Санкт-Петербург, Санкт-Петербургский государственный экономический университет, 2023, с. 84–87.
- [3] OWASP Top 10 for Large Language Model Applications. URL: <https://owasp.org/www-project-top-10-for-large-language-model-applications/> (accessed 25.08.2025).
- [4] Юрьев А.С. Фаззинг полиморфных систем в структурах. *Труды Института системного программирования РАН*, 2024, т. 36, № 1, с. 45–60. [https://doi.org/10.15514/ISPRAS-2024-36\(1\)-4](https://doi.org/10.15514/ISPRAS-2024-36(1)-4)