УДК 004.896

СТАТИЧЕСКИЙ АНАЛИЗ ФРЕЙМВОРКОВ МАШИННОГО ОБУЧЕНИЯ

Менисов А.Б., Захаров О.О. (Военно-космическая академия им. А.Ф.Можайского)
Научный руководитель – к.т.н. Менисов А.Б.
(Военно-космическая академия им. А.Ф.Можайского)

Введение. Внедрение технологий искусственного интеллекта требует учитывать ряд возникающих рисков и новых угроз. Это целый стек технологий, состоящий из методов и алгоритмов машинного обучения, фреймворков машинного обучения (TensorFlow, PyTorch), а также инфраструктурных решений для их поддержки (облачные системы, специализированные аппаратные системы и др.). Помимо классических уязвимостей программного обеспечения, эти технологии являются источниками принципиально новых типов ошибок и уязвимостей, которые предоставляют новые возможности для проведения атак злоумышленниками [1].

Основная часть. Статический анализ кода показал дефекты, связанные с:
1. Неверной обработкой ошибок (error handling issues);
2. Проблемами хранения целых чисел (integer handling issues);
3. Неинициализированными членами классов (uninitialized members);
4. Нарушением потока управления (control flow issues);
5. Разыменовыванием нулевого указателя (null pointer dereference);
6. Некорректными выражениями (incorrect expression);
7. Проблемами использования API (API usage issues);
8. Различными другими ошибками (medium impact quality);
9. Нарушением доступа к данным из нескольких потоков (concurrent data access violation);
10. Небезопасной обработкой данных (insecure data handling);
11. Проблемами с производительностью (program hangs).

Перечисленные дефекты среднего уровня опасности могут повлечь за собой отказ в обслуживании (Denial of Service) – например, разыменовывание нулевого указателя может привести к аварийному завершению работы программы. Проблемы с хранением целых чисел могут повлечь за собой исполнение произвольного кода, привести к неопределенному поведению программы и, как следствие, аварийному завершению работы, либо нестабильной работе фреймворка. Нарушение потока управления влияет на логику работы программы, то есть в ряде случаев не будет выполняться ожидаемый код, поскольку условия его выполнения заведомо невозможные (dead code). Нарушение доступа к данным из разных потоков может привести к неопределенному поведению программы, что неприемлимо в системах, спроектированных с расчетом на высокую отказоустойчивость [2].

Выводы. Дефекты и уязвимости фреймворков машинного обучения могут повлечь за собой нарушение всех составляющих информационной безопасности – конфиденциальности, целостности и доступности: повреждения памяти могут повлечь за собой исполнение произвольного кода, несанкционированный доступ к конфиденциальной информации, а также атаки типа DoS (Denial of Service); неинициализированные переменные зачастую содержат «мусорные» данные – это может повлиять на стабильность работы систем искусственного интеллекта [3].

Список использованных источников:

1. Безопасное использование TensorFlow // URL: https://github.com/tensorflow/tensorflow/blob/master/SECURITY.md (дата обращения: 31.01.2023).

2. Papernot N. et al. Technical report on the cleverhans v2. 1.0 adversarial examples library //arXiv preprint arXiv:1610.00768. – 2016.
3. Melis M. et al. secml: A python library for secure and explainable machine learning //arXiv preprint arXiv:1912.10013. – 2019.