Гандзюк Д.А., Шевелев Р.В.

ВЫЯВЛЕНИЕ УЯЗВИМОСТЕЙ В СМАРТ-КОНТРАКТАХ С ПОМОЩЬЮ ГЛУБОКОГО ОБУЧЕНИЯ

Департамент программной инженерии и искусственного интеллекта ИМКТ ДВФУ
Научный руководитель – ст. преподаватель О.А. Крестникова

В настоящее время получила признание технология блокчейн [2], благодаря суждению, что следующий этап развития интернета - Web3, будет основываться именно на блокчейне и его возможностях безопасности и децентрализации. Крупные технологические компании вкладывают деньги в исследования в области блокчейна и внедряют эти технологии в свои продукты.

Неотъемлемой частью блокчейна являются смарт-контракты [2], которые являются программами, исполняемыми в особой среде блокчейна и запускающими децентрализованные приложения. Смарт-контракты хранятся публично, а также не имеют возможности быть изменёнными после загрузки в блокчейн, поэтому критически важно проводить тщательную проверку кода на возможные уязвимости. Проведение аудита поможет выявить ошибки в безопасности на ранней стадии, что защитит от возможной кражи средств и других рисков.

На данный момент используют следующие методы обнаружения уязвимостей в смарт-контрактах:

- статический анализ [2] — метод анализа программного кода, при котором происходит оценка его структуры, синтаксиса и свойств без фактического выполнения программы (производится специальным ПО).
- динамический анализ [1] — метод анализа программного кода, который осуществляется путем фактического выполнения программы или её части.
- ручной анализ — анализ программного кода человеком-аудитором без использования ПО.
- фаззинг-тестирование [1] — техника тестирования, проверяющая может ли при обработке некорректных входных данных возникнуть странности в поведении программы, которые не закладывали разработчики.

В таблице ниже представлено сравнение нескольких программных средств для проведения аудита кода смарт-контракта на основе нескольких ключевых критериев, которые используются для оценки качества работы алгоритмов в задачах машинного обучения.

Таблица

Сравнение программных средств

Название ПС	Oyente[3]	Mythril[4]	Securify[5]	Slither[6]	MythX[7]
Тип анализа	Статический	Статический	Статический	Статический	Динамический
Доступность	Бесплатно	Бесплатно	Бесплатно	Бесплатно	Платно

Продолжение таблицы

Точность	Средняя	Высокая	Низкая	Средняя	Средняя
Полнота	Высокая	Высокая	Средняя	Средняя	Средняя
Удобство использования	Среднее	Высокое	Низкое	Высокое	Высокое
Производительность	Низкая	Средняя	Высокая	Высокая	Низкая

Как видно из таблицы, рассмотренные программные средства, за исключением одного – Mythril, не предоставляют инструментов для наиболее полного и точного анализа кода смарт-контракта в целях выявить в нём возможные уязвимости. Помимо этого, все представленные программные средства не используют самые актуальные данные во время проведения анализа. Следовательно, требуется разработка системы, способной с максимально высокой точностью и полнотой анализировать код, обучаясь на самых актуальных данных в области смарт-контрактов.

Список литературы

1. Использование нейросетей для анализа кода / [Электронный ресурс] // Deep-learning Based Solution for smart contract vulnerabilities detection: [сайт] — URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10654660/ (дата обращения: 20.03.2024).
2. Глоссарий блокчейн-терминов / [Электронный ресурс] // Glossary of blockchain: [сайт] — URL: https://blockchaintrainingalliance.com/pages/glossary-of-blockchain-terms (дата обращения: 20.03.2024).
3. Oyente / [Электронный ресурс] // [сайт] — URL: https://github.com/enzymefinance/oyente (дата обращения: 22.03.2024).
4. Mythril / [Электронный ресурс] // [сайт] — URL: https://github.com/Consensys/mythril (дата обращения: 22.03.2024).
5. MythX / [Электронный ресурс] // [сайт] — URL: https://mythx.io/ (дата обращения: 25.03.2024).
6. Securify / [Электронный ресурс] // [сайт] — URL: https://github.com/eth-sri/securify2 (дата обращения: 27.03.2024).
7. Slither / [Электронный ресурс] // [сайт] — URL: https://github.com/crytic/slither (дата обращения: 28.03.2024).