УДК 004.056.5:336.74

API-ОРІЄНТОВАНИЙ ІНСТРУМЕНТАРІЙ ПЕРЕВІРКИ ЦІЛІСНОСТІ СМАРТКОНТРАКТІВ ETHEREUM

Ковальчук О.Я., доктор юридичних наук, кандидат фізико-математичних наук, доцент
Західноукраїнський національний університет
Україна, Тернопіль
ORCID 0000-0001-6490-9633

Анотація. *Розроблено комплексний інструментарій для автоматизованої перевірки безпеки смартконтрактів у мережі Ethereum, що поєднує методи статичного аналізу коду та верифікації через API Etherscan. Результати тестування підтвердили ефективність запропонованого алгоритму для масштабованого моніторингу безпеки децентралізованих фінансових платформ і раннього виявлення вразливостей.*
Ключові слова: *смартконтракт; блокчейн; Ethereum; кібербезпека; DeFi; автоматизована верифікація; код.*

Вступ. Стрімкий розвиток електронної комерції, цифрових платформ та віртуальних ринків створює нову реальність, де традиційні механізми взаємодії поступаються місцем інноваційним цифровим рішенням. Проте революція у сфері фінтеху, попри свою інноваційність, створила нові вразливості для кібербезпеки фінансової екосистеми, зокрема криптовалютного сектора [1]. Сучасні дослідження свідчать про те, що традиційні методи забезпечення безпеки все частіше виявляються недостатньо ефективними проти новітніх видів кіберзагроз [2]. Технологія блокчейн та смартконтракти пропонують революційний підхід до захисту даних, заснований на розподіленому реєстрі та криптографічних алгоритмах [3], однак самі можуть ставати об'єктами кібератак [4]. Інтеграція блокчейн-технологій у фінансову систему створила парадоксальну ситуацію: інструменти, розроблені для підвищення безпеки та прозорості операцій, самі потребують комплексного захисту від кіберзагроз.

Ethereum став першою платформою, що запропонувала концепцію «програмованого блокчейну». Він працює як глобальний, децентралізований комп'ютер. Кожна операція в мережі потребує «газу» – внутрішньої валюти для оплати обчислювальних ресурсів. Існуючі підходи до виявлення фішингу в Ethereum зазвичай використовують машинне навчання або методи вбудовування графів транзакцій для ізольованої ідентифікації шахраїв. Проте такі заходи не забезпечують можливість ефективно виявляти низку пов'язаних із шахрайством транзакційних рахунків (так звані «банди»). Оскільки рахунки в Ethereum є псевдонімними, ці невиявлені акаунти змовників становлять потенційні ризики для системи [5]. Така ситуація вимагає розробки нових підходів до забезпечення безпеки в екосистемі Ethereum. Особливо критичним є створення механізмів автоматичної верифікації смартконтрактів, які б дозволяли виявляти потенційні вразливості ще на етапі розробки.

Метою роботи є розробити комплексний інструментарій для автоматизованої перевірки смартконтрактів, що включає: (1) ідентифікацію адрес (контракти або зовнішні облікові записи); (2) перевірку їхньої верифікації на Etherscan; (3) класифікацію контрактів за архітектурою (незмінний/оновлюваний); (4) обробку помилок і масштабовану підтримку великої кількості адрес. Такий може забезпечити оцінювання поточного стану прозорості та безпеки екосистеми децентралізованих фінансів (DeFi).

Основна частина. Запропонований системний підхід передбачає аналіз та перевірку смартконтрактів в мережі Ethereum через API Etherscan. Він забезпечує отримання інформації

про смартконтракти з мережі Ethereum, аналіз коду смартконтрактів, перевірку їхньої цілісності, відповідності стандартам та пошук потенційних вразливостей.

1. Імпорт та налаштування

На першому етапі імпортуються дві бібліотеки:

– requests: застосовується для виконання HTTP-запитів до API Etherscan;

– random: може бути використана для генерації випадкових значень, наприклад, для тестових сценаріїв чи випадкового вибору параметрів

та визначається константа, яка зберігає ключ доступу до API Etherscan. Цей ключ потрібен для автентифікації та забезпечення доступу до сервісів API.

API Etherscan надає доступ до наступних функціональних можливостей:

– отримання коду смартконтракту: можна витягувати вихідний код контрактів, які верифіковані в мережі Ethereum;

– аналіз транзакцій: скрипт може отримувати історію транзакцій для адреси контракту;

– перевірка балансу або стану контракту: за допомогою відповідних запитів можна отримувати інформацію про поточний стан контракту.

2. Перевірка, чи є вказана Ethereum-адреса смартконтрактом

Далі визначається, чи є Ethereum-адреса контрактом, чи звичайною зовнішньою адресою (EOA, Externally Owned Account). Ця процедура виконується шляхом перевірки наявності байт-коду за вказаною адресою.

3. Перевірка, чи верифікований контракт на Etherscan

Перевіряється, чи пройшов смартконтракт процес верифікації на Etherscan. Верифікація означає, що вихідний код контракту було завантажено на платформу Etherscan, і він відповідає збереженому в блокчейні байт-коду. Якщо контракт верифікований, ми отримуємо відповідь, що це так. Якщо ні, отримуємо повідомлення про відсутність верифікації.

4. Перевірка смартконтрактів на незмінність

На наступному етапі ідентифікується тип контракту: є смартконтракт «immutable» (незмінним) або «upgradable» (з можливістю оновлення). Результатом перевірки вихідного коду смартконтракту для визначення його типу є один із наступних варіантів:

– «immutable» (незмінний): код контракту після розгортання не можна змінити;

– «upgradable» (з можливістю оновлення): контракт підтримує оновлення логіки через проксі-патерни або інші механізми.

Запропонований алгоритм використовує бібліотеку requests для надсилання HTTP-запиту до API Etherscan. Відповідь перетворюється у формат JSON для подальшого аналізу. Результатом є текст коду, який був верифікований на Etherscan.

Код перевіряється на наявність ключових слів:

– «upgradeable»: якщо знайдено, контракт визначається як «з можливістю оновлення».

– «immutable»: якщо знайдено, контракт визначається як «незмінний».

Якщо жодне з ключових слів не знайдено, буде видано повідомлення «Contract type is unknown». Якщо запит до API був невдалим або вихідний код недоступний, видається повідомлення «Contract type information not available».

Така процедура є корисною для автоматизованого аналізу смартконтрактів у блокчейні, особливо якщо необхідно швидко визначити їхню модифікованість або стійкість до змін.

5. Перевірка, чи є задана адреса смартконтрактом та чи верифікований він на Etherscan

Розроблений алгоритм аналізує, чи є задана адреса смартконтрактом, перевіряє його верифікацію на Etherscan та визначає його тип. Така процедура означає ідентифікацію факту завантаження вихідного коду контракту на Etherscan. Якщо контракт верифіковано,

виводиться повідомлення про це, інакше – про відсутність верифікації. Якщо адреса не є смартконтрактом, про це буде виведено відповідне повідомлення.

Тестування запропонованого підходу проведено на наборі з 100 випадкових адрес Ethereum [6]. Окремі з них вивились адресами смартконтрактів, а інші – просто гаманцями.

У межах проведеного дослідження було розроблено та протестовано Python-скрипт для аналізу смартконтрактів в мережі Ethereum. Інструмент дозволяє автоматично визначати, чи є задана Ethereum-адреса смартконтрактом, перевіряти верифікацію контракту на платформі Etherscan та аналізувати тип контракту (immutable або upgradable). Тестування виконано на 100 випадкових API-адресах Ethereum. Приклади отриманих результатів представлено на Рисунку 1.



Рисунок. 1. Фрагмент результатів застосування алгоритму перевірки цілісності смартконтрактів Ethereum

Серед проаналізованих адрес 80% виявились смартконтрактами, з яких лише 60% були верифіковані на Etherscan. Це свідчить про значні ризики для безпеки через непрозорість 40% контрактів. 20% контрактів ідентифіковано як незмінні (immutable), 10% – як оновлювані (upgradable). Для 30% смартконтрактів тип не встановлено. Результати підтверджують ефективність автоматизованого підходу для оцінювання безпеки DeFi-платформ і виявлення потенційних вразливостей на ранніх етапах розгортання контрактів.

Висновки. У роботі розроблено та апробовано системний підхід до автоматизованого аналізу безпеки смартконтрактів у мережі Ethereum, що базується на інтеграції API Etherscan та статичного аналізу вихідного коду. Запропонований інструмент забезпечує комплексну перевірку IP адрес на предмет їх ідентифікації як смартконтрактів, верифікації на платформі Etherscan та класифікації за архітектурним типом. Тестування на вибірці зі 100 випадкових Ethereum-адрес продемонструвало наступні результати: 80 адрес (80%) ідентифіковано як смартконтракти, з яких 60 (60% від загальної кількості) пройшли верифікацію на Etherscan. Архітектурний аналіз верифікованих контрактів виявив 20 незмінних (immutable) контрактів (20%), 10 оновлюваних (upgradable) контрактів (10%). Для 30 контрактів (30%) тип залишився невизначеним через відсутність відповідних маркерів у коді. Отримані результати виявили критичну проблему прозорості екосистеми DeFi: 40% контрактів залишаються неверифікованими, що створює суттєві ризики для користувачів через неможливість аудиту їхнього вихідного коду. Розроблений інструмент довів свою ефективність для масштабованого аналізу смартконтрактів і може бути використаний для моніторингу безпеки DeFi-платформ,

раннього виявлення потенційних вразливостей та підвищення рівня довіри до децентралізованих фінансових сервісів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Kovalchuk, O., Shevchuk, R., & Banakh, S. (2024). Cryptocurrency crime risks modeling: Environment, e-commerce, and cybersecurity issue. *IEEE Access*, 12, 50673–50688. doi: 10.1109/ACCESS.2024.3386428.
[2] Singh, A., & Parizi, R. M. (2023). Security vulnerabilities in smart contracts: A comprehensive survey of attacks and defenses. *IEEE Access*, 11, 45298–45315.
[3] Krichen, M. (2023). Strengthening the security of smart contracts through the power of artificial intelligence. *Computers*, 12(5), 107. doi: 10.3390/computers12050107.
[4] Wang, S., Ding, W., Li, J., Yuan, Y., Ouyang, L., & Wang, F.-Y. (2019). Decentralized autonomous organizations: Concept, model, and applications. *IEEE Transactions on Computational Social Systems*, 6(5), 870–878. doi: 10.1109/TCSS.2019.2938190.
[5] Chen, Z., Huang, J., Liu, S., & Long, H. (2024). Multiscale feature fusion and graph convolutional network for detecting Ethereum phishing scams. *Electronics*, 13(6), 1012. doi: 10.3390/electronics13061012.
[6] Etherscan. (n.d.). Retrieved from https://etherscan.io.

Kovalchuk O.Ya.

API-based toolkit for Ethereum smart contract integrity verification

Abstract. *A comprehensive toolkit has been developed for automated security verification of smart contracts on the Ethereum network, combining static code analysis methods and verification through the Etherscan API. Testing results confirmed the effectiveness of the proposed algorithm for scalable security monitoring of decentralized financial platforms and early vulnerability detection*.

Keywords: *smart contract; blockchain; Ethereum; cybersecurity; DeFi; automated verification; code.*