УДК 004.4'24

ОГЛЯД НАЯВНИХ ІНСТРУМЕНТІВ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ СМАРТ-КОНТРАКТІВ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Терещенко Олександр Ігорович
к.т.н., завідувач кафедри ІПЗ Комлева Наталія Олегівна
Національний університет «Одеська політехніка», УКРАЇНА

АНОТАЦІЯ. В роботі проаналізовані підходи до виявлення вразливостей смарт-контрактів на основі машинного навчання. Проведено порівняльний аналіз наявних інструментів виявлення вразливостей на основі машинного навчання та виявлено їх недоліки.

Вступ. Втрати, спричинені проблемами з безпекою смарт-контрактів, постійно зростають та сприяють поширенню недовіри до технології блокчейну. Зокрема, зловмисникам вдалося вкрасти більше 4 мільярдів доларів шляхом експлуатації вразливостей в смарт-контрактах в 2022 році. Таким чином, створення засобів для ефективного виявлення вразливостей смарт-контрактів є актуальною задачею. Класичними методами виявлення вразливостей є символічне виконання, фаззинг та формальна перевірка. Перелічені методи демонструють недостатню точність, повільність та слабку адаптивність до нових вразливостей. Останнім часом дослідники почали активно застосовувати машинне навчання для вирішення задачі виявлення вразливостей в коді програм.

Мета роботи. Метою роботи є порівняльний аналіз підходів до виявлення вразливостей на основі машинного навчання та визначення основних недоліків поточних інструментів.

Основна частина роботи. Класичні методи виявлення вразливостей в смарт-контрактах, такі як формальна верифікація, символьне виконання, фаззинг-тестування і проміжне представлення, вимагають значних обчислювальних витрат, що негативно впливає на тривалість аналізу контрактів. Наприклад, метод символьного виконання полягає в дослідженні всіх існуючих шляхів в програмі, що зазвичай є дуже трудомісткою та неефективною задачею. В свою чергу, підхід до виявлення вразливостей на основі машинного навчання значно підвищує ефективність виявлення за рахунок автоматизації вилучення ознак. Крім цього, можна досягти вищого рівня узагальненості, розширюваності, адаптивності та точності в порівнянні з засобами на основі класичних методів [1].

Існуючі підходи виявлення вразливостей смарт-контрактів на основі машинного навчання можна класифікувати за трьома категоріями в залежності від типу трансформованої задачі. В першому підході, скомпільований опкод або байткод контракту розглядається як фрагмент тексту, таким чином задача виявлення вразливостей трансформується в задачу класифікації тексту. Прикладами інструментів та моделей, що використовують цей підхід, є *RecChecker*, *SaferSC* та *ESCORT*. В другому підході код смарт-контракту перетворюють в неевклідів граф, в якому містяться залежності потоку даних і потоку керування смарт-контракту, завдяки чому він може добре характеризувати семантику смарт-контракту. При цьому, задача виявлення вразливостей трансформується в задачу виявлення топології графа. Прикладами інструментів та моделей, що використовують цей підхід, є *DR-GCN, TMP* та *ContractWard*. Третій підхід полягає в перетворенні байт-коду смарт-контракту в закодоване зображення. В результаті, задача виявлення вразливостей трансформується в задачу класифікації та розпізнавання зображень. Цей підхід не отримав широкого розповсюдження через те, що операції згортки і об'єднання можуть призвести до того, що будуть порушені контекстні зв’язки в коді [2].

Таким чином було визначено 6 інструментів, що базуються на використанні машинного навчання, а саме *RecChecker, SaferSC, DR-GCN, TMP, ContractWard* та *ESCORT*.

RecChecker – це інструмент, заснований на глибокому навчанні та використовує покращену версію методу виявлення вразливостей коду *VulDeePecker*. *RecChecker* фіксує основну семантичну інформацію та інформацію про залежності потоку керування в смарт-контракті шляхом перетворення вихідного коду контракту у форму фрагментів коду контракту. Цей

інструмент використовує двонаправлену довгу короткочасну пам'ять (*BLSTM*) із механізмом уваги для досягнення автоматичного виявлення вразливостей.

Модель *SaferSC* є однією з перших моделей виявлення вразливостей на основі глибокого навчання для смарт-контрактів. SaferSC аналізує опкоди смарт-контракту та використовує мережу *LSTM* для побудови моделі на рівні коду операції для досягнення точного виявлення вразливостей.

Модель *DR-GCN* досліджує граф з високим ступенем семантичної представленості, який був отриманий шляхом перетворення коду смарт-контракту. Для побудови моделі виявлення вразливостей було вперше використано графову згорткову нейронну мережу (*GCN*). *DR-GCN* аналізує смарт-контракти на двох платформах, а саме *Ethereum і VNT Chain* [3].

TMP (*Temporal message propagation*) — це модель машинного навчання, що використовує граф контракту для виявлення вразливостей смарт-контрактів. При побудові графу смарт-контракту, ключові функції та змінні стають вузлами, які наділені важливою семантичною інформацією. Спрямовані ребра зображують етапи виконання програми, а саме залежності потоку керування та даних.

Інструмент *ContractWard* використовує методи машинного навчання для виявлення вразливостей у смарт-контрактах шляхом вивчення шаблонів вразливих контрактів у навчальних зразках. *ContractWard* витягує біграми з кодів операцій смарт-контрактів і використовує різноманітні алгоритми машинного навчання, які можуть ефективно та результативно виявляти вразливості на основі статичних характеристик.

Фреймворк *ESCORT* заснований на глибоких нейронних мережах (*DNN*) та підтримує легке трансферне навчання для раніше невідомих вразливостей, тому є розширюваним та узагальненим. *ESCORT* використовує багатовихідну нейронну мережеву архітектуру, яка складається з двох частин, а саме загального екстрактору ознак, який вивчає семантику вхідного смарт-контракту, та багатогілочної структури, де кожна гілка вивчає певний тип вразливості на основі ознак, отриманих від екстрактору ознак [3].

В таблиці 1 наведений перелік вразливостей смарт-контрактів на рівні *Solidity-коду, EVM* та залежності блоків.

Таблиця 1 – Перелік вразливостей

№	Рівень	Вразливість
1	Solidity-код	Повторний вхід
2		Цілочисельне переповнення/недоповнення
3		Контроль доступу
4		Неправильна обробка винятку
5		Відмова в обслуговуванні
6		Невідповідність типу
7		Виклик невідомої функції
8		Замороження ефіру
9	EVM	Використання неіснуючої адреси
10		Переповнення стеку викликів
11		Використання джерела транзакції
12	Залежність блоків	Залежність від мітки часу
13		Залежність від порядку транзакції

Основні дані про розглянуті інструменти, а саме рівень робити та перелік вразливостей, що виявляються цими інструментами, наведено в таблиці 2.

Наведені інструменти виявлення вразливостей смарт-контрактів на основі глибоких нейронних мереж мають як мінімум один з наступних недоліків:

1. Вони за своєю суттю є нерозширюваними та негнучкими, оскільки включення будь-яких нових типів вразливостей вимагатиме навчання нових моделей;

2. Вони розрізняють лише вразливі та невразливі смарт-контракти (бінарна класифікація) без можливості визначення типів вразливостей;

3. Інструменти вимагають вихідного коду смарт-контракту, що обмежує сферу їх застосування;

4. Вони підтримують виключно статичний аналіз. Однак статичний аналіз має тенденцію пропускати можливі шляхи виконання програми. Через відсутність динамічної взаємодії із зовнішніми контрактами, це зазвичай призводить до високого рівня хибно позитивних або хибно негативних результатів;

5. Вони мають низьку точність через те, що не існує єдиного стандартизованого набору смарт-контрактів з вразливостями для навчання моделей, який можна використовувати як еталон.

Таблиця 2 – Основні дані про інструменти

Інструмент	Рівень роботи	Вразливості
SaferSC	байткод, Solidity-код	1, 8, 9
RecChecker	Solidity-код	1
DR-GCN	Solidity-код	1, 5, 12
TMP	Solidity-код	1, 5, 12
ContractWard	Solidity-код	1, 2, 10, 12, 13
ESCORT	байткод	1, 5, 8, 10, 13

Висновки. В представленій роботі було розглянуто наявні підходи до виявлення вразливостей смарт-контрактів на основі машинного навчання, зокрема підходи, що трансформують задачу виявлення вразливостей в задачу класифікації тексту, в задачу виявлення топології графа та в задачу класифікації і розпізнавання зображень. Для кожного з розглянутих підходів було наведено приклади наявних інструментів, а саме RecChecker, SaferSC, DR-GCN, TMP, ContractWard та ESCORT. Перелічені інструменти було проаналізовано з точки зору переліку вразливостей, які вони можуть виявляти. Кожен з розглянутих інструментів дозволяє виявляти лише невеликий перелік вразливостей, через що користувачам доводиться використовувати декілька інструментів одночасно, щоб підвищити ймовірність того, що смарт-контракт не має вразливостей. В результаті аналізу наявних інструментів було виявлено, що кожен з них має як мінімум один з наступних недоліків: нерозширюваність та негнучкість, бінарність класифікації, орієнтованість на вихідний код смарт-контрактів, статичність аналізу, низька точність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

4. Komleva N. O., Tereshchenko O. I. Requirements for the development of smart contracts and an overview of smart contract vulnerabilities at the Solidity code level on the Ethereum platform. *Herald of Advanced Information Technology*. 2023. Vol. 6, No 1. P. 54–68.
5. Hajdu A., Jovanovic D. Solc-verify: a modular verifier for solidity smart contracts. *Proceedings of the Working Conference on Verified Software: Theories, Tools, and Experiments*. 2019. P. 161–179.
6. Mazurok I. E., Leonchyk Y. Y., Antonenko O. S., Volkov K. S. Smart contract sharding with proof of execution. *Applied Aspects of Information Technology*. 2021. Vol. 4, No 3. P. 271–281.

OVERVIEW OF AVAILABLE TOOLS FOR DETECTING SMART CONTRACT VULNERABILITIES BASED ON MACHINE LEARNING

Tereshchenko Oleksandr

PhD (Eng), Associate Professor, Head of Software Engineering Department Nataliia Komleva

Odessa National Polytechnic University, UKRAINE

ANNOTATION. The paper analyzes the approaches of detecting vulnerabilities of smart contracts based on machine learning. A comparative analysis of available vulnerability detection tools based on machine learning was conducted and their shortcomings were identified.