Разработка инструмента для исследования множества контрактов сети Ethereum

Прошутинский А. В., 4 курс, кафедра системного программирования СПбГУ, st046460@student.spbu.ru

Ханов А. Р., ст. преп. кафедры системного программирования СПбГУ, awengar@gmail.com

Аннотация

Важной частью сети Ethereum является возможность использования контрактов, программ, описывающих заложенную бизнес-логику. Ошибки при их написании могут привести к серьезным экономическим потерям. В данной работе рассматривается инструмент для поиска уязвимых контрактов сети Ethereum.

Введение

Ethereum является платформой для создания децентрализованных приложений на базе блокчейна, работающих с помощью смарт-контракт-ов (или умных контрактов). Блокчейн – это непрерывная цепочка записей, называемых блоками, которые связаны с помощью криптографии. Каждый блок содержит хэш предыдущего, временную метку и данные транзакций. Блокчейн по сути является устойчивой к изменению данных децентрализованной базой данных [1]. Смарт-контракт – это компьютерная программа, в которой заложены соглашения между сторонами. Использование контрактов выгодны в ситуациях, когда классические средства обеспечения выполнения контракта слишком дороги или стороны не имеют доступа к общему арбитру или юридической системе. Основной принцип умного контракта состоит в полной автоматизации и достоверности исполнения договорных отношений. Умные контракты в Ethereum представлены в виде классов, которые могут быть реализованы на различных языках, включая визуальное программирование и компилируются в байт-код Ethereum Virtual Machine (EVM) перед отправкой в блокчейн. Изменение состояния виртуальной машины может быть записано на полном по Тьюрингу языке сценариев.

Помимо множества возможностей для построения приложений смарт-контракты привнесли множество опасностей, связанных с уязвимостями компиляторов и ошибками программистов. Существует множество инструментов для обнаружения уязвимостей смарт-контрактов[3][4][6][7][2], но не

существует актуальной базы данных со всеми контрактами, их уязвимостями и подробной информацией по ним, а по данным etherscan.io [12] количество верифицированных контрактов очень мало: лишь 53915 [8] из более чем 14 миллионов [10]. Для таких контрактов доступна лишь информация по уязвимостям различных версий компилятора и исходный код. Проверка всех контрактов анализаторами не представляется возможной без должной подготовки исходных данных, так как нужно фильтровать ложные срабатывания, анализ может занимать до нескольких минут, а новые контракты появляются в блокчейне каждые несколько десятков секунд.

Исследователям безопасности и разработчикам верификаторов нужен инструмент для исследования рабочей сети Ethereum и построение базы данных с хранением информации и статистики по контрактам может решить эту проблему. Такая база будет хранить сигнатуры функций, код, результаты проверки наиболее популярными верификаторами, похожие контракты. Все вызовы всех контрактов сохранены в блокчейне, поэтому эта информация доступна всем, но нигде не описана.

Благодаря созданию приложения по сбору статистики по контрактам и своевременной передаче информации об уязвимостях через Ethereum Bounty Program [9] можно добиться минимальных потерь как со стороны пользователей, так и со стороны владельцев распределенных приложений. Вознаграждение за уязвимости контрактов в рамках этой программы не назначается, но будет оказана помощь в доведении информации до разработчиков.

Разработанное решение

За основу был взят Parity-Ethereum[11], предоставляющий доступ к нужным данным блокчейна, благодаря наличию полной базы данных со всеми вызовами, в отличие от официальных узлов. В качестве базы данных - PostgreSQL.

В базе данных хранятся:

- Адрес контракта;
- Исходный код;
- Блок создания;
- Сигнатуры функций;
- Является ли контракт дубликатом и адрес оригинального.

Все данные заполняются с приходом нового блока в реальном времени, поиск дубликатов реализован с помощью индекса по хеш-суммам от исходного кода. После проверки на уникальность исходный код контракта дизассемблируется и из него выделяются сигнатуры функций, по которым строится граф зависимостей контрактов, использующих одинаковые функции.

Так взяв уже известные уязвимые контракты, с использованием данного графа и индекса, мы можем найти точные копии и список ”подозрительных” контрактов.

Заключение

В процессе анализа множества контрактов сети Ethereum было установлено, что из 14 миллионов лишь 200 тысяч являются уникальными, различных сигнатур функций - тоже около 200 тысяч.

Для проверки работы решения был взят адрес одного из самых известных уязвимых контрактов – The DAO[13]. Как оказалось, в блокчейне Ethereum существует 62 точные копии данного контракта, а похожими по набору функций являются ещё около 48 тысяч контрактов. При анализе Parity Multi-Sig Wallet[14] было найдены 652 точных копии и 160 похожих уникальных контрактов.

Благодаря этому инструменту исследователи безопасности при анализе одного смарт-контракта смогут выявить все его копии и связанные с ним, чтобы заранее уведомить их владельцев и сократить время анализа.

Литература

[1] Ethereum White-Paper. https://github.com/ethereum/wiki/wiki/White-Paper

[2] Oyente. Making Smart Contracts Smarter. https://www.comp.nus.edu.sg/ loiluu/papers/oyente.pdf

[3] Mythril Classic: Security analysis tool for Ethereum smart contracts. https://github.com/ConsenSys/mythril-classic

[4] MAIAN: automatic tool for finding trace vulnerabilities in Ethereum smart contracts. https://github.com/MAIAN-tool/MAIAN

[5] Decompiler and Security Analysis tool for Blockchain-based Ethereum Smart-Contracts. https://github.com/comaeio/porosity

[6] Manticore. Symbolic execution tool. https://github.com/trailofbits/manticore

[7] Securify. Security Scanner for Ethereum Smart Contracts. https://securify.chainsecurity.com/

[8] Verified Contracts listing. https://etherscan.io/contractsVerified

[9] Ethereum Bounty Program. https://bounty.ethereum.org/

[10] Ethereum contracts creation transactions. https://blockchair.com/ethereum/calls?q=type(create)#

[11] Parity-Ethereum. The fast, light, and robust EVM and WASM client. https://github.com/paritytech/parity-ethereum

[12] Etherscan. Block Explorer and Analytics. https://etherscan.io

[13] The DAO contract. https://etherscan.io/address/0xbb9bc244d798123fde783fcc1c72d3bb8c189413#code

[14] Parity Multi-Sig Wallet. https://www.parity.io/a-postmortem-on-the-parity-multi-sig-library-self-destruct/