Обеспечение информационной безопасности смарт-контрактов в сети Ethereum

Ensuring information security of smart-contracts in the network Ethereum

Елецкий Егор Николаевич
студент,
Российский государственный университет нефти и газа
(национальный исследовательский университет) имени И.М. Губкина,
РФ, г. Москва
e-mail: eletskiy.egor@mail.ru

Eletskiy Egor Nikolaevich
student,
National University of Oil and Gas «Gubkin University»,
RF, Moscow
e-mail: eletskiy.egor@mail.ru

Аннотация.
В данной статье в центре внимания оказался вопрос об обеспечении информационной безопасности смарт-контрактов в сети Ethereum. Рассматриваются основные направления уязвимостей, характерные для смарт-контрактов. Предложена классификация этих уязвимостей и способы их решений. Помимо этого, рассматривается такой вопрос как тенденции развития в криптографии и его последствия. В итоге делается вывод, что смарт-контракт в сети Ethereum остается еще полноценно неизученным вопросом.

Annotation.
This article focuses on the issue of ensuring the information security of smart contracts in the Ethereum network. The main directions of vulnerabilities typical for smart contracts are considered. A classification of these vulnerabilities and ways of solving them are proposed. In addition, such an issue as development trends in cryptography and its consequences is considered. As a result, it is concluded that the smart contract in the Ethereum network is still a fully unexplored issue.

Ключевые слова: информационная безопасность, смарт-контракт, уязвимость.

Key words: Information Security, smart contract, vulnerability.

Информационная безопасность (ИБ) – это состояние рассматриваемого комплекса, при котором оно наименее восприимчиво к вмешательству и нанесению ущерба злоумышленником.

Смарт-контракт – это программный алгоритм, который контролирует совершение сделки.

Уязвимость - Недостаток (слабость) программного (программно-технического) средства или информационной системы в целом, который(ая) может быть использован(а) для реализации угроз безопасности информации [2].

Я уверен, что многие уже знакомы с таким понятием как блокчейн Ethereum. Однако с другой стороны, может кто-то и вовсе не слышал об этом. Хорошо это или плохо – сложно сказать. Но все же, если использовать смарт-контракты в сети Ethereum, необходимо осознавать, с чем люди связываются и каковы могут быть последствия. В Интернете, безусловно, уже было много сказано о платформе Ethereum. Я же хочу разобрать смарт-контракт в сети Ethereum с точки зрения информационной безопасности. Так как смарт-контракт в основном связан с финансами, то тему безопасности необходимо…

Собственно с чего же начать? Я считаю, что для обеспечения информационной безопасности смарт-контракта в сети Ethereum, необходимо иметь представление об его создании и конечно платформе Ethereum. Разобрав это, можно выделить основные уязвимости, разобрать их. Зная основные направления уязвимостей смарт-контрактов, можно принять меры по нейтрализации их.

Я думаю, что каждый человек в какой-то степени да слышал про Bitcoin. Пару слов скажу о нем. В 2008 году от имени Сатоши Накамото был опубликован документ «Bitcoin: A Peer-to-Peer Electronic Cash System». В

нем описана концепция реализации распределенного реестра в виде цепочки блоков доверие, которая обеспечивается не за счёт третей доверенной стороны, а криптографическими методами. Документ утверждал создание системы расчетов с электронным аналогом наличных денег. Главное условие в том, чтобы система полностью была доступна каждому участнику и не требовала выделенного финансового института для функционирования. Реализация распределенного реестра, которую предложил Сатоши, получило название блокчейн. Но блокчейн - это не единственная возможная реализация концепции распределенного реестра, всего лишь одна из реализации механизма хранения данных в реестре. Следует разделять термины блокчейн и распределенный реестр.

Но жизнь не стоит на месте и с развитием технологии построения распределенного реестра на базе блокчейна стал Ethereum. В 2013 году Виталик Бутерин предложил свое видение развития сети Bitcoin. Он запустил новую сеть в 2015 году. Основная идея, положенная в основу запуска сети Ethereum было мнение Бутырина о том, что для написания кода необходимо продвинутый скриптовый язык программирования, который мог бы гарантировать не только достоверность финансовых действий, но и способность выполнять все условия сделки, которые прописываются в коде программы.

Фактически Ethereum получил такую популярность исключительно из-за наличия смарт-контрактов. В сети Ethereum под смарт-контрактом понимается программный алгоритм, который контролирует совершение сделки. Эта программа заносится в сеть Ethereum без возможности совершения каких-либо изменений. Значимый результат действия данного алгоритма - эта модификация распределённого реестра, достоверность которого гарантируется за счет свойств блокчейн.

Ключевой фактор, который привлекает внимание к смарт-контрактам, то, что в отличие от любой обычной программы, корректность работы и неизменность результата работы гарантируется криптографически внутри блокчейна, не привлекая третьей доверенной стороны.

Исходя из сказанного, хочется выделить следующее; блокчейн, программирование, криптография. Именно эти слова являются ключом к выявлению уязвимостей смарт-контракта в сети Ethereum.

Уязвимости смарт-контракта в сети Ethereum.

Безопасность смарт-контрактов можно рассматривать на нескольких уровнях, представленных на рисунке 1.

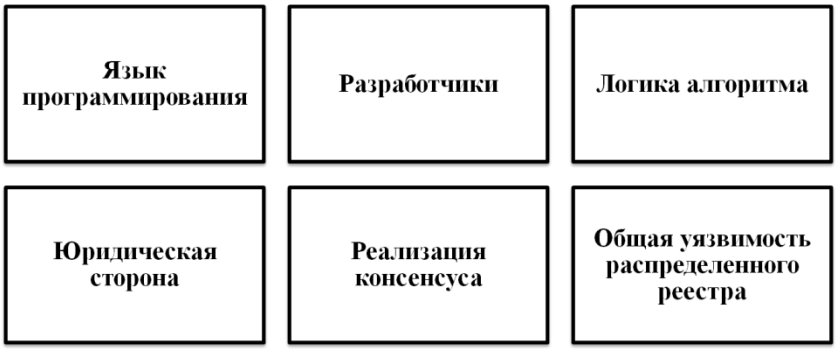

Рисунок 1. Уязвимости смарт-контракта в сети Ethereum

Язык программирования.

Первый уровень проблем связан с самим языком Solidity, на котором пишется код. Этот язык является JavaScript подобным и изначально не был ориентирован на безопасную разработку. Конструкции типа множество наследования, возможности вызова функции из внешних смарт-контрактах с одной стороны повышают гибкость языка, но с другой стороны повышают вероятность ошибок и атак на смарт-контракты.

Разработчики (программисты).

Следующий уровень проблем с безопасность смарт-контрактов кроется в процессе разработки самих программистов. В первую очередь надо сказать, что это отрасль программирование очень молода. Достаточно сложно найти опытных разработчиков имеющих навыки разработки надежных систем в Solidity. Зачастую за разработку берутся люди, недавно изучившие язык. Новички в основном используют шаблоны кода из популярных библиотек. Использование внешних библиотек само по себе может являться источником уязвимостей. Безусловно, стоит отметить, что очень сложно изменить код смарт-контракта после размещения его в сети Ethereum.

Логика алгоритма.

Еще один уровень, на котором возникают уязвимости смарт контрактов - это сам код, а более точно его логика. Этот тип уязвимости тесно связан с юридической стороной. Отмечу то, что из-за специфики языка программирования Solidity, на котором написан алгоритм смарт-контракта, имеет тоже уязвимости (так называемые внутренние уязвимости). По этим причинам существенно измениться алгоритм, нарушится его логика. А ведь код «контролирует» совершение сделки.

Общая уязвимость распределенного реестра.

Следующий уровень проблем безопасности это то, что все взаимодействия между пользователем и блокчейном происходят через мобильное приложение или веб-сайт. Используя уязвимости мобильных приложений, которые были разработаны для работы с блокчейн компонентами, злоумышленники могут отправлять произвольные транзакции, воспользовавшись правами пользователя приложения.

Юридическая сторона смарт-контракта.

С юридической точки зрения смарт-контракт понимается как умный договор – это правовое соглашение между лицами, условия которого заключены в электронном виде. Сегодня нет четкого понимания о том, каким образом технологии распределённого реестра могут регулироваться. В блокчейн Ethereum, с использованием смарт-контракта, любой участник сети может записать любую информацию, в том числе и запрещенную к распространению. Механизмов для блокировки распространения такой информации или удаления её не существует за исключением блокировки полного доступа ко всей сети Ethereum. Поэтому возникает множество вопросов по обеспечению судебной защиты со стороны государства.

Реализация консенсуса.

Консенсус - это механизм достижения договоренности участниками сети о том, какая версия реестра является актуальной и валидный, при условии, что некоторые участники сети пытаются внести в реестр изменения, которые нарушают его целостность.

Рассмотрим уязвимости на примере консенсуса Proof-of-Work.

Если используется данный алгоритм достижения консенсуса, то сеть подвержена так называемой атаки 51 процент. Атака 51 процент становится возможной, когда более 51-го процента вычислительной мощности сети попадает под контроль одного субъекта. В этом случае становится возможным проводить транзакции с двойным расходованием, отменять транзакции, делать копии распределённого реестра.

В качестве примера успешной реализации атаки 51 процент можно назвать серию из трех атак на криптовалюту Verge, в результате которой было похищено несколько миллионов долларов. Атаку на валюту Bitcoin Gold, с помощью которой была совершена двойная трата почти 19 миллионов долларов и атака на валюту ZenCash в результате трех двойных трат была похищена около 700 тысяч долларов.

Еще пример атаки, связанный с возможностью так называемых Long-Range Attack. В рамках этой атаки злоумышленник создает свою альтернативную версию блокчейна, начиная с нулевого блока и потом подменяя

ей актуальную версию. В этой альтернативной версии могут присутствовать как двойные траты, так и измененные транзакции.

Вывод об уязвимостях смарт-контрактов.

Существует разное мнение о процентном соотношений уязвимости смарт-контракта. Я выделяю, что уязвимости смарт-контрактов занимают примерно треть от всех выявленных уязвимостей системы блокчейн.

Анонимность.

Хочу разобрать такой «щекотливый» момент как анонимность. Вообще вопрос анонимности сразу же возникает при знакомстве с блокчейн. Безусловно, в сети нет механизмов идентификации владельцев адресов. Все транзакции, включая адреса отправителей и получателей, суммы в кошельке публично доступны. Безопасность обеспечивается криптографическими методами. Но все же эта сеть функционируют не в вакууме, а в традиционной инфраструктуре, в которой способов отслеживания идентификации источников трафика немало. Установить владельца адреса блокчейна можно по протоколам сетевых сессий с IP адресами, данными сохраняемыми в браузере. Поэтому вопрос об анонимности – это палка о двух концах.

Чего ожидать дальше? Тренды развития и криптоапокалипсис.

Прежде чем задуматься над этим вопросом, отмечу что главная особенность любого распределенного реестра - это криптографическая подпись транзакции. Для подписи клиента необходимо иметь два ключа закрытый и открытый.

Одним из минусов тренда развития - это квантовая криптография. Существует два квантовых алгоритма угрожающих распределенным реестрам, построенные на традиционных принципах. В случае распространения квантовой криптографии:

во-первых - это алгоритм Гровера, который сокращает время подбора исходного значения хэш-функции. Таким образом, злоумышленник, обладающий квантовым компьютером, безболезненно проводит атаки двойной траты на распределённый реестр и вообще может изменить значение распределённого реестра после достижения консенсуса.

во-вторых - это так называемый алгоритм Шора, который состоит в сведении сложно-вычислительных на классическом компьютере задач к вычислению порядка некой функции. Говоря простыми словами, благодаря использованию данного квантового алгоритма становится возможным вычисление закрытого ключа на основании открытого ключа.

В итоге, это приведет к обвалу всю используемую современную криптографию. Полностью компрометирует цифровую подпись. Это настолько серьезная проблема, что в 2017 году конгресс США дал указ, получающий NIST (National Institute of Standards and Technology) начать разработку стандартов для постквантовой криптографии.

Интересные тренды наблюдается в области обеспечения анонимности транзакций. В частности рассматривается вопрос о применении в рамках сетей Ethereum протокол zk – STARK, который позволяет полностью маскировать отправителя и получателя транзакции.

Способы нейтрализаций уязвимостей.

Исходя из рассмотренных уязвимостей, еще раз подчеркнем, что смарт-контракт - это программа, реализованная на языке программирования Solidity. Это полноценный алгоритм, который имеет и способен организовывать сложные ветвления, циклы и множество других элементов программирования

Исходя из перечисленных уязвимостей, для поддерживания безопасности, немало важно проводить все возможные тестирования кода – например, пентесты и аудит смарт-кантрактов, с целью выявить уязвимости

алгоритма. Но стоит помнить, что аудит в этой область не является юридическим, который будет гарантировать 100 % безопасность кода.

Следующий вопрос, на который следует ответить, кто будет реализовывать код? Необходимо сформировать определенные требования к опыту и квалификации программиста, разработчика; подготавливать кадры специальной направленности.

Для нейтрализации угроз, которые исходят от уязвимостей, связанных с недостаточно проработанной правовой стороны смарт-контракта, необходимо юридически закрепить все функции, которые выполняет смарт-контракт. Сделать это можно с помощью подписания соглашения между сторонами об использовании смарт-контракта, в котором излагаются все условия, реализуемые смарт-контрактом, и последствия выполнения этих условий.

Заключение

Во-первых, что хочется отметить – это то, что смарт-контракт в сети Ethereum взаимосвязаны с деньгами, с финансами. А значит и высоки риски ИБ. Поэтому, для уменьшения вероятности реализации нехорошего алгоритма злоумышленником, необходимо уделять особое внимание к безопасности в области смарт-контрактов.

Во-вторых, для реализации информационной безопасности смарт-контрактов необходимо владеть основными знаниями в этой области. В статье были разобраны основные уязвимости как для смарт-контракта, так и общие характерные для систем на основе распределенных реестров.

В-третьих, блокчейн, благодаря криптографии, отлично подходит для обеспечения целостности информации, но не дает особых преимуществ с точки зрения анонимности, если сравнивать с иными технологиями. Отсюда следует то, что система блокчейн не является решением для кибератак, поэтому традиционные функции безопасности по-прежнему являются важной частью инфраструктуры информационной безопасности.

В итоге, я пришел к выводу, что смарт- контракт в сети Ethereum остается еще полноценно неизученным вопросом, пока этой технологии не хватает зрелости, чтобы выйти из категории инноваций в мейнстрим.

Список используемой литературы:

1. https://consensys.github.io/smart-contract-best-practices/
2. ГОСТ Р 56546-2015 Защита информации. Уязвимости информационных систем. Классификация уязвимостей информационных систем.
3. Колесников П., Бекетнова Ю., Крылов Г. Технология блокчейн. Анализ атак, стратегии защиты. Изд-во LAP LAMBERT Academic Publishing RU, 2017. 77 с