УДК 004.056

Н. Карапетьянц[1]*

Виртуальный испытательный стенд для проведения экспериментов в области анализа транзакций сети Bitcoin

[1] Национальный исследовательский ядерный университет «МИФИ», г. Москва, Российская Федерация
* e-mail: nkarapetyants@mephi.ru

Аннотация. В данной статье представлена реализация виртуального испытательного стенда для проведения экспериментов в области анализа транзакций в сети Bitcoin. Цель данного исследования – разработка виртуального испытательного стенда для обеспечения воспроизводимости экспериментов в рамках исследований транзакций сети Bitcoin. Результаты исследований показали, что для обеспечения анализа транзакций в сети Bitcoin требуется большие вычислительные ресурсы по причине большого объема данных транзакций, генерируемых участниками сети Bitcoin. Рассматривается реализация виртуального испытательного стенда для проведения экспериментов в области анализа транзакций с использованием инструментов автоматизации. Результаты настоящей работы предоставляют возможность осуществить автоматизированное развертывание виртуальной инфраструктуры, состоящей из набора инструментов для проведения репрезентативных экспериментов оценки существующих или разработки новых методов идентификации средств, используемых в незаконной деятельности, и их источников.

Ключевые слова: Bitcoin, Blockchain, KYT, ETL, автоматизация

N. Karapetyants[1]*

Virtual Test Bench for Conducting Experiments in the Field of Transaction Analysis of the Bitcoin Network

[1] National Research Nuclear University MEPhI, Moscow, Russian Federation
* e-mail: nkarapetyants@mephi.ru

Abstract. This article presents the implementation of a virtual test bench for conducting experiments in the field of transaction analysis on the Bitcoin network. The purpose of this study is to develop a virtual test bench to ensure reproducibility of experiments within the framework of Bitcoin network transaction research. The research results have shown that large computing resources are required to provide transaction analysis on the Bitcoin network due to the large volume of transaction data generated by Bitcoin network participants. The implementation of a virtual test bench for conducting experiments in the field of transaction analysis using automation tools is being considered. The results of this work provide an opportunity to implement automated deployment of a virtual infrastructure consisting of a set of tools for conducting representative experiments to evaluate existing or develop new methods for identifying funds used in illegal activities and their sources.

Keywords: Bitcoin, Blockchain, KYT, ETL, automatization

Введение

Платежная сеть Bitcoin, функционирующая на основе распределенной технологии блокчейн, обеспечивает как децентрализованный доступ, так и прозрачность проводимых транзакций сети. Архитектурные особенности протокола

Bitcoin исключают возможности идентификации пользователей сети, осуществляющих платежные операции, как и возможность анонимно проводить платежи. Таким образом идентифицировать участников сети Bitcoin возможно, но при условии использования специализированных средств и методов «Знай свою транзакцию» (англ. KYT, Know Your Transaction).

Основными клиентами инструментов анализа транзакций сети Bitcoin являются биржи, регуляторы и правоохранительные органы. С точки зрения правоохранительных органов данные инструменты используются для расследования преступлений, связанных с финансированием терроризма, отмыванием денег, мошенничеством, кражей криптовалюты, киберпреступлений и других видов незаконной деятельности, в которых задействованы криптовалюты. Регулятор, например, центральный банк, осуществляет надзор за действиями в сети Bitcoin для обеспечения экономической безопасности кредитно-финансовой системы. Биржам также необходимы данные инструменты в соответствии с требованиями государственных и международных регуляторов для осуществления деятельности, направленной на борьбу с отмыванием денег.

В данной статье приводится реализация виртуального испытательного стенда для проведения экспериментов в области анализа транзакций сети Bitcoin с целью обеспечения возможности повторного воспроизведения проведенных научных исследований.

В данной области проводят исследования следующие ученые: Стрельцов А. С., Французова Г. А., Басыня Е. А. Албычев А. С., Ильин Д. Ю., Лю Ф., Шноринг Х., Тюркауф Д., Жако В., Влахавас Г., Сян Й. [1–8].

Методы и материалы

Виртуальный испытательный стенд представляет собой виртуальную сетевую инфраструктуру, функционирующую на базе технологий виртуализации и контейнеризации. Данные технологии позволяют создавать сложную и воспроизводимую информационную инфраструктуру для обеспечения работоспособности сервисов и приложений как общего, так и специализированного предназначения.

Программное решение, реализующее технологию виртуализации, именуется гипервизором. Наиболее распространёнными на сегодняшний день гипервизорами являются KVM, Hyper-V, VMware ESXi и VirtualBox. В данном исследовании в качестве гипервизора будет использоваться решение Proxmox Virtual Environment [9], функционирующее на базе KVM. Выбор данного программного решения обусловлен следующими причинами: оно распространяется с открытым исходным кодом, имеет возможность интеграции со средствами автоматизации Ansible и Terraform, имеет графический интерфейс, а также поддерживает файловую систему ZFS.

К средствам, обеспечивающим функционирование технологии контейнеризации, относятся Kubernetes, Docker и LXC [10]. Kubernetes предлагает высокий уровень автоматизации, управления контейнерами и масштабируемости, но требует значительных затрат на развертывание и настройку. LXC позволяет развер-

тывать системные контейнеры, в которых функционирует полноценная операционная система. К недостаткам LXC можно отнести высокое потребление ресурсов по сравнению с контейнерами приложений, где функционирует ограниченное количество процессов. Это решение можно использовать как альтернативу виртуальным машинам, в которых функционирует операционная система на базе ядра Linux. Для развертывания контейнеров приложений используется Docker. Он позволяет создавать и управлять множеством контейнеров, в каждом из которых функционирует лишь один сервис. Также Docker включает в себя режим Swarm, который представляет собой систему оркестрации контейнеров для поддержания высокой доступности сервисов, работающих внутри контейнеров. Таким образом, использование Docker Swarm в рамках реализации виртуального испытательного стенда позволяет создать воспроизводимую и масштабируемую виртуальную инфраструктуру.

Далее необходимо обеспечить получение актуальной информации о транзакциях сети Bitcoin. Сделать это можно с помощью клиент-серверного приложения Bitcoin Core, который хранит все данные об операциях сети Bitcoin [11]. Автоматизация процесса сборки Bitcoin Core из исходного кода и его настройки достигается путем контейнеризации. Впоследствии полученный контейнер позволит минимизировать время развертывания виртуального стенда. И только процесс получения информации занимает продолжительное время, поскольку зависит от производительности центрального процессора для осуществления проверки цепочки блоков и пропускной способности канала связи с общедоступной сетью Интернет. На данный момент объем сети Bitcoin достигает более 550 гигабайт и с каждым днем неуклонно растет.

После того, как будет получена вся информация о транзакциях сети Bitcoin, необходимо её извлечь. Данный процесс осуществляется с помощью протокола удаленного вызова процедур JSON RPC, входящего в состав клиент-серверного приложения Bitcoin Core, и языка программирования Python. Получаемая информация представляет собой данные в формате JSON, которые впоследствии можно загрузить в базу данных или любую другую систему хранения данных для последующего анализа.

Поскольку получение данных напрямую через Bitcoin Core имеет низкую скорость и производительность, возникает необходимость использования системы управления базами данных (СУБД). Протокол сети Bitcoin постоянно развивается и улучшается, результатом чего является внесение новых полей данных в блок транзакций. С учетом данного фактора и большого объема данных о транзакциях требуется высокопроизводительная не реляционная база данных с возможностью горизонтального масштабирования. Лучше всего подходит система управления базами данных Cassandra [12], которая помимо вышеперечисленных требований имеет возможность интеграции с открытой распределенной вычислительной системой Apache Spark, предназначенной для обработки больших объемов данных [13].

Использование системы Apache Spark в составе виртуального испытательного стенда позволяет осуществлять подробный анализ транзакций с использо-

ванием машинного обучения, производить обработку данных сети Bitcoin в реальном времени, подключать сторонние инструменты, например, графовую базу данных или ELK (Elasticsearch, Logstash, and Kibana) Stack [14].

ELK Stack является технологическим стеком для централизованного сбора, обработки и визуализации данных. Сервис Elasticsearch может быть использован для быстрого и гибкого поиска по транзакциям. Для получения данных о транзакциях из базы данных Cassandra с последующей их отправкой в Elasticsearch используется сервис Logstash. Визуализацию данных обеспечивает сервис Kibana, который позволяет создавать диаграммы и графики из данных полученных в Elasticsearch.

Процесс автоматизированного развертывания обеспечивается за счет использования инструментов Ansible и Terraform [15]. Инструмент Ansible необходим для осуществления автоматической настройки виртуальных сервисов, развертывания кластера Docker Swarm, а также для запуска сервисов Apache Cassandra, ELK Stack и Bitcoin Core. Средство Terraform обеспечивает создание и предварительную настройку виртуальных машин в среде Proxmox.

Автоматизированное развертывание виртуального испытательного стенда состоит из трех этапов:

1) установка и настройка операционной системы Proxmox Virtual Environment;
2) конфигурирование Ansible и Terraform на компьютере, с которого будет производиться управление стендом;
3) запуск подготовленных сценариев развертывания виртуальной сетевой инфраструктуры.

Установка операционной системы Proxmox Virtual Environment осуществляется вручную в соответствии с параметрами, заданными разработчиком по умолчанию. Последующая настройка Proxmox включает в себя выполнение следующих операций:

– создание пользователя с именем «terraform»;
– предоставление прав доступа пользователю «terraform» к корневому каталогу и каталогу хранилища с ролью «Administrator»;
– добавление токена API для пользователя «terraform»;
– создание шаблона виртуальной машины.

С учетом требований разработчиков вышеописанных инструментов в качестве основной операционной системы виртуальных машин будет использоваться Ubuntu 22.04 LTS (Jammy). Подробная инструкция по созданию шаблона содержится в официальной документации Proxmox [16]. Но прежде, чем конвертировать установленную виртуальную машину, необходимо убедиться, что все пакеты операционной системы обновлены, а также установлен пакет гостевых дополнений «qemu-guest-agent».

Конфигурирование инструментов Ansible [17] и Terraform заключается в создании пары ключей SSH и копировании открытого ключа в поле «SSH public key» в разделе «Cloud init» в настройках шаблона виртуальной машины. Также необходимо в поле «User» того же раздела установить имя пользователя, с помощью которого будет осуществляться подключение к виртуальной машине по

протоколу SSH. Также для работы Terraform с Proxmox требуется плагин «terraform-provider-proxmox» [18], который доступен для скачивания с официального сайта разработчика.

В перечень конфигурационных файлов для развертывания виртуальной инфраструктуры стенда входят файл Terraform для создания виртуальных машин, сценарии Ansible для развертывания кластера Docker Swarm, СУБД Cassandra, системы обработки данных Apache Spark, ELK Stack и Bitcoin Core.

Конфигурационный файл Terraform включает в себя следующие директивы: подключение плагина «terraform-provider-proxmox», данные для подключения к Proxmox (имя пользователя, токен и секретный ключ), перечень параметров разворачиваемых виртуальных машин и перечень сценариев Ansible. Общее количество машин, необходимое для развертывания всех сервисов, составляет 5 штук. Одна виртуальная машина необходима для обеспечения функционирования мастер-ноды кластера Docker Swarm, а каждая из остальных виртуальных машин предназначена для функционирования каждого из сервисов.

Результаты и обсуждение

Разработка виртуального испытательного стенда для проведения экспериментов в области анализа транзакций сети Bitcoin производилась с использованием управляющего компьютера с предустановленной операционной системой Manjaro и сервера, характеристики которого указаны в табл. 1.

Для обеспечения качества и проверки функционирования испытательного виртуального стенда было проведено стресс-тестирование. Целью данного тестирования является оценка поведения системы при высоких нагрузках и определение оптимальных параметров виртуальных машин для корректного функционирования сервисов.

При проведении стресс-тестов производилось тестирование каждой виртуальной машины по отдельности с целью выявления минимальных параметров виртуальных машин, при которых функционирует каждый из сервисов. После этого производилось тестирование всего стенда. В результате проведенных тестов были получены параметры виртуальных машин для функционирования стенда, которые отображены в табл. 2.

Таблица 1

Экспериментальная рабочая среда

Параметр	Значение
Процессор	AMD 6800H (8 ядер, 3,2 ГГц)
Оперативная память	64 ГБ
Постоянная память	1920 ГБ (SSD)
Сеть внешняя	1 ГБит/с
Сеть внутренняя	10 Гбит/с
Файловая система	ZFS
Операционная система	Proxmox 8.1.10

Таблица 2

Параметры виртуальных машин стенда

Виртуальная машина	Количество ядер процессора	Объем оперативной памяти	Объем постоянной памяти
Мастер-нода Docker swarm	1 vcpu	2 ГБ	12 ГБ
Bitcoin Core	1 vcpu	8 ГБ	700 ГБ
ELK Stack	2 vcpu	14 ГБ	200 ГБ
Apache Spark	2 vcpu	16 ГБ	300 ГБ
Cassandra	2 vcpu	16 ГБ	400 ГБ

Также тестирование показало, что для анализа транзакций в первую очередь необходим большой размер постоянной памяти. Лучше всего подходят серверные твердотельные накопители, поскольку они имеют высокую производительность и надежность для обеспечения нормальной работы систем хранения и баз данных.

Заключение

В ходе исследования был разработан виртуальный испытательный стенд для проведения экспериментов в области анализа транзакций сети Bitcoin. Практическим результатом работы являются сценарии автоматизированного развертывания инструментов хранения, обработки и анализа для проведения воспроизводимых экспериментов. В ходе работы было проведено стресс-тестирование, по итогам которого были выявлены минимальные системные параметры виртуальных машин для обеспечения корректного функционирования сервисов Bitcoin Core, Apache Cassandra, Apache Spark и ELK Stack в кластере Docker Swarm.

В рамках дальнейших исследований планируется дополнить перечень инструментов виртуального испытательного стенда для проведения репрезентативных экспериментов в области анализа транзакций Bitcoin.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Стрельцов А. С., Французова Г. А., Басыня Е. А. Разработка системы сбора, обработки, анализа, идентификации корреляции событий информационной инфраструктуры предприятия //Системы анализа и обработки данных. – 2023. – №. 1 (89). – С. 101-113.

2. Албычев А. С., Ильин Д. Ю. Выбор стека технологий вычислительной инфраструктуры для экспериментальных исследований цифровых валют //International Journal of Open Information Technologies. – 2023. – Т. 11. – №. 4. – С. 95-101.

3. Liu F. et al. Bitcoin address clustering based on change address improvement //IEEE Transactions on Computational Social Systems. – 2023.

4. Schnoering H., Porthaux P., Vazirgiannis M. Assessing the Efficacy of Heuristic-Based Address Clustering for Bitcoin //arXiv preprint arXiv:2403.00523. – 2024.

5. Thürkauf D. Address Clustering Heuristics for Account-Based Blockchain Networks: An Analysis Based on a Decentraland User Set //Available at SSRN 4589925. – 2023.

6. Jacquot V., Hammad N., Donnet B. Efficient and Reliable Service Detection on Bitcoin //IEEE International Conference on Blockchain and Cryptocurrency (ICBC). – IEEE, 2024.

7. Vlahavas G., Karasavvas K., Vakali A. Unsupervised clustering of bitcoin transactions //Financial Innovation. – 2024. – Т. 10. – №. 1. – С. 25.

8. Xiang Y. et al. Babd: A bitcoin address behavior dataset for pattern analysis //IEEE Transactions on Information Forensics and Security. – 2023.

9. Mitchell J. C. On the Practical Applications of Virtualization Software in Business Operations : дис. – Utica College, 2020.

10. Chen C. C. et al. Docker and Kubernetes //Industry 4.1: Intelligent Manufacturing with Zero Defects. – 2021. – С. 169-213.

11. Tschannen P., Ahmed A. Bitcoin's apis in open-source projects: Security usability evaluation //Electronics. – 2020. – Т. 9. – №. 7. – С. 1077.

12.Shah R. S., Bhatia A. Bitcoin data analytics: Exploring research avenues and implementing a hadoop-based analytics framework //Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 34th International Conference on Advanced Information Networking and Applications (WAINA-2020). – Springer International Publishing, 2020. – С. 178-189.

13. Begum M. S. et al. Predicting The Prices Of Bitcoin Using Data Analytics //Turkish Journal of Computer and Mathematics Education (TURCOMAT). – 2021. – Т. 12. – №. 10. – С. 1487-1501

14. Bambuch V. Platform for Cryptocurrency Address Collection. – 2020.

15. Басыня Е. А., Малышев Е. А. Обеспечение достоверности результатов научно-практических изысканий с применением программной инженерии // ЗАЩИТА ИНФОРМАЦИИ. ИНСАЙД. – 2023. – Т. 112 – №. 4. – С. 14-21.

16. Cloud-Init_Support // Proxmox Wiki. – URL: https://pve.proxmox.com/wiki/Cloud-Init_Support (дата обращения: 24.04.2024).

17. Басыня Е. А., Лукина М. С. Автоматизированная установка и конфигурирование серверных решений //Современные материалы, техника и технологии. – 2016. – №. 2 (5). – С. 21-26.

18. terraform-provider-proxmox // Github. – URL: https://github.com/Telmate/terraform-provider-proxmox (дата обращения: 24.04.2024).