Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/49/20241063

Assessing and neutralizing multi-tiered security threats in blockchain systems

Yize Cai

International School of Information Science and Engineering, Dalian University of Technology, Dalian, 116024, China

7585001@mail.dlut.edu.cn

Abstract. Blockchain technology, the backbone of digital cryptocurrencies, has rapidly ascended as a pivotal tool in modern commerce due to its decentralized, immutable nature. It offers fresh, innovative avenues for overcoming trust issues inherent in traditional trading systems. Yet, the unique traits that make blockchain advantageous also render it vulnerable. Cybercriminals are ceaselessly innovating, devising new tactics to exploit these vulnerabilities and resulting in a surge of security incidents that have led to substantial economic losses. The increasing frequency and sophistication of these attacks jeopardize the integrity and stability of blockchain networks. This paper offers a comprehensive study of blockchain system architecture, the principles underlying various attack methods, and viable defense strategies, all organized within a hierarchical framework. Initially, the paper categorizes blockchain attacks according to the hierarchy of blockchain systems, providing a detailed exploration of the characteristics and principles behind these attacks at each level. Next, the paper summarizes existing countermeasures and proposes effective new strategies for bolstering blockchain security. The paper concludes with a recap of its key findings and outlines the landscape for future research in blockchain security.

Keywords: Blockchain Attacks, Mitigation Strategies, Hierarchical Level.

1. Introduction

Since its inception in 2009, digital cryptocurrencies have garnered widespread attention. At the core of these digital currencies is blockchain technology, which revolutionizes the accounting landscape through intricate coding structures. Blockchain's decentralized approach effectively eliminates the need for intermediaries like governments, thus addressing the longstanding trust issues in traditional transactions. The evolution of blockchain can be segmented into three significant phases. Blockchain 1.0 marks the era of programmable currency, epitomized by Bitcoin. During this phase, the foundational theoretical framework of blockchain was laid out, and its guiding principles were successfully realized [1]. This era solidified the basic technological infrastructure for blockchain's future.

Blockchain 2.0, or the era of programmable applications, is most notably represented by Ethereum. In this phase, innovative technologies like smart contracts and consensus algorithms were incorporated, substantially bolstering blockchain's programmability and versatility. While these advances broadened the technology's applicability, they also introduced more intricate and economically impactful security risks, placing increased responsibilities on developers.

DOI: 10.54254/2755-2721/49/20241063

Lastly, we have Blockchain 3.0, termed the "Programmable Social Era." In this stage, blockchain seamlessly integrates with multiple industries, giving rise to a socially interconnected blockchain ecosystem. Currently, the academic research on blockchain is nearing the end of its 2.0 phase, while practical applications are in a transitional period, moving from the 1.0 to the 2.0 era.

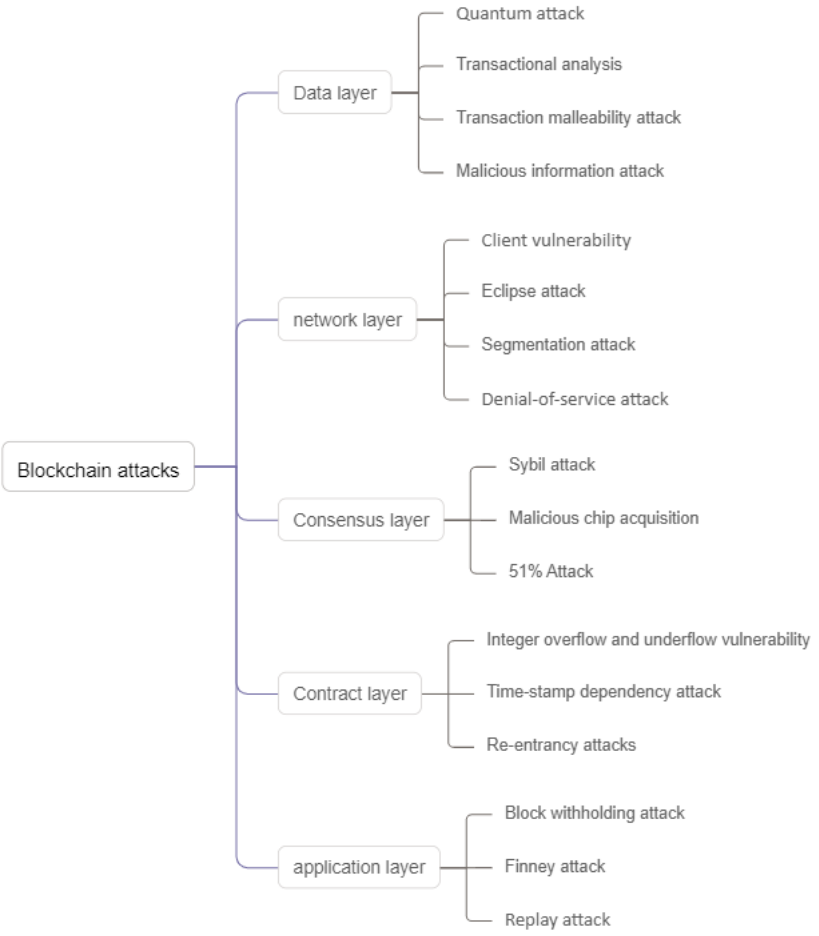

Figure 1. Blockchain attack classification (Photo/Picture credit: Original).

With blockchain playing an increasingly important role in business, there is a growing number of attempts to steal cryptocurrencies through attacking the blockchain. Moreover, in an era where internet privacy is hard to protect, transaction privacy and user identities become targets for attackers.

As shown in Figure 1, existing attack methods can be classified based on the hierarchical level of the targeted blockchain:

Data Layer: The data layer serves as the foundational framework for data management in blockchain technology. It represents the lowest level of this technology and plays a crucial role in encapsulating the diverse data structures within the blockchain.

Network Layer: The network layer is the fundamental level within the blockchain technology ecosystem. It ensures the decentralization and immutability of the blockchain. The network layer primarily faces attacks targeting the peer-to-peer (P2P) network.

Consensus Layer: The consensus layer is the core architecture of the blockchain technology ecosystem, encapsulating the consensus algorithm of the blockchain. In the authorized consensus mechanism, each node has equal influence over the consensus mechanism. In contrast, in non-consensus mechanisms, nodes compete for the right to validate transactions and earn rewards based on the "chips" they possess.

DOI: 10.54254/2755-2721/49/20241063

Contract Layer: The contract layer is a crucial component in ensuring trustworthy peer-to-peer interactions within the blockchain. The contract layer encapsulates various types of script codes, algorithm mechanisms, and smart contracts.

Application layer: The application layer is the application carrier of blockchain technology, providing solutions for various businesses.

2. Attacks on Blockchain's Data Layer

The data layer of blockchain utilizes cryptographic tools and a distributed architecture to ensure decentralization and immutability. However, attackers can exploit vulnerabilities in the data structure and public information to launch attacks.

2.1. Attacks Exploiting Private Data

Quantum attack: Quantum computers use Grover's search algorithm for mining, requiring significantly fewer hash functions than classical computers. Using the Grover search algorithm, searching for a desired item among N items typically requires $O(\sqrt{N})$ queries in a quantum computer, while a classical computer usually requires $\Omega(N)$ queries to achieve the same task [2, 3]. The emergence of quantum attacks will deal a devastating blow to the existing information security system. Aspects such as the mining mechanism of Bitcoin, the immutability of blockchain, and the confidentiality of blockchain data will all face significant challenges.

Transactional analysis: Attackers gather a large amount of users' publicly available information through methods like eavesdropping and create identity profiles for anonymous accounts [4]. They exploit user behavior and transaction patterns analysis to associate and obtain target users' identity privacy and transaction privacy.

2.2. Attacks Altering Blockchain Data

Transaction malleability attack: The attacker can modify transaction data before it is confirmed, creating a new transaction with a different hash. If the modified transaction is recorded first, miners will consider the original transaction to have a double-spending issue and refuse to include it in the block. The attacker can then appeal to the exchange, potentially resulting in a loss of funds if the appeal is approved [5].

Malicious information attack: Attackers write malicious information in the blockchain, such as virus signatures, malicious advertisements, politically sensitive topics, exploiting the immutable nature of the blockchain, leading to user side antivirus software reporting and causing politically sensitive issues [6].

2.3. Mitigation Strategies

For quantum resistance, we need to look for an alternative proof-of-work method that provides "No quantum advantage". Momentum is a good candidate in this regard. By utilizing the "Momentum" algorithm, a blockchain system can increase the difficulty of resisting quantum attacks, thereby enhancing the security of the network [7]. This is because attackers would need to allocate a significant amount of memory resources to carry out an attack, making it more challenging and expensive.

Below are two defense strategies that can be considered against Transactional analysis:

Data obfuscation: data can be subjected to obfuscation techniques such as introducing noise, perturbations, or transformations. These methods increase the complexity and difficulty for attackers to analyze the data effectively.

Covert transmission: Utilizing techniques like steganography or other covert methods during communication increases the difficulty for attackers to obtain the information.

A solution to transaction malleability attack is Segregated Witness [8]. The hash value of a block header is entirely determined by transaction information. therefore, the hash of a transaction will not change due to alterations in the signature information. Through this method, attackers cannot modify the hash value of a transaction unless they possess the private key.

To mitigate malicious information attacks, adding additional review processes would undermine the decentralized nature of blockchain. Therefore, the following measures can be implemented:

DOI: 10.54254/2755-2721/49/20241063

Restricting data formats: Implement restrictions on specific data formats to prevent the upload of viruses or malicious files onto the blockchain.

Implementing incentive mechanisms: Introduce a reward and punishment system to discourage the upload of malicious information. Nodes that upload malicious content should face penalties, while nodes that report such content should receive rewards.

3. Attacks on Blockchain's Network Layer

The P2P network architecture at the network layer is the fundamental technological structure of blockchain, and therefore attacks at this layer are targeted at the P2P network.

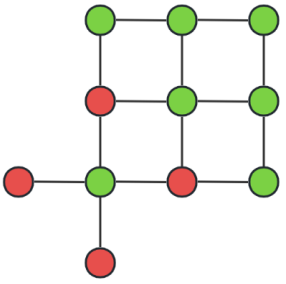

Figure 2. Eclipse attack (Photo/Picture credit: Original).

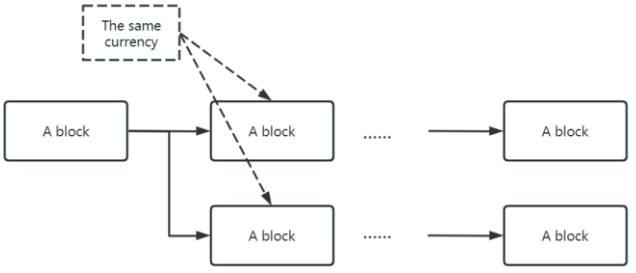

Figure 3. Segmentation attack (Photo/Picture credit: Original).

3.1. Attacks on Peer-to-Peer Networks

Client vulnerability: Attackers can use techniques such as 0-day vulnerability scanning to identify vulnerabilities in client systems, and then exploit these vulnerabilities to launch various attacks [9]. In 2018, blockchain security company PeckShield released a security vulnerability that allows attackers to cause two-thirds of Ethereum nodes to go offline by sending specific malicious messages to Ethereum clients.

Eclipse attack: As shown in Figure 2, the attacker manipulates multiple nodes to maintain long-term transmission with the attacked node, causing the number of online connections to reach the upper limit of the attacked node's connections. resulting in the target node being unable to maintain the blockchain ledger [10].

Segmentation attack: The Border Gateway Protocol (BGP) is one of the components of the Internet, and hijacking BGP can disrupt transaction and consensus processes. As shown in Figure 3, through BGP attacks, the blockchain is forked, allowing attackers to use the same electronic currency for transactions across multiple branches. After the blockchain is normalized, only one transaction will be preserved [11].

Denial-of-service (DoS) attack: Attackers use a large amount of network resources to attack computer systems or networks, causing them to stop responding thus deny service. In 2017, Poloniex suffered a DoS attack, which resulted in Bitcoin prices being locked and users unable to execute transactions normally [12].

DOI: 10.54254/2755-2721/49/20241063

3.2. Mitigation Strategies

In the attack scenario of Client vulnerability, the vulnerabilities in client-side code can often be attributed to negligence or coding errors during the development phase. As a result, it is nearly impossible to completely eliminate client-side vulnerabilities. However, developers can employ security assessment techniques such as Fuzzing during client-side development to identify potential vulnerabilities and mitigate security threats [13].

BlockQuick proposed by Letz can effectively prevent Eclipse attacks. When a miner submits a new block to the blockchain, nodes in the network will verify the miner's signature and compare it with the identities of known miners in the consensus reputation table. The new block is only accepted when the consensus score exceeds 50% [14].

The Segmentation attack can be addressed by tackling BGP hijacking attacks. AP2Vec leverages the network topology and inter-domain relationships of internet routing data, achieving a high accuracy in hijack detection and displaying good robustness against novel hijacking attacks [15].

For DoS attacks, the combination of sniffing technique and firewall can effectively address DoS attacks. Once the sniffing system detects unusual patterns and identifies them as attacks, it will promptly alert the firewall to isolate the attacker's original Internet Protocol (IP) address. As a result, all communication from the attacker's host to the target will be terminated [16].

4. Attacks on Blockchain's Consensus Layer

The consensus layer encapsulates various consensus algorithms of the blockchain and is critical to ensuring that all network nodes of the blockchain achieve correct consensus.

4.1. Attacks Exploiting Recentralization

Malicious chip acquisition: In non-permissioned consensus mechanisms, the more "chips" a node holds, the greater the likelihood of gaining the right to validate transactions. However, if a node holds too many chips, it violates the principle of decentralization in blockchain. Attackers can increase their chances of gaining validation rights by acquiring "chips" through attacks.

51% attack: In an non-permissioned consensus mechanisms, once the 'chips' of malicious nodes exceed half of the total' chips', they can obtain accounting rights and reach certain consensus. Alternatively, using the advantage of chip holdings, one could generate a blockchain fork and gradually replace the main chain with the forked chain [17].

Sybil attack: Attackers in the network operate on multiple different identities to achieve specific goals. In a blockchain network, attackers can achieve erroneous consensus among honest nodes by conducting a Sybil attack [18].

4.2. Attacks Involving Malicious Competition Among Mining Pools

Selfish mining attack: Selfish miners gain an advantage by mining blocks ahead of other nodes and selectively propagating them, deliberately creating forks. The attackers continue mining along the previous chain, invalidating the blocks of other miners. This attack is a legitimate operation, but it renders the work of other nodes ineffective and wastes a significant amount of computational power [19].

4.3. Mitigation Strategies

To counter attacks in non-permissioned mechanisms (preventing attackers from owning more than half of the "chips" in the entire network), Ethereum has proposed the Casper mechanism [20]. The Casper mechanism requires miners who generate blocks to lock up a certain amount of Ether as collateral. If the miners act honestly, they will receive rewards. Otherwise, Ethereum will seize the locked Ether from the miners.

P Swathi et al. proposed a solution for mitigating Sybil attacks. In their solution, each node actively monitors the behavior of other nodes in the network. If a node is found to be selectively propagating blocks of only specific users, it is swiftly identified, verified, and then reported to other nodes. This

DOI: 10.54254/2755-2721/49/20241063

proactive monitoring and reporting mechanism helps to restrict and mitigate the impact of Sybil attacks on the blockchain network [21].

The forced fairness algorithm is a method to mitigate the Selfish Mining attack. It establishes a mature defense system by utilizing the expected transaction confirmation height and block publishing height to detect selfish mining behavior [22].

5. Attacks on Blockchain's Contract Layer

The contract layer serves as a significant hallmark of the blockchain 2.0 technological framework, encapsulating smart contracts, algorithmic mechanisms, and script code.

5.1. Attacks on Smart Contracts

Integer overflow and underflow vulnerability: In the code of smart contracts, integer variables have upper and lower limits. When these limits are exceeded, the value of the number can change. Attackers often exploit vulnerabilities related to integer overflow to modify address pointers and execute malicious code [23]. In the attack on the BeautyChain (BEC) smart contract in 2018, the attacker successfully transferred a significant amount of BEC tokens by exploiting the Integer overflow and underflow vulnerability [24]. This incident caused massive panic, leading to a substantial sell-off of BEC tokens in the market.

Time-stamp dependency attack: The execution of smart contracts often relies on the timestamp of blocks, as different timestamps can yield varying results for smart contracts. For instance, suppose a smart contract requires determining the account eligible for receiving rewards based on the current timestamp. In this case, an attacker could potentially manipulate the process by attempting to pre-validate different timestamps during the mining process, thereby diverting the rewards to their desired account [25].

Re-entrancy attack: Attackers exploit reentrancy vulnerabilities in smart contract code to initiate attacks that result in the recursive invocation of two smart contracts [26]. In the Decentralized Autonomous Organization (DAO) attack [27], the attacker uses smart contract A to request a withdrawal from smart contract B. During this process, B transfers funds to A and triggers A's callback function. If the attacker has inserted a withdrawal request from A to B within the callback function, the process can be repeated recursively, depleting the account balance until it becomes insufficient.

Call deep attack: When a smart contract invokes itself or other smart contracts, the call stack depth of the transaction increases by 1. When the call stack depth reaches 1024, invoking the smart contract again will result in an exception being thrown. Attackers can exploit this exception to launch an attack. For instance, in the case of a smart contract containing a transfer transaction, an attacker can intentionally approach the call stack limit beforehand. When the transfer function is called, an exception is thrown, causing the transfer to fail [28].

5.2. Mitigation Strategies

Smart contracts are pieces of code written and deployed on the blockchain by developers. Attacks at the contract level can be the result of vulnerabilities caused by developer negligence or intentionally inserted by malicious developers. Therefore, it is necessary to impose certain requirements on developers:

To prevent common vulnerabilities such as Integer overflow and underflow vulnerability, developers must ensure rigor in their code development.

Developers can adopt various coding approaches during smart contract development to avoid complete reliance on timestamps for the execution of the contract.

Developers can protect existing deployed contracts against re-entrancy attack by utilizing a novel smart contract security technology called Sereum [29], which ensures backward compatibility. Sereum offers a means to safeguard deployed contracts from re-entrancy attacks without requiring any modifications to the existing contracts.

DOI: 10.54254/2755-2721/49/20241063

To mitigate Call deep attack, a smart contract can implement an alert mechanism. When the call stack limit is about to be reached, the smart contract issues a warning to the user. Additionally, the smart contract can invoke a penalty contract to punish the user who made the final call to the smart contract.

6. Attacks on Blockchain's Application Layer

6.1. Attacks Targeting Trading Accounts

Dusting attack: In the Bitcoin system, amounts of Bitcoin that are less than or equal to 100 satoshis are referred to as dust. Attackers send dust to target user wallets, and when users make transactions using this dust, attackers track the dust transactions to trace the user's address and even identify the individual associated with the wallet [30].

Replay attack: Replay attack occurs in a hard fork situation. Due to the two hard forked chains, whose addresses and private key production algorithms are the same, and the transaction format is completely the same, transactions on one chain are likely to be completely legal on the other chain [31].

6.2. Attacks Against Trading Platforms

Application programming interface (API) attack: Users typically perform order confirmation, cancellation, and other operations through private API interfaces within the trading platform. These interfaces require an API Key for authentication. However, if the API Key is leaked, it can cause significant losses to the user [32].

Brute-force attack: For a login interface, attackers can send multiple requests targeting the desired value in an attempt to obtain it through brute-force methods. In the case of password verification, attackers can use a large password dictionary to attempt a brute-force attack on an account's password [33].

6.3. Mitigation Strategies

In application layer attacks, there are primarily attacks targeting transaction accounts and trading platforms. The ultimate goal is to obtain the account information of the target node and illegally acquire user assets. To defend against these attacks, the following measures can be taken.

Adopt a password rating mechanism to notify users when their passwords have a low-security level.

Introduce additional authentication mechanisms, like an API call interface authentication, to vulnerable systems.

Cultivate security awareness and establish a system defense system.

Implement offline key management to prevent the use of identical passwords across different accounts.

The Anti-Dust security model proposed by Yunpeng Wang et al. can be utilized to counter the Dusting attack. This model analyzes the characteristics of dust attacks and draws insights from historical transaction data to construct a dust recognition algorithm based on Gaussian distribution [30].

A blockchain-based decentralized framework can be used to defend against Replay attacks. It utilizes a Bayesian inference mechanism that employs locally reported attack probabilities, which is specifically designed for a blockchain framework [34].

In a blockchain system based on the UTXO (Unspent Transaction Output) structure, each transaction is an atomic operation, and it is not possible to deduct a UTXO more than once. In a blockchain system based on Hyperledger Fabric, each account has a Nonce value that serves as a marker for the number of transactions originating from that account. When miners validate a transaction, they can compare the Nonce value included in the transaction with the Nonce value in the sender's account. Only when they are equal is the transaction considered legitimate.

7. Conclusion

Blockchain technology, marked by its decentralized architecture and resistance to tampering, represents a landmark innovation with a myriad of promising applications. This article classifies blockchain attacks

DOI: 10.54254/2755-2721/49/20241063

into five distinct layers: the data layer, the network layer, the consensus layer, the contract layer, and the application layer. Within the data layer, attacks are further categorized into those that exploit private data and those that alter blockchain data. Network layer attacks focus specifically on peer-to-peer networks, while consensus layer assaults involve the exploitation of recentralization and malicious competition among mining pools. At the contract layer, the focus is on smart contract vulnerabilities, and at the application layer, attacks target trading accounts and trading platforms. Central to the blockchain's integrity is the consensus algorithm, which serves as the primary safeguard for system security and trustworthiness. However, prevalent consensus algorithms are not without their challenges, especially in the context of security. The presence of malicious nodes within a distributed system has the potential to compromise the consensus algorithm's effectiveness. Future research avenues include the development of more robust and secure Byzantine Fault Tolerant (BFT) consensus algorithms. By doing so, the aim is to thwart manipulation attempts by malicious nodes and ensure the seamless operation of the blockchain ecosystem.

References

[1] Tian, G., Hu, Y., & Chen, X. (2021). Research progress on attacks and defense techniques in blockchain systems. Journal of Software, 32(5), 1495-1525.

[2] Aggarwal, D., Brennen, G. K., Lee, T., Santha, M., & Tomamichel, M. (2017). Quantum attacks on Bitcoin, and how to protect against them. arXiv preprint arXiv:1710.10377.

[3] Gao, Y. L., Chen, X. B., Chen, Y. L., Sun, Y., Niu, X. X., & Yang, Y. X. (2018). A secure cryptocurrency scheme based on post-quantum blockchain. Ieee Access, 6, 27205-27213.

[4] Awan, M. K., & Cortesi, A. (2017). Blockchain transaction analysis using dominant sets. In Computer Information Systems and Industrial Management: 16th IFIP TC8 International Conference, CISIM 2017, Bialystok, Poland, June 16-18, 2017, Proceedings 16 (pp. 229-239). Springer International Publishing.

[5] Decker, C., & Wattenhofer, R. (2014). Bitcoin transaction malleability and MtGox. In Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19 (pp. 313-326). Springer International Publishing.

[6] Fuji, R., Usuzaki, S., Aburada, K., Yamaba, H., Katayama, T., Park, M., ... & Okazaki, N. (2019, March). Investigation on sharing signatures of suspected malware files using blockchain technology. In International Multi Conference of Engineers and Computer Scientists (IMECS) (pp. 94-99).

[7] Larimer, D. A. N. I. E. L. (2014). Momentum–a memory-hard proof-of-work via finding birthday collisions. Tech. Rep., Oct. 2013. [Online]. Available: http:// invictus -innovations. Com /s/MomentumProofOfWork-hok9. pdf.

[8] Kedziora, M., Pieprzka, D., Jozwiak, I., Liu, Y., & Song, H. (2023). Analysis of segregated witness implementation for increasing efficiency and security of the Bitcoin cryptocurrency. Journal of Information and Telecommunication, 7(1), 44-55.

[9] Moubarak, J., Filiol, E., & Chamoun, M. (2018, April). On blockchain security and relevant attacks. In 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM) (pp. 1-6). IEEE.

[10] Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse attacks on {Bitcoin's}{peer-to-peer} network. In 24th USENIX security symposium (USENIX security 15) (pp. 129-144).

[11] Apostolaki, M., Zohar, A., & Vanbever, L. (2017, May). Hijacking bitcoin: Routing attacks on cryptocurrencies. In 2017 IEEE symposium on security and privacy (SP) (pp. 375-392). IEEE.

[12] Chaganti, R., Boppana, R. V., Ravi, V., Munir, K., Almutairi, M., Rustam, F., ... & Ashraf, I. (2022). A comprehensive review of denial of service attacks in blockchain ecosystem and open challenges. IEEE Access.

[13] Clarke, T. (2009). Fuzzing for software vulnerability discovery. Department of Mathematic, Royal Holloway, University of London, Tech. Rep. RHUL -MA -2009 -4.

DOI: 10.54254/2755-2721/49/20241063

[14] Letz, D. (2019). Blockquick: Super-light client protocol for blockchain validation on constrained devices. Cryptology ePrint Archive.
[15] Shapira, T., & Shavitt, Y. (2022). AP2Vec: an unsupervised approach for BGP hijacking detection. IEEE Transactions on Network and Service Management, 19(3), 2255-2268.
[16] Fidele, K. A., & Hartanto, A. (2019). DoS Attack Prevention Using Rule-Based Sniffing Technique and Firewall in Cloud Computing. In E3S Web of Conferences (Vol. 125, p. 21004). EDP Sciences.
[17] Aponte-Novoa, F. A., Orozco, A. L. S., Villanueva-Polanco, R., & Wightman, P. (2021). The 51% attack on blockchains: A mining behavior study. IEEE Access, 9, 140549-140564.
[18] Douceur, J. R. (2002, March). The sybil attack. In International workshop on peer-to-peer systems (pp. 251-260). Berlin, Heidelberg: Springer Berlin Heidelberg.
[19] Grunspan, C., & Pérez-Marco, R. (2018). On profitability of selfish mining. arXiv preprint arXiv:1805.08281.
[20] Jain, A., Arora, S., Shukla, Y., Patil, T., & Sawant-Patil, S. (2018). Proof of stake with casper the friendly finality gadget protocol for fair validation consensus in ethereum. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 3(3), 291-298.
[21] Swathi, P., Modi, C., & Patel, D. (2019, July). Preventing sybil attack in blockchain using distributed behavior monitoring of miners. In 2019 10th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-6). IEEE.
[22] Saad, M., Njilla, L., Kamhoua, C., & Mohaisen, A. (2019, February). Countering selfish mining in blockchains. In 2019 International Conference on Computing, Networking and Communications (ICNC) (pp. 360-364). IEEE.
[23] Anley, C., Heasman, J., Lindner, F., & Richarte, G. (2011). The shellcoder's handbook: discovering and exploiting security holes. John Wiley & Sons.
[24] Sun, T., & Yu, W. (2020). A formal verification framework for security issues of blockchain smart contracts. Electronics, 9(2), 255.
[25] Wang, Q., He, L., Zhu, X., Huang, Y., & Li, Z. (2021, September). Privacy protection of blockchain security development status. In 2021 4th International Conference on Information Systems and Computer Aided Education (pp. 2592-2596).
[26] Rodler, M., Li, W., Karame, G. O., & Davi, L. (2018). Sereum: Protecting existing smart contracts against re-entrancy attacks. arXiv preprint arXiv:1812.05934.
[27] Mehar, M. I., Shier, C. L., Giambattista, A., Gong, E., Fletcher, G., Sanayhie, R., ... & Laskowski, M. (2019). Understanding a revolutionary and flawed grand experiment in blockchain: the DAO attack. Journal of Cases on Information Technology (JCIT), 21(1), 19-32.
[28] Alharby, M., & Van Moorsel, A. (2017). Blockchain-based smart contracts: A systematic mapping study. arXiv preprint arXiv:1710.06372.
[29] Rodler, M., Li, W., Karame, G. O., & Davi, L. (2018). Sereum: Protecting existing smart contracts against re-entrancy attacks. arXiv preprint arXiv:1812.05934.
[30] Wang, Y., Yang, J., Li, T., Zhu, F., & Zhou, X. (2018, July). Anti-dust: a method for identifying and preventing blockchain's dust attacks. In 2018 international conference on information systems and computer aided education (ICISCAE) (pp. 274-280). IEEE.
[31] Anita, N., & Vijayalakshmi, M. (2019, July). Blockchain security attack: A brief survey. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.
[32] Kiktenko, E. O., Kudinov, M. A., & Fedorov, A. K. (2019, June). Detecting brute-force attacks on cryptocurrency wallets. In International Conference on Business Information Systems (pp. 232-242). Cham: Springer International Publishing.
[33] Anderson, R. (2020). Security engineering: a guide to building dependable distributed systems. John Wiley & Sons.

DOI: 10.54254/2755-2721/49/20241063

[34] Ramanan, P., Li, D., & Gebraeel, N. (2021). Blockchain-based decentralized replay attack detection for large-scale power systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(8), 4727-4739.