iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.78603

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

A Concise Review of Blockchain Attack Vectors: A Layered Taxonomy

Ali Rezaei
VAISR Research Group, Tehran, Iran

Abstract: ***This paper is a systematic examination of the security environment in decentralized ledger systems. As the ecosystem of blockchain systems is in a constant state of evolution, threats have been identified in various areas of the system stack, from the base network stack to the intricacies of the economics of the system. We have identified these threats in five main areas of the system: Infrastructure & Network, Smart Contract & VM, Economic & Composability, dApp & Application, and Forensic & Data Analysis.***

Keywords: Blockchain, blockchain Security, DAPP Security, Smart Contract

I. INTRODUCTION

This rapid evolution of the underlying technology has caused it to move from a niche cryptographic experiment into a pillar of the modern digital economy [1]. It has redefined the way in which trust is built and delivered in financial systems, supply chains, and governance [2]. As the TVL of these ecosystems continues to scale to unprecedented heights, so has the surface area for sophisticated cyber-attacks [3,4].

Whilst security risks were traditionally centered on the integrity of consensus algorithms, such as the 51% attack or double spending [5], the current risk landscape is significantly more complex.

The advent of programmable logic in the form of Smart Contracts and the emergence of Decentralized Finance (DeFi) have created a whole new generation of security risks [6]. These include code-based risks such as reentrancy and arithmetic overflow, as well as complex economic attacks such as flash loans and oracle attacks [7].

However, despite the "code is law" ideology often linked with decentralized systems, these aspects of the system remain weak. Classic web attacks such as phishing and cross-site scripting have been modified to attack decentralized applications as well [8], while data analysis is used more and more to attack the privacy of users through de-anonymization [9].

This paper seeks to create a comprehensive taxonomy of the current attack landscape on blockchain systems. By dividing the attack domains into five distinct areas of Infrastructure, Smart Contracts, Economic Composability, dApp Architecture, and Forensic Analysis, we hope this paper provides a holistic approach for researchers and developers to better comprehend the risks of the blockchain system and how to mitigate them [10].

II. BACKGROUND AND PRELIMINARIES

A. Blockchain Technology

Blockchain technology, being a decentralized, distributed ledger technology, enables the recording of transactions among a network of computers in a way that data cannot be altered or compromised, thus providing immutability and transparency of data [1, 2]. At its core, blockchain technology utilizes a Consensus Mechanism, which may be based on a Proof of Work (PoW) or a Proof of Stake (PoS), to validate transactions and achieve a consensus among a network of computers without the need for a central authority [4, 11]. However, the decentralized nature of the network poses a number of network-level threats, which may compromise the integrity of the consensus among a network of computers, including Sybil and Eclipse attacks.

B. Smart Contracts and the EVM

Smart contracts are programs written on the blockchain that execute automatically and facilitate, verify, or enforce the negotiation of a contract [1, 6]. Smart contracts in the Ethereum environment are written in high-level programming languages such as Solidity and executed by the Ethereum Virtual Machine (EVM) [8, 11].

Smart contracts, although eliminating the middleman, are vulnerable to code-related attacks such as reentrancy and arithmetic overflow, which may result in devastating losses [6, 13].

C. *Decentralized Applications (dApps)*

Decentralized Applications, or dApps, are software applications running on a distributed computing system, generally using smart contracts as the backend logic [8, 14]. Unlike conventional applications, dApps communicate with the blockchain, and the frontend interfaces and proxy structures of dApps have become a primary vector for web-based attacks such as XSS and phishing [10, 15]. These types of vulnerabilities stem from the dichotomy between the nature of decentralized protocols and the centralized web infrastructure used to access these protocols [3, 16].

D. *The Mempool and Transaction Ordering*

The Mempool, or Memory Pool, can be defined as the waiting area for unconfirmed transactions, which are then taken by miners or validators and added to a block [15]. Since the mempool is public, it allows anyone to view the unconfirmed transactions, which leads to "Transaction Ordering Attacks" like front-running [6, 16]. This environment makes it possible for attackers to exploit the sequence of execution for their own benefit, also known as Miner Extractable Value (MEV) [15, 18].

III. A MULTI-LAYERED TAXONOMY OF BLOCKCHAIN ATTACKS

The security of a blockchain ecosystem is not a monolithic problem but a distributed responsibility over a number of layers. This section offers a detailed analysis of the attack vectors identified in our taxonomy.

A. *Infrastructure and Network Layer*

The base layer of any blockchain system primarily focuses on consensus mechanisms and P2P networking. Attacks on this level target the integrity of the blockchain itself.

1) Consensus Manipulations: In this type of attack, the adversary can conduct a 51% Attack, where the adversary controls the majority of the hash rate, allowing it to manipulate the blockchain [4, 9]. In other cases, the adversary can conduct a Selfish Mining attack, where the adversary can selfishly mine blocks to obtain unfair benefits [19]. In other instances, the adversary can conduct a Bribery Attack, where the adversary bribes miners to collude and attack the blockchain protocol [19]. In the case of the Proof of Stake blockchain, the Nothing at Stake attack allows adversaries to validate two or more branches of the blockchain to ensure profit [4, 21].

2) Network Disruptions: Network Disruptions: Sybil Attacks involve the creation of many fake identities to overwhelm the network's influence and consensus participation [9, 22]. In order to attack certain nodes, Eclipse Attacks involve the creation of a malicious environment around a node, which provides it with false information or denies it the chance to view the longest valid chain [12, 23]. Timejacking Attacks involve the interference with the synchronization of nodes through the provision of false timestamps, which causes the creation of forks in the blockchain [24], while Routing Attacks involve BGP hijacking, which interferes with communication between nodes at the ISP level [24,25].

B. *Smart Contract and Virtual Machine (VM) Layer*

This domain encompasses vulnerabilities within the execution logic (often Solidity) and the EVM environment.

1) Logic and Execution Faults: Logic and Execution Faults: The infamous Reentrancy vulnerability is a situation where a contract is executed multiple times before the state is updated, resulting in unexpected funds being depleted. On the other hand, Arithmetic Overflow/Underflow occurs when a number exceeds its representation limit, for example, when a number exceeds 2^256, thus making it possible for attackers to skip balance checks. Additionally, attackers may exploit Timestamp Dependence by manipulating the block timestamp, for example, in gambling contracts and time-locked withdrawable contracts. Finally, improper Access Control and visibility modifiers may allow unauthorized users to assume administrative privileges, thus compromising the fundamental functionality of a contract [26, 27,28].

2) Low-Level Memory Issues: The misuse of Delegatecall can result in arbitrary code execution outside the intended context, as it allows the target contract to change the state of the calling contract [6, 13]. There are also other risks, such as Calls to the Unknown, which can result in unexpected fallbacks and can be abused for reentrancy or denial-of-service [7, 29]. Additionally, the Uninitialized Storage variables can result in attack entries, where critical contract pointers are overwritten, while the Short Address Attacks can result from the flaws in the EVM buffer checking, which can be used to manipulate the parameters during transaction decoding.

C. Economic and Composability Layer

As protocols interact, complex market-based threats emerge from their synergy.

1) Market Exploits: Flash Loan Attacks use massive, uncollateralized loans to manipulate markets or arbitrage opportunities in a single block [6, 31]. These attacks have been paired with Oracle Manipulation, where the external inputs used by smart contracts to determine the asset price are tampered with, creating the ability to borrow assets at false exchange rates [13, 32]. These types of attacks have shown the potential risks in the composability of decentralized financial systems [33].
2) MEV and Transaction Ordering: MEV and Transaction Ordering: Actors in this attack type could carry out a Front-running attack in which the attacker searches the mempool for profitable transactions and "jumps ahead" of them by sending the same transaction with a higher gas fee attached in order to execute the transaction first [14, 16]. This is a key aspect of the Maximal Extractable Value (MEV) attack in which the searcher and the validator exploit the ordering of the transactions in order to maximize their value [17, 34]. On the other hand, the attacker could carry out a Mempool DoS attack in which the attacker sends a number of "dust" transactions into the network in order to fill the mempool with useless transactions, thus increasing the gas fees and effectively "paralyzing" the network [9, 35].

D. dApp and Application Layer

This layer bridges the gap between decentralized protocols and the end-user, often inheriting traditional web risks.

1) Interface Vulnerabilities: Traditional Web Vulnerabilities: Common web-based threats like Cross-Site Scripting (XSS) and SQL Injection have been found to be used to attack the frontend interfaces of dApps, allowing the attacker to inject malicious scripts into the user session [7, 36]. Despite the use of robust cryptographic protocols, Phishing has been found to be the most prominent and successful attack vector, where a user is tricked into revealing their private keys or signing a malicious transaction [9, 37]. These types of attacks have been found to take advantage of the "human-in-the-loop" vulnerability, bypassing robust technical security [8, 38].
2) Architecture Risks: Proxy Contract Attacks are attacks against the dApps' upgradeability feature, where attackers take advantage of the inconsistency between the proxy contract and the implementation contract to divert users or modify the dApps' logic [15, 39]. Critical problems also stem from poor Key Management, which may result in a complete loss of control [7, 36]. Moreover, the application of Weak Field Modifiers, such as marking data as "private" in Solidity, does not provide confidentiality, as this data may be easily inferred or retrieved from the public chain history [5, 40].

E. Forensic and Data Analysis Layer

These attacks prioritize the extraction of metadata and the de-anonymization of participants.

1) Privacy Invasions: Dust Attack is a technique where small quantities of currency are broadcast to thousands of users in order to link unconnected user accounts to a single user identity through the use of a co-spending mechanism in the next transaction [8, 41]. In addition, advanced Blockchain Ingestion tools and side-channel analysis allow the scraping and aggregation of public ledger data, which can be used to perform targeted social engineering and/or predatory price manipulation based on user patterns [9, 16]. These techniques take advantage of the pseudo-anonymous nature of public blockchains in order to reduce the privacy of the participants [2, 42].
2) Side-Channel Vectors: Side-Channel Attacks involve monitoring the performance of a system, the timing of execution, and other information to achieve a competitive or malicious edge over other participating entities [8, 43]. In the context of a blockchain, side-channel attacks may allow for the discovery of key information, such as a private key, through a power analysis of a node's execution or network propagation delay to de-anonymize the originating entity of a transaction [42, 44]. By monitoring non-functional attributes of a node's hardware or software execution, attackers may be able to circumvent cryptographic security without compromising the algorithms used to secure a system [9, 45].

Table 1: Classification of Security Threats Across Blockchain Domains

Domain	Primary Focus	Key Attack Examples
Infrastructure	Consensus & P2P Networking	51% Attack, Sybil, Eclipse
Smart Contract	Logic & EVM Execution	Reentrancy, Overflow, Access Control
Economic	Market & Composability	Flash Loans, Oracle Manipulation, Front-running
Application	User Interface & Management	Phishing, Proxy Attacks, Key Theft
Forensic	Privacy & Data Analysis	Dust Attack, Blockchain Ingestion

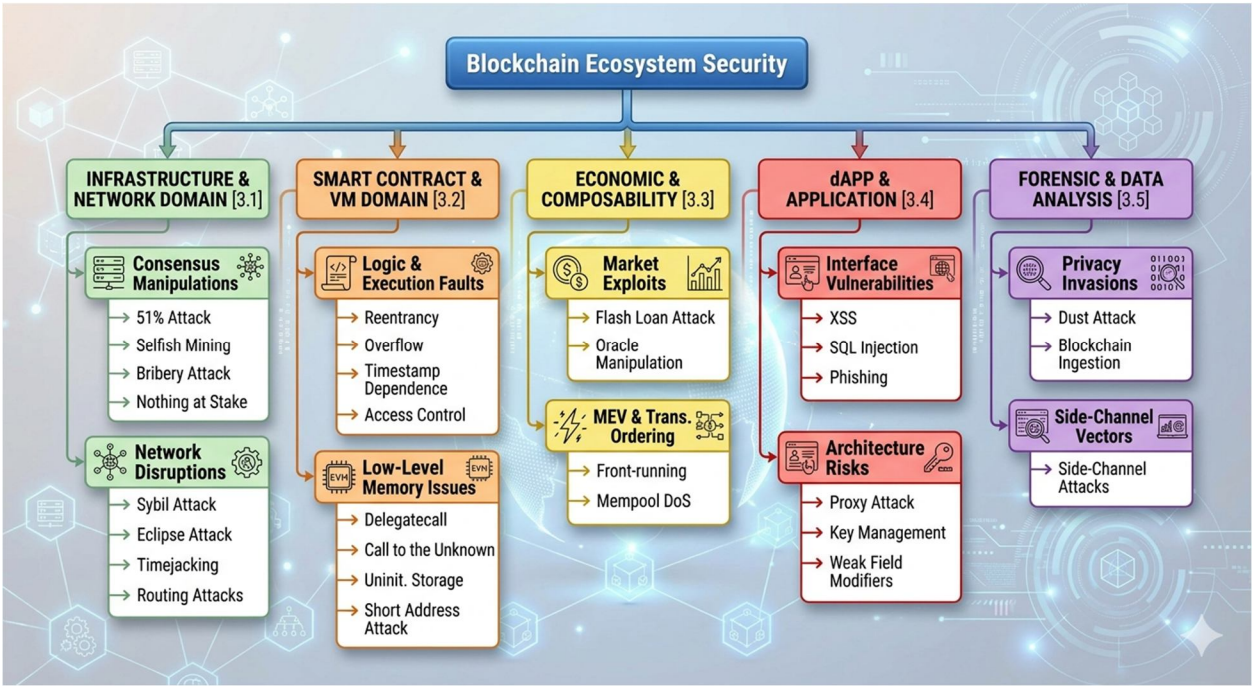


Figure 1: Blockchain Echosystem Security

IV. CONCLUSION

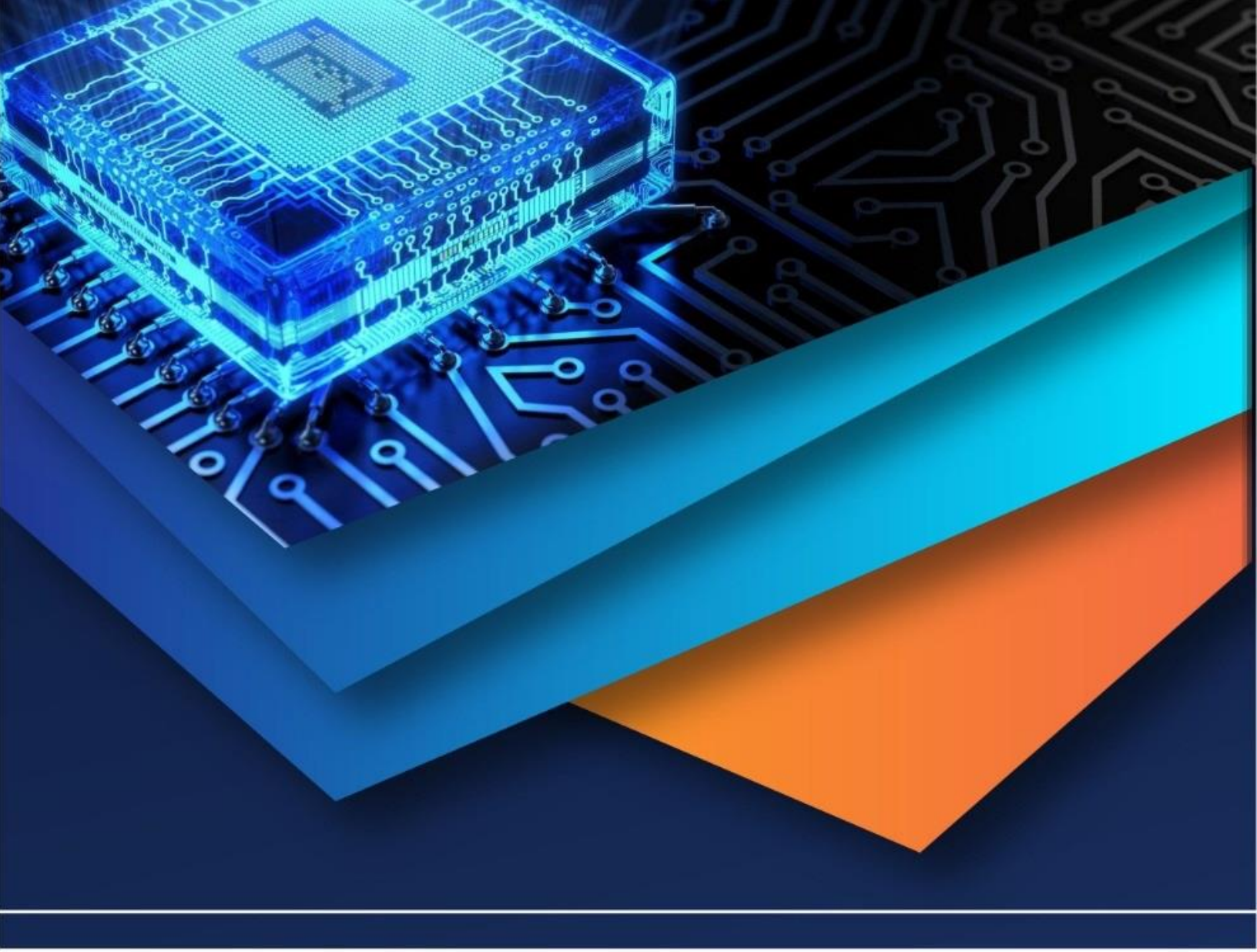
This shift towards a decentralized digital economy has created a multifaceted and complex threat landscape. As illustrated in this taxonomy, the threats are not limited to a specific layer but have a presence throughout the entire blockchain stack, from the fundamental P2P network and consensus algorithms to the advanced logic in smart contracts and the financial interactions in decentralized finance.

Our analysis has shown that, although the blockchain technology has the fundamental property of immutability, it is not impervious to common web-based threats such as phishing and XSS, nor is it immune to the latest threats in the decentralized ecosystem, such as flash loan attacks and oracle manipulation. Moreover, although the transparency provided by the public ledger is a fundamental property, it also makes the technology vulnerable to advanced de-anonymization using dust attacks and ingesting forensic data. As the ecosystem develops, the "code is law" principle must be complemented by security audits, key management, and defensive techniques. As a next step, we should investigate mitigation techniques that can effectively protect users from the combined effect of technical and economic attacks. To secure the future of blockchains, we need a comprehensive approach that recognizes the interdependency of infrastructure, code, and incentives.

REFERENCES

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.
[2] Tapscott, D., & Tapscott, A. (2016). Blockchain revolution. Penguin.
[3] Deirmentzoglou, E., Papakyriakopoulos, G., & Patsakis, C. (2019). A survey on blockchain surveys. IEEE Access, 7, 6452–6476.
[4] Pishdar, M., & Manzoor, J. (2026). Why no consensus on consensus? A deep dive into blockchain consensus protocols (arXiv:2603.08629). arXiv.
[5] Sayeed, S., & Marco-Gisbert, H. (2019). Assessing blockchain consensus and security mechanisms. Algorithms, 12(1), 10.
[6] Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts (SoK). In Principles of Security and Trust (POST).
[7] Werner, S. M., et al. (2021). SoK: Decentralized finance (DeFi). In IEEE Computer Security Foundations Symposium (CSF).
[8] Chen, H., et al. (2020). A survey on Ethereum systems security: Vulnerabilities, attacks, and defenses. ACM Computing Surveys (CSUR), 53(3), 1–35.
[9] Conti, M., et al. (2018). A survey on security and privacy issues of Bitcoin. IEEE Communications Surveys & Tutorials, 20(4), 3416–3452.
[10] Saad, M., et al. (2020). Exploring the attack surface of blockchain: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(3), 1977–2008.
[11] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger (Ethereum Project Yellow Paper).
[12] Heilman, E., et al. (2015). Eclipse attacks on Bitcoin's peer-to-peer network. In USENIX Security Symposium.
[13] Luu, L., et al. (2016). Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.
[14] Eskandari, S., et al. (2021). SoK: Oracles from the ground truth to market manipulation (arXiv:2106.00667). arXiv.
[15] Daian, P., et al. (2020). Flash boys 2.0: Front-running in decentralized exchanges. In IEEE Symposium on Security and Privacy (S&P).
[16] Fröwis, M., & Böhme, R. (2018). In code we trust? Measuring the adaptability of smart contracts. In International Conference on Blockchain (ICBC).
[17] Eskandari, S., et al. (2019). SoK: Transparent dishonesty: Front-running attacks on blockchain. In Financial Cryptography and Data Security (FC).

[18] Qin, K., et al. (2022). Quantifying blockchain extractable value: Before and after the merge. In IEEE Symposium on Security and Privacy (S&P).
[19] Eyal, I., & Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. Communications of the ACM, 57(7), 95–102.
[20] Bonneau, J. (2016). Why buy when you can rent? Bribery attacks on Bitcoin-style proof-of-work. In Financial Cryptography and Data Security.
[21] Li, W., et al. (2017). A survey on the security of blockchain consensus algorithms. In IEEE Conference on Communications and Network Security (CNS).
[22] Douceur, J. R. (2002). The Sybil attack. In International Workshop on Peer-to-Peer Systems (IPTPS).
[23] Marcus, Y., et al. (2018). Low-resource eclipse attacks on Ethereum's P2P network. In ACM SIGSAC Conference on Computer and Communications Security.
[24] Culver, K. (2011). Bitcoin-Timejacking (Technical Report).
[25] Apostolaki, M., et al. (2017). Hijacking Bitcoin: Routing attacks on advertising protocols. In IEEE Symposium on Security and Privacy (S&P).
[26] Pishdar, M., Lei, Y., Harfoush, K., & Manzoor, J. (2025). Denial-of-service attacks on permissioned blockchains: A practical study. Journal of Cybersecurity and Privacy, 5(3), 39.
[27] Torres, C. F., et al. (2018). Osiris: Trustworthy smart contract execution. In Annual Computer Security Applications Conference (ACSAC).
[28] Pishdar, M., Bahaghighat, M., Kumar, R., & Xin, Q. (2024). Major vulnerabilities in Ethereum smart contracts: Investigation and statistical analysis. EAI Endorsed Transactions on Internet of Things, 11.
[29] Fattahdizaji, A., Pishdar, M., & Shukur, Z. (2024, October). Investigating cyber threats against proof-of-work blockchain networks. In 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS) (pp. 1–14). IEEE.
[30] Fattahdizaji, A., Pishdar, M., & Shukur, Z. (2026). SmartGraphical: A human-in-the-loop framework for detecting smart contract logical vulnerabilities via pattern-driven static analysis and visual abstraction (arXiv:2603.08580). arXiv.
[31] Payer, U., et al. (2018). A survey of security vulnerabilities in Ethereum smart contracts. In International Conference on Information Security.
[32] Zhang, P., et al. (2020). Smart contract security: A survey. In IEEE International Conference on Blockchain.
[33] Nikolic, I., et al. (2018). Finding the greedy, prodigal, and suicidal smart contracts at scale. In Annual Computer Security Applications Conference (ACSAC).
[34] Feist, J., et al. (2019). Slither: A static analysis framework for smart contracts. In WETSEB.
[35] Karame, G. O., et al. (2012). Two Bitcoins at the price of one? Double-spending attacks on fast payments. In ACM SIGSAC Conference on Computer and Communications Security.
[36] Qin, K., et al. (2021). Attacking the DeFi ecosystem with flash loans for fun and profit. In Financial Cryptography and Data Security.
[37] Angeris, G., et al. (2020). Analysis of Uniswap markets. Cryptoeconomic Systems.
[38] Gudgeon, L., et al. (2020). DeFi is the future of finance, but is it secure? ACM SIGMETRICS Performance Evaluation Review.
[39] Zhou, L., et al. (2021). Just-in-time liquidity: Risks and rewards in DeFi (arXiv:2106.01830). arXiv.
[40] Johnson, A., et al. (2020). Liveness and denial-of-service in Ethereum. In ACM Conference on Advances in Financial Technologies (AFT).
[41] He, Y., et al. (2020). Security analysis of cryptocurrency wallets. In ACM Conference on Advances in Financial Technologies (AFT).
[42] Vasek, M., & Moore, T. (2015). There's no free lunch, even using Bitcoin: Tracking the popularity and profits of virtual currency scams. In Financial Cryptography and Data Security.
[43] Onaolapo, J., et al. (2016). The adventures of a discarded Bitcoin wallet. In Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs).
[44] Mense, A., & Boxler, C. (2020). Analysis of the proxy pattern in Ethereum smart contracts. In International Electronics Communication Conference (IECC).
[45] Biryukov, A., & Tikhomirov, S. (2019). Security and privacy of mobile wallet applications. In IEEE Symposium on Security and Privacy (S&P).
[46] Meiklejohn, S., et al. (2013). A fistful of Bitcoins: Characterizing payments among men with no names. In Internet Measurement Conference (IMC).
[47] Biryukov, A., et al. (2014). Deanonymisation of clients in Bitcoin P2P network. In ACM SIGSAC Conference on Computer and Communications Security.
[48] Kocher, P., et al. (2011). Introduction to differential power analysis. Journal of Cryptographic Engineering.
[49] Fanti, G., & Viswanath, P. (2017). Deanonymization in the Bitcoin P2P network. In Advances in Neural Information Processing Systems (NIPS).
[50] Genkin, D., et al. (2017). LVI: Low-value information leakage in Bitcoin-style systems. In USENIX Security Symposium.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY