Incident Analysis of Decentralized Finance

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Decentralized Finance (DeFi) has emerged as a transformative force in the financial landscape, introducing novel financial instruments built on blockchain technology. However, this rapid evolution has brought about unique challenges in ensuring the security of DeFi protocols. While several studies have contributed valuable insights into DeFi security from various perspectives, the landscape remains characterized by a persistent surge in incidents. Therefore, this paper systematically examines prominent incidents that have transpired from June 2022 to May 2023. Our findings underscore the significance of continuous vigilance in DeFi operations and propose future directions for bolstering blockchain security.**

***Index Terms*—Decentralized Finance, DeFi, Flash loan, Oracle, Reentrancy**

I. Introduction

Blockchain technology possesses many powerful features such as decentralization, persistence, anonymity, and auditability [1]. These features enable the emergence of innovative applications such as decentralized finance (DeFi), decentralized autonomous organizations (DAO), decentralized identities (DID), and more. Notably, DeFi has emerged as a compelling solution to rectify the inherent deficiencies of conventional financial systems, often referred to as CeFi. These limitations include inefficiency, centralized control, restricted accessibility, opacity, and a dearth of interoperability [2].

The features of DeFi have attracted a substantial influx of capital. In April 2022, the total value locked in DeFi reached a peak of 200 billion USD [3]. Despite experiencing a significant decline afterward, as of April 2023, the total value locked in DeFi still amounts to a substantial sum of 49 billion USD.

However, the high level of transparency and anonymity in DeFi also provides attackers with opportunities to exploit capital. In March 2023, Euler Finance, a decentralized lending protocol, suffered an attack that drained cryptocurrencies equivalent to 197 million USD [4]. Notably, up to 6 auditing firms were partnering with Euler Finance, yet none of them identified vulnerabilities resulting in this incident. This indicates that the DeFi industry still faces security challenges.

Despite previous efforts to systematically classify DeFi incidents [5], as well as develop scanning tools to detect vulnerabilities, DeFi incidents continue to arise frequently. We systematically categorized DeFi incidents from June 2022 to May 2023 based on prior studies. Then, we selected several scanning tools to scan these vulnerable contracts and verify whether these existing scanning tools can effectively identify vulnerabilities before attacks occur.

The contributions of this paper are:

(1) Analyze the DeFi incidents with a loss of more than 1 million dollars from June 2022 to May 2023 using incident types proposed by [5] and [6].
(2) Examine the vulnerable contracts using open-source scanning tools to determine whether these tools can detect the vulnerabilities that led to these incidents.
(3) Provide protection measures and directions for future research.

II. Analysis of Vulnerabilities

Despite many academic institutions and businesses conducting systematic analyses, there is currently no globally accepted standard for classifying blockchain vulnerabilities and attacks. This is because blockchain technology has been evolving rapidly, and new vulnerabilities and attack methods are constantly emerging, making it difficult to devise a complete and comprehensive classification standard.

After a thorough review of research papers [5]–[9], the 5-layer framework proposed by [5] presents the most comprehensive coverage of all vulnerabilities in blockchain, shown in Table I. We have opted to utilize this framework to categorize the vulnerabilities in DeFi, as it enables a more comprehensive and precise analysis.

A. Network Layer (NET)

In the Network Layer, nodes communicate by transmitting messages through various network protocols, such as TCP/IP and DNS [10]. Common incident types in this layer include:

1) Network Transparency: Exposing transaction data to the public network could provide attackers with the opportunity to create counterfeit transactions. Blockchain systems should create secure TLS connections to miners, and confidentially share all credentials [11].

2) Improper Peer Discovery: Peer-discovery may expose the location and identity of nodes, enabling attackers to pinpoint nodes, such as eclipse attacks [12]. Attackers may also abuse poorly designed peer discovery algorithms to launch denial of service attacks.

3) Network Congestion: This usually refers to denial of service attacks.

TABLE I
INCIDENT LAYERS, TYPES, AND CAUSES

Layer	Incident Cause	Incident Type
Network	Network layer transparency	Transaction content transparency
		Propagation transparency
	Improper peer discovery / churning logic	Eclipse attack
		Sybil attack
	Network congestion	Denial of service (DoS)
	Exposed Internet service	Sensitive DNS servers
		Unreliable BGP messages
Consensus	Blockchain protocol vulnerabilities	-
	Unstable incentive mechanism	Majority / 51% attack
		Block reorganization
		Selfish mining
		Double spending
		Feather forking
		Bribery attacks
		Mining difficulty adjustment
	Unfair sequencing	Sequencer transaction order manipulation
		Transaction censoring
Smart Contract	Untrusted or unsafe calls	Direct call to untrusted contract
		Reentrancy
		Delegate call / code injection
	Coding mistake	Unhandled or mishandled exception
		Locked or frozen asset
		Integer overflow or underflow
		Absence of coding logic or sanity check
		Short address
		Casting
		Unbounded or gas-costly operation
		Arithmetic mistakes
	Access control mistake	Inconsistent access control
		Visibility error and unrestricted action
	State transition design mistakes	Under-priced opcodes
		Outdated compiler version
Protocol	Transaction order dependency mistake	Front-running
		Back-running
		Sandwiching
	Replayable design	Transaction / strategy replay
	Block state dependency mistake	Randomness
		Other block state dependency
	Permissionless interaction	Camouflage a token contract
		Camouflage a non-token contract
	Unsafe dependency	On-chain oracle manipulation
		Governance attack
		Token standard incompatibility
		Liquidity borrow, purchase, mint, deposit
		Unsafe call to phantom function
	Unfair or unsafe interaction	Unfair slippage protection
		Unfair liquidity providing
		Unsafe or infinite token approval
Auxiliary Services	Faulty web development	-
	Faulty operation	Compromised private key / wallet
		Weak password
		Deployment mistake
	Off-chain oracle manipulation	Malicious oracle updater
		Malicious data source
		External market manipulation
	Greedy operator	Backdoor / Honeypot
		Insider trade or other activity
		Phishing attack
		Authority control or breach of promise
	Faulty blockchain service provider	Faulty wallet provider
		Faulty API / RPC

4) Exposed Internet Services: A blockchain system must secure network access services to prevent BGP hijacking and DNS hijacking. This incident cause was first observed in an attack against MyEtherWallet. Due to a BGP flaw [13], an attacker can announce ownership of any prefix to its neighbor routers to redirect users to a fake website and compromise victims' private keys.

B. Consensus Layer (CON)

The Consensus Layer is used to determine which transactions are included in a block and to ensure the consistency and security of the blockchain. There are multiple algorithms for the Consensus Layer [14], such as Proof of Work (PoW) in Bitcoin, and Proof of Stake (PoS) in Ethereum. Common incident types in this layer include:

1) Unstable Incentive Mechanism: Incident types in this category are quite diverse. All of them are enumerated below with brief explanations.

- **Majority / 51% Attack:** When a group of nodes controls over 50% of the total computing power of a blockchain, it allows them to rewrite blocks [15]. Even if the group controls lower but close to 51%, they still have chances to overwrite blocks.
- **Block Reorganization:** When a block is removed from the blockchain due to a longer chain being discovered, it results in the loss of previously confirmed transactions. During the reorganization, attackers may exploit the asynchrony of information to launch attacks [16].
- **Selfish Mining:** A group of miners solves a hash, creates a new block, and keeps it hidden. This creates a fork, which is mined to surpass the public blockchain. If the group's chain outpaces the honest one, they overwrite the original blockchain [17].
- **Double Spending:** A user attempts to spend the same cryptocurrency twice. Though consensus algorithms prevent spending multiple times, this can occur in a majority / 51% attack.
- **Feather Forking:** A small group of nodes creates its own version of the blockchain. For instance, the group refuses to build on any chain that contains a block with the unwanted transaction. As long as this group has the longest fork path [18], all other honest miners may have a chance to include the forked chain in the main chain.
- **Bribery Attacks:** Bribe miners to manipulate blockchain consensus or transaction order [19], leading to attacks such as double spending, historical alterations, transaction delays, etc.
- **Mining Difficulty Adjustment:** Manipulating the rate of block creation.

2) Unfair Sequencing: The attackers alter the sequencing of transactions to gain a favorable price. Or, they censor and obstruct the transactions of other users.

C. Smart Contract Layer (SC)

Smart contracts are programs stored on blockchains to execute automatically when predefined conditions are satisfied. These conditions are typically based on the terms of an agreement or a specific workflow. In essence, smart contracts serve to eliminate the need for third parties or intermediaries in transactions.

The smart contract layer allows for the creation and deployment of smart contracts, enabling transaction-level atomic state transitions and storing various states of the blockchain system, such as users' cryptocurrency balances, cryptocurrency addresses, block numbers, and other relevant information. The most popular programming language for writing smart contracts is Solidity.

1) Untrusted Or Unsafe Calls: Smart contracts typically use external calls to interact with other smart contracts, external data sources, or users to achieve more complex functionality. However, this provides attackers with opportunities to embed malicious code logic.

- **Reentrancy:** Repeatedly calling a function within a smart contract before the initial function call is completed.
- **Delegate Call / Code Injection:** Delegate call is a low-level mechanism in Solidity that permits one contract to execute the code of another contract within the current context. This implies the data and state are shared with the called contract, potentially allowing attackers to bypass authorization checks or modify the state.

2) Coding Mistake: This refers to technical errors made by developers during the coding process, resulting in incorrect smart contract behavior, as opposed to design issues.

3) Access Control Mistake: Attackers utilize external calls, fallback functions, forged wallet addresses, etc. to bypass identity verification. Smart contract developers must ensure that resource access is properly authenticated by any means.

4) State Transition Design Mistakes: Each opcode supported by the Ethereum Virtual Machine (EVM) is associated with a specific gas cost. These gas costs are designed to reflect the underlying resources consumed by each operation on the nodes that constitute the Ethereum network. A mismatch between the cost of an operation and the actual resource consumption (such as CPU time and memory) has several implications. It can be exploited for malicious purposes such as excessive block processing times and skewed block gas limits [20].

D. Protocol Layer (PRO)

The protocol layer is a collection of standardized DeFi protocols that define basic rules to manage cryptocurrency. For instance, ERC-20 establishes the fundamental functionalities of cryptographic tokens, including token transfers, balance inquiries, and approvals. This means that despite belonging to different DeFi projects, all ERC-20-compliant tokens are compatible with each other, which promotes interoperability between DeFi protocols.

1) Transaction Order Dependency Mistake: An agent can benefit from front-running by gaining early knowledge of pending transactions and strategically setting a higher gas fee to prioritize their own transaction execution before the victim's transaction. This unfair practice allows the agent to

obtain more favorable outcomes while causing losses to other participants. Similarly, in back-running or sandwiching, an agent intentionally places its own transaction orders around the target transaction to manipulate the price.

2) Replayable Design: Replicating a transaction executed on one blockchain to another blockchain. This attack is commonly observed during cryptocurrency forks or when transactions are not specific to a particular chain.

3) Block State Dependency Mistake: Block states such as `block.blockhash` or `block.timestamp` can be manipulated by malicious miners, so a contract shouldn't use block states for decision-making. Neither should they use block states as random seeds [13].

4) Permissionless Interaction: Victims only constrain the function interface of a contract, without considering how the contract is implemented. As a result, attackers exploit contracts that comply with the accepted ABI interface but contain incompatible implementation logic, causing harm [5].

5) Unsafe Dependency: The reliance on external protocols or some standards provides attackers with more attack vectors. For example, attackers can exploit the quotes provided by an oracle to manipulate prices. They can also acquire a sufficient amount of governance tokens to propose and execute malicious contract code.

6) Unfair or Unsafe Interaction: Participants can obtain more favorable prices via unfair slippage protection or unfair liquidity. Unsafe or infinite token approval refers to a protocol granting approval for an unlimited or excessively high number of tokens to another address or contract.

E. Auxiliary Service (AUX)

The functionalities that do not belong to the first four layers are classified as the Auxiliary Service Layer, including website management, code/contract deployment, etc. These services typically do not directly participate in the operation of the blockchain system, but they provide necessary support and management to make the blockchain system more user-friendly. In general, vulnerabilities in the auxiliary service layer are usually caused by backdoors, phishing, or trust in malicious data sources.

III. Analysis of Attacks and Incidents

In this section, we introduce the attack events proposed by [6]. Then, we apply these classification systems to the investigated incidents to better understand the characteristics of these incidents.

A. Analysis of Attacks

1) Utilization of Flash Loan: Flash loans typically refer to the process of borrowing and repaying funds within a single block, with various transactions intertwined through smart contracts. The borrower first obtains a loan amount by pledging a certain amount of cryptocurrency assets and then using rapid trading, repayment, and other operations to achieve borrowing and repayment of funds, thereby achieving arbitrage.

2) Private Key Leakage: Attackers either steal or exploit the accidental leakage of private keys from the project team, enabling them to gain unauthorized access to deploy and manage the smart contracts. With these permissions, they have the ability to arbitrarily mint and transfer tokens.

3) Reentry Attack: Attackers insert malicious code within the "fallback" or other external functions, exploiting the reentrancy vulnerability inherent in such functions. By repeatedly calling functions, the attackers can execute the malicious code multiple times, bypassing the contract's intended logic and control flow.

4) Arithmetic Bug: Arithmetic bugs in DeFi applications arise from flaws in mathematical operations and calculations. These bugs can result in inaccurate balances, exchange rates, or reward calculations, leading to financial losses or overpayments.

5) Others: Oracle attacks exploit price manipulation, leading to the reception of incorrect price information by the contract. Phishing attacks are executed by injecting malicious scripts to deceive users into performing wallet operations. In addition to the aforementioned attack categories, there exist various other attack patterns that deserve comprehensive investigation.

B. Data Source

The investigation scope of this paper is limited to DeFi incidents that occurred from June 2022 to May 2023, involving direct or indirect losses of 1 million USD or more. The incident data sources[1] primarily rely on (i) Rekt News; (ii) DeFiHackLabs; (iii) Slowmist, and official post-mortem reports. To the best of our knowledge, these data sources provide timely and comprehensive coverage of the reported incidents, ensuring that significant events with losses exceeding 1 million USD are not overlooked.

This paper focuses solely on DeFi incidents. Any incidents involving CeFis (e.g. FTX, Binance), DAOs, or NFTs will not be included in the scope of this paper.

C. Analysis of Incidents

The final result in Table II comprises 35 events, with a total loss exceeding 950 million USD. The table provides information about the loss amount, incident type, whether they underwent professional auditing, occurrence date, and a citation to detailed post-mortem.

Table II reveals that out of 35 incidents, 33 victims had professional audits in place; however, this did not prevent these attacks from occurring. This underscores the need for a more thorough examination of auditing practices and a reevaluation of the factors contributing to the vulnerabilities. Upon an in-depth investigation of 35 case studies, we identified that some incidents stem from the following human errors:

- **Leave Audited Risks Unresolved:** Quantstamp audit suggested Nomad Bridge validate the `_leaf` input of

[1]Links: (i) https://rekt.news/; (ii) https://github.com/SunWeb3Sec/DeFiHackLabs/; and (iii) https://www.slowmist.com/

TABLE II
DEFI INCIDENTS THAT LOST OVER 1 MILLION USD OCCURRED FROM JUNE 2022 TO MAY 2023.

Project	Loss	Incident Type	Attack Event	Date	Audit	Post-Mortem
Jimbos Protocol	7.5M	Unfair slippage protection	Flash loan	May 29, 2023		[21]
Swaprum	3.0M	Authority control or breach of promise	Rug pull	May 18, 2023	✓	[22]
Level Finance	1.1M	Absence of coding logic or sanity check	Flash loan	May 02, 2023	✓	[23]
0vix	2.0M	On-chain oracle manipulation	Flash loan, Oracle attack	Apr 28, 2023	✓	[24]
Merlin DEX	1.8M	Authority control or breach of promise	Rug pull	Apr 26, 2023	✓	[25]
Hundred Finance	7.4M	Absence of coding logic or sanity check	Flash loan	Apr 15, 2023	✓	[26]
Yearn	11.6M	Absence of coding logic or sanity check	Attacks related to contract	Apr 13, 2023	✓	[27]
Sushi	3.3M	Visibility error and unrestricted action	Attacks related to contract	Apr 09, 2023	✓	[28]
SafeMoon	8.9M	Visibility error and unrestricted action	Attacks related to contract	Mar 28, 2023	✓	[29]
Kokomo Finance	4.0M	Direct call to untrusted contract	Attacks related to contract	Mar 27, 2023	✓	[30]
Euler Finance	197.0M	Absence of coding logic or sanity check	Flash loan	Mar 13, 2023	✓	[31]
Hedera	12.2M	Inconsistent access control	Attacks related to contract	Mar 09, 2023	✓	[32]
Hope Finance	1.9M	Deployment mistake	Rug pull	Feb 20, 2023	✓	[33]
Dexible	2.0M	Direct call to untrusted contract	Attacks related to contract	Feb 17, 2023	✓	[34]
Platypus Finance	8.5M	Absence of coding logic or sanity check	Flash loan	Feb 16, 2023	✓	[35]
dForce Network	3.6M	Reentrancy	Flash loan, Reentrancy	Feb 09, 2023	✓	[36]
Orion Protocol	3.0M	Reentrancy	Flash loan, Reentrancy	Feb 04, 2023	✓	[37]
Rubic	1.5M	Direct call to untrusted contract	Attacks related to contract	Dec 25, 2022	✓	[38]
Raydium	4.4M	Compromised private key / wallet	Private key leakage	Dec 16, 2022	✓	[39]
Lodestar Finance	6.5M	On-chain oracle manipulation	Oracle attack	Dec 10, 2022	✓	[40]
DFXFinance	4.0M	Reentrancy	Flash loan, Reentrancy	Nov 10, 2022	✓	[41]
Skyward Finance	3.2M	Visibility error and unrestricted action	Attacks related to contract	Nov 02, 2022		[42]
Team Finance	15.8M	Inconsistent access control	Attacks related to contract	Oct 27, 2022	✓	[43]
Mango Markets	115.0M	External market manipulation	Oracle attack	Oct 12, 2022	✓	[44]
Transit Swap	21.0M	Visibility error and unrestricted action	Attacks related to contract	Oct 02, 2022	✓	[45]
Wintermute	162.0M	Randomness	Attacks related to contract	Sep 20, 2022	✓	[46]
Acala Network	1.6M	Arithmetic mistakes	Attacks related to contract	Aug 14, 2022	✓	[47]
Nomad Bridge	190.0M	Absence of coding logic or sanity check	Attacks related to contract	Aug 02, 2022	✓	[48]
Reaper.Farm	1.7M	Inconsistent access control	Attacks related to contract	Aug 01, 2022	✓	[49]
Nirvana Finance	3.5M	Liquidity borrow, purchase, mint, deposit	Flash loan	Jul 29, 2022	✓	[50]
Crema Finance	8.8M	Visibility error and unrestricted action	Attacks related to contract	Jul 03, 2022	✓	[51]
Harmony Bridge	100.0M	Compromised private key / wallet	Private key leakage	Jun 24, 2022	✓	[52]
Inverse Finance	5.8M	On-chain oracle manipulation	Flash loan, Oracle attack	Jun 16, 2022	✓	[53]
Gym Network	2.1M	Visibility error and unrestricted action	Attacks related to contract	Jun 08, 2022	✓	[54]
Wintermute	27.6M	Transaction / strategy replay	Attacks related to contract	Jun 05, 2022	✓	[55]

The Amount column is expressed in millions (M) of US dollars. The Incident Type column indicates the type proposed by Table I.

the `Replica.sol:prove`, with QSP-19 Proving With An Empty Leaf. But the Nomad team seemed to misunderstand the issue an leave it unresolved.

- **Deploy New Code Without Audit:** Gym Network releases new features without being extensively audited. Dexible only had their experienced engineers review new contracts.
- **Partially Audit:** Kokomo Finance's audit report only covered the token contract, rather than the protocol at large. Euler Finance introduced vulnerable code `EToken.sol:donateToReserve` [31], however, Omniscia only performed an audit of the Chainlink integration component.
- **Use Unsafe Vanity Address:** Wintermute used the Profanity tool to generate addresses with multiple leading zeros. The private keys were compromised by brute force.
- **Rug Pull:** A member of Hope Finance deployed a fake router and deceived the other three owners into approving a multi-signature wallet, thereby siphoning off the funds. Merlin DEX directly inserted a backdoor into the contract. Certik did indeed raise this trust issue in their audit report, but Certik marked it as resolved without the code being genuinely fixed.

The practical value of an audit becomes limited when a project is unable to effectively prevent human errors. This highlights the need for rigorous processes to prevent human errors and oversights. The next section will propose some prevention methods.

D. Analysis of Layers, Loss, and Occurrences

TABLE III
LOSSES AND OCCURRENCE OF LAYERS

Layer	Loss	Count	Loss / Count
NET	0	0	-
CON	0	0	-
SC	512.3M	22	23.3M
PRO	214.9M	7	30.7M
AUX	226.0M	6	37.7M
Total	953.3M	35	27.2M

Table III presents the losses, occurrence frequencies, and average losses of individual layer attack events. It is noteworthy that neither NET Layer nor CON Layer was involved in the 35 incidents. This observation suggests a higher level of security in these layers, making it challenging for attackers to target them. The most common incident causes belong to SC Layer, accounting for 22 out of 35 cases (62%).

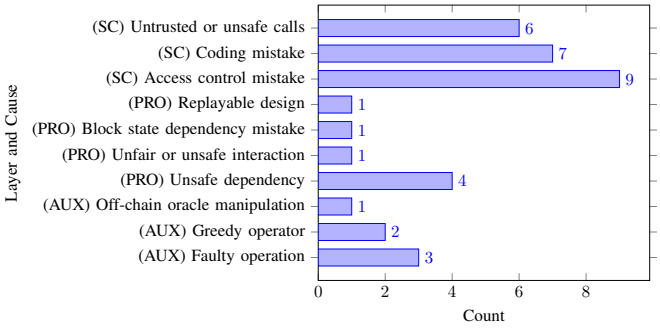

Fig. 1. Occurrences of Incident Causes

Figure 1 shows the frequency of incident causes. In SC Layer, Access Control Mistake is the most common incident cause. Most of the victims deployed flawed authentication logic when updating new contracts. In PRO Layer, Unsafe Dependency is the most common incident cause, which implies that DeFi projects should not blindly trust external data sources, such as oracles. In AUX Layer, Faulty Operation and Greedy Operation are common causes. Preventing the leakage of private keys and guarding against rug pulls are critical aspects of security in this context.

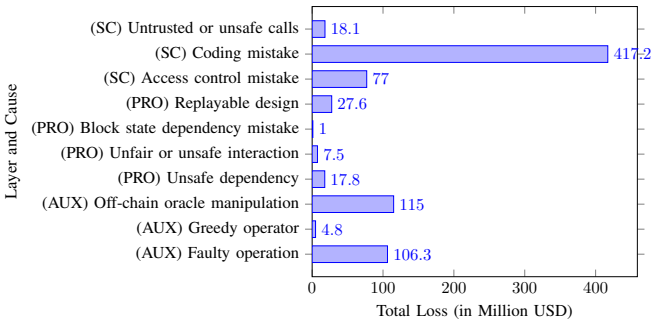

Fig. 2. Total Loss of Incident Causes

Figure 2 shows that the losses incurred due to Coding Mistake in SC Layer significantly outweigh those caused by other factors. The losses in the AUX Layer are also substantial. The losses incurred by these vulnerabilities are exceptionally costly. Even a single occurrence of such a vulnerability event could result in the flourishing project's bankruptcy.

IV. Threat Mitigation Strategies

A. Audit Services

During the auditing process, all pertinent information, including the whitepaper, business requirements, technical specifications, and so forth, must be shared with auditors. This facilitates a comprehensive understanding of the smart contracts by the auditing firms. Incorporating the assessments of multiple distinct audit firms can reduce the chances of vulnerabilities undetected.

B. Code Analyzers

SmartBugs [56] integrates 19 open-source static code analyzers. However, static code analysis lacks a concise and coherent overview, multiple-repository applications support, and standardized third-party framework integration [57]. Many of these scanning tools have not received updates for years, potentially leading to inadequate support for new vulnerabilities.

Within the SmartBugs framework, we selectively employed four tools—Mythril, Manticore, Slither, and Solhint—to conduct scans on contracts of Dexible, dForce, Euler Finance, Platypus, and others. The primary aim was to assess the tools' capability to identify the vulnerabilities. Regrettably, the outcome revealed that none of the tools successfully detected the vulnerabilities related to the incidents in this study. Hence, there is still substantial room for improvement in existing code analysis tools.

C. Human Error Protections

- **Full Audit Coverage:** All smart contracts must undergo rigorous auditing, not solely those considered critical by the development team. If possible, third-party libraries should also undergo auditing. Furthermore, a re-audit should be conducted before deploying any new contract. Lastly, all risks mentioned in the audit report should be unambiguously clarified in their meaning and necessary corrections made based on the recommendations of the audit team.
- **Multisignature Wallet:** Multisig wallets typically require signatures from multiple independent parties to ensure that any significant financial transactions necessitate unanimous consent, thereby making rug pulls more challenging, as no single entity can unilaterally control or compromise the funds. Unless the entire team has premeditated rug pulls.

D. Oracle Manipulation Protections

- **M-out-of-N reporters:** Multiple oracles provide redundancy, ensuring alternative price references when one oracle's data is erroneous or manipulated. This design forces attackers to incur higher costs and complexity when attempting an attack, reducing the probability of successful manipulation. Although it cannot eliminate risks, multiple oracles can lower the chances of being targeted.
- **Oracle Performance Monitor:** Regularly verifying incoming data instead of always assuming that oracles operate normally. For example, creating a script that periodically compares the price provided by the oracle with values from other data sources, such as another oracle, and checks for significant discrepancies. If the oracle's behavior becomes suspicious, consider temporarily pausing the smart contract of the protocol or temporarily switching to another oracle.

V. Conslusion

This paper employs the 5-layer framework to summarize 35 real-world DeFi incidents from June 2022 to May 2023, with each loss exceeding one million US dollars. The results revealed that while the majority of DeFi projects undergo professional audits, certain key issues, such as human error and oracle manipulation, still lead to security incidents. This underscores the need for DeFi projects to maintain vigilance throughout their operational phases. The future development of blockchain security may involve the establishment of operational standards. Furthermore, enhancing the collaboration model between DeFi developers and auditors holds the potential to elevate the reliability of audits.

References

[1] A. A. Monrat, O. Schelén, and K. Andersson, "Survey of blockchain from the perspectives of applications, challenges and opportunities," *IEEE Access*, vol. PP, pp. 1–1, 08 2019.

[2] S. Borisov, "DeFi – Potential, Advantages and Challenges," *Economic Studies journal*, no. 4, pp. 33–54, 2022. [Online]. Available: https://ideas.repec.org/a/bas/econst/y2022i4p33-54.html

[3] "DefiLlama," https://defillama.com/, 2023, [Accessed 22-07-2023].

[4] E. Nicolle and S. Shukla, "Defi lender euler finance hit by $197 million hack, experts say," *Bloomberg*, 03 2023. [Online]. Available: https://www.bloomberg.com/news/articles/2023-03-13/defi-s-euler-finance-hit-by-197-million-hack-experts-say

[5] L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, and A. Gervais, "Sok: Decentralized finance (defi) attacks," 2023.

[6] W. Li, J. Bu, X. Li, and X. Chen, "Security analysis of defi: Vulnerabilities, attacks and advances," 2022.

[7] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17318332

[8] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, and H.-N. Lee, "Systematic review of security vulnerabilities in ethereum blockchain smart contract," *IEEE Access*, vol. 10, pp. 6605–6621, 2022.

[9] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, 2019.

[10] R. Braden, *RFC 1122 Requirements for Internet Hosts - Communication Layers*, Internet Engineering Task Force, October 1989. [Online]. Available: http://tools.ietf.org/html/rfc1122

[11] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?" 2021.

[12] K. Wüst and A. Gervais, "Ethereum eclipse attacks," ETH Zürich, Zürich, Report, 2016.

[13] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks and defenses," 2019.

[14] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Express*, vol. 6, no. 2, pp. 93–97, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240595951930164X

[15] F. A. Aponte-Novoa, A. L. S. Orozco, R. Villanueva-Polanco, and P. Wightman, "The 51

[16] J. Lovejoy, "An empirical analysis of chain reorganizations and double-spend attacks on proof-of-work cryptocurrencies," *MIT DSpace*, 2020. [Online]. Available: https://dspace.mit.edu/handle/1721.1/127476

[17] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," 2013.

[18] A. Magnani, L. Calderoni, and P. Palmieri, "Feather forking as a positive force: Incentivising green energy production in a blockchain-based smart grid," in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, ser. CryBlock'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 99–104. [Online]. Available: https://doi.org/10.1145/3211933.3211951

[19] X. Sun, "Bribes to miners: Evidence from ethereum," 2022.

[20] "EIP-1884: Repricing for trie-size-dependent opcodes — eips.ethereum.org," https://eips.ethereum.org/EIPS/eip-1884, Mar 2019, [Accessed 20-09-2023].

[21] "Decoding Jimbo's Protocol $7.5M Exploit — QuillAudits — quillaudits.medium.com," https://quillaudits.medium.com/decoding-jimbos-protocol-7-5m-exploit-quillaudits-772ad1db6c07, Jun 2023, [Accessed 14-10-2023].

[22] "Decoding Swaprum Finance $3 Million Rug Pull — QuillAudits — quillaudits.medium.com," https://quillaudits.medium.com/decoding-swaprum-finance-3-million-rug-pull-quillaudits-2c6f9527b589, May 2023, [Accessed 14-10-2023].

[23] "Rekt - Level Finance - REKT — rekt.news," https://rekt.news/level-finance-rekt/, May 2023, [Accessed 14-10-2023].

[24] "Decoding Ovix Protocol's $2 Million Exploit — QuillAudits — quillaudits.medium.com," https://quillaudits.medium.com/decoding-ovix-protocols-2-million-exploit-quillaudits-92befc250e7c, Apr 2023, [Accessed 14-10-2023].

[25] "Rekt - Merlin DEX - REKT — rekt.news," https://rekt.news/merlin-dex-rekt/, Apr 2023, [Accessed 14-10-2023].

[26] "Rekt - Hundred Finance - REKT 2 — rekt.news," https://rekt.news/hundred-rekt2/, Apr 2023, [Accessed 14-10-2023].

[27] "Rekt - Yearn - REKT 2 — rekt.news," https://rekt.news/yearn2-rekt/, Apr 2023, [Accessed 14-10-2023].

[28] "Rekt - SushiSwap - REKT — rekt.news," https://rekt.news/sushi-yoink-rekt/, Apr 2023, [Accessed 14-10-2023].

[29] "Rekt - Safemoon - REKT — rekt.news," https://rekt.news/safemoon-rekt/, Mar 2023, [Accessed 14-10-2023].

[30] "Rekt - Kokomo Finance - REKT — rekt.news," https://rekt.news/kokomo-finance-rekt/, Mar 2023, [Accessed 14-10-2023].

[31] "Euler Finance Incident Post-Mortem — omniscia.io," https://medium.com/@omniscia.io/euler-finance-incident-post-mortem-1ce077c28454, Mar 2023, [Accessed 14-10-2023].

[32] A. P. Joe Blanchard, "Analysis & Remediation of the Precompile Attack on the Hedera Network — Hedera — hedera.com," https://hedera.com/blog/analysis-remediation-of-the-precompile-attack-on-the-hedera-network, Mar 2023, [Accessed 14-10-2023].

[33] "Rekt - Hope Finance - REKT — rekt.news," https://rekt.news/hope-finance-rekt/, Feb 2023, [Accessed 14-10-2023].

[34] "Rekt - Dexible - REKT — rekt.news," https://rekt.news/dexible-rekt/, Feb 2023, [Accessed 14-10-2023].

[35] "Rekt - Platypus Finance - REKT — rekt.news," https://rekt.news/platypus-finance-rekt/, Feb 2023, [Accessed 14-10-2023].

[36] "Rekt - dForce Network - REKT — rekt.news," https://rekt.news/dforce-network-rekt/, Feb 2023, [Accessed 14-10-2023].

[37] "Rekt - Orion Protocol - REKT — rekt.news," https://rekt.news/orion-protocol-rekt/, Feb 2023, [Accessed 14-10-2023].

[38] Neptune Mutual, "How Was Rubic Protocol Hacked? — neptunemutual.com," https://neptunemutual.com/blog/how-was-rubic-protocol-hacked/, Dec 2022, [Accessed 14-10-2023].

[39] "Rekt - Raydium - REKT — rekt.news," https://rekt.news/raydium-rekt/, Dec 2022, [Accessed 14-10-2023].

[40] Waffle, "Post Mortem Summary — blog.lodestarfinance.io," https://blog.lodestarfinance.io/post-mortem-summary-13f5fe0bb336, Dec 2022, [Accessed 14-10-2023].

[41] "V2 Vulnerability Post Mortem — dfxfinance," https://medium.com/@dfxfinance/v2-vulnerability-post-mortem-b05232bc6550, Nov 2022, [Accessed 15-10-2023].

[42] Neptune Mutual, "Decoding Skyward Finance Smart Contract Vulnerability — medium.com," https://medium.com/neptune-mutual/decoding-skyward-finance-smart-contract-vulnerability-3e38c5d0e312, Nov 2022, [Accessed 15-10-2023].

[43] "Rekt - Team Finance - REKT — rekt.news," https://rekt.news/teamfinance-rekt/, Oct 2022, [Accessed 15-10-2023].

[44] "Rekt - Mango Markets - REKT — rekt.news," https://rekt.news/mango-markets-rekt/, Oct 2022, [Accessed 15-10-2023].

[45] "Rekt - Transit Swap - REKT — rekt.news," https://rekt.news/transit-swap-rekt/, Oct 2022, [Accessed 15-10-2023].

[46] "Rekt - Wintermute - REKT 2 — rekt.news," https://rekt.news/wintermute-rekt-2/, Sep 2022, [Accessed 15-10-2023].

[47] "Rekt - Acala Network - REKT — rekt.news," https://rekt.news/acala-network-rekt/, Aug 2022, [Accessed 15-10-2023].

[48] Sm4rty, "Nomad Bridge's $200 Million Exploit Postmortem — sm4rty.medium.com," https://sm4rty.medium.com/nomad-bridges-200-million-exploit-postmortem-9d1cd83db1f7, Aug 2022, [Accessed 15-10-2023].
[49] Gabriel Sieng, "Reaper Farm Just Lost US$1.7 Million From Exploit - ChainDebrief — chaindebrief.com," https://chaindebrief.com/reaper-farm-got-hacked/, Aug 2022, [Accessed 15-10-2023].
[50] "Nirvana Finance Incident Analysis — certik.com," https://www.certik.com/resources/blog/1UBzEHHu35dJdJOGsuf85D-nirvana-finance-incident-analysis, Jul 2022, [Accessed 15-10-2023].
[51] "Crema Finance Exploit — certik.com," https://www.certik.com/resources/blog/4XzSJEeWC2bRppR9CeBckw-crema-finance-exploit, Jul 2022, [Accessed 15-10-2023].
[52] "Harmony Incident Analysis — certik.com," https://www.certik.com/resources/blog/2QRuMEEZAWHx0f16kz43uC-harmony-incident-analysis, Jun 2022, [Accessed 15-10-2023].
[53] "Inverse Finance Incident Analysis — certik.com," https://www.certik.com/resources/blog/6LbL57WA3iMNm8zd7q111R-inverse-finance-incident-analysis, Jun 2022, [Accessed 15-10-2023].
[54] "Rekt - Gym Network - REKT — rekt.news," https://rekt.news/gymnet-rekt/, Jun 2022, [Accessed 15-10-2023].
[55] "Rekt - Wintermute - REKT — rekt.news," https://rekt.news/wintermute-rekt/, Jun 2022, [Accessed 15-10-2023].
[56] J. F. Ferreira, P. Cruz, T. Durieux, and R. Abreu, "Smartbugs: A framework to analyze solidity smart contracts," in *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, 2020, pp. 1349–1352.
[57] A. Walker, M. Coffey, P. Tisnovsky, and T. Černý, "On limitations of modern static analysis tools," in *Information Science and Applications*. Springer Berlin, Heidelberg, 01 2020, pp. 577–586.