

Security Analysis of DeFi: Vulnerabilities, Attacks and Advances

Wenkai Li*, Jiuyang Bu*, Xiaoqi Li†, Xianyi Chen

School of Cyberspace Security, Hainan University, Haikou, China

Email: liwenkai871@gmail.com, 736011577@qq.com, csxqli@gmail.com, 1550615836@qq.com

Abstract—Decentralized finance (DeFi) in Ethereum is a financial ecosystem built on the blockchain that has locked over 200 billion USD until April 2022. All transaction information is transparent and open when transacting through the DeFi protocol, which has led to a series of attacks. Several studies have attempted to optimize it from both economic and technical perspectives. However, few works analyze the vulnerabilities and optimizations of the entire DeFi system. In this paper, we first systematically analyze vulnerabilities related to DeFi in Ethereum at several levels, then we investigate real-world attacks. Finally, we summarize the achievements of DeFi optimization and provide some future directions.

Index Terms—Smart contract, Ethereum, Decentralized finance, DeFi

I. INTRODUCTION

The popularity of blockchain 2.0 technology has resulted in a wide range of related services. Decentralized finance (DeFi) is an example of a financial service built on blockchains to provide transaction transparency. From January 2020 to April 2022, the total value locked in DeFi climbs from 600 million USD to around 200 billion USD [1]. However, there was a sharp drop in May 2022, which caused us to ponder the safety of DeFi system.

Attacks have emerged gradually with the rapid development of DeFi. Security incidents against DeFi continue to proliferate, and there has been a lot of research to improve the security of blockchain [2]–[11]. [4] presented BLOCKEYE, a real-time threat detection solution for Ethereum-based DeFi deployments. [6] presented an online framework SODA to identify assaults on smart contracts. However, none of them consider economic security beyond detecting vulnerabilities against technical aspects. [7] distinguished between technological and economic security and demonstrated that economic security is not flawless. [8] investigated the scope of loan marketplaces and assess the risk of lending agreements. [9] investigated how errors in design and pricing volatility of DeFi protocols might lead to DeFi crises. [10] analyzed the differences between centralized finance (CeFi) and DeFi, covering legislation, economy, security, privacy, and market manipulation.

However, DeFi is not secure enough that attacks on it keep appearing, such as the Ronin Bridge incident [12]. It is shown in [11] that the existing defenses do not reduce the

number of attacks. Therefore, the security against DeFi needs to strengthen.

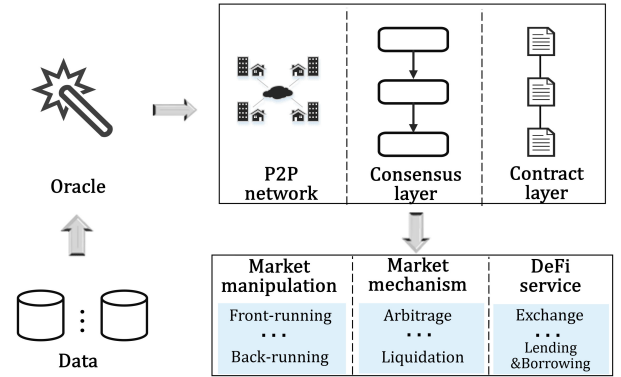


Fig. 1: Overview of the Analysis Path

Unfortunately, there is still a lack of systematic analysis of DeFi system security. To assist in subsequent studies, our research analyzes the technical and economic risks to which DeFi is vulnerable at the system level and comprehensively follows the path in Fig. 1. We also summarize existing real-world attacks in a way that provides a good foundation for future research, then we summarize classical related protection techniques. Finally, we provide an outlook on the possible improvements that need to strengthen in this area.

The main contributions of this paper are as follows:

- (1) To the best of our knowledge, we conduct the *first* systematic examination on the security issues of the DeFi ecosystem built on blockchain.
- (2) We systematically summarize the vulnerabilities of the Ethereum-based DeFi system, investigate real-world attack events related to DeFi and classify them according to their vulnerability principles.
- (3) We survey the security optimizations in DeFi from the system level and give some suggestions for future research directions in this area.

II. BACKGROUND

ETHEREUM: Ethereum is a public blockchain system that uses the Turing-complete programming language Solidity and Vyper, etc to develop smart contracts [13]–[15]. Anyone can deploy decentralized applications (dapps) on the Ethereum chain that can communicate with others, and the most popular

* Wenkai Li and Jiuyang Bu are co-first authors.

† The corresponding author.

application in the financial field is DeFi, which provides a wide range of financial services.

GAS: To avoid overuse of network resources, all transactions on Ethereum are paid a cost, and the total gas cost equals the amounts of gas multiplied by `gasPrice` [16], [17]. The user who proposes transactions sets the `gasPrice`, and the transaction is conducted earlier if the `gasPrice` is high.

MINER EXTRACTABLE VALUE (MEV): It refers to the profit miners make by performing a series of operations on the blocks they mine [18], such as transaction inclusion, exclusion, and reordering. Miners reorder transactions to optimize the initial ordering of transactions. Earning additional ordering optimization (OO) fees [19] is also a source of MEV.

DEVELOPMENT OF DeFi: The introduction of blockchain technology [20] has changed the traditional financial ecosystem. With the advent of Ethereum, smart contracts became the basis for the development and implementation of DeFi. Since the landing of MAKERDAO in 2014 which is the first Ethereum-based DeFi project, several DeFi protocols have emerged to implement functions of traditional CeFi, such as lending platforms, exchanges, derivatives, and margin trading systems [21]. As liquidity mining is mentioned in 2020, DeFi is pushed into high gear with the emergence of decentralized exchanges such as COMPOUND, which are entirely managed by smart contracts. MONEY LEGOS brings unlimited creativity to DeFi products. It means that a new financial product can be realized by combining the underlying DeFi protocols [22]. In 2022, regulated decentralized finance (rDeFi) becomes the new trend in DeFi development [23].

III. ANALYSIS OF VULNERABILITIES

To summarize threats in decentralized finance, we focus on data, consensus, contract, and application layers.

1) Data Security Vulnerabilities:

Oracle Mechanism Vulnerabilities: The oracle is an automated service mechanism that allows the system to obtain off-chain asset data as input [7]. However, as Fig. 1 shows, the risk to oracle grows drastically when a single point of failure occurs. For example, over 3 million sETH were arbitrated (Address: `0x93819f6...`) due to oracle errors in SYNTHETIX, a protocol that converts entity into synthetic [24].

Inappropriate Key Management: In the DeFi ecosystem, wallets are used to manage private keys, and authentication is based on keys in most cases. However, even the safest cryptocurrency hardware wallets have security issues [25] caused by the architecture design. For example, the Ronin Bridge was hacked for 624 million USD, which occurred in March 2022 [12]. The hackers used a backdoor attack to get the signatures of a third-party validator and four other local verifier signatures for stealing.

2) Consensus Mechanism Vulnerabilities:

Certain malevolent activity leverages the rules of consensus to influence the sequences of transactions. There are a variety of attacks combined with MEV, such as flash loans [26], [27], sandwich attacks [18], [28], and forking attacks [19].

Transaction Order Vulnerability: It will be used to describe this phenomenon in which an attacker alters the initial sequence of transactions by leveraging the miner's desire for profit. The sandwich attack is a typical example, the attacker anticipates that the victim will buy asset A, and pays a higher gas fee to miners to acquire it before the victim at a lower price. And then sells A at a higher price for arbitrage since the victim's purchase boosts the price [28].

Forking Vulnerability: Forking events in DeFi are generally associated with transaction fee-based forks and time-bandit attacks [19]. Mining revenue incentivizes miners to perform honestly, but the OO fee motivates them to reorder transactions in the block, enhancing the income.

3) Smart Contract Vulnerabilities:

There are 20 types of smart contract vulnerabilities in Ethereum defined in [29], of which Table I shows the security weaknesses that attackers might use to make a profit. [30] detected over 18K real-world smart contracts and achieved an average coverage rate of 92 percent above the average of six vulnerabilities that can be categorized into the three types as detailed below:

Suicidal and Greedy Contracts: Smart contracts usually include a provision enabling the owner to commit suicide if the contract is challenged. This suicide procedure can be carried out for any cause under the suicidal contract [31], [32]. Greedy contracts do not have functions related to the extraction [31]. The contract locks all ether and cannot withdraw. Therefore, making sure there are means to get ether out before transferring it to a contract [29].

Block Info Dependency: In Ethereum, the discrepancy between successive blocks is valid when the timestamps are within 12 minutes [33]. However, if the logic of the contract combines states in the block, the miner can control this information for profit [29]. For example, when the `block.timestamp` is used as the seed in a pseudo-random function running in a contract, miners with access to this block can replicate the process of producing random numbers to attack the contract.

Unchecked External Call: The return value or the arguments of an external call can affect the states of the code, and many contracts do not check the return value leads to vulnerabilities. Multiple functions are nested, and the external call does not check the return value of the internal call in time can go wrong. Smart contracts in the DeFi trade by using external call functions including `call()`, `send()`, `delegatecall()`. More crucially, a failed external call in these methods results in a transaction not being rolled back, which can cause logical effects.

4) Application Layer Vulnerabilities:

The vulnerabilities at the application level are based on the manipulation of prices, and we summarize the existing flaws from the following four perspectives.

Lending Market Imperfection: When the prices in the market are out of balance, it will result in bad debts for one of the participants in the lending market. To get more loans, attackers can boost the cryptocurrency exchange rate on the

TABLE I: Smart Contract Vulnerabilities in DeFi

Categories	Causes	Categories	Causes
Unchecked External Calls	Without checking return values	Reentry	Repeated calls before completed
Transaction State Dependency	Failure to check permissions	Nested Call	Unrestricted call depth
DoS Under External Influence	External exceptions inside loops	Missing Return	Denote return but no value
Unmatched ERC-20 Standard	Not follow the standard	Greedy Contracts	Receive but not withdraw Ethers
Strict Balance Equality	Balance check failed	Block Info Dependency	Status leakage
Misleading Data Location	Incorrect storage type	Missing Interrupter	No backdoor to handle crises
Transaction State Dependency	Error using tx.origin	Arithmetic Operations	Unmatched type to values

oracle by modifying the real-time price-related status before the loan is made. For example, an attacker can gain a larger quantity of tokens by directly manipulating token prices in the asset pool or increasing the price of collateral before lending [34], putting the borrower in danger of bad debt.

Cryptocurrency Instability: The large fluctuations of cryptocurrencies come from many reasons, one of which is the Pump-and-Dump. The instability can easily trigger liquidation procedures. Exchanges have chosen stablecoin, which is tied to the price of real money, as the pricing standard to minimize losses, but they still exist a risk. For example, a 99.98 % plunge on May 11, 2022, in the price of the luna coin whose value is tied to a stablecoin called Terra, leaving the entire crypto market with over 700 million in collateral liquidated [35].

Design imperfection: The attackers make use of incorrectly configured functionality or specific convenience features of DeFi platform exchanges [36]. Flash loan is designed as risk-free loans to be a convenient improvement to the loan that need to borrow the flash loan, exchange it for currency and repay the loan in an atomic transaction. For example, attackers borrow the flash loan to receive collateral at a premium and make a profit in this atomic transaction [37], this results in bad debts for the users who borrow money from attackers.

Abusive Exposure Transaction: Exchanges disclose all transactions as soon as feasible to ensure complete behavioral transparency because off-chain matching services are not automated. Unfortunately, exchanges can restrict access to select users and launch denial of service attacks [38] to dominate the market, audit transactions and even front run the orders.

IV. ANALYSIS OF ATTACK EVENTS

In this section, we investigate real-world attacks in DeFi and analyze the vulnerabilities exploited in the attacks.

1) Utilization of Flash Loan:

Flash loan is a type of unsecured lending that relies on the atomicity of blockchain transactions at the point of execution [26] and adds dynamism to DeFi. Unfortunately, attackers can exploit flaws in their existing protocols [36]. There are several other attacks caused by using the flash loan in Table II.

From Table II, the various attacks against the flash loan service have caused significant financial damage to the DeFi ecosystem. Attackers borrowed money from lending platforms, e.g, DYDX [39], with the flash loan services, then used the borrowed funds to manipulate the price of tokens [34] to make an arbitrage.

Flash loans facilitate the execution and reduce the cost of attacks. The GRIM FINANCE and POPSICLE FINANCE incidents borrowed tokens by flash loans to enable reentry attacks and double claiming attacks, respectively.

TABLE II: Attacks Related to Flash Loan

Victims	Date	Amount (million USD)
Harvest Finance	Oct 26, 2020	24
Alpha Homora	Feb 13, 2021	37
XToken	May 12, 2021	24
PancakeBunny	May 19, 2021	200
Belt Finance	May 28, 2021	50
Cream Finance	Oct 27, 2021	130
Beanstalk Farms	Apr 18, 2022	182

2) Private Key Leakage:

Ethereum-based DeFi applications need to interact with the wallet, like METAMASK, and Ethereum provides the API [40] that enables this interaction. Attackers get the private key of the original contract deployers or administrators to control the contract to mint or transfer tokens to others under their control. According to Table III, the exposure of the private key has lost hundreds of millions of dollars.

TABLE III: Attacks Related to Private Key Leakage

Victims	Date	Amount (million USD)
Meerkat Finance	Mar 04, 2021	31
Paid Network	Mar 05, 2021	160
EasyFi	Apr 19, 2021	80
bZx	Nov 05, 2021	55
Vulcan Forged	Dec 13, 2021	140
Ronin Bridge	Mar 29, 2022	624

3) Reentry Attack:

The most significant reentry attack in Ethereum was the DAO attack [41] that caused a hard fork of Ethereum. Reentry attacks were applied to the DeFi protocol with its development.

The reentry attacks that occurred on the DFORCE and GRIM FINANCE [42] platforms, together caused a loss of 54 million USD, in Table IV. The DFORCE incident was caused by the fact that the ERC-777 which is a standard for token contracts interfaces and behaviors allows transaction notifications to be sent to the recipient in the form of callbacks. This means that ERC-777 token indirectly results in the recipient having control of the execution [7].

In the GRIM FINANCE security incident, the attacker publishes a malicious contract whose callback function contains

a call to the `depositFor()` function in the GRIMBOOST-VAULT CONTRACT. `depositFor()` returns proof of investment Spirit-LP to the user. Therefore, it will call the callback function in the malicious contract again to obtain multiple Spirit-LP proofs. This allows the attacker to gain more additional revenue.

4) Arithmetic Bug:

Almost all DeFi applications involve arithmetic operations on currencies. These operations consist of adding or subtracting from account balances and converting exchange rates between different tokens [7]. Attackers typically target weaknesses in arithmetic operations. This can be seen in the case of the accuracy loss in URANIUM FINANCE incident, when checking the contract balance, the bug resulted in the final contract calculating 100 times larger than the actual balance [43] and losing 50 million USD.

TABLE IV: Attacks Related to Contract Bugs

Victims	Date	Amount (million USD)
dForce	Apr 19, 2020	24
Uranium Finance	Apr 28, 2021	50
Compound	Sep 30, 2021	80
Grim Finance	Dec 19, 2021	30

Another example of arithmetic vulnerabilities is the integer underflow of COMPOUND FINANCE (Address: `0x75442Ac...`). Its reward payouts `CompSpeed` can be set to 0, which indicates that reward payouts are suspended, and the market award index `supplyIndex` is 0. For new users, their award index `supplierIndex` initialized to `CompInitialIndex` preset by COMPOUND as 10^{36} . Parameters variation causes the formula, `deltaIndex=sub_(supplierIndex=0, supplierIndex= 10^{36})`, for calculating the difference in the reward index to overflow, while the calculation of the reward relies on the value of `deltaIndex`. There was no attacker in this security incident, but rather an overpayment of rewards due to an underflow vulnerability in the contract.

5) Other Bugs:

Attacks Related to Oracle: Oracle serves as an information channel between the DeFi and the outside world, giving external asset values as an input source to the DeFi [44]. VEE FINANCE requires that price variations in the mining pool of more than 3% be re-inputted using oracle. Because its oracle solely utilizes the prices in the mining pool as an input source, the attacker can manipulate the token price in the pool, forcing the oracle to update the price. As a result, the contract received incorrect price information, skipping the slippage protection [45] and resulting in a loss of 35 million USD for VEE FINANCE.

Phishing Attack: DeFi website embedded scripts that we can interact with the user's wallet API, which could facilitate a phishing attack [46]. The attackers used a phishing attack on BadgerDAO, causing it to lose 120 million USD. First, the attacker used the email address of the administrator to create three BADGER accounts, one of which passed official authentication. After that, the attacker accessed the BADGER

application website through this account and injected a malicious script into the website. The script intercepted web3 transactions and prompted the user to allow the attacker to manipulate the tokens in their wallet [47].

Attacks Related to Contract: The WORMHOLE incident [48] caused about 320 million USD in damage on February 3, 2022. The attacker first calls the `verify_signature()` function to obtain signatures for the function `post_vaa()`. However, the `load_instruction_at()` function called in the `verify_signatures()` function does not verify the authenticity of the account, so the account can use the obtained valid signatures to send messages to the contract. Finally, the attacker used this vulnerability to send a message casting 120,000 wETH to the contract.

Double-Claiming Attack: In Table V, the POPSICLE FINANCE event [49] was attacked similarly to the double-spending attack which creates multiple transactions using the same cryptocurrency.

TABLE V: Attacks Related to Other Bugs

Victims	Date	Amount (million USD)
Spartan Protocol	May 02, 2021	30
Popsicle Finance	Aug 03, 2021	25
Poly Network	Aug 10, 2021	26
Vee Finance	Sep 21, 2021	37
BadgerDAO	Dec 02, 2021	120
Qubit Finance	Jan 28, 2022	80
Wormhole	Feb 03, 2022	326

First, the attacker deposits funds via POPSICLE FINANCE, and the platform returns a PLP Token certificate of deposit. Then, the attacker transfers the certificate to other contracts under his control. POPSICLE FINANCE calculates the user's reward incrementally via the `_fee0Earned()` function. The rewards are accumulated even if there is no asset in the user's account. Finally, the attacker controls the contract by calling the `withdraw()` function to remove the deposited funds and rewards.

V. ANALYSIS OF SECURITY OPTIMIZATION

1) Data Security Optimization:

Oracle optimization Schemes: Due to the necessity for off-chain asset information such as pricing, as discussed in III-1, there is an expanding demand for superior oracles [50]. Our research looked at the real-world oracle optimization choices of Defi systems. The COMPOUND that aggregates pricing from off-chain to on-chain via the CHAINLINK [44], delivers multi-party data directly to the contract through reputation from providers, forming a reference pricing network where nodes in the chain may get price data to stay up to current. However, quantitative reputation cannot match the oversized price makes it can only apply on a small scale. Another form is MAKERDAO [51], which collects off-chain data through the central medianizer which is an aggregator. It utilizes the median of prices for pricing and delays price updates by one

hour before uploading on the chain so that governors and users can react to faults to secure the process.

Wallet Key Security Optimization: Users initiate a transaction and sign it using the key pair, the assets in the account are lost when the key leaks to an adversary. Some studies [52]–[54] proposed specific solutions for wallet management and wallet architecture. According to [25], existing hardware wallets migrated from the PC wallet architecture, resulting in a bad design that does not fundamentally fix the problem when just utilizing authentication and communication encryption. For interactive authentication, adds several signatures and keys to the original wallet structure, which prevents attackers from manipulating the keys for transactions using a malfunctioning wallet. Combined with software and hardware, two android applications created in [54] for a cold wallet with key storage in the form of QR codes and a hot wallet for sending transactions, respectively, provide privacy protection.

2) Smart Contract Security Optimization:

The smart contract, which is a part of the DeFi project connecting the data and the application layer, might alter the state of a transaction, and cause errors, so it's critical to improve the security of contracts.

Smart Contract Vulnerability Detection: Much research [30], [55]–[58] has been undertaken to discover contract vulnerabilities using various technological tools, such as formal verification, and machine learning. Combined with dynamic testing extends the ability of symbolic execution techniques to detect unknown vulnerabilities, thus improving the robustness of programs. Fig. 2 shows an overview of ILF [57] that combines fuzzing, machine learning, and symbolic execution.

The system used the symbolic execution for a portion of the contracts to generate transaction sequences as the training dataset for a new model consisting of GRU which is a type of neural network and a fully connected network so that the model can learn the fuzzing in the state after the symbolic execution to test contracts with high coverage.

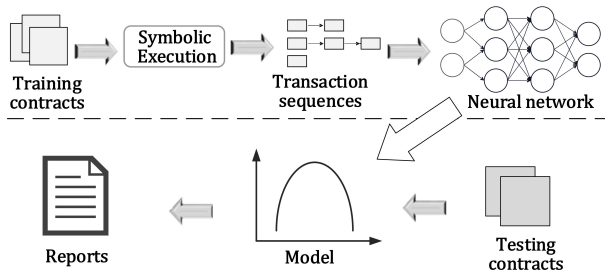


Fig. 2: Schematic Diagram of ILF Process Framework

Smart Contract Operation Regulation: However, [11] showed that contract vulnerability detection can increase contract defensibility, but assaults have not decreased, indicating that contract regulation has to be further improved. It has been studied in [4], [6], [11], [59]–[61], and we briefly introduce Sereum [61] in Fig. 3, a security tool focused on runtime monitoring and verification of reentry vulnerability.

Transaction Manager converts all control flows into conditional jump instructions in the bytecode interpreter, and then the taint engine identifies data flows in conditional jump instructions, tagging `storage` variables as key variables and writing into the lock. The attack detector detects the variables, if the modification occurs, the whole transaction is rolled back to the point where the variable was marked, which is the starting point of the entire transaction.

3) Consensus Layer Optimization:

The consensus layer and the incentive layer are interdependent, and the design of the consensus mechanism directly affects the behavior of miners, although many consensus mechanisms have been proposed, there is little regulation of the consensus and incentive levels.

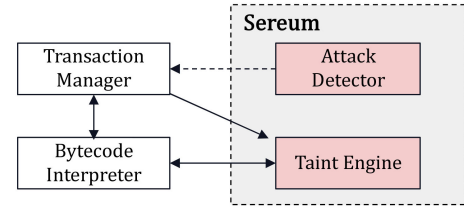


Fig. 3: Diagram of Sereum System Architecture

As described in III-2, fork attacks might affect blockchain security in terms of consensus mechanism, [27] developed DEFIPoser to monitor fork behaviors. Fig. 4 shows the process of DEFIPoser, it heuristically prunes the patches after building the DeFi graph and then does a greedy search of the negative cycle in the directed transaction flow graph, which means finding all possible profitable cycles in the trade flow graph, to detect arbitrage transactions in cyclic or more complicated scenarios. A binary search of all the paths finds the most profitable one. If it is within the quantization threshold quantified by the Markov decision process, there is an opportunity to motivate a fork attack by miners using MEV.

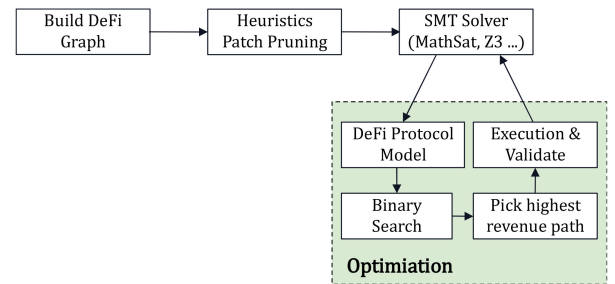


Fig. 4: Diagram of DEFIPoser Core Process

4) P2P Network Optimization:

The transactions initiated by each node in Ethereum are transmitted through P2P networks to achieve self-governing without relying on a third party; however, the lack of authentication and other features leads to a series of attacks, such as the eclipse attack [62]–[64] and sybil attack [65]. An information eclipse attack occurs when an aggressor removes nodes from a network to restrict access to information from nodes.

However, [63] suggests a series of protection methods against eclipse attacks on the Ethereum, two of which are also adopted by `geth`. When a node restarts, the client's seeding is triggered every hour, or `lookup()` is called on an empty table which stores the information in memory, but the seeding is available only if the table is empty. However, node IDs should always be inserted into the table to prevent attacks. Specifically, `geth` runs a `lookup()` on three random targets during seeding to add more legitimate nodes from the db which stores the information on disk to the table to prevent attackers from inserting their node IDs into an empty table during seeding.

5) Application Layer Optimization:

Although there is a correlation between the various layers, methods for lower levels can not fully recognize the attacks against the application layer. There still exists some research [4], [34] that makes contributions.

[4] designed BLOCKEYE divides the detection work into two phases. In Fig. 5, the first phase uses symbolic execution analysis in oracle to check whether state data streams are externally manipulated to detect vulnerable DeFi, and during the second phase, transaction monitors under the chain collect transactions to extract the features and further analysis to monitor the attack.

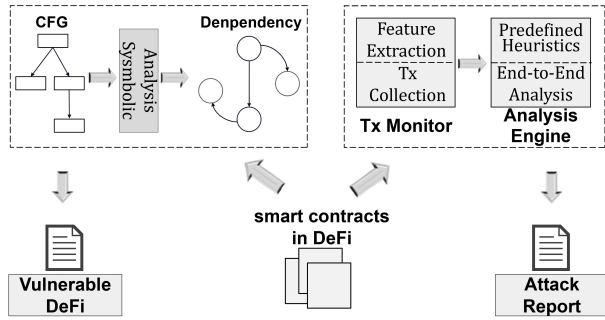


Fig. 5: Diagram of BLOCKEYE Core Process

6) Insurance Optimization:

As the DeFi market expands, the insurance on it is critical to ensuring its stability [66]. Our research divides risks in DeFi into market risks, technical risks, and credit risks. However, the damages experienced by regular users as a result of technical or credit risks are enormous, and an insurance system is required to safeguard the properties of users. It can classify as centralized and decentralized.

For example, OPYN [67], which focuses on insurance for option trading products, enables users to choose options to hedge risks based on ERC20 tokens, and the protocol is automatically performed by smart contracts for multiparty governance. SMART CONTRACT COVER, which provides smart contract insurance, is evaluated by the company NEXUS MUTUAL [66] internal assessor to determine the cost of the insurance.

VI. CONCLUSION AND FUTURE DIRECTION

The focus of this paper is on the security of DeFi, and we summarize a series of security risks of DeFi by analyzing

their projects deployed in Ethereum. For each vulnerability, we explore its causes with real-world cases. Finally, we investigate the optimization options for decentralized finance and suggest possible future directions.

Comprehensive knowledge of security and risk problems is critical to improving blockchain and establishing powerful defense capabilities in practice. There is a strong possibility to combine static detection with dynamic supervision technologies to protect DeFi at the consensus mechanism, smart contract, and application levels for the future development of DeFi application security.

REFERENCES

- [1] "Defillama," <https://defillama.com/>, 2022.
- [2] T. Chen, Z. Li, H. Zhou, J. Chen, X. Luo, X. Li, and X. Zhang, "Towards saving money in using smart contracts," in *2018 IEEE/ACM 40th international conference on software engineering: New ideas and emerging technologies results (ICSE-NIER)*, 2018, pp. 81–84.
- [3] T. Chen, X. Li, Y. Wang, J. Chen, Z. Li, X. Luo, M. H. Au, and X. Zhang, "An adaptive gas cost mechanism for ethereum to defend against under-priced dos attacks," in *International conference on information security practice and experience (ISPEC)*, 2017, pp. 3–24.
- [4] B. Wang, H. Liu, C. Liu, Z. Yang, Q. Ren, H. Zheng, and H. Lei, "Blockeye: Hunting for defi attacks on blockchain," in *Proceedings of the 43rd IEEE/ACM International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2021, pp. 17–20.
- [5] X. Li *et al.*, "Hybrid analysis of smart contracts and malicious behaviors in ethereum," 2021.
- [6] R. Cao, T. Chen, T. Li, X. Luo, G. Gu, Y. Zhang, Z. Liao, H. Zhu, G. Chen, Z. He, Y. Tang, X. Lin, and X. Zhang, "Soda: A generic online detection framework for smart contracts," in *Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*, 2020, pp. 1–17.
- [7] S. M. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. J. Knottenbelt, "Sok: Decentralized finance (defi)," *arXiv preprint arXiv:2101.08778*, 2021.
- [8] K. Qin, L. Zhou, P. Gamito, P. Jovanovic, and A. Gervais, "An empirical study of defi liquidations: Incentives, risks, and instabilities," in *Proceedings of the 21st ACM Internet Measurement Conference (IMC)*, 2021, pp. 336–350.
- [9] L. Gudgeon, D. Perez, D. Harz, B. Livshits, and A. Gervais, "The decentralized financial crisis," in *Proceedings of Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2020, pp. 1–15.
- [10] K. Qin, L. Zhou, Y. Afonin, L. Lazzaretti, and A. Gervais, "Cefi vs. defi—comparing centralized to decentralized finance," *arXiv preprint arXiv:2106.08157*, 2021.
- [11] C. Ferreira Torres, A. K. Iannillo, A. Gervais *et al.*, "The eye of horus: Spotting and analyzing attacks on ethereum smart contracts," in *Proceedings of International Conference on Financial Cryptography and Data Security (FC)*, 2021, pp. 33–52.
- [12] "Community alert: Ronin validators compromised," <https://roninblockchain.substack.com/p/community-alert-ronin-validators?s=w>, 2022.
- [13] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, pp. 841–853, 2020.
- [14] T. Chen, Z. Li, Y. Zhu, J. Chen, X. Luo, J. C.-S. Lui, X. Lin, and X. Zhang, "Understanding ethereum via graph analysis," *ACM Transactions on Internet Technology (TOIT)*, pp. 1–32, 2020.
- [15] X. Li, T. Chen, X. Luo, T. Zhang, L. Yu, and Z. Xu, "Stan: Towards describing bytecodes of smart contract," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, 2020, pp. 273–284.
- [16] T. Chen, X. Li, X. Luo, and X. Zhang, "Under-optimized smart contracts devour your money," in *Proceedings of the 24th IEEE international conference on software analysis, evolution and reengineering (SANER)*, 2017, pp. 442–446.
- [17] T. Chen, Y. Feng, Z. Li, H. Zhou, X. Luo, X. Li, X. Xiao, J. Chen, and X. Zhang, "Gaschecker: Scalable analysis for discovering gas-inefficient smart contracts," *IEEE Transactions on Emerging Topics in Computing*, pp. 1433–1448, 2020.

- [18] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?" *arXiv preprint arXiv:2101.05511*, 2021.
- [19] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *Proceedings of IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 910–927.
- [20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.
- [21] Z. Wang, K. Qin, D. V. Minh, and A. Gervais, "Speculative multipliers on defi: Quantifying on-chain leverage risks," *Financial Cryptography and Data Security*, 2022.
- [22] A.-D. Popescu *et al.*, "Decentralized finance (defi)—the lego of finance," *Social Sciences and Education Research Review*, pp. 321–349, 2020.
- [23] "Regulated decentralized finance (rdefi) is paving the way for exponential growth," <https://www.globenewswire.com/news-release/2022/05/04/2435410/0/en/Regulated-Decentralized-Finance-rDeFi-is-Paving-The-Way-for-Exponential-Growth.html>, 2022.
- [24] "Synthetix response to oracle incident," <https://blog.synthetix.io/response-to-oracle-incident/>, 2019.
- [25] A. Dabrowski, K. Pfeiffer, M. Reichel, A. Mai, E. R. Weippl, and M. Franz, "Better keep cash in your boots—hardware wallets are the new single point of failure," in *Proceedings of ACM CCS Workshop on Decentralized Finance and Security (DeFi)*, 2021, pp. 1–8.
- [26] K. Qin, L. Zhou, B. Livshits, and A. Gervais, "Attacking the defi ecosystem with flash loans for fun and profit," in *Proceedings of International Conference on Financial Cryptography and Data Security (FC)*, 2021, pp. 3–32.
- [27] L. Zhou, K. Qin, A. Cully, B. Livshits, and A. Gervais, "On the just-in-time discovery of profit-generating transactions in defi protocols," in *Proceedings of IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 919–936.
- [28] L. Zhou, K. Qin, C. F. Torres, D. V. Le, and A. Gervais, "High-frequency trading on decentralized on-chain exchanges," in *Proceedings of IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 428–445.
- [29] J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo, and T. Chen, "Defining smart contract defects on ethereum," *IEEE Transactions on Software Engineering*, pp. 327 – 345, 2020.
- [30] J. Choi, D. Kim, S. Kim, G. Grieco, A. Groce, and S. K. Cha, "Smartian: Enhancing smart contract fuzzing with static and dynamic data-flow analyses," in *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 227–239.
- [31] X. Li, T. Chen, X. Luo, and C. Wang, "Clue: towards discovering locked cryptocurrencies in ethereum," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC)*, 2021, pp. 1584–1587.
- [32] X. Li, T. Chen, X. Luo, and J. Yu, "Characterizing erasable accounts in ethereum," in *International Conference on Information Security (ISC)*, 2020, pp. 352–371.
- [33] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, pp. 1–32, 2014.
- [34] S. Wu, D. Wang, J. He, Y. Zhou, L. Wu, X. Yuan, Q. He, and K. Ren, "Defiranger: Detecting price manipulation attacks on defi applications," *arXiv preprint arXiv:2104.15068*, 2021.
- [35] "Over \$700 million liquidated as terra (luna) crashes below \$1," <https://cryptopotato.com/over-700-million-liquidated-as-terra-luna-crashes-below-1/>, 2022.
- [36] D. Wang, S. Wu, Z. Lin, L. Wu, X. Yuan, Y. Zhou, H. Wang, and K. Ren, "Towards a first step to understand flash loan and its applications in defi ecosystem," in *Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Computing (SBC)*, 2021, pp. 23–28.
- [37] "All you need to know about defi flash loans," <https://medium.com/coinmonks/all-you-need-to-know-about-defi-flash-loans-ca0ff4592d90>, 2021.
- [38] C. Baum, B. David, and T. K. Frederiksen, "P2dex: privacy-preserving decentralized cryptocurrency exchange," in *Proceedings of International Conference on Applied Cryptography and Network Security (ACNS)*, 2021, pp. 163–194.
- [39] "dydx," <https://dydx.exchange/>, 2022.
- [40] "Ethereum provider api," <https://docs.metamask.io/guide/ethereum-provider.html>, 2022.
- [41] "Critical update re: Dao vulnerability," <https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/>, 2016.
- [42] "Analysis of the grim finance hack," <https://slowmist.medium.com/analysis-of-the-grim-finance-hack-bc440108b069>, 2021.
- [43] "Slowmist: Analysis of uranium finance's hacked event," <https://slowmist.medium.com/slowmist-analysis-of-uranium-finance-s-hacked-event-9c9d11af7b2b>, 2021.
- [44] B. Liu, P. Szalachowski, and J. Zhou, "A first look into defi oracles," in *Proceedings of IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, 2021, pp. 39–48.
- [45] "The main cause of vee finance attack," <https://slowmist.medium.com/the-main-cause-of-vee-finance-attack-52fc8e5fb13d>, 2021.
- [46] P. Winter, A. H. Lorimer, P. Snyder, and B. Livshits, "What's in your wallet? privacy and security issues in web 3.0," *arXiv preprint arXiv:2109.06836*, 2021.
- [47] "Badgerdao exploit technical post mortem," <https://badger.com/technical-post-mortem>, 2021.
- [48] "Wormhole bridge exploit analysis," <https://certik.medium.com/worm-hole-bridge-exploit-analysis-5068d79cbb71>, 2022.
- [49] A. Begum, A. Tareq, M. Sultana, M. Soheli, T. Rahman, and A. Sarwar, "Blockchain attacks analysis and a model to solve double spending attack," *International Journal of Machine Learning and Computing*, pp. 352–357, 2020.
- [50] M. Kaleem and W. Shi, "Demystifying pythia: A survey of chainlink oracles usage on ethereum," in *Proceedings of International Conference on Financial Cryptography and Data Security (FC)*, 2021, pp. 115–123.
- [51] "The maker protocol: Makerdao's multi-collateral dai (mcd) system," <https://makerdao.com/en/whitepaper/>, 2022.
- [52] S. He, Q. Wu, X. Luo, Z. Liang, D. Li, H. Feng, H. Zheng, and Y. Li, "A social-network-based cryptocurrency wallet-management scheme," *IEEE Access*, pp. 7654–7663, 2018.
- [53] X. He, J. Lin, K. Li, and X. Chen, "A novel cryptocurrency wallet management scheme based on decentralized multi-constrained derangement," *IEEE Access*, pp. 185 250–185 263, 2019.
- [54] A. G. Khan, A. H. Zahid, M. Hussain, and U. Riaz, "Security of cryptocurrency using hardware wallet and qr code," in *Proceedings of International Conference on Innovative Computing (ICIC)*, 2019, pp. 1–10.
- [55] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of ACM SIGSAC conference on computer and communications security (CCS)*, 2016, pp. 254–269.
- [56] J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo, and T. Chen, "Defectchecker: Automated smart contract defect detection by analyzing evm bytecode," *IEEE Transactions on Software Engineering*, 2021.
- [57] J. He, M. Balunović, N. Ambroladze, P. Tsankov, and M. Vechev, "Learning to fuzz from symbolic execution with application to smart contracts," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019, pp. 531–548.
- [58] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: Towards healthier blockchain technology," in *Proceedings of world wide web conference (WWW)*, 2018, pp. 1409–1418.
- [59] M. Rodler, W. Li, G. O. Karame, and L. Davi, "Evmpatch: Timely and automated patching of ethereum smart contracts," in *Proceedings of the 30th USENIX Security Symposium (USENIX Security)*, 2021, pp. 1289–1306.
- [60] C. Ferreira Torres, M. Baden, R. Norvill, and H. Jonker, "Ægis: Smart shielding of smart contracts," in *Proceedings of ACM SIGSAC conference on computer and communications security (SIGSAC)*, 2019, pp. 2589–2591.
- [61] M. Rodler, W. Li, G. O. Karame, and L. Davi, "Sereum: Protecting existing smart contracts against re-entrancy attacks," *arXiv preprint arXiv:1812.05934*, 2018.
- [62] K. Wüst and A. Gervais, "Ethereum eclipse attacks," *ETH Zurich, Tech. Rep.*, 2016.
- [63] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," *Cryptology ePrint Archive*, 2018.
- [64] G. Xu, B. Guo, C. Su, X. Zheng, K. Liang, D. S. Wong, and H. Wang, "Am i eclipsed? a smart detector of eclipse attacks for ethereum," *Computers & Security*, p. 101604, 2020.
- [65] J. R. Douceur, "The sybil attack," in *Proceedings of International workshop on peer-to-peer systems (IPTPS)*, 2002, pp. 251–260.
- [66] A.-D. Popescu, "Transitions and concepts within decentralized finance (defi) space," *Research Terminals in the social sciences*, 2020.
- [67] "Opyn," <https://www.defipulse.com/projects/opyn>, 2022.