Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.031488

REVIEW

A Review on the Security of the Ethereum-Based DeFi Ecosystem

Yue Xue[1], Dunqiu Fan[2], Shen Su[1,3,*], Jialu Fu[1], Ning Hu[1], Wenmao Liu[2] and Zhihong Tian[1,*]

[1]Cyberspace Institute of Advanced Technology, Guangzhou University, Guanzhou, 510000, China

[2]Innovation Center, NSFOCUS Inc., Beijing, 100089, China

[3]Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education, Beijing, 100816, China

*Corresponding Authors: Shen Su. Email: sushen@gzhu.edu.cn; Zhihong Tian. Email: tianzhihong@gzhu.edu.cn

Received: 20 June 2023 Accepted: 07 September 2023 Published: 30 December 2023

ABSTRACT

Decentralized finance (DeFi) is a general term for a series of financial products and services. It is based on blockchain technology and has attracted people's attention because of its open, transparent, and intermediary free. Among them, the DeFi ecosystem based on Ethereum-based blockchains attracts the most attention. However, the current decentralized financial system built on the Ethereum architecture has been exposed to many smart contract vulnerabilities during the last few years. Herein, we believe it is time to improve the understanding of the prevailing Ethereum-based DeFi ecosystem security issues. To that end, we investigate the Ethereum-based DeFi security issues: 1) inherited from the real-world financial system, which can be solved by macro-control; 2) induced by the problems of blockchain architecture, which require a better blockchain platform; 3) caused by DeFi invented applications, which should be focused on during the project development. Based on that, we further discuss the current solutions and potential directions of DeFi security. According to our research, we could provide a comprehensive vision to the research community for the improvement of Ethereum-based DeFi ecosystem security.

KEYWORDS

Blockchain; smart contract; decentralized finance; DeFi; security

1 Introduction

Decentralized finance (DeFi) refers to a series of financial projects based on the blockchain platforms [1], including Ethereum, BSC, Polygon, Solana, Aptos and many more, conducts financial business (e.g., digital assets [2,3] trading and investment) on top of blockchain ledgers. DeFi has become one of the most attractive investment objectives during the last two years owing to its open, decentralized, and highly transparent characters. Hundreds of billions of dollars [4] are flowing into the DeFi market. Among many blockchains, the Ethereum-based blockchain's locked value accounts for more than 80% [5] of the DeFi market, so this article mainly discusses the Ethereum-based DeFi ecosystem.

From February 2020 to October 2021, the DeFi ecosystem experienced more than 100 incidents with $1.8 billion lost [6]. Project owners had to expend enormous amounts of money and time to compensate the project participants. Worse still, such incidents further harm the investors' confidence and retard the trend of DeFi ecosystem development.

In its role as the financial activities platform over the blockchain ecosystem, DeFi functions similarly to the traditional centralized financial system (CeFi). It can freely combine between applications to improve capital utilization.

However, most Ethereum-based DeFi activities are programmed by a smart contract instead of traditional software, therefore, the financial activities of DeFi are hard to be interrupted in the form of transactions in the blockchain. As a result, systematic problems could also spread more rapidly in DeFi.

At the same time, The DeFi ecosystem is based on the blockchain and smart contract architecture [7] brings advantages: (1) flexibility of the application and absolute control over individual funds from smart contract, ensuring the security of user funds; (2) open source and transparent code, promoting the development of ecology; (3) decentralization, no need for trusted third parties, ensures no restriction from third parties; (4) adopt cryptography mechanism and more secure.

However, there are also many disadvantages: (1) vulnerability in the open-source smart contract code reduces the attack threshold and makes the attack very frequent; (2) the propagation of code reuse enables the same type of vulnerability to be spread in DeFi projects with similar code; (3) the high correlation between the code and the fund makes the attack very profitable; (4) transaction execution has time delay, resulting in the production of Miner Extractable Value; (5) non-discriminate protection of privacy leads to money laundering.

Therefore, this article delves deeper into understanding the intricate dynamics of risk transmission between traditional centralized finance (CeFi) and the emerging decentralized finance (DeFi) platforms. Furthermore, we anchor our discussion on real-world security breach instances. By doing so, we meticulously categorize and dissect the security vulnerabilities inherent in the DeFi sector. Furthermore, we provide actionable and practical solutions to address these challenges. In the subsequent portions of this document, readers will be introduced to the current structural and operational layout of the DeFi ecosystem. A side-by-side, in-depth comparison will be drawn between DeFi and its CeFi counterparts, especially focusing on the unique security risks each poses. Lastly, we will project forward, outlining prospective research avenues and the challenges that lie ahead.

2 Composition of Ethereum-Based DeFi Ecosystem

In its role as a financial system, DeFi uses smart contracts to perform various functions of an economic system, including issuing assets, circulating, and other financial activities. With its broad range of decentralized applications (DApps), DeFi encompasses assets exchanges, investments, etc., for data exchange and value circulation. Therein, we investigated DeFi applications from DappRadar [8], DeFilLama [9], which could be classified from two perspectives, including token systems, decentralized exchange (DEX), decentralized financial intermediaries, oracles, and cross-chain bridges, as shown in Fig. 1.

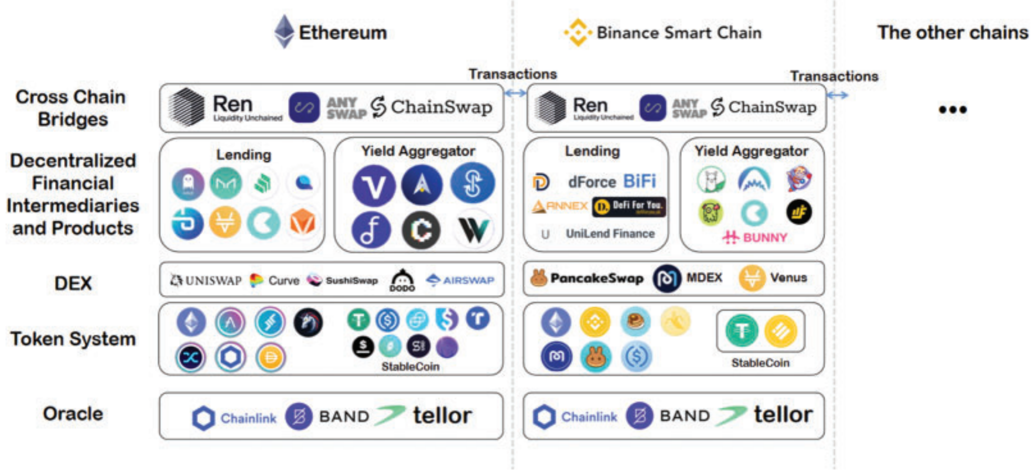

Figure 1: The landscape of DeFi

2.1 *Application Composition*

In order to describe the relationship between various parts of DeFi in a more comprehensive way, we have sorted out the DeFi ecology in Table 1 and classified and symbolized the elements.

Table 1: Elements of DeFi

Description	Expression
Maximum number of tokens that can be created	$MaxSupply = \{x \mid x \in N,\ x > 0\}$
Total number of tokens that has been created	$TotalSupply = \{x \mid x \in N,\ 0 < x <= MaxSupply\}$
Total value of token in the market	$MarketValue = \{x \mid x = TotalSupply * TokenPrice\}$
Price of Token in market	$TokenPrice = \{x \mid x \in N,\ x > 0\}$
Management mechanism of right for *use(u), ownership(o)* and *transfer(t)*	$PermManage = \{x \mid x = f(u,\ o,\ t)\}$
Methods and rules for issuing tokens include *status(s)* and *quantity(q)*	$IssueManage = \{x \mid x = f(s,\ q)\}$
The *frequency(fr)* and *quantity(q)* of token destruction	$DesManage = \{x \mid x = f(fr, q)\}$
Mechanism in the process of token transfer include *tax(t)* and *mechanism(m)*	$TransManage = \{x \mid x = f(t, m)\}$
Distribution mechanism of project income based on *revenue(r)* per token	$IncomeManage = \{x \mid x = f(UserHold,\ MarketValue,\ TotalSupply, r)\}$
Total amount of *tokens locked by users(tl)* through adding liquidity	$TVL = \{x \mid x = \Sigma(tl * TokenPrice\ )\}$

(Continued)

Table 1 (continued)

Description	Expression
Total value of token transactions in the system based on *token amount in one transaction(ta)*	$TradeVolume = \{x\|x = \Sigma(ta * TokenPrice\ )\}$
Management of risks in the system, such as *default risk(dr)* and *credit risk(cr)*	$RiskManage = \{x\|x = \Sigma(f(dr, cr))\}$
Process of matching sales orders such as AMM or off-chain matching	$OrderMatchingManage\ =\ \{x \mid x = \{CFMM,\ CPMM,\ CSMM,\ CMMM,\ off-chain\}$
Process of determining the price of tokens	$PriceManage = \{x \mid x = f(TokenPrice,\ OrderMatchingManage,\ TVL)\}$
Monetary fee charged by the borrower to the borrower	$BorrowRate = \{x \mid x = f(PriceManage,\ TokenPrice,\ TradeVolume,\ IssueManage,\ TotalSupply,\ TVL)\}$
Product yield level	$YieldManage = \{x \mid x = f(BorrowRate,\ PriceManage,\ TradeVolume,\ IssueManage,\ TotalSupply,\ TokenPrice,\ TVL)\}$

2.1.1 Token System

As a general equivalent in centralized finance, currency exists as a medium of exchange and a store of value. Aside from inheriting CeFi's features of currency, DeFi's token ensures autonomy for the business operations of the project, as well as limiting the risk of project assets being lost due to fluctuations in digital asset prices. To implement primary functions such as token transfers, minting, and burning in Ethereum-based blockchains, DeFi applications should inherit the ERC-20 and ERC-777, etc., token standards, token system can be formally expressed as:

$$Token\ System\ =\ (MaxSupply,\ TotalSupply,\ MarketValue,\ TokenPrice,\ Tokenomics) \tag{1}$$

$$MarketValue\ =\ \{x \mid x\ =\ TotalSupply *\ TokenPrice\} \tag{2}$$

$$Tokenomics = \{PermManage,\ IssueManage,\ DesManage,\ IncomeManage\} \tag{3}$$

$$PermManage = \ \{x \mid x\ = f(use,\ ownership,\ transfer)\} \tag{4}$$

$$IssueManage\ =\ \{x \mid x\ =\ f(status,\ quantity)\} \tag{5}$$

$$DesManage\ =\ \{x \mid x\ =\ f(frequency, quantity)\} \tag{6}$$

$$IncomeManage\ =\ \{x \mid x\ =\ f(UserHold,\ MarketValue,\ TotalSupply, r)\} \tag{7}$$

To ensure that the value of digital assets does not fluctuate drastically due to price changes, it is necessary to introduce stablecoin, which serves as a price benchmark in the token system as a unique token. Stablecoin is a digital asset with the property of "anchoring", by anchoring assets off-chain and guaranteeing price stability by maintaining the same value [10]. Take the USDT token as an example:

Tether [11], the company that issued USDT, guarantees that every USDT will be backed by $1. Known as a mortgage stablecoin, USDT can be exchanged for USD anytime, ensuring price stability.

However, the mortgage stablecoin needs a large amount of off-chain asset anchoring to maintain its value, leading to lower capital utilization. To solve this problem, a token called algorithmic stablecoin appears, which retains its value by encouraging the market to speculate on tokens using their agreements. And control the issuance through various methods, to have a high capital utilization rate without a mortgage.

Price adjustment of algorithmic stablecoins can be divided into the bond mechanism (Basis Token [12]) and the airdrop-burn mechanism (Ampleforth [13]). When the token price is below $1, the project owner reduces the market supply by issuing bonds or burning the coins directly to increase the price. As the price rises, the project owner increases market supply by buying back bonds or airdropping coins, decreasing the stablecoin's value.

However, there is the risk of adjusting the supply quantity. The reduction in supply will prompt people to become more concerned about the face value of their stablecoin shrinking. This is based on the market confidence of holders. Thus, resulting in a vicious circle [10], the value of stablecoin may be reduced to zero due to panic selling for other tokens.

2.1.2 Decentralized Exchanges

The CeFi marketplace provides a trading platform for buyers and sellers to exchange financial assets. In DeFi, tokens are freely traded on the blockchain through its Decentralized Exchanges (DEXs), and the market determines the price without the intervention of centralized institutions.

In contrast to financial markets in CeFi, DEXs do not escrow users' token assets anytime, ensuring control over funds and preventing fraud. Current DEXs primarily use the Automated Market Maker (AMM) model, and Ethereum, for example, has used AMM as the core protocol in all of its top-ranked DEXs, including Curve, Uniswap, SushiSwap, Bancor, and Balancer.

- *AMM.* In the centralized financial system, market makers purchase and sell assets through their accounts to provide liquidity to the exchange. Essentially, they facilitate the exchange of goods and services between buyers and sellers and profit from the price difference. This role is known as the Auto Market Maker (AMM) in DEXs, based on smart contracts. Using a smart contract token account maintains a pool of liquid assets and facilitates the trading of tokens. Users can perform several actions: liquidity operations, liquidity mining, and token exchange.

- *Token exchange.* Users swap between the two tokens that already exist in the liquidity pool. Liquidity pools calculate how many tokens users are expected to receive and invest based on their pricing methodology and token reserves. In this operation, the price of tokens in the liquidity pool will change (for example, token A can be exchanged for token B, causing B's price to rise against token A).

- *Liquidity operations.* Upon transferring two tokens to the liquidity pool, users will receive proof tokens proving how much they have provided for token trading, or they can redeem the added liquidity using the liquidity tokens.

- *Liquidity mining.* Liquidity mining is generally considered to be an adjunct to liquidity operations. In order to encourage and reward users for providing and maintaining liquidity, DEX rewards users who redeposit their acquired proof tokens into the mining contract with DEX tokens based on the number of liquidity pool proof tokens.

Therefore, DEX can be formally expressed as:

$$DEX = (TVL,\ TradeVolume,\ PriceManage,\ RiskManage,\ OrderMatchingManage) \tag{8}$$

$$PriceManage = f(TokenPrice,\ TVL) \tag{9}$$

$$RiskManage = \Sigma(f(default\ risk, credit\ risk)) \tag{10}$$

$$OrderMatchingManage = \{CFMM,\ CPMM,\ CSMM,\ CMMM,\ off\ chain\} \tag{11}$$

2.1.3 Decentralized Financial Intermediaries and Products

Decentralized financial intermediaries offer a variety of financial products that allow them to pool users' funds for significant investments in order to raise and transfer funds efficiently. In addition, intermediaries spread risk by investing in a number of DeFi projects, thereby reducing transaction costs.

Furthermore, by uploading the contract code to the blockchain browser, intermediaries are able to be transparent in their investment behavior, eliminating the possibility of insider trading and other drawbacks associated with centralized financial intermediaries. A number of financial products are available in DeFi, including collateralized lending, flash loans, asset management, and others.

• *Collateralized lending.* Through collateralized lending, anyone can supply crypto assets, provide liquidity for lending, receive rewards, and overcollateralize crypto assets. Anyone can act as a liquidator when the collateral value falls below a certain liquidation threshold in order to purchase the collateral at a discount and close the borrower's account.

• *Flash loan.* In the DeFi ecosystem, flash loans are a highly innovative mechanism that provides users with a wider array of capital options and a more comprehensive range of operating options. A flash loan on the blockchain is intended for arbitrage, repayment of collateralized loans, and self-liquidation. The loan is executed through a smart contract that completes borrowing and repayment in one block transaction without pledging any assets. Borrowed assets can be used for custom operations after lending. A user is only required to return the borrowed funds and fees at the end of the transaction. Otherwise, the transaction will be rolled back based on the assertion mechanism in the smart contract as if nothing has happened. Marble protocol [14] first introduced the flash loan concept, and other protocols popularized it.

• *Asset management.* DeFi provides a variety of money management applications, such as portfolio investments [15], that are designed to provide users with more flexibility and higher returns on their assets. In the case of the yield aggregator, once a user pledges their tokens or liquidity pool tokens to the yield aggregator, the yield aggregator automatically selects the platform with the highest returns within the current DeFi ecosystem. Users can withdraw profits and collateral at any time. Users receive a portion of the profits in the form of original rewards, the remaining profits are distributed to users in equivalent yield aggregator tokens. This approach ensures that users receive the revenue they deserve and facilitates the circulation of yield aggregator tokens, thereby allowing the company to grow faster. However, pricing the collateral value of the user's investments can be risky when issuing equivalent returns to investment users, as we will discuss in Section 5.

Take lending and yield aggregator as example, which can be formally expressed as:

$$Aggregator = (TVL,\ Price,\ IncomeManage,\ YieldManage,\ PriceManage, BorrowRate) \tag{12}$$

$$IncomeManage = \{x \mid x = f(UserHold,\ MarketValue,\ TotalSupply, r)\} \tag{13}$$

$$YieldManage = \{x \mid x = f(BorrowRate, TradeVolume, Issue, Supply, Price, TVL)\} \tag{14}$$

$$PriceManage = \{x \mid x = f(TokenPrice, TVL)\} \tag{15}$$

$$BorrowRate = \{x \mid x = f(PriceManage, Price, TradeVolume, Issue, Supply, TVL)\} \tag{16}$$

$$Issue = \{x \mid x = f(status, quantity)\} \tag{17}$$

2.1.4 Decentralized Autonomous Organization

In centralized finance, financial regulatory mechanisms refer to how an organization allocates financial regulation and development planning authority. In DeFi, there is also a need for an organization to make project management transparent and democratized. The community will be better able to collaborate among investors in the project, and revenue distribution and development planning will be more efficient. The technical staff presented decentralized autonomous organizations (DAOs) as a result. Many community rules can be established by DAO, including benchmarks for token issuance, revenue criteria for applications, core contract addresses, and other information.

The members of the community vote on proposals to add or modify community rules using governance tokens issued by the community. By using smart contracts to set regulations and initiate proposals, a DAO draws its strength from the decentralized nature of blockchains and the flexible rule design of smart contracts. There is no possibility of a centralized community manager being malicious and modifying rules without consent because there is no centralized manager, and smart contracts store all governance information.

2.1.5 Coin Shuffle

Coin shuffle is a project that protects users' transfer privacy by randomly mapping account addresses so their transfer history cannot be tracked. Following a preliminary study [16], many other studies [17–20] analyzed or implemented coin shuffles, by including many inputs and outputs in a single transaction, the continuity of transactions is disrupted, thereby severing the connection between inputs and outputs.

Using a contract call, users can deposit digital assets into the pool, obtain deposit credentials, and withdraw previously deposited digital assets to any address. Due to the lack of the voucher itself in the data during the generation and withdrawal phases of the deposit voucher, it is impossible to link the deposits and withdrawals through the deposit voucher, ensuring the transfers of funds to and from the pool are entirely independent and concealing intermediate transfer records [21–23].

Tornado Cash [24] is one of the most widely known coin shuffle programs on Ethereum. By using zero-knowledge proofs, ETH and ERC20 tokens can be sent to any address in an untraceable manner, protecting the privacy of transactions.

2.2 Technical Composition

2.2.1 Blockchain and Smart Contract

DeFi's goals are met through blockchain and smart contracts, including high transparency of transactions, no licenses, no time limit, and absolute control over personal funds. DeFi-related transactions are issued by any node, propagated through the blockchain network, and collected by miners, who place them in their node's pending pools and perform packaged mining. Mined blocks are propagated through the blockchain to other nodes, which receive and verify them, modify their blockchain status, and connect them to the local blockchain.

The smart contract code specifies rules for processing transactions and is stored in the node database. Financial activities between two parties can be conducted without trust when these rules are enforced by the blockchain's consensus. The contract code for a transaction runs in all nodes as soon as it has been packaged and propagated in the blockchain, and all the blockchain data has been updated to a consistent state. The sender of the transaction charges a transaction fee to cover the cost of the state update.

2.2.2 Oracle

Due to the closed-loop ecosystem of DeFi on the chain, which may be insecure and inaccurate due to attacks or unexpected issues, DeFi needs a large amount of off-chain data to support the operation of various tokens and derivatives prices. However, if each node reads the data individually, the result would be different state transition results when executing smart contracts. This would constitute a violation of the blockchain consistency requirement.

Therefore, DeFi utilizes oracle technology in order to ensure the stability and consistency of its project on the chain. In the blockchain industry, this is one of the infrastructures that allow information to be entered into the network from outside the blockchain. By integrating the oracle with the blockchain, events, data, and payment messages from outside the blockchain can be written to smart contracts on the blockchain, enabling a more comprehensive range of applications for smart contracts.

DeFi uses an oracle that collects data from multiple centralized exchanges or commodity prices in the real world and writes it into a smart contract through contract calls. Users can access this data through the contract interface. Currently, oracles such as Chainlink have been widely used in the DeFi ecosystem, including DEXs such as Uniswap, and Aave and financial derivatives projects such as Synthetix and Compound.

2.2.3 Cross-Chain

DeFi's cross-chain exchange between tokens enhances the vitality of the ecosystem by ensuring interoperability between multiple blockchains, as well as the flow of funds. By using a two-way anchor, DeFi users can exchange assets across different blockchains, allowing digital assets to be purchased on one chain and received on another. Many researchers [25–28] investigated cross-chain and proven industrial implementation. An example is Multichain [29], which receives assets from users on the BSC chain and releases assets of the same value to users on the Ethernet chain. Its breadth is enhanced by expanding the ecology of DeFi from a single chain to multiple chains.

3 Inherent Risks from Centralized Finance

DeFi has designed many of its protocols to implement the functions of the centralized financial system. However, these protocols are not entirely secure. Similar risks as centralized finance are introduced: 1) A lack of adequate financial supervision has led to many frauds called rug pull, which are more prevalent in decentralized models; 2) Risk propagation has increased due to the complex interactions between DeFi projects, and financial logic based on highly automated smart contract code has been unable to contain risks on time as a result.

3.1 Rug Pull

Rug pull [30] is one of the most common digital asset scams. Typically, a rug pull manifests as a significant violation by the project's creator who absconds with the invested funds due to operational difficulties or fraudulent intentions. Perpetrators of such nefarious projects amass large quantities of tokens on Decentralized Exchanges (DEXs), pair them with other digital assets such as USDT in a liquidity pool, and entice potential victims via social media platforms. Unwitting investors trade their USDT for these listed tokens, increasing the amount of USDT in the liquidity pool. Consequently, the fraudulent project creators can trade their tokens for substantial profits, leading to a price collapse and resulting in severe losses for many investors.

Further exacerbating the situation is the excessive permissions often granted to project creators. One notable characteristic of a Decentralized Application (DApp) is its decentralization, implying an ideal DeFi application should not be under any single entity's control. The operation of a DeFi project should ideally be governed solely by the smart contract.

However, many DeFi projects retain centralized elements, with the project creators managing funds in the smart contract. Many DeFi whitepapers state that owners can only execute emergency actions, such as suspending contract execution or withdrawing tokens under exceptional circumstances, like security breaches. Nonetheless, this grants the owner near-total control over the funds throughout the project's lifecycle. Therefore, should the project owner harbor malicious intentions or prove incapable of maintaining the project, they can withdraw the funds from the smart contract without any prior notice.

Further, the absence of a regulatory mechanism within the DeFi ecosystem means that no individual or institution can be held accountable for such fraud. Coupled with the difficulty of tracing laundered money due to coin shuffling, the resulting financial loss must be borne by the participants. This condition is undoubtedly detrimental to the DeFi ecosystem's health and development.

Rug pull incidents have occurred repeatedly in the DeFi ecosystem, as shown in Table 2. A total of 14 rug pull incidents have transpired from September 2020 to June 2021, involving several major public chains such as Ethereum, BSC, causing losses of more than $110 million.

Table 2: Rug pull incidents

Date	Platform	Protocol	Loss
19 Sept 2020	BSC	Bantiample	$81,000
20 Sept 2020	ETH	LV finance	$4,000,000
26 Sept 2020	ETH	GemSwap	Unknown
10 Oct 2020	ETH	UniCats	$200,000
10 Nov 2020	Tron	SharkTron	$10,000,000
27 Jan 2021	ETH	refi.finance	$154,512
31 Jan 2021	BSC	popcornswap	$1,920,000
01 Feb 2021	BSC	Multi finance	$200,000
04 Mar 2021	BSC	Meerkat finance	$31,000,000
10 Mar 2021	HECO	HSO	$77,655
20 Mar 2021	BSC	Turtle.dex	$2,000,000
23 May 2021	BSC	DeFi100	$32,000,000
24 Jun 2021	BSC	StableMagnet	$22,000,000
01 Nov 2021	BSC	SQUID	$12,000,000
05 Jan 2022	ETH	Bored bunny	$7,000,000
06 Jan 2022	BSC	Arbix finance	$10,000,000

(Continued)

Table 2 (continued)

Date	Platform	Protocol	Loss
16 Feb 2022	BSC	Bnb42	$2,700,000
23 Feb 2022	BSC	W3M	$235,000
22 Mar 2022	BSC	REALSWAK	$526,500
29 Mar 2022	BSC	BNB DEFI	$112,200

Xia et al. [31] proposed a machine learning approach to detect and flag rug pull risk using malicious tokens based on a malicious token dataset with associated transactions.

3.2 Systemic Risks

Systemic risk is a global risk triggered by external factors and cannot be mitigated by any other diversification investment mechanism. A typical example is the subprime mortgage crisis in centralized finance, which is highly correlated with deteriorating liquidity in the market and interinstitutional collaboration [32]. There are still risks associated with DeFi.

Due to the fact that DeFi uses cryptocurrency as its backing assets, external factors may adversely affect the confidence of investors and market liquidity. Since cryptocurrency prices fluctuate significantly, many investors may sell digital assets in order to protect their investments, resulting in decreased market liquidity and increased systemic risk.

Additionally, smart contract-based DeFi projects are highly programmable. The ability to build projects on top of others can speed up the development process and allow projects to interact with each other, thus causing a chain reaction, increasing the possibility of risk propagation.

On March 12, 2020, the cryptocurrency market saw a black swan event due to a variety of real-world factors, with a large number of cryptocurrencies plummeting, (e.g., a 30% drop in the price of Bitcoin). It resulted in a $400 million reduction in the DeFi total locked value, a 33% drop in 24 h. Furthermore, platforms underwent massive liquidation; for example, MakerDao and Compound reached $10 million and $6 million, respectively. MakerDao had $8.32 million worth of collateral auctioned off at zero price and incurred around $5 million in non-performing debt that could only be repaid through the subsequent auction of its platform token MKR [33].

Several studies have examined the systemic risk associated with DeFi, and Gudgeon et al. [34] and Nadler et al. [35] explored the interconnections and dependencies among DeFi items. In addition, Tolmach et al. [36] proposed a formal process-algebraic technique that models DeFi protocols compositionally for efficient property verification. In the absence of effective collaboration mechanisms and multiple project management systems, it can be challenging to identify and defend against the risks associated with complex systems involving multiple DeFi projects connected to a public chain.

From the perspective of decentralized exchanges, Wang [37] studied the mathematical model of AMM and Angeris et al. [38] analyzed the early Uniswap. In another study, Angeris et al. [39] proposed a method to construct constant function market makers, and Liu et al. [40] proposed decentralized lending protocols. The study by Gudgeon et al. [41] explored the possible risks faced by DeFi and proposed a stress testing framework for DeFi lending protocols, and Bartoletti et al. [42] investigated various implementations of existing lending applications. Perez et al. [43] conducted the first empirical study on the liquidation of lending agreements, which helped DeFi lending programs optimize their contracts to avoid unexpected liquidations in the case of dramatic token changes and helped reduce the propagation of risk in the DeFi ecosystem.

4 Inherent Risks of Blockchain and Related Technologies

In recent years, a number of contract auditing methods based on formal auditing have been proposed to address the security problem associated with blockchain technology. In spite of this, there are still many security issues that cannot be solved by traditional auditing.

At the consensus layer, calculating the block packing order based on gas fees has many problems, leading to front-running attacks (4.3), which affect the outcome of transactions and cause losses to users.

At the contract layer, non-mandatory contract auditing and source code disclosure has significantly lowered the threshold for attacks. As well, many new projects blindly copy contract code from large projects without auditing it, increasing the risk of security breaches across multiple projects.

At the data layer, the existing oracle architecture is not entirely secure, and corresponding security risks may arise. Furthermore, attackers frequently use the coin shuffle to shield their transactions for laundering money from being tracked.

4.1 Project Vulnerability

In addition, the propagation of security issues due to code reuse can lead to attacks on many projects, as will be explained in 4.1.2.

4.1.1 Smart Contract Vulnerability

The DeFi project is based on smart contracts, and smart contract security vulnerabilities have been discussed in studies [44–46]. Therefore, this section does not analyze all vulnerabilities of smart contracts, but only discusses the security issues exploited in DeFi. DeFi security incidents are most frequently caused by smart contract vulnerabilities. From 2020 to May 2022, there have been more than 40 security incidents caused by smart contracts, which accounts for more than 40% of the 112 DeFi security incidents [6] (Fig. 2).

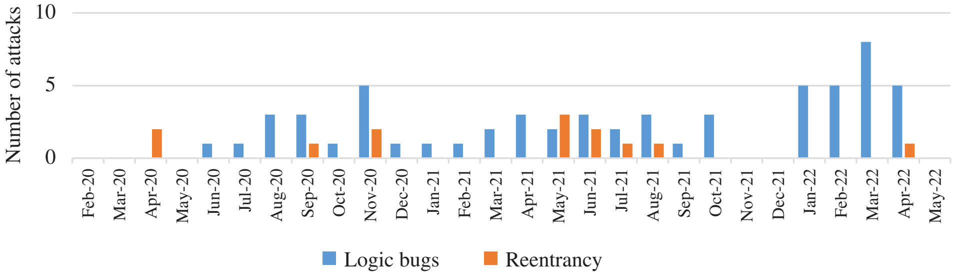

Figure 2: Security issues caused by smart contract vulnerabilities

- Logic Errors

The most common security issue faced by smart contracts is the logical fallacy generated by human error. In DeFi, logic errors are often caused by: 1) Developers specify the contract's essential functions with the incorrect visibility modifier, allowing attackers to manipulate data or obtain permissions arbitrarily. A VETH attack [47] in July 2020 caused $900,000 in damage by exploiting the visibility of the changeExcluded() function without permission. 2) Attackers use incorrect permission judgments to steal vital information or bypass critical logic checks, resulting in severe security problems. In November 2020, the Pickle Finance attack [48] caused approximately $20 million in losses by passing in malicious parameters to the contract that was not validated, resulting in the exchange of real tokens with fake tokens.

- Overflow

Since smart contracts are lightweight and efficient, there are no essential security checks like data security computation and boundary checks, which can result in overflows.

Overflow has been extensively studied as a traditional smart contract vulnerability in [49–51] and has been adequately addressed in contracts. Most DeFi applications use the SafeMath contract or Solidity version 0.8.0 to achieve sufficiently safe integer calculations. However, developers should be careful to use the original operator calculation method in Solidity versions lower than 0.8.0 to avoid overflow.

- Reentrancy

Aside from logic errors and reentrancy, smart contract calls generate complex runtime state spaces that are difficult to handle correctly, thereby creating reentrancy problems.

During the execution of smart contracts, it is difficult to ensure the security of the runtime space. The reentrancy attack may occur when a smart contract calls a malicious external contract. Reentrancy attacks are one of the most destructive forms of smart contract attacks. An example is the DAO attack [52]. The DAO attack was one of the most severe attacks on Ethereum in the early days. The attack not only led to the loss of more than 3.6 million ETH, but it also resulted in the hard fork that severely damaged the Ethereum community consensus.

It is noted that the early reentrancy attack was caused by the *Call()* function while transferring native cryptocurrency. With the development of DeFi, a new type of attack has emerged, in which the attacker adds venture logic into the token transfer function or token receive function to attack.

On April 18 and 19, 2020, the DEX project Uniswap and the lending project Lendf.me suffered reentrancy attacks [53]. Uniswap and Lendf.me must call the token contract function to transfer ETH when performing business functions. The attackers exploited this by releasing a malicious token contract with reentrancy logic in its transfer function. Then, whenever the victim project made a token transfer, the execution of the transaction would enter an insecure runtime space specified by the attackers. Based on the above exploitation basis, the attackers used reentrancy to attack Uniswap and Lendf.me, causing a total loss of about $25 million. Similar attacks primarily involve two token standards, ERC777 and ERC20 (Table 3).

Despite the development of smart contracts, these security issues have not been completely resolved. Numerous research has been conducted on smart contract vulnerability detection. Examples include fuzzing-based ContractFuzzer [54] Echidna [55], EOSFuzzer [56], sFuzz [51], CONFUZZIUS [57], Harvey [58], and ILF [59]; tools based on symbolic execution or CFG analysis, such as teether[60], Oyente [61], Mythril [62], Osiris [63], Seraph [64], SPCON [65] and WANA [66]; studies based on formal validation, such as the formal verification framework for smart contracts proposed by Bhargavan et al. [67], by Hirai et al. [68], in ZEUS [69] and Securify [49]; and research based on machine learning methods such as the predictive model proposed by Momeni et al. [70], in SmartEmbed [71], and in Solidity-coverage [72]. Chen et al. [73], TXSPECTOR [74], EthScope [75], and XBlock-ETH [76] found historical attacks caused by smart contract vulnerability based on transactions in Ethereum.

Table 3: Reentrancy issues

Date	Protocol	ERC Token standard	Loss
18 Apr 2020	Uniswap	ERC777	$220,000
19 Apr 2020	Lendf.Me	ERC777	$24,696,616
13 Nov 2020	Akropolis	ERC20	$2,030,000
17 Nov 2020	OUSD	ERC20	$7,000,000
20 Jun 2021	PolyDEX	ERC777	$500,000
13 Jul 2021	DeFiPie	ERC20	$124,999
30 Apr 2022	RariCapital	ERC777	$80,000,000

4.1.2 Code Reuse

There is a substantial amount of code reuse among many projects in order to reduce development time and costs. Hence, the same vulnerability can affect multiple projects if there is a security vulnerability in the original project's code.

For example, as shown in Table 4, yield aggregators on a BSC chain (AutoShark [77], MerlinLabs [78], PancakeHunny [79], ApeRocket [80]) reused codes from the yield aggregator PancakeBunny. On May 20, 2021, there was a price manipulation attack on PancakeBunny [81]. It was reported that AutoShark and MerlinLabs were attacked on May 25th and 26th; PancakeHunny and ApeRocket were attacked on June 3rd and July 14th. The four subsequent attacks followed almost identical strategies, suggesting that if a DeFi project is found to be vulnerable, its vulnerability may spread to other projects as well. It is a hazardous situation for DeFi at the moment.

Table 4: PancakeBunny-based code reuse security issues

Date	Platform	Protocol	Attack	Loss	
2021.5.20	BSC	PancakeBunny	Price manipulation	$45,000,000	
2021.5.25	BSC	AutoShark	Price manipulation	$750,000	Forked from PancakeBunny
2021.5.26	BSC	MerlinLabs	Price manipulation	$6,800,000	
2021.6.3	BSC	PancakeHunny	Price manipulation	$113,004	
2021.7.14	BSC	ApeRocket	Price manipulation	$1,260,000	

4.2 Insecure Off-Chain Data

Oracle provides off-chain data for DeFi applications. Even so, oracle itself has security and trustworthiness issues, and if security issues arise in oracle projects, it may have an impact on the entire DeFi ecosystem.

Current Oracle solutions are divided into centralized and decentralized oracles based on the number of nodes of the data source under the chain:

- The data source of the centralized oracle comes from a single off-chain service node with a relatively trusted execution environment [82].

• The data source of the decentralized oracle relies on multiple third-party independent nodes to report off-chain data. For example, *Chainlink*, a well-known Oracle project, utilizes 21 independent third-party nodes in order to prevent centralized manipulation of one data source, and also uses token collateral and node reproduction evaluation to prevent multi-node collaboration fraud.

However, with the expanding scale, the amount of money involved increases, and the potential profit from node falsification increases as well, reducing the credibility of the oracle. Meanwhile, the oracle may also be subject to malicious attacks. In the case of a DOS attack against Oracle, for example, the attacker may send large numbers of price requests on Ethereum to increase the cost of uploading price data, which in turn leads to large gas fees on oracle nodes, which eventually exhaust ETH reserves on those nodes.

At the same time, as the cost of uploading prices rises, the price of the corresponding oracle incentive tokens also rises, allowing attackers to take advantage of the situation. In September 2020, attackers used this approach to launch an attack against nine nodes in the *Chainlink*, obtaining many CHI tokens to sell and causing a loss of about 700 ETH, worth $335,000 [83].

4.3 Front-Running Attack

A front-running attack is an act of preemptively purchasing tokens that will be purchased by the victim by raising gas fees on DEX, and then selling the tokens after the victim has bought the tokens, thereby generating a profit on the transaction.

Front-running utilized the risk of transaction order dependency. A transaction that is visible within a transaction pool results in predictable execution results, which may be exploited maliciously. As transactions are sent on the blockchain, they enter the pending pool, where miners select the transactions to package. Miners usually prefer transactions with higher gas fees. But actually, it is also possible for miners to decide based on other criteria, resulting in different orders for issuing transactions and packaging, leading to different transaction execution results.

As shown in Fig. 3, an ordinary user exchanges token X for token Y at a specific price on the AMM-based DEX, increasing the price of Y against X (Fig. 3a). An attacker monitors ordinary users' transactions and maliciously sends a higher fee transaction to buy Y via X before the victim's transaction. As soon as the victim's trade is complete, the attacker sends a transaction to sell Y for X.

Furthermore, miners have the ability to modify the order of transactions when they package transactions into a block. The attacker raises the transaction gas fee paid to miners to execute their transactions preemptively. In this case, the redundant gas fee from the attacker is known as the miner extractable value (MEV), which was first proposed by Daian et al. [84] and is the value miners gain by using the power to rank the order of block transactions.

Apart from front-running attacks, another common form of MEV is the use of arbitrage robots between two or more DEXs. Arbitrage space arises when the token prices of two exchanges deviate. As the degree of adoption of DeFi and the liquidity of decentralized exchanges increase, such arbitrage opportunities continue to emerge with increasing profit margins.

Daian et al. [84] investigated the use of bots in DeFi exchanges for robotic scrambling trades and MEV and profit strategy. Qin et al. [85] quantified the pre-emption transactions due to transaction order dependency and MEV to identify possible pre-emption transactions based on transaction history data. Zhou et al. [86,87] analyzed the possibility of an attack by quantifying it for the robocall attacks generated in the DeFi exchange. Wang et al. [88] discussed general implications for users, DeFi applications, and the community.

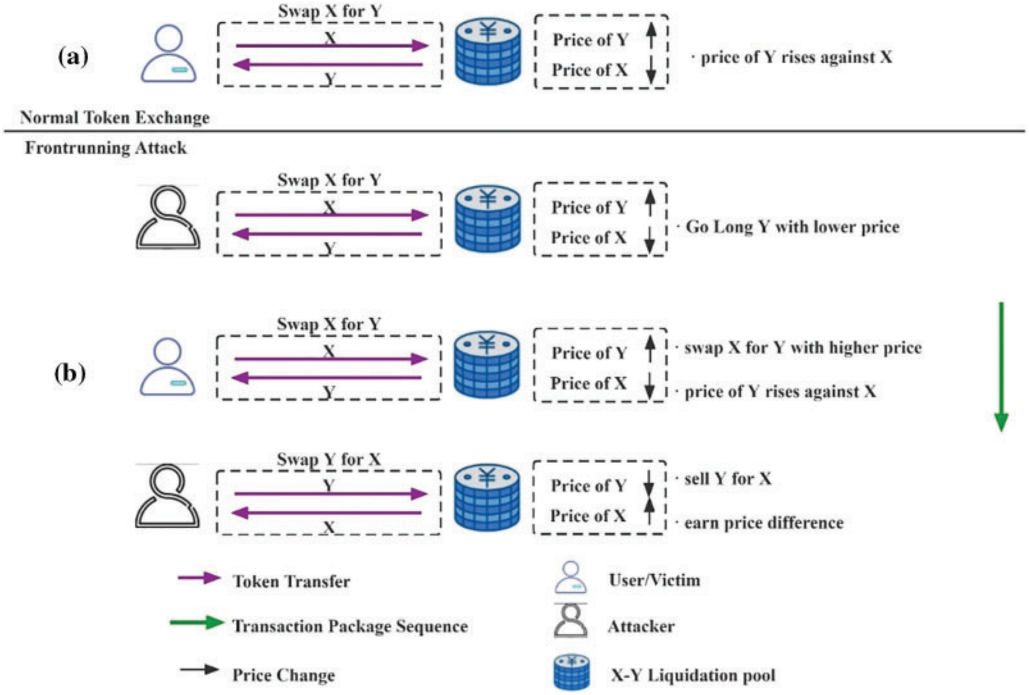

Figure 3: (a) Normal token exchange in decentralized exchange (DEX) and (b) front-running attack in DEX

There is an increasing number of such attacks today, and we think there are two main options:

- Enhanced privacy and speed of transactions: Uniswap V3 utilizes layer2 [89] to execute all transactions off-chain on centralized servers while ensuring the privacy of user transactions and consistency of results from off-chain to on-chain transactions using optimistic rollup. In addition, Chainlink is developing a fair sorting service to solve the MEV problem finally, by separating transaction sorting from block generation and predetermining sorting rules to eliminate miner value extraction.

- MEV auctions, Flashbots [90] proposed an auction system to reduce the external impact and risk posed by MEV on smart contract blockchains to solve the MEV crisis. Front-running attackers can conduct their activities within a limited framework of rules, preventing disorderly MEV activities that can negatively impact DeFi transactions.

4.4 Money Laundering

By using cross-chain and coin shuffle illegally, money is laundered in an untraceable manner. Cross-chain allows users to transfer assets between chains, while coin shuffle protects transaction privacy. Attackers can use them to quickly transfer on-chain assets after an attack, evading the tracking of on-chain security personnel.

As part of the PolyYeld Finance attack on July 28, 2021, the attackers exchanged $250,000 worth of YELD tokens for the exact value of ETH on Ethereum using a cross-chain bridge. Then hid the transaction via Tornado Cash, making the loss of funds from the DeFi project untraceable. Many DeFi attacks proceed through this step [91–93], which is the final step in the process.

5 Risks from Features of DeFi

Due to the lack of regulation and the combination of projects, DeFi aggravates the systemic risk and rug pull issues (Section 3) compared with centralized finance; it also caused a series of new problems in DeFi:

1) DeFi utilizes the DAO to boost users' trust and monitor the project effectively. On the other hand, the calculation of governance power is highly dependent on the number of governance tokens. Thus, users who hold most of the governance tokens can easily control the rights of a DAO, undermining fairness and creating security issues.

2) It is possible that the flash loan can be maliciously exploited by hackers in combination with one or more security vulnerabilities, resulting in even greater losses. Currently, most attacks in DeFi are flash loan related, and they have become the number one DeFi security issue.

3) The lack of an effective regulatory mechanism for DeFi and the fragility of the RFQ (request for quote) design of financial instruments, possibly raise the risk of price manipulation. Data that is abnormal may adversely affect the value assessment of user assets, resulting in serious results.

5.1 Price Manipulation

In Asset management of DeFi applications, the pricing mechanism is fundamental to accurately value the user-pledged DeFi financial assets in real-time; however, a pricing mechanism that is overly dependent on data from one DeFi ecosystem can be easily manipulated.

The yield aggregator calculates the value of user assets based on two types of information: the exchange ratio and the reserve of tokens in the DEX liquidity pool. However, both types of data are manipulable [94]: in some decentralized exchanges, the exchange rate between tokens is calculated by a specific formula, and huge deal transactions can lead to large abnormal fluctuations; in decentralized exchanges, anyone can operate the liquidity pool, allowing them to control the token reserve through significant liquidity additions and withdrawals.

Consequently, by manipulating the corresponding data, the financial instrument may overestimate collateral assets, issue unusual rewards, or withdraw more collateral than was pledged. Afterward, the attacker can resell the rewards or collateral.

Fortunately, such an attack requires considerable financial support, so few individuals are capable of carrying it out successfully. However, flash loans have significantly reduced the threshold for these attacks. The study by Qin et al. [95] provided a preliminary investigation of this attack. The flash-loan-based price manipulation attack is described in detail in Section 5.2.

Since 2021, price manipulation has been one of DeFi's primary attacks (Table 5). A number of DeFi applications, such as PancakeBunny [96], use Oracle to obtain token exchange ratios, thereby avoiding the risk of manipulation of the exchange ratio data obtained from the liquidity pool. However, token reserve data can only be obtained from liquidity pools, which are readily manipulated. It is for this reason that PancakeHunny was attacked on October 20, 2021, despite using the Chainlink Oracle to ensure access to the actual token exchange ratio after being attacked on June 03, 2021.

From an audit perspective, price manipulation is a critical vulnerability that demands attention. According to the research by Zhang et al. [97], not only is the issue of price manipulation frequently observed in the real world, but it also emerges as the most commonly identified vulnerability during the audit processes. Intriguingly, price manipulation is a flaw that requires detection by the fewest average number of individuals. This suggests that uncovering this type of vulnerability often necessitates the expertise of multiple auditors with substantial experience. Such a finding indirectly underscores the challenges in identifying and fully mitigating price manipulation vulnerabilities.

Table 5: Price manipulation attacks

Date	Platform	Flash loan source	Protocol	Loss
18 Dec 2020	ETH	Uniswap	Warp finance	$7,700,000
09 Feb 2021	ETH	Uniswap	BT.Finance	$1,500,000
13 May 2021	ETH	dYdX	xToken	$25,000,000
16 May 2021	BSC	Cream	bEarn Fi	$11,000,000
20 May 2021	BSC	PancakeSwap	PancakeBunny	$45,000,000
22 May 2021	BSC	PancakeSwap	Boogged Finance	$3,000,000
25 May 2021	BSC	PancakeSwap	AutoShark	$750,000
26 May 2021	BSC	PancakeSwap	MerlinLabs	$6,800,000
28 May 2021	BSC	JulSwap	JulSwap	$1,500,000
30 May 2021	BSC	PancakeSwap	Belt finance	$6,200,000
03 Jun 2021	BSC	PancakeSwap	PancakeHunny	$113,004
25 Jun 2021	BSC	PancakeSwap	xWin finance	$281,599
14 Jul 2021	BSC	PancakeSwap	ApeRocket finance	$1,260,000
17 Jul 2021	POLYGON	AAVE	PancakeBunny	$2,402,462
19 Jul 2021	ETH	AAVE	ArrayFinance	$515,000
08 Aug 2021	*[Unknown]*	*[Unknown]*	Zerogoki	$930,000
17 Aug 2021	BSC	PancakeSwap	XSURGE	$5,000,000
25 Aug 2021	Polkadot	*[Unknown]*	Dot.Finance	$429,724
29 Aug 2021	ETH	*[Unknown]*	xToken	$4,500,000
2 Oct 2021	BSC	*[Unknown]*	AutoShark finance	$580,000
06 Oct 2021	BSC	PancakeSwap	My farm pet	$31,424
20 Oct 2021	BSC	Cream finance	Pancake hunny	$1,900,000
23 Nov 2021	BSC	PancakeSwap	Ploutoz finance	$365,000
15 Mar 2022	FTM	*[Unknown]*	Deus finance	$3,000,000
22 Mar 2022	FTM	*[Unknown]*	OneRing	$2,000,000
31 Mar 2022	Fuse	*[Not Flash loan]*	Ola finance	$4,000,000
02 Apr 2022	ETH	*[Not Flash loan]*	Inverse finance	$15,600,000
13 Apr 2022	BSC	PancakeSwap	Elephant money	$22,000,000
28 Apr 2022	FTM	*[Unknown]*	Deus finance	$13,400,000

In DeFi, traditional smart contract vulnerability detectors cannot identify specific vulnerabilities related to financial logic. Therefore, studies specifically target smart contract security issues in DeFi. SciviK [98] was a proposed framework for specifying and verifying smart contracts, which uses an expressive annotation system with EVM low-level execution semantics and SMT solvers to detect vulnerabilities included in DeFi smart contracts. DeFiRanger [99] proposed a method to prevent price manipulation by recovering the transaction data as DeFi semantics and detecting attacks with the flow of funds. BlockEye [100] used the transaction data executed by the DeFi project to perform symbolic inference to detect the presence of price manipulation vulnerabilities in the contract and to defend against the attack.

5.2 Flash Loan Attack

The flash loan is an innovative DeFi application mechanism that provides users with more options for financial activity, but it may pose substantial security risks.

There are many security issues related to centralized finance or blockchain in DeFi, but the cost of an attack limits the damage it can cause and the threshold required to attack. However, a flash loan allows attackers to obtain enormous amounts of money in a single transaction. Consequently, the capital threshold for an attack is drastically reduced, and a large amount of cash magnifies the damage. As a result, flash loan attacks recur in the current DeFi ecosystem, causing many risks, including smart contract vulnerabilities, price manipulation, governance risks, and systemic risks, combine with flash loans to produce significant negative consequences.

Many researchers have tried to analyze flash loans: Wang et al. [101] investigated the role and identification of the flash loan among DeFi; and Cao et al. proposed Flashot [102], a standardized tool and method to describe the form and funding flow of flash loan attacks, and analyzed some existing cases. We compiled statistics based on all relevant security on DeFi and extracted flash loan-related attacks. Out of all 112 DeFi attacks from February 2020 (the first flash loan attack) to October 2021, 48 flash loan attacks have occurred; moreover, their occurrence increased in 2021 (Fig. 4).

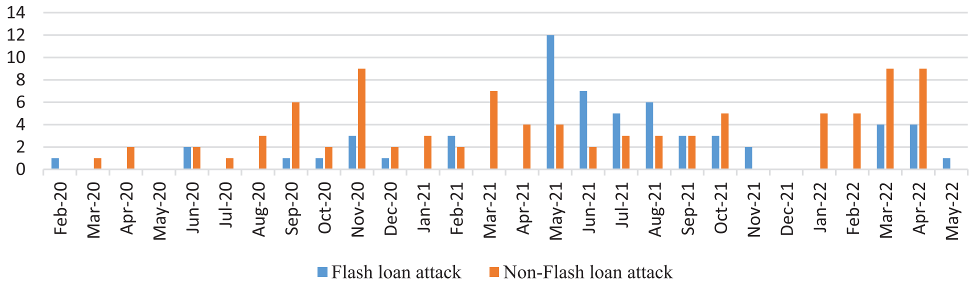

Figure 4: Comparison of flash loan and non-flash loan attacks

A typical case is the governance attack based on a flash loan, which stems from the Maker Dao governance risk, described in Section 5.3. One of the fundamentals of the attack is the accumulation of MKR tokens, which can be achieved by crowdsourcing the required tokens and paying each attack participant a portion of the prize. In order to do so, the attackers must accumulate approximately 50,000 MKR tokens without being detected.

Nevertheless, the advent of flash loan has lowered this threshold. It is possible to attack in other ways since anyone can get large amounts of money at a low cost. After the first bZx attack [103,104] based on flash loan on February 15, 2020, the community's developers became aware of this problem. They passed a vote on February 21, 2020, to activate a governance security module [105] to prevent such flash loan attacks.

Another typical case is the flash-loan-based price manipulation attack. For example, the Pancake-Bunny attack was divided into four steps: flash loan, collateralization, price manipulation, and return of the flash loan.

Under normal circumstances, the PancakeBunny user will provide the BNB-USDT liquidity pool proof token (BNB-USDT LP) to the yield aggregator and call the getRward() function. The yield aggregator will exchange the BNB-USDT LP into BNB-BUNNY LP to evaluate the BNB-USDT LP value that the user deposited and relies on the reserve of BNB in the BNB-BUNNY liquidity pool at that point to mint the reward token BUNNY.

However, due to the existence of the flash loan, the attacker used a large number of BNB tokens to try to manipulate the number of BNB in the BNB-BUNNY liquidity pool on May 20, 2021, causing the minting of a large number of reward tokens BUNNY. Finally, the attacker monetized the BUNNY for the corresponding attack profit, causing the price of BUNNY to plummet, resulting in a loss of $45 million for PancakeBunny. The specific process of the attack is shown in Fig. 5.

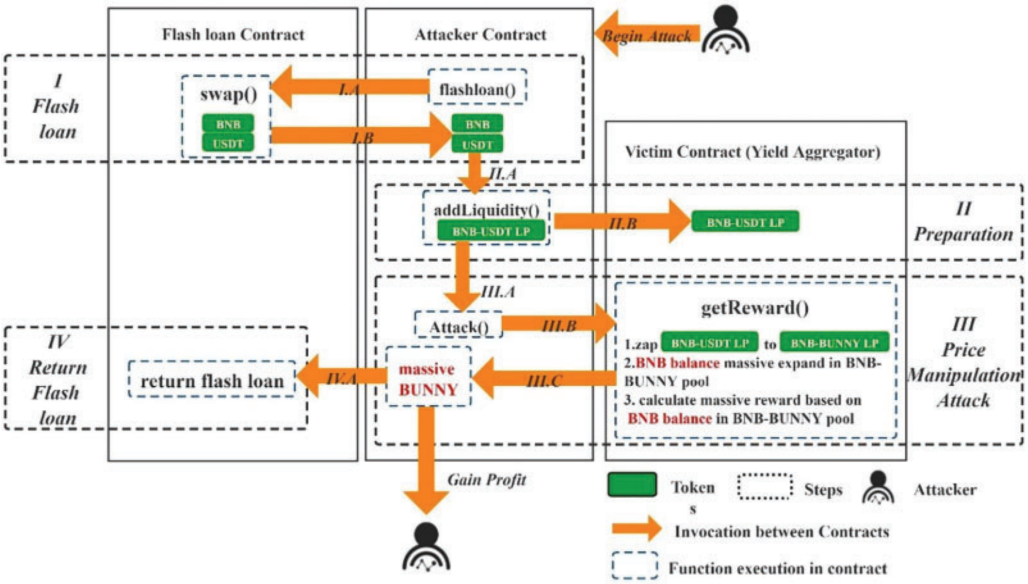

Figure 5: Flash loan-based price manipulation attack

The attack flow in Fig. 5 is as follows:

STEP I.A.-I.B. Flash loan. After the attacker launches the attack, the attacker contract invokes the flash loan contract to initiate the borrowing. Enough *BNB* and *USDT* assets are obtained to make a dramatic change in the price of the pool.

STEP II.A.-II.B. Collateralization. The attacker pledges *BNB* with *USDT* assets to obtain *BNB-USDT LP* and transfers to the victim contract (yield aggregator).

STEP III.A.-III.C. Price manipulation to obtain huge *BUNNY* token rewards. The attacker contract calls the *getReward( )* in the yield aggregator. The yield aggregator exchanges a large amount of *BNB-USDT LP* held into a large amount of *BNB-BUNNY LP* for evaluating the user's deposited *BNB-USDT LP* value. This will cause the amount of *BNB* in the *BNB-BUNNY* liquidity pool to expand significantly, eventually minting a massive amount of bonus tokens *BUNNY*.

STEP IV. A. Return flash loan. The attacker monetizes the massive amount of *BUNNY* into *BNB* and *USDT*, the flash loan is returned, and the remaining part is the profit from the attack.

5.3 *Governance Regulation Attack*

Using a DAO for governance, DeFi has lowered its governance threshold to attract users, and users can vote on the operation of the DeFi community based on how many tokens they hold. The vote will determine the project's core contract address and key parameters like its return rate. This means that users with more tokens have more control over the community. An extreme increase in this number may result in malicious manipulation of key contracts in the community, resulting in a substantial negative impact on the community's security.

In the Maker Dao governance risk event in December 2019, Micah Zoltu specified how to attack the Maker Dao community's governance contracts. By accumulating enough MKR tokens, the attacker can replace the existing governance contract with his malicious one. This allows the attacker to take control of the system and withdraw collateral in Dai, causing the Maker Dao community to lose funds.

Among the reasons for this risk, we believe the voting power calculation in the Maker Dao governance community is based solely on the number of MKR tokens owned by participants. The wealthy with a large number of MKR tokens benefit from this overly monolithic governance calculation.

The same problem occurs in other DeFi governance communities, such as Uniswap, whose governance depends on its platform token UNI. Almost all large DeFi projects involve governance communities that use their governance tokens to evaluate governance rights.

The quantification of the fragmentation of governance power among DeFi applications and the assessment of the capabilities and limitations of token governance helps governance communities optimize their governance power design. However, the current governance community faces design flaws in governance power sources [106–108]; designers of governance communities should consider balancing the pervasiveness and security of voting power mechanisms. As described in Section 5, the high pervasiveness of governance rules can lower the threshold to attract users to govern. However, it will lead to a simple voting power design, which users with many governance tokens easily exploit. In contrast, low pervasiveness and complex voting power design may be more secure. Still, they may be targeted explicitly by some users with ulterior motives to find loopholes in the rules and profit for themselves.

5.4 *Financial Model Attack*

In decentralized finance (DeFi), financial models constitute the cornerstone of any DeFi project. They delineate the methods of fund allocation, operation, and management across the project. Given that these models are often predicated on sophisticated algorithms and strategies, they inevitably become focal points for attackers. Malefactors aim to exploit potential vulnerabilities or inadequacies in these models to achieve illicit gains. Apart from the typical financial model attacks outlined in Sections 5.1–5.3, DeFi is rife with vulnerabilities inherent in its economic models, as delineated below:

Delay in Interest Update: It is an incorrect sequence of operations [109–111]. When code statements that update or accrue interest or exchange rates are executed after those that modify user balances, stakes, fees, loans, or rewards, it can result in miscalculated financial outcomes. The correct logic demands that interest or exchange rates be updated prior to any asset recalculations. This misordering can inadvertently introduce financial discrepancies, potentially leading to significant losses or unfair gains within the DeFi ecosystem.

Misordered Checkpoint Accounting: This vulnerability [112,113] emerges when user checkpoints are invoked after calculations affecting balances, shares, stakes, fees, loans, or rewards. Such a sequence can inadvertently result in incorrect accounting, potentially leading to unintended value distributions or imprecise financial records. This misordering can compromise the integrity of the contract and disrupt the intended functionality. Proper sequencing—invoking user checkpoints before any financial calculation—is crucial to ensure the accuracy and reliability of DeFi smart contracts.

Initial Liquidity Manipulation: Such vulnerability [114] arises when the initial depositor can unduly influence the total share or minted amount. In scenarios where the total supply or liquidity equals zero, the first depositor is endowed with the power to set the total share equivalent to their first deposit. This can potentially compromise the minting of shares or even lead to liquidity drainage, impacting all users. To mitigate, a protocol can send the initial minimum liquidity LP tokens to a zero address, ensuring share dilution and preventing exploitation.

Based on the provided vulnerabilities, the commonalities and vulnerabilities present in the financial models of decentralized finance (DeFi) projects can be distilled into the following points, linking them with the definition of a financial model:

Sequence Vulnerabilities: Both "Delay in interest update" and "Misordered Checkpoint Accounting" highlight the inherent risks associated with the order of operations in DeFi contracts. Financial models are essentially a sequence of calculations and logic. If the order of these operations is incorrect, it can drastically alter the expected outcome. Such errors can have financial consequences, as miscalculations can result in significant losses or unintended distributions.

Initial State Vulnerabilities: The "Initial Liquidity Manipulation" underscores the susceptibility of DeFi systems during their inception or zero-state. Financial models often make assumptions about the state of the system, and if these assumptions don't hold, the model's predictions and operations can go awry. In DeFi, where systems are initialized and can grow autonomously, ensuring that initial conditions are robust is crucial.

Reliance on Accurate Algorithms and Logic: At the heart of DeFi is the automation of financial transactions and strategies. These are predicated on algorithms and logical processes outlined in smart contracts. The slightest deviation, be it in sequencing, setting initial conditions, or post-transaction activities, can distort the system's behavior from its intended purpose.

In conclusion, financial models in DeFi serve as blueprints for automated financial systems. They define how funds are allocated, managed, and operated across a project. However, their decentralized and autonomous nature makes them vulnerable to a host of potential issues. Proper sequencing of operations, careful initialization, and rigorous post-transaction checks are essential to ensuring the security and functionality of these systems. Moreover, a thorough understanding of the inherent vulnerabilities can aid in the creation of more resilient and trustworthy DeFi platforms.

6 Open Problems on DeFi Security and Future Directions

We have divided and analyzed the advantages and disadvantages and security risks faced by the DeFi ecosystem (Table 6). Based on such categories, DeFi may face several security challenges as the following are the left open problems to our best knowledge.

Table 6: Current advantages and disadvantages and risk

Key to DeFi implementation	Risk	Open problem
Inherent from centralized finance	Rug pull Systemic risks	Regulatory mechanism
New feature in DeFi	Price manipulation Flash loan Governance attack	DeFi project vulnerability detection and on-chain attack defense
Inherent from blockchain	Smart contract vulnerability Vulnerability propagation	
	Off-chain data source unsafety	Regulatory mechanism
	Transaction order dependency	Miner extractable value
	Money laundering	Regulatory mechanism

6.1 DeFi Project Vulnerability Detection

Detecting vulnerabilities in decentralized finance (DeFi) smart contracts, especially those tied to financial business logic, remains a challenge. While tools like SciviK, DeFiRanger, BlockEye, and the methodologies proposed by Wang et al. provide insights into generic smart contract vulnerabilities, they often fail to delve into the nuances of financial logic inherent in DeFi contracts. These nuances, especially concerning price manipulation, flash loans, and token operations (transfer, burn, and minting), demand specialized attention.

A major obstacle in pinpointing business logic vulnerabilities is the limitations of contemporary detection tools [115], such as symbolic execution and fuzzing. While these are apt at identifying broad vulnerabilities, they grapple with extracting insights specific to DeFi vulnerabilities due to their design that gravitates towards general detection. For example, DeFi smart contracts' peculiar parameters and triggers often necessitate advanced mutation techniques beyond basic random mutations to unveil concealed vulnerabilities.

To address these challenges, leveraging generative artificial intelligence models, such as ChatGPT, emerges as a promising solution. ChatGPT's inherent capability in interpreting code [116,117], especially in business logic analysis, offers unique advantages. By utilizing such capabilities, it becomes feasible to bridge the gap between business scenarios and detection tool rules. However, the specifics of this implementation method still warrant further exploration.

Furthermore, the unique characteristics of business logic within DeFi amplify the complexity of vulnerability detection [118,119]. Each DeFi initiative often brings forth innovative mechanisms, economic structures, and operating protocols. Given the custom nature of these contracts, comparing them with historical vulnerabilities becomes intricate. Unlike conventional financial systems where established practices might render vulnerability patterns somewhat discernible, the ever-evolving DeFi landscape implies that each new venture could usher in unparalleled security risks.

To tackle such intricacies, a plausible approach is to modularize and structure DeFi. This can be achieved by dissecting DeFi into distinct sub-functionalities [120], meticulously examining the security concerns and norms associated with each, and integrating conventional financial standards into these security norms. Through this method, one can systematically analyze the functional constituents of DeFi, as well as the financial models and potential vulnerabilities stemming from them.

Moreover, many DeFi vulnerabilities are intricately linked with the contract's on-chain state. A specific vulnerability might only be evident under particular chain conditions, or its exploitability could be profoundly influenced by the blockchain's current state. Current detection tools often divorce smart contracts from their on-chain state during examination, neglecting the dynamic relationship between the two. To address this, integrating real-time on-chain state retrieval in these tools is paramount. By incorporating genuine DeFi data into the vulnerability analysis process, the accuracy of such assessments can be considerably enhanced.

In conclusion, while the perpetual evolution of DeFi paves the way for fresh financial possibilities and breakthroughs, it concurrently births unprecedented security dilemmas. The custom-tailored nature of DeFi projects, coupled with the innate deficiencies of existing tools to discern and adjust to distinctive business environments, underscores the need for vulnerability detection to be a perpetually adaptive and specialized domain.

6.2 On-Chain Attack Defense

In the domain of Ethereum's decentralized framework, immutability has emerged as both a boon and a bane. The inalterability of smart contracts, which ensures that once stipulations are coded and deployed, they remain irreversible, also heralds challenges. Any inadvertent error or subsequently discovered vulnerability within the smart contract becomes immutable, often culminating in financial vulnerabilities and compromised system integrity.

The innovative introduction of proxy contracts emerges as a solution to this conundrum. These contracts function as intermediaries between end-users and the core logic or data contract. Instead of directly interfacing with the primary smart contract, users engage with this proxy. Such a bifurcation permits the modification or upgrading of the foundational logic or data contract without necessitating a change in the address that users predominantly interact with. Conceptually, this proxy contract functions as a dynamic gateway to the contemporaneous version of logic or data. When one contextualizes this within the ambit of DeFi projects, which are notorious for their large value transactions, the significance of this proxy becomes paramount. Through it, a plethora of security layers, transaction filters, and permissioned controls can be implemented, ensuring the sanctity and legitimacy of operations that gain approval.

Shifting our lens to the off-chain transactional ecosystem, it is evident that analysis in this realm is instrumental in fortifying DeFi platforms. By scrutinizing transactions relegated to the mempool (effectively a repository for transactions awaiting confirmation), one can discern patterns and potential anomalies indicative of malicious intent. Instances of unexpected escalations in gas prices or a conspicuous focus on particular contracts might allude to attempts at front-running or analogous malevolent strategies.

The phenomenon of front-running, while not novel to traditional financial systems, has acquired unique dimensions within the decentralized context. Within Ethereum's sphere, it pertains to the act of preemptively discerning a transaction-in-waiting within the mempool and subsequently initiating another transaction, typically with a higher gas bid, ensuring its preferential processing. Through meticulous off-chain analyses, potential front-running strategies can be identified, empowering DeFi platforms to recalibrate their defenses or alert their user base. Implementing protective stratagems—be it through introducing stochastic delays, employing commit-reveal schemes, or leveraging the capabilities of layer2 solutions with expedited confirmation cycles—can serve as effective deterrents against these malevolent tactics.

To encapsulate, the inherent rigidity of Ethereum-based blockchains, while posing intrinsic challenges, also underscores the ingenuity and adaptability of the community. As the landscape evolves, it is imperative to adopt a stratified approach amalgamating the strengths of on-chain proxy mechanisms, rigorous off-chain analytical frameworks, and proactive defense against front-running. Such a holistic strategy will be instrumental in upholding the resilience and credibility of the burgeoning DeFi sector.

6.3 Regulatory Mechanism

Expanding on the aforementioned perspective on the DeFi ecosystem, it is essential to delve deeper into the systemic risks that arise from code cloning and project dependencies, as well as the crucial need for monitoring contracts.

In the fast-paced world of DeFi, it is common for projects to "fork" or clone existing codebases, tweak them slightly, and then launch them as new protocols. This is partly due to the open-source nature of many DeFi projects. While this can speed up innovation and the spread of good ideas, it also introduces systemic risk. If the original codebase has a flaw, that flaw might proliferate across many projects, creating a potential domino effect. Just like in traditional finance, where a common point of failure can lead to cascading collapses (think about the 2008 financial crisis), in DeFi, a shared vulnerability in many cloned projects can cause massive losses in a short period.

Additionally, the interdependencies between projects, often termed "money legos" in the DeFi space, is another potential point of systemic risk. Projects are often intertwined in a complex web of smart contracts and interlinked liquidity pools. If one project faces a bug or a liquidity crisis, it can quickly ripple through the ecosystem, affecting many other projects and users.

Given these risks, the need for monitoring contracts becomes apparent. Such contracts can act as guardians, overseeing the interactions between different projects. They can ensure that projects adhere to standard best practices and that the interactions between different projects do not introduce unforeseen vulnerabilities. These monitoring contracts can be both proactive and reactive: proactively checking and validating interactions, and reactively halting or modifying operations when unusual patterns or potential threats are detected.

However, the implementation of monitoring contracts does bring up several challenges:

Trust: Who oversees these monitoring contracts? If these are controlled by a centralized entity, it goes against the ethos of decentralization inherent to DeFi.

Efficiency: With an ever-increasing number of transactions and interactions in the DeFi space, the monitoring contracts must be efficient to ensure they do not cause unnecessary congestion or latency in the system.

Updatability: As DeFi evolves, so too will the best practices and standards. These contracts must be updatable to adapt to the changing landscape.

In conclusion, while the promise of DeFi is immense, its decentralized nature brings about unique challenges that the centralized financial world does not face. Regulation at the interface and interaction level, through tools like monitoring contracts, could be a balanced approach, offering protection without stifling innovation. But careful design, robust implementation, and a community-wide consensus are vital for such solutions to be effective and accepted.

6.4 Miner Extractable Value

As for MEV, in addition to transaction privacy protection offered by BackRunMe and Flashbots, some possible solutions include algorithmic trading, where traders buy or sell on a large scale in financial markets, which has a significant impact on an illiquid market. In order to reduce the adverse effects of market volatility on the exchange, traders usually split the orders, divide the large-scale transactions into smaller ones at the right time to minimize the associated transaction costs, and ensure that the whole transaction process reaches the target price level. In DeFi, a token exchange transaction for DEX is split into multiple smaller transactions, and the transactions are mixed and gridded in stages to reduce the profit margin caused by a front-running attack.

Meanwhile, another layer2 implementation, zkRollup [121], is a layer2 project launched by zkSync [122]. Which product zkEVM [123] can already partially support smart contract execution, and DEX projects such as uniswap have a high possibility of migrating to zkEVM in the future. With zkEVM's centralized server approach, transactions can be executed almost in real-time, which alleviates the risk of exploiting transaction information caused by delayed transactions in the chain.

6.5 Governance for DAO

To prevent the DAO from the issues described in Section 5, the scheme proposed by Wulf A. Kaal focuses on the following points: 1) the anonymity of DAO members, achieved through the hiding of personal identities avoids the impact of real-world identities on DAO governance; 2) the irreplaceability of governance tokens, which is different from traditional governance tokens based on ERC-20 protocols. The value of members can be differentiated and quantified by non-tradable tokens, avoiding the rich people who have greater control over the DAO by purchasing tokens in previous cases.

7 Conclusion

The realm of DeFi presents an intriguing juxtaposition against traditional centralized financial systems. Unlike most existing research [124–126], this study stands out by specifically examining the paradigm shift from centralized to decentralized financial mechanisms. Beyond just a surface analysis, we take a deep dive into concrete case studies of DeFi security incidents. Most distinctively, our research introduces innovative, forward-looking strategies in the domains of vulnerability detection and transaction interception, areas largely unexplored in conventional studies.

This multi-dimensional perspective on DeFi, enriched by its focus on decentralized governance, intricate coding nuances, and consensus algorithms, offers a fresh viewpoint on the evolution and challenges of the ecosystem. While blockchain technology continues to evolve at breakneck speed, the ultimate outcome of the DeFi narrative remains an unfolding story. Still, the relentless pursuit of innovation persists.

Acknowledgement: The authors would like to express their gratitude to all who supported this research. We are thankful for the insightful comments from anonymous reviewers, which have greatly improved this manuscript. We also appreciate the financial assistance provided by the funding agencies.

Funding Statement: This work is supported by the Key-Area Research and Development Program of Guangdong Province 2020B0101090003, CCF-NSFOCUS Kunpeng Scientific Research Fund (CCF-NSFOCUS 2021010), Innovation Fund Program of the Engineering Research Center for Integration and Application of Digital Learning Technology of Ministry of Education under Grant No. 1221027, National Natural Science Foundation of China (Grant Nos. 61902083, 62172115, 61976064), Guangdong Higher Education Innovation Group 2020KCXTD007 and Guangzhou Higher Education Innovation Group (No. 202032854), and Guangzhou Fundamental Research Plan of "Municipal-School" Jointly Funded Projects (No. 202102010445), and Guangdong Province Science and Technology Planning Project (No. 2020A1414010370).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yue Xue, Shen Su; data collection: Yue Xue, Jialu Fu; analysis and interpretation of results: Yue Xue, Dunqiu Fan, Zhihong Tian; draft manuscript preparation: Yue Xue, Wenmao Liu, Ning Hu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The vulnerability data used in this study were collected from two sources: https://web3sec.xrex.io/ and SlowMist Hacked (https://hacked.slowmist.io/). The former provides a database of DeFi vulnerabilities and related information. The latter documents security incidents in the blockchain ecosystem. These sources are publicly available online. The detailed data are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Chen, Y., Bellavitis, C. (2020). Blockchain disruption and decentralized finance: The rise of decentralized business models. *Journal of Business Venturing Insights, 13,* e00151. https://doi.org/10.1016/j.jbvi.2019.e00151
2. Zheng, M., Robbins, H., Chai, Z., Thapa, P., Moore, T. (2018). Cybersecurity research datasets: Taxonomy and empirical analysis. *11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)*, Baltimore, MD, USENIX Association. https://www.usenix.org/conference/cset18/presentation/zheng (accessed on 22/11/2023)
3. Martins, F. F., Matos, D. R., Pardal, M. L., Correia, M. (2020). Recoverable token: Recovering from intrusions against digital assets in Ethereum. *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pp. 1–9. Cambridge, MA, USA, IEEE. https://doi.org/10.1109/NCA51143.2020.9306738
4. DappRadar (2023). https://dappradar.com/defi (accessed on 22/11/2023)
5. DefiLlama (2023). https://defillama.com/chains (accessed on 22/11/2023)
6. SlowMist Hacked (2023). https://hacked.slowmist.io/ (accessed on 22/11/2023)
7. Ethereum and DeFi (2023). https://ethereum.org/en/defi/#ethereum-and-defi (accessed on 22/11/2023)
8. DappRadar (2023). https://dappradar.com/ (accessed on 22/11/2023)
9. DeFiLlama (2023). https://defillama.com/ (accessed on 22/11/2023)
10. Bullmann, D., Klemm, J., Pinna, A. (2019). In search for stability in crypto-assets: Are stablecoins the solution? *Occasional Paper Series 230.* European Central Bank. https://doi.org/10.2866/969389
11. Tether (2023). https://tether.to/ (accessed on 22/11/2023)
12. Al-Naji, N., Chen, J., Diao, L. (2017). Basis: A price-stable cryptocurrency with an Algorithmic Central Bank. https://www.basis.io/basis_whitepaper_en.pdf (accessed on 22/11/2023)
13. Ampleforth (2023). https://docs.ampleforth.org/ (accessed on 22/11/2023)
14. Wolff, M. (2018). Introducing Marble. https://medium.com/marbleorg/introducing-marble-a-smart-contract-bank-c9c438a12890 (accessed on 22/11/2023)

15. Feng, F., Weickmann, B. (2019). Set: A protocol for baskets of tokenized assets. https://www.setprotocol.com/pdf/set_protocol_whitepaper.pdf (accessed on 22/11/2023)
16. Le, D. V., Gervais, A. (2020). AMR: Autonomous coin mixer with privacy preserving reward distribution. arXiv preprint arXiv:2010.01056.
17. Stone, D. (2021). Trustless, privacy-preserving blockchain bridges. arXiv preprint arXiv:2102.04660.
18. Zhu, S. (2021). *Privacy preservation & security solutions in blockchain network (Ph.D. Thesis)*. Georgia State University, USA.
19. Conti, M., Sandeep Kumar, E., Lal, C., Ruj, S. (2018). A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials, 20(4),* 3416–3452. https://doi.org/10.1109/COMST.2018.2842460
20. Li, C., Palanisamy, B. (2018). Decentralized privacy-preserving timed execution in blockchain-based smart contract platforms. *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, pp. 265–274. Bengaluru, India. https://doi.org/10.1109/HiPC.2018.00037
21. Ruffing, T., Moreno-Sanchez, P., Kate, A. (2014). CoinShuffle: Practical decentralized coin mixing for bitcoin. In: Kutyłowski, M., Vaidya, J. (Eds.), *Computer security–ESORICS 2014*, pp. 345–364. Cham: Springer International Publishing.
22. Glaeser, N., Maffei, M., Malavolta, G., Moreno-Sanchez, P., Tairi, E. et al. (2022). Foundations of coin mixing services. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, in CCS'22*, pp. 1259–1273. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3548606.3560637
23. Ni, W. Z., Cheng, P., Chen, L. (2022). Mixing transactions with arbitrary values on blockchains. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2602–2614. Kuala Lumpur, Malaysia. https://doi.org/10.1109/ICDE53745.2022.00240
24. Tornado Cash (2023). https://tornadocash.eth.link/ (accessed on 22/11/2023)
25. Herlihy, M. (2018). Atomic cross-chain swaps. *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, in PODC'18*, pp. 245–254. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3212734.3212736
26. Wang, H., Cen, Y., Li, X. (2018). Blockchain router: A cross-chain communication protocol. *Proceedings of the 6th International Conference on Informatics, Environment, Energy and Applications*, pp. 94–97. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3070617.3070634
27. Deng, L., Chen, H., Zeng, J., Zhang, L. J. (2018). Research on cross-chain technology based on sidechain and hash-locking. In: Liu, S., Tekinerdogan, B., Aoyama, M., Zhang, L. J. (Eds.), *Edge computing–EDGE 2018*, pp. 144–151. Cham: Springer.
28. Jiang, Y., Wang, C., Wang, Y., Gao, L. A. (2019). A Cross-chain solution to integrating multiple blockchains for IoT data management. *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1035–1041. Halifax, NS, Canada. https://doi.org/10.1109/Cybermatics_2018.2018.00192
29. Multichain (2023). https://multichain.org/ (accessed on 22/11/2023)
30. CoinMarketCap (2023). Rug pull. https://coinmarketcap.com/alexandria/glossary/rug-pull (accessed on 22/11/2023)
31. Xia, P., Wang, H., Gao, B., Su, W., Zhou, Y. et al. (2021). Demystifying scam tokens on uniswap decentralized exchange. arXiv preprint arXiv:2109.00229v1.
32. Demyanyk, Y., van Hemert, O. (2011). Understanding the subprime mortgage crisis. *The Review of Financial Studies, 24(6),* 1848–1880.

33. Igor Igamberdiev (2020). Black Thursday for MakerDAO: $8.32 million was liquidated for 0 DAI. https://medium.com/@whiterabbit_hq/black-thursday-for-makerdao-8-32-million-was-liquidated-for-0-dai-36b83cac56b6 (accessed on 22/11/2023)
34. Gudgeon, L., Werner, S., Perez, D., Knottenbelt, W. J. (2020). DeFi protocols for loanable funds: Interest rates, liquidity and market efficiency. *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pp. 92–112. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3419614.3423254
35. Nadler, M., Schär, F. (2020). Decentralized finance, centralized ownership? An iterative mapping process to measure protocol token distribution. arXiv preprint arXiv:2012.09306.
36. Tolmach, P., Li, Y., Lin, S., Liu, Y. (2021). Formal analysis of composable DeFi protocols. arXiv preprint arXiv:2103.00540.
37. Wang, Y. (2020). Automated market makers for decentralized finance (DeFi). arXiv preprint arXiv:2009.01676.
38. Angeris, G., Kao, H., Chiang, R., Noyes, C., Chitra, T. (2019). An analysis of Uniswap markets. arXiv preprint arXiv:1911.03380.
39. Angeris, G., Evans, A., Chitra, T. (2021). Replicating market makers. arXiv preprint arXiv:2103.14769.
40. Liu, B., Szalachowski, P., Zhou, J. (2020). A first look into DeFi oracles. arXiv preprint arXiv:2005.04377.
41. Gudgeon, L., Perez, D., Harz, D., Livshits, B., Gervais, A. (2020). The decentralized financial crisis. arXiv preprint arXiv:2002.08099.
42. Bartoletti, M., Chiang, J. H., Lafuente, A. L. (2021). SoK: Lending pools in decentralized finance. *Financial Cryptography and Data Security. FC 2021 International Workshops*, pp. 553–578. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-662-63958-0_40
43. Perez, D., Werner, S. M., Xu, J., Livshits, B. (2020). Liquidations: DeFi on a knife-edge. arXiv preprint arXiv:2009.13235.
44. Atzei, N., Bartoletti, M., Cimoli, T. (2017). A survey of attacks on ethereum smart contracts (SoK). In: Maffei, M., Ryan, M. (Eds.), *Principles of Security and Trust*, pp. 164–186. Berlin: Springer.
45. Perez, D., Livshits, B. (2019). Smart contract vulnerabilities: Does anyone care? arXiv preprint arXiv:1902.06710.
46. Tsankov, P., Dan, A., Drachsler-Cohen, D., Gervais, A., Bünzli, F. et al. (2018). Securify: Practical security analysis of smart contracts. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 67–82. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3243734.3243780
47. Transaction Details (2020). https://etherscan.io/tx/0xdd1120a90ed4112b634266d6a244b93ca86785317bc75f0e170ab0cd97c65224 (accessed on 22/11/2023)
48. Kuhn, D., Reynolds, K. (2020). DeFi protocol Pickle Finance token loses almost half its value after $19.7M hack. https://www.coindesk.com/markets/2020/11/22/defi-protocol-pickle-finance-token-loses-almost-half-its-value-after-197m-hack/ (accessed on 22/11/2023)
49. Lai, E., Luo, W. (2020). Static analysis of integer overflow of smart contracts in ethereum. *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, pp. 110–115. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3377644.3377650
50. Sun, J., Huang, S., Zheng, C., Wang, T., Zong, C. et al. (2022). Mutation testing for integer overflow in ethereum smart contracts. *Tsinghua Science and Technology, 27(1),* 27–40. https://doi.org/10.26599/TST.2020.9010036
51. Nguyen, T. D., Pham, L. H., Sun, J., Lin, Y., Minh, Q. T. (2020). sFuzz: An efficient adaptive fuzzer for solidity smart contracts. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, in ICSE'20*, pp. 778–788. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3377811.3380334

52. Finley, K. (2016). A $50 million hack just showed that the DAO was all too human. https://www.wired.com/2016/06/50-million-hack-just-showed-dao-human/ (accessed on 22/11/2023)

53. Tokenlon DEX (2020). About recent Uniswap and Lendf.Me reentrancy attacks. https://medium.com/imtoken/about-recent-uniswap-and-lendf-me-reentrancy-attacks-7cebe834cb3 (accessed on 22/11/2023)

54. Jiang, B., Liu, Y., Chan, W. K. (2018). ContractFuzzer: Fuzzing smart contracts for vulnerability detection. *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pp. 259–269. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3238147.3238177

55. Grieco, G., Song, W., Cygan, A., Feist, J., Groce, A. (2020). Echidna: Effective, usable, and fast fuzzing for smart contracts. *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 557–560. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3395363.3404366

56. Huang, Y., Jiang, B., Chan, W. K. (2020). EOSFuzzer: Fuzzing EOSIO smart contracts for vulnerability detection. arXiv preprint arXiv:2007.14903.

57. Torres, C. F., Iannillo, A. K., Gervais, A., State, R. (2021). ConFuzzius: A data dependency-aware hybrid fuzzer for smart contracts. arXiv preprint arXiv:2005.12156.

58. Wüstholz, V., Christakis, M. (2020). Harvey: A greybox fuzzer for smart contracts. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1398–1409. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3368089.3417064

59. He, J., Balunović, M., Ambroladze, N., Tsankov, P., Vechev, M. (2019). Learning to fuzz from symbolic execution with application to smart contracts. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 531–548. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3319535.3363230

60. Krupp, J., Rossow, C. (2018). TEETHER: Gnawing at ethereum to automatically exploit smart contracts. *Proceedings of the 27th USENIX Conference on Security Symposium*, pp. 1317–1333. USA, USENIX Association.

61. Luu, L., Chu, D. H., Olickel, H., Saxena, P., Hobor, A. (2016). Making smart contracts smarter. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 254–269. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/2976749.2978309

62. Mueller, B. (2017). Mythril. https://github.com/Consensys/mythril (accessed on 22/11/2023)

63. Torres, C. F., Schütte, J., State, R. (2018). Osiris: Hunting for integer bugs in ethereum smart contracts. *Proceedings of the 34th Annual Computer Security Applications Conference, in ACSAC '18*, pp. 664–676. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3274694.3274737

64. Yang, Z., Liu, H., Li, Y., Zheng, H., Wang, L. et al. (2020). Seraph: Enabling cross-platform security analysis for EVM and WASM smart contracts. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, pp. 21–24. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3377812.3382157

65. Liu, Y., Li, Y., Lin S. W., Artho, C. (2022). Finding permission bugs in smart contracts with role mining. *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 716–727. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3533767.3534372

66. Wang, D., Jiang, B., Chan, W. K. (2020). WANA: Symbolic execution of Wasm Bytecode for cross-platform smart contract vulnerability detection. arXiv preprint arXiv:2007.15510.

67. Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G. et al. (2016). Formal verification of smart contracts: Short paper. *Proceedings of the 2016 ACM Workshop on Programming*

Languages and Analysis for Security, pp. 91–96. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/2993600.2993611

68. Hirai, Y. (2016). Formal verification of deed contract in Ethereum name service. https://www.coinresear.ch/snapshots/Hirai2016 (accessed on 22/11/2023)

69. Kalra, S., Goel, S., Dhawan, M., Sharma, S. (2018). ZEUS: Analyzing safety of smart contracts. *25th Annual Network and Distributed System Security Symposium*, San Diego, California, USA, NDSS. http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_09-1_Kalra_paper.pdf

70. Momeni, P., Wang, Y., Samavi, R. (2019). Machine learning model for smart contracts security analysis. *2019 17th International Conference on Privacy, Security and Trust (PST)*, pp. 1–6. Fredericton, NB, Canada. https://doi.org/10.1109/PST47121.2019.8949045

71. Gao, Z., Jayasundara, V., Jiang, L., Xia, X., Lo, D. et al. (2019). SmartEmbed: A tool for clone and bug detection in smart contracts through structural code embedding. *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 394–397. Cleveland, OH, USA. https://doi.org/10.1109/ICSME.2019.00067

72. Alex Rea, A. R. (2016). Code Coverage for Solidity. https://blog.colony.io/code-coverage-for-solidity-eecfa88668c2/ (accessed on 22/11/2023)

73. Chen, T., Li, Z., Zhang, Y., Luo, X., Chen, A. et al. (2019). DataEther: Data exploration framework for ethereum. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1369–1380. Dallas, TX, USA. https://doi.org/10.1109/ICDCS.2019.00137

74. Zhang, M., Zhang, X., Zhang, Y., Lin, Z. (2020). TXSPECTOR: Uncovering attacks in ethereum from transactions. *Proceedings of the 29th USENIX Conference on Security Symposium*, USA, USENIX Association.

75. Wu, L., Wu, S., Zhou, Y., Li, R., Wang, Z. et al. (2020). EthScope: A transaction-centric security analytics framework to detect malicious smart contracts on ethereum. arXiv preprint arXiv:2005.08278v1.

76. Zheng, P., Zheng, Z., Wu, J., Dai, H. N. (2020). Xblock-ETH: Extracting and exploring blockchain data from Ethereum. *IEEE Open Journal of the Computer Society, 1,* 95–106. https://doi.org/10.1109/OJCS.2020.2990458

77. Transaction (2021). https://bscscan.com/tx/0x8769f7ee2c8e010fc8791bd0e42569b7ced9b2f67b721e6f0c6a6435b4d6670f (accessed on 22/11/2023)

78. Transaction (2021). https://bscscan.com/tx/0x903ae34f48d4e00da8d7ca5dfad26f8f37e80cde2156907580cf551a63317f76 (accessed on 22/11/2023)

79. Transaction (2021). https://bscscan.com/tx/0x1b698231965b72f64d55c561634600b087154f71bc73fc775622a45112a94a77 (accessed on 22/11/2023)

80. Transaction (2021). https://bscscan.com/tx/0x701a308fba23f9b328d2cdb6c7b245f6c3063a510e0d5bc21d2477c9084f93e0 (accessed on 22/11/2023)

81. Jamie, C. (2021). BSC Flash Loan Attack: PancakeBunny. https://www.coindesk.com/markets/2021/05/20/flash-loan-attack-causes-defi-token-bunny-to-crash-over-95/ (accessed on 22/11/2023)

82. Zhang, F., Cecchetti, E., Croman, K., Juels, A., Shi, E. (2016). Town crier: An authenticated data feed for smart contracts. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 270–282. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/2976749.2978326

83. Southhurst, J. (2020). Chainlink exploits lead to ETH losses–again. https://coingeek.com/chainlink-exploits-lead-to-eth-losses-again/ (accessed on 22/11/2023)

84. Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X. et al. (2019). Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. arXiv preprint arXiv:1904.05234.

85. Qin, K., Zhou, L., Gervais, A. (2021). Quantifying blockchain extractable value: How dark is the forest? arXiv preprint arXiv:2101.05511.

86. Zhou, L., Qin, K., Torres, C. F., Le, D. V., Gervais, A. (2021). High-frequency trading on decentralized on-chain exchanges. *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 428–445. https://doi.org/10.1109/SP40001.2021.00027

87. Zhou, L., Qin, K., Cully, A., Livshits, B., Gervais, A. (2021). On the just-in-time discovery of profit-generating transactions in defi protocols. *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 919–936. https://doi.org/10.1109/SP40001.2021.00113

88. Wang, Y., Zuest, P., Yao, Y., Lu, Z., Wattenhofer, R. (2022). Impact and user perception of sandwich attacks in the DeFi ecosystem. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–15. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3491102.3517585

89. Uniswap (2022). Introducing Uniswap V3. https://uniswap.org/blog/uniswap-v3/ (accessed on 22/11/2023)

90. Flashbots (2022). Welcome to Flashbots. https://docs.flashbots.net/ (accessed on 22/11/2023)

91. Akhtar, T. (2021). Belt Finance to compensate users following $6.23 M attack. https://www.coindesk.com/business/2021/06/02/belt-finance-to-compensate-users-following-623m-attack/ (accessed on 22/11/2023)

92. SlowMist (2021). Value DeFi vSwap Module Hack Analysis. https://slowmist.medium.com/slowmist-value-defi-vswap-module-hack-analysis-64e8909ef6a2 (accessed on 22/11/2023)

93. Merlin Lab (2022). Incident Case Analysis. https://medium.com/valixconsulting/merlin-lab-incident-case-analysis-d023d4205bd7 (accessed on 22/11/2023)

94. Wang, S., Wu, C. Liang, Y. Hsieh, L. Hsiao, H. (2021). ProMutator: Detecting vulnerable price oracles in DeFi by mutated transactions. *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 380–385. Los Alamitos, CA, USA, IEEE Computer Society. https://doi.org/10.1109/EuroSPW54576.2021.00047

95. Qin, K., Zhou, L., Livshits, B., Gervais, A. (2020). Attacking the DeFi ecosystem with flash loans for fun and profit. arXiv preprint arXiv:2003.03810.

96. PancakeBunny (2023). https://pancakebunny.finance/ (accessed on 22/11/2023)

97. Zhang, Z., Zhang, B., Xu, W., Lin, Z. (2023). Demystifying Exploitable Bugs in Smart Contracts. *Proceedings of the 45th International Conference on Software Engineering*, pp. 615–627. Melbourne, Victoria, Australia, IEEE Press. https://doi.org/10.1109/ICSE48619.2023.00061

98. Lin, S., Sun, X., Yao, J., Gu, R. (2021). SciviK: A versatile framework for specifying and verifying smart contracts. arXiv Preprint arXiv:2103.02209.

99. Wu, S., Wang, D., He, J., Zhou, Y., Wu, L. et al. (2021). DeFiRanger: Detecting price manipulation attacks on DeFi applications. arXiv preprint arXiv:2104.15068.

100. Wang, B., Liu, H., Liu, C., Yang, Z., Ren, Q. et al. (2021). BLOCKEYE: Hunting for DeFi attacks on blockchain. *IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pp. 17–20. Madrid, Spain. https://doi.org/10.1109/ICSE-Companion52605.2021.00025

101. Wang, D., Wu, S., Lin, Z., Yuan, X., Zhou, Y. et al. (2021). Towards a first step to understand flash loan and its applications in DeFi ecosystem. *Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Computing*, pp. 23–28. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3457977.3460301

102. Cao, Y., Zou, C., Cheng, X. (2021). Flashot: A snapshot of flash loan attack on DeFi ecosystem. arXiv preprint arXiv:2102.00626.

103. Block #9484688 (2020). https://etherscan.io/block/9484688 (accessed on 22/11/2023)

104. Transaction (2020). https://etherscan.io/tx/0x4f4fc1aa665264ee53180b6da4e2195d81fad4530e55190932461f8b8681a8a6 (accessed on 22/11/2023)

105. The Governance Security Module (GSM) (2019). https://blog.makerdao.com/governance-security-module-gsm/ (accessed on 22/11/2023)

106. Raul Amoros (2017). The Bitcoin Wealth Distribution. https://howmuch.net/articles/bitcoin-wealth-distribution (accessed on 22/11/2023)

107. Beck, R., Müller-Bloch, C., King, J. L. (2018). Governance in the blockchain economy: A framework and research agenda. *Journal of the Association for Information Systems, 19(10)*.

108. Mehar, M., Shier, C., Giambattista, A., Gong, E., Fletcher, G. et al. (2019). Understanding a revolutionary and flawed grand experiment in blockchain: The DAO attack. *Journal of Cases on Information Technology (JCIT), 21(1),* 19–32. https://doi.org/10.4018/JCIT.2019010102

109. BugBusters (2023). Interested calculated is amplified by multiple of 1000 in_debt_interest_since_last_update. https://github.com/sherlock-audit/2023-06-unstoppable-judging/issues/191 (accessed on 22/11/2023)

110. Heiko Fisch (2021). MSDController._withdrawReserves does not update interest before withdrawal. https://solodit.xyz/issues/msdcontroller_withdrawreserves-does-not-update-interest-before-withdrawal-consensys-dforce-lending-protocol-review-markdown (accessed on 22/11/2023)

111. 0x52 (2022). Vault_Base_ERC20#_updateVirtualPrice calculates interest incorrectly if updated frequently. https://solodit.xyz/issues/m-10-vault_base_erc20_updatevirtualprice-calculates-interest-incorrectly-if-updated-frequently-sherlock-isomorph-isomorph-git (accessed on 22/11/2023)

112. Leastwood (2022). Users Will Lose Rewards if the Shelter Mechanism is Enacted Before A Recent Checkpoint. https://solodit.xyz/issues/m-10-users-will-lose-rewards-if-the-shelter-mechanism-is-enacted-before-a-recent-checkpoint-code4rena-concur-finance-concur-finance-contest-git (accessed on 22/11/2023)

113. Unforgiven, 0xDjango (2022). User can steal all rewards due to checkpoint after transfer. https://solodit.xyz/issues/h-01-user-can-steal-all-rewards-due-to-checkpoint-after-transfer-code4rena-backd-backd-contest-git (accessed on 22/11/2023)

114. cmichel (2021). Pools can be created without initial liquidity. https://solodit.xyz/issues/m-05-pools-can-be-created-without-initial-liquidity-code4rena-spartan-protocol-spartan-protocol-contest-git (accessed on 22/11/2023)

115. Chaliasos, S., Charalambous, M. A., Zhou, L., Galanopoulou, R., Gervais, A. et al. (2023). Smart contract and defi security: Insights from tool evaluations and practitioner surveys. arXiv preprint arXiv:2304.02981.

116. Surameery, N. M. S., Shakor, M. Y. (2023). Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC), 3(1),* 17–22. https://doi.org/10.55529/ijitc.31.17.22

117. Chen, E., Huang, R., Chen, H. S., Tseng, Y. H., Li, L. Y. (2023). GPTutor: A ChatGPT-powered programming tool for code explanation. arXiv preprint arXiv:2305.01863.

118. Li, W., Bu, J., Li, X., Chen, X. (2022). Security analysis of DeFi: Vulnerabilities, attacks and advances. *2022 IEEE International Conference on Blockchain (Blockchain)*, pp. 488–493. https://doi.org/10.1109/Blockchain55522.2022.00075

119. Wang, B., Yuan, X., Duan, L., Ma, H., Su, C. et al. (2022). Defiscanner: Spotting DeFi attacks exploiting logic vulnerabilities on blockchain. *IEEE Transactions on Computational Social Systems*, pp. 1–12. https://doi.org/10.1109/TCSS.2022.3228122

120. Babel, K., Daian, P., Kelkar, M., Juels, A. et al. (2023). Clockwork finance: Automated analysis of economic security in smart contracts. *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2499–2516. IEEE. https://doi.org/10.1109/SP46215.2023.10179346

121. ZK-Rollups (2023). https://ethereum.org/en/developers/docs/scaling/zk-rollups/ (accessed on 22/11/2023)

122. zkSync (2023). https://zksync.io/ (accessed on 22/11/2023)

123. zkEVM FAQ (2023). https://docs.zksync.io/zkevm/ (accessed on 22/11/2023)

124. Werner, S., Perez, D., Gudgeon, L., Sok, A., Klages-Mundt, D. et al. (2022). SoK: Decentralized finance (DeFi). *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, pp. 30–46. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3558535.3559780

125. Xu, J., Paruch, K., Cousaert, S., Sok, S. Feng, Y. et al. (2023). SoK: Decentralized exchanges (DEX) with automated market maker (AMM) protocols. *ACM Computing Surveys, 55(11),* 1–50. https://doi.org/10.1145/3570639

126. Amler, H., Eckey, L., Faust, S., Kaiser, M. Sandner, P. et al. (2021). DeFi-ning DeFi: Challenges & pathway. *2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pp. 181–184. Paris, France. https://doi.org/10.1109/BRAINS52497.2021.9569795