FAIRYPROOF

Review of Blockchain Security in Year 2021

Presented by Fairyproof
January, 2022

OVERVIEW

Year 2021 was a hot year for cryptos.

According to coinmarketcap.com's data, Bitcoin's price rose from $28994.01 on Jan 1, 2021 to $46306.45 on December 31, 2021 and hit an all time high of $68789.63 on November 10, 2021, Ethereum's price rose from $737.71 on January 1, 2021 to $3682.63 on December 31, 2021 and hit an all time high of $4891.7 on November 16, 2021, and the total crypto currency's market cap rose from $773 billion on January 1, 2021 to $2256 billion on December 31, 2021

Not only the market was hot but also the whole crypto industry welcomed extremely huge development and growth. Numerous innovations and applications disrupted our understanding of technologies, businesses and culture. We saw Ethereum's layer 2 solutions boom[1][2], which we hadn't expected to happen this fast. We saw DeFi 2.0 reshape conventional DeFi business models[3]. We saw NFT based JPEGs become a symbol of social status in metaverse[4][5]. We saw blockchain games have groundbreaking play-to-earn models[6][7].

A coin has two sides. On one hand we saw and experienced the boom of the crypto ecosystem and on the other hand we experienced its dark side as well. Security issues were definitely its dark side. There were 189 publicly reported security incidents taking place in 2021. And more than 7.6 billion USDs of crypto assets were exploited.

These exploits not only caused huge losses to crypto holders but also extremely hindered the whole ecosystem's long-term development.

We studied these 189 incidents and compose our findings, analysis and best practices in this report.

BACKGROUND

Before we proceed, it is necessary to introduce some basic concepts that we frequently mention in this report.

WHAT IS A BLOCKCHAIN

A blockchain is a growing list of data records, called blocks that are linked together using cryptography[8][9][10][11]. Each block except the genesis one contains a cryptographic hash of the previous block, a timestamp, and transaction data.[12] These blocks together form a chain that grows permanently. In general a blockchain is irreversible and tamper-proof because the data in any given block cannot be modified without modifying all subsequent blocks.

Bitcoin was the first implementation of the blockchain technology. It opened the infinite possibilities of blockchain ecosystem's development. After Bitcoin, various blockchain implementations have emerged.

PERMISSIONLESS BLOCKCHAINS VS PERMISSIONED BLOCKCHAINS

Basically all the existing blockchain implementations can be categorized into two categories: permissionless blockchains and permissioned blockchains.

A permissionless blockchain, also referred to as a public blockchain, is an open network which allows everyone to participate in its consensus process, data and block validations without access control[13]. All the nodes that participate in the network activities are trustless. Bitcoin was the first permissionless blockchain.

A permissioned blockchain, is a closed network which only grants access to designated parties that are members of a consortium, an organization or a company etc[14]. These parties participate in its consensus process, data and block validations.

Because permissionless blockchains allow anyone to interact with them and participate in their activities, they gain huge traction of passionate engineers all over the globe to work in and build their ecosystems.

In addition, to maintain a permissionless blockchain's autonomous operation, a mining mechanism to reward the nodes with crypto currencies, that successfully generate valid blocks is widely adopted such that actors would be incentivize to participate in its consensus.

Therefore permissionless blockchains grow much faster, more active and prosperous than permissioned blockchain ecosystems.

Because permissionless blockchains allow anyone including malicious actors to participate in their activities and quite often these malicious actors would obtain huge economic benefits from attacks by exploiting crypto currencies, permissionless blockchains are much more likely to suffer from attacks both from within the blockchains and outside the blockchains.

These factors all together add tremendous complexity in handling security issues of a permissionless blockchain.

WHAT IS A DAPP

A decentralized application is an application implemented in one or multiple smart contracts that run on a blockchain[15][16]. If the blockchain that a DApp runs on is a permissionless blockchain the DApp is able to run or function autonomously without a centralized entity in control.

DApps of this kind are often open source, transparent, and allow everyone to permissionlessly interact with them. In order to both attract as many users as possible to use a DApp and maintain the DApp's long term development, the team behind it, in general, would launch crypto tokens for the DApp and would reward the tokens to its users and team members. These lucrative tokens are always the targets of cyber hackers.

These characteristics make DApps as prone to be attacked as permissionlesss blockchains as well.

FOCUS OF THIS REPORT

Cyber attacks against permissionless blockchains, DApps and centralized entities that do crypto related businesses have been a wide spread issue and a hot research topic in the current blockchain industry. And the attackers in most of these attacks eventually exchange the exploited crypto assets to fiat-pegged stablecoins or fiat currencies for profits.

In this report we will list our statistics collected from typical security incidents that happened in the blockchain industry in Year 2021, give an in-depth analysis of their root causes and present our recommended best practices.

STATISTICS AND ANALYSIS OF SECURITY INCIDENTS OF YEAR 2021

We studied 189 publicly reported security incidents that took place in 2021 and present our statistics and analysis based on the suffering targets and root causes.

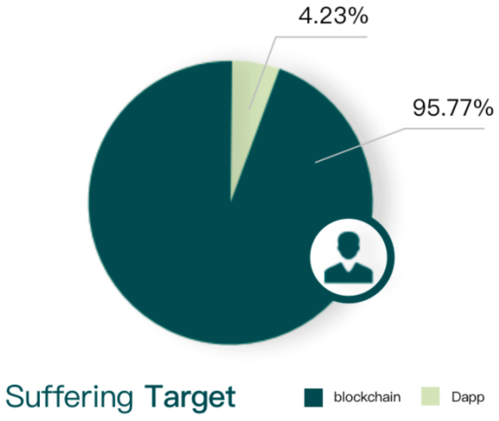
INCIDENTS CATEGORIZED BY SUFFERING TARGET

Based on the type of suffering target, our researched incidents can be categorized into two types: blockchains and DApps.

A blockchain related incident is one in which a blockchain is attacked by malicious actors from either inside or outside or both such that its operation would go out of order, or a blockchain fails to work properly due to either software issues or hardware issues or both such that attackers would be able to exploit the consensus for profits.

A DApp related incident is one in which a DApp is attacked or a DApp's daily operation goes out of order such that attackers would be able to exploit users crypto assets that are under custody of the DApp.

There were 189 incidents in total and here is a figure that shows the percentages of DApp related incidents and blockchain related incidents respectively.



The above figure shows that the number of DApp related incidents accounted for more than 95% of the total incidents. Among 189 incidents, 181 were DApp related and only 8 incidents were blockchain related.

BLOCKCHAIN RELATED INCIDENTS

Among the incidents that happened to blockchains, we further categorize them into three sub-categories: blockchain mainnets, side chains and layer 2 solutions.

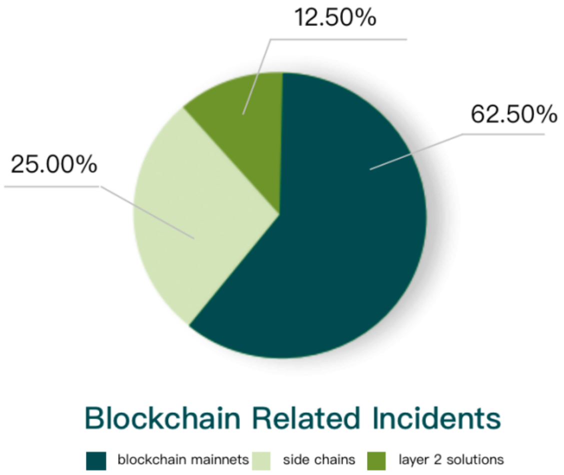
A blockchain mainnet, also known as lay 1 is an independent blockchain that has its own network with its own protocol, consensus and validators. A blockchain mainnet can validate the transactions, data and blocks that are generated in its network by its own validators and reach a finality. Bitcoin and Ethereum are typical blockchain mainnets.

A side chain is a separate blockchain which runs in parallel and independently to a blockchain mainnet. It has its own network, consensus and validators. It is connected to a blockchain mainnet e.g. by a two-way peg[17].

A layer 2 solution refers to a protocol or network that relies to a blockchain as its base layer for security and finality[18]. The base layer is also referred to as a layer 1. A layer 2's main purpose is to solve the base layer's scalability issues. A layer 2 solution processes transactions much faster and costs much less compared to its base layer. The Ethereum blockchain, for instance, saw a huge surging in the growth and development of layer 2 solutions pertaining to it in 2021.

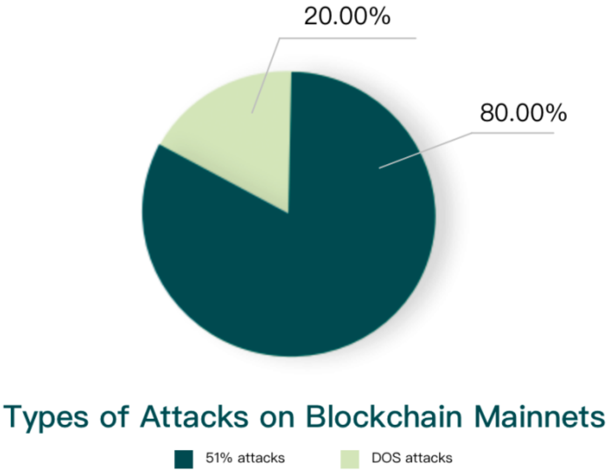
Both side chains and layer 2 solutions are to solve blockchain mainnets' s scalability issues. The significant difference between a side chain and a layer 2 solution is that a side chain doesn't necessarily rely on its blockchain mainnet for security or finality but a layer 2 solution does.

There were 8 blockchain related incidents in total in 2021 and here is a figure that shows the percentages of blockchain mainnets related incidents, side chain related incidents and layer 2 related incidents respectively.



The figure shows that the number of blockchain mainnet related incidents accounted for 62.5% of the total incidents. Among 8 incidents, 5 were blockchain mainnet related including Solana[19], ETC[20], BSV[21], Verge[22] and Firo[23], 2 was side chain related including Polygon[24] and Liquid Network[25], and 1 was layer 2 solution related including Arbitrum One[26].

Among those 5 blockchain mainnet related incidents, ETC, BSV, Verge and Firo all suffered from 51% attacks[27]. The root cause was that these mainnets' hash rates were relatively low and it was easy for hackers to rent hash power to launch attacks. Apart from that, Solana suffered from DOS attacks[28].



DAPP RELATED INCIDENTS

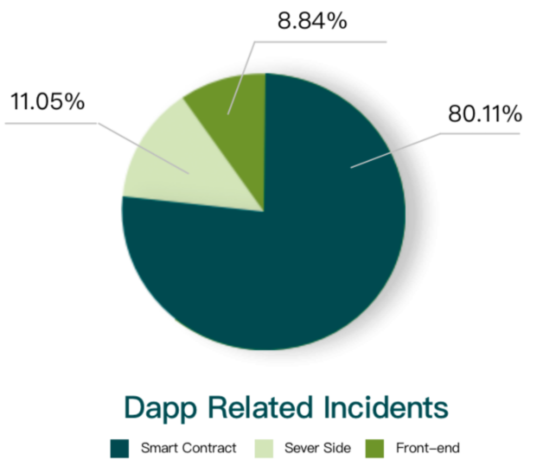
Among the incidents that happened to DApps we further categorize them into three sub-categories:DApp's front-end, DApp's server side and DApp's smart contract(s).

DApp's front-end related incidents are those in which vulnerabilities in conventional client side are exploited such that users' account information, personal details would be stolen and used to exploit users' crypto assets.

DApp's server side related incidents are those in which vulnerabilities in conventional server side are exploited such that communication between off-chain and on-chain would be hijacked and users' crypto assets would be exploited.

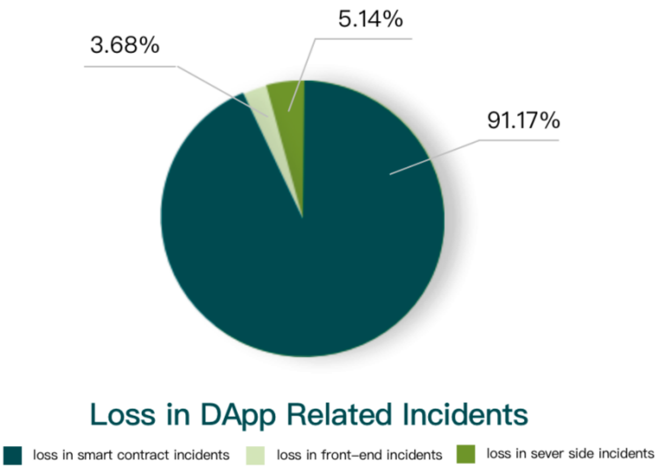
Smart contract related incidents are those in which vulnerabilities in smart contract design or implementation are exploited such that users' crypto assets in the smart contracts would be exploited.

There were 181 incidents in total and here is a figure that shows the percentages of front-end related incidents, server side related incidents and smart contract related incidents respectively.



The above figure shows that the number of smart contract related incidents accounted for 80.11% of the total incidents. Among 181 DApp related incidents, 16 were front-end related, 20 were server side related and 145 were smart contract related.

We went further and studied the loss in these three sub-categories and got the following figure:

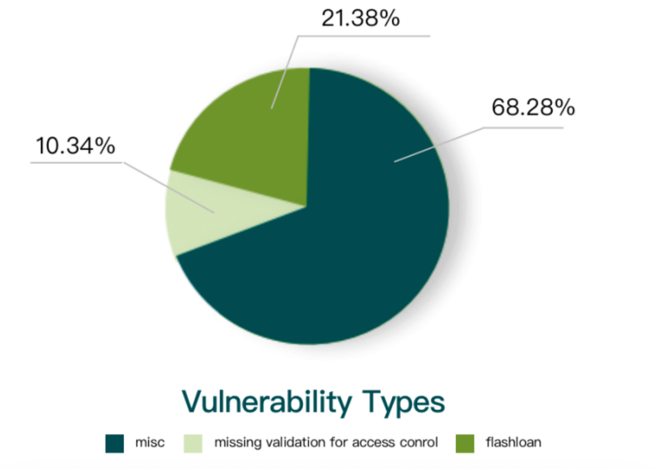


In our statistics the loss in front-end related incidents was $280 million, the loss in server side related incidents was $391 million and the loss in smart contract related incidents was $6.93 billion.

Although the number of front-end related incidents didn't account for a great percentage, there were cases each of which suffered a huge loss. For example Vulcan Forged[29] lossed $140 million, BadgerDAO[30] lossed $120 million and Farmer World[31] lossed $15.70 million.

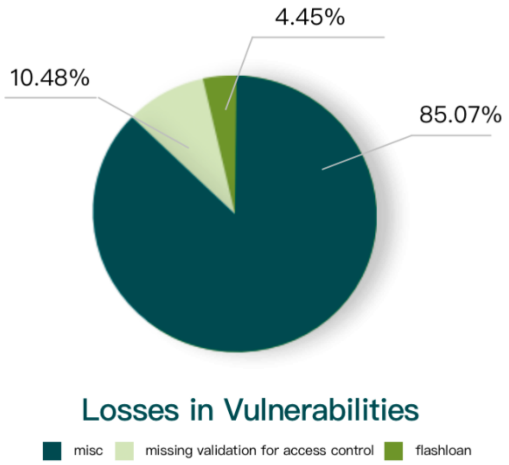
Clearly smart contract related incidents were the biggest issue. Typical vulnerabilities we found pertaining to smart contracts in 2021 included flashloans[32], missing validation for access control, incorrect calculation of token precision, interger overflow/underflow, re-entracy attacks, incorrect swap algorithms, fake token deposit, double-spend, governance attacks etc.

We studied the number of incidents based on the vulnerability types and got the following figure:



The figure shows that the number of incidents that suffered from flashloan attacks ranked No. 1 among all smart contract related incidents and it was followed by the number of incidents that suffered from missing validation for access control. A total of 31 incidents suffered from flashloan attacks and 15 suffered from attacks from missing validation for access control.

After we studied the loss in each of the vulnerability types we got the following figure:



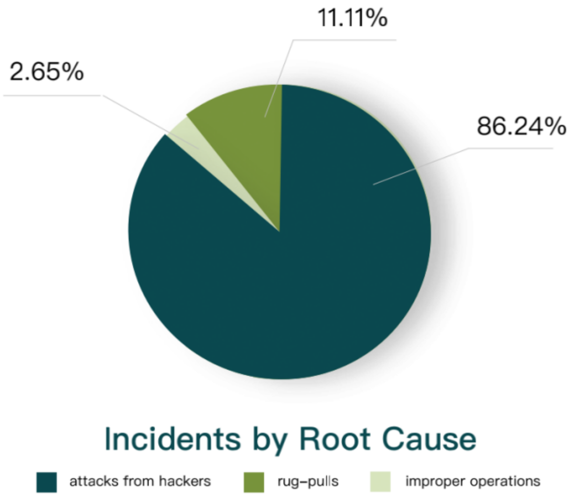
The interesting thing that the figure shows is that although the number of incidents that suffered from missing validation for access control was far less than the number of incidents that suffered from flashloan attacks, the amount of loss of the former was far more than the latter.

In 2021, flashloans had become a frequently used tool to attack DeFi applications and quite a few popular DeFi applications such as Cream Finance[33], Spartan Protocol[34], YFI[35], Indexed Finance[36] suffered from flashloan attacks. Some suffered even more than once. For example AutoShark[37] was attacked three times by flashloans, Pancake Bunny[38], BurgerSwap[39] and Cream Finance each was attacked twice by flashloans.

INCIDENTS CATEGORIZED BY ROOT CAUSE

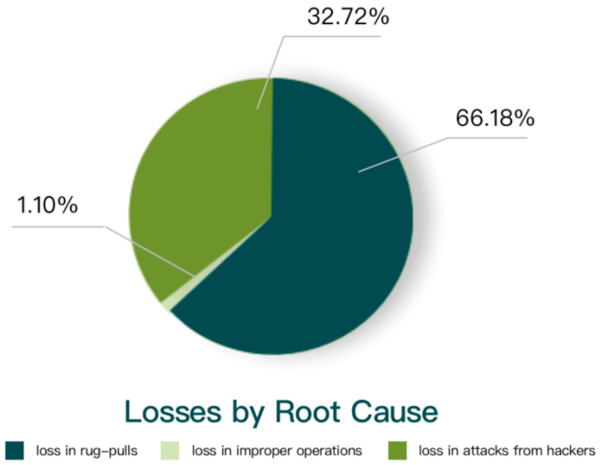
Based on the root cause we categorize these incidents into three categories: attacks from hackers, improper operations and rug-pulls.

We studied the incidents and got the following figure.



The above figure shows that the number of attacks from hackers accounted for 86.24% of the total incidents. Among the total 189 incidents, 163 suffered from attacks from hackers, 5 suffered from improper operations and 21 suffered from rug-pulls. Among the total 189 incidents, the number of incidents that suffered from attacks from hackers was 163, the number of incidents that suffered from improper operations was 5 and the number of incidents that suffered from rug-pulls was 21.

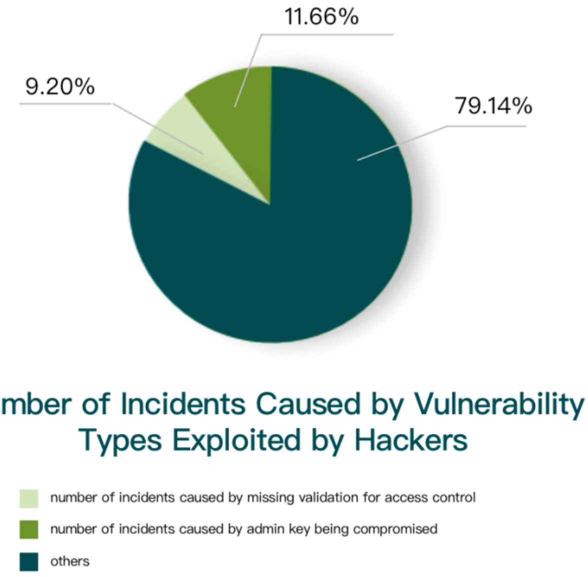
We studied the losses based on the root causes and got the following figure:



The above figure shows that the loss in the incidents that suffered from rug-pulls accounted for 66.18% of the total loss. That loss far surpassed the the loss caused by attacks from hackers and the loss caused by improper operations. Among the total loss of $7.6 billion, the loss in the incidents that suffered from attacks from hackers was $2.49 billion, the loss in the incidents that suffered from improper operations was $83.54 million and the loss in the incidents that suffered from rug-pulls was $5.03 billion.

ATTACKS FROM HACKERS

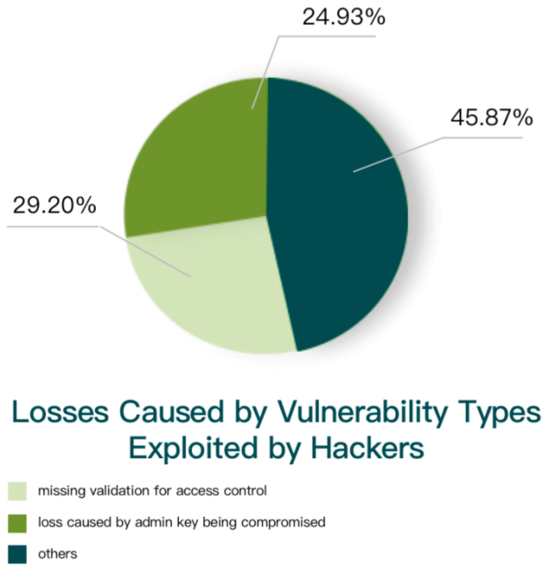
We studied the types of vulnerability that were exploited by hackers and got the following figure:



Number of Incidents Caused by Vulnerability Types Exploited by Hackers

The above figure shows that neither the number of incidents caused by admin key being compromised nor the number of incidents caused by missing validation for access control accounted for a significant percentage. The former accounted for 11.66% and the latter accounted for 9.2%.

After we studied the loss in each of them we got the following figure:



Losses Caused by Vulnerability Types Exploited by Hackers

In contrast to the previous figure, this figure shows that both the loss in the incidents caused by admin key being compromised and the loss in the incidents caused by missing validation for access control accounted for a significant percentage respectively. The former accounted for 24.93% and the latter accounted for 29.2%.

Those that suffered huge losses from admin key being compromised included multiple centralized exchanges. For example BitMEX[40] loss $150 million, Liquid[41] loss $91 million, AscendEX[42] loss $77 million, HitBtc[43] loss $40 million and Bilaxy[44] loss $21.70 million.

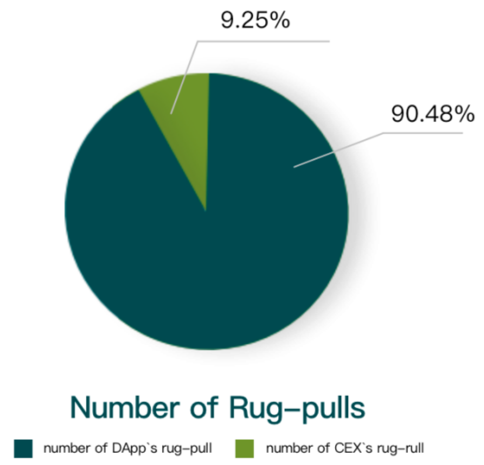
It is worth noting that the incidents discussed here included the incidents related to smart contracts, front-end and server side. If we consider only the incidents related to front-end and server side, admin key being compromised was the biggest risk.

RUG-PULLS

Rug-pulls happened to various DApps in 2021 covering DeFi applications and centralized exchanges.

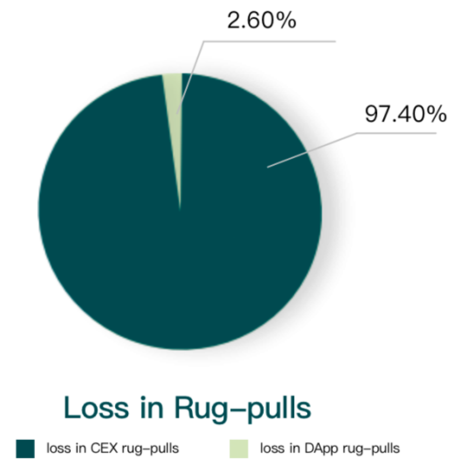
Among all the 21 rug pulls, 2 were centralized exchanges and the rest 19 cases were DApps.

We studied these cases and got the following figure:



The above figure shows that CEX’s rug-pulls only accounted for 9.52% of the total cases while DApp’s rug-pulls accounted for 90.48%.

We studied the losses in the two rug-pulls and got the following figure:



The above figure shows that although the number of CEX’s rug-pulls was tiny compared to the DApp’s rug-pulls, its loss accounted for 97.4% of the total loss.

IMPROPER OPERATIONS

There were 5 incidents caused by improper operations. All of them were DApps, more specifically, they were all DeFi applications including popular ones such as Compound[45] and dYdX[46]. The total loss in these incidents was $83.54 million.

RESEARCH FINDINGS

With regard to attacks on blockchain mainnets, 51% attacks were still the most widely adopted attacks, which accounted for 80% of the total attacks against mainnets in 2021.

Besides commonly known incidents with blockchain mainnets, incidents pertaining to layer 2 solutions emerged but the number of cases was still less than that of side chains in 2021.

With regard to the suffering types, attacks from hackers accounted for nearly 90% of the total incidents. Attacks from hackers were still the main threat to the whole crypto industry.

The three types of incidents that a DApp suffered from were front-end related incidents, server-side related incidents and smart contract related incidents. Based on the number of incidents and the loss in the incidents, smart contract related incidents accounted for a far more percentage than the other two in 2021. With regard to the number of incidents, smart contract related incidents accounted for 80.11% of the total, followed by the server-side related incidents which accounted for 11.05% and then by the front-end related incidents which accounted for 8.84%. With regard to the loss in the incidents, smart contract related incidents accounted for 91.17% of the total, followed by the server-side related incidents which accounted for 5.14% and then by the front-end related incidents which accounted for 3.68%.

Front-end incidents didn't gain attention in the past but they caused huge losses to some projects in 2021. There were two cases each of which suffered a loss of more than $100 millon. The victims were Vulcan Forged and BadgerDAO losing $140 million and $120 million respectively.

With regard to the incidents related to front-end and server side, admin key being compromised was definitely the biggest risk in 2021.

After we studied the smart contract related incidents we found out that the number of incidents that suffered from flashloan attacks was far more than any other root causes and ranked No 1, and the number of incidents that suffered from missing validation for access control ranked No 2. But in contrast, the amount of loss in incidents of the latter was far more than any other and ranked No 1, and the amount of loss in the former ranked No. 2.

With regard to the root causes of the overall incidents, although the number of incidents that suffered from hackers was far more than the number of incidents that suffered from rug-pulls. However the loss caused by the latter far surpassed the loss caused by the former.

The rug-pulls covered both DApps and centralized exchanges in 2021. And the number of DApp's rug-pulls was much greater than that of CEX's rug-pulls. However the loss of the latter far exceeded the loss of the former. It indicates that CEX's rug-pulls were still the main risk crypto holders should face and be cautious about.

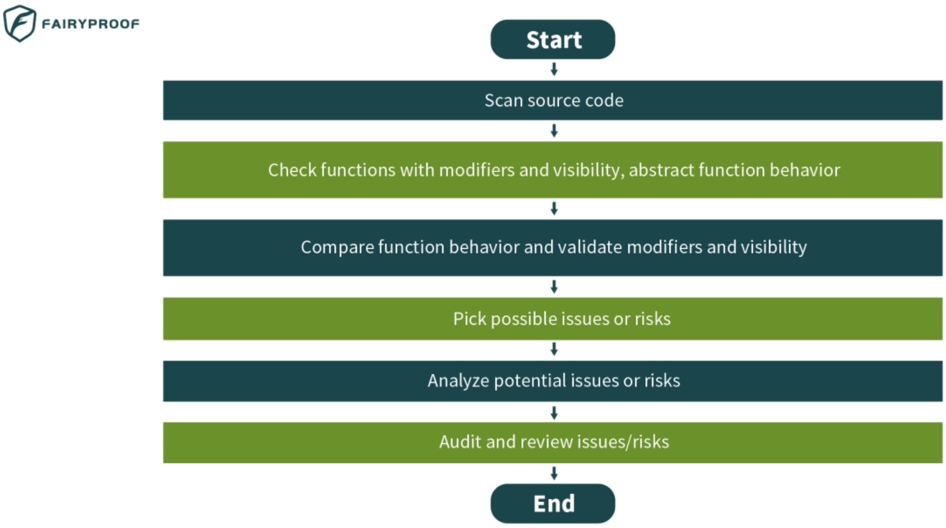
FAIRYPROOF’S SOLUTIONS

Based on our research and analysis, the biggest challenges in security were flashloan attacks, missing validation for access control and rug-pulls. These threats widely existed in smart contracts.

In this section we will introduce Fairyproof’s solutions to these threats.

VULNERABILITY DETECTION SYSTEM

In order to automatically detect a project’s vulnerabilities, especially those related to flashloans and validation for access control, our system follows the steps below:


FAIRYPROOF
Start
Scan source code
Check functions with modifiers and visibility, abstract function behavior
Compare function behavior and validate modifiers and visibility
Pick possible issues or risks
Analyze potential issues or risks
Audit and review issues/risks
End

Step 1: it scans the whole source code.

Step 2: it checks functions with modifiers and functions’ visibility, and abstracts every function’s behavior.

Step 3: it compares every function’s behavior with the behaviors in our database and validates whether or not that function’s modifier or visibility is appropriate.

Step 4: for each function whose behavior has been abstracted, it is compared with typical threats that are saved in our threat database. Each typical threat in the database has a pending checklist. If an item matches the key characteristics of a typical threat it will be put into the checklist of that threat.

Step 5: every item in these checklists is analyzed with our developed behavior curve-fitting algorithm which is a machine learning algorithm to validate whether or not it is a potential threat.

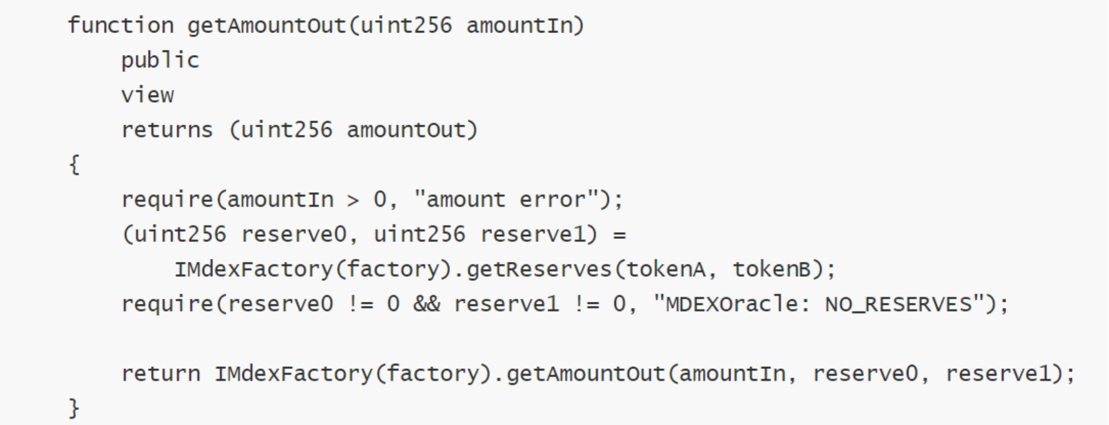
Step 6: if a function’s behavior is determined to have potential issues or risks in Step 5, it will be marked and sent to our engineers.

Step 7: our engineers will review and audit all the functions sent by Step 6.

This flow greatly automates an audit process, reduces tedious manual work and speed up our audit work.

We utilized this tool set to discover various typical issues or risks including the biggest threats: flashloans and missing validation for access control.

Here is an an example of flashloan attack in our audit. In the following code section, the getAmountOut() function got a reserve value from a trading pair in a decentralized exchange and calculated the price of UBT. This would possibly be attacked by flashloans/manipulation of oracles

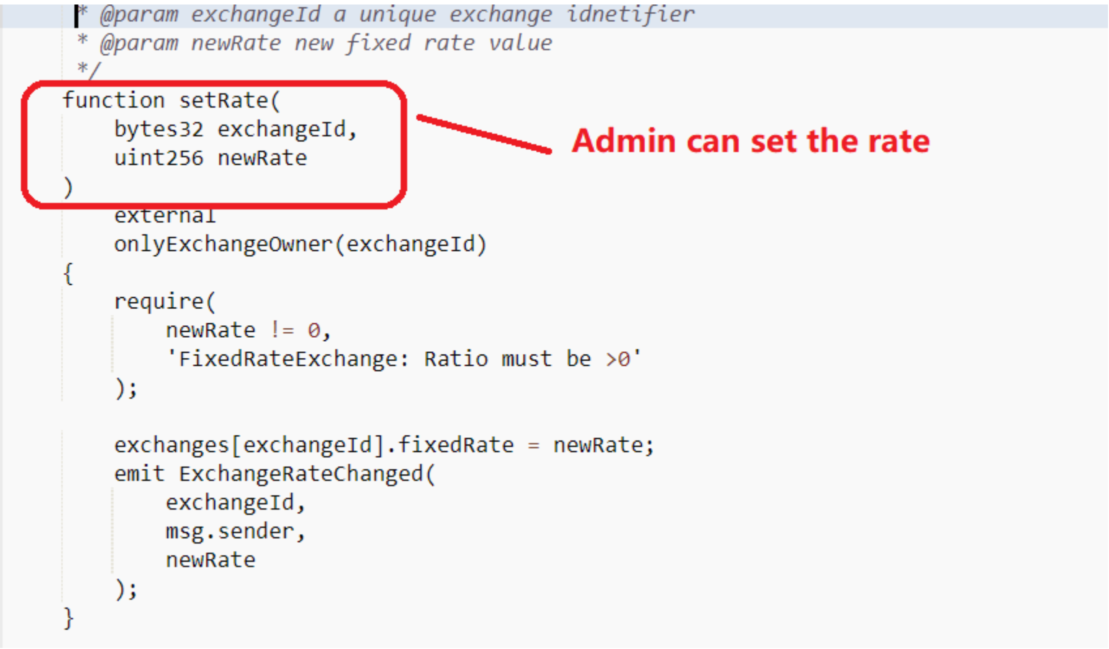


```
function getAmountOut(uint256 amountIn)
    public
    view
    returns (uint256 amountOut)
{
    require(amountIn > 0, "amount error");
    (uint256 reserve0, uint256 reserve1) =
        IMdexFactory(factory).getReserves(tokenA, tokenB);
    require(reserve0 != 0 && reserve1 != 0, "MDEXOracle: NO_RESERVES");

    return IMdexFactory(factory).getAmountOut(amountIn, reserve0, reserve1);
}
```

We suggested to use a safe oracle to get its price.

Here is an example of missing validation for access control in our audit. In the following code sections, the admin could set the rate by calling the “setRate” function and this would cause the swap transaction being front-run.

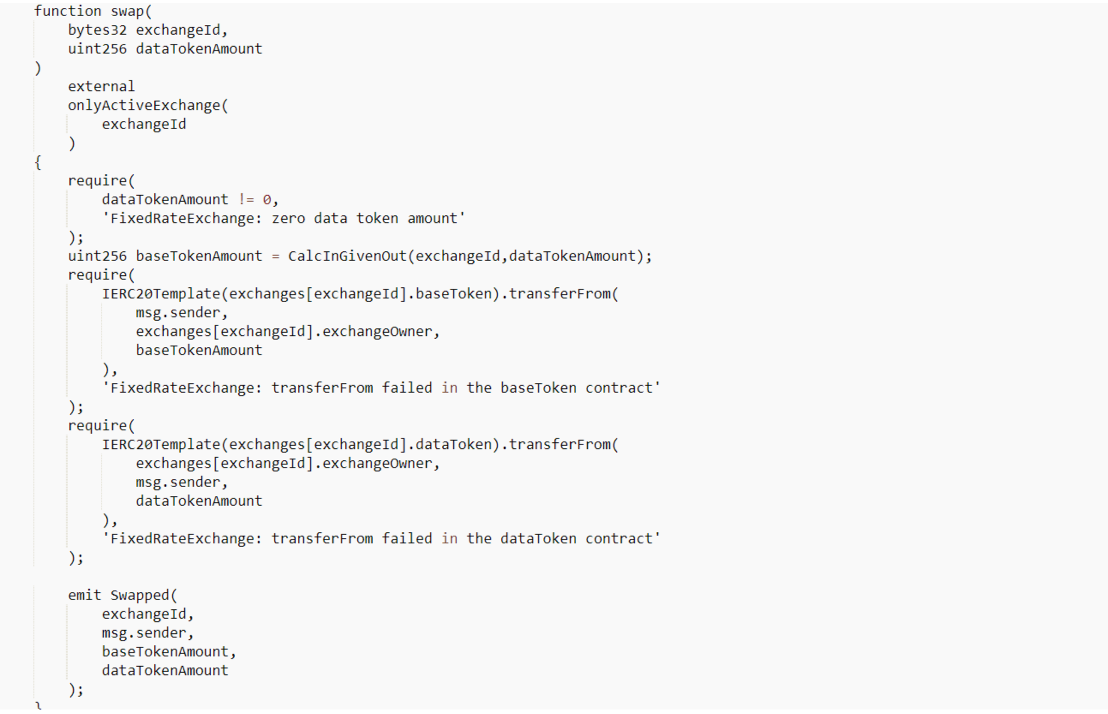


```
 * @param exchangeId a unique exchange idnetifier
 * @param newRate new fixed rate value
 */
function setRate(
    bytes32 exchangeId,
    uint256 newRate
)
    external
    onlyExchangeOwner(exchangeId)
{
    require(
        newRate != 0,
        'FixedRateExchange: Ratio must be >0'
    );

    exchanges[exchangeId].fixedRate = newRate;
    emit ExchangeRateChanged(
        exchangeId,
        msg.sender,
        newRate
    );
}
```

Admin can set the rate

The following function might be attacked by front-run.



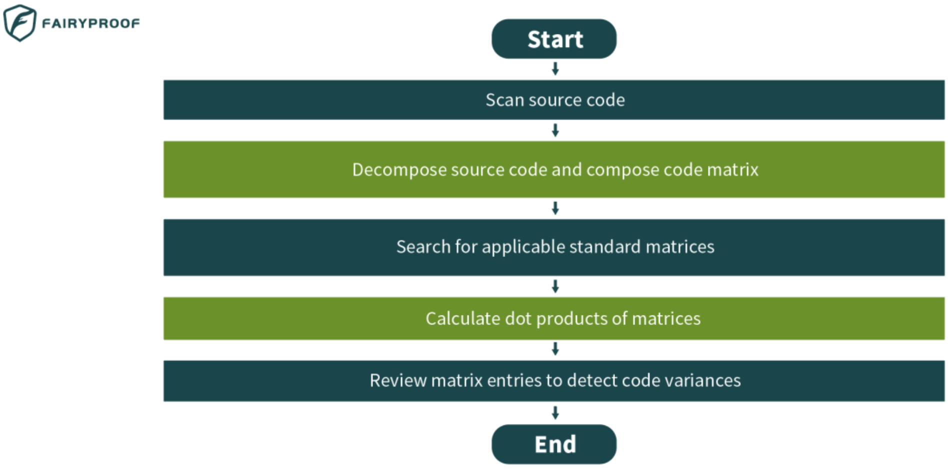
```
function swap(
    bytes32 exchangeId,
    uint256 dataTokenAmount
)
    external
    onlyActiveExchange(
        exchangeId
    )
{
    require(
        dataTokenAmount != 0,
        'FixedRateExchange: zero data token amount'
    );
    uint256 baseTokenAmount = CalcInGivenOut(exchangeId,dataTokenAmount);
    require(
        IERC20Template(exchanges[exchangeId].baseToken).transferFrom(
            msg.sender,
            exchanges[exchangeId].exchangeOwner,
            baseTokenAmount
        ),
        'FixedRateExchange: transferFrom failed in the baseToken contract'
    );
    require(
        IERC20Template(exchanges[exchangeId].dataToken).transferFrom(
            exchanges[exchangeId].exchangeOwner,
            msg.sender,
            dataTokenAmount
        ),
        'FixedRateExchange: transferFrom failed in the dataToken contract'
    );

    emit Swapped(
        exchangeId,
        msg.sender,
        baseTokenAmount,
        dataTokenAmount
    );
```

This validation issue was quickly discovered by our system and we suggested to remove or limit the admin’s access control to the setRate function.

TOKEN VARIANCE DETECTION SYSTEM

In order to automatically detect the variances of a token’ s implementation to discover potential issues or risks based to the existing token standards specifically in the Ethereum ecosystem, our system follows the steps below:


FAIRYPROOF
Start
Scan source code
Decompose source code and compose code matrix
Search for applicable standard matrices
Calculate dot products of matrices
Review matrix entries to detect code variances
End

Step 1: it scans the whole source code

Step 2: it decomposes the source code based on its functions, interfaces, inheritance relationships, etc and composes an $m_i \times n$ matrix A that represents the source code’ s characteristics.

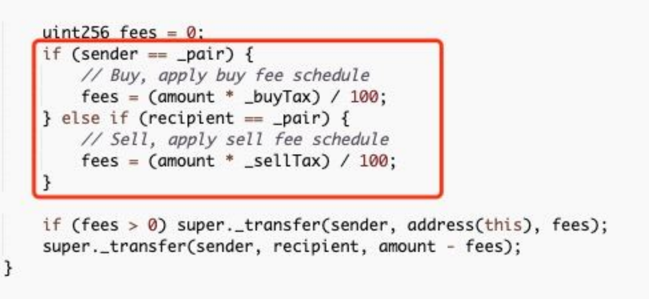
Step 3: it searches for various standard models that represent typical token standards such as ERC-20 token, ERC-721 token, ERC-1155 token etc in the database. These models are represented as $n \times m_j$ matrices as $B_1, B_2, \ldots B_j$.

Step 4: it calculates the dot products of $A \cdot B_1, A \cdot B_2, \ldots, A \cdot B_j$ and gets an $m_i \times m_1$matrix C_1, an $m_i \times m_2$ matrix C_2, ... and an $m_i \times m_j$ matrix C_i.

Step 5: each of the entries in these matrices indicates a possible issue. The higher the value of an entry is, the riskier the code section that is represented by that entry is.

Step 6: our technical team reviews and examines all the code sections that may have potential issues indicated by these matrix entries and draws a final conclusion on the token’ s legitimacy.

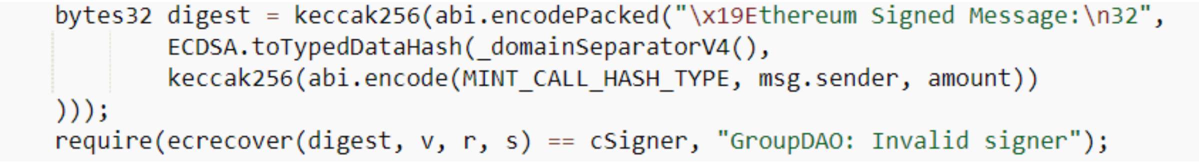
Following this automation procedure we quickly identified the “tax” mechanism in the MaskDAO’ s token shown as follows:



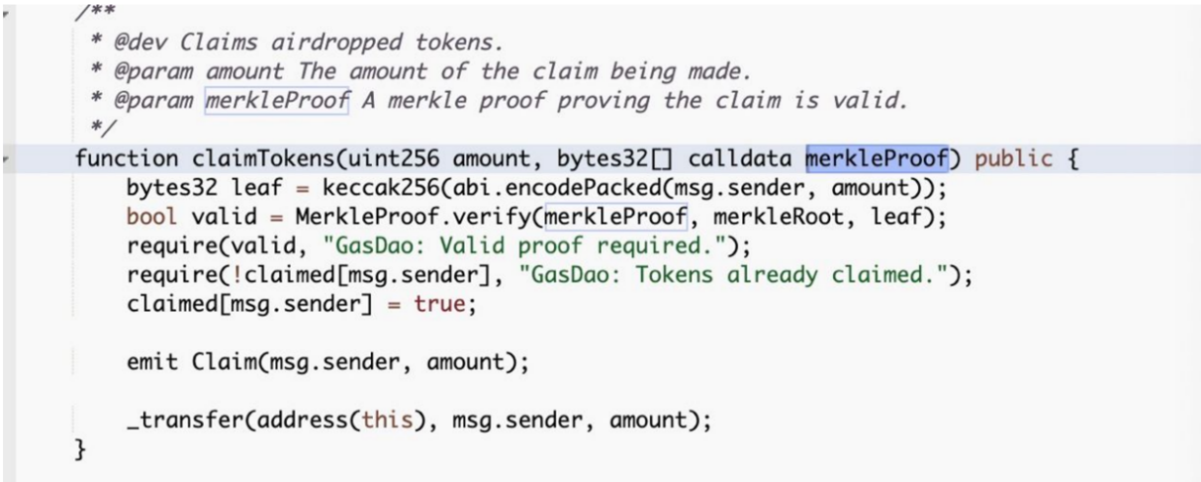
```
    uint256 fees = 0;
    if (sender == _pair) {
        // Buy, apply buy fee schedule
        fees = (amount * _buyTax) / 100;
    } else if (recipient == _pair) {
        // Sell, apply sell fee schedule
        fees = (amount * _sellTax) / 100;
    }

    if (fees > 0) super._transfer(sender, address(this), fees);
    super._transfer(sender, recipient, amount - fees);
}
```

The automation tool also helped us quickly pinpoint an implementation of off-chain message that was commonly used in the recent meme tokens such as SOS, MASK, GDO, GAS etc. Here were two examples of code section:



```
bytes32 digest = keccak256(abi.encodePacked("\x19Ethereum Signed Message:\n32",
        ECDSA.toTypedDataHash(_domainSeparatorV4(),
        keccak256(abi.encode(MINT_CALL_HASH_TYPE, msg.sender, amount))
)));
require(ecrecover(digest, v, r, s) == cSigner, "GroupDAO: Invalid signer");
```



```
/**
 * @dev Claims airdropped tokens.
 * @param amount The amount of the claim being made.
 * @param merkleProof A merkle proof proving the claim is valid.
 */
function claimTokens(uint256 amount, bytes32[] calldata merkleProof) public {
    bytes32 leaf = keccak256(abi.encodePacked(msg.sender, amount));
    bool valid = MerkleProof.verify(merkleProof, merkleRoot, leaf);
    require(valid, "GasDao: Valid proof required.");
    require(!claimed[msg.sender], "GasDao: Tokens already claimed.");
    claimed[msg.sender] = true;

    emit Claim(msg.sender, amount);

    _transfer(address(this), msg.sender, amount);
}
```

BEST PRACTICES TO PREVENT SECURITY ISSUES

In this section we present some best practices to help both blockchain developers and users manage the risks posed by the incidents that happened in 2021 and support coordinated and efficient response to crypto security incidents. Both blockchain developers and users are recommended to apply these practices to the greatest extent possible based on availability of their resources.

Note: the blockchain developer here refers to not only developers of blockchains but also every developer that participates in development of DApps, blockchains or systems pertaining to crypto currencies. The blockchain user here refers to everyone that participates in activities pertaining to crypto system's management, operation, trading etc.

FOR BLOCKCHAIN DEVELOPERS

The best way to protect a PoW based permissionless blockchain's security is to develop and grow its ecosystem to incentivize as many miners as possible to join the network and increase the whole network's hash rate.

Among the solutions that scaled blockchain mainnets, the number of incidents related to side chains was more than that of lay 2 solutions. But layer 2 solutions are a new area. Awareness of security with layer 2 solutions should be raised as more layer 2 solutions will emerge and develop in the coming years. Solutions to tackle the security challenges in this new area need to be researched and developed promptly.

With regard to a DApp's security, the security pertaining to its smart contracts still has the highest priority but the security pertaining to its front-end and service side needs increasingly more awareness. Audit of a DApp's front-end and server side should no longer be ignored.

It should be a must-have step to transfer an admin's access control to a multi-sig wallet or a DAO to manage access control to crypto assets or critical operations.

Flashloans and validation for access control are two biggest security challenges to a smart contract's security. Proper handling of these two should have the highest priority for a smart contract developer when designing and coding a smart contract.

FOR BLOCKCHAIN USERS

Users should be cautious about holding or trading a PoW based crypto currencie whose underlying blockchain's hash rate is too low because of the possibility of suffering from 51% attacks.

Both side chains and layer 2 solutions are young, immature and not immune to security issues. It would be better to check a side chain or a layer 2 solution's security status before participating in activities on top of it.

When interacting with a DApp, while it is still necessary to pay great attention to its smart contracts' security, it is increasingly more important to pay attention to the security of its user interface and be cautious about any suspicious messages, prompts and behavior presented by the user interface.

It is strongly suggested users should check whether or not a project has audit reports and should read its audit reports before proceeding with further actions.

Use a cold wallet to manage crypto assets that are not for frequent trading. Be careful about using a hot wallet and make sure the hardware in which a hot wallet is installed is safe and secure.

Be cautious about a DApp whose team members are unknown, unheard of or lack of reputation since such a DApp may eventually be a rug-pull. Be cautious about a centralized exchange which hasn't built a reputation or doesn't have tracked transaction data on third party media since it may eventually be a rug-pull as well.

REFERENCES

[1] Arbitrum Portal, https://portal.arbitrum.one/

[2] Optimism, https://www.optimism.io/

[3] "DeFi 2.0: A beginner's guide to the second generation of DeFi protocols". https://cointelegraph.com/defi-101/defi2-0-a-beginners-guide-to-the-second-generation-of-defi-protocols.

[4] CryptoPunks. https://www.larvalabs.com/cryptopunks

[5] BAYC. https://boredapeyachtclub.com/

[6] Axie Infinity. https://axieinfinity.com/

[7] "Play-To-Earn Gaming Is Driving NFT And Crypto Growth". https://www.forbes.com/sites/robertfarrington/2021/12/13/play-to-earn-gaming-is-driving-nft-and-crypto-growth/?sh=7f3afd1dc2dc . December 13, 2021

[8] Morris, David Z. (15 May 2016). "Leaderless, Blockchain-Based Venture Capital Fund Raises $100 Million, And Counting". Fortune. Archived from the original on 21 May 2016. Retrieved 23 May 2016.

[9] Popper, Nathan (21 May 2016). "A Venture Fund With Plenty of Virtual Capital, but No Capitalist". The New York Times. Archived from the original on 22 May 2016. Retrieved 23 May 2016.

[10] "Blockchains: The great chain of being sure about things". The Economist. 31 October 2015. Archived from the original on 3 July 2016. Retrieved 18 June 2016. The technology behind bitcoin lets people who do not know or trust each other build a dependable ledger. This has implications far beyond the crypto currency.

[11] Narayanan, Arvind; Bonneau, Joseph; Felten, Edward; Miller, Andrew; Goldfeder, Steven (2016). Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton: Princeton University Press. ISBN 978-0-691-17169-2.

[12] Blockchain. https://en.wikipedia.org/wiki/Blockchain. January 4, 2022

[13] Stifter N, Judmayer A, Schindler P, et al. What is Meant by Permissionless Blockchains?[J]. IACR Cryptol. ePrint Arch., 2021, 2021: 23

[14] Stifter N, Judmayer A, Schindler P, et al. What is Meant by Permissionless Blockchains?[J]. IACR Cryptol. ePrint Arch., 2021, 2021: 23.

[15] DApp,[1] "CVC Money Transmission Services Provided Through Decentralized Applications (DApps)" (PDF). FinCEN. Retrieved 2019-05-09.

[16] dApp,[2] "IEEE DAPPS 2020". ieeedapps.net. Archived from the original on 2020-04-26. Retrieved 2020-08-15.

[17] Sidechains. https://ethereum.org/en/developers/docs/scaling/sidechains/

[18] Layer-2. https://academy.binance.com/en/glossary/layer-2

[19] Solana. https://solana.com/

[20] ETC. https://ethereumclassic.org/

[21] BSV. https://bitcoinsv.com/

[22] Verge. https://vergecurrency.com/

[23] Firo. https://firo.org/

[24] Polygon. https://polygon.technology/

[25] Liquid Network. https://river.com/learn/terms/l/liquid-network/

[26] Arbitrum One. https://portal.arbitrum.one/

[27] “What Is a 51% Attack?”. https://www.coindesk.com/learn/what-is-a-51-attack/ . October 12, 2021

[28] Denial-of-service attack. https://en.wikipedia.org/wiki/Denial-of-service_attack . January, 2022

[29] Vulcan Forged. https://vulcanforged.com/

[30] Badger DAO. https://app.badger.com/

[31] Farmers World. https://farmersworld.io/

[32] Flash-loans. https://aave.com/flash-loans/

[33] Cream Finance. https://app.cream.finance/

[34] Spartan Protocol. https://spartanprotocol.org/

[35] Yearn Finance. https://yearn.finance/#/home

[36] Indexed Finance. https://indexed.finance/

[37] AutoShark. https://autoshark.finance/

[38] Pancake Bunny. https://pancakebunny.finance/

[39] BurgerSwap. https://burgerswap.org/

[40] BitMEX. https://www.bitmex.com/

[41] Liquid. https://www.liquid.com/

[42] AscendEX. https://ascendex.com/

[43] HitBTC. https://hitbtc.com/zh_CN

[44] Bilaxy. https://bilaxy.com/

[45] Compound Finance. https://compound.finance/

[46] dYdX. https://dydx.exchange/