Immunefi

The True Origin of Hacks

Top Web3 Vulnerabilities

PREPARED BY IMMUNEFI

Vulnerabilities Report

PREPARED BY IMMUNEFI

OVERVIEW

After years of reporting on crypto losses, Immunefi has investigated the true causes of hacks in the web3 space, namely vulnerabilities. The term vulnerability refers to an absence or weakness of a safeguard in any type of relevant asset—for example, in smart contracts, code implementation, or infrastructure. Exploitation of these vulnerabilities often lead to a devastating impact on the project.

In 2022, we saw losses of **$3,948,856,037** across the web3 ecosystem. $3,773,906,837 of this was lost to hacks across 134 specific incidents, and the remaining $174,949,200 was lost to fraud across 34 specific incidents.

Losses are defined as a combination of hacks and alleged fraud incidents. The latter account for a large proportion of losses, but are distinct from an attack on the technological stack of a project.

Web3 projects are incredibly complex. They include infrastructure, code inherited from different projects and handwritten code, various cryptography, oracle modules, and myriad other components. Each of these may contain flaws. Furthermore, these are often closely interconnected, which can further increase attack vectors. However, not all attack vectors occur equally or cause the same amount of damage.

2. Key Takeaways in 2022

Vulnerabilities Report

KEY TAKEAWAYS IN 2022

- Infrastructure is king. **46.5%** of all hacks in 2022 in monetary terms occurred via infrastructure, e.g. poor private key handling. Developers and researchers generally focus on designing and coding the smart contract protocol, which forms the core of web3 projects, but all too often the danger lurks one level below. It comes as no surprise that infrastructure in particular is the major difference between DeFi and CeFi projects. 11 of 13 exploits in CeFi were infrastructural in nature.

- The biggest infrastructural issue is private key management, which is essential to maintaining self-custody of crypto assets. Typically, private key management is not something that undergoes a security audit, and not all web3 projects adequately care about rigorous key management policies, practices, or emergency plans.

- For smart contract-related vulnerabilities, the more complex the use cases, the more likely it is that bugs will arise within the flow of logic.

- Developers make mistakes and introduce vulnerabilities far too often in smart contracts when it comes to access control, input validation, and arithmetic operations. This accounts for nearly **37.5%** of all incidents. Fortunately, their damage in cash is small (5% of all damage in cash).

- Bridge hacks play an important role in losses. Blockchains are highly isolated environments; inter-blockchain communication is not easy, and third parties often step in to build what's known as a bridge to find some way to connect the two blockchains together. The basic functionality of a bridge is to lock funds from one blockchain and release the equivalent value of funds on the other blockchain. If there's a minor problem with such proof generation or verification, a malicious actor could steal funds on one side of the bridge.

"Web3 projects are incredibly complex and can be attacked through multiple vectors. The standard methodology we developed highlights the fact that infrastructural issues remain a predominant category of vulnerabilities and a costly concern for the industry."

Mitchell Amador
Founder and CEO at Immunefi

3. Classification of Vulnerabilities

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Immunefi has analyzed 128 technical vulnerabilities that resulted in hacks and losses in 2022*. In order to discover the nature of the vulnerabilities, Immunefi distinguished technical vulnerabilities from Fraud (social engineering, scams, and rug pulls), since they are not triggered by any code or contract design flaws.

It revealed that causes of Hacks fall into three clearly identifiable categories: flaws in the logic, implementation, and infrastructure of the project.

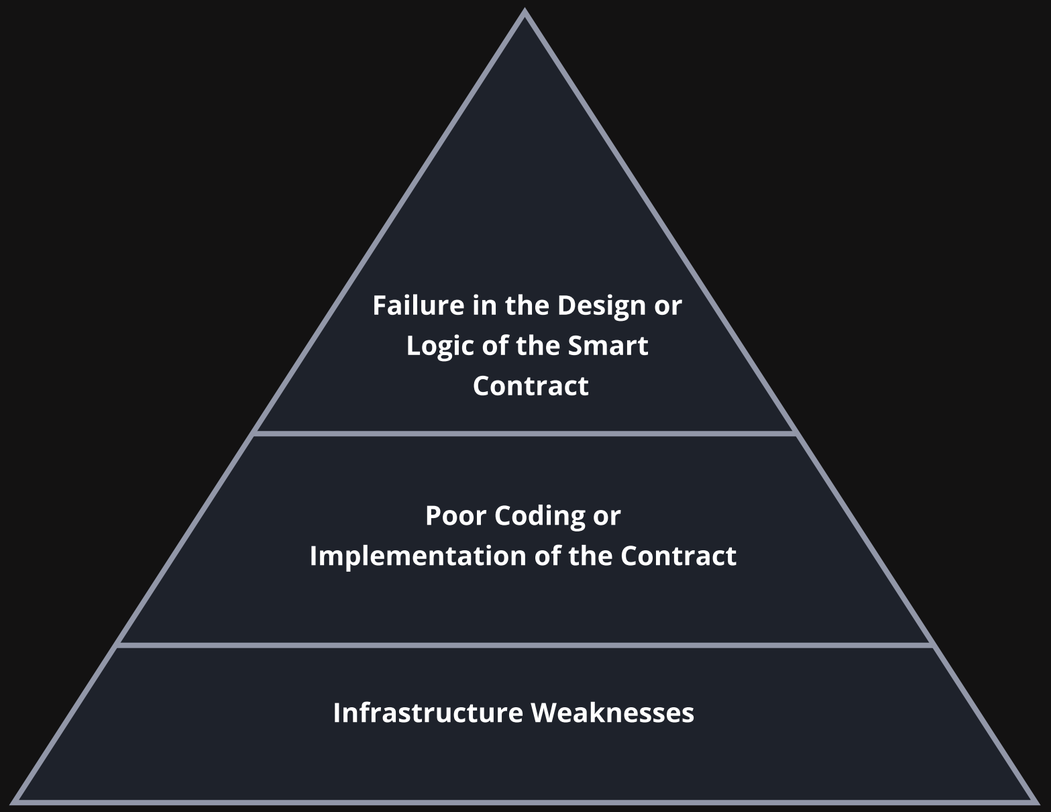
Crypto Losses

- Fraud:
  - Occasional scam
  - Rug pull
- Hack:
  - Failure in the design/logic of the smart contract
  - Poor coding/implementation of the contract
  - Infrastructure weaknesses

*Immunefi's vulnerability classification is based on the analysis of hacking incidents across **2022**.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES



Description

- **Failure in the design/logic of the smart contract:** when the project outlined on paper behaves improperly.
- **Poor coding/implementation of the contract:** when the design and infrastructure are secure, but the code contains flaws.
- **Infrastructure weaknesses:** the IT-infrastructure on which a smart contract operates—for example virtual machines, private keys, etc. Infrastructure exposure can lead to hacks and losses, even if the smart contract itself has been designed, written, and tested well.

Examples

- **Logic: BNBChain** (Q4, $570 million). The bridge's failure to completely verify the Merkle tree root hash created a vulnerability. This allowed the attacker to create forged proofs from an earlier, legitimate one and mint BNB directly to their own wallet.
- **Implementation: Qubit** (Q1, $80 million). The protocol was duped into believing that attackers had deposited money when they hadn't. The hacker called `deposit()` in the QBridge #eth contract without making any deposit, and emitted the Deposit event. The exploit was caused by `tokenAddress.safeTransferFrom` in QBridgeHandler.sol which didn't revert the transaction when the tokenAddress is the 0x0.
- **Infrastructure: Ronin Network** (Q1, $625 million). The hacker used hacked private keys to forge fake withdrawals.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Immunefi's experts have analyzed all hacks in 2022 and dissected each one to discover the root cause of the issue. As a result of this research, Immunefi has divided three major domains of vulnerabilities into smaller, more precise sub-domains of vulnerabilities listed below.

- **High-Level Failures in Design/Logic**
  - Cryptographic issues
  - Contract misconfiguration
- **Poor Coding/Implementation of the Contract**
  - Improper handling of external dependencies
  - Unsafe external calls; Usage of pools' spot prices;
  - Weak/missing access control and/or input validation
  - Arithmetic overflows, truncations, and other errors in calculation
  - Others
- **Infrastructure Weaknesses**

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

High-Level Failures in Design/Logic:

- **Cryptographic issues**
  - Merkle Tree errors (including Inferior hashing)
  - Signature replayability (e.g., lack of nonce usage)
  - Predictable random number generation (e.g., using block number)

Cryptographic issues are a common type of vulnerability in DeFi smart contracts that occur due to errors in implementing cryptographic algorithms or protocols. Some examples of cryptographic issues in DeFi include Merkle tree errors, inferior hashing, and signature replayability. Merkle tree errors occur when there is an error in the construction or verification of a Merkle tree. This is a data structure used in DeFi to efficiently prove the inclusion or absence of data in a large dataset.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

High-Level Failures in Design/Logic:

- **Contract misconfiguration**
  - Uninitialized state/storage/function (e.g. Parity wallet freeze)

Contract misconfiguration is a vulnerability in smart contracts that occurs when a contract is not properly configured, leading to unexpected behavior or security risks. This can occur due to errors in the contract's configuration parameters, such as incorrect or missing addresses, incorrect contract dependencies, or incorrect initialization parameters. For example, a contract misconfiguration could allow an attacker to bypass certain security checks, such as input validation or access control, and potentially modify the contract's state or steal funds from the contract. Additionally, a misconfigured contract could expose sensitive information to unauthorized parties, such as private keys or other authentication tokens, compromising the security of the contract and its users.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Poor Coding/Implementation of the Contract:

- **Improper handling of external dependencies**
  - Oracle manipulations/staleness due to attacking the logic of a third-party contract
  - Price manipulation due to relying on low liquidity pools
  - Lack of staleness check-in feeds

Improper handling of external dependencies. Smart contracts tend to rely on the values provided by third-party contracts in an insecure manner. In particular, this can be seen in the use of oracles as well as the dependence on low-liquidity liquidity pools which are subjected to price manipulation attacks. Oracles are used in DeFi to fetch data from external sources, such as price feeds, and to verify the occurrence of certain events—for example the settlement of a futures contract. If an oracle is compromised or manipulated, it can provide incorrect data to the smart contract, leading to financial losses for users.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Poor Coding/Implementation of the Contract:

- **Unsafe external calls; Usage of pools' spot prices;**
  - Reentrancy vulnerabilities
  - Forwarder issues

Unsafe external call is a vulnerability in DeFi smart contracts that occurs when a contract makes an external call to another contract without proper validation or protection. This allows an attacker to manipulate the behavior of the smart contract and potentially steal funds or cause other types of harm. In DeFi, smart contracts often interact with other contracts to perform complex actions, such as exchanging tokens or accessing external liquidity pools. These interactions can introduce security risks if the calling contract does not validate the data or behavior of the called contract. Reentrancy attacks are also of this nature.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Poor Coding/Implementation of the Contract:

- **Weak/missing access control and/or input validation**
  - Incorrect privilege
  - Bypass safety checks due to incorrect logic
  - Incorrect input data validation [or] missing validation

Weak or missing access control and input validation are vulnerabilities in DeFi smart contracts that occur when there is insufficient validation of user inputs, or when there is a lack of proper access control mechanisms. This leads to unauthorized access or the manipulation of smart contract data. Weak input validation occurs when a smart contract does not properly validate or sanitize user inputs, allowing an attacker to exploit vulnerabilities in the contract's logic and potentially execute arbitrary code or manipulate the contract's state. Missing access control occurs when a smart contract does not implement proper access control mechanisms to restrict access to sensitive functions or data. This also includes replay attacks and signature malleability.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Poor Coding/Implementation of the Contract:

- **Arithmetic overflows, truncations and other errors in calculation**
  - Using a wrong fixed-data size variable for the actual data
  - Rounding errors

Arithmetic overflows, truncations, and other errors in calculation refer to vulnerabilities in smart contracts that occur when the contract's code does not properly handle arithmetic operations, resulting in unexpected or incorrect results. For example, arithmetic overflows occur when the result of an arithmetic operation exceeds the maximum value that can be represented by the data type used to store the result. Truncations occur when the result of an arithmetic operation is rounded or truncated, leading to a loss of precision. Rounding errors and various incorrect calculations are also an issue. Attackers can exploit these vulnerabilities to steal funds or manipulate the contract's state. For example, an attacker could exploit an arithmetic overflow vulnerability to create or mint new tokens, leading to an inflation of the token supply and a devaluation of the token's value. Alternatively, an attacker could exploit a truncation vulnerability to steal funds from the contract or manipulate the contract's state in unintended ways.

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Poor Coding/Implementation of the Contract:

- **Others**
  - Governance attack
  - Incorrect logging
  - Denial of Service (gas consumption, storage bloat, unbounded loop)
  - Proxy issues (uninitialized proxy, storage collisions)

Vulnerabilities Report

CLASSIFICATION OF VULNERABILITIES

Infrastructure Weaknesses

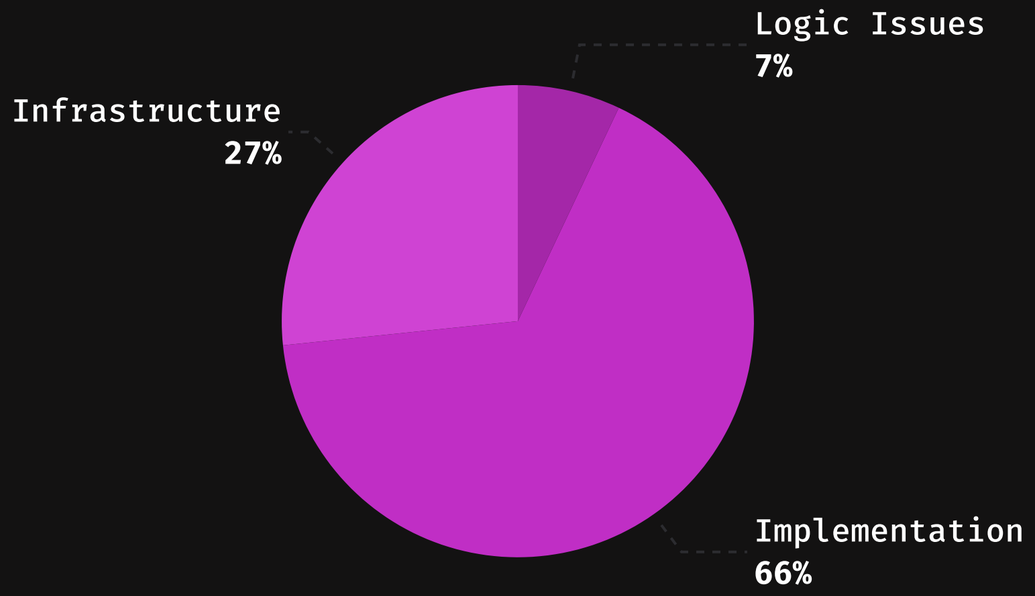
- Private key leakage, including using the private keys on an insecure communication channel
- Weak passphrase for the key vault that leads to brute force decryption
- Problems with 2FA
- DNS hijacking
- BGP hijacking
- Hot wallet compromise
- Using weak encryption methods or storing them in plaintext

By infrastructure, we mean IT infrastructure on top of which smart contracts operate, e.g., virtual machines, storage infrastructure to target private keys, etc. Exposure of infrastructure leads to hacks and losses, even though the smart contract itself is well-designed, written, and tested.

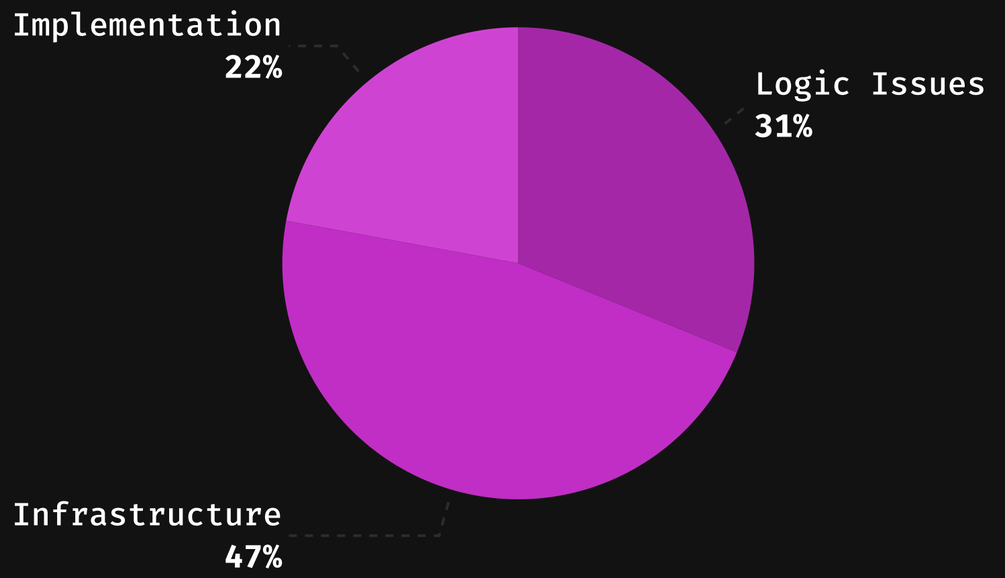
4. Vulnerabilities in Cases and Cash

Vulnerabilities 2022

VULNERABILITIES IN CASES



VULNERABILITIES IN CASH



Logic issues are often costly; a single one could destroy an entire project. Therefore, we highly recommend analyzing the logic and structure of your contract (and the way it utilizes cryptography) as thoroughly as possible.
When it comes to DeFi/CeFi, Infrastructure becomes crucial, because 11 of 13 exploits in CeFi were infrastructural in nature.

Vulnerabilities 2022

VULNERABILITIES IN CASES

Vulnerability	Cases	Share
Infrastructure weaknesses	34	26.56%
Cryptographic issues	2	1.56%
Contract misconfiguration	7	5.47%
Unsafe external calls; Usage of pools' spot prices	6	4.69%
Weak/missing access control and/or input validation	39	30.47%
Improper handling of external dependencies	27	21.09%
Arithmetic overflows, truncations and other errors in calculation	9	7.03%
Other	4	3.13%
Total	128	

Vulnerabilities 2022

VULNERABILITIES IN CASH

Vulnerability	Damage	Share
Infrastructure weaknesses	$1,716,192,510	46.48%
Cryptographic issues	$760,000,000	20.58%
Contract misconfiguration	$403,798,713	10.94%
Unsafe external calls; Usage of pools' spot prices	$110,440,000	2.99%
Weak/missing access control and/or input validation	$170,460,074	4.62%
Improper handling of external dependencies	$327,101,147	8.86%
Arithmetic overflows, truncations and other errors in calculation	$16,842,949	0.46%
Other	$187,371,444	5.07%
Total	**$3,692,206,837**	

Cost per vulnerability*

VULNERABILITIES 2022

Based on the data, Immunefi has calculated the average cost* of each type of vulnerability and ranked them. Poor logic or infrastructure negligence is significantly more costly than flaws in coding and implementation.

Vulnerability	Cost
Cryptographic issues	$380,000,000
Contract misconfiguration	$57,685,530
Infrastructure weaknesses	$50,476,250
Unsafe external calls; Usage of pools' spot prices	$18,406,666
Improper handling of external dependencies	$12,114,857
Weak/missing access control and/or input validation	$4,370,771
Arithmetic overflows, truncations and other errors in calculation	$1,871,438

Vulnerabilities Report

THE SCOPE

Since the subject of our study is the origin of hacks, this study specifically focuses on the technical nature of the relevant vulnerabilities and hacks.

Some price manipulation issues were excluded from the scope of the study because they are heavily dependent on various trading manipulations due to the economic peculiarities of a given project.

Immunefi has also grouped infrastructure vulnerabilities into a separate category.

Immunefi

Immunefi is the leading bug bounty and security services platform for web3 protecting over $50 billion in user funds. Immunefi features a massive community of whitehat hackers who review projects' blockchain and smart contract code, find and responsibly disclose vulnerabilities, and get paid for making crypto safer. With Immunefi, whitehat hackers are rewarded based on the severity of the vulnerability that they discover, creating incentives for as many experts as possible to examine project code for vulnerabilities.

Immunefi has pioneered the scaling web3 bug bounties standard, meaning that rewards should be priced accordingly with the severity of an exploit and the volume of funds at risk, which resulted in the company building the largest community of security talent in the web3 space.

TOTAL BOUNTIES PAID

Immunefi has paid out over **$85 million** in total bounties, while saving over **$25 billion** in user funds.

TOTAL BOUNTIES AVAILABLE

Immunefi offers over **$150 million** in available bounty rewards.

SUPPORTED PROJECTS

Trusted by established, multi-billion dollar projects like Chainlink, Wormhole, MakerDAO, TheGraph, Synthetix, and more, Immunefi now supports more than 300 projects across multiple crypto sectors.

LARGEST BUG BOUNTY PAYMENTS IN THE HISTORY OF SOFTWARE

Immunefi has facilitated the largest bug bounty payments in the history of software:

- **$10 million** for a vulnerability discovered in Wormhole, a generic cross-chain messaging protocol.
- **$6 million** for a vulnerability discovered in Aurora, a bridge, and a scaling solution for Ethereum.
- **$2.2 million** for a vulnerability discovered in Polygon, a decentralized Ethereum scaling platform that enables developers to build scalable, user-friendly dApps.

Note:

- Immunefi's vulnerability classification is based on the analysis of hacking incidents across 2022.

More:

- If you’re a developer thinking about a bug-hunting career in web3, we got you. Check out our **Web3 Security Library**, and start taking home some of the over $150M in rewards available on Immunefi — the leading bug bounty platform for web3.

For more information, please visit **https://immunefi.com/**