

HACKHUB 2024

The latest trends in cryptocurrency-enabled hacks, money laundering techniques, DeFi exploits, and more

Table of Contents

1. Key Takeaways	03
2. Executive Summary	04
3. Smart Contract Exploits	08
a. Smart Contract Exploits	09
📁 <i>Case Study: The Deus DAO Hack</i>	11
b. Reentrancy Vulnerabilities	11
📁 <i>Case Study: Sturdy Finance Hack</i>	13
c. Arithmetic Issues	14
📁 <i>Case Study: Poolz Finance Exploit</i>	14
4. Hacks and Other Vulnerabilities	16
a. Hot Wallets Attacks in CeFi	16
📁 <i>Case Study: The Alphapo Exploit</i>	18
b. Private Key Compromises in DeFi	18
📁 <i>Case Study: The LianGo Exploit</i>	19
c. Flash Loan Attacks	19
📁 <i>Case Study: The KyberSwap Exploit</i>	20
d. Oracles & Price Manipulation Attacks	21
📁 <i>Case Study: BonqDAO Exploit</i>	21
e. MEV & Sandwich Attacks	21
📁 <i>Case Study: Sandwich the Ripper</i>	23
5. Trends in Crypto Laundering Techniques	24
a. The use of cross chain bridges	24
📁 <i>Case Study: LastPass Hack</i>	25
b. Use of Coinjoins & Mixers	26
📁 <i>Case Study: Tornado Cash</i>	27
c. Use of dApps for token Swap	27
📁 <i>Case Study: Decentralized Exchanges</i>	28
6. Hackhub Methodology	30
7. Glossary	32
8. References	33

Expert Speak



“DeFi rather than CeFi continues to be the top target for hackers, highlighting the evolving landscape of cyber threats in the cryptocurrency sector. Many smart contracts, fundamental to DeFi platforms, have not undergone comprehensive security audits, leaving significant vulnerabilities exposed. Couple this with the substantial amounts of funds locked within these contracts, DeFi has become an increasingly attractive target for those looking to exploit these weaknesses. As this sector continues to grow, the imperative for robust security measures and vigilant oversight becomes ever more critical to safeguard against these emerging threats.”

Matt Swenson

Retired HSI Cyber Division Chief, CEO BlackRainbow - North America



“The only constant in the cryptocurrency industry is change. The common attack vectors this year are different from the year before, and those from the year before that. With each innovation, new vulnerabilities emerge, making it imperative for everyone in the space to keep their security knowledge up-to-date through resources like HACKHUB. By understanding the latest trends in hacks, we empower ourselves to anticipate and mitigate potential threats more effectively.”

Charles Rettig

Former Commissioner of the Internal Revenue Service



About HACKHUB

This report sheds light on the latest trends in criminal activity in the blockchain industry by conducting a deep analysis on 2023's smart contract exploits, security breaches, attacks on exchanges, NFT platforms, cross-chain bridges, and wallets. This report documents the flow of stolen funds from some of the major hacks of 2023, the vulnerabilities that were exploited, and some mitigation strategies that could have prevented the loss of billions of dollars of users' funds to bad actors.

About Merkle Science

Founded in 2018, Merkle Science is the next-generation predictive cryptocurrency risk and intelligence platform that helps crypto companies, DeFi participants, financial institutions, and government agencies detect, investigate, and prevent illegal activities involving cryptocurrencies. The core team comes from Paypal, Forter, Luno, Bank of America, FBI, and the US DOJ.

Merkle Science's proprietary Behavioral Rule Engine enables our tools to go beyond blacklists so that compliance teams may fulfill their local KYC and AML obligations and industry players may keep pace with increasingly complex illicit activities.

Merkle Science envisions a world powered by crypto and is creating the infrastructure necessary to ensure the safe and healthy growth of the cryptocurrency industry as it becomes a key pillar of the \$22 trillion financial services ecosystem. We enable businesses to scale and mature so that a full range of individuals, entities, and services may transact with crypto safely.

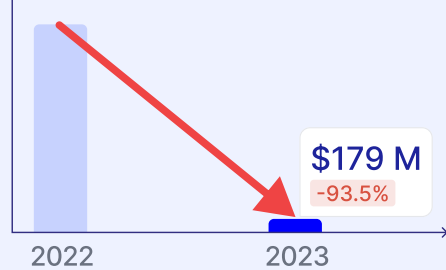
Top 10 Key Takeaways

Total losses from crypto hacks in 2023 amounted to

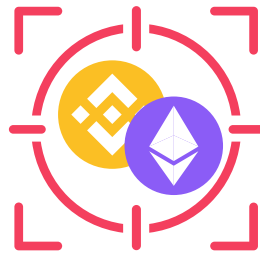
\$3.3 billion

a 15% decline from 2022

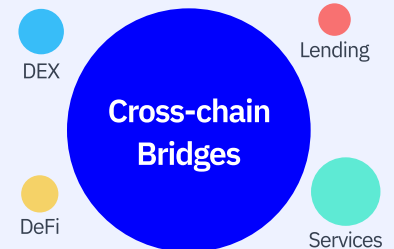
Amount lost due to smart contract vulnerabilities



Amount lost due to smart contract vulnerabilities drops 93.5% (\$179 million) when compared to 2022 (\$2.6 billion), yet these vulnerabilities still make up nearly half of the total number of attacks in 2023.



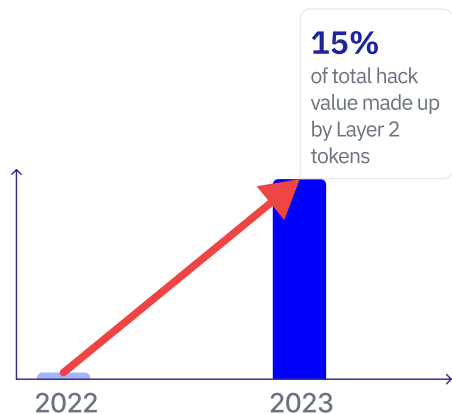
DeFi rather than CeFi continues to be the top target for hackers, with smart contracts and protocols on **Ethereum and Binance Smart Chain**—two of the top EVM-chains by total value locked (TVL) at the time—suffering the most exploits.



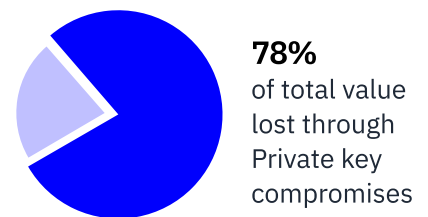
Attacks on cross-chain bridges led to the highest amount lost, followed by services (like wallet providers and payment gateways), decentralized exchanges, DeFi projects and DeFi lending platforms.



Tokens on Layer 2 (L2) protocols made up 15% of total hack value in 2023— a significant increase from the previous year where they accounted for less than 0.5% of the total attacks.



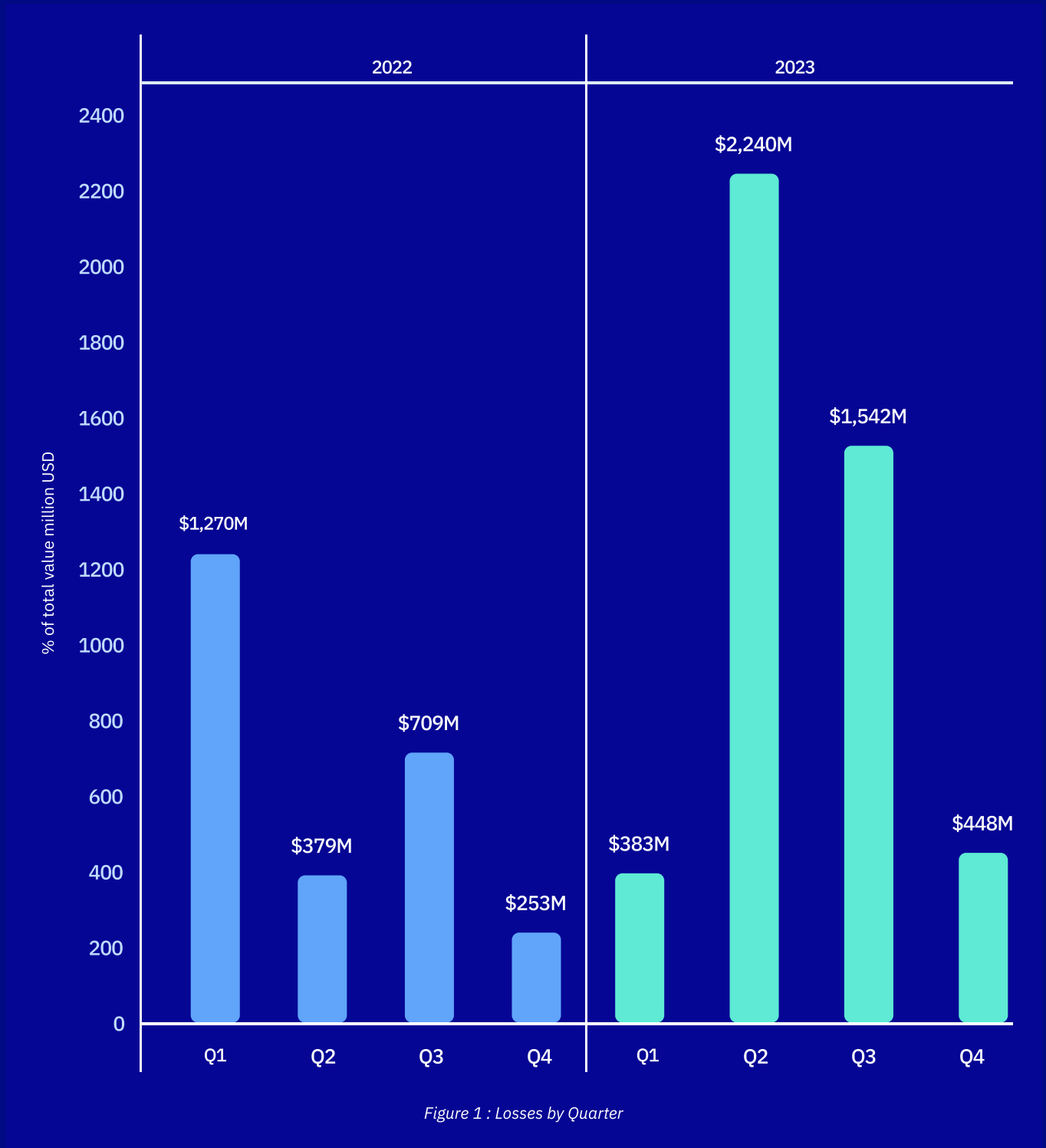
The North Korean **Lazarus Group** continues to carry out major hacks, netting over \$359 million in 2023 through attacks on Atomic Wallet, CoinEx, Alphapo, Stake.com, and CoinsPaid.



Private key compromises across the DeFi sector and hot wallet attacks in CeFi have surged, becoming the largest attack vectors in 2023, responsible for over 78% of total value lost—a staggering \$2.5 billion.

Executive Summary

2023 was a year that epitomized both the persistent security challenges facing the blockchain ecosystem, and its ecosystem players' resilience against these challenges. Despite total losses in 2023 amounting to a staggering \$3.3 billion—only a 15% decline from 2022's record breaking \$3.9 billion—a deeper analysis of these losses reveals a shift in the patterns of attacks, signaling a potential turning point for the industry.



While 2023 saw more attacks, attackers netted fewer funds on average

Top 5 Hacks

Hack Name	Amount lost
Poly Network Exploit	\$1,526,000,000
Mixin Network Exploit	\$200,000,000
Euler Finance Exploit	\$197,000,000
Multichain Exploit	\$144,000,000
Poloniex Exploit	\$127,000,000

2022 was defined by a series of high-profile, high-yield heists that resulted in one of the worst years on record with a staggering \$3.9 billion drained through hacks and exploits. **Losses in 2023, on the other hand, dropped by 15%** when compared to 2022. However, despite this decline in the value stolen, the number of attacks in 2023 saw an increase of approximately 10% from the previous year, exposing significant vulnerabilities and security gaps across the DeFi ecosystem in particular.

The continued involvement of sophisticated criminal syndicates, such as the notorious ransomware actor the Lazarus Group, further highlight the persistent threat facing crypto businesses. Attacks by the Lazarus group resulted in a staggering \$359 million in losses across just five hacks in 2023.

Attackers continue to focus on vulnerabilities in DeFi projects rather than CeFi






In both 2022 and 2023, DeFi projects remained the primary focus of attacks, accounting for over 95% of the total attacks occurring both years.

While Ethereum remained the most attacked blockchain in 2023, the number of attacks dropped significantly compared to 2022, going from 70% to 50%. Losses from attacks on Ethereum projects plummeted in 2023. In 2022, attackers stole a whopping \$2.8 billion. By the end of 2023, that number dropped by nearly half, down to \$1.49 billion.

This decrease is likely a result of the total value locked (TVL) in various DeFi projects beginning to spread across other EVM chains as the DeFi ecosystem continues to grow.

This is further evident as the amount lost by Binance Smart Chain (BSC)-based projects grew from 23% in 2022 to 38% in 2023. Other exploited projects in 2023 included those on high TVL EVM and EVM-compatible chains like Polygon, TRON, Arbitrum, and Optimism.

Top 5 Blockchains Exploited

BlockChain	Amount lost
 ETH	\$1,492,000,000
 BSC	\$1,275,000,000
 MATIC	\$218,000,000
 TRX	\$120,000,000
 BTC	\$71,000,000

Amount lost due to smart contract vulnerabilities drops 92%, yet they still make up nearly half of all hacks in 2023

Merkle Science researchers found that while smart contract vulnerabilities remained a prevalent target, accounting for 46% of all hacks in 2023, the total value stolen from these exploits exhibited a significant decline. Losses attributable to smart contract vulnerabilities in 2023 amounted to \$179 million, representing a substantial decrease of 92% compared to the nearly \$2.6 billion lost in 2022.*

**This positive development is likely due to a confluence of factors. Developers and auditors are prioritizing security best practices, leading to the creation of more robust smart contracts. Advancements in security tools are empowering further protection, and cybercriminals may be shifting their focus to softer targets that need comparatively lesser technical expertise and knowledge.*

Hackers continue to use Flash Loans to exploit smart contract vulnerabilities

Flash loan attacks have seen a notable rise from 0.5% of the total losses in 2022 to approximately 8% in 2023. Flash Loans allow DeFi users to borrow large sums of cryptocurrency instantly, without needing collateral. Due to this ease of access, they have become a prime tool for bad actors looking to exploit new vulnerabilities in smart contracts. This trend is likely to continue to grow as new and unaudited smart contracts for DeFi projects continue to proliferate with the growing total value locked across various EVM blockchains.

The 2024 crypto landscape finds itself at an inflection point

While the threat of crime persists, there is a palpable sense of optimism and determination to create a safer and more secure ecosystem. Through a collaborative effort between industry stakeholders, regulatory bodies, and enforcement agencies, the crypto community stands poised to turn the tide against attackers.

Advancements in blockchain analytics offer a powerful tool in the fight against cryptocurrency-enabled crime. By leveraging cutting-edge technologies, authorities gain valuable insights into criminal behavior, enabling more effective detection, prevention, and prosecution of illicit activities. At Merkle Science, we remain steadfast in our commitment to supporting this pursuit of a safer cryptocurrency ecosystem.

Private key compromises and hot wallet attacks account for more than half of stolen funds in 2023

In 2023, private key compromises in DeFi became the largest attack vector, responsible for over 56% of total losses and reaching a staggering \$1.88 billion. Some of the biggest hacks of the year, including the Multichain hack, Poly Network hack, and Heco Bridge hack, were all the result of compromised private keys.

Moreover, attacks on CeFi hot wallets emerged as the second leading cause of losses in 2023, indicating an alarming trend. Over 22% of stolen funds, amounting to a staggering \$748 million, resulted from hot wallet breaches. Furthermore, two out of the five biggest hacks of the year targeted hot wallets, highlighting the severity of the issue.

Crypto Hacks Deep Dive

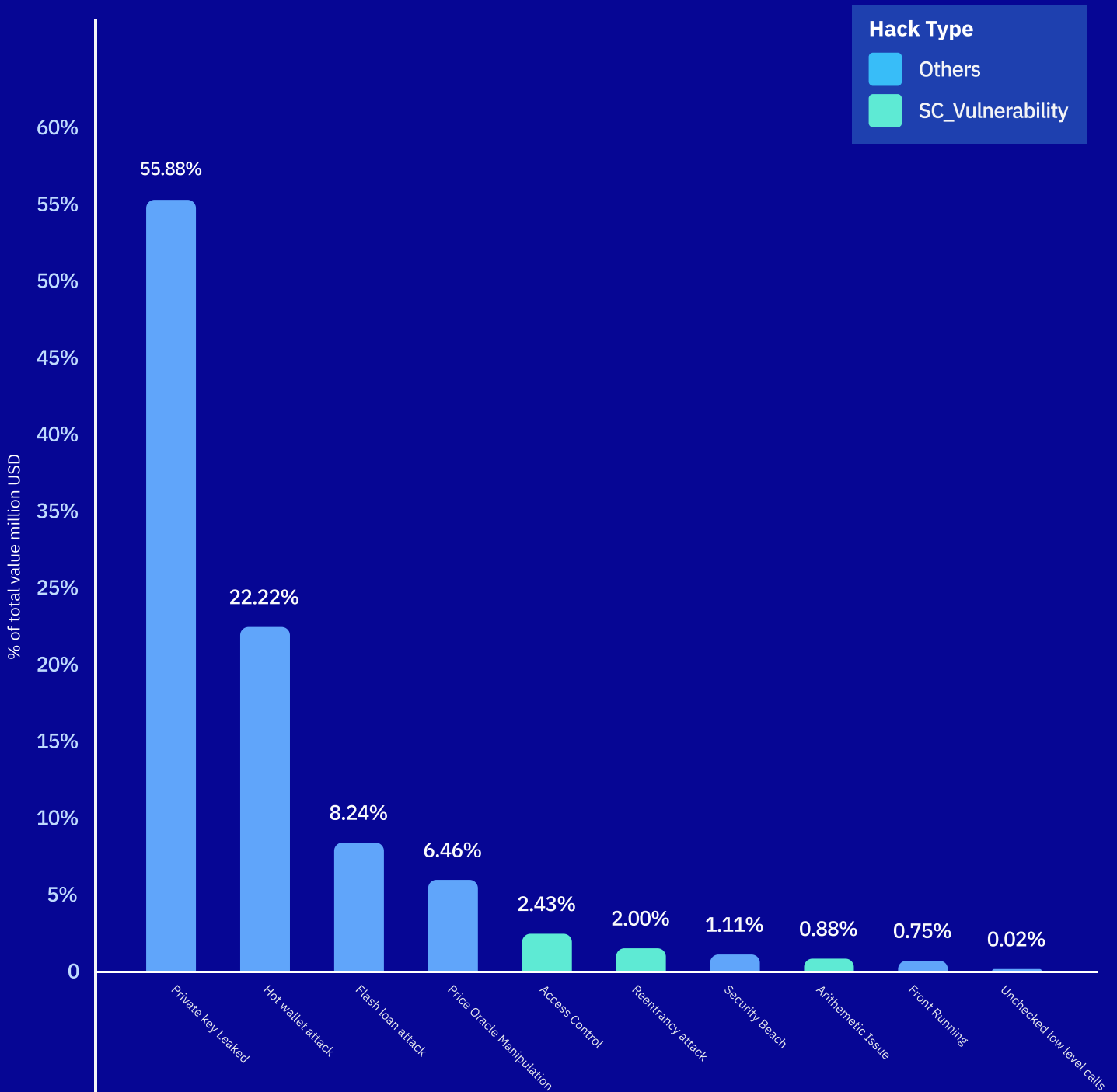
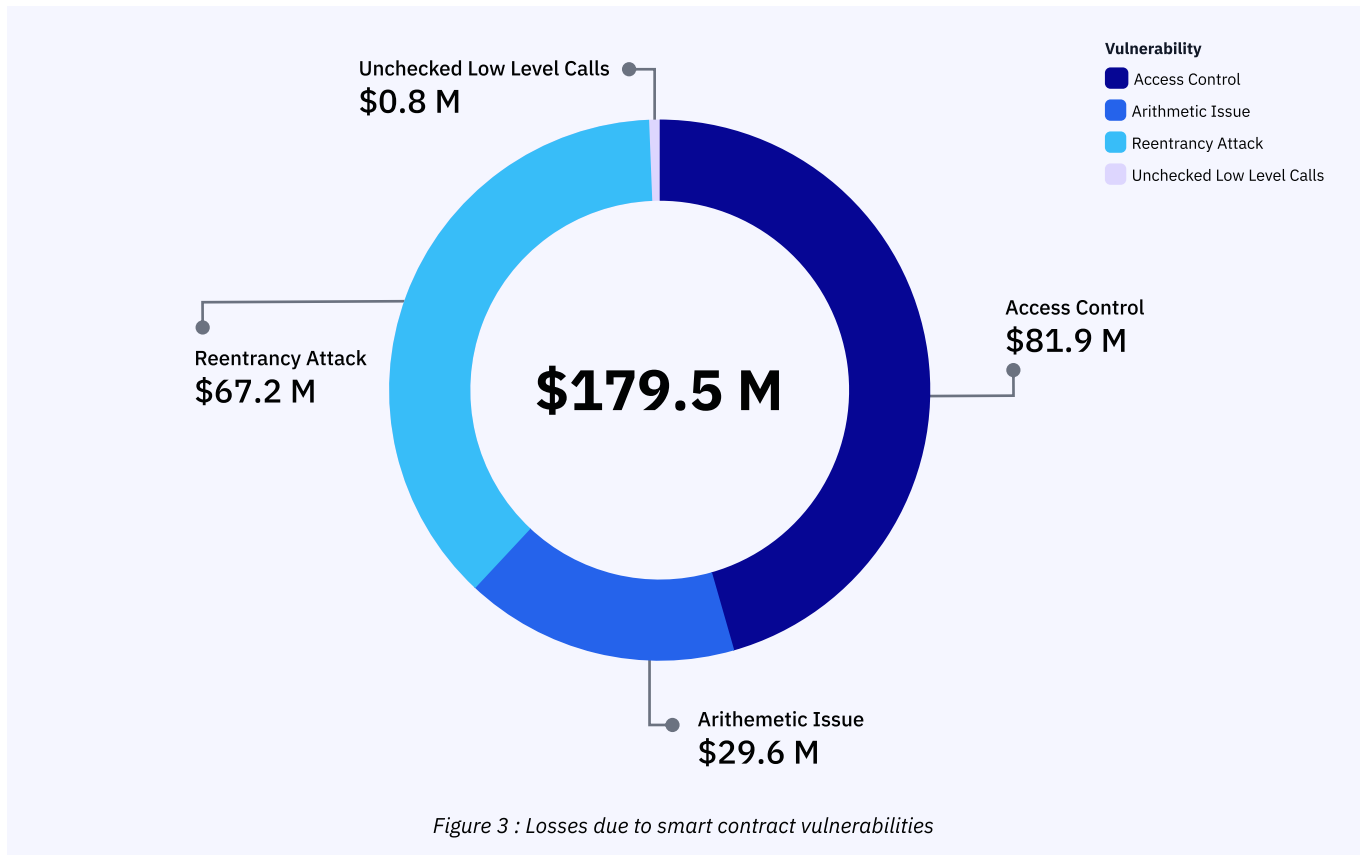


Figure 2 : Total Losses by Vulnerabilities

A staggering \$3.3 billion was lost to crypto-enabled hacks and exploits in 2023. The most prevalent vulnerability, accounting for nearly 56% of total losses, was private key compromise. Other notable breach types included hot wallet attacks (22%), flash loan exploits (8.24%), price oracle manipulation (6.46%), access control exploit (2.43%), reentrancy attacks (2%), front running attacks (0.75%), arithmetic issues (0.88%), and unchecked low level calls (0.02%) (cf. Figure 2).

Common Smart Contract Vulnerabilities

Total loss due to smart contract vulnerabilities in 2023: **\$179 million**



Smart contract (SC) vulnerabilities have proven to be a significant concern for the blockchain industry, resulting in substantial losses for investors and users. The year 2022 witnessed staggering losses totaling \$2.6 billion due to exploits on vulnerabilities in SCs. However, there is a glimmer of hope in the data from 2023, where losses due to SC vulnerabilities decreased significantly to \$179 million. While this reduction signals progress in addressing vulnerabilities, it is crucial to recognize that the threat posed by SC vulnerabilities still persists.

The complexity of smart contract systems, coupled with the evolving nature of blockchain technology, enables new vulnerabilities to emerge rapidly. Even though efforts are made to enhance security measures, malicious actors continue to exploit weaknesses for their financial gain.

Below we have provided a detailed analysis of how smart contracts are exploited, the magnitude of financial losses incurred in 2023, and how platforms and developers can mitigate such attacks.

Access Control

In 2023, more than **\$81 million** was lost solely due to access control vulnerabilities in smart contracts deployed by various platforms. Well-known platforms like Yearn Finance, a yield aggregator, and Dues DAO, a DAO project, suffered losses of millions of dollars in users' funds due to inadequate access control measures.

The Critical Role of Access Control in Smart Contract Security

Learning about the proper use of access control is of utmost importance since it is crucial for managing the most critical functions within a smart contract. The combined losses due to access control vulnerability in 2022 and 2023 already exceed \$2 billion.

To understand how access control works, think of a security system in a high-tech building that decides who can enter certain zones, access valuable items, or modify important settings. Similar to this system, in smart contracts, access control guards important functions like generating of tokens, locking of funds, withdrawing of assets, approving proposals, or modifying of the contract's terms.

What is access control?

[Access control](#) involves implementing various protective measures such as:



Authentication

Verifying the identity of a user



Authorization

Ensuring that a user has appropriate access to resources



Accountability

Logging activities performed

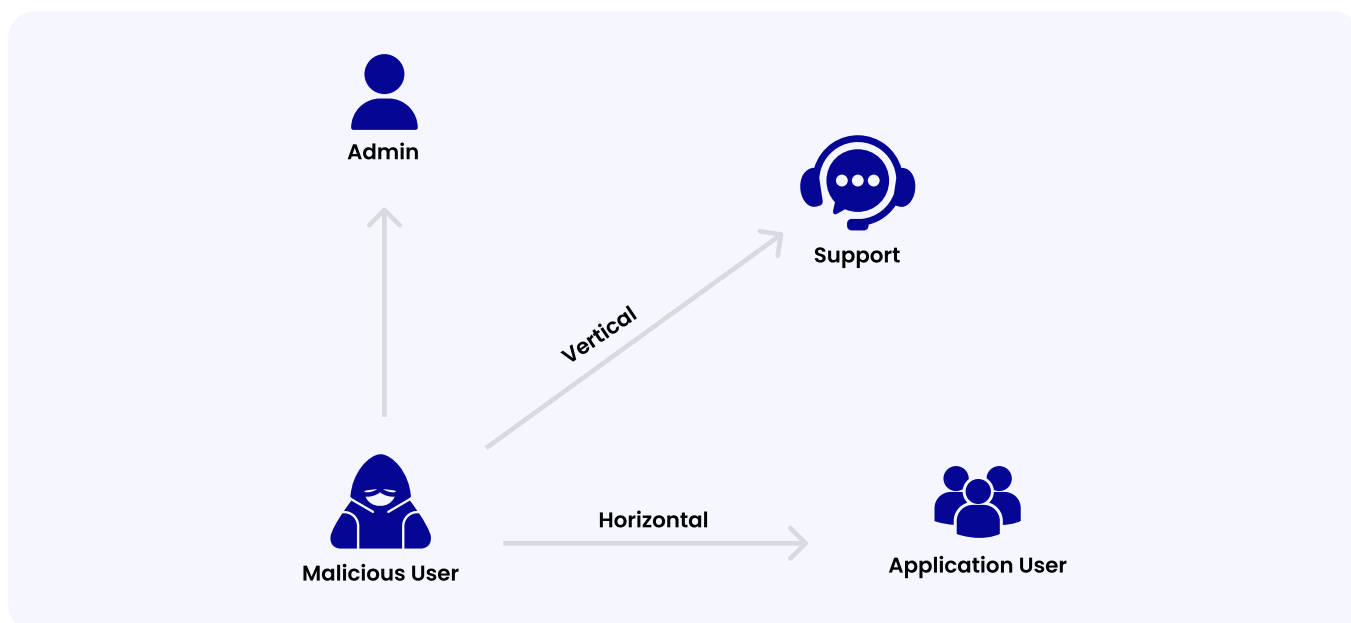
When a specific access function within a smart contract is left open or malfunctions, attackers leverage the opportunity and compromise protocol security by gaining unauthorized privileges, accessing sensitive data, executing commands, and avoiding detection.

Types of access control

Some common access control vulnerabilities:

- **Broken Authentication Flaw:** Such a flaw occurs when hackers bypass the authentication system due to developer oversights (for example, systems with bugs or poor configuration). These errors are very common and are exploited by hackers without extensive technical skills.
- **Vertical Privilege Escalation:** Happens when a hacker gains access to functions reserved for specific roles. For instance, they might use a mint function only available to administrators. The escalation is often achieved through metadata manipulation, breaching the principle of least privilege that ensures users only have access to necessary functions.

- **Horizontal Privilege Escalation:** Occurs when a user gains access to resources that belong to another user, instead of accessing only their own resources of that type. Hackers exploit a known flaw to gain access to functionalities with similar ownerships, like taking over other admin accounts after compromising one. This process allows them to gain the necessary permissions to sign off on critical actions like multi-sig transactions.



Mitigating Access Control vulnerabilities

There are two main causes of access control vulnerabilities:

- **Specification:** Incorrect privileges, permissions, or ownerships are not explicitly defined for either the user or the admin.
- **Enforcement:** Errors within the mechanism prevent proper enforcement of specified access control requirements. For example, allowing users to define their own privileges or allowing syntactically incorrect access control lists (ACLs) to create insecure settings. This issue arises within the program itself, failing to enforce the intended security policy specified by the administrator.

Access control attacks aren't exclusive to smart contracts; Some of the best ways to prevent these attacks are similar across different technologies. Developers need to configure access control carefully, creating clear distinctions between different privilege levels. When setting up these distinctions, developers should follow the principle of least privilege, ensuring users only have access to what they need for their role. Following the concept of least privilege, developers minimize the risk of security breaches.

In May 2023, [Deus DAO](#), a DeFi protocol offering options and derivatives trading, fell victim to a security exploit that resulted in the loss of approximately \$6.5 million.

The culprit behind the hack was a flaw in the code of Deus DAO's smart contract, specifically the `burnFrom` function. `burnFrom` was designed to allow users to burn (remove from circulation) their DEI tokens through another user (acting as a spender). However, as a result of a parameter reversal error, the function became susceptible to manipulation. The attacker found the vulnerability by altering the parameters transmitted to the `burnFrom` function. Post the identification, the attacker approved the spending of a large amount of another user's DEI tokens, even though the user never intended to grant such permission. Crucially, the attacker triggered the `burnFrom` function with a burn amount of zero. This counterintuitively resulted in the attacker being granted full access to the victim's DEI token allowance. With unauthorized access, the attacker then transferred the victim's DEI tokens to their own wallet, effectively draining a significant amount of funds from the protocol.

Reentrancy Vulnerabilities

In 2023, reentrancy vulnerabilities led to the theft of nearly **\$67 million** of users' funds. In 2022, almost \$441 million was stolen due to reentrancy gaps. Though there is a significant decrease in the amount stolen in 2023, reentrancy vulnerabilities are still one of the major concerns related to smart contracts. Attacks on platforms like Paribus and Sturdy Finance were among the worst incidents carried out by exploiting reentrancy gaps in smart contracts in 2023.

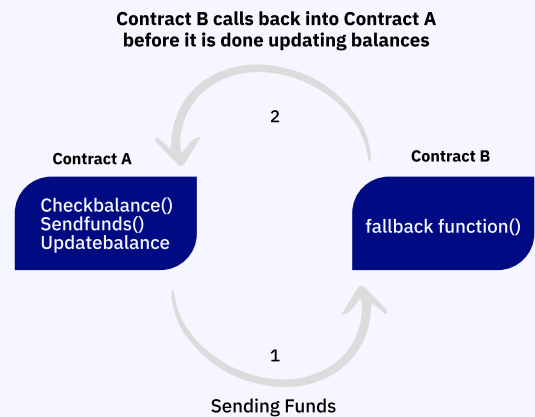
A [reentrancy](#) attack occurs when an attacker invokes a function in a vulnerable contract. This exploitation is caused due to the target (vulnerable) contract's failure to immediately update its state after a given function (such as withdrawal or transfer) is called. As a consequence, the attacker repeatedly invokes the same functions, leading the target contract to repeat the same action in a continuous loop.

Imagine using an ATM card to withdraw cash. If the ATM or the bank doesn't immediately reconcile your account balance after each withdrawal, it's like having a balance that never changes. You could keep withdrawing funds endlessly, and your account balance would stay the same. However, behind the scenes, you've actually withdrawn more money than you have, leaving you owing money to the bank.

Here's how a reentrancy attack works:

Let's consider 'Contract A' as the vulnerable contract and 'Contract B' as the attacker.

- The attacker initiates a call to 'Contract A' to transfer funds to 'Contract B'.
- 'Contract A' complies and transfers the specified amount of funds to 'Contract B'.
- After receiving the funds, 'Contract B' triggers a fallback function that loops back into 'Contract A' before it updates its balance, restarting the process.
- This cycle repeats until 'Contract A' is depleted of all its funds.



Exploring how a re-entrancy attack works

source: <https://hackernoon.com/hack-solidity-reentrancy-attack>

Types of Reentrancy Attacks



Single-Function Reentrancy: A single-function reentrancy attack occurs when a vulnerability exists within the same function that the attacker recursively calls. Contracts using functions like call, send, or transfer end up redirecting flow to an external contract or externally owned account with a fallback function. If the vulnerable contract fails to update its state, the contract's state remains incomplete even after the transfer. Consequently, triggering the fallback function causes control flow not to return to the previous state, allowing the caller to perform unexpected actions such as calling the function again or interacting with other contracts.

Cross-Function Reentrancy : Cross-function attacks occur when two different functions or contracts share the same state. Unlike single-function attacks where the vulnerable function is recursively called by the attacker, in cross-function attacks, the reentered function is not the one making the external call.



Cross-Contract Reentrancy: occurs when multiple contracts share the same state variable, and some contracts update that variable in an insecure manner. Cross-contract reentrancy is considered a complex issue because it is comparatively difficult to detect.

Read-Only Reentrancy: Read-only reentrancy occurs when a view function is reentered, potentially leading to incorrect state reporting. This deceives protocols relying on return values and prompts unwanted actions. Attackers exploit read-only reentrancy by manipulating prices, often seen in protocols integrating with DeFi platforms to access token prices or wrapped token values.



Mitigating Reentrancy Attacks



Updating Balance Before Transferring: Ensure that the contract's balance is updated before transferring funds to prevent multiple changes to the contract's state during a single transaction.

Re-entrancy Guard: Implement re-entrancy guards to prevent contracts from being called again before the previous call finishes executing, thereby preventing theft and malicious code execution.



Use of Mutex Locks: Employ mutual exclusion locks to prevent multiple threads from executing the same code simultaneously, thereby mitigating re-entrancy attacks.

Checking Call Stack Depth: Monitor the call stack depth to prevent stack overflows and vulnerabilities to re-entrancy attacks. Split large functions into smaller ones with fewer variables to avoid the "[Stack Too Deep](#)" error.



Use of Require Statements: Utilize "require statements" to ensure that a contract's state is updated before any external calls are made, preventing recursive calls to a withdraw function and fund drainage.

Use of Function Modifiers: Employ function modifiers to restrict access to certain functions, allowing only authorized users to call them and check the contract's state before executing transactions to prevent unexpected behavior.



Case Study: The Sturdy Finance Hack



The [Sturdy Finance](#) hack occurred on June 12, 2023, resulting in the loss of approximately \$800,000 worth of ether (ETH).

The attacker exploited a reentrancy vulnerability present in Balancer's system, combined with the manipulation of token B-stETH-STABLE price data. The attacker seized the opportunity to repeatedly call a function within a single transaction before completing the initial function call. Exploiting the loophole, the attacker managed to withdraw more funds than they were legitimately entitled to. Once in control of the function calls, the attacker successfully manipulated the price oracle, effectively draining funds from Sturdy Finance.

Arithmetic Issues

Arithmetic issues in smart contracts resulted in losses nearing **\$30 million** in 2023. Exploiting this vulnerability, attackers targeted platforms such as Level Finance, Jimbo Protocol, and Poolz Finance. The preceding year, in 2022, hackers capitalized on arithmetic issues within smart contracts across multiple platforms, leading to losses totaling nearly \$45 million.

Arithmetic problems, particularly integer overflow and underflow, are frequent vulnerabilities in smart contracts that arise when unsigned integers surpass their maximum value or drop below their minimum value, resulting in unexpected behavior due to rounding off.

For instance, if contract balance approaches the maximum uint value (2^{256}), it wraps back to zero, which poses potential risks depending on the implementation. It is therefore crucial to assess whether the uint value can reach such a large number. The same principle applies to underflow: If a uint is forced to be less than zero, it triggers an underflow and resets to its maximum value.

Both forms of arithmetic problems stem from assigning a value to a variable that surpasses or falls short of the data capacity. The overflow or underflow occurs due to the EVM mandating fixed-size data types for integers, limiting the range of representable numbers.

In typical scenarios, the fixed-size data range suffices. However, it falters in exceptional cases where individuals consciously or inadvertently try to store values exceeding the maximum or falling below the minimum.

Mitigating Reentrancy Attacks

For mitigation, the developers need to check how the uint variable changes state and who has the authority to modify it. Exercise caution with smaller data types like uint8, uint16, uint24, as they are more prone to hitting their maximum value. One simple solution to mitigate the common mistakes for overflow and underflow is to use SafeMath.sol [library](#) for arithmetic functions. In contrast to software, which are patched in real-time to address bugs, any flaws in smart contracts are essentially permanent once they are deployed. Hence, it is vital that smart contracts undergo thorough testing and refinement from the outset to rectify any issues before they are deployed.

Case Study: The Poolz Finance Exploit

In March 2023, Poolz Finance, a decentralized platform designed to streamline fundraising for crypto projects, fell victim to an exploit specifically due to a logic error known as an Arithmetic Overflow in its vesting contract which resulted in an improper integer handling during the calculation of claimable vested tokens from a vesting pool.

The attacker submitted a massive number of tokens to a vesting pool, off-shooting the intended upper limit for the data type used in the calculation. This triggered an arithmetic overflow, causing the value to wrap around to a minimal value. Following this, the smart contract perceived a significantly smaller amount of tokens as vested, enabling the attacker to withdraw a much larger quantity than they were rightfully entitled to.

The attack was carried out first on Binance Smart Chain and then repeated on Ethereum and Polygon, resulting in multiple stolen tokens. These tokens were then converted into 1,291 BNB (\$397,668), 54.2 ETH (\$91,216) and other assets like KMON, POOLz and MATIC bringing the total loss up to \$550,000.

Smart Contract Security and Threat Mitigation

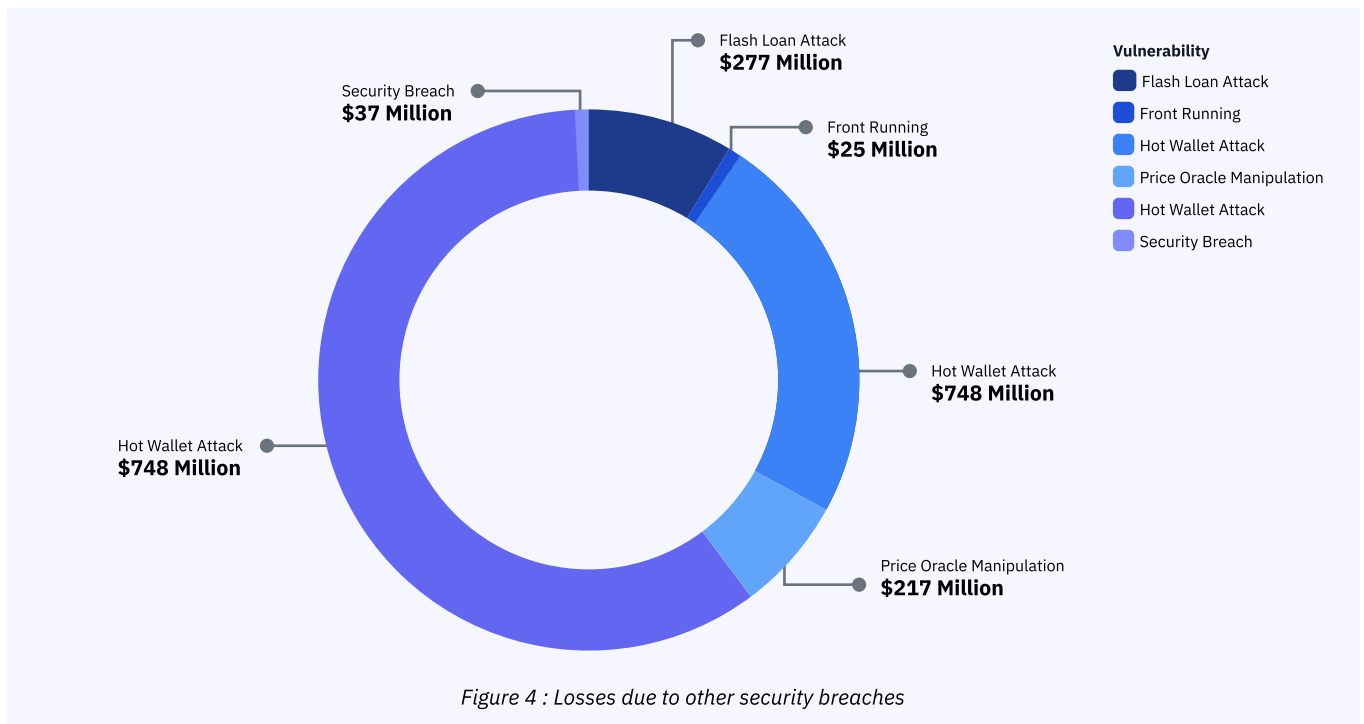
Smart contract security is increasingly pivotal not only within decentralized finance (DeFi) but also in the many centralized financial institutions that are beginning to leverage the benefits of smart contracts more extensively. As smart contracts become increasingly integral to the operations of CeFi platforms, the implications of their security vulnerabilities become a critical concern for the ecosystem as a whole.

Smart contracts execute automated actions based on predefined conditions, offering efficiency and transparency. However, without proper knowledge, understanding, and necessary security measures, smart contracts are susceptible to various vulnerabilities and exploits, from simple bugs to complex logical errors. For example, reentrancy attacks, integer overflow, or improper access control could allow attackers to drain funds from contracts, manipulate transaction values, or gain unauthorized access to financial operations.

Despite the positive trend of reduced smart contract losses observed in 2023, indicating progress in addressing security concerns, it remains one of the most pressing issues confronting the blockchain industry. In this section, we delve into the most common pitfalls associated with smart contract security and steps that help mitigate them. This section covers vulnerabilities that were exploited in 2023, along with those from previous years, as well as other typical traps that smart contract developers and protocols should be aware of.

As both traditional and crypto-native CeFi institutions continue to integrate smart contracts into their systems, understanding and mitigating the associated security risks becomes crucial. Compliance teams must be proactive in adopting sophisticated security measures and remain vigilant to the evolving attack vectors facing smart contracts across the ecosystem.

Hacks Due to Others Vulnerabilities



Private key compromise proved to be the most prevalent attack vector 2023, resulting in a significant loss of \$1.88 billion. Hot wallet vulnerabilities followed closely behind, leading to \$748 million in stolen funds. Flash loan attacks, a technique exploiting temporary liquidity, emerged as a growing threat, causing \$277 million in losses. Price oracle manipulation (\$217 million) and various other security breaches (\$37 million), contributed an additional \$254 million to the overall loss.

Hot Wallets Attacks in CeFi

In 2023, attacks on hot wallets emerged as one of the leading causes of losses, indicating an alarming trend. Over 22% of stolen funds, amounting to a staggering \$748 million, resulted from hot wallet breaches. Moreover, two out of the five biggest hacks of the year targeted hot wallets, emphasizing the severity of the issue.

Notably, in 2022, hot wallet attacks led to \$204 million in stolen funds. Losses due to hot wallet attacks increased by approximately 266% from 2022 to 2023. This massive jump shows that hackers are still capitalizing on older hacking techniques and continuously searching for overlooked security gaps to exploit.

While the cryptocurrency landscape has evolved with improved security measures and heightened awareness, hot wallet hacks in centralized finance (CeFi) platforms continue to pose significant challenges. Despite a reduction in such incidents over the years largely due to better security practices, 2023 witnessed a substantial rise in funds lost to these breaches, highlighting persistent vulnerabilities.

Hot wallets are cryptocurrency wallets designed to maintain continuous internet connectivity and real-time interaction with the network. In the context of a crypto business, these wallets are used for daily transactional purposes, enabling quick and efficient user transactions. Unlike cold wallets, which are stored offline and used for holding cryptocurrencies more securely over the long term, hot wallets provide the liquidity and functionality necessary for ongoing trading and exchanges on CeFi platforms.

However, the lack of air gapping of hot wallets often makes them more susceptible to various cybersecurity risks that may expose their private keys, leaving attackers with full access to any and all funds within the hot wallet.

Private Key Compromise

Private keys are the cornerstone of security in the cryptocurrency world. They serve as the digital signature for all transactions from a specific wallet or account, essentially acting as the owner's authorization to transfer funds. Unlike traditional banking systems where a lost password or PIN can be reset, a lost or stolen private key often means that the associated funds are irretrievably lost. Thus, the security of private keys is paramount in the cryptocurrency ecosystem, and understanding their management and protection is critical for anyone involved in the handling of digital assets.

A compromise of these keys gives malicious actors access to the associated wallets, leading to direct financial loss and significant damage to the reputation of the involved parties. Attackers exploit various methods to obtain private keys, including social engineering tactics, phishing attacks, the exploitation of software vulnerabilities, and the use of malware. Once a private key is obtained, attackers access the associated wallet's stored assets and transfer funds without the owner's consent.

Mitigating Private Key Compromises



Secure Key Creation and Storage: Generate keys in a secure, offline environment, and store them in encrypted formats. Employ cold storage for significant holdings and limit access to essential personnel.

Implementation of Multi-Factor Authentication (MFA): Secure key management systems with MFA to prevent unauthorized access by spreading authentication across multiple private keys.



Regular Security Audits and Employee Training: Conduct audits and vulnerability assessments regularly. Train employees on the importance of key security and awareness of common attack vectors.

Exercise Caution: Always exercise caution when sharing personal information or downloading any documents or software on a device that is used for conducting crypto transactions. Verify the legitimacy of sources, seek feedback from other users, and stay vigilant against phishing and online scams targeting crypto assets.



Additionally, innovative key management technologies that are growing in popularity amongst EVM chains include:



Smart Account (ERC-4337) Key Creation: Utilize ERC-4337 smart accounts, a new standard for creating user accounts on Ethereum without traditional private keys. These accounts operate using smart contracts that manage permissions and have the option to integrate recovery and security features directly into the account, enhancing security against key loss or theft.

Multi-Party Computation (MPC) Wallet Key Creation: Adopt MPC technology for key generation and management. MPC enables the division of a private key into multiple parts, with no single party accessing the complete key. This method not only secures the key from external threats but also mitigates risks of internal compromises.



Case Study: The Alphapo Exploit



On the 22nd of July, [Alphapo](#), a cryptocurrency payment gateway suffered a massive security breach leading to more than \$110 million being drained from its hot wallets on Ethereum and Tron blockchain.

The attack took place in the following steps:

1. The attacker stole \$101 million from Alphapo hot wallets on the Ethereum blockchain
2. The attacker then hopped over to the Bitcoin blockchain, where the funds moved to 67 newly minted bitcoin addresses
3. The attacker also gained access the platform's hot wallet on the Tron blockchain and stole over 118 million TRX tokens amounting to approximately \$9.5 million

The Rise of Private Key Compromise in DeFi

Private key compromises were the largest attack vectors of 2023, with nearly 56% of the total annual losses, reaching roughly \$1.88 billion. Private key compromises have historically been a significant concern for the industry. However, in the previous years, there had been a downward trend in reported incidents of private key compromises, suggesting improved security practices and heightened awareness among users. Despite these positive developments, private key exploitation resurged in 2023—particularly amongst DeFi players. Some of the biggest hacks of the year including Multichain hack, Poly Network and Heco Bridge were all caused due to private key compromises.

Learning from the various attacks on crypto exchanges in the early days of crypto, many CeFi institutions have developed robust security measures and best practices to protect their users' assets. The growing DeFi sector, on the other hand, has witnessed a significant rise in incidents related to private key compromises. In 2023, private key compromises in DeFi accounted for over 56% of the total losses from crypto hacks, highlighting the critical need for improved security measures.

Unlike CeFi, where the control and management of private keys are centralized, the operations of most DeFi projects are run on smart contracts. Smart contracts have no associated private key and cannot initiate transactions—they only execute operations, such as calling another contract or transferring funds, after an

externally owned account (EOA) interacts with them.

So, if smart contracts don't have private keys, how can the assets stored in them be stolen through private key compromise?

Many DeFi projects use smart contracts that are enabled to be controlled or upgraded by specific addresses known as admin addresses. If the private keys to these **admin addresses** are compromised, an attacker can execute transactions that could alter the smart contract's rules or redirect funds. This is exactly what we saw in the LianGo Exploit, which we detail in the case study below.

Mitigating private key compromise for these admin addresses would follow the same recommendations in our previous section covering hot wallet hacks.

Case Study: The LianGo Exploit



On February 7, 2023, there was a drastic drop in the value of the [LianGo](#) Protocol's token (LGT), indicating a problem with the system. Investigations showed that LGT tokens were moved from the LGT Pool contract, likely due to a compromised private key. The LP token address in the LGT Pool contract was altered by the LGT contract owner, leading to funds being drained into a malicious contract deployed by the exploiter.

During the attack, a significant amount of LGT was transferred to the exploiter, who then exchanged it for \$1,628,168.69 and moved it to another address. Subsequent attempts by other accounts to withdraw tokens from the LGT pool were unsuccessful due to depleted funds.

The attack took place in the following way:

- Attacker obtained the private keys of the LianGo Protocol's token (LGT) contract.
- Attacker created a malicious contract to carry out the exploit.
- A custom pool is created and a large quantity of fake LP tokens are deposited.
- The attacker then proceeds to increase the supply of the fake LP tokens and drains out the LGT pool using a manipulated reward contract.
- What followed was a transfer of the siphoned tokens to an address which further transferred the stolen \$1.6 million to Tornado Cash.

Flash Loan Attacks

Flash loans are a cornerstone feature of Decentralized Finance (DeFi). They allow users to borrow large sums of cryptocurrency instantly, without needing collateral. This rapid access to capital unlocks unique financial opportunities, but also introduces security risks. Prevalent even in 2022, the exploitation has contributed to 8.24% of the total loss of 2023. The total amount lost due to flash loan attacks in 2023 clocks at \$277 million.

The ease of obtaining flash loans makes them a prime target for malicious actors. These attackers exploit vulnerabilities in DeFi smart contracts to steal funds through "flash loan attacks." These attacks are popular because they require minimal technical expertise and resources compared to other hacking methods.

While attackers exploit them, flash loans also offer legitimate benefits. They enable complex financial maneuvers, such as arbitrage opportunities. Arbitrage involves taking advantage of price discrepancies between different markets to make a profit. Users borrow funds through a flash loan, execute a trade that makes use of a price difference, and repay the loan within the same transaction, all with minimal upfront investment.

Case Study: The KyberSwap Exploit



In November 2023, KyberSwap Elastic, the decentralized exchange's concentrated liquidity protocol, experienced a hack. By exploiting a vulnerability in the protocol, the attacker stole an estimated \$44.8 million.

The KyberSwap hacker took advantage of a weakness in the protocol's concentrated liquidity calculations. Performing similar mathematical operations in two different ways, and assuming they would yield the same result, introduced a vulnerability. This vulnerability allowed for a meticulously crafted swap to drain the liquidity pools of the protocol.

The stolen funds were transferred through multiple currencies including ETH \$7.5 million, ARB \$20 million, OP \$15 million, MATIC \$2 million, AVAX \$23,493 in tokens. On BASE, KyberSwap Exploiter stole over \$315K in tokens, including USDC, WETH.

Analysis of multiple flash loan exploits reveals common attack vectors:

- **Arbitrage Ops:** Attackers use flash loan capital to conduct risk-free arbitrage trades across decentralized exchanges. By exploiting lag in price feeds or temporarily distorting prices, they generate profit, and siphon liquidity from affected pools.
- **Oracle Manipulation:** Attackers influence asset prices significantly in the short-term by conducting a series of large trades using flash loans. They then provide manipulated price data to the oracle feeds of lending protocols to get favorable collateral ratios or increased loan amounts.
- **Draining Pools:** Attackers identify flaws in the token bonding curves or reward distribution conditions set in smart contracts governing liquidity pools. Flash loans allow them to construct complex transactions that unfairly drain funds from the pools.

These attacks succeed mainly due to technical vulnerabilities in smart contract code and business logic. Measures like formal verification, modular architecture, and comprehensive audits during development improves resilience.

Oracles and Price Manipulation Attacks

Blockchain oracles connect off-chain data sources with on-chain DeFi platforms to feed external information to transaction logic and smart contract code. While 2022 saw \$249 million hacked due to price oracle manipulation, the figures in 2023 total at \$217 million, exactly 6.46% of the total amount lost.

Assessing Centralization Risks of Blockchain Oracles

Single Point of Failure: Relying completely on one centralized oracle creates a bottleneck in data flows towards contracts. Outages at the oracle lead to platform failures. Attackers also target it to provide false data or conduct denial of service.

Data Source Centralization: Even decentralized oracle networks often source from the same APIs, market feeds, and web endpoints. Compromising the few underlying common sources essentially puts all data at risk.

Mitigating measures would involve diversifying and independently verifying data sourced from multiple types of APIs, feeds, sensor networks and web sources. On-chain verification of data signatures adds more robust authentication as well. This reduces the chances of widespread data manipulation.

The risks around flash loans and oracles described need to be managed proactively to secure the DeFi ecosystem.

Case Study: BonqDAO Exploit

BonqDAO, a decentralized autonomous organization (DAO), experienced a severe smart contract breach, leading to an approximate loss of \$120 million. The exploit stemmed from an oracle hack that allowed manipulation of AllianceBlock (ALBT) token prices within Bonq's protocol. The attacker artificially inflated the price of ALBT, resulting in the creation of a significant amount of BEUR tokens, which were subsequently exchanged on Uniswap. As a consequence, the value of ALBT plummeted, causing the liquidation of assets.

MEV and Sandwich Attacks

In 2023, one Maximal Extractable Value (MEV) bot stole more than \$25 million through a single front-running attack alone. MEV represents both an operational characteristic and a potential vulnerability within the DeFi ecosystem. While inherently not illicit, MEV involves the extraction of value from blockchain transactions during the production of blocks. It's a complex topic where ethical and functional aspects intertwine, especially when discussing forms like front running and sandwich attacks.

MEV or **Maximal Extractable Value** remains a nuanced aspect of DeFi, embodying both necessary market functions and potential avenues for exploitation. Understanding its mechanisms and impacts is crucial for both users and developers within the ecosystem. Employing robust mitigation techniques enables the DeFi community to safeguard participants from malicious practices while preserving the efficiencies that MEV can offer.

Front running in the context of MEV occurs when someone (usually a MEV Searcher operating through an autonomous bot) sees a lucrative pending transaction intended for the next block to be added to the blockchain. (e.g., a large trade on a DEX) and places their own transaction ahead of it with a higher gas fee to profit from the

anticipated price movement. This, in turn, increases the swap price for the target transaction conducting the initial trade resulting in a less favorable outcome for the end user. The MEV Searcher then immediately sell the tokens directly after the victim's swap as they have further moved the price of the token, netting a substantial profit for the Searcher. This is known as a **sandwich transaction** or, more commonly, a **sandwich attack**.

The role of MEV in DeFi

While MEV is often viewed in a negative light due to predatory practices such as sandwich transactions, it is also a necessary component of DeFi. Arbitrage, for instance, helps in maintaining the price parity across different exchanges or trading platforms, thereby contributing to market efficiency and liquidity. It's important to recognize that not all activities classified under MEV are harmful or unethical; many are integral to the proper functioning of decentralized markets.

The primary targets of these more predatory types of MEV are end users engaging in swaps or trades on decentralized platforms. Simple transfers of funds (i.e., sending cryptocurrency from one wallet to another without the intention of trading) are not affected by MEV. New or uninformed DeFi users are especially vulnerable as they might not be aware of the potential for such activities or know how to protect against them. Unfortunately, accurately estimating the total amount truly lost to MEV is challenging, if not impossible, for many factors, such as needing to accurately estimate how a transaction would have settled if it had not been frontrun and identifying every single MEV bot transacting on the blockchain to understand which transactions were targeted in the first place.

Mitigating MEV

To protect against MEV, both users and DeFi platforms can adopt several strategies

For Users:



Using Slippage Tolerance: Setting a maximum slippage tolerance can prevent transactions from being executed at undesirably high prices.

Timing Transactions: Avoiding times of high network congestion can reduce the visibility and vulnerability of transactions to front running.



Regular Security Audits and Employee Training: Conduct audits and vulnerability assessments regularly. Train employees on the importance of key security and awareness of common attack vectors.

Private Transactions: Some services offer the ability to hide transaction details until they are executed, minimizing exposure to predatory MEV strategies.



For Protocols/Projects



Fair Sequencing Services: Implementing solutions that offer fair ordering of transactions can help mitigate the advantage that arbitrage bots and front runners might have.

Improved Transaction Batching: Batch processing transactions in a way that obscures the details and intentions of individual trades can reduce the likelihood of being targeted by MEV strategies.



MEV-Resistant DeFi Designs: Designing protocols that inherently resist MEV by using mechanisms like frequent batch auctions instead of continuous order books can significantly decrease the potential for exploitative MEV.

Case Study: Sandwich the Ripper



On April 3rd, a sophisticated attack unfolded on the Ethereum network, orchestrated by a malicious validator known as "Sandwich the Ripper." This incident exposed vulnerabilities in the handling of MEV bundles (bundles of MEV transactions sent from MEV bots directly to Ethereum Block Builders for inclusion on-chain), leading to significant financial repercussions.

The malicious validator executed a carefully planned strategy by setting up honeypot transactions within the public mempool. These transactions were designed to appear highly lucrative from an MEV perspective, thereby attracting MEV Searcher bots. The vulnerability exploited was in the MEV Relay operations (MEV Relays play a pivotal role on Ethereum post Proposer/Builder Separation, transferring blocks from Block Builders to Validators where they can then be added to the blockchain). At the time, MEV Relays did not verify certain block header parameters—specifically, the parent and state root. Taking advantage of this oversight, "Sandwich the Ripper" managed to manipulate the transactions by intercepting and altering them before they were included in a new, validly proposed block.

Despite the complexity and initial success of the attack, which netted approximately \$25 million, the malicious validator was eventually penalized, being slashed 32 ETH for their actions and booted from the network. However, the financial damage inflicted on the victims was substantial, highlighting a critical security lapse in the system.

The "Sandwich the Ripper" attack serves as a stark reminder of the vulnerabilities inherent in decentralized networks, particularly those involving complex transaction mechanisms like MEV. The swift response to shore up security in the aftermath of the attack reflects the ongoing challenges and adaptations needed to secure blockchain ecosystems against evolving threats.

Crypto Laundering Techniques and Trends

The rise of DeFi technologies, such as decentralized exchanges (DEXs) and cross-chain bridges, have notably eased the transfer of capital across different blockchain networks. Nevertheless, beyond their legitimate applications in decentralized trading, payments, and other financial transactions, criminals leverage the perceived anonymity of DeFi transactions to move billions of dollars in cryptocurrency across diverse assets and blockchains.

However, the use of DeFi protocols has often proven ineffective as a way of concealing illicit flow of funds, and instead serve more as an effective way to transfer low liquidity stolen assets into higher liquidity assets that would be easier to off-ramp.

In 2023, illicit actors majorly exploited bridges, swap services and coin mixers to launder and off-ramp their illicit gains. In this section, we highlight and analyze the most frequently used laundering techniques employed by bad actors across the crypto ecosystem.

The Use of Cross Chain Bridges

Intended Use: Swap assets between different blockchains **Illicit Use:** Chain hopping in attempt to obfuscate flow of funds

A cross-chain bridge is a protocol designed to enable the transfer of digital assets between independent blockchains. Below is a high-level breakdown of how a typical cross-chain bridge works behind the scenes. This explanation provides a high-level technical overview. The specific implementation details vary significantly depending on the bridge's design and underlying protocols.

- 1. Initiating the Transfer:** The process begins with a user's request to transfer assets from the source chain (Chain A) to the destination chain (Chain B).
- 2. Smart Contract Interaction (Chain A):** A smart contract on Chain A receives the transfer request for a designated amount of tokens. The token contract locks or burns the tokens, essentially removing them from circulation on Chain A.
- 3. The smart contract on Chain A logs following details in the transaction:**
 - Transfer amount: Amount of tokens being transferred.
 - Destination address: Recipient's address on Chain B.
 - Unique identifier: To track the transaction across chains.
- 4. Cross-Chain Communication:**
 - Message Delivery: Oracles or validators monitor for the locking of tokens on the originating blockchain. These oracles or validators verify the proof submitted by the user and ensure that the tokens are indeed locked and ready for transfer.
- 5. Relay nodes on Chain B:**
 - Verify the data packet's authenticity and integrity.
 - Ensure sufficient funds are locked on Chain A to cover the transfer.

6. Mint or Release/unlock: Based on the bridge type:

- **Trust-based bridges:** A custodian entity holds the locked assets on Chain A. Upon receiving the validated data, the custodian releases an equivalent amount of tokens on Chain B.
- **Trustless bridges:** A smart contract on Chain B mints a new token representing the transferred asset. This minted token is often referred to as a wrapped token (e.g., wBTC on Ethereum).

7. Transaction Finalization: The minted tokens (or released assets) are credited to the recipient's address on Chain B. The successful transfer is reflected on both Chain A's (locked tokens) and Chain B's (minted/released tokens) ledger states.

Case Study: LastPass Hack

The LastPass hack of 2023 might not be one of the biggest hacks of the year, but it is interesting due to its laundering technique. Around \$4.4 million was stolen across Ethereum, Polygon, and Binance Smart Chain. Most of these funds ended up in prominent exchanges on Ethereum, BSC, and Polygon. However, a portion, roughly 816.5 ETH, was routed through the ThorSwap cross-chain bridge to the Bitcoin blockchain, where it was spread across multiple Bitcoin addresses (cf. Figure 5).

In another instance from the same exploit, some of the stolen proceeds were first converted into approximately 11 wrapped Bitcoin (wBTC) on the Ethereum blockchain before transitioning to the Bitcoin blockchain in their unwrapped form. Once again, the funds were dispersed across multiple addresses.

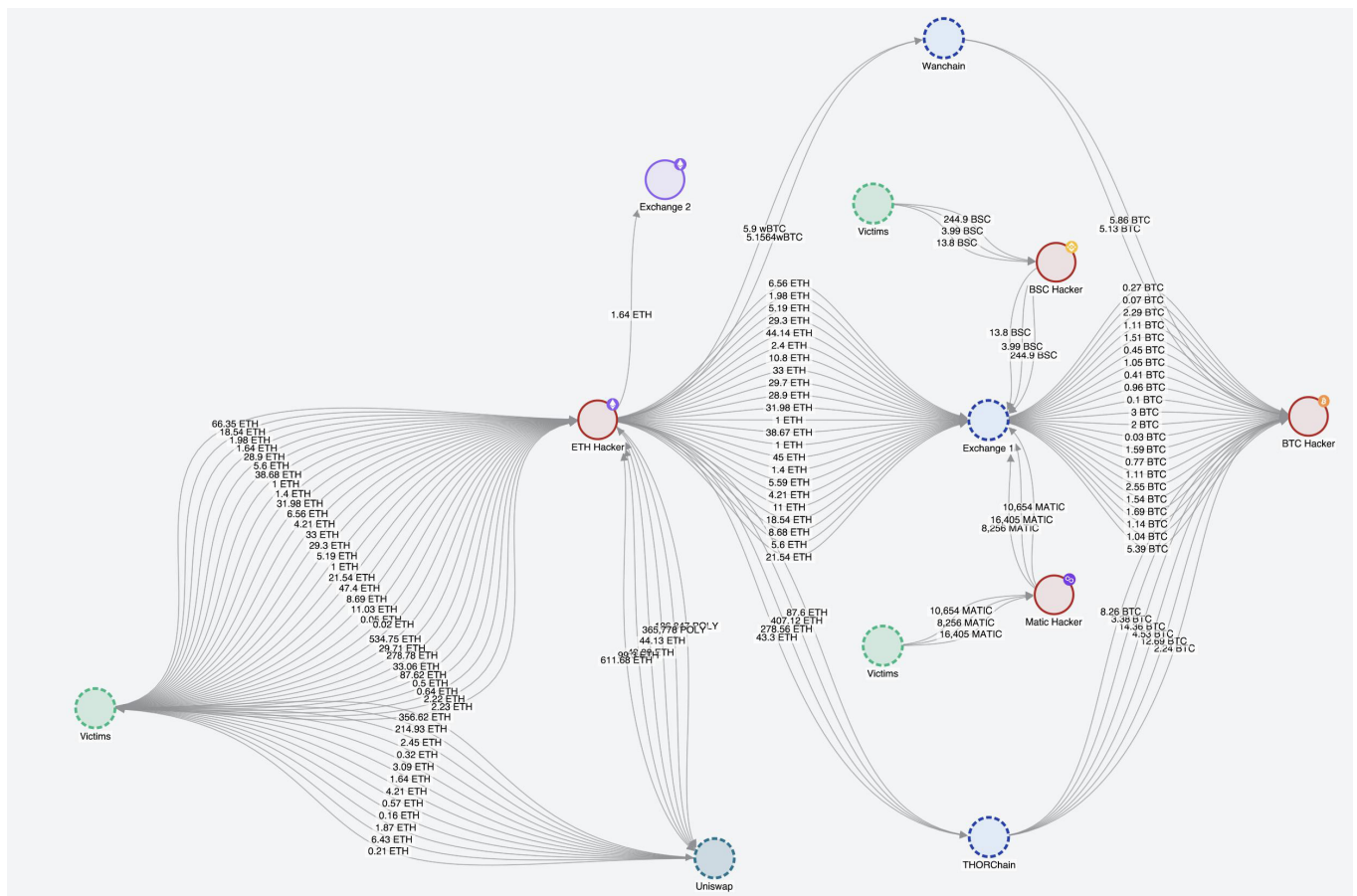


Figure 5 : Hackers sending funds to bridges to hop from one blockchain to another

Use of Coinjoins and Mixers

Intended Use:

Enhance privacy of blockchain transactions by obscuring the source/destination of funds.

Illicit Use:

Obfuscating the trail of stolen assets

CoinJoins and mixers are often used to obfuscate the origin and destination of cryptocurrency transactions. While this process is intended to be used to enhance financial privacy in an open blockchain system, bad actors can also reap the benefits of these privacy enhancing tools to assist with obfuscating their flow of funds as they attempt to launder and off-ramp their illicit gains.

Both mixers and CoinJoins serve to enhance privacy by obfuscating the origins and destinations of funds on the blockchain, but they operate very differently and are suited to different types of blockchain technologies.

Mixers, also commonly referred to as tumblers, are services that take cryptocurrency from multiple users, mix the funds together (often in a single hot wallet), and then redistribute it to users in different amounts from different sources, making it difficult to trace the original funds. Mixers can be standalone, centralized services or integrated into other platforms. Unlike mixers operating on the Bitcoin blockchain, mixers on EVM-based chains like Ethereum operate as decentralized smart contracts that cannot be shut down or seized. This is the case with Tornado Cash, which we will discuss further in our case study below.

Mixers often require trusting a third party, which could be malicious or potentially saving logs of where you are moving funds to, and the degree of obfuscation depends on the volume of coins mixed and the policies of the mixer.

A CoinJoin, on the other hand, is a method specific to UTXO (Unspent Transaction Output) cryptocurrencies like Bitcoin. It involves multiple parties combining their transactions into one large transaction with multiple inputs and outputs. By doing so, they obscure which outputs belong to which inputs, complicating the task of tracing individual transactions. CoinJoins do not rely on an external service or additional fees beyond standard transaction fees, making them distinct from mixers. However, because the transactions that make up the inputs in a CoinJoin also make up the outputs, they are often far easier to trace and link back to the original source of funds.

Both methods still face challenges in providing complete anonymity, especially against sophisticated blockchain analysis techniques that can sometimes de-anonymize such transactions.

For money launderers and other criminals, leveraging these mixing services enables them to convert tainted crypto linked to unlawful activities into seemingly clean coins. One of the most widely utilized mixers, Tornado Cash, faced sanctions from the U.S. Department of the Treasury's Office of Foreign Assets Control (OFAC) in August 2022 due to its involvement in extensive money laundering activities. However, this did not stop criminals from using their services as, even in 2023, some of the largest attacks found their ways to coin mixers like Tornado Cash and Sinbad.io, despite regulatory actions.

Case Study: Tornado Cash

Tornado Cash continues to hold a prominent reputation among those who wish to obfuscate stolen funds and 2023 was no different. The image below illustrates how Tornado Cash was used to mix stolen funds from multiple incidents such as Rodeo Finance exploit, Conic Finance Exploit and Arcadia Exploit etc (cf. Figure 6).

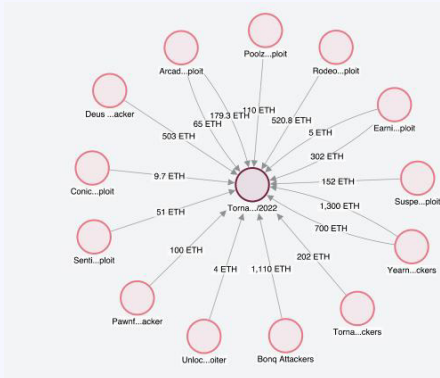


Figure 6 : Hackers sending funds to Tornado Cash after the theft

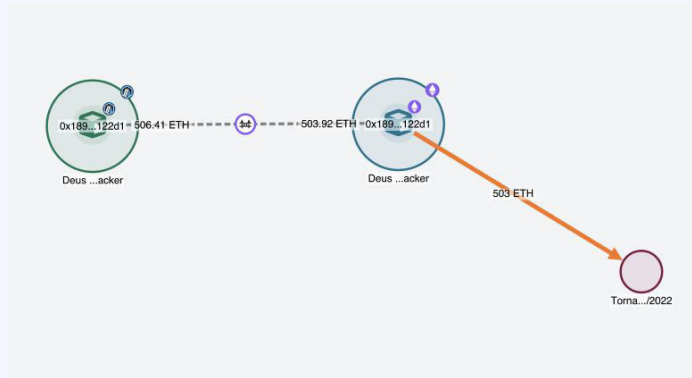


Fig 7 : Deus DAO attacker sent funds to Tornado Cash

What attracts our attention further, is the combination of more than one method to launder funds. As shown in figure 7, in the case of Deus Finance Exploit of 2023 where a portion of the funds, approximately 506 ETH, transcended from the Arbitrum layer-2 blockchain into the Ethereum Mainnet and then sent into the coin mixer Tornado Cash. Such usage of a combination of laundering techniques poses a significantly larger threat in our efforts to assist with the tracking and recovery of stolen funds.

Use of Dapps for Token Swap

Intended Use:

Exchange crypto without the need to sign up for an account first, or provide any identity documents

Illicit use:

Move stolen funds from one blockchain to another without KYC or other scrutiny

Decentralized Exchange (DEX) Model:

Coin swap platforms primarily operate as DEXs, facilitating peer-to-peer (P2P) cryptocurrency exchanges without the need for a central intermediary.

Transaction Mechanism:

- 1. User initiates swap:** A user selects the desired tokens for exchange (e.g., swapping ETH for USDT).
- 2. Liquidity Pools:** Instead of directly matching buyers and sellers, coin swaps rely on liquidity pools. These are smart contract-managed reserves containing one or more pairs of tokens (e.g., ETH and USDT).
- 3. Automated Market Maker (AMM) Algorithm:** Swap platforms utilize an AMM algorithm to determine the exchange rate between the tokens in a liquidity pool. Common AMM algorithms include:
 - Constant product market makers: Maintain a constant product (price x quantity) for the token pair.
 - Constant sum market makers: Maintain a constant total value locked (TVL) in the pool.

4. Trade execution: Based on the user's swap request and the AMM formula, the platform:

- Deducts the desired amount of one token (e.g., ETH) from the pool.
- Credits the user with the equivalent amount of the other token (e.g., USDT) based on the calculated exchange rate.

Technical aspects:

- **Smart contracts:** Play a crucial role in managing liquidity pools, executing trades based on the AMM algorithm, and ensuring secure and transparent transactions.
- **Oracles:** In some cases, external oracles might be used to feed external price data into the AMM algorithm, especially for assets without readily available on-chain price information.

Additional functionalities:

- **Fees:** Platforms typically charge a small fee on each swap to incentivize liquidity providers and cover operational costs.
- **Staking:** Some platforms offer staking options where users lock their tokens to earn rewards.

In 2023, illicit actors heavily relied on coin swaps to covertly exchange assets. Most laundering activities involved utilizing swap protocols such as Uniswap, Sushiswap, 1inch, Multichain, PancakeSwap, Curve, Bancor, and Balancer.

Case Study: Decentralized Exchanges



The example below illustrates how swap services have been utilized in the past in an attempt to obscure funds obtained from hacks and exploits. Unlike exchanges, swap services don't necessitate hosting wallet services and thus don't require KYC from customers. However, this lack of KYC is not synonymous with heightened anonymity, as all transactions on a DEX are completely visible in the public blockchain. The same is not true for a CEX, where the flow of funds is broken and a subpoena is necessary to uncover which transactions have occurred after criminal funds have entered their hot wallet.

DEXs do, however, enable exploiters to convert stolen funds—especially low liquidity tokens— into higher liquidity tokens like USDT, USDC, ETH, or TRX, which make them much easier to move and off-ramp. In 2023, the use of swap services continued to surge, with attackers employing them in over 90% of hacks to funnel stolen funds. (cf. Figure 8)

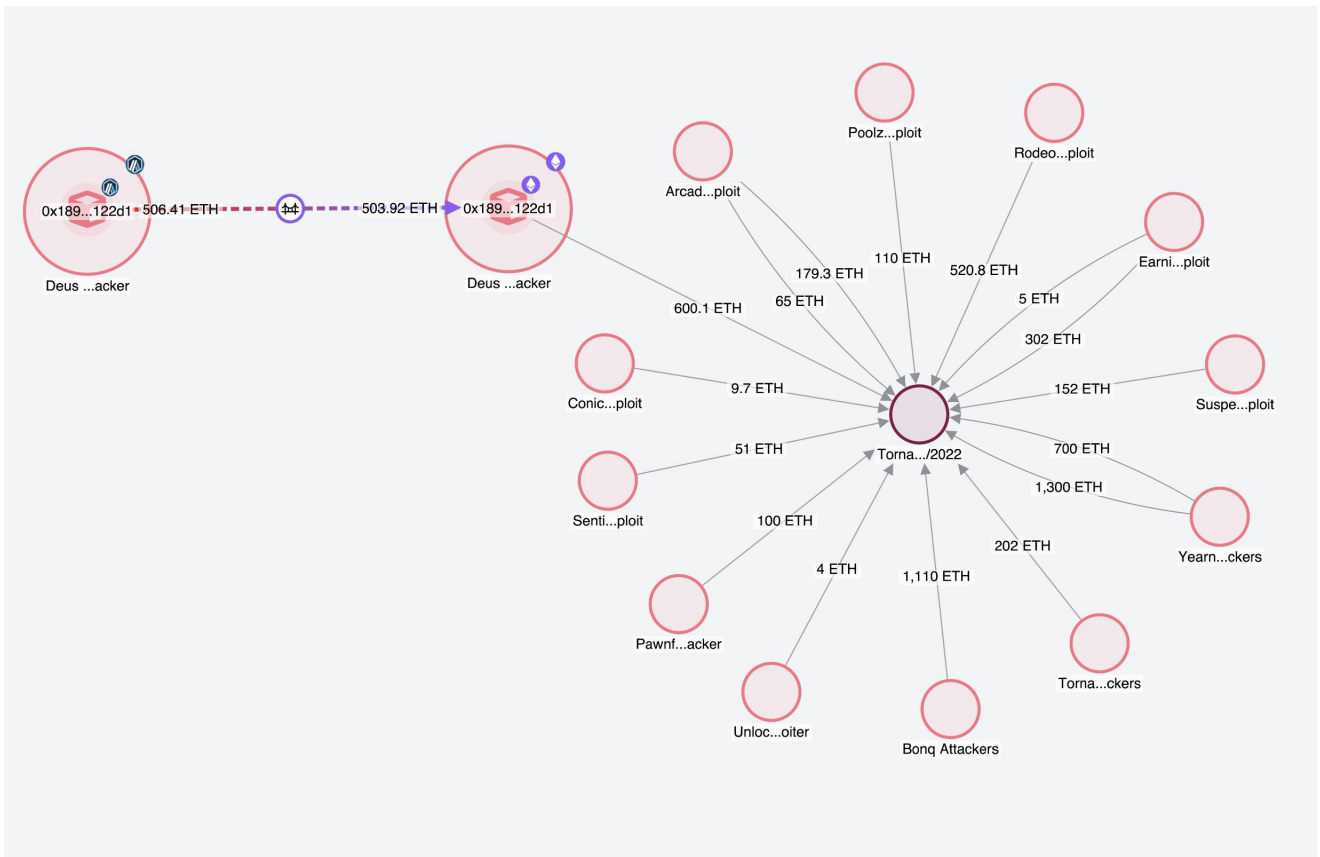
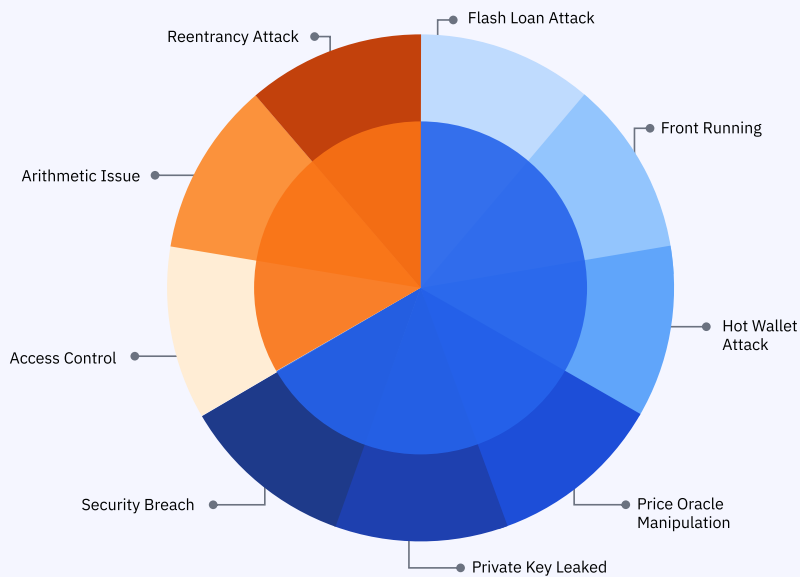


Figure 8 : Funds from multiple hacks being sent to swap services to hop to other blockchains smoothly without customer due diligence

HACKHUB Methodology



Based on secondary research such as articles, news and social posts released by experts of the industry, representatives of the hacked platforms, and Merkle Science’s in-house investigations, we were able to classify the hacks and abuses of the year 2023 in the 2 broad categories of

1. SC Vulnerability: Where the hacks/thefts/abuses have occurred directly and solely due to the exploitation of a smart contract vulnerability.

2. Other: Where the hacks/thefts/abuses have occurred either

- due to other security gaps (Hot Wallet Hacked, Security Breach)
- Indirectly due to a smart contract vulnerability which has resulted in other forms of exploitation. (Flash Loan, Price Manipulation)

SC Vulnerability/Other	Subcategory
SC Vulnerability	Access Control
SC Vulnerability	Arithmetic Issue
SC Vulnerability	Reentrancy Attack
Other	Flash Loan Attack
Other	Front Running
Other	Hot Wallet Attack
Other	Price Oracle Manipulation
Other	Private Key Compromise
Other	Security Breach

Smart Contract Vulnerability

Within the smart contract vulnerabilities, DASP has categorized smart contract vulnerabilities into the following subcategories:

- 1. Reentrancy:** A malicious external contract executes a recursive call on the victim smart contract in a virtually endless loop.
- 2. Access Control:** A vulnerability that allows an attacker to manipulate the validation/verification step thereby replacing ownership of the contract.
- 3. Arithmetic:** Integer overflow/underflow that is a resultant of the use of unsigned integers in the smart contract code.
- 4. Unchecked Low-Level Calls:** This occurs when the creator fails to check the validity of low-level calls that can result in unexpected behavior in the program logic.
- 5. Denial of Services:** When an attacker can potentially block the functioning of a smart contract.
- 6. Bad Randomness:** Occurs when a smart contract uses randomness, it is often predictable. Hence a miner or attacker can guess, crack/manipulate this randomness to attain favorable outcomes.
- 7. Front Running:** It occurs when an attacker manipulates the order of transactions to enable the transaction of interest to occur before.
- 8. Time Manipulation:** A malicious miner can report an incorrect time of the transaction so that the transaction ends up in the block he is mining to attain favorable results.
- 9. Short Addresses:** When an attacker bypasses the required parameter with a shorter than expected parameter, however, the Ethereum Virtual Machine appends '0' at the end thereby facilitating a bypass of the authentication process.
- 10. Unknown:** Any other smart contract vulnerability can be classified under unknown.

Other

- 1. Flash Loan Exploit:** Flash loan exploits usually occur due to smart contract vulnerabilities. However, since the attacks affect a different bridge/platform than the lending platform, these attacks have been classified as Flash Loan Exploits.
- 2. Price Oracle Manipulation:** Smart contract vulnerabilities facilitate borrowing of an unsecured flash loan and the flash loan in turn facilitates the attacker to manipulate the price of a token. However, since the final cause of the exploit is a price manipulation of tokens in one way or another, we have categorized such hacks as Price Oracle Manipulation.
- 3. Hot Wallet Hacked & Private Key Compromise:** Both Hot Wallet Hacks and Private Key Compromises are similar in nature. However, if an attacker gains access to the hot wallet that can be air gapped, we have categorized it as 'Hot Wallet Hacked' whereas if the private keys of entities (including smart contract like in the case of attacks on bridges) are stolen or taken over, we have classified such attacks as 'Private Key Compromises'.
While both Hot Wallet Hacked and Private Key Compromises are a type of access control, we have specifically segregated them for ease of understanding.
- 4. Other Security Breach:** All other non-smart contract-related security and infrastructural breaches are classified under 'Other Security Breach', which are inclusive of but not limited to 2FA failure, frontend manipulation, and fake account creation.
As per our investigation, some hacks are combination of 2 or more categories/subcategories of hack classification, however, we have attributed the final cause of the hack as the basis of classification

Note: To provide the most complete picture possible, this report includes some reported hack incidents, even though their loss amounts may vary from our independent analysis. We encourage you to review all data points for a well-rounded understanding.

Glossary

- **Air drop:** Free tokens distributed to cryptocurrency wallet addresses.
- **Air gap:** A security measure isolating a system from the internet.
- **Algorithms:** Defined sets of instructions used in cryptography, mining, and other blockchain functions.
- **Arbitrage:** Taking advantage of temporary price discrepancies between different markets.
- **Block reward:** Cryptocurrency awarded for adding a new block to the blockchain.
- **Chain hopping:** Moving cryptocurrency assets between different blockchains.
- **Collateral:** Assets deposited as security for a loan in DeFi protocols.
- **Customer due diligence (CDD):** Verifying a customer's identity and assessing risks.
- **DAO (Decentralized Autonomous Organization):** A community-governed organization on a blockchain.
- **Flash loans:** Uncollateralized loans in DeFi that must be repaid within the same transaction.
- **Floor price:** The lowest potential price point at which an NFT might sell.
- **Gas:** The fee required to execute transactions on a blockchain network.
- **Layer 0 blockchain:** The underlying infrastructure layer for blockchains.
- **Layer 1 blockchain:** The main blockchain where transactions and smart contracts reside.
- **Layer 2 blockchain:** Blockchain solutions built on top of layer 1 chains for scalability.
- **Liquidity pool:** A collection of cryptocurrency assets for trading on decentralized exchanges.
- **Mempool:** A temporary pool where pending transactions wait before being included in a block.
- **MEV (Miner Extractable Value):** Additional profit miners extract by manipulating transaction order.
- **Mnemonic phrase:** A list of words used in sequence to access crypto assets.
- **Multi-sig (multi-signature):** A security feature requiring multiple approvals for transactions.
- **Non-custodial wallet:** A type of cryptocurrency wallet where users hold the private keys.

- **Multi-sig (multi-signature):** A security feature requiring multiple approvals for transactions.
- **Non-custodial wallet:** A type of cryptocurrency wallet where users hold the private keys.
- **OTC (Over-The-Counter):** Trading digital assets directly between two parties.
- **Payment gateway:** A platform enabling merchants to accept cryptocurrency payments.
- **Reentrancy guard:** A security feature in smart contracts preventing theft through multiple function calls.
- **Slippage:** The difference between the expected and actual price of a trade.
- **Stablecoin:** A cryptocurrency pegged to a stable asset (e.g., USD) to minimize price fluctuations.
- **Wallet provider:** A service that allows users to store, send, and receive cryptocurrency.
- **zkp (Zero-knowledge proof):** A cryptographic technique for proving information without revealing it.

Reference

- <https://csrc.nist.gov/glossary>
- <https://dasp.co/index.html>
- <https://swcregistry.io/>
- <https://consensys.github.io/smart-contract-best-practices/attacks/>
- <https://scsfg.io/>
- <https://docs.soliditylang.org/en/v0.4.21/bugs.html>
- <https://github.com/pcaversaccio/reentrancy-attacks>
- <https://dev-docs.poly.network/GLOSSARY.html>