Cybersecurity Issues and Countermeasures in Crypto-Asset-Related Businesses

Research paper

June 30, 2026

Deloitte Tohmatsu LLC

Acknowledgments and Disclaimer

Acknowledgments

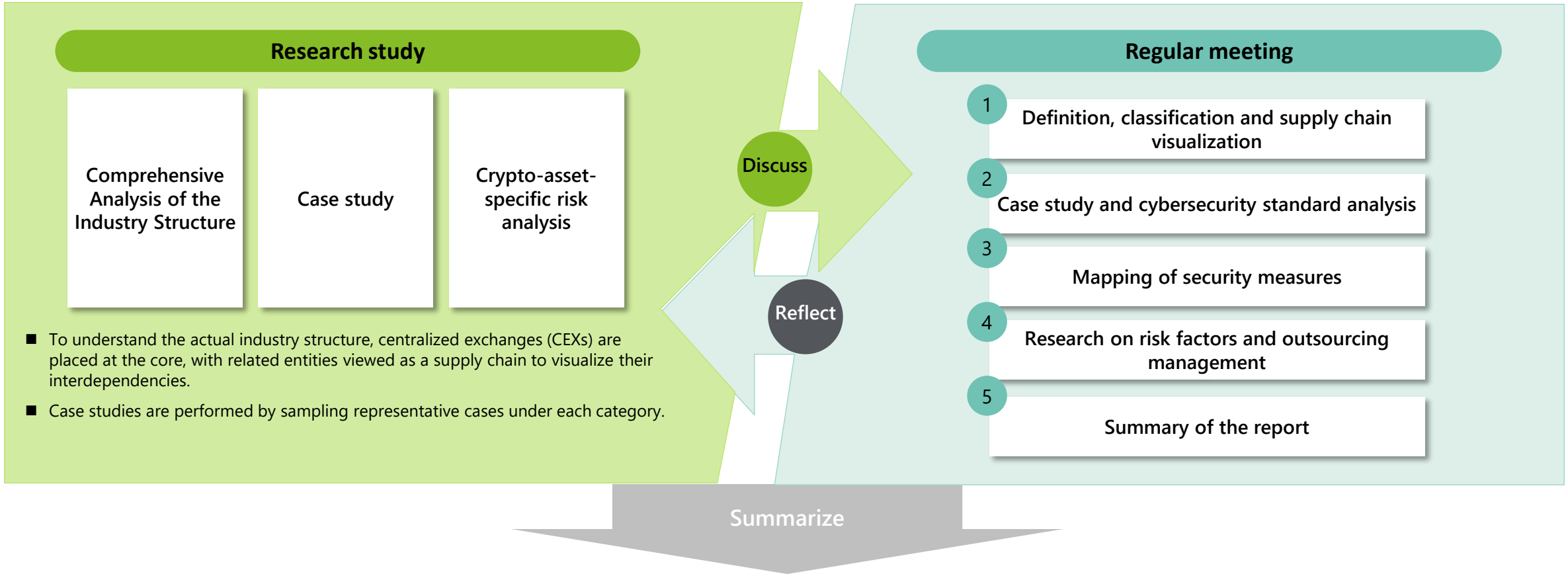
- In the preparation of this report, we received valuable advice and comments from Professor Emeritus Naoyuki Iwashita of Kyoto University, and Research Professor Shinichiro Matsuo of Georgetown University. We also received insightful suggestions and advice from observers at the Bank of Japan and the Digital Agency, as well as officials from the Financial Services Agency.
- Furthermore, we drew upon analysis reports and insights obtained through individual interviews with several companies, including Fireblocks, Thales DIS Japan K.K., and DFNS. In addition, we conducted interviews with organizations and business operators related to incidents referenced in this report, and greatly benefited from their valuable advice and comments.
- We also conducted interviews with some related crypto-asset service providers regarding the details of the cyberattacks.
- In February 2026, at the JFSA Blockchain Roundtable held during Japan Fintech Week 2026, we received valuable advice from the participating experts. In March 2026, we also received valuable advice from participants at the Blockchain Governance Initiative Network (BGIN) Block #14 meeting.
- Nevertheless, any errors in this report are the sole responsibility of Deloitte Tohmatsu LLC, the contractor.

Disclaimer

- The contents of this report do not represent the official views of the Financial Services Agency.
- For content other than past or present facts stated in this report, the outlook is based on information available at the time of writing, and actual trends may vary due to various uncertainties.
- The case studies in this report have been organized for research purposes based on publicly available materials concerning each case and information available at the time of writing, and do not indicate the attribution of responsibility or any legal assessment regarding individual cases.

Discussed the research status at regular meetings, reflected expert advice in the research contents, and compiled reports.

The research approach



Given the increasing attacks and vulnerabilities stemming from outsourcing, this research illustrates the supply chain, maps the associated risks, and sets out proposals to strengthen comprehensive security management and controls in line with international standards.

Research background and objectives

- In recent years, cyberattacks targeting centralized exchanges and decentralized finance (DeFi) protocols have occurred with increasing frequency, resulting in numerous incidents of crypto-asset losses. Security issues stemming in particular from third-party outsourcing have come to the fore, and attack vectors are becoming ever more sophisticated. In light of these trends, it is indispensable to strengthen comprehensive, industry-wide information security management and controls that extend beyond the management of signing keys.
- Specifically, (i) taking into account the intent of the “Policy for Strengthening Cybersecurity in Crypto-Asset Exchange Services,” published by the JFSA in April 2026, this study examines areas requiring further enhancement and identifies key priorities that crypto-asset exchange service providers should place greater emphasis on, with a view to raising the level of cybersecurity required under the supervisory guidelines. In addition, from practical perspectives such as contract management and audit frameworks, it organizes appropriate response measures and derives insights for the revision of the supervisory guidelines. Furthermore, (ii) it aims to facilitate cooperation between service providers and authorities and, recognizing that strengthening cybersecurity in the crypto-asset industry is a global challenge, is expected to serve as a foundation for international discussions. Moreover, (iii) in order to enable service providers to establish appropriate security frameworks through the use of standards and guidelines, the study identifies vulnerabilities in supply chains and systems and clarifies concrete measures to be taken.
- Items studied in this research
  1. **A comprehensive industrial structure study**
     Identify the major players in the crypto-asset industry and clarify their roles and business activities. In parallel, visualize relationships with third-party service providers, and examine outsourcing arrangements and technical integrations.
  2. **Case study**
     For representative domestic and international cyberattack cases, examine the attack techniques employed and the vulnerabilities exploited.
     With reference to international and domestic standards and guidelines, including NIST CSF 2.0, the FISC Security Guidelines, and the JFSA’s Cybersecurity Guidelines, discuss the controls that should have been implemented by the affected entities.
  3. **An analysis of risks specific to crypto assets**
     Examine new risk factors unique to crypto-asset-related business operators. In particular, from practical perspectives such as contract management with third-party service providers, audit frameworks, and security verification methodologies, consider corresponding countermeasures and propose enhancements to existing guidelines.
     Aim to strengthen the security posture of crypto-asset-related business operators and contribute to more advanced supervisory oversight and guidance.

Table of Contents

Glossary

Terminology	Definition
AML/CFT	Anti Money Laundering and Combating the Financing of Terrorism
BIS	Bank for International Settlements
BPT	Balancer Pool Token
CCSS	CryptoCurrency Security Standard
CDN	Content Delivery Network
CEX	Centralized Exchange
CGTF	Crypto Governance Task Force
CICD	Continuous Integration/Continuous Delivery (or Deployment)
DAO	Decentralized Autonomous Organization
DeFi	Decentralized Finance
DEX	Decentralized Exchange
DNS	Domain Name System
DoS	Denial of Service
ENISA	European Union Agency for Cybersecurity

Glossary

Terminology	Definition
ERC	Ethereum Request for Comments
ETH	Ether (Ethereum)
FATF	Financial Action Task Force
FSB	Financial Stability Board
HSM	Hardware Security Module
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LP	Liquidity Provider/Liquidity Pool
LST／LRT	Liquid Staking Token／Liquid Restaking Token
MEV	Maximal Extractable Value
MPC	Multi-Party Computation
RAT	Remote Access Trojan
TVL	Total Value Locked

Summary of Research Findings

- In recent years, cyberattacks targeting crypto-asset-related businesses have continued to occur and grow in scale, with attack vectors becoming increasingly diverse, sophisticated, and organized year by year. In particular, as the use of AI technologies and other factors accelerates vulnerability discovery and the identification of attack paths, combined attacks (e.g. compromises of external service providers, smart contract vulnerabilities, malware, phishing, and privilege escalation) have become apparent. Attackers are increasingly targeting not only systems themselves, but also operational infrastructure (e.g. signing keys, wallets, access controls, CI/CD, cloud IAM/KMS, and signing services), **making it important to address supply chain attacks conducted via external systems as well as attacks against a company's own systems**.
- Blockchain has been regarded as a technology that redefines the basis of trust; however, real-world crypto-asset-related services are not operated under a purely decentralized model, but rather as **an architecture that embodies the risks of both centralized and autonomously "decentralized" systems**. **Based on this structural premise, it is necessary to conduct comprehensive risk assessments** that encompass not only on-chain code and protocols, but also off-chain components, operational controls, external dependencies, and governance.
- In this research, we analyzed recent attack cases against crypto-asset-related services, including DeFi, focusing on major attack methods that resulted in large losses, such as **third-party compromises, smart contract vulnerabilities, rug pulls, flash loan attacks, and malware attacks**. We also analyzed recent incidents targeting crypto-asset-related services, including DeFi. As a result, **we confirmed attacks that did not involve the theft of signing keys themselves, but instead tampered with components** (e.g., UI, APIs, CI/CD, unsigned transaction generation logic, and production programs), causing fund outflows by abusing legitimate signing processes. The analysis also indicated potential issues in controls at third parties (e.g., social engineering and malware countermeasures, production environment access management, program change management, verification at the time of signing, smart contract upgrade privileges, and business logic verification).

Summary of Research Findings

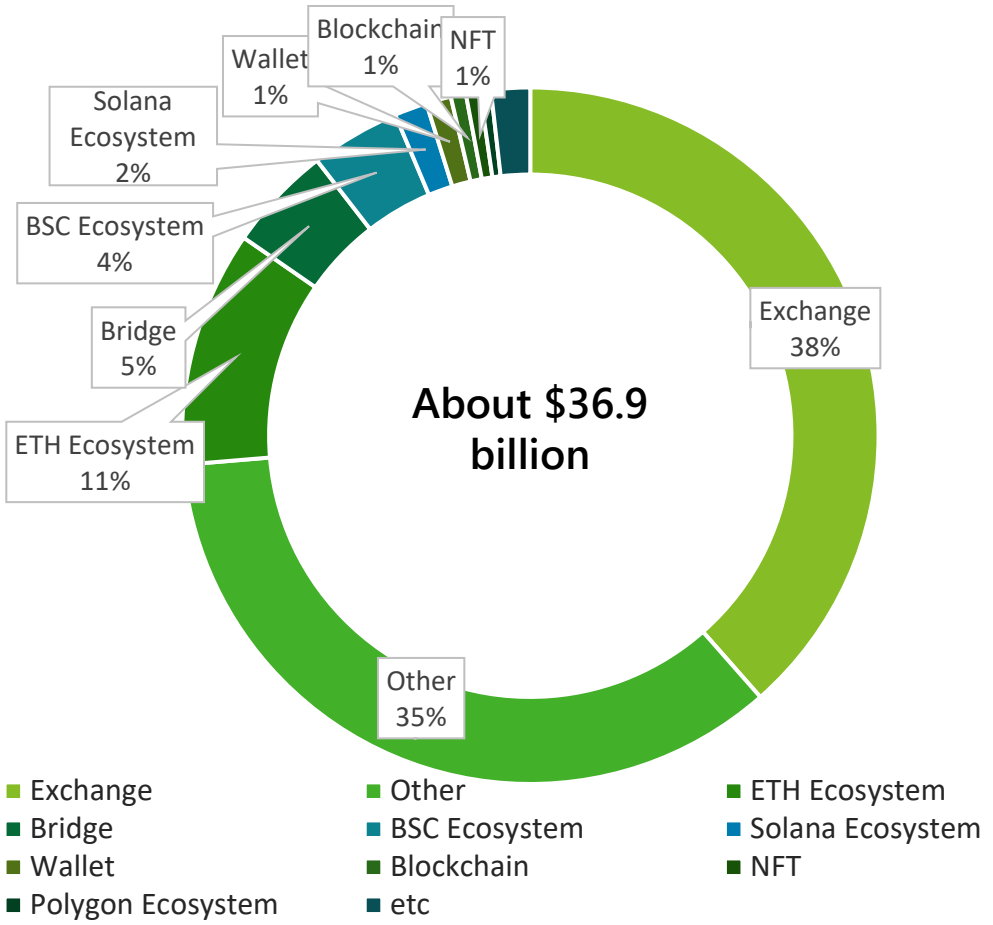
- Based on the analysis of the crypto-asset ecosystem and case studies, this research performed a comparative analysis of the identified attack methods and vulnerabilities against existing guidelines (e.g., NIST CSF 2.0, FSA Administrative Guidelines, FSA Cybersecurity Guidelines, the FISC Security Guidelines, JVCEA rules and standards, and OWASP SCSVS). As a result, while existing guidelines include items that address major vulnerabilities, it was confirmed that **crypto-asset-related business operators need to further specify and deepen their controls according to their own system configurations, outsourcing relationships, wallet and key management methods, and the status of their use of DeFi and external services.** Based on these findings, five areas were extracted and examined as priority areas: **enhancement of third-party risk management; measures to prevent malicious code injection or program tampering; measures to prevent unauthorized transfers of crypto-assets; measures for smart contracts and DeFi; and leveraging external assessments**.
- In particular, when a crypto-asset-related business operator uses external service providers for critical operations (e.g., crypto-asset withdrawals, signing, key management, wallet operations, CI/CD, and cloud infrastructure), it is necessary to thoroughly evaluate the system of the external service providers (e.g., system architecture, governance structure, access privileges, subcontracting arrangements, log monitoring, incident response, and program change management). Even when reviewing external assessments or third-party assurance reports, the operator is also required to assess relevant matters (e.g., scope, assessment timing, exclusions, evaluator qualifications, and remediation status), and to address any identified gaps. In addition, given that a breach at an external service provider could spill over to the operator's own services or user assets, it is important to establish self-protective measures (e.g., additional approvals, usage limits, anomaly detection, automated suspension, and alternative arrangements).
- Given that service quality varies across crypto-asset-related business operators, **it is important to establish common international data standards for threat and vulnerability information related to crypto-assets**. Furthermore, in light of the acceleration of vulnerability discovery and the identification of attack paths in the AI era, **crypto-asset-related business operators are required to establish a cybersecurity framework that integrates technical measures and governance** (e.g. reducing attack surfaces, isolating critical systems, controlling development environments, educating executives and employees, and redesigning responsible disclosure processes).

1. Crypto-asset-related cyberattack trends and risk analysis

1.1 Cyberattack trends

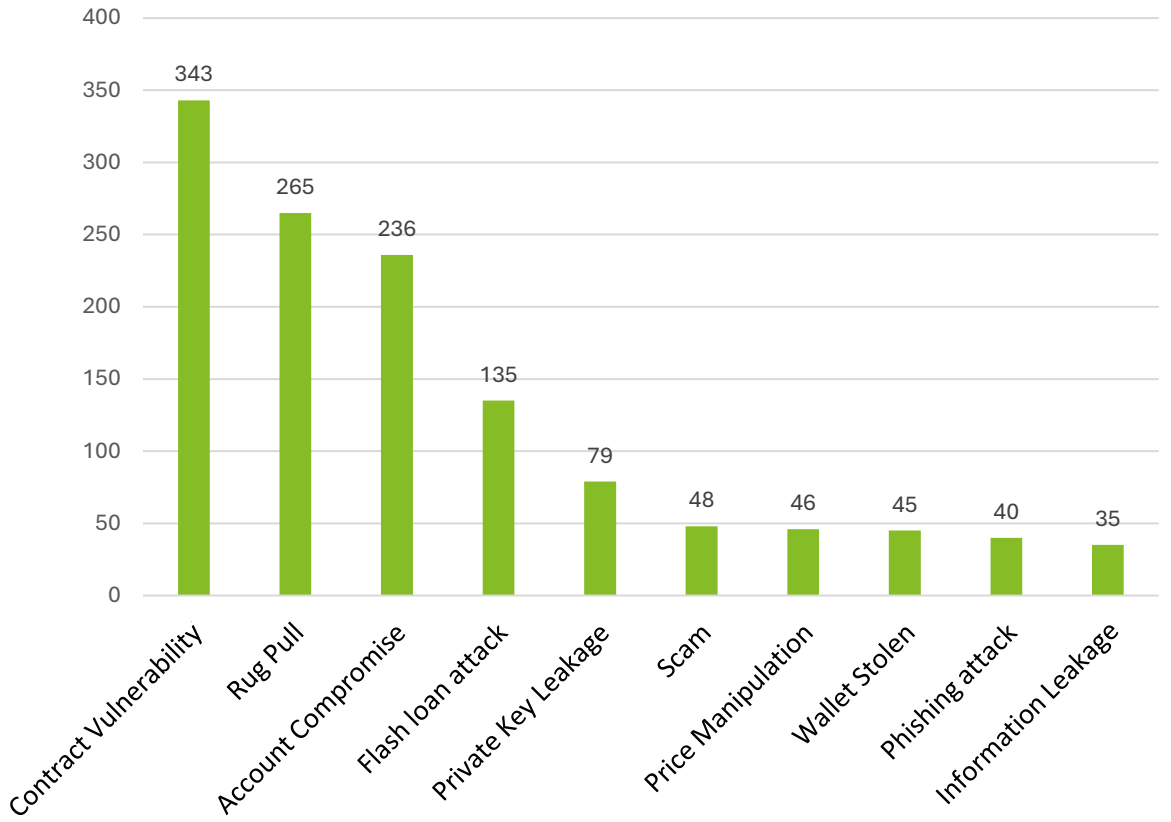
Crimes against crypto-asset-related businesses are diversifying.

Amount of loss by category (cumulative since 2012)



*"etc" is an aggregation of categories of 1% or less individually, and "Other" refers to items that cannot be classified to the above categories.
Source : SlowMist Hacked - SlowMist Zone (SlowMist Hacked) As of March 2026

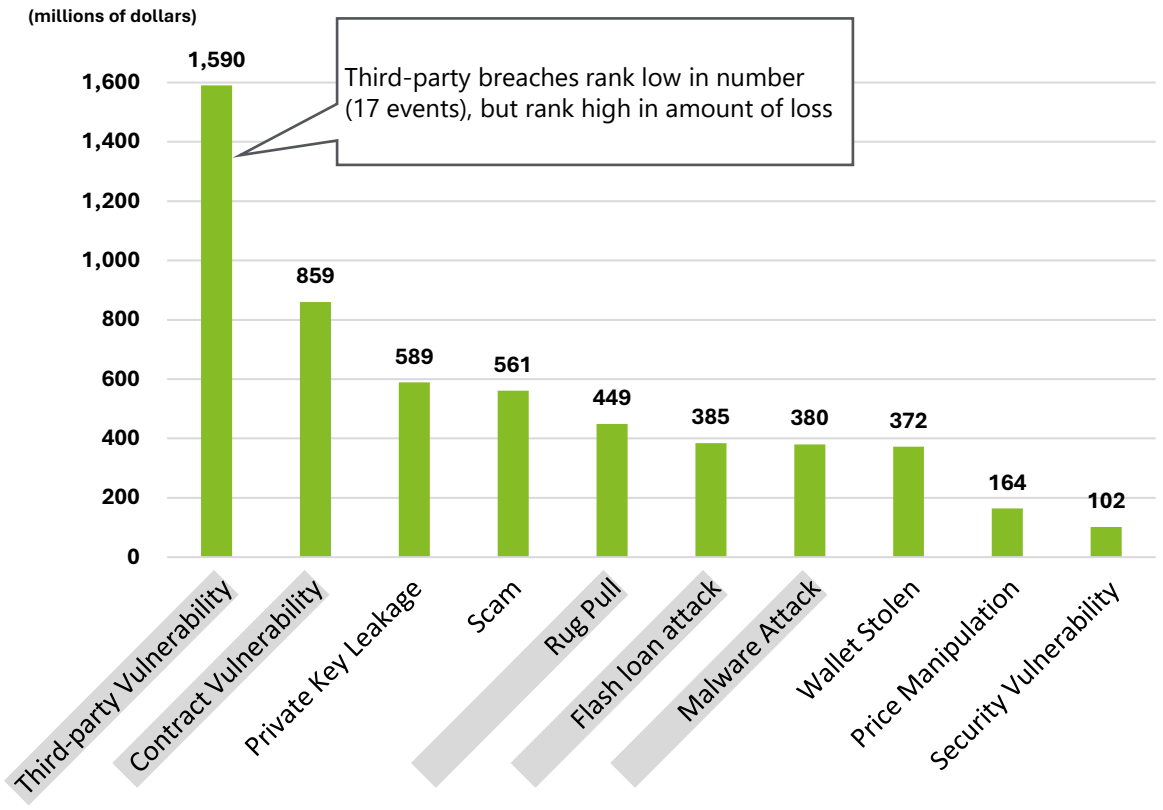
Number of hacks by attack method (cumulative since 2012)



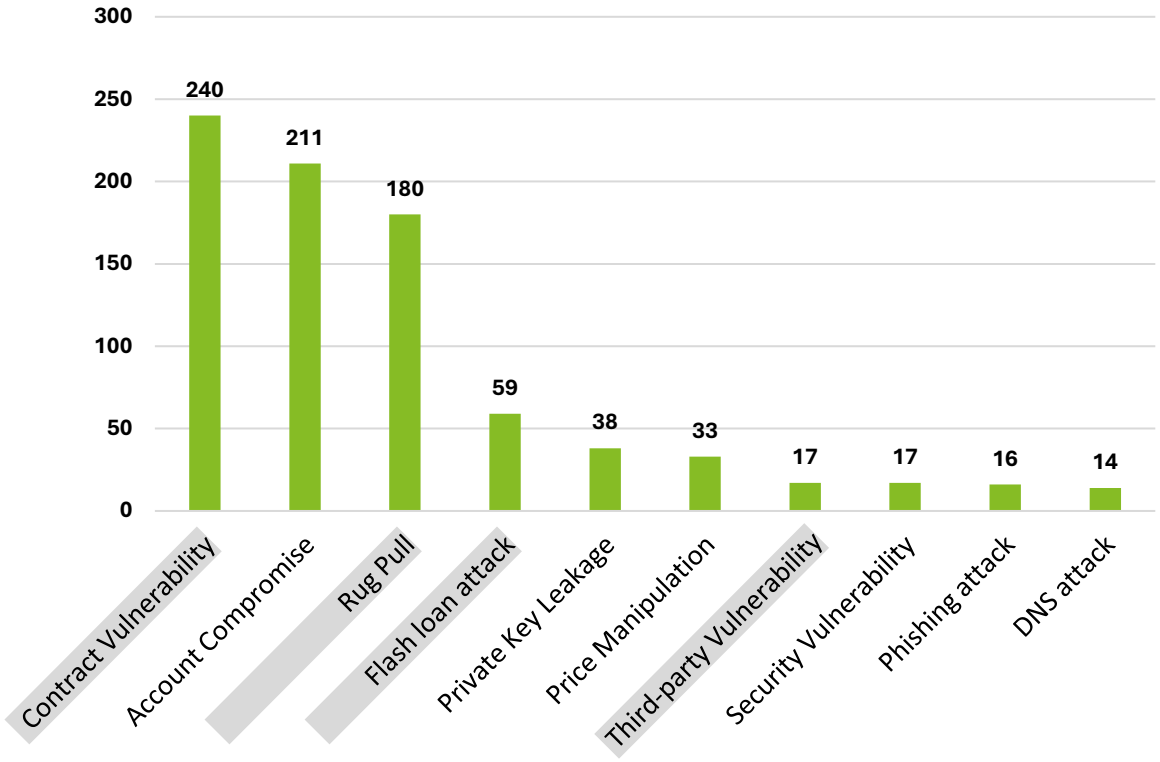
* Although some incidents categorized as “rug pulls” may involve attacks exploiting smart contract vulnerabilities or similar technical issues, many such incidents primarily relate to market manipulation or fraud rather than cybersecurity issues.

Since 2023, there has been a significant number of attacks targeting third-party vulnerabilities or smart contract vulnerabilities.

Amount of loss by attack method (cumulative since 2023)



Number of incidents by attack method (cumulative since 2023)

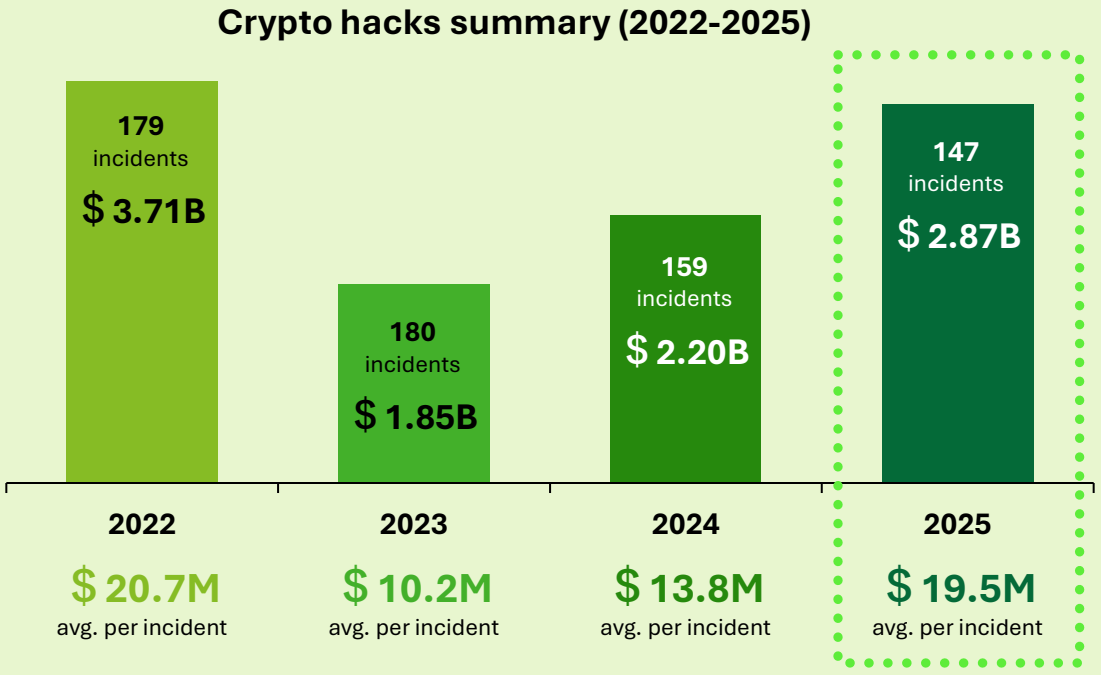


Source : SlowMist Hacked - SlowMist Zone (SlowMist Hacked) _ Statistics of major blockchain hacking incidents occurred between January 6, 2012 and January 21, 2026

In 2025, total losses from crypto-asset-related incidents reached a record high, with damages increasingly concentrated in a small number of large-scale incidents.

Incident Trends (1/5)

- According to TRM Labs, illicit crypto volume reached an **all-time high of USD 158 billion** in 2025.
- Of this total, USD 2.87 billion was stolen across approximately 150 incidents involving hacking and the exploitation of vulnerabilities, with the top 10 incidents accounting for 81% of the annual total, highlighting a marked concentration of losses in large-scale incidents.



In 2025, **while the average incident size rose to USD 19.5 million,** the median incident size fell to USD 1.3 million.

The Bybit breach in February accounted for USD 1.46 billion (51%) of all funds stolen in 2025. **The top 10 incidents represented 81% of the annual total**, signaling an excessive concentration of risk on specific players.

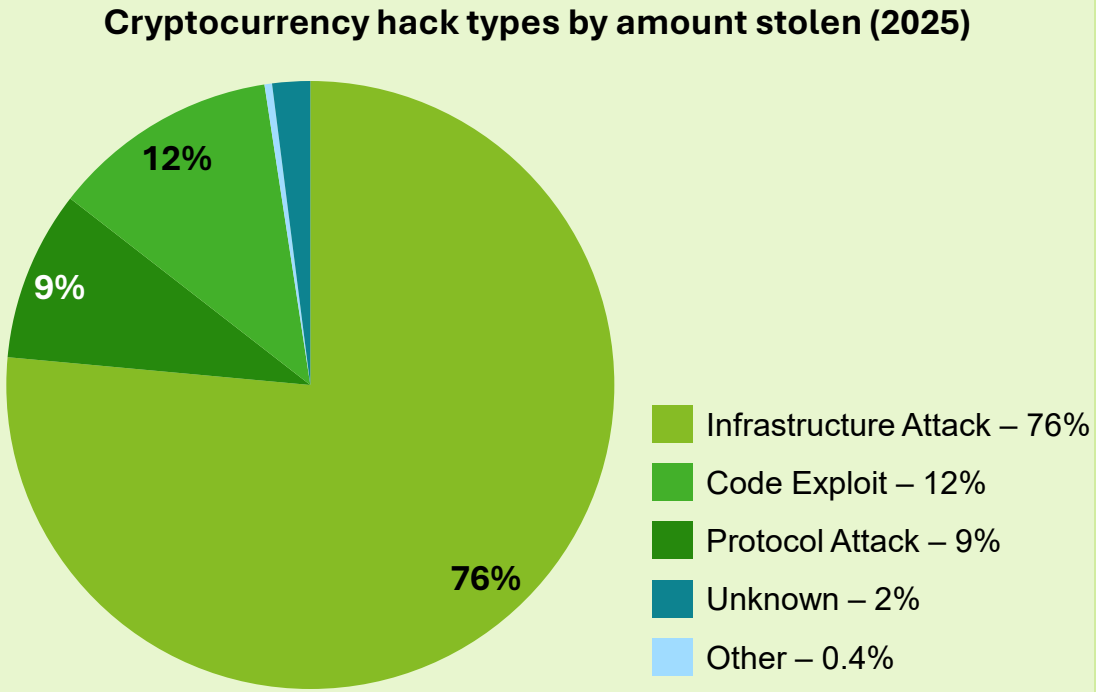
Sophisticated actors, particularly state-sponsored threat actors, are no longer just exploiting code — **they are compromising the operational foundations of crypto-asset services and the ecosystems around them**, while a high volume of smaller incidents continues to inflict persistent damage across the crypto ecosystem.

Source : 2026 Crypto Crime Report – Illicit Crypto Trends & Typologies (TRM Labs) As of February 2026

[Factor 1 Behind the Concentration of Losses in Large-Scale Incidents]
In 2025, there was a notable shift toward directly targeting infrastructure with higher attack return on investment

Incident Trends (2/5)

- In 2025, while incident volume remained broadly consistent with recent years, the severity of losses increased sharply due to **a structural shift in attack vectors**:
- The adversaries moved up the stack, **targeting operational infrastructure, including keys, wallets, and access controls.**
- For highly sophisticated adversaries, compromising the operational infrastructure of centralized entities such as CEXs has become the most cost-effective attack approach.



TRM's dataset classifies hacks and exploits into five top-level categories:

- Infrastructure Attack (that directly targets the operational infrastructure such as keys and servers)
- Code Exploit (that exploits a vulnerability or bug in a program)
- Protocol Attack (that exploits a flaw in the specification or design of a system)
- Unknown
- Other

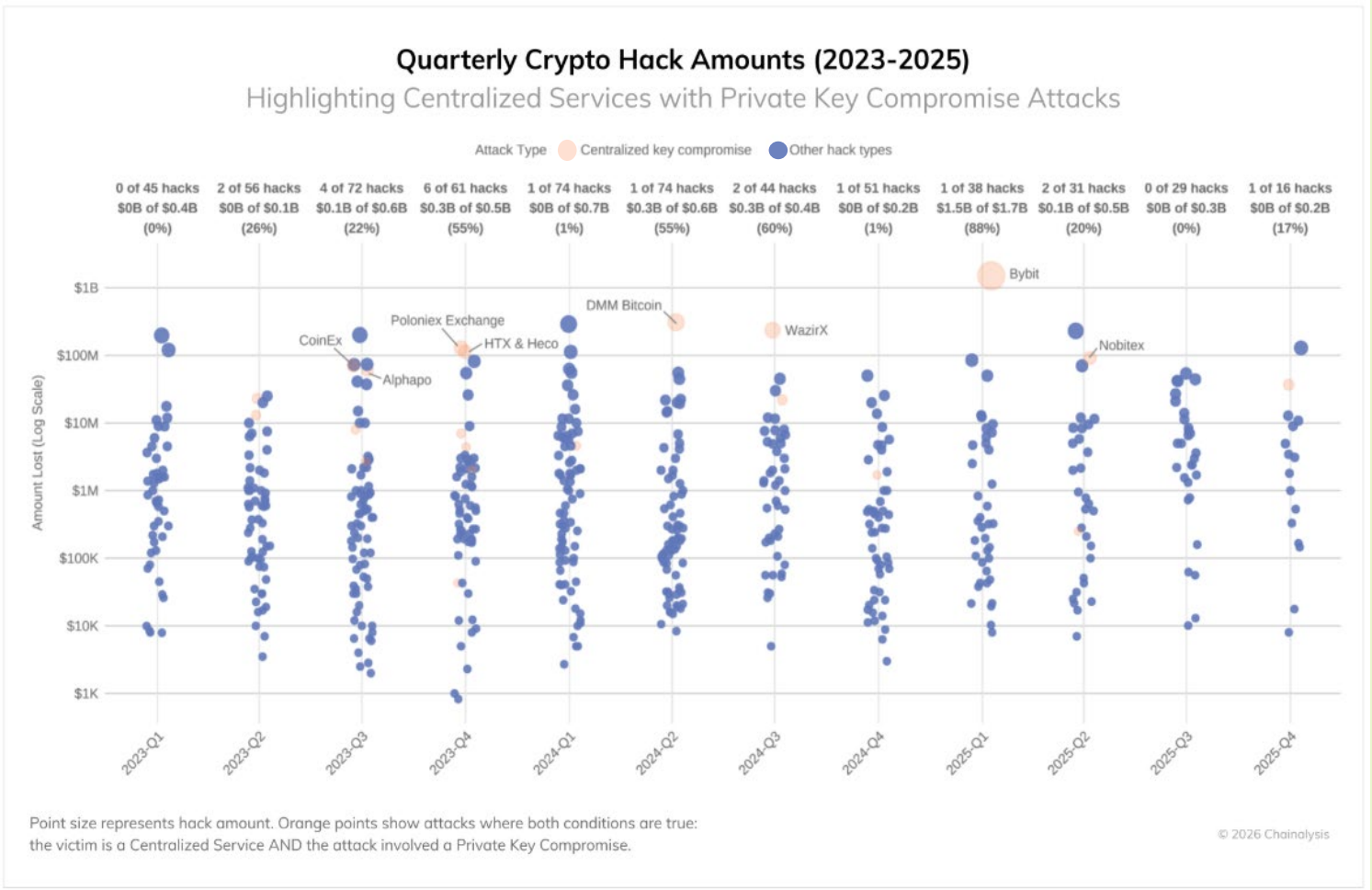
1. **Infrastructure Attacks —** which include compromises of private keys / seed phrases, wallet infrastructure, privileged access, and front-end surfaces — **drove USD 2.2 billion in losses (76%) across 45 incidents,** averaging approximately USD 48.5 million per incident. **The dominant 2025 pattern was operational compromise, often enabled by social engineering, developer environment penetration, or weaknesses in access controls and withdrawal governance.**
2. Code Exploits were the most frequent category (52 incidents), but they accounted for a comparatively smaller share of total losses (USD 350 million, 12.1%) — about USD 6.7 million per incident on average.
3. Protocol Attacks were less frequent (25 incidents) but more severe on average (USD 277 million, 9.6%) — roughly USD 11.1 million per incident.

Source : 2026 Crypto Crime Report – Illicit Crypto Trends & Typologies (TRM Labs) As of February 2026

[Factor 2 Behind the Concentration of Losses in Large-Scale Incidents]
While attacks on infrastructure do not occur frequently, they tend to result in significantly larger losses when they do occur.

Incident Trends (3/5)

- **Centralized services are experiencing increasingly large losses due to sophisticated attacks on private key infrastructure and signing processes.** This type of compromise drives enormous shares of stolen volumes, **88% in Q1 2025 due to Bybit.**
- While such compromises are infrequent, their scale is extremely large when they do occur.
- Despite their institutional resources and professional security teams, these platforms remain vulnerable to advanced threats that can circumvent cold wallet controls. **Many attackers have developed methods to exploit third-party wallet integrations and trick legitimate signers into authorizing malicious transactions.**
- State-backed threat actors are increasingly achieving these outsized results often by embedding IT workers – one of their principal attack vectors – inside crypto services to gain privileged access and enable high-impact compromises.



Source : 2025 Crypto Theft Reaches $3.4 Billion (Chainalysis) As of December 2025

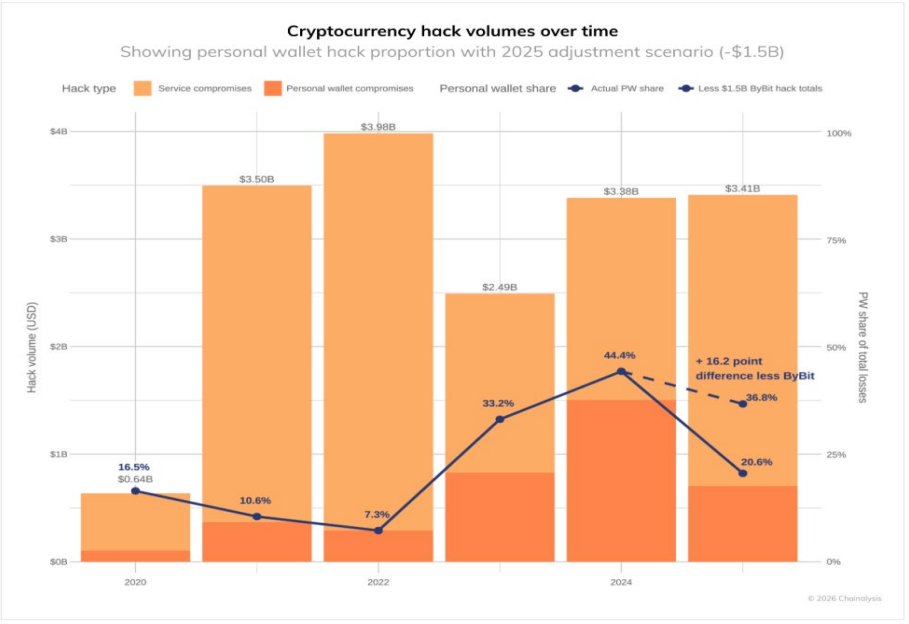
Legend (notes added by Deloitte)

Attacks on the signing-key infrastructure, etc., of centralized services
Other attacks
*The size of each circle indicates the amount of loss

[Factor 3 Behind the Concentration of Losses in Large-Scale Incidents]
In recent years, the proportion of attacks targeting individual wallets has increased sharply, widening the disparity in the scale of losses.

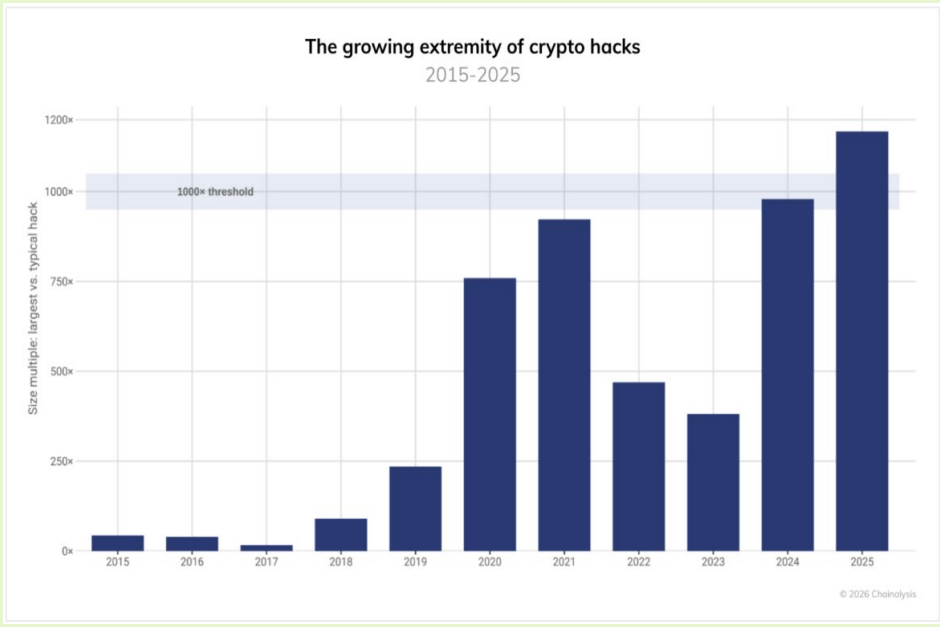
Incident Trends (4/5)

- The cryptocurrency industry witnessed over $3.4 billion in theft from January through early December 2025, with the February compromise of Bybit alone accounting for $1.5 billion of that total.
- Recent incidents have shown a shift from ‘Service compromises’ to ‘Personal wallet compromises’ and a growing extremity in amount of loss.



Legend (notes added by Deloitte)

- Attacks on crypto-asset services
- Attacks on personal wallets



When classifying incidents as ‘Service compromises’ and ‘Personal wallet compromises’, the data reveal important shifts in the composition of these thefts. ‘Personal wallet compromises’ have grown substantially, increasing from just 7.3% of total stolen value in 2022 to 44% in 2024. In 2025, the share would have been 37% if it weren’t for the outsized impact of the Bybit attack.

Stolen fund activity has always been outlier-driven, but 2025 reveals a striking escalation: the ratio between the largest hack and median of all incidents has crossed the 1,000x threshold for the first time. The top three hacks in 2025 account for 69% of all service losses. **This growing discrepancy has concentrated losses dramatically.**

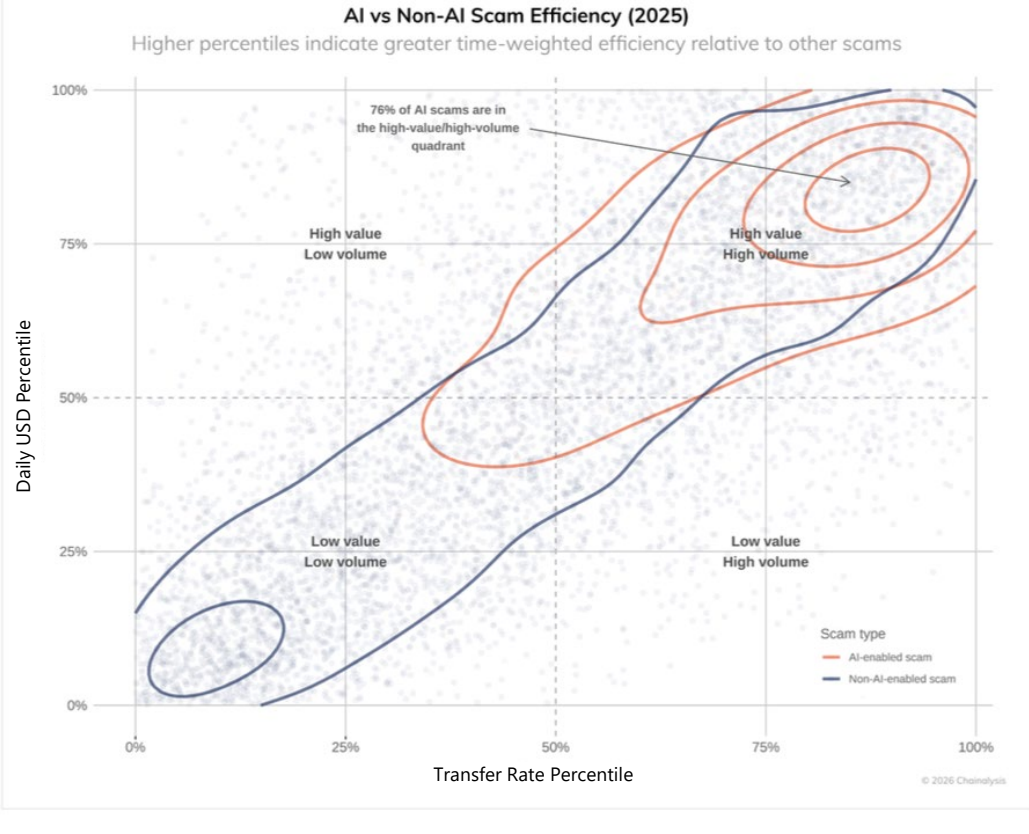
Source : 2025 Crypto Theft Reaches $3.4 Billion (Chainalysis) As of December 2025

* The above report does not include information after January 2026. Nevertheless, recent incidents indicate that attacks targeting crypto-asset services, particularly DeFi-related services, have continued to increase.

Scammers are increasingly leveraging deepfake and face-swapping technology and AI-generated content.

Incident Trends (5/5)

- Chainalysis's analysis revealed that, on average, scams with on-chain links to AI vendors extract $3.2 million per operation compared to $719,000 for those without an on-chain link — 4.5 times more revenue per scam.
- These AI-related operations also demonstrate significantly greater time-weighted efficiency: higher daily revenue of $4,838 vs median daily revenue of $518.
- State-sponsored threat actors have also been observed targeting the crypto-asset and DeFi sectors and using AI-generated videos to deceive victims.



We are moving toward a future in which virtually all scams will incorporate AI into their operations to some degree. In the figure, 'AI Enabled Scam' and 'Non-AI Enabled Scam' are plotted and visualized on 'Daily USD Percentile' (vertical axis) and 'Transfer Rate Percentile' (horizontal axis). Compared to non-AI operations (blue), **AI enabled cases (orange) tend to have higher incoming transfer rates and higher daily USD volumes**.

AI-related operations demonstrated increased transaction volume, averaging 35.1 transfers per day, **about 9 times** higher than the 3.89 average transfers per day observed in non-AI scams.

In AI-enabled scams, scammers are increasingly leveraging deepfake technology and AI-generated content to create convincing impersonations in romance and investment scams. Not only does the adoption of AI make scams more convincing and persuasive, AI is also enabling scammers to **reach and manage more victims simultaneously**.

The vertical axis in the left figure shows the loss amount (Daily USD Percentile), and the horizontal axis shows the number of incidents (Transfer Rate Percentile).

Legend (notes added by Deloitte)
- AI Enabled Scam
- Non-AI Enabled Scam

Source : The 2026 Crypto Crime Report - English - Chainalysis (Chainalysis) As of March 2026
「UNC1069 Targets Cryptocurrency Sector with New Tooling and AI-Enabled Social Engineering」（Google Cloud） As of February 2026

1.Crypto-asset-related cyberattack trends and risk analysis
1.2 Mapping of key players and analysis of cybersecurity characteristics

Blockchain technology was expected to redefine the basis of trust, but the real-world operations have "composite architectures" with a mix of centralized elements and decentralized infrastructure, thus analysis based on this assumption is necessary.

Comparison of centralized and autonomously distributed systems regarding cybersecurity

	Ideological difference		Real-world operations
	Centralized systems (Third party dependent)	Autonomously distributed systems (Protocol dependent)	Operations in crypto-asset-related business (Both)
Basis of trust	**Third parties** Security is ensured and managed by specific entities	**Cryptographic proof** Security is ensured by mathematical correctness and decentralized consensus	**Business operators** A model in which management of distributed assets is entrusted to specific entities
Ideology of defense	**Multi-layer defense/ Zero trust** Stricter authentication and authorization, monitoring under the assumption of being attacked	**Self-sovereign defense management** It's all about managing signing keys, systems cannot deny access	**Isolation of critical assets** While defending with web technology, assets (keys) are isolated with cold wallets, etc.
Access control	**ID provider dependent** Can be reset and stopped by the administrator	**Signing key dependent** Loss of key means loss of the asset, no one authorizes nor stops others	**ID management + Delegated key management** Users use ID/PW, but operators manage signing keys at the back
Tamper resistance (Immutability)	**Managed by the administrator** Authorized administrator can edit the records	**Tamper resistance** Impossible to cancel transactions even if they are deemed illegal by the administrator	**Managed through double books with consistency** Business operator guarantees synchronization of its database and the data on blockchain
Highest risk	**Information leakage/ Abuse of access** Data is copied and the originals are often kept in safe	**Irreversible loss of assets** Loss of an asset means that the asset itself is displaced and cannot be recovered	**Concentrated attacks on a single point of failure** Centralized defenses get compromised and decentralized assets get stolen

what is actually happening

- **Incidents occurring**
  - Blockchain can be understood as a technology that “decentralizes where trust is placed and expands the scope of what can be made verifiable.” However, designing a fully autonomously decentralized system is difficult. The organizations and operations needed to complement this limitation tend to be centralized, and large-scale hacking incidents caused by vulnerabilities in these centralized components continue to occur.
- **Structural issues**
  - **An architecture that embodies the risks of both centralized and autonomously decentralized systems:**

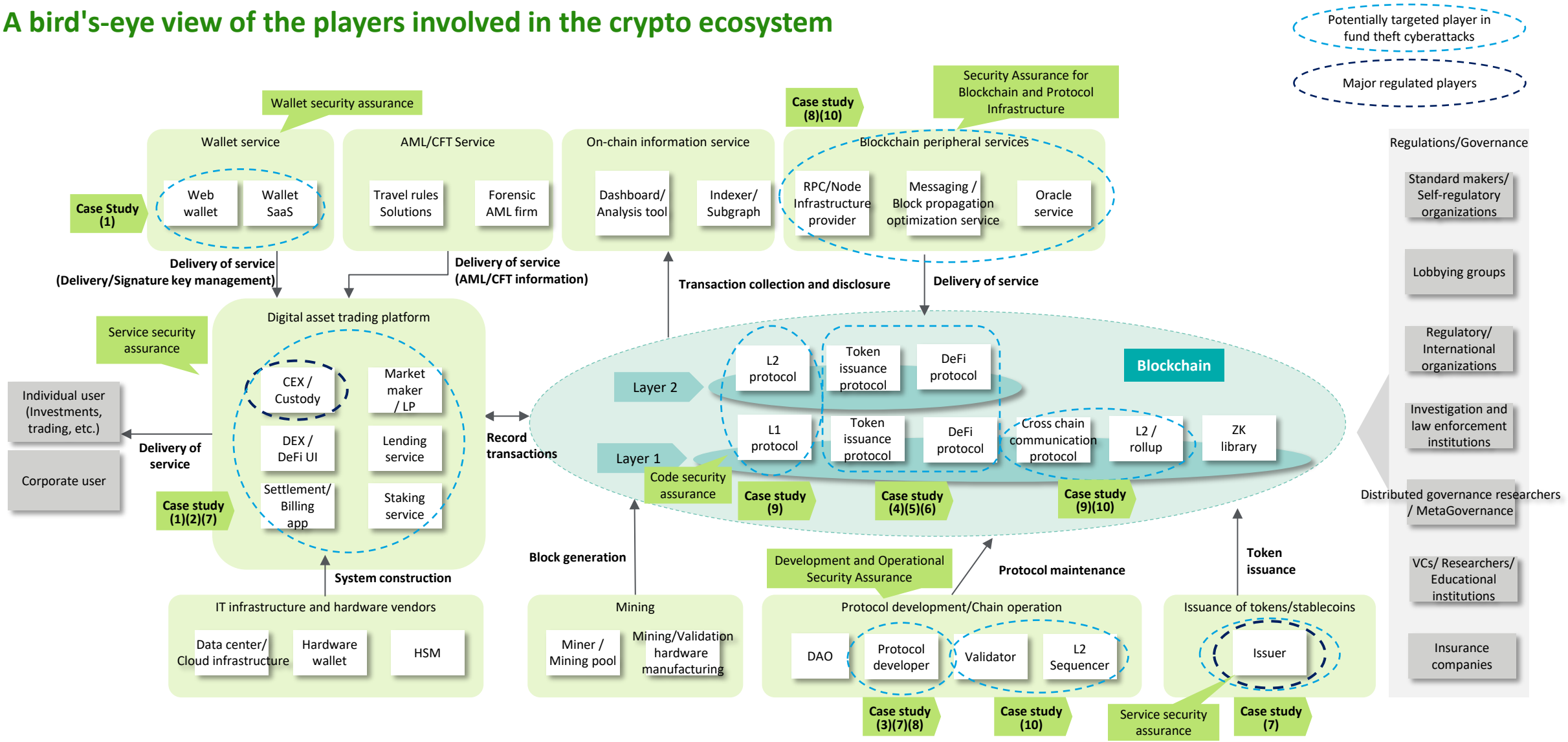
    Internet-based value transfer
    Highly liquid assets are constantly connected to public networks globally, and the expected return of successful attacks is extremely high.
    Mutual amplification of risk
    Centralized components such as UI/API, key management, and bridges are becoming single points of failure, while transaction suspension and rewinding are difficult due to on-chain determination after an incident occurs. It is a structure that suffers from both the fragility of centralization and the irreversibility of decentralization.
  - **Structural increase in defense cost:**

    Highest grade immediate risk of loss
    Due to the global instant settlement, anonymity, and difficulty of rewinding, the scale of losses in the event of a successful attack is extremely large.
    Return to operation-dependent security
    It is necessary to supplement the defense with advanced key management (MPC/HSM), multi-layer authorization, continuous monitoring, external audit, etc., thus the defense cost is structurally increasing.

■Standpoint of this study report
Real-world operations are not purely autonomously distributed, but "composite architectures" with a mix of centralized elements and decentralized infrastructure. Based on this assumption, this study conducted a comprehensive risk analysis including off-chain elements, operational controls, and governance.

The crypto ecosystem is composed of a diverse set of players, many of whom are not directly regulated or supervised. Therefore, it is important to take a holistic view of the ecosystem and identify attack targets and the potential impact of attacks.

A bird's-eye view of the players involved in the crypto ecosystem



Following the approaches taken in academic papers and public reports※, we decided to take an approach of revealing the industrial structure and identifying risks in security through layering, visualizing and analyzing interdependencies of crypto ecosystem, for this research.

Research approach based on preliminary research

Preliminary research - Academic papers -

In academic papers on cybersecurity of crypto-asset-related businesses, technological features of autonomous distributed systems were observed as follows.

- Data's tamper resistance and transparency
- Decentralized implementation and access management
- Introduction of consensus mechanisms
- Smart contracts, etc.

An approach of reorganizing layered system model incorporated with new technical features, categorizing threats and vulnerabilities in security, and analyzing risk factors, has been widely adopted.

Preliminary research - Public reports, etc. -

Public reports, etc. extracted the technical features in the same way as above, and in addition to consideration of the technical elements, they also tend to discuss and analyze risks from the following perspectives.

- Operation and governance development and challenges
- Importance of key management, authority management, and wallet operation
- Financial service delivery players and market dynamics
- Existing regulations and standards

※ Please refer to Appendix1 for preliminary research details.

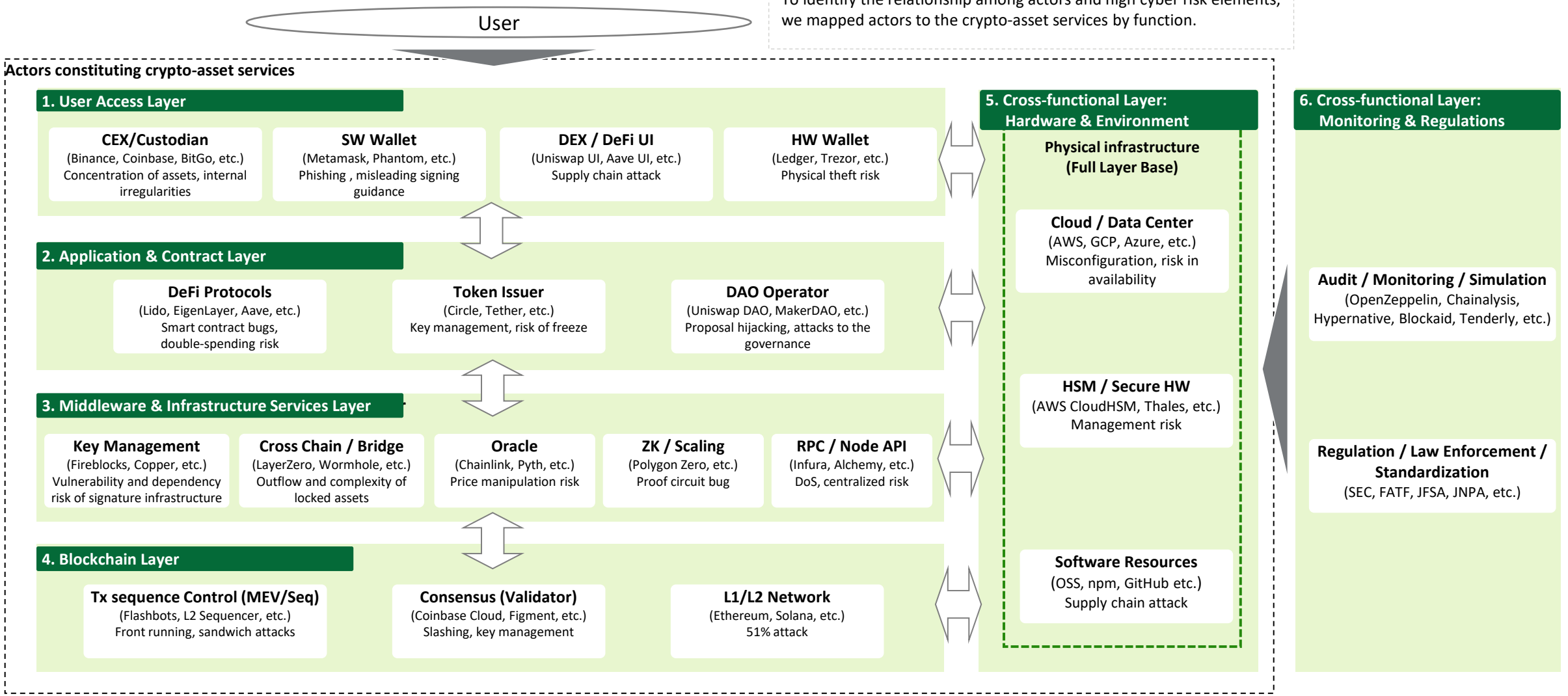
1.Crypto-asset-related cyberattack trends and risk analysis
1.3 Supply chain by function and attack vectors

Crypto-asset-related services are provided through a combination of various functions, thus we identify high cyber risk elements by revealing the interdependencies among them.

Actors by function in crypto-asset-related businesses

Legend: **Function** (Representative actors) Examples of attack methods

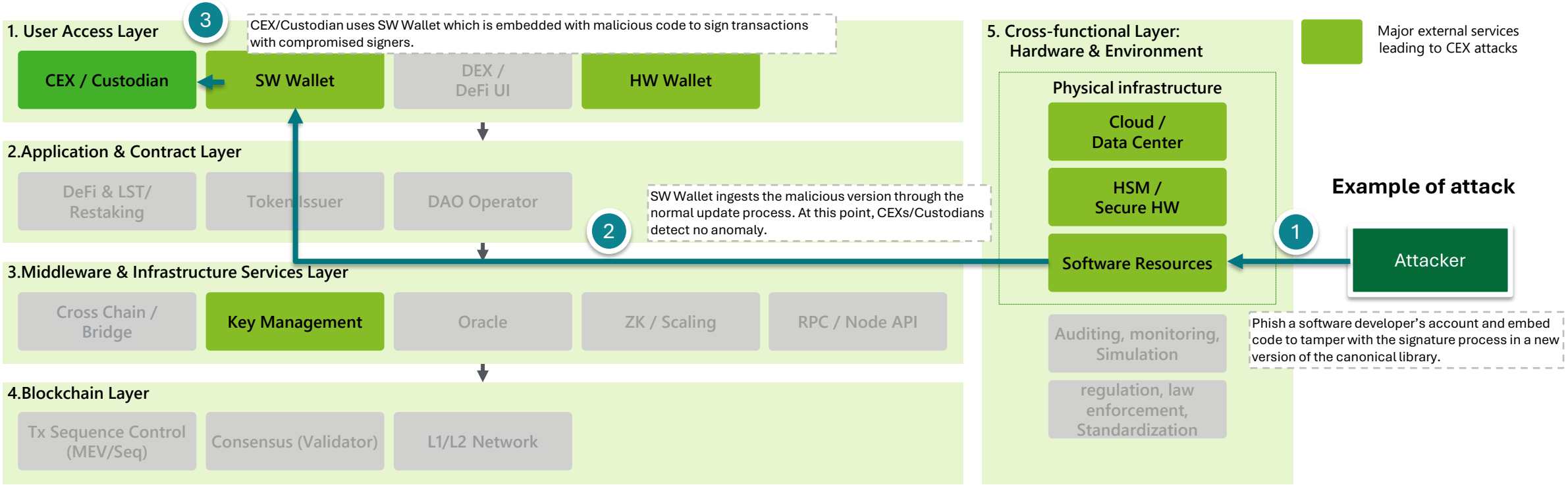
To identify the relationship among actors and high cyber risk elements, we mapped actors to the crypto-asset services by function.



Because of the dependency on external services in CEX operations, in important areas such as signing, it is recommended to appropriately manage the supply chain, including subcontractors.

Overview of actors by function

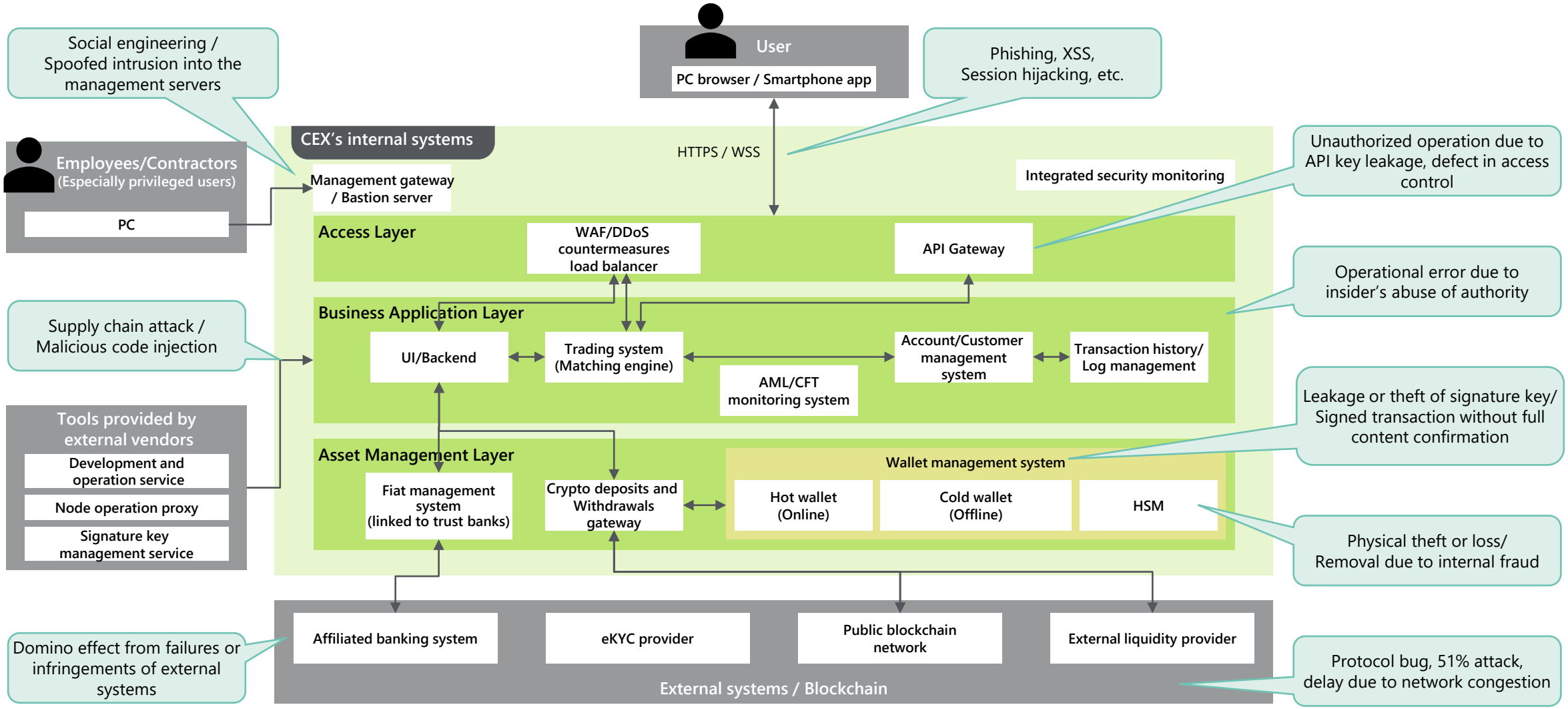
- **Limits of control and supply chain attacks**
  CEX/Custodian only have direct control over its own infrastructure and employees' authorities and procedures.
  On the other hand, the security level and operation of external services such as wallet SaaS, development infrastructure, and monitoring infrastructure, depend on the discretion of each service provider, so CEX/Custodian has only indirect controls based on the contracts or certain monitoring activities. External services, especially those involved in key management and signing processes, can be "critical points" that could lead to asset leakage in the event of a breach.
- **Black box resulted from multi-level outsourcing**
  CEX/Custodian's external vendors also re-contract to sub-vendors, and a lack of transparency can create a chain of attacks. The importance of properly confirming and supervising sub-outsourcing is increasing, but multi-level outsourcing can be difficult to trace in practice.



Note: The above figure is an example in a layered structure, and the external services used by the CEX / Custodian may differ by company.

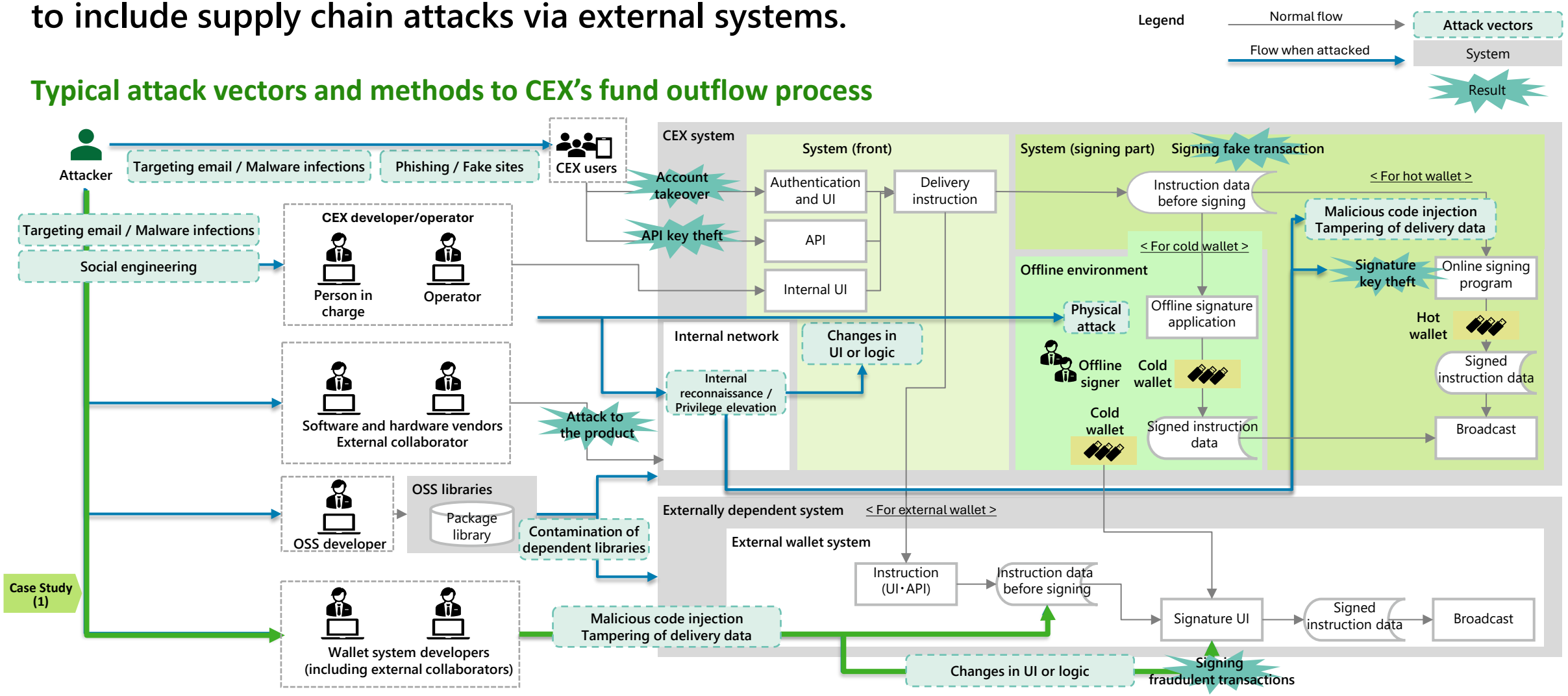
It is necessary to identify potential channels through which attackers can access the CEX's systems, and take appropriate countermeasures accordingly.

Attacks mapped to functions of a CEX



The attack surface and vectors targeting CEXs are diverse, extending beyond internal systems to include supply chain attacks via external systems.

Typical attack vectors and methods to CEX's fund outflow process



Attack via CEX developers/operators: A method in which attackers access the internal network through social engineering or malware targeting CEX developers/operators, and attack the core of the CEX system, such as withdrawal instructions and signing, by injecting malicious code, modifying the UI or logic, or conducting physical attacks.

Attack via software and hardware vendors / external collaborators: A method in which attackers access the internal network through social engineering or malware targeting external development collaborators, and attack the core of the CEX system by injecting malicious code, modifying the UI or logic, or conducting physical attacks.

Attack via OSS (open-source software) developers: A method in which attackers compromise OSS developers through targeted emails or similar techniques, contaminate OSS libraries, and cause the impact to propagate to the CEX system or the external wallet system as a whole. (*OSS: software whose source code is publicly available / Library: a program used as a component by other systems)

Attack via wallet system developers: A method in which attackers infect the PCs of wallet system developers with malware through social engineering or similar techniques, gain access to external wallet systems, and attack the core of the external wallet system by injecting malicious code or modifying the UI or logic.

2. Study of causes and countermeasures of recent incidents

2.1 Sampling for case study

Attacks against crypto-asset-related businesses range from classic phishing and malware attacks, to exploits of vulnerabilities in smart contracts or blockchain-related software systems.

Examples of attack vectors against crypto-asset-related businesses

Social engineering / Phishing

Attacks that exploit human psychology and carelessness, rather than system vulnerabilities, to obtain authentication credentials or confidential information, including phishing attacks that use fraudulent emails or SMS messages to mislead users into entering IDs, passwords, and other sensitive information.

Malware attack

Malicious software and malware attacks are used by bad actors to target and steal crypto assets. Types of malware can include:
- Keyloggers, that capture keystrokes and allow attackers to record sensitive information,
- Phishing software, used to perform phishing campaigns as discussed above,
- Remote Access Trojan (RAT), which allows attackers to gain control over a victim's hardware, enabling access to wallets and secret information, and
- Cryptojacking, which involves hijacking a user's computing resources to mine cryptocurrency.

API Compromise

If the APIs are not safely secured attackers can bypass authentication and access sensitive and private data, such as the developer's PC and internal systems, to ingest API credentials that control critical functions such as fund withdrawal, transaction signing, and account management. Using these keys, attackers can hijack transaction management systems, identify vulnerabilities in authorization process, and perform large-scale fraudulent transfers.

Weak authentication system

Often, bad actors may choose to attack via 'brute force' - when bad actors easily guess simple or common passwords chosen by users. Additionally, if users reuse their passwords across several platforms, several accounts may be compromised as a result of one weak protection method.

Smart Contract Vulnerability

Oversights by developers who write the Smart Contracts may sometimes leave room for vulnerabilities and flaws, which can be taken advantage of by hackers. Commonly these can include:
- Reentrancy Attacks : where the hacker exploits a function that interacts with an external contract, prior to the update of the original contract. For example, an attacker could continuously call a function that withdraws funds, before the original smart contract has a chance to update the balance, and the attacker could withdraw more funds than were available.
- Access Control Failures : when a smart contract does not have robust security for permission of access, an attacker could invoke restricted functions, that allow them to transfer funds or access assets.
- Logic bugs : simple but frequent coding errors or oversights, such as incorrect conditions, or poorly defined terms in the smart contract logic, can allow attackers to perform actions such as draining the contract's funds.

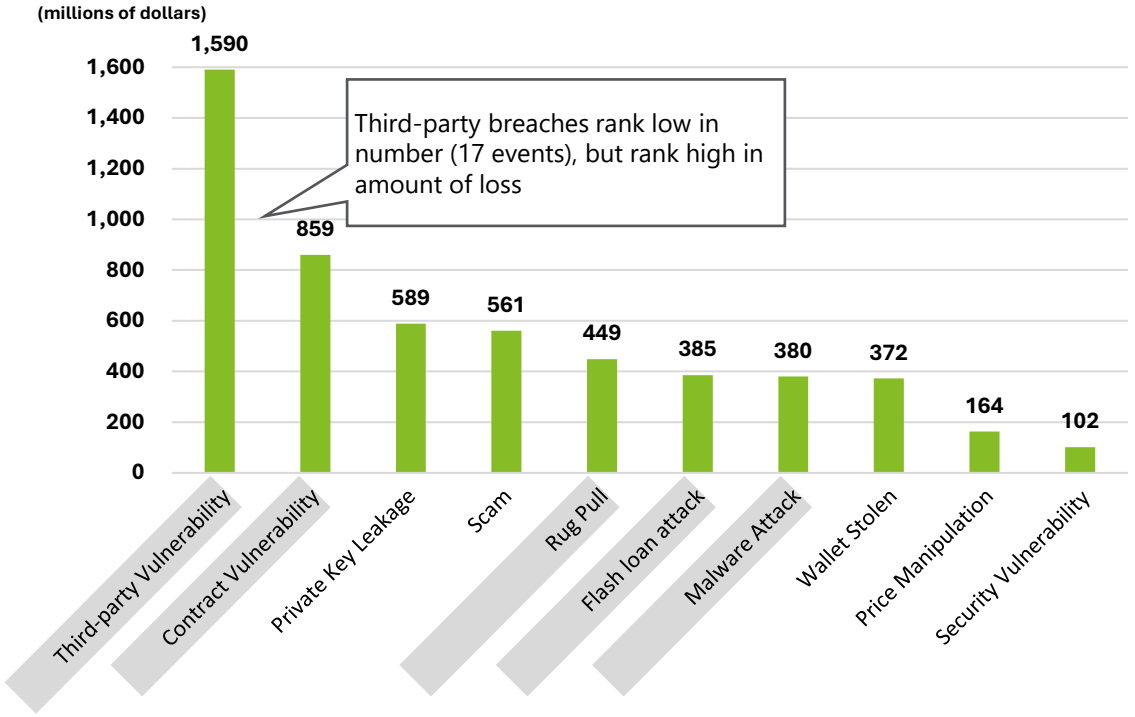
Flash loan attack

Flash loans allow users to borrow large sums of cryptocurrency without collateral as long as they return the loan within the same transaction, so hackers use these loans to manipulate prices on decentralized exchanges, or exploit vulnerable logic in the protocols to steal funds. Commonly these can include:
- Price oracle manipulation: manipulating prices on a particular exchange to borrow assets at a lower price or to liquidate collateral fraudulently from another lending protocol that uses the fake prices as a reference,
- Liquidity pool exploit: profit by operating liquidity from the liquidity pool to manipulate token prices, and
- Use of vulnerability in governance: stealing funds by temporarily collecting a large number of voters to get malicious proposals passed.

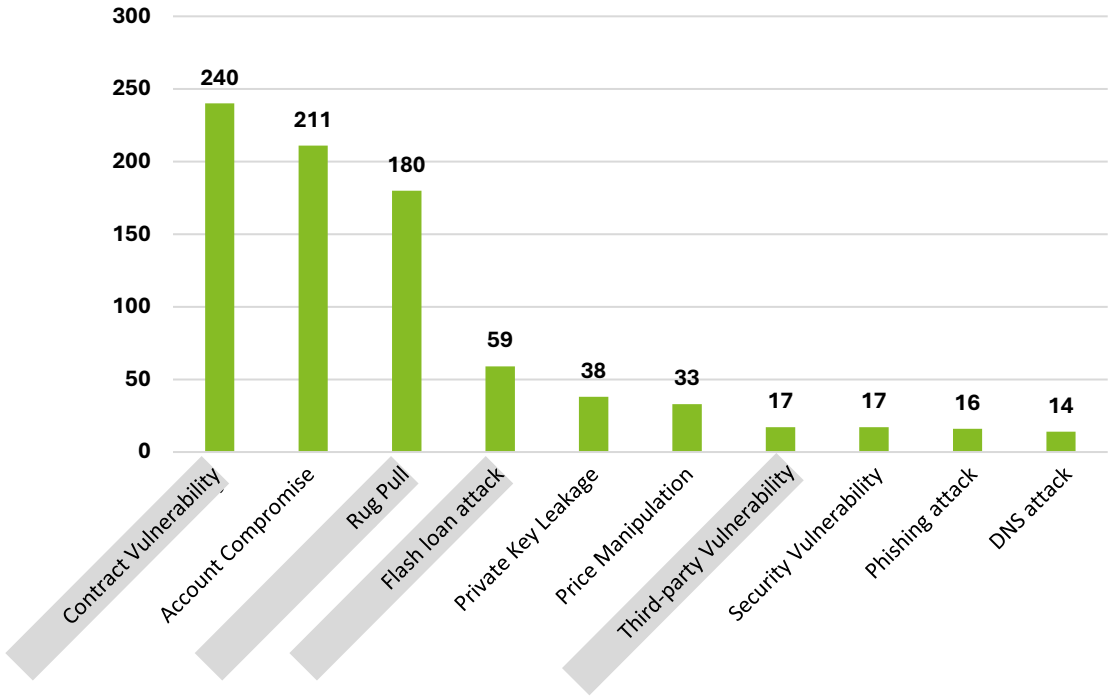
Source:「Crypto Wallets Threat Intelligence Report」（NOMINIS）_Nov 2024

Since 2023, there has been a significant increase in attacks targeting third-party vulnerabilities or smart contract vulnerabilities.

Amount of loss by attack method (cumulative since 2023)



Number of incidents by attack method (cumulative since 2023)



Source : SlowMist Hacked - SlowMist Zone (SlowMist Hacked) _ Statistics of major blockchain hacking incidents occurred between January 6, 2012 and January 21, 2026

- For the top five attack methods by amount of loss, the incidents with the largest amount of loss are outlined from the next page.
- With advances in AI and misuse of AI, incidents targeting DeFi protocols have increased in recent years, and this study highlights cases featuring distinctive attack techniques.
- Incidents categorized as “Private key leakage” often have unknown details of how keys were leaked by game platform company or the attack techniques used, therefore we did not sample from this category for further case study.

This study focused on representative cases selected for each major attack vector with significant loss impacts

Incidents with top amount of loss by attack method (January 2023 - January 2026) （1/2）

Third-party Vulnerability

	Victim	Incident Date	Loss (M$)	Outline
No.1	Bybit Case study (1)	February 2025	1,460	A crypto-asset-related company suffered an outflow of funds due to a phishing attack against a third-party wallet service developer.
No.2	Swiss Borg Case study (2)	September 2025	42	The credentials of an engineer at a third-party staking service were misused, causing a leak of staked assets.
No.3	BigONE	July 2025	27	A crypto-asset-related business operator suffered a leakage of funds due to tampering with the logic of third-party software that controls the operation of hot wallets.
No.4	Fixed Float	February 2024	26	A crypto-asset-related company lost control of its main servers, including its wallet system, and funds were stolen.
No.5	Fortress	September 2023	15	At the custodian, an employee from a third-party was targeted by social engineering that caused customer account compromises and asset loss.

Contract vulnerability

Victim	Incident Date	Loss (M$)	Outline
Cetus	May 2025	230	At the DEX, a large amount of assets were withdrawn from the liquidity pool with minimal token deposits.
Balancer v2 Case study (4)	November 2025	121	Token prices were manipulated by exploiting rounding errors in smart contracts, and buying and exchanging undervalued tokens for large amounts of money.
GMX	July 2025	42	At the DEX, token prices were manipulated and cashed out, and the assets were transferred out.
Penpie	September 2024	27	In the DeFi protocol, malicious contracts were created, and a large stake fee was leaked.
Thala	November 2024	26	In the DeFi protocol, a large amount of money was withdrawn from the liquidity pool despite the account's insufficient token balance.

Rug pull

Victim	Incident Date	Loss (M$)	Outline
LIBRA	February 2025	250	The Argentine president's propaganda sent the price of tokens skyrocketing, then plummeting, leaving many wallets with losses.
Zkasino	April 2024	33	At the gambling platform, the refund of the customer's collateral token was refused and the platformers moved the collateral token outside.
BALD	July 2023	26	On the Layer2 network, memecoin issuers cashed out a lot of tokens when prices skyrocketed.
IBX trade	October 2024	22	User funds were transferred to the project organizer's address during project funding.
Essence Finance	October 2024	20	In the stablecoin project, the price collapsed immediately after a large amount of collateral tokens were withdrawn.
Kokomo Finance Case study (6)	March 2023	4	The organizer replaced the contract with malicious code and transferred the user assets to an external address.

Note: The terms used for attack methods (e.g., 'Third-Party Vulnerability' and 'Contract Vulnerability') are quoted directly from the original source.
Note: Some attacks can be classified into multiple attack categories.
【Data Source】https://hacked.slowmist.io/en/ SlowMist Hacked - SlowMist Zone As of January 2026

This study focused on representative cases selected for each major attack vector with significant loss impacts

Incidents with top amount of loss by attack method (January 2023 - January 2026) （2/2）

Flash loan attack

	Victim	Incident Date	Loss (M$)	Outline
No.1	Euler Finance Case study (5)	March 2023	197	The attacker took advantage of a vulnerability in a smart contract which allowed users to donate an unlimited amount of collateral tokens, intentionally causing a forced liquidation, and earned a large reward by impersonating a liquidator.
No.2	Hedgey	April 2024	45	The attacker used a flash-loan smart contract vulnerability to drain funds from two chains, Ethereum and Arbitrum.
No.3	Sonne Finance	May 2024	20	By exploiting a known flaw in the Compound v2 fork (Precision Loss), the attacker was able to extract more underlying assets by returning fewer collateral tokens than needed.
No.4	Polter Finance	November 2024	12	The protocol had vulnerabilities to price oracle manipulation, allowing the attacker to borrow a flash loan, then change the perceived price of the token, borrow and repay it with inflated collateral, thus earn the difference in price.
No.5	Platypus	February 2023	9	A vulnerability in the protocol's stablecoin solvency check feature allowed the attacker to use flash loan borrowings as collateral and then withdraw the money without paying the debt.

Malware attack

Victim	Incident Date	Loss (M$)	Outline
Radiant Capital Case study (3)	October 2024	50	A developer’s PC was infected with malware that appeared to be PDF files, and transactions were tampered with on the infected PC, resulting in an outflow of funds.
Poly Network	July 2023	10	The program’s compilation environment was infected with malware and the signing key of the validator node was stolen. The attacker used this key to generate forged block headers and verification signatures, bypassing the verification process and stealing funds.
Truflation	June 2024	6	The attacker used malware to break into the system, collected several signing keys of wallets, and then signed malicious transactions to drain funds.
Tapioca DAO	October 2024	5	A developer’s PC was infected with malware that leaked signing keys, which were then used to hijack a token vesting contract and steal the tokens in the contract.
CoinStats	June 2024	2	An employee was social-engineered to download malware onto internal computers that gained access to AWS infrastructure, which led to the theft of signing keys and fund drain.

Note: Some attacks can be classified into multiple attack categories.
【Data Source】https://hacked.slowmist.io/en/ SlowMist Hacked - SlowMist Zone As of January 2026

We also included cases from February 2026 onward, focusing primarily on attacks targeting crypto-asset-related services, including DeFi.

Typical incidents after February 2026

Victim	Incident Date	Loss (M$)	Method	Outline
Resolv Protocol Case study (7)	March 2026	25	Social Engineering +Governance vulnerability	The attacker compromised an external collaborator's GitHub credentials, gained access to Resolv's repository, and stole additional credentials. The attacker then infiltrated the cloud environment, modified the KMS policy to seize signing privileges, and minted a large amount of USR tokens without authorization.
Drift Case study (8)	April 2026	285	Social Engineering +Governance vulnerability	This case involved social engineering to obtain a pre-signed transaction using a durable nonce, enabling the attacker to seize admin privileges. The attacker then posted spoofed tokens as collateral to withdraw real assets such as USDC and SOL without authorization.
Litecoin Case study (9)	April 2026	0.6	Blockchain reorganization (reorg) attack	This case exploited inadequate verification of MWEB, triggering a 13-block chain reorganization (reorg). The deposit on the Litecoin chain was reverted, but the assets remaining on the Bitcoin chain were withdrawn without authorization.
Kelp DAO /LayerZero Case study (10)	April 2026	293	Infrastructure-level attack	This case abused the verification process of the LayerZero bridge (OFT), resulting in the unauthorized minting of unbacked rsETH (approximately 116,000 tokens). The unbacked rsETH was then posted as collateral in DeFi to borrow and withdraw assets, resulting in a fund outflow.

Data Source https://hacked.slowmist.io/en/ SlowMist Hacked - SlowMist Zone As of May 2026
https://neuralwired.com/2026/04/26/litecoin-mweb-zero-day-13-block-reorg/（Litecoin 13-Block Reorg: MWEB Zero-Day Exploit Explained 2026）As of April 2026

2. Study of causes and countermeasures of recent incidents
2.2 Case study

This case started with compromising the development environment of the external service provider, and legitimate signature process was executed via disguised UI, eventually resulting in the theft of funds.

Overview and timeline of Bybit case

■ The attacker launched a social engineering attack against Safe, the external wallet service used by Bybit, and tampered with the UI used to generate transactions. Through the compromised UI, Bybit personnel approved the generation of a malicious transaction. Although the details of the malicious transaction were displayed on the hardware wallet used for signing, Bybit signers signed it without sufficiently verifying the contents, and the malicious transaction was broadcast to the blockchain. As a result, control of the wallet was transferred to the attacker, leading to the outflow of funds.

Attack Timeline

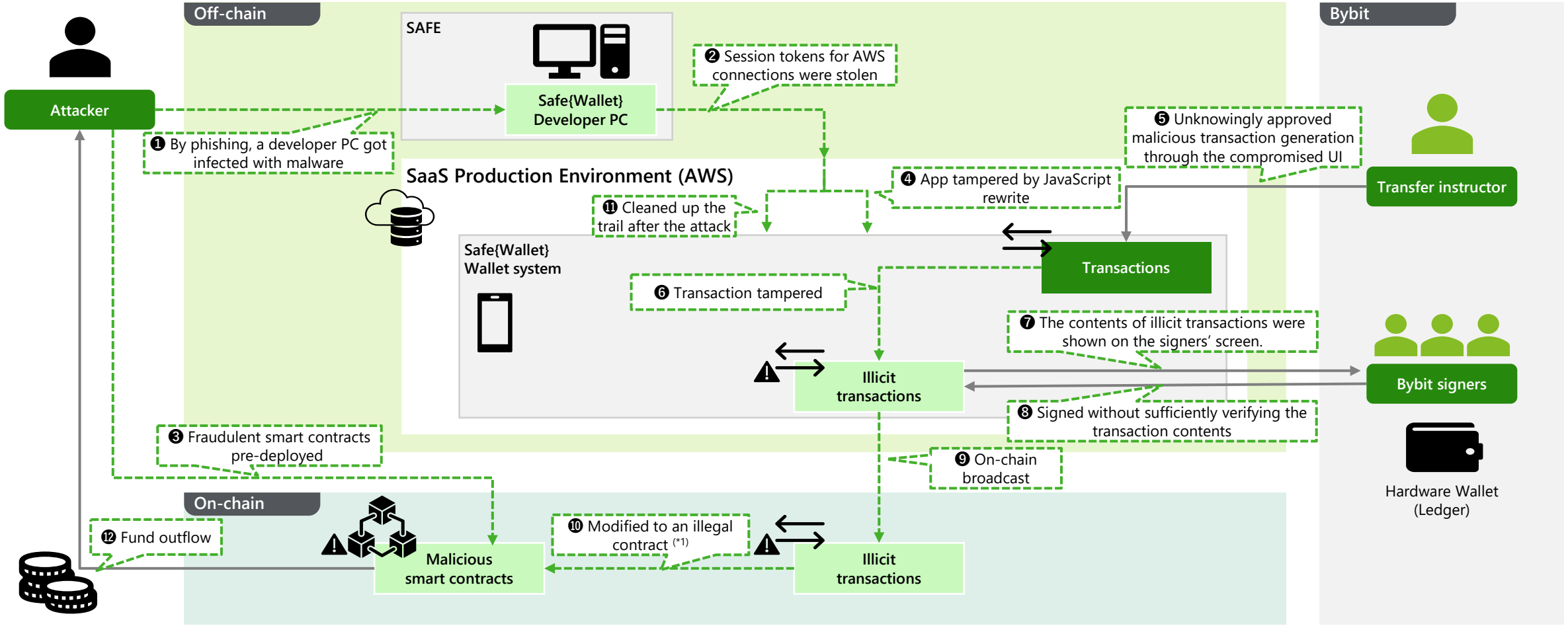
	No.	Date and time	Attacker/Victim	Description
Early invasion	❶ ❷	February 4, 2025 ~ (UTC)	**Attacker**	By phishing, **a wallet vendor's developer PC got infected with malware**. AWS access tokens started to be exploited.
	❸	February 18, 2025 15:39 (UTC)	**Attacker**	The attacker **deployed a malicious contract**. This contract is called from an illegal transaction that calls a contract implemented with a withdrawal function.
	❸	February 18, 2025 18:00 (UTC)	**Attacker**	The attacker deployed another malicious contract **implemented with a withdrawal function**.
	❹	February 19, 2025 15:29 (UTC)	**Attacker**	With malicious code, the attacker **tampered with JavaScript files** stored on AWS S3 in Safe{Wallet}.
Transaction tampering /Signature induction	❺ ❻	February 21, 2025 13:30 (UTC)	**Bybit**	A transaction started by Bybit via Safe{Wallet} was modified by **malicious JavaScript files**.
	❼ ❽	February 21, 2025 14:11 (UTC)	**Bybit**	All Bybit signers **signed the malicious transaction via a compromised Safe{Wallet} UI**. The compromised UI was sending fraudulent transactions to the hardware wallet while displaying details about the original correct contents of the transaction.
	❾ ❿	February 21, 2025 14:13 (UTC)	**Bybit**	The signed malicious transaction was broadcast to the Ethereum blockchain and Bybit **upgraded the cold wallet contract** to execute the contract's backdoor functions sweepETH() and sweepERC20() to drain the funds.
Concealment/ Fund outflow	⓫	February 21, 2025 14:15 (UTC)	**Attacker**	Two minutes after the attack, the attacker restored the JavaScript files on Safe{Wallet} AWS S3 to the original harmless files, **cleaning up the trail and hiding the breach**.
	⓬	February 21, 2025 14:16 (UTC)	**Bybit**	Via a smart contract with an embedded backdoor, **all crypto-assets in the cold wallet were stolen**.

【Source】Bybit Incident Investigation Preliminary Report (Verichain) As of January 2026

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case started with compromising the development environment of the external service provider, and legitimate signature process was executed via disguised UI, eventually resulting in the theft of funds.

The attack methods in Bybit case: from off-chain infringement to on-chain fund outflow



(*1) Consisting of two smart contracts, the attacker replaced the implementation contract called from the proxy contract with a malicious contract created by the attacker to steal crypto assets.

- Proxy contract: stores the address of the implementation contract in storage and invokes the implementation contract when instructed to do so
- Implementation contract (MasterCopy): defines rules such as multi-sig, has delegate call capability (delegates processing to an external contract)

This case started with compromising the development environment of the external service provider, and legitimate signature process was executed via disguised UI, eventually resulting in the theft of funds.

Vulnerabilities and countermeasures based on major facts

	Facts	Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
SAFE	■ SAFE offers wallet applications (hosted in AWS/S3 buckets, such as JavaScript) for crypto-asset-related businesses. ■ AWS environment credentials are stored on the developer's device.					
Attacker	■ After compromising the developer's device by phishing, the attacker stole valid session tokens for connecting to the AWS environment and accessed the AWS environment.	Inadequate anti-phishing measures	■ Education and simulated training for phishing countermeasures	Phishing (T1566)	Acquire Accounts (ADT3001)	Message Analysis (D3-MA) User Behavior Analysis (D3-UBA)
		Insufficient production credential management and storage of production environment credentials	■ Short-lived authentication, MFA, prohibition of long-term credential storage on devices, credential rotation	Steal Application Access Token (T1528) Application Access Token (T1550.001)	Supply Chain Compromise (ADT1195) Exploit External Services (ADT3008)	Credential Hardening (D3-CH) Credential Rotation (D3-CRO)
		Insufficient cloud access controls and anomaly detection	■ Conditional access, JIT access, IP/device restrictions, operation log monitoring	Valid Accounts: Cloud Accounts (T1078.004) Supply Chain Compromise (T1195)	Exploit External Services (ADT3008) Supply Chain Compromise (ADT1195)	Access Mediation (D3-AMED) Resource Access Pattern Analysis(D3-RAPA)
	■ After compromising the developer's device by phishing, the attacker stole valid session tokens for connecting to the AWS environment and accessed the AWS environment. ■ The attacker replaced the JavaScript used by the signer with a malicious one, which will: ① Execute only under certain conditions (such as transactions involving Bybit), ② Tamper transactions with malformed smart contract upgrade, ③ Manipulate the wallet UI to make the transaction appear legitimate to the signer, ④ Replace with legitimate transactions after signatures obtained.	Insufficient controls to suppress and detect unapproved program changes Insufficient detection of malicious logic Insufficient measures to prevent and detect tampering with unsigned transactions Insufficient signing confirmation process	■ CI/CD controls, code signing, change approval, host asset integrity monitoring ■ Code diff monitoring, pre-signing Tx decoding, hash verification, context validation ■ Tx verification on independent devices, dual-confirmation of signing contents	Transmitted Data Manipulation(T1565.002) Masquerading (T1036)	Exploiting Smart Contract Implementation (ADT3012) Blind Signing (similar to ADT3012.006)	File Integrity Monitoring(D3-FIM) Transaction Verification (D3-TV) File Analysis (D3-FA) File Metadata Consistency Validation(D3-FMCV)

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case started with compromising the development environment of the external service provider, and legitimate signature process was executed via disguised UI, eventually resulting in the theft of funds.

Vulnerabilities and countermeasures based on major facts

Facts	Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Bybit ■ Signers sign with a hardware wallet (Ledger). The wallet was unable to convert the contract contents into a human-readable format, thus signers did not notice the fraud. ■ As a result, a malicious smart contract upgrade was broadcast. ① Attacker Contract: Invoked by a delegate call when all signers approve an illegal transaction and rewrites the implementation code to address in ②. ② Backdoor Contract (Attacker's New Implementation): The contract contained built-in functions for draining large amounts of crypto assets.	Partially blind signing (*1) Unseparated authorities for execution of transfer and upgrade Centralized asset management in the same wallet	■ Introducing wallet functionality of human-readable message display ■ Education and training of personnel in charge to prevent signing for cyberattacks ■ Separation of authorities for execution of transfer and upgrade ■ Decentralized asset management across multiple wallets	Transmitted Data Manipulation(T1565.002) User Execution (T1204)	Insider-Assisted Access (ADT3017) Exploiting Smart Contract Implementation (ADT3012) Evil Contract (ADT3012.002)	Application Hardening (D3-AH) Application Hardening (D3-TV) System Configuration Permissions (D3-SCP) Access Mediation (D3-AMED) Inbound Session Volume Analysis (D3-ISVA)
Attacker ■ The attacker transferred the stolen funds to multiple addresses in pieces or moved to other chains by chain hopping, making it difficult to trace.				Cross-Chain Swaps (Hopping) (*2)(ADT3005) Siphon Funds (ADT3028) Layering (*3) (ADT3028.003)	

If the signers had noticed that the message hash displayed on the hardware wallet was different from the usual one, they could have detected the anomaly. However, the signers were not necessarily technically proficient and failed to recognize it.

Bybit

Experts

Using a wallet with highly readable messages does not eliminate the need to verify and understand the smart contract code, particularly when signing wallet upgrade transactions.

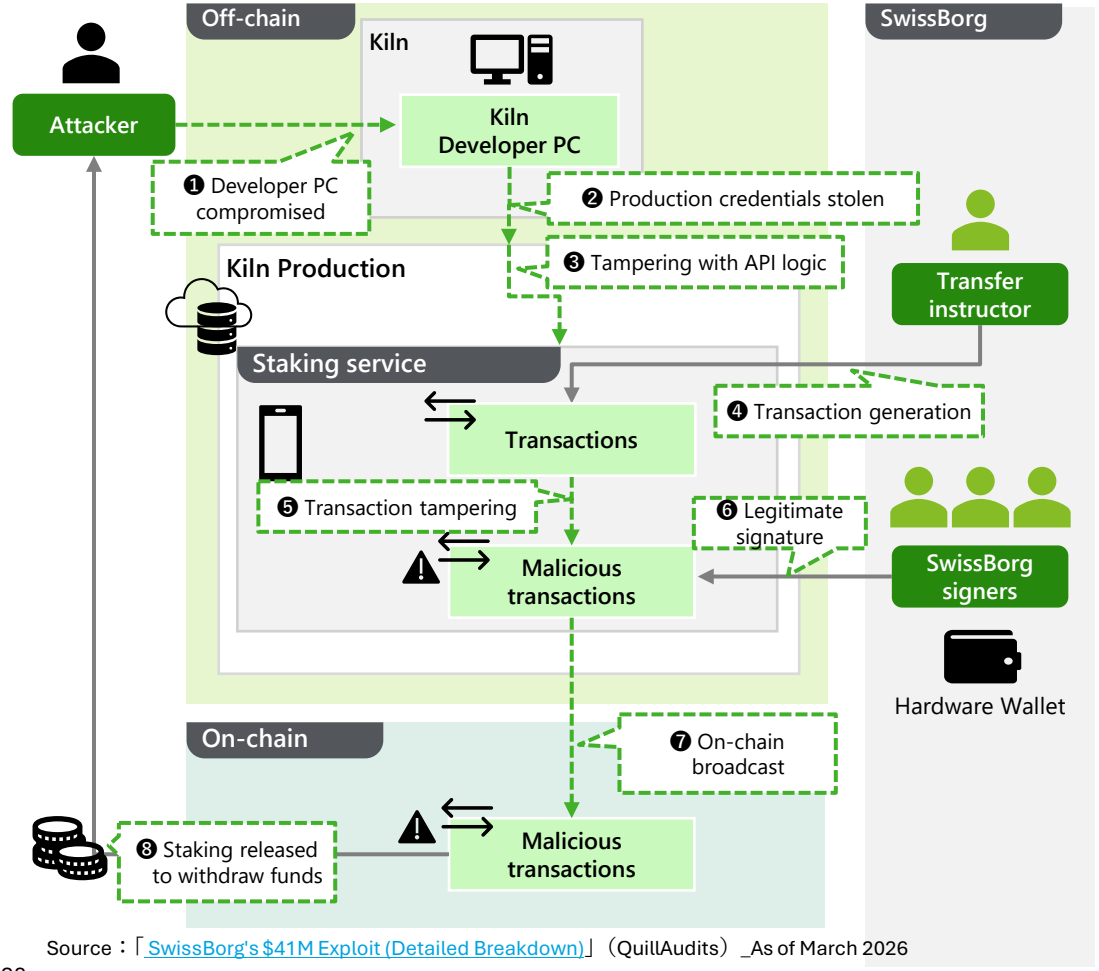
(*1): Signing a complex transaction including a smart contract without the user being able to confirm the entire signed contents on the screen. (Refer to P65)
(*2): Trading crypto assets across different blockchains to hide their origin
(*3): Tiering large amounts of crypto into smaller transactions that can be moved across different accounts, wallets, and platforms to make them harder to trace.

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case was caused by API tampering at a third-party staking service, where assets deposited for staking were stolen.

Overview and timeline of SwissBorg case

- SwissBorg, which provides operational functionality by simultaneously connecting to multiple CEXs and DEXs, staked its crypto assets on the services of Kiln, a staking platform.
- The attacker tampered with transactions by gaining unauthorized access to and modifying an external wallet service program, then seized staking withdrawal privileges without SwissBorg noticing and stole the crypto assets being staked.



Source：「SwissBorg's $41M Exploit (Detailed Breakdown)」（QuillAudits）_As of March 2026

Attack Timeline

	No.	Date and time	Attacker/ Victim	Description
Early invasion	❶ ❷	Unknown	Attacker	By compromising the GitHub access token of Kiln's infrastructure engineers (by an unknown method), the attacker was able to **steal credentials that allowed access to production systems.**
Transaction tampering/ Signature induction	❸ ❺	Unknown	Attacker	The attacker **altered the API endpoint logic** of Kiln production to return a malicious transaction in addition to a legitimate unstake transaction.
	❹ ❻ ❼	August 31, 2025 09:55 (UTC)	SwissBorg	The tampered transaction was **decoded and signed without confirmation.**
Fund outflow	❽	September 8, 2025 12:02 (UTC)	Attacker	The tampered transaction gave the attacker access to withdraw funds from the staking account, the attacker then unstaked and withdrew funds.

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case was caused by API tampering at a third-party staking service, where assets deposited for staking were stolen.

Vulnerabilities and countermeasures based on major facts

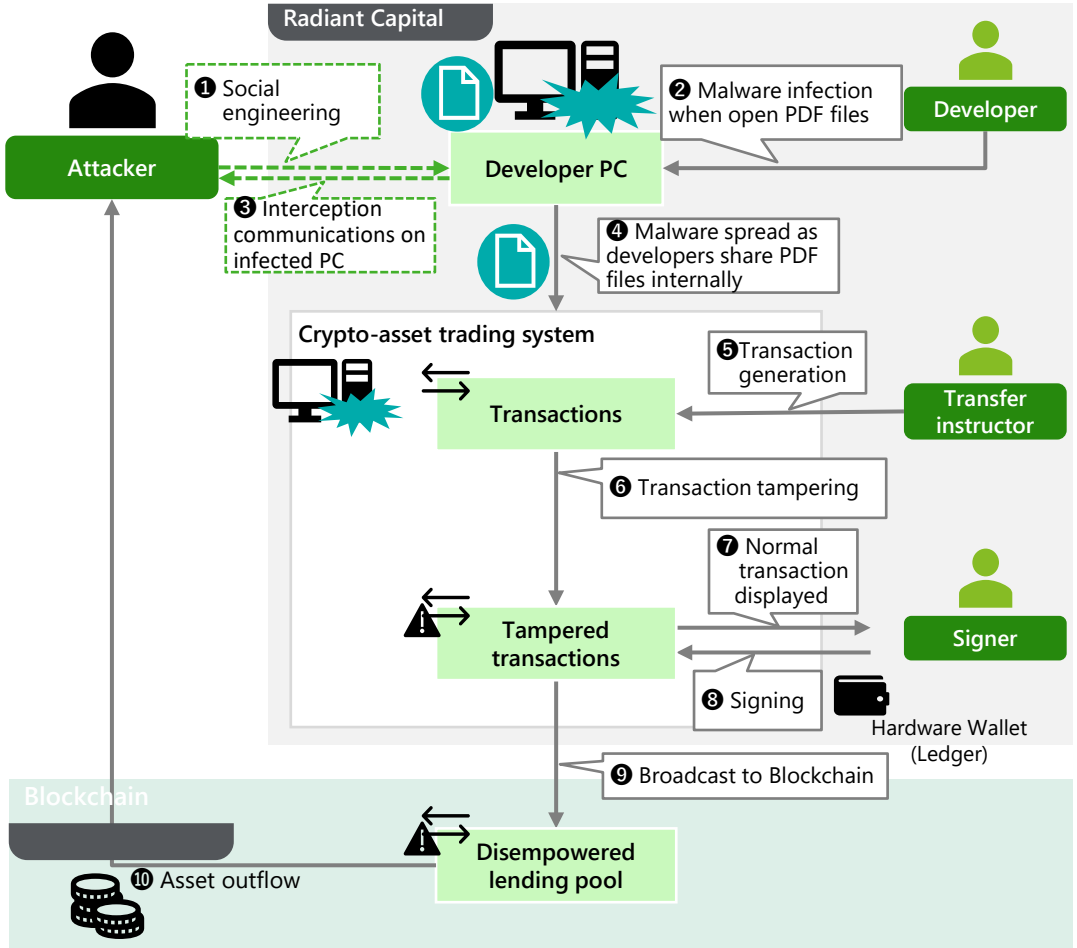
Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Swiss Borg	SwissBorg, a crypto investment platform, used the services of Kiln, a staking platform, to provide staking services.	Insufficient management of GitHub tokens, MFA enforcement and revocation controls	■ Strict management of production credentials, including short-lived credentials and rotation ■ Change approval and API integrity monitoring, API response validation, and Tx decoding ■ Verification and validation of API responses ■ Monitoring of privilege change events and threshold-based monitoring	Steal Application Access Token (T1528) Valid Accounts (T1078) Unsecured Credentials (T1552) Transmitted Data Manipulation (T1565.002) Event Triggered Execution (T1546) Supply Chain Compromise (T1195)	Acquire Accounts (ADT3001) Exploit External Services (ADT3008) Exploiting Smart Contract Implementation (ADT3012) Evil Contract (ADT3012.002) Supply Chain Compromise (ADT1195)	Credential Hardening (D3-CH) Multi-factor Authentication (D3-MFA) Credential Rotation (D3-CRO) Access Mediation (D3-AMED) File Integrity Monitoring(D3-FIM) Application Hardening (D3-AH) Transaction Verification (D3-TV) Domain Logic Validation (D3-DLV)
Kiln	Kiln is a staking platform, and its security was assessed by external reports and smart contract audits. (* 1)	Insufficient strict management of production credentials Insufficient controls for API change and deployment governance				
Attacker	① The attacker somehow compromised the GitHub access token of Kiln's infrastructure engineers and stole credentials that allowed access to production systems. ② Attackers tampered with the production Kiln API endpoint logic to append a malicious transaction to a legitimate unstake transaction. The malicious transaction changed the withdrawal authority only if the balance was over a certain amount.	Insufficient API and Tx validation Insufficient detection of triggered malicious logic				
Swiss Borg	SwissBorg signers signed and executed transactions received from the tampered API without decoding and verifying them. (* 2) As a result, the attacker gained access to the staking account.	Blind signing (*3) Unseparated authorities for execution of transfer and upgrade Centralized asset management in the same wallet	■ Introducing wallet functionality of Human-readable message display ■ Separation of smart contract authorities ■ Decentralized asset management across multiple wallets	User Execution (T1204) Data Manipulation (T1565)	Evil Contract (ADT3012.002)	Transaction Verification (D3-TV) Content Validation (D3-CV)
Attacker	The attacker executed unstaking and withdrawal to steal crypto assets.					

(*1) As this study did not obtain these external assessment results, it has not verified the scope or methodology of the assessments, nor whether any issues were identified.
(*2) Signing a transaction without decoding (content parsing) it. Signing without confirming the contents is risky and not recommended. It is mainly used for special transactions and advanced DeFi operations, such as when adding signatures.
(*3) Signing a complex transaction (including smart contract interactions) without the user being able to review the full details of what is being signed on the screen. (Refer to P65)

This case represents an incident in which transactions were manipulated, and funds were stolen through social engineering attacks targeting developers, combined with malware.

Overview and timeline of Radiant Capital case

- Radiant Capital ('Radiant') offers decentralized cross-chain financings (DeFi).
- The attacker infiltrated the system via malware that infected a developer, tampered with transactions, and induced Radiant Capital to generate electronic signatures without noticing, thereby stealing the crypto assets.



Source：「Anatomy of a $53 Million Hack: How Radiant Capital's Multisig Failed」（CoinsBench/Marcellus Nwankwo）_As of March 2026

Attack Timeline

	No.	Date and time	Attacker/ Victim	Description
Intrusions and malware infections	❶	September 11, 2024	Attacker	The attacker pretended to be Radiant's former contractor and sent malware executables disguised as PDF files.
	❷		Radiant	The malware installed a backdoor when a developer opened the PDF files and infected the developer's PC.
	❸		Attacker	The attacker intercepted communications on the infected PC.
	❹		Radiant	The malware spread as developers shared PDF files internally, infecting the crypto-asset trading systems.
Transaction tampering/ Signature induction	❺ ❻	October 16, 2024 14:47(UTC)	Radiant	The generated transfer transactions had been tampered with, and the transactions that reached the hardware wallet included a function that delegated control of the lending pool that stored Radiant's own funds.
	❼ ❽		Radiant	The transactions appeared to be normal on PCs, and the tampered transactions were signed in the hardware wallet (Ledger) without confirmation of the full contents.
	❾		Radiant	Once signed, the tampered transactions were broadcast to the blockchain.
Fund outflow	❿	October 16, 2024 17:09(UTC)	Radiant	The tampered transactions disempowered the lending pool and resulted in the outflow of funds.

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case represents an incident in which transactions were manipulated, and funds were stolen through social engineering attacks targeting developers, combined with malware.

Vulnerabilities and countermeasures based on major facts（1/2）

Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Radiant Capital	Radiant Capital (‘Radiant’) offers decentralized cross-chain financings (DeFi).					
Attacker	The attacker pretended to be Radiant’s former contractor and sent malware as follows. ① The malware was disguised as PDF files that opened as professional, detailed PDFs, but malware set up a backdoor	Insufficient external contact verification and anti-phishing measures	■ Counterparty verification, out-of-band confirmation, and anti-phishing education and training	Phishing (T1566)	Acquire Accounts (ADT3001) Exploit External Services (ADT3008)	Message Analysis (D3-MA) User Behavior Analysis (D3-UBA)
	② The attacker intercepted communications on infected PCs	Insufficient communication protection and device trust verification	■ Dedicated signing devices, communication authentication, and certificate pinning	Network Sniffing (T1040) Adversary-in-the-Middle (T1557)	Acquire Accounts (ADT3001)	Credential Transmission Scoping (D3-CTS) Message Authentication (D3-MAN)
	③ When a wallet operation is performed on an infected PC, the transaction got tampered with, and the transactions that reached the hardware wallet included a function that delegated control of the lending pool	Insufficient controls to prevent tampering with unsigned Tx	■ Tx hash verification, pre-signing verification, and confirmation on an independent device	Transmitted Data Manipulation(T1565.002)	Exploiting Smart Contract Implementation (ADT3012)	Transaction Verification (D3-TV) File Metadata Consistency Validation(D3-FMCV)
	④ The transaction appears to be normal on the PC	Overreliance on UI displays blind signing *1	■ Verification of hardware wallet display contents and dual-channel confirmation	Masquerading (T1036)	Blind Signing (similar to ADT3012.006)	Content Validation (D3-CV) Domain Logic Validation (D3-DLV)

(*1): Signing a complex transaction including a smart contract without the user being able to confirm the entire signed contents on the screen. (Refer to P65)

This case represents an incident in which transactions were manipulated, and funds were stolen through social engineering attacks targeting developers, combined with malware.

Vulnerabilities and countermeasures based on major facts（2/2）

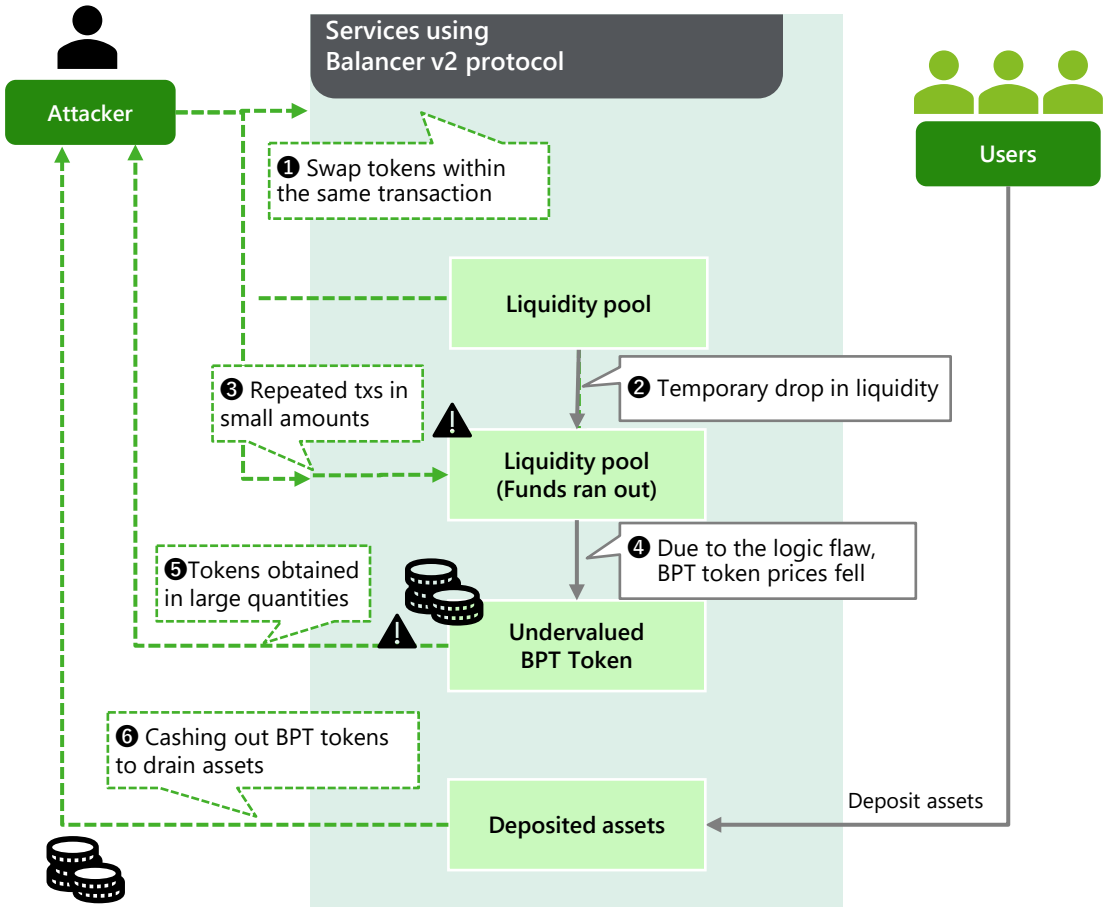
Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Radiant Capital	① Malware spread by developers sharing PDFs internally, infecting the PCs of transfer operators	Insufficient file execution controls and malware scanning Insufficient user execution countermeasures and EDR Insufficient controls for internal file sharing and lateral movement	■ Attachment isolation, file execution controls, EDR, application control, user education, attachment scan, endpoint isolation, and dedicated terminals for fund transfers	Masquerading (T1036) Obfuscated Files or Information (T1027) Malicious File (T1204.002) Internal Spearphishing (T1534)	Acquire Accounts (ADT3001)	Message Analysis (D3-MA) User Behavior Analysis (D3-UBA) File Analysis (D3-FA) Service Binary Verification (D3-SBV)
	② When a wallet operation was performed on an infected PC, it seems that trading simulations (Tenderly) were also run.	Insufficient malware detection and endpoint monitoring	■ EDR, behavior-based detection, persistence monitoring, and endpoint isolation	Ingress Tool Transfer (T1105) Command and Scripting Interpreter (T1059)		Application Hardening (D3-AH) File Integrity Monitoring(D3-FIM)
	③ The tampered transaction was signed in the hardware wallet (Ledger) without confirmation of the full contents.	Blind signing *1 Insufficient verification of privilege-changing transactions	■ Verification of hardware wallet display contents and dual-channel confirmation ■ High-risk classification of privilege changes, multi-party approval, and allowlisting	Masquerading (T1036) Transmitted Data Manipulation(T1565.002)	Blind Signing (similar to ADT3012.006) Changing Contract Ownership (ADT3012.001)	Content Validation (D3-CV) Domain Logic Validation (D3-DLV) System Configuration Permissions (D3-SCP) Access Mediation (D3-AMED)
	④ The tampered transactions disempowered the lending pool and resulted in the outflow of funds	Centralized asset management in the same wallet	■ Asset diversification	Transmitted Data Manipulation(T1565.002)	Siphon Funds (ADT3028)	File Metadata Consistency Validation(D3-FMCV)

(*1): Signing a complex transaction including a smart contract without the user being able to confirm the entire signed contents on the screen. (Refer to P65)

This case is an attack by manipulating token prices using a smart contract logic flaw of rounding errors and making a profit on the sale of tokens.

Overview and timeline of Balancer v2 case

- Balancer v2 is a DEX protocol with an Automated Market Maker (AMM) and adopted by several platforms.
- The protocol has a batchSwap function that allows successive swaps across multiple liquidity pools to be executed in a single transaction. The batchSwap function used spot settlement rules, permitting temporary borrowing or insufficient balances within a transaction.
- The attacker exploited rounding errors in the smart contract to deliberately drive down the token price, then acquired and sold the token for profit.



Attack Timeline

No.	Date and time	Attacker/ Victim	Description
			Obtaining tokens at an undervalued price by taking advantage of vulnerabilities
❶	November 3,2025 07:46 (UTC)	**Attacker**	The attacker abused the spot settlement rules to swap tokens within the same transaction.
❷		**Balancer**	A particular liquidity pool had become nearly empty.
❸		**Attacker**	The attacker **repeatedly traded small amounts in the pool.**
❹		**Balancer**	The fractions are rounded in the calculation process of transaction amount, since the pool held few assets (little liquidity), **the portion of rounded fractions in the total amount increased** for the pool. As a result, the value of the constant indicating the asset value of the pool decreased and **the price of BPT tokens (proof of deposit of assets) decreased**.
			Sale of tokens
❺ ❻		**Attacker**	The attacker obtained and exchanged undervalued BPT tokens for a large amount of money.

Source：「Understanding the Balancer v2 Exploit: Technical Analysis and Lessons for the Blockchain Industry」（OpenZeppelin）_As of March 2026

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case is an attack by manipulating token prices using a smart contract logic flaw of rounding errors and making a profit on the sale of tokens.

Vulnerabilities and countermeasures based on major facts

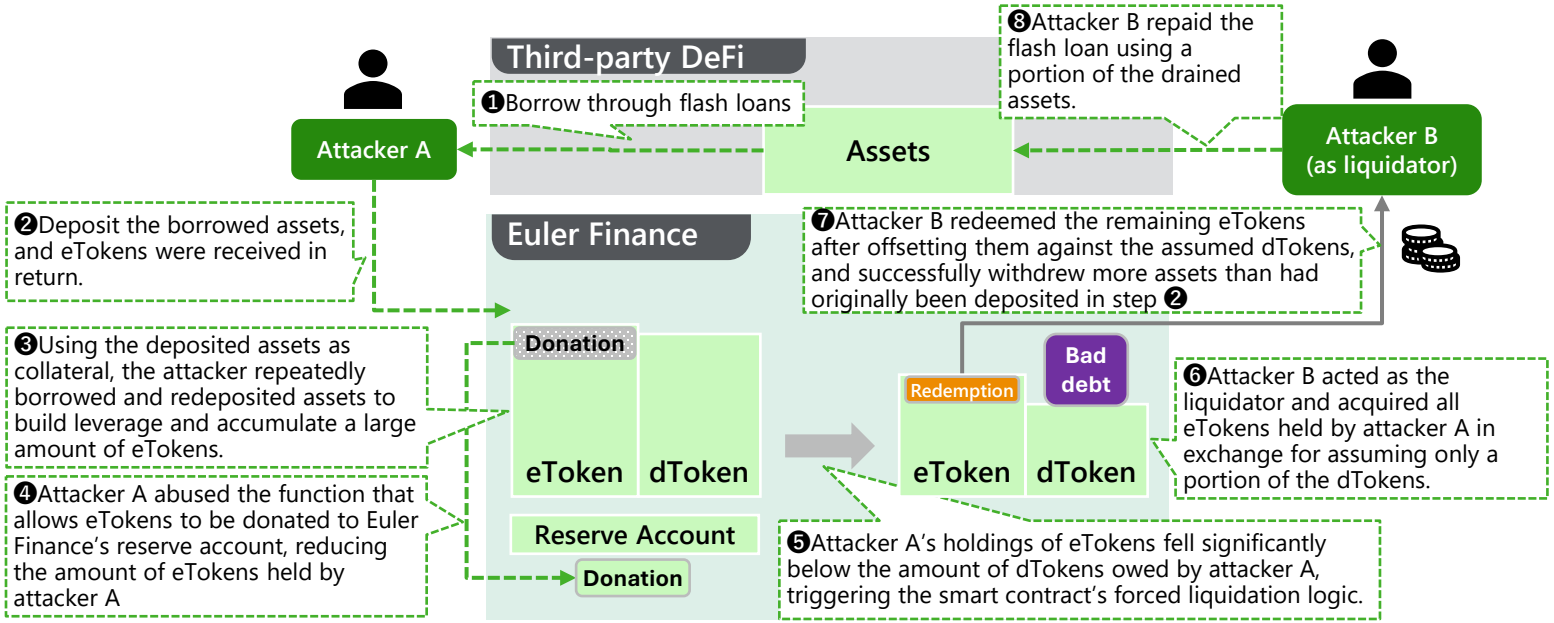
Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Balancer	■ Balancer v2 is the protocol of DEX and its Automatic Market Maker (AMM). ■ The batchSwap function allows successive swaps across multiple liquidity pools to be performed in a single transaction, saving on gas costs. In batchSwap, there were spot settlement rules to allow for temporary borrowing or insufficient balances within a transaction. ■ Balancer v2's code had been audited by multiple entities, including OpenZeppelin. (* 1).					
Attacker	① By abusing spot settlement rules, the attacker swapped tokens within the same transaction, nearly emptying a particular liquidity pool. ② The attacker repeatedly traded small amounts in the pool. Fractions were rounded in the calculation of the transaction. (Vulnerability) ③ Since the pool had little assets (little liquidity), the portion of rounded fractions in the total amount increased for the pool. ④ The value of the constant indicating the asset value of the pool decreased and the price of BPT tokens (proof of deposit of assets) decreased. ⑤ The attacker obtained and exchanged undervalued BPT tokens for a large amount of money.	Insufficient business logic validation for complex functions	■ Smart contract and formal verification, including function combinations		Smart Contract Implementation Analysis (ADT3029)	Domain Logic Validation (D3-DLV)
		Insufficient anomalous transaction controls	■ Liquidity thresholds, anomalous transaction detection, and transaction limits	Data Manipulation (T1565)	Exploiting Smart Contract Implementation (ADT3012)	Platform Monitoring (D3-PM)
		Insufficient validation of rounding errors and fractional processing	■ Rounding review, boundary value testing, and economic impact assessment			
		Insufficient safety controls under low-liquidity conditions	■ Minimum liquidity requirements and low-liquidity restrictions		Market Manipulation (ADT3021)	
		Anomaly detection for pricing logic	■ Price deviation monitoring, invariant monitoring, and oracle-assisted validation	Stored Data Manipulation (T1565.001)		File Metadata Consistency Validation(D3-FMCV)
		Insufficient safeguards against illicit profits	■ Circuit breakers and trading suspension		Siphon Funds (ADT3028)	

(*1): Among the publicly available smart contract audit reports, the most recent report (dated September 2022) covering the attacked component, Composable Stable Pools, excluded the functions responsible for rounding/decimal handling from the audit scope.

This case is a fund theft by exploiting a smart contract vulnerability to perform transactions using funds obtained with flash loans.

Overview and timeline of Euler Finance case

- Euler Finance ("Euler") is a DeFi protocol that issues eTokens representing a user's share of deposited assets. When a user borrows assets from Euler, the user receives dTokens representing the corresponding debt obligation.
- On Euler, users can build leveraged positions by repeatedly borrowing and redepositing assets using the deposited assets as collateral.
- The attacker exploited a vulnerability in the donation function contract, which failed to perform an account health check after a donation, together with the liquidation mechanism that allows liquidators to acquire collateral at a discount, to steal funds.



	No.	Date and time	Attacker/ Victim	Description
Flash loan borrowing	❶	March 13, 2023 08:50 (UTC)	Attacker A	Attacker A **borrowed large amounts of crypto using other DeFi flash loans.**
	❷		Attacker A	The borrowed crypto assets were then transferred to Euler's attacker account for deposit and eTokens were received in return.
Transaction abusing the donation function	❸		Attacker A	Using the deposited assets as collateral, the attacker repeatedly borrowed and redeposited assets to build leverage and accumulate a large amount of eTokens.
	❹		Attacker A	By donating a portion of eTokens to the contract, **the attacker's account became insolvent, but it wasn't blocked.**
	❺		Euler	Attacker A's holdings of eTokens fell significantly below the amount of dTokens owed by attacker A, triggering the smart contract's forced liquidation logic.
	❻		Attacker B	Attacker B acted as the liquidator and acquired all eTokens held by attacker A in exchange for assuming only a portion of the dTokens.
Flash Loan Repayment and Profit Extraction	❼		Euler	Attacker B redeemed the remaining eTokens after offsetting them against the assumed dTokens, and **successfully withdrew more assets than had originally been deposited in ❷.**
	❽		Attacker B	**Attacker B repaid the flash loan using a portion of the drained assets.**

Source：「$197 Million Stolen: Euler Finance Flash Loan Attack Explained」（Chainalysis）_As of March 2026

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case is a fund theft by exploiting a smart contract vulnerability to perform transactions using funds obtained with flash loans.

Vulnerabilities and countermeasures based on major facts

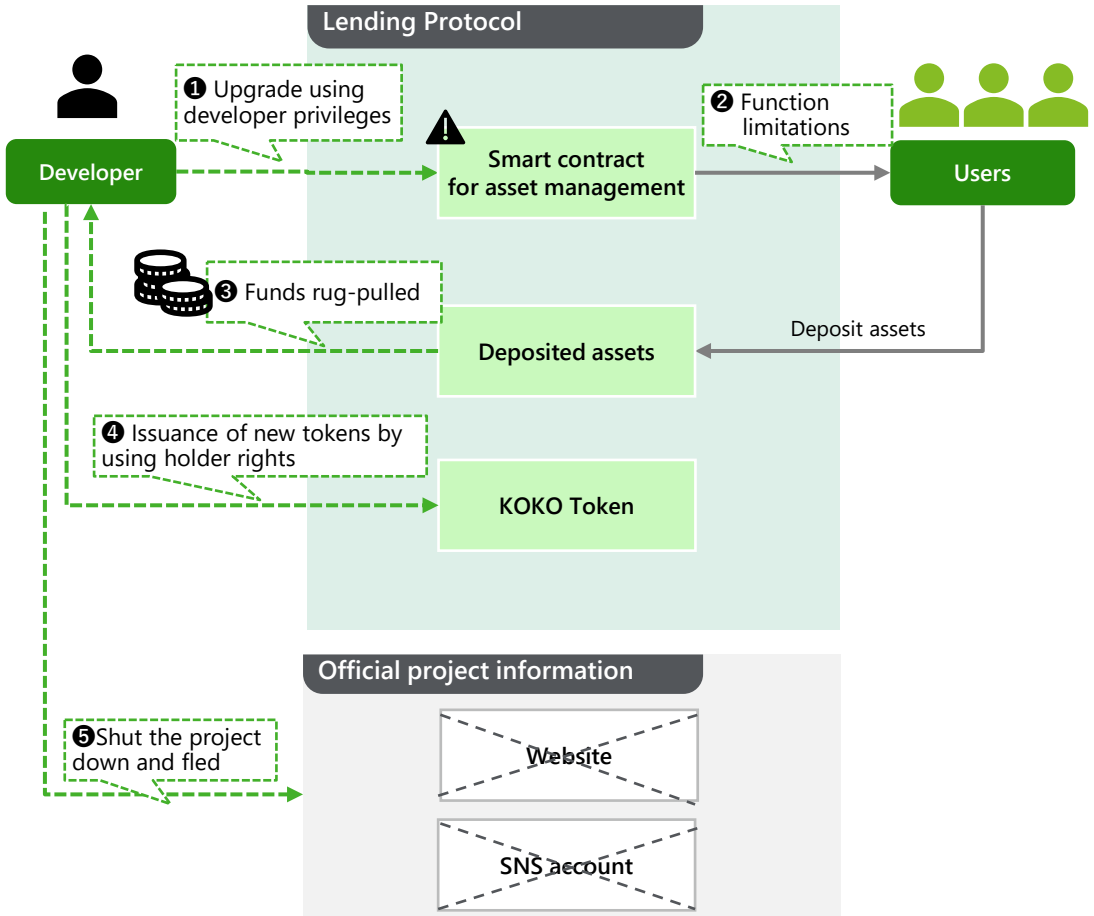
Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Euler	■ Euler Finance (‘Euler’) builds DeFi lending protocols on the ETH chain, where users can lend and borrow crypto assets. ■ A donation function was added to the smart contract, but the implementation lacked logic to block donations that would cause an account to become insolvent (vulnerability). ■ There was a smart contract audit conducted, but no vulnerability was found in the donation function(*1).	Smart contract audits did not prevent the breach				
Attacker	The funds were stolen by the following steps. ① The attacker borrowed a large amount of crypto-assets using other DeFi flash loans. ② The borrowed crypto assets were then transferred to Euler’s attacker account for deposit and received collateral tokens (Euler issues collateral tokens based on the funds deposited by the users). ③ Using the deposited assets as collateral, the attacker repeatedly borrowed and redeposited assets to build leverage and accumulate a large amount of eTokens. ④ By donating a portion of the collateral tokens to the contract, the attacker’s holdings of eTokens fell significantly below the amount of dTokens owed by attacker A, triggering the smart contract’s forced liquidation logic. ⑤ The attacker redeemed the remaining eTokens after offsetting them against the assumed dTokens. ⑥ The attacker successfully withdrew more assets than had originally been deposited, and repaid the flash loan using a portion of the drained assets.	Insufficient risk controls assuming large-scale instant liquidity Potential abuse of collateral token minting logic Insufficient controls over leverage and debt positions Lack of insolvency prevention logic in the donation function Insufficient validation of liquidation logic abuse Insufficient liquidation reward caps and anomaly detection Insufficient emergency suspension mechanisms	■ Flash loan resilience testing and large transaction monitoring ■ Protocol-wide state transition validation ■ Position limits and risk parameter monitoring ■ Safety condition validation for donation functions and insolvency safeguards ■ Pre- and post-liquidation validation and anomalous liquidation detection ■ Liquidation reward caps and anomalous reward detection ■ Transaction-time invariant monitoring and emergency suspension	Data Manipulation (T1565)	Flash Loan (ADT3015) Intercept API Communication (ADT3019) Exploiting Smart Contract Implementation (ADT3012) Market Manipulation (ADT3021) Siphon Funds (ADT3028)	Platform Monitoring (D3-PM) Domain Logic Validation (D3-DLV) Transaction Verification (D3-TV)

(*1) The auditing firm, which overlooked the vulnerability in the donation function, later admitted the flaws in the smart contract audit and paid compensation to Euler Finance.

This case is a rug pull by DeFi protocol developers who abused developer privileges over the smart contract.

Overview and timeline of Kokomo Finance case

- The Kokomo Finance project (“Kokomo”) started operating on Optimism (Ethereum’s Layer 2 network) as an open-source lending protocol and also issued KOKO tokens. Soon after it went live, it was listed on a major blockchain data platform, and many users deposited funds into it.
- Shortly thereafter, the operators abruptly transferred users’ deposited assets and generated funds through large-scale minting, then disappeared.



Attack Timeline

No.	Date and time	Attacker/ Victim	Description
Contract tampering			
❶	March 26, 2023 09:00(UTC)	Kokomo	Once enough money had been raised, Kokomo tampered (upgraded) the smart contract using KOKO token developer privileges that manage deposited assets from users.
❷		Users	The tampered smart contracts limited users' ability to receive repayment or borrow money.
Fund outflow			
❸	March 26, 2023 14:02 (UTC)	Kokomo	Kokomo operated to consume users’ deposited assets and transferred them to an external address.
❹		Kokomo	Kokomo made a profit by issuing new tokens by using the function that allows KOKO token holders to issue a large number of new tokens (mint) to any address.
❺	March 26, 2023	Kokomo	Kokomo shut down their official website and SNS account related to the project and ran away.

Source：「Optimism DeFi Protocol Kokomo Finance Rug Pulls Users for $4 Million」（Binance News）_As of March 2026

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case is a rug pull by DeFi protocol developers who abused developer privileges over the smart contract.

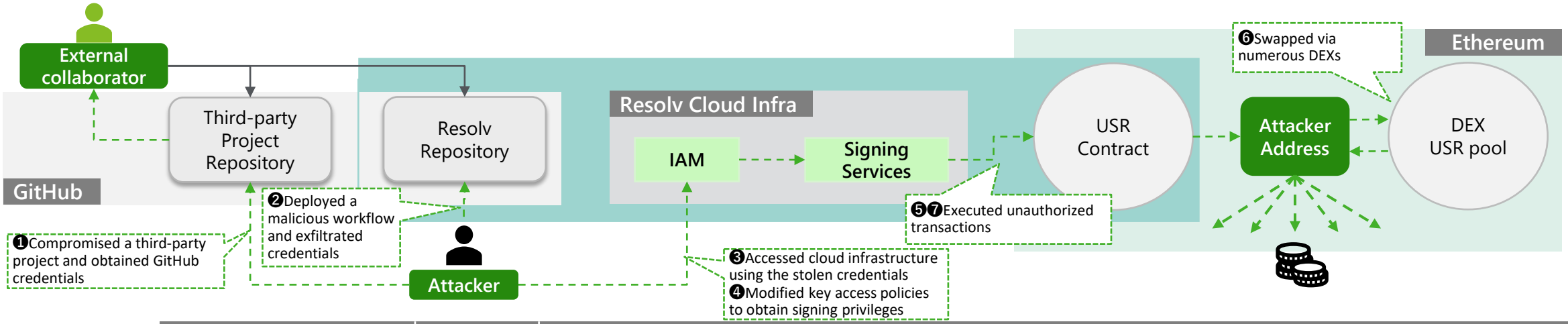
Vulnerabilities and countermeasures considered as major facts

Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE		
				ATT&CK	AADAPT	D3FEND
Kokomo Finance	■ The Kokomo Finance project (‘Kokomo’) started operating on Optimism (layer 2 network of Ethereum) as an open-source lending protocol and also issued KOKO tokens. Smart contracts used in this project had been audited. ■ Soon after the project went live, it was listed on a major blockchain data platform, and many users deposited funds into it. ■ Once enough money had been raised, the following steps were performed ① Kokomo tampered (upgraded) the smart contract using KOKO token developer privileges that manage deposited assets from users. ② The tampered smart contracts limited users’ ability to receive repayment or borrow money. Kokomo operated to consume users’ deposited assets and transferred them to an external address (rug-pulled). ③ In addition, Kokomo made a profit by issuing new tokens using the function that allows KOKO token holders to issue a large number of new tokens (mint) to any address. ④ Kokomo shut down their official website and SNS account related to the project and ran away.	It is highly likely that this project aimed to scam users for funds from the beginning, and it is considered difficult for users to take IT controls or security measures by themselves to prevent such cases. Therefore, analysis of security measures is omitted for this case.		Data Manipulation (T1565)	Market Manipulation (ADT3021) Contract Ownership Changes(ADT3012.001) Generate Counterfeit Tokens (ADT3016) Siphon Funds (ADT3028)	System Configuration Permissions (D3-SCP) Access Mediation (D3-AMED) Domain Logic Validation (D3-DLV)

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case demonstrates that smart contract audits alone are insufficient for DeFi and require integrated evaluation of GitHub, CI/CD, cloud IaaS/KMS, and signing services.

Overview and timeline of Resolv case



	No.	Date and time	Attacker/ Victim	Description
Preparation	❶	–	Attacker	An external collaborator of Resolv compromised a previously associated project and obtained their GitHub credentials.
GitHub／CI･CD compromise	❷	–	Attacker	Using the compromised GitHub credentials, the attacker accessed certain repositories, embedded a malicious workflow, and exfiltrated sensitive credentials.
	❸	–		Using the exfiltrated credentials, the attacker accessed the cloud infrastructure and performed reconnaissance, including service enumeration, API key discovery, and privilege assessment.
	❹	–		The attacker modified key access policies via infrastructure policy management functions to obtain signing privileges for the mint completion process.
Compromise of Signing Privileges / Illicit Minting	❺	March 22, 2026 02:21(UTC)		Executed the first unauthorized transaction against the Resolv contract, depositing 100,000 USDC and illicitly minting 50 million USR.
	❻	March 22, 2026 02:21～03:41 (UTC)	Attacker	Converted the illicitly minted USR into ETH and other assets via multiple wallets and numerous DEX swaps.
	❼	March 22, 2026 03:41 (UTC)		Executed the second unauthorized transaction against the Resolv contract, depositing 100,000 USDC and illicitly minting 30 million USR.
Incident Response		March 22, 2026 05:16/05:30 (UTC)	Resolv	Detected anomalous transactions through real-time monitoring, suspended all related contracts, and revoked the compromised credentials.
		–		Rotated credentials for CI/CD, cloud API keys, authentication services, and VPN, and froze approximately 46 million USR.

【Source】Resolv Postmortem: March 22, 2026 Incident

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case demonstrates that smart contract audits alone are insufficient for DeFi and require integrated evaluation of GitHub, CI/CD, cloud IaaS/KMS, and signing services.

Vulnerabilities and countermeasures based on major facts (1/2)

Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
External collaborator	■ A third-party project associated with an external collaborator was compromised, and GitHub credentials were stolen ■ The stolen credentials were used to gain initial access to Resolv's GitHub repository	Over-privileged external collaborator access Insufficient credential revocation and access review	■ Restrict external collaborator access and enforce least privilege ■ Use fine-grained tokens, enforce MFA, and PATs ■ Immediately revoke access upon project completion and enforce time-bound external collaborator privileges	Valid Accounts (T1078)	Acquire Accounts (ADT3001) Exploit External Services (ADT3008) Siphon Funds (ADT3028)	Access Mediation (D3-AMED) Credential Hardening (D3-CH) Credential Rotation (D3-CRO) Multi-factor Authentication (D3-MFA) User Account Permissions (D3-UAP)
Resolv Protocol	■ Exploited GitHub workflows to exfiltrate sensitive credentials ■ Used the stolen credentials to access the cloud environment and conduct reconnaissance, including API key discovery and service enumeration	Insufficient protection of CI/CD credentials and GitHub monitoring	■ Transition to short-lived authentication (e.g., OIDC), isolate signing operations from the CI/CD authentication chain, require mandatory reviews for workflow changes	Steal Application Access Token (T1528) Supply Chain Compromise (T1195)	Supply Chain Compromise (ADT1195)	System Configuration Permissions (D3-SCP) Credential Rotation (D3-CRO)
		Potential lateral movement using cloud credentials Insufficient anomaly detection for cloud API usage and privilege exercise	■ Segregate cloud privileges, enforce conditional access, and apply JIT access ■ Monitor API usage, alert on privilege exercise, and monitor configuration changes	Cloud Accounts (T1078.004) Cloud Service Dashboard (T1538) Account Discovery (T1087)	Exploit External Services (ADT3008)	Access Mediation (D3-AMED) Resource Access Pattern Analysis (D3-RAPA) Platform Monitoring (D3-PM)

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case demonstrates that smart contract audits alone are insufficient for DeFi and require integrated evaluation of GitHub, CI/CD, cloud IaaS/KMS, and signing services.

Vulnerabilities and countermeasures based on major facts (2/2)

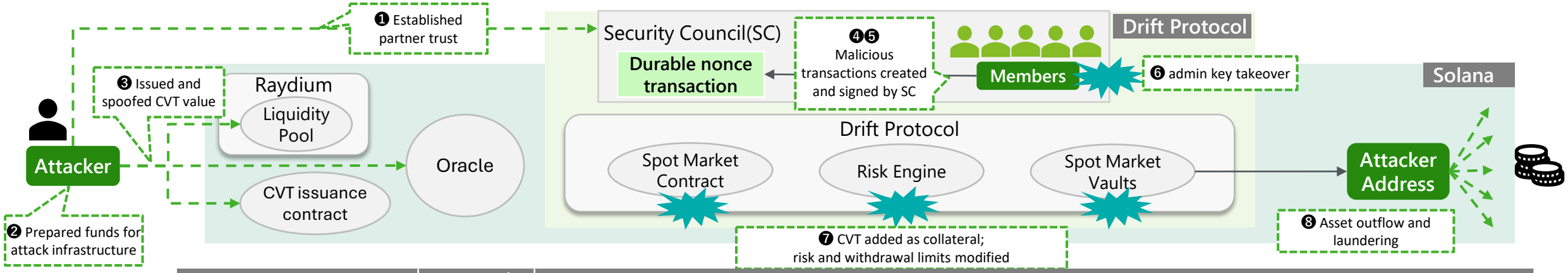
Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Attacker	■ The attacker obtained signing privileges by modifying the KMS key policy.	Insufficient KMS privilege separation	■ Dual approval and segregation of duties for KMS key policy changes ■ Real-time detection of KMS key policy change events	Additional Cloud Roles (T1098.003) Valid Accounts (T1078)	Acquire Accounts (ADT3001)	System Configuration Permissions (D3-SCP) Access Mediation (D3-AMED)
	■ The attacker illicitly minted USR using completeSwap.	Overreliance on off-chain signing privileges	■ Threshold-based signing keys and multi-party approval ■ Transaction context validation for signing targets ■ Automatic privilege revocation for high-risk operations	Data Manipulation (T1565)		Domain Logic Validation (D3-DLV) Access Mediation (D3-AMED) Transaction Verification (D3-TV)
		Insufficient mint limits and price validation	■ Mint limits and oracle price validation ■ Anomalous mint detection and automatic suspension ■ Invariant monitoring of issuance and collateral value		Generate Counterfeit Tokens (ADT3016) Siphon Funds (ADT3028)	

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case involved prolonged social engineering and pre-signed transactions compromising protocol administrative privileges, enabling borrowing with fraudulent collateral.

Overview and timeline of Drift case

- The attacker obtained pre-signed transactions using Solana's durable nonce through long-term trust-building with Drift Protocol stakeholders.
- After compromising Drift Protocol's administrative privileges, the attacker enabled the use of worthless self-issued tokens as collateral and withdrew tokens (USDC, SOL).



	No.	Date and time	Attacker/ Victim	Description
Preparation	❶	From autumn 2025 to spring 2026	Attacker	The attacker engaged Drift Protocol stakeholders via international conferences and platforms such as Telegram, and built trust as a partner through activities such as vault onboarding and deposits exceeding USD 1 million.
	❷	March 11, 2026	Attacker	The attacker withdrew 10 ETH originating from Tornado Cash and used it as preparatory funds for attack infrastructure and the creation of fraudulent collateral.
	❸	Around March 12, 2026	Attacker	The attacker created CarbonVote Token (CVT) and minted approximately 750 million tokens. Minimal liquidity was placed on the DEX Raydium, and self-trading was used to spoof a price of around USD 1. The attacker-controlled oracle then supplied the price data to Drift.
Administrative privilege takeover	❹	March 23–30, 2026	Attacker / Drift	The attacker created a durable nonce account and obtained pre-signed transactions from Security Council signers, which included the transfer of administrative privileges and other actions.
	❺	Around March 26–27, 2026	Drift	The Security Council transitioned to a 2-of-5 multisig configuration with zero timelock, and the attacker secured the required signatures under the new setup.
	❻	April 1, 2026 16:05 (UTC)	Attacker	The attacker executed the pre-signed transactions and transferred the admin key to an attacker-controlled address.
Asset outflow	❼	Immediately after the takeover	Attacker	The attacker added CVT to the collateral market, modified risk parameters and withdrawal limits, and deposited 500 million CVT.
	❽	April 1, 2026 ～18:31 (UTC)	Attacker	The attacker executed 31 withdrawals to drain legitimate assets, swapped the withdrawn funds on Solana, bridged them to other chains such as Ethereum, and converted them into ETH and distributed them.

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

【Source】Incident Recovery Update – April 16, 2026

This case involved prolonged social engineering and pre-signed transactions compromising protocol administrative privileges, enabling borrowing with fraudulent collateral.

Vulnerabilities and countermeasures based on major facts(1/2)

	Facts	Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Drift	■ The attacker created a durable nonce account ■ The attacker obtained pre-signed transactions that appeared to be routine transactions. ■ In fact, the transactions constituted blind signing and included the transfer of administrative privileges and other actions.	Blind signing *1 Insufficient durable nonce controls	■ Dedicated signing devices and transaction decoding ■ No durable nonce for critical actions		Reuse Signing (similar to ADT3012.006) Contract Ownership Changes (ADT3012.001)	Transaction Verification (D3-TV) Credential Transmission Scoping (D3-CTS)
	■ The Security Council transitioned to a 2-of-5 multisig configuration with zero timelock. ■ The attacker transferred the admin key using the pre-signed transactions.	Insufficient thresholds and timelocks Insufficient admin transfer controls	■ Higher signature thresholds, e.g., 3-of-5 or 4-of-7 ■ 24–72 hour timelocks for critical operations ■ Admin transfer allowlist and real-time alerts		Contract Ownership Changes (ADT3012.001)	System Configuration Permissions (D3-SCP) Access Mediation (D3-AMED) Authorization Event Thresholding (D3-AZET)
	■ The attacker added CVT to the collateral market. ■ Borrow limits and risk parameters were relaxed. ■ 500 million CVT was recognized as collateral worth approximately USD 500 million.	Insufficient risk parameter controls Insufficient new collateral caps	■ Gradual increases in collateral factors and borrow caps ■ Separation of duties for market additions, oracle selection, and risk changes		Exploiting Smart Contract Implementation (ADT3012) Oracle Manipulation (ADT3012.004)	Content Validation (D3-CV) File Metadata Consistency Validation(D3-FMCV)
	■ Withdrawal limits were increased and circuit breakers were weakened. ■ JLP, USDC, SOL and other assets were withdrawn in 31 withdrawals. ■ Asset transfers continued for approximately 2.5 hours.	Insufficient withdrawal limit controls Insufficient kill switches for abnormal conditions	■ Caps on parameter-change multipliers ■ No simultaneous changes across multiple markets ■ Kill switches for abnormal large withdrawals			Resource Access Pattern Analysis (D3-RAPA)

(*1): Signing a complex transaction including a smart contract without the user being able to confirm the entire signed contents on the screen. (Refer to P65)

This case involved prolonged social engineering and pre-signed transactions compromising protocol administrative privileges, enabling borrowing with fraudulent collateral.

Vulnerabilities and countermeasures based on major facts(2/2)

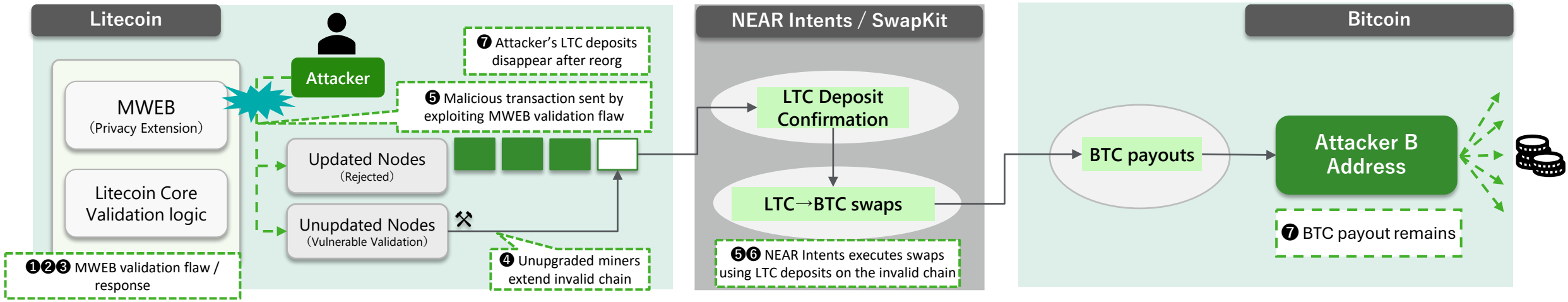
Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE		
				ATT&CK	AADAPT	D3FEND
Attacker	■ The attacker posed as a quantitative trading firm and maintained long-term contact with Drift stakeholders. ■ The attacker built trust through activities such as vault onboarding and deposits exceeding USD 1 million.	Insufficient counterparty verification	■ Signer social engineering training ■ Separated signing flow / dual-channel verification		Social Engineering/ Account Manipulation	Message Analysis (D3-MA) User Behavior Analysis (D3-UBA) Network Traffic Analysis (D3-NTA)
	■ The attacker created CVT and minted approximately 750 million tokens. ■ The attacker spoofed a price of around USD 1 through minimal liquidity and self-trading. ■ An attacker-controlled oracle referenced the spoofed price.	Insufficient new collateral screening Insufficient oracle and liquidity validation	■ Minimum liquidity / holder distribution / trading-period requirements ■ Multiple oracles / auto-rejection of abnormal prices		Generate Counterfeit Tokens (ADT3016) Oracle Manipulation (ADT3012.004)	Network Traffic Analysis (D3-NTA) Domain Logic Validation (D3-DLV) File Metadata Consistency Validation(D3-FMCV)

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case demonstrates that, absent a design premised on chain reorganization risk, deposits on an invalid chain can be converted into actual losses via third-party swaps.

Overview and timeline of Litecoin/Near intents case

- The attacker exploited a fork on the Litecoin chain and used LTC deposits on a branch chain that would later be invalidated to execute LTC-to-BTC swaps via NEAR Intents.
- NEAR Intents treated the LTC deposits as confirmed and completed the BTC payout on the Bitcoin side.
- Subsequently, Litecoin reverted to the valid chain, and the original LTC deposits were invalidated. As a result, the attacker obtained BTC without actual payment of LTC.



	No.	Date and time	Attacker/ Victim	Description
	❶	March 19, 2026	LTC Developer /Miner	Insufficient MWEB validation was identified through an internal review, and an illicit peg-out of approximately 85,000 LTC was confirmed through a chain scan.
	❷	March 19-24, 2026		The Litecoin Foundation deployed an emergency miner-only release and temporarily froze Attacker A's outpoint.
	❸			Attacker A returned the funds, which were pegged back into MWEB, and the resulting rebalancing output was frozen to restore the MWEB balance.
	❹	～April 25, 2026		While updated miners had difficulty continuing normal mining, miners without the MWEB fix extended the invalid chain by 13 blocks.
Swap on Invalid Chain	❺	April 25, 2026 8:40:16（UTC）	Attacker B	Using LTC deposits on the invalid chain, the attacker created and processed LTC→BTC swaps via NEAR Intents/SwapKit, completing the BTC payout.
	❻	April 25, 2026 8:51:13（UTC）	Attacker B	The attacker attempted additional LTC→ETH swaps. However, the swaps failed, possibly preventing further losses.
Reorg	❼	April 25, 2026	LTC Miner	The valid chain overtook the invalid chain, and the 13 illicit blocks were invalidated through a reorg.

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

【Source】Litecoin MWEB Security Incident Postmortem

This case demonstrates that, absent a design premised on chain reorganization risk, deposits on an invalid chain can be converted into actual losses via third-party swaps.

Vulnerabilities and countermeasures based on major facts

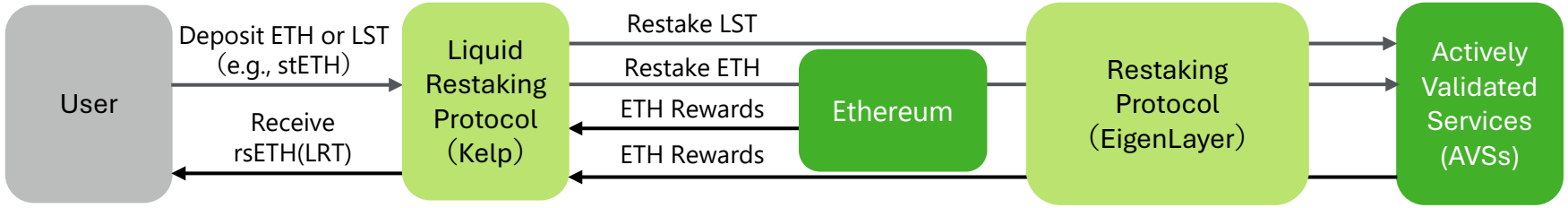
	Facts	Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Litecoin	■ It was possible to falsify UTXO metadata referenced by MWEB inputs, making a small input appear to support a much larger output under the validation logic. ■ Updated nodes rejected the invalid block; however, their behavior of retaining MWEB block data interfered with mining RPC processing for subsequent valid blocks.	Missing metadata validation Patch deployment lag Overreliance on emergency miner coordination	■ Monitor effective miner/node upgrade rates, not just public patch releases	Data Manipulation (T1565) Endpoint Denial of Service (T1499)	Exploit Blockchain Technology Specific Vulnerabilities (ADT3013)	Domain Logic Validation (D3-DLV) Operational Logic Validation (D3-OLV) Source Code Hardening (D3-SCH) Network Isolation (D3-NI)
	■ After the valid chain overtook the invalid chain, 13 invalid blocks were removed by reorg, causing the Litecoin-side deposits to disappear.	Insufficient reorg/fork resilience and finality assessment	■ Monitor invalid chains and forks ■ Maintain emergency contact ■ Prepare procedures for reconvergence to the valid chain ■ Link confirmation requirements to reorg risk ■ Increase confirmation depth based on PoW finality			Transaction Verification (D3-TV) File Metadata Consistency Validation(D3-FMCV) Network Traffic Analysis (D3-NTA)
Near Intents/ Attacker	■ NEAR Intents/SwapKit confirmed LTC deposits on the invalid chain and completed the swaps and BTC payouts. ■ Before the 13 invalid blocks were removed by the reorg, the attacker used LTC on the invalid chain as deposits for cross-chain swaps.	Overreliance on deposit confirmation from invalid block Insufficient limits and pause conditions	■ Apply swap limits and delayed settlement		Siphon Funds (ADT3028) Cross-Chain Swaps (Hopping) (ADT3005)	Access Mediation (D3-AMED) Operational Logic Validation (D3-OLV) Network Isolation (D3-NI) Transaction Verification (D3-TV)

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case involved the exploitation of a cross-chain verification vulnerability, leading to the unauthorized issuance of the LRT rsETH and spillover risks to the DeFi market.

Kelp

- Kelp provides a Liquid Restaking protocol on Ethereum. Users deposit ETH or LST and receive rsETH (LRT), which represents an economic claim on staking and restaking rewards.
- rsETH can be utilized in DeFi markets for collateral, liquidity provision, and other yield strategies while earning rewards.

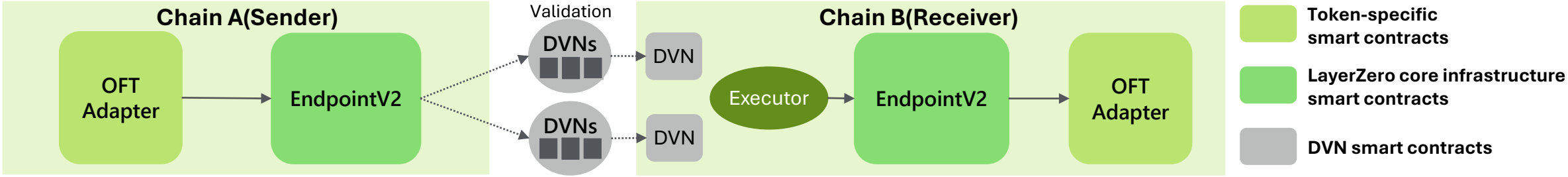


※ Liquid Staking: A mechanism where users stake ETH on Ethereum and receive LST in return
※ Restaking: A mechanism where already staked ETH is restaked into additional decentralized services (AVS) with extra slashing conditions
※ Liquid Restaking: A mechanism where users deposit ETH/LST for restaking into AVS and receive LRT in return
※ Actively Validated Service(AVS): A decentralized service that leverages Ethereum's validator set to secure and operate its own services
https://kerneldao.gitbook.io/kernel/getting-started/kelp

LayerZero Omnichain Fungible Token（OFT）

- LayerZero's Omnichain Fungible Token (OFT) standard enables a single token to move across multiple chains while maintaining total supply consistency.
- Under this standard, cross-chain transfers using LayerZero operate as follows:
  - ① The OFT Adapter on Chain A burns/locks tokens, and EndpointV2 generates a message packet
  - ② The Decentralized Verifier Network (DVN) verifies the message for Chain B
  - ③ After all designated DVNs submit verification results on Chain B, the Executor invokes EndpointV2
  - ④ EndpointV2 executes the function, and the OFT Adapter on Chain B mints/unlocks tokens

※ DVN: An independent mechanism composed of multiple nodes that verifies the authenticity and integrity of cross-chain messages. Its configuration can be flexibly defined by the token issuer based on required security policies



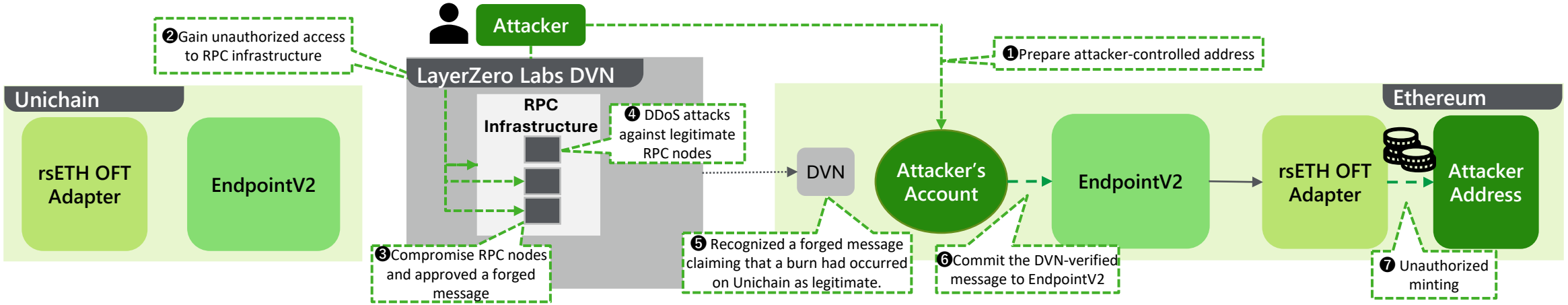
*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

【Source】LayerZero V2 OFT Quickstart

This case involved the exploitation of a cross-chain verification vulnerability, leading to the unauthorized issuance of the LRT rsETH and spillover risks to the DeFi market.

Overview and timeline of Kelp/LayerZero case

- The attacker caused the DVN to recognize a forged cross-chain message as valid by spoofing a burn of rsETH on Unichain.
- As a result, even though no legitimate burn had occurred on Unichain, the rsETH OFT Adapter on Ethereum processed an unauthorized issuance instruction and minted rsETH to the attacker's address.



	No.	Date and time	Attacker/ Victim	Description
Preparation	❶	April 18, 2026 8:39:59（UTC）	Attacker	Funds originating from Tornado Cash were deposited into the attacker-controlled account.
Unauthorized Access and DDoS Attacks	❷	～ 17:20:00	Attacker	The attacker accessed configuration information for the RPC infrastructure referenced by the LayerZero Labs DVN.
	❸			The attacker compromised two RPC nodes running on separate clusters and approved a forged message.
Function Call Using an Unauthorized Message	❹	17:20:00 ～	Attacker	The attacker launched DDoS attacks against uncompromised RPC nodes, causing the DVN's RPC requests to time out and triggering failover to the compromised RPC nodes.
	❺	17:33:35	LayerZero /Kelp	The compromised RPC nodes used by the DVN returned messages as if a cross-chain transaction that had not actually occurred existed. As a result, the LayerZero Labs DVN recognized an unauthorized message as valid.
	❻	17:35:11	Attacker	On Ethereum, the attacker committed the DVN-verified state to EndpointV2 using an attacker-controlled account.
Unauthorized minting	❼	17:35:35	Attacker	The rsETH OFT Adapter processed an unauthorized issuance instruction, resulting in the minting of approximately 116,500 rsETH.

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

【Source】KelpDAO Incident Statement As of April 2026

This case involved the exploitation of a cross-chain verification vulnerability, leading to the unauthorized issuance of the LRT rsETH and spillover risks to the DeFi market.

Vulnerabilities and countermeasures based on major facts (1/2)

Facts		Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
LayerZero Labs	■ OApp/OFT developers (e.g. token issuers) were allowed to customize DVN settings, including configurations that relied solely on a single DVN (1-of-1 DVN)	Insufficient DVN setting restrictions	■ Mandatory multi-DVN and RPC path configuration		Exploit Blockchain Technology Specific Vulnerabilities (ADT3013)	Network Isolation (D3-NI)
	■ The RPC paths used by the LayerZero Labs DVN were poisoned, but the anomaly was not fully detected because normal data was returned to monitoring systems.	Insufficient RPC anomaly detection	■ Monitor unauthorized node binary changes ■ Insider threat and cross-system monitoring controls ■ Additional consistency checks		Exploit External Services (ADT3008)	File Metadata Consistency Validation(D3-FMCV) Network Traffic Analysis (D3-NTA) RPC Traffic Analysis (D3-RTA)
	■ Compromised RPC nodes returned malicious data to the DVN	Overreliance on DVN verification results and insufficient integrity assurance	■ Multi-RPC verification ■ Message authentication and tamper resistance	Data Manipulation (T1565) Adversary-in-the-Middle (T1557)	Exploit Blockchain Technology Specific Vulnerabilities (ADT3013)	Message Hardening (D3-MH) Message Authentication (D3-MAN)
Kelp DAO	■ A configuration relying on the LayerZero Labs DVN was selected.	Overreliance on a single-DVN configuration	■ Multi-DVN configuration			
		Insufficient DVN setting restrictions	■ Explicit RPC path configuration			

This case highlights the existence of supply chain dependency risks in DeFi and the need for appropriate third-party vendor management.

Experts

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case involved the exploitation of a cross-chain verification vulnerability, leading to the unauthorized issuance of the LRT rsETH and spillover risks to the DeFi market.

Vulnerabilities and countermeasures based on major facts (2/2)

Facts	Vulnerabilities that gave rise to the incident	Measures that should have been implemented	Reference：MITRE ATT&CK	Reference：MITRE AADAPT	Reference：MITRE D3FEND
Attacker ■ The attacker accessed configuration information for the RPC infrastructure referenced by the LayerZero Labs DVN	Predictable RPC configuration information	■ Protection of RPC configuration information, access control, and limited configuration disclosure	Cloud Service Dashboard (T1538) Valid Accounts (T1078)	Exploit External Services (ADT3008)	System Configuration Permissions (D3-SCP) Credential Hardening (D3-CH)
■ The attacker compromised two external RPC nodes on separate clusters and replaced the op-geth binaries.	Insufficient risk assessment of external RPC dependencies Insufficient intrusion prevention for RPC nodes Insufficient executable binary integrity verification	■ RPC path control and external dependency risk assessment ■ Node hardening, access control, and intrusion detection ■ Executable integrity verification and tamper detection	Exploit Public-Facing Application (T1190) Modify System Image (T1601) Compromise Host Software Binary (T1554)	Exploit External Services (ADT3008) Supply Chain Compromise (ADT1195)	Resource Access Pattern Analysis (D3-RAPA) Access Mediation (D3-AMED) Service Binary Verification (D3-SBV) File Integrity Monitoring (D3-FIM)
■ The attacker launched DDoS attacks against legitimate RPC nodes	Insufficient DDoS resilience	■ DDoS attack detection logic and redundant RPC nodes	Network Denial of Service (T1498)	Exploit External Services (ADT3008)	Inbound Traffic Filtering (D3-ITF)
■ A forged message verified by the DVN was committed to EndpointV2 on Ethereum.	Insufficient additional integrity validation	■ Additional event integrity validation ■ Prevention of convergence to compromised RPCs ■ Allowance for time-based delays	Data Manipulation (T1565)	Exploit Blockchain Technology Specific Vulnerabilities (ADT3013)	Domain Logic Validation (D3-DLV) Network Traffic Analysis (D3-NTA)
■ Unauthorized token issuance was executed through the rsETH OFT Adapter.	Insufficient token issuance controls	■ Rate limits and cap controls for token issuance		Generate Counterfeit Tokens (ADT3016) Siphon Funds (ADT3028)	

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

This case involved the exploitation of a cross-chain verification vulnerability, leading to the unauthorized issuance of the LRT rsETH and spillover risks to the DeFi market.

DeFi Market Impact and Protocol Responses Following the Kelp/LayerZero case

- Aave faced growing bad debt concerns after the unauthorized rsETH was supplied as collateral, leading to severe liquidity stress across tokens, including stablecoins, and declines in token prices.
- The incident undermined confidence in LayerZero's OFT standard, prompting some DeFi protocols with no direct rsETH exposure to temporarily pause their bridges.
- Under the DeFi United framework, compensation funding, attacker position liquidation, and the rsETH recovery plan were carried out in phases.

Protocol	Impact	Responses
Kelp LayerZero	■ Credit risk spread to lending and yield markets using rsETH as collateral or in other strategies, significantly disrupting rsETH's ETH parity. About one week after the incident, 1 rsETH fell to approximately 0.8795 ETH, representing an 18.5% decline in asset value.	■ Froze all bridge functions via emergency multisig (Kelp). ■ Disclosed the incident on the same day, including the prevention of an additional $200M outflow (Kelp). ■ Isolated and replaced compromised RPC paths, prohibited 1-of-1 DVN configurations, and requested migration to multi-DVN setups (LayerZero) ■ Replenished approximately 20,373 rsETH to the OFT Adapter on May 25, 2026, completing the recovery plan (Kelp).
Aave （Circle）	■ Of the 116,500 rsETH minted without authorization, the attacker supplied 89,567 rsETH as collateral on Aave and borrowed 82,650 WETH and 821 wstETH. As a result, WETH liquidity pool balances across chains such as Ethereum, Arbitrum, and Base fell to as low as $20. ■ The decline in rsETH prices made collateral recovery less certain for liquidators, slowing liquidation activity and increasing bad debt concerns. (Total bad debt was estimated at $123.7M–$230.1M.) ■ The depletion of WETH liquidity triggered withdrawals of other safe assets, also depleting USDC liquidity. Within 24 hours, USDC supply and borrowings each declined by about $60M, while TVL fell by $8.5B over 48 hours. ■ Credit concerns over Aave caused the AAVE token price to decline by more than 20% within several days. ※ On Aave, positions are generally liquidated when the Health Factor (= collateral value × liquidation threshold ÷ borrowed amount) falls below 1. Liquidators repay the borrower's debt and receive the collateral plus a liquidation bonus.	■ Froze the WETH pool to contain spillover to other liquidity pools, including stablecoins (Aave). ■ Submitted a governance proposal to raise interest rates, aiming to attract new lending and resolve withdrawal queues (Circle). ■ Secured over $300M in ETH and advanced a phased rsETH replenishment plan as a cross-industry recovery framework to restore rsETH and normalize markets such as Aave (DeFi United). ■ Liquidated the attacker's remaining rsETH collateral positions on Ethereum/Arbitrum by May 7, 2026, and transferred the collateral to the DeFi United wallet (Aave Labs).
Morpho, Euler, ether.fi etc. **Lido**	■ Although direct exposure to rsETH was limited across DeFi protocols, confidence concerns over the broader LayerZero OFT route led to an approximately $1.5B decline in deposits on Morpho. ■ Lido's EarnETH vault had direct rsETH exposure through rsETH/ETH positions on Aave, creating a risk of unfavorable position exits amid market unwinding and liquidation of rsETH-related positions. ※ Lido Earn is Lido's asset management function that accepts tokens and allocates them to DeFi strategies such as Aave, Uniswap, and Morpho. EarnETH is the vault for ETH-denominated tokens such as ETH, WETH, stETH, and wstETH.	■ Temporarily paused OFT transfers (Morpho, Euler, ether.fi). ■ Built a joint "escape hatch" as an emergency measure to mitigate cascading damage across protocols, providing Aave users with an exit route (Fluid, Lido, Ether.fi, 1inch, etc.). ■ Paused EarnETH deposits and withdrawals (Lido). ■ Activated DAO-funded first-loss protection for potential EarnETH losses (Lido). ■ Submitted a DAO proposal to contribute up to 2,500 stETH to an rsETH relief vehicle (Lido).

*Prepared for research purposes based on publicly available information as of the time of writing; not a statement of responsibility or legal assessment.

2. Study of causes and countermeasures of recent incidents
2.3 Lessons learned from case study

Rather than attempting to steal signing keys themselves, attackers have adopted techniques aimed at exfiltrating crypto assets by exploiting vulnerabilities in the systems and processes leading up to the signing workflow.

Lessons learned from case study (1/4)

Attacker's aim	What's common in sampled cases and why they were not prevented from happening		Lessons learned
Third-party Vulnerability Case study (1)(2) Malware attack Case study (3) **To cause fraudulent transactions to be signed by tampering with the transaction or the system program leading up to the signing**	The starting point of these incidents is the compromise of developer privileges through social engineering or phishing, or the deployment of malware.	• Insufficient measures were taken to prevent credentials from being leaked through social engineering or phishing • The implemented detection or anti-malware measures failed to detect unauthorized access	• From these case studies, it was reaffirmed that cybersecurity measures must consider not only the leakage or theft of the signing keys themselves, but also scenarios involving unauthorized intrusion into, or tampering with, the systems and processes leading up to the signing step. • Where a crypto-asset-related business operator uses a third-party service for its wallet system, it is important — even if signing keys are not entrusted to the third party— to confirm that the third party has appropriately established and operates controls such as cyberattack countermeasures, production-environment access management, and program/change management. • Considering that the attacker proceeded from unauthorized access to tampering with programs in the production environment, it can be inferred that there may also have been issues with controls over developers' access to the production environment, management of the source code used in production, and the security of build/deployment processes for promoting changes to production. It is also necessary that such program change management, etc., be appropriately established and operated by third parties. • In some cases, crypto-asset businesses had reviewed external assessment results; however, rather than relying solely on those results, they are considered to need—taking into account the assessment's scope and period, the criteria used, and the qualifications/competence of the independent auditor—to conduct additional verification as necessary and introduce complementary controls. • In the cases reviewed, it might have been possible to prevent damage by strictly verifying transaction details when executing electronic signatures. When using an external system, it is important—after understanding the system's structure and governance/management framework—to analyze the impact of the external service provider's security incidents on their own company and relevant leakage risk scenarios, and, after considering what "self-defensive measures" are needed to prevent leakage, to establish a verification process to be performed at signing.
	Attackers tampered with signed transactions, as well as programs of production-system UIs/APIs	• Lack of strict control over production credentials and access, allowing attackers to tamper with production programs • Ineffective controls to prevent or detect unauthorized program changes to the production environment • Insufficient mechanisms to prevent or detect tampering with unsigned transactions	
	Tampering with UIs/APIs made it difficult for signers to notice that the transaction is fraudulent, and thus they proceeded with signing.	• Because there was no strictly designed signing framework or process that anticipated the possibility of tampering with the wallet system's UI/API, the signers were unable to notice abnormalities in the transaction contents.	

* Please refer to Appendix5 SEAL Certification Framework, Appendix 6 CryptoCurrency Security Standard, Appendix7 JVCEA Crypto Asset Security Management Standard

In utilizing smart contracts and DeFi, it is necessary to take into account not only vulnerabilities in the underlying logic but also potential shortcomings in the governance of developers and operating entities.

Lessons learned from case study (2/4)

Attacker's aim	What's common in sampled cases and why they were not prevented from happening		Lessons learned
To directly steal crypto assets stored in a smart contract by exploiting vulnerabilities in the smart contract	**Contract Vulnerability Case study (4)** **Flash loan attack Case study (5)** Attackers took advantage of vulnerabilities that result from the combination of multiple smart contracts or the lack of consideration in design	• Insufficient validation checks that take into account all possible transaction patterns under the protocol, and lack of mechanisms to detect abnormalities in prices, pools, etc. • It is possible that, in smart contract audits, the vulnerabilities in terms of business logic result from the combination of multiple smart contracts had not been examined	• The case suggests that the losses could potentially have been prevented if the smart contract had incorporated sufficient validation checks and anomaly detection mechanisms. This highlights the importance of implementing robust safeguards against misuse, taking into account that vulnerabilities in business logic can directly lead to the theft of crypto assets. • The case also indicates that the attackers likely conducted the attack with a thorough understanding of the smart contract's behavior and logic vulnerabilities, suggesting a high level of analytical capability on the attacker's side, potentially enhanced by advances in AI. • Given that the takeover of control rights through smart contract upgrades has been employed as an attack method in multiple cases, it is necessary for crypto-asset-related service providers—whether they incorporate smart contracts into their own services or rely on third parties that utilize them—to implement measures that address such risks. These include, for example, the segregation of privileges and the strict management of signing keys associated with high-level permissions. • In addition, unlike conventional contractual arrangements with third parties, the use of DeFi protocols may in some cases place the full burden of risk on the user. Accordingly, careful risk assessment should be undertaken when utilizing such protocols.
	Third-party Vulnerability Case study (1)(2) **Malware attack Case study (3)** Attackers abused the smart contract upgrade function to take control of the contract	• It is possible that the design did not separate authorities for asset transactions and upgrades which were executed with the same signing key. • It is possible that the signers had a poor understanding of the specifications and risks of the smart contract and did not notice that normal transactions had been tampered with to perform upgrades.	
Rug pull Case study (6) **To raise money for a fraudulent project and run away with it**	Attackers used anonymity to mislead users by acting like a trusted project without revealing their identities.	• Risks of projects run by people with unknown identity	• In DeFi protocol, there are projects in which the identity of the operator is unknown, and even in such cases, the operator has the privilege to freely upgrade the contract or move funds out of the pool, therefore when using it, it is necessary to **consider the lack of governance at the operator**. If the fraud risk cannot be sufficiently reduced, one should judge carefully whether to use it.

 * Please refer to Appendix4 OWASP SCSVS

It is important to verify transaction details at the time of signing through multiple measures, including wallet functions that improve message readability.

Lessons learned from case study (3/4)

Third-party Vulnerability Case study (1)(2) → Malware attack Case study (3) → Privilege compromise Case study (8)

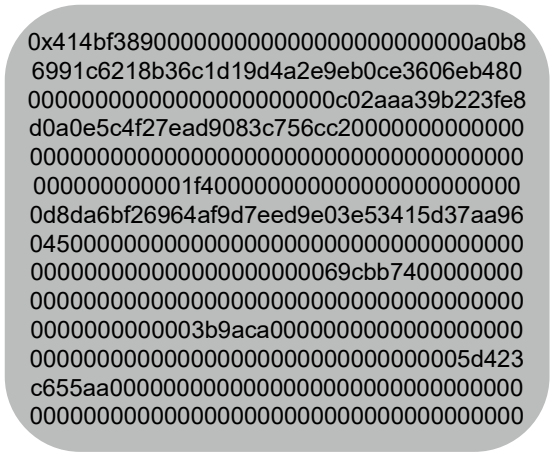
- For complex smart contract transactions, such as updates to smart contracts related to DeFi protocols or wallets, wallets may not be able to present transaction details in a human-readable format. Calldata or EIP-712 messages may instead be displayed as hexadecimal strings or data that is difficult for humans to interpret. Hardware wallets, in particular, may have limited ability to clearly present complex transaction details due to constraints such as screen size and UI design.
- When signing complex transactions involving smart contracts, signers should understand not only the recipient address and transfer amount, but also what operation the transaction will perform, what privileges it will grant, and to whom.

Clear Signing

- On May 12, 2026, the Ethereum Foundation, together with wallet developers, security firms, and other ecosystem participants, announced Clear Signing as a framework aimed at achieving “What You See Is What You Sign” at the time of signing, with the goal of reducing blind signing.
- ERC-7730, the core technology of Clear Signing, is a JSON-based descriptor standard for presenting structured data, such as smart contract calls and EIP-712 messages, in a human-readable format. Originally created as a draft ERC in February 2024, ERC-7730 has since evolved into a broader ecosystem standard. In particular, ERC-7730 v2 expanded its applicability to wider use cases, including cross-chain interactions and software wallets.
- Clear Signing is designed as a comprehensive framework that includes a neutral, mirrorable registry for distributing descriptors, a mechanism that enables independent reviews and attestations of descriptor accuracy, and implementation tools for wallet and protocol developers.
- Clear Signing can be effective in reducing the risk of asset outflows caused by insufficient pre-signing verification of complex transactions, such as bridge and cross-chain transactions, staking, privilege grants, and smart contract upgrades and administration.
- However, the effectiveness of Clear Signing depends on the accuracy of the displayed information, the authenticity of descriptors, and the model adopted by each wallet. Incorrect descriptors or compromised distribution channels may result in misleading displays. Therefore, the information presented through Clear Signing should not be treated as unconditionally safe, and Clear Signing should be used together with other measures, including double-checking approvals, anti-malware controls, third-party risk management, segregation of duties, and anomaly detection.

<Image>

calldata



Clear Signing

Action	Swap
Send	1.000USDC
Receive min.	0.42WETH
Protocol	Uniswap V3
Network	Ethereum
…	

The root causes of asset outflows in DeFi are primarily attributed to control deficiencies outside the blockchain and represent risks that are common across crypto-asset-related operations, including CEXs.

Lessons learned from case study (4/4)

Attacker's aim	What's common in sampled cases and why they were not prevented from happening	
Attackers do not necessarily target signing keys or smart contract vulnerabilities directly. Instead, they compromise surrounding infrastructure and operational processes, such as CI/CD, cloud IAM/KMS, and administrator privileges, and steal assets by undermining the assumptions that DeFi protocols rely on for issuance, collateralization, withdrawals, and cross-chain verification.	Incidents originated from compromises in off-chain domains, including GitHub, CI/CD, cloud IAM/KMS, and signing processes.	■ Controls and assessments of the overall system, including the supply chain, may have been insufficient.
	The sampled cases involved the abuse of high-authority operations, such as admin rights, signing authority, minting permissions, and risk-parameter control.	■ Controls over high-risk operations may have been insufficient. (e.g., segregation of duties, multi-party approval, timelocks, change limits, and destination restrictions.)
	The attackers abused human signing and approval actions as attack vectors. (e.g., blind signing and pre-signed transactions.)	■ Human controls and signing controls may have been insufficient. (e.g., mechanisms and operational procedures to verify the substantive details of signing targets, as well as signer education.)
	The attackers abused structures that rely on trust in external dependencies, such as cross-chain verification and chain finality, causing false data or transactions on invalid chains to be accepted.	■ Design and monitoring for external risks may have been insufficient. (e.g., single-RPC reliance, insufficient multi-source verification, insufficient reorg resilience, and insufficient anomaly detection.)
	Even after execution, unauthorized minting and withdrawals continued, with the attackers executing multiple transactions in rapid succession.	■ Real-time monitoring and deterrence functions may have been insufficient. (e.g., anomalous transaction detection, economic invariant monitoring, circuit breakers, and automated suspension.)

Lessons learned

- In sampled cases, the security of DeFi protocols should be assessed not only based on whether smart contract audits have been conducted, but also through an integrated risk assessment covering GitHub, CI/CD, cloud IAM/KMS, signing services, oracles, RPC, external collaborator management, and chain finality.
- For high-privilege operations, organizations should combine controls rather than relying solely on multisig, such as segregation of duties, enhanced thresholds, timelocks, transfer-destination allowlists, dual approval, decoded verification of signing contents, and transaction context validation for signing targets.
  In particular, admin transfers, KMS policy changes, additions of collateral markets, oracle changes, issuance limit changes, and withdrawal limit changes should be treated as high-risk operations distinct from ordinary transfers.
- When using cross-chain or bridge mechanisms, organizations should avoid reliance on a single RPC by introducing multi-source verification, confirmation requirements based on finality risk, delayed settlement, circuit breakers, and monitoring of invariants for issuance volume and collateral value.
- The root cause of the attack lies in deficiencies in privilege management, credential management, CI/CD, cloud configuration, signing controls, third-party management, program change management, and cross-chain connection management. Similar risks also arise in centralized services (CEXs, custodians, wallet service providers, etc.)
- In particular, compromises in off-chain domains (GitHub, CI/CD, cloud IAM/KMS, signing services, etc.) should not be regarded as issues specific to DeFi, as they may also affect multiple areas at crypto-asset exchange service providers (hot wallet management, transfer approval, key management infrastructure, API infrastructure, asset movement management, etc.). Blind signing, insufficient verification of signing details, excessive concentration of administrative privileges, reliance on external services, and insufficient automated suspension in abnormal situations are also common risks in operations by CEXs and other service providers (withdrawal approval, asset movement, wallet updates, external custody connections, etc.). Therefore, multi-layered controls are required for both DeFi and CeFi.
- When centralized service providers use or connect to DeFi infrastructure or protocols, they should verify third-party controls over those external services and conduct a scenario-based assessment of the spillover risks to their own assets and customer assets arising from failures, compromises, or specification changes in those services. They should also establish self-protective measures within their own organizations (limits, suspension conditions, monitoring, additional approvals, etc.)

3. Proposals for the operation of crypto-asset-related businesses

3.1 Comparative analysis of potential countermeasures and existing guidelines

We've compared the attack vectors and possible countermeasures learned from the risk analysis by function and the case study with existing cybersecurity guidelines and rules.

Comparison of measures and guidelines

Possible attack methods by function to crypto-asset-related businesses

Regarding the functions that are considered within the scope of controls that should be taken by the business operator itself or controls that should be included in its outsourcing management, we've summarized possible attack methods by referring to MITRE, etc.

- CEX
- HW wallet
- SW wallet
- MPC / Key Manager
- Physical infrastructure (Cloud, HSM, Software Resources, etc.)

Attack vectors leading to fund outflows identified from the case studies

By analyzing the attack methods used in high-loss incidents since 2023, we've identified several vulnerabilities accordingly.

- Third-party vulnerability
- Contract vulnerability
- Rug pull
- Flash loan attack
- Malware attack

Security measures to be considered

In this chapter, we will examine and list possible security measures against attack techniques and vulnerabilities.

(Major items)
- Measures against the risk of private key leakage
- Detection and prevention of unauthorized transactions
- Anti-phishing / Anti-malware measures
- Access control
- Program management
- Measures against supply chain attack

etc.

Comparison with guidelines, etc.

In this chapter, we will also compare the considered measures with existing guidelines, etc.

(Guidelines, etc.)
- NIST CSF 2.0
- Financial Services Agency Administrative Guidelines
- Financial Services Agency Cyber Security Guidelines
- FISC Security Measures Standard
- MITRE ATT&CK
- MITRE AADAPT
- JVCEA Crypto Asset Security Management Standard
- OWASP SCSVS

etc.

List of guidelines and threat analysis frameworks referenced in this study.

List of guidelines and threat analysis frameworks referenced in this study

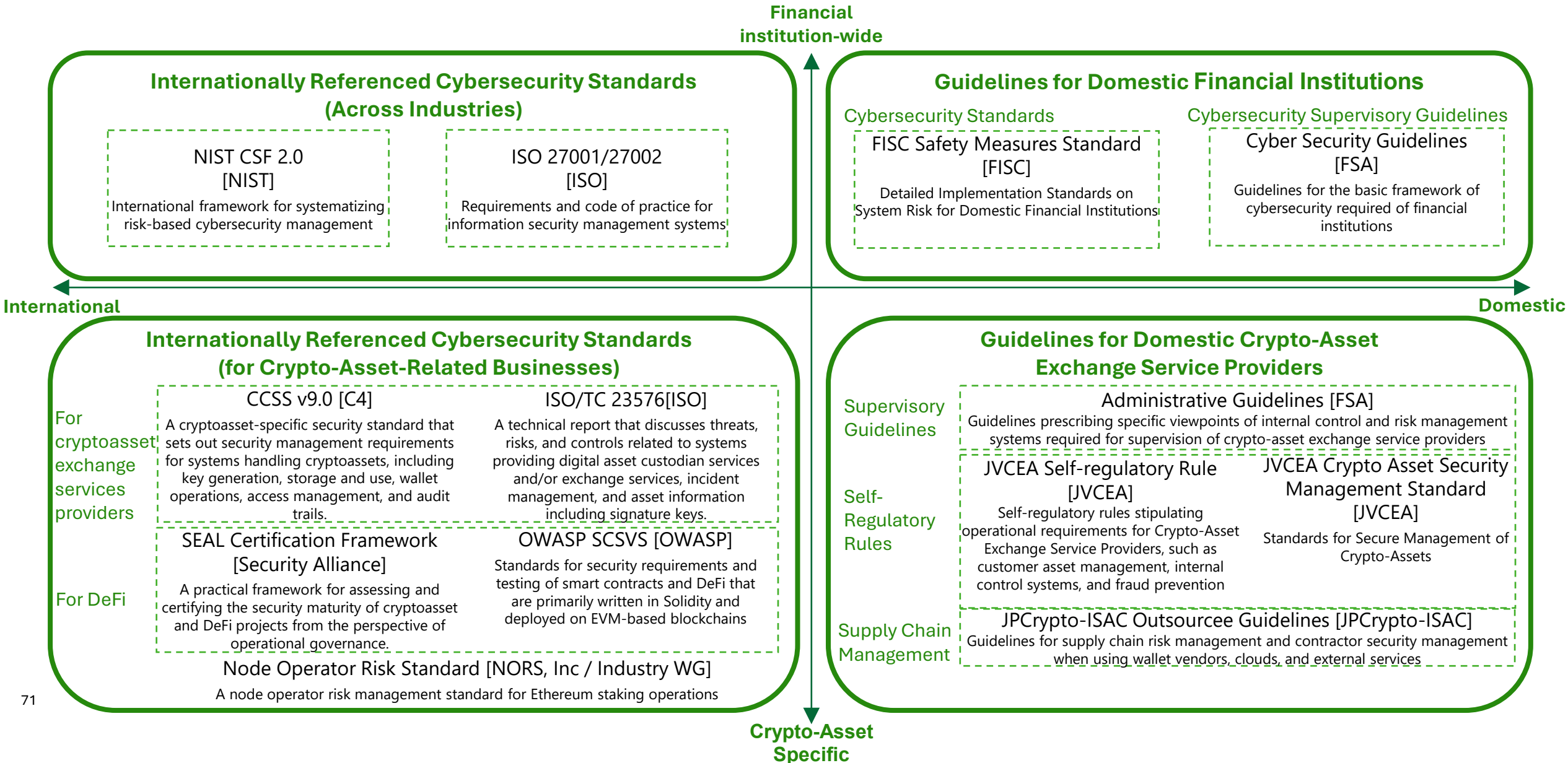
Referenced guidelines/ threat analysis frameworks	Official name	Publication Date/Edition	Publisher	Summary
NIST CSF	Framework for Improving Critical Infrastructure Cybersecurity	2024.2.26 /Version 2.0	National Institute of Standards and Technology (NIST)	International framework for systematizing risk-based cybersecurity management
Financial Services Agency Cyber Security Guidelines	Guidelines on Cybersecurity for the Financial Sector	2024.10.4	Financial Services Agency (FSA)	Guidelines for the basic framework of cybersecurity required of financial institutions
FISC Safety Measures Standard	For computer systems of financial institutions Safety measures standards and manuals	2026.3.25 / Edition 14	The Center for Financial Industry Information Systems(FISC)	Detailed Implementation Standards on System Risk for Domestic Financial Institutions
JVCEA Self-regulatory Rule	Concerning a Crypto-Asset Exchange Service Provider self-regulatory rule	-	Japan Virtual and Crypto assets Exchange Association (JVCEA)	Self-regulatory rules stipulating operational requirements for Crypto-Asset Exchange Service Providers, such as customer asset management, internal control systems, and fraud prevention
JVCEA Crypto Asset Security Management Standard	Crypto asset security management standard	2025.6	Japan Virtual and Crypto assets Exchange Association (JVCEA)	Standards for Secure Management of Crypto-Assets
Financial Services Agency Administrative Guidelines	Third volume: Financial companies 16.Crypto-Asset Exchange Service Providers	2025.4.1	Financial Services Agency (FSA)	Guidelines Prescribing Specific Viewpoints of Internal Control and Risk Management System Required for Supervision of Crypto-Asset Exchange Service Providers
JPCrypto-ISAC Outsourcee Guidelines	Guidelines for Management of Contractors in Crypto-Assets-Related Businesses	2026.1.26	JPCrypto-ISAC	Guidelines for Supply Chain Risk Management and Contractor Security Management when Using Wallet Vendors, Clouds, and External Services
OWASP SCSVS	Smart Contract Security Verification Standard	2024	Open Web Application Security Project(OWASP)	Standards for security requirements and testing of smart contracts and DeFi that are primarily written in Solidity and deployed on EVM-based blockchains

List of guidelines and threat analysis frameworks referenced in this study.

List of guidelines and threat analysis frameworks referenced in this study

Referenced guidelines/ threat analysis frameworks	Official name	Publication Date/Edition	Publisher	Summary
CCSS v9.0	CryptoCurrency Security Standard v9.0	2024.12.17	CryptoCurrency Certification Consortium (C4)	A cryptoasset-specific security standard that sets out security management requirements for systems handling cryptoassets, including key generation, storage and use, wallet operations, access management, and audit trails.
SEAL Certification Framework		-	Security Alliance (SEAL)	A practical framework for assessing and certifying the security maturity of cryptoasset and DeFi projects from the perspective of operational governance.
MITRE ATT&CK	MITRE ATT&CK Framework	2025.10.28 /v18.1	MITRE Corporation	A comprehensive framework that organizes the tactics and techniques of attackers against systems based on real-world observations
MITRE AADAPT	MITRE AADAPT Framework	2025.7.14	MITRE Corporation	A comprehensive framework that systematizes attackers' tactics and techniques in digital asset management systems such as crypto assets
MITRE D3FEND	MITRE D3FEND Framework	2025.12.16 /v1.3.0	MITRE Corporation	Knowledge base that systemizes defense technologies and countermeasures against cyberattacks. Support the design and evaluation of defense methods by associating with MITRE ATT&CK
ISO/TC 23576:2020	Blockchain and distributed ledger technologies — Security management of digital asset custodians	2020.12 / Technical Report	International Organization for Standardization（ISO）	A technical report that discusses threats, risks, and controls related to systems providing digital asset custodian services and/or exchange services, incident management, and asset information including signature keys.
NORS	Node Operator Risk Standard	2024.8 / Certification framework	NORS, Inc. / industry working group	A node operator risk management standard for Ethereum staking operations

List of guidelines referenced in this study (relation map)



We observed from a mapping of possible attack vectors and existing guidelines that the guidelines cover risky areas, but more concrete and in-depth measures and a deeper understanding by business operators will lead to more effective utilization.

Mapping of vulnerabilities learned from case studies and existing guidelines, etc. (1/2)

Vulnerabilities	Case study	Possible security measures	Representative measures listed in the existing guidelines (* 1)
Insufficient anti-phishing measures (* 2)	(1), (3)	■ Education on anti-phishing measures based on case studies ■ Regular reminders and checks	➢ Provide awareness and training to individuals (NIST/PR.AT-02) ➢ Education and training for third-party personnel (the Financial Services Agency/2.3.2. (2)) ➢ Education in Cybersecurity and Social Engineering (FISC/T14)
Insufficient anti-malware measures (* 2)	(1), (3)	■ In addition to signature-based anti-malware software, introduction and monitoring of anti-malware software with behavior detection functions and EDR	➢ Monitor software and data (NIST/DE.CM-09) ➢ Protecting Your System Against Malware Infections (the Financial Services Agency/2.3.4.1. (4)) ➢ Take measures to prevent and detect malicious programs (FISC/J20, J21)
Insufficient measures against unauthorized access (* 2)	(1), (2), (3)	■ Logical and physical separation and segmentation of the network ■ Strict control of access to the production environment	➢ The network is protected from unauthorized access (NIST/PR.IR-01) ➢ Physical separation of networks, segmentation (the Financial Services Agency/2.3.4.4. (1), c) ➢ Restrict connectivity devices accessible from external networks (FISC/J15)
Replacement with unauthorized programs or API tampering	(1), (2), (3)	■ Prevention of production program changes ■ Change management, code scanning, and log monitoring	➢ Execute and monitor secure software development practices (NIST/PR.PS-06) ➢ Separation of development and production environments and prevention of unauthorized changes (the Financial Services Agency/2.3.4.4.b) ➢ Clarification of confirmation and verification procedures in system development and modification (FISC/J75)
Tampering with the signing transaction file	(1), (2), (3)	■ Technical measures to prevent and detect tampering of signing transactions	➢ Preventing Unintended Data Contamination in Transaction Signatures (JVCEA Standard)
Blind signing	(1), (2), (3), (8)	■ Introducing a wallet feature that makes messages more human-readable ■ Providing cyberattack risk training for signers	➢ Confirmation of appropriateness of transfer destination, target assets and quantity (JVCEA Standard) ➢ Verifying the Content of Signed Transaction Data (JVCEA Standard)
Centralized asset management in the same wallet	(1), (2), (3), (4), (5)	■ Detection and restriction of transactions per unit time ■ Decentralization across multiple wallets	➢ Measures to reduce leakage risks such as distribution of wallets (JVCEA Regulation/Article 20) ➢ Setting of the maximum amount of transfer of crypto-assets according to the risk of leakage (JVCEA Regulation/Article 26)
Exploiting the upgrade function of smart contracts to steal authorities	(1), (2), (3)	■ Separation of permissions for each operation and strict management of upgrade permissions	➢ Execute and monitor secure software development practices (NIST/PR.PS-06) ➢ The risk impact of the change is assessed (NIST/ID.RA-07) ➢ Security by Design Implementation (the Financial Services Agency/2.3.4.3. (1)) ➢ System development and modification procedures, including security by design (FISC/J75) ➢ Comprehensive test coverage including unit, integration and security tests (OWASP SCSVS/S3) ➢ Deploying Role Based Access Management (RBAC) (OWASP SCSVS/S4) ➢ Multisig Ops (SEAL Certification Framework)
Insufficient mechanisms to prevent abnormal transactions in contracts	(4), (5)	■ Functional verification that takes into account the vulnerability of the entire system caused by the addition or combination of contract functions, and introduction of checking functions to prevent abnormal transactions	

(*1) Referenced guidelines include: NIST, NIST CSF 2.0; the Financial Services Agency, Guidelines on Cybersecurity for the Financial Sector; FISC, Standard and Manual for Security Measures for Computer Systems of Financial Institutions (Version 13); JVCEA Regulations, Rules for the Management of User Assets Pertaining to Crypto-Asset Exchange Services; JVCEA Standard, Crypto-Asset Security Management Standard; OWASP SCSVS, OWASP Smart Contract Security Verification Standard
(*2) We still listed them as vulnerabilities out of the possibility that they may be insufficient although no information showed that they were not implemented in the incidents.

We observed from a mapping of possible attack vectors and existing guidelines that the guidelines cover risky areas, but more concrete and in-depth measures and a deeper understanding by business operators will lead to more effective utilization.

Mapping of vulnerabilities learned from case studies and existing guidelines, etc. (2/2)

Vulnerabilities	Case study	Possible security measures	Representative measures listed in the existing guidelines (*1)
Excessive privileges granted to external collaborators **Insufficient revocation and review of credentials** **Potential for lateral movement to CI and cloud environments**	(7), (10)	■ Least privilege, JIT access, short-lived credentials, periodic access reviews, and enhanced management of external service providers	➢ Minimization and periodic review of access privileges (NIST / PR.AA-05) ➢ Management of identities and credentials (NIST / PR.AA-01) ➢ CI/CD and cloud configuration change management (NIST / PR.PS-01) ➢ Ongoing assessment and monitoring of third-party risks (NIST / GV.SC-07) ➢ Enhanced access control and authentication (OWASP SCSVS / SCSVS-AUTH) ➢ Secure development, testing, and deployment (OWASP SCSVS / SCSVS-CODE) ➢ CI/CD, cloud, and supply chain controls (SEAL / DevOps & Infrastructure) ➢ Account inventory and credential management (SEAL / Identity & Accounts) ➢ Key management, revocation of access privileges, and least privilege (CCSS / 1.04.1・2・3, 1.05.1・3, 1.06.2)
Insufficient protection of CI/CD credentials **Inadequate access control for cloud APIs and KMS**	(7)	■ Monitoring of workflow changes and mandatory reviews ■ Blocking of signing operations from CI/CD ■ Multi-party approval for KMS policy changes	
Insufficient segregation of duties **Insufficient thresholds** **No timelock configured** **No restrictions on changes**	(7), (8), (10)	■ Enhanced thresholds (e.g. 3 of 5) ■ Timelocks ■ Allowlists for change destinations ■ Limits on change magnitudes	➢ Business logic and economic mechanism security (OWASP SCSVS / SCSVS-GOV) ➢ Multisig, timelocks, and key rotation (OWASP SCWE / SCWE-155) ➢ Multisig operations and threshold management (SEAL / Multisig Ops)
External dependencies **Insufficient reorg resilience**	(8), (9), (10)	■ Verification against multiple sources ■ Delayed settlement ■ Inconsistency detection	➢ Identification and management of external data flows (NIST / ID.AM-03) ➢ Continuous monitoring of external services (NIST / DE.CM-06) ➢ Analysis of inconsistencies and anomalous events (NIST / DE.AE-02, DE.AE-03) ➢ Oracle and external data validation (OWASP SCSVS / SCSVS-ORACLE) ➢ Ensuring the integrity of bridge and state management (OWASP SCSVS / SCSVS-BRIDGE) ➢ Verification of DeFi-specific risks and economic invariants (OWASP SCSVS / SCSVS-DEFI) ➢ Emergency stop and circuit breakers (OWASP SCWE / SCWE-156) ➢ Monitoring, response playbooks, and drills (SEAL / Incident Response) ➢ DeFi risk management and transaction monitoring (SEAL / Treasury Ops) ➢ Treasury operations, DeFi risk management, and transaction monitoring (SEAL / Treasury Ops) ➢ Assessment of digital asset custodian and exchange service systems handling asset information including signature keys (ISO/TR 23576) ➢ Due diligence of Ethereum staking service providers and node operators based on operational security, private key management, and business continuity controls (NORS / Risk & Control Matrix)
Insufficient real-time monitoring and anomaly prevention capabilities	(7), (8), (9), (10)	■ Monitoring of economic invariants ■ Anomalous withdrawal detection ■ Automatic suspension ■ Issuance caps and withdrawal limits	

(*1) Referenced guidelines include: NIST, NIST CSF 2.0; the Financial Services Agency, Guidelines on Cybersecurity for the Financial Sector; FISC, Standard and Manual for Security Measures for Computer Systems of Financial Institutions (Version 13); JVCEA Regulations, Rules for the Management of User Assets Pertaining to Crypto-Asset Exchange Services; JVCEA Standard, Crypto-Asset Security Management Standard; OWASP SCSVS, OWASP Smart Contract Security Verification Standard; ISO/TR 23576, Blockchain and distributed ledger technologies — Security management of digital asset custodians; NORS, Node Operator Risk Standard

3. Proposals for the operation of crypto-asset-related businesses
3.2 Proposed Recommendations for the operation of crypto-asset-related businesses

Crypto-asset-related service providers need to consider and analyze measures to strengthen the management of critical operations that depend on external services.

Proposed recommendations derived from analyses and case studies (1/5)

Enhancement of external service/system provider management

- Crypto-asset-related service providers are required to clarify the interdependencies among functions within their supply chains and to analyze risks that may affect the services they provide. Such risks should not be understood merely as “third-party vulnerabilities,” but should also be analyzed as risks arising from unverified external dependencies, where critical external inputs, interfaces, or functions are trusted without sufficient technical visibility, independent verification, or controls to limit the impact of compromise. In particular, when relying on external service providers for critical operations such as asset transfers and transaction signing, it is necessary to enhance third-party risk management on the premise that a compromise at a service provider may directly lead to the outflow of the firm’s assets. In the cases analyzed in this study, compromises or misconfigurations involving service providers, external collaborators, and external infrastructure have resulted in transaction tampering, takeover of signing authority, unauthorized minting, and failures in cross-chain verification.
- Specifically, it is important to periodically assess the cybersecurity posture of service providers through a layered approach, including phishing and malware countermeasures, access controls for production environments, CI/CD management, cloud IAM/KMS management, program change management, and log monitoring. In addition, service providers should be required to implement least-privilege access controls, just-in-time (JIT) access provisioning, short-lived credentials, periodic access reviews, and immediate revocation of access rights upon project completion, including for subcontractors.
- To this end, organizations should establish governance frameworks that enable them to independently verify the effectiveness of control, including those of subcontractors, through periodic audits and continuous monitoring of service providers. Furthermore, outsourcing agreements should incorporate requirements necessary for such verification, including audit rights, reporting obligations, credential management, remediation measures in the event of violations, log retention, and subcontracting obligations. Appropriate due diligence should also be conducted on the structure of the services and systems provided by service providers. Where assessment results are deemed insufficient, decisions should be made to implement additional defensive measures, restrict usage, consider alternative providers, or review existing outsourcing arrangements.
- Crypto-asset-related service providers should take into account the risk that attackers may obtain unauthorized access privileges to critical systems through IT and systems personnel within their own organizations or at external service providers. They should therefore ensure that risk management for external services addresses not only system and technical controls, but also personnel security controls. In particular, they should refer to CCSS when reviewing whether appropriate controls are in place for personnel of external service providers, subcontractors, and other external parties. Such controls may include identity verification, legally permissible background checks, security training, and periodic access privilege reviews.

Note: The recommendations provided here are general reference information. Addressing only the measures described herein is not sufficient, additional considerations and controls tailored to each company’s circumstances and risk profile are necessary.

For critical operations such as crypto-asset withdrawals and transaction signing, it is necessary to enhance control measures that explicitly code tampering stemming from unauthorized access.

Proposed recommendations derived from analyses and case studies (2/5)

Measures to prevent the malicious code injection or program tampering

- Case studies demonstrate that malicious code injection and tampering have often resulted in the compromise of the UI, APIs, CI/CD pipelines, cloud environments, and unsigned transaction generation logic leading up to signing (e.g., wallet UI JavaScript tampering in Bybit, API logic tampering in SwissBorg, malware-based transaction tampering in Radiant, and credential theft through abuse of GitHub workflows in Resolv). In these cases, the attacks abused legitimate signing processes.
- Concrete countermeasures include strict management of production credentials, rigorous CI/CD review processes, development and maintenance of software bills of materials (SBOMs), integrity verification of source code and files, validation of API responses, and operational monitoring frameworks capable of promptly detecting and responding to tampering or anomalous behavior. For privileged operations involving signing processes, KMS, CI/CD, cloud IAM, and production environments, crypto-asset service providers should implement network segmentation, segregation of duties, hardware-backed credentials, short-lived authentication, just-in-time privilege elevation with automatic expiry, elimination of standing administrator access, and multi-party approval for high-risk changes such as KMS policy modifications. In addition, crypto-asset-related service providers should implement controls to prevent direct access from CI/CD environments or developer endpoints to high-risk operations, including signing processes, KMS operations, and privileged actions in production environments (e.g., network segmentation, segregation of privileges, short-lived authentication, and multi-party approval for KMS policy changes). Furthermore, when incorporating external libraries, it may be beneficial to establish clear requirements, such as version pinning and prohibiting automatic execution, in order to prevent contamination of internal systems through the introduction of compromised or malicious libraries.
- For wallet operation terminals, it is also necessary to reduce the risk of pre-signing tampering by combining multiple security measures (e.g., Endpoint Detection and Response (EDR), behavioral detection, anti-malware controls, dedicated devices, restrictions on the execution of external files, and network traffic monitoring). Separately, build and deployment environments for wallet, signing, CI/CD, smart-contract, and cloud-control components should be isolated, auditable, hermetic, and reproducible, with integrity checks and change approvals applied before production deployment.

Note: The recommendations provided here are general reference information. Addressing only the measures described herein is not sufficient, additional considerations and controls tailored to each company’s circumstances and risk profile are necessary.

To prevent unauthorized transfers of crypto-assets, in addition to multi-layered controls, rigorous verification at the time of signing is indispensable.

Proposed recommendations derived from analyses and case studies (3/5)

Measures to prevent unauthorized transfers of crypto-assets

- To prevent unauthorized transfers of crypto-assets, crypto-asset service providers should recognize that it is not sufficient to introduce multi-signature arrangements or hardware wallets. To ensure that signing activities are consistently conducted in accordance with the purpose of the transaction and approved policies, it is important to implement multi-layered controls over transfer transactions as part of their operational requirements.
- Case studies (Bybit, SwissBorg, and Radiant) demonstrate that unauthorized privilege changes and fund outflows were caused by signers approving transactions that appeared legitimate on the screen. For transfer transactions—particularly complex smart contract transactions related to DeFi or wallet operations, including smart contract upgrades—it is desirable to operate a combination of multiple controls (e.g., additional approvals based on factors such as transaction amount, frequency, new destination addresses, unknown contracts, privilege changes, and large withdrawals; delayed execution; offline signing; transaction simulation; technical review of decoded transaction data; and improved readability through Clear Signing or similar mechanisms.) Where transaction data or approval screens are generated through external APIs or wallet infrastructure, they should also consider pre-signing attestation or independent verification mechanisms that enable cryptographic verification of transaction content and policy compliance before signing.
- In addition, they should design systems to minimize damage even if certain controls are bypassed (e.g., through distributed wallet management, per-address asset holding limits, withdrawal limits, issuance limits, abnormal withdrawal detection, economic invariant monitoring, and automatic suspension and circuit breakers). When verifying transaction details before signing, they should consider not only checks and balances to prevent fraud or operational errors, but also methods for verifying transaction contents and the qualifications and backgrounds of the personnel responsible for verification. Those personnel should receive adequate training.
- Furthermore, they should assume the risk of large-scale fund outflows occurring at once and consider measures to avoid concentration of balances, such as distributed wallet management and per-address asset holding limits.
- It is also useful for preventing unauthorized fund outflows to collect, analyze, share, and review fund outflow incidents, and to identify new attack methods and trends at an early stage by observing patterns in smaller-scale incidents and reflect them in controls, both internally and through industry association.

Note: The recommendations provided here are general reference information. Addressing only the measures described herein is not sufficient, additional considerations and controls tailored to each company's circumstances and risk profile are necessary.

Crypto-asset-related service providers need to consider and analyze measures to prepare for the expanding use of smart contracts and DeFi.

Proposed recommendations derived from analyses and case studies (4/5)

Smart Contracts and DeFi

- For smart contracts and DeFi, it is not sufficient to verify only vulnerabilities in the code itself. The overall risk profile of a protocol may change depending on post-deployment operations (e.g. administrator privileges, upgrade privileges, mint privileges, oracle settings, risk parameters, withdrawal limits, and cross-chain settings). Therefore, as with the operations implemented by centralized crypto-asset service providers for signing key management and ensuring transaction authenticity, operational controls are also important for smart contracts and DeFi (e.g. authority management, change management, monitoring, and emergency suspension).
- In Balancer v2 and Euler, vulnerabilities arising from smart contract logic and the addition or combination of functions were exploited. In contrast, in Drift, Litecoin/Near Intents, and Kelp/LayerZero, losses resulted from operational and configuration deficiencies (e.g. administrative privileges, oracles, blockchain reorg resilience, dependence on single points of failure, minting privileges, and cross-chain verification). These cases demonstrate the need for controls premised on real-world operations (e.g. code reviews, smart contract audits, authority design, change management, monitoring, emergency suspension, and management of external dependencies).
- Specifically, in addition to technical verification related to the design, implementation, and changes of smart contracts, controls over the exercise of administrative privileges should be established. For high-risk operations, it is desirable to combine controls such as segregation of duties, multi-party approval, timelocks, allowlists for change targets, limits on the magnitude of changes, initial caps, phased increases, and automated suspension (e.g. minting, burning, locking/unlocking, adding collateral, oracle changes, risk parameter changes, withdrawal limit changes, admin transfers, and contract upgrades).
- In addition, in DeFi and cross-chain domains, dependencies on external parties and components may become risk factors (e.g. external oracles, RPCs, DVNs, bridges, and other protocols), and risk management should therefore cover these external dependencies. Such measures may include multiple oracles, minimum liquidity, holder distribution, and trading-period requirements, automated rejection of abnormal prices, multi-source verification, confirmation requirements based on reorg risk, delayed settlement, swap limits, redundancy in cross-chain verification and external data retrieval paths, avoidance of reliance on a single verification entity or data source, and invariant monitoring of issuance volume and collateral value.
- Accordingly, in reducing risks related to smart contracts and DeFi, it is essential to control the full range of operations (e.g. verification of code vulnerabilities through audits, design of administrative privileges and signing privileges, change approvals, parameter changes, external dependencies, monitoring and detection, and emergency suspension). Even when smart contract audits have been conducted, residual risks may remain in areas not covered by the audit (e.g. functions outside the audit scope, subsequent upgrades, external dependencies, operational privileges, and cross-chain configurations). Therefore, it is necessary not to rely solely on audit results, but to combine continuous operational management with complementary controls.

Note: The recommendations provided here are general reference information. Addressing only the measures described herein is not sufficient, additional considerations and controls tailored to each company's circumstances and risk profile are necessary.

Although external assessments are effective, it is important to verify their coverage and limitations and to combine them with supplementary checks and compensating controls aligned with the firm's specific risk exposure.

Proposed recommendations derived from analyses and case studies (5/5)

Leveraging third-party assessments

- Crypto-asset service providers should leverage third-party assessments for external service providers, wallet services, staking services, DeFi protocols, bridges, smart contracts, cloud, CI/CD, and signing services. For infrastructure providers supporting critical operations, they should use such assessments to confirm baseline assurance (e.g., SOC 2 or ISO/IEO 27001 attestation, vulnerability assessments, penetration tests, evidence of completed code audits and, where applicable, smart contract audits, security certifications, and bug bounty results). However, third-party assessments are not a panacea and should be used only after confirming the assessment scope, timing, methodology, assumptions, exclusions, and the independence and qualifications of the third-party assessors.
- The case studies indicate that, even when smart contract audits or third-party assurance are in place, related controls and processes may not have been sufficiently evaluated (e.g. components outside the audit scope, functions added later, integrations with external services, external data retrieval paths, cross-chain verification mechanisms, development and release processes, cloud-based privilege management, key management, and signing-related services). Therefore, when reviewing third-party assessment results, they should understand not only "what has been assessed," but also "what has not been assessed," and address any gaps based on their own outflow risk scenarios.
- In addition, the mere fact that a third-party assessment has been conducted should not be treated as evidence of security. It is desirable to confirm the findings identified in the assessment results, the status of remediation, whether any follow-up assessment has been conducted, post-audit change management, and the actual operational practices of external service providers. As necessary, they should combine audits of external service providers, additional questionnaires, review of technical evidence, on-site reviews, continuous monitoring, and complementary controls implemented by themselves.

Note: The recommendations provided here are general reference information. Addressing only the measures described herein is not sufficient, additional considerations and controls tailored to each company's circumstances and risk profile are necessary.

3. Proposals for the operation of crypto-asset-related businesses
3.3 Other considerations

To address crypto-asset risks characterized by anonymity, cross-border nature, and immediacy, it is necessary to enhance industry-wide and internationally coordinated information-sharing and analysis functions centered on information-sharing organizations

Recommendation: Establishment of an internationally coordinated analytical and supervisory framework

Challenges

- Crypto-asset transactions are characterized by anonymity, cross-border nature, and immediacy, and attack methods exploiting these characteristics are becoming increasingly sophisticated.
- Therefore, it is becoming increasingly important for crypto-asset service providers to share information across operators (e.g. attack information, vulnerability information, illicit wallets, asset-freezing requests, and incident response knowledge).
- Given this risk structure, internal controls by individual crypto-asset service providers alone have limitations in real-time detection, tracing, asset freezing, and prevention of further damage.

Recommendations

■ **Recommendations on the concept**

To fundamentally address the structural issues above, it is desirable that initiatives such as the following advance internationally.

1. **Establishment of common data standards and formats**
   Develop standards and formats that can be shared internationally for relevant data items (e.g. wallet information, transaction information, risk attributes, asset-freezing requests, attack patterns, and vulnerability information).

2. **Enhancement of information-sharing and analysis functions centered on information-sharing organizations, such as JPCrypto-ISAC**
   ➢These organizations should aggregate and analyze threat information and incident information, and be responsible for issuing alerts to members, sharing examples of responses, conducting exercises and training, and accumulating practical knowledge on external service provider management.
   ➢They should also play a role in international information sharing and analysis through coordination with overseas ISACs and other relevant organizations, while facilitating information sharing among domestic crypto-asset service providers.

3. **Implementation of joint supervision and joint response**
   ➢Continuously collect and analyze attack patterns, including small-scale cases and attempted incidents, and reflect the insights obtained in practical measures and guidance (e.g. guidelines, external service provider management, signing and withdrawal controls, and training scenarios).

■ **Recommendations for implementation**

As various industry-led initiatives advance, it is desirable to strengthen a framework that goes beyond responses by individual crypto-asset service providers and enables information-sharing organizations, such as JPCrypto-ISAC, to aggregate and analyze relevant information and share it rapidly among operators (e.g. threat information, vulnerability information, illicit wallet information, and incident response knowledge). These organizations should advance the establishment of common data items and sharing formats, coordination with overseas ISACs, and the accumulation of practical knowledge on exercises, training, and external service provider management. It is also important to establish a mechanism through which crypto-asset service providers continuously reflect the insights obtained in practical measures and guidance (e.g. guidelines, signing and withdrawal controls, external service provider management, and training scenarios).

In light of advances in AI, urgent measures are required that integrate technology and operations, including reducing attack surfaces, isolating critical systems, providing education and training, and redesigning vulnerability disclosure frameworks

Recommendations: AI-Era Cybersecurity Measures for Crypto-Asset Service Providers

Challenges

- Anthropic announced that its frontier AI model, Claude Mythos, discovered high-severity vulnerabilities in critical software, including major operating systems and browsers. Malicious uses of AI have also been observed, (e.g., the mass deployment of crypto-asset scams and improved impersonation quality in identity verification and recruitment processes). The acceleration of vulnerability discovery and exploitation through AI is becoming a realistic cyber risk for crypto-asset service providers.
- While the discovery of vulnerabilities, including zero-day vulnerabilities, is becoming faster and more scalable through the use of AI, cooperative vulnerability disclosure may become more difficult if these providers' responses to identified vulnerabilities, as well as the design of reward and recognition systems for vulnerability researchers, fail to keep pace. If such frameworks are insufficient, this may lead to significant problems (e.g. vulnerability reports being left unaddressed, insufficient explanations or rewards for researchers, and a lack of transparency in the response process). As a result, researchers may disengage from cooperative disclosure processes and shift to non-cooperative actions (e.g. public disclosure, sale, or retaliatory use of vulnerability information).
- Attackers are also combining weaknesses in business operations with technical vulnerabilities to carry out attacks. In this industry, many processes rely heavily on human judgment and approval (e.g. signing key management, withdrawal approval, multisig operations, CI/CD, and administrator privilege management). If controls over these areas are insufficient, attackers can cause asset outflows or steal critical information through operational weaknesses (e.g. privilege concentration, circumvention of approval flows, abuse of exception handling, and unauthorized changes to development/operational environments).

In light of advances in AI, urgent measures are required that integrate technology and operations, including reducing attack surfaces, isolating critical systems, providing education and training, and redesigning vulnerability disclosure frameworks

Recommendations: AI-Era Cybersecurity Measures for Crypto-Asset Service Providers

Recommendations

Crypto-asset service providers need to establish technical measures and governance controls in an integrated manner, with a view to minimizing external attack risks and preventing attackers from abusing human judgment and approval processes. Specifically, the following measures may be considered.

- **Reduction of attack surfaces:** They should continuously identify externally reachable attack surfaces, and promptly detect and remediate unnecessary public services, misconfigurations, and excessive access privileges.
- **Restrict Internet Exposure of Critical Systems:** Internet connectivity to critical systems directly linked to the protection of user assets (e.g. wallets, private keys, signing infrastructure, authentication and privilege management, CI/CD, and monitoring and log management) should be limited to the minimum necessary outbound communications, with inbound Internet access prohibited.
- **Security management of development environments:** Given that development and CI/CD environments may be used as attack paths to production systems, they should be treated as security management targets as critical as production systems (e.g. privilege management, change approval, log monitoring, and supply chain risk management).
- **Isolation of critical systems:** Production, development/deployment, asset management, and administrative systems should be functionally separated according to their roles and risk characteristics, so that any compromise does not unnecessarily expand its scope of impact.
- **Management of audit trails:** They might use Infrastructure as Code (IaC) to the extent possible for relevant components (e.g. system infrastructure, networks, security settings, and privilege policies), with a view to reducing manual configuration errors and configuration drift; establish a framework for continuously understanding and verifying configurations, settings, and change histories; and review and approve IaC code itself, inspect it for misconfigurations, and manage changes, while separately managing secret information and privileged access granted to individual users (e.g. private keys and API keys).
- **Rapid and safe change management:** They should establish processes and technical mechanisms for rapid, controlled emergency changes within compressed timeframes, while maintaining segregation of duties, auditability, and production safety (e.g., changes to wallets, signing infrastructure, CI/CD, cloud IAM/KMS, bridges, RPC/DVN configurations, and smart-contract parameters).
- **Training and education:** They should provide continuous education and training for their executives and employees, and clarify specific verification procedures and reporting channels for suspicious requests (e.g. impersonation using deepfakes, urgent requests for granting privileges, exceptional withdrawal approval requests, and suspicious change requests related to CI/CD or administrator privileges).
- **Redesign of vulnerability disclosure frameworks:** They might need to consider redesigning the Responsible Disclosure framework (e.g. report quality, verification scope, disclosure deadlines, and coordination with law enforcement and regulatory authorities).

Recommendations: Future-Proofing and Cryptographic Agility

Recommendations

Crypto-asset service providers should prepare for the transition to post-quantum cryptography. In particular, **cryptographic agility**—the ability to migrate cryptographic algorithms and key-management mechanisms without significant re-engineering of underlying systems—should be regarded as an important security requirement, and they should develop medium- to long-term migration plans accordingly.

Appendix1. Preliminary research

In analyzing cybersecurity of crypto-asset-related business, the academics often take an approach of using a layered system model to locate risks.

Preliminary research - Academic papers (1/2)

<table>
<tr><th></th><th>Title</th><th>Author</th><th>Publication information</th><th>Approach to analyze cybersecurity</th><th>Insights: differences between autonomous distributed systems and traditional systems</th></tr>
<tr><td>1</td><td>A Survey on the Security of Blockchain Systems</td><td>Xiaoqi Li et al.</td><td>At Future Generation Computer Systems 107 in August 2017</td><td>This paper analyzed security issues in blockchain systems, and as shown in the table below, it categorized threats, attacks, and countermeasures into a ‘Taxonomy-of-blockchain’s-risks’, based on layered system structure.
<table><tr><th>Risk</th><th>Cause</th></tr><tr><td>51% vulnerability</td><td>Consensus mechanism</td></tr><tr><td>Private key security</td><td>Public-key encryption scheme</td></tr><tr><td>Criminal activity</td><td>Cryptocurrency application</td></tr><tr><td>Double spending</td><td>Transaction verification mechanism</td></tr><tr><td>Transaction privacy leakage</td><td>Transaction design flaw</td></tr><tr><td>Criminal smart contracts</td><td>Smart contract application</td></tr><tr><td>Vulerabilities in small contracts</td><td>Program design flaw</td></tr><tr><td>Under-optimized smart contracts</td><td>Program writing flaw</td></tr><tr><td>Under-priced operations</td><td>EVM design flaw</td></tr></table></td><td>Comparing to traditional system structure consisted of OS, network and applications, this paper expanded it to include the technical features of autonomous distributed systems, such as ‘Consensus mechanism’ and ‘Smart contract’.</td></tr>
<tr><td>2</td><td>SoK: Decentralized Finance (DeFi) Attacks</td><td>Liyi Zhou et al.</td><td>At 2023 IEEE Symposium on Security and Privacy</td><td>This paper analyzed DeFi attacks and mapped them to a layered system model shown in the upper figure.
In addition, when analyzing Adversarial Capabilities, it introduced a parameter by classifying the knowledge (access to information) of an actor into three groups: Public/Sequencer/Insider.</td><td>The same with #1 above, the layered model used in this paper also expanded to include ‘Consensus mechanism’ and ‘Smart contract’.

Furthermore, when analyzing DeFi attacks, it not only included traditional way of hacking, but also market manipulation and front-running (the attacks to the ‘Pro’ and ‘AUX’ layers in the upper figure), in which the code is correct, but the weak points in market mechanisms or operations were taken advantage of.

This paper also provided an insight that, in autonomous distributed systems, access to information and authority management have changed from the past, and it is necessary to consider such elements.</td></tr>
</table>

In analyzing cybersecurity of crypto-asset-related business, the academics often take an approach of using a layered system model to locate risks.

Preliminary research - Academic papers (2/2)

	Title	Author	Publication information	Approach to analyze cybersecurity	Insights: differences between autonomous distributed systems and traditional systems
3	SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) Protocols	Jiahua Xu et al.	ACM Computing Surveys/Vol. 55, in 2023	This paper studied only the AMM (Automated Market Maker) type decentralized exchanges. When analyzing the risk factors, it mapped categorized security attacks to a simplified system model with three layers: Infrastructure layer/ Middleware layer/ Application layer.	Similar to the papers in previous page, the layered model used in this paper also expanded to include the technical features of autonomous distributed systems, such as **'Smart contract'.** Furthermore, in the analysis of risk factors, **the risks derived from the transparency and traceability of blockchain technology are grouped as 'privacy concern',** and it argued that it is a cause of security attacks.
4	A Survey on Ethereum Systems Security: Vulnerabilities, Attacks and Defenses	Huashan Chen et al.	ACM Computing Surveys/Vol. 53, in 2020	This paper studied the Ethereum systems security, and analyzed the vulnerabilities in security by mapping them to the 'layered system model + environmental factors'. In the grouping and analysis of vulnerabilities, causes are identified in very much details.	Similar to the above papers, the layered model used in this paper also expanded to reflect the technical features of autonomous distributed systems, it defined **'Consensus'** as a layer, and incorporated elements such as **'Smart contract'.** This paper also provided a viewpoint of mapping 'environmental factors', which are applicable to both autonomous distributed systems and traditional systems, separately from the layered system model. In response to the technical features of autonomous distributed systems, this paper identified **new risk areas such as 'Smart contract programming'and 'Design and implementation (of consensus layer)', in which there may be exploits taking advantage of those flaws in programming or design.**

Public/international organizations or authorities analyze cybersecurity of crypto-asset-related business from not only technology, but also governance, operations and markets, and regulations perspectives.

Preliminary research - Public reports, etc. (1/3)

<table>
<tr><th></th><th>Title</th><th>Author</th><th>Publication information</th><th>Approach to analyze cybersecurity</th><th>Insights: differences between autonomous distributed systems and traditional systems</th></tr>
<tr><td>1</td><td>NISTIR 8202: Blockchain Technology Overview</td><td>NIST</td><td>October 2018</td><td>This public report from NIST systematically describes blockchain technology.

It summarized the nature of blockchain technology as follows.
1) Implemented in a distributed fashion (= Without central authority, run by consensus mechanisms)
2) Tamper evident and tamper resistant

On the other hand, this report pointed out that the use of blockchain technology is not a silver bullet, and there are issues, usually beyond technology itself, that must be considered. such as:
■ The concept of immutability for blockchain ledgers can be violated;
■ How to deal with malicious users;
■ How controls are applied;
■ Limitations of the implementations;
■ Compliance issues arising from data visibility;
■ Other operational and governance issues.</td><td>Blockchain technology has decisively different features from traditional systems, such as
consensus mechanisms, decentralized authority, immutability and data visibility.

On the other hand, there are also operational and governance limitations in the use of blockchain technology, and in some cases, security issues can arise.</td></tr>
<tr><td>2</td><td>Crypto Currency Security Standard (CCSS)</td><td>Crypto Currency Certification Consortium (C4)</td><td>Version 9.0
December 2024</td><td>CCSS is a crypto-asset specific standard for practices, published by C4.
The security requirements in CCSS are specified in categories and aspects as shown in the table.
<table><tr><th>Category</th><th>Aspect</th></tr><tr><td>1. Cryptographic Asset Management</td><td>1.01 Key Material Generation</td></tr><tr><td></td><td>1.02 Wallet Generation</td></tr><tr><td></td><td>1.03 Key Material Storage</td></tr><tr><td></td><td>1.04 Key Material Access</td></tr><tr><td></td><td>1.05 Key Material Usage</td></tr><tr><td></td><td>1.06 Data Sanitization Documentation</td></tr><tr><td>2. Operations</td><td>2.01 Security Tests/ Audits</td></tr><tr><td></td><td>2.02 Log and Monitor</td></tr><tr><td></td><td>2.03 Governance and Risk</td></tr><tr><td></td><td>2.04 Key Compromise Documentation</td></tr></table></td><td>Instead of the traditional focus on server defense, CCSS specifies controls over key management, access management and wallet operations as the core of its security requirements, in response to the ‘irreversible’ nature of blockchain technology.</td></tr>
</table>

Public/international organizations or authorities analyze cybersecurity of crypto-asset-related business from not only technology, but also governance, operations and markets, and regulations perspectives.

Preliminary research - Public reports, etc. (2/3)

	Title	Author	Publication information	Approach to analyze cybersecurity	Insights: differences between autonomous distributed systems and traditional systems
3	Views on Security Measures for Crypto-Asset Custodians - Fifth Edition -	Cryptoassets Governance Task Force (CGTF)	April 2024	This is a public report from CGTF, a group of security experts and cryptoasset exchangers established for the purpose of developing security standards for risk management to protect users and consumers, prior to consideration by formal financial regulators or self-regulatory organizations. This report outlined a model of **crypto-asset custody systems and operation flows**, and used this model as the basis for analysis throughout the report. This report summarized the **features of crypto-assets in blockchain and distributed ledgers** as follows. ■ **Importance of signing keys**: More attention should be paid to the theft, unauthorized use and loss of signing keys in crypto-assets because of the irreversible nature of the technology. ■ **Diversed implementation**: Due to the high diversity of implementation for each crypto-asset, there are cases in which security measures that are effective for a particular type of crypto-asset cannot be implemented with another type of crypto-asset. ■ **Potential blockchain forks**: There are a wide variety of forks and splits on blockchain so it can be difficult to cope with all of them, and countermeasures need to be considered depending on the risk. ■ **Risk in authorization of transactions**: Sending a transaction only directs the application for a transfer and does not mean that the asset has been immediately transferred. A transaction requires approval to complete so there are risks in this process. **In the transfer of crypto–assets, the signing key plays a key role therefore, has high risks.** **T**his report analyzed such risks by **mapping the threats to the factors associated with signing key**. In addition to signing keys, this report also analyzed risks related to asset data, internet infrastructure, Web PKI, device environment, blockchain, business cooperation with external parties, etc., and presented the basic concept of security management for various risks **in accordance with ISO requirements for cyber security management systems ISO/IEC 27001:2013 (JIS Q 27001: 2014) and the code for practice ISO/IEC 27002:2013 (JIS Q 27002: 2014)*.**	Taking into account the features of crypto-assets, **comprehensive approach from the perspective of both technology and operations** would make it easier to identify or locate the risks. Autonomous distributed systems has decisively different features from traditional systems, such as **the key role of signing keys, a distributed way of ledger implementation, blockchain forks and splits, and consensus mechanisms.** **In the transfer of crypto–assets, the signing key plays a key role and has high risks**, therefore, key management tends to be the core of analyzing and considering countermeasures against security threats. Based on the differences between autonomous distributed systems and traditional systems,when analyzing security risks, **mapping to existing standards or guidelines** is a useful way to identify risks and consider countermeasures.

*The standards listed above are those referenced at the time. Current versions: ISO/IEC 27001:2022 + Amd 1:2024 (JIS Q 27001:2025); ISO/IEC 27002:2022 (JIS Q 27002:2024).

Public/international organizations or authorities analyze cybersecurity of crypto-asset-related business from not only technology, but also governance, operations and markets, and regulations perspectives.

Preliminary research - Public reports, etc. (3/3)

<table>
<tr><th></th><th>Title</th><th>Author</th><th>Publication information</th><th>Approach to analyze cybersecurity</th><th>Insights: differences between autonomous distributed systems and traditional systems</th></tr>
<tr><td>4</td><td>The crypto ecosystem key elements and risks</td><td>BIS</td><td>July 2023</td><td>This report broke down to the following elements to analyze risks associated with crypto-assets.
Risks within the crypto ecosystem
Risks in DeFi
Risks interconnected with the traditional financial system
Risks to emerging market and developing economies</td><td>This report analyzed that rather than risks derived from technology itself, other elements such as ecosystem structure, interconnections and operations, and markets can amplify risks.</td></tr>
<tr><td>5</td><td>Quarterly Review - DeFi risks and the decentralisation illusion</td><td>BIS</td><td>December 2021</td><td>This report compared DeFi with traditional financial operations by mapping financial functions and services, as shown in the table below. And based on the analysis it pointed out that, there is a “decentralisation illusion” in DeFi since the need for governance makes some level of centralisation inevitable.
Crypto vs traditional financial system<table><tr><th rowspan="2">Function</th><th rowspan="2">Service</th><th colspan="2">Crypto financial system</th><th rowspan="2">Traditional finance</th></tr><tr><th>Decentralised finance (DeFi)</th><th>Centralised finance (CeFi)</th></tr><tr><td rowspan="3">Trading</td><td>Funds transfer</td><td>DeFi stablecoins (DAI)</td><td>CeFi stablecoins (USDT, USDC)</td><td>Traditional payment platforms</td></tr><tr><td>Asset trading</td><td>Crypto asset DEX (Uniswap)</td><td rowspan="2">Crypto CEX (Binance, Coinbase)</td><td rowspan="2">Exchanges and OTC brokers</td></tr><tr><td>Derivatives trading</td><td>Crypto derivatives DEX (Synthetix, dYdX)</td></tr><tr><td rowspan="2">Lending</td><td>Secured lending</td><td>Crypto decentralised lending platforms (Aave, Compound)</td><td>Crypto centralised lending platforms (BlockFi, Celsius)</td><td>Broker-dealers active in repo and securities lending</td></tr><tr><td>Unsecured lending</td><td>Crypto credit delegation (Aave)</td><td>Crypto banks (Silvergate)</td><td>Commercial banks and non-bank lenders</td></tr><tr><td>Investing</td><td>Investment vehicles</td><td>Crypto decentralised portfolios (yearn, Convex)</td><td>Crypto funds (Grayscale, Galaxy)</td><td>Investment funds</td></tr></table>CEX = centralised exchanges; DEX = decentralised exchanges; OTC = over-the-counter; USDC = USD Coin; USDT = Tether.</td><td>Autonomous distributed system technology is implemented in a decentralized manner, but
centralised governance is needed to take strategic and operational decisions in each player (including wallet operators, exchanges, and affiliated traditional financial institutions), especially in terms of risk management and compliance in order to properly provide financial services.

It also should be noted that consensus mechanism itself, which is an important feature in technology of autonomous distributed systems, tends to favour a concentration of decision power.</td></tr>
</table>

Appendix2. Risk analysis by function in the Crypto-Asset Ecosystem

To visualize structural vulnerabilities specific to crypto-asset-related businesses, this cyber risk assessment was conducted incorporating multifaceted risk factors.

Cyber risk analysis for crypto-asset-related functions

Objectives

- The objective of this risk analysis is to visualize, by function (as defined in Section 1.3), "where is structurally easy to be targeted" in the crypto ecosystem.
- Comparing to the conventional scoring system of “Impact x Frequency“, this risk analysis emphasizes on differences across functions in risk factors (qualitative differences).

Assumptions

- The framework adopted for this analysis identifies risk concentration areas across the ecosystem, it does not rank the safety nor indicate the availability of specific services.
- This analysis covers only the typical service model for each function and does not assess or comprehensively guarantee the safety of individual operators with diverse business models.
- Even within the same function, actual risk may differ depending on whether the operator is subject to regulation and on differences in operating and control frameworks.
- Risk levels are relative and may change over time as technologies evolve or new threats emerge.

Approach

Risk factors were selected from the following perspectives.

① **Attacker's view**	Motivation to attack (Concentration of assets) Ease of attack (Attack exposure)
② **System structure**	Impact spread (Domino effect) Concentration of authorities / Governance
③ **Defense environment**	External controls (Lack of regulations, etc.) Vulnerabilities in operations (Human errors / Operational risk)

It is also composed with reference to the following public / international frameworks:

- NIST SP 800-30 (Risk Assessment)
- NIST SP 800-53 (Security Controls Catalog)
- NIST CSF 2.0 (6 functional models, etc.)
- BIS/FSB (DeFi Interoperability Risk)
- ENISA/Verizon DBIR (Human Factors)

Risk factors were independently selected based on existing international standards, taking into account the characteristics of distributed systems, such as irreversibility, composability, etc.

Defined risk factors

① **Concentration of assets**	How attractive is the target to the attacker?
② **Domino effect**	How far will the impact spread if there is a failure in this function?
③ **Attack exposure**	How wide are the contacts that can be penetrated from the outside?
④ **Lack of regulations, etc.**	Are independent audits, regulations and third-party verification functioning?
⑤ **Human errors / Operational risk**	Can non-technical factors cause failure?

*Each factor was assessed in three degrees as “High/Medium/Low" without scoring
*Based on the “weakest-link” principle: overall security is determined by the weakest component.

Expected utilization

- For regulators/auditers: help to identify key areas or form viewpoints for verification
- For business operators: help the overall risk assessment of the business and sorting priorities
- For outsourcing management: help to identify critical external service providers

The risk analysis of the crypto ecosystem actors is conducted by defining five risk factors, and qualitatively evaluating each factor in three degrees: high, medium, and low.

Definition and level of risk factors

① Concentration of assets

The importance of certain assets in the system, and/or the degree of concentration of funds or data to which makes it a prime target for attack

Level	Description
High	Concentration of vast customer assets and/or critical data in a single location or contract
Medium	A state in which certain assets are held but in a decentralized manner, and losses in the event of a single failure are limited
Low	A state in which customer assets are not directly held, and direct losses in the event of a single failure are small

② Domino effect

The degree of dependencies on external systems or other protocols, and the risk of failures in effected area resulting in failures cascading across the ecosystem

Level	Description
High	In a fundamental position of the ecosystem where failures can have catastrophic effects on others
Medium	While having linkage to several external systems, the impact of failures is limited to specific services and users
Low	High degree of independency in operations thus the impact of failures to other systems is small

③ Attack exposure

The number of accessible points from outside(such as endpoints) for attackers to break into or tamper with the system

Level	Description
High	Extremely large number of accessible points of communication such as contracts, APIs, etc.
Medium	Accessible points exist but each only has single function and can limit the access from attackers
Low	Minimal connect with external network or physical isolation making attacks difficult

④ Lack of regulations, etc.

Whether the entity or system is subject to regulatory supervision, independent audits by third parties, and/or ongoing external verification and control

Level	Description
High	A state in which there is little external control and is not subject to regulatory supervision or third-party audits
Medium	A state in which there is limited external control that a certain degree of regulatory compliance and third-party review is in place
Low	A state in which the entity or system is under regulatory supervision based on laws and regulations, and is subject to regular third-party audits

⑤ Human errors / Operational risk

Operational risk caused by psychological traps (phishing, etc.) targeting who operate the system, inadequacies in operating procedures, misconfiguration, etc.

Level	Description
High	High dependency on users' IT literacy, resulting in significant losses from operational errors or phishing
Medium	While being operated by experts, there is still possibility of being targeted because of complex misconfigurations or through social engineering
Low	High degree of automation of operational processes with less human intervention, therefore fewer human errors

* This assessment is qualitative rather than based on quantitative metrics; therefore, the results may involve a degree of subjectivity.

At the User Access Layer, human and operational factors tend to be the primary points of compromise, and the rigor of operational controls has a significant impact on the likelihood of leakage.

Risk analysis by function (1/4): User Access Layer

Function	representative actors	Key roles	Reasons to be targeted/ Vulnerabilities	Risk profile	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors / Operational risk	Representative examples from MITRE AADAPT
CEX / Custodian	Binance Coinbase BitGo	Trading, custody and fiat currency exchange	A huge asset concentration point where user assets and personal information (KYC) are processed. In addition to external attacks, there are various attack vectors such as insiders and API key leaks.	**The concentration of customer assets and the relatively broad attack exposure make it more likely to be targeted,** and **human and operational factors can increase the risk of fund outflows.** On the other hand, when external audits function well, there is room for correction and deterrence, and a combination of asset concentration, attack exposure, and human factors could be the primary risk structure.	High	Medium	High	Low	High	■ Acquire Accounts (ADT3001): Acquire Accounts with Forged or Stolen IDs ■ Scrape KYC Data (ADT3026): Collection and disclosure of personal information ■ Intercept API Communication (ADT3018): Intercept API keys and communication ■ Insider-Assisted Access (ADT3017) ■ Aggregate Private Key Generation Data (ADT3002): Analyzing the Key Generation Process and Exploiting Entropy Deficiency
SW Wallet	Metamask Phantom Safe	key management and signing interface	It's always online, making it a easy target to misidentify phishing malware approves. Most frequent incidents of total loss of personal assets	While not holding the assets directly, **signature authority is concentrated,** and in addition to attack exposure and domino effect, **human and operational factors tend to affect risk levels**. External audits, etc. are often limited, and user's misoperation or being scammed may lead to losses.	Medium	Medium	High	Medium	High	■ Manipulate Transaction History: Address Poisoning (ADT3020.001) ■ Manipulate Transaction History: Zero-Value Transfer Phishing (ADT3020.002) ■ Unsecured Credentials: Private Keys (ADT1552.004): Theft of signing keys by malware, etc.
DEX / DeFi UI	Uniswap UI GMX UI Jupiter	front end of trading operations	Supply chain compromises (external library contamination) or DNS hijacking to induce users to malicious signings. Biggest entry point for attackers even though no assets held directly	While asset concentration is low, **domino effect tend to happen due to multiple protocol connections,** and **fraud can occur** through human and operational factors. **External regulations are usually weak** and managing the attack exposure at the entrance of the front end can be important.	Low	Medium	Medium	High	High	■ Supply Chain Compromise (ADT1195): Malicious Code Injection into UI Library ■ Exploit External Services (ADT3008): Front-end (website) exploits and phishing ■ Supply Chain Compromise: Compromise Software Dependencies and Development Tools (ADT1195.001): dependency contamination
HW Wallet	Ledger Trezor Keystone	Storage and signing in physical devices	Remote infringement is extremely difficult when it's offline. However, physical supply chain attacks and mis-storage of recovery phrases (analog leakage) are the risks.	While attack exposure and domino effect tend to be relatively contained, **human and operational factors such as lost and mismanagement of device** can be the main risk. External audits are often limited to the product level, and users' operational quality may affect safety.	Medium	Medium	Medium	Medium	Medium	■ Supply Chain Compromise (ADT1195): Device tampering during logistics ■ Fault-Injection Attack (ADT3014): Glitch attack by voltage manipulation, etc. ■ Side-Channel Attack (ADT3027): Key analysis from power consumption and electromagnetic waves ■ Exploit Obsolete Device (ADT3010): Unupdated FW exploit

Risk analysis by function (1/4): User Access Layer

Function	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors/Operational risk
CEX/Custodian	High	Medium	High	Low	High
	When customer assets and critical information are aggregated into a specific operational infrastructure, they can become a prime target.	They are often linked to multiple external systems (e.g., payments, deposits and withdrawals, market connections, etc.), and the impact may spread to specific areas.	With many public APIs and user-facing features, it is relatively easy to increase the number of externally reachable contacts.	They operate under financial regulations, often requiring external audits and regulatory reporting, and certain external controls function.	There are situations in which human factors such as internal improprieties, operational errors, and phishing, directly lead to losses.
SW Wallet	Medium	Medium	High	Medium	High
	The assets themselves are on the chain, but if signing rights are concentrated on devices or apps, they could be a hot target.	It is a point of interaction with various applications, which may affect users and specific services.	Since the external environment is contacted as a browser extension, mobile app, etc., the exposed contact easily expands.	While some providers may have audit and governance arrangements in place, OSS-driven implementations may limit ongoing external controls.	It is highly dependent on the judgment and operation of the user, and fraud inducement and erroneous operation can easily lead to losses.
DEX / DeFi UI	Low	Medium	Medium	High	High
	If the UI itself does not directly hold customer assets, direct losses due to asset concentration may be relatively limited.	It is often a conduit for multiple protocols, and the impact can spill over to a particular set of users.	Although the front end is open to the public and may be subject to tampering, the function may be relatively limited.	They are often protocol-driven and generally do not have regulatory oversight or regular third-party audits.	There are situations in which it is difficult for users to accurately determine the content of transactions, and unauthorized inducement may lead to losses.
HW Wallet	Medium	Medium	Medium	Medium	Medium
	Although the assets are on the chain, they can be targeted as a means of storing signature authority (especially in terms of supply chains, forgeries, etc.).	While risks are relatively low if operations are conducted offline, in the case of online signing, there is a possibility that risks may propagate if the integrity of transaction information cannot be ensured.	Although remote accessibility is limited, attacks may still involve physical access or user interaction.	While audits and evaluations may be conducted at the product level, ongoing external oversight during the use phase may be limited.	Analog factors remain, such as lost, theft and improper storage, and there is a certain level of human risk.

‘DeFi & LST’ function has a high concentration of assets, as it is remained unregulated and unaudited, making it a significant risk factor in the overall ecosystem

Risk analysis by function (2/4): Application & Contract Layer

Function	representative actors	Key roles	Reasons to be targeted/ Vulnerabilities	Risk profile	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors / Operational risk	Representative examples from MITRE AADAPT
DeFi & LST / Restaking	Uniswap Aave Lido EigenLayer	Financial logic staking operations	Huge TVL concentrated in smart contracts. Logic bugs and reentrancies. LST/Restaking has combined risk of “locked asset + operational logic"	**With high asset concentration, domino effect and attack exposure,** in a case of limited external control, the impact of an incident may be widespread**.** It is considered that the maturity of design, implementation and operations can greatly affect the residual risk.	High	Medium	High	High	High	■ Exploit Smart Contract Implementation: Reentrancy (ADT3012.005) ■ Exploit Smart Contract Implementation (ADT3012): exploiting implementation bugs ■ Flash Loan (ADT3015): Price manipulation and attacks using flash loans ■ Smart Contract Implementation Analysis (ADT3029): vulnerability discovery through code analysis
Token Issuer	Circle (USDC) Tether Ondo	Asset backing, issuance and freezing	The contract itself is robust, but can be forged or wrongly frozen if administrative signing keys are leaked. Mainly the risks regarding internal control and governance, rather than cyberattacks	Due to the **concentration of underlying assets and issuance amount,** domino effect is relatively high and there may be issues of **dependencies on a single entity**. On the other hand, when external regulations function well, there is room for damage mitigation and correction, and the operational quality can affect the risk levels in practice.	High	High	Medium	Low	Medium	■ Unsecured Credentials: Private Keys (ADT1552.004): administrative signing key leak ■ Generate Counterfeit Tokens (ADT3016): illegal token issuance ■ Exploit Smart Contract Implementation: Contract Ownership Changes (ADT3012.001)
DAO Operator	Uniswap DAO Maker DAO	Parameter changes and fund allocation	“Proposal takeover" through governance token acquisitions or flash loans. Abuse of protocol update rights to extract funds through legitimate process	Even formally decentralized, **human and operational factors and information asymmetry can distort decision-making**. With limited external control, domino effect and participant behavior can be at the core of risk.	Medium	Medium	Medium	High	High	■ Flash Loan (ADT3015): A governance attack that allows banks to borrow large amounts of money and pass proposals ■ Exploit Smart Contract Hierarchical Ownership (ADT3011): Exploiting Hierarchical Ownership Structures ■ Exploit Smart Contract Implementation: Contract Ownership Changes (ADT3012.001): Revoke parameter change authority

Risk analysis by function (2/4): Application & Contract Layer

Function	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors/Operational risk
DeFi & LST protocol	High	Medium	High	High	High
	If value such as TVL is aggregated into a contract, it can be a prime target for attacks.	Chained use with other protocols is often assumed, and some failures may spread widely.	Since it has public logic and complicated state transitions, there is relatively large room for external abuse.	Many are operated autonomously, and there is generally no regulatory oversight or ongoing third-party audits.	Deficiencies in design, implementation, or deployment can have a significant impact, but they tend to vary depending on organizational maturity.
Token issuer	High	High	Medium	Low	Medium
	When the underlying assets or outstanding balances are large, the value or credit can be concentrated in a single entity.	When used as a key asset, they can have a broad impact on markets and many services.	There are external contacts such as smart contracts, but the functions may be relatively limited.	As a corporate entity, it is often subject to regulatory supervision and accounting audits, which may bring certain functionality of external audits.	Due to the reliance on internal controls and procedures, there is possibility of operational deficiencies or insider risks.
DAO operator	Medium	Medium	Medium	High	High
	They may hold treasuries, but the degree of concentration may vary depending on size and liquidity.	Parameter changes can affect the target system, but the level of impact depends on the system design.	They have contacts such as voting and forums, and can be the target of attack and disturbance.	Operating entities are often unclear, and regulatory oversight and ongoing third-party controls may be limited.	Decision-making may be distorted by apathy, information asymmetry, social engineering, etc.

‘Cross-chain’ function serves as bridges across assets on different chains, therefore with high asset concentration and domino effect, a failure at a single point can lead to damage on many other chains.

Risk analysis by function (3/4): Middleware & Infrastructure Services Layer

Function	representative actors	Key roles	Reasons to be targeted/ Vulnerabilities	Risk profile	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors / Operational risk	Representative examples from MITRE AADAPT
Cross Chain / Bridge	LayerZero Wormhole Across	Transfer or relay of assets between chains	Large amount of assets are locked in bridge contracts and verification logic is extremely complex. Most lucrative, with the largest number of past cases of losses	**The case of high concentration of locked assets, domino effect, and attack exposure** is likely to occur, and if external control is weak, the impact can be widespread. Implementation and operations quality can significantly affect risk levels.	High	High	High	High	Medium	■ Exploit Smart Contract Implementation: Signature Replay Attack (ADT3012.006): unauthorized withdrawals due to signature reuse ■ Exploit Blockchain Technology Specific Vulnerabilities (ADT3013): Exploiting chain-specific specifications (validation logic) ■ Cross-Chain Swaps (Hopping) (ADT3005): A way for attackers to move funds, but also in the context of exploiting bridge vulnerabilities
Key Manage-ment	Fireblocks Copper Dfns	Provision of infrastructure for key distributed management and signature	The center that manages CEX and agency signing key shares. Implementation bugs, misconfigurations, and internal irregularities can be a single technical point of failure resulting in theft of customer assets en masse	Even where assets are not held directly, **when signing authority is concentrated**, the importance of key management may increase, and the entity may also have a greater systempic impact as an underlying infrastructure provider. External regulations may work to some extent, but **human elements remain** due to complex operational flows.	Medium	High	Medium	Medium	Medium	■ Aggregate Private Key Generation Data (ADT3002): Analyzing the Key Generation Process and Exploiting Entropy Deficiency ■ Insider-Assisted Access (ADT3017) — Collusion by key share insiders ■ Unsecured Credentials: Private Keys (ADT1552.004): Decentralized key recovery and disclosure
Oracle	Chainlink Pyth RedStone	supply of off-chain pricing information	By manipulating or delaying the price feed, erroneous clearing or arbitrage may happen at DeFi. No directly held assets but may trigger market disruption	Asset concentration may be small, but **domino effect may be high given its price reference role**. External control is often limited, resulting in poor data quality and operations that can have far-reaching consequences.	Low	High	Medium	High	Medium	■ Exploit Smart Contract Implementation: Oracle Manipulation (ADT3012.004): Tampering with External Data Sources for Price Manipulation ■ Smart Contract Implementation Analysis: Oracle Analysis (ADT3029.005): Identifying Dependent Oracle and Investigating Vulnerabilities
ZK / Scaling	zkSync Polygon Zero	certification, verification, and compression	Fraudulent withdrawals can be made with false proof due to bugs in the cryptographic proof circuit. It is extremely technically difficult.	While domino effect is large given its fundamental role, external control may be weak. The attack exposure depends on the implementation, and may be influenced by human and operational factors.	Low	High	Medium	High	Low	■ Exploit Blockchain Technology Specific Vulnerabilities (ADT3013): Exploit bugs specific to new technologies (e.g. circuits) ■ Generate Counterfeit Tokens: Cryptographic Protocol Analysis (ADT3016.001): Impersonation by parsing cryptographic protocols (for example, ZK certificate forgery)
RPC / Node API	Infura Alchemy QuickNode	API access, Tx relay	No access to assets. Often used for service denial (DoS) or information spoofing with phishing sites	While asset concentration is low, domino effect and attack exposure can be high given its **communications hub** role. Alternative channels may exist, but dependencies on specific business operators and operational quality tend to affect the risk levels.	Low	High	High	Medium	Low	■ Eclipse Attack (ADT3006): Blocking/forging information to nodes, giving false information. ■ Exploit Gas-Free RPCs (ADT3009): unauthorized access to privileged RPCs ■ Intercept API Communication (ADT3018) — Data interception on a communication path.

Risk analysis by function (3/4): Middleware & Infrastructure Services Layer

Function	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors/Operational risk
Cross-chain / Bridge	High	High	High	High	Medium
	They tend to have a structure in which locked assets are concentrated and can be a major target.	Since multiple networks are directly connected, problem in one can cause problem in many others.	Verification logic and points of contact can easily become complicated, which can increase the exposure of external attacks.	Many of them operate outside the regulatory framework and may not be subject to continuous external verification and control.	Due to advanced implementation and operation, design deficiencies or operational errors can affect.
MPC / Key Manager	Medium	High	Medium	Medium	Medium
	Even if the assets are not held directly, the aggregated signing authority can increase their importance.	If it is used as a basis for multiple operators, the impact of failures or infringements may spread.	While restrained by access control, administrative contacts may remain.	Corporate services may be subject to SOC audits, etc., but the scope of ongoing audits of actual operations may vary by business operator.	Approval flows and operational procedures are likely to become complex, and human errors and procedural deficiencies can affect.
Oracle	Low	High	Medium	High	Medium
	They usually do not hold assets, and direct losses from asset concentration are considered to be limited.	When a large number of protocols are relying upon them as price references, the impact can be widespread.	There are contact points such as data acquisition source and node operation, and a certain level of exposure can occur.	In some cases, audits are conducted, but the ongoing supervision system tends to be limited.	Even when automated, risks in operations, settings, and response decisions may remain.
ZK / Scaling	Low	High	Medium	High	Low
	Assets may be aggregated around a bridge, and the degree of concentration may depend on the configuration.	If there is a lot of use as an infrastructure between L1 and L2, the impact can be widespread.	There is a cryptographic implementation and operation interface, and a certain room for attack may remain.	The system is operated mainly by protocols, and usually there is no regulatory supervision or regular audits.	When automation advances, human intervention is relatively small, but implementation errors, etc. need to be considered separately.
RPC / Node API	Low	High	High	Medium	Low
	They usually do not hold assets, and direct losses from asset concentration are considered to be limited.	They are communication hubs between users and the network, and a failure can have a wide impact.	They have a lot of public access and are easily to be targeted in DDoS.	Large business operators have SOC or other auditings in place, but overall the level of control varies by operator.	It is often automated in the standard operation, but the influence of settings and operational decisions may remain.

The “Software Resources” function has a broad attack surface, and once malicious code is injected, the impact can easily propagate to many projects.

Risk analysis by function (4/4): Blockchain Layer, Cross-functional Layer: Hardware & Environment

Function	representative actors	Key roles	Reasons to be targeted/ Vulnerabilities	Risk profile	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors / Operational risk	Representative examples from MITRE AADAPT
Tx Sequence Control (MEV/Seq)	Flashbots L2 Sequencer	Transaction sequence control	Censorship and outage risk of sequencers (centralized servers) / Exploits of user profits through MEV (Sandwich attack)	Although there is no asset custody, there may be large domino effect to the whole market. External control such as regulations is limited, and **design transparency and operational policies can influence the risk degree**.	Medium	High	Medium	High	Medium	■ Market Manipulation (ADT3021): Market manipulation such as sandwich attacks ■ Smart Contract Implementation Analysis: Timestamp Dependence Analysis (ADT3029.006): exploiting sequence and timestamp dependencies ■ Induce Legal and Regulatory Penalties (ADT3019): Induce risk of regulatory violations through censorship
Consensus (Validator)	Coinbase Cloud Figment	Block generation and verification	Thrashing due to validator key leakage (asset forfeiture). Risk of censorship and reorg increases in the case of concentrated stakeholders	As consensus basis there is a large domino effect, and the concentration of staking assets may vary by logic design. Under the same regulations the operation model may still differ, **human and operational elements such as key management and infrastructure operations may be the cause of incidents**.	Medium	High	Medium	High	Medium	■ Exploit Consensus Logic (ADT3007): Consensus Vulnerability Exploit ■ Chain Reorganization (ADT3003): Reorganizing a chain to invalidate transactions ■ Unsecured Credentials: Private Keys (ADT1552.004): thrashing with validator key leak
L1/L2 Network	Ethereum Solana Optimism	Ledger records and consensus	A 51% attack on a major chain is unrealistic as it requires a national budget level of cost. However, it can be done cheaply on minor chains	This is an infrastructure layer that tends to have **high asset concentration, domino effect and attack exposure**, and external controls are often limited. While decentralization may mitigate some risk, protocol and operational quality can affect residual risk.	High	High	Medium	High	Low	■ Exploit Consensus Logic: Circumvent Voting Majority Control (ADT3007.001): 51% attacks ■ Exploit Consensus Logic: Double-Spending Attack (ADT3007.002) ■ Exploit Consensus Logic: Sybil Node Creation (ADT3007.004): Network Domination with Large Number of False Nodes
Physical Infrastructure (Cloud/HSM)	AWS Thales CloudHSM	Server, network and physical key protection	The attacks will aimed at "DoS" or physical destruction, not asset theft. HSMs are deadly if breached, but such cases are rare	while dependencies on upper layer tend to cause asset concentration and domino effect, in many cases, external audits and mature operations are seen. **Concentration on specific service providers and attacks on management APIs** could be the main points of discussion.	Medium	High	Medium	Low	Low	■ Fault-Injection Attack (ADT3014): Induction of malfunction by physical stress ■ Side-Channel Attack (ADT3027): physical information leak ■ Reputation Damage (ADT3024): Credit damage due to DoS
Software Resources	OSS npm GitHub	Software development, code management and libraries	Supply chain attacks that exploit dependencies. Injected malicious code bringing damages to many projects where it’s used	In a situation where external control such as regulations is limited, its open sourced nature, as well as human and operational factors, can **make it easy to be a target of supply chain compromises and cause cascading damage to a wide range of systems**.	Low	High	High	High	High	■ Supply Chain Compromise (ADT1195): Malicious code intrusion into open source dependencies and source code repositories ■ Exploit External Services (ADT3008): Exploit vulnerabilities in external services, such as third-party providers, to gain initial access to broadly connected systems

Risk analysis by function (4/4): Blockchain Layer, Cross-functional Layer: Hardware & Environment

Function	① Concentration of assets	② Domino effect	③ Attack exposure	④ Lack of regulations, etc.	⑤ Human errors/Operational risk
Tx Sequence Control (MEV/Seq)	Medium	High	Medium	High	Medium
	They do not hold assets, but can be targeted because value extraction opportunities can be concentrated.	Widespread contagion can occur because the transaction sequence can affect market behavior.	There are communication and operational interfaces, remaining certain room for attacks.	In many cases, there is no clear control and supervisory framework, and continuous external verification may be limited.	While automation may advance, design and transparency issues may remain.
Consensus (Validator)	Medium	High	Medium	High	Medium
	Staking assets may be aggregated, and the degree of concentration may vary with distribution.	They serve as a basis for consensus building, and a failure can affect the entire network.	There are P2P communications and operational infrastructures, remaining certain room for attacks.	In the case of corporate operator, there may be a certain level of control, but in the case of individual operating nodes, external regulations may not reach.	Inadequacies in key management or operation settings can cause accidents.
L1/L2 Network	High	High	Medium	High	Low
	Overall values and activities tend to be concentrated in the infrastructure layer, and the impact in the event of a serious incident can be large.	They are the execution foundation for all applications, and failures can spread widely.	There are a wide range of layers from consensus to execution, which can increase the attack exposure.	They are often operated as decentralized protocols and may not have regulatory oversight or ongoing third-party controls.	They are usually operated automatically by protocols, but there may be other factors related to operation or implementation.
Physical Infrastructure (Cloud/HSM)	Medium	High	Medium	Low	Low
	Nodes and keys may be aggregated in a particular infrastructure, and the degree of concentration may depend on the adoption.	Failures can propagate because the upper layers widely depend on them.	Even in the case of strict control, external contact points such as administrative APIs may remain.	Third-party verification or auditing are usually in place, and a relatively high level of external auditing can be expected.	Sophisticated operational procedures are usually in place, but it is not zero risk and may depend on the operational quality.
Software Resources	Low	High	High	High	High
	Since they do not directly hold customer assets, direct economic losses from asset concentration are considered to be limited.	They often serve as the foundation upon which many projects depend, and failures and breaches can spread widely throughout the ecosystem.	They are widely accessible serving as a public repository or package manager, and there are relatively many external contact points for malicious code injection.	They are often open sourced or community-driven, and tend to have limited regulatory oversight.	Human factors, such as inadequate account management by developers and social engineering, can easily lead to supply chain attacks.

Appendix3. MITRE Framework

The MITRE framework, a comprehensive knowledge database of attack, detection, and defense, provides a foundation for evaluating the completeness and effectiveness of security measures.

Overview of MITRE ATT&CK® / MITRE AADAPT™ / MITRE D3FEND ™

What is MITRE?

MITRE is a non-profit organization in the United States that supports researches and development in areas such as cybersecurity, national security, and medical care for public institutions such as the U.S. federal government.
In cybersecurity, MITRE established frameworks for databased attack tactics & techniques and vulnerability management programs such as CVEs.

Attack	**MITRE ATT&CK® (*1)**	■ A comprehensive framework for databased attack tactics & techniques against systems based on real-world observations. ■ The attack methods are classified by tactics such as reconnaissance, resource development, and initial access. Each technique is associated with sub-techniques, mitigations and detections. ■ The tactics & techniques are databased in each system domain such as Enterprise/Mobile/ICS (industrial control system).
	MITRE AADAPT™ (*2)	■ A comprehensive framework for databased attack tactics & techniques against management systems of digital assets such as crypto assets. ■ Based on the MITRE ATT&CK® model, it complements the attacker's view specific to digital assets. ■ It consists of 11 tactics and associated techniques, mitigations, and detections. There is no description of tactics and techniques for each system domain such as Enterprise/Mobile/ICS.
Defense	**MITRE D3FEND ™ (*3)**	■ MITRE D3FEND is a knowledge database that systematizes defense technologies and countermeasures. ■ This framework is mapped to MITRE ATT & CK and describes specific measures for detection, prevention, and response.

Utilization in this research

- ➢ In case study, we mapped sampled cases to relevant attack techniques from this framework
- ➢ We used it as a reference to guide the formulation of effective countermeasures and improvement plans
  - This research covers only the Enterprise domain (MITRE ATT&CK®)

(*1) MITRE ATT&CK® : MITRE Adversarial Tactics, Techniques, and Common Knowledge -
(*2) MITRE | AADAPT™: MITRE Adversarial Actions in Digital Asset Payment Technologies
(*3) MITRE D3FEND™: MITRE Digital Defensive Framework for Enterprise Network

MITRE ATT&CK® is a knowledge database that systematizes ‘Tactics & Techniques’ of attackers against IT systems based on actual observations.

(Reference) Tactics & Techniques in MITRE ATT&CK® (1/2)

Tactics & Techniques

- ‘Tactics’ refers to the tactical goals (intentions) of an adversary in each phase of an infringement or attack. The "why" of the technique, that is, the reason why the attacker takes the action. It is organized into 14 categories, and it is a framework to divide stages and aims in terms of the objective of attack, and specific means (techniques) to achieve are linked to each tactic.
- ‘Techniques’ are specific ways and means used by an attacker to achieve a tactical goal.

	tactics	Content	technique number	typical technique	Contents of typical techniques
1	Reconnaissance	Gather information that can be used to plan future operations.	11	Active scanning	Scan networks and public systems to identify running services and vulnerabilities.
2	Resource Development	Establish resources they can use to support operations.	8	Acquire Infrastructure	Acquire and build physical or cloud servers, domains, and third-party web services to provide an attack surface.
3	Initial Access	Gain their initial foothold within a network.	11	Phishing	Use fake emails and sites to trick users into providing credentials and running malware.
4	Execution	Run malicious code.	17	Command and Scripting Interpreter	Abuse command or script interpreters to perform unauthorized processing or control.
5	Persistence	Keep access to systems across restarts, changed credentials, and other interruptions that could cut off the access.	23	Boot or Logon Autostart Execution	Allow programs to run automatically at system startup or login to maintain persistence or gain higher privileges on compromised systems.
6	Privilege Escalation	Gain higher-level permissions on a system or network.	14	Exploitation for Privilege Escalation	Exploit a software vulnerability to elevate the privilege level.
7	Defense Evasion	Avoid detection throughout the compromise.	47	Obfuscated Files or Information	Obfuscate the contents of a system or in transit by means of encryption, encoding, etc., in order to make it difficult to find or analyze executable files.
8	Credential Access	Get credentials include keylogging or credential dumping.	17	OS Credential Dumping	Extract from OS cache and memory to obtain account login and authentication information.
9	Discovery	Gain knowledge about the system and internal network.	34	Network Service Discovery	Obtain a list of services running on a remote host or network infrastructure device.
10	Lateral Movement	Move through the environment and control remote systems on a network.	9	Remote Services	Exploit remote services to gain unauthorized access to internal systems when inside the network.

Source : https://attack.mitre.org/ (MITRE Corporation) As of March 2026

MITRE ATT&CK® is a knowledge database that systematizes ‘Tactics & Techniques’ of attackers against IT systems based on actual observations.

(Reference) Tactics & Techniques in MITRE ATT&CK® (2/2)

	tactics	Content	technique number	typical technique	Contents of typical techniques
11	Collection	Gather information and the sources information is collected from that are relevant to following through on the adversary's objectives.	17	Data from Local System	Collect local files and databases on compromised devices.
12	Command and Control	Communicate with systems under their control within a victim network.	18	Application Layer Protocol	Uses OSI application layer protocols to blend into existing traffic to avoid detection and network filtering and communicate.
13	Exfiltration	Steal data from the network.	9	Exfiltration Over C2 Channels	Using existing C2 communications to steal data in a non-detectable manner.
14	Impact	Disrupt availability or compromise integrity by manipulating business and operational processes.	15	Data Encrypted for Impact	Encrypts data on a target system or many systems in a network, disrupting the availability of system and network resources.

Source : https://attack.mitre.org/ (MITRE Corporation) As of March 2026

MITRE ATT&CK® is a knowledge database that systematizes 'Tactics & Techniques' of attackers against IT systems based on actual observations.

(Reference) Mitigations and Detections in MITRE ATT&CK®

Mitigations

- It systematizes the concept of defense measures to prevent or make it difficult to execute attack techniques, and each measure is organized as a Mitigation ID.
- Classifications are defined by domain (Enterprise/Mobile/ICS)
- Currently, the total number is 109 (Enterprise: 44, Mobile: 13, ICS: 52).

typical mitigation measures

	ID	Name	Summary
1	M1032	Multi-factor Authentication	Prevents unauthorized logins by requiring multiple authentication factors, such as one-time codes and biometric information, in addition to passwords.
2	M1026	Privileged Account Management	Strictly control the use of administrator and high-privilege accounts, with minimal privileges for normal operations.
3	M1030	Network Segmentation	Divide the network into zones and segments, limiting communication paths and making it difficult for attackers to deploy laterally.
4	M1036	Account Use Policies	Reduce the risk of unauthorized access by setting account usage rules and enforcing login time restrictions and lockouts.
5	M1015	Active Directory Configuration	Detect, block, and remove known malware and suspicious processes using signatures and behavioral analysis.
6	M1016	Vulnerability Scanning	Automatically or independently detect and assess system, application, and network vulnerabilities on a regular basis to determine priority for remediation.

Detections

- A framework that organizes the concept of detection in order to observe and identify the actions of attackers. It is provided as a design concept rather than a single rule.
- Classification starts with Detection Strategies and is organized into Analytics and Data Components structures.
- Currently, the total number is 3049 (Detection Strategies: 898/Analytics: 2032/Data Components: 119).

typical detection measure

	ID	Name	Summary
1	DET0597	Detect Unauthorized Access to Password Managers	Detects unauthorized access to password manager processes (such as 1Password) through abnormal process injection, memory reading, or command line use of associated DLLs.
2	DET0320	Detection of System Network Connections Discovery Across Platforms	Detect enumeration behavior of a system or network connection. Capture traces of network searches through API calls.
4	DET0515	Detection Strategy for T1528 - Stealing Application Access Token	After accessing and retrieving a container service account token, detect unauthorized API requests to use the token to interact with the service.
5	DET0070	Detection Strategy for Phishing across platforms.	Detect cross-platform phishing activity.

Source : https://attack.mitre.org/ (MITRE Corporation) As of March 2026

MITRE AADAPT™ is a knowledge database that systematizes 'Tactics & Techniques' of attackers against digital asset (such as crypto assets) management systems

(Reference) Components of MITRE AADAPT™

Tactics & Techniques

- Tactics refers to the tactical goals (intentions) of an adversary in each phase of an infringement or attack. The "why" of the technique, that is, the reason why the attacker takes the action. It is organized into 11 categories, and it is a framework to divide stages and aims in terms of the objective of attack, and specific means (techniques) to achieve are linked to each tactic.
- Techniques are specific ways and means used by an attacker to achieve a tactical goal.

	tactics	Content	technique number	typical technique	Contents of typical techniques
1	Reconnaissance	Gather information they can use to plan future operations.	2	Smart Contract Implementation Analysis	Analyzing the code and specifications of publicly available smart contracts to identify potential vulnerabilities that could exploit the system.
2	Resource Development	Establish resources they can use to support operations.	2	Acquire Accounts	Create an active digital asset system account to embezzle funds. use fake personal information to create seemingly legitimate accounts.
3	Initial Access	Gain their initial foothold within a network.	4	Exploit External Services	Exploit vulnerabilities in external services to gain unauthorized access to digital asset systems.
4	Execution	Run malicious code.	6	Exploit Smart Contract Implementation	Exploit vulnerabilities in the implementation of smart contracts in digital asset systems to perform unauthorized processing such as transaction operations and fund theft.
5	Privilege Escalation	Gain higher-level permissions on a system or network.	1	Exploit Smart Contract Hierarchical Ownership	Exploiting vulnerabilities in the hierarchical ownership structure of smart contracts to gain unauthorized control and privileges within the contract.
6	Defense Evasion	Avoid detection throughout the compromise.	4	Use Anonymization Services	Use anonymization techniques to avoid tracking and detection of financial flows and attacks.
7	Credential Access	Steal signature keys, account credentials, etc.	2	Unsecured Credentials	Explore compromised systems, obtain credentials that are not properly secured, and gain access.
8	Lateral Movement	Move through the environment and control remote systems on a network.	2	Exploit Gas-Free RPCs	Leverage RPC without standard transaction fees (gas) to extend reach to multiple addresses and environments.
9	Collection	Gather information and the sources information is collected from that are relevant to following through on the adversary's objectives.	4	Aggregate Private Key Generation Data	Collect information about signature key generation to prepare for future unauthorized use or expansion of compromise.
10	Impact	Disrupt availability or compromise integrity by manipulating business and operational processes.	5	Burn Wallets	Transfer fraudulently obtained funds to burn wallets (abandoned wallets or wallets that have lost their signing keys) so that no one can recover the assets.
11	Fraud	Create, acquire, or utilize value-form illicitly.	7	Chain Reorganization	Create a longer alternative blockchain to overwrite the existing chain to manipulate transaction history and avoid detection.

Source : https://aadapt.mitre.org/ (MITRE Corporation) As of March 2026

MITRE D3FEND™ is a knowledge database mapped to MITRE ATT & CK that systematizes defense technologies and countermeasures against cyberattacks, to support the design and evaluation of defenses

(Reference) Components of MITRE D3FEND™

Tactics & Techniques

- Tactics are high-level categories that indicate the purpose and direction of defense activities. A concept that organizes "what defense is to achieve" against cyberattacks, and systemizes defense objectives such as detection, isolation, and reinforcement.
- Technique is a concrete defense technique to realize tactics. An implementable defense technology that protects, monitors, and analyzes systems and networks

	tactics	Content	technique number	typical technique	Contents of typical techniques
1	Model	Optimize and understand assets and environment for defense design	4	Asset Inventory	Comprehensive understanding and management of IT assets such as servers, terminals, network devices, and applications within the organization to serve as a foundation for defense design and risk management
2	Harden	reduce the attack surface and make infringement less likely	6	Application Hardening	Reduce the attack surface of applications by disabling unnecessary functions, strengthening settings, and taking measures against vulnerabilities, thereby reducing the risk of unauthorized use and infringement.
3	Detect	Monitor for signs and detect breaches early	8	File Analysis	A technique for examining suspicious file behavior and signatures in detail. Combining static and dynamic analysis to detect unknown malware and attack intent at an early stage, leading to rapid response
4	Isolate	Isolate the scope of the infringement and prevent its extension	5	Network Isolation	Technique for logically isolating a compromised terminal or segment. Minimize damage by physically preventing the spread of infection (horizontal deployment) and communication with external attack command servers
5	Deceive	use decoys to guide attackers and visualize their tactics	2	Decoy Environment	Guiding an attack to a decoy system, collecting attack techniques, entry routes, and IoC
6	Evict	Eliminate infringing elements and prevent reentry	3	Process Eviction	Terminate and stop unauthorized or compromised processes from the system, thereby blocking the attacker's activities and malware execution and preventing further damage
7	Restore	Restore safety to ensure business continuity	2	Restore Object	Recover corrupted or deleted files or system objects using backups or snapshots

Source : https://d3fend.mitre.org/ (MITRE Corporation) As of March 2026

Appendix4. OWASP SCSVS

“Standardization of verification criteria” for smart contract-specific risks will enable crypto-asset-related business operators to enhance security controls, including the area of external dependencies.

Overview of OWASP SCSVS

What is SCSVS?

- Smart Contract Security Verification Standard established by OWASP
- A framework for improving security of smart contracts from both design and operation perspectives
- It systematized security requirements for smart contracts in 11 areas by classifying them into 3 different levels

	Security Requirements	Contents
S1	Architecture, Design, and Threat Modeling	Identify, evaluate, and reduce threats by designing smart contracts based on modularity, upgradability, and module (contract and function) partitioning so that they can be safely operated, upgraded, and maintained.
S2	Policies, Procedures, and Code Management	Develop development policies and procedures that promote secure coding, thorough code review, and comprehensive testing to prevent vulnerabilities and improve code maintainability and clarity
S3	Business Logic and Economic Security	Ensure the integrity of token handling and transaction flows by ensuring that business logic and economic models are resistant to abuse and unexpected behavior against threats from incentive design, tokonomics, and logic vulnerabilities
S4	Access Control and Authentication	Establish robust access control and authentication, including role-based access control (RBAC), authorization mechanisms, and distributed identity management so that only authorized actors can perform sensitive operations
S5	Secure Interactions and Communications	Secure inter-contract cooperation, Oracle cooperation, cross-chain cooperation, and bridge communication and interaction, and establish a secure cooperation protocol through secure handling of external calls, data integrity, and handling at the time of failure.
S6	Cryptographic Practices	Secure cryptographic implementations for key management, signature verification, and random number generation to protect the integrity and authenticity of transactions and data
S7	Arithmetic and Logic Security	Prevent arithmetic vulnerabilities such as overflow/underflow, and ensure the integrity of computations in contracts through correct computation and logical processing
S8	Denial of Service (DoS)	Establish mechanisms and designs for gas efficiency, fallback, and resource exhaustion tolerance (rate limiting, etc.) to prevent DoS that impedes contract function and availability.
S9	Blockchain Data and State Management	Establish practices for safe, efficient, and consistent management of data/state on the blockchain to prevent state corruption and unexpected behavior
S10	Gas Usage, Efficiency, and Limitations	Establish practical methods to optimize gas consumption to reduce costs and improve performance
S11	Component-Specific Security	Establish component-specific security practices and standards to mitigate vulnerabilities specific to each component, such as tokens, NFTs, vaults, and liquidity pools

Reference: "https://scs.owasp.org/SCSVS/" (OWASP) As of March 2026

Appendix5. SEAL Certification Framework

Framework for third-party certification that utilizes standards specialized in DeFi/Web3 that are important from a security perspective and evaluates them on a modular, evidence-based basis

SEAL Certification Framework Overview

- An operational security authentication framework for the DeFi Web3 protocol developed by the Security Alliance (SEAL).
  ※ Security Alliance (SEAL): A non-profit organization that provides 24/7 incident response, threat intelligence, and security coordination for the cryptoasset industry
- The framework aims to (1) enable the protocol itself to self-assess its own security posture and projects, (2) enable third-party assessments to show that the protocol meets a certain level of operational security, and (3) create security standards that can be compared between cryptoasset protocols.
- Cryptoassets do not cover the broad areas of SOC2 and ISO27001, but are limited to highly influential areas of operations specific to cryptoasset protocols.
- SEAL also offer a third-party certification system, and an on-chain verifiable certificate is issued via the Ethereum Attestation Service (EAS) if the certification is successful.
  - Each item is evaluated on a four point scale: "Implemented," "Partially Implemented," "Not Implemented," and "N/A."
  - Modular, meaning that the customer does not have to meet all of the criteria at once, but can gradually increase the maturity level in the most important areas
  - Based on evidence supporting implementation, such as documents, logs, setting screens, staffs, and review frequency

DevOps & Infrastructure

Development environment, source code, CI/CD, cloud infrastructure, supply chain

Prevent protocol development, deployment, and infrastructure operation from becoming attack vectors

DNS Security

Domain management, DNS controls, registrar security, email authentication

Prevent front-end hijacking, DNS hijacking, fake site induction, and fraudulent certificate issuance

Incident Response

Threat modeling, monitoring, response playbooks, drills

Define in advance who will do what, in what order, and with what authority when the protocol is attacked

Multisig Ops

Governance, signer security, transaction verification, emergency procedures

Prevent infringement of multi-sig operation such as Admin/upgrade/treasury/emergency pause authority

Treasury Ops

Treasury architecture, transaction security, custody, DeFi risk management

Treat financial management as a risk-based asset preservation process, not just a wallet management

Identity & Accounts

Organizational account inventory, phishing-resistant MFA, credential management, takeover monitoring

Maintain control of devices, accounts, MFA, credentials, software, telecommunications, on/off boarding, education, insider risk, and third-party access

Sources: https://frameworks.securityalliance.org/certs/overview

Framework for third-party certification that utilizes standards specialized in DeFi/Web3 that are important from a security perspective and evaluates them on a modular, evidence-based basis

Modules	Sections/Controls
DevOps & Infrastructure	• Governance & Development Environment(DevOps Security Owner, DevOps Security Policy, Development Environment Isolation, Development Tools Approval and Approved List) • Source Code & Supply Chain Security(Repository Security, Secret Scanning, External Contributor Review, Dependency and Supply Chain Security) • CI/CD Pipeline Security(Pipeline Security Controls, Secrets Management, Security Testing Integration) • Infrastructure & Cloud Security(Infrastructure as Code, Infrastructure Access Controls, Backup and Disaster Recovery, Cloud Security Monitoring)
DNS Security	• DNS & Registrar(Domain management, DNS controls, registrar security, email authentication) • Governance & Domain Management(Domain Security Owner, Domain Inventory and Documentation) • Risk Assessment & Classification(Domain Classification and Compliance, Enterprise Registrar Security Requirements) • Access Control & Authentication(Registrar Access Control, Dedicated Domain Security Contact Email, Change Management for Domain Operations) • Technical Security Controls(DNS Security Standards, Email Authentication Standards, Domain Lock Implementation, TLS Certificate Lifecycle Management) • Monitoring & Detection(Domain and DNS Monitoring, Certificate Transparency Monitoring, Domain Expiration Prevention) • Incident Response(Alerting and Emergency Contacts, Domain Incident Response Plan)
Incident Response	• Governance & Team Structure(Incidents Response Team and Role Assignments, Stakeholder Coordination and Contacts) • Monitoring, Detection & Alerting(Monitoring Coverage, Alerting, Paging, and Escalation, Logging Integrity and Retention) • Response & Emergency Operations(Response Playbooks, Signer Reachability and Coordination, Emergency Transaction Readiness) • Communication & Coordination(Incident Communication Channels, Internal Status Updates, Public Communication and Information Management) • Testing & Continuous Improvement(IR Drills and Testing)
Multisig Ops	• Governance & Inventory(Named Multisig Operations Owner, Multisig Registry and Documentation) • Risk Assessment & Management(Multisig Classification and Risk-Based Controls, Contract-Level Security Controls, Exception Approval Process, Wallet Segregation) • Signer Security & Access Control(Signer Address Verification, Signer Key Management Standards, Seed Phrase Backup and Protection, Signer Lifecycle Management, Signer Training and Assessment, Hardware Wallet Standards, Secure Signing Environment, Signer Diversity) • Operational Procedures(Transaction Proposal, Verification, and Execution Procedures, Transaction Records and Evidence Retention, Multisig Tool and Platform Evaluation, Backup Signing Infrastructure) • Communication & Coordination(Secure Communication Procedures, Emergency Contact List) • Emergency Operations(Emergency Playbooks, Signer Reachability and Escalation, Multisig Monitoring and Alerts, Emergency Drills and Improvement)
Treasury Ops	• Governance & Treasury Architecture(Treasury Operations Owner, Treasury Registry and Documentation, Custody Architecture Rationale, Treasury Infrastructure Change Management) • Risk Classification & Fund Allocation(Treasury Wallet Risk Classification, Portfolio Concentration Limits and Rebalancing, Per-Actor and Per-Path Exposure Limits) • Access Control & Platform Security(Custody Platform Security Configuration, Credential and Secret Management, Access Reviews for Treasury Systems, Personnel Operational Security, Privileged Access and Root Account Management) • Transaction Security(Transaction Verification and Execution, Signer and Approver Knowledge, Secure Communication Procedures) • Protocol Deployments(Protocol Evaluation and Exposure Limits, Position Lifecycle Management) • Monitoring & Incident Response(Monitoring and Threat Awareness, Incident Response Plan) • Vendor & Infrastructure(Vendor Security Management, Backup Infrastructure and Alternate Access) • Accounting & Reporting(Financial Recordkeeping and Reconciliation, Insurance Coverage)
Identity & Accounts	• Governance & Inventory(Organizational Account Security Owner, Organizational Account Inventory) • Authentication & Credentials(Phishing-Resistant Multi-Factor Authentication, Credential Management and Individual Accountability, Recovery Methods Restricted to Organizational Channels) • Access & Lifecycle(Account Lifecycle Management) • Monitoring & Third-Party(Organizational Account Takeover Monitoring, Third-Party Access Management)

Sources: https://frameworks.securityalliance.org/certs/overview

Appendix6. CryptoCurrency Security Standard（CCSS）

CCSS is a specific standard for cryptoasset, including private key management, operation control, audit trail, and access control, for exchanges, wallets, custody services, and web apps.

CryptoCurrency Security Standard Overview

- The CryptoCurrency Security Standard (CCSS) is a set of requirements for all information systems that use cryptoassets.
- Updated regularly to address the rapidly evolving cryptoasset industry (Version 9.0, the latest version was released on December 17, 2024)
- Designed to complement existing information security standards (e.g., ISO/IEC 27001:2022) by introducing guidance on cybersecurity best practices for cryptoasset
- The following three systems, not the business entity, are certified. Systems are certified as CCSS Level1, Level2, and Level3 as security levels increase
  - Self-Custody: A system that independently controls the private key that manages the entity's own funds
  - CCSS Qualified Service Provider: A system that provides a portion of the custody service to other systems.
  - CCSS Full System: A system that meets all applicable CCSS requirements as a whole
- CCSS implementation and audit have separate roles
  - CCSS Implementer (CCSSI): A supporter for businesses to design, implement and document controls in line with CCSS
  - CCSS Auditor (CCSSA): An auditor who audits the system and makes an evaluation based on CCSS
- The audit covers the status of control operations for the most recent 12 months and is expected to be conducted at least once a year in principle.
- Audit results are reviewed by CCSSA peer reviewer, and in the event of a dispute, CCSS Steering Committee intervenes

Category	Aspect	Detail
1. Cryptographic Asset Management Controls for securely generating, storing, accessing, using, and sanitizing key material	1.01 Key Material Generation	Confidentiality of key generation, randomness, the key generation environment and procedures
	1.02 Wallet Generation	Signing configuration, redundancy, geographic and organizational distribution, and wallet generation policy
	1.03 Key Material Storage	Encryption, backups, environmental protection, access control, tamper evidence, and backup encryption
	1.04 Key Material Access	Key material access grants/revocations, approved communication channels, and audit trails
	1.05 Key Material Usage	Access authentication, trusted environment isolation, personnel screening, training, and spend verification
	1.06 Data Sanitization Documentation	Media sanitization, destruction, and audit trails for key material
2. Operations Controls for auditing, monitoring, governance, and compromise response across the key management environment	2.01 Security Tests/Audits	Third-party security assessments, penetration testing, and smart contract code audits
	2.02 Log and Monitor	Audit logs, log backups, log monitoring, and blockchain state monitoring
	2.03 Governance and Risk	Executive management responsibility, threat modeling, risk management, and service provider management
	2.04 Key Compromise Documentation	Key inventory, key compromise policy, and training/rehearsals

Sources: https://cryptoconsortium.org/standards-2/

CCSS is a specific standard for cryptoasset, including private key management, operation control, audit trail, and access control, for exchanges, wallets, custody services, and web apps.

Control details（1/3）

Aspect	Controls	Detail
1.01 Key Material Generation	1. Actor-generated Key Material	Key material is generated by the using actor and is not received from another actor.
	2. Validation of Generation Methodology	The key material generation methodology is validated prior to use to ensure no value restrictions, data transfer to another actor, or determinism.
	3. DRBG Compliance	The generation mechanism for key material conforms to NIST SP 800-90A.
	4. Entropy Pool	Key material is generated on a Key Management System with sufficient entropy to ensure that it is not generated with any bias toward a reduced range of values or other deterministic properties.
1.02 Wallet Generation	1. Signing Configuration	Single-signer or multi-signer mechanisms are selected appropriately based on the wallet's criticality, customer-fund impact, compromise risk, and implemented security controls.
	2. Key Material Redundancy	Multi-signer wallets must have at least one redundant key for recovery purposes.
	3. Geographic Key Material Distribution	Multi-signer wallet key material must be stored in geographically separate locations to mitigate risks from disasters, break-ins, and localized disruptions.
	4. Entity Key Material Distribution	Multi-signer wallet key material must be stored by distinct operators within the entity or separate entities to reduce organizational single points of failure.
	5. Wallet Generation Policy Documentation	The entity must document a wallet generation policy that details internal procedures.
1.03 Key Material Storage	1. Encryption of Operational Key Material	Key material is stored with the use of strong encryption when not in use.
	2. Key Material Backup(s)	Backups must exist for operational key material and for all key material used in wallets.
	3. Environmental Protection for Key Material Backup(s)	Backups are protected against environmental risks and stored in geographically separate locations from the storage and usage of operational key material.
	4. Key Material Backup(s) Have Access Control	Key material backups must be protected by access controls, such as safes, safe deposit boxes, or locked storage, to prevent unauthorized access.
	5. Tamper-evident Key Material Backup(s)	Key material backups must use tamper-evident mechanisms, such as seals or serial-numbered bags, to detect unauthorized access or tampering.
	6. Key Material Backup(s) Encryption	Key material backups must be stored with strong encryption equal to or greater than the security prescribed for operational key material.

CCSS is a specific standard for cryptoasset, including private key management, operation control, audit trail, and access control, for exchanges, wallets, custody services, and web apps.

Control details （2/3）

Aspect	Controls	Detail
1.04 Key Material Access	1. Grant/Revoke Documentation	The entity must manage personnel transitioning into or out of key holder roles using least-privilege checklists.
	2. Approved Communication Channel	Key holder grant/revoke requests must be conducted over Approved Communication Channels.
	3. Grant/Revoke Audit Trail	The entity's checklists include audit information that records the identity of personnel performing key holder grant/revoke operations.
1.05 Key Material Usage	1. Access Authentication to Key Material	Access to key material must require authentication.
	2. Operational Key material Environment	Operational key material must be used only in a CCSS Trusted Environment, not on general-purpose endpoints.
	3. Operator Reference Checks	Operator reference checks must be performed before granting key access.
	4. Operator ID Checks	Operator ID checks must be performed before granting key access.
	5. Operator Background Checks	Operator background checks must be performed to the extent permitted by law.
	6. Key Management Training	Key management training must be provided upon hiring, before key access, and annually.
	7. Key Management Responsibilities	Personnel with key management responsibilities must acknowledge their responsibilities in writing.
	8. Spend Verification	Spend destination and amount must be verified over Approved Communication Channels before signing.
	9. Multi-Signer Mechanism Usage	Multi-signer wallet key material must be stored and used on separate logical or physical devices.
	10. DRBG Compliance for Signatures	Digital signature randomness must comply with cryptographic best practices for the applicable signing algorithm.
1.06 Data Sanitization Documentation	1. Data Sanitization Policy Existence	Policy and procedure documents must define the sanitization and destruction requirements for media holding key material.
	2. Media Sanitization Audit Documentation	Audit trails must be maintained for media sanitization involving key material, including media identifiers, methods, personnel, and other relevant information.

CCSS is a specific standard for cryptoasset, including private key management, operation control, audit trail, and access control, for exchanges, wallets, custody services, and web apps.

Control details （3/3）

Aspect	Aspect Controls	Detail
2.01 Security Tests/Audits	1. Security Development and Documentation	Security expertise, independent assessments, and remediation must be integrated throughout the system lifecycle.
	2. Smart Contract Software Code Audit Documentation	Smart contract software code audits must be completed and documented, with all identified concerns risk-assessed and addressed.
2.02 Log and Monitor	1. Application Audit Logs	Audit logs are captured and retained for operations performed within the CCSS Trusted Environment.
	2. Audit Log Backup	Audit logs are backed up to a separate environment to ensure that evidence is preserved in the event of log deletion or tampering.
	3. Audit Log Monitoring	Audit logs are monitored, and alerts are issued when suspicious activity is detected.
	4. Blockchain State Monitoring	Relevant on-chain states are continuously monitored to detect suspicious activities such as asset movements.
2.03 Governance and Risk	1. Governance	Executive management is formally assigned responsibility for system security and provides written acknowledgement of that responsibility.
	2. Risk Management	The entity identifies threats to the CCSS Trusted Environment and implements controls based on its threat model and risk management program.
	3. Service Provider Management	Service providers that may affect the CCSS Trusted Environment are subject to due diligence before engagement and ongoing reviews at least annually.
2.04 Key Compromise Documentation	1. Key Compromise Policy Existence	The entity maintains an inventory of all key material and documents the policy, approved communication channels, and roles and responsibilities for responding to key material compromise.
	2. Key Compromise Policy Training and Rehearsals	The key compromise policy is trained and rehearsed at least annually, and the results, improvement items, and next scheduled rehearsal are documented.

Appendix7.
JVCEA Crypto Asset Security Management Standard

To prevent unauthorized transactions, the Crypto Asset Security Management Standard describes the control framework that exchangers should have in place in eight major risk areas, from key management to transaction management, API, external connection, and automation.

Overview of JVCEA Crypto Asset Security Management Standard

What is Crypto Asset Security Management Standard?

- The Crypto Asset Security Management Standard (Version as of June 2025) is a management standard for members of the Japan Virtual and Crypto assets Exchange Association (JVCEA) that organizes the key points of security management requirements for crypto-asset exchangers into eight areas A to H.
- To prevent the unauthorized leakage, loss, and manipulation of crypto assets, this standard covers the management of important information such as signature keys/seeds, as well as the risks associated with transaction control, vulnerability countermeasures, APIs, external connections, application distribution, and automation.

	Area	Summary
A	Critical information management	Clarify management methods for generation, storage, and backup of important data necessary for transfer, such as signature keys and seeds, to prevent unauthorized leakage and loss.
B	Withdrawal control of crypto assets	Prevent unauthorized leakage and loss by monitoring, checking, distributing authority, and off-line operation at the time of withdrawal of crypto assets.
C	Deposit control of crypto assets	Prevent unauthorized manipulation of asset data by establishing operations such as monitoring, checking, and setting the number of approvals at the time of crypto-currency deposit.
D	Specification changes and vulnerability response	Formulate and periodically review policies for informing customers, restricting/suspending transactions, and determining resumption of transactions in preparation for announcements of changes in specifications and the possibility of attacks.
E	API Management	Establish management such as authentication, authorization, invalidation, separation, testing, and monitoring according to the use and risk when API is provided
F	External connection management	When external network connections are required, ensure a connection environment that meets the risks in terms of availability, confidentiality, and integrity.
G	App. distribution management	Specify the presence or absence of official applications, etc., and take appropriate safety measures to prevent the introduction of unauthorized applications, etc.
H	Risks associated with automation	Since errors, inconsistencies, and unintended transactions may occur due to unexpected defects due to automation of business processes, etc., detection and recovery procedures, etc. shall be considered from the design and test stages.

Reference: Crypto Asset Security Management Standard (JVCEA) As of March 2026

Appendix8. Outsourcing management

It is important to raise the level of requirements for external service providers of critical operations and consider taking additional measures based on an understanding of the risks of outsourced operations.

Challenges and countermeasures in outsourcing management

Challenges at the external service providers (Example case)

- External service providers' security management may not meet expectations despite risks and criticality of their business.
  - External service providers in developing industries may not have controls equivalent to those of financial institutions or crypto-asset-related business operators in areas such as risk management, asset management, and customer information management.
  - In the absence of clear regulations or standards for such providers on information security and cybersecurity management, or clear visibility into their compliance status, technology adoption and service development may proceed ahead of control development.

- Crypto-asset-related business operators should select external service providers that have developed the necessary control environment according to the risks and importance of outsourced operations.

(Sample viewpoints)

  - Whether external service providers handling crypto assets have controls equivalent to those of crypto-asset-related business operators.
  - Whether external service providers operate controls based on standards and external assessments, pursue continuous improvement, and have experienced personnel.
  - Whether external service providers properly manage their own outsourcing.
  - Whether contracts require external service providers to provide sufficient information for evaluating outsourced operations and to meet other required conditions.

Challenges at the Crypto-asset-related business operators (Example case)

- Crypto-asset-related business operators may not fully understand the actual conditions of their external service providers.
  - Crypto-asset-related business operators may rely on external service providers' capabilities without fully understanding their systems, technologies, and controls or exhaustively identifying relevant risk scenarios.
  - Crypto-asset-related business operators may lack a risk control framework to prevent breaches at external service providers from impacting their own services.

- Crypto-asset-related business operators should deepen their understanding of external service providers and take effective measures based on risk assessment.

(Sample viewpoints)

  - Deeper understanding and evaluation of information security and cybersecurity management
  - Awareness of major or non-recurring system changes at external service providers, and assessment of their impact
  - Strict checking of API responses and isolation of high-risk anomalies
  - Vulnerability detection based on SBOM and other vendor-provided information, and mitigation measures based on the severity of threats
  - Mechanism to verify the authenticity of transactions independent of external service providers, and strict checking processes

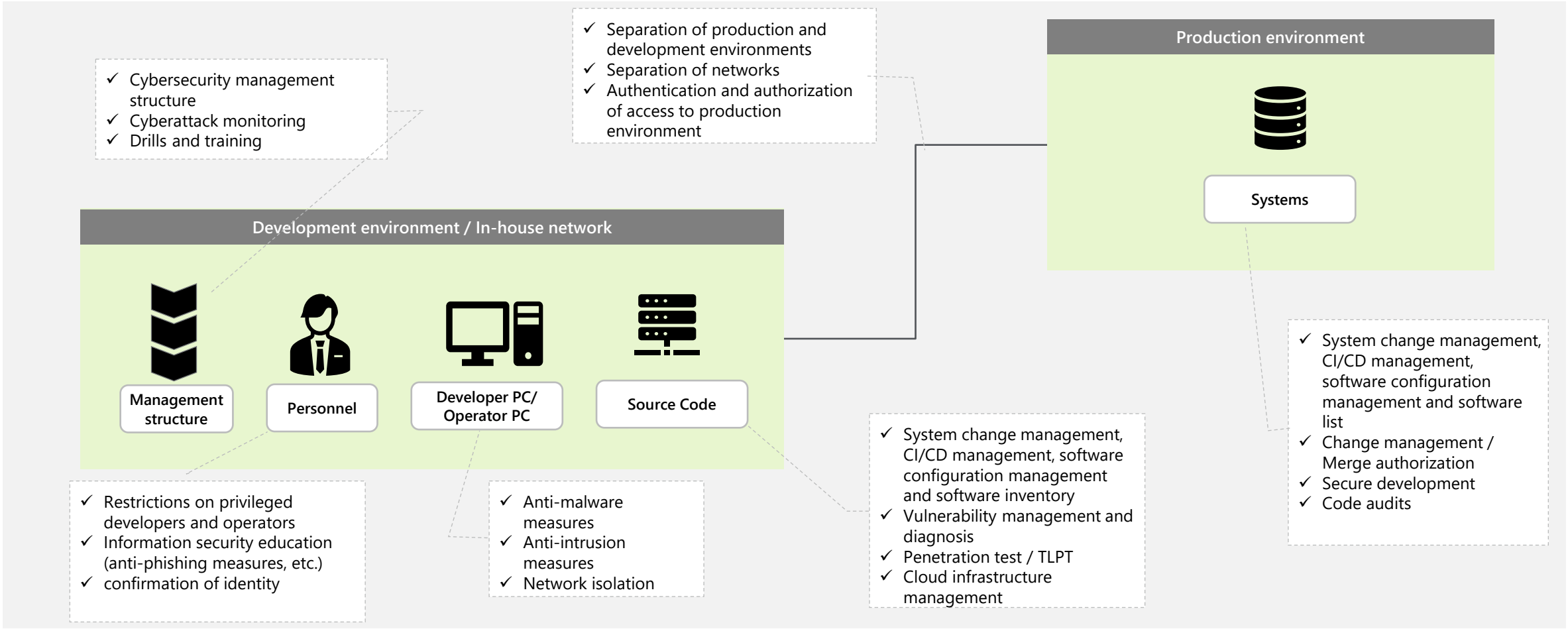
It is considered important to develop a management structure that enables effective control and monitoring over external service providers throughout the life cycle of outsourcing management.

Outsourcing management lifecycle and key points (Example)

	Key Points	Examples of Control Measures
Outsourcing management structure	Development of outsourcing management structure	Establishment of a three-line defense system, support by IT and legal departments, and development of internal policies, procedures and ledgers
	Risk-based outsourcing management	Risk assessment based on a thorough understanding of outsourced operations and confirmation according to its importance
Evaluation and selection of external service providers	Information security and cybersecurity management	Measures against cyberattacks, phishing and malwares, and separation of environment and network
	Technical capabilities and IT controls	Expertise, experience, secure development and operation, authentication and access management, and incident response
Monitoring and auditing	Implementation of monitoring	Establishment of monitoring items and methods according to risk and importance
	Confirmation through external audits, etc.	Confirmation of the results of external audits, third-party assessment reports, smart contract audits, etc. and evaluation of their validity and uncovered risk areas
Contracting	Confirmation at the time of conclusion and termination of contract	Security measures, reporting and disposal of personal information in the event of an incident, and access blocking
Incident response	Development of incident management	Education and training in normal times, confirmation of incident response procedures, and conduction of communication and reporting in the event of an incident

For critical operations involving wallets, personal information management, etc., crypto-asset-related business operators are expected to understand external service providers' system environments and conduct deeper assessments of their cybersecurity management.

Review of information security and cybersecurity management for external service providers (Example)



Crypto-asset-related business operators should consider whether the necessary security requirements are included in the contract with their third-party service providers.

Confirmation at the time of contract conclusion

■ TPSP(Third-Party Service Providers) arrangements should be governed by legally binding written agreements that clearly describe the rights, obligations, responsibilities and expectations of all parties.

1. Considerations for all contracts / Items to be included in contracts for critical TPSP arrangements

- KPI
- Right of the company to receive accurate, comprehensive and timely information
- Rights relating to TPSPs and key nth parties
- Company's right to access (including facilities), audit, and obtain relevant information from third parties
- Supervisors' right to access (including facilities), audit, and obtain information from third parties (Provided, however, that the laws and regulations of each jurisdiction)
- Obligations and Responsibilities for Business Continuity and Disaster Recovery
- Cost structure
- Ownership, access, and use of logical assets (data, etc.) and physical assets and ease of transfer of rights in a timely and appropriate manner, including at the time of termination
- Obligations and responsibilities related to security, resilience and other technical configurations
- Locations (countries, regions, etc.) where business operations and related data are processed and stored
- Confidentiality of proprietary and strategic information held by the Company and use of confidentiality agreements
- Addressing the risk that corporate information may be mixed with other third-party customer information
- Company's right to be compensated in certain circumstances
- Customer complaint handling and dispute resolution system
- Governing law and jurisdiction in case of dispute (Prefer to apply the laws of the company's domicile or business jurisdiction, if possible)
- Default and termination conditions
- Framework for modifying existing arrangements for reasons such as changes in regulatory and supervisory requirements
- Provision to support the company's exit strategy in case of termination

2. Additional items to be included in important contracts

- Primary party terms (Examples: advance notification of use or change, incident reporting)
- Additional KPI metrics and measurement methods (Examples: SLAs and service standards, BCP test results, control effectiveness test results, customer complaint information)
- The right of the company to receive the information contained in the SLA in an accurate, comprehensive and timely manner (Includes incidents and significant changes to third-party and major party services)
- Company's right to access, audit, and obtain relevant information from key nth parties
- Right of supervisory authorities to access, audit and obtain information from key nth parties (Provided, however, that the laws and regulations of each jurisdiction)
- Obligations and responsibilities for business continuity and disaster recovery plans (Includes minimum uptime, maximum downtime, RTO, and RPO)

Reference: Basel Committee on Banking Supervision, Principles for the sound management of third-party risk, December 2025.

Crypto-asset-related business operators should also consider exit plans and strategies for the planned termination of TPSP arrangements.

Confirmation at the time of termination

3. Handling at the time of termination

- In an exit plan for the termination of TPSP arrangements, the following should be considered.
  ① Transition period
  ② Integrity of contractual rights
  ③ Appropriate budget allocation
  ④ Identification of areas of responsibility

- Exit plans for critical TPSP arrangements may include, but are not limited to, the following:
  ① Process for timely and appropriate transfer of logical assets (data, etc.), physical assets, and human resources
  ② Measures necessary for coordination with all stakeholders

- An exit strategy for unplanned termination should be maintained on an appropriate and proportionate basis for all TPSP arrangements, based on reasonable scenarios and reasonable assumptions, and taking into account the importance and substitutability of the services provided by TPSP.

- Exit strategies for critical TPSP arrangements should include:
  ① Asset transfer process
  ② Periodic update of emergency response personnel
  ③ Process for budget approval to secure additional costs.

Reference: Basel Committee on Banking Supervision, Principles for the sound management of third-party risk, December 2025.