SERAPH

BY HALBORN



// MARCH 2025

HALBORN.COM

How Seraph Offers Protection Against Bybit-Style Hacks

INTRODUCTION

In February 2025, cryptocurrency exchange Bybit was the victim of the most expensive known hack in history. The previous largest hack in the crypto space — the 2022 hack of the Ronin Network — had a price tag of “only” $624 million, compared to Bybit’s astronomical $1.4 billion.

Historically, most major DeFi hacks have exploited protocols that failed to implement security best practices. For example, most hacks exploited smart contracts that were not properly audited before being deployed on-chain.

Bybit is one of several hacks breaking this mold where sophisticated threat actors exploited protocols that went above and beyond the minimum for security. Bybit was using multi-signature wallets implemented with Safe and had signing keys stored on hardware wallets.

The Lazarus Group - a notorious state-sponsored North Korean APT (Advanced Persistent Threat) group - evaded these defenses via a social engineering attack, tricking signers into blindly approving a malicious upgrade to the project’s cold wallet contract. Once approved, this modification implemented malicious sweepETH and sweepERC20 functions that allowed them to steal $1.4 billion in ETH, stETH, cmEYH, and mETH from the contract.

BREAKING DOWN THE BYBIT ATTACK

The Bybit hack was the result of a sophisticated attack by the Lazarus Group targeting Safe and Bybit. Let's break down the details of the incident to highlight the key events and failures that made the theft possible:

Exploiting Blind Signing Vulnerabilities in Multi-Sig Wallets

One of the root causes of the Bybit hack was "blind signing." This occurs when the various signers of a multi-sig wallet review and approve a transaction without understanding its purpose and functionality.

In Bybit's case, the Lazarus Group was aware that Bybit periodically made transfers from its cold to hot wallets to top up these wallets. The attackers waited for and modified one of these transactions containing an estimated 30,000 ETH to include the malicious functionality, making the fraudulent transfer seem plausible.

How Attackers Compromise Safe's AWS Infrastructure to Serve Malicious Code

Bybit uses Safe to implement its multi-sig wallet. The various signers would view the transaction's contents within the Safe user interface and trigger approval and signing from there.

Investigation of the Bybit hack revealed that the Safe user interface code hosted on AWS was compromised and used to serve a malicious version of app.safe.javascript. This modification was the result of a phishing attack on a Safe developer.

This modified code was designed specifically to target Bybit, only triggering for transactions that were intended for the Safe address and had a default (0) transaction operation. If the source of the transfer was the attacker's address or the Bybit proposal wallet, the page was reloaded. This design prevented the malicious code from accidentally triggering and revealing itself for anything except the target transaction.

Manipulating Transaction Data within the Safe User Interface

Once the malicious JavaScript was deployed on Safe's AWS, the attackers waited for a transaction that met their requirements. When a transfer from the cold wallet to the hot wallet was approved, the transaction modified its contents within the Safe interface.

The goal of this modification was to trick the signers into approving different transaction data than what they were seeing. The malevolent code saved the original transaction data, then modified the transaction to point to the attacker's address and include a malicious payload.

This insidious payload was designed to configure the target contract to perform a delegatecall to the attacker's malicious contract by injecting that code into the Storage[0x0] storage location in the Gnosis Safe multisig proxy contract. This performed a malicious upgrade and enabled the sweepETH and sweepERC20 backdoors used to steal funds. The malicious JavaScript code then used the built-in signTransaction or executeTransaction functions to sign or execute the malicious code.

Hiding Malicious Modifications by Masking the Safe Wallet Interface

The Bybit hack was made possible by a mismatch between what the signers were seeing and what happened behind the scenes. When the malicious JavaScript inserted into the Safe UI identified a matching transaction, it hijacked that transaction for its own purposes behind the scenes.

However, the UI was masked to ensure that everything that the signer saw appeared legitimate. The original transaction data was saved by the JavaScript before modifying it and was restored once the malicious operations were complete — successfully or otherwise. When the signers reviewed the transaction data and the Safe URL, everything looked legitimate because the malicious code was being served from Safe's AWS environment.

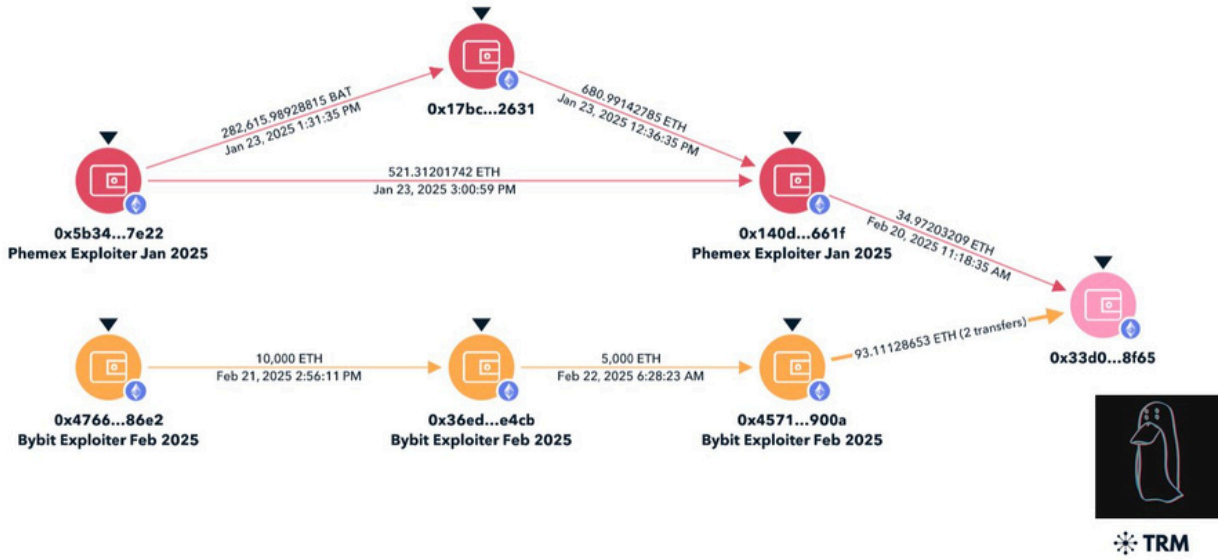
Restoring the original transaction data in the result and transaction ensured that any subsequent processing performed on the transaction was done on the "clean" data. However, the signed transaction was the nefarious one.

Draining Funds: The $1.4 Billion Token Theft

The objective of the fraudulent transaction was to insert hidden sweepETH and sweepERC20 functions into Bybit's Safe smart contract. Once these backdoors were inserted, the attacker was able to call these functions to drain the cold wallet of its funds.

In total, the attacker stole an estimated $1.4 billion in ETH and various ERC tokens. These tokens were then laundered through thousands of addresses across multiple blockchains, according to an FBI advisory.

Later analysis by the renowned crypto security analyst and Paradigm adviser ZachXBT linked the Bybit hack to the September 2024 hack of BingX and the January 2025 Phemex hack, indicating that the Lazarus Group was behind the theft. The image below shows that funds stolen in the two incidents were eventually sent to the same addresses, indicating that the same party was behind both.



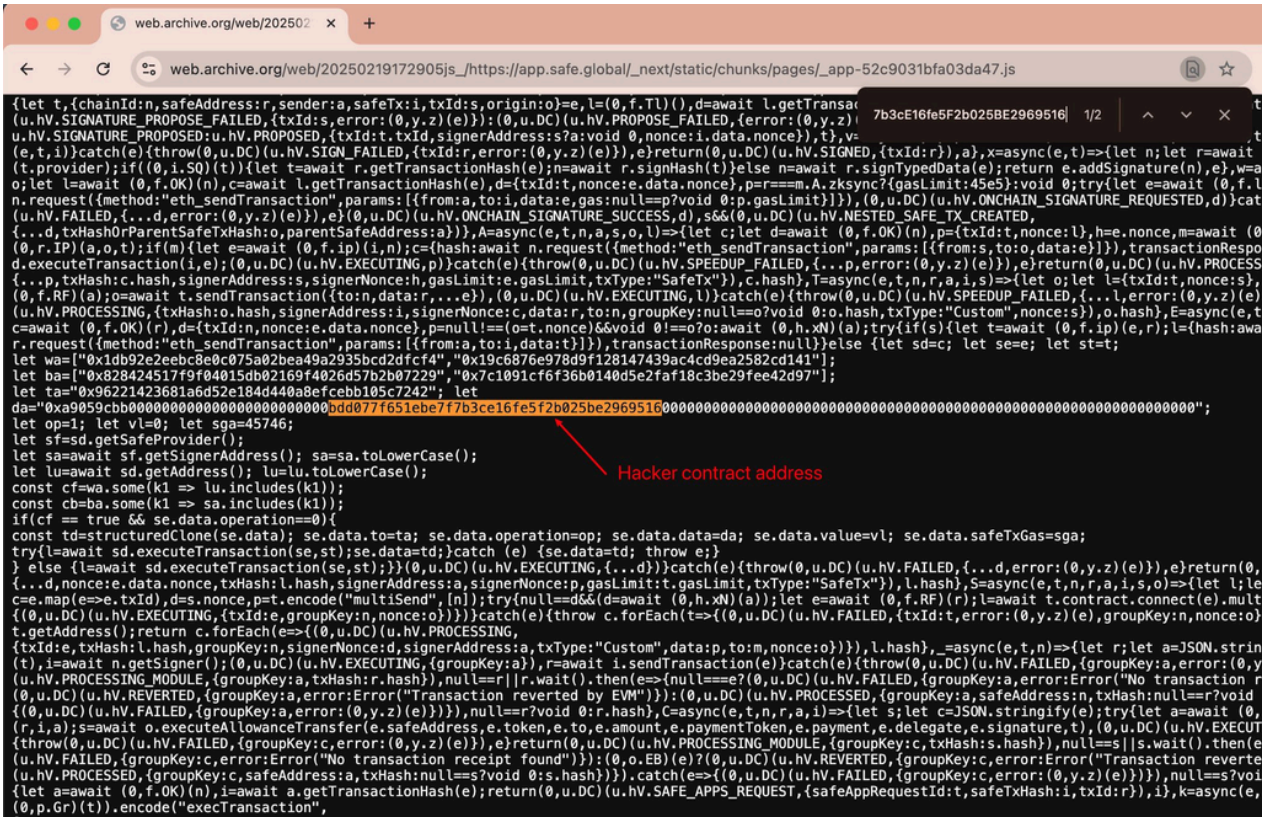
Transaction graph linking the Phemex, BingX, and Bybit hacks (Source: ZachXBT)

HALBORN

CRITICAL FAILURES AND VULNERABILITIES LEADING TO THE BYBIT HACK

The $1.4 billion theft from Bybit was the result of a sophisticated attack by the Lazarus Group. The threat actor gained access to Safe's AWS environment via a compromised developer machine and temporarily replaced the Safe app with a modified variant. This code would only execute under certain circumstances and was replaced by a clean version minutes after the attack in an attempt to cover the attackers' tracks.

The hack was made possible by the fact that the attackers had full control over what the signers could see on an untrusted, "hot" device in the Safe UI. Within the UI, everything that the signers could see included the original transaction data. The malicious JavaScript running behind the scenes is shown in the figure below.



Malicious JavaScript code retrieved from Wayback Machine

The signers had limited ability to validate the contents of the transaction on their hardware wallets, so the switch went undetected. As a result, the modified transaction data sent to these devices for signing went undetected. Once signatures were applied to the malicious transaction, the attackers had everything that they needed.

EVALUATING SECURITY COUNTERMEASURES AND THEIR LIMITATIONS IN CRYPTO SYSTEMS

Bybit was the victim of the largest hack to date despite implementing strong security. Their use of multi-sig wallets and hardware wallets via Safe eliminated many of the vulnerabilities that lead to off-chain hacks.

Despite this, a sufficiently sophisticated and patient attacker — the Lazarus Group — found a hole in their defenses. In the wake of the attack, several solutions were proposed that may have prevented it. However, most of them wouldn't have actually helped.

The Role of Dedicated Hardware Devices in Strengthening Crypto Security

One common suggestion for mitigating the Bybit hack is that the signers should have used dedicated devices for signing. The rationale is that these devices have a lower probability of being infected with malware that would allow this type of attack to be performed.

While this is true, these devices would need to interact with external systems to receive transactions and submit them for inclusion on the blockchain. Additionally, they would need to undergo regular updates to eliminate vulnerabilities that could be exploited during these times. A sufficiently motivated and sophisticated attacker may have been able to gain a foothold on these devices as a result.

In this case, the security of the devices used for signing is irrelevant since the compromise occurred within Safe's AWS environment. A brand-new laptop turned on for the first time to perform this transaction would have been equally vulnerable to the attack.

How Communication Breakdowns Contribute to Transaction Vulnerabilities

The root cause of the Bybit hack was blind signing or approving a transaction without understanding its full purpose. This is due to the limited visibility provided by a hardware wallet and — potentially — a lack of communication among team members regarding the purpose of the transaction.

However, if the attacker controlled either the signers' devices or Safe's application (which was the case here), team communication provides no real benefit or protection. The Lazarus Group waited for a legitimate transaction to come through that they nefariously modified while masking the Safe UI. If all of the signers were sitting next to one another and approved the transaction together, they still would have been tricked.

Preventing Unauthorized Transactions Through Sandboxed Simulation

The fatal flaw in Bybit's setup was that it lacked an environment that was both fully trusted and provided complete transaction visibility. The Safe UI used to validate the transaction's contents was compromised by the attacker. The hardware wallets used to digitally sign transactions — the real approval — didn't display all of the information required to verify that the transaction being signed was the same one as displayed on the Safe UI.

The only way to prevent hacks like the Bybit incident is to use a secure environment that validates that the signed transaction data did what the signer expected. Simulating signed transactions before posting them on-chain can allow unexpected or malicious side effects to be detected and provide an opportunity for a fraudulent transaction to be blocked before it does damage.

HALBORN'S SERAPH: PREVENTING HACKS VIA TRANSACTION-LEVEL AUDITS

Halborn developed Seraph to provide the missing layer of blockchain security: preventative security. Seraph is a policy enforcement tool for smart contracts that acts as an additional line of defense, allowing signed transactions posted to the blockchain to be reverted if they violate organizational procedures.

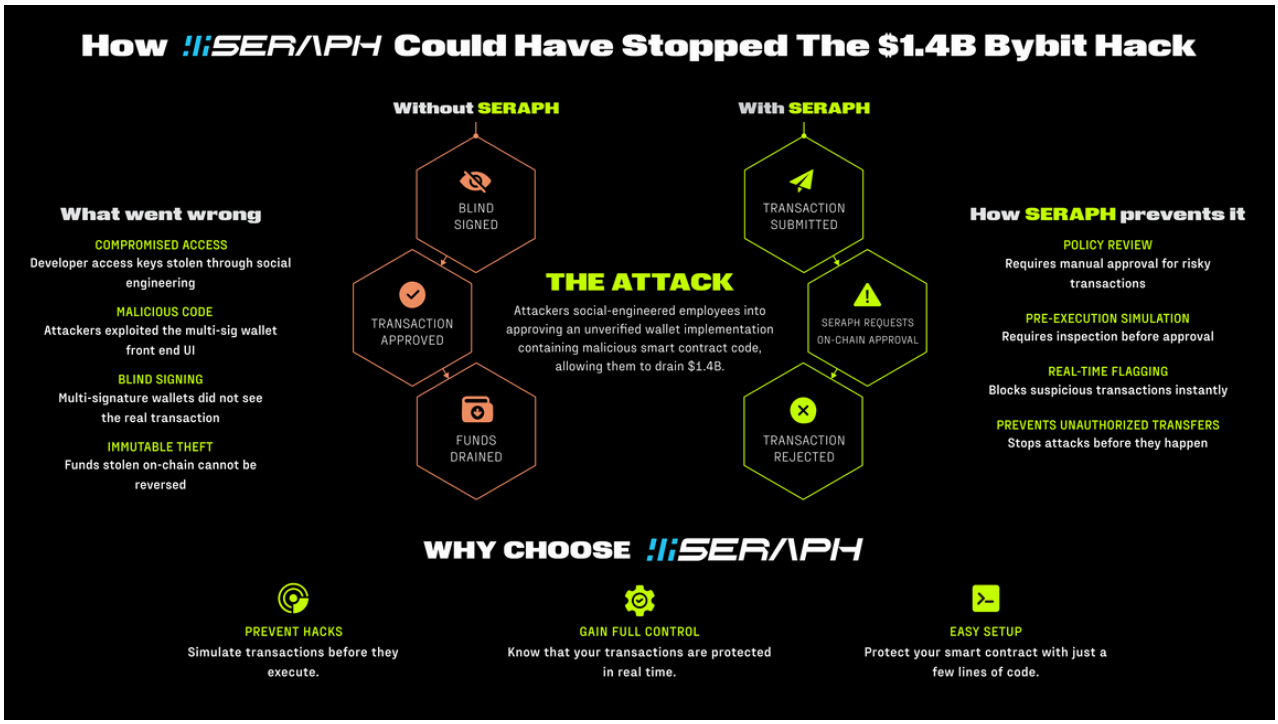
Any transactions submitted to a smart contract protected by Seraph must do so through a dedicated Seraph RPC (Remote Procedure Call) endpoint. This endpoint simulates the transaction's execution and determines if it calls any Seraph-protected functions. If so, an administrator is alerted to perform final validation and approval of the transaction before it is executed on-chain.

In the case of the Bybit hack, if the on-chain function that was called by the blind-signed transaction was protected with Seraph, this would have revealed that the nefarious transaction included an upgrade to Bybit's cold storage smart contract. As a result, an administrator would have been notified of this fact and asked to perform final approval (to confirm the transaction) or rejection (to revert the transaction).

Since this action would not have matched the intended function of the original transaction — performing a routine transfer of assets from cold to hot wallets — the administrator would have rejected it. This would have caused the Seraph smart contract to throw an exception on transaction execution, causing the hack to fail.

This hack was similar to the Ledger incident of 2023 when Ledger's trusted code was compromised, and a Ledger Live update was infected with malicious code. Like the Bybit hack, this was made possible due to compromised developer credentials.

Since the Lazarus Group only compromised the Safe interface in the Bybit hack, they would have no visibility into or control over the secondary Seraph approval. Halborn is a SOC2-certified security company that protects all developer credentials and access with strong multi-factor authentication. Since the Seraph GitHub is secured against malicious updates, its UI is protected against a similar masking attack. As a result, if Seraph had been deployed for the smart contracts of the Bybit platform, the fraudulent, signed Bybit transaction would have been caught and flagged, preventing a $1.4 billion hack.



How Seraph defends against Bybit-style hacks

HALBORN'S SERAPH: A PROVEN DEFENSE AGAINST CRYPTO HACKS

At its root, the Bybit hack was made possible by two things: a compromised AWS deployment and blind signing. Multi-sig and hardware wallets only provide so much protection if you're trusting the user interface on a "hot" device to accurately display transaction data.

Halborn's Seraph provides an additional, preventative line of defense against this type of threat to on-chain projects and assets: if Seraph had been deployed on Bybit, the largest hack to date would not have happened. Halborn also offers advisory services to support projects throughout the development lifecycle and help ensure compliance with security best practices and regulatory requirements.

Contact Halborn today for a demo of Seraph and learn how our transaction-level audits can secure your platform.

Sources

- Bybit Incident Investigation Preliminary Report — Verichains
- Bybit Interim Investigation Report — Sygnia
- Safe{Wallet} Statement on Targeted Attack on Bybit
- Incident Update: Unauthorized Activity Involving ETH Cold Wallet
- Bybit's $1.5 Billion Theft Unveiled: Safe{Wallet} Front-End Code Tampered — Slowmist
- ZachXBT Twitter/X
- North Korea Responsible for $1.5 Billion Bybit Hack - FBI
- Ben Zhou's Livestream on the Latest ETH Wallet Incident

HALBORN

MEET OUR TEAM

Halborn was founded in 2019 by renowned ethical hacker Steven Walbroehl and growth hacker Rob Behnke. The fully remote organization has since grown to over 100 of the best and brightest offensive security engineers in the world.



JACQUES BOSCHUNG

CEO

STEVEN WALBROEHL

CO-FOUNDER & CTO





ROB BEHNKE

CO-FOUNDER &
EXECUTIVE CHAIRMAN

HALBORN

WHY WORK WITH US?

Halborn's proactive security measures, including in-depth vulnerability assessments, are critical for maintaining trust and reliability in a multi-blockchain landscape. By incorporating proactive security methods, you can protect sensitive information while fostering a sustainable and resilient network where all participants can thrive.

For financial institutions venturing into the blockchain and DeFi space, prioritizing these security solutions is essential and can help you bridge the narrowing gap between traditional and decentralized finance securely and with confidence.

GET IN TOUCH

Sales: sales@halborn.com

Marketing: marketing@halborn.com

For more info on our services and to read client testimonials and case studies, visit **halborn.com**