$15 Billion Rugpull Vulnerability in Convex Finance protocol Uncovered and Resolved

OPENZEPPELIN SECURITY | APRIL 4, 2022

Security Audits

TLDR: In late 2021, as part of a security audit for a client, OpenZeppelin conducted a security review of the Convex Finance protocol. As part of the audit, the Security Research Team uncovered a vulnerability that, if exploited by two of three anonymous multi-signature wallet (multisig) signers, would have given the Convex multisig direct control over Convex's locked value—then approximately $15 billion. Convex documentation specifically stated such control was not possible. This vulnerability has since been patched by the Convex Team.

The vulnerability represented a potential rugpull, a new and significant threat vector in the DeFi space. As OpenZeppelin's mission is to protect the decentralized economy, the research team created a plan of action to help resolve the vulnerability with minimal risk to the funds in any possible scenario.

The Convex protocol had a serious bug

Convex—a major DeFi protocol—contained a bug that put $15B at risk of a rugpull. The bug is fixed now, and no funds were lost. But the situation provides some interesting lessons.

The Security Research Team found that if two of the three signers of the Convex multisig executed a specific series of steps, those users would be provided with unrestricted access to LP tokens staked in a target pool configured with the LP token and target gauge. (Curve uses gauges to allot financial rewards relative to a given user's contribution of liquidity.) Convex's documentation at the time, since updated, stated that this should not be possible—hence the cautious approach to resolution. The vulnerability, which was remedied via a patch on December 14, 2021, could have been exploited as follows:

1. Call the `revertControl` function of the `PoolManagerV2` contract to become the `poolManager` of the `Booster` contract
2. Add a new pool configured with the target gauge and an attacker-controlled LP token
3. Add another new pool configured with target LP token and an attacker-controlled gauge
4. Deposit to the second pool an amount of attacker-controlled LP token equal to the amount of LP tokens currently deposited in the target gauge
5. Call the `withdrawAll` function of the `Booster` contract to withdraw from the first pool: this will withdraw all the LP tokens from targeted gauge and leave them in the `CurveVoterProxy` contract
6. Deposit at least 1 LP token in the second pool by calling the deposit function of the Booster contract: this will approve the entire targeted LP token balance held by the `CurveVoterProxy` to the fake gauge and call a deposit function on it which can be used to drain the `CurveVoterProxy`'s balance

Bug disclosure was complicated due to an anonymous counterpart

If a security researcher finds a protocol vulnerability that can only be exploited (or patched) by that protocol's developer team—and that team is anonymous—what is the best way to disclose the vulnerability? The answer is not so simple.

DeFi protocol vulnerabilities—of varying severities—are not uncommon. A key tenet of the open-source philosophy is the expectation that developers who find them will step forward, disclose them, and in some cases claim a reward for doing so. This dynamic is a key source of the battlethe case here.

The dynamics of contacting anonymous teams about issues can be complex. In many cases, a vulnerability in open-source software can be exploited by anyone who finds it. In this specific instance, however, the vulnerability could only be exploited—or patched—by Convex's anonymous developers.

From the outset, OpenZeppelin's analysis of the code (on behalf of a client) and the effort required by Convex to exploit it gave the Security Research Team a high degree of confidence that the vulnerability was unintentional. However, at the time, this could not be known with absolute certainty. Moreover, even if it were unintentional and Convex was unaware of it, disclosure created a perverse incentive for Convex's developers with $15 billion on the line. There was reason to believe that Convex's developers were good-faith actors, but the potential costs of being wrong in this belief were astronomical.

From OpenZeppelin's point of view, its concerns could be alleviated if the identities of Convex's developers were known. Convex, on the other hand, was faced with the legitimate security concerns associated with potential loss of its anonymity. For these reasons, both parties had strong incentives to be cautious.

The incentive structure outlined above left the Security Research Team with three theoretical courses of action.

1. **Disclose the vulnerability details to the Convex team.** If the vulnerability had been intentional, this course of action would have risked prompting Convex's anonymous developers to execute the rugpull. Even the discovery of an unintentional vulnerability posed risks, however. Convex's anonymity meant that sharing the vulnerability's details created a significant perverse incentive for its developers. If disclosure led to a rugpull, OpenZeppelin would have had some degree of culpability.
2. **Disclose the vulnerability details publicly.** Some might argue that the Security Research Team should have provided the community with information and allowed Convex users to make decisions based on their own judgment of the risks involved. OpenZeppelin believes, however, that this course of action would have been irresponsible. If the vulnerability had

second course of action, public disclosure also created perverse incentives for Convex's developers.

3. **Attempt to obtain assurances that the vulnerability would not be exploited and then disclose it to the Convex team.** This course of action presented the greatest number of difficulties for both parties, but was the only available option that could potentially prompt a patch while hedging against a rugpull.

After considering these options, the Security Research Team chose Option 3. They determined that the optimal approach was to reach out to bug bounty partner Immunefi for an introduction to an intermediary between OpenZeppelin and Convex.

Disclosure was conducted with the help of publicly known parties to reduce risk

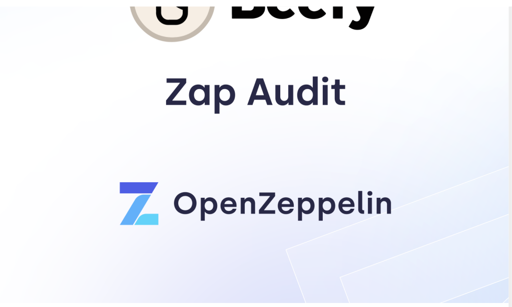
The addition of publicly known persons to the Convex multisig was a key breakthrough.

OpenZeppelin Security Research Team along with Convex anonymous developers agreed that the best course of action to this dilemma was to incorporate additional publicly known parties to the multisig, making a rugpull impossible.

At this point, the Security Research Team commenced open communication with Convex, providing full vulnerability details and a testing method. Shortly thereafter, Convex patched the vulnerability.

OpenZeppelin's mission is to grow and protect the open economy. To achieve this mission, OpenZeppelin provides the Contract library, the gold-standard for smart contract development, security audits, and Defender, the trusted Security Operations platform for Web3. If you want to be part of the team that is securing the future of Web3, we are hiring!

Related Posts



Beefy Zap Audit

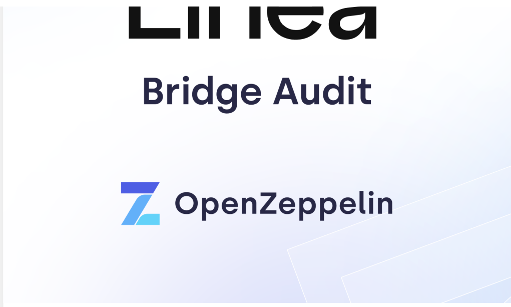
BeefyZapRouter serves as a versatile intermediary designed to execute users' orders through routes...

Security Audits

OpenBrush Contracts Library Security Review

OpenBrush is an open-source smart contract library written in the Rust programming language and the...

Security Audits



Linea Bridge Audit

Linea is a ZK-rollup deployed on top of Ethereum. It is designed to be EVM-compatible and aims to...

Security Audits

© Zeppelin Group Limited 2023

Privacy | Terms of Use