0x Protocol

Smart Contract Security Assessment

March 13, 2025

ABSTRACT

Dedaub was commissioned to perform a security audit of a Gnosis Safe guard developed for the 0x protocol. The guard implements a timelock mechanism, ensuring owners have sufficient time to respond to potentially malicious transactions.

Overall we found **no issues** with the contract.

BACKGROUND

Gnosis Safe is one of the most widely used multisignature smart contract wallets in production today, allowing multiple users to collectively manage assets securely. A Gnosis Safe Guard is an optional security layer designed to enforce additional pre- and post-transaction conditions, ensuring heightened security.

The audited guard specifically implements a **timelock mechanism**, which requires transactions to be queued using the `enqueue` function. Each enqueued transaction must pass a predefined delay before it can be executed. The primary functions enforcing this logic are `checkTransaction` and `checkAfterExecution`, which verify the transaction has been properly enqueued and that the timelock has elapsed along with some additional checks to prevent the safe from ending up in a corrupted state.

This timelock approach mitigates risks by providing a grace period during which potentially malicious transactions can be identified and canceled (`cancel`), effectively protecting users from immediate compromise. Additionally, the guard restricts certain high-risk operations, explicitly disallowing `DelegateCall` operations to prevent potentially catastrophic state modifications.

SETTING & CAVEATS

The audit covers the `SafeGuard.sol` contract in pull request #253 at commit: `0d5a970a9ba5dc02f1ec64793ac3cf0705222b4e`.

The audit's main target is security threats, i.e., what the community understanding would likely call "hacking", rather than the regular use of the protocol. Functional correctness (i.e. issues in "regular use") is a secondary consideration. Typically it can only be covered if we are provided with unambiguous (i.e. full-detail) specifications of what is the expected, correct behavior. In terms of functional correctness, we often trusted the code's calculations and interactions, in the absence of any other specification. Functional correctness relative to low-level calculations (including units, scaling and quantities returned from external protocols) is generally most effectively done through thorough testing rather than human auditing.

VULNERABILITIES & FUNCTIONAL ISSUES

This section details issues affecting the functionality of the contract. Dedaub generally categorizes issues according to the following severities, but may also take other considerations into account such as impact or difficulty in exploitation:

Category	Description
CRITICAL	Can be profitably exploited by any knowledgeable third-party attacker to drain a portion of the system's or users' funds OR the contract does not function as intended and severe loss of funds may result.
HIGH	Third-party attackers or faulty functionality may block the system or cause the system or users to lose funds. Important system invariants can be violated.
MEDIUM	Examples: • User or system funds can be lost when third-party systems misbehave. • DoS, under specific conditions. • Part of the functionality becomes unusable due to a programming error.
LOW	Examples: • Breaking important system invariants but without apparent consequences. • Buggy functionality for trusted users where a workaround exists. • Security issues which may manifest when the system evolves.

Issue resolution includes "dismissed" or "acknowledged" but no action taken, by the client, or "resolved", per the auditors.

CRITICAL SEVERITY:

[No critical severity issues]

HIGH SEVERITY:

[No high severity issues]

MEDIUM SEVERITY:

[No medium severity issues]

LOW SEVERITY:

[No low severity issues]

OTHER / ADVISORY ISSUES:

This section details issues that are not thought to directly affect the functionality of the project, but we recommend considering them.

ID	Description	STATUS
A1	Compiler bugs	**INFO**
The code is compiled with Solidity `0.8.25`, which as of the time of writing has no known issues.		

DISCLAIMER

ABOUT DEDAUB

Dedaub offers significant security expertise combined with cutting-edge program analysis technology to secure some of the most prominent protocols in DeFi. The founders, as well as many of Dedaub's auditors, have a strong academic research background together with a real-world hacker mentality to secure code. Protocol blockchain developers hire us for our foundational analysis tools and deep expertise in program analysis, reverse engineering, DeFi exploits, cryptography and financial mathematics.