Immunefi

Vault Safe

Security Assessment & Correctness

February 13th, 2023

Audited By:
Angelos Apostolidis
angelos.apostolidis@ourovoros.io
George Delkos
george.delkos@ourovoros.io

Overview

Project Summary

Project Name	Immunefi - Vault Safe
Website	Immunefi
Description	Bug Bounty Vaults
Platform	Ethereum; Solidity, Yul

Codebase	GitHub Repository
Commits	3883128ea79f23d09e8e1d5be58f0f525fa505a9

Audit Summary

Delivery Date	February 13th, 2023
Method of Audit	Static Analysis, Manual Review

Vulnerability Summary

Total Issues	4
Total Major	0
Total Minor	2
Total Informational	2

Files In Scope

Contract	Location
src/Withdrawable.sol	https://github.com/immunefi-team/vault-safe-poc-contracts/tree/3883128ea79f23d09e8e1d5be58f0f525fa505a9/src/Withdrawable.sol
src/Splitter.sol	https://github.com/immunefi-team/vault-safe-poc-contracts/tree/3883128ea79f23d09e8e1d5be58f0f525fa505a9/src/Splitter.sol

Findings

ID	Title	Type	Severity
F-1	Ambiguous event definition	Gas Optimization	informational
F-2	Usage of `transfer()` for sending Ether	Volatile Code	minor
F-3	Inexistent input sanitization	Volatile Code	minor
F-4	Redundant use of `virtual`	Coding Style	informational

F-1: Ambiguous event definition

Type	Severity	Location
Gas Optimization	informational	Withdrawable L20, L38

Description:

The `LogWithdraw` event includes data unrelated to the ERC-20 token standard, i.e. a `tokenId`, leading to static data logging during the said event's emission.

Recommendation:

We advise to remove the `tokenId` parameter from the `LogWithdraw` event.

F-2: Usage of `transfer()` for sending Ether

Type	Severity	Location
Volatile Code	minor	Withdrawable L33

Description:
After EIP-1884 was included in the Istanbul hard fork, it is not recommended to use `.transfer()` for transferring ether as these functions have a hard-coded value for gas costs making them obsolete as they are forwarding a fixed amount of gas, specifically `2300`. This can cause issues in case the linked statements are meant to be able to transfer funds to other contracts instead of EOAs.

Recommendation:
We advise that the linked `.transfer()` calls are substituted with the utilization of the `sendValue()` function from the `Address.sol` implementation of OpenZeppelin either by directly importing the library or copying the linked code.

F-3: Inexistent input sanitization

Type	Severity	Location
Volatile Code	minor	Splitter L37, L68, L101, L110

Description:
The linked code expressions fail to check the address-based values against the zero address.

Recommendation:

We advise to add `require` statements, checking the aforementioned values against the zero address.

F-4: Redundant use of `virtual`

Type	Severity	Location
Coding Style	informational	Splitter L127

Description:
The linked function contains the `virtual` keyword without the intention of being extended.

Recommendation:
We advise to remove the `virtual` keyword from the linked function.

Disclaimer

Reports made by Ourovoros are not to be considered as a recommendation or approval of any particular project or team. Security reviews made by Ourovoros for any project or team are not to be taken as a depiction of the value of the “product” or “asset” that is being reviewed.

Ourovoros reports are not to be considered as a guarantee of the bug-free nature of the technology analyzed and should not be used as an investment decision with any particular project. They represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Each company and individual is responsible for their own due diligence and continuous security. Our goal is to help reduce the attack parameters and the high level of variance associated with utilizing

new and consistently changing technologies, and in no way claim any guarantee of security or functionality of the technology we agree to analyze.