Smart Contract Audit Conclusion

by Vinh Le

2th October 2022

This is the review of Vinh Le ’s Multisig-wallet repo. No critical bug found, but there exists several issues.

Findings

ID	Severity	Subject
CVF-1	High	Contract locking ether
CVF-2	Medium	Reentrancy
CVF-3	Medium	Improper Solidity version
CVF-4	Medium	Improper approach
CVF-5	Minor	Different pragma directives are used
CVF-6	Minor	Missing zero address validation
CVF-7	Minor	Function declaration

Contents

1. Document properties

Version

Version	Date	Author	Description
0.1	Aug. 24, 2022	Vinh Le	Initial Draft

Contact

Vinh Le

minhthangminh1992@gmail.com

2. Introduction

The following document provide the results of the audit performed by Vinh Le Consulting. The audit goal is a general review of the smart contracts structure, critical/major bugs detection and issuing the general recommendations.

We have audited the Multisig-wallet Concretely, the following file was audited:

- packages\hardhat\contracts\MultiSigFactory.sol
- packages\hardhat\contracts\MultiSigWallet.sol

3. Detailed Results

3.1 CVF-1 Contract locking ether

- Severity High
- Category Withdraw

Description Contract with a payable function, but without a withdrawal capacity.

Recommendation Remove the payable attribute or add a withdraw function.

Listing 1: Contract locking ether

Contract MultiSigFactory (contracts/MultiSigFactory.sol#12-136) has payable functions:

- MultiSigFactory.create2(uint256,address[],uint256,string)

(contracts/MultiSigFactory.sol#67-112)

But does not have a function to withdraw the ether

3.2 CVF-2 Reentrancy

- Severity Medium
- Category Suboptimal

Description Detection of the reentrancy bug.

Recommendation Apply the check-effects-interactions pattern.

Listing 2: Reentrancy

- multiSig.init(_chainId,_owners,_signaturesRequired) (contracts/MultiSigFactory.sol#98)

State variables written after the call(s):

- multiSigs.push(multiSig) (contracts/MultiSigFactory.sol#100)

3.3 CVF-3 Improper Solidity version

- Severity Medium
- Category Suboptimal

Description Pragma version^0.8.13 (src/CharityFactory.sol#2) necessitates a version too recent to betrusted.

Recommendation Consider deploying with 0.6.12/0.7.6/0.8.7

Listing 3: Improper Solidity version

```
2 pragma solidity ^0.8.13;
```

3.4 CVF-4 Improper approach

- Severity Medium
- Category Suboptimal

Description Low level call in Cutie.withdraw(). The use of low-level calls is error-prone. Low-level calls do not check for code existence or call success.

Recommendation Avoid low-level calls. Check the call success. If the call is meant for a contract, check for code existence.

Listing 4: Improper approach

(success,result) = to.call{value: value}(data) (contracts/MultiSigWallet.sol#211)

3.5 CVF-5 Different pragma directives are used

- Severity Minor
- Category Suboptimal

Description Different versions of Solidity are used throughout the OpenZepellin dependencies and src/Badge.sol, src/Charity.sol
Recommendation Use one Solidity version.

Listing 5: Different pragma directives are used

Version used: ['>=0.8.0<0.9.0', '^0.8.0']

3.6 CVF-6 Missing zero address validation

- Severity Minor
- Category missing-zero-check

Description Detect missing zero address validation.

Recommendation Check that the address is not zero.

Listing 6: Missing zero address validation

(success,result) = to.call{value: value}(data) (contracts/MultiSigWallet.sol#211)

3.7 CVF-7 Function declaration

- Severity Minor
- Category Suboptimal

Description Public functions that are never called by the contract should be declared external to save gas.

Recommendation numberOfOwners() should be declared external:

Listing 7: Function declaration

```
- MultiSigWallet.numberOfOwners() (contracts/MultiSigWallet.sol#259-261)
```