Security Audit Report

Poolz

28/6/2022

PREPARED FOR:
Poolz, poolz.finance

ARCADIA CONTACT INFO
Email: audits@arcadiamgroup.com
Telegram: https://t.me/thearcadiagroup

Table of Contents

Executive Summary

A Representative Party of Poolz("**CLIENT**") engaged The Arcadia Group ("Arcadia"), a software development, research, and security company, to conduct a review of the following Poolz smart contracts on the **The-Poolz/MultiSig** github repository at Commit **#79dacb75bfbad1460a639c31667cc2f2c1ea4f6b**

The scope of this audit included the following files:

1. **Multisig**.sol
2. **MultiSigConfirmer**.sol
3. **MultiSigEvents**.sol
4. **MultiSigInitiator**.sol
5. **MultiSigModifiers**.sol

The scope does not include the following files:

1. **ERC20Token**.sol
2. **TokenInterface**.sol

Arcadia completed this security review using various methods primarily consisting of dynamic and static analysis. This process included a line-by-line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities and limiters, and reference against intended functionality.

There were **3** issues found, **0** of which were deemed to be 'critical', and **0** of which were rated as 'high'.

Severity Rating	Number of Original Occurrences	Number of Remaining Occurrences
CRITICAL	0	0
HIGH	0	0
MEDIUM	0	0
LOW	0	0
INFORMATIONAL	3	0

Findings

1. Check zero address before set

Issue: **POOLZMULTISIG-1**
Severity: **INFORMATIONAL**

Target:**Multisig.sol**
Finding Type: **DYNAMIC**

https://github.com/The-Poolz/MultiSig/blob/master/contracts/MultiSig.sol#L19

When adding more zero addresses that are larger than the MinSigners, the contract cannot be executed.

Action Recommended:

Add require(Authorized[index] != address(0)) function before set.

2. Check MinSignersAmount variable

Issue: **POOLZMULTISIG-2**
Severity: **INFORMATIONAL**

Target: **Multisig.sol**
Finding Type: **DYNAMIC**

https://github.com/The-Poolz/MultiSig/blob/master/contracts/MultiSig.sol#L15
Should check the variable MinSignersAmount > 1 to make sense contract

Action Recommended:

Add require(MinSignersAmount > 1);

3. Call isMinter function

Issue: **POOLZMULTISIG-3**
Severity: **INFORMATIONAL**

Target: **MultiSigInitiator.sol**
Finding Type: **DYNAMIC**

Call the “isMinter” function to make sure the contract can mint and transfer owner of minting tokens

Action Recommended:

Add require function in InitiateMint and InitiateTransferOwnership

```
require(
          IERC20(TokenAddress).isMinter(address(this)) ,
          "Not minter"
      );
```

Conclusion

Arcadia identified issues that occurred at hash **#79dacb75bfbad1460a639c31667cc2f2c1ea4f6b**.

Disclaimer

While best efforts and precautions have been taken in the preparation of this document, The Arcadia Group and the Authors assume no responsibility for errors, omissions, or damages resulting from the use of the provided information. Additionally, Arcadia would like to emphasize that the use of Arcadia's services does not guarantee the security of a smart contract or set of smart contracts and does not guarantee against attacks. One audit on its own is not enough for a project to be considered secure; that categorization can only be earned through extensive peer review and battle testing over an extended period.