SAFETIN

AUDIT

Equal9 Timelock

August 6th, 2022

TABLE OF CONTENTS

AUDIT SUMMARY

This report was written for Equal9 in order to find flaws and vulnerabilities in the Equal9 project's source code, as well as any contract dependencies that weren't part of an officially recognized library given they were provided.

A comprehensive examination has been performed, utilizing Static Analysis, Manual Review, and Equal9 Deployment techniques. The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors
- Assessing the codebase to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
- Through line-by-line manual review of the entire codebase by industry expert

AUDIT OVERVIEW

PROJECT SUMMARY

Project name	Equal9
Description	Utility token for a blockchain incubator company, the first Dapp EqualsSport is already published, a tournament platform for competitive E-sports.
Platform	Harmony One
Language	Solidity
Codebase	https://github.com/equals9-com/eq9-token-contracts/blob/master/contracts/timelocks/TokenTimelock.sol Commit: 9e74845f015590e1c0f38ca5694b121b50018406

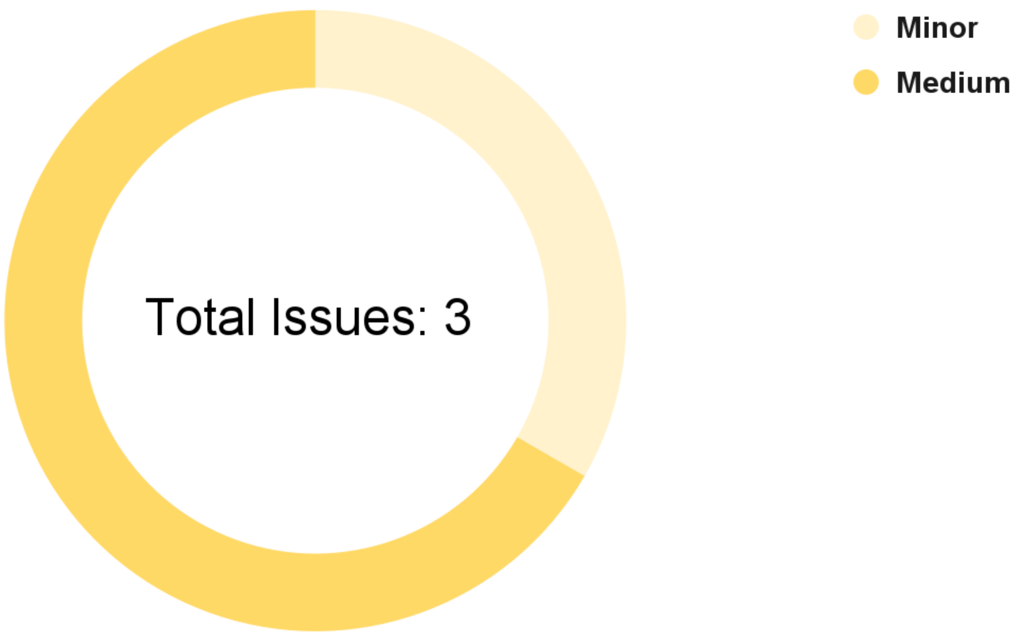
FINDINGS SUMMARY

Vulnerability	Total	Resolved
● Critical	0	0
● Major	0	0
● Medium	2	0
● Minor	1	0
● Informational	0	0

EXECUTIVE SUMMARY

There have been no critical issues related to the codebase and all findings listed here range from informational to medium. The medium security problem relates to the check-effect-interaction pattern and Array index validation.

AUDIT FINDINGS



Code	Title	Severity
RE-1	Check-Effect-Interaction pattern	● Medium
THRE-1	Array index validation	● Medium
BLOC-1	Use of block.timestamp	● Minor

RE-1 | Check-Effect-Interaction pattern

Description

Some functions within Equal9's contracts make external function calls before relevant modifying state variables. This can lead to re-entrancy where the function can be called multiple times before the completion of the first execution. This can be problematic as such multiple invocation of said functions can succeed even though they should have failed. Functions identified with this issue have been listed below.

- release -> Line: 71

Recommendation

We recommend amending this function to have the modification of the currentIndex (line 82) state variable take place before calling the safeTransfer function (line 80). As the operations within the release function are index sensitive, it may be necessary to assign a local variable equated to currentIndex before changing currentIndex.

THRE-1 | Array index validation

Description

Within the constructor (line 46) there is no validation in place to ensure that the _releaseTimes and _releaseAmounts arrays are equal to each other in length. This can break the functionality of the release (line 71) function as for example if currentIndex is 2, _releaseTimes[2] may exist but _releaseAmounts[2] may not.

Recommendation

We recommend amending this constructor by adding validation to ensure that both _releaseTimes and _releaseAmounts arrays are equal to each other in length. This can be achieved by comparing _releaseTimes.length is equal to _releaseAmounts.length in a require statement within the constructor.

BLOC-1 | Use of block.timestamp

Description

The use of block.timestamp can be problematic. The timestamp can be partially manipulated by the miner (see https://cryptomarketpool.com/block-timestamp-manipulation-attack/).

Recommendation

We fully understand that the use of block.timestamp within the Equal9 Protocol is required for certain functionality such as releasing time locked tokens. Nevertheless, it is still useful to point out this kind of potential security problem.

Global security warnings

These are safety issues for the whole project. They are not necessarily critical problems but they are inherent in the structure of the project itself. Potential attack vectors for these security problems should be monitored.

Compliance with industry standards

The way the contract is developed and its compliance with industry standards are part of the project. In order to increase the optimization of the latter, we recommend refining the code to best fit industry best practices, in particular the use of error messages and library utilization.

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement.

This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement.

This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without Safetin's prior written consent.This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Safetin to perform a security assessment.

This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance. This report should not be used in any way

Safetin security assessment to make decisions around investment or involvement with any particular project.

This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intended to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Safetin's position is that each company and individual are responsible for their own due diligence and continuous security. Safetin's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or fun.