SAFETIN

AUDIT

Equal9

July 20th, 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 [Equal9 Token Team]:
Platform	Harmony One
Language	Solidity
Codebase	https://explorer.harmony.one/address/0x598228643d6faa1b5569c3d996cb8cf8ca1fdb92

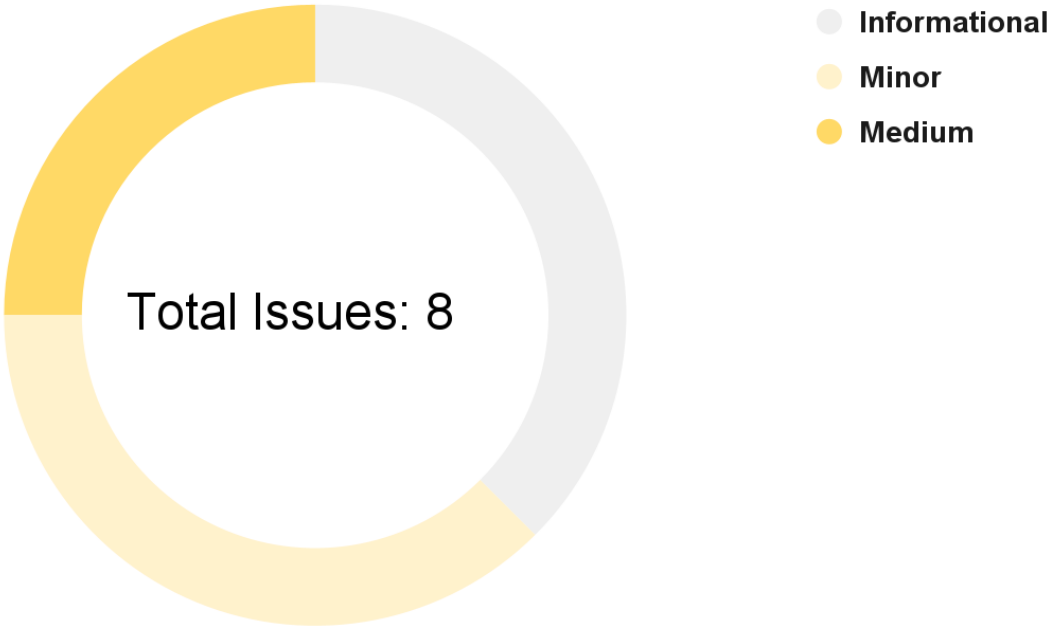
FINDINGS SUMMARY

Vulnerability	Total	Resolved
Critical	0	0
Major	0	0
Medium	2	0
Minor	3	0
Informational	3	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 centralization of token distribution, and external dependencies.

AUDIT FINDINGS



Code	Title	Severity
CENT-1	Centralization of initial token distribution	Medium
EXT-1	External protocol dependencies	Medium
GAS-1	Overly long error messages	Minor
GAS-2	Unoptimised function type	Minor
COMP-1	Unlocked compiler version	Minor
FUNC-1	Unused function	Informational
BP-1	Too many zeros	Informational

BP-2	Minor typo	Informational

CENT-1 | Centralization of initial token distribution

Description

A constructor (line 562) within the contract mints the entirety of the initial token supply to the deployer address (msg.sender). This initially centralizes token supply to the deployer address.

Recommendation

We recommend decentralizing tokens as soon as possible, matching the project's intentions. Examples of this are burning tokens or adding tokens to a liquidity pool (locked). We also recommend being fully transparent with the community about token distribution.

EXT-1 | Dependence to external protocol

Description

The contract interacts with SushiSwap protocols. The scope of the audit would treat these third party entities as black boxes and assume they are fully functional. However in the real world, third parties may be compromised thus leading assets to be lost or stolen. We fully understand that the business logic of the Equal9 project is designed to work with SushiSwap protocols. This extends to other protocols and interfaces not within the scope of this audit.

Recommendation

We encourage the team to constantly monitor the security level of the entirety of SushiSwap protocols interacted with, as the security of the project is highly dependent on the security of these decentralized exchange platforms.

GAS-1 | Overly long error messages

Description

The smart contract has some error messages that are too long. The industry standards specify error messages must have a maximum length of 32 bytes. We recommend having the short error messages within 32 bytes to optimize gas costs. Require statements with this issue have been listed below:

- line -> 300
- line -> 341
- line -> 368
- line -> 369
- line -> 374
- line -> 418
- line -> 423
- line -> 452
- line -> 453

Recommendation

We recommend shortening these error messages to be 32 characters or less in length.

GAS-2 | Unoptimized function type

Description

Throughout Equal9's contracts some functions are of type public although they are never called within the contract. External functions require significantly less gas to call. Such found functions are listed below:

- name -> Line 204
- symbol -> Line 212
- totalSupply -> Line 236
- balanceOf -> Line 243
- transfer -> Line 255
- allowance -> Line 263
- approve -> Line 274
- transferFrom -> Line 292
- increaseAllowance -> Line 320
- decreaseAllowance -> Line 339
- stake -> Line 574
- unstake -> Line 584
- amountStakes -> Line 593

Recommendation

We recommend reviewing each of the functions listed above and where possible switch their type from public to external.

COMP-1 | Unlocked compiler version

Description

Equal9's contract does not have locked compiler versions, meaning a range of compiler versions can be used. This can lead to differing bytecodes being produced depending on the compiler version, which can create confusion when debugging, as bugs may be specific to a specific compiler version(s).

Recommendation

To rectify this, we recommend setting the compiler to a single version, the version tested the most to be compatible with the code, an example of this change can be seen below.

```
pragma solidity 0.8.0;
```

FUNC-1 | Unused functions

Description

Multiple functions within Equal9's contract are defined as private or internal but are never called within the contract. This wastes contract space as there is a maximum size a contract can have. Functions found with this issue have been listed below:

- _msgData -> Line 138
- reset -> Line 541

Recommendation

We recommend removing these functions from the contract.

BP-1 | Too many zeros

Description

Some arithmetic operations within Equal9's contract contain integers with too many zeros. This can make it hard to read the code and maintain. This can lead to coding errors such as adding an extra zero or missing one during development The lines within the contract where this issue was identified has been listed below.

- Line 562

Recommendation

Such integers should be represented in standard index form by utilizing powers of 10, for example 3000000 can be represented as 3 * 10**6.

BP-2 | Minor typo

Description

A simple typo was found within the contract, in the function name amountStakes (line 593). This does not affect the contract however it's worth mentioning.

Recommendation

We recommend remaining the amountStakes function to amountStaked which fits the context better.

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 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.

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.