CERTIK

Security Assessment

StoneAgeNFT

Oct 10th, 2021

Table of Contents

Summary

This report has been prepared for StoneAgeNFT to discover issues and vulnerabilities in the source code of the StoneAgeNFT project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review 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.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	StoneAgeNFT
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/StoneAgeNFT/GES/tree/d344e45151624e41774e558b569315644e3cd6ab
Commit	d344e45151624e41774e558b569315644e3cd6ab

Audit Summary

Delivery Date	Oct 10, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

Vulnerability Level	Total	⊙ Pending	⊗ Declined	ⓘ Acknowledged	⊙ Partially Resolved	⊘ Resolved
● Critical	0	0	0	0	0	0
● Major	2	0	0	2	0	0
● Medium	1	0	0	1	0	0
● Minor	0	0	0	0	0	0
● Informational	1	0	0	1	0	0
● Discussion	0	0	0	0	0	0

Audit Scope

ID			File	SHA256 Checksum

Findings

🟥 **Critical**				**0**	(0.00%)
🟧 **Major**				**2**	(50.00%)
🟨 **Medium**				**1**	(25.00%)
🟨 **Minor**				**0**	(0.00%)
🟦 **Informational**				**1**	(25.00%)
🟩 **Discussion**				**0**	(0.00%)
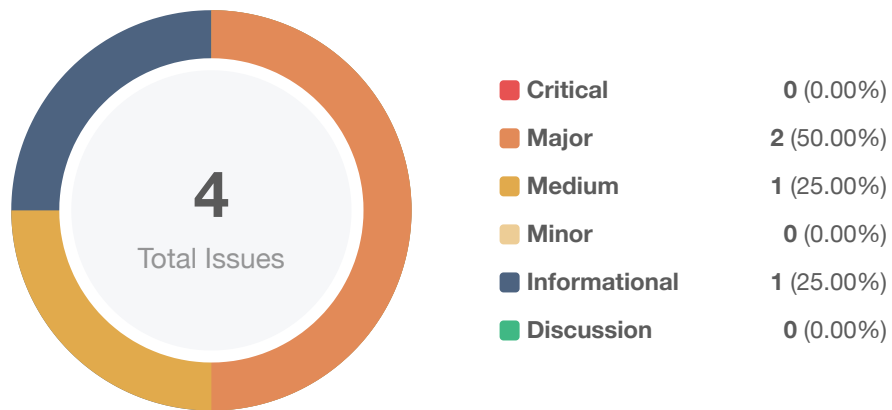
ID	Title	Category	Severity	Status
GES-01	Initial Token Distribution	**Centralization / Privilege**	🟠 **Major**	ⓘ Acknowledged
GES-02	Centralization Risk	**Centralization / Privilege**	🟠 **Major**	ⓘ Acknowledged
GES-03	approveSwitch can be a modifier	Coding Style	🔵 Informational	ⓘ Acknowledged
GES-04	Ineffective approval prevention	Logical Issue	🟡 Medium	ⓘ Acknowledged

GES-01 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	contracts/GES.sol: 19	ⓘ Acknowledged

Description

All of the GES tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute GES tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process.

Alleviation

[StoneAgeNFT Team]: This is to avoid 3rd party service dependencies which might cause major impact to our future operation and project plan. we will manually split the private key and recovery key in different stackholders to avoid the centralized risk. This is because we experienced in using other 3rd party services and due to them, we have to perform contract migration.

GES-02 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	contracts/GES.sol: 18, 35~37	ⓘ Acknowledged

Description

Contract deployer is able to pause all GES token transfers and approvals, as well as mint as many tokens as possible which can be used to saturate the market. However, pausing approvals can be bypassed by the `approveV2` function.

Recommendation

If the client wants to remove the version of approvals that can be paused:

- Remove `approve` and the `approveSwitch` mechanism, and keep `approveV2`, but since `approveV2` just returns `super.approve(spender, amount)` the developer should remove `approveV2` to avoid redundancy.

We also advise the client to carefully manage the `Owner` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

[StoneAgeNFT Team]: this is due to we need to disable the "Add Liquidity" feature in other defi system until we are ready, however, before this, we will need to allow the owner to be able to perform Buy NFT action in our marketplace, and therefore this approveV2 function is specially designed for the marketplace.

GES-03 | approveSwitch can be a modifier

Category	Severity	Location	Status
Coding Style	● Informational	contracts/GES.sol: 24~26	ⓘ Acknowledged

Description

approveSwitch checking logic can be separated into a modifier.

Recommendation

Can extract the code that checks approveSwitch into a modifier, and apply the modifier to the approve function.

GES-04 | Ineffective approval prevention

Category	Severity	Location	Status
Logical Issue	● Medium	contracts/GES.sol: 23~33	ⓘ Acknowledged

Description

The function `approve` only allows the user to approve GES tokens if and only if `approveSwitch` != 1. When `approveSwitch` = 1, approvals done through the `approve` function cannot be made. This will pause users and dApps who uses the `approve` function, which is common since GES are treated as an ERC20, but not those who use the `approveV2` function.

Recommendation

Remove the `approveV2` function, or remove both approval functions made in the `GES.sol` contract and the `approveSwitch` if `approveV2` is supposed to be the intended approve function.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

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 you ("Customer" or 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, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

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 CertiK 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 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. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK'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 functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.