Security Assessment

Fans Force

Jul 10th, 2021

Table of Contents

Summary

This report has been prepared for Fans Force to discover issues and vulnerabilities in the source code of the Fans Force 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 given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Fans Force
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/FansForceNFT/fansforce-contract
Commit	b5b582cf7cae3bdefc19169a6a4db0c43d8b6e49

Audit Summary

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

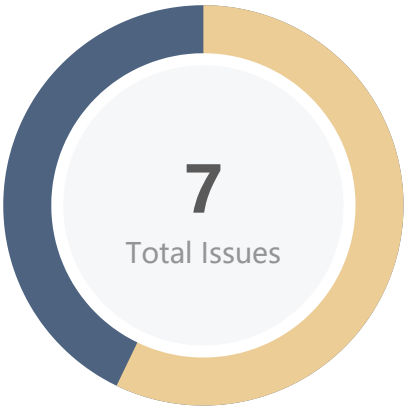
Vulnerability Summary

Vulnerability Level	Total	Pending	Partially Resolved	Resolved	Acknowledged	Declined
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	0	0	0	0	0	0
Minor	4	0	0	1	3	0
Informational	3	0	0	3	0	0
Discussion	0	0	0	0	0	0

Audit Scope

ID	file	SHA256 Checksum
FFN	FFNToken.sol	b87497965e49659fda26eb9ee8891b17d320f070afe3b5b0c305ca4c45fd23f5
FFF	FansForceNFT.sol	b9f09be66dfc659d2305da6293003c4fcd8151aebe13d254e927f6296f2080a6
FFT	FansForceNFTManager.sol	b546cc3dc8e1a4da4c324dbd4b384d2f963a6d6b24a5adb31933d4383c765489
NFT	NFTShareTemp.sol	117a94e4e4ea8a33cf1e90e37f8e8769d10421ea536d2171cfb5d839207cea80

Findings



Critical	**0** (0.00%)
Major	**0** (0.00%)
Medium	**0** (0.00%)
Minor	**4** (57.14%)
Informational	**3** (42.86%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
FFF-01	Conflict between judgement condition and exception message	Logical Issue	Minor	Resolved
FFF-02	Variable pending never updated	Logical Issue	Minor	Acknowledged
FFF-03	Unfair to the other guy	Logical Issue	Minor	Acknowledged
FFF-04	Missing zero address validation	Optimization	Informational	Resolved
FFF-05	Public function that could be declared external	Gas Optimization	Informational	Resolved
FFN-01	Initial token distribution	**Centralization / Privilege**	**Minor**	**Acknowledged**
NFT-01	Missing zero address validation	Optimization	Informational	Resolved

FFF-01 | Conflict between judgement condition and exception message

Category	Severity	Location	Status
Logical Issue	Minor	projects/FFNContract/FansForceNFT.sol: 227	Resolved

Description

Conflict between judgement condition and exception message. When `NFTInfo.sharing` is not true, it will throw the exception: `burn:tokenId already sharing`

Recommendation

Consider making two of them Unanimous.

Alleviation

[Fans Force Team]: This is our design.The NFT is occur sharing can not be burnd;

FFF-02 | Variable pending never updated

Category	Severity	Location	Status
Logical Issue	Minor	projects/FFNContract/FansForceNFT.sol: 246	Acknowledged

Description

Function `_beforeTokenTransfer` has a check on the `NFTInfo.pending`. It will need `pending` being false to transfer if token exists. However, the `NFTInfo.pending` never updated to false so that the NFT token can never be tansfered.

Recommendation

Consider adding function which can update the variable `pending`.

Alleviation

[Fans Force Team]: Yes， now it is not allown to change the status. There are some conditions to do this. But we are not decision, we will update the contract for next verion to do this.

FFF-03 | Unfair to the other guy

Category	Severity	Location	Status
Logical Issue	Minor	projects/FFNContract/FansForceNFT.sol: 124	Acknowledged

Description

Only the creator or the owner of the NFT token can share. However, One of them can share without the other person's knowledge. The share for them can be extremely unfair under these circumstances.

Recommendation

Consider adding approval mechanism. For Example:

```
    contract xxx {
        mapping(uint=>address[]) tokenVoteAddress;
        function proposalOrVote(uint tokenId) {
            address owner = ERC721Upgradeable.ownerOf(tokenId);
            address creator = _nftInfos[tokenId].creatorAddress;
            require(owner == _msgSender() || creator == _msgSender(), "vote:only owner or creator
can vote ");
            address[] storage voteAddress = tokenVoteAddress[tokenId];
            for(uint i = 0;i < voteAddress.length;i++) {
                if(voteAddress[i] == _msgSender()) {
                    return;
                }
            }
            voteAddress.push(_msgSender());
        }

        function share(...) {
            require(tokenVoteAddress[tokenId].length == 2, "")
        }
    }
```

Alleviation

[Fans Force Team]: The creator and owner have the same auth with this function. The proportion is to make a deal offline between them. If someone breaks the rule, this is one of the conditions to reject change pending to true.

FFF-04 | Missing zero address validation

Category	Severity	Location	Status
Optimization	Informational	projects/FFNContract/FansForceNFT.sol: 148~151	Resolved

Description

Missing zero address validation.

Recommendation

Check that the address is not zero. For example:

```
require(shareErc20Temp != address(0), "shareErc20Temp invalid");
```

```
require(nftContractAddr != address(0). "nftContractAddr invalid");
```

Alleviation

Team heeded our advise and change is applied in commit `b5b582cf7cae3bdefc19169a6a4db0c43d8b6e49`.

FFF-05 | Public function that could be declared external

Category	Severity	Location	Status
Gas Optimization	Informational	projects/FFNContract/FansForceNFT.sol: 111~117, 74~80	Resolved

Description

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

Recommendation

Use the external attribute for functions never called from the contract.

Alleviation

Team heeded our advise and change is applied in commit `b5b582cf7cae3bdefc19169a6a4db0c43d8b6e49`.

FFN-01 | Initial token distribution

Category	Severity	Location	Status
Centralization / Privilege	**Minor**	projects/FFNContract/FFNToken.sol: 10	**Acknowledged**

Description

All of the `FFNToken` tokens are sent to the contract deployer when deploying the contract.

Recommendation

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

Alleviation

[Fans Force]: Not worry. Before FFN flows to the market. We will transfer to another address, just like our whitepaper says.

NFT-01 | Missing zero address validation

Category	Severity	Location	Status
Optimization	Informational	projects/FFNContract/NFTShareTemp.sol: 22~28	Resolved

Description

Missing zero address validation.

Recommendation

Check that the address is not zero. For example:

```
require(shareErc20Temp != address(0), "shareErc20Temp invalid");
```

```
require(nftContractAddr != address(0). "nftContractAddr invalid");
```

Alleviation

Team heeded our advise and change is applied in commit `b5b582cf7cae3bdefc19169a6a4db0c43d8b6e49`.

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.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

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.

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 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 CertiK'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 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.

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.