DeHacker

Code Security Assessment

Verse Farms (2023)

July 13th, 2023

Contents

Summary

DeHacker's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.
Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow/underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service/logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Issue Categories

Every issue in this report was assigned a severity level from the following:

Critical severity issues

A vulnerability that can disrupt the contract functioning in a number of scenarios or creates a risk that the contract may be broken.

Major severity issues

A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.

Medium severity issues

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

Minor severity issues

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

Informational

A vulnerability that has informational character but is not affecting any of the code.

Overview

Project Summary

Project Name	Verse Farms (2023)
Platform	Ethereum
Website	https://www.bitcoin.com/
Type	DeFi, Staking
Language	Solidity

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	1	0	0	0	0	1
Major	1	0	0	1	0	0
Medium	0	0	0	0	0	0
Minor	3	0	0	1	0	2
Informational	0	0	0	0	0	0
Discussion	0	0	0	0	0	0

Audit scope

ID	File	SHA256 Checksum
SFB	contracts/SimpleFarm.sol	fe869870d5a0cf9efa67092de1c569d7f0abcf62576246e8c876e9aae9f4bb56
TWB	contracts/TokenWrapper.sol	ec7886ee1b6beaf386caa957058bf7fc5ec1eba6e032ed4d0c54d002dd82ef2
IER	contracts/IERC20.sol	215e6566be35c9700ee4d29c4738bf46cb78b72b2a8ba1072a71a6a2ff44305e
SER	contracts/SafeERC20.sol	6e1eeda04a44b13b163c6a350643d8bd7cd95a54db25704ef8474d6d1d890bd3

Findings

ID	Category	Severity	Status
TWB-01	Logical Issue	Critical	Resolved
SFB-01	Centralization /Privilege	Major	Acknowledged
SFB-02	Volatile Code	Minor	Resolved
TWB-02	Volatile Code	Minor	Acknowledged
TWB-03	Logical Issue	Minor	Resolved

INFORMATIONAL

TWB-01|FUNCTION _transfer() SHOULD UPDATE REWARD

Category	Severity	Location	Status
Logical Issue	Critical	contracts/ TokenWrapper. sol: 112~113	Resolved

Description

The token serves as a receipt for staking and is used to compute the user's reward. When a user's balance changes, his orher reward must be calculated immediately.

Recommendation

We recommend the team update the reward accounts for both sides of the transaction before the transfer.

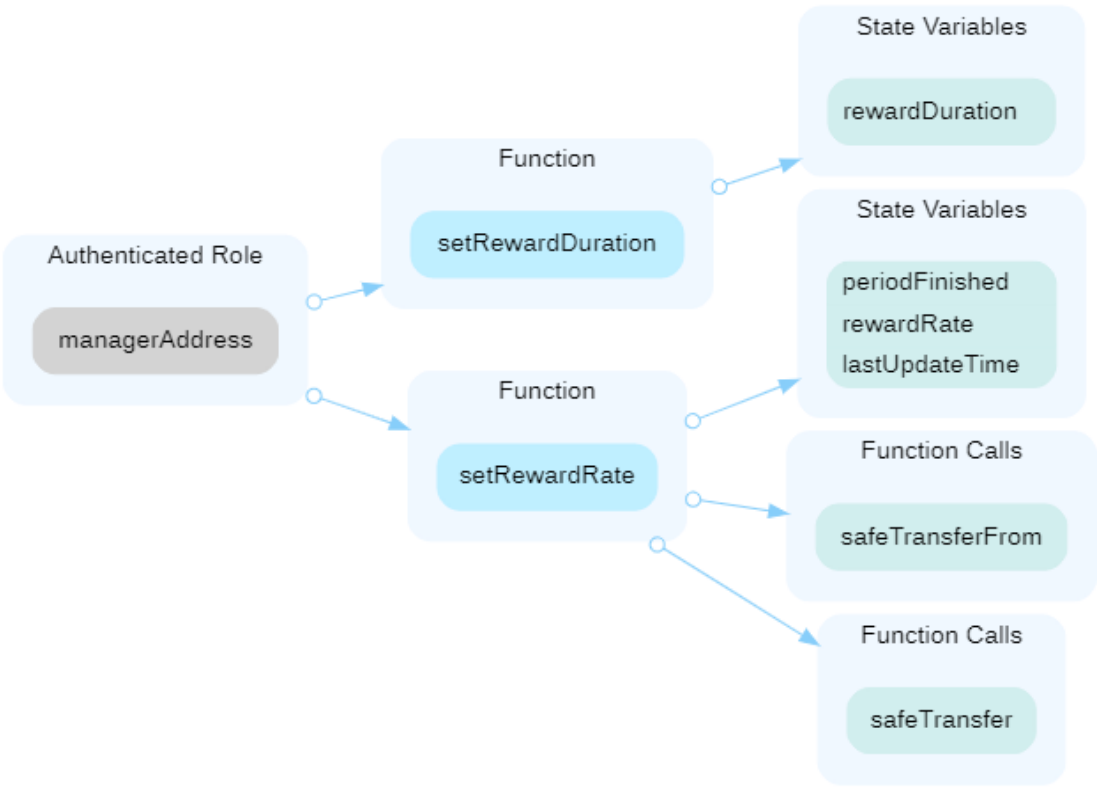
INFORMATIONAL

SFB-01|CENTRALIZATION RISKS IN SIMPLEFARM.SOL

Category	Severity	Location	Status
Centralization / Privilege	Major	contracts/ SimpleFarm.sol: 291 , 325, 372, 400	Acknowledged

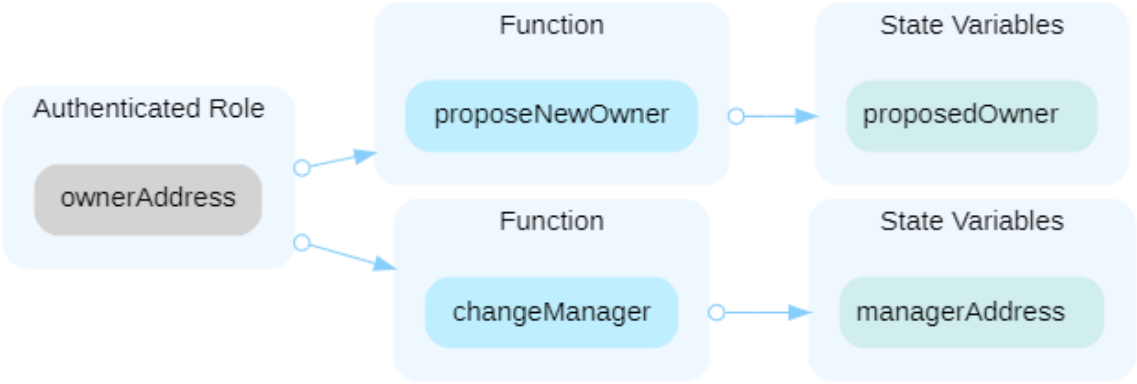
Description

In the contract SimpleFarm the role managerAddress has authority over the functions shown in the diagram below. Any compromise to the manager Address account may allow the hacker to take advantage of this authority and change therewardDuration and rewardRate .



In the contract SimpleFarm the role ownerAddress has authority over the functions shown in the diagram below. Anycompromise to the ownerAddress account may allow the hacker to take advantage of this authority and change the owneror manager of the contract.

In the contract SimpleFarm the role managerAddress has authority over the functions shown in the diagram below. Any compromise to the manager Address account may allow the hacker to take advantage of this authority and change therewardDuration and rewardRate .



Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level ofdecentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefullymanage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommendcentralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accountswith enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that wouldalso mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:
Timelock and Multi sign (2/3,3/5) combination mitigate by delaying the sensitive operation and avoiding a single point of keymanagement failure.
Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND
Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private keycompromised;
AND
A medium/blog link for sharing the timelock contract and multi-signers addresses information with the publicaudience.

Long Term:
Timelock and DAO, the combination, mitigate by applying decentralization and transparency.
Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
AND
Introduction of a DAO/governance/voting module to increase transparency and user involvement.
AND
A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the publicaudience.

Permanent:
Renouncing the ownership or removing the function can be considered fully resolved.
Renounce the ownership and never claim back the privileged roles.
OR
Remove the risky functionality.

INFORMATIONAL

SFB-02|MISSING ZERO ADDRESS VALIDATION IN SimpleFarm.sol

Category	Severity	Location	Status
Volatile Code	Minor	contracts/ SimpleFarm.sol: 297, 331	Resolved

Description

Addresses should be checked before assignment or external call to make sure they are not zero addresses.

```
297         proposedOwner = _newOwner;
```

_newOwner is not zero-checked before being used.

```
331         managerAddress = _newManager;
```

_newManager is not zero-checked before being used.

Recommendation

We advise adding a zero-check for the passed-in address value to prevent unexpected errors..

INFORMATIONAL

TWB-02|MISSING ZERO ADDRESS VALIDATION INTokenWrapper.sol

Category	Severity	Location	Status
Volatile Code	Minor	contracts/ TokenWrapper.sol: 94, 138~139, 162, 214, 233	Acknowledged

Description

The aforementioned parameters are missing the zero address check. It is not suitable to transfer tokens to a zero address,approve allowances to a zero address, or transfer from a zero address.

Recommendation

We recommend zero address checks for these parameters to avoid wastes of gas.

INFORMATIONAL

TWB-03|MISSING EMIT EVENTS

Category	Severity	Location	Status
Logical Issue	Minor	contracts/ TokenWrapper.sol: 57, 75	Resolved

Description

The event Transfer should be emitted in the function _stake and _withdraw to support user account tracking in theexplorer.

Recommendation

We recommend the team emit Transfer event in these functions.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

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.

Coding Style

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

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

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.

About

DeHacker is a team of auditors and white hat hackers who perform security audits and assessments. With decades of experience in security and distributed systems, our experts focus on the ins and outs of system security. Our services follow clear and prudent industry standards. Whether it's reviewing the smallest modifications or a new platform, we'll provide an in-depth security survey at every stage of your company's project. We provide comprehensive vulnerability reports and identify structural inefficiencies in smart contract code, combining high-end security research with a real-world attacker mindset to reduce risk and harden code.

BLOCKCHAIINS

Ethereum

Cosmos

Eos

Substrate

TECH STACK

Python

Solidity

Rust

c++

CONTACTS

https://dehacker.io

https://twitter.com/dehackerio

https://github.com/dehacker/audits_public

https://t.me/dehackerio

https://blog.dehacker.io/

July 2023