Smart Contract Security Audit

Audit details:

Audited project:	NafterX
Deployer address:	0xe3AA0302B4aa87C4f57FAe7Ec4235B54396dCa2F
Client contacts:	NafterX team
Blockchain:	Binance Smart Chain
Project website:	https://nafterx.com

May, 2021
TechRate

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by NafterX to perform an audit of smart contracts:

- *https://bscscan.com/address/0x7927c99a9a48b737b155f929ba5813266398c362#code*

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts details

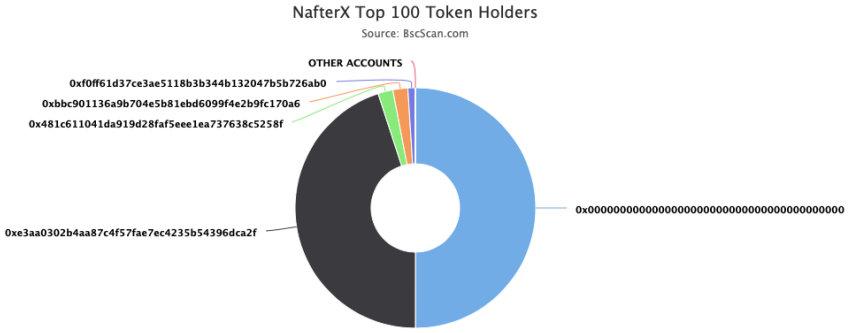
Token contract details for 26.05.2021.

Contract name:	NafterX
Contract address:	0x7927c99a9A48B737b155f929ba5813266398C362
Total supply:	1000000000000000
Token ticker:	NAFTX
Decimals:	9
Token holders:	5
Transactions count:	5
Top 100 holders dominance:	100.00
Contract deployer address:	0xe3AA0302B4aa87C4f57FAe7Ec4235B54396dCa2F
Contract’s current owner address:	0xe3AA0302B4aa87C4f57FAe7Ec4235B54396dCa2F

NafterX token distribution

The top 100 holders collectively own 100.00% (1,000,000,000,000,000.00 Tokens) of NafterX

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 5



(A total of 1,000,000,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

NafterX contract interaction details

Time Series: Token Contract Overview

Fri 21, May 2021 - Fri 21, May 2021

NafterX top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x0000000000000000000000000000000000000000	500,000,000,000,000	50.0000%
2	0xe3aa0302b4aa87c4f57fae7ec4235b54396dca2f	450,000,000,000,000	45.0000%
3	0x481c611041da919d28faf5eee1ea737638c5258f	20,000,000,000,000	2.0000%
4	0xbbc901136a9b704e5b81ebd6099f4e2b9fc170a6	20,000,000,000,000	2.0000%
5	0xf0ff61d37ce3ae5118b3b344b132047b5b726ab0	10,000,000,000,000	1.0000%

Contract functions details

+ NafterX
  - [Pub] <Constructor> #
  - [Pub] initialBurn #
  - [Pub] _transferDevFunds #
  - [Pub] _transferMarketingFunds #
  - [Pub] _transferCharityFunds #
  - [Ext] getPreSaleDone
  - [Ext] disablePresale #
    - modifiers: onlyOwner
  - [Ext] enablePresale #
    - modifiers: onlyOwner
  - [Ext] getTotalSupply
  - [Ext] getPricePerToken
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] getTokenBalance
  - [Ext] getContractAddress
  - [Ext] getPreSaleBalance
  - [Pub] preSalePurchase ($)
  - [Ext] transfer #
  - [Int] _transfer #
  - [Pub] payoutToPresaleOwner ($)
    - modifiers: onlyOwner
  - [Ext] airDrop #
    - modifiers: onlyOwner
  - [Ext] destroyBlackFunds #
    - modifiers: onlyOwner
  - [Pub] transferFrom #
  - [Pub] approve #
  - [Ext] allowance
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Prv] _approve #

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] div
  - [Int] mod
  - [Int] mod

Issues Checking Status

№	Issue description.	Checking status
1	Compiler errors.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model of the contract.	High issues
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	High issues
19	Cross-function race conditions.	Passed
20	Safe Open Zeppelin contracts implementation and usage.	Passed
21	Fallback function security.	Passed

Security Issues

High Severity Issues

1. Exceeding presale limit

Issue:

- ❏ The preSalePurchase() function compares preSaleDone to preSaleLimit without adding new values. In corner cases, this can lead to the preSaleLimit being exceeded.

```
require(preSaleDone <= preSaleLimit, "Pre Sale Completed");
preSaleDone = preSaleDone.add(totalTokenValue);
```

Recommendation:

Compare preSaleLimit with preSaleDone adding totalTokenValue to the last one.

2. Wrong airDrop

Issue:

- ❏ The airDrop() function changes balances without affecting to contract economy model (ex.: changing totalSupply), that may cause wrong calculations of further transactions, totalSupply changing and others.
- ❏ _addresses.length not compared with _amounts.length to avoid out of bounds mistakes.

```
for (uint256 i = 0; i < _addresses.length; i++) {
    balanceOf[_addresses[i]] = balanceOf[_addresses[i]].add(
        _amounts[i]
    );
    emit Transfer(address(this), _addresses[i], _amounts[i]);
}
```

Recommendation:

Refactor logic of the function with correcting the calculations.

3. destroyBlackFunds error

Issue:

- ❏ destroyBlackFunds function changes balances without affecting to contract economy model (ex.: changing totalSupply, sending withdrawn amount to specific address). This may cause wrong calculations of further transactions, totalSupply changing and others.
- ❏ Owner could nullify any users balance

```
function destroyBlackFunds(address _blackListedUser) external onlyOwner {
    uint256 dirtyFunds = balanceOf[_blackListedUser];
    balanceOf[_blackListedUser] = 0;
    emit DestroyedBlackFunds(_blackListedUser, dirtyFunds);
}
```

Recommendation:

Refactor logic of the function with correcting the calculations.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

No low severity issues found.

Owner privileges (In the period when the owner is not renounced)

- ❏ Owner can enable & disable presale.

Conclusion

Smart contracts contain high severity issues.

Techrate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.