



Smart Contract Security Audit Report

AxeDAO

November 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

AxeDAO



Deployer address

0xAD92618e40356671057d05C1FD0F5862ebe1f39A



Client contacts

AxeDAO Team



Blockchain

Ethereum



Website

Not provided

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) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by AxeDAO to perform an audit of smart contracts:

- <https://etherscan.io/token/0x30AC8317DfB0ab4263CD8dB1C4F10749911B126C#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 understood 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.

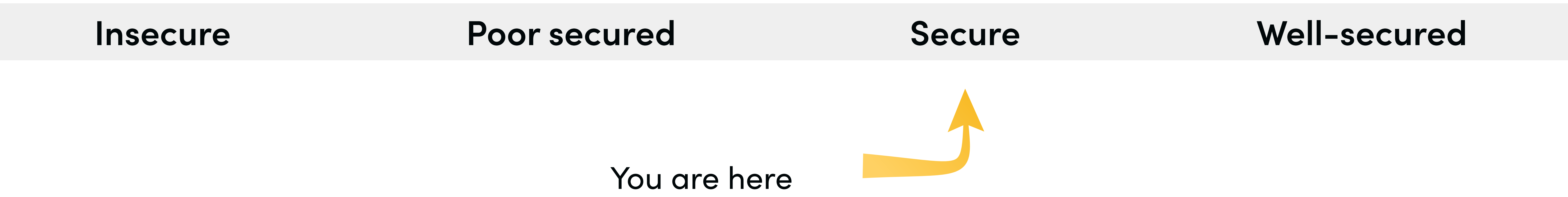
Contract Details

Token contract details for 26.11.2022

Token Type	: DAO
Contract name	: AxeERC20
Contract address	: 0x30AC8317DfB0ab4263CD8dB1C4F10749911B126C
Total supply	: 3,113,555.363980538
Token ticker	: AXE
Decimals	: 9
Token Holders	: 111
Transactions count	: 2,383
Compiler version	: v0.7.5+commit.eb77ed08
Contract deployer address	: 0xAD92618e40356671057d05C1FD0F5862ebe1f39A
Owner address	: 0x3586075F3997d52C866B00FA6bbb8Eb6E1b702Cd
Contract vault address	: 0xd2039621Cc042567092fAaee89B03Ef959F89712

Audit Summary


According to the standard audit assessment, Customer`s solidity smart contracts are **“Secure”**. This contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.




We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 0 medium and 1 low.

AxeDAO Distribution

 The top 100 holders collectively own 100.00% (3,113,555.36 Tokens) of AxeDAO

 Token Total Supply: 3,113,555.36 Token





|

Total Token Holders: 111



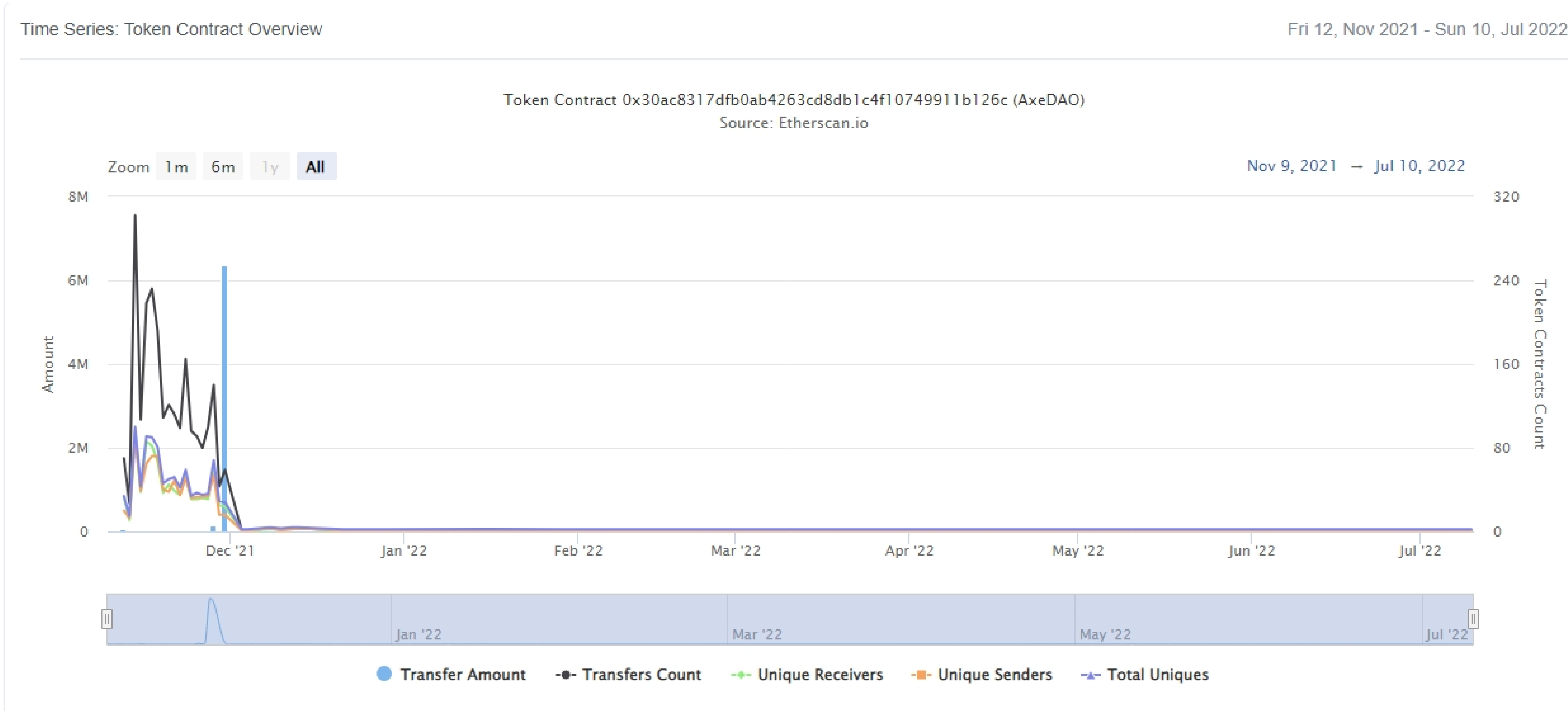
AxeDAO Top 20 Token Holders

(A total of 3,113,555.36 tokens held by the top 100 accounts from the total supply of 3,113,555.36 token)

Rank	Address	Quantity (Token)	Percentage
1	0x04052181207eaf35da27c0d042afd589a871b79d	2,884,012.441739197	92.6276%
2	 Uniswap V2: AXE-DAI	134,648.595907327	4.3246%
3	 0x7f4d186b76a39a2da32804a8c35b3d04e0e174fd	31,949.223275533	1.0261%
4	Null Address: 0x00...dEaD	16,372.212417041	0.5258%
5	0xc17293df5741aaa4d3bd8f5f54748c47c568cf73	16,056.188923782	0.5157%
6	0xeb09aeae8fe0868fb63a0fa1134e4bd18df723c0	11,031.195925241	0.3543%
7	 0xbae21d4247dd3818f720ab4210c095e84e980d96	10,000	0.3212%
8	0xcd4087d136f33ae6225e993028e969caa3a120cc	3,599.548974895	0.1156%
9	0x117acf9277ec297bfed9157ce26638f51ea29dcc	1,726.433516331	0.0554%
10	 0x77116eb7c6c827efc47b99685a6a22f5fefcbcb55	550.261447103	0.0177%
11	0x3a614e0d52501dda6d241520f9ea322c619b7668	476.530894258	0.0153%
12	0x60a7bdc04030162ebc7768e8afe7a99cf1a436fe	471.261026785	0.0151%
13	0x37b2360a8f0425924544dafbf5b79d07e3e864dd	470.522532364	0.0151%
14	0xe7323e665a37ff90ddd40d13261aa76a3a05e2e4	323.452313217	0.0104%
15	0x6470c7ecb0f882705940ecd22f9dbfb755bdfcdc	200	0.0064%
16	0x57780b68026e4bbc82280e2725aa706e6a6260e2	184.316082921	0.0059%
17	0x8a74921bb8b4b6b48a8b8d3c1173b94993afc8b7	166.832396382	0.0054%
18	0xf76c14d259d56afe02c6dabab592b23671f5ba6b	166.661240742	0.0054%
19	0xb4dbc39c3ea0dcb67db6c4ce35c6cf0f9f1cd47c	132.863566859	0.0043%
20	0x8001d28203b62970e9c819473a0dc82e8344f638	119.749693653	0.0038%

AxeDAO Distribution

AxeDAO Contract Overview



Contract functions details

AxeERC20.sol

+VaultOwned (Ownable)

-[Ext] setVault #

-modifiers: onlyOwner

-[Pub] vault

+AxeERC20 (ERC20Permit, VaultOwned)

-<constructor>

-[Ext] mint #

-modifiers: onlyVault

-[Pub] burn #

-[Pub] burnFrom #

-[Pub] _burnFrom #

SafeMath.sol

+**[Lib]** SafeMath

-[Int] add

-[Int] sub

-[Int] sub

-[Int] mul

-[Int] div

-[Int] div

-[Int] sqrrt

ERC20.sol

+ERC20 (IERC20)

-<constructor>

-[Pub] name

-[Pub] symbol

-[Pub] decimals

-[Pub] totalSupply

-[Pub] balanceOf

-[Pub] transfer #

-[Pub] allowance

-[Pub] approve #

-[Pub] transferFrom #

-[Pub] increaseAllowance #

-[Pub] decreaseAllowance #

-[Int] _transfer #

Contract functions details

- [Int] _mint #
- [Int] _burn #
- [Int] _approve #
- [Int] _beforeTokenTransfer #

+[Int] IERC2612Permit

- [Ext] permit
- [Ext] nonces

+ERC20Permit (ERC20, IERC2612Permit)

- <constructor>
- [Pub] permit

+[Lib] SafeERC20

- [Int] safeTransfer #
- [Int] safeTransferFrom #
- [Int] safeApprove #
- [Int] safeIncreaseAllowance #
- [Int] safeDecreaseAllowance #
- [Pvt] _callOptionalReturn #

Ownable.sol

+[Int] IOwnable

- [Ext] owner
- [Ext] renounceManagement
- [Ext] pushManagement
- [Ext] pullManagement

+Ownable (IOwnable)

- <constructor>
- [Pub] owner
- [Pub] renounceManagement #
 - modifiers: onlyOwner
- [Pub] pushManagement #
 - modifiers: onlyOwner
- [Pub] pullManagement #
 - modifiers: onlyOwner

Address.sol

+[Lib] Address

Contract functions details

- [Int] isContract
- [Int] sendValue
- [Int] functionCall
- [Int] functionCall
- [Int] functionCallWithValue
- [Int] functionCallWithValue
- [Pvt] _functionCallWithValue
- [Int] functionStaticCall
- [Int] functionStaticCall
- [Int] functionDelegateCall
- [Int] functionDelegateCall
- [Pvt] _verifyCallResult
- [Int] addressToString

Counters.sol

- +[Lib] Counters
 - [Int] current
 - [Int] increment
 - [Int] decrement

IERC20.sol

- +[Int] IERC20
 - [Ext] name
 - [Ext] symbol
 - [Ext] decimals
 - [Ext] totalSupply
 - [Ext] balanceOf
 - [Ext] allowance
 - [Ext] approve
 - [Ext] transfer
 - [Ext] transferFrom

(\$) = payable function

= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	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.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Low issue

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

✔ Critical Severity Issues

No critical severity issue found.

✔ High Severity Issues

No high severity issue found.

✔ Medium Severity Issues

No medium severity issue found.

✔ Low Severity Issues

One low severity issue found.

1. Old compiler version

- **Description**

Contract has been deployed using too old solidity version.

- **Recommendation**

It is advisable to deploy contract using any of the latest version of solidity

Centralization

Owner/vault Privileges :

- AxeDAO Contract:
 - Owner can renounce and transfer ownership.
 - Owner can mint.
 - Owner can set vault.
 - Owner can set new owner.

This smart contract has some functions which can be executed by the Vault (Owner) only. If the admin wallet private key would be compromised, then it would create trouble as smart contract ownership has not been renounced. Following are Admin/vault functions:

- renounceManagement
- mint
- setVault
- pushManagement

Conclusion

Smart contract contains low severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe 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.