



Smart Contract Security Audit Report

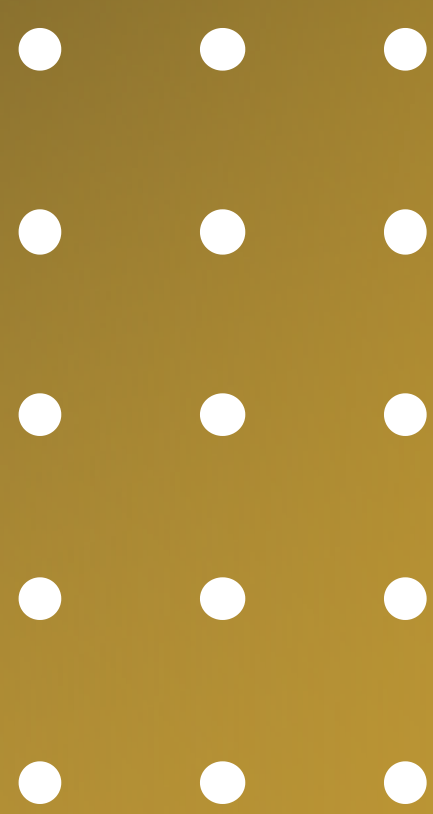
Milk Protocol

August 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

Milk Protocol



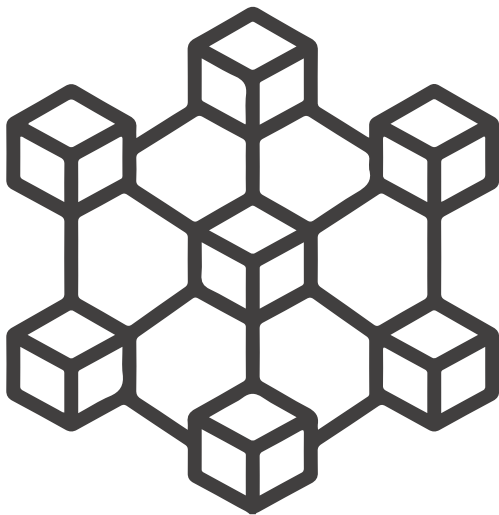
Deployer address

0xAE1F48653Ab26D13A706DB78988c28C493321ae2



Client contacts

Milk Protocol team



Blockchain

Binance Smart chain



Website

<https://stakecow.com/>

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 Milk Protocol to perform an audit of smart contracts:

- <https://bscscan.com/address/0x8e9f5173e16ff93f81579d73a7f9723324d6b6af#code>

The purpose of the audit was to achieve the

- 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 22.08.2022

Token Type	: BEP20
Contract name	: Milk
Contract address	: 0x8E9f5173e16Ff93F81579d73A7f9723324d6B6aF
Compiler version	: v0.5.16+commit.9c3226ce
Total supply	: 13,292
Token Ticker	: MILK
Decimals	: 18
Token Holders	: 1,824
Transactions count	: 22,031
Contract deployer address	: 0xAE1F48653Ab26D13A706DB78988c28C493321ae2
Owner address	: No Owner

Social profiles

Twitter profile : <https://twitter.com/StakeCow>

Telegram Profile : <https://t.me/StakeCow>

Audit Summary

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

Insecure	Poor secured	Secure	Well-secured
----------	--------------	--------	--------------

You are here



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 and some very low-level issues. These issues are not critical ones.

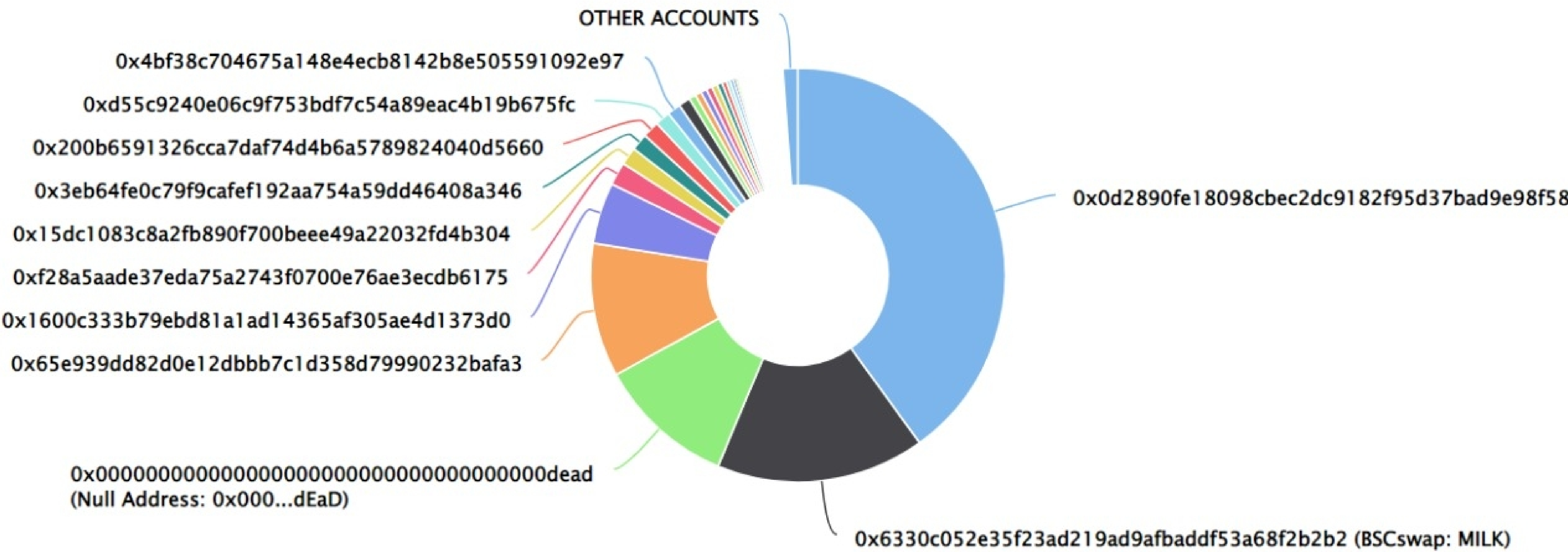
Milk Protocol Token Distribution

💡 The top 100 holders collectively own 98.85% (13,139.69 Tokens) of Milk Protocol

💡 Token Total Supply: 13,292.00 Token | Total Token Holders: 1,824






Milk Protocol Top 100 Token Holders

Source: BscScan.com



Milk Protocol Top 20 Token Holders

(A total of 13,139.69 tokens held by the top 100 accounts from the total supply of 13,292.00 token)

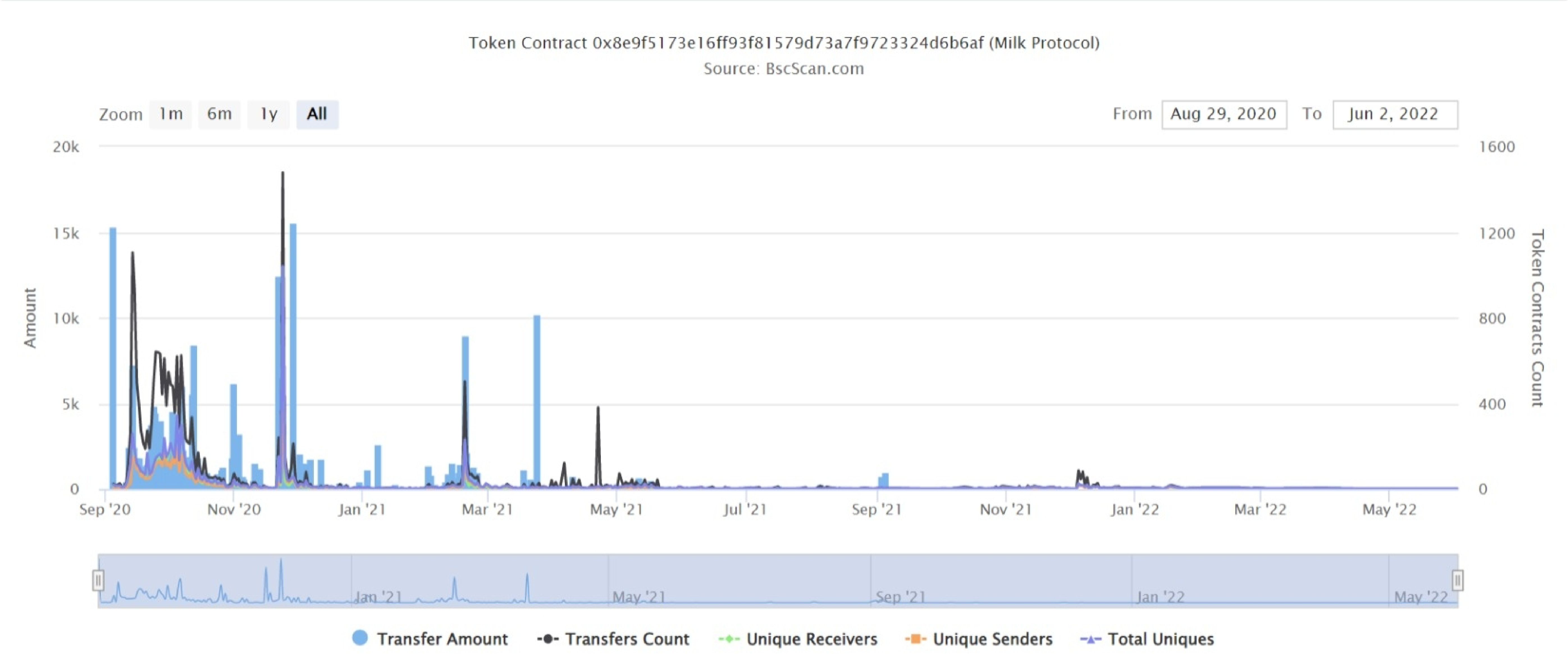
Rank	Address	Quantity (Token)	Percentage
1	0x0d2890fe18098cbec2dc9182f95d37bad9e98f58	5,319.289436802580742249	40.0187%
2	 BSCswap: MILK	2,165.113771395212620966	16.2888%
3	Null Address: 0x000...dEaD	1,424.597069627440135687	10.7177%
4	0x65e939dd82d0e12dbbb7c1d358d79990232bafa3	1,389.069412318743090239	10.4504%
5	 0x1600c333b79ebd81a1ad14365af305ae4d1373d0	636.3968	4.7878%
6	0xf28a5aade37eda75a2743f0700e76ae3ecdb6175	240.078524059864192223	1.8062%
7	 0x15dc1083c8a2fb890f700beee49a22032fd4b304	185.070729485158698923	1.3923%
8	0x3eb64fe0c79f9cafef192aa754a59dd46408a346	178.479462909828793152	1.3428%
9	0x200b6591326cca7daf74d4b6a5789824040d5660	173.210717093117759686	1.3031%
10	0xd55c9240e06c9f753bdf7c54a89eac4b19b675fc	154.388895959387810821	1.1615%
11	 0x4bf38c704675a148e4ecb8142b8e505591092e97	138.499999999999998	1.0420%
12	0x137936bb17ea85a70dc9fc44c17b3cee0452883f	123.838938243246215177	0.9317%
13	0x112a6a20c69da8aa432e68b7f67e823f1a05195b	70.656104076834400281	0.5316%
14	0x2ad2200a2d056c79b04592a9fe956e58e81d3dda	66.264942863545656736	0.4985%
15	0xb079a72c627d0a34b880aee0504b901cbce64568	62.474271784143407157	0.4700%
16	0xb95ae81e620e0419acd1835fa73a0f81a2d3de14	60	0.4514%
17	0x381a2d76f12984bdc91d35b5a841ea7891d25505	58.289169487596262157	0.4385%
18	0xdfda9a73cc842d17b84869df47a1e132b8d156f3	51.29795831179699825	0.3859%
19	 0x3c51a88134ef088c722f946202b3c4a197ec5bde	47.975795862560012122	0.3609%
20	0xb0a53b05cecf8ac547aa5887d6082e1f70d2f494	37.742791331710639249	0.2840%

Milk Protocol Token Distribution

Milk Protocol Contract Overview

Time Series: Token Contract Overview

Sat 5, Sept 2020 - Thu 2, Jun 2022



Contract functions details

+ Context

- [Int] <constructor>
- [Int] _msgSender
- [Int] _msgData

+ [Int] IERC20

- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] transfer
- [Ext] allowance
- [Ext] approve
- [Ext] transferFrom

+ [Lib] SafeMath

- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] div
- [Int] mod
- [Int] mod

+ ERC20 (Context, IERC20)

- <constructor>
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer #
- [Pub] allowance
- [Pub] approve #
- [Pub] transferFrom #
- [Pub] increaseAllowance
- [Pub] decreaseAllowance
- [Int] _transfer #
- [Int] _mint#
- [Int] _burn #
- [Int] _approve #
- [Int] _burnFrom #

Contract functions details

+ERC20Detailed (IERC20)

-[Pub] <constructor> #

-[Pub] name

-[Pub] symbol

-[Pub] decimals

+ Milk (ERC20, ERC20Detailed)

-[Pub] <constructor> #

(\$) = payable function

= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
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	Passed

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 issues found.

✔ Low Severity Issues

One low severity issue found.

1. Unlocked Compiler Version.

- **Description**

The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

It is advisable that the compiler version is alternatively locked at the lowest version possible so that the contract can be compiled. For example, for version 0.5.0 the contract should contain the following line:

```
pragma solidity 0.5.16;
```

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.