0.7

ShibaSwap V3 Process Quality Review

Score: 10%

Overview

This is a ShibaSwap Process Quality Review completed on August 31st 2021. It was performed using the Process Review process (version 0.7.3) and is documented here. The review was performed by Rex of DeFiSafety. Check out our Telegram. The previous version of the review is here. The first version of the review came out just as they launched before the Certik audit and had a score of 3%. The second version came out after the audit on July 10, with a 35% score. This score was unfairly high as it gave a 10% on Q1. DeFiSafety did this to smooth the fur of the Shib Army but this was unfair to our process. We are correcting this now with this report.

The final score of the review is **10%, a fail**. The breakdown of the scoring is in Scoring Appendix. For our purposes, a pass is 70%.

Summary of the Process

Very simply, the review looks for the following declarations from the developer's site. With these declarations, it is reasonable to trust the smart contracts.

- **Here are my smart contracts on the blockchain**
- **Here is the documentation that explains what my smart contracts do**
- **Here are the tests I ran to verify my smart contract**
- **Here are the audit(s) performed on my code by third party experts**
- **Here are the admin controls and strategies**

Disclaimer

This report is for informational purposes only and does not constitute investment advice of any kind, nor does it constitute an offer to provide investment advisory or other services. Nothing in this report shall be considered a solicitation or offer to buy or sell any security, token, future, option or other financial instrument or to offer or provide any investment advice or service to any person in any jurisdiction. Nothing contained in this report constitutes investment advice or offers any opinion with respect to the suitability of any security, and the views expressed in this report should not be taken as advice to buy, sell or hold any security. The information in this report should not be relied upon for the purpose of investing. In preparing the information contained in this report, we have not taken into account the investment needs, objectives and financial circumstances of any particular investor. This information has no regard to the specific investment objectives, financial situation and particular needs of any specific recipient of this information and investments discussed may not be suitable for all investors.

Any views expressed in this report by us were prepared based upon the information available to us at the time such views were written. The views expressed within this report are limited to DeFiSafety and the author and do not reflect those of any additional or third party and are strictly based upon DeFiSafety, its authors, interpretations and evaluation of relevant data. Changed or additional information could cause such views to change. All information is subject to possible correction. Information may quickly become unreliable for various reasons, including changes in market conditions or economic circumstances.

Chain

This section indicates the blockchain used by this protocol.

> ✓ **Chain: Ethereum**

Guidance:
Ethereum
Binance Smart Chain
Polygon

Code and Team

This section looks at the code deployed on the Mainnet that gets reviewed and its corresponding software repository. The document explaining these questions is here. This review will answer the questions;

1) Are the executing code addresses readily available? (%)
2) Is the code actively being used? (%)
3) Is there a public software repository? (Y/N)
4) Is there a development history visible? (%)
5) Is the team public (not anonymous)? (Y/N)

1) Are the executing code addresses readily available? (%)

> ⚠ Answer: 0%

Executing addresses could not be found on the website, Documentation, or Github. There has been **NO official developer-supported documentation** that has included any Smart Contract Addresses in their documentation. A contract address was found from a medium article, which is not affiliated with the site, but our process requires the protocol to publish their addresses. For this reason the score is 0%. As per our process, this means that the audit score is 0% also (See Q17).

From the medium article referenced above we got the following code;

https://etherscan.io/address/0x94235659cf8b805b2c658f9ea2d6d6ddbb17c8d7#code
etherscan.io

This code is used for subsequent questions.

Note: Although a token address is available, the executing smart contract addresses are not, and that is what we are primarily looking for.

Guidance:

100%	Clearly labelled and on website, docs or repo, quick to find
70%	Clearly labelled and on website, docs or repo but takes a bit of looking
40%	Addresses in mainnet.json, in discord or sub graph, etc
20%	Address found but labelling not clear or easy to find
0%	Executing addresses could not be found

How to improve this score

Make the Ethereum addresses of the smart contract utilized by your application available on either your website or your GitHub (in the README for instance). Ensure the addresses is up to date. This is a very important question wrt to the final score.

2) Is the code actively being used? (%)

> ✓ Answer: 100%

Activity exceeds 300 transactions a day as indicated in the Appendix.

Percentage Score Guidance

100%	More than 10 transactions a day
70%	More than 10 transactions a week
40%	More than 10 transactions a month
10%	Less than 10 transactions a month
0%	No activity

3) Is there a public software repository? (Y/N)

> ⚠ Answer: No

ShibaSwap has a private GitHub repository.

Is there a public software repository with the code at a minimum, but normally test and scripts also (Y/N). Even if the repo was created just to hold the files and has just 1 transaction, it gets a Yes. For teams with private repos, this answer is No.

4) Is there a development history visible? (%)

> ⚠ Answer: 0%

ShibaSwap has a Private Development Repository.

This checks if the software repository demonstrates a strong steady history. This is normally demonstrated by commits, branches and releases in a software repository. A healthy history demonstrates a history of more than a month (at a minimum).

Guidance:

100%	Any one of 100+ commits, 10+branches
70%	Any one of 70+ commits, 7+branches
50%	Any one of 50+ commits, 5+branches
30%	Any one of 30+ commits, 3+branches
0%	Less than 2 branches or less than 30 commits

How to improve this score

Continue to test and perform other verification activities after deployment, including routine maintenance updating to new releases of testing and deployment tools. A public development history indicates clearly to the public the level of continued investment and activity by the developers on the application. This gives a level of security and faith in the application.

5) Is the team public (not anonymous)? (Y/N)

> ⚠ Answer: No

No public team members.

For a yes in this question the real names of some team members must be public on the website or other documentation. If the team is anonymous and then this question is a No.

Documentation

This section looks at the software documentation. The document explaining these questions is here.

Required questions are;

6) Is there a whitepaper? (Y/N)
7) Are the basic software functions documented? (Y/N)
8) Does the software function documentation fully (100%) cover the deployed contracts? (%)
9) Are there sufficiently detailed comments for all functions within the deployed contract code (%)

10) Is it possible to trace from software documentation to the implementation in code (%)

6) Is there a whitepaper? (Y/N)

Answer: Yes

Location:
https://raw.githubusercontent.com/shytoshikusama/woofwoofpaper/main/SHIBA_INU_WOOF_WOOF.pdf

Note: It's the same whitepaper as the Shiba Inu whitepaper. There was a small section added to include some information about ShibaSwap.

7) Are the basic software functions documented? (Y/N)

Answer: No

No software function documentation was found in any **official developer-supported documentation.**

How to improve this score

Write the document based on the deployed code. For guidance, refer to the SecurEth System Description Document.

8) Does the software function documentation fully (100%) cover the deployed contracts? (%)

Answer: 0%

No software function documentation was found in any **official developer-supported documentation.**

Guidance:

100%	All contracts and functions documented
80%	Only the major functions documented
79-1%	Estimate of the level of software documentation
0%	No software documentation

How to improve this score

This score can improve by adding content to the requirements document such that it comprehensively covers the requirements. For guidance, refer to the SecurEth System Description Document . Using tools that aid traceability detection will help.

9) Are there sufficiently detailed comments for all functions within the deployed contract code (%)

> Answer: 24%

ShibaSwap has a Comment to Code ratio of 24%.

The Comments to Code (CtC) ratio is the primary metric for this score.

Guidance:
100%	CtC > 100 Useful comments consistently on all code
90-70%	CtC > 70 Useful comment on most code
60-20%	CtC > 20 Some useful commenting
0%	CtC < 20 No useful commenting

How to improve this score

This score can improve by adding comments to the deployed code such that it comprehensively covers the code. For guidance, refer to the SecurEth Software Requirements.

10) Is it possible to trace from software documentation to the implementation in code (%)

> Answer: 0%

No software documentation in any **official developer-supported documentation.**

Guidance:
100%	Clear explicit traceability between code and documentation at a requirement
	level for all code
60%	Clear association between code and documents via non explicit traceability
40%	Documentation lists all the functions and describes their functions
0%	No connection between documentation and code

How to improve this score

This score can improve by adding traceability from requirements to code such that it is clear where each requirement is coded. For reference, check the SecurEth guidelines on traceability.

Testing

This section looks at the software testing available. It is explained in this document. This section answers the following questions;

11) Full test suite (Covers all the deployed code) (%)
12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)

13) Scripts and instructions to run the tests (Y/N)
14) Report of the results (%)
15) Formal Verification test done (%)
16) Stress Testing environment (%)

11) Is there a Full test suite? (%)

> ⚠ Answer: 0%

There are no tests or test reports that are publicly available in any **official developer-supported documentation, resources, or github.**

This score is guided by the Test to Code ratio (TtC). Generally a good test to code ratio is over 100%. However the reviewers best judgement is the final deciding factor.

Guidance:

100%	TtC > 120% Both unit and system test visible
80%	TtC > 80% Both unit and system test visible
40%	TtC < 80% Some tests visible
0%	No tests obvious

How to improve this score

This score can improve by adding tests to fully cover the code. Document what is covered by traceability or test results in the software repository.

12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)

> ⚠ Answer: 0%

There was no indication of any code coverage test having been done that were then published in any **official developer-supported documentation channels.**

Guidance:

100%	Documented full coverage
99-51%	Value of test coverage from documented results
50%	No indication of code coverage but clearly there is a reasonably complete set of tests
30%	Some tests evident but not complete
0%	No test for coverage seen

How to improve this score

This score can improve by adding tests achieving full code coverage. A clear report and scripts in the software repository will guarantee a high score.

13) Scripts and instructions to run the tests (Y/N)

> ⚠ Answer: No

There are no ShibaSwap tests availible publicly, and there is **NO official developer-supported documentation** with any scripts and installations available.

How to improve this score

Add the scripts to the repository and ensure they work. Ask an outsider to create the environment and run the tests. Improve the scripts and docs based on their feedback.

14) Report of the results (%)

> ⚠ Answer: 0%

There are **NO official developer-supported documentation** that include any report of the test resuts.

Guidance:

100%	Detailed test report as described below
70%	GitHub Code coverage report visible
0%	No test report evident

How to improve this score

Add a report with the results. The test scripts should generate the report or elements of it.

15) Formal Verification test done (%)

> ⚠ Answer: 0%

There is **NO official developer-supported documentation** that indicates that any formal verification testing has been done.

16) Stress Testing environment (%)

> ⚠ Answer: 0%

There is **NO official developer-supported documentation** that indicates that any Kovan or Ropsten TestNet addresses are availible.

Security

This section looks at the 3rd party software audits done. It is explained in this document. This section answers the following questions;

17) Did 3rd Party audits take place? (%)
18) Is the bounty value acceptably high?

17) Did 3rd Party audits take place? (%)

> ✓ Answer: 90%

Certik released an audit on 2021/07/27 of ShibaSwap.

It is a solid audit with changes implemented before deployment. It has 34 findings which is higher than normal. The audit highlights very high centralization risk through the multisigs, which have control of user funds.

However as per our guidance below the score must be 0% as no smart contract addresses are found.

Guidance:
100% Multiple Audits performed before deployment and results public and implemented or not required
90% Single audit performed before deployment and results public and implemented or not required
70% Audit(s) performed after deployment and no changes required. Audit report is public

50% Audit(s) performed after deployment and changes needed but not implemented
20% No audit performed
0% Audit Performed after deployment, existence is public, report is not public and no improvements deployed OR smart contract address' not found, question

Deduct 25% if code is in a private repo and no note from auditors that audit is applicable to deployed code

18) Is the bounty value acceptably high (%)

> ⚠ Answer: 0%

No ShibaSwap Bug Bounty program was found.

Guidance:

100% Bounty is 10% TVL or at least $1M AND active program (see below)
90% Bounty is 5% TVL or at least 500k AND active program

80%   Bounty is 5% TVL or at least 500k
70%   Bounty is 100k or over AND active program
60%   Bounty is 100k or over
50%   Bounty is 50k or over AND active program
40%   Bounty is 50k or over
20%   Bug bounty program bounty is less than 50k
0%   No bug bounty program offered

Active program means a third party actively driving hackers to the site. Inactive program would be static mention on the docs.

Access Controls

This section covers the documentation of special access controls for a DeFi protocol. The admin access controls are the contracts that allow updating contracts or coefficients in the protocol. Since these contracts can allow the protocol admins to "change the rules", complete disclosure of capabilities is vital for user's transparency. It is explained in this document. The questions this section asks are as follow;

19) Can a user clearly and quickly find the status of the admin controls?
20) Is the information clear and complete?
2`) Is the information in non-technical terms that pertain to the investments?
22) Is there Pause Control documentation including records of tests?

19) Can a user clearly and quickly find the status of the access controls (%)

> ✓ Answer: 100%

The highly centralized status of the access controls is detailed in the Certik audit. The multisig has very high control.

Guidance:
100%   Clearly labelled and on website, docs or repo, quick to find
70%   Clearly labelled and on website, docs or repo but takes a bit of looking
40%   Access control docs in multiple places and not well labelled
20%   Access control docs in multiple places and not labelled
0%   Admin Control information could not be found

20) Is the information clear and complete (%)

> ⓘ Answer: 60%

b) Multisig function is clearly outlined in the ShibaSwap whitepaper. (30%)
c) The capabilities for change are described clearly in the audit. (30%)

Guidance:
All the contracts are immutable -- 100% OR

a) All contracts are clearly labelled as upgradeable (or not) -- 30% AND
b) The type of ownership is clearly indicated (OnlyOwner / MultiSig / Defined Roles) -- 30% AND
c) The capabilities for change in the contracts are described -- 30%

How to improve this score

Create a document that covers the items described above. An example is enclosed.

21) Is the information in non-technical terms that pertain to the investments (%)

> ✓ Answer: 90%

Guidance:
100%   All the contracts are immutable
90%   Description relates to investments safety and updates in clear, complete non-software l language
30%   Description all in software specific language
0%   No admin control information could not be found

How to improve this score

Create a document that covers the items described above in plain language that investors can understand. An example is enclosed.

22) Is there Pause Control documentation including records of tests (%)

> ⚠ Answer: 0%

No Pause Control or similar function is listed in the ShibaSwap documentation.

Guidance:
100%   All the contracts are immutable or no pause control needed and this is explained OR
100%   Pause control(s) are clearly documented and there is records of at least one test within 3 months
80%   Pause control(s) explained clearly but no evidence of regular tests
40%   Pause controls mentioned with no detail on capability or tests
0%   Pause control not documented or explained

How to improve this score

Create a document that covers the items described above in plain language that investors can understand. An example is enclosed.

Appendices

Author Details

The author of this review is Rex of DeFi Safety.

Email : rex@defisafety.com Twitter : @defisafety

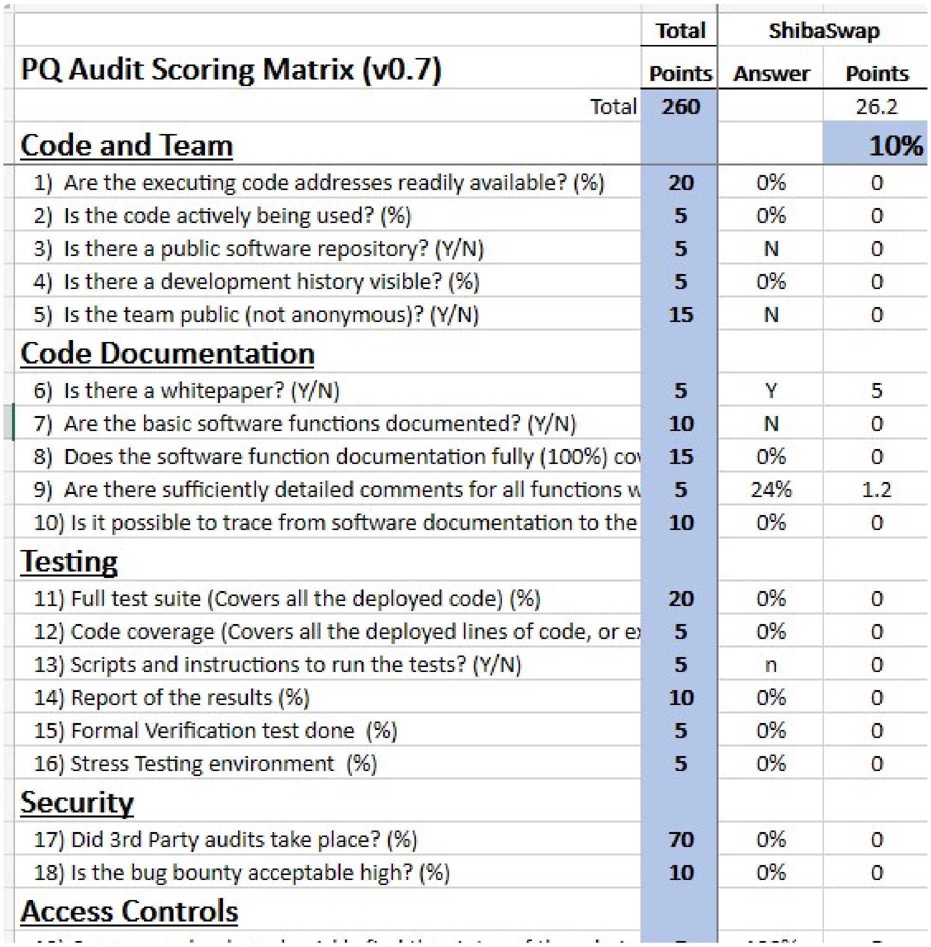
I started with Ethereum just before the DAO and that was a wonderful education. It showed the importance of code quality. The second Parity hack also showed the importance of good process. Here my aviation background offers some value. Aerospace knows how to make reliable code using quality processes.

I was coaxed to go to EthDenver 2018 and there I started SecuEth.org with Bryant and Roman. We created guidelines on good processes for blockchain code development. We got EthFoundation funding to assist in their development.

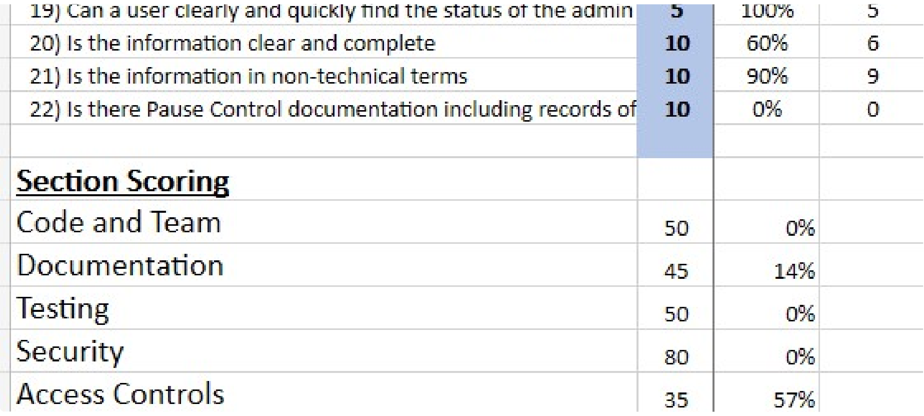
Process Quality Reviews are an extension of the SecurEth guidelines that will further increase the quality processes in Solidity and Vyper development.

DeFiSafety is my full time gig and we are working on funding vehicles for a permanent staff.

Scoring Appendix



	Total	ShibaSwap	
PQ Audit Scoring Matrix (v0.7)	Points	Answer	Points
Total	260		26.2
Code and Team			10%
1) Are the executing code addresses readily available? (%)	20	0%	0
2) Is the code actively being used? (%)	5	0%	0
3) Is there a public software repository? (Y/N)	5	N	0
4) Is there a development history visible? (%)	5	0%	0
5) Is the team public (not anonymous)? (Y/N)	15	N	0
Code Documentation			
6) Is there a whitepaper? (Y/N)	5	Y	5
7) Are the basic software functions documented? (Y/N)	10	N	0
8) Does the software function documentation fully (100%) cov	15	0%	0
9) Are there sufficiently detailed comments for all functions w	5	24%	1.2
10) Is it possible to trace from software documentation to the	10	0%	0
Testing			
11) Full test suite (Covers all the deployed code) (%)	20	0%	0
12) Code coverage (Covers all the deployed lines of code, or ex	5	0%	0
13) Scripts and instructions to run the tests? (Y/N)	5	n	0
14) Report of the results (%)	10	0%	0
15) Formal Verification test done (%)	5	0%	0
16) Stress Testing environment (%)	5	0%	0
Security			
17) Did 3rd Party audits take place? (%)	70	0%	0
18) Is the bug bounty acceptable high? (%)	10	0%	0
Access Controls			

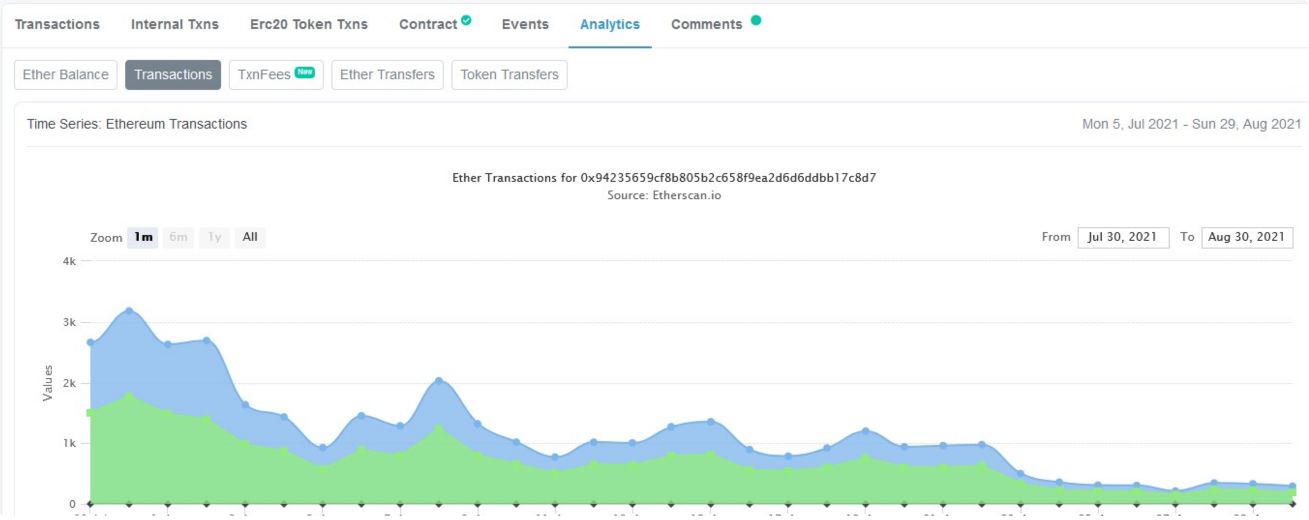


19) Can a user clearly and quickly find the status of the admin	5	100%	5
20) Is the information clear and complete	10	60%	6
21) Is the information in non-technical terms	10	90%	9
22) Is there Pause Control documentation including records of	10	0%	0
Section Scoring			
Code and Team	50	0%	
Documentation	45	14%	
Testing	50	0%	
Security	80	0%	
Access Controls	35	57%	

Executing Code Appendix

null

Code Used Appendix



Time Series: Ethereum Transactions

Mon 5, Jul 2021 - Sun 29, Aug 2021

Ether Transactions for 0x94235659cf8b805b2c658f9ea2d6d6ddbb17c8d7
Source: Etherscan.io

Example Code Appendix

```

    // function to claim all the tokens locked by user, after the locking period
    function claimAll(uint256 r) external {
        claimAllForUser(r, msg.sender);
    }

    // function to get claimable amount for any user
    function getClaimableAmount(address _user) external view returns(uint256) {
        LockInfo[] memory lockInfoArrayForUser = lockInfoByUser[_user];
        uint256 totalTransferableAmount = 0;
        uint i;
        for (i=latestCounterByUser[_user]; i<lockInfoArrayForUser.length; i++){
            uint256 lockingPeriodHere = lockingPeriod;
            if(lockInfoArrayForUser[i]._isDev){

                lockingPeriodHere = devLockingPeriod;
            }
            if(now >= (lockInfoArrayForUser[i]._timestamp.add(lockingPeriodHere))){
                totalTransferableAmount = totalTransferableAmount.add(lockInfoArrayForUser
            } else {
                break;
            }
        }
        return totalTransferableAmount;
    }

    // get the left and right headers for a user, left header is the index counter till wh
    function getLeftRightCounters(address _user) external view returns(uint256, uint256){
        return(latestCounterByUser[_user], lockInfoByUser[_user].length);
    }

    // in cases of emergency, emergency address can set this to true, which will enable em
    function setEmergencyFlag(bool _emergencyFlag) external {
        require(msg.sender == emergencyAddress, "This function can only be called by emerge
        emergencyFlag = _emergencyFlag;
    }

    // function for owner to transfer all tokens to another address
    function emergencyWithdrawOwner(address _to) external onlyOwner{
        uint256 amount = boneToken.balanceOf(address(this));
        require(boneToken.transfer(_to, amount), 'MerkleDistributor: Transfer failed.');
    }

    // emergency address can be updated from here
    function setEmergencyAddr(address _newAddr) external {
        require(msg.sender == emergencyAddress, "This function can only be called by emerge
        require(_newAddr != address(0), "_newAddr is a zero address");
        emergencyAddress = _newAddr;
    }

    // function to update/change the normal & dev locking period
    function setLockingPeriod(uint256 _newLockingPeriod, uint256 _newDevLockingPeriod) exte
        lockingPeriod = _newLockingPeriod;
        devLockingPeriod = _newDevLockingPeriod;
        emit LockingPeriod(msg.sender, _newLockingPeriod, _newDevLockingPeriod);
    }
}

```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex

Solidity	1	797	105	132	560	72

Comments to Code 132 / 560 = 24%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	0	0	0	0	0	0

Tests to Code 0/0 = 0%