OBELISK

OBELISK

Part of Tibereum Group

AUDITING REPORT

Version Notes

Version	No. Pages	Date	Revised By	Notes
1.0	Total: 15	31th of March 2021	MrTeaThyme, Hebilicious, Plemonade, Zapmore	Audit Draft
2.0	Total: 16	3rd of April 2021	MrTeaThyme, Hebilicious, Plemonade, Zapmore	Final Audit Report

Audit Notes

Audit Date	6th of March 2021 - 31st of March 2021
Auditor/Auditors	MrTeaThyme, Hebilicious, Plemonade
Auditor/Auditors Contact Information	Twitter: @ObeliskOrg
Notes	Code and contracts are audited for security flaws. UI/UX (website), logic, and tokenomics are not audited.
Audit Report Number	OB85684668

Disclaimer

Obelisk Auditing

Binance Smart Chain (BSC) launched in fall 2020 as a faster and cheaper alternative to Ethereum. Since then BSC has seen exponential growth to a point where there is a multitude of new projects created every day. In a fast phased world like this, there will also be an enormous amount of scams. The scams have become so elaborate that it’s hard for the common investor to trust a project, even though it could be legit. We saw a need for creating high-quality audits at a fast phase to keep up with the constantly expanding market. With the Obelisk stamp of approval, a legitimate project can easily grow its user base exponentially in a world where trust means everything. Obelisk Auditing consists of a group of security experts that specialize in security and structural operations, with previous work experience from among other things, PricewaterhouseCoopers. All our audits will always be conducted by at least two independent auditors for maximum security and professionalism.

As a comprehensive security firm, Obelisk provides all kinds of audits and project assistance.

Table of Content

Project Information

Project Name	Superpepebros
Description	Superpepebros is a farm with a twist. The longer you stay in a pool, the higher the bonus multiplier you get, and in return, the higher your staking reward becomes in that specific pool. The project at this time has released the farming part, but there will be many more things coming soon.
Contact	Shoostah
Contact information	https://t.me/SmokehouseFinance
Token Name(s)	superpepebros.finance
Token Short	SPB
Programming Language	Solidity
Blockchain	Binance Smart Chain (BSC)

Executive Summary

The audit of SuperPepeBros was conducted by three of Obelisks' security experts between the 6th of March 2021 - 31st of March 2021. As the contracts were not yet live as the audit started, the audit took place on the unpublished source code using our comprehensive testing tools and manual in-depth review.

After finishing the full audit, Obelisk can say that there were no critical security issues found within the audited contracts belonging to SuperPepeBros. This means that there were no findings of malicious code that could be used either by an outsider or the team in order to liquidate the projects or some parts of it. The team has not reviewed the UI/UX, logic, or tokenomics of the SuperPepeBros project.

Even though we don't actively look for logic issues, during this audit for SuperPepeBros, we found an issue with the ClaimReward logic. As our auditors forwarded this information to the SuperPepeBros team, they worked with Tibereum consultants to redesign the logic of their contracts and released SuperPepeBros v2, which we then audited again.

During the review, there were some minor issues found which can be seen in the summary table. These issues are described further in this audit.

Summary Table

Severity	Found	Mitigated	Still Open
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	3	0	3
Informational	0	0	0
Total	**3**	**0**	**3**

Audited Part	Status	Note	Mitigated
Loop over unbounded data structure	Unsolved	Low Risk	No
FeeAdressBuyBurn Is Externally Owned	Unsolved	Low Risk	No
Missing TimeLock	Unsolved	Low Risk	No

Result of Audit

The result of this audit is: PASS w. comments

Introduction

Obelisk was commissioned by SuperPepeBros on the 5th of March 2021 to conduct a comprehensive audit of their contract code. The following audit was conducted between the 6th of March 2021 - 31st of March 2021 and delivered as a final audit on the 3d of April 2021. During this time 3 of Obelisks' technical specialists did a comprehensive analysis of the underlying code and other relevant documents to find out if any vulnerabilities could be used by outsiders or the team. The comprehensive test was conducted in a specific test environment that utilized exact copies of the relevant contracts. Since the audit was done on the unpublished contracts, Obelisk came back and did an on-chain analysis once the contracts went live, to ensure that the deployed contracts matched the tested ones.

While conducting the audit, the Obelisk security team uses best practices to ensure that the reviewed contracts are thoroughly examined against all angles of attack. This is done by evaluating the codebase and whether it gives rise to significant risks. During the audit, Obelisk assesses the risks and assigns a risk level to each section together with an explanatory comment.

While starting the audit on version 1 of SuperPepeBros, we directly saw that there were significant problems with the leveling system that made the structure and implementation not functioning like it was intended by their team. Once the realization of the problems came to light, the SuperPepeBros worked with Tibereum consultants recommended by Obelisk to help figure out the cause of the problems and to code a version 2 of SuperPepeBros. Version 2 was built following BDD best practices and was deemed to be working as the original plan by the SuperPepeBros team, which then forwarded the v2 source code to Obelisk for auditing.

This audit is based on version 2 of the SuperPepeBros contracts that were not yet live in a production environment. During the audit, our security experts found some Low Impact security issues that can't resolve in the loss of user funds but could lead to undesired consequences in some scenarios.

While doing the on-chain analysis, we found that the provided Timelock contract wasn't deployed. We recommend setting MasterPepe owner to a Timelock contract of at least 24hours.

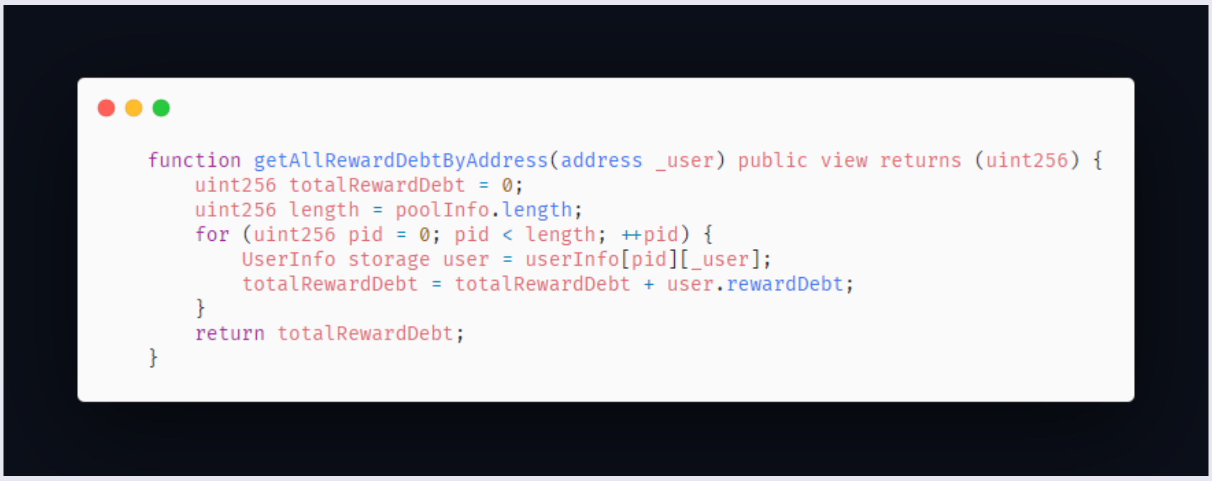
Code Audits

Manual Analysis

Loop over unbounded data structure

SEVERITY	Low Risk
LOCATION	*MasterPepe.sol => Line 293* *MasterPepe.sol => Line 460*

/MasterPepe.sol l#293



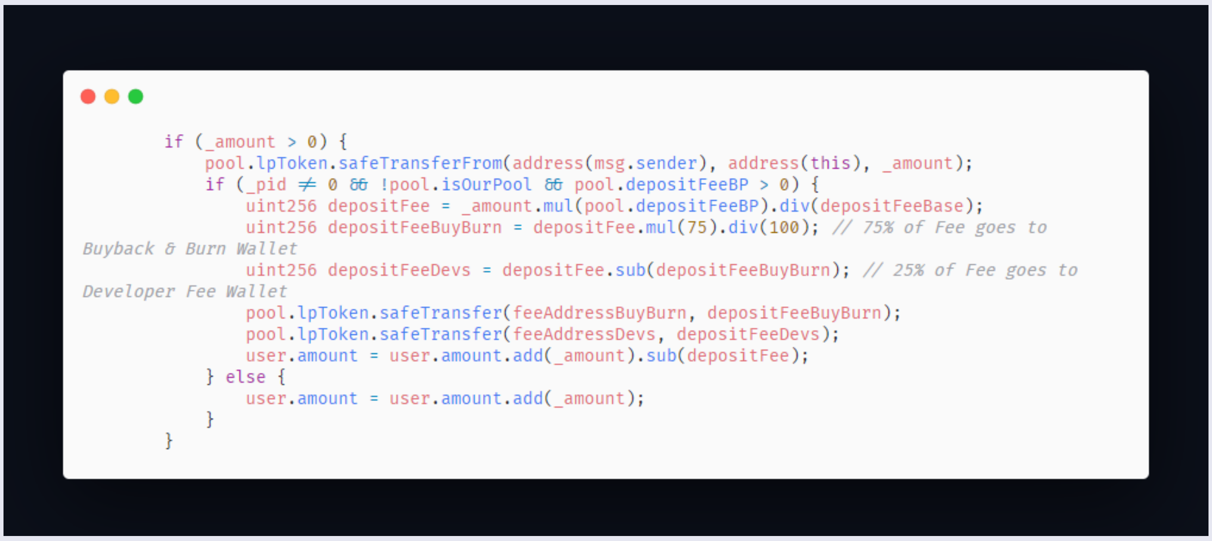
```
function getAllRewardDebtByAddress(address _user) public view returns (uint256) {
    uint256 totalRewardDebt = 0;
    uint256 length = poolInfo.length;
    for (uint256 pid = 0; pid < length; ++pid) {
        UserInfo storage user = userInfo[pid][_user];
        totalRewardDebt = totalRewardDebt + user.rewardDebt;
    }
    return totalRewardDebt;
}
```

DESCRIPTION	If the pool length grows too large, the gas required to execute the code will exceed the block gas limit, effectively causing a denial-of-service condition.
RECOMMENDATION	Avoid iteration when possible, and increment a variable for each user that tracks the appropriate value.
MITIGATED	N/A

FeeAdressBuyBurn Is Externally Owned

SEVERITY	Low Risk
LOCATION	*MasterPepe.sol Line 386*

/MasterPepe.sol#386



```
        if (_amount > 0) {
            pool.lpToken.safeTransferFrom(address(msg.sender), address(this), _amount);
            if (_pid ≠ 0 && !pool.isOurPool && pool.depositFeeBP > 0) {
                uint256 depositFee = _amount.mul(pool.depositFeeBP).div(depositFeeBase);
                uint256 depositFeeBuyBurn = depositFee.mul(75).div(100); // 75% of Fee goes to
Buyback & Burn Wallet
                uint256 depositFeeDevs = depositFee.sub(depositFeeBuyBurn); // 25% of Fee goes to
Developer Fee Wallet
                pool.lpToken.safeTransfer(feeAddressBuyBurn, depositFeeBuyBurn);
                pool.lpToken.safeTransfer(feeAddressDevs, depositFeeDevs);
                user.amount = user.amount.add(_amount).sub(depositFee);
            } else {
                user.amount = user.amount.add(_amount);
            }
        }
```

DESCRIPTION	FeeAddressBuyBurn is a fee taken on every deposit to the contract. Externally owned addresses can be used at the discretion of their owners. No contract has been provided that implements a logic that buys and burns tokens automatically.
RECOMMENDATION	Consider implementing this functionality through the smart contract.
MITIGATED	N/A

Static Analysis

No Findings

On-Chain Analysis

Summary

After conducting an on-chain analysis of the deployed contracts,
we can observe that:

- PepeToken deployed code matches the audited code
- MasterPepe deployed code matches the audited code
- No Timelock was deployed

The following addresses have been set to EOA (Externally Owned Addresses)

- devAddr, feeAddressDevs : **0xed1af9702c3ccc772dd50047e6b1e17266db85ca**
- feeAddressBuyBurn: **0xc323def8f8c8a27e426e1f6f2115a04b9af55af8**
- owner: **0xe08c3ac6ec5df597967fc3e97445eb47f8e729ff**

PepeToken owner has been set to MasterPepe.

Missing TimeLock

SEVERITY	Low Risk
DESCRIPTION	Upon reviewing the deployed contracts we have found no evidence of a TimeLock on any sensitive functions.
RECOMMENDATION	Change MasterPepe owner to a Timelock contract.
MITIGATED	N/A

Appendix A - Reviewed Documents

Document	**Deployed address**
MasterPepe.sol	0x15144a607722f6Ac15C30e7141ec8b16E4e6F6F8
PepeToken.sol	0xFcb055630D7F29CF16d70a3B5b51A2Aa0cC92ed0
Timelock.sol	-

Appendix B - Risk Ratings

Risk	Description
High Risk	A fatal vulnerability that can cause immediate loss of Tokens / Funds
Medium Risk	A vulnerability that can cause some loss of Tokens / Funds
Low Risk	A vulnerability that can be mitigated
Informational	No vulnerability

Appendix C - Icons

Icon	Explanation
📋	Solved by Project Team
🔍	Under Investigation of Project Team
⚠	Unsolved

Appendix C - Testing Standard

An ordinary audit is conducted using these steps.

1. Gather all information
2. Conduct a first visual inspection of documents and contracts
3. Go through all functions of the contract manually (2 independent auditors)
    a. Discuss findings
4. Use specialized tools to find security flaws
    a. Discuss findings
5. Follow up with project lead of findings
6. If there are flaws, and they are corrected, restart from step 2
7. Write and publish a report

During our audit, a thorough investigation has been conducted employing both automated analysis and manual inspection techniques. Our auditing method lays a particular focus on the following important concepts:

- Ensuring that the code and codebase use best practices, industry standards, and available libraries.
- Testing the contract from different angles ensuring that it works under a multitude of circumstances.
- Analyzing the contracts through databases of common security flaws.

OBELISK

Part of Tibereum Group