OBELISK

OBELISK

Part of Tibereum Group

AUDITING REPORT

Version Notes

Version	No. Pages	Date	Revised By	Notes
1.0	Total: 16	2021-10-02	Zapmore, Donut	Audit Final

Audit Notes

Audit Date	2021-06-18 - 2021-10-02
Auditor/Auditors	Donut, Hebilicious
Auditor/Auditors Contact Information	tibereum-obelisk@protonmail.com
Notes	Specified code and contracts are audited for security flaws. UI/UX (website), logic, team, and tokenomics are not audited.
Audit Report Number	OB58888759

Disclaimer

This audit is not financial, investment, or any other kind of advice and is for informational purposes only. This report is not a substitute for doing your own research and due diligence. Obelisk is not responsible or liable for any loss, damage, or otherwise caused by reliance on this report for any purpose. Obelisk has based this audit report solely on the information provided by the audited party and on facts that existed before or during the audit being conducted. Obelisk is not responsible for any outcome, including changes done to the contract/contracts after the audit was published. This audit is fully objective and only discerns what the contract is saying without adding any opinion to it. The audit is paid by the project but neither the auditors nor Obelisk has any other connection to the project and has no obligations other than to publish an objective report. Obelisk will always publish its findings regardless of the outcome of the findings. The audit only covers the subject areas detailed in this report and unless specifically stated, nothing else has been audited. Obelisk assumes that the provided information and material were not altered, suppressed, or misleading. This report is published by Obelisk, and Obelisk has sole ownership of this report. Use of this report for any reason other than for informational purposes on the subjects reviewed in this report including the use of any part of this report is prohibited without the express written consent of Obelisk.

Obelisk Auditing

Defi is a relatively new concept but has seen exponential growth to a point where there is a multitude of new projects created every day. In a fast-paced 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.

Audit Information

The auditors always conducted a manual visual inspection of the code to find security flaws that automatic tests would not find. Comprehensive tests are also conducted in a specific test environment that utilizes exact copies of the published contract.

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. Take note that the comments from the project team are their opinion and not the opinion of Obelisk.

Table of Content

Project Information

Project Name	Cafeswap
Description	CafeSwap is a yield farming and staking platform on BSC Chain and Polygon.
Website	https://cafeswap.finance/
Contact	@CafeSwapChef
Contact information	@CafeSwapChef on TG
Token Name(s)	N/A
Token Short	N/A
Contract(s)	See Appendix A
Code Language	Solidity
Chain	Polygon

Executive Summary

The audit of Cafeswap was conducted by two of Obelisks' security experts between the 18th of June 2021 and the 2nd of October 2021.

Only the Polygon farming contract of the Cafeswap project has been audited by Obelisk. The Low-Risk finding in the contract has been mitigated. Finding #3 is partially solved since a timelock is implemented on the MasterChefV2 contract, which is set to an externally owned account, but the timelock is only set to 12 hours instead of the recommended 72 hours.

The team has not reviewed the UI/UX, logic, team, or tokenomics of the Cafeswap project.

Please read the full document for a complete understanding of the audit.

Summary Table

Audited Part	ID	Severity	Status
Minters Cannot Be Easily Identified	#0001	Low Risk	Closed
Loss Of Precision When Calculation Rewards	#0002	Informational	Open
No Timelock Contract	#0003	High Risk	Partially Mitigated

Introduction

Obelisk was commissioned by Cafeswap on the 17th of June 2021 to conduct a comprehensive audit of Cafeswap's Polygon farming contract. The following audit was conducted between the 18th of June 2021 and the 2nd of October 2021. Two of Obelisk's security experts went through the related contracts using industry standards to find if any vulnerabilities could be exploited.

The reason for the unusually long audit time was purely due to the wait of deploying the contracts on Polygon in order for us to be able to do an On-Chain analysis.

The Cafeswap Polygon farming contracts were checked for vulnerabilities prior to deployment. The audit found a Low-Risk issue in the contracts related to that the minters can't easily be identified by the users. As anyone should be able to verify and account for all minters, this could pose a Low-Risk issue. This issue was solved by the Cafeswap team and is fully mitigated.

After the contracts were deployed, there was an on-chain analysis conducted in order to check on the implementation of the contracts and if they match the audited ones. The deployed contracts matched the audited ones. The issue #3 which were found during the on-chain analysis was partially solved by implementing a timelock with a 12 hour delay for the ownership of the MasterChefV2 contract. For the issue to be fully solved, there needs to be a 72 hour delay in order for people to have time to react to changes.

The informational finding are good to know while interacting with the project but don't directly damage the project in its current state, hence it's up to the project team if it's worth solving the issue.

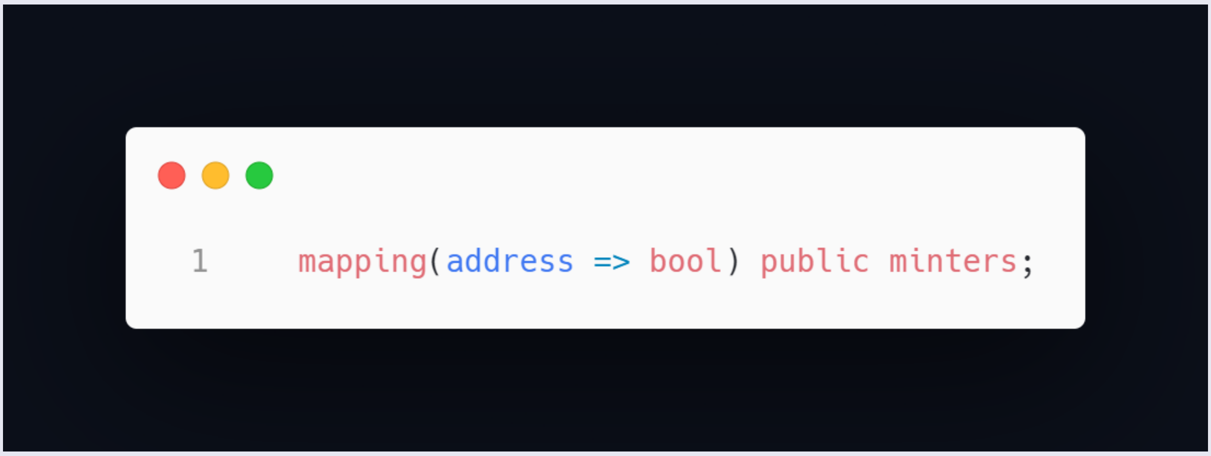
Please see each section of the audit to get a full understanding of the audit.

Findings

Manual Analysis

Minters Cannot Be Easily Identified

SEVERITY	Low Risk
FINDING ID	#0001
STATUS	**Closed**
LOCATION	CoffeeToken.sol -> 13

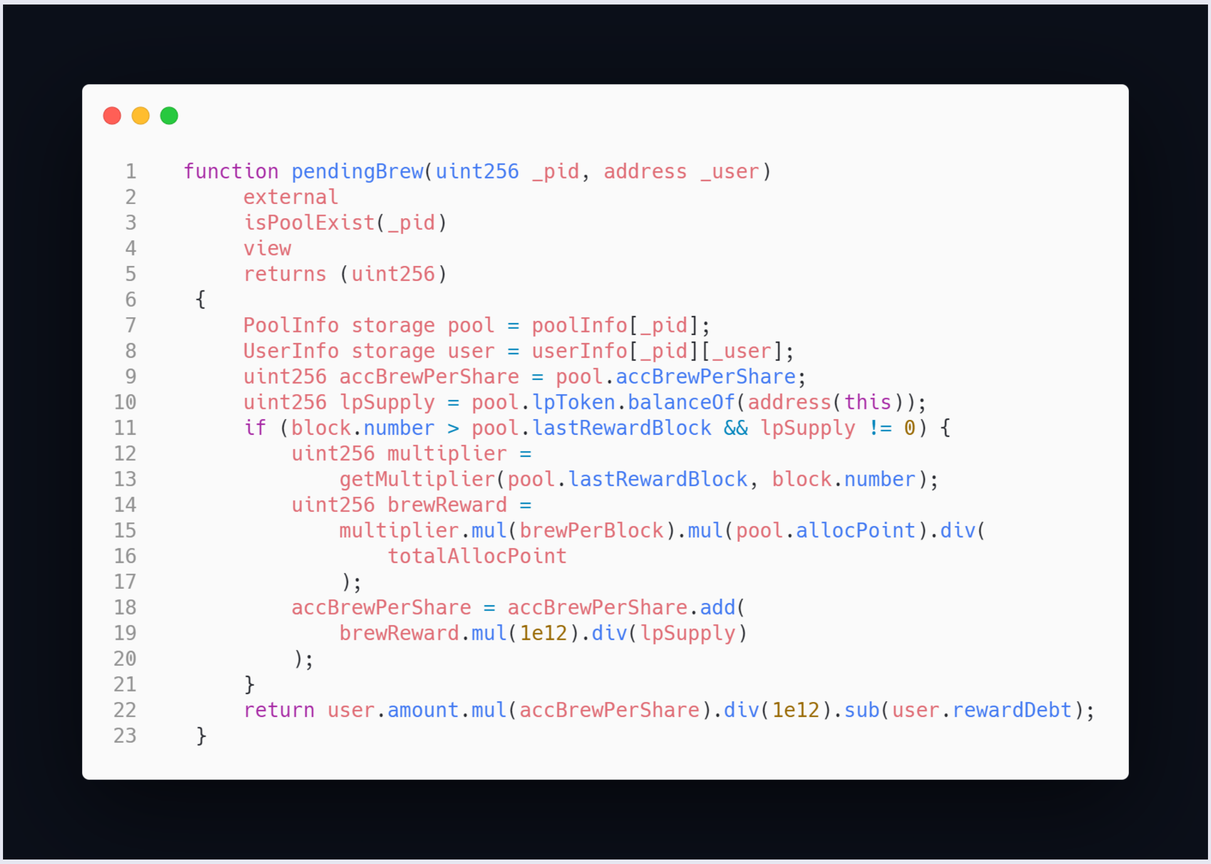


```
mapping(address => bool) public minters;
```

DESCRIPTION	The minting addresses cannot be easily identified by users. It is important to ensure that all addresses that can mint are accounted for.
RECOMMENDATION	Add an enumerated list of all minter addresses. Add events that emit when an address is added or removed as a minter.
RESOLUTION	A list of possible minters was added as well as an event emitted upon the change of minters. Note: listOfMinters may contain addresses no longer considered minters, but they can be checked

Loss Of Precision When Calculation Rewards

SEVERITY	Informational
FINDING ID	#0002
STATUS	**Open**
LOCATION	MasterChefV2.sol -> 165-187



```
function pendingBrew(uint256 _pid, address _user)
     external
     isPoolExist(_pid)
     view
     returns (uint256)
 {
     PoolInfo storage pool = poolInfo[_pid];
     UserInfo storage user = userInfo[_pid][_user];
     uint256 accBrewPerShare = pool.accBrewPerShare;
     uint256 lpSupply = pool.lpToken.balanceOf(address(this));
     if (block.number > pool.lastRewardBlock && lpSupply != 0) {
         uint256 multiplier =
             getMultiplier(pool.lastRewardBlock, block.number);
         uint256 brewReward =
             multiplier.mul(brewPerBlock).mul(pool.allocPoint).div(
                 totalAllocPoint
             );
         accBrewPerShare = accBrewPerShare.add(
             brewReward.mul(1e12).div(lpSupply)
         );
     }
     return user.amount.mul(accBrewPerShare).div(1e12).sub(user.rewardDebt);
 }
```

DESCRIPTION	The factor used to calculate the rewards is 1e12. The typical token has 18 figures of precision. As a result of this discrepancy, small amounts of reward will be lost to rounding.
RECOMMENDATION	Use a factor of 1e18 when calculating rewards.
RESOLUTION	N/A

Static Analysis

No Findings

On-Chain Analysis

No Timelock Contract

SEVERITY	High Risk
FINDING ID	#0003
STATUS	**Partially Mitigated**
LOCATION	CoffeeToken 0xb5106A3277718eCaD2F20aB6b86Ce0Fee7A21F09 MasterChefV2 0xca2DeAc853225f5a4dfC809Ae0B7c6e39104fCe5

DESCRIPTION	The ownership of the CafeSwap Token (pBREW) token and the MasterChefV2 contract is set to an externally owned account (EOA). This allows the external account to freely mint tokens, modify pools, and the token emission rate.
RECOMMENDATION	Provide a timelock to allow users to respond to changes. A time lock of at least 72 hours is recommended.
RESOLUTION	A timelock was added for these contracts: CoffeeToken Timelock 0xf9dED50213ffdDd2c51Da47750eE943b5460F39e delay: 48 hours MasterChefV2 Timelock 0x83f839b7E9Ee8C322a9CcfeC469283A1E3184f70 delay: 12 hours Project team comment: “The MasterChef timelock is kept to the standard of 12 hours which is used by pancakeswap and cafeswap from february, it will allow us to add new farms and partnership pools faster and will allow us to proceed with business development faster. The token contract has a 48 hours timelock which is more than enough according to us.”

Appendix A - Reviewed Documents

Document	Address
CafeSwapSafeTransfer.sol	0x8110C0755Cd5f1e082b3aA2f05c69419Abdfb65F
CoffeeToken.sol	0xb5106A3277718eCaD2F20aB6b86Ce0Fee7A21F09
MasterChefV2.sol	0xca2DeAc853225f5a4dfC809Ae0B7c6e39104fCe5
Timelock.sol	0xf9dED50213ffdDd2c51Da47750eE943b5460F39e 0x83f839b7E9Ee8C322a9CcfeC469283A1E3184f70

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 D - 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.

Follow Obelisk Auditing for the Latest Information

ObeliskOrg ObeliskOrg

OBELISK

Part of Tibereum Group