Security Assessment

BSCStation

Aug 26th, 2021

Table of Contents

Summary

This report has been prepared for BSC Station Ltd to discover issues and vulnerabilities in the source code of the BSCStation project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	BSCStation
Platform	Ethereum, BSC
Language	Solidity
Codebase	https://bscscan.com/address/0x587d14bb7bdc34aa1061c4e6f7f495d6af8fad36#code https://github.com/BSCStationSwap/contract
Commit	https://bscscan.com/address/0x587d14bb7bdc34aa1061c4e6f7f495d6af8fad36#code fb8394d1064709be0e90fb0a808d824c7228cc71

Audit Summary

Delivery Date	Aug 26, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

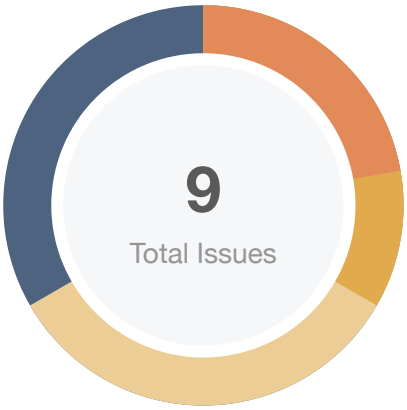
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	2	0	0	1	0	1
Medium	1	0	0	1	0	0
Minor	3	0	0	1	1	1
Informational	3	0	0	1	1	1
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
DPE	DPETBSCSStartPools.sol	24317a11bfe51c1b06cdbe424b19c2feaad5594d5566f3b7e456d58096f92a6b

Findings



Critical	**0** (0.00%)
Major	**2** (22.22%)
Medium	**1** (11.11%)
Minor	**3** (33.33%)
Informational	**3** (33.33%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
DPE-01	Privileged ownership	**Centralization / Privilege**	**Major**	Acknowledged
DPE-02	Redundant Statement	Volatile Code	Informational	Resolved
DPE-03	Lack of Input Validation	Volatile Code	Informational	Partially Resolved
DPE-04	Incompatibility With Deflationary Tokens	Volatile Code	Minor	Acknowledged
DPE-05	Mint and burn tokens	Logical Issue	Informational	Acknowledged
DPE-06	Centralized risk in `collectFee`	**Centralization / Privilege**	**Minor**	Partially Resolved
DPE-07	Lack of using safeTransfer	Volatile Code	Minor	Resolved
DPE-08	Lack of reward distributing	Logical Issue	Medium	Acknowledged
DPE-09	Risk of transfer the staked token	Logical Issue	Major	Resolved

DPE-01 | Privileged ownership

Category	Severity	Location	Status
Centralization / Privilege	**Major**	DPETBSCSStartPools.sol: 814, 1214, 1226, 1252, 1267, 1274, 1278, 1283, 1294, 1315, 1335, 1355, 1393, 1432, 1596, 1624	Acknowledged

Description

The owner of contract `DPETBSCSStartPools` has the permission to:

1. removing the staking token by calling the function `emergencyRemoval`,
2. withdraw reward tokens by calling the function `emergencyRewardWithdraw`,
3. update `feePeriod`, `unStakingFee`, `feeCollector`, `poolLimitPerUser`, `hasUserLimit`, `poolCap`, `hasPoolLimit`, `rewardPerBlock`, `startBlock`, `bonusEndBlock`, `lastRewardBlock`, `stakingBlock`, `stakingEndBlock` and `unStakingBlock`.
4. call the function `stopReward`,
5. call the function `recoverWrongTokens`,
6. call the function `lock`,
7. add/remove reward tokens,

without obtaining the consensus of the community.

Recommendation

Renounce ownership when the timing is appropriate, or gradually migrate to a timelock plus multisig governing procedure and let the community monitor in respect to transparency considerations.

Alleviation

No alleviation.

DPE-02 | Redundant Statement

Category	Severity	Location	Status
Volatile Code	Informational	DPETBSCSStartPools.sol: 1279	Resolved

Description

`_newFee` is a uint256, so `_newFee >= 0` will be always true.

Recommendation

Consider removing the redundant statement.

Alleviation

The development team heeded our advice and resolved this issue in commit d0af25923bf60dcc22d51ae923b6b9036cdc293d

DPE-03 | Lack of Input Validation

Category	Severity	Location	Status
Volatile Code	Informational	DPETBSCSStartPools.sol: 1007~1008, 1014, 1018	Partially Resolved

Description

Below address-type arguments in the constructor are not validated as the non-zero addresses to prevent error:

1. _stakedToken
2. _rewardTokens
3. _feeCollector
4. _admin

Recommendation

We advise the client to check if the values are set as the non-zero addresses.

Alleviation

The development team heeded our advice and partially resolved this issue in commit e62d804eebe42d6ca530b8f4f82ee9e9645af4fd.

DPE-04 | Incompatibility With Deflationary Tokens

Category	Severity	Location	Status
Volatile Code	Minor	DPETBSCSStartPools.sol	Acknowledged

Description

When transferring standard ERC20 deflationary tokens, the input amount may not be equal to the received amount due to the charged transaction fee. For example, if a user stakes 100 deflationary tokens (with a 10% transaction fee) in the `DPETBSCSStartPools` contract, only 90 tokens will actually arrive in the contract. However, the user can still withdraw 100 tokens from the contract, which causes the contract to lose 10 tokens in this manner of transaction.

The `DPETBSCSStartPools` takes the pool token balance into account when calculating the users' rewards. An attacker can repeat the process of deposit and withdraw to lower the token balance in a deflationary token pool and cause the contract to increase the reward amount.

Reference: https://thoreum-finance.medium.com/what-exploit-happened-today-for-gocerberus-and-garuda-also-for-lokum-ybear-piggy-caramelswap-3943ee23a39f

Recommendation

We advise the client to regulate the set of pool tokens supported and add necessary mitigation mechanisms to keep track of accurate balances if there is a need to support deflationary tokens.

Alleviation

No alleviation.

DPE-05 | Mint and burn tokens

Category	Severity	Location	Status
Logical Issue	Informational	DPETBSCSStartPools.sol: 1124	Acknowledged

Description

The contract will mint the tokens to the user who user deposits the `stakedToken` token and burns those tokens from the user when said is user withdrawing, is this designed as expected? What if the user transferred these tokens out by mistake before he/she is withdrawing?

Recommendation

Consider informing the user to keep the minted tokens when depositing, since those minted tokens are required, to withdraw their staking token.

Alleviation

No alleviation.

DPE-06 | Centralized risk in `collectFee`

Category	Severity	Location	Status
Centralization / Privilege	**Minor**	DPETBSCSStartPools.sol: 1185	Partially Resolved

Description

```
1     function collectFee(uint256 _amount, UserInfo memory user)
2         internal
3         returns (uint256)
4     {
5         uint256 blockPassed = block.number.sub(user.lastStakingBlock);
6         if (feePeriod == 0 || (feePeriod > 0 && feePeriod >= blockPassed)) {
7             uint256 collectedAmt = _amount.mul(unStakingFee).div(10000);
8             ERC20(stakedToken).transfer(feeCollector, collectedAmt);
9             return _amount.sub(collectedAmt);
10        }
11        return _amount;
12    }
```

The `collectFee` function calls the `ERC20(stakedToken).transfer()` function with the `to` address specified as `feeCollector` for acquiring the fee. As a result, over time the `feeCollector` address will accumulate a significant portion of staking token. If the `feeCollector` is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Recommendation

We recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e. Multisignature wallets.

Indicatively, here are some feasible solutions that would also mitigate the potential risk:

- Time-lock with reasonable latency, i.e. 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent single point of failure due to the private key;
- Introduction of a DAO / governance / voting module to increase transparency and user involvement.

Alleviation

No alleviation.

DPE-07 | Lack of using safeTransfer

Category	Severity	Location	Status
Volatile Code	Minor	DPETBSCSStartPools.sol: 1158, 1216	Resolved

Description

The balance of the `rewardTokens` maybe less than the amount to be transferred when transferring.

Recommendation

Consider using the safe transfer pattern as shown below:

```
function safeERC20Transfer(ERC20 erc20, address _to, uint256 _amount) internal {
    uint256 balance = erc20.balanceOf(address(this));
    if (_amount > balance) {
        erc20.transfer(_to, balance);
    } else {
        erc20.transfer(_to, _amount);
    }
}
```

Alleviation

The development team heeded our advice and resolved this issue in commit fb8394d1064709be0e90fb0a808d824c7228cc71.

DPE-08 | Lack of reward distributing

Category	Severity	Location	Status
Logical Issue	Medium	DPETBSCSStartPools.sol: 1624	Acknowledged

Description

Lack of reward distributing before removing a reward token, is this designed as expected?

Recommendation

Consider distributing the reward token to users first before removing a reward token.

Alleviation

No alleviation.

DPE-09 | Risk of transfer the staked token

Category	Severity	Location	Status
Logical Issue	Major	DPETBSCSStartPools.sol: 1748~1753, 1764~1768	Resolved

Description

There are two functions, `transfer` and `transferFrom`, that are able to transfer staked tokens. It will result in a future failure of deposits. Example:

1. Alice stakes 20 tokens,
2. Bob stakes 100 tokens. We assume that `rewardDebt[rewardTokens[i]]` of Bob is `10*x`,
3. Bob transfer 90 tokens to Alice, then, Bob has the balance of 10 tokens,
4. Bob try to deposit another 100 tokens,

we also assume that step 2, 3 and 4 happen in the same block for easy understanding, so in step 4, we need to calculate the pending reward of Bob by the following formula:

```
1103      pending = user
1104      .amount
1105      .mul(accTokenPerShare[rewardTokens[i]])
1106      .div(PRECISION_FACTOR[rewardTokens[i]])
1107      .sub(user.rewardDebt[rewardTokens[i]]);
```

It is noted that the calculation of `rewardDebt` as below:

```
1127      user.rewardDebt[rewardTokens[i]] = user
1128      .amount
1129      .mul(accTokenPerShare[rewardTokens[i]])
1130      .div(PRECISION_FACTOR[rewardTokens[i]]);
```

It is easy to understand that the calculation of `pending` in line 1103 is just subtract `rewardDebt` for now to the before `rewardDebt`. Pay attention that `rewardDebt` is proportional to staking amount, thus, it is easy to conclude that the `rewardDebt` for now will be `x` which is less than the `rewardDebt` before which is `10*x`, it because after the step 3, Bob's staked amount is now 10 that is 10 percent as it was before. Thus steps 2, 3, and 4 will happen in the same block. Finally, the calculation in line 1103 will revert.

The reverting of calculation after the transfer of staked token is also easier to happen even if the step 2, 3 and 4 are not in the same block.

Recommendation

Consider forbidding the transferring of staked tokens in the contract.

Alleviation

The development team heeded our advice and resolved this issue in commit 305b883aa66e8eef871e11e11c8f320bb66ed16e.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.