CERTIK

Code Security Assessment

BOT PLANET

Jan 31st, 2022

Table of Contents

Summary

This report has been prepared for BOT PLANET to discover issues and vulnerabilities in the source code of the BOT PLANET 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	BOT PLANET
Platform	bsc
Language	Solidity
Codebase	https://bscscan.com/address/0x6fa690040946f49b9257b2f04f0611de55a5af9f#code https://testnet.bscscan.com/address/0xFd109F06cBe28b0c7c07c08edd1ECF413f2b8652#code
Commit	N/A

Audit Summary

Delivery Date	Jan 31, 2022
Audit Methodology	Static Analysis, Manual Review

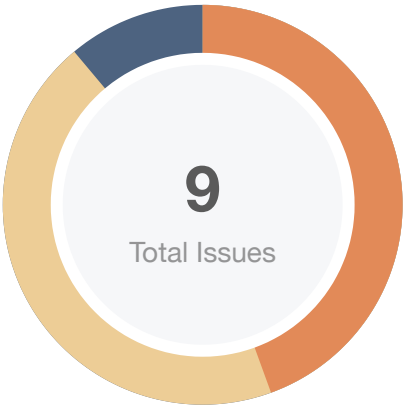
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Critical	0	0	0	0	0	0	0
Major	4	0	0	0	0	1	3
Medium	0	0	0	0	0	0	0
Minor	4	0	0	0	0	0	4
Informational	1	0	0	0	0	0	1
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
TOK	Token.sol	a9ddede5c2766736fd4cfaa4628238da5031dc754886e1f60afb8b4d440b156e

Findings



Critical	**0** (0.00%)
Major	**4** (44.44%)
Medium	**0** (0.00%)
Minor	**4** (44.44%)
Informational	**1** (11.11%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Centralization Related Risks	**Centralization / Privilege**	**Major**	Resolved
TOK-01	Initial Token Distribution	**Centralization / Privilege**	**Major**	Mitigated
TOK-02	Possible to Gain Ownership after Renouncing the Contract Ownership	Logical Issue	Major	Resolved
TOK-03	Incorrect Error Message	Logical Issue	Minor	Resolved
TOK-04	Variable `_rOwned[account]` Not Updated in Function `removeFromWhiteList()`	Control Flow	Minor	Resolved
TOK-05	Missing Initialization for Important Variables	Control Flow	Minor	Resolved
TOK-06	Upper Limit for Total Fees is not Reasonable	Logical Issue	Major	Resolved
TOK-07	Redundant Code	Logical Issue	Informational	Resolved
TOK-08	Excessive gas consumption in function `_transfer()`	Gas Optimization	Minor	Resolved

GLOBAL-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	Major	Global	Resolved

Description

In the contract `Token.sol`, the role `owner` has authority over the following functions:

- `excludeFromReward()`
- `includeInReward()`
- `excludeFromFee()`
- `includeInFee()`
- `setAllFeePercent()`
- `setBuybackUpperLimit()`
- `setMaxTxPercent()`
- `setMaxWalletPercent()`
- `setSwapAndLiquifyEnabled()`
- `recoverBEP20()`

Any compromise to the `owner` account may allow a hacker to take advantage of this authority and disrupt the operation of the token contract.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND

- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles. OR
- Remove the risky functionality.

Alleviation

Privileged functions are either removed from the contract or can't be accessed by the owner anymore.

TOK-01 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	**Major**	Token.sol (1): 791	Mitigated

Description

All of the `BOT` tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute `BOT` tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

Alleviation

[Bot Planet team] The tokens of this contract after its creation will go to one "Gnosis Safe Multisig" wallet, which is now protected by a multi-signature. We would also like to draw your attention to our explanation of how the tokens will be distributed. This information is transparent to the entire community even before the creation of a contract on our website https://www.botpla.net/#tokenomics.

TOK-02 | Possible To Gain Ownership After Renouncing The Contract Ownership

Category	Severity	Location	Status
Logical Issue	Major	Token.sol (1): 523~536	Resolved

Description

An owner is possible to gain ownership of the contract even if he calls function `renounceOwnership` to renounce the ownership. This can be achieved by performing the following operations:

1. Call `lock` to lock the contract. The variable `_previousOwner` is set to the current owner.
2. Call `unlock` to unlock the contract.
3. Call `renounceOwnership` to leave the contract without an owner.
4. Call `unlock` to regain ownership.

Recommendation

We advise updating/removing `lock` and `unlock` functions in the contract; or removing the `renounceOwnership` if such a privilege retains at the protocol level. If timelock functionality could be introduced, we recommend using the implementation of Compound finance as reference.

Reference: https://github.com/compound-finance/compound-protocol/blob/master/contracts/Timelock.sol

Alleviation

The team heeded our advice and resolved the issue in the new version.

TOK-03 | Incorrect Error Message

Category	Severity	Location	Status
Logical Issue	Minor	Token.sol (1): 914	Resolved

Description

The error message in `require(_isExcluded[account], "Already excluded")` does not describe the error correctly.

Recommendation

The message "Already excluded" can be changed to "Account is not excluded" .

Alleviation

The team heeded our advice and resolved the issue in the new version.

TOK-04 | Variable `_rOwned[account]` Not Updated In Function `removeFromWhiteList()`

Category	Severity	Location	Status
Control Flow	Minor	Token.sol (1): 913~924	Resolved

Description

The function below has a known bug.

```
913     function includeInReward(address account) external onlyOwner() {
914         require(_isExcluded[account], "Account is not excluded");
915         for (uint256 i = 0; i < _excluded.length; i++) {
916             if (_excluded[i] == account) {
917                 _excluded[i] = _excluded[_excluded.length - 1];
918                 _tOwned[account] = 0;
919                 _isExcluded[account] = false;
920                 _excluded.pop();
921                 break;
922             }
923         }
924     }
```

Variable `_rOwned[account]` is not updated in the function `includeInReward()`, which will make the accounts included siphon off the tokens out of the balances of all token holders.

Details of this finding can be seen in this article from Pera Finance: Link

Recommendation

We recommend updating `_rOwned[account]` before setting `_tOwned[account]` to 0.

Sample code:

```
    function includeInReward(address account) external onlyOwner() {
        require(_isExcluded[account], "Already excluded");
        for (uint256 i = 0; i < _excluded.length; i++) {
            if (_excluded[i] == account) {
                _excluded[i] = _excluded[_excluded.length - 1];
                _rOwned[account] = _tOwned[account].mul(_getRate());  // update
_rOwned[account]
                _tOwned[account] = 0;
                _isExcluded[account] = false;
                _excluded.pop();
                break;
```

```
            }
        }
    }
```

Alleviation

The team deleted this function in the new version.

TOK-05 | Missing Initialization For Important Variables

Category	Severity	Location	Status
Control Flow	Minor	Token.sol (1): 761~763	Resolved

Description

In the current implementation, some important global variables are not properly initialized in the declaration. This might disrupt some functionalities of the token contract if the deployer failed to pass proper values of these variables to the constructor function.

```
762     uint256 public _maxTxAmount;
763     uint256 public _maxWalletAmount;
764     uint256 public numTokensSellToAddToLiquidity;
```

Recommendation

We recommend setting proper initial values for the linked variables instead of the constructor phase assignment.

Alleviation

The team heeded our advice and resolved the issue in the new version.

TOK-06 | Upper Limit For Total Fees Is Not Reasonable

Category	Severity	Location	Status
Logical Issue	Major	Token.sol (1): 935~946	Resolved

Description

The current upper limit for transaction fees can be set as high as 50%, which is not a reasonable value.

```
  935    function setAllFeePercent(uint8 taxFee, uint8 liquidityFee, uint8 burnFee, uint8
walletFee, uint8 buybackFee) external onlyOwner() {
  936          require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
  937          require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
  938          require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
  939          require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
  940          require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
  941          _taxFee = taxFee;
  942          _liquidityFee = liquidityFee;
  943          _burnFee = burnFee;
  944          _buybackFee = buybackFee;
  945          _walletFee = walletFee;
  946      }
```

Recommendation

We recommend adding a upper limit for total fee and set it to an appropriate value such as 15%.

Alleviation

The team heeded our advice and resolved the issue in the new version.

TOK-07 | Redundant Code

Category	Severity	Location	Status
Logical Issue	Informational	Token.sol (1): 1260~1261	Resolved

Description

The condition `!_isExcluded[sender] && !_isExcluded[recipient]` can be included in `else` .

Recommendation

The following code can be removed:

```
... else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
    _transferStandard(sender, recipient, amount);
} ...
```

Alleviation

The team heeded our advice and resolved the issue in the new version.

TOK-08 | Excessive Gas Consumption In Function `_transfer()`

Category	Severity	Location	Status
Gas Optimization	Minor	Token.sol (1): 1127	Resolved

Description

```
1   if(buybackFee !=0) {
2       uint256 balance = address(this).balance;
3       if (balance > uint256(1 * 10**_decimals)) {
4           if (balance > _buyBackUpperLimit) {
5               balance = _buyBackUpperLimit;
6           }
7
8           // 1% of balance/buyBackUpperLimit for buyBack
9           uint256 amountBuyBackTokens = balance.div(100);
10          buyBackTokens(amountBuyBackTokens);
11      }
12  }
```

In the current implementation of `_transfer()`, each token transaction will trigger `buyBackTokens()`, which significantly increases the gas consumption of token transactions.

Recommendation

We recommend the client use an "all at once" method to deal with the buyback logic. The code snippet below provides a possible solution:

```
            if(overMinTokenBalance) {
                contractTokenBalance = numTokensSellToAddToLiquidity;
                //add liquidity
                swapAndLiquify(contractTokenBalance);
                if(_buybackFee !=0){
                  uint256 balance = address(this).balance;
                  if (balance > uint256(1 * 10**18)) {

                      if (balance > buyBackUpperLimit)
                          balance = buyBackUpperLimit;

                      buyBackTokens(balance);
                  }
              }
          }
```

Alleviation

The team heeded our advice and resolved the issue at this version: https://github.com/BOTDeFi/Smart-Contracts/tree/4b85d9733270da6a3bceacbfa66b0e2a4822e676.

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.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

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.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

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.