CERTIK

Preliminary Comments

Bolix Token

May 20th, 2022

Table of Contents

Summary

This report has been prepared for Bolix to discover issues and vulnerabilities in the source code of the Bolix Token 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	Bolix Token
Platform	Ethereum
Language	Solidity
Codebase	Private Codebase
Commit	

Audit Summary

Delivery Date	May 20, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

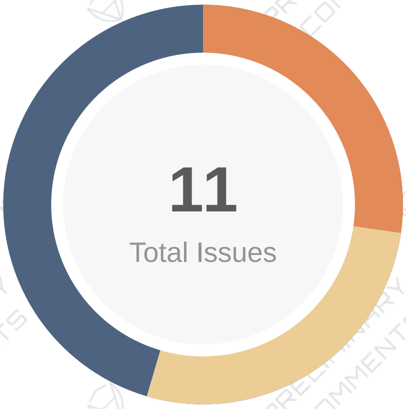
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0	0
● Major	3	3	0	0	0	0	0
● Medium	0	0	0	0	0	0	0
● Minor	3	3	0	0	0	0	0
● Informational	5	5	0	0	0	0	0
● Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
BCK	Bolix.sol	69329518b3e90f3f1b48c8e74459892d8cc9eabcdda702a8621e4286a70a723a

Findings



🟥 **Critical**	**0** (0.00%)
🟧 **Major**	**3** (27.27%)
🟨 **Medium**	**0** (0.00%)
🟨 **Minor**	**3** (27.27%)
🟦 **Informational**	**5** (45.45%)
🟩 **Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
BCK-01	Centralization Related Risks In Contract `Bolix`	**Centralization / Privilege**	● **Major**	Pending
BCK-02	Unused Event	Coding Style	● Informational	Pending
BCK-03	Initial Token Distribution	**Centralization / Privilege**	● **Major**	Pending
BCK-04	Centralization Risk Related To `addLiquidityETH`	**Centralization / Privilege**	● **Major**	Pending
BCK-05	Potential Sandwich Attacks	Logical Issue	● Minor	Pending
BCK-06	Usage Of `transfer()` For Sending Ether	Volatile Code	● Minor	Pending
BCK-07	Potential Denial-of-Service Attack	Logical Issue	● Minor	Pending
BCK-08	Unhandled Return Value	Logical Issue	● Informational	Pending
BCK-09	Missing Emit Events	Coding Style	● Informational	Pending
BCK-10	Function Visibility Optimization	Gas Optimization	● Informational	Pending
BCK-11	Redundant SafeMath Usage	Language Specific	● Informational	Pending

BCK-01 | Centralization Related Risks In Contract `Bolix`

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	Bolix.sol: 317, 322, 630, 636, 642, 648, 654, 661, 666, 671, 675, 683, 691, 696, 708, 717, 725, 898	ⓘ Pending

Description

In contract `Bolix`, the owner has the authority over the following functions:

- `Bolix.removeTokens`: Transfer tokens owned by the contract
- `Bolix.enableTrading`: Enable trading
- `Bolix.removeLimits`: Remove limits after token is stable
- `Bolix.disableTransferDelay`: Disable transfer delay
- `Bolix.updateSwapTokensAtAmount`: Change the minimum amount of tokens to sell from fees
- `Bolix.updateMaxTxnAmount`: Change max transaction allowed
- `Bolix.updateMaxWalletAmount`: Set new max wallet size amount
- `Bolix.excludeFromMaxTransaction`: Exclude an address from max transaction restriction
- `Bolix.updateBuyFees`: Change the buy fee
- `Bolix.updateSellFees`: Change the sell fee
- `Bolix.excludeFromFees`: Exclude address from being charged a fee
- `Bolix.setAutomatedMarketMakerPair`: Set address for AMM pair
- `Bolix.updateFeeWallet`: Change address to receive fees
- `Bolix.setSnipers`: Add address to black list
- `Bolix.delSnipers`: Remove address from black list
- `Bolix.withdrawFees`: Withdraw fees
- `Ownable.renounceOwnership`: Remove the owner
- `Ownable.transferOwnership`: Change the owner address

Any compromise to the owner account may allow a hacker to take advantage of this authority and change fees or send and withdraw all the fees and tokens owned by the contract to a desired address.

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., multisignature 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.

BCK-02 | Unused Event

Category	Severity	Location	Status
Coding Style	● Informational	Bolix.sol: 569~570, 574~575, 575~576	ⓘ Pending

Description

The following events are declared but never used.

- `UpdateUniswapV2Router`
- `AutoNukeLP`
- `ManualNukeLP`

Recommendation

We recommend removing the unused event or emitting it in the right place.

BCK-03 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	Bolix.sol: 627~628	ⓘ Pending

Description

All of the Bolix tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute Bolix 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.

BCK-04 | Centralization Risk Related To `addLiquidityETH`

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	Bolix.sol: 863	ⓘ Pending

Description

In `addLiquidityETH` function, `uniswapV2Router.addLiquidityETH{value: ethAmount}` will be called with the `to` argument set to the address `owner()` for acquiring the generated LP tokens from the corresponding pool. As a result, over time the wallet with address `owner()` address will accumulate a significant portion of LP tokens. Mishandling its private key can have devastating consequences to the project as a whole.

```
858         uniswapV2Router.addLiquidityETH{value: ethAmount}(
859             address(this),
860             tokenAmount,
861             0, // slippage is unavoidable
862             0, // slippage is unavoidable
863             owner(),
864             block.timestamp
865         );
866     }
```

Recommendation

We advise the `to` address of the `uniswapV2Router.addLiquidityETH` function call to be replaced by the contract itself, i.e. `address(this)`, and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the `_owner` account is compromised. In general, we strongly 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 a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

BCK-05 | Potential Sandwich Attacks

Category	Severity	Location	Status
Logical Issue	● Minor	Bolix.sol: 846~847, 861~863	ⓘ Pending

Description

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one of the assets and make profits by backrunning (after the transaction being attacked) a transaction to sell the asset.

The following functions are called without setting restrictions on slippage or minimum output amount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

- `_swapTokensForEth`
- `_addLiquidity`

Recommendation

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the aforementioned functions.

BCK-06 | Usage Of `transfer()` For Sending Ether

Category	Severity	Location	Status
Volatile Code	● Minor	Bolix.sol: 899~900	ⓘ Pending

Description

After EIP-1884 was included in the Istanbul hard fork, it is not recommended to use `.transfer()` or `.send()` for transferring ether as these functions have a hard-coded value for gas costs making them obsolete as they are forwarding a fixed amount of gas, specifically `2300`. This can cause issues in case the linked statements are meant to be able to transfer funds to other contracts instead of EOAs.

Recommendation

We advise that the linked `.transfer()` and `.send()` calls are substituted with the utilization of the `sendValue()` function from the `Address.sol` implementation of OpenZeppelin either by directly importing the library or copying the linked code.

BCK-07 | Potential Denial-of-Service Attack

Category	Severity	Location	Status
Logical Issue	● Minor	Bolix.sol: 750	ⓘ Pending

Description

The internal function `_transfer()` sets the `_isSniper` flag of the recipient of a token transfer to `true` if the transaction timestamp is the same as `_launchTime`.

```
750         if (block.timestamp == _launchTime) _isSniper[to] = true;
```

As addresses whose `_isSniper` flag is `true` are unable to use functions that call `_transfer()`, this leads to a possible denial-of-service attack where an attacker obtains tokens before `_launchTime`, such as from the owner, and transfers a token to an address at `_launchTime`. This allows the possibility for important addresses, such as the associated UniswapV2 pair/router to be flagged.

This issue can be fixed through the function `delSnipers()`, but there may be a service disruption shortly after `_launchTime`.

Recommendation

We recommend only allowing transfers after `_launchTime` if the sender and recipient are not the owner, or being very careful about how the initial tokens are distributed.

BCK-08 | Unhandled Return Value

Category	Severity	Location	Status
Logical Issue	● Informational	Bolix.sol: 858~859	ⓘ Pending

Description

In `_addLiquidity()` the return value from `uniswapV2Router.addLiquidityETH()` is not properly handled. If the return values are ignored, this could create unexpected exceptions. This could especially happen in circumstances where the called functions are not reverted automatically.

Recommendation

We recommend checking the return value of the function linked and handling both sided of success and failure cases based on logic needed.

BCK-09 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	Bolix.sol: 630~631, 642~643, 648~649, 654~655, 661~662, 666~667, 671~672, 675~676, 683~684, 691~692, 717~718, 725~726	ⓘ Pending

Description

The function that affects the status of sensitive variables should be able to emit events as notifications.

- `enableTrading`: Set `tradingActive` to true
- `removeLimits`: Set `limitsInEffect` to true
- `disableTransferDelay`: Set `transferDelayEnabled` to false
- `updateSwapTokensAtAmount`: Change the minimum amount of tokens to sell from fees
- `updateMaxTxnAmount`: Set maximum transfer amount
- `updateMaxWalletAmount`: Set new max wallet size amounts
- `excludeFromMaxTransaction`: Exclude an address from the max transaction restriction
- `updateBuyFees`: Update buy fees
- `updateSellFees`: Update sell fees
- `excludeFromFees`: Exclude an address from buy/sell fees
- `setSnipers`: Blacklist an address
- `delSnipers`: Remove an address from blacklist

Recommendation

Consider adding events for sensitive actions, and emit them in the function.

BCK-10 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	● Informational	Bolix.sol: 696~697, 713~714, 717~718, 725~726	ⓘ Pending

Description

The following functions are declared as `public`, contain array function arguments, and are not invoked in any of the contracts contained within the project's scope. The functions that are never called internally within the contract should have external visibility.

- `setAutomatedMarketMakerPair`
- `isExcludedFromFees`
- `setSnipers`
- `delSnipers`

Recommendation

We advise that the functions' visibility specifiers are set to `external` and the array-based arguments change their data location from `memory` to `calldata`, optimizing the gas cost of the function.

BCK-11 | Redundant SafeMath Usage

Category	Severity	Location	Status
Language Specific	● Informational	Bolix.sol: 243	ⓘ Pending

Description

Solidity version >=0.8.0 includes checked arithmetic operations and underflow/overflow by default, making SafeMath redundant.

Recommendation

We recommend removing the SafeMath library and use standard arithmetic operators to reduce code complexity.

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.

Volatile Code

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

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

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.

CERTIK