CERTIK

Security Assessment

TheKillBox

Dec 2nd, 2021

Table of Contents

Summary

This report has been prepared for TheKillBox to discover issues and vulnerabilities in the source code of the TheKillBox 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	TheKillBox
Platform	bsc
Language	Solidity
Codebase	https://bscscan.com/address/0x3523d58d8036B1C5C9A13493143c97aEfC5Ad422
Commit	

Audit Summary

Delivery Date	Dec 02, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	ERC20

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	2	0	0	2	0	0
Medium	0	0	0	0	0	0
Minor	2	1	0	1	0	0
Informational	5	0	0	5	0	0
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
KBX	KBOXToken.sol	c8309b8536a35307481a50d2733a2d5837a26dda56480e0776db431b28512d4a

Overview

TheKillBox created a standard ERC20 token with the `minter` role that can mint `KBOXToken` tokens before the circulating tokens number surpasses the total supply.

The auditors have confirmed that the contract `KBOXToken` has been deployed at address 0x3523d58d8036B1C5C9A13493143c97aEfC5Ad422.

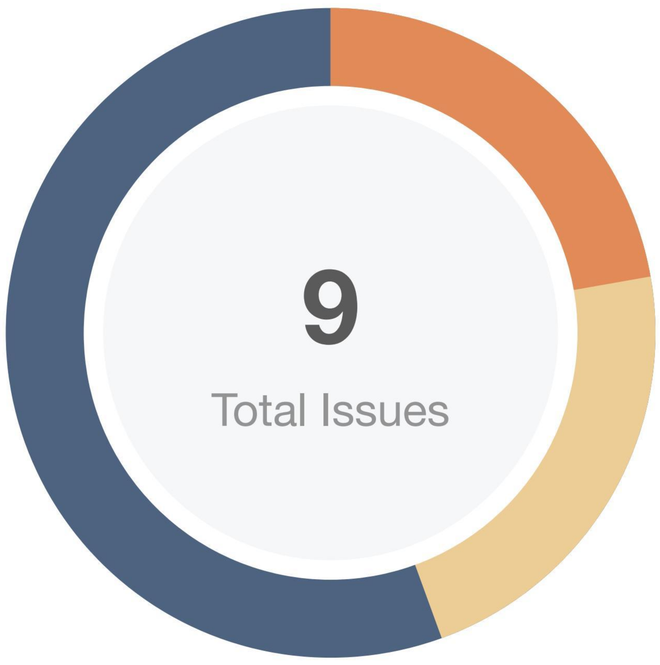
Privileged Roles

To set up the project correctly, improve overall project quality and preserve upgradability, the following roles are adopted in the codebase:

- The `owner` role is adopted in contract `Ownable` to transfer/renounce the ownership.
- The `owner` role is adopted in contract `KBOXToken` to add/delete/get minters.
- The `minter` role is adopted in contract `KBOXToken` to mint `KBOXToken` tokens.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Any plan to invoke the aforementioned functions should be also considered to move to the execution queue of the `Timelock` contract.

Findings



Critical	**0** (0.00%)
Major	**2** (22.22%)
Medium	**0** (0.00%)
Minor	**2** (22.22%)
Informational	**5** (55.56%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
KBX-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
KBX-02	Initial Token Distribution	**Centralization / Privilege**	**Major**	Acknowledged
KBX-03	Potential Integer Underflow	Mathematical Operations	Minor	Acknowledged
KBX-04	Unlocked Compiler Version	Language Specific	Informational	Acknowledged
KBX-05	Multiple `pragma` Being Declared	Language Specific	Informational	Acknowledged
KBX-06	Redundant Internal Function	Coding Style, Gas Optimization	Informational	Acknowledged
KBX-07	Function Can be Declared as `external`	Gas Optimization	Informational	Acknowledged
KBX-08	Lack of Event Emission for Significant Transactions	Coding Style	Informational	Acknowledged
KBX-09	Burning Tokens Not Affecting TotalSupply	Logical Issue	Minor	Pending

KBX-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/KBOXToken/KBOXToken.sol (015d0d3): 692, 701, 1019, 1027, 1032, 1045	Acknowledged

Description

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

- `Ownable.renounceOwnership()`: Renounce the ownership of the contract and set the `owner` address to the zero address.
- `Ownable.transferOwnership()`: Transfer the ownership of the contract to an arbitrary non-zero address.

In contract `KBOXToken`, the `minter` role has the authority over the following function:

- `KBOXToken.mint()`: Mint any amount of `KBOXToken` tokens before reaching the `totalSupply` to an arbitrary address.

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

- `KBOXToken.addMinter()`: Grant the `minter` role to an arbitrary non-zero address.
- `KBOXToken.delMinter()`: Retract an address's `minter` role.
- `KBOXToken.getMinter()`: Get the list of addresses with the `minter` role.

Any compromise to any account with any privileged role may allow the hacker to take advantage of this and disrupt operations involving this token.

Recommendation

We advise the client to carefully manage the `owner` and `minter` accounts' private keys carefully to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

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

- Time-lock with reasonable latency, e.g., 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.

Alleviation

[**CertiK**]: The auditors have confirmed that the contract `KBOXToken` has been deployed at address 0x3523d58d8036B1C5C9A13493143c97aEfC5Ad422.

According to the information available on bscscan at UTC 7:00 PM, November 23rd, 2021, the owner of contract `KBOXToken` is 0x8F6e763B47257d7D2B277ea1562dd77C34C4e9e1, which is an externally owned account. There is no account being granted with `minter` role at UTC 7:00 PM, November 23rd, 2021.

KBX-02 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/KBOXToken/KBOXToken.sol (015d0d3): 1015	Acknowledged

Description

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

Recommendation

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

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

KBX-03 | Potential Integer Underflow

Category	Severity	Location	Status
Mathematical Operations	Minor	projects/KBOXToken/KBOXToken.sol (015d0d3): 1046	Acknowledged

Description

In the function `KBOXToken.getMinter()`, the input `_index` is checked to ensure the index is valid. However, it is possible that `getMinterLength() == 0` which might lead to integer underflow in the calculation:

```
1046          require(_index <= getMinterLength() - 1, "KBOXToken: index out of
bounds");
```

Recommendation

We advise the client to use the SafeMath library for all of the mathematical operations.

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

KBX-04 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	Informational	projects/KBOXToken/KBOXToken.sol (015d0d3): 6	Acknowledged

Description

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.6.2` the contract should contain the following line:

```
pragma solidity 0.6.2;
```

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

KBX-05 | Multiple `pragma` Being Declared

Category	Severity	Location	Status
Language Specific	● Informational	projects/KBOXToken/KBOXToken.sol (015d0d3): 6, 29, 109, 327, 643, 711	ⓘ Acknowledged

Description

There are multiple `pragma` declarations on the aforementioned lines.

Recommendation

We recommend removing the redundant `pragma` declarations.

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

KBX-06 | Redundant Internal Function

Category	Severity	Location	Status
Coding Style, Gas Optimization	Informational	projects/KBOXToken/KBOXToken.sol (015d0d3): 619	Acknowledged

Description

The function `ERC20._setupDecimals()` has the `internal` visibility. However, it is not called by any function in the contract `ERC20`.

Recommendation

We recommend removing the redundant function `ERC20._setupDecimals()`.

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

KBX-07 | Function Can be Declared as `external`

Category	Severity	Location	Status
Gas Optimization	Informational	projects/KBOXToken/KBOXToken.sol (015d0d3): 387, 395, 412, 426, 438, 446, 457, 475, 493, 512, 566, 692, 701, 1019, 1027, 1032, 1045	Acknowledged

Description

The following functions are never called by the contract, so they can be declared `external` to save gas:

- `ERC20.name()`
- `ERC20.symbol()`
- `ERC20.decimals()`
- `ERC20.balanceOf()`
- `ERC20.transfer()`
- `ERC20.allowance()`
- `ERC20.approve()`
- `ERC20.transferFrom()`
- `ERC20.increaseAllowance()`
- `ERC20.decreaseAllowance()`
- `ERC20.burn(uint256)`
- `Ownable.renounceOwnership()`
- `Ownable.transferOwnership()`
- `KBOXToken.mint()`
- `KBOXToken.addMinter()`
- `KBOXToken.delMinter()`
- `KBOXToken.getMinter()`

Recommendation

We recommend changing the visibilities of the functions on the aforementioned lines to `external`.

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

KBX-08 | Lack of Event Emission for Significant Transactions

Category	Severity	Location	Status
Coding Style	Informational	projects/KBOXToken/KBOXToken.sol (015d0d3): 1027, 1032	Acknowledged

Description

The following functions that affect the status of sensitive variables should emit events for better tracking contract status:

- `KBOXToken.addMinter()`: Grant the `minter` role to an arbitrary non-zero address.
- `KBOXToken.delMinter()`: Retract an address's `minter` role.

Recommendation

We recommend adding events for the aforementioned sensitive actions, and emitting the events within the functions.

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

KBX-09 | Burning Tokens Not Affecting TotalSupply

Category	Severity	Location	Status
Logical Issue	Minor	projects/KBOXToken/KBOXToken.sol (015d0d3): 587	Pending

Description

While buring `KBOXToken`s, the `_totalSupply` of `KBOXToken` will not decrease. We want to check with the team whether this is an intended design.

Alleviation

[TheKillBox]: The team acknowledged the issue and decided not to make any changes to the current version.

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.

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

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.

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.