Security Assessment

eBSO

Sept 15th, 2021

Table of Contents

Summary

This report has been prepared for Blockben Financial Services OU to discover issues and vulnerabilities in the source code of the eBSO 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.

Additionally, this audit is based on a premise that all external contracts were implemented safely.

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	eBSO
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/blockben-official/ebso/tree/main/src/contracts
Commit	1a3b90807274e78d918d8e2b2bb37cb398b60ed6 66f109faf7dfdf6055760be65c4d1d7827792d89

Audit Summary

Delivery Date	Sept 15, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

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

Audit Scope

ID	File	SHA256 Checksum
EBS	EBlockStock.sol	df6dcb9193f171dea8a3a5dfa279f41e699a001ef9f8ec81c2fce64aae2c906f
EBA	EBlockStockACL.sol	6c26044dd24c2f1e5857db8a1a7d775d9ae22801fe1fd69c3729b6865ca055b6

Understandings

Overview

The EBlockStock contract is an ERC20 deflation token contract. It contains three important roles: `TOKEN_ADMIN`, `TREASURY_ADMIN`,`AML_ADMIN`.

The `TOKEN_ADMIN` role can pause, unpause, destroy the contract, and set all important parameters of the contract.
The `AML_ADMIN` role can set/cancel the source blacklist and destination blacklist.
The `TREASURY_ADMIN` role can mint tokens to any account and burn tokens from `treasuryAddress` account.
Two parts of fees are charged for transfer, `generalFee` and `bsoFee`. When they are not 0, the amount received by the receiver will be less than the amount sent by the sender.

Privileged Functions

The contract contains the following privileged functions that are restricted by some modifiers and roles. They are used to modify the contract configurations and address attributes. We grouped these functions below:

The `TOKEN_ADMIN` role:

Contract `EBlockStockACL`:

- setUrl(string calldata _newUrl)
- setTreasuryAddress(address _newAddress)
- setFeeAddress(address _newAddress)
- setBsoPoolAddress(address _newAddress)
- setGeneralFee(uint16 _newFee)
- setBsoFee(uint16 _newFee)
- pause()
- unpause()

The `TREASURY_ADMIN` role:

Contract `EBlockStock`:

- mint(address _account, uint256 _amount)
- burn(uint256 _amount)

The `AML_ADMIN` role:

Contract `EBlockStockACL`:

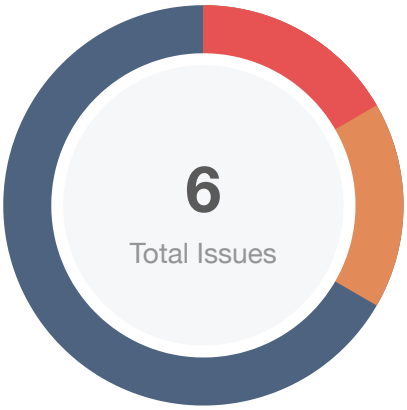
- setSourceAccountBL(address _account, bool _lockValue)
- setDestinationAccountBL(address _account, bool _lockValue)
- setBatchSourceAccountBL(address[] calldata _addresses, bool _lockValue)
- setBatchDestinationAccountBL(address[] calldata _addresses, bool _lockValue)

The `whenNotPaused` modifier:

Contract `EBlockStock`:

- transfer(address _to, uint256 _value)
- transferFrom(address _from, address _to, uint256 _value)
- approve(address _spender, uint256 _value)
- increaseAllowance(address _spender, uint256 _addedValue)
- decreaseAllowance(address _spender, uint256 _subtractValue)
- mint(address _account, uint256 _amount)
- burn(uint256 _amount)

Findings



Critical	1 (16.67%)
Major	1 (16.67%)
Medium	0 (0.00%)
Minor	0 (0.00%)
Informational	4 (66.67%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Centralization Risk	Centralization / Privilege	Major	Acknowledged
EBA-01	Unlocked Compiler Version Declaration	Language Specific	Informational	Resolved
EBA-02	Missing Input Validation	Logical Issue	Informational	Resolved
EBS-01	Unlocked Compiler Version Declaration	Language Specific	Informational	Resolved
EBS-02	Boolean equality	Coding Style	Informational	Resolved
EBS-03	`EBSO_ADMIN` Role Can Destroy The Contract	Logical Issue	Critical	Resolved

GLOBAL-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	Global	Acknowledged

Description

In the contract `EBlockStockACL`, the role `AML_ADMIN` has the authority over the following function:

1. set/cancel any account to the source blacklist through `setSourceAccountBL` function.
2. set/cancel any account to the destination blacklist through `setDestinationAccountBL` function.
3. set/cancel accounts to the source blacklist in batch through `setBatchSourceAccountBL()` function.
4. set/cancel accounts to the destination blacklist in batch through `setBatchDestinationAccountBL()` function.

In the contract `EBlockStockACL`, the role `TOKEN_ADMIN` has the authority over the following function:

1. modify `url` through `setUrl` function.
2. modify treasury address through `setTreasuryAddress` function.
3. modify fee address through `setFeeAddress` function.
4. modify `BSO` token pool address through `setBsoPoolAddress` function.
5. modify general fee rate through `setGeneralFee` function.
6. modify `bsoFee` rate through `setBsoFee` function.
7. pause the contract through `pause` function.
8. unpause the contract through `unpause` function.

In the contract `EBlockStock`, the role `TREASURY_ADMIN` has the authority over the following function:

1. mint token to any account through `mint` function.
2. burn token from treasury account through `burn` function.

In the contract `EBlockStock`, the role `TOKEN_ADMIN` has the authority over the following function:

1. destroy the contract and transfer all balance of the contract to account `_toCashOut` through `kill` function.

without obtaining the consensus of the community.

Recommendation

We advise the client to carefully manage the `TOKEN_ADMIN`, `AML_ADMIN`, `TREASURY_ADMIN` role accounts' private key 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 are some feasible suggestions that would also mitigate the potential risk at the different levels in terms 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

Client response:

- multi-sig wallets are not in scope now.
- we have stored the addresses properly at the moment time-lock is not necessary, as such changes are subject to GTC changes that we always announce in advance.
- BlockBen Financial Services OÜ, as a financial institution, uses all the privileged rights according to the GTC, taking into account the legislation in force. It is also guaranteed that the admin accounts are owned by BlockBen Financial Services OÜ so it uses its ownership in governance and make the privileged operations according to following (like admin add, remove as, burn etc).

BlockBen guarantees that all ethereum accounts private keys with admin privileges are stored with due care to avoid the possibility of hacking. (It can be taken into account that access to the super admin ethereum account is stored in a vault and printed on a paper) In the case of privileged operations, BlockBen acts in accordance with the GTC and the legislation in force in Estonia and audited by a regulatory body. Treasury account required for EBSO issuance and withdrawal, and blacklist due to AML requirements from the regulatory body.

You can find out more about how the token works at https://blockben.com/products/ebso.

EBA-01 | Unlocked Compiler Version Declaration

Category	Severity	Location	Status
Language Specific	Informational	EBlockStockACL.sol: 2	Resolved

Description

The compiler version utilized throughout the project uses the `^` prefix specifier, denoting that a compiler version that is greater than the version will be used to compile the contracts. It is recommended the compiler version be consistent throughout the codebase.

Recommendation

It is a general practice to instead lock the compiler at a specific version rather than allow a range of compiler versions to be utilized to avoid compiler-specific bugs and thus be able to identify ones more easily. We recommend locking the compiler at the lowest possible version that supports all the capabilities wished by the codebase. This will ensure that the project utilizes a compiler version that has been in use for the longest time and as such is less likely to contain yet-undiscovered bugs.

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 66f109faf7dfdf6055760be65c4d1d7827792d89.

EBA-02 | Missing Input Validation

Category	Severity	Location	Status
Logical Issue	Informational	EBlockStockACL.sol: 92, 97, 102, 46	Resolved

Description

The given input is missing the sanity check for non-zero address in the aforementioned line.

Recommendation

We recommend adding the check for the passed-in values to prevent unexpected error as below:
constructor：

```
46  require(_superadmin != address(0), "_superadmin cannot be 0");
```

```
92  require(_newAddress != address(0), "_newAddress cannot be 0");
```

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 66f109faf7dfdf6055760be65c4d1d7827792d89.

EBS-01 | Unlocked Compiler Version Declaration

Category	Severity	Location	Status
Language Specific	Informational	EBlockStock.sol: 2	Resolved

Description

The compiler version utilized throughout the project uses the `^` prefix specifier, denoting that a compiler version that is greater than the version will be used to compile the contracts. It is recommended the compiler version be consistent throughout the codebase.

Recommendation

It is a general practice to instead lock the compiler at a specific version rather than allow a range of compiler versions to be utilized to avoid compiler-specific bugs and thus be able to identify ones more easily. We recommend locking the compiler at the lowest possible version that supports all the capabilities wished by the codebase. This will ensure that the project utilizes a compiler version that has been in use for the longest time and as such is less likely to contain yet-undiscovered bugs.

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 66f109faf7dfdf6055760be65c4d1d7827792d89.

EBS-02 | Boolean equality

Category	Severity	Location	Status
Coding Style	Informational	EBlockStock.sol: 86, 102, 121, 122	Resolved

Description

Boolean constants can be used directly and do not need to be compared to true or false.

Recommendation

We recommend changing it as following:

burn():

```
86    require(!getSourceAccountBL(treasuryAddress), 'Blacklist: treasury');
```

_mint():

```
102    require(!getDestinationAccountBL(_account), 'Blacklist: target');
```

_transfer():

```
121    require(!getSourceAccountBL(_sender), 'Blacklist: sender');
122    require(!getDestinationAccountBL(_recipient), 'Blacklist: recipient');
```

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 66f109faf7dfdf6055760be65c4d1d7827792d89.

EBS-03 | `EBSO_ADMIN` Role Can Destroy The Contract

Category	Severity	Location	Status
Logical Issue	Critical	EBlockStock.sol: 93	Resolved

Description

The `EBlockStock` contract is a token contract. If the `TOKEN_ADMIN` role destroys the contract, EBSO tokens will not be able to trade and circulate, it will cause huge losses to users who hold tokens.

Recommendation

We recommend removing the `kill` function.

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 66f109faf7dfdf6055760be65c4d1d7827792d89.

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.

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.