SecureWise

AUDIT REPORT

SecureWise

ALL BLUE

https://securewise.info/

https://t.me/securewisehub

https://github.com/securewise

Table of Contents

Disclaimer

SecureWise provides the smart contract audit of solidity. Audit and report are for informational purposes only and not, nor should be considered, as an endorsement to engage with, invest in, participate, provide an incentive, or disapprove, criticise, discourage, or purport to provide an opinion on any particular project or team.

This audit report doesn't provide any warranty or guarantee regarding the nature of the technology analysed. These reports, in no way, provide investment advice, nor should be used as investment advice of any sort. Investors must always do their own research and manage their risk.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and SecureWise and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) SecureWise owe no duty of care towards you or any other person, nor does SecureWise make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SecureWise hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SecureWise hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SecureWise, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

Overview

Token Name: All Blue **(ALLBLUE)**

Methodology: Automated Analysis, Manual Code Review

Language: Solidity

Contract Address: 0xA02cDBe4323C52921815F6Bb0028a74ebE7889DA

ContractLink: https://bscscan.com/address/0xA02cDBe4323C52921815F6Bb0028a74ebE7889DA

Network: BSC

Supply: 5500000000

Website: https://allblue.games/

Twitter: https://twitter.com/AllBlue_metvrse

Telegram: https://t.me/AllBlue_Group

Report Date: April 10, 2023

Quick Result

SecureWise has applied the automated and manual analysis of Smart Contract and were reviewed for common contract vulnerabilities and centralized exploits

⚠	The owner can mint tokens after the initial deployment without limit
⚠	The owner can burn tokens without limit

Page 10 for more details

All Blue (ALLBLUE) has succesfully **PASSED** the smart contract audit with **HIGH** **severity issue**

Auditing Approach and Methodologies

SecureWise has performed starting with analyzing the code, issues, code quality, and libraries. Reviewed line-by-line by our team. Finding any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

Methodology

- Understanding the size, scope and functionality of your project's source code
- Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Testing and automated analysis of the Smart Contract to determine proper logic has been followed throughout the whole process
- Deploying the code on testnet using multiple live test
- Analyzing a program to determine the specific input that causes different parts of a program to execute its functions.
- Checking whether all the libraries used in the code are on the latest version.

Goals

Smart Contract System is secure, resilient and working according to the specifications and without any vulnerabilities.

Risk Classification

High: Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, of the contract and its functions. Must be fixed as soon as possible.

Medium: Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Must be fxed as soon as possible.

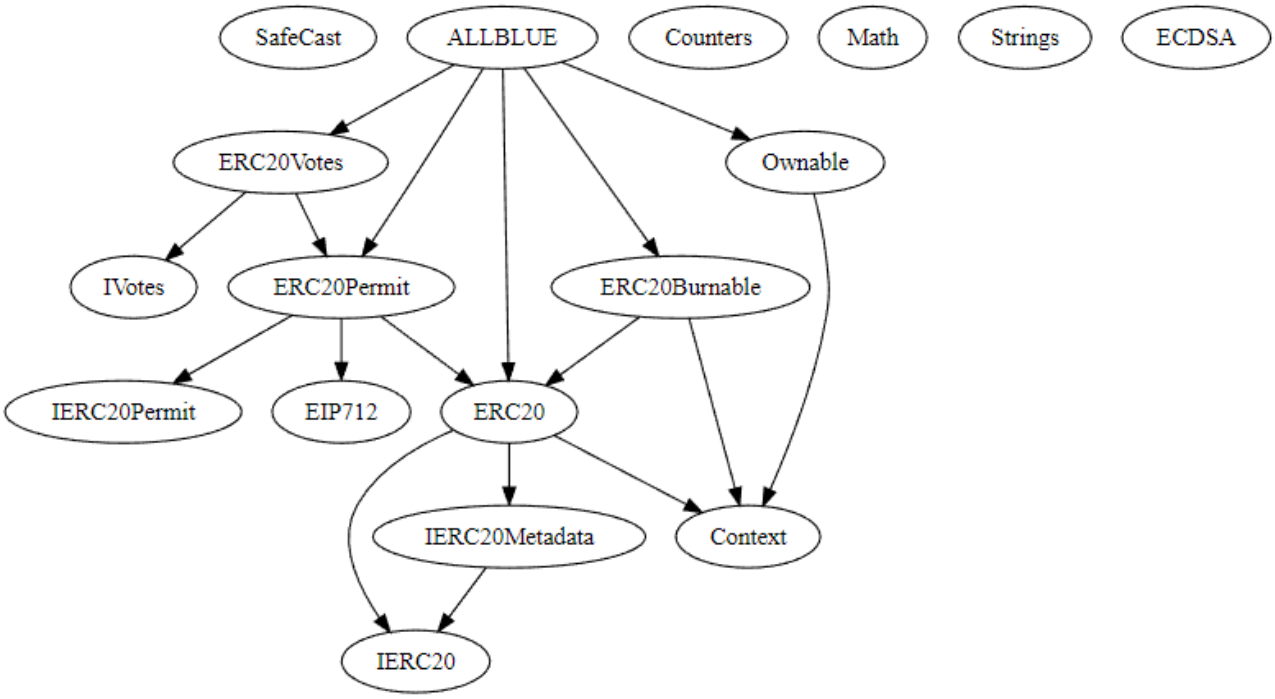
Low: Effects are minimal in isolation and do not pose a signifcant danger to the project or its users. Issues under this classifcation are recommended to be fixed nonetheless.

Automated Analysis

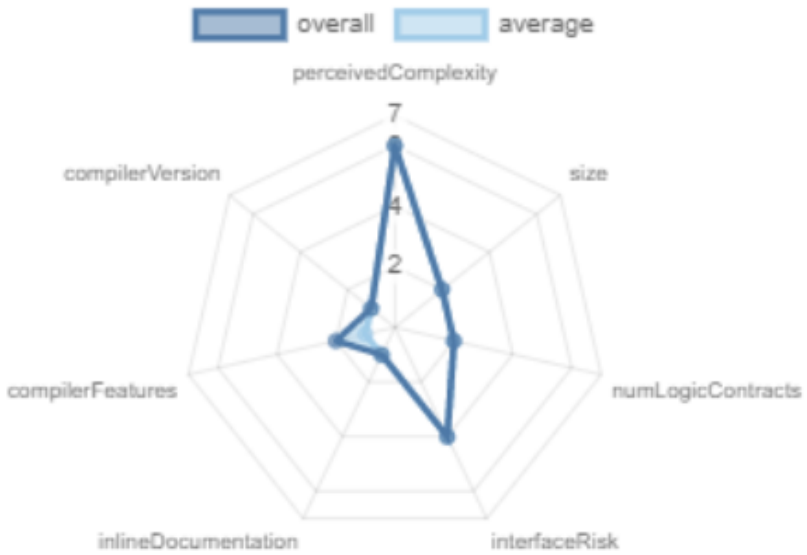
Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

ERC20Burnable	Implementation	Context, ERC20		
L	burn	Public ❗	🔴	NO ❗
L	burnFrom	Public ❗	🔴	NO ❗
ALLBLUE	Implementation	ERC20, ERC20Burnable, Ownable, ERC20Permit, ERC20Votes		
L		Public ❗	🔴	ERC20 ERC20Permit
L	mint	Public ❗	🔴	onlyOwner
L	_afterTokenTransfer	Internal 🔒	🔴	
L	_mint	Internal 🔒	🔴	
L	_burn	Internal 🔒	🔴	

Inheritance Graph



Risk



Source Lines

Contract Summary

Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
13	4	3018	2767	995	1603	743	
13	**4**	**3018**	**2767**	**995**	**1603**	**743**	

Components

Contracts	Libraries	Interfaces	Abstract
2	5	4	6

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

Public	Payable
46	0

External	Internal	Private	Pure	View
19	162	8	95	36

StateVariables

Total	Public
21	0

Capabilities

Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
^0.8.0 ^0.8.9		—	yes (7 asm blocks)	—

Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRecover	New/Create/Create2
—	—	—	yes	yes	—

TryCatch	Σ Unchecked
—	yes

Manual Review

The owner can mint tokens after the initial deployment without limit

```
ftrace | funcSig
function mint(address to, uint256 amount) public onlyOwner {
    _mint(to, amount);
}
```

During the audit, it was identified that the smart contract allows the owner to mint tokens even after the initial deployment without any limit. This means that the owner has the ability to create an unlimited number of tokens, which could potentially lead to inflation and devaluation of the token's value.

Recommendation

Based on the audit findings, we recommend that the contract owner implements a limit on token minting to mitigate the risks of potential inflation and fraudulent activities. or can'table to mint after the initial deployment. By implementing these measures, the contract owner can help ensure that the token issuance process is fair, transparent, and in line with community expectations. It is recommended that these measures are implemented as soon as possible to mitigate the risks associated with unlimited token minting.

The owner can burn tokens without limit

```
ftrace | funcSig
function _burn(address account, uint256 amount)
    internal
    override(ERC20, ERC20Votes)
{
    super._burn(account, amount);
}
}
```

```
function burn(uint256 amount) public virtual {
    _burn(_msgSender(), amount);
}
```

During the audit, it was identified that the smart contract allows the owner to burn tokens without any limit. This means that the owner has the ability to destroy an unlimited number of tokens, which could potentially lead to a shortage of tokens and an increase in their value.

Recommendation

Based on the audit findings, we recommend that the contract owner implements a limit on token burning to mitigate the risks of potential fraudulent activities while still allowing for necessary token burning operations. By implementing these measures, the contract owner can help ensure that the token burning process is fair, transparent, and in line with community expectations. It is recommended that these measures are implemented as soon as possible to mitigate the risks associated with unlimited token burning.

SecureWise

AUDIT REPORT

SecureWise

https://securewise.info/ https://t.me/securewisehub https://github.com/securewise