

# Boost Smart Contract Audit Report



10 Aug 2021



# **TABLE OF CONTENTS**

#### Audited Details

- Audited Project
- Blockchain
- Addresses
- Project Website
- Codebase

#### Summary

- Contract Summary
- Audit Findings Summary
- Vulnerabilities Summary

### Conclusion

#### Audit Results

#### Smart Contract Analysis

- Detected Vulnerabilities

#### **Disclaimer**

### About Us



# AUDITED DETAILS

### Audited Project

| Project name | Token ticker | Blockchain |  |
|--------------|--------------|------------|--|
| Boost        | BOOST        | Ethereum   |  |

### Addresses

| Contract address          | 0x4e0fca55a6c3a94720ded91153a27f60e26b9aa8 |  |
|---------------------------|--|--|
| Contract deployer address | 0xEE72635914C510637c77B85232572d10f72b5A0E |  |

### Project Website

#### https://boostco.in/

### Codebase

https://etherscan.io/address/0x4e0fca55a6c3a94720ded91153a27f60e26b9aa8#code



# SUMMARY

"Boost is a new Digital Ecosystem for the modern world. Its mission is to disrupt traditional forms of financial tools which provide liquidity to the broader economy. We believe that the power of finance should be in the hands of the many, not the few, and that technology should be optimized in the best interest of human beings, to work for us and not the other way around. Today, large corporations earn billions of dollars each year by leveraging the value of our collective earnings, which they risk by investing in ventures large and small, for the benefit of the few, not the many. "

### Contract Summary

#### **Documentation Quality**

Boost provides a very good documentation with standard of solidity base code.

• The technical description is provided clearly and structured and also dont have any high risk issue.

#### **Code Quality**

The Overall quality of the basecode is standard.

• Standard solidity basecode and rules are already followed by Boost with the discovery of several low issues.

#### **Test Coverage**

Test coverage of the project is 100% (Through Codebase)

### Audit Findings Summary

- SWC-100 SWC-108 | Explicitly define visibility for all state variables on lines 666.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 102, 134, 157, 158, 193, 229, 645, 645, 646, 646, 669, 669, 670, 670, 671, 671, 806, 808, 828, 830, 878, 1059, 808 and 830.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 6.
- SWC-110 SWC-123 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 807, 808, 808, 829, 830, 830, 917, 918, 1060, 1060, 1061 and 1062.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 872.



# CONCLUSION

We have audited the Boost project released on August 2021 to discover issues and identify potential security vulnerabilities in Boost Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

The security audit report provides a satisfactory result with some low-risk issues.

The issues found in the Boost smart contract code do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a floating pragma is set, a state variable visibility is not set, tx.origin as a part of authorization control and out of bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value.



# AUDIT RESULT

| Article                              | Category           | Description   | Result         |  |
|--------------------------------------|--------------------|---|----------------|--|
| Default Visibility                   | SWC-100<br>SWC-108 | Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously. |                |  |
| Integer Overflow<br>and Underflow    | SWC-101            | If unchecked math is used, all math operations should be safe from overflows and underflows.                          | ISSUE<br>FOUND |  |
| Outdated Compiler<br>Version         | SWC-102            | It is recommended to use a recent version of the Solidity compiler.   | PASS           |  |
| Floating Pragma                      | SWC-103            | Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.          | ISSUE<br>FOUND |  |
| Unchecked Call<br>Return Value       | SWC-104            | The return value of a message call should be checked.   | PASS           |  |
| Unprotected Ether<br>Withdrawal      | SWC-105            | Due to missing or insufficient access controls, malicious parties can withdraw from the contract.                     | PASS           |  |
| SELFDESTRUCT<br>Instruction          | SWC-106            | The contract should not be self-destructible while it has funds belonging to users.                                   | PASS           |  |
| Reentrancy                           | SWC-107            | Check effect interaction pattern should be followed if the code performs recursive call.                              | PASS           |  |
| Uninitialized<br>Storage Pointer     | SWC-109            | Uninitialized local storage variables can point to unexpected storage locations in the contract.                      | PASS           |  |
| Assert Violation                     | SWC-110<br>SWC-123 | Properly functioning code should never reach aISSLfailing assert statement.FOUR                                       |                |  |
| Deprecated Solidity<br>Functions     | SWC-111            | Deprecated built-in functions should never be used.   | PASS           |  |
| Delegate call to<br>Untrusted Callee | SWC-112            | Delegatecalls should only be allowed to trusted addresses.  | PASS           |  |



| DoS (Denial of<br>Service)             | SWC-113<br>SWC-128            | Execution of the code should never be blocked by a specific contract state unless required.  | PASS           |
|--|-------------------------------|--|----------------|
| Race Conditions                        | SWC-114                       | Race Conditions and Transactions Order Dependency should not be possible.  | PASS           |
| Authorization through tx.origin        | SWC-115                       | tx.origin should not be used for authorization.  | ISSUE<br>FOUND |
| Block values as a proxy for time       | SWC-116                       | Block numbers should not be used for time calculations.  | PASS           |
| Signature Unique<br>ID                 | SWC-117<br>SWC-121<br>SWC-122 | Signed messages should always have a unique id. A transaction hash should not be used as a unique id.  | PASS           |
| Incorrect<br>Constructor Name          | SWC-118                       | SWC-118 Constructors are special functions that are called only once during the contract creation.   |                |
| Shadowing State<br>Variable            | SWC-119                       | State variables should not be shadowed.  | PASS           |
| Weak Sources of<br>Randomness          | SWC-120                       | Random values should never be generated from Chain<br>Attributes or be predictable.  | PASS           |
| Write to Arbitrary<br>Storage Location | SWC-124                       | The contract is responsible for ensuring that only<br>authorized user or contract accounts may write to<br>sensitive storage locations.        |                |
| Incorrect<br>Inheritance Order         | SWC-125                       |  | PASS           |
| Insufficient Gas<br>Griefing           | SWC-126                       | SWC-126Insufficient gas griefing attacks can be performed on<br>contracts which accept data and use it in a sub-call on<br>another contract.PA |                |
| Arbitrary Jump<br>Function             | SWC-127                       | As Solidity doesnt support pointer arithmetics, it is impossible to change such variable to an arbitrary value.                                | PASS           |



| Typographical<br>Error        | SWC-129            | A typographical error can occur for example when the intent<br>of a defined operation is to sum a number to a variable.  |  |
|-------------------------------|--------------------|--|--|
| Override control<br>character | SWC-130            | Malicious actors can use the Right-To-Left-Override unicode<br>character to force RTL text rendering and confuse users as<br>to the real intent of a contract. |  |
| Unused variables              | SWC-131<br>SWC-135 |  |  |
| Unexpected Ether<br>balance   | SWC-132            | Contracts can behave erroneously when they strictly assume a specific Ether balance.   |  |
| Hash Collisions<br>Variable   | SWC-133            | Using abi.encodePacked() with multiple variable length arguments can, in certain situations, lead to a hash collision.   |  |
| Hardcoded gas<br>amount       | SWC-134            | The transfer() and send() functions forward a fixed amount of 2300 gas.  |  |
| Unencrypted<br>Private Data   | SWC-136            | It is a common misconception that private type variables cannot be read.   |  |



# **SMART CONTRACT ANALYSIS**

| Started          | Monday Aug 09 2021 03:13:45 GMT+0000 (Coordinated Universal Time)  |  |  |
|------------------|--|--|--|
| Finished         | Tuesday Aug 10 2021 11:53:02 GMT+0000 (Coordinated Universal Time) |  |  |
| Mode             | Standard   |  |  |
| Main Source File | BoostToken.sol   |  |  |

### Detected Issues

| ID      | Title                                | Severity | Status       |
|---------|--------------------------------------|----------|--------------|
| SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED  | low      | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED | low      | acknowledged |

### SYSFIXED

| SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED                    | low | acknowledged |
|---------|--|-----|--------------|
| SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED                   | low | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED                   | low | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED                    | low | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED                   | low | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED                    | low | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED                    | low | acknowledged |
| SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED                   | low | acknowledged |
| SWC-101 | COMPILER-REWRITABLE " <uint> - 1" DISCOVERED</uint>    | low | acknowledged |
| SWC-101 | COMPILER-REWRITABLE " <uint> - 1" DISCOVERED</uint>    | low | acknowledged |
| SWC-103 | A FLOATING PRAGMA IS SET.                              | low | acknowledged |
| SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.                  | low | acknowledged |
| SWC-115 | USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL. | low | acknowledged |



| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
|---------|----------------------------|-----|--------------|--|
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |
| SWC-110 | OUT OF BOUNDS ARRAY ACCESS | low | acknowledged |  |



# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 102

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
101 function add(uint256 a, uint256 b) internal pure returns (uint256) {
102 uint256 c = a + b;
103 require(c >= a, "SafeMath: addition overflow");
104
105 return c;
106
```



# SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

**LINE 134** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
133 require(b <= a, errorMessage);
134 uint256 c = a - b;
135
136 return c;
137 }
138</pre>
```



# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 157

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
156
157 uint256 c = a * b;
158 require(c / a == b, "SafeMath: multiplication overflow");
159
160 return c;
161
```



# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

**LINE 158** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
157 uint256 c = a * b;
158 require(c / a == b, "SafeMath: multiplication overflow");
159
160 return c;
161 }
162
```



# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

**LINE 193** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
192 require(b > 0, errorMessage);
193 uint256 c = a / b;
194 // assert(a == b * c + a % b); // There is no case in which this doesn't hold
195
196 return c;
197
```



# SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

**LINE 229** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
228 require(b != 0, errorMessage);
229 return a % b;
230 }
231 }
232
233
```



### SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

**LINE 645** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
644 uint256 private constant MAX = ~uint256(0);
645 uint256 private constant _tTotal = 1000000000 * 10**18;
646 uint256 private _rTotal = (MAX - (MAX % _tTotal));
647 uint256 private _tFeeTotal;
648
649
```



### SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

**LINE 645** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
644 uint256 private constant MAX = ~uint256(0);
645 uint256 private constant _tTotal = 1000000000 * 10**18;
646 uint256 private _rTotal = (MAX - (MAX % _tTotal));
647 uint256 private _tFeeTotal;
648
649
```



### SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 646

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
645 uint256 private constant _tTotal = 1000000000 * 10**18;
646 uint256 private _rTotal = (MAX - (MAX % _tTotal));
647 uint256 private _tFeeTotal;
648
649 string private constant _name = 'Boost';
650
```



### SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 646

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
645 uint256 private constant _tTotal = 1000000000 * 10**18;
646 uint256 private _rTotal = (MAX - (MAX % _tTotal));
647 uint256 private _tFeeTotal;
648
649 string private constant _name = 'Boost';
650
```



### SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

**LINE 669** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
668
669 uint256 private _maxTxAmount = 5000000 * 10**18;
670 uint256 private constant _numOfTokensToExchangeForTeam = 5000 * 10**18;
671 uint256 private _maxWalletSize = 9500000 * 10**18;
672
673
```



### SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

**LINE 669** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
668
669 uint256 private _maxTxAmount = 5000000 * 10**18;
670 uint256 private constant _numOfTokensToExchangeForTeam = 5000 * 10**18;
671 uint256 private _maxWalletSize = 9500000 * 10**18;
672
673
```



# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

**LINE 670** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
669 uint256 private _maxTxAmount = 5000000 * 10**18;
670 uint256 private constant _numOfTokensToExchangeForTeam = 5000 * 10**18;
671 uint256 private _maxWalletSize = 9500000 * 10**18;
672
673 event botAddedToBlacklist(address account);
674
```



# SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

**LINE 670** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
669 uint256 private _maxTxAmount = 5000000 * 10**18;
670 uint256 private constant _numOfTokensToExchangeForTeam = 5000 * 10**18;
671 uint256 private _maxWalletSize = 9500000 * 10**18;
672
673 event botAddedToBlacklist(address account);
674
```



# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

**LINE 671** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
670 uint256 private constant _numOfTokensToExchangeForTeam = 5000 * 10**18;
671 uint256 private _maxWalletSize = 9500000 * 10**18;
672
673 event botAddedToBlacklist(address account);
674 event botRemovedFromBlacklist(address account);
675
```



# SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

**LINE 671** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
670 uint256 private constant _numOfTokensToExchangeForTeam = 5000 * 10**18;
671 uint256 private _maxWalletSize = 9500000 * 10**18;
672
673 event botAddedToBlacklist(address account);
674 event botRemovedFromBlacklist(address account);
675
```



# SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

**LINE 806** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
805 require (_isBlackListedBot[account], 'Account is not blacklisted');
806 for (uint256 i = 0; i < _blackListedBots.length; i++) {
807 if (_blackListedBots[i] == account) {
808 _blackListedBots[i] = _blackListedBots[_blackListedBots.length - 1];
809 _isBlackListedBot[account] = false;
810
```



# SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

**LINE 808** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
807 if (_blackListedBots[i] == account) {
808 _blackListedBots[i] = _blackListedBots[_blackListedBots.length - 1];
809 _isBlackListedBot[account] = false;
810 _blackListedBots.pop();
811 break;
812
```



# SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

**LINE 828** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
827 require(_isExcluded[account], "Account is not excluded");
828 for (uint256 i = 0; i < _excluded.length; i++) {
829 if (_excluded[i] == account) {
830 _excluded[i] = _excluded[_excluded.length - 1];
831 _tOwned[account] = 0;
832
```



# SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

**LINE 830** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
829 if (_excluded[i] == account) {
830 _excluded[i] = _excluded[_excluded.length - 1];
831 _tOwned[account] = 0;
832 _isExcluded[account] = false;
833 _excluded.pop();
834
```



# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

**LINE 878** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
877 uint256 tokenBalanceRecipient = balanceOf(recipient);
878 require(tokenBalanceRecipient + amount <= _maxWalletSize, "Recipient exceeds max
wallet size.");
879 }
880 // is the token balance of this contract address over the min number of
881 // tokens that we need to initiate a swap?
882
```



# SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1059

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
1058 uint256 tSupply = _tTotal;
1059 for (uint256 i = 0; i < _excluded.length; i++) {
1060 if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
1061 rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1062 tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1063
```



### SWC-101 | COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED

**LINE 808** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
807 if (_blackListedBots[i] == account) {
808    _blackListedBots[i] = _blackListedBots[_blackListedBots.length - 1];
809    _isBlackListedBot[account] = false;
810    _blackListedBots.pop();
811    break;
812
```



### SWC-101 | COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED

**LINE 830** 

#### **Iow SEVERITY**

This plugin produces issues to support false positive discovery within mythril.

#### Source File

- BoostToken.sol

```
829 if (_excluded[i] == account) {
830 _excluded[i] = _excluded[_excluded.length - 1];
831 _tOwned[account] = 0;
832 _isExcluded[account] = false;
833 _excluded.pop();
834
```



### SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

#### **Iow SEVERITY**

The current pragma Solidity directive is ""^0.6.12"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

#### Source File

- BoostToken.sol

```
5 // SPDX-License-Identifier: Unlicensed
6 pragma solidity ^0.6.12;
7
8 abstract contract Context {
9 function _msgSender() internal view virtual returns (address payable) {
10
```





### SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 666

#### **Iow SEVERITY**

It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwap" is internal. Other possible visibility settings are public and private.

#### Source File

- BoostToken.sol

```
665
666 bool inSwap = false;
667 bool public swapEnabled = true;
668
669 uint256 private _maxTxAmount = 5000000 * 10**18;
670
```



# SWC-115 | USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.

LINE 872

#### **Iow SEVERITY**

Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" unless you really know what you are doing.

#### Source File

- BoostToken.sol

#### Locations

871 require(!\_isBlackListedBot[msg.sender], "You are blacklisted");
872 require(!\_isBlackListedBot[tx.origin], "You are blacklisted");
873 if(sender != owner() && recipient != owner()) {
874 require(amount <= \_maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
875 }
876</pre>



**LINE 807** 

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
806 for (uint256 i = 0; i < _blackListedBots.length; i++) {
807 if (_blackListedBots[i] == account) {
808 _blackListedBots[i] = _blackListedBots[_blackListedBots.length - 1];
809 _isBlackListedBot[account] = false;
810 _blackListedBots.pop();
811</pre>
```



**LINE 808** 

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
807 if (_blackListedBots[i] == account) {
808 _blackListedBots[i] = _blackListedBots[_blackListedBots.length - 1];
809 _isBlackListedBot[account] = false;
810 _blackListedBots.pop();
811 break;
812
```



**LINE 808** 

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
807 if (_blackListedBots[i] == account) {
808 _blackListedBots[i] = _blackListedBots[_blackListedBots.length - 1];
809 _isBlackListedBot[account] = false;
810 _blackListedBots.pop();
811 break;
812
```



**LINE 829** 

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
828 for (uint256 i = 0; i < _excluded.length; i++) {
829 if (_excluded[i] == account) {
830 _excluded[i] = _excluded[_excluded.length - 1];
831 _tOwned[account] = 0;
832 _isExcluded[account] = false;
833</pre>
```



**LINE 830** 

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
829 if (_excluded[i] == account) {
830 _excluded[i] = _excluded[_excluded.length - 1];
831 _tOwned[account] = 0;
832 _isExcluded[account] = false;
833 _excluded.pop();
834
```



**LINE 830** 

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
829 if (_excluded[i] == account) {
830 _excluded[i] = _excluded[_excluded.length - 1];
831 _tOwned[account] = 0;
832 _isExcluded[account] = false;
833 _excluded.pop();
834
```



LINE 917

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
916 address[] memory path = new address[](2);
917 path[0] = address(this);
918 path[1] = uniswapV2Router.WETH();
919
920 _approve(address(this), address(uniswapV2Router), tokenAmount);
921
```



LINE 918

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
917 path[0] = address(this);
918 path[1] = uniswapV2Router.WETH();
919
920 _approve(address(this), address(uniswapV2Router), tokenAmount);
921
922
```



LINE 1060

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
1059 for (uint256 i = 0; i < _excluded.length; i++) {
1060 if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
1061 rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1062 tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1063 }
1064
```



LINE 1060

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
1059 for (uint256 i = 0; i < _excluded.length; i++) {
1060 if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
1061 rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1062 tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1063 }
1064
```



LINE 1061

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
1060 if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
1061 rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1062 tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1063 }
1064 if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
1065</pre>
```



LINE 1062

#### **Iow SEVERITY**

The index access expression can cause an exception in case of use of invalid array index value.

#### Source File

- BoostToken.sol

```
1061 rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1062 tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1063 }
1064 if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
1065 return (rSupply, tSupply);
1066
```



## 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 Sysfixed'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 Sysfixed 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 is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

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.

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 Sysfixed and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Sysfixed) owe no duty of care.



# ABOUT US

Sysfixed is a blockchain security certification organization established in 2021 with the objective to provide smart contract security services and verify their correctness in blockchain-based protocols. Sysfixed automatically scans for security vulnerabilities in Ethereum and other EVM-based blockchain smart contracts. Sysfixed a comprehensive range of analysis techniques—including static analysis, dynamic analysis, and symbolic execution—can accurately detect security vulnerabilities to provide an in-depth analysis report. With a vibrant ecosystem of world-class integration partners that amplify developer productivity, Sysfixed can be utilized in all phases of your project's lifecycle. Our team of security experts is dedicated to the research and improvement of our tools and techniques used to fortify your code.