Revoluzion Audit


# Audit Report

| | |
|---|---|
| Name | : **DBDT TOKEN** |
| Symbol | : **DBDT** |
| Decimals | : **18** |
| Address | : **0x000c7603cc3de5360c56bb5429f371932675cec7** |
| Owner | : **0x26bce0516066613b5c9BfB73e646b300b4D24e13** |
| Network | : **Binance Smart Chain (Mainnet)** |
| Type | : **BEP20** |
| Audited on | : **16 November 2022** |
| Updated on | : **19 November 2022** |

# Revoluzion Audit

## Contents

## Project Overview

| | |
|---|---|
| Name | DBDT TOKEN |
| Symbol | DBDT |
| Decimals | 18 |
| Total Supply | 1,000,000,000 |
| Tax | Buy 3% \| Sell 16% — ( Variable Tax ) |
| Compiler Version | v0.8.17+commit.8df45f5f |
| Optimization | Yes with 200 runs |
| License Type | MIT |
| Explorer Link | https://bscscan.com/address/0x000c7603cc3de5360c56bb5429f371932675cec7 |
| Create Tx | 0x9146a77133d8b997fd2e9ec2759f9ae6920fceb8412d629df7371f05ce799a29 |
| Creator | 0x26bce0516066613b5c9BfB73e646b300b4D24e13 |
| Featured Wallet | Company Wallet — 0x55271ae133fbf13c0e7b2621a4bc0b4e306ea701<br>Market Wallet — 0x7f4fa204d3f5686e6c91ff24c8d573dcffafc75a<br>Pool Game Wallet — 0xd6c40901122dd702a35a9e2e84b52fa9e622bdaf |
| GitHub Link | https://github.com/dbdtoken-caudit |
| Website | https://www.dbdtoken.com |

# Project Description

### According to their whitepaper

DBDT TOKEN was developed to financially manage the entire self-sustaining ecosystem developed and linked to Blockchain and Web3 technology! Responsible for governing the Marketplace, Game, Metaverse, CrossChain and Services.

**Release Date** : TBA

**Category** : Utility Token

# Online Presence

## About Website

**Registrar** : https://www.register.it

**Domain Expiration** : 2023-10-15

**SSL Certificate** : Issued by cPanel, Inc.

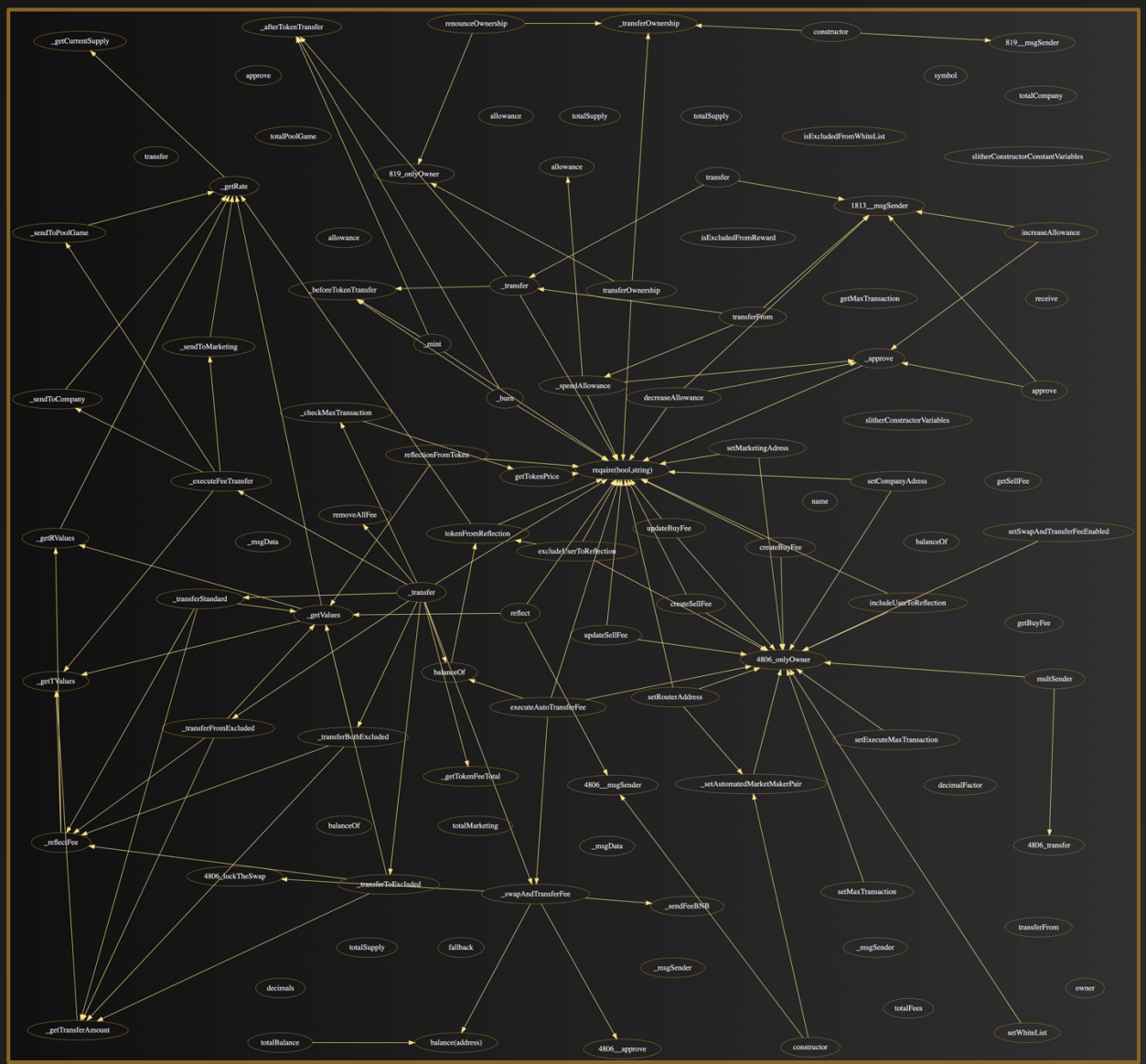
## Official Links

| | |
|---|---|
| Website | https://dbdtoken.com |
| Instagram | https://instagram.com/dbdt.token |
| Twitter | https://twitter.com/dbdt_token |
| GitHub | https://github.com/dbdtoken-caudit |
| Telegram News | https://t.me/dbdtokennews |
| Telegram Portuguese | https://t.me/dbdtokenpt |
| Telegram English | https://t.me/dbdtokenglobal |
| Telegram Chinese | https://t.me/dbdtokench |

## The Team

| | |
|---|---|
| **About** | We only interacted with the owner for the audit. However, there are no KYC procedure being conducted by Revoluzion on any of DBDT TOKEN's team members. |
| **KYC Issuer** | N/A |
| **Member's KYC'd** | N/A |
| **KYC Date** | N/A |
| **Certificate Link** | N/A |
| **Task Completed** | N/A |

# Contract Functions Interaction



# Audit Overview

### Threat Level

When conducting audit on smart contract(s), we first look for known vulnerabilities and issues within the code because any exploitation on such vulnerabilities and issues by malicious actors could potentially result in serious financial damage to the projects. All the issues and vulnerabilities will be categorized into the categories as provided below.

**Critical**

This category provides issues and vulnerabilities that are critical to the performance/functionality of the smart contract and should be fixed by project creator before moving to a live environment.

**Medium**

This category provides issues and vulnerabilities that are not that critical to the performance/functionality of the smart contract but is recommended to be fixed by project creator before moving to a live environment.

**Minor**

This category provides issues and vulnerabilities that are minor to the performance/functionality of the smart contract and can remain unfixed by project creator before moving to a live environment.

**Informational**

This category provides issues and vulnerability that have insignificant effect on the performance/functionality of the smart contract and can remain unfixed by project creator before moving to a live environment. However, fixing them can further improve the efficacy or security for features with a risk-free factor.

## Notable Information

- Contract Owner cannot stop or pause transactions.
- Contract Owner cannot transfer tokens from specific address.
- Contract Owner cannot mint new tokens after deploying smart contract.
- Contract Owner cannot burn tokens from specific wallet.
- Contract Owner cannot blacklist wallets from selling.
- There are no compiler warnings when compiling the smart contracts.
- Contract is using safe Zeppelin modules.

## Caution

- According to owner, buy and sell fee will be set to a total of 3% and 16% respectively. Please note that the distribution can be adjust to not more than 25% in total for both buy and sell fee. Both fee will only be set once the pool has been finalised.

- Daily transaction has a maximum limit of at least 2 BNB worth of tokens and owner cannot set the limit to be any lower than that.

## Bugs and Optimizations Detection

This table is based on the result obtained from running the smart contract through Slither's Solidity static analysis.

| What it detects | Impact | Confidence | Status |
|---|---|---|---|
| Storage abiencoderv2 array | High | High | Passed |
| transferFrom uses arbitrary from | High | High | Passed |
| Modifying storage array by value | High | High | Passed |
| The order of parameters in a shift instruction is incorrect. | High | High | Passed |
| Multiple constructor schemes | High | High | Passed |
| Contract's name reused | High | High | Passed |
| Detected unprotected variables | High | High | Passed |
| Public mappings with nested variables | High | High | Passed |
| Right-To-Left-Override control character is used | High | High | Passed |
| State variables shadowing | High | High | Passed |
| Functions allowing anyone to destruct the contract | High | High | Passed |
| Uninitialized state variables | High | High | Passed |
| Uninitialized storage variables | High | High | Passed |
| Unprotected upgradeable contract | High | High | Passed |

| | | | |
|---|---|---|---|
| transferFrom uses arbitrary from with permit | High | Medium | Passed |
| Functions that send Ether to arbitrary destinations | High | Medium | Moderated |
| Tainted array length assignment | High | Medium | Passed |
| Controlled delegatecall destination | High | Medium | Passed |
| Payable functions using delegatecall inside a loop | High | Medium | Passed |
| msg.value inside a loop | High | Medium | Passed |
| Reentrancy vulnerabilities (theft of ethers) | High | Medium | Moderated |
| Signed storage integer array compiler bug | High | Medium | Passed |
| Unchecked tokens transfer | High | Medium | Passed |
| Weak PRNG | High | Medium | Passed |
| Detects ERC20 tokens that have a function whose signature collides with EIP-2612's DOMAIN_SEPARATOR() | Medium | High | Passed |
| Detect dangerous enum conversion | Medium | High | Passed |
| Incorrect ERC20 interfaces | Medium | High | Passed |
| Incorrect ERC721 interfaces | Medium | High | Passed |
| Dangerous strict equalities | Medium | High | Passed |
| Contracts that lock ether | Medium | High | Passed |

| | | | |
|---|---|---|---|
| Deletion on mapping containing a structure | Medium | High | Passed |
| State variables shadowing from abstract contracts | Medium | High | Passed |
| Tautology or contradiction | Medium | High | Passed |
| Unused write | Medium | High | Moderated |
| Misuse of Boolean constant | Medium | Medium | Passed |
| Constant functions using assembly code | Medium | Medium | Passed |
| Constant functions changing the state | Medium | Medium | Passed |
| Imprecise arithmetic operations order | Medium | Medium | Passed |
| Reentrancy vulnerabilities (no theft of ethers) | Medium | Medium | Passed |
| Reused base constructor | Medium | Medium | Passed |
| Dangerous usage of tx.origin | Medium | Medium | Passed |
| Unchecked low-level calls | Medium | Medium | Passed |
| Unchecked send | Medium | Medium | Passed |
| Uninitialized local variables | Medium | Medium | Passed |
| Unused return values | Medium | Medium | Passed |
| Modifiers that can return the default value | Low | High | Passed |
| Built-in symbol shadowing | Low | High | Passed |

| | | | |
|---|---|---|---|
| Local variables shadowing | Low | High | Passed |
| Uninitialized function pointer calls in constructors | Low | High | Passed |
| Local variables used prior their declaration | Low | High | Passed |
| Constructor called not implemented | Low | High | Passed |
| Multiple calls in a loop | Low | Medium | Moderated |
| Missing Events Access Control | Low | Medium | Passed |
| Missing Events Arithmetic | Low | Medium | Passed |
| Dangerous unary expressions | Low | Medium | Passed |
| Missing Zero Address Validation | Low | Medium | Moderated |
| Benign reentrancy vulnerabilities | Low | Medium | Moderated |
| Reentrancy vulnerabilities leading to out-of-order Events | Low | Medium | Moderated |
| Dangerous usage of block.timestamp | Low | Medium | Moderated |
| Assembly usage | Informational | High | Moderated |
| Assert state change | Informational | High | Passed |
| Comparison to boolean constant | Informational | High | Passed |
| Deprecated Solidity Standards | Informational | High | Passed |

| | | | |
|---|---|---|---|
| Un-indexed ERC20 event parameters | Informational | High | Passed |
| Function initializing state variables | Informational | High | Passed |
| Low level calls | Informational | High | Moderated |
| Missing inheritance | Informational | High | Passed |
| Conformity to Solidity naming conventions | Informational | High | Moderated |
| If different pragma directives are used | Informational | High | Passed |
| Redundant statements | Informational | High | Passed |
| Incorrect Solidity version | Informational | High | Passed |
| Unimplemented functions | Informational | High | Passed |
| Unused state variables | Informational | High | Passed |
| Costly operations in a loop | Informational | Medium | Moderated |
| Functions that are not used | Informational | Medium | Moderated |
| Reentrancy vulnerabilities through send and transfer | Informational | Medium | Passed |

| | | | |
|---|---|---|---|
| Variable names are too similar | Informational | Medium | **Moderated** |
| Conformance to numeric notation best practices | Informational | Medium | **Passed** |
| State variables that could be declared constant | Optimization | High | **Passed** |
| Public function that could be declared external | Optimization | High | **Passed** |

## Contract Diagnostic

**Link for initial smart contract commit being audited on GitHub:**

https://github.com/RevoluzionToken/Revoluzion-Audits/commit/cfb524495e96462fdb795e9cac1a5bd4e6a309fe

| CODE | SEVERITY | DESCRIPTION |
|---|---|---|
| SWC-103 | Minor | A floating pragma is set. |
| VS | Minor | Variables are used locally prior to their declaration. |
| CaL | Minor | Loops with multiple calls. |
| CoL | Informational | Loop with costly operations. |
| SS | Informational | Function state shadowing other function. |
| DC | Informational | Dead code. |
| NC | Informational | Naming convention. |
| SN | Informational | Similar name. |
| US | Informational | Unused state variable. |
| CS | Informational | State variable that can be declared as constant. |

### SWC-103 — A floating pragma is set

| | |
|---|---|
| **SEVERITY** | Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L10 |
| **DESCRIPTION** | The current pragma Solidity directive is set as ""^0.8.16"". |
| **RECOMMENDATIONS** | Project creator is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. It is important if the project rely on bytecode-level verification of the code. |
| **STATUS** | **N/A** |

### VS — Variables are used locally prior to their declaration

| | |
|---|---|
| **SEVERITY** | Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L2016 |
| **DESCRIPTION** | [SmartContToken.getTokenPrice.ResBNB] (#L2016) potentially used before declaration at #L2020<br><br>[SmartContToken.getTokenPrice.ResToken] (#L2016) potentially used before declaration at #L202 |
| **RECOMMENDATIONS** | Project creator might want to move all the declarations of the variable prior to any usage and make sure that reaching a variable declaration does not depend on some conditional if it is used unconditionally. |
| **STATUS** | **FIXED by Revoluzion**<br><br>https://github.com/RevoluzionToken/Revoluzion-Audits/commit/bc8690990d0072b6ef30d6a49b01a862474e16e0 |

## CaL — Loops with multiple calls

| | |
|---|---|
| **SEVERITY** | Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L1907-1938, 1946-1952, 2007-2024 |
| **DESCRIPTION** | [SmartContToken._swapAndTransferFee] (#L1907-1938) has external calls inside a loop at line #L1911, 1918-1924<br><br>[SmartContToken._sendFeeBNB] (#L1946-1952) has external calls inside a loop at line #L1948<br><br>[SmartContToken.getTokenPrice] (#L2007-2024) has external calls inside a loop at line #L2015, 2016, 2020 |
| **RECOMMENDATIONS** | Project creator can choose to either make use of pull over push strategy for external calls or ignore the issues since the logic does require such function(s) |
| **STATUS** | **N/A** |

## CoL — Loop with costly operations

| | |
|---|---|
| **SEVERITY** | Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L1273-1277, 1447-1458, 1461-1554, 1637-1652, 1789-1798 |
| **DESCRIPTION** | [SmartContToken.lockTheSwap] (#L1273-1277) has costly operations inside a loop at #L1274, 1276<br><br>[SmartContToken.includeUserToReflection] (#L1447-1458) has costly operations inside a loop at #L1454<br><br>[SmartContToken._transfer] (#L1461-1554) has costly operations inside a loop at #L1480, 1481, 1482, 1483, 1494, 1495, 1496, 1497, 1511, 1513, 1514, 1521, 1538, 1552<br><br>[SmartContToken._reflectFee] (#L1637-1652) has costly operations inside a loop at #L1645, 1646, 1647, 1648, 1649<br><br>[SmartContToken.removeAllFee] (#L1789-1798) has costly operations inside a loop at #L1794, 1795, 1796, 1797 |
| **RECOMMENDATIONS** | Project creator can use local variables instead to hold the loop computation result for the fees. |
| **STATUS** | **N/A** |

## SS — Function state shadowing other function

| | |
|---|---|
| **SEVERITY** | Informational — Medium |
| **LOCATION(S)** | DBDTTOKEN.sol#L857, 1199 |
| **DESCRIPTION** | [SmartContToken._allowances] (#L1199) shadows [BEP20._allowances] (#L857) |
| **RECOMMENDATIONS** | Project creator can choose to either rename the parameters to something else or completely ignore them.<br><br>However, the codes at #L1199 can completely be removed since it is not being used anywhere within the contract. |
| **STATUS** | **FIXED by Revoluzion**<br><br>https://github.com/RevoluzionToken/Revoluzion-Audits/commit/911258c97e8f0bc998e60746e541f8f220797ecb |

## DC — Dead code

| | |
|---|---|
| **SEVERITY** | Informational — Medium |
| **LOCATION(S)** | DBDTTOKEN.sol#L376-378, 598-600, 614-621, 645-656, 674-686, 706-711, 719-725, 738-750, 758-769, 771-799, 1028-1051, 1062-1072, 1085-1100, 1166-1170, 1186-1190 |
| **DESCRIPTION** | [Context._msgData()] (#L376-378) is never used and should be removed.<br><br>[SafeMath.mod()] (#L598-600) is never used and should be removed.<br><br>[SafeMath.mod()] (#L614-621) is never used and should be removed.<br><br>[Address.isContract()] (#L645-656) is never used and should be removed.<br><br>[Address.sendValue()] (#L674-686) is never used and should be removed.<br><br>[Address.functionCall()] (#L706-711) is never used and should be removed.<br><br>[Address.functionCall()] (#L719-725) is never used and should be removed.<br><br>[Address.functionCallWithValue()] (#L738-750) is never used and should be removed.<br><br>[Address.functionCallWithValue()] (#L758-769) is never used and should be removed.<br><br>[Address._functionCallWithValue()] (#L771-799) is never used and should be removed.<br><br>[BEP20._transfer()] (#L1028-1051) is never used and should be removed.<br><br>[BEP20._mint()] (#L1062-1072) is never used and should be removed. |

| | |
|---|---|
| | [BEP20._burn()] (#L1085-1100) is never used and should be removed.<br><br>[BEP20._beforeTokenTransfer()] (#L1166-1170) is never used and should be removed.<br><br>[BEP20._afterTokenTransfer()] (#L1186-1190) is never used and should be removed. |
| **RECOMMENDATIONS** | Based on our analysis, the Address, Context and SafeMath smart contracts is the standard that is a direct fork from Open Zeppelin while BEP20 smart contract is a derivative from Open Zeppelin standard. These smart contracts were used within the contract itself. Except for Address smart contract which is not being used anywhere throughout the smart contract as a whole.<br><br>It is recommended for project creator to remove those functions to further optimize the smart contract since they are not used anywhere at all. Doing so will reduce the amount of gas required when deploying the smart contract. |
| **STATUS** | **N/A** |

## NC — Naming convention

| | |
|---|---|
| **SEVERITY** | Informational — Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L73, 75, 106, 155, 394, 858, 1213, 1214, 1215, 1216, 1219, 1220, 1221, 1222, 1244-1248, 1257, 1258, 1260, 1284, 1805, 1806, 1807, 1818, 1819, 1820, 1854, 1855, 1856, 1857, 1869, 1870, 1871, 1872, 1984, 1988, 1992 |
| **DESCRIPTION** | [IUniswapV2Pair.DOMAIN_SEPARATOR()] (#L73) is not in mixedCase.<br>[IUniswapV2Pair.PERMIT_TYPEHASH()] (#L75) is not in mixedCase.<br>[IUniswapV2Pair.MINIMUM_LIQUIDITY()] (#L106) is not in mixedCase.<br>[IUniswapV2Router01.WETH()] (#L155) is not in mixedCase.<br>[Ownable._owner] (#L394) is not in mixedCase.<br>[BEP20._totalSupply] (#L858) is not in mixedCase.<br>[SmartContToken.MarketAddress] (#L1213) is not in mixedCase.<br>[SmartContToken.CompanyAddress] (#L1214) is not in mixedCase.<br>[SmartContToken.PoolGameAddress] (#L1215) is not in mixedCase.<br>[SmartContToken._bnb_Address] (#L1216) is not in mixedCase.<br>[SmartContToken._TAX_FEE] (#L1219) is not in mixedCase.<br>[SmartContToken._COMPANY_FEE] (#L1220) is not in mixedCase.<br>[SmartContToken._MARKETING_FEE] (#L1221) is not in mixedCase.<br>[SmartContToken._POOLGAME_FEE] (#L1222) is not in mixedCase. |

| | |
|---|---|
| | [SmartContToken.userSell] (#L1244-1248) is not in CapWords.<br><br>[SmartContToken._DECIMALFACTOR] (#L1257) is not in mixedCase.<br><br>[SmartContToken._tTotal] (#L1258) is not in UPPER_CASE_WITH_UNDERSCORES.<br><br>[SmartContToken._MaxTransaction] (#L1260) is not in mixedCase.<br><br>[SmartContToken._TokenReserv] (#L1284) is not in mixedCase.<br><br>[SmartContToken.createBuyFee()._txFee] (#L1805) is not in mixedCase.<br><br>[SmartContToken.createBuyFee()._MarketingFee] (#L1806) is not in mixedCase.<br><br>[SmartContToken.createBuyFee()._CompanyFee] (#L1807) is not in mixedCase.<br><br>[SmartContToken.updateBuyFee()._txFee] (#L1818) is not in mixedCase.<br><br>[SmartContToken.updateBuyFee()._MarketingFee] (#L1819) is not in mixedCase.<br><br>[SmartContToken.updateBuyFee()._CompanyFee] (#L1820) is not in mixedCase.<br><br>[SmartContToken.createSellFee()._txFee] (#L1854) is not in mixedCase.<br><br>[SmartContToken.createSellFee()._MarketingFee] (#L1855) is not in mixedCase.<br><br>[SmartContToken.createSellFee()._CompanyFee] (#L1856) is not in mixedCase.<br><br>[SmartContToken.createSellFee()._PoolGameFee] (#L1857) is not in mixedCase.<br><br>[SmartContToken.updateSellFee()._txFee] (#L1869) is not in mixedCase.<br><br>[SmartContToken.updateSellFee()._MarketingFee] (#L1870) is not in mixedCase. |

| | |
|---|---|
| | [SmartContToken.updateSellFee()._CompanyFee] (#L1871) is not in mixedCase.<br><br>[SmartContToken.updateSellFee()._PoolGameFee] (#L1872) is not in mixedCase.<br><br>[SmartContToken.setCompanyAdress()._CompanyAddress] (#L1984) is not in mixedCase.<br><br>[SmartContToken.setMarketingAdress()._MarketingAddress] (#L1988) is not in mixedCase.<br><br>[SmartContToken.setMaxTransaction().amount_BNB] (#L1992) is not in mixedCase. |
| **RECOMMENDATIONS** | Based on our analysis, the IUniswapRouter01 and IUniswapV2Pair smart contracts are direct fork from Uniswap while Ownable is the direct fork from Open Zeppelin while BEP20 smart contract is a derivative from Open Zeppelin standard. Although the name doesn't conform to the standard convention, it's still okay to leave it be to avoid from potentially breaking any external function.<br><br>However, for SmartContToken smart contract, it is okay for project creator to update the name of the parameters in those functions so that they conform to the standard naming convention. |
| **STATUS** | **FIXED by Revoluzion**<br><br>https://github.com/RevoluzionToken/Revoluzion-Audits/commit/bc8690990d0072b6ef30d6a49b01a862474e16e0 |

## SN — Similar name

| | |
|---|---|
| **SEVERITY** | Informational — Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L160, 161, 1420, 1471, 1586, 1587, 1600, 1601, 1614, 1615, 1629, 1630, 1713, 1765, 1907, 1942, 1954 |
| **DESCRIPTION** | [IUniswapV2Router01.addLiquidity().amountADesired] (#L160) is too similar to [IUniswapV2Router01.addLiquidity().amountBDesired] (#L161).<br><br>[SmartContToken.reflectionFromToken().rTransferAmount] (#L1420) is too similar to [SmartContToken._transferToExcluded().tTransferAmount] (#L1587).<br><br>[SmartContToken.reflectionFromToken().rTransferAmount] (#L1420) is too similar to [SmartContToken._transferFromExcluded().tTransferAmount] (#L1601).<br><br>[SmartContToken.reflectionFromToken().rTransferAmount] (#L1420) is too similar to [SmartContToken._transferBothExcluded().tTransferAmount] (#L1615).<br><br>[SmartContToken.reflectionFromToken().rTransferAmount] (#L1420) is too similar to [SmartContToken._transferStandard().tTransferAmount] (#L1630).<br><br>[SmartContToken.reflectionFromToken().rTransferAmount] (#L1420) is too similar to [SmartContToken._getTransferAmount().tTransferAmount] (#L1765).<br><br>[SmartContToken._transferToExcluded().rTransferAmount] (#L1586) is too similar to [SmartContToken._transferToExcluded().tTransferAmount] (#L1587).<br><br>[SmartContToken._transferToExcluded().rTransferAmount] (#L1586) is too similar to |

| | |
|---|---|
| | [SmartContToken._transferFromExcluded().tTransferAmount] (#L1601).<br><br>[SmartContToken._transferToExcluded().rTransferAmount] (#L1586) is too similar to [SmartContToken._transferBothExcluded().tTransferAmount] (#L1615).<br><br>[SmartContToken._transferToExcluded().rTransferAmount] (#L1586) is too similar to [SmartContToken._transferStandard().tTransferAmount] (#L1630).<br><br>[SmartContToken._transferToExcluded().rTransferAmount] (#L1586) is too similar to [SmartContToken._getTransferAmount().tTransferAmount] (#L1765).<br><br>[SmartContToken._transferFromExcluded().rTransferAmount] (#L1600) is too similar to [SmartContToken._transferToExcluded().tTransferAmount] (#L1587).<br><br>[SmartContToken._transferFromExcluded().rTransferAmount] (#L1600) is too similar to [SmartContToken._transferFromExcluded().tTransferAmount] (#L1601).<br><br>[SmartContToken._transferFromExcluded().rTransferAmount] (#L1600) is too similar to [SmartContToken._transferBothExcluded().tTransferAmount] (#L1615).<br><br>[SmartContToken._transferFromExcluded().rTransferAmount] (#L1600) is too similar to [SmartContToken._transferStandard(address,address,uint256).tTransferAmount] (#L1630).<br><br>[SmartContToken._transferFromExcluded().rTransferAmount] (#L1600) is too similar to [SmartContToken._getTransferAmount().tTransferAmount] (#L1765).<br><br>[SmartContToken._transferBothExcluded().rTransferAmount] (#L1614) is too similar to [SmartContToken._transferFromExcluded(address,address,uint256).tTransferAmount] (#L1601). |

| | |
|---|---|
| | [SmartContToken._transferBothExcluded().rTransferAmount] (#L1614) is too similar to [SmartContToken._transferToExcluded().tTransferAmount] (#L1587).<br><br>[SmartContToken._transferBothExcluded().rTransferAmount] (#L1614) is too similar to [SmartContToken._transferBothExcluded().tTransferAmount] (#L1615).<br><br>[SmartContToken._transferBothExcluded().rTransferAmount] (#L1614) is too similar to [SmartContToken._transferStandard().tTransferAmount] (#L1630).<br><br>[SmartContToken._transferBothExcluded().rTransferAmount] (#L1614) is too similar to [SmartContToken._getTransferAmount().tTransferAmount] (#L1765).<br><br>[SmartContToken._transferStandard().rTransferAmount] (#L1629) is too similar to [SmartContToken._transferToExcluded().tTransferAmount] (#L1587).<br><br>[SmartContToken._transferStandard().rTransferAmount] (#L1629) is too similar to [SmartContToken._transferFromExcluded().tTransferAmount] (#L1601).<br><br>[SmartContToken._transferStandard().rTransferAmount] (#L1629) is too similar to [SmartContToken._transferBothExcluded().tTransferAmount] (#L1615).<br><br>[SmartContToken._transferStandard().rTransferAmount] (#L1629) is too similar to [SmartContToken._transferStandard().tTransferAmount] (#L1630).<br><br>[SmartContToken._transferStandard().rTransferAmount] (#L1629) is too similar to [SmartContToken._getTransferAmount().tTransferAmount] (#L1765).<br><br>[SmartContToken._getValues().rTransferAmount] (#L1713) is too similar to |

| | |
|---|---|
| | [SmartContToken._transferToExcluded().tTransferAmount] (#L1587).<br><br>[SmartContToken._getValues().rTransferAmount] (#L1713) is too similar to [SmartContToken._transferFromExcluded().tTransferAmount] (#L1601).<br><br>[SmartContToken._getValues().rTransferAmount] (#L1713) is too similar to [SmartContToken._transferBothExcluded().tTransferAmount] (#L1615).<br><br>[SmartContToken._getValues().rTransferAmount] (#L1713) is too similar to [SmartContToken._transferStandard().tTransferAmount] (#L1630).<br><br>[SmartContToken._getValues().rTransferAmount] (#L1713) is too similar to [SmartContToken._getTransferAmount().tTransferAmount] (#L1765).<br><br>[SmartContToken.executeAutoTransferFee().tokemAmount] (#L1954) is too similar to [SmartContToken._transfer().tokenAmount] (#L1471).<br><br>[SmartContToken.executeAutoTransferFee().tokemAmount] (#L1954) is too similar to [SmartContToken._swapAndTransferFee().tokenAmount] (#L1907).<br><br>[SmartContToken.executeAutoTransferFee().tokemAmount] (#L1954) is too similar to [SmartContToken._getTokenFeeTotal().tokenAmount] (#L1942). |
| **RECOMMENDATIONS** | Based on our analysis, the IUniswapV2Router smart contract is a direct fork from Uniswap. Although their names are too similar, it's still okay to leave them be for the purpose of following the standard parameter declaration that is widely used as reference.<br><br>However, for the SmartContToken smart contract, it is recommended for project creator to make the changes to the names to prevent variables from having similar names which will then make the code hard to review.. |

| STATUS | **FIXED by Revoluzion**<br>https://github.com/RevoluzionToken/Revoluzion-Audits/commit/bc8690990d0072b6ef30d6a49b01a862474e16e0 |
| --- | --- |

## US — Unused state variable

| | |
|---|---|
| **SEVERITY** | Informational — Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L1177-1211 |
| **DESCRIPTION** | [SmartContToken._DECIMALFACTOR] (#L1257) is never used.<br><br>[SmartContToken._allowances] (#L1257) is never used. |
| **RECOMMENDATIONS** | Based on our analysis, project creator can either create a function to call the value for preview or completely remove it from the code. Project creator can also opt to keep them as it doesn't give any issue other than increasing the gas fee when deploying the smart contract. |
| **STATUS** | **FIXED by Revoluzion**<br><br>https://github.com/RevoluzionToken/Revoluzion-Audits/commit/bc8690990d0072b6ef30d6a49b01a862474e16e0 |

## CS — State variable that can be declared as constant

| | |
|---|---|
| **SEVERITY** | Informational — Minor |
| **LOCATION(S)** | DBDTTOKEN.sol#L1252, 1253, 1254, 1257 |
| **DESCRIPTION** | [SmartContToken._name] (#L1252) should be constant.<br>[SmartContToken._symbol] (#L1253) should be constant.<br>[SmartContToken._decimals] (#L1254) should be constant<br>[SmartContToken._DECIMALFACTOR] (#L1257) should be constant. . |
| **RECOMMENDATIONS** | Based on our analysis, these variables should be declared as constant since they don't change throughout smart contract. |
| **STATUS** | **FIXED by Revoluzion**<br>https://github.com/RevoluzionToken/Revoluzion-Audits/commit/bc8690990d0072b6ef30d6a49b01a862474e16e0 |

# Disclaimer

**This report only shows findings based on our limited project analysis** according to the good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall online presence and team transparency details of which are set out in this report. To get a full view of our analysis, **it is important for you to read the full report.** Under no circumstances did Revoluzion Audit receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. **Our team provides no guarantees against the sale of team tokens or the removal of liquidity by the project** audited in this document.

While **we have done our best to conduct thorough analysis to produce this report**, it is crucial to note that you should not rely solely on this report and use the content provided in this document as financial advice or a reason to buy any investment. The Our team disclaims any liability against us for the resulting losses based on the you decision made by relying on the content of this report. **You must conduct your own independent investigations before making any decisions** to protect yourselves from being scammed. We go into more detail on this in the disclaimer clause in the next page — please make sure to read it in full.

## Full Disclaimer Clause

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 all copies of this report downloaded and/or printed by you. This report is provided for information purposes only, on a non-reliance basis and does not constitute to any investment advice.

No one shall have any right to rely on the report or its contents, and Revoluzion Audit and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (collectively known as Revoluzion) owe no duty of care towards you or any other person, nor do we 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 Revoluzion 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, Revoluzion hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Revoluzion, 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.

The analysis of the security is purely based on the smart contracts, website, social media, and team.