expelee

Building the Futuristic Blockchain Ecosystem

Audit Report

FOR

Block Vest

OVERVIEW

The Expelee team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

Audit Result	Passed with high Risk
KYC Verification	Done
Audit Date	26 Sep 2022

Why high risk?

owner has privileges to disable trades, or set a high amount of tax

-Team Expelee

PROJECT DESCRIPTION

Block Vest

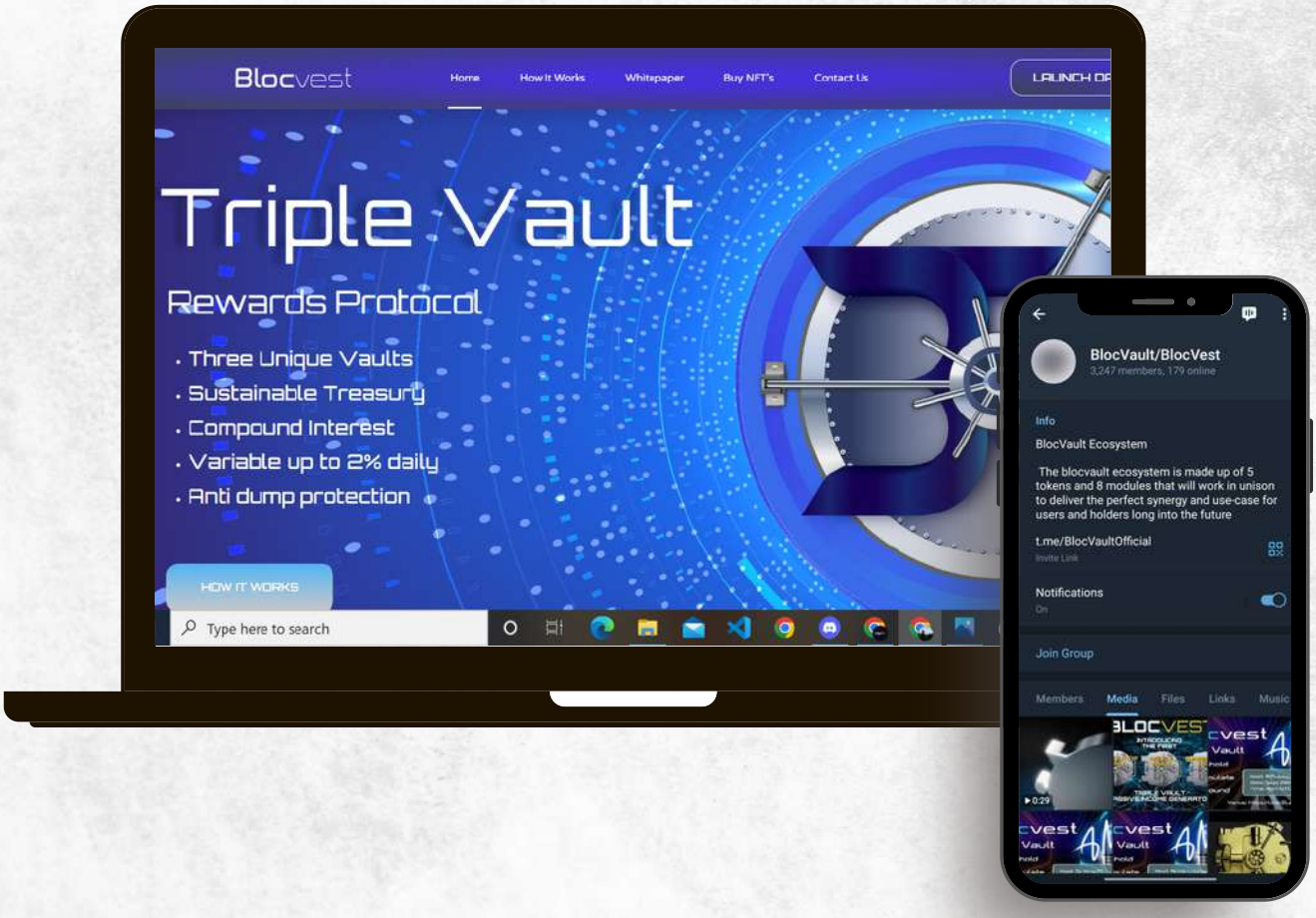
Blocvest is a revolutionary deflationary token with three very unique vaults. The Shareholder, Accumulator and the trickle vault. The system was developed to offer different rewards and in different ways depending on how you decide to utilize them. The BVST token is paired to BNB which means it will see significant gains once the market stabilizes and starts to recover. We have also added anti whale functions and higher tax thresholds to ensure a stable future for the ecosystem over time

expelee

expelee.com

Social Media Profiles

Block Vest



https://blocvest.io/

https://t.me/BlocVaultOfficial

https://twitter.com/BlocVaultAPP

It's always good to check the social profiles of the project, before making your investment.

-Team Expelee

expelee

expelee.com

CONTRACT DETAILS

Token Name

BlocVest Token

Symbol

BVST

Network

BSC

Language

Solidity

Contract Address (Verified)

0x592032513b329a0956b3f14d661119880f2361a6

Total Supply

10,000,000

Decimals

18

Compiler

v0.8.17+commit.8df45f5f

License

MIT license

Contract SHA-256 Checksum:

bc0acef1885918032ab15d84df74e850fe449ca4d6eaea3ac81559e674b8166a

AUDIT METHODOLOGY

Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.

Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability

Tools

- DE
- Open Zeppelin
- Code Analyzer
- Solidity Code
- Complier
- Hardhat

FUNCTION OVERVIEW

Can Take Back Ownership	Not Detected
Owner Change Balance	Not Detected
Blacklist	Not Detected
Modify Fees	Not Detected
Proxy	Not Detected
Whitelisted	Not Detected
Anti Whale	Not Detected
Trading Cooldown	Not Detected
Transfer Pausable	Not Detected
Cannot Sell All	Not Detected
Hidden Owner	Not Detected
Mint	Not Detected

VULNERABILITY CHECKLIST

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions & reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

AUDIT SUMMARY

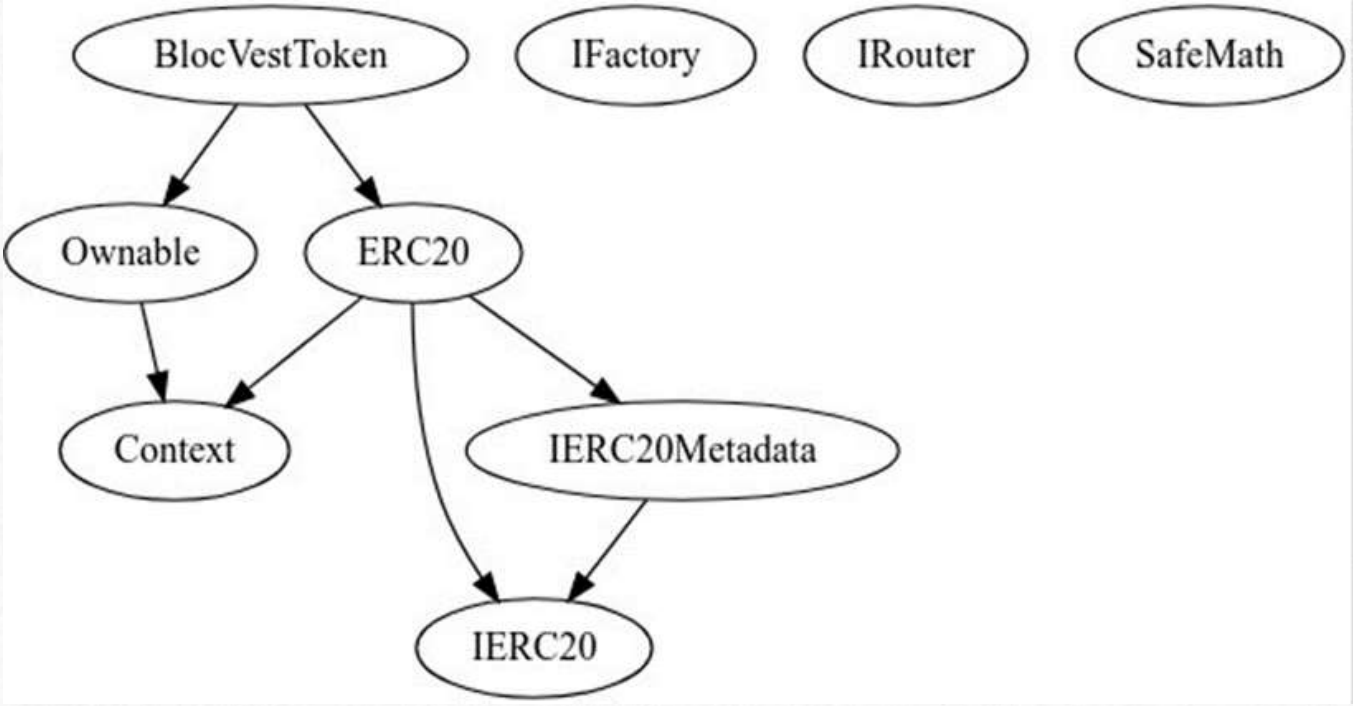
Ownership:

Current owner of BVST is:
0x258c7f570ae021a17bbfb9f6e7dde673701ad397

Contracts & Inheritance Tree:

all of below contractsare in this audit scope

BVST.sol



Summary

- There are 3 tax periods after launch, each with a specifiectax amount
- first period takes 3 blocks and 100% taxes all buys (anti-sniper)
- second period takes 3600 seconds which is 5 minutes taxes are 10% on buy and 30% on sell.
- third period takes 2 days and has 10% on buy and 30% on sell. taxes get accumolated and then sent to dev wallet (owner wallet) no rewards or reflections
- taxes can be hight to 100%
- owner is able to disable trading at any time

Launch on local blockchain

we launched BVST on our local blockchain and created liquiditypool on pancakeswap.
we performed couple of buy & sells to check taxes & swapAndLiqufity operations. everything worked as expected withouterrors, i.e taxes go collectedinside the contractand then swapped to BNB and BUSD:

Buys:

```
🟢Action : Buy
Trying to buy : 2486.3028900465589518 12 BVST
🔸Buyer : 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
🔸Received Amount : 2237.6726010419030566 31 BVST
🔸Buy Tax : 9.99 %
🔸Gas Used : 258038
🔸Monitored Wallets :
Contract : 248.6302890046558951 81 BVST
Contract : 0.0 ETH
Tax Wallet : 0.0 BVST
Tax Wallet : 0.0 ETH
Tax Wallet : 0.0 BVST
==================================
🟢Action : Buy
Trying to buy : 2473.9516862529411431 55 BVST
🔸Buyer : 0x90F79bf6EB2c4f870365E785982E1f101E93b906
🔸Received Amount : 2226.5565176276470288 4 BVST
🔸Buy Tax : 9.99 %
🔸Gas Used : 187296
🔸Monitored Wallets :
Contract : 496.0254576299500094 96 BVST
Contract : 0.0 ETH
Tax Wallet : 0.0 BVST
Tax Wallet : 0.0 ETH
Tax Wallet : 0.0 BVST
==================================
🟢Action : Buy
Trying to buy : 2461.6923356437626046 97 BVST
🔸Buyer : 0x15d34AAf54267DB7D7c367839AAf71A00a2C6A65
🔸Received Amount : 2215.5231020793863442 28 BVST
🔸Buy Tax : 9.99 %
🔸Gas Used : 187296
🔸Monitored Wallets :
Contract : 742.1946911943262699 65 BVST
```

Sells:

```
🔸Seller : 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
🔸Sell Amount : 1002.0
🔸Sell Tax : -64.05 %
🔸Gas Used : 559975
🔸Block Number : 23
🔸Monitoring Wallets:
Contract : 100.3728607759131441 29 BlocVest Token
Contract : 0.0 ETH
Tax Wallet : 0.0 BlocVest Token
Tax Wallet : 0.045035329664499853 ETH
Tax Wallet : 3740285.8090468919883886 25 Stable Coin
==================================
🟠Action : Sell
2226.556517627647028 84
🔸Seller : 0x90F79bf6EB2c4f870365E785982E1f101E93b906
🔸Sell Amount : 1002.0
🔸Sell Tax : -0.01 %
🔸Gas Used : 454870
🔸Block Number : 25
🔸Monitoring Wallets:
Contract : 100.2292026005941976 65 BlocVest Token
Contract : 0.0 ETH
Tax Wallet : 0.0 BlocVest Token
Tax Wallet : 0.0511101477117084 62 ETH
Tax Wallet : 4244598.4443564043052421 06 Stable Coin
==================================
🟠Action : Sell
2215.5231020793863442 28
🔸Seller : 0x15d34AAf54267DB7D7c367839AAf71A00a2C6A65
🔸Sell Amount : 1002.0
🔸Sell Tax : 0 %
🔸Gas Used : 454870
🔸Block Number : 27
🔸Monitoring Wallets:
Contract : 100.2291630169808907 45 BlocVest Token
```

MANUAL AUDIT

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to a methodology based on
OWASP standards.
Vulnerabilities are divided into three primary risk categories: high, medium, and low.

High-level considerations for vulnerabilities span the followingkey areas when conducting assessments:

- Malicious Input Handling
- Escalation of privileges
- Arithmetic
- Gas use

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Findings Summary

- High Risk Findings:7
- Medium Risk Findings:2
- Low Risk Findings:3
- Suggestions & discussion: 2
- Gas Optimizations : 5

High Risk Findings

Centralization Risks

- Owner is able to disabletrading by using deactivateTrading function. there is not any restrictions on this functionand this action can be performed at any time.
- Owner is able to disablea wallet from buying/selling/transfering tokens using blockAccount function. if token is not launched yet owner is able to add any wallet to blacklisted wallets,but if token is launchedand 172800 seconds is passed since launch time, this function would be disabledand owner can not blacklista wallet anymore
- Owner is able to cancel launch before launch tax periods (launch1,launch2, launch3), this period is 2 days in total, canceling launch means disablingtrades and resetinglaunch start time to 0.
- Owner is able to set buy and sell taxes up to 100% using setBaseFeesOnBuy and
- setBaseFeesOnSell.
- Suggestion: set a reasonable limit for buy and sell fees
- Owner is able to set a maximumamount for buying/selling/transfering amounts using setMaxTransactionAmount function, this maximumamount can even be 0 which disableseveryone from trading his/her tokens.

- Owner is able to set a maximumamount for holdingamount (i.e non-whitelisted wallets are not able to hold more than that amount) using setMaxWalletAmount function, this maximum amount can even be 0 which disables everyonefrom buying/transfering tokens.
- liquidityWallet is getting LP tokens at _addLiquidity function, since this wallet is an EOA If a hacker gains access to this address, they can use the lp tokens to drain the liquiditypool of funds. liquidityWallet : 0x3f953098f5eba8bcb7e2ab42f1d6c10d7b0f8760

Medium Risk Findings

Logical

- Since only 1 pool is created for BVST token and only this pool is recognized as an automatedMarketMakerPairs, then trades from other pools will not be considered a buy or sell transaction in _transfer function. this can affect amount of taxes
- Suggestion: declare a setAutomatedMarketMakerPair external onlyOwner functionto be able to set pools as market makers in the contract.

```
bool isBuyFromLp = automatedMarketMakerPairs[from];
bool isSelltoLp = automatedMarketMakerPairs[to];
```

- Any time the owner of the BSVT contract calls deactivateTrading() , the functionexecutes and updatesisTradingEnabled to false, regardlessof whether the bool value is already false.
- Addition,_tradingPausedTimestamp also gets updated to the current timestamp. This can pose problems if the functionis called when isTradingEnabled is already false, since the timestamp will change, and several other lines of logic in the contractrely on this timestamp.

Low Risk Findings

Logical

- No dead address validation at setBUSDAddress and setBLVTAddress. suggestion: check if input address is dead address or not
- Setting BUSD address to dead address or an address that is not an ERC-20 token or doesn't have pool on pancakeswap can revert some transaction due to an error in swapAndLiquify function.
- Error message is not matching the require statement

require(amount <= maxTxAmount, "BlocVest: Buy amount exceeds the maxTxBuyAmount.")

- suggestion:

change error message to
"amount exceeds the maxTxAmount"

Suggestions

- you can use up to 3 indexed arguments in events, make sure to use this 3 in your events
- Solidity versions >=0.8.0 include checked arithmetic operations and underflow/overflow (for signedandunsigned integers) by default, making the usage of multiple SafeMath libraries redundant. The underflow/overflow check is performedat every step in a calculation.

Gas Optimizations

- BLVT address never used inside the contract, its not clear what is usage of this variable.
- define _blockedTimeLimit as constant
- Lots of reading from storage for isTradingEnabled state vairable in _transfer, after Istanbul
- hardfork price of some opcodes increased, including SLOAD opcode which changed from 400 to
- 800 gas, by declaring a memory vairable and assigning isTradingEnabled to it you can reduce gas
- usage from 800 to only 3 per using the variable.

at this block of code at _transfer

```
if (_isInLaunch && (currentTimestamp - _launchStartTimestamp) <= 300) {
if (to != owner() && isBuyFromLp && (currentTimestamp -
_buyTimesInLaunch[to]) > 60) {
_buyTimesInLaunch[to] = currentTimestamp;
}
}
```

- its possible to lower gas usage by moving:
- _buyTimesInLaunch[to] = currentTimestamp;
- to:
- if (_isInLaunch && (currentTimestamp - _launchStartTimestamp) <= 300 &&

```
isBuyFromLp) {
require((currentTimestamp - _buyTimesInLaunch[to]) > 60, "BlocVest:
Cannot buy more than once per min in first 5min of launch");
}
```

- under require statement, since they are checking same conditions
- at _adjustTaxes function, you can declare memory vairables for this storage variables:

_liquidityFee = 0;
_devFee = 0;
_buyBackFee = 0;
_shareholderFee = 0;
can be:
uint256 m_liquidityFee;
uint256 m_devFee;
uint256 m_buyBackFee;
uint256 m_shareholderFee;
then you can do all the operations, and at the end assign _totalFee to sum of all this memory vairables,
and also pass this variables to FeesApplied event and also assign each one to its storage alias:
_liquidityFee = m_liquidityFee;
_devFee = m_devFee;
_buyBackFee = m_buyBackFee;
_shareholderFee =m_shareholderFee;
this will lower gas usage on buy/sells/transfers a lot

ABOUT EXPELEE

Expelee is a product-based aspirational Web3 Start-up. Coping up with numerous solutions for blockchain Security and constructing a Web3 Ecosystem from Deal making platform to developer hosting open platform, while also developing our own commercial and sustainable blockchain.

www.expelee.com

expeleeofficial

expelee

Expelee

expelee

expelee_official

expelee-co

expelee

Building the Futuristic **Blockchain Ecosystem**

DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always Do your own research and protect yourselves from being scammed. The Expelee team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools.

Under no circumstances did Expelee receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams.

This document should not be presented as a reason to buy or not buy any particular token. The Expelee team disclaims any liability for the resulting losses.