



Smart Contract Security Audit Report

Belka

January 2023

Security Status



www.hacksafe.io



Audit Details



Audited project

Belka



Deployer address

0x66629195f6a31d24a7f1c732865dda612ee1db71



Client contacts

Belka Team



Blockchain

Binance smart chain



Website

not provided

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at 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.

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 TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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 alone. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by Belka to perform an audit of smart contracts:

- <https://bscscan.com/token/0x1E7681E86027D8556B0c7eFC7bA213B8940e2788#code>

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be understood to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

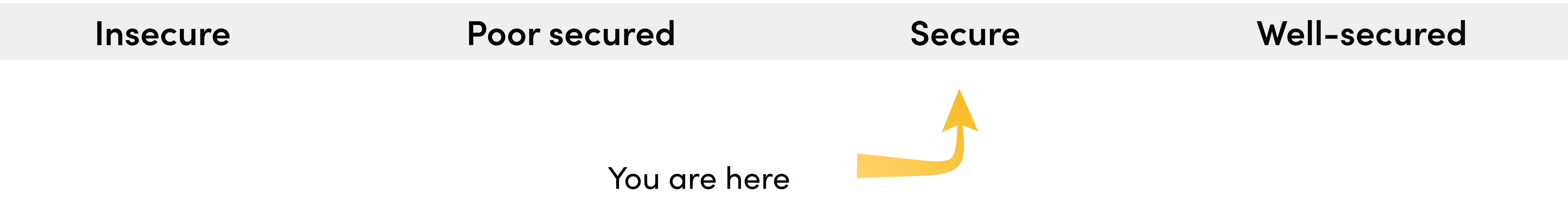
Contract Details

Token contract details for 02.01.2023

Token Type	: MEME
Contract name	: Belka
Contract address	: 0x1E7681E86027D8556B0c7eFC7bA213B8940e2788
Total supply	: 1,000,000,000
Token ticker	: BLK
Decimals	: 9
Token Holders	: 1,131
Transactions count	: 5,498
Compiler version	: v0.8.7+commit.e28d00a7
Contract deployer address	: 0x66629195f6a31d24a7f1c732865dda612ee1db71
Owner address	: 0x66629195f6a31d24a7f1c732865dda612ee1db71

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Secure”**. This token contract does contain owner control as ownership has not been renounced, which do not make it fully decentralized.



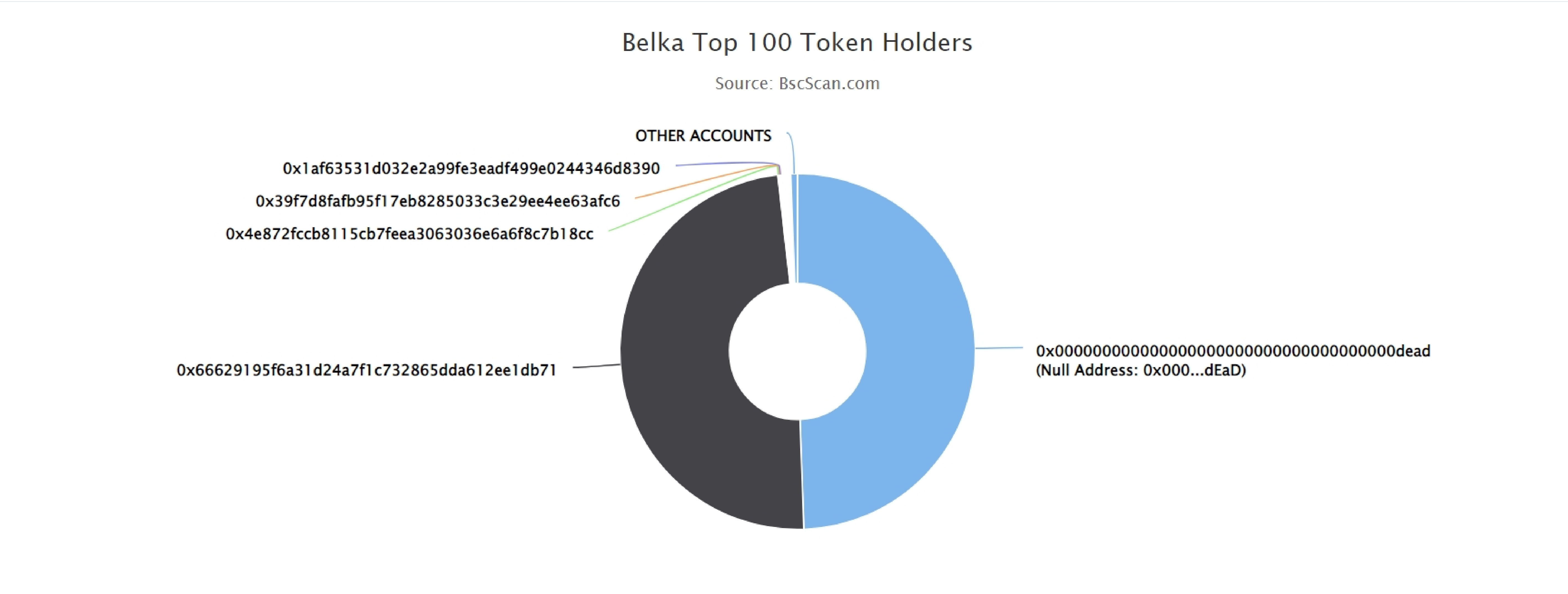
We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 0 medium and 0 low.



Belka Token Distribution

 The top 100 holders collectively own 99.39% (993,886,269.37 Tokens) of Belka

 Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 1,131

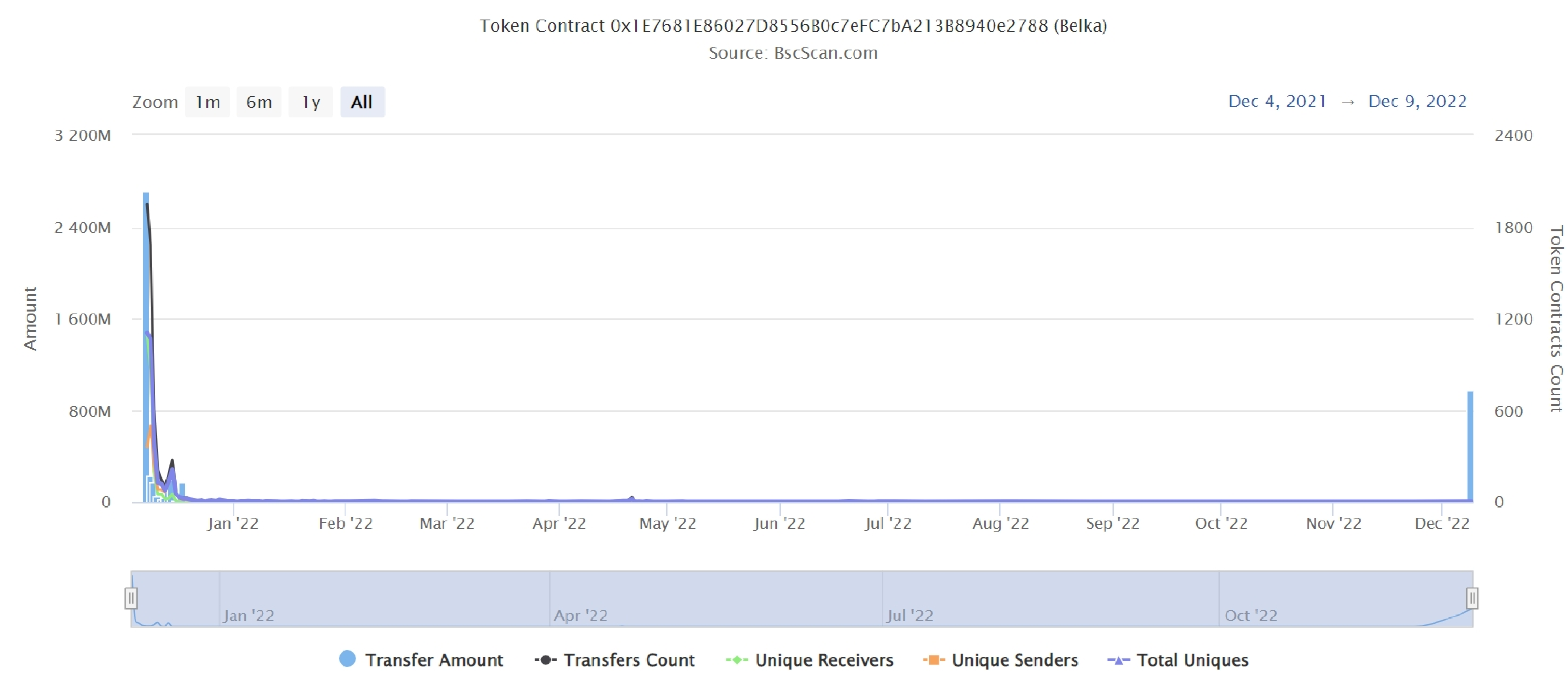


Belka Top 20Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x000...dEaD	494,394,903.326585226	49.4395%
2	0x66629195f6a31d24a7f1c732865dda612ee1db71	487,886,297.835796592	48.7886%
3	0x4e872fccb8115cb7feea3063036e6a6f8c7b18cc	1,063,392.038463771	0.1063%
4	0x39f7d8fafb95f17eb8285033c3e29ee4ee63afc6	880,454.356846356	0.0880%
5	 0x1af63531d032e2a99fe3eadf499e0244346d8390	742,592.110738538	0.0743%
6	0x05221b6a0775011f6c87ce5646bf33f57c1e08ef	652,034.393022892	0.0652%
7	0x7013abc1c401bc2a99be25e99c655329adf3a5c5	637,878.249757885	0.0638%
8	0x7537ecdc8338e819e0d4cff7813a64e94384c5ea	595,033.535127019	0.0595%
9	0xa90f0ebbc788ddedfd624b9782a35f27732f20a3	527,014.703710233	0.0527%
10	Binance: Hot Wallet 13	500,475.063244386	0.0500%
11	0x3034e22f293b73c1d441495f5c1ff86469b456b2	466,587.615542219	0.0467%
12	0x52e25a28d6134a06a7ca6ee0306829dbf54b3727	340,010.419956041	0.0340%
13	0xf0b864feddd1f7bb9c8736be74172695dbb9aa13	277,706.377708724	0.0278%
14	0xc9fd50a879387b93fee758c7ee93d347bcd60f62	170,528.048699165	0.0171%
15	0xa61a2f54267ae1016186307f199256e56f0eadc	164,301.657688436	0.0164%
16	0x4393142332c4f7dc0c95d74fa2e8f5868dfaa863	156,001.267470935	0.0156%
17	0x777758ce5553fe8a4b49fa83df82f83b0fa3b5d2	144,501.105433456	0.0145%
18	0x499ac5ffe6a1a33480cfb6ca76a578a133faa09f	142,940.315928397	0.0143%
19	 0x1e7681e86027d8556b0c7efc7ba213b8940e2788	131,309.166625062	0.0131%
20	0xf7a4c1197103e9fca4625e36578be6bcf381e031	117,140.330827317	0.0117%

Belka Token Distribution

Belka Contract Overview



Contract functions details

+Context

-[Int] _msgSender

+ [Int] IERC20

-[Ext] totalSupply

-[Ext] balanceOf

-[Ext] transfer #

-[Ext] allowance

-[Ext] approve #

-[Ext] transferFrom #

+ [Lib] SafeMath

-[Int] add

-[Int] sub

-[Int] sub

-[Int] mul

-[Int] div

-[Int] div

+Ownable (Context)

-[Int] <constructor> #

-[Pub] owner

-[Pub] renounceOwnership #

- modifiers: onlyOwner

-[Pub] transferOwnership #

- modifiers: onlyOwner

+ [Int] IUniswapV2Factory

-[Ext] createPair #

+ [Int] IUniswapV2Router02

-[Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

-[Ext] factory

-[Ext] WETH

-[Ext] addLiquidityETH (\$)

+ Belka (Context, IERC20, Ownable)

-[Pub] <Constructor> #

-[Pub] name

-[Pub] symbol

-[Pub] decimals

-[Pub] totalSupply

Contract functions details

- [Pub] balanceOf
- [Pub] transfer #
- [Pub] allowance
- [Pub] approve #
- [Pub] transferFrom #
- [Ext] setIsExcludedFromFee #
 - modifiers: onlyOwner
- [Pub] isExcludedFromFee
- [Ext] setProjectWallet #
 - modifiers: onlyOwner
- [Ext] setBuybackWallet #
 - modifiers: onlyOwner
- [Ext] setMarketingWallet #
 - modifiers: onlyOwner
- [Ext] setOperationsWallet #
 - modifiers: onlyOwner
- [Pvt] tokenFromReflection
- [Pvt] removeAllFee #
- [Pvt] restoreAllFee #
- [Ext] setRemoveAllFee #
 - modifiers: onlyOwner
- [Ext] setRestoreAllFee #
 - modifiers: onlyOwner
- [Pvt] _approve #
- [Pvt] _transfer #
- [Pvt] swapTokensForEth #
 - modifiers: lockTheSwap
- [Pvt] _tokenTransfer #
- [Pvt] _transferStandard #
- [Pvt] _reflectFee #
- [Pvt] _calculateReflectTransfer #
- [Ext] <Fallback> (\$)
- [Pvt] _getRate
- [Pvt] _getCurrentSupply
- [Ext] setMaxTxPercent #
 - modifiers: onlyOwner
- [Ext] setBuyFee #
 - modifiers: onlyOwner

Contract functions details

-[Ext] setSellFee #

- modifiers: onlyOwner

-[Ext] withdrawResidualBnb #

- modifiers: onlyOwner

-[Ext] withdrawResidualErc20 #

- modifiers: onlyOwner

(\$) = payable function

= non-constant function

Issues Checking Status

No.	Title	Status
1.	Compiler error	Passed
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

✔ Critical Severity Issues

No critical severity issue found.

✔ High Severity Issues

No high severity issue found.

✔ Medium Severity Issues

No medium severity issue found.

✔ Low Severity Issues

No low severity issue found.

Notes:

Operations wallet get the rest of swapped amount.

Centralization

Owner privileges :

- Belka Contract:
 - Owner Can Change Project, Buyback, Marketing And Operations Wallet.
 - Owner Can Remove And Restore Fees.
 - Owner Can Change The Maximum Transaction Amount.
 - Owner Can Change Buy And Sell Fees.
 - Owner Can Withdraw Contract Tokens And Bnbs.

This smart contract has some functions which can be executed by the admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble, as smart contract ownership has not been renounced.

- renounceownership
- transferownership
- setisexcludedfromfee
- setprojectwallet
- setbuybackwallet
- setmarketingwallet
- setoperationswallet
- setremoveallfee
- setrestoreallfee
- setmaxtxpercent
- setbuyfee
- setsellfee
- withdrawresidualbnb
- withdrawresidualerc20

Conclusion

Smart contract contains no low severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.