Bepe Grok Token

RugfreeCoins Verified on January 29th, 2024

Overview

✅ No mint function found, the owner cannot mint tokens after initial deployment.

✅ The owner can't set a max transaction limit

✅ The owner can't pause trading once it's enabled

❌ The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

✅ The owner can't change fees over 10%.

✅ The owner can't blacklist wallets.

✅ The owner can't set a max wallet limit

✅ The owner can't claim the contract's balance of its own token.

❗ HIGH SEVERITY ISSUES

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function enableTrading() external onlyOwner {
    require(!isTradingEnabled, "Trading already enabled");
    isTradingEnabled = true;
    emit _enableTrading();
}
```

Contents

Audit details

Audited project

Bepe Grok Token

Contract Address

0x482f7653DefC8dE3f7e649CdE1339184A47ff5b7

Client contact

Bepe Grok Token Team

Blockchain

Binance Smart chain

Project website

https://www.bepegrok.com

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 disclaimer below – **please make sure to read it in full.**

Background

RugfreeCoins was commissioned by the **Bepe Grok Token Team** to perform an audit of the smart contract.

https://bscscan.com/token/0x482f7653defc8de3f7e649cde1339184a47ff5b7

This audit focuses on verifying that the smart contract is secure, resilient, and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, and long-term sustainability, and as a guide to improving the smart contract's security posture by remediating the identified issues.

Tokenomics

▲ 7% tax when buying & selling

5% of trade goes to the marketing wallet in BNB
1% of trade goes to the dev wallet in BNB
1% of trade goes to the liquidity pool

Target market and the concept

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's ready to earn a passive income by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part in the Bepe Grok token ecosystem.
- Anyone who's interested in taking part in the future plans of Bepe Grok Token.
- Anyone who's interested in making financial transactions with any other party Bepe Grok Token as the currency.

Potential to grow with score points

⚡ Project efficiency	**8** / 10
🌟 Project uniqueness	**8** / 10
📊 Information quality	**8** / 10
💧 Service quality	**8** / 10
💻 System quality	**8** / 10
🌍 Impact on the community	**8** / 10
💼 Impact on the business	**9** / 10
🔮 Preparing for the future	**8** / 10
🔒 Smart contract security	**8** / 10
🛠️ Smart contract functionality assessment	**10** / 10
🏆 **Total Score**	**8.3** / **10**

Contract details

Token contract details for 29th of January 2024

Contract name	**Bepe Grok**
Contract address	**0x482f7653DefC8dE3f7e649CdE1339184A47ff5b7**
Token supply	**420,000,000,000,000,000**
Token ticker	**BepeGrok**
Decimals	**9**
Token holders	**3**
Transaction count	**6**
Contract deployer address	**0x3Bb00D621F5F84877e40FdaF085F0f06330ffa01**
Contract's current owner address	**0x3Bb00D621F5F84877e40FdaF085F0f06330ffa01**

Contract code function details

Nº	Category	Item	Result
1	Coding conventions	ERC20 Token standards	PASS
		Compile errors	PASS
		Compiler version security	PASS
		Visibility specifiers	PASS
		Gas consumption	PASS
		SafeMath features	PASS
		Fallback usage	PASS
		tx.origin usage	PASS
		Deprecated items	PASS
		Redundant code	PASS
		Overriding variables	PASS
2	Function call audit	Authorization of function call	PASS
		Low level function (call/delegate call) security	PASS
		Returned value security	PASS
		Self destruct function security	PASS
3	Business security & centralisation	Access control of owners	HIGH
		Business logics	PASS
		Business implementation	PASS
4	Integer overflow/underflow		PASS
5	Reentrancy		PASS
6	Exceptional reachable state		PASS
7	Transaction ordering dependence		PASS
8	Block properties dependence		PASS
9	Pseudo random number generator (PRNG)		PASS
10	DoS (Denial of Service)		PASS
11	Token vesting implementation		PASS
12	Fake deposit		PASS
13	Event security		PASS

Contract description table

The below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions, and implementations with their visibility and mutability.

Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
Context	**Implementation**			
L		Public ❗	🛑	NO ❗
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
Ownable	**Implementation**	Context		
L		Public ❗	🛑	NO ❗
L	owner	Public ❗		NO ❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
L	_setOwner	Private 🔐	🛑	
IFactoryV2	**Interface**			
L	getPair	External ❗		NO ❗
L	createPair	External ❗	🛑	NO ❗
IV2Pair	**Interface**			
L	factory	External ❗		NO ❗

L	getReserves	External ❗			NO ❗
L	sync	External ❗	🛑		NO ❗
IRouter01	**Interface**				
L	factory	External ❗			NO ❗
L	WETH	External ❗			NO ❗
L	addLiquidityETH	External ❗	💵		NO ❗
L	addLiquidity	External ❗	🛑		NO ❗
L	swapExactETHForTokens	External ❗	💵		NO ❗
L	getAmountsOut	External ❗			NO ❗
L	getAmountsIn	External ❗			NO ❗
IRouter02	**Interface**	**IRouter01**			
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑		NO ❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵		NO ❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑		NO ❗
L	swapExactTokensForTokens	External ❗	🛑		NO ❗
IERC20	**Interface**				
L	totalSupply	External ❗			NO ❗
L	decimals	External ❗			NO ❗

BepeGrok	Implementation	Context, Ownable, IERC20		
L	symbol	External ❗		NO ❗
L	name	External ❗		NO ❗
L	getOwner	External ❗		NO ❗
L	balanceOf	External ❗		NO ❗
L	transfer	External ❗	🛑	NO ❗
L	allowance	External ❗		NO ❗
L	approve	External ❗	🛑	NO ❗
L	transferFrom	External ❗	🛑	NO ❗
BepeGrok	**Implementation**	**Context, Ownable, IERC20**		
L	totalSupply	External ❗		NO ❗
L	decimals	External ❗		NO ❗
L	symbol	External ❗		NO ❗
L	name	External ❗		NO ❗
L	getOwner	External ❗		NO ❗
L	allowance	External ❗		NO ❗
L	balanceOf	Public ❗		NO ❗
L		Public ❗	🛑	NO ❗
L		External ❗	💵	NO ❗
L	transfer	Public ❗	🛑	NO ❗

L	approve	External ❗	🛑	NO ❗
L	_approve	Internal 🔒	🛑	
L	transferFrom	External ❗	🛑	NO ❗
L	isNoFeeWallet	External ❗		NO ❗
L	setNoFeeWallet	Public ❗	🛑	onlyOwner
L	isLimitedAddress	Internal 🔒		
L	is_buy	Internal 🔒		
L	is_sell	Internal 🔒		
L	canSwap	Internal 🔒		
L	changeLpPair	External ❗	🛑	onlyOwner
L	toggleCanSwapFees	External ❗	🛑	onlyOwner
L	_transfer	Internal 🔒	🛑	
L	changeWallets	External ❗	🛑	onlyOwner
L	takeTaxes	Internal 🔒	🛑	
L	swapAndLiquify	Internal 🔒	🛑	inSwapFlag
L	internalSwap	Internal 🔒	🛑	inSwapFlag
L	setPresaleAddress	External ❗	🛑	onlyOwner
L	enableTrading	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable
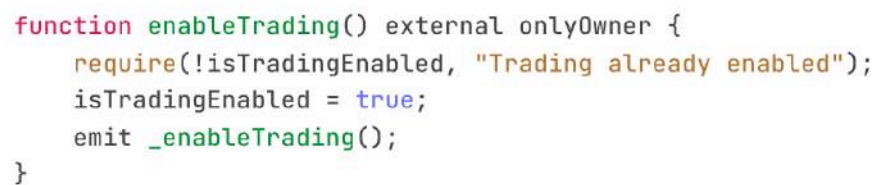
Inheritance Hierarchy
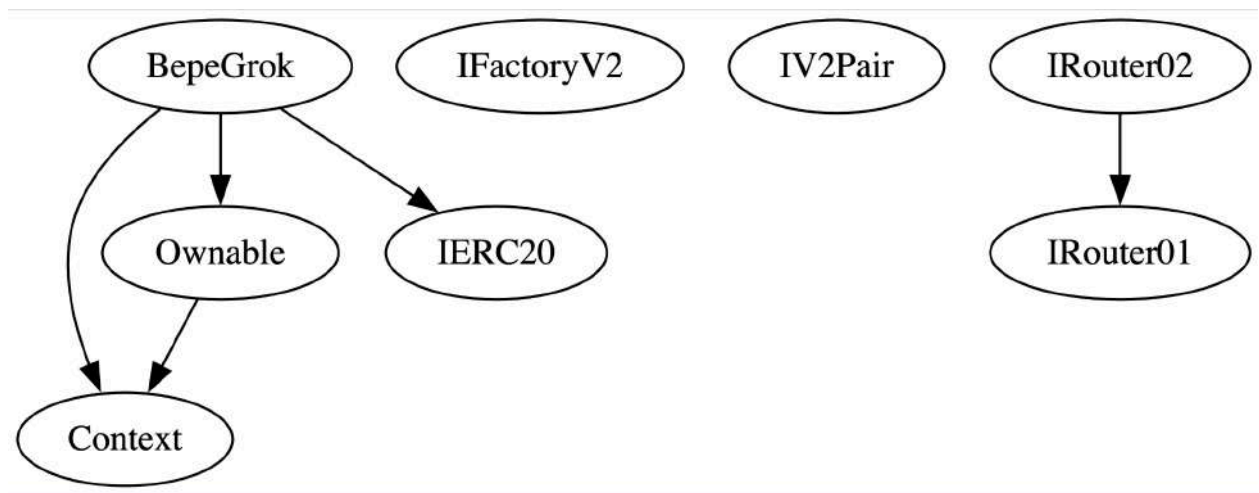
Security issue checking status

❖ High severity issues

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function enableTrading() external onlyOwner {
    require(!isTradingEnabled, "Trading already enabled");
    isTradingEnabled = true;
    emit _enableTrading();
}
```

❖ Medium severity issues

No medium severity issues found

❖ Low severity issues

No high severity issues found

Owner privileges

❖ Owner can include/exclude wallets from fees

```solidity
function setNoFeeWallet(address account, bool enabled) public onlyOwner {
    _noFee[account] = enabled;
}
```

❖ Owner can add/remove new lp pairs

```solidity
function changeLpPair(address newPair) external onlyOwner {
    isLpPair[newPair] = true;
    emit _changePair(newPair);
}
```

❖ Owner can enable/disable swapping

```solidity
function toggleCanSwapFees(bool yesno) external onlyOwner {
    require(canSwapFees != yesno,"Bool is the same");
    canSwapFees = yesno;
    emit _toggleCanSwapFees(yesno);
}
```

❖ Owner can change marketing and dev wallet address

```solidity
function changeWallets(address newBuy, address newDev) external onlyOwner {
    require(newBuy != address(0),"Freddy: Address Zero");
    require(newDev != address(0),"Freddy: Address Zero");
    marketingAddress = payable(newBuy);
    devWallet = payable(newDev);
    emit _changeWallets(newBuy, newDev);
}
```

❖ Owner can include/exclude pre sale address

```solidity
function setPresaleAddress(address presale, bool yesno) external onlyOwner {
    require(isPresaleAddress[presale] != yesno,"Same bool");
    isPresaleAddress[presale] = yesno;
    _noFee[presale] = yesno;
    liquidityAdd[presale] = yesno;
    emit _setPresaleAddress(presale, yesno);
}
```

❖ Owner can enable trading,once enabled can not disable again

```solidity
function enableTrading() external onlyOwner {
    require(!isTradingEnabled, "Trading already enabled");
    isTradingEnabled = true;
    emit _enableTrading();
}
```

Audit conclusion

RugFreeCoins team has performed in-depth testing, line-by-line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status:	PASS ⌄
Smart contract security Status:	HIGH ISSUES ⌄
Number of risk issues:	01
Solidity code functional issue level:	PASS ⌄
Number of owner privileges:	06
Centralization risk correlated to the active owner:	HIGH ⌄
Smart contract active ownership:	ACTIVE ⌄