Error 404 Token

RugfreeCoins Verified on February 19th, 2024

Overview

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

❌ The owner can set the buy and sell without any limit

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

✅ The owner can't change fees.

✅ 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.

Contents

Audit details

Audited project

Error 404 Token

Contract Address

0xc9E46c11c0810Ae41Aa2693be9323dDBe8AA0dBb

Client contact

Error 404 Token Team

Blockchain

Ethereum

Project website

https://error404.meme/

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 **Error 404 Token Team** to perform an audit of the smart contract.

https://etherscan.io/token/0xc9e46c11c0810ae41aa2693be9323ddbe8aa0dbb

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

▲ **0% tax when buying & selling**

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 Error 404 token ecosystem.
- Anyone who's interested in taking part in the future plans of Error 404 Token.
- Anyone who's interested in making financial transactions with any other party Error 404 Token as the currency.

Potential to grow with score points

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

Contract details

Token contract details for 19th of February 2024

Contract name	**Error 404**
Contract address	**0xc9E46c11c0810Ae41Aa2693be9323dDBe8AA0dBb**
Token supply	**10000**
Token ticker	**ERROR404**
Decimals	**18**
Token holders	**0**
Transaction count	**0**
Contract deployer address	**0x3aFE0636a4716df78bAFe6866f033d95763468c8**
Contract's current owner address	**0x3aFE0636a4716df78bAFe6866f033d95763468c8**

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
ReentrancyGuard	**Implementation**			
L		Public ❗	🛑	NO ❗
L	_nonReentrantBefore	Private 🔐	🛑	
L	_nonReentrantAfter	Private 🔐	🛑	
L	_reentrancyGuardEntered	Internal 🔒		
SignedMath	**Library**			
L	max	Internal 🔒		
L	min	Internal 🔒		
L	average	Internal 🔒		
L	abs	Internal 🔒		
Math	**Library**			
L	tryAdd	Internal 🔒		
L	trySub	Internal 🔒		
L	tryMul	Internal 🔒		
L	tryDiv	Internal 🔒		
L	tryMod	Internal 🔒		
L	max	Internal 🔒		
L	min	Internal 🔒		
L	average	Internal 🔒		

L	ceilDiv	Internal 🔒		
L	mulDiv	Internal 🔒		
L	mulDiv	Internal 🔒		
L	sqrt	Internal 🔒		
L	sqrt	Internal 🔒		
L	log2	Internal 🔒		
L	log2	Internal 🔒		
L	log10	Internal 🔒		
L	log10	Internal 🔒		
L	log256	Internal 🔒		
L	log256	Internal 🔒		
L	unsignedRoundsUp	Internal 🔒		
Strings	**Library**			
L	toString	Internal 🔒		
L	toStringSigned	Internal 🔒		
L	toHexString	Internal 🔒		
L	toHexString	Internal 🔒		
L	toHexString	Internal 🔒		
L	equal	Internal 🔒		
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
L	_contextSuffixLength	Internal 🔒		
Pausable	**Implementation**	**Context**		
L		Public ❗	🛑	NO ❗
L	paused	Public ❗		NO ❗

L	_requireNotPaused	Internal 🔒		
L	_requirePaused	Internal 🔒		
L	_pause	Internal 🔒	🛑	whenNotPaused
L	_unpause	Internal 🔒	🛑	whenPaused
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO ❗
L	owner	Public ❗		NO ❗
L	_checkOwner	Internal 🔒		
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
L	_transferOwnership	Internal 🔒	🛑	
Ownable2Step	**Implementation**	**Ownable**		
L	pendingOwner	Public ❗		NO ❗
L	transferOwnership	Public ❗	🛑	onlyOwner
L	_transferOwnership	Internal 🔒	🛑	
L	acceptOwnership	Public ❗	🛑	NO ❗
ERC721Receiver	**Implementation**			
L	onERC721Received	External ❗	🛑	NO ❗
ERC404	**Implementation**	**Ownable 2Step**		
L		Public ❗	🛑	Ownable
L	setWhitelist	Public ❗	🛑	onlyOwner
L	ownerOf	Public ❗		NO ❗
L	tokenURI	Public ❗		NO ❗
L	approve	Public ❗	🛑	NO ❗
L	setApprovalForAll	Public ❗	🛑	NO ❗

L	transferFrom	Public ❗	🛑	NO ❗
L	transfer	Public ❗	🛑	NO ❗
L	safeTransferFrom	Public ❗	🛑	NO ❗
L	safeTransferFrom	Public ❗	🛑	NO ❗
L	_transfer	Internal 🔒	🛑	
L	_getUnit	Internal 🔒		
L	_mint	Internal 🔒	🛑	
L	_burn	Internal 🔒	🛑	
L	_setNameSymbol	Internal 🔒	🛑	
Error404	**Implementation**	**ERC404, Pausable, Reentrancy Guard**		
L		Public ❗	🛑	ERC404
L	setLimit	Public ❗	🛑	onlyOwner
L	_mint	Internal 🔒	🛑	whenNotPaused
L	startApplyingLimit	External ❗	🛑	onlyOwner
L	stopApplyingLimit	External ❗	🛑	onlyOwner
L	_transfer	Internal 🔒	🛑	whenNotPaused
L	setTokenURI	Public ❗	🛑	onlyOwner
L	setNameSymbol	Public ❗	🛑	onlyOwner
L	tokenURI	Public ❗		NO ❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Inheritance Hierarchy
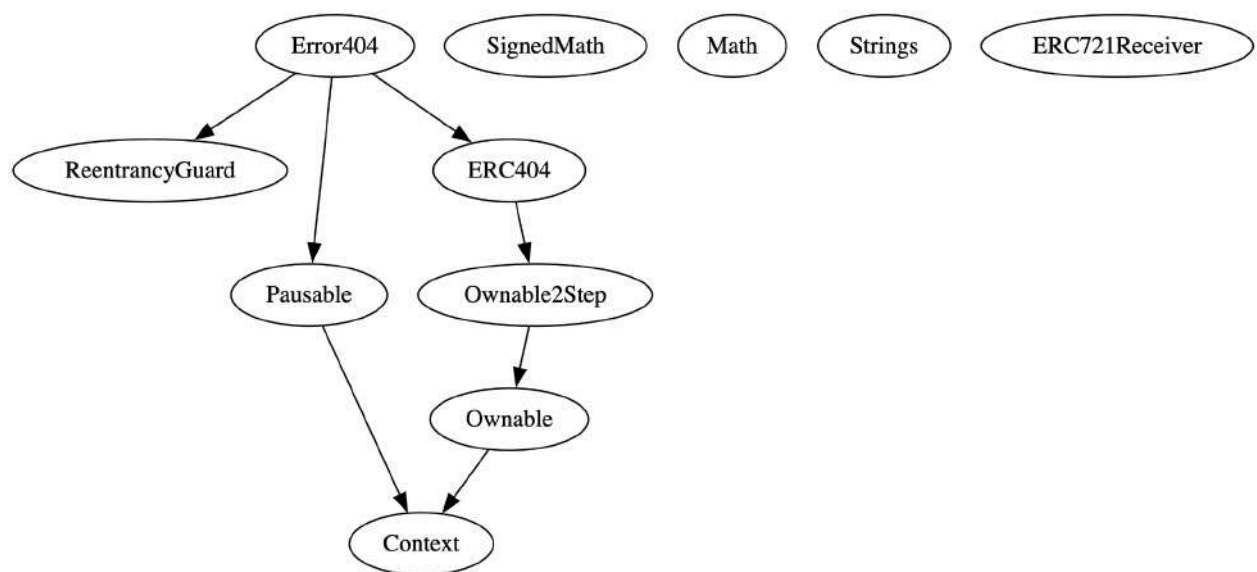
Security issue checking status

❖ High severity issues

Owner can set buy and sell limits without any limit

```
function setLimit(uint256 _buylimit, uint256 _selllimit) public onlyOwner{
    buyLimit = _buylimit;
    sellLimit = _selllimit;
}
```
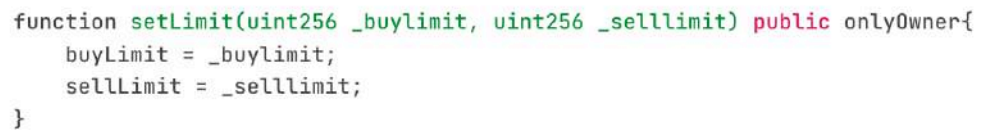
❖ Medium severity issues

No medium severity issues found

❖ Low severity issues

No high severity issues found

Owner privileges

❖ The owner can enable disable buy and sell limits

```
function startApplyingLimit() external onlyOwner{
    applyTxLimit = true;
}

function stopApplyingLimit() external onlyOwner{
    applyTxLimit = false;
}
```

❖ The owner can change the token URI

```
function setTokenURI(string memory _tokenURI) public onlyOwner {
    baseTokenURI = _tokenURI;
}
```

❖ The owner can change buy and sell limits

```
function setLimit(uint256 _buylimit, uint256 _selllimit) public onlyOwner{
    buyLimit = _buylimit;
    sellLimit = _selllimit;
}
```

❖ The owner can whitelist users from the contract (Buy and sell limits will not apply for whitelisted users)

```solidity
function setWhitelist(address target, bool state) public onlyOwner {
    whitelist[target] = state;
}
```

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:	04
Centralization risk correlated to the active owner:	HIGH ▾
Smart contract active ownership:	ACTIVE ▾