SMART CONTRACT SECURITY AUDIT
of

Token Overview

0xSafe received the application for a smart contract security audit of **EMERALD CAT** on March 9, 2022.

Details

Client: EmeraldCat [$ECAT]

Blockchain: Binance Smart Chain

Contract: 0xA7BEd2a446c9e59D3502301EA0BB7283Ac84076A

Compiler: v0.7.4+commit.3f05b770

Optimization: Yes with 200 runs

Website: https://emerald-cat.com/

Table of Contents

Methodology

Audit Details

This comprehensive audit report provides an overview of the **EmeraldCat** token smart contract. 0xSafe utilizes a combination of static, automated, and manual analysis tools to check for any potential vulnerabilities or hacks in the system.

Code Quality

This includes a full review of the smart contract code. The prime areas of focus are:

- Accuracy
- Exploits
- Functionality
- Readability
- Security
- Vulnerabilities

Scope of work

EmeraldCat's team provided us with the files that need to be tested (BSCscan, Etherscan, Github, etc.). The focus of the security audit is the main token smart contract.

Tools

Ganache, Mithril, MythX, Open Zeppelin Code Analyzer, Proprietary tests, Remix IDE, Solidity Compiler, SWC Registry.

Risk Classification

!Critical	This signifies vulnerabilities with the smart contract's functionality or performance. Issues should be resolved immediately.
!Medium	This signifies vulnerabilities that can potentially cause problems and should eventually be fixed.
!Minor	Minor vulnerabilities may or may not impact smart contract functionality.
!Informational	This is there to offer suggestions for improvement

Audit Findings

Critical Issues

-no critical issues found-

Medium Issues

-no medium issues found-

Minor Issues

Issue	Type	Line #(s)	Description
#1	A floating pragma is set.	27	Current pragma directive is: "^v0.7.4"
#2	State variable visibility is not set.	188, 196-198, 200-202, 215, 217, 354-356, 362, 367-268, 374-377, 394-395, 403, 411	It is best practice to set the visibility of state variables explicitly (possible visibility settings are internal, public, and private)

SWC Attacks

SWC ID	Description	Status
SWC-100	Function Default Visibility	PASSED
SWC-101	Integer Overflow and Underflow	PASSED
SWC-102	Outdated Compiler Version	PASSED
SWC-103	Floating Pragma	MINOR
SWC-104	Unchecked Call Return Value	PASSED
SWC-105	Unprotected Ether Withdrawal	PASSED
SWC-106	Unprotected SELFDESTRUCT Instruction	PASSED
SWC-107	Reentrancy	PASSED
SWC-108	State Variable Default Visibility	MINOR
SWC-109	Uninitialized Storage Pointer	PASSED
SWC-110	Assert Violation	PASSED
SWC-111	Use of Deprecated Solidity Functions	PASSED
SWC-112	Delegatecall to Untrusted Callee	PASSED
SWC-113	DoS with Failed Call	PASSED
SWC-114	Transaction Order Dependence	PASSED
SWC-115	Authorization through tx.origin	PASSED
SWC-116	Block values as a proxy for time	PASSED
SWC-117	Signature Malleability	PASSED
SWC-118	Incorrect Constructor Name	PASSED
SWC-119	Shadowing State Variables	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	PASSED

SWC-122	Lack of Proper Signature Verification	PASSED
SWC-123	Requirement Violation	PASSED
SWC-124	Write to Arbitrary Storage Location	PASSED
SWC-125	Incorrect Inheritance Order	PASSED
SWC-126	Insufficient Gas Griefing	PASSED
SWC-127	Arbitrary Jump with Function Type Variable	PASSED
SWC-128	DoS With Block Gas Limit	PASSED
SWC-129	Typographical Error	PASSED
SWC-130	Right-To-Left-Override control character (U+202E)	PASSED
SWC-131	Presence of unused variables	PASSED
SWC-132	Unexpected Ether balance	PASSED
SWC-133	Hash Collisions With Multiple Variable Length Arguments	PASSED
SWC-134	Message call with hardcoded gas amount	PASSED
SWC-135	Code With No Effects	PASSED
SWC-136	Unencrypted Private Data On-Chain	PASSED

Important Notes

Authorized addresses can exclude from fees

```
function setIsFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}
```

Authorized addresses can exclude from transaction limits

```
function setIsTxLimitExempt(address holder, bool exempt) external authorized {
    isTxLimitExempt[holder] = exempt;
}
```

Authorized addresses can exclude from timelock

```
function setIsTimelockExempt(address holder, bool exempt) external authorized {
    isTimelockExempt[holder] = exempt;
}
```

Authorized addresses can exclude from rewards

```
function setIsDividendExempt(address holder, bool exempt) external authorized {
    require(holder != address(this) && holder != pair);
    isDividendExempt[holder] = exempt;
    if(exempt){
        distributor.setShare(holder, 0);
    }else{
        distributor.setShare(holder, _balances[holder]);
    }
}
```

Owner can blacklist addresses

```
function manage_blacklist(address[] calldata addresses, bool status) public onlyOwner {
    for (uint256 i; i < addresses.length; ++i) {
        isBlacklisted[addresses[i]] = status;
    }
}
```

Owner can set the max wallet balance to 0%

```
function setMaxWalletPercent_base1000(uint256 maxWallPercent_base1000) external onlyOwner() {
    _maxWalletToken = (_totalSupply * maxWallPercent_base1000 ) / 1000;
}
```

Owner can set the max transaction to 0%

```
function setMaxTxPercent_base1000(uint256 maxTXPercentage_base1000) external onlyOwner() {
        _maxTxAmount = (_totalSupply * maxTXPercentage_base1000 ) / 1000;
}
```

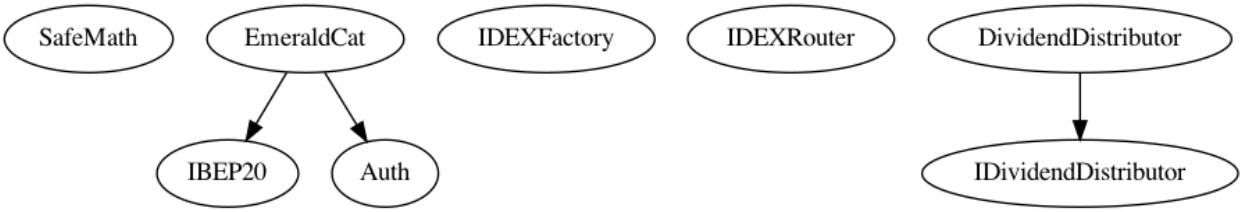
Owner can pause trading

```
function tradingStatus(bool _status) public onlyOwner {
        tradingOpen = _status;
}
```

Good Practices

- Owner cannot mint new tokens after initial deployment
- Owner cannot set fees more than 50%
- The smart contract utilizes “SafeMath” to prevent overflows

Inheritance Tree



Contract Inspection

Below is a visual description report comprising of information about the system's files, contracts, and their functions.

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable
🔒	Internal function
NO ❗	Function has no modifier

Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library				
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
IBEP20	Interface				
└	totalSupply	External ❗		NO ❗
└	decimals	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	name	External ❗		NO ❗

└	getOwner	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
Auth	Implementation			
└	<Constructor>	Public ❗	🛑	NO ❗
└	authorize	Public ❗	🛑	onlyOwner
└	unauthorize	Public ❗	🛑	onlyOwner
└	isOwner	Public ❗		NO ❗
└	isAuthorized	Public ❗		NO ❗
└	transferOwnership	Public ❗	🛑	onlyOwner
IDEXFactory	Interface			
└	createPair	External ❗	🛑	NO ❗
IDEXRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
IDividendDistributor	Interface			
└	setDistributionCriteria	External ❗	🛑	NO ❗
└	setShare	External ❗	🛑	NO ❗
└	deposit	External ❗	💵	NO ❗
└	process	External ❗	🛑	NO ❗
DividendDistributor	Implementation	IDividendDistributor		
└	<Constructor>	Public ❗	🛑	NO ❗
└	setDistributionCriteria	External ❗	🛑	onlyToken
└	setShare	External ❗	🛑	onlyToken

└	deposit	External ❗	💵	onlyToken
└	process	External ❗	🛑	onlyToken
└	shouldDistribute	Internal 🔒		
└	distributeDividend	Internal 🔒	🛑	
└	claimDividend	External ❗	🛑	NO ❗
└	getUnpaidEarnings	Public ❗		NO ❗
└	getCumulativeDividends	Internal 🔒		
└	addShareholder	Internal 🔒	🛑	
└	removeShareholder	Internal 🔒	🛑	
EmeraldCat	Implementation	IBEP20, Auth		
└	<Constructor>	Public ❗	🛑	Auth
└	<Receive Ether>	External ❗	💵	NO ❗
└	totalSupply	External ❗		NO ❗
└	decimals	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	name	External ❗		NO ❗
└	getOwner	External ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	allowance	External ❗		NO ❗
└	approve	Public ❗	🛑	NO ❗
└	approveMax	External ❗	🛑	NO ❗
└	transfer	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
└	setMaxWalletPercent_base1000	External ❗	🛑	onlyOwner
└	setMaxTxPercent_base1000	External ❗	🛑	onlyOwner
└	setTxLimit	External ❗	🛑	authorized
└	_transferFrom	Internal 🔒	🛑	
└	_basicTransfer	Internal 🔒	🛑	
└	checkTxLimit	Internal 🔒		
└	shouldTakeFee	Internal 🔒		
└	takeFee	Internal 🔒	🛑	
└	shouldSwapBack	Internal 🔒		
└	clearStuckBalance	External ❗	🛑	authorized
└	clearStuckBalance_sender	External ❗	🛑	authorized
└	set_sell_multiplier	External ❗	🛑	onlyOwner
└	tradingStatus	Public ❗	🛑	onlyOwner
└	cooldownEnabled	Public ❗	🛑	onlyOwner
└	swapBack	Internal 🔒	🛑	swapping
└	setIsDividendExempt	External ❗	🛑	authorized

└	enable_blacklist	Public ❗	🛑	onlyOwner
└	manage_blacklist	Public ❗	🛑	onlyOwner
└	setIsFeeExempt	External ❗	🛑	authorized
└	setIsTxLimitExempt	External ❗	🛑	authorized
└	setIsTimelockExempt	External ❗	🛑	authorized
└	setFees	External ❗	🛑	authorized
└	setFeeReceivers	External ❗	🛑	authorized
└	setSwapBackSettings	External ❗	🛑	authorized
└	setTargetLiquidity	External ❗	🛑	authorized
└	setDistributionCriteria	External ❗	🛑	authorized
└	setDistributorSettings	External ❗	🛑	authorized
└	getCirculatingSupply	Public ❗		|NO ❗ |
└	getLiquidityBacking	Public ❗		|NO ❗ |
└	isOverLiquified	Public ❗		|NO ❗ |
└	multiTransfer	External ❗	🛑	onlyOwner
└	multiTransfer_fixed	External ❗	🛑	onlyOwner

Audit Results

EmeraldCat does not contain any severe issues or risks. The security of the smart contract was tested by 0xSafe using static, automated, and manual analysis. The

AUDIT PASSED

Note:
Please check the disclaimer below and note the audit makes no statements or warranties on the business model, investment attractiveness, or code sustainability of this project. The security audit report is provided for the only contract mentioned in this report.

Disclaimer

0xSafe.io provides contract auditing, KYC, development, and launch services for blockchain projects. The purpose of the security audit is to analyze the on-chain smart contract source code and to provide an easy-to-understand assessment of the crypto project and the smart contract. **0xSafe.io provides information as is.**

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 0xSafe.io and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (0xSafe.io) owes no duty of care towards you or any other person, nor does 0xSafe.io 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 0xSafe.io 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, 0xSafe.io hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against 0xSafe.io, 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 the use of this report, and any reliance on this report.

This report should not be considered as an endorsement or disapproval of any project or team. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. Conduct your own due diligence and consult your financial advisor before making any investment decisions.

https://0xsafe.io