



**SAFUAUDIT**  
SMART CONTRACT AUDITING

## 3POINTS SMART CONTRACT AUDIT



March 11, 2022

# INTRODUCTION

---

<b>Client</b>	ThreePoints (TPS)
<b>Language</b>	Solidity
<b>Contract address</b>	0xF252bfF330C303a01c3Fc8A2D8EFB93C275342c5
<b>Decimals</b>	18
<b>Supply</b>	1,000,000,000
<b>Platform</b>	Binance Smart Chain
<b>Compiler</b>	v0.8.4+commit.c7e474f2
<b>Optimization</b>	Yes, with 200 runs
<b>Website</b>	<a href="http://3pointsdesign.com/">http://3pointsdesign.com/</a>
<b>Telegram</b>	<a href="https://t.me/ThreePointsOfficial">https://t.me/ThreePointsOfficial</a>
<b>Twitter</b>	<a href="https://twitter.com/threepointsmeta">https://twitter.com/threepointsmeta</a>

## Description

3Points is a community platform for automotive paint design. It provides fans with 3D models to create NFT works, and connects offline car beauty shops to paint on real cars, realizing the connection between the virtual world and reality.

# TABLE OF CONTENTS

## 01 INTRODUCTION

---

Introduction	02
Approach	04
Risk classification	05

## 02 ABSTRACT

---

Abstract	06
----------	----

## 03 VULNERABILITIES TEST

---

Vulnerabilities Test	07
----------------------	----

## 04 MANUAL ANALYSIS

---

Manual analysis	09
Contract Inspection	10
Inheritance Tree	16
Important Snippets	17
Good Practices	18

## 05 WEBSITE

---

Website Audit	19
---------------	----

## 06 CONCLUSIONS

---

Disclaimer	20
Audit Results	21
SafuScore	22
Summary	23

# Approach

---



## 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

- Remix IDE
- MythX, Mytrhl
- SWC Registry
- Open Zeppelin Code Analyzer
- Solidity Code Complier

# RISK CLASSIFICATION

---

## CRITICAL

---

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

## MEDIUM

---

Issues on this level could potentially bring problems and should eventually be fixed.

## MINOR

---

Issues on this level are minor details and warning that can remain unfixed but would be better fixed at some point in the future

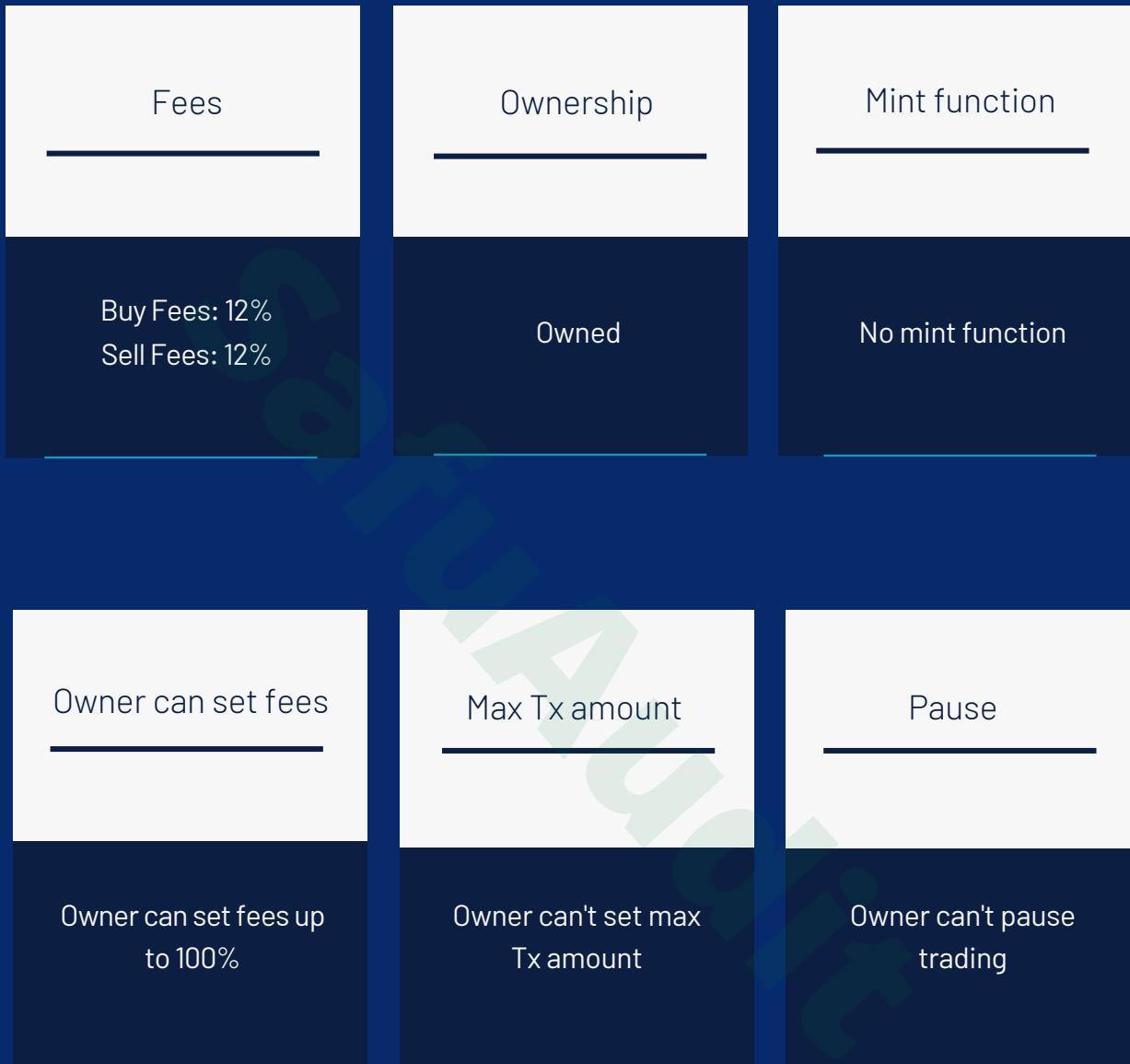
## INFORMATIONAL

---

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

# ABSTRACT

---



# Vulnerabilities Test

SWC ID	Description	
<b>SWC-100</b>	Function Default Visibility	<b>Passed</b>
<b>SWC-101</b>	Integer Overflow and Underflow	<b>Passed</b>
<b>SWC-102</b>	Outdated Compiler Version	<b>Passed</b>
<b>SWC-103</b>	FloatingPragma	<b>Minor</b>
<b>SWC-104</b>	Unchecked Call Return Value	<b>Passed</b>
<b>SWC-105</b>	Unprotected Ether Withdrawal	<b>Passed</b>
<b>SWC-106</b>	Unprotected SELF-DESTRUCT Instruction	<b>Passed</b>
<b>SWC-107</b>	Re-entrancy	<b>Passed</b>
<b>SWC-108</b>	State Variable Default Visibility	<b>Passed</b>
<b>SWC-109</b>	Uninitialized Storage Pointer	<b>Passed</b>
<b>SWC-110</b>	Assert Violation	<b>Passed</b>
<b>SWC-111</b>	Use of Deprecated Solidity Functions	<b>Passed</b>
<b>SWC-112</b>	Delegate Call to Untrusted Callee	<b>Passed</b>
<b>SWC-113</b>	DoS with Failed Call	<b>Passed</b>
<b>SWC-114</b>	Transaction Order Dependence	<b>Passed</b>
<b>SWC-115</b>	Authorization through tx.origin	<b>Minor</b>

<b>SWC-116</b>	Block values as a proxy for time	<b>Passed</b>
<b>SWC-117</b>	Signature Malleability	<b>Passed</b>
<b>SWC-118</b>	Incorrect Constructor Name	<b>Passed</b>
<b>SWC-119</b>	Shadowing State Variables	<b>Passed</b>
<b>SWC-120</b>	Weak Sources of Randomness from Chain Attributes	<b>Passed</b>
<b>SWC-121</b>	Missing Protection against Signature Replay Attacks	<b>Passed</b>
<b>SWC-122</b>	Lack of Proper Signature Verification	<b>Passed</b>
<b>SWC-123</b>	Requirement Violation	<b>Passed</b>
<b>SWC-124</b>	Write to Arbitrary Storage Location	<b>Passed</b>
<b>SWC-125</b>	Incorrect Inheritance Order	<b>Passed</b>
<b>SWC-126</b>	Insufficient Gas Griefing	<b>Passed</b>
<b>SWC-127</b>	Arbitrary Jump with Function Type Variable	<b>Passed</b>
<b>SWC-128</b>	DoS With Block Gas Limit	<b>Passed</b>
<b>SWC-129</b>	Typographical Error	<b>Passed</b>
<b>SWC-130</b>	Right-To-Left-Override control character (U+202E)	<b>Passed</b>
<b>SWC-131</b>	Presence of unused variables	<b>Passed</b>
<b>SWC-132</b>	Unexpected Ether balance	<b>Passed</b>
<b>SWC-133</b>	Hash Collisions With Multiple Variable Length Arguments	<b>Passed</b>
<b>SWC-134</b>	Message call with the hardcoded gas amount	<b>Passed</b>
<b>SWC-135</b>	Code With No Effects (Irrelevant/Dead Code)	<b>Passed</b>
<b>SWC-136</b>	Unencrypted Private Data On-Chain	<b>Passed</b>

# MANUAL ANALYSIS

The contract is verified to check if functions do and work as they should and malicious code is not inserted.

	Tested	Result
<b>Transfer</b>	Yes	<b>Passed</b>
<b>Total Supply</b>	Yes	<b>Passed</b>
<b>Buy Back</b>	Yes	<b>N/A</b>
<b>Burn</b>	Yes	<b>Passed</b>
<b>Mint</b>	Yes	<b>N/A</b>
<b>Rebase</b>	Yes	<b>N/A</b>
<b>Pause</b>	Yes	<b>N/A</b>
<b>Blacklist</b>	Yes	<b>Passed</b>
<b>Lock</b>	Yes	<b>N/A</b>
<b>Max Transaction</b>	Yes	<b>N/A</b>
<b>Transfer Ownership</b>	Yes	<b>Passed</b>
<b>Renounce Ownership</b>	Yes	<b>Passed</b>

# CONTRACT INSPECTION



** ERC20**   Interface		
L   totalSupply   External     NO		
L   balanceOf   External     NO		
L   transfer   External     <span style="color: red;">●</span>   NO		
L   allowance   External     NO		
L   approve   External     <span style="color: red;">●</span>   NO		
L   transferFrom   External     <span style="color: red;">●</span>   NO		
** Context**   Implementation		
L   _msgSender   Internal		
L   _msgData   Internal		
** UniswapV2Router01**   Interface		
L   factory   External     NO		
L   WETH   External     NO		
L   addLiquidity   External     <span style="color: red;">●</span>   NO		
L   addLiquidityETH   External        NO		
L   removeLiquidity   External     <span style="color: red;">●</span>   NO		
L   removeLiquidityETH   External     <span style="color: red;">●</span>   NO		
L   removeLiquidityWithPermit   External     <span style="color: red;">●</span>   NO		
L   removeLiquidityETHWithPermit   External     <span style="color: red;">●</span>   NO		
L   swapExactTokensForTokens   External     <span style="color: red;">●</span>   NO		
L   swapTokensForExactTokens   External     <span style="color: red;">●</span>   NO		
L   swapExactETHForTokens   External        NO		
L   swapTokensForExactETH   External     <span style="color: red;">●</span>   NO		
L   swapExactTokensForETH   External     <span style="color: red;">●</span>   NO		
L   swapETHForExactTokens   External        NO		
L   quote   External     NO		
L   getAmountOut   External     NO		
L   getAmountIn   External     NO		
L   getAmountsOut   External     NO		
L   getAmountsIn   External     NO		

**IUniswapV2Router02**   Interface   IUniswapV2Router01
L   removeLiquidityETHSupportingFeeOnTransferTokens   External     NO
L   removeLiquidityETHWithPermitSupportingFeeOnTransferTokens   External     NO
L   swapExactTokensForTokensSupportingFeeOnTransferTokens   External     NO
L   swapExactETHForTokensSupportingFeeOnTransferTokens   External     NO
L   swapExactTokensForETHSupportingFeeOnTransferTokens   External     NO
**IUniswapV2Factory**   Interface
L   feeTo   External     NO
L   feeToSetter   External     NO
L   getPair   External     NO
L   allPairs   External     NO
L   allPairsLength   External     NO
L   createPair   External     NO
L   setFeeTo   External     NO
L   setFeeToSetter   External     NO
**IUniswapV2Pair**   Interface
L   name   External     NO
L   symbol   External     NO
L   decimals   External     NO
L   totalSupply   External     NO
L   balanceOf   External     NO
L   allowance   External     NO
L   approve   External     NO
L   transfer   External     NO
L   transferFrom   External     NO
L   DOMAIN_SEPARATOR   External     NO
L   PERMIT_TYPEHASH   External     NO
L   nonces   External     NO
L   permit   External     NO
L   MINIMUM_LIQUIDITY   External     NO
L   factory   External     NO
L   token0   External     NO
L   token1   External     NO
L   getReserves   External     NO
L   price0CumulativeLast   External     NO
L   price1CumulativeLast   External     NO

```
| L | kLast | External | | NO | | |
| L | burn | External | | ● | NO | |
| L | swap | External | | ● | NO | |
| L | skim | External | | ● | NO | |
| L | sync | External | | ● | NO | |
| L | initialize | External | | ● | NO | |
|||||
| **IERC20Metadata** | Interface | IERC20 |||
| L | name | External | | NO | |
| L | symbol | External | | NO | |
| L | decimals | External | | NO | |
|||||
| **Ownable** | Implementation | Context |||
| L | <Constructor> | Public | | ● | NO | |
| L | owner | Public | | NO | |
| L | renounceOwnership | Public | | ● | onlyOwner |
| L | transferOwnership | Public | | ● | onlyOwner |
|||||
| **SafeMath** | Library | ||
| L | add | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
|||||
| **SafeMathInt** | Library | ||
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | add | Internal 🔒 | | |
| L | abs | Internal 🔒 | | |
| L | toUint256Safe | Internal 🔒 | |
```

```
| **SafeMathUint** | Library | ||| | |
| L | toInt256Safe | Internal 🔒 | |||
|||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata ||
| L | <Constructor> | Public | 🔒 | NO! |
| L | name | Public | 🔒 | NO! |
| L | symbol | Public | 🔒 | NO! |
| L | decimals | Public | 🔒 | NO! |
| L | totalSupply | Public | 🔒 | NO! |
| L | balanceOf | Public | 🔒 | NO! |
| L | transfer | Public | 🔒 | NO! |
| L | allowance | Public | 🔒 | NO! |
| L | approve | Public | 🔒 | NO! |
| L | transferFrom | Public | 🔒 | NO! |
| L | increaseAllowance | Public | 🔒 | NO! |
| L | decreaseAllowance | Public | 🔒 | NO! |
| L | _transfer | Internal 🔒 | 🔒 | |||
| L | _cast | Internal 🔒 | 🔒 | |||
| L | _burn | Internal 🔒 | 🔒 | |||
| L | _approve | Internal 🔒 | 🔒 | |||
| L | _beforeTokenTransfer | Internal 🔒 | 🔒 | |||
|||||
| **DividendPayingTokenInterface** | Interface | |||
| L | dividendOf | External | 🔒 | NO! |
| L | withdrawDividend | External | 🔒 | NO! |
|||||
| **DividendPayingTokenOptionalInterface** | Interface | |||
| L | withdrawableDividendOf | External | 🔒 | NO! |
| L | withdrawnDividendOf | External | 🔒 | NO! |
| L | accumulativeDividendOf | External | 🔒 | NO! |
|||||
| **DividendPayingToken** | Implementation | ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface ||
| L | <Constructor> | Public | 🔒 | NO! | ERC20 |
| L | distributeCAKEDividends | Public | 🔒 | NO! | onlyOwner |
| L | withdrawDividend | Public | 🔒 | NO! |
| L | _withdrawDividendOfUser | Internal 🔒 | 🔒 | |||
| L | dividendOf | Public | 🔒 | NO! |
```

```
| └ withdrawableDividendOf | Public ! | |NO! | |
| └ withdrawnDividendOf | Public ! | |NO! |
| └ accumulativeDividendOf | Public ! | |NO! |
| └ _transfer | Internal 🔒 | ○ | ||
| └ _cast | Internal 🔒 | ○ | ||
| └ _burn | Internal 🔒 | ○ | ||
| └ _setBalance | Internal 🔒 | ○ | ||
|||||
| **TokenDividendTracker** | Implementation | Ownable, DividendPayingToken |||
| └ <Constructor> | Public ! | ○ | DividendPayingToken |
| └ _transfer | Internal 🔒 | |||
| └ withdrawDividend | Public ! | |NO! |
| └ setMinimumTokenBalanceForDividends | External ! | ○ | onlyOwner |
| └ excludeFromDividends | External ! | ○ | onlyOwner |
| └ updateClaimWait | External ! | ○ | onlyOwner |
| └ getLastProcessedIndex | External ! | |NO! |
| └ getNumberOfTokenHolders | External ! | |NO! |
| └ isExcludedFromDividends | Public ! | |NO! |
| └ getAccount | Public ! | |NO! |
| └ getAccountAtIndex | Public ! | |NO! |
| └ canAutoClaim | Private 🔑 | |||
| └ setBalance | External ! | ○ | onlyOwner |
| └ process | Public ! | ○ |NO! |
| └ processAccount | Public ! | ○ | onlyOwner |
| └ MAPGet | Public ! | |NO! |
| └ MAPGetIndexOfKey | Public ! | |NO! |
| └ MAPGetKeyAtIndex | Public ! | |NO! |
| └ MAPSize | Public ! | |NO! |
| └ MAPSet | Public ! | ○ |NO! |
| └ MAPRemove | Public ! | ○ |NO! |
|||||
| **RedKing** | Implementation | ERC20, Ownable |||
| └ <Constructor> | Public ! | 🚀 | ERC20 |
| └ <Receive Ether> | External ! | 💸 |NO! |
| └ updateMinimumTokenBalanceForDividends | Public ! | ○ | onlyOwner |
| └ updateUniswapV2Router | Public ! | ○ | onlyOwner |
| └ excludeFromFees | Public ! | ○ | onlyOwner |
| └ excludeMultipleAccountsFromFees | Public ! | ○ | onlyOwner |
| └ setMarketingWallet1 | External ! | ○ | onlyOwner |
```

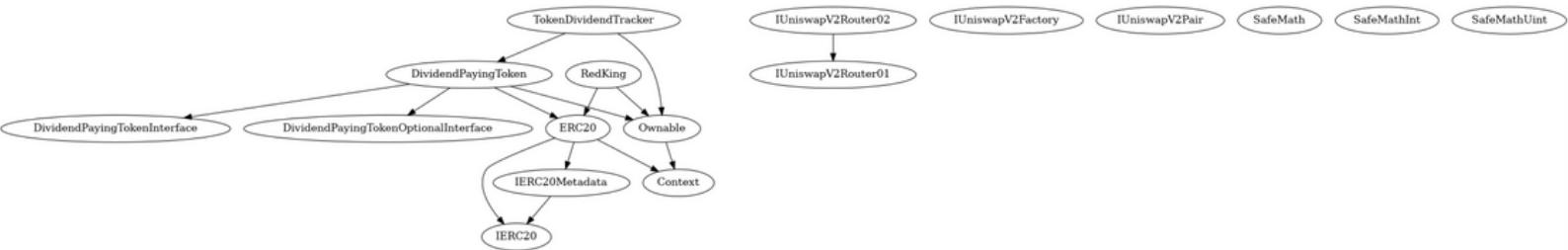
```

| L | setMarketingWallet2 | Public | 🔒 | 🔴 | onlyOwner |
| L | setAutomatedMarketMakerPair | Public | 🔒 | 🔴 | onlyOwner |
| L | _setAutomatedMarketMakerPair | Private | 🔒 | 🔴 || 
| L | EnemyAddress | External | 🔒 | 🔴 | onlyOwner |
| L | updateGasForProcessing | Public | 🔒 | 🔴 | onlyOwner |
| L | updateClaimWait | External | 🔒 | 🔴 | onlyOwner |
| L | getClaimWait | External | 🔒 | NO! |
| L | getTotalDividendsDistributed | External | 🔒 | NO! |
| L | isExcludedFromFees | Public | 🔒 | NO! |
| L | withdrawableDividendOf | Public | 🔒 | NO! |
| L | dividendTokenBalanceOf | Public | 🔒 | NO! |
| L | excludeFromDividends | External | 🔒 | 🔴 | onlyOwner |
| L | isExcludedFromDividends | Public | 🔒 | NO! |
| L | getAccountDividendsInfo | External | 🔒 | NO! |
| L | getAccountDividendsInfoAtIndex | External | 🔒 | NO! |
| L | processDividendTracker | External | 🔒 | 🔴 | NO! |
| L | claim | External | 🔒 | 🔴 | NO! |
| L | getLastProcessedIndex | External | 🔒 | NO! |
| L | getNumberOfDividendTokenHolders | External | 🔒 | NO! |
| L | setSwapTokensAtAmount | Public | 🔒 | 🔴 | onlyOwner |
| L | setTokenRewardsFee | Public | 🔒 | 🔴 | onlyOwner |
| L | setLiquidityFee | Public | 🔒 | 🔴 | onlyOwner |
| L | setMarketingFee | Public | 🔒 | 🔴 | onlyOwner |
| L | setTechnologyFee | Public | 🔒 | 🔴 | onlyOwner |
| L | setDeadFee | Public | 🔒 | 🔴 | onlyOwner |
| L | _transfer | Internal | 🔒 | 🔴 || 
| L | swapAndSendToFee | Private | 🔒 | 🔴 || 
| L | swapAndLiquify | Private | 🔒 | 🔴 || 
| L | swapTokensForEth | Private | 🔒 | 🔴 || 
| L | swapTokensForCake | Private | 🔒 | 🔴 || 
| L | addLiquidity | Private | 🔒 | 🔴 || 
| L | swapAndSendDividends | Private | 🔒 | 🔴 || 

```

Symbol	Meaning
🔴	Function can modify state
\$	Function is payable
🔒	Private function
🔓	Internal function
NO!	Function has no modifier

# INHERITANCE TREE



Inheritance is a feature of the object-oriented programming language. It is a way of extending the functionality of a program, used to separate the code, reduces the dependency, and increases the re-usability of the existing code. Solidity supports inheritance between smart contracts, where multiple contracts can be inherited into a single contract.

# Important Snippets



## Exclude multiple accounts from fees

```
function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }
    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

## Blacklisted addresses are not permitted to transfer their tokens

```
function EnemyAddress(address account, bool value) external onlyOwner{
    _isEnemy[account] = value;
}
```

## Exclude from dividends

```
function excludeFromDividends(address account) external onlyOwner{
    dividendTracker.excludeFromDividends(account);
}
```

## Owner can set fees up to 100%

```
function setTokenRewardsFee(uint256 amount) public onlyOwner {
    tokenRewardsFee = amount;
}
function setLiquidityFee(uint256 amount) public onlyOwner {
    liquidityFee = amount;
}
function setMarketingFee(uint256 amount) public onlyOwner {
    marketingFee1 = amount;
}

function setTechnologyFee(uint256 amount) public onlyOwner {
    marketingFee2 = amount;
}

function setDeadFee(uint256 amount) public onlyOwner {
    deadFee = amount;
}
```

# GOOD PRACTICES ✓

---

- The owner cannot stop or pause the smart contract
- The owner cannot mint new tokens after deployment
- The owner cannot set max Tx
- The smart contract utilizes "SafeMath" to prevent overflows

```
library SafeMath {  
    function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {  
        unchecked {  
            uint256 c = a + b;  
            if (c < a) return (false, 0);  
            return (true, c);  
        }  
    }  
  
    function trySub(uint256 a, uint256 b) internal pure returns (bool, uint256) {  
        unchecked {  
            if (b > a) return (false, 0);  
            return (true, a - b);  
        }  
    }  
  
    function tryMul(uint256 a, uint256 b) internal pure returns (bool, uint256) {  
        unchecked {  
            // Gas optimization: this is cheaper than requiring 'a' not being zero, but  
            // benefit is lost if 'b' is also tested.  
            // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522  
            if (a == 0) return (true, 0);  
            uint256 c = a * b;  
            if (c / a != b) return (false, 0);  
            return (true, c);  
        }  
    }  
  
    function tryDiv(uint256 a, uint256 b) internal pure returns (bool, uint256) {  
        unchecked {  
            if (b == 0) return (false, 0);  
            return (true, a / b);  
        }  
    }  
  
    function tryMod(uint256 a, uint256 b) internal pure returns (bool, uint256) {  
        unchecked {  
            if (b == 0) return (false, 0);  
            return (true, a % b);  
        }  
    }  
}
```

# WEBSITE



<b>Website</b>	<a href="http://3pointsdesign.com/">http://3pointsdesign.com/</a>
<b>Domain Registry</b>	<a href="http://www.godaddy.com">http://www.godaddy.com</a>
<b>Domain Expiry Date</b>	2023-01-05
<b>Response Code</b>	200
<b>SSL Checker and HTTPS Test</b>	Medium
<b>Deprecated HTML tags</b>	Passed
<b>Robots.txt</b>	Informational
<b>Sitemap Test</b>	Informational
<b>SEO Friendly URL</b>	Passed
<b>Responsive Test</b>	Passed
<b>JS Error Test</b>	Passed
<b>Console Errors Test</b>	Informational
<b>Site Loading Speed Test</b>	1.96 seconds - Passed
<b>HTTP2 Test</b>	Medium
<b>Safe Browsing Test</b>	Passed

# DISCLAIMER

---

SafuAudit.com is not a financial institution and the information provided on this website does not constitute investment advice, financial advice, trading advice or any other sort of advice. You should not treat any of the website's content as such. Investing in crypto assets carries a high level of risk and does not hold guarantees for not sustaining financial loss due to their volatility.

## Accuracy of Information

SafuAudit will strive to ensure accuracy of information listed on this website although it will not hold any responsibility for any missing or wrong information. SafuAudit provides all information as is. You understand that you are using any and all information available here at your own risk. Any use or reliance on our content and services is solely at your own risk and discretion.

The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project.

While we have used all the information available to us for this straightforward investigation, you should not rely on this report only – we recommend proceeding with several independent audits. Be aware that smart contracts deployed on a blockchain aren't secured enough against external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, SafuAudit does not guarantee the explicit security of the audited smart contract. 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.

# AUDIT RESULTS

---

## CRITICAL

---

No critical severity issues have been found.

## MEDIUM

---

- Owner can set fees to 100%. It can be required to change taxes, but it should be restricted beyond a certain range.

## MINOR

---

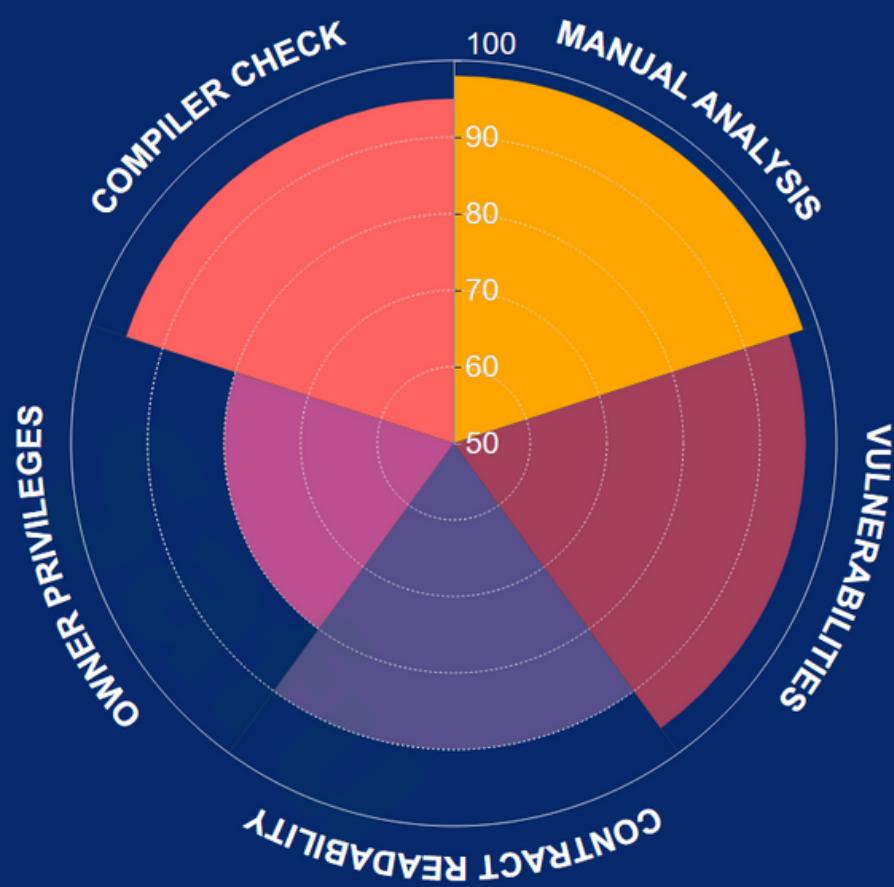
- A floating pragma is set. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.
- Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" instead.

## INFORMATIONAL

---

The standard audit model does not offer suggestions and consulting for improvements of efficacy.

# SAFUSCORE



Manual Analysis



Vulnerabilities



Contract Readability



Owner Privileges

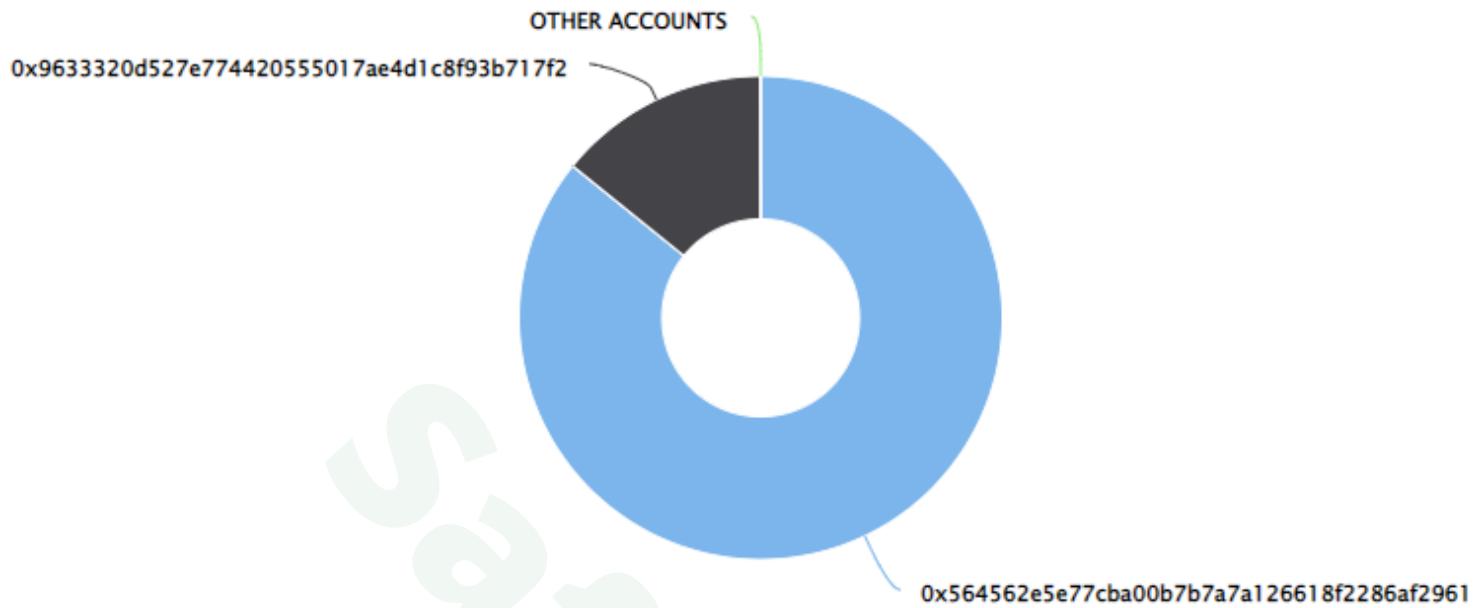


Compiler Check

**Final Score: 91.8**

# SUMMARY

## Top 10 holders



Rank	Address	Quantity (Token)	Percentage
1	0x564562e5e77cba00b7b7a7a126618f2286af2961	858,200,000	85.8200%
2	0x9633320d527e774420555017ae4d1c8f93b717f2	141,800,000	14.1800%

## Conclusion

Project ThreePoints does not contain any severe issues or risk characteristics. Owner should have been limited to set fees beyond an agreed rate.

SafuAudit has tested the security based on manual and automated tests. Please note that we don't offer any warranties for business model.





**SafuAudit.com**

