



SAFUAUDIT
SMART CONTRACT AUDITING

THREEPOINTS

SMART CONTRACT AUDIT



March 11, 2022

INTRODUCTION

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

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.

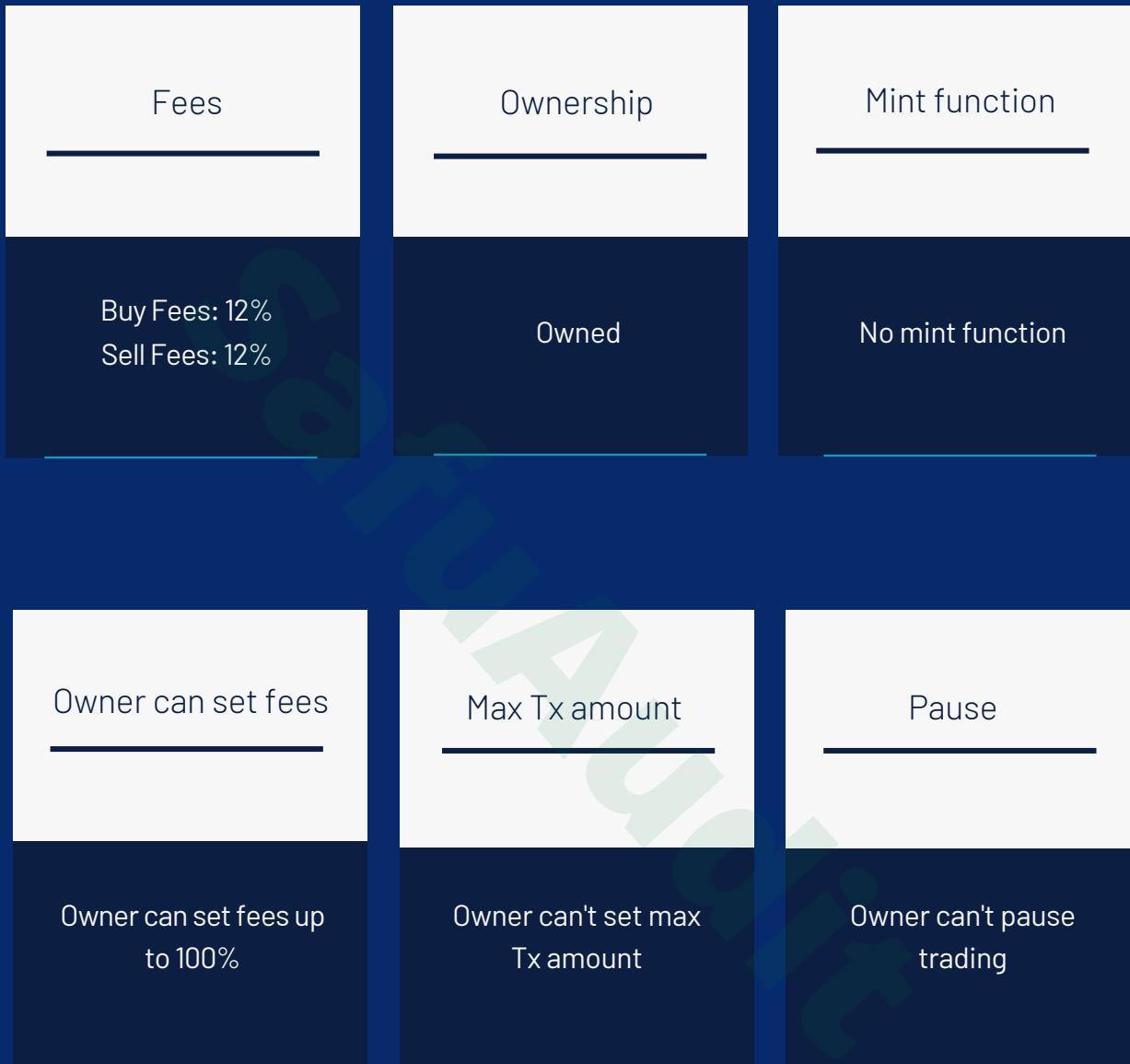
MINOR

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SWC-102	Outdated Compiler Version	Passed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

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 the hardcoded gas amount	Passed
SWC-135	Code With No Effects (Irrelevant/Dead Code)	Passed
SWC-136	Unencrypted Private Data On-Chain	Passed

MANUAL ANALYSIS

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

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

CONTRACT INSPECTION



** ERC20** Interface		
L totalSupply External NO		
L balanceOf External NO		
L transfer External ● NO		
L allowance External NO		
L approve External ● NO		
L transferFrom External ● NO		
** Context** Implementation		
L _msgSender Internal		
L _msgData Internal		
** UniswapV2Router01** Interface		
L factory External NO		
L WETH External NO		
L addLiquidity External ● NO		
L addLiquidityETH External NO		
L removeLiquidity External ● NO		
L removeLiquidityETH External ● NO		
L removeLiquidityWithPermit External ● NO		
L removeLiquidityETHWithPermit External ● NO		
L swapExactTokensForTokens External ● NO		
L swapTokensForExactTokens External ● NO		
L swapExactETHForTokens External NO		
L swapTokensForExactETH External ● NO		
L swapExactTokensForETH External ● 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 |
```

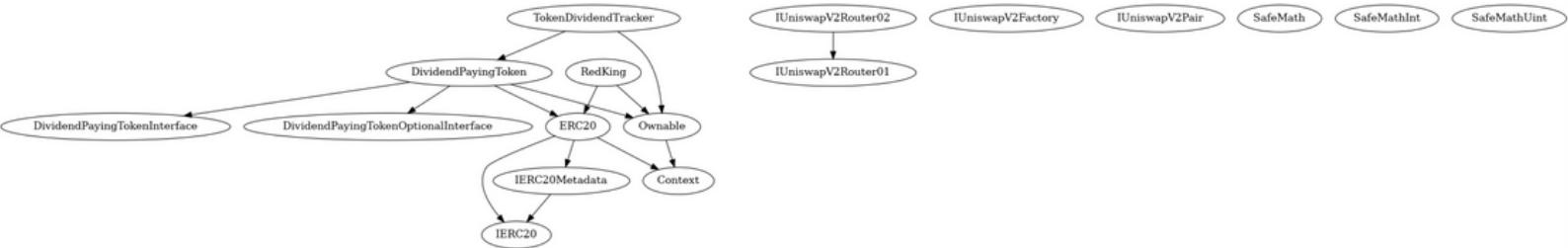
```

| 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



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

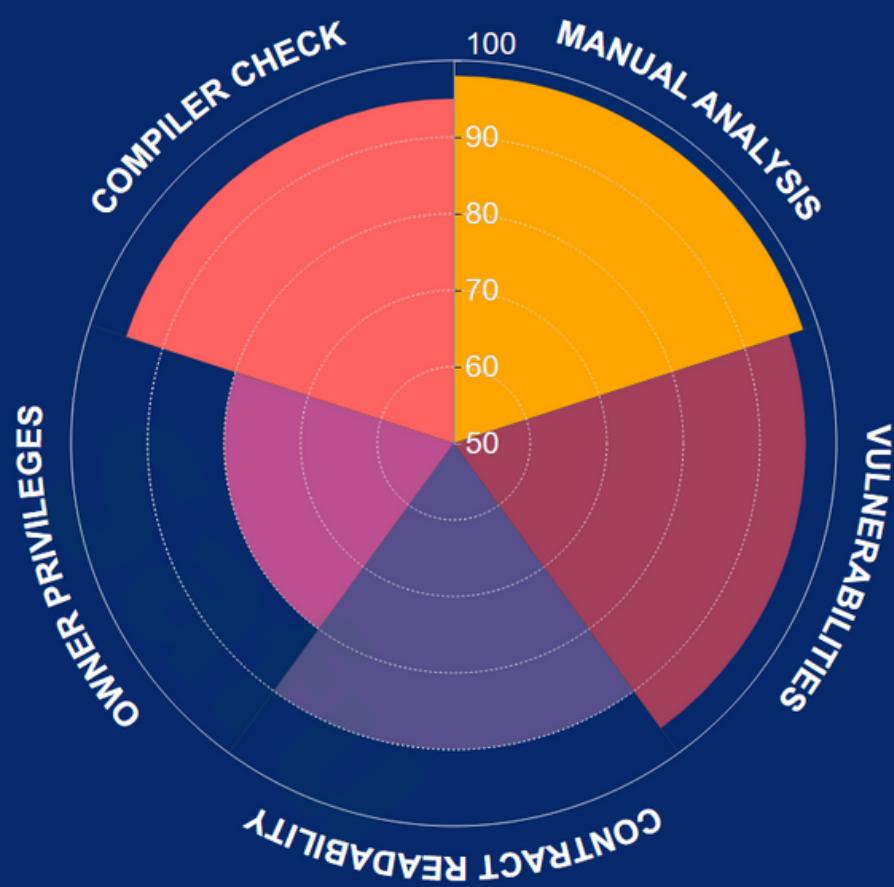
MINOR

- 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

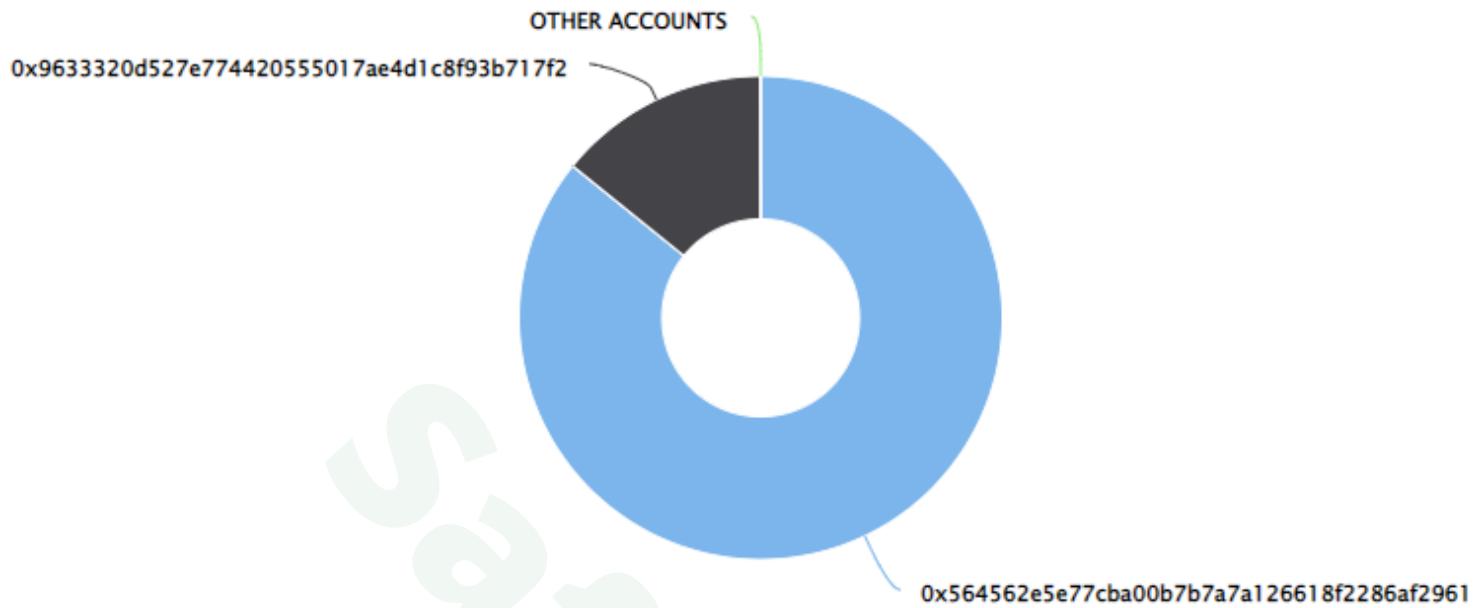


Compiler Check

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

