



Smart Contract Audit

FOR

bankp2pgold

DATED : 9 June 23'

HIGH RISK FINDING

Centralization – Trades must be enabled

Severity: **High**

function: enableTrading

Status: **Resolved** (Contract is owned by safu developer)

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function enableTrading() external onlyOwner {  
    require(tradingEnabled == false, "Trading is already enabled");  
    tradingEnabled = true;  
}
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1. Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.
2. Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.
3. Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades



AUDIT SUMMARY

Project name – bankp2pgold

Date: 9 June, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	1
Acknowledged	0	0	0	0	0
Resolved	0	1	0	0	0

USED TOOLS

Tools:

1- Manual Review:

A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :

The code has undergone static analysis using Slither.

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0xc9952C25f69711C522Fdb99a858583261f15A366>



Token Information

Token Name : bankp2pgold

Token Symbol: bankp2pgold

Decimals: 18

Token Supply:1,000,000,000

Token Address: ---

Checksum:

b568b8d4cfcfc26718929b54abd9ac85b25e5079

Owner: ---

Deployer: ---



TOKEN OVERVIEW

Fees:

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 10%

Fees Privilege: Owner

Ownership: ---

Minting: None

Max Tx Amount/ Max Wallet Amount: Yes

Blacklist: No

Other Privileges:- initial distribution of tokens

- including or excluding from fees
 - changing swap threshold
 - enabling trades
-
-



AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
 - Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
 - Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
 - Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
 - Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
 - Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.
-

VULNERABILITY CHECKLIST

- | | |
|------------------------------------|-------------------------------|
| ✓ Return values of low-level calls | ✓ Gasless Send |
| ✓ Private modifier | ✓ Using block.timestamp |
| ✓ Multiple Sends | ✓ Re-entrancy |
| ✓ Using Suicide | ✓ Tautology or contradiction |
| ✓ Gas Limitand Loops | ✓ Timestamp Dependence |
| ✓ Address hardcoded | ✓ Revert/require functions |
| ✓ Exception Disorder | ✓ Use of tx.origin |
| ✓ Using inline assembly | ✓ Integer overflow/underflow |
| ✓ Divide before multiply | ✓ Dangerous strict equalities |
| ✓ Missing Zero Address Validation | ✓ Using SHA3 |
| ✓ Compiler version not fixed | ✓ Using throw |
-



CLASSIFICATION OF RISK

Severity

Description

◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

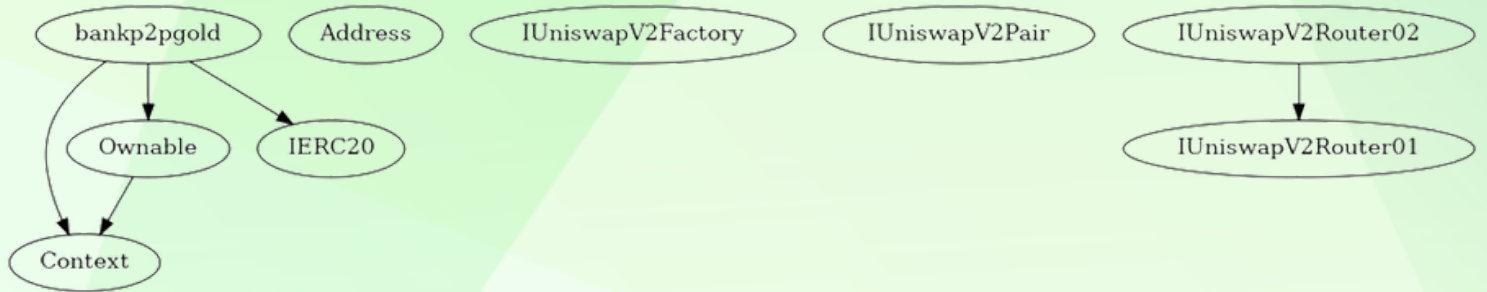
Severity

Found

◆ Critical	0
◆ High-Risk	1
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	1



INHERITANCE TREE



POINTS TO NOTE

- owner is not able to change fees (10% for buy and sell, 0% for transfers)
 - owner is not able to blacklist an arbitrary wallet
 - owner is not able to set limit for buy/sell/transfer/holding amounts
 - owner is not able to mint new tokens
 - owner is not able to disable trades
-



CONTRACT ASSESMENT

```
| Contract |      Type      |      Bases      |      |      |
|:-----:|:-----:|:-----:|:-----:|:-----:|
|  L  | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
|||||
| **Context** | Implementation | |||
|  L  | _msgSender | Internal 🔒 | | |
|  L  | _msgData | Internal 🔒 | | |
|||||
| **Ownable** | Implementation | Context |||
|  L  | <Constructor> | Public ! | ⬤ | NO ! |
|  L  | owner | Public ! | | NO ! |
|  L  | renounceOwnership | Public ! | ⬤ | onlyOwner |
|  L  | transferOwnership | Public ! | ⬤ | onlyOwner |
|||||
| **IERC20** | 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 ! |
|||||
| **Address** | Library | |||
|  L  | isContract | Internal 🔒 | | |
|  L  | sendValue | Internal 🔒 | ⬤ | |
|  L  | functionCall | Internal 🔒 | ⬤ | |
|  L  | functionCall | Internal 🔒 | ⬤ | |
|  L  | functionCallWithValue | Internal 🔒 | ⬤ | |
|  L  | functionCallWithValue | Internal 🔒 | ⬤ | |
|  L  | _functionCallWithValue | Private 🔒 | ⬤ | |
|||||
| **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 | |||
```



CONTRACT ASSESMENT






























```
|  | name | External ! | |NO ! |
|  | symbol | External ! | |NO ! |
|  | decimals | External ! | |NO ! |
|  | totalSupply | External ! | |NO ! |
|  | balanceOf | External ! | |NO ! |
|  | allowance | External ! | |NO ! |
|  | approve | External ! | ● |NO ! |
|  | transfer | External ! | ● |NO ! |
|  | transferFrom | External ! | ● |NO ! |
|  | DOMAIN_SEPARATOR | External ! | |NO ! |
|  | PERMIT_TYPEHASH | External ! | |NO ! |
|  | nonces | External ! | |NO ! |
|  | permit | External ! | ● |NO ! |
|  | MINIMUM_LIQUIDITY | External ! | |NO ! |
|  | factory | External ! | |NO ! |
|  | token0 | External ! | |NO ! |
|  | token1 | External ! | |NO ! |
|  | getReserves | External ! | |NO ! |
|  | price0CumulativeLast | External ! | |NO ! |
|  | price1CumulativeLast | External ! | |NO ! |
|  | kLast | External ! | |NO ! |
|  | burn | External ! | ● |NO ! |
|  | swap | External ! | ● |NO ! |
|  | skim | External ! | ● |NO ! |
|  | sync | External ! | ● |NO ! |
|  | initialize | External ! | ● |NO ! |
|  |  |  |  |  |
| **IUniswapV2Router01** | Interface |  |  |
|  | factory | External ! | |NO ! |
|  | WETH | External ! | |NO ! |
|  | addLiquidity | External ! | ● |NO ! |
|  | addLiquidityETH | External ! | $D |NO ! |
|  | removeLiquidity | External ! | ● |NO ! |
|  | removeLiquidityETH | External ! | ● |NO ! |
|  | removeLiquidityWithPermit | External ! | ● |NO ! |
|  | removeLiquidityETHWithPermit | External ! | ● |NO ! |
|  | swapExactTokensForTokens | External ! | ● |NO ! |
|  | swapTokensForExactTokens | External ! | ● |NO ! |
|  | swapExactETHForTokens | External ! | $D |NO ! |
|  | swapTokensForExactETH | External ! | ● |NO ! |
|  | swapExactTokensForETH | External ! | ● |NO ! |
```






CONTRACT ASSESMENT

```
|  | swapETHForExactTokens | External ! |  | NO ! |
|  | quote | External ! |  | NO ! |
|  | getAmountOut | External ! |  | NO ! |
|  | getAmountIn | External ! |  | NO ! |
|  | getAmountsOut | External ! |  | NO ! |
|  | getAmountsIn | External ! |  | NO ! |
|  |  |
|  |  |
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
|  | removeLiquidityETHSupportingFeeOnTransferTokens | External ! |  | NO ! |
|  | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! |  | NO ! |
|  | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! |  | NO ! |
|  | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! |  | NO ! |
|  | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! |  | NO ! |
|  |  |
| **bankp2pgold** | Implementation | Context, IERC20, Ownable |||
|  | <Constructor> | Public ! |  | NO ! |
|  | name | Public ! |  | NO ! |
|  | symbol | Public ! |  | NO ! |
|  | decimals | Public ! |  | NO ! |
|  | totalSupply | Public ! |  | NO ! |
|  | balanceOf | Public ! |  | NO ! |
|  | transfer | Public ! |  | NO ! |
|  | allowance | Public ! |  | NO ! |
|  | approve | Public ! |  | NO ! |
|  | transferFrom | Public ! |  | NO ! |
|  | increaseAllowance | Public ! |  | NO ! |
|  | decreaseAllowance | Public ! |  | NO ! |
|  | isExcludedFromReward | Public ! |  | NO ! |
|  | totalReflectionDistributed | Public ! |  | NO ! |
|  | deliver | Public ! |  | NO ! |
|  | reflectionFromToken | Public ! |  | NO ! |
|  | tokenFromReflection | Public ! |  | NO ! |
|  | excludeFromReward | Public ! |  | onlyOwner |
|  | includeInReward | External ! |  | onlyOwner |
|  | <Receive Ether> | External ! |  | NO ! |
|  | claimStuckTokens | External ! |  | onlyOwner |
|  | _reflectFee | Private  |  |
|  | _getValues | Private  |  |
|  | _getTValues | Private  |  |
|  | _getRValues | Private  |  |
|  | _getRate | Private  |  |
```

CONTRACT ASSESMENT

^L	_getCurrentSupply	Private 			
^L	_takeLiquidity	Private 			
^L	_takeMarketing	Private 			
^L	calculateTaxFee	Private 			
^L	calculateLiquidityFee	Private 			
^L	calculateMarketingFee	Private 			
^L	removeAllFee	Private 			
^L	setBuyFee	Private 			
^L	setSellFee	Private 			
^L	isExcludedFromFee	Public !		NO !	
^L	_approve	Private 			
^L	enableTrading	External !		onlyOwner	
^L	_transfer	Private 			
^L	swapAndSendMarketing	Private 			
^L	setSwapTokensAtAmount	External !		onlyOwner	
^L	setSwapEnabled	External !		onlyOwner	
^L	_tokenTransfer	Private 			
^L	_transferStandard	Private 			
^L	_transferToExcluded	Private 			
^L	_transferFromExcluded	Private 			
^L	_transferBothExcluded	Private 			
^L	excludeFromFees	External !		onlyOwner	
^L	changeMarketingWallet	External !		onlyOwner	

Legend

Symbol	Meaning
	Function can modify state
	Function is payable
	Function is payable



STATIC ANALYSIS

```
Variable bankp2pgold._getValues(uint256).rTransferAmount (contracts/Token.sol#614) is too similar to bankp2pgold._getValues(uint256).tTransferAmount (contracts/Token.sol#613)
Variable bankp2pgold.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#557) is too similar to bankp2pgold._getValues(uint256).tTransferAmount (contracts/Token.sol#613)
Variable bankp2pgold._getValues(uint256).rTransferAmount (contracts/Token.sol#614) is too similar to bankp2pgold._getTValues(uint256).tTransferAmount (contracts/Token.sol#623)
Variable bankp2pgold._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to bankp2pgold._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#819)
Variable bankp2pgold._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to bankp2pgold._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#856)
Variable bankp2pgold._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to bankp2pgold._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#837)
Variable bankp2pgold._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#854) is too similar to bankp2pgold._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#875)
Variable bankp2pgold._getValues(uint256).rTransferAmount (contracts/Token.sol#614) is too similar to bankp2pgold._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#875)
Variable bankp2pgold.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#557) is too similar to bankp2pgold._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#875)
Variable bankp2pgold._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#817) is too similar to bankp2pgold._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#875)
Variable bankp2pgold._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#636) is too similar to bankp2pgold._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#875)
Variable bankp2pgold._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to bankp2pgold._getTValues(uint256).tTransferAmount (contracts/Token.sol#623)
Variable bankp2pgold._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#873) is too similar to bankp2pgold._getValues(uint256).tTransferAmount (contracts/Token.sol#613)
Variable bankp2pgold._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#854) is too similar to bankp2pgold._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#819)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

bankp2pgold.DEAD (contracts/Token.sol#417) is never used in bankp2pgold (contracts/Token.sol#378-903)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

bankp2pgold.DEAD (contracts/Token.sol#417) should be constant
bankp2pgold._decimals (contracts/Token.sol#392) should be constant
bankp2pgold._name (contracts/Token.sol#390) should be constant
bankp2pgold._symbol (contracts/Token.sol#391) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

bankp2pgold._tTotal (contracts/Token.sol#395) should be immutable
bankp2pgold._liquidityFeeonBuy (contracts/Token.sol#402) should be immutable
bankp2pgold._liquidityFeeonSell (contracts/Token.sol#403) should be immutable
bankp2pgold._marketingFeeonBuy (contracts/Token.sol#405) should be immutable
bankp2pgold._marketingFeeonSell (contracts/Token.sol#406) should be immutable
bankp2pgold._taxFeeonBuy (contracts/Token.sol#399) should be immutable
bankp2pgold._taxFeeonSell (contracts/Token.sol#400) should be immutable
bankp2pgold._totalBuyFees (contracts/Token.sol#412) should be immutable
bankp2pgold._totalSellFees (contracts/Token.sol#413) should be immutable
bankp2pgold._uniswapV2Pair (contracts/Token.sol#420) should be immutable
bankp2pgold._uniswapV2Router (contracts/Token.sol#419) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output



FUNCTIONAL TESTING

1- Adding liquidity (passed):

<https://testnet.bscscan.com/tx/0xe1d23839577d021a9627cb22bab951fe02bb95bb65f2533bbe01e4b67865d8a>

2- Buying when excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0xbb5b19f4b8b9159ab2ca61a1ede0741b89b7c0c577897e247820d7908a3e1710>

3- Selling when excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x19bbf85f19704d1ab701e2fc0ecbff89617e71ef1fd77afbf7fa0c196ef33bb9>

4- Transferring when excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x19444b2ea10bbd3c8f3d8d5b3c3bf1aac71332f7eae33db65e3d4027d0b96ef3>

5- Buying when not excluded from fees (10% static tax) (passed):

<https://testnet.bscscan.com/tx/0xa355426458e97ce1004ce4c13e004f72193656010289ec2aa82471538295b39c>

6- Selling when not excluded from fees (10% static tax) (passed):

<https://testnet.bscscan.com/tx/0x3144465757b4022cc3fc54dce6fcf6b2b31cd664294f0fc87fc0ec5f582dce79>



FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x0349f50fb65998fb3c6fd80afdd9606044790157dfabea3c560e3350c4c5eb98>

8- Internal swap (passed):

<https://testnet.bscscan.com/address/0x41f38cc1fb5d9e3b3d535827ad1c77f586f17796#internaltx>

FUNCTIONAL TESTING

Centralization – Trades must be enabled

Severity: **High**

function: enableTrading

Status: **Resolved** (Contract is owned by safu developer)

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function enableTrading() external onlyOwner {  
    require(tradingEnabled == false, "Trading is already enabled");  
    tradingEnabled = true;  
}
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1. Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.
2. Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.
3. Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades



FUNCTIONAL TESTING

Suggestion – Immutable tax

Severity: **Informational**

Status: Open

Overview:

Buy/Sell taxes can not be updated later (10% static tax). Owner might need to change fees based on different market conditions.

Suggestion

It is suggested to have a function to be able to update buy/sell/transfer tax in a safe range (0-10%)



DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



ABOUT AUDITACE

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



<https://auditace.tech/>



https://t.me/Audit_Ace



https://twitter.com/auditace_



<https://github.com/Audit-Ace>
