

AUDIT REPORT

DATE APRIL 7TH

FOR

P  **RK** **s w a p**



Disclaimer

SolidGroup reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. Solid group do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc’...)

SolidGroup Audits do not provide any warranty or guarantee **regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors. SolidGroup Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.** SolidGroup Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. **SolidGroup’s position is that each company and individual are responsible for their own due diligence and continuous security.** SolidGroup in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Overview

Network: Binance Smart Chain
Website: <https://porkswap.finance>
Twitter Group: https://twitter.com/pork_swap
Telegram Group: <https://t.me/porkswap>

Description

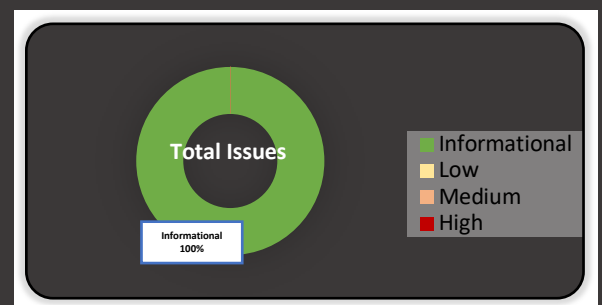
Decentralized Spot and Futures Trading on BSC. Automated market maker powered, taking De-Fi game to the next level. Make instant spot exchange, leveraged trades, provide liquidity to earn returns and rewards

Files In Scope

Contract Name	MD5
PSWAP.sol	15aaa5eaf9fa4e171f6b8115aa73bc18

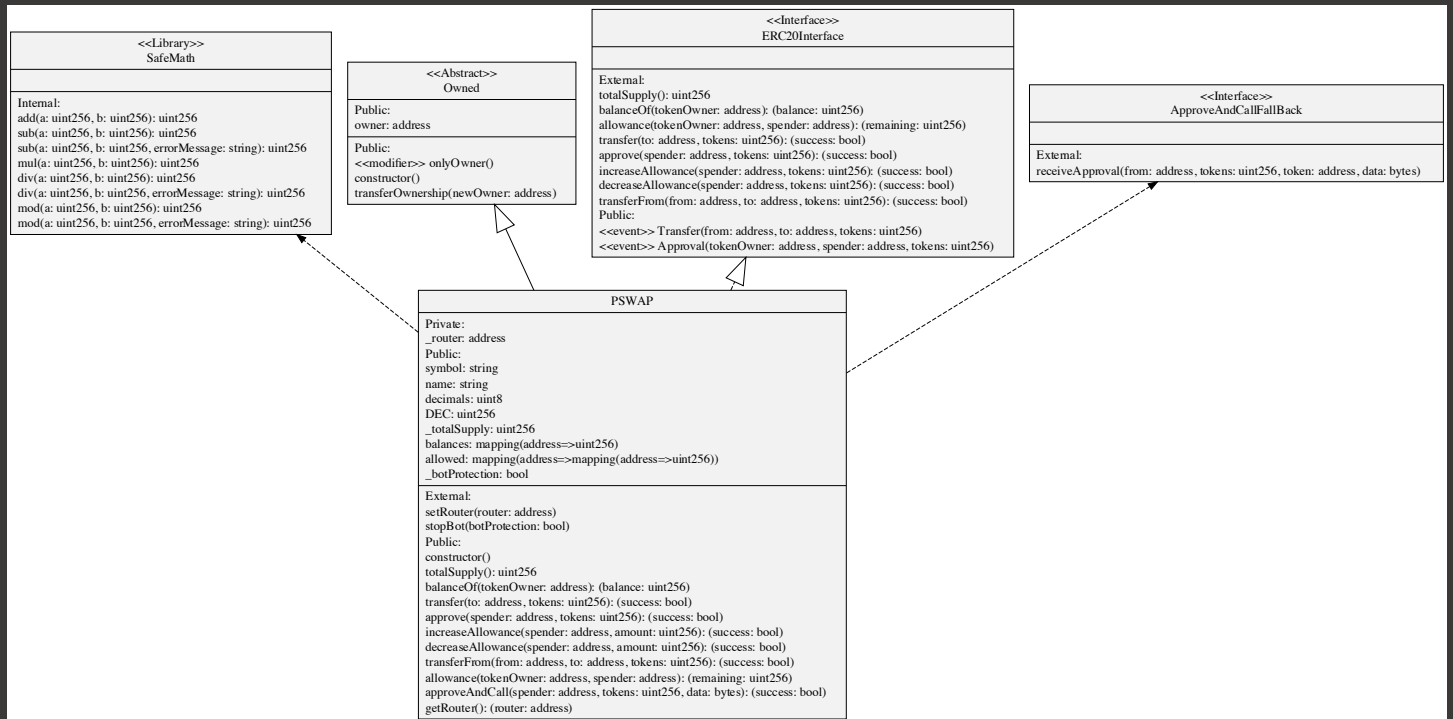
Vulnerability Summary

● Informational severity Issues	3
● Low severity issues	0
● Medium severity issues	0
● High severity issues	0



UML

PSWAP.sol



BEP-20's Conformance

This test checks for BEP-20's conformance.

- All the functions are present
- All the events are present
- Functions return the correct type
- Functions that must be view are view
- Events' parameters are correctly indexed
- The functions emit the events
- Derived contracts do not break the conformance

Function	present	type	Correct Return value	events	
totalSupply	✓	✓ view	✓		
balanceOf(address)	✓	✓ view	✓		
transfer(address,uint256)	✓	✓ external	✓	✓ Transfer	
transferFrom(address, address, uint256)	✓	✓ external	✓	✓ Transfer	
approve(address,uint256)	✓	✓ external	✓	✓ Approval	
allowance(address, address)	✓	✓ view	✓		
name	✓	✓ view	✓		
symbol	✓	✓ view	✓		

Check Events:

- ✓ Transfer
- ✓ Approve

General:

- ✓ No external mint function
- ✓ No Volatile Code

The contract that was tested is the token's contract: PSWAP.sol

General Notes

- Basic BEP-20 contract with some additional features.

Main Features:

- `_tradingEnabled` variable was added for the protection from premature listing.
- When `_tradingEnabled` is turned off only the owner and Unicrypt ILO contract can call `approve`, and `allowance` functions.
- The owner can call `enableTrade()` for turning on `_tradingEnabled`.
- Once the owner called `enableTrade()` It's irreversible.
- In our audit report we don't cover integration with external contracts (such as Uniswap, Unicrypt, PancakeSwap ... etc) We raised our concern regarding the integration with Unicrypt ILO contracts but the team reassured us that the integration was tested and everything worked as expected. The team added two fallback functions in case there will be an error "`setUnicrypt`" and "`enableTrade`".

PSWAP.sol

Description – Basic BEP-20 Contract

Issue #1:

Type	Severity	Location
Lack of events	● Informational	PSWAP.sol

Description:

The function “transferOwnership” is not emitting event.

Recommendation:

Our recommendation is to add events in critical parts of the contract, such as when transferring ownership. Events are great for integrating with DApps in the future and also for the integration with Blockchain explorers. We recommend considering emitting events when state is changed.

Issue #2:

Type	Severity	Location
Best Practice	● Informational	PSWAP.sol

Description:

State variables that are never changed should be constant.

Recommendation:

The state variable “decimals”, “name”, “symbol” are never changed consider changing their attribute to constant to save on gas. z

Issue #3:

Type	Severity	Location
Gas Optimization	● Informational	PSWAP.sol

Description:

The public function “getRouter” should be declared as external.

Recommendation:

These functions are only called outside the contract consider using external attribute instead of public.

Summary

● Informational severity Issues	3
● Low severity issues	0
● Medium severity issues	0
● High severity issues	0