



SOLIDRATE
<https://solidrate.io>

SMART CONTRACT AUDIT

POP PIG
date **04 MAY 2024**



Table Of Contents

Contract Details	3
Contract Analysis	
Trading Security	4
Token Authority	6
SWC Security	8
Community Trust	
Social Media Presence	11
Listings and Partnerships	11
Audit Summary	12
Audit Methodology	13
Disclaimer	14

Contract Details

Contract Address	9BenhXdVfF8wNHq9FWAFmZqQw6qBFgkSQ8CNjSzYndip
Contract Name	POPPIG
Symbol	POPPIG
Network	Solana Network
Compiler Version	v0.8.19+commit.7dd6d404
Licence	MIT License
Decimals	9
Current Supply	999,921,419.56
Deployer Address	5aMMj7jeWnPhuLyjTFxzHeh8AwPgwlD7D3RQhamTEtB
Owner Address	TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA

Contract Analysis

Trading Security

Thorough manual examination of the code, including line-by-line analysis and a meticulous review of trading constraints, taxes, and owner privileges. Our assessment ensures a comprehensive understanding of potential security risks.

Trading Constraints

Safety Overview

Passed

Failed

Analysis

Results

No pause function

Passed

No trading cooldown function

Passed

No blacklist function

Passed

No whitelist function

Passed

Contract Features

Safety Overview

Passed

Failed

Analysis

Results

Fees below 5%

Passed

No max transaction amount feature

Passed

No max wallet feature

Passed

No limited transactions feature

Passed

Token Authority

Contract Checks

Our assessment covers meticulous contract checks to ensure industry standards compliance and a focused review of owner privileges for a secure token governance framework.

Safety Overview

Passed

Failed

Analysis

Results

Contract Verified

Yes

External Call Risk

No

Proxy Contract

No

Self Destruct

No

Antibot

No

Antiwhale

No

Owner Privileges

Safety Overview

Passed

Failed

Analysis

Results

Ownership renounced

Yes

No regain ownership function

No

No hidden mint function

No

Modifiable taxes

No

Modifiable max amount or max transaction

No

No high amount owner token

No

Smart Contract Weakness Classification Security Analysis

Detected vulnerabilities during our security audit scan of the smart contract. Our assessment includes thorough SWC-Registry checks and overflow assessments.

Severity High Medium Low Passed

Id	Name	Error	Result
SWC-100	State Variable Default Visibility		Passed
SWC-101	Code With No Effects		Passed
SWC-102	Unencrypted Private Data On-Chain		Passed
SWC-103	Floating Pragma		Passed
SWC-104	Unchecked Call Return Value		Passed
SWC-105	Unprotected Ether Withdrawal		Passed
SWC-106	Unprotected SELFDESTRUCT		Passed
SWC-107	Reentrancy		Passed
SWC-108	State Variable Default Visibility		Passed
SWC-109	Uninitialized Storage Pointer		Passed
SWC-110	Assert Violation		Passed

Severity High Medium Low Passed

Id	Name	Error	Result
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		Passed
SWC-121	Missing Protection against Signature Replay Attacks		Passed
SWC-122	Lack of Proper Signature Verification		Passed
SWC-123	Requirement Violation		Passed

Severity High Medium Low Passed

Id	Name	Error	Result
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		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

Community Trust

Our commitment to community trust extends to a thorough evaluation of social media channels, ensuring active engagement and responsiveness. Investor relations and partnerships are also scrutinized for credibility and transparency.

Social Media Presence

Website Health	Medium
Twitter	High Activity
Telegram	High Activity
Youtube	Not Available
Discord	Not Available
	Active Community
+230 Followers	
+500 Subscribers	

Listings and Partnerships

Listings	Not Available
Partnerships	Not Available

Audit Summary

Throughout the audit process, our team of auditors and smart contract developers conducted a cautious evaluation to identify security-related issues, code quality, and adherence to specification and best practices.

The smart contract does not contain high severity issues.

Security Score : 93%

Audit Methodology

Our smart contract audit methodology is designed to provide a thorough and comprehensive analysis of any contract code. We believe that a collaborative and multi-faceted approach is the best way to ensure the reliability and accuracy of our results.

Our audits are performed by experienced blockchain developers and security experts who work independently to provide a more comprehensive and accurate assessment. The auditors begin by reading the contract code to understand its structure and purpose, and examining the functional and technical requirements as well as other relevant documents provided by the client.

Next, automated tools are used in a controlled environment to search for security vulnerabilities and best practices. These tools can help us identify potential vulnerabilities, such as Smart Contract Weakness vulnerabilities, and provide a more complete picture of the contract's functionality.

Data flow diagrams are generated to visualise all possible states and interactions with other contracts. This allows the auditors to track changes in data and funds flow and identify any potential issues or risks.

Line-by-line review of the contract code is conducted to check for hidden malicious code or other security risks. This meticulous and thorough review is essential for protecting our clients' interests and ensuring the success of their projects.

Once the audit is complete, we provide a comprehensive report on our findings and recommendations. Our methodology is standardised to ensure consistent and reliable results, and we are committed to providing our clients with the information they need to improve the security and functionality of their contracts. By conducting a thorough and comprehensive audit, we can help our clients protect their interests and ensure the success of their projects.

Disclaimer

This audit report is intended to provide a comprehensive analysis of the contract code and its potential vulnerabilities. The findings and recommendations in this report are based on our best efforts and knowledge at the time of the audit, but they are not guaranteed to be complete or accurate.

All smart contract audits performed by us are provided for informational purposes only, and should not be considered legal or financial advice. We do not guarantee the security or functionality of any smart contracts that we audit, and we shall not be liable for any losses or damages arising from the use of any audited smart contracts.

We recommend that our clients carefully review the report and consider its findings and recommendations, but they should not rely on it as the sole basis for making any decisions regarding the contract. We encourage our clients to conduct their own independent analysis and seek additional advice as needed. This report is provided for informational purposes only, and it does not constitute legal, financial, or other professional advice. It does not create any warranties, representations, or guarantees, and it does not establish any legal or contractual obligations.

By accepting this report, our clients acknowledge and agree to the terms of this disclaimer. We appreciate the opportunity to conduct the audit and provide our findings, and we are committed to assisting our clients in any way we can.