SafePal Token

Smart Contract Audit Report

SYSFIXED

17 Dec 2020

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
SafePal Token	SFP	Binance Smart Chain

Addresses

Contract address	0xd41fdb03ba84762dd66a0af1a6c8540ff1ba5dfb
Contract deployer address	0xBAc93b5b19FeC3D8Da65A81bBf79F23D33A50a2D

Project Website

https://www.safepal.com/

Codebase

https://bscscan.com/address/0xd41fdb03ba84762dd66a0af1a6c8540ff1ba5dfb#code

SUMMARY

Although blockchain promises to cut out the middleman and boost security, crypto still has its gatekeepers: Long learnir

Contract Summary

Documentation Quality

SafePal Token provides a very good documentation with standard of solidity base code.

- The technical description is provided clearly and structured and also dont have any high risk issue.

Code Quality

The Overall quality of the basecode is standard.

- Standard solidity basecode and rules are already followed by SafePal Token with the discovery of several low issu

Test Coverage

Test coverage of the project is 100% (Through Codebase)

Audit Findings Summary

- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 7.

CONCLUSION

We have audited the SafePal Token project released on December 2020 to discover issues and identify potential securit

The security audit report provides satisfactory results with low-risk issues.

The issues found in the SafePal Token smart contract code do not pose a considerable risk. The writing of the contract

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels	PASS
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows	PASS
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that t	ISSUE FOUND
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
Unprotected Ether Withdrawal	SWC-105	Due to missing or insufficient access controls, malicious parties can withdraw	PASS
SELFDESTRUCT Instruction	SWC-106	The contract should not be self-destructible while it has funds belonging to us	PASS
Reentrancy	SWC-107	Check effect interaction pattern should be followed if the code performs recu	PASS
Uninitialized Storage Pointer	SWC-109	Uninitialized local storage variables can point to unexpected storage locations	PASS
Assert Violation	SWC-110 SWC-123	Properly functioning code should never reach a failing assert statement.	PASS
Deprecated Solidity Functions	SWC-111	Deprecated built-in functions should never be used.	PASS
Delegate call to Untrusted Callee	SWC-112	Delegatecalls should only be allowed to trusted addresses.	PASS

DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a spec	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency sh	PASS
Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	PASS
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A tran	PASS
Incorrect Constructor Name	SWC-118	Constructors are special functions that are called only o	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain A	PASS
Write to Arbitrary Storage Location	SWC-124	The contract is responsible for ensuring that only author	PASS
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they hav	PASS
Insufficient Gas Griefing	SWC-126	Insufficient gas griefing attacks can be performed on co	PASS
Arbitrary Jump Function	SWC-127	As Solidity doesnt support pointer arithmetics, it is impo	PASS

Typographical Error	SWC-129	A typographical error can occur for example when the	PASS
Override control character	SWC-130	Malicious actors can use the Right-To-Left-Override un	PASS
Unused variables	SWC-131 SWC-135	Unused variables are allowed in Solidity and they do no	PASS
Unexpected Ether balance	SWC-132	Contracts can behave erroneously when they strictly a	PASS
Hash Collisions Variable	SWC-133	Using abi.encodePacked() with multiple variable length	PASS
Hardcoded gas amount	SWC-134	The transfer() and send() functions forward a fixed am	PASS
Unencrypted Private Data	SWC-136	It is a common misconception that private type variabl	PASS

SMART CONTRACT ANALYSIS

Started	Wednesday Dec 16 2020 23:20:36 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Dec 17 2020 14:18:26 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	SafePalToken.sol

Detected Issues

ID	Title	Severity	Status
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 7

low SEVERITY

The current pragma Solidity directive is "">=0.6.0<0.8.0"". It is recommended to specify a fixed compiler version to ensur

Source File

- SafePalToken.sol

Locations

```
6
7   pragma solidity >=0.6.0 <0.8.0;
8
9   /*
10   * @dev Provides information about the current execution context, including the
11
```

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, di

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This re

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date o

This report should not be used in any way to make decisions around investment or involvement with any particular proje

This report is provided for information purposes only and on a non-reliance basis and does not constitute investment ad

ABOUT US

Sysfixed is a blockchain security certification organization established in 2021 with the objective to provide smart contra