CFG NINJA AUDITS

Security Assessment

Saiyan Doge Token

October 23, 2023

Audit Status: Pass

Audit Edition: Pinksale

POWERED BY BLADE POOL

Risk Analysis

Classifications of Manual Risk Results

Classification	Description
Critical	Danger or Potential Problems.
High	Be Careful or Fail test.
Low	Pass, Not-Detected or Safe Item.
Informational	Function Detected

Manual Code Review Risk Results

Contract Priviledge	Description
Buy Tax	4%
Sale Tax	4%
Cannot Sale	Pass
Cannot Sale	Pass
Max Tax	25
Modify Tax	Yes
Fee Check	Pass
Is Honeypot?	Not Detected
Trading Cooldown	Not Detected
Can Pause Trade?	Pass

Contract Priviledge	Description
Pause Transfer?	Not Detected
Max Tx?	Pass
Is Anti Whale?	Not Detected
Is Anti Bot?	Not Detected
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Not Detected
Can Mint?	Pass
Is Proxy?	Not Detected
Can Take Ownership?	Not Detected
Hidden Owner?	Not Detected
Owner	0x69642B4a44699e9a4bbc4F9f7d046BAff0599553
Self Destruct?	Not Detected
External Call?	Not Detected
Other?	Detected
Holders	1
Auditor Confidence	Low
KYC Completed	No

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Project Overview

Token Summary

Parameter	Result
Address	0x036eF1b2c7c597AB246443Cfd566F6BaC85f1D0e
Name	Saiyan Doge
Token Tracker	Saiyan Doge (SDoge)
Decimals	18
Supply	55,555
Platform	Binance Smart Chain
compiler	v0.8.4+commit.c7e474f2
Contract Name	BABYTOKEN
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/token/0x036eF1b2c7c597AB246443Cfd566F6BaC85f1D0e#code
Payment Tx	Corporate

Main Contract Assessed
Contract Name

Name	Contract	Live
Saiyan Doge	0x036eF1b2c7c597AB246443Cfd566F6BaC85f1D0e	Yes

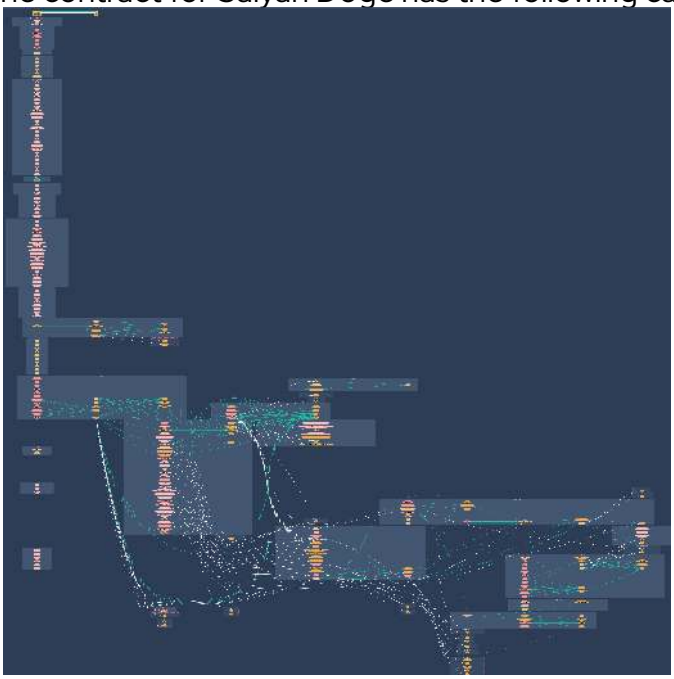
TestNet Contract was Not Assessed

Solidity Code Provided

SolID	File Sha-1	FileName
BBT	a7c274a751395ecc183d558f9ed0b5422a385fc4	BabyToken.sol

Call Graph

The contract for Saiyan Doge has the following call graph structure.



Smart Contract Vulnerability Checks

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) while overlaying a wide range of weakness variants that are specific to smart contracts.

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	BabyToken.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	BabyToken.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	BabyToken.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	BabyToken.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	BabyToken.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	BabyToken.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	BabyToken.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	BabyToken.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	BabyToken.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	BabyToken.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	BabyToken.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-111	Pass	Use of Deprecated Solidity Functions.	BabyToken.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	BabyToken.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	BabyToken.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	BabyToken.sol	L: 0 C: 0
SWC-115	low	Authorization through tx.origin.	BabyToken.sol	L: 3123 C: 12,L: 3223 C: 20
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	BabyToken.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	BabyToken.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	BabyToken.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	BabyToken.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	BabyToken.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	BabyToken.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	BabyToken.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	BabyToken.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	BabyToken.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-125	Pass	Incorrect Inheritance Order.	BabyToken.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	BabyToken.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	BabyToken.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	BabyToken.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	BabyToken.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	BabyToken.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	BabyToken.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	BabyToken.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	BabyToken.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	BabyToken.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	BabyToken.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	BabyToken.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Smart Contract Vulnerability Details

SWC-115 - Authorization through tx.origin

CWE-477: Use of Obsolete Function

Description:

tx.origin is a global variable in Solidity which returns the address of the account that sent the transaction. Using the variable for authorization could make a contract vulnerable if an authorized account calls into a malicious contract. A call could be made to the vulnerable contract that passes the authorization check since tx.origin returns the original sender of the transaction which in this case is the authorized account.

Remediation:

tx.origin should not be used for authorization. Use msg.sender instead.

References:

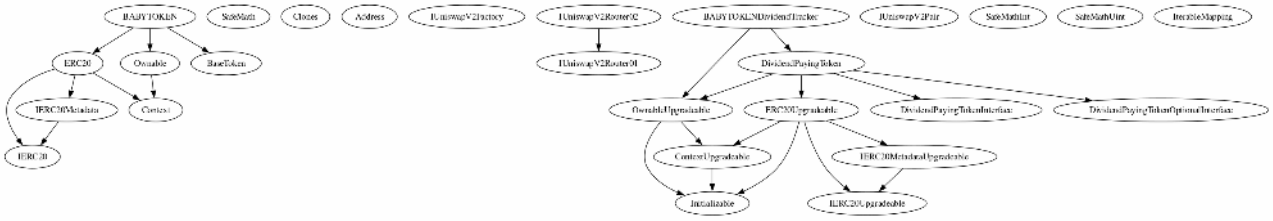
Solidity Documentation - tx.origin

Ethereum Smart Contract Best Practices - Avoid using tx.origin

SigmaPrime - Visibility.

Inheritance

The contract for Saiyan Doge has the following inheritance structure.



Smart Contract Advance Checks

ID	Severity	Name	Result	Status
SDoge-01	Low	Potential Sandwich Attacks.	Pass	Not Detected
SDoge-02	Informational	Function Visibility Optimization	Pass	Not Detected
SDoge-03	Low	Lack of Input Validation.	Pass	Not Detected
SDoge-04	High	Centralized Risk In addLiquidity.	Pass	Not Detected
SDoge-05	Low	Missing Event Emission.	Pass	Not Detected
SDoge-06	Low	Conformance with Solidity Naming Conventions.	Pass	Not Detected
SDoge-07	Low	State Variables could be Declared Constant.	Pass	Not Detected
SDoge-08	Low	Dead Code Elimination.	Pass	Not Detected
SDoge-09	High	Third Party Dependencies.	Pass	Not Detected
SDoge-10	High	Initial Token Distribution.	Pass	Not Detected
SDoge-11	High	claimStuckTokens can claim own tokens.	Pass	Not Detected
SDoge-12	High	Centralization Risks In The X Role	Pass	Not Detected
SDoge-13	Informational	Extra Gas Cost For User..	Pass	Not Detected
SDoge-14	Medium	Unnecessary Use Of SafeMath	Pass	Not Detected
SDoge-15	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not Detected

ID	Severity	Name	Result	Status
SDoge-16	Medium	Taxes can be up to 100%	Pass	Not Detected
SDoge-17	Logical Issue	Highly Permissive Role Access.,`	Pass	Not Detected
SDoge-18	Critical	Stop Transactions by using Enable Trade.	Pass	Not Detected

Technical Findings Summary

Classification of Risk

Severity	Description
Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
High	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
Low	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
Critical	0	0	0
High	0	0	0
Medium	0	0	0
Low	0	0	0
Informational	0	0	0
Total	0	0	0

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/SaiyanDoge_bsc	Pass
Other		Fail
Website	https://saiyandoge.xyz/	Pass
Telegram	https://t.me/saiyandoge_bsc	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Assessment Results

Score Results

Review	Score
Overall Score	89/100
Auditor Score	80/100
Review by Section	**Score**
Manual Scan Score	18/33
SWC Scan Score	35 /37
Advance Check Score	36 /30

The Following Score System Has been Added to this page to help understand the value of the audit, the maximun score is 100, however to attain that value the project most pass and provide all the data needed for the assessment. Our Passing Score has been changed to 80 Points, if a project does not attain 80% is an automatic failure. Read our notes and final assessment below.

Audit Passed

Assessment Results

Important Notes:

- No issues or vulnerabilities were found.
- This is a Pinksale Generated BabyToken.
- Please DYOR on the project.
- The contract gives Ethereum https://bscscan.com/address/0x2170ed0880ac9a755fd29b2688956bd959f933f8
- this contract depends on volume and buy/sale then distribution of rewards may happen.
- This type of contract may fail if the fees are set to 0.

Auditor Score =80
Audit Passed

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

CFGNINJA has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided 'as is, and CFGNINJA is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will CFGNINJA or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by CFGNINJA are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.