



CFG NINJA AUDITS

Security Assessment

GrowToken Token

October 29, 2023

Audit Status: Pass

Audit Edition: Advance



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	8%
🟢 Modify Tax	Pass
🟢 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?	Not-Detected
Is Proxy?	Not Detected
Can Take Ownership?	Not Detected
Hidden Owner?	Not Detected
Owner	
Self Destruct?	Not Detected
External Call?	Not Detected
Other?	Not Detected
Holders	1
Auditor Confidence	High
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	
Name	GrowToken
Token Tracker	GrowToken (NGR)
Decimals	18
Supply	
Platform	Ethereum
compiler	v0.8.19+commit.7dd6d404
Contract Name	NextGenROI
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://
Payment Tx	Corporate





Main Contract Assessed

Contract Name

Name	Contract	Live
GrowToken		No

TestNet Contract was Not Assessed

Solidity Code Provided

SolidID	File Sha-1	FileName
GrowToken	5062758fbb002019f5bf826f03945480b4b32b5e	GrowToken.sol



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	GrowToken.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	GrowToken.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	GrowToken.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	GrowToken.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	GrowToken.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	GrowToken.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	GrowToken.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	GrowToken.sol	L: 0 C: 0
SWC-108	Low	State variable visibility is not set..	GrowToken.sol	L: 892 C: 9
SWC-109	Pass	Uninitialized Storage Pointer.	GrowToken.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	GrowToken.sol	L: 0 C: 0





ID	Severity	Name	File	location
SWC-111	Pass	Use of Deprecated Solidity Functions.	GrowToken.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	GrowToken.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	GrowToken.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	GrowToken.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	GrowToken.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	GrowToken.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	GrowToken.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	GrowToken.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	GrowToken.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randomness.	GrowToken.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	GrowToken.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	GrowToken.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	GrowToken.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	GrowToken.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	GrowToken.sol	L: 0 C: 0



ID	Severity	Name	File	location
SWC-126	Pass	Insufficient Gas Griefing.	GrowToken.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	GrowToken.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	GrowToken.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	GrowToken.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	GrowToken.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	GrowToken.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	GrowToken.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	GrowToken.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	GrowToken.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	GrowToken.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	GrowToken.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-108 - State Variable Default Visibility

CWE-710: Improper Adherence to Coding Standards

Description:

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

Remediation:

Variables can be specified as being public, internal or private. Explicitly define visibility for all state variables.

References:

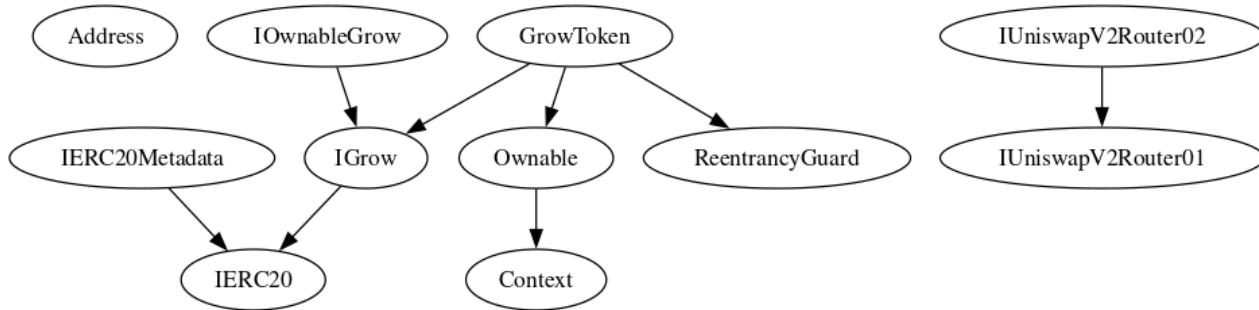
Ethereum Smart Contract Best Practices - Explicitly mark visibility in functions and state variables





Inheritance

The contract for GrowToken has the following inheritance structure.



Privileged Functions (onlyOwner)

Please Note if the contract is Renounced none of this functions can be executed.

Function Name	Parameters	Visibility
renounceOwnership		Public
transferOwnership	address newOwner	Public
ActivateToken		External
setFeeExemption	address Contract, bool exempt	External
setExecutorAddress	address executorAddress, bool exempt	External
updateShares	uint256 newDevShare, uint256 newLiquidityShare	External
updateDevAddress	address newDev	External
updateFees	uint256 newSellFee, uint256 newMintFee, uint256 newTransferFee	External
setStableToken	address _stable,	



bool _accept



Public



exchangeTokens



address_from,
address_to,
address_router



External



Smart Contract Advance Checks

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



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



NGR-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	● Low	GrowToken.sol: L: 1529 C: 14	🕒 Detected

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes. The linked code does not create an event for the transfer.

Remediation

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.



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.
ℹ️ 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	1	0	0
ℹ️ Informational	0	0	0
Total	1	0	0



Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/nextgenroi	Pass
Other		Fail
Website		Fail
Telegram	https://t.me/nextgenroi	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

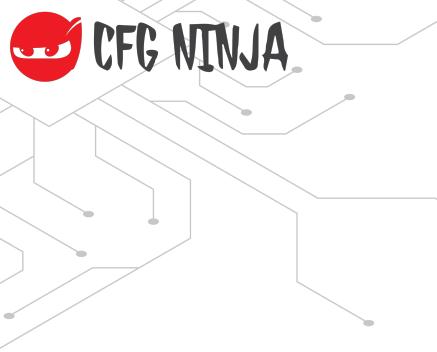
Score Results

Review	Score
Overall Score	100/100
Auditor Score	95/100
Review by Section	Score
Manual Scan Score	48/33
SWC Scan Score	35/37
Advance Check Score	35/30

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must 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:

- Token developed by Semi.

Auditor Score =95

Audit Passed



Appendix

Finding Categories

Centralization / Privilege

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

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM 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 how `block.timestamp` works.

Control Flow

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

Volatile Code

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

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the 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 setter function.



Coding Best Practices

ERC 20 Coding Standards are a set of rules that each developer should follow to ensure the code meet a set of criterias 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 advocacy 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 depreciation 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.

