Security Assessment:
Fort Block Games Token

January 30, 2024

- Audit Status: **Pass**
- Audit Edition: **Standard**

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 Privilege	Description
🟠 Buy Tax	40%
🟠 Sale Tax	40%
🟢 Cannot Sale	Pass
🟢 Cannot Sale	Pass
🟢 Max Tax	5%
🟡 Modify Tax	Yes
🟢 Fee Check	Pass
🟢 Is Honeypot?	Not Detected
🟢 Trading Cooldown	Not Detected
🟢 Can Pause Trade?	Pass
🟢 Pause Transfer?	Not Detected
🟢 Max Tx?	Pass
🟢 Is Anti Whale?	Not Detected
🟢 Is Anti Bot?	Not Detected

Contract Privilege	Description
🟢 Is Blacklist?	Not Detected
🟢 Blacklist Check	Pass
🟢 is Whitelist?	Detected
🟢 Can Mint?	Pass
🟢 Is Proxy?	Not Detected
🟢 Can Take Ownership?	Not Detected
🟢 Hidden Owner?	Not Detected
ℹ️ Owner	0x0580Fc95d0531E8e45Cfa30b910cB4cc5374540b
🟢 Self Destruct?	Not Detected
🟢 External Call?	Detected
🟢 Other?	Not Detected
🟢 Holders	1
🟢 Auditor Confidence	High
🟡 KYC Present	Yes
🟡 KYC URL	https://assuredefi.com/projects/fort-block-games/

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	Fort Block Games
Token Tracker	Fort Block Games (FGB)
Decimals	18
Supply	10,000,000
Platform	ETHEREUM
compiler	0.8.19+commit.7dd6d404
Contract Name	FGB
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	
Payment Tx	Corporate

Main Contract Assessed
Contract Name

Name	Contract	Live
Fort Block Games		No

TestNet Contract Assessed
Contract Name

Name	Contract	Live
Fort Block Games	0x4298C180d1608dCd2B1b10962d09903F6Cb52119	No

Solidity Code Provided

SolID	File Sha-1	FileName
fgbv2	a287c23fc2a1b96832a69cf0f4e33c729a18049a	fgbv2.sol
fgbv2		
fgbv2		
fgbv2		
fgbv2		
fgbv2	undefined	

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	fgbv2.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	fgbv2.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	fgbv2.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	fgbv2.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	fgbv2.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	fgbv2.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	fgbv2.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	fgbv2.sol	L: 0 C: 0
SWC-108	Low	State variable visibility is not set..	fgbv2.sol	L: 82 C: 12
SWC-109	Pass	Uninitialized Storage Pointer.	fgbv2.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	fgbv2.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	fgbv2.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	fgbv2.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	fgbv2.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	fgbv2.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-115	Pass	Authorization through tx.origin.	fgbv2.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	fgbv2.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	fgbv2.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	fgbv2.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	fgbv2.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	fgbv2.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	fgbv2.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	fgbv2.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	fgbv2.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	fgbv2.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	fgbv2.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	fgbv2.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	fgbv2.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	fgbv2.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	fgbv2.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	fgbv2.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	fgbv2.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	fgbv2.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	fgbv2.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	fgbv2.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	fgbv2.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	fgbv2.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 Fort Block Games has the following inheritance structure.

The Project has a Total Supply of 10,000,000
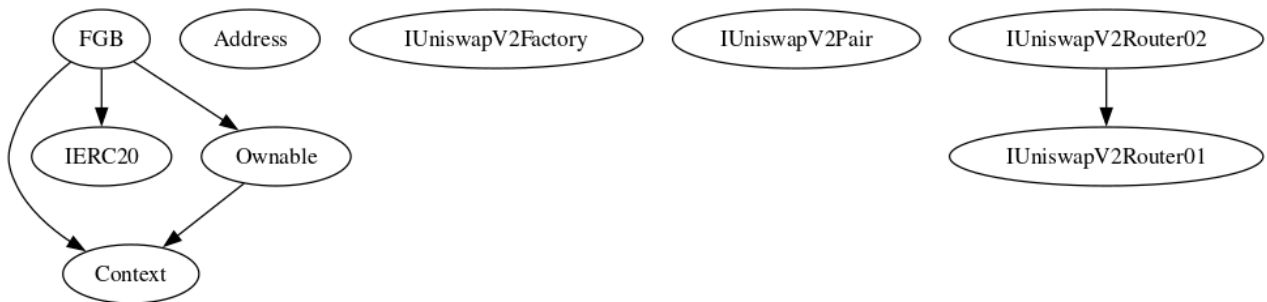
FGB-10 | Initial Token Distribution.

Category	Severity	Location	Status
Centralization / Privilege	🟠 High	fgbv2.sol: L: 811 C: 14	📄 Detected

Description

All of the Fort Block Games tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute tokens without obtaining the consensus of the community.

Remediation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

Project Action

emit Transfer(address(0), _msgSender(), _tTotal);

FGB-11 | Initial Tax is 40% and then lowered to 5%.

Category	Severity	Location	Status
Optimization	🟢 Low	fgbv2.sol: L: 05 C: 14	Detected

Description

Initial Tax 40% and then lowered to 5% using setEndTime.

Remediation

We recommend a lower tax at initial.

Project Action

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	1	0	0
🟡 Medium	0	0	0
🔶 Low	1	1	0
ⓘ Informational	0	0	0
Total	2	2	0

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/FortBlockGames	Pass
Other		N/A
Website	https://fortblockgames.com/	Pass
Telegram	https://t.me/+UXApDkYgDK1mMjFh	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:

ASSURE
DEFI

OFFICIAL PARTNER

Assessment Results

Score Results

Review	Score
Overall Score	89/100
Auditor Score	85/100
Review by Section	Score
Manual Scan Score	22
SWC Scan Score	35
Advance Check Score	32

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 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below.

Audit Passed

Assessment Results

Important Notes:

- Code audited and tested in bsc testnet.ı

- Always DYOR.

Auditor Score =85
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

Assure Defi 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 Assure Defi is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will Assure Defi 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 Assure Defi 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.

ASSURE DEFI™
THE VERIFICATION **GOLD STANDARD**