CFG NINJA AUDITS

Security Assessment

HYPERFLXTOKEN

Token

August 17, 2023

Audit Status: Pass with KYC

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	2%
Sale Tax	2%
Cannot Sale	Pass
Cannot Sale	Pass
Max Tax	10%
Modify Tax	Yes
Fee Check	Pass
Is Honeypot?	Not Detected
Trading Cooldown	Not Detected
Can Pause Trade?	Owner need to enable trade.

Contract Priviledge	Description
Pause Transfer?	Detected
Max Tx?	Fail
Is Anti Whale?	Detected
Is Anti Bot?	Detected
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Detected
Can Mint?	Pass
Is Proxy?	Not Detected
Can Take Ownership?	Not Detected
Hidden Owner?	Detected, safeManager can withdraw tokens and liquidy.
Owner	0x8e210528fCF50e538bfeD58af32c21384a7240ab
Self Destruct?	Not Detected
External Call?	Not Detected
Other?	Not Detected
Holders	1
Auditor Confidence	Low

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	0xC574f9f7DE8B07ac7594B195907d3A92124141b4
Name	HYPERFLXTOKEN
Token Tracker	HYPERFLXTOKEN (HYFX)
Decimals	18
Supply	1,000,000,000,000
Platform	Ethereum
compiler	v0.8.19+commit.7dd6d404
Contract Name	HYPERFLXTOKEN
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://etherscan.io/address/0xC574f9f7DE8B07ac7594B195907d3A92124141b4#code
Payment Tx	0x4caf6ecfce7c566c9ca2672987205e909f32f76ad6890d20fc9ba312ef5eec87

Main Contract Assessed Contract Name

Name	Contract	Live
HYPERFLXTOKEN	0xC574f9f7DE8B07ac7594B195907d3A92124141b4	Yes

TestNet Contract Assessed Contract Name

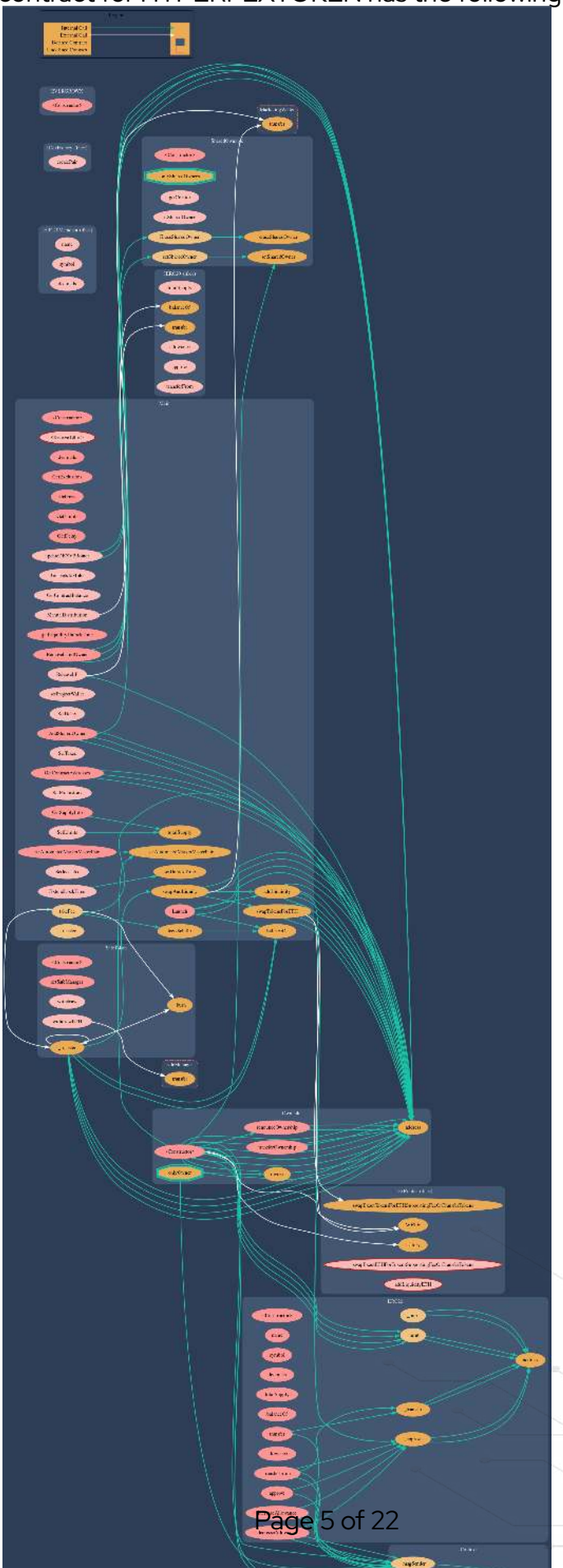
Name	Contract	Live
HYPERFLXTOKEN	0x38D44b6b1F16963bB13CEE257847EC80EAC0A0c2	Yes

Solidity Code Provided

SolID	File Sha-1	FileName
Hyperflux	b38149b59029281b94b30fa61fd7f1350b680423	hyperflux.sol
Hyperflux		
Hyperflux		
Hyperflux		
Hyperflux		
Hyperflux		

Call Graph

The contract for HYPERFLXTOKEN 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	hyperflux.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	hyperflux.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	hyperflux.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	hyperflux.sol	L: 10 C: 0
SWC-104	Pass	Unchecked Call Return Value.	hyperflux.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	hyperflux.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	hyperflux.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	hyperflux.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	hyperflux.sol	L: 170 C: 20
SWC-109	Pass	Uninitialized Storage Pointer.	hyperflux.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	hyperflux.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-111	Pass	Use of Deprecated Solidity Functions.	hyperflux.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	hyperflux.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	hyperflux.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	hyperflux.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	hyperflux.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	hyperflux.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	hyperflux.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	hyperflux.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	hyperflux.sol	L: 0 C: 0
SWC-120	Low	Potential use of block.number as source of randonmness.	hyperflux.sol	L: 425 C: 22, L: 485 C: 34,L: 528 C: 53,L: 529 C: 33
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	hyperflux.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	hyperflux.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	hyperflux.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-124	Pass	Write to Arbitrary Storage Location.	hyperflux.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	hyperflux.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	hyperflux.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	hyperflux.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	hyperflux.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	hyperflux.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	hyperflux.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	hyperflux.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	hyperflux.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	hyperflux.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	hyperflux.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	hyperflux.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	hyperflux.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-120 - Weak Sources of Randomness from Chain Attributes

CWE-330: Use of Insufficiently Random Values

Description:

Solidity allows for ambiguous naming of state variables when inheritance is used. Contract A with a variable x could inherit contract B that also has a state variable x defined. This would result in two separate versions of x, one of them being accessed from contract A and the other one from contract B. In more complex contract systems this condition could go unnoticed and subsequently lead to security issues.

Shadowing state variables can also occur within a single contract when there are multiple definitions on the contract and function level.

Remediation:

Using commitment scheme, e.g. RANDAO. Using external sources of randomness via oracles, e.g. Oraclize. Note that this approach requires trusting in oracle, thus it may be reasonable to use multiple oracles. Using Bitcoin block hashes, as they are more expensive to mine.

References:

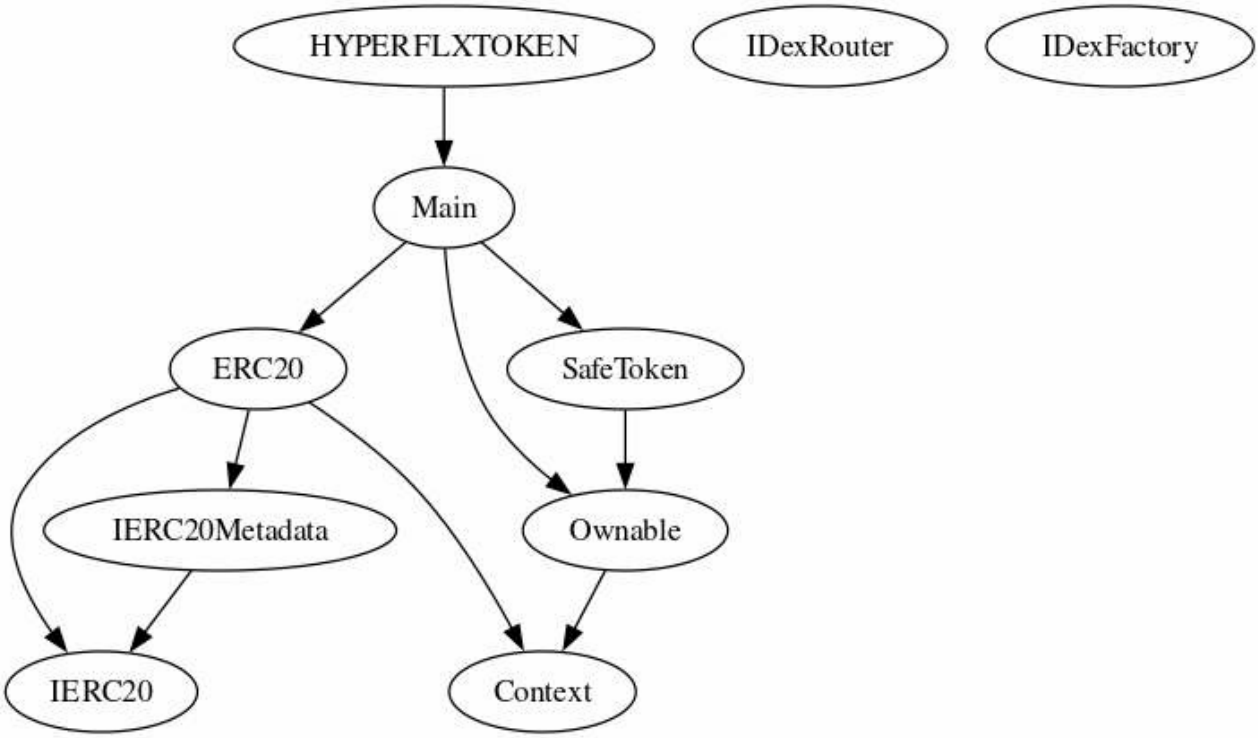
How can I securely generate a random number in my smart contract?)

When can BLOCKHASH be safely used for a random number? When would it be unsafe?

The Run smart contract.

Inheritance

The contract for HYPERFLXTOKEN has the following inheritance structure.



Smart Contract Advance Checks

ID	Severity	Name	Result	Status
HYFX-01	Low	Potential Sandwich Attacks.	Pass	Not-Found
HYFX-02	Informational	Function Visibility Optimization	Pass	Resolved
HYFX-03	Low	Lack of Input Validation.	Pass	Resolved
HYFX-04	High	Centralized Risk In addLiquidity.	Fail	Detected
HYFX-05	Low	Missing Event Emission.	Pass	Resolved
HYFX-06	Low	Conformance with Solidity Naming Conventions.	Pass	Not-Found
HYFX-07	Low	State Variables could be Declared Constant.	Pass	Not-Found
HYFX-08	Low	Dead Code Elimination.	Pass	Not-Found
HYFX-09	High	Third Party Dependencies.	Pass	Not Detected
HYFX-10	High	Initial Token Distribution.	Pass	Not-Found
HYFX-11	High	A special ownership library is used for this contract named SharedOwnable.	Pass	Not Detected
HYFX-12	High	Centralization Risks In The X Role	Pass	Not-Found
HYFX-13	Informational	Extra Gas Cost For User..	Pass	Not Detected
HYFX-14	Medium	Unnecessary Use Of SafeMath	Pass	Not Detected

ID	Severity	Name	Result	Status
HYFX-15	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not Detected
HYFX-16	Medium	Taxes can be up to 100%	Pass	Not Detected
HYFX-17	Logical Issue	Highly Permissive Role Access.,`	Pass	Not Detected
HYFX-18	Critical	Stop Transactions by using Enable Trade.	Fail	Detected

HYFX-04 | Centralized Risk In addLiquidity.

Category	Severity	Location	Status
Coding Style	High	hyperflux.sol: L: 573 C: 14	Detected

Description

uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this), tokenAmount, 0, 0, owner(), block.timestamp);

The addLiquidity function calls the uniswapV2Router.addLiquidityETH function with the to address specified as owner() for acquiring the generated LP tokens from the HYFX-WBNB pool.

As a result, over time the _owner address will accumulate a significant portion of LP tokens.If the _owner is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Remediation

We advise the to address of the uniswapV2Router.addLiquidityETH function call to be replaced by the contract itself, i.e. address(this) , and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the _owner account is compromised. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e. Multisignature wallets.

1. Indicatively, here are some feasible solutions that would also mitigate the potential risk:

2. Time-lock with reasonable latency, i.e. 48 hours, for awareness on privileged operations;

3. Assignment of privileged roles to multi-signature wallets to prevent single point of failure due to the private key;

Introduction of a DAO / governance / voting module to increase transparency and user involvement

Project Action

Liquidity get added to the contract, and this can be recovered with the Withdrawal function.

HYFX-18 | Stop Transactions by using Enable Trade.

Category	Severity	Location	Status
Logical Issue	Critical	hyperflux.sol: L: 414 C: 14	Detected

Description

Enable Trade is presend on the following contract and when combined with Exclude from fees it can be considered a whitelist process, this will allow anyone to trade before others and can represent and issue for the holders.

Remediation

We recommend the project owner to carefully review this function and avoid problems when performing both actions.

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	1	0	0
High	1	0	0
Medium	0	0	0
Low	0	0	2
Informational	0	0	0
Total	2	0	2

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/hyperflxtoken	Pass
Other	https://www.tiktok.com/@hyperflxtoken	Pass
Website	https://hyperflxtoken.com	Pass
Telegram	https://t.me/hyperflxtoken	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	80/100
Auditor Score	80/100
Review by Section	**Score**
Manual Scan Score	24/33
SWC Scan Score	35 /37
Advance Check Score	21 /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:

- The contract has a process to add liquidity to its own address, lock it and later can be withdrawn by safeManager.
- SafeManager can withdraw tokens and liquidy from the contract.
- The owner needs to enable trade or launch the contract.
- The customer will do kyc with pinksale.
- Please DYOR on the project.

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.