



\$HIIT

BSC

utility

**Contract:**

0x30fb.....4506bC

Issues**Found****Fixed**

- Critical
- Medium
- Minor
- Informational

0
2
0
3

0
0
0
0

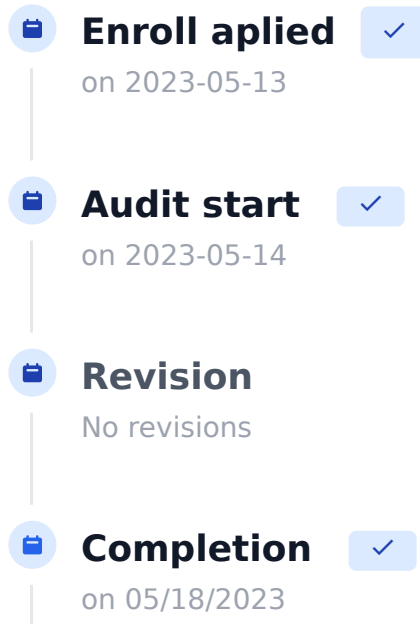
9.0/₁₀**AUDIT PASSED** **Project Description**

HIIT.finance is a decentralized application platform and utility token for the global fitness industry. HIIT is the utility token that is used for: - App based fitness rewards - B2B and B2C transactions within the fitness industry Example use cases include: -Earn HIIT wh

[Enroll](#)

you complete a workout in-app. This means you can literally get paid to train. -Have HIIT awarded to you (by a coach or gym) -As HIIT.finance grows, more gyms and personal trainers accept HIIT as payment

Project details



Language: **Solidity**

Contract address: **0x30...06bC**

Owner: **0x1f...a764**

SHA-256:
a74...2a0

Supply: **300,000,000**

Decimals: **18**

Deployer: **0x1f...a764**

Chain: **Binance Smart Chain**

Compiler: **v0.8.4+commit.c7e474f2**

Optimization: **Yes with 200 runs**

License: **Unlicensed**

Verified: **Yes**

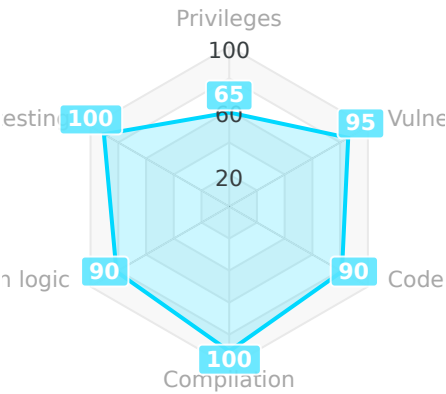
Contract type: **bep20**

Contract Name: **HIIT**

Enroll

Overview

Doge Inu 



Minting[?]

Can't mint new tokens

Blacklist[?]

Can't blacklist addresses

Ownership

Owned

Honeypot

Fees without limits

Buy fees[?]

Owner can set buy fees without limits

Sell fees[?]

Owner can set buy fees without limits

Proxy[?]

No proxy detected

Max tx amount[?]

Can set max tx value without a lower limit

Trading Cooldown

No cooldown

Anti Bot

Not present

Pause[?]

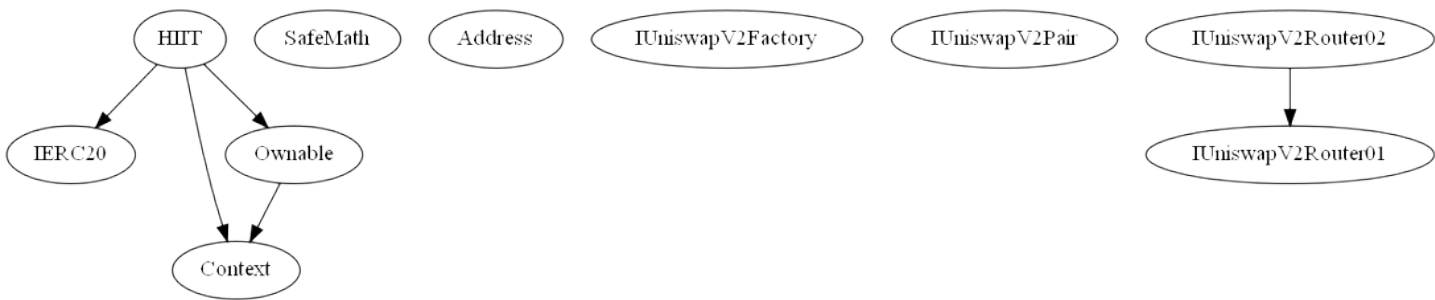
not present, but transactions can still be blocked through maxTx

Hidden Owner

No hidden owner

- Safe
- Caution advised
- Be careful
- Extreme caution

Inheritance



Contract details [0x30...06bC](#)

Contracts and libraries collapse

Contract	Type	
L	**Function Name**	
IERC20	Interface	
SafeMath	Library	
Context	Implementation	
Address	Library	
Ownable	Implementation	Context
IUniswapV2Pair	Interface	
IUniswapV2Router01	Interface	
IUniswapV2Router02	Interface	IUniswapV2Router01

| ****HIIT**** | Implementation | Context, IERC20, Ownable |||

▼ Modifiers [collapse](#)

Contract SafeMath

Function	Modifiers
tryAdd	[]
trySub	[]
tryMul	[]
tryDiv	[]
tryMod	[]
add	[]
sub	[]
mul	[]
div	[]
mod	[]
sub	[]

▼ Constructor calls [collapse](#)

```
#####
##### HIIT #####
#####
```

```
## Constructor Call Sequence
- Ownable
- HIIT
```

```
## Constructor Definitions
```

```
### Ownable
```

```
    constructor () {
        _owner = 0x1fe54b3dDBbf04612bd65c24c3464Ee47c1ca764;
        emit OwnershipTransferred(address(0), _owner);
    }
```

▼ Requires [collapse](#)

Contract SafeMath

Function	require or assert

	tryAdd			
	trySub			
	tryMul			
	tryDiv			
	tryMod			
	add			
	sub			
	mul			
	div			
	mod			
	sub		require(bool,string)(b <= a,errorMessage)	

Findings

High Severity

Medium Severity

✓ Centralization - maxTx value

Lines: #536-540

Description: The owner can set maxTx value without limits using setMaxTxPercent function.

Possible remediation: Add a require in setMaxTxPercent function and set a lower limit for _maxTxAmount, for example, 0.1% of the total supply.

✓ Fees without a limit

Lines: #527-535

Description: The owner can set buy and sell fees up to 100% with setTaxFeePercent, setDevelopmentFeePercent, and setLiquidityFeePercent functions. This increases the centralization risk.

Possible remediation: Add a totalFees variable and limit it to a max of 25%.

Low Severity

No low severity issues found

Informational

✓ Floating pragma

Lines: #7

Description: Contracts should be deployed with the same compiler version

Possible remediation: Lock the pragma version and also consider known bugs (<https://github.com/ethereum/solidity/releases>) for the compiler version that is chosen.

✓ State variable visibility is not set

Lines: #395

Description: The default visibility for "inSwapAndLiquify" is internal.

Possible remediation: It is best practice to set the visibility of state variables explicitly.

✓ Dead code

Lines: #20-87

Description: SafeMath is no longer needed starting with Solidity 0.8. The compiler now has built in overflow checking

Possible remediation: Remove the library

SWC Vulnerabilities

✓ 35/37 tests passed [collapse](#)

Code #	Name	Result	Severity
SWC-100	Function Default Visibility	Passed	n/a
SWC-101	Integer Overflow and Underflow	Passed	n/a
SWC-102	Outdated Compiler Version	Passed	n/a
SWC-103	Floating Pragma	Failed	Informational
SWC-104	Unchecked Call Return Value	Passed	n/a
SWC-105	Unprotected Ether Withdrawal	Passed	n/a
SWC-106	Unprotected SELF-DESTRUCT Instruction	Passed	n/a
SWC-107	Reentrancy	Passed	n/a
SWC-108	State Variable Default Visibility	Failed	Informational

Enroll

Code #	Name	Result	Severity
SWC-109	Uninitialized Storage Pointer	Passed	n/a
SWC-110	Assert Violation	Passed	n/a
SWC-111	Use of Deprecated Solidity Functions	Passed	n/a
SWC-112	Delegatecall to Untrusted Callee	Passed	n/a
SWC-113	DoS with Failed Call	Passed	n/a
SWC-114	Transaction Order Dependence	Passed	n/a
SWC-115	Authorization through tx.origin	Passed	n/a
SWC-116	Block values as a proxy for time	Passed	n/a
SWC-117	Signature Malleability	Passed	n/a
SWC-118	Incorrect Constructor Name	Passed	n/a
SWC-119	Shadowing State Variables	Passed	n/a
SWC-120	Weak Sources of Randomness from Chain Attributes	Passed	n/a
SWC-121	Missing Protection against Signature Replay Attacks	Passed	n/a
SWC-122	Lack of Proper Signature Verification	Passed	n/a
SWC-123	Requirement Violation	Passed	n/a
SWC-124	Write to Arbitrary Storage Location	Passed	n/a
SWC-125	Incorrect Inheritance Order	Passed	n/a
SWC-126	Insufficient Gas Griefing	Passed	n/a
SWC-127	Arbitrary Jump with Function Type Variable	Passed	n/a
SWC-128	DoS With Block Gas Limit	Passed	n/a
SWC-129	Typographical Error	Passed	n/a
SWC-130	Right-To-Left-Override control character (U+202E)	Passed	n/a
SWC-131	Presence of unused variables	Passed	n/a
SWC-132	Unexpected Ether balance	Passed	n/a
SWC-133	Hash Collisions With Multiple Variable Length Arguments	Passed	n/a
SWC-134	Message call with the hardcoded gas amount	Passed	n/a
SWC-135	Code With No Effects	Passed	n/a
SWC-136	Unencrypted Private Data On-Chain	Passed	n/a

Other findings

✓ Costly operations inside a loop

HIIT.includeInReward(address) (token.sol#498-509) has costly operations inside a loop: `_excluded.pop()` (token.sol#505. Incrementing state_variable in a loop incurs a lot of gas because of expensive SSTOREs, which might lead to an out-of-gas.

- Use a local variable to hold the loop computation result.

Additional information: [link](#)

✓ Variable should be constant

HIIT._decimals (token.sol#385), HIIT._developmentWalletAddress, HIIT._name (token.sol#383), HIIT._symbol, HIIT._tTotal, HIIT._tTotal, HIIT.numTokensSellToAddToLiquidity should be declared constant to save gas.

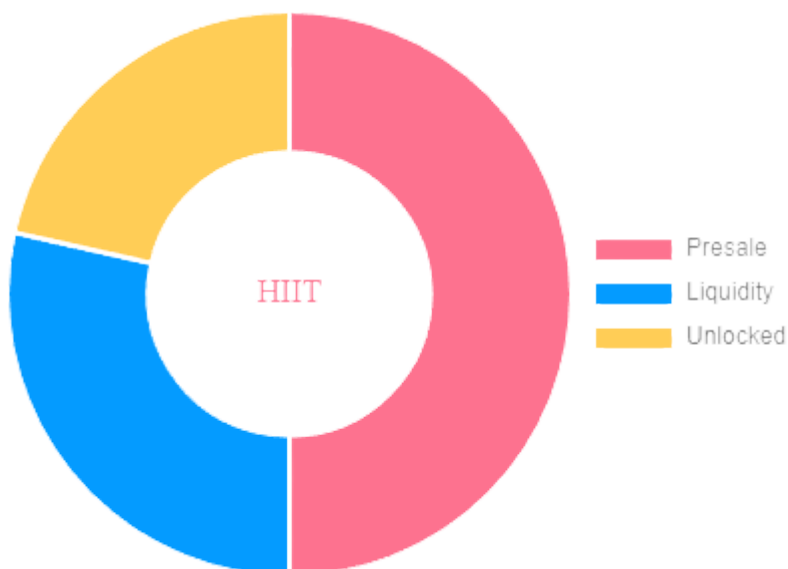
- Add the constant attribute to state variables that never change.

Additional information: [↔ link](#)

Holders stats (top 10)

0x73...b62a	78.50%
0x1f...a764	21.50%
0x00...0000 locked	0.00%

Presale

[View presale](#)

Soft Cap	100 BNB
Hard Cap	Fair Launch
Presale Start Time	2023.05.25 20:30 (UTC)
Presale End Time	2023.05.30 08:30 (UTC)
Liquidity Percent	60%

[Enroll](#)

KYC & DOXX

- **KYC**

✕ No information

- **DOXX**

✕ No information

Website audit <https://hiit.finance/>

- **Domain Registry**

GoDaddy.com

- **Domain Expiry Date**

2024-05-13

- **Response Code**

200

- **SSL Check and HTTPS test**

Yes, passed

- **HTTP2**

Yes, passed

- **Safe Browsing Test**

Yes, passed

Disclaimer

GuardGate is not a financial institution and the information provided on this website does not constitute investment advice, financial advice, trading advice, or any other sort of advice. You should not treat any of the website's content as such. Investing in crypto assets carries a high level of risk and does not hold guarantees for not sustaining

financial loss due to their volatility. Accuracy of Information GuardGate will strive to ensure the accuracy of the information listed on this website although it will not hold any responsibility for any missing or wrong information. GuardGate provides all information as is. You understand that you are using any and all information available here at your own risk. Any use or reliance on our content and services is solely at your own risk and discretion. The purpose of the audit is to analyze the on-chain smart contract source code and to provide a basic overview of the project. While we have used all the information available to us for this straightforward investigation, you should not rely on this report only - we recommend proceeding with several independent audits Be aware that smart contracts deployed on a blockchain aren't secured enough against external vulnerability or a hack. Be aware that active smart contract owner privileges constitute an elevated impact on the smart contract safety and security. Therefore, GuardGate does not guarantee the explicit security of the audited smart contract. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.



GuardGate

PRODUCTS

[Audit](#)

[Scanner](#)

[AMA](#) *coming soon

[Airdrop](#) *coming soon

COMPANY

[Pricing](#)

[About us](#)

[How it works](#)

[Enroll](#)

[Disclaimer](#)

STAY IN TOUCH



© 2023 GuardGate Co. All rights reserved.

Enroll