OXSCANS

DGI Game

AI Generated at 04:45 PM, UTC

February 10, 2024

OVERVIEW

This audit has been perpared for 'DGI Game' to review the main aspects of the project to help investors make an informative decision during their research process

You will find a summarized review of the following key points:

- Contract's source code
- Owner wallets
- Tokenomics
- Team transparency and goals
- Website's age, code, security and UX
- Whitepaper and roadmap
- Social media and online presence

Table of Content

General Information

DGI Game

DGI - Decentralized Gaming Income Token, represents a pioneering venture within the realm of gaming guilds, offering you the most effective way to generate passive income from gaming.

Name	**DGI Game**
Direction	**Ethereum Ecosystem**
Info	Website, Telegram Bot

General Information

Tokenomics

Ticker	DGI
Network	ethereum
Contract Address	0xe453c3409f8ad2b1fe1ed08e189634d359705a5b

General Analysis

Audit Review Process

1. Testing the smart contracts against both common and uncommon vulnerabilities
2. Assessing the codebase to ensure compliance with current best practices and industry standards
3. Ensuring contract logic meets the specifications and intentions of the client
4. Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
5. Thorough line-byline AI review of the entire codebase by industry

Token Transfer Stats

Transactions (Latest Mine Block)	Token holders	Compiler
1	1274	v0.8.17

Smart Contract Stats

Functions	Events	Constructor
28	12	1

Detail Analysis

Threat Level

High	Issues on this level are critical to the smart contract's performace/functionality and should be fixed before moving to a live enviroment
Medium	Issues on this level are critical to the smart contract's performace/functionality and should be fixed before moving to a live enviroment
Low	Issues on this level are minor details and warning that can remain unfixed
Informational	Informational level is to offer suggestions for improvement of efficacy or secruity for fratures with risk free factor

Threat Level

High	**2** threats found
Medium	**4** threats found
Low	**2** threats found
Informational	**2** threats found

Detail Analysis

Vulnerability Check

13 Passed | 8 Fail

● Arbitrary Jump/Storage Write	● Centralization of Control		
● Compiler Issues	● Delegate Call to Untrusted Contract		
● Dependence on Predictable Variables	● Ether/Token Theft		
● Flash Loans	● Front Running	● Improper Events	
● Improper Authorization Scheme	● Integer Over/Underflow		
● Logical Issues	● Oracle Issues	● Outdated Compiler Version	
● Race Conditions	● Reentrancy	● Signature Issues	● Sybil Attack
● Unbounded Loops	● Unused Code	● Overall Contract Safety	

Detail Analysis

Detail Analysis

13 Passed

8 Fail

CATEGORY	STATUS	NOTES
Arbitrary Jump/Storage Write	●	No arbitrary jumps or storage writes detected, standard ERC20 and Ownable patterns used.
Centralization of Control	●	The contract uses an 'onlyOwner' modifier, centralizing control in the owner's hands.
Compiler Issues	●	Compiled with a recent Solidity version (v0.8.0) without known compiler issues.
Delegate Call to Untrusted Contract	●	No delegate calls to external contracts present in the contract.
Dependence on Predictable Variables	●	No critical dependence on variables like block.timestamp or block.number.

Detail Analysis

Detail Analysis

13 Passed

8 Fail

CATEGORY	STATUS	NOTES
Ether/Token Theft		Standard ERC20 transfer mechanisms, no functions that could lead to Ether or token theft.
Flash Loans		Flash loan attack vectors not applicable, no external calls or token price dependencies.
Front Running		Some functions might be susceptible to front-running due to public visibility and transfer mechanics, although no direct financial risk observed.
Improper Events		All external state-changing functions emit appropriate events.
Improper Authorization Scheme		Centralized authorization scheme with 'onlyOwner', potential risk if the owner is compromised.
Integer Over/Underflow		Solidity 0.8.0 inherently protects against integer overflow and underflow.

Detail Analysis

Detail Analysis

13 Passed

8 Fail

CATEGORY	STATUS	NOTES
Logical Issues	●	Logic appears sound, but some custom functions like 'setRule' and 'blacklist' need careful review to ensure they behave as intended.
Oracle Issues	●	No external oracles or dependencies on off-chain data.
Outdated Compiler Version	●	Compiled with a recent and stable version of Solidity.
Race Conditions	●	Potential for race conditions in functions like 'transfer' and 'approve', common in ERC20 tokens.
Reentrancy	●	No external calls that could lead to reentrancy attacks.
Signature Issues	●	No signature-based functionalities in the contract.

Detail Analysis

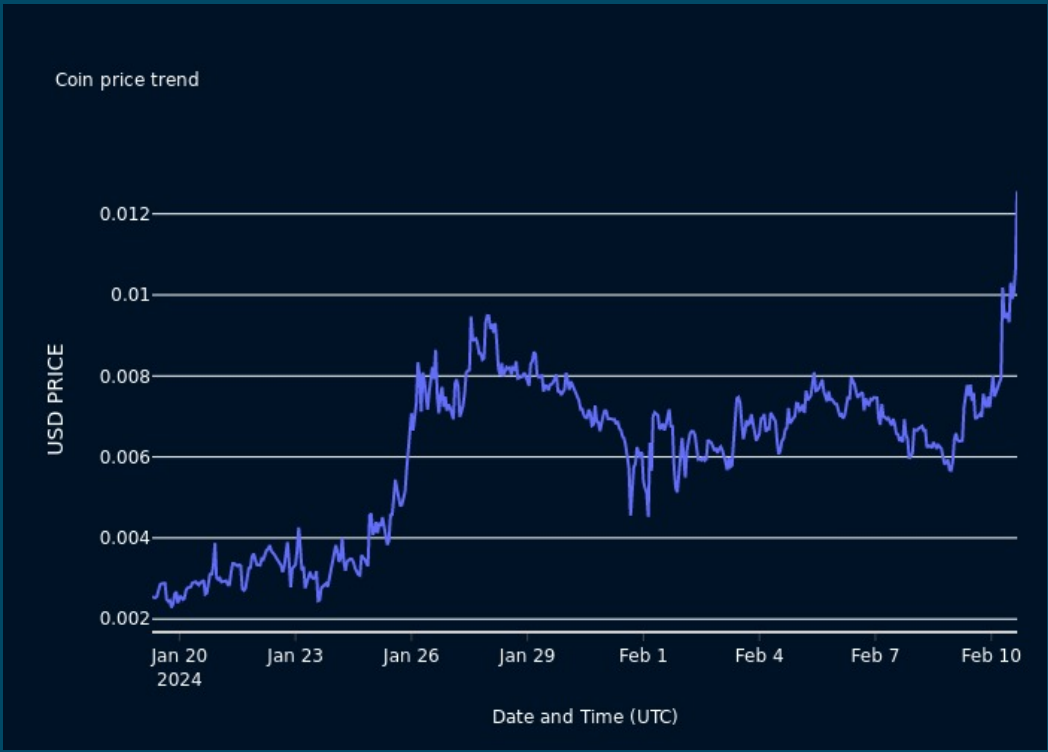
Detail Analysis

13 Passed

8 Fail

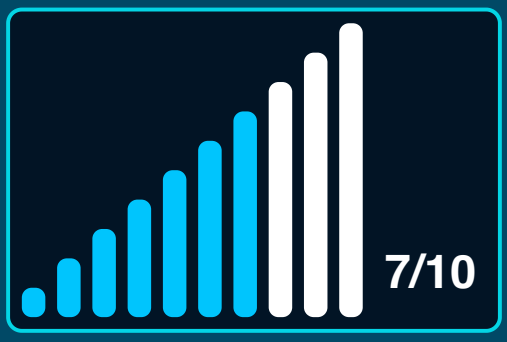
CATEGORY	STATUS	NOTES
Sybil Attack	🔴	While not directly vulnerable, ERC20 tokens can potentially be affected by sybil attacks in broader ecosystem.
Unbounded Loops	🟢	No unbounded loops that could lead to gas limit issues.
Unused Code	🔴	Some code paths are not used in current contract logic but do not pose a risk.
Overall Contract Safety	🔴	While the contract follows common ERC20 and Ownable patterns, centralized control and potential logical issues warrant caution.

Market Analysis

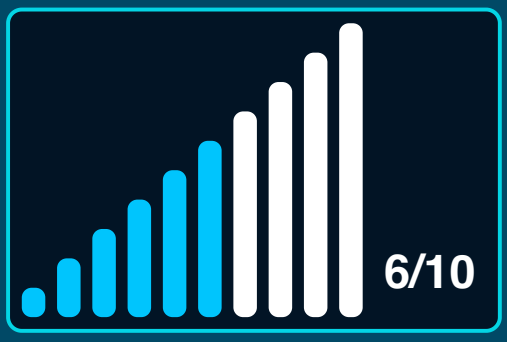


Score

Total Audit Score



Security Score



Legal Disclaimer

0xscans operates as an automated system for smart contract due diligence, acknowledging the possibility of bugs or vulnerabilities impacting token values. We do not hold specific obligations regarding your trading outcomes or the utilization of audit content. Users release 0xscans from any liability associated with content obtained through the tool.

AI generated by 0xscans AI technology

Chat with us

Telegram

For more information. Visit below:

Twitter

Github