OXSCANS

MetaZero Counter-Strike 2: Guns

AI Generated at 06:18 PM, UTC

February 27, 2024

OVERVIEW

This audit has been perpared for 'MetaZero Counter-Strike 2: Guns' 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

MetaZero Counter-Strike 2: Guns

Name **MetaZero Counter-Strike 2: Guns**

Info

General Information

Tokenomics

Contract Address 0x40909204bf1d4ff728700bDdd927dfFd85252d87

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	0	v0.8.22

Smart Contract Stats

Functions	Events	Constructor
54	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	**4** threats found
Medium	**0** threats found
Low	**1** threats found
Informational	**1** threats found

Detail Analysis

Vulnerability Check

16 Passed | 5 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- Outdated Compiler Version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

16 Passed 5 Fail

CATEGORY	STATUS	NOTES
Arbitrary Jump/Storage Write	●	The contract does not contain arbitrary jumps or storage writes.
Centralization of Control	●	The contract has centralized control through the owner role, which can set minter addresses, update base URI, and change royalty information.
Compiler Issues	●	The contract is compiled with Solidity version ^0.8.19 which is a stable and widely used version.
Delegate Call to Untrusted Contract	●	There are no delegate calls to untrusted contracts in the given contract code.
Dependence on Predictable Variables	●	The contract relies on predictable variables such as block.timestamp or block.number for nonce validation in userMint function.

Detail Analysis

Detail Analysis

16 Passed 5 Fail

CATEGORY	STATUS	NOTES
Ether/Token Theft	●	The contract's functions are well-structured to prevent unauthorized access to funds.
Flash Loans	●	This contract does not interact with flash loan functionalities.
Front Running	●	The contract is potentially vulnerable to front-running attacks, especially in the userMint function where nonces and signatures are used.
Improper Events	●	All external and security-critical internal functions emit events correctly.
Improper Authorization Scheme	●	The contract uses a single owner authorization scheme which creates a central point of control and increases risks.
Integer Over/Underflow	●	The contract uses Solidity ^0.8.19 which has built-in checks for integer overflows and underflows.

Detail Analysis

Detail Analysis

16 Passed

5 Fail

CATEGORY	STATUS	NOTES
Logical Issues		No logical issues or inconsistencies apparent in the contract code.
Oracle Issues		The contract does not use external oracles.
Outdated Compiler Version		The contract uses a recent compiler version (0.8.19).
Race Conditions		No race conditions detected in the contract.
Reentrancy		The contract functions are structured to avoid reentrancy vulnerabilities.
Signature Issues		The contract does not involve Ethereum signatures in its logic.

Detail Analysis

Detail Analysis

16 Passed

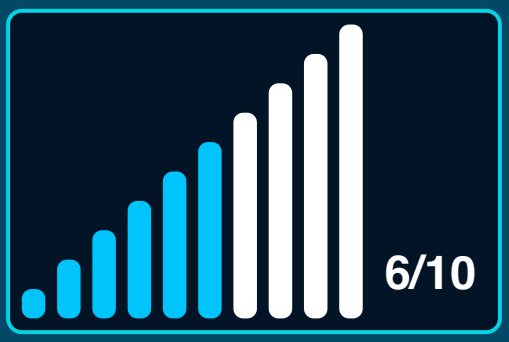
5 Fail

CATEGORY	STATUS	NOTES
Sybil Attack		There's no functionality in the contract that could be vulnerable to Sybil attacks.
Unbounded Loops		No unbounded loops that could lead to gas limit issues and DoS attacks.
Unused Code		The contract does not contain significant amounts of unused code.
Overall Contract Safety		The contract has several centralization issues and is potentially vulnerable to front-running, making it less safe.

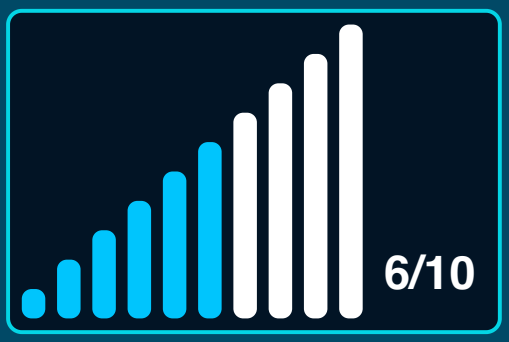
Market Analysis

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