SolidityScan

An open to all quick scanning extension designed to view results in simple terms. Initiate a smart contract scan by selecting from a wide range of supported protocols and get a quick analysis report within seconds.

Ethereum - (etherscan.io)

Goerli Testnet

0×989f3c571837Ae36866df22af9c735B9d8eF5A98

Start Scan

Security Score

Your Security Score is GREAT

VULNERABILITIES DETECTED

Critical	High	Medium
0	0	0

Low	Informational	Gas
0	0	0

SCAN STATISTICS

Security Score	100.00/100
Duration	5 seconds
Lines of Code	2

DETAILED RESULT

This contract has been analyzed by more than 130 proprietary vulnerability patterns of SolidityScan. Vulnerability details and mechanisms to remediate the risks tailored specific to the contract are now available in the link below.

View Detailed Result →

THREAT SCAN SUMMARY

Threat Score

ThreatScan, a smart contract analysis tool, is built by the SolidityScan team. It is designed to assist users in identifying potential rug pull scams by providing an in-depth analysis of a smart contract's code and highlighting any potential red flags that may indicate a scam.

IS SOURCE CODE VERIFIED

The contract's source code is verified.
Source code verification provides transparency for users interacting with smart contracts. Block explorers validate the compiled code with the one on the blockchain. This also gives users a chance to audit the contracts.

PRESENCE OF MINTING FUNCTION

The contract cannot mint new tokens. The `_mint` functions was not detected in the contracts.
Mint functions are used to create new tokens and transfer them to the user's/owner's wallet to whom the tokens are minted. This increases the overall circulation of the tokens.

PRESENCE OF BURN FUNCTION

The tokens can not be burned in this contract.
Burn functions are used to increase the total value of the tokens by decreasing the total supply.

SOLIDITY PRAGMA VERSION

The contract can not be compiled with an older Solidity version.
Pragma versions decide the compiler version with which the contract can be compiled. Having older pragma versions means that the code may be

compiled with outdated and vulnerable compiler versions, potentially introducing vulnerabilities and CVEs.

PROXY-BASED UPGRADABLE CONTRACT

This is not an upgradable contract.
Having upgradeable contracts or proxy patterns allows owners to make changes to the contract's functions, token circulation, and distribution.

OWNERS CANNOT BLACKLIST TOKENS OR USERS

Owners cannot blacklist tokens or users.
If the owner of a contract has permission to blacklist users or tokens, all the transactions related to those entities will be halted immediately.

PAUSABLE CONTRACTS

This is not a Pausable contract.
If a contract is pausable, it allows privileged users or owners to halt the execution of certain critical functions of the contract in case malicious transactions are found.

CRITICAL ADMINISTRATIVE FUNCTIONS

Critical functions that add, update, or delete owner/admin addresses are not detected
These functions control the ownership of the contract and allow privileged users to add, update, or delete owner or administrative addresses. Owners are usually allowed to control all the critical aspects of the contract.

CONTRACT/TOKEN SELF DESTRUCT

The contract cannot be self-destructed by owners.
`selfdestruct()` is a special function in Solidity that destroys the contract and transfers all the remaining funds to the address specified during the call. This is usually access-control protected.

ERC20 RACE CONDITION

The contract is not vulnerable to ERC-20 approve Race condition vulnerability.
ERC-20 approve function is vulnerable to a frontrunning attack which can be exploited by the token receiver to withdraw more tokens than the

allowance. Proper mitigation steps should be implemented to prevent such vulnerabilities.

RENOUNCED OWNERSHIP

The contract's owner was not found.
Renounced ownership shows that the contract is truly decentralized and once deployed, it can't be manipulated by administrators.

USERS WITH TOKEN BALANCE MORE THAN 5%

Some addresses contains more than 5% of circulating token supply.
0xee2a7b2c72217f6ebf0401dabb407c7a600d910f 100.0%
. Token distribution plays an important role when controlling the price of an asset.

OVERPOWERED OWNERS

The contracts have not defined any owner-controlled functions.
Giving too many privileges to the owners via critical functions might put the user's funds at risk if the owners are compromised or if a rug-pulling attack happens.

COOLDOWN FEATURE

The contract does not have a cooldown feature.
Cooldown functions are used to halt trading or other contract workflows for a certain amount of time so as to prevent users from repeatedly executing transactions or buying and selling tokens.

OWNERS WHITELISTING TOKENS/USERS

Owners can not whitelist tokens or users.
If the owner of a contract has permission to whitelist users or tokens, it'll be unfair toward other users or the transaction flow may not be executed impartially.

OWNERS CAN SET/UPDATE FEES

Owners can not set or update Fees in the contract.

HARDCODED ADDRESSES

The contract was not hardcoding addresses in the code.

OWNERS UPDATING TOKEN BALANCE

The contract does not have any owner-controlled functions modifying token balances for users or the contract

FUNCTION RETRIEVING OWNERSHIP

No such functions were found
If this function exists, it is possible for the project owner to regain ownership even after relinquishing it.

Why SolidityScan ?

Smart-contract scanning tool built to discover vulnerabilities & mitigate risks in your code.

Initiate Scans

Publish Reports

130+ Vulnerability Checks