



Coinsult

Advanced Manual Smart Contract Audit




Project: LegoX

Website: <http://www.legox.xyz>

 **Low-Risk**

1 low-risk code
issues found

 **Medium-Risk**

0 medium-risk code
issues found

 **High-Risk**

0 high-risk code
issues found

Contract Address

0x3dfFCeE85E33b0D051bFdC35Eb58FE23703e64dA (ETH)

Disclaimer: Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

Disclaimer

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xd8e9bbe1ac1b542773cb1101a9bd7cc98319228e	1,000,000,000	100.0000%

Source Code

Coinsult was comissioned by LegoX to perform an audit based on the following smart contract:

<https://etherscan.io/address/0x3dffcee85e33b0d051bfdc35eb58fe23703e64da#code>

Manual Code Review

In this audit report we will highlight all these issues:

Low-Risk

1 low-risk code
issues found

Medium-Risk

0 medium-risk code
issues found

High-Risk

0 high-risk code
issues found

The detailed report continues on the next page...

● **Low-Risk:** Could be fixed, will not bring problems.

Redundant Statements

Detect the usage of redundant statements that have no effect.

```
// File: @openzeppelin/contracts/utils/Context.sol

// OpenZeppelin Contracts v4.4.1 (utils/Context.sol)

// File: @openzeppelin/contracts/token/ERC20/IERC20.sol

// File: @openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol

// OpenZeppelin Contracts v4.4.1 (token/ERC20/extensions/IERC20Metadata.sol)

// File: @openzeppelin/contracts/token/ERC20/ERC20.sol

// File: $LEGOx/SimpleToken.sol
```

Recommendation

Remove redundant statements if they congest code but offer no value.

Exploit scenario

```
contract RedundantStatementsContract {

    constructor() public {
        uint; // Elementary Type Name
        bool; // Elementary Type Name
        RedundantStatementsContract; // Identifier
    }

    function test() public returns (uint) {
        uint; // Elementary Type Name
        assert; // Identifier
        test; // Identifier
        return 777;
    }
}
```

Each commented line references types/identifiers, but performs no action with them, so no code will be generated for such statements and they can be removed.

Owner privileges

- Owner cannot set fees higher than 25%
- Owner cannot pause trading
- Owner cannot change max transaction amount

Extra notes by the team

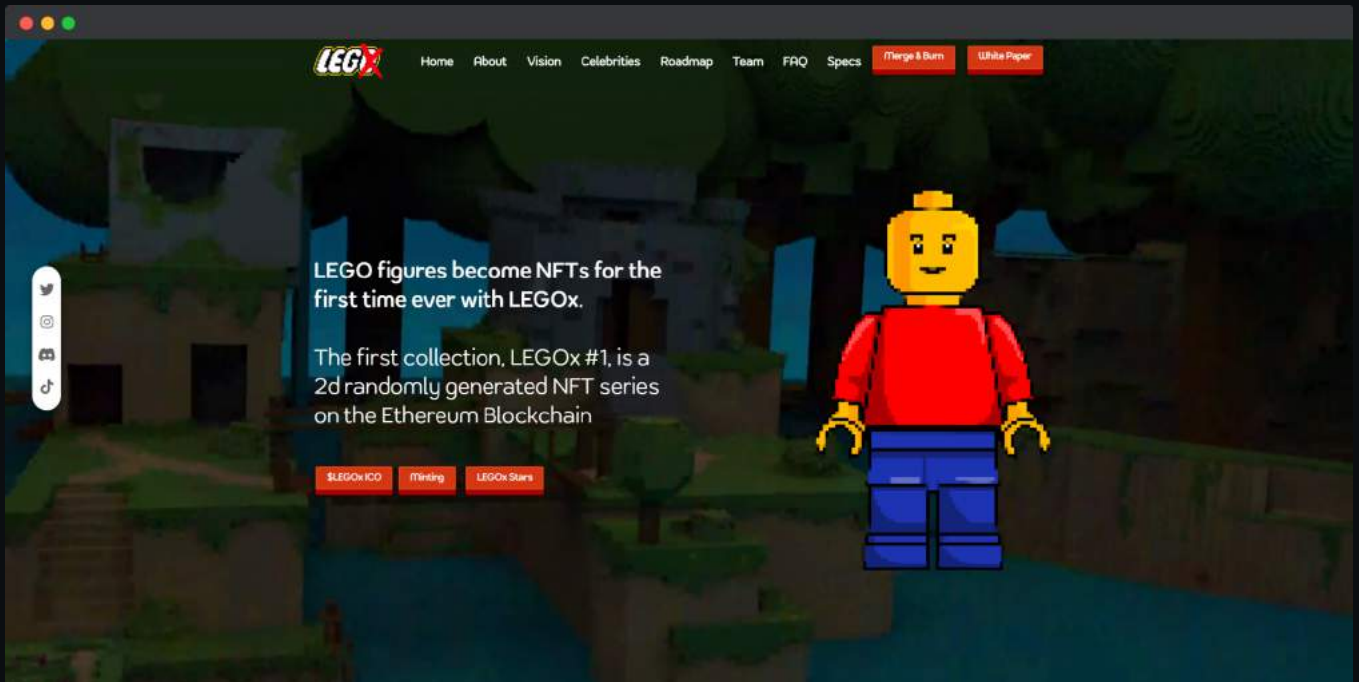
Contract is very basic, no real issues detected.

Contract Snapshot

```
contract SimpleToken is ERC20 {  
  
    constructor(  
        string memory name,  
        string memory symbol,  
        uint256 totalSupply_  
    ) ERC20(name, symbol) {  
        _mint(msg.sender, totalSupply_);  
    }  
  
}
```

Website Review

Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.



- Mobile Friendly
- Does not contain jQuery errors
- SSL Secured
- No major spelling errors

Project Overview

● Not KYC verified by Coinsult

AUDITED
BY COINSULT.NET

