



Coinsult

Advanced Manual Smart Contract Audit



Project: Vulcano

Website: <https://www.vulcano.gg>

Low-risk

3 low-risk code
issues found

Medium-risk

0 medium-risk code
issues found

High-risk

0 high-risk code
issues found

Contract address

Not deployed yet

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

Not deployed yet

Source code

Coinsult was commissioned by Vulcano to perform an audit based on the following smart contract:

<https://github.com/NWCrypto/vulcano/blob/main/packages/hardhat/contracts/deployable/Token.sol> (PRIVATE)

Note: This project uses openzeppelin imports. While we do check the full contract for vulnerabilities at the time of the audit, we can not ensure the correctness of these imported modules.

Manual Code Review

● Low-risk

3 low-risk code issues found.

Could be fixed, will not bring problems.

- State variable written after a call

External calls:

pinkAntiBot.setTokenOwner(msg.sender) (Token.sol#28)

State variables written after the call(s):

antiBotEnabled = true (Token.sol#29)

Recommendation: Apply the check-effects-interactions pattern.

Additional information: [Slither](#)

```
constructor(  
    string memory tokenName,  
    string memory symbol,  
    address pinkAntiBot_  
) ERC20(tokenName, symbol) {  
    _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);  
    _grantRole(PAUSER_ROLE, msg.sender);  
    _mint(msg.sender, 1000000000 * 10**decimals());  
  
    // Create an instance of the PinkAntiBot variable from the  
provided address  
    pinkAntiBot = IPinkAntiBot(pinkAntiBot_);  
    // Register the deployer to be the token owner with  
PinkAntiBot. You can  
    // later change the token owner in the PinkAntiBot contract  
    pinkAntiBot.setTokenOwner(msg.sender);  
    antiBotEnabled = true;  
}
```

- Literals with many digits are difficult to read and review.

Recommendation: Use Ether suffix, Time suffix, or The scientific notation

```
_mint(msg.sender, 1000000000 * 10**decimals());
```

- Multiple Solidity versions are used

Recommendation: Try to use one Solidity version.

```
- Version used: ['>=0.5.0', '^0.8.0', '^0.8.4']
- ^0.8.0 (@openzeppelin/contracts/access/AccessControl.sol#4)
- ^0.8.0 (@openzeppelin/contracts/access/IAccessControl.sol#4)
- ^0.8.0 (@openzeppelin/contracts/security/Pausable.sol#4)
- ^0.8.0 (@openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- ^0.8.0 (@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
- ^0.8.0
(@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol#4)
- ^0.8.0
(@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
- ^0.8.0 (@openzeppelin/contracts/utils/Context.sol#4)
- ^0.8.0 (@openzeppelin/contracts/utils/Strings.sol#4)
- ^0.8.0 (@openzeppelin/contracts/utils/introspection/ERC165.sol#4)
- ^0.8.0 (@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
- ^0.8.4 (contracts/deployable/Token.sol#2)
- >=0.5.0 (contracts/pink/IPinkAntiBot.sol#2)
```

● Medium-risk

0 medium-risk code issues found.

Should be fixed, could bring problems.

● High-risk

0 high-risk code issues found

Must be fixed, and will bring problems.

Extra notes by the team

Note: This project uses openzeppelin imports. While we do check the full contract for vulnerabilities at the time of the audit, we can not ensure the correctness of these imported modules.

- Anti bot system implemented
- Contract should be deployed with PinkAntiBot:
0x8efdb3b642eb2a20607ffe0a56cfeff6a95df002
- Dev notes should be removed before launch
 - Changed by the team after the audit
- Owner can pause and unpause the contract

Contract Snapshot

```
contract VulcanoToken is ERC20, ERC20Burnable, Pausable, AccessControl
{
    IPinkAntiBot public pinkAntiBot;
    bool public antiBotEnabled;
    bytes32 public constant PAUSER_ROLE = keccak256("PAUSER_ROLE");

    constructor(
        string memory tokenName,
        string memory symbol,
        address pinkAntiBot_
    ) ERC20(tokenName, symbol) {
        _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);
        _grantRole(PAUSER_ROLE, msg.sender);
        _mint(msg.sender, 1000000000 * 10**decimals());

        // Create an instance of the PinkAntiBot variable from the
provided address
        pinkAntiBot = IPinkAntiBot(pinkAntiBot_);
        // Register the deployer to be the token owner with
PinkAntiBot. You can
        // later change the token owner in the PinkAntiBot contract
        pinkAntiBot.setTokenOwner(msg.sender);
        antiBotEnabled = true;
    }

    function pause() public onlyRole(PAUSER_ROLE) {
        _pause();
    }

    function unpause() public onlyRole(PAUSER_ROLE) {
        _unpause();
    }

    function _beforeTokenTransfer(
        address from,
        address to,
        uint256 amount
    ) internal override whenNotPaused {
        super._beforeTokenTransfer(from, to, amount);
    }
}
```

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
- Contains no jQuery errors
- SSL Secured
- No major spelling errors

Note: The website is graphically dependent on the visitor's graphics card; The website might lag if the visitor has an outdated graphics card.

Loading speed: 88%

Rug-pull Review

Based on the available information analyzed by us, we come to the following conclusions:

- Locked Liquidity (no liquidity yet)
- Large unlocked wallets
 - Note: Tokens not distributed yet
- Doxxed Team

Honeypot Review

Based on the available information analyzed by us, we come to the following conclusions:

- Ability to sell
- Owner is able to pause the contract
- Router not hard coded in the contract

Note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.