Coinsult

Advanced Manual Smart Contract Audit

Project: SonicDoge

Website: https://sonicdoge.dog

Low-Risk	Medium-Risk	High-Risk
2 low-risk code issues found	0 medium-risk code issues found	0 high-risk code issues found

Contract Address

0x69D3E8279209b78dA6514981A962680b7C77d9F1

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

-

Source Code

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

https://explorer.dogechain.dog/address/0x69D3E8279209b78dA6514981A962680b7C77c

Manual Code Review

In this audit report we will highlight all these issues:

Low-Risk

2 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.

No zero address validation for some functions

Detect missing zero address validation.

```
function changeTreasuryWallet(address _treasuryWallet) external onlyOwner {
    require(_treasuryWallet != treasuryWallet, 'Treasury wallet is already that address');
    treasuryWallet = _treasuryWallet;
}
```

Recommendation

Check that the new address is not zero.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

Bob calls `updateOwner` without specifying the `newOwner`, soBob loses ownership of the contract.

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

Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
function updateSellFees(
    uint256 _liquiditySellFee,
    uint256 _treasurySellFee,
    uint256 _firePitSellFee
) external onlyOwner {
    liquiditySellFee = _liquiditySellFee;
    treasurySellFee = _treasurySellFee;
    firePitSellFee = _firePitSellFee;
    totalSellFee = liquiditySellFee + treasurySellFee + firePitSellFee;

    require(totalSellFee &lt;= 10, &#039;Fees must be less than 10%&#039;);
}
```

Recommendation

Emit an event for critical parameter changes.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

`updateOwner()` has no event, so it is difficult to track off-chain changes in the buy price.

Owner privileges

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

Extra notes by the team

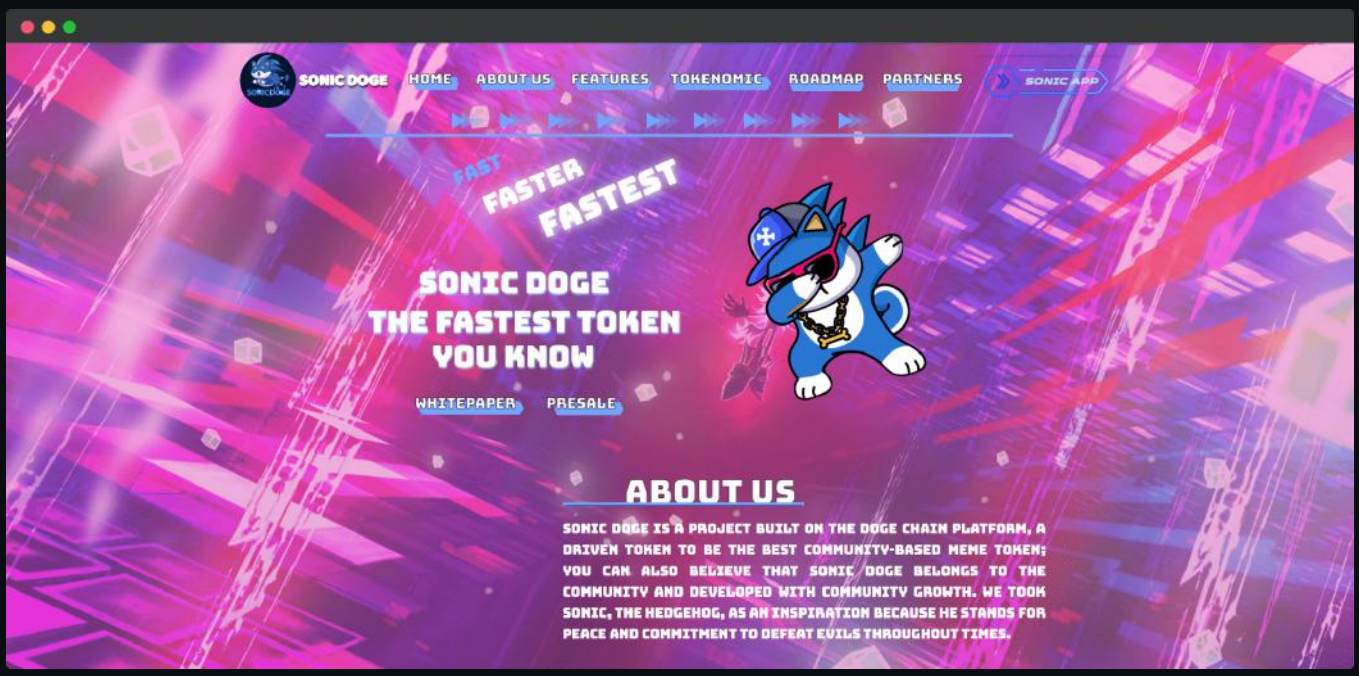
No notes

Contract Snapshot

```
contract SonicDoge is ERC20, Ownable {
mapping(address =&gt; uint256) _rBalance;
mapping(address =&gt; mapping(address =&gt; uint256)) private _allowances;
mapping(address =&gt; bool) private _isExcludedFromFees;

uint256 public liquidityBuyFee;
uint256 public treasuryBuyFee;
uint256 public totalBuyFee;

uint256 public liquiditySellFee;
uint256 public treasurySellFee;
uint256 public firePitSellFee;
uint256 public totalSellFee;

IDogeSwapV2Router02 public uniswapV2Router;
address public uniswapV2Pair;
address public sustainabilityWallet;
address public treasuryWallet;

address private DEAD = 0x000000000000000000000000000000000000dEaD;

mapping(address =&gt; bool) public automatedMarketMakerPairs;

uint256 private initialSupply;
uint256 private rSupply;
uint256 private constant MAX = type(uint256).max;
uint256 private _totalSupply;

bool public swapEnabled = true;
bool private inSwap;
uint256 private swapThreshold;
uint256 public lastSwapTime;
uint256 public swapInterval;

modifier swapping() {
    inSwap = true;
    _;
```

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

SonicDoge

Audited by Coinsult.net

Date: 18 August 2022

Advanced Manual Smart Contract Audit