Coinsult

Advanced Manual Smart Contract Audit

Project: Shaggy Inu

Website: https://www.shaggytoken.com/

🟢 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

0x287C2fB71F034F089f24baB8B58b6F7653febe09

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	0x5232cf4195ec413725cb66a7b18d4902e0fee04d	1,000,000,000,000	100.0000%

Source Code

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

https://bscscan.com/address/0x287c2fb71f034f089f24bab8b58b6f7653febe09#code

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.

Contract contains Reentrancy vulnerabilities

Additional information: This combination increases risk of malicious intent. While it may be justified by some complex mechanics (e.g. rebase, reflections, buyback).

More information: Slither

```
function _transfer(address from, address to, uint256 amount) private {

    if(from != owner() &amp;&amp; to != owner()){
        require(tradingOpen,"Trading not open yet");

        if(SHAG &amp;&amp; from == uniswapV2Pair){
            isBlacklisted[to] = true;
        }
    }

    if(blacklistMode &amp;&amp; !SHAG){
        require(!isBlacklisted[from],"Blacklisted");
    }

    require((amount &lt;= _maxTxAmount) || isTxLimitExempt[from] || isTxLimitExempt[to], &quot;Max T

    if (!isWalletLimitExempt[from] &amp;&amp; !isWalletLimitExempt[to] &amp;&amp; to != uniswapV2Pai
        require((balanceOf(to) + amount) = _maxTxAmount) {
            contractTokenBalance = _maxTxAmount - 1;
        }

        bool overMinTokenBalance = contractTokenBalance &gt;= swapThreshold;
```

Recommendation

Apply the check-effects-interactions pattern.

Exploit scenario

```
function withdrawBalance(){
    // send userBalance[msg.sender] Ether to msg.sender
    // if mgs.sender is a contract, it will call its fallback function
    if( ! (msg.sender.call.value(userBalance[msg.sender])() ) ){
        throw;
    }
    userBalance[msg.sender] = 0;
}
```

Bob uses the re-entrancy bug to call withdrawBalance two times, and withdraw more than its initial deposit to the contract.

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

Redundant Statements

Detect the usage of redundant statements that have no effect.

```
function _msgData() internal view virtual returns (bytes calldata) {
    this; // silence state mutability warning without generating bytecode - see https://github.com/e
    return msg.data;
}
```

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
- 🟡 Owner can exclude from fees

⚠️ Owner has hardcoded max transaction amount at totalsupply/50

⚠️ Owner has hardcoded max wallet balance at totalsupply/50

Extra notes by the team

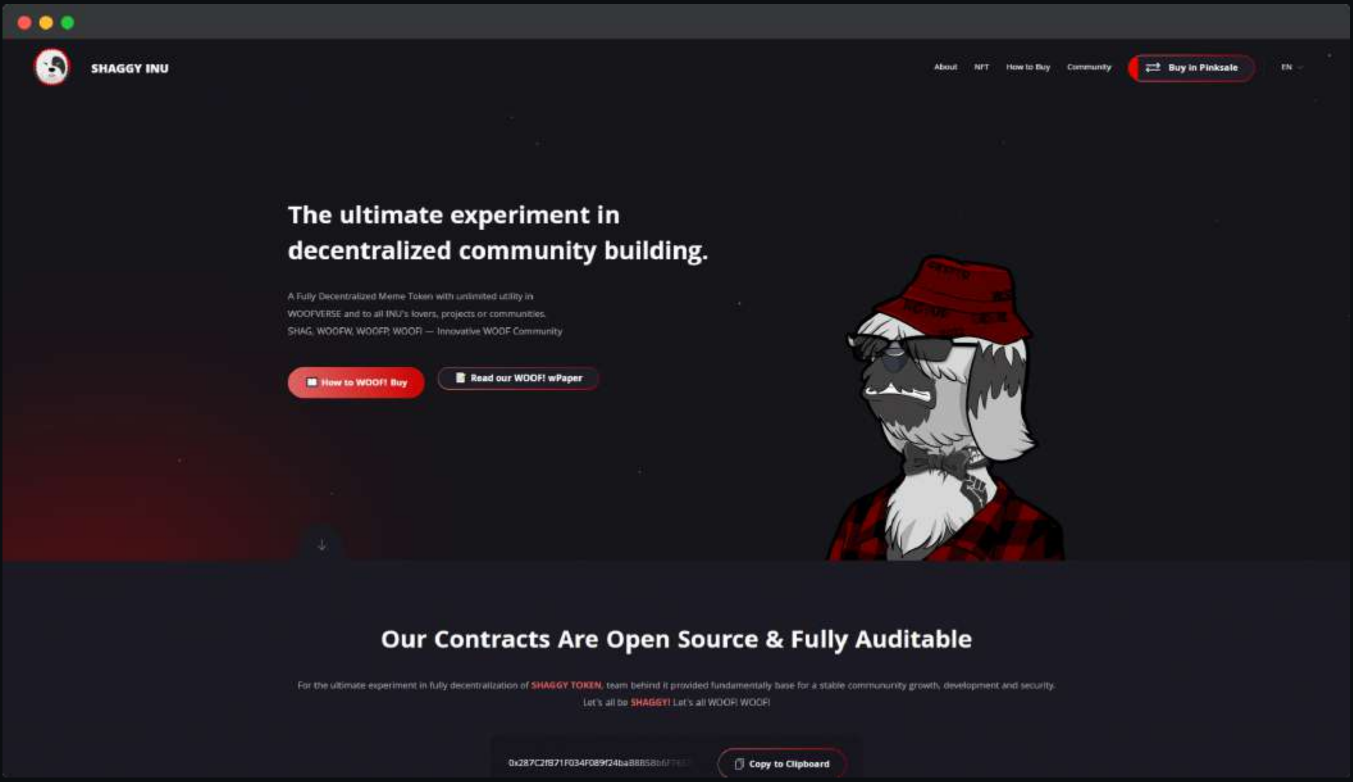
No notes

Contract Snapshot

```
contract SHAGGYTOKEN is Context, IERC20, Ownable {
using Address for address;

mapping (address =&gt; uint256) public _balance_reflected;
mapping (address =&gt; uint256) public _balance_total;
mapping (address =&gt; mapping (address =&gt; uint256)) private _allowances;

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

bool public blacklistMode = true;
mapping (address =&gt; bool) public isBlacklisted;

bool public tradingOpen = true;
bool public SHAG = 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.



The ultimate experiment in decentralized community building.

A Fully Decentralized Meme Token with unlimited utility in WOOFVERSE and to all INU's lovers, projects or communities. SHAG, WOOFW, WOOFP, WOOF! — Innovative WOOF Community

Our Contracts Are Open Source & Fully Auditable

For the ultimate experiment in fully decentralization of SHAGGY TOKEN, team behind it provided fundamentally base for a stable commununity growth, development and security. Let's all be SHAGGY! Let's all WOOF! WOOF!

0x287C2f871F034F089f24baB8858b6F7E

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

Project Overview

● Not KYC verified by Coinsult

Shaggy Inu

Audited by Coinsult.net

Date: 2 August 2022

✓ Advanced Manual Smart Contract Audit