Coinsult

Advanced Manual Smart Contract Audit

Project: Siuuu Token

Website: https://siu2022.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

0x16E33015C466c0183cd1c2adbC5B4E7bf97548Da

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	0xb2d6cbdb479f3e4ae9efcc8040167344b1c3aec2	625,000,000	100.0000%

Source Code

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

https://bscscan.com/address/0x16E33015C466c0183cd1c2adbC5B4E7bf97548Da#code

Contract uses SiuRouter which points to '0x090da105df89079cac0eb643911b375272550ebe', it is a TransparentUpgradeableProxy contract. Which is not audited by Coinsult.

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.

Avoid relying on block.timestamp

block.timestamp can be manipulated by miners.

```
if (
    antiBotTime &gt; block.timestamp &amp;&amp;
    amount &gt; antiBotAmount &amp;&amp;
    sender != address(this) &amp;&amp;
    recipient != address(this) &amp;&amp;
    sender == uniswapV2Pair
) {
```

Recommendation

Do not use `block.timestamp`, now or `blockhash` as a source of randomness

Exploit scenario

```
contract Game {

    uint reward_determining_number;

    function guessing() external{
      reward_determining_number = uint256(block.blockhash(10000)) % 10;
    }
}
```

Eve is a miner. Eve calls `guessing` and re-orders the block containing the transaction. As a result, Eve wins the game.

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

Too many digits

Literals with many digits are difficult to read and review.

```
uint256 public antiBotAmount = 700000 * 10**18;
uint256 public minSwapAmount = 200000 * 10**18;
```

Recommendation

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

Exploit scenario

```
contract MyContract{
    uint 1_ether = 10000000000000000000;
}
```

While `1_ether` looks like `1 ether`, it is `10 ether`. As a result, it's likely to be used incorrectly.

Owner privileges

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

Extra notes by the team

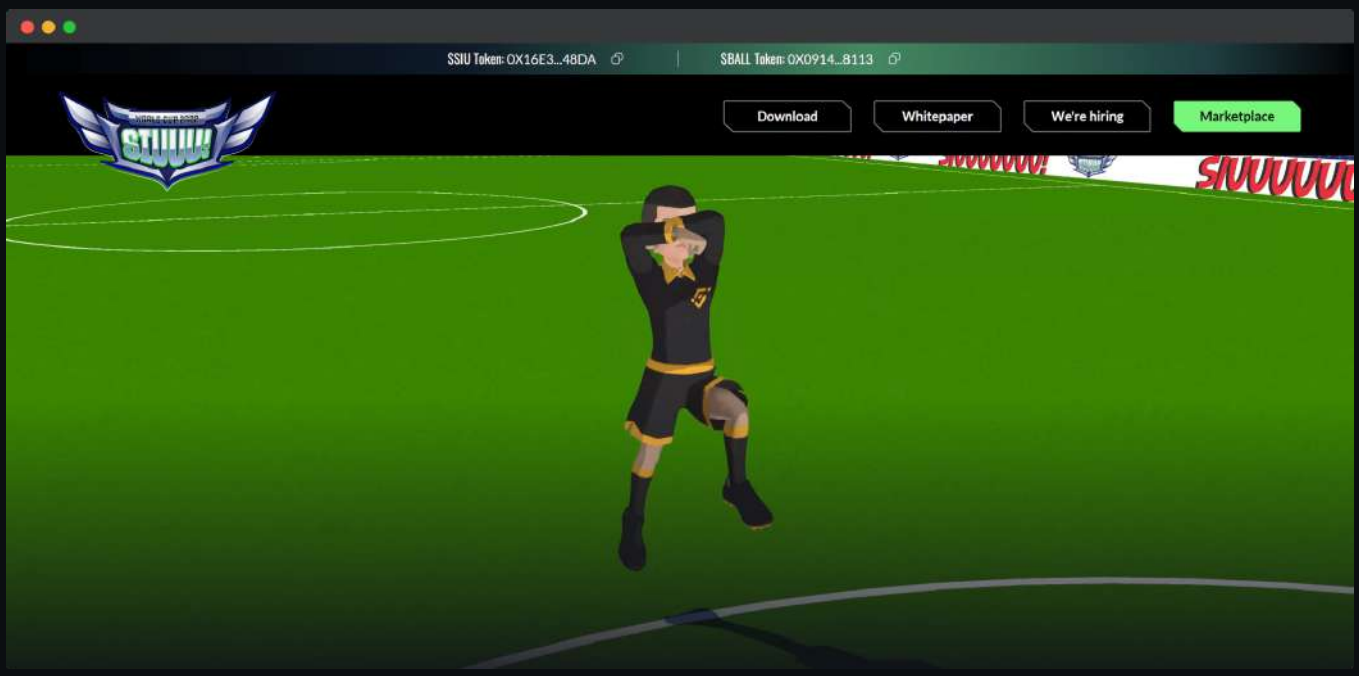
Contract uses SiuRouter which points to ‘0x090da105df89079cac0eb643911b375272550ebe’, it is a TransparentUpgradeableProxy contract. Which is not audited by Coinsult.

Contract Snapshot

```
contract Siuuu is ERC20, Ownable {
using SafeMath for uint256;

uint256 public constant maxSupply = 625 * 10**6 * 10**18;

IUniswapV2Router02 public uniswapV2Router;
SiuRouter public siuRouter;
address public uniswapV2Pair;
address public treasury;
uint256 public sellFeeRate = 6; // 6% fee to Treasury
uint256 public buyFeeRate = 6; // 6% fee to Referral
bool inSwap = false;

address public usdt;
address public idoAddress;

uint256 public antiBotDuration = 10;
uint256 public antiBotTime;
uint256 public antiBotAmount = 700000 * 10**18;
uint256 public minSwapAmount = 200000 * 10**18;

constructor(
    string memory name,
    string memory symbol,
    address _usdt,
    address _siuRouter
) ERC20(name, symbol) {
    uniswapV2Router = IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);
    usdt = _usdt;

    siuRouter = SiuRouter(_siuRouter);

    uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory()).createPair(address(this), usdt);

    _mint(_msgSender(), maxSupply);

    treasury = _msgSender();
```

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

Siuuu Token

Audited by Coinsult.net

Date: 21 July 2022

Advanced Manual Smart Contract Audit