Coinsult

Advanced Manual Smart Contract Audit

Project: Spring Rolls Kitchen
Website: https://www.springrolls.kitchen/

🟢 **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

0x684e1C37813363bA2Dc9F9AAcc3573bC2D1F0A3A

Disclaimer

Tokenomics

- Not applicable

Source code

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

https://bscscan.com/address/0x684e1C37813363bA2Dc9F9AAcc3573bC2D1F0A3A#code

Manual Code Review

🟢 Low-risk

3 low-risk code issues found.
Could be fixed, will not bring problems.

- Avoid relying on block.timestamp
  block.timestamp can be manipulated by miners.

```
lastHarvest[msg.sender] = block.timestamp;
```

- Missing zero address validation
  Check that the new address is not the zero address.

```
    function whitelistWallet(address Wallet, bool isWhitelisted)
public onlyOwner {
        Whitelisted[Wallet] = isWhitelisted;
    }
```

- Literals with many digits are difficult to read and review.
  Recommendation: Use Ether suffix, Time suffix, or The scientific notation

```
marketSpringRolls = 108000000000;
```

🟡 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

🟡 Owner can whitelist wallets.

🟡 The ownership of the contract isn't renounced.

🟢 Funds are locked inside the contract and are given out daily.

Contract Snapshot

```solidity
contract SpringRollsKitchen is Context, Ownable {
    using SafeMath for uint256;

    uint256 private SPRING_ROLL_TO_HIRE_CHEFS = 1080000;
    uint256 private PSN = 10000;
    uint256 private PSNH = 5000;
    uint256 private devFeeVal = 2;
    uint256 private marketingFeeVal = 2;
    uint256 private webFeeVal = 2;
    uint256 private teamFeeVal = 2;
    bool private initialized = false;
    address payable private recAdd;
    address payable private marketingAdd;
    address payable private teamAdd;
    address payable private webAdd;
    mapping (address => uint256) private springRollChefs;
    mapping (address => uint256) private claimedSpringRolls;
    mapping (address => uint256) private lastHarvest;
    mapping (address => address) private referrals;
    bool private whitelistActive = true;
    mapping(address => bool) public Whitelisted;
    uint256 private marketSpringRolls;
```

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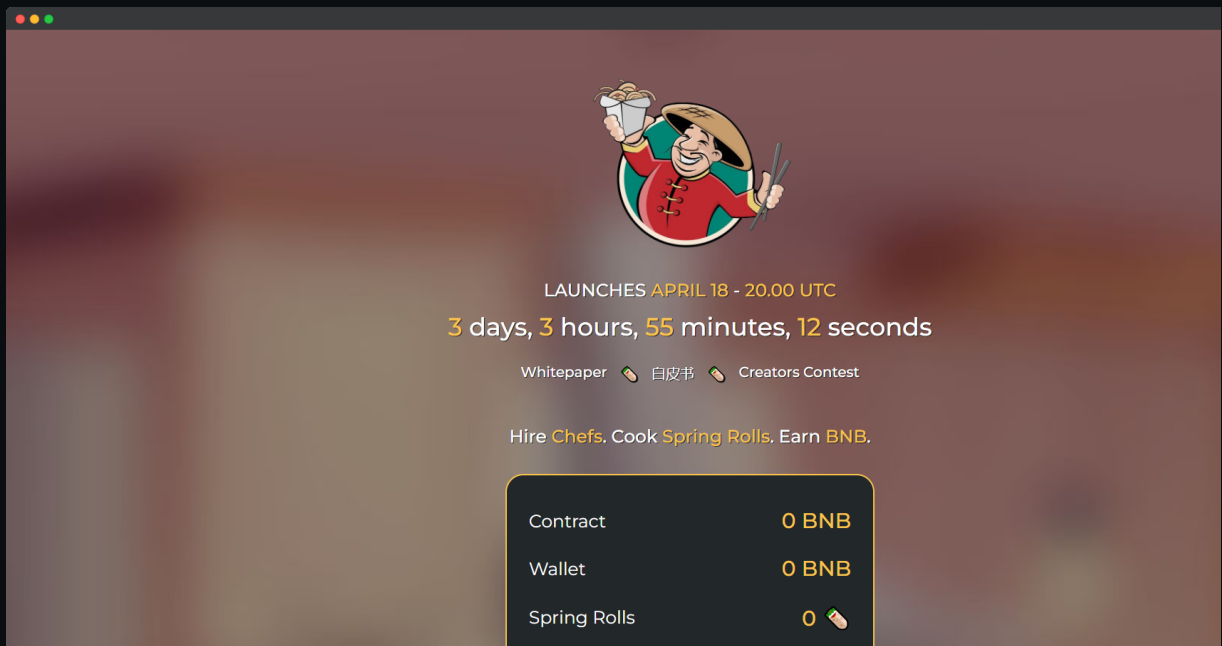
Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.

- 🟢 Mobile Friendly
- 🟢 Contains no jQuery errors
- 🟢 SSL Secured
- 🟢 No major spelling errors

Loading speed: 82%

Rug-pull Review

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

🟢 Locked Liquidity
- Not applicable

🟢 Large unlocked wallets
- Not applicable

🔴 No doxxed Team

Honeypot Review

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

🟢 Ability to sell

🟢 Owner is not able to pause the contract

🟢 Router can be changed
- Not applicable

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.