Coinsult

Advanced Manual Smart Contract Audit

Project: Football Move

Website: https://footballmove.io/

Low-Risk	Medium-Risk	High-Risk
4 low-risk code issues found	1 medium-risk code issues found	0 high-risk code issues found

Contract Address

0x1F461B82234b70b44780bb21C8D97456161EBCD7

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	Null Address: 0x000...dEaD	25,000,000	50.0000%
2	0xc57235cf7cc2ecd029355d19b0f11a0f64efddce	12,500,355	25.0007%
3	0x407993575c91ce7643a4d4ccacc9a98c36ee1bbe	12,499,645	24.9993%

Source Code

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

https://bscscan.com/address/0x1F461B82234b70b44780bb21C8D97456161EBCD7#code

Manual Code Review

In this audit report we will highlight all these issues:

🟢 Low-Risk

4 low-risk code
issues found

🟡 Medium-Risk

1 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.

```
function _transfer(
    address sender,
    address receiver,
    uint256 amount
  ) internal virtual override {
    require(
      receiver != address(this),
      string("No transfers to contract allowed.")
    );

    require(!isBlacklist[sender], "User blacklisted");

    uint256 taxAmount = 0;

    if (liquidityPool[sender] == true) {
      //It's an LP Pair and it's a buy
      taxAmount = (amount * buyTax) / MAX_PERCENT;
    } else if (liquidityPool[receiver] == true) {
      //It's an LP Pair and it's a sell
      taxAmount = (amount * sellTax) / MAX_PERCENT;

      uint256 transactionTime = block.timestamp;
```

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.

No zero address validation for some functions

Detect missing zero address validation.

```
function setMarketingPool(address _marketingPool) external onlyOwner {
  marketingPool = _marketingPool;
  emit ChangeMarketingPool(_marketingPool);
}
```

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.

Conformance to Solidity naming conventions

Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
function setBlacklist(address _wallet, bool _status) external onlyOwner {
  isBlacklist[_wallet] = _status;
  emit ChangeBlacklist(_wallet, _status);
}
```

Recommendation

Follow the Solidity naming convention.

Rule exceptions

- Allow constant variable name/symbol/decimals to be lowercase (ERC20).
- Allow _ at the beginning of the `mixed_case` match for private variables and unused parameters.

🟢 **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 returns (bytes memory) {
  this;
  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.

🟡 **Medium-Risk:** Should be fixed, could bring problems.

double variable type specification

```
require(
  receiver != address(this),
  string("No transfers to contract allowed.")
);
```

Recommendation

Leave out the string() type specification

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 can blacklist addresses
- ⚠️ Owner can set trade cooldown

Extra notes by the team

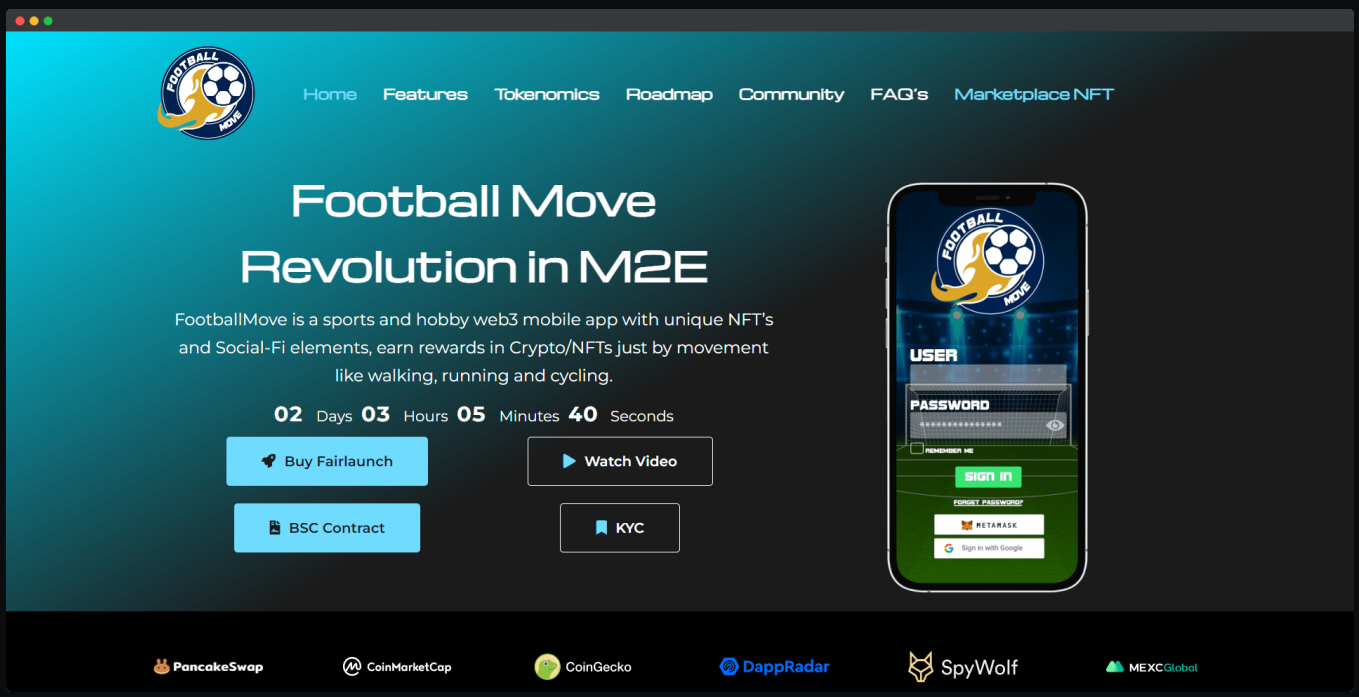
No notes

Contract Snapshot

```
contract SuperToken is BEP20Detailed, BEP20 {

uint256 constant private MAX_PERCENT = 100;

mapping(address =&gt; bool) private isBlacklist;
mapping(address =&gt; bool) private liquidityPool;
mapping(address =&gt; bool) private whitelistTax;
mapping(address =&gt; uint256) private lastTrade;

uint8 private buyTax;
uint8 private sellTax;
uint8 private tradeCooldown;
uint8 private transferTax;
//uint256 private taxAmount;
```

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

Football Move

Audited by Coinsult.net

Date: 4 June 2022

✔ Advanced Manual Smart Contract Audit