Coinsult

Advanced Manual Smart Contract Audit

Project: MetaFarmer Game

Website: https://metafarmergame.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

0xD858f2361FC7290290F926697388d612edA632bB

Disclaimer: Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

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	0x41cad05ca8e6fb43047fe1d5ec523d1de9598c8b	1,000,000,000	100.0000%

Source Code

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

https://bscscan.com/address/0xD858f2361FC7290290F926697388d612edA632bB#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.

Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
/**
 * @dev Leaves the contract without owner. It will not be possible to call
 * `onlyOwner` functions anymore. Can only be called by the current owner.
 *
 * NOTE: Renouncing ownership will leave the contract without an owner,
 * thereby removing any functionality that is only available to the owner.
 */
function renounceOwnership() public virtual onlyOwner {
    _setOwner(address(0));
}
```

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.

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

Redundant Statements

Detect the usage of redundant statements that have no effect.

```
// import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
// import "@openzeppelin/contracts/access/Ownable.sol";
// import "@openzeppelin/contracts/utils/math/SafeMath.sol";
// import "contracts/BaseToken.sol";
```

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

Extra notes by the team

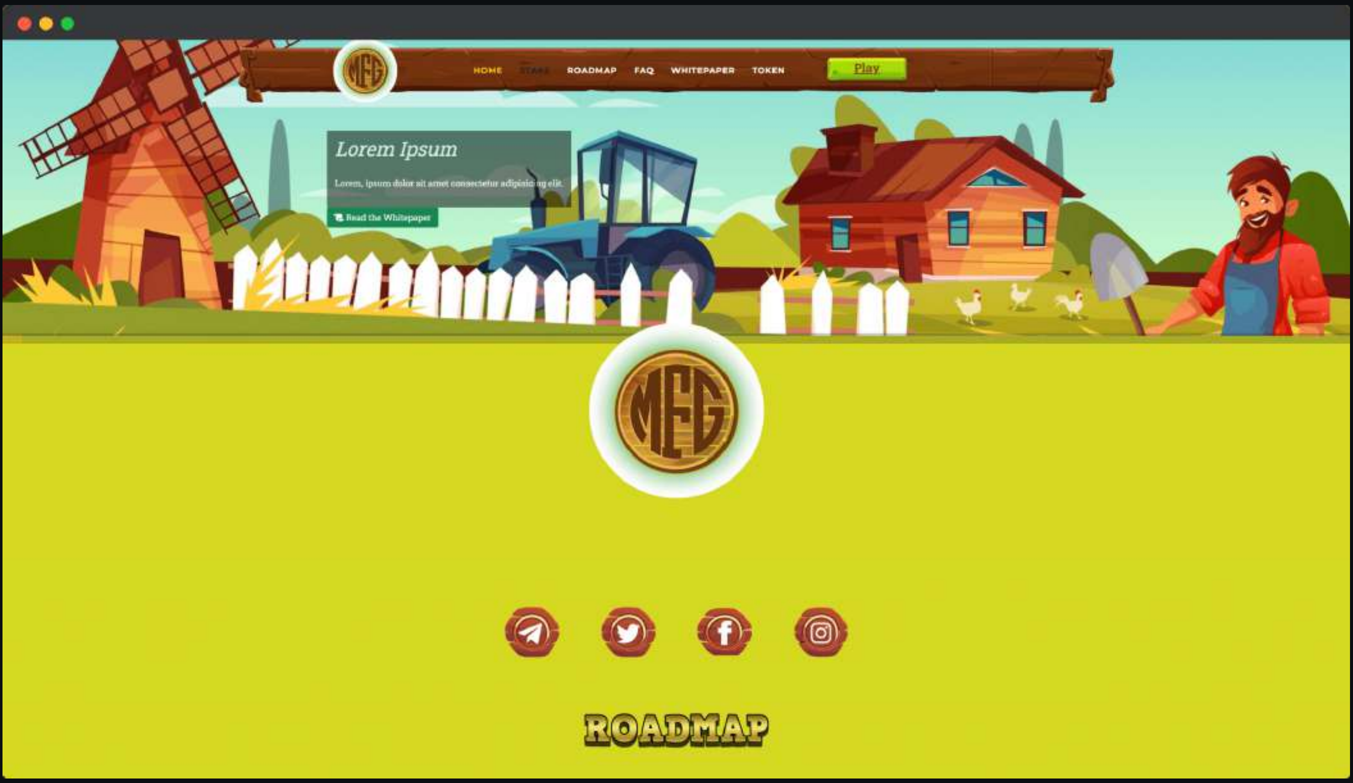
No notes

Contract Snapshot

```
contract StandardToken is IERC20, Ownable, BaseToken {
using SafeMath for uint256;

uint256 public constant VERSION = 1;

mapping(address =&gt; uint256) private _balances;
mapping(address =&gt; mapping(address =&gt; uint256)) private _allowances;

string private _name;
string private _symbol;
uint8 private _decimals;
uint256 private _totalSupply;

constructor(
    string memory name_,
    string memory symbol_,
    uint8 decimals_,
    uint256 totalSupply_,
    address serviceFeeReceiver_,
    uint256 serviceFee_
) payable {
    _name = name_;
    _symbol = symbol_;
    _decimals = decimals_;
    _mint(owner(), totalSupply_);

    emit TokenCreated(owner(), address(this), TokenType.standard, VERSION);

    payable(serviceFeeReceiver_).transfer(serviceFee_);
}

/**
 * @dev Returns the name of the token.
 */
function name() public view virtual returns (string memory) {
    return _name;
}
```

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.



Lorem Ipsum

Lorem, ipsum dolor sit amet consectetur adipisicing elit.

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

Note:

There is a large amount of lorem ipsum text.

Project Overview

● Not KYC verified by Coinsult