



Coinsult

Advanced Manual Smart Contract Audit



Project: Private Land

Website: <https://myprivateland.io/>

Low-Risk

4 low-risk code
issues found

Medium-Risk

0 medium-risk code
issues found

High-Risk

0 high-risk code
issues found

Contract Address

0xd883b14d0b2db45ff6c789ab83feba4183b6fc77

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	0xfbbf797569fcf9f02d0420b14f6492ae8428d847	10,000,000,000,000	100.0000%

Source Code

Coinsult was comissioned by Private Land to perform an audit based on the following smart contract:

<https://bscscan.com/address/0xd883b14d0b2db45ff6c789ab83feba4183b6fc77#code>

Manual Code Review

In this audit report we will highlight all these issues:

Low-Risk

4 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.

Require statement from other contract

```
function setBuyTax(uint256 buyTax) public onlyOwner {  
    require(buyTax < 25, "wofo: max tax 25%");  
    _Tax_On_Buy = buyTax;  
}  
function setSellTax(uint256 sellTax) public onlyOwner {  
    require(sellTax < 25, "wofo: max tax 25%");  
    _Tax_On_Sell = sellTax;  
}
```

Recommendation

Current require statements originate from the 'WOFO' token. This might confuse investors.

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

No zero address validation for some functions

Detect missing zero address validation.

```
function setGameAddress(address _gameAddress) external onlyOwner {
    Wallet_Game = payable(_gameAddress);
}
function setMarketingAddress(address _marketingAddress) external onlyOwner {
    Wallet_Dev = payable(_marketingAddress);
}
```

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, so Bob loses ownership of the contract.

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

Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
function setBuyTax(uint256 buyTax) public onlyOwner {
    require(buyTax <= 25, "wofo: max tax 25%");
    _Tax_On_Buy = buyTax;
}
function setSellTax(uint256 sellTax) public onlyOwner {
    require(sellTax <= 25, "wofo: max tax 25%");
    _Tax_On_Sell = sellTax;
}
```

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.

Conformance to Solidity naming conventions

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

```
address payable public Wallet_Game = payable(0xbce51C2D3607e85401d8c5045239D9a25422756A);
address payable public Wallet_Dev = payable(0xa3B936eA9c7620bb775a018071AeF7F773577746);
address payable public constant Wallet_Burn = payable(0x00000000000000000000000000000000dEaD);
```

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.

Owner privileges

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

Extra notes by the team

No notes

Contract Snapshot

```
contract PrivateLand is Context, IERC20 {
    using SafeMath for uint256;
    using Address for address;
    address private _owner;

    event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);
    event ExcludeFromFees(address indexed account, bool isExcluded);

    function owner() public view virtual returns (address) {
        return _owner;
    }

    modifier onlyOwner() {
        require(owner() == _msgSender(), "Ownable: caller is not the owner");
        _;
    }

    function renounceOwnership() external virtual onlyOwner {
        emit OwnershipTransferred(_owner, address(0));
        _owner = address(0);
    }

    function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "WOF0: new owner is the zero address");
        emit OwnershipTransferred(_owner, newOwner);
        _owner = newOwner;
    }

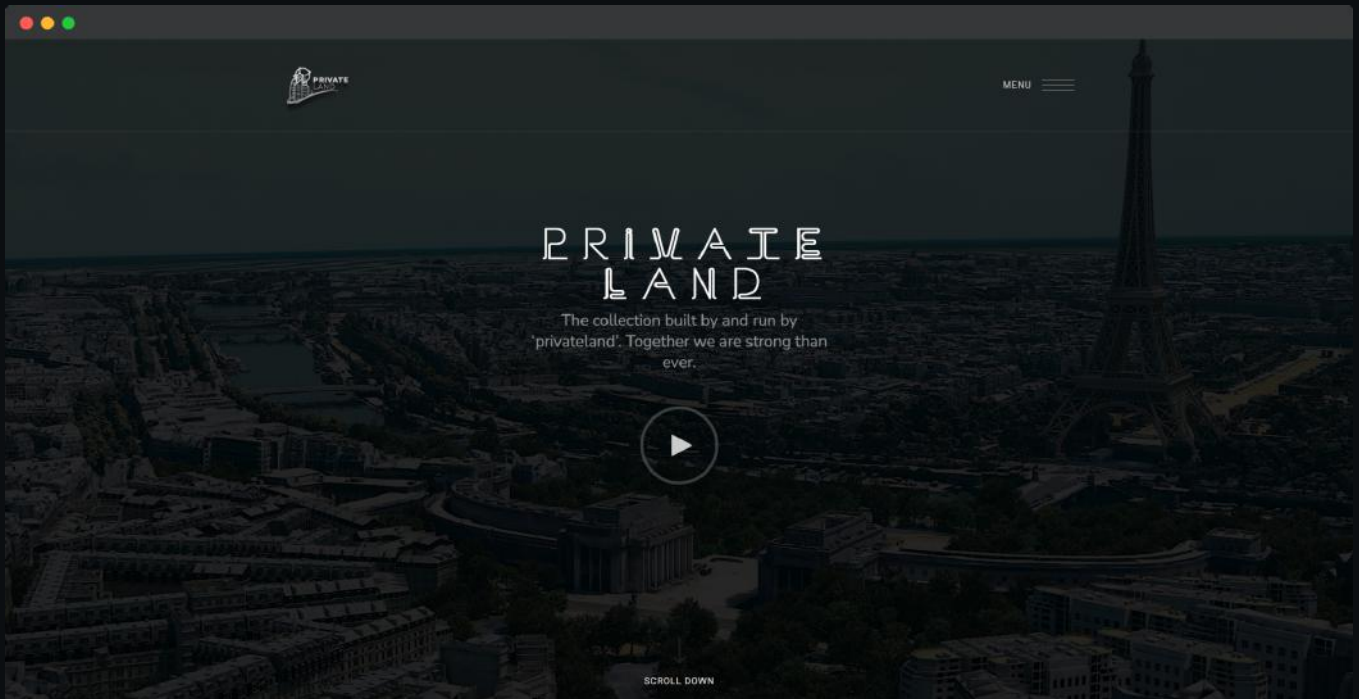
    mapping (address => uint256) private _tOwned;
    mapping (address => mapping (address => uint256)) private _allowances;
    mapping (address => bool) public _isExcludedFromFee;

    address payable public Wallet_Game = payable(0xbce51C2D3607e85401d8c5045239D9a25422756A);
    address payable public Wallet_Dev = payable(0xa3B936eA9c7620bb775a018071AeF7F773577746);
    address payable public constant Wallet_Burn = payable(0x00000000000000000000000000000000dEaD);

    uint8 private constant _decimals = 9;
```

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

Private Land

Audited by Coinsult.net



Date: 14 August 2022

✓ Advanced Manual Smart Contract Audit