Audit Report for **Eutopia**

Date: 28 August 2024

Audit result: **Passed with Medium Risk.**

**Token Address:** 0xb6551900d7FB1b51ebb29FA0143a6EBf0FB0D47F

**Name:** Eutopia 

**Symbol:** EUTO

**Decimals**: 18

**Network:** Ether Scan

**Token Type**: ERC1967Proxy

**Owner**: 0x3F7a733857300043E9c9E2707836dD5d272415cd

**Token Supply:** 4,000,000,000

**Checksum:** 5bb72de43a65baf58885adfc77e0c9ad

**Token Overview:**

**Buy Fee:** 0-13%

**Sell Fee:** 0-18%

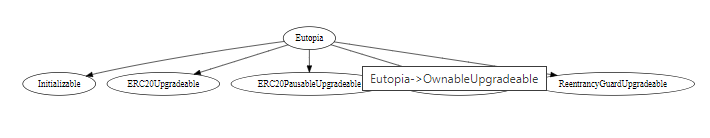
**Transfer Fee:** 0-0%

**Fee Privilege:** Owner

**Ownership:** Owned

**Minting:** No

**Max Tx:** No  
**Blacklist:** No

**Inheritance Tree**

**Ownership Privileges:**

- The owner can setFeeExempt.  
- The owner can setTargetLiquidity.  
- The owner can setFeeReceivers.  
- The owner can setFees.  
- The owner can clearStuckBalance.  
- The owner can setAutoRebase.  
- The owner can setRebaseFrequency.  
- The owner can setRewardYield.  
- The owner can setNextRebase.  
- The owner can pause/unpause..

**Findings:**

**Critical**: 0

**High**: 0

**Medium**: 3

**Low**: 1

**Informational** **&** **Optimizations**: 1  
 **Centralization - Ownership of Proxy and Implementation contract.**

**Severity**: **Medium**

**Overview:** The owner has renounced the ownership of the implementation contract but didn’t renounce the ownership of the proxy contract. Which can lead to concern by manipulating the functionality of implementation contract through proxy contract.

**Suggestion:** Make sure the owner has to renounce the ownership of the proxy contract.  
 **Centralization – Missing Require Check.**

**Severity**: **Medium**

**Function**: setFeeReceivers

**Status:** Open

**Overview:**

The owner can set any arbitrary address including zero address as this is not recommended because if the owner sets the address to the contract address, then the ETH will not be sent to that address and the transaction will fail, leading to a potential honeypot in the contract.

*function* setFeeReceivers(

        address \_liquidityReceiver,

        address \_treasuryReceiver,

        address \_essrReceiver

    ) external onlyOwner {

        liquidityReceiver **=** \_liquidityReceiver;

        treasuryReceiver **=** \_treasuryReceiver;

        essrReceiver **=** \_essrReceiver;

**emit** SetFeeReceivers(

            \_liquidityReceiver,

            \_treasuryReceiver,

            \_essrReceiver

        );

    }

**Suggestion:** It is recommended that the address should not be able to set as a contract address.  
  
**Centralization – Liquidity is added to EOA.**

**Severity**: **Medium**

**function**: \_addLiquidity

**Status:** Acknowledge

**Overview:**

Liquidity is added to EOA. It may be drained by the liquidityReceiver.

*function* \_addLiquidity(uint256 \_tokenAmount, uint256 \_ethAmount) private {

        uniswapRouter.addLiquidityETH{value**:** \_ethAmount}(

            address(**this**),

            \_tokenAmount,

*0*,

*0*,

            liquidityReceiver,

**block**.timestamp

        );

    }

**Suggestion:**

It is suggested that the address should be a contract address or a dead address but the client   
  
  
  
**Centralization – Missing Zero Address**

**Severity**: **Low**

**Subject**: Zero Check

**Status:** Open

**Overview:**

functions can take a zero address as a parameter (0x00000...). If a function parameter of address type is not properly validated by checking for zero addresses, there could be serious consequences for the contract's functionality.

*function* setFeeReceivers(

        address \_liquidityReceiver,

        address \_treasuryReceiver,

        address \_essrReceiver

    ) external onlyOwner {

        liquidityReceiver **=** \_liquidityReceiver;

        treasuryReceiver **=** \_treasuryReceiver;

        essrReceiver **=** \_essrReceiver;

**emit** SetFeeReceivers(

            \_liquidityReceiver,

            \_treasuryReceiver,

            \_essrReceiver

        );

    }

**Suggestion:**

It is suggested that the address should not be zero or dead.  
  
**Optimization**

**Severity**: **Optimization**

**Subject**: Remove unused code.

**Status:** Open

**Overview:**

Unused variables are allowed in Solidity, and they do. not pose a direct security issue. It is the best practice. though to avoid them.  
  
*function* pause() public onlyOwner {   
\_pause();   
  
}  
  
*function* unpause() public onlyOwner {   
\_pause();   
  
}