Cyberscope

Audit Report

Eagle AI

May 2024

Network	BASE
Address	0x6797b62244fa75f2e78cdffc3a4eb169332b730cc
Audited by	© cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	MFC	Misleading Fee Comments	Unresolved
●	PLPI	Potential Liquidity Provision Inadequacy	Unresolved

Table of Contents

Review

Contract Name	EAGLEAI
Compiler Version	v0.8.24+commit.e11b9ed9
Optimization	200 runs
Explorer	https://basescan.org/address/0x6797b6244fa75f2e78cdffc3a4eb169332b730cc
Address	0x6797b6244fa75f2e78cdffc3a4eb169332b730cc
Network	BASE
Symbol	EAI
Decimals	18
Total Supply	100,000,000
Badge Eligibility	Yes

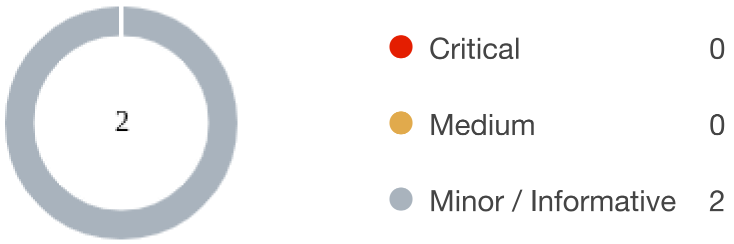
Audit Updates

Initial Audit	15 May 2024 https://github.com/cyberscope-io/audits/blob/main/1-eai/v1/audit.pdf
Corrected Phase 2	20 May 2024

Source Files

Filename	SHA256
EAGLEAI.sol	e6825a6475b53352f3559b4d9134ca9a249686fbf823056ce03b9600cf75d419

Findings Breakdown



Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	2	0	0	0

MFC - Misleading Fee Comments

Criticality	Minor / Informative
Location	EAGLEAI.sol#L384,1213
Status	Unresolved

Description

The contract is initializing the fee variables for both buying and selling transactions correctly. However, the comments associated with the segments where these fees are applied are misleading. The comments reflect incorrect fee values that do not accurately match the actual fee variables initialized in the contract. For instance, the `buyCoinWalletTaxPer` is set to 1, but the comment states it is 2%. Similarly, the `sellReflectionTax` is set to 2, but the comment mentions 1%. This discrepancy can lead to confusion and potential misinterpretation of the contract's behavior.

```
//Buy tax percentage
uint256 public buyReflectionTax=1;
uint256 public buyCoinWalletTaxPer=1;
uint256 public buyLiquidityTaxPer=1;
uint256 public buyBurnTaxPer= 0;
//sell tax percentage
uint256 public sellReflectionTax=2;
uint256 public sellCoinWalletTaxPer=1;
uint256 public sellLiquidityTaxPer=2;
uint256 public sellBurnTaxPer= 1;
…

if (isBuy) {
refAmt =    buyReflectionTax; //1 %
coinOperation = buyCoinWalletTaxPer; //2 %
liquidty = buyLiquidityTaxPer; //2 %
burn = buyBurnTaxPer; //0%
...
else if (isSell) {
refAmt = sellReflectionTax; //1%
coinOperation = sellCoinWalletTaxPer; //2%
liquidty = sellLiquidityTaxPer; //2%
burn = sellBurnTaxPer;    //0%
```

Recommendation

It is recommended to update the comments to reflect the actual fee values accurately. If the comments are deemed unnecessary or prone to being outdated, consider removing them to avoid any potential confusion. Ensuring that the comments are consistent with the actual code will enhance the clarity and maintainability of the contract.

PLPI - Potential Liquidity Provision Inadequacy

Criticality	Minor / Informative
Location	EAGLEAI.sol#L1283
Status	Unresolved

Description

The contract operates under the assumption that liquidity is consistently provided to the pair between the contract's token and the native currency. However, there is a possibility that liquidity is provided to a different pair. This inadequacy in liquidity provision in the main pair could expose the contract to risks. Specifically, during eligible transactions, where the contract attempts to swap tokens with the main pair, a failure may occur if liquidity has been added to a pair other than the primary one. Consequently, transactions triggering the swap functionality will result in a revert.

```
_approve(address(this), address(uniswapV2Router), tokenAmount);

// make the swap
uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
    tokenAmount,
    0, // accept any amount of ETH
    path,
    address(this),
    block.timestamp
);
```

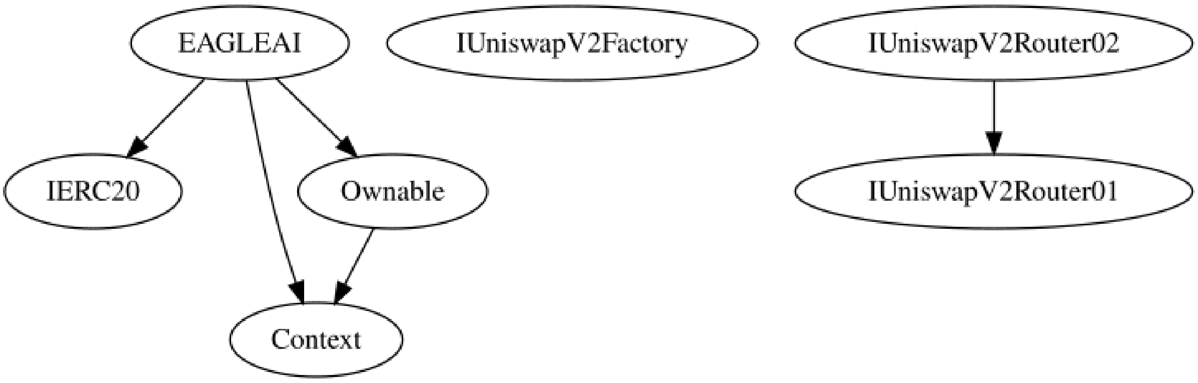
Recommendation

The team is advised to implement a runtime mechanism to check if the pair has adequate liquidity provisions. This feature allows the contract to omit token swaps if the pair does not have adequate liquidity provisions, significantly minimizing the risk of potential failures.

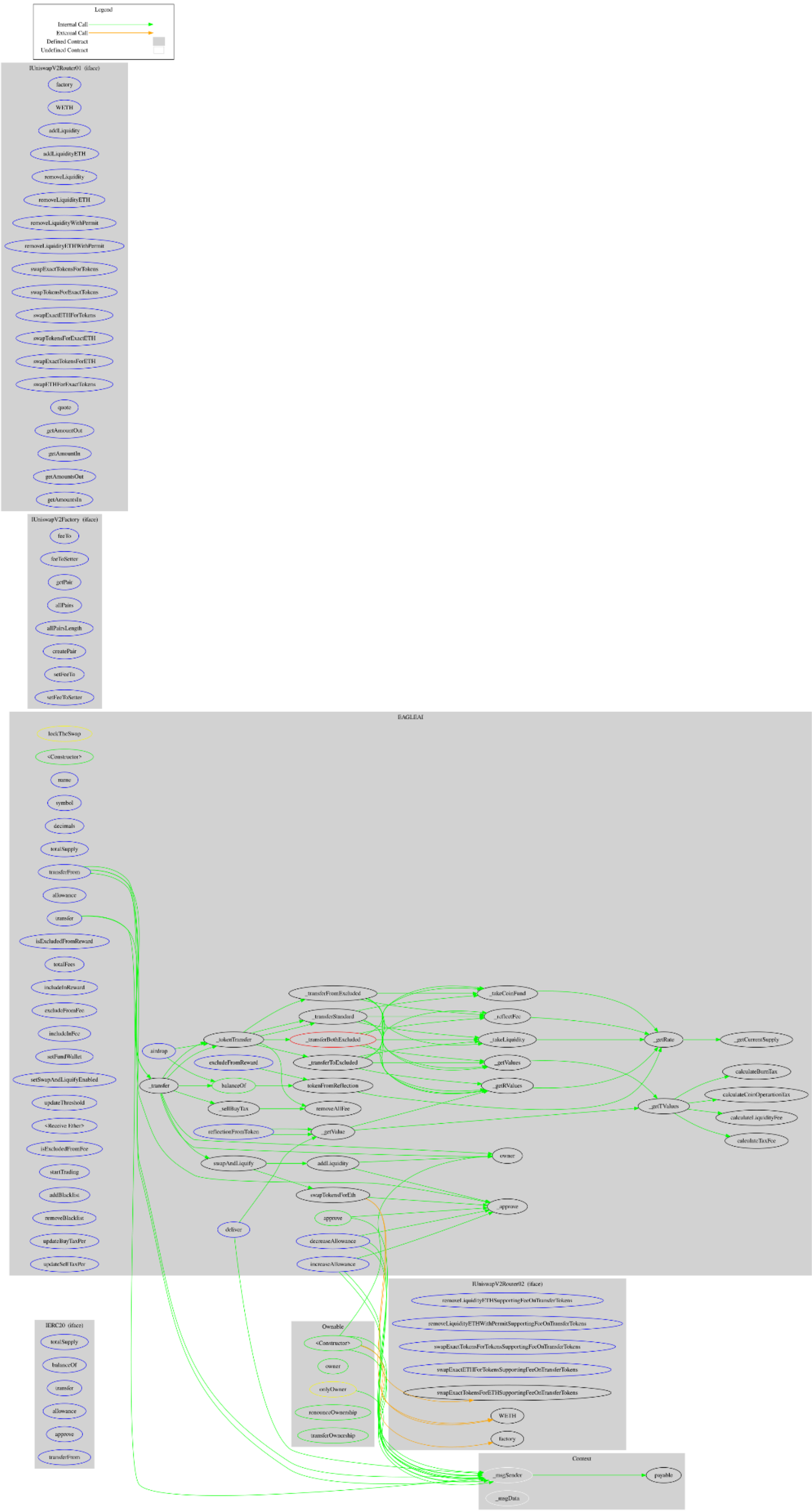
Furthermore, the team could ensure the contract has the capability to switch its active pair in case liquidity is added to another pair.

Additionally, the contract could be designed to tolerate potential reverts from the swap functionality, especially when it is a part of the main transfer flow. This can be achieved by executing the contract's token swaps in a non-reversible manner, thereby ensuring a more resilient and predictable operation.

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IUniswapV2Router01 (iface)
factory
WETH
addLiquidity
addLiquidityETH
removeLiquidity
removeLiquidityETH
removeLiquidityWithPermit
removeLiquidityETHWithPermit
swapExactTokensForTokens
swapTokensForExactTokens
swapExactETHForTokens
swapTokensForExactETH
swapExactTokensForETH
swapETHForExactTokens
quote
getAmountOut
getAmountIn
getAmountsOut
getAmountsIn
IUniswapV2Factory (iface)
feeTo
feeToSetter
getPair
allPairs
allPairsLength
createPair
setFeeTo
setFeeToSetter
EAGLEAI
lockTheSwap
<Constructor>
name
symbol
decimals
totalSupply
transferFrom
allowance
transfer
isExcludedFromReward
totalFees
includeInReward
excludeFromFee
includeInFee
setFundWallet
setSwapAndLiquifyEnabled
updateThreshold
<Receive Ether>
isExcludedFromFee
startTrading
addBlacklist
removeBlacklist
updateBuyTaxPer
updateSellTaxPer
airdrop
_transfer
_tokenTransfer
excludeFromReward
balanceOf
_sellBuyTax
reflectionFromToken
swapAndLiquify
deliver
_transferFromExcluded
_transferStandard
_transferBothExcluded
_transferToExcluded
tokenFromReflection
removeAllFee
_getValue
addLiquidity
swapTokensForEth
approve
decreaseAllowance
increaseAllowance
_takeCoinFund
_reflectFee
_takeLiquidity
_getValues
_getRValues
owner
_approve
_getRate
_getCurrentSupply
_getTValues
calculateBurnTax
calculateCoinOperartionTax
calculateLiquidityFee
calculateTaxFee
IUniswapV2Router02 (iface)
removeLiquidityETHSupportingFeeOnTransferTokens
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
swapExactTokensForTokensSupportingFeeOnTransferTokens
swapExactETHForTokensSupportingFeeOnTransferTokens
swapExactTokensForETHSupportingFeeOnTransferTokens
WETH
factory
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Ownable
<Constructor>
owner
onlyOwner
renounceOwnership
transferOwnership
Context
_msgSender
payable
_msgData

Summary

Eagle AI contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Shanghai Dragon is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of a maximum 24% fee on buy and sell transactions.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io