March 20, 2025

Express Audit Report for

Sock Coin [SOCK]

De.Fi

De.Fi

Project Summary

Project Name	Token(Sock Coin)
Address	**0x3e097678ab790aacd8b42721099b3983f40aa827**
Network	56

De.Fi

Issue ID	183
Severity	Optimization
Status	High
Description Code	**address** dead = 0x000000000000000000000000000000000000dEaD;
Location	Token.dead (Token.sol#703) should be constant

De.Fi

Issue ID	183
Severity	**Optimization**
Status	High
Description Code	**uint8 public** maxBurnFee = 10;
Location	Token.maxBurnFee (Token.sol#707) should be constant

De.Fi

Issue ID	183
Severity	Optimization
Status	High
Description Code	**uint8 public** maxBuybackFee = 10;
Location	Token.maxBuybackFee (Token.sol#709) should be constant

De.Fi

Issue ID	183
Severity	**Optimization**
Status	High
Description Code	**uint8 public** maxLiqFee = 10;
Location	Token.maxLiqFee (Token.sol#705) should be constant

De.Fi

Issue ID	183
Severity	Optimization
Status	High
Description Code	**uint8 public** maxTaxFee = 10;
Location	Token.maxTaxFee (Token.sol#706) should be constant

De.Fi

Issue ID	183
Severity	Optimization
Status	High
Description Code	**uint8 public** maxWalletFee = 10;
Location	Token.maxWalletFee (Token.sol#708) should be constant

De.Fi

Issue ID	183
Severity	Optimization
Status	High
Description Code	**uint8 public** minMxTxPercentage = 1;
Location	Token.minMxTxPercentage (Token.sol#710) should be constant

De.Fi

Issue ID	183
Severity	**Optimization**
Status	High
Description Code	**uint8 public** minMxWalletPercentage = 1;
Location	Token.minMxWalletPercentage (Token.sol#711) should be constant

De.Fi

Issue ID	183
Severity	**Optimization**
Status	High
Description Code	**bool public** mintedByMudra = true;
Location	Token.mintedByMudra (Token.sol#730) should be constant

De.Fi

Issue ID	183
Severity	Optimization
Status	High
Description Code	**address public** router = 0x10ED43C718714eb63d5aA57B78B54704E256024E;
Location	Token.router (Token.sol#722) should be constant

De.Fi

Issue ID	179
Severity	Informational
Status	Medium

Description Code	function includeInReward(address account) external onlyOwner() {
	require(_isExcluded[account], "Already excluded");
	for (uint256 i = 0; i < _excluded.length; i++) {
	if (_excluded[i] == account) {
	_excluded[i] = _excluded[_excluded.length - 1];
	_tOwned[account] = 0;
	_isExcluded[account] = false;
	_excluded.pop();
	break;
	}
	}
	}

Location	Token.includeInReward(address) (Token.sol#910-921) has costly operations inside a loop: - _excluded.pop() (Token.sol#917)

Issue ID	156
Severity	**Low**
Status	**Medium**
Description Code	

```
function swapAndLiquify(uint256
contractTokenBalance) private lockTheSwap {
//This needs to be distributed among burn, wallet
and liquidity
//burn
uint8 totFee = _burnFee + _walletFee + _liquidityFee +
_buybackFee;
uint256 spentAmount = 0;
uint256 totSpentAmount = 0;
if(_burnFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_burnFee);
_tokenTransferNoFee(address(this), dead,
spentAmount);
totSpentAmount = spentAmount;
}
if(_walletFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_walletFee);
_tokenTransferNoFee(address(this), feeWallet,
spentAmount);
totSpentAmount = totSpentAmount + spentAmount;
}
if(_buybackFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_buybackFee);
swapTokensForBNB(spentAmount);
totSpentAmount = totSpentAmount + spentAmount;
}
if(_liquidityFee != 0){
contractTokenBalance =
contractTokenBalance.sub(totSpentAmount);
// split the contract balance into halves
uint256 half = contractTokenBalance.div(2);
uint256 otherHalf = contractTokenBalance.sub(half);
// capture the contract's current ETH balance.
// this is so that we can capture exactly the amount
of ETH that the
// swap creates, and not make the liquidity event
```

Issue ID	156
Severity	**Low**
Status	**Medium**
Description Code	

```
function swapAndLiquify(uint256
contractTokenBalance) private lockTheSwap {
//This needs to be distributed among burn, wallet
and liquidity
//burn
uint8 totFee = _burnFee + _walletFee + _liquidityFee +
_buybackFee;
uint256 spentAmount = 0;
uint256 totSpentAmount = 0;
if(_burnFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_burnFee);
_tokenTransferNoFee(address(this), dead,
spentAmount);
totSpentAmount = spentAmount;
}
if(_walletFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_walletFee);
_tokenTransferNoFee(address(this), feeWallet,
spentAmount);
totSpentAmount = totSpentAmount + spentAmount;
}
if(_buybackFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_buybackFee);
swapTokensForBNB(spentAmount);
totSpentAmount = totSpentAmount + spentAmount;
}
if(_liquidityFee != 0){
contractTokenBalance =
contractTokenBalance.sub(totSpentAmount);
// split the contract balance into halves
uint256 half = contractTokenBalance.div(2);
uint256 otherHalf = contractTokenBalance.sub(half);
// capture the contract's current ETH balance.
// this is so that we can capture exactly the amount
of ETH that the
// swap creates, and not make the liquidity event
```

Issue ID	156
Severity	**Low**
Status	**Medium**
Description Code	

```
function swapAndLiquify(uint256
contractTokenBalance) private lockTheSwap {
//This needs to be distributed among burn, wallet
and liquidity
//burn
uint8 totFee = _burnFee + _walletFee + _liquidityFee +
_buybackFee;
uint256 spentAmount = 0;
uint256 totSpentAmount = 0;
if(_burnFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_burnFee);
_tokenTransferNoFee(address(this), dead,
spentAmount);
totSpentAmount = spentAmount;
}
if(_walletFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_walletFee);
_tokenTransferNoFee(address(this), feeWallet,
spentAmount);
totSpentAmount = totSpentAmount + spentAmount;
}
if(_buybackFee != 0){
spentAmount =
contractTokenBalance.div(totFee).mul(_buybackFee);
swapTokensForBNB(spentAmount);
totSpentAmount = totSpentAmount + spentAmount;
}
if(_liquidityFee != 0){
contractTokenBalance =
contractTokenBalance.sub(totSpentAmount);
// split the contract balance into halves
uint256 half = contractTokenBalance.div(2);
uint256 otherHalf = contractTokenBalance.sub(half);
// capture the contract's current ETH balance.
// this is so that we can capture exactly the amount
of ETH that the
// swap creates, and not make the liquidity event
```

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function name**() **public view returns** (**string memory**) {
	return _name;
	}

Location	name() should be declared external: - Token.name() (Token.sol#816-818)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function symbol**() **public view returns** (**string memory**) {
	return _symbol;
	}

Location	symbol() should be declared external: - Token.symbol() (Token.sol#820-822)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function decimals**() **public view returns** (**uint8**) {
	return _decimals;
	}

Location	decimals() should be declared external: - Token.decimals() (Token.sol#824-826)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function totalSupply**() **public view override returns** (**uint256**) {
	return _tTotal;
	}

Location	totalSupply() should be declared external: - Token.totalSupply() (Token.sol#828-830)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function transfer**(**address** recipient, **uint256** amount) **public override returns** (**bool**) {
	_transfer(_msgSender(), recipient, amount);
	return true;
	}

Location	transfer(address,uint256) should be declared external: - Token.transfer(address,uint256) (Token.sol#837-840)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function allowance**(**address** owner, **address** spender) **public view override returns** (**uint256**) {
	return _allowances[owner][spender];
	}

Location	allowance(address,address) should be declared external: - Token.allowance(address,address) (Token.sol#842-844)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function approve**(**address** spender, **uint256** amount) **public override returns** (**bool**) {
	_approve(_msgSender(), spender, amount);
	return true;
	}

Location	approve(address,uint256) should be declared external: - Token.approve(address,uint256) (Token.sol#846-849)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function transferFrom**(**address** sender, **address** recipient, **uint256** amount) **public override returns** (**bool**) {
	_transfer(sender, recipient, amount);
	_approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount, "ERC20: transfer amount exceeds allowance"));
	return true;
	}

Location	transferFrom(address,address,uint256) should be declared external: - Token.transferFrom(address,address,uint256) (Token.sol#851-855)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function increaseAllowance**(**address** spender, **uint256** addedValue) **public virtual returns** (**bool**) {
	_approve(_msgSender(), spender,
	_allowances[_msgSender()]
	[spender].add(addedValue));
	return true;
	}

Location	increaseAllowance(address,uint256) should be declared external: - Token.increaseAllowance(address,uint256) (Token.sol#857-860)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function decreaseAllowance**(**address** spender, **uint256** subtractedValue) **public virtual returns** (**bool**) {
	_approve(_msgSender(), spender, _allowances[_msgSender()][spender].sub(subtractedValue, "ERC20: decreased allowance below zero"));
	return true;
	}

Location	decreaseAllowance(address,uint256) should be declared external: - Token.decreaseAllowance(address,uint256) (Token.sol#862-865)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function isExcludedFromReward**(**address** account) **public view returns** (**bool**) {
	return _isExcluded[account];
	}

Location	isExcludedFromReward(address) should be declared external: - Token.isExcludedFromReward(address) (Token.sol#867-869)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function totalFees**() **public view returns** (**uint256**) {
	return _tFeeTotal;
	}

Location	totalFees() should be declared external: - Token.totalFees() (Token.sol#871-873)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function deliver**(**uint256** tAmount) **public** {
	address sender = _msgSender();
	require(!_isExcluded[sender], "Excluded addresses cannot call this function");
	(**uint256** rAmount,,,,,) = _getValues(tAmount);
	_rOwned[sender] = _rOwned[sender].sub(rAmount);
	_rTotal = _rTotal.sub(rAmount);
	_tFeeTotal = _tFeeTotal.add(tAmount);
	}

Location	deliver(uint256) should be declared external: - Token.deliver(uint256) (Token.sol#875-882)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function reflectionFromToken**(**uint256** tAmount, **bool** deductTransferFee) **public view returns**(**uint256**) {
	require(tAmount <= _tTotal, "Amt must be less than supply");
	if (!deductTransferFee) {
	(**uint256** rAmount,,,,,) = _getValues(tAmount);
	return rAmount;
	} **else** {
	(,**uint256** rTransferAmount,,,,) = _getValues(tAmount);
	return rTransferAmount;
	}
	}

Location	reflectionFromToken(uint256,bool) should be declared external: - Token.reflectionFromToken(uint256,bool) (Token.sol#884-893)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	function excludeFromReward(address account) public onlyOwner() {
	require(!_isExcluded[account], "Account is already excluded from reward");
	if(_rOwned[account] > 0) {
	_tOwned[account] = tokenFromReflection(_rOwned[account]);
	}
	_isExcluded[account] = true;
	_excluded.push(account);
	}

Location	excludeFromReward(address) should be declared external: - Token.excludeFromReward(address) (Token.sol#901-908)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	function excludeFromFee(address account) public onlyOwner {
	_isExcludedFromFee[account] = true;
	}

Location	excludeFromFee(address) should be declared external: - Token.excludeFromFee(address) (Token.sol#924-926)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function includeInFee**(**address** account) **public onlyOwner** {
	_isExcludedFromFee[account] = false;
	}

Location	includeInFee(address) should be declared external: - Token.includeInFee(address) (Token.sol#928-930)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function buyBackUpperLimitAmount**() **public view returns** (**uint256**) {
	return buyBackUpperLimit;
	}

Location	buyBackUpperLimitAmount() should be declared external: - Token.buyBackUpperLimitAmount() (Token.sol#945-947)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
	swapAndLiquifyEnabled = _enabled;
	emit SwapAndLiquifyEnabledUpdated(_enabled);
	}

Location	setSwapAndLiquifyEnabled(bool) should be declared external: - Token.setSwapAndLiquifyEnabled(bool) (Token.sol#967-970)

De.Fi

Issue ID	184
Severity	Optimization
Status	High

Description Code	**function isExcludedFromFee**(**address** account) **public view returns**(**bool**) {
	return _isExcludedFromFee[account];
	}

Location	isExcludedFromFee(address) should be declared external: - Token.isExcludedFromFee(address) (Token.sol#1068-1070)

De.Fi

Issue ID	184
Severity	**Optimization**
Status	**High**

Description Code	**function recoverBEP20**(**address** tokenAddress, **uint256** tokenAmount) **public onlyOwner** {
	// do not allow recovering self token
	require(tokenAddress != **address**(this), "Self withdraw");
	IERC20(tokenAddress).transfer(owner(), tokenAmount);
	}

Location	recoverBEP20(address,uint256) should be declared external: - Token.recoverBEP20(address,uint256) (Token.sol#1325-1329)

De.Fi

Issue ID	172
Severity	Informational
Status	High
Description Code	**uint8 private** _previousTaxFee = _taxFee;
Location	Token._previousTaxFee (Token.sol#737) is set pre-construction with a non-constant function or state variable: - _taxFee

De.Fi

Issue ID	172
Severity	**Informational**
Status	High
Description Code	**uint8 private** _previousLiquidityFee = _liquidityFee;
Location	Token._previousLiquidityFee (Token.sol#740) is set pre-construction with a non-constant function or state variable: - _liquidityFee

De.Fi

Issue ID	172
Severity	Informational
Status	High
Description Code	**uint8 private** _previousBurnFee = _burnFee;
Location	Token._previousBurnFee (Token.sol#743) is set pre-construction with a non-constant function or state variable: - _burnFee

De.Fi

Issue ID	172
Severity	Informational
Status	High
Description Code	**uint8 private** _previousWalletFee = _walletFee;
Location	Token._previousWalletFee (Token.sol#746) is set pre-construction with a non-constant function or state variable: - _walletFee

De.Fi

Issue ID	172
Severity	**Informational**
Status	High
Description Code	**uint8 private** _previousBuybackFee = _buybackFee;
Location	Token._previousBuybackFee (Token.sol#749) is set pre-construction with a non-constant function or state variable: - _buybackFee

De.Fi

Issue ID	173
Severity	**Informational**
Status	**High**

Description Code	**function sendValue**(**address payable** recipient, **uint256** amount) **internal** {
	require(**address**(this).balance >= amount, "Address: insufficient balance");
	// solhint-disable-next-line avoid-low-level-calls, avoid-call-value
	(**bool** success,) = recipient.call{ value: amount }("");
	require(success, "Address: unable to send value, recipient may have reverted");
	}

Location	Low level call in Address.sendValue(address,uint256) (Token.sol#293-299): - (success) = recipient.call{value: amount}() (Token.sol#297)

De.Fi

Issue ID	173
Severity	**Informational**
Status	**High**

Description Code	**function _functionCallWithValue**(**address** target, **bytes memory** data, **uint256** weiValue, **string memory** errorMessage) **private returns** (**bytes memory**) {
	require(isContract(target), "Address: call to non-contract");
	// solhint-disable-next-line avoid-low-level-calls
	(**bool** success, **bytes memory** returndata) = target.call{ value: weiValue }(data);
	if (success) {
	return returndata;
	} **else** {
	// Look for revert reason and bubble it up if present
	if (returndata.length > 0) {
	// The easiest way to bubble the revert reason is using memory via assembly
	// solhint-disable-next-line no-inline-assembly
	assembly {
	let returndata_size := mload(returndata)
	revert(add(32, returndata), returndata_size)
	}
	} **else** {
	revert(errorMessage);
	}
	}
	}

Location	Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (Token.sol#359-380): - (success,returndata) = target.call{value: weiValue}(data) (Token.sol#363)

De.Fi

Issue ID	167-a
Severity	**Low**
Status	**Medium**

Description Code	**function setAllFeePercent**(**uint8** taxFee, **uint8** liquidityFee, **uint8** burnFee, **uint8** walletFee, **uint8** buybackFee) **external onlyOwner**() {
	require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
	require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
	require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
	require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
	require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
	_taxFee = taxFee;
	_liquidityFee = liquidityFee;
	_burnFee = burnFee;
	_buybackFee = buybackFee;
	_walletFee = walletFee;
	}

Location	Token.setAllFeePercent(uint8,uint8,uint8,uint8,uint8) (Token.sol#932-943) should emit an event for: - _taxFee = taxFee (Token.sol#938) - _liquidityFee = liquidityFee (Token.sol#939) - _burnFee = burnFee (Token.sol#940) - _buybackFee = buybackFee (Token.sol#941) - _walletFee = walletFee (Token.sol#942)

De.Fi

Issue ID	167-a
Severity	**Low**
Status	**Medium**
Description Code	**function setBuybackUpperLimit(uint256** buyBackLimit) **external onlyOwner**() { buyBackUpperLimit = buyBackLimit * 10**18; }
Location	Token.setBuybackUpperLimit(uint256) (Token.sol#949-951) should emit an event for: - buyBackUpperLimit = buyBackLimit * 10 ** 18 (Token.sol#950)

De.Fi

Issue ID	167-a
Severity	Low
Status	Medium

Description Code	**function setMaxTxPercent(uint256** maxTxPercent) **external onlyOwner**() {
	require(maxTxPercent >= minMxTxPercentage &&
	maxTxPercent <=100,"err");
	_maxTxAmount = _tTotal.mul(maxTxPercent).div(
	10**2
	);
	}

Location	Token.setMaxTxPercent(uint256) (Token.sol#953-958) should emit an event for: - _maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 2) (Token.sol#955-957)

De.Fi

Issue ID	167-a
Severity	Low
Status	Medium

Description Code	function setMaxWalletPercent(uint256 maxWalletPercent) external onlyOwner() {
	require(maxWalletPercent >= minMxWalletPercentage && maxWalletPercent <=100,"err");
	_maxWalletAmount = _tTotal.mul(maxWalletPercent).div(10**2);
	}

Location	Token.setMaxWalletPercent(uint256) (Token.sol#960-965) should emit an event for: - _maxWalletAmount = _tTotal.mul(maxWalletPercent).div(10 ** 2) (Token.sol#962-964)

De.Fi

Issue ID	168
Severity	**Low**
Status	**Medium**
Description Code	**address payable** _feeWallet
Location	Token.constructor(address,string,string,uint8,uint256, uint8,uint8,address)._feeWallet (Token.sol#779) lacks a zero-check on : - feeWallet = _feeWallet (Token.sol#790)

De.Fi

Issue ID	182
Severity	**Informational**
Status	**Medium**
Description Code	**address** dead = 0x000000000000000000000000000000000000dEaD;
Location	Contract Token uses literals with too many digits: - dead = 0x000000000000000000000000000000000000dEaD (Token.sol#703)

De.Fi

Issue ID	209
Severity	Critical
Status	High

Description Code	**function transfer**(**address** recipient, **uint256** amount) **public override returns** (**bool**) {
	_transfer(_msgSender(), recipient, amount);
	return true;
	}

Location	Transfer Fee: Token.transfer(address,uint256) (Token.sol#837-840) - in nested function: _getTValues - in expression: _amount.mul(_taxFee).div(10 ** 2) - in expression: _amount.mul(_liquidityFee + _burnFee + _walletFee + _buybackFee).div(10 ** 2)

De.Fi

Issue ID	7
Severity	Data
Status	High
Description Code	
Location	Transfer fee variables

De.Fi

Issue ID	8
Severity	Data
Status	High
Description Code	
Location	Transfer fee limits

De.Fi

Issue ID	211
Severity	Critical
Status	High

Description Code	**function transferFrom**(**address** sender, **address** recipient, **uint256** amount) **public override returns** (**bool**) {
	_transfer(sender, recipient, amount);
	_approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount, "ERC20: transfer amount exceeds allowance"));
	return true;
	}

Location	Transfer amount limits in: Token.transferFrom(address,address,uint256) (Token.sol#851-855) - In expression: amount <= _maxTxAmount - In expression: contractBalanceRecepient + amount - In expression: contractBalanceRecepient + amount <= _maxWalletAmount

De.Fi

Issue ID	211
Severity	Critical
Status	High

Description Code	**function transfer**(**address** recipient, **uint256** amount) **public override returns** (**bool**) {
	_transfer(_msgSender(), recipient, amount);
	return true;
	}

Location	Transfer amount limits in: Token.transfer(address,uint256) (Token.sol#837-840) - In expression: amount <= _maxTxAmount - In expression: contractBalanceRecepient + amount - In expression: contractBalanceRecepient + amount <= _maxWalletAmount

De.Fi

Issue ID	6
Severity	**Data**
Status	**High**
Description Code	
Location	Transfer limits

De.Fi

Issue ID	152
Severity	**Informational**
Status	**High**

Description Code	**function setAllFeePercent**(**uint8** taxFee, **uint8** liquidityFee, **uint8** burnFee, **uint8** walletFee, **uint8** buybackFee) **external onlyOwner**() {
	require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
	require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
	require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
	require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
	require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
	_taxFee = taxFee;
	_liquidityFee = liquidityFee;
	_burnFee = burnFee;
	_buybackFee = buybackFee;
	_walletFee = walletFee;
	}

Location	Token.setAllFeePercent(uint8,uint8,uint8,uint8,uint8) (Token.sol#932-943) contains a tautology or contradiction: - require(bool,string)(burnFee >= 0 && burnFee <= maxBurnFee,BF err) (Token.sol#935)

De.Fi

Issue ID	152
Severity	Informational
Status	High

Description Code	**function setAllFeePercent**(**uint8** taxFee, **uint8** liquidityFee, **uint8** burnFee, **uint8** walletFee, **uint8** buybackFee) **external onlyOwner**() {
	require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
	require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
	require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
	require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
	require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
	_taxFee = taxFee;
	_liquidityFee = liquidityFee;
	_burnFee = burnFee;
	_buybackFee = buybackFee;
	_walletFee = walletFee;
	}

Location	Token.setAllFeePercent(uint8,uint8,uint8,uint8,uint8) (Token.sol#932-943) contains a tautology or contradiction: - require(bool,string)(buybackFee >= 0 && buybackFee <= maxBuybackFee,BBF err) (Token.sol#937)

De.Fi

Issue ID	152
Severity	Informational
Status	High

Description Code	**function setAllFeePercent**(**uint8** taxFee, **uint8** liquidityFee, **uint8** burnFee, **uint8** walletFee, **uint8** buybackFee) **external onlyOwner**() {
	require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
	require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
	require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
	require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
	require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
	_taxFee = taxFee;
	_liquidityFee = liquidityFee;
	_burnFee = burnFee;
	_buybackFee = buybackFee;
	_walletFee = walletFee;
	}

Location	Token.setAllFeePercent(uint8,uint8,uint8,uint8,uint8) (Token.sol#932-943) contains a tautology or contradiction: - require(bool,string)(walletFee >= 0 && walletFee <= maxWalletFee,WF err) (Token.sol#936)

De.Fi

Issue ID	152
Severity	Informational
Status	High

Description Code	**function setAllFeePercent**(**uint8** taxFee, **uint8** liquidityFee, **uint8** burnFee, **uint8** walletFee, **uint8** buybackFee) **external onlyOwner**() {
	require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
	require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
	require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
	require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
	require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
	_taxFee = taxFee;
	_liquidityFee = liquidityFee;
	_burnFee = burnFee;
	_buybackFee = buybackFee;
	_walletFee = walletFee;
	}

Location	Token.setAllFeePercent(uint8,uint8,uint8,uint8,uint8) (Token.sol#932-943) contains a tautology or contradiction: - require(bool,string)(taxFee >= 0 && taxFee <= maxTaxFee,TF err) (Token.sol#933)

De.Fi

Issue ID	152
Severity	Informational
Status	High

Description Code	**function setAllFeePercent**(**uint8** taxFee, **uint8** liquidityFee, **uint8** burnFee, **uint8** walletFee, **uint8** buybackFee) **external onlyOwner**() {
	require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
	require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
	require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
	require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
	require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
	_taxFee = taxFee;
	_liquidityFee = liquidityFee;
	_burnFee = burnFee;
	_buybackFee = buybackFee;
	_walletFee = walletFee;
	}

Location	Token.setAllFeePercent(uint8,uint8,uint8,uint8,uint8) (Token.sol#932-943) contains a tautology or contradiction: - require(bool,string)(liquidityFee >= 0 && liquidityFee <= maxLiqFee,LF err) (Token.sol#934)

De.Fi

Issue ID	158
Severity	**Informational**
Status	**Medium**

Description Code	**function recoverBEP20**(**address** tokenAddress, **uint256** tokenAmount) **public onlyOwner** {
	// do not allow recovering self token
	require(tokenAddress != **address**(this), "Self withdraw");
	IERC20(tokenAddress).transfer(owner(), tokenAmount);
	}

Location	Token.recoverBEP20(address,uint256) (Token.sol#1325-1329) ignores return value by IERC20(tokenAddress).transfer(owner(),tokenAmount) (Token.sol#1328)

De.Fi

Issue ID	237
Severity	Low
Status	High

Description Code	**function transferFrom**(**address** sender, **address** recipient, **uint256** amount) **public override returns** (**bool**) {
	_transfer(sender, recipient, amount);
	_approve(sender, _msgSender(), _allowances[sender][_msgSender()].sub(amount, "ERC20: transfer amount exceeds allowance"));
	return true;
	}

Location	whitelisted function: Token.transferFrom(address,address,uint256) (Token.sol#851-855) - in internal call: Token._transfer(address,address,uint256) (Token.sol#1080-1141) - in expression _isExcludedFromFee[from]

De.Fi

Issue ID	237
Severity	Low
Status	High

Description Code	**function transfer**(**address** recipient, **uint256** amount) **public override returns** (**bool**) {
	_transfer(_msgSender(), recipient, amount);
	return true;
	}

Location	whitelisted function: Token.transfer(address,uint256) (Token.sol#837-840) - in internal call: Token._transfer(address,address,uint256) (Token.sol#1080-1141) - in expression _isExcludedFromFee[from]