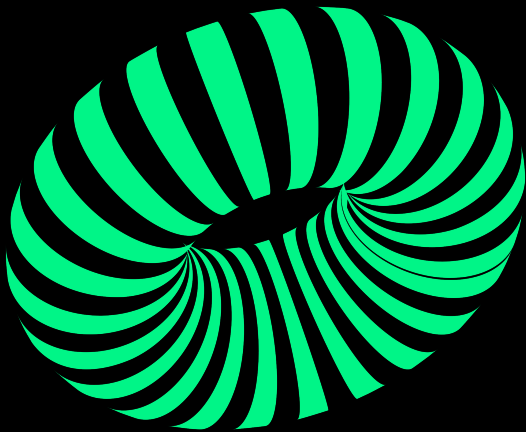


Advanced Manual

Smart Contract Audit



Audit (Simple Report)

PEPE

Decembre 2024



Network	ETH
Address	0xd34a8b2a6ec4a0775c315181d4fe9ce9eeb78300
Audited by	Oxaudit

Table of Contents

Table of Contents	1
Contract Review	2
Source Files	2
Audit Updates	2
Contract Analysis	4
OCTD - Owner Contract Tokens Drain	5
Description	5
Recommendation	5
Contract Diagnostics	6
UC - Unused Code	7
Description	7
Recommendation	7
L01 - Public Function could be Declared External	8
Description	8
Recommendation	8
L02 - State Variables could be Declared Constant	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
L05 - Unused State Variable	11
Description	11
Recommendation	11
Contract Functions	12
Contract Flow	15



Summary	16
Disclaimer	17
About Oxaudit	18



Contract Review

Contract Name	PEPE
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xd34a8b2a6ec4a0775c315181d4fe9ce9eeb78300
Symbol	PP
Decimals	9
Total Supply	100,000,000,000,000,000
Domain	pepe.monster

Source Files

Filename	SHA256
contract.sol	be6dcce897c9de217ab0f36b17d3d0e4228c58902b7e3f24b1c330e1907dc45c

Audit Updates

Initial Audit	19th April 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor ● Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L397,403

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `manualswap` and `manualsend` functions.

```
function manualswap() external {
    require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

function manualsend() external {
    require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
    uint256 contractETHBalance = address(this).balance;
    sendETHToFee(contractETHBalance);
}
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

● Critical
● Medium
● Minor

Severity	Code	Description
●	UC	Unused Code
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable

UC - Unused Code

Criticality	minor
Location	contract.sol#435

Description

There are segments that contain unused public functions.

```
function noUse() public only Dev {

}
```

Recommendation

Remove the unused public function



L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L128,134,203,207,211,215,223,228,232,235,240,322,328,335,409,420,424,428

Description

Public functions that are never called by the contract should be declared external to save gas.

```
exclude Multiple AccountsFromFees
toggle Swap
noUse
set Fee
set NewMarketing Address
set NewDevAddress
rescue ForeignTokens
transfer From
approve
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L111

Description

Constant state variables should be declared constant to save gas.

```
_previous Owner
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L52,321,327,334,322,424,151,164,165,166

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow `_` at the beginning of the mixed_case match for private variables and unused parameters.

```

_decimals
_symbol
_name
_tTotal
_swapEnabled
_amount
_to
_tokenAddr
marketingAddressUpdated
...

```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>

L05- Unused State Variable

Criticality	minor
Location	contract.sol#L111,146

Description

There are segments that contain unused state variables.

```
_tOwned
_previousOwner
```

Recommendation

Remove unused state variables.



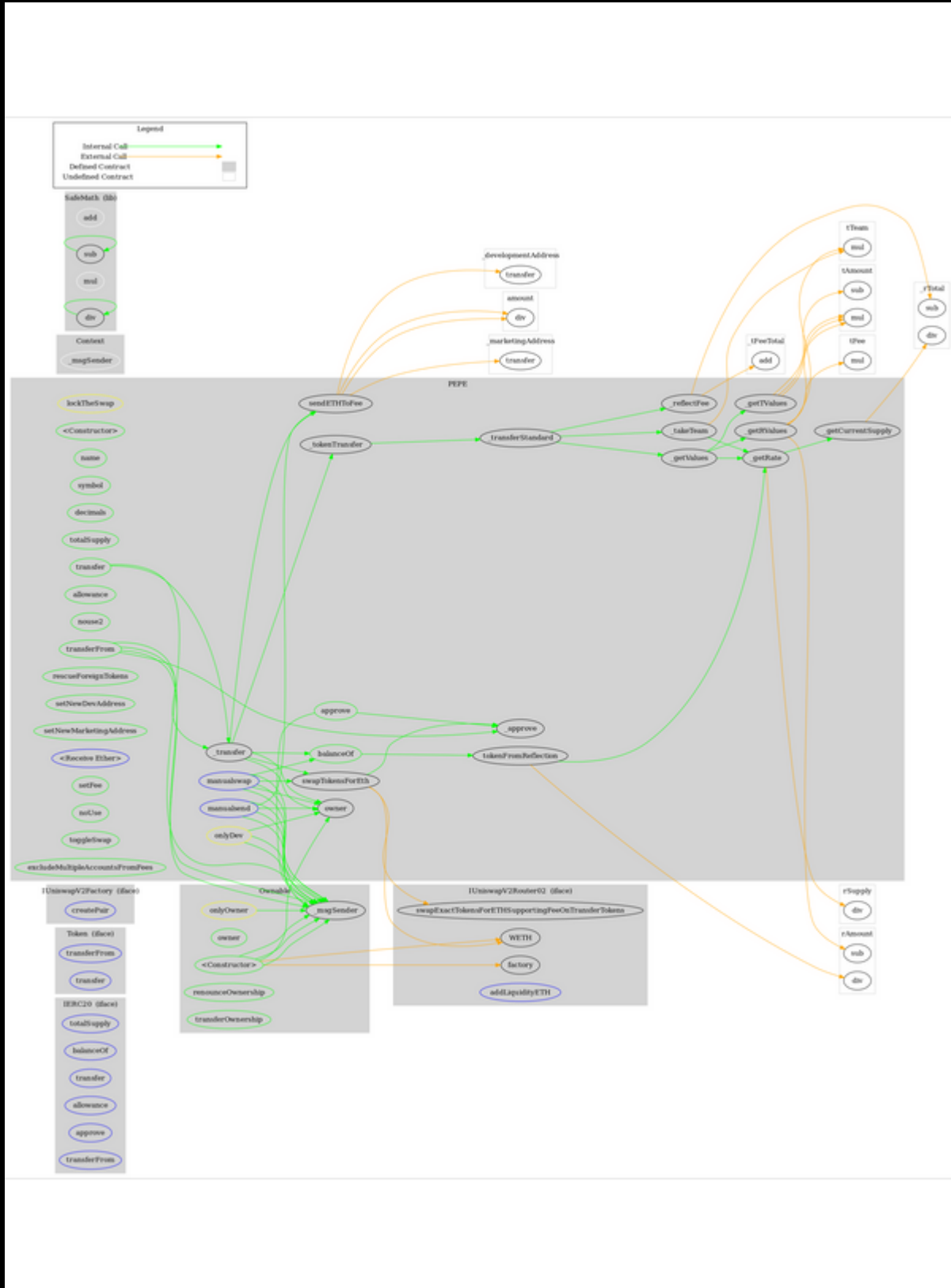
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Token	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
Context	Implementation			
	_msgSender	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		

	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
Ownable	Implementation	Cont ext		
	<Construct or>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
PEPE	Implementation	Cont ext, IERC20, Ownable		
	<Construct or>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	nouse2	Public		-
	approve	Public	✓	-
	transferF rom	Public	✓	-
	tokenFromRe flection	Private		
	_approve	Private	✓	
	_transfer	Private	✓	
	swapT okensF orE th	Private	✓	lockTheSwap
	sendETHT oFee	Private	✓	
	_tokenTransfer	Private	✓	
	rescueF oreignTokens	Public	✓	onlyDev
	setNewDevAddr ess	Public	✓	onlyDev
	setNewMark etingAddress	Public	✓	onlyDev
	_transferS tandard	Private	✓	
	_takeTeam	Private	✓	

	_reflectFee	Private	✓	
	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getTV alues	Private		
	_getRV alues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	manualswap	External	✓	-
	manualsend	External	✓	-
	setFee	Public	✓	onlyDev
	noUse	Public	✓	onlyDev
	toggleSwap	Public	✓	onlyDev
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner

Contract Flow



Summary

PEPE Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The ownership is present and the team can manually drain the contract's balance from accumulated fees. There is also a limit of max 16% fees.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Oxaudit team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Oxaudit team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Oxaudit receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Oxaudit team disclaims any liability for the resulting losses.



About Oxaudit

OxAudit, formerly known as Reflecting Audit, has rebranded to establish itself as a premier authority in blockchain security. With expertise spanning Smart Contract Audits, DApp Security Audits, Layer 2 (L2) Solutions Audits, and Penetration Testing, OxAudit delivers comprehensive, tiered services tailored to meet the needs of various blockchain projects.

The OxAudit process involves a meticulous analysis of smart contracts and decentralized applications, categorizing vulnerabilities into High, Medium, and Low risks. By providing detailed, actionable reports and a streamlined generic score, OxAudit empowers users with the insights needed to trust the projects they interact with.

At its core, OxAudit is committed to making blockchain safer and smarter for everyone. Through its free-to-use tools, structured education modules, and user-focused auditing framework, OxAudit equips the blockchain community with the essential resources to draw their own informed conclusions while fostering trust in the rapidly evolving crypto space.



Oxaudit team

<https://www.oxaudit.com>