AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Porn AI

DATED : 10 May 23'

AUDIT SUMMARY

Project name – Porn AI

Date: 10 May, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:

a line by line code review has been performed by audit ace team.

2- BSC Test Network:

all tests were done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

https://testnet.bscscan.com/token/0x85eFA81cDD6 6cf6766Ae7925fFE5311A03380d35#balances

Token Information

Token Name : PornAI

Token Symbol: PORN

Decimals: 18

Token Supply:69,000,000

Token Address:
0x7D82b51fF67F94189c7C59A6648fDC34B3038C5E

Checksum:
6c9983423d2f18b11ac8f73f5c1a834214cceafa

Owner:
0x24B1d0e4b3C9c9dF26076D40564A1aebfC06E737

TOKEN OVERVIEW

Fees:

Buy Fees: 0 %

Sell Fees: 0 %

Transfer Fees: 0%

Fees Privilige: none

Ownership : none

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: none

AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.

- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.

- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.

- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.

- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.

- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

VULNERABILITY CHECKLIST

- ✅ Return values of low-level calls
- ✅ **Gasless Send**
- ✅ Private modifier
- ✅ Using block.timestamp
- ✅ Multiple Sends
- ✅ Re-entrancy
- ✅ Using Suicide
- ✅ Tautology or contradiction
- ✅ Gas Limitand Loops
- ✅ Timestamp Dependence
- ✅ Address hardcoded
- ✅ Revert/require functions
- ✅ Exception Disorder
- ✅ Use of tx.origin
- ✅ Using inline assembly
- ✅ Integer overflow/underflow
- ✅ Divide before multiply
- ✅ Dangerous strict equalities
- ✅ Missing Zero Address Validation
- ✅ Using SHA3
- ✅ Compiler version not fixed
- ✅ Using throw

CLASSIFICATION OF RISK

Severity	Description
◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE

POINTS TO NOTE

- Owner is not able to set buy/sell/transfer taxe (0% all)
- Owner is not able to set a max buy/transfer/wallet/sell amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
FullFeatureToken	Implementation	ERC20, ERC20Burnable, ERC20Pausable, Ownable		
└	\<Constructor\>	Public ❗	🛑	ERC20
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	isPausable	Public ❗		NO❗
└	isMintable	Public ❗		NO❗
└	isBurnable	Public ❗		NO❗
└	isBlacklistEnabled	Public ❗		NO❗
└	isWhitelistEnabled	Public ❗		NO❗
└	isMaxAmountOfTokensSet	Public ❗		NO❗
└	isDocumentUriAllowed	Public ❗		NO❗
└	isForceTransferAllowed	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	getWhitelistedAddresses	External ❗		NO❗
└	setDocumentUri	External ❗	🛑	onlyOwner
└	setMaxTokenAmountPerAddress	External ❗	🛑	onlyOwner
└	blackList	External ❗	🛑	onlyOwner whenNotPaused
└	removeFromBlacklist	External ❗	🛑	onlyOwner whenNotPaused
└	transfer	Public ❗	🛑	whenNotPaused validateTransfer
└	transferFrom	Public ❗	🛑	whenNotPaused validateTransfer
└	mint	External ❗	🛑	onlyOwner whenNotPaused
└	burn	Public ❗	🛑	onlyOwner whenNotPaused
└	burnFrom	Public ❗	🛑	onlyOwner whenNotPaused
└	pause	External ❗	🛑	onlyOwner
└	unpause	External ❗	🛑	onlyOwner
└	renounceOwnership	Public ❗	🛑	onlyOwner whenNotPaused
└	transferOwnership	Public ❗	🛑	onlyOwner whenNotPaused
└	updateWhitelist	External ❗	🛑	onlyOwner
└	_addManyToWhitelist	Private 🔐	🛑	
└	_clearWhitelist	Private 🔐	🛑	
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	\<Constructor\>	Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗

CONTRACT ASSESMENT

└	allowance	Public ❗			NO❗
└	approve	Public ❗		🛑	NO❗
└	transferFrom	Public ❗		🛑	NO❗
└	increaseAllowance	Public ❗		🛑	NO❗
└	decreaseAllowance	Public ❗		🛑	NO❗
└	_transfer	Internal 🔒		🛑	
└	_mint	Internal 🔒		🛑	
└	_burn	Internal 🔒		🛑	
└	_approve	Internal 🔒		🛑	
└	_spendAllowance	Internal 🔒		🛑	
└	_beforeTokenTransfer	Internal 🔒		🛑	
└	_afterTokenTransfer	Internal 🔒		🛑	
IERC20	Interface			
└	totalSupply	External ❗			NO❗
└	balanceOf	External ❗			NO❗
└	transfer	External ❗		🛑	NO❗
└	allowance	External ❗			NO❗
└	approve	External ❗		🛑	NO❗
└	transferFrom	External ❗		🛑	NO❗
IERC20Metadata	Interface	IERC20			
└	name	External ❗			NO❗
└	symbol	External ❗			NO❗
└	decimals	External ❗			NO❗
Context	Implementation			
└	_msgSender	Internal 🔒			
└	_msgData	Internal 🔒			
Ownable	Implementation	Context			
└	<Constructor>	Public ❗		🛑	NO❗
└	owner	Public ❗			NO❗
└	_checkOwner	Internal 🔒			
└	renounceOwnership	Public ❗		🛑	onlyOwner
└	transferOwnership	Public ❗		🛑	onlyOwner
└	_transferOwnership	Internal 🔒		🛑	
ERC20Burnable	Implementation	Context, ERC20			
└	burn	Public ❗		🛑	NO❗
└	burnFrom	Public ❗		🛑	NO❗

CONTRACT ASSESMENT

ERC20Pausable	Implementation	ERC20, Pausable		
└	_beforeTokenTransfer	Internal 🔒	🛑	
Pausable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	paused	Public ❗		NO❗
└	_requireNotPaused	Internal 🔒		
└	_requirePaused	Internal 🔒		
└	_pause	Internal 🔒	🛑	whenNotPaused
└	_unpause	Internal 🔒	🛑	whenPaused

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
FullFeatureToken._clearWhitelist().i (contracts/Token.sol#1427) is a local variable never initialized
FullFeatureToken._addManyToWhitelist(address[]).i (contracts/Token.sol#1407) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

FullFeatureToken.constructor(string,string,uint256,uint8,address,FullFeatureToken.ERC20ConfigProps,uint256,string).tokenOwner (contracts/Token.sol#1024) lacks a zero-check on
:
        - initialTokenOwner = tokenOwner (contracts/Token.sol#1042)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Helpers.safeTransferETH(address,uint256) (contracts/Token.sol#898-908) uses assembly
        - INLINE ASM (contracts/Token.sol#901-904)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['^0.8.0', '^0.8.17']
        - ^0.8.0 (contracts/Token.sol#8)
        - ^0.8.0 (contracts/Token.sol#37)
        - ^0.8.0 (contracts/Token.sol#131)
        - ^0.8.0 (contracts/Token.sol#164)
        - ^0.8.0 (contracts/Token.sol#274)
        - ^0.8.0 (contracts/Token.sol#694)
        - ^0.8.0 (contracts/Token.sol#732)
        - ^0.8.0 (contracts/Token.sol#776)
        - ^0.8.17 (contracts/Token.sol#869)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Context._msgData() (contracts/Token.sol#25-27) is never used and should be removed
Helpers.safeTransferETH(address,uint256) (contracts/Token.sol#898-908) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (contracts/Token.sol#8) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#37) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#131) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#164) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#274) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#694) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#732) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#776) allows old versions
Pragma version^0.8.17 (contracts/Token.sol#869) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

FullFeatureToken.initialDocumentUri (contracts/Token.sol#919) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```
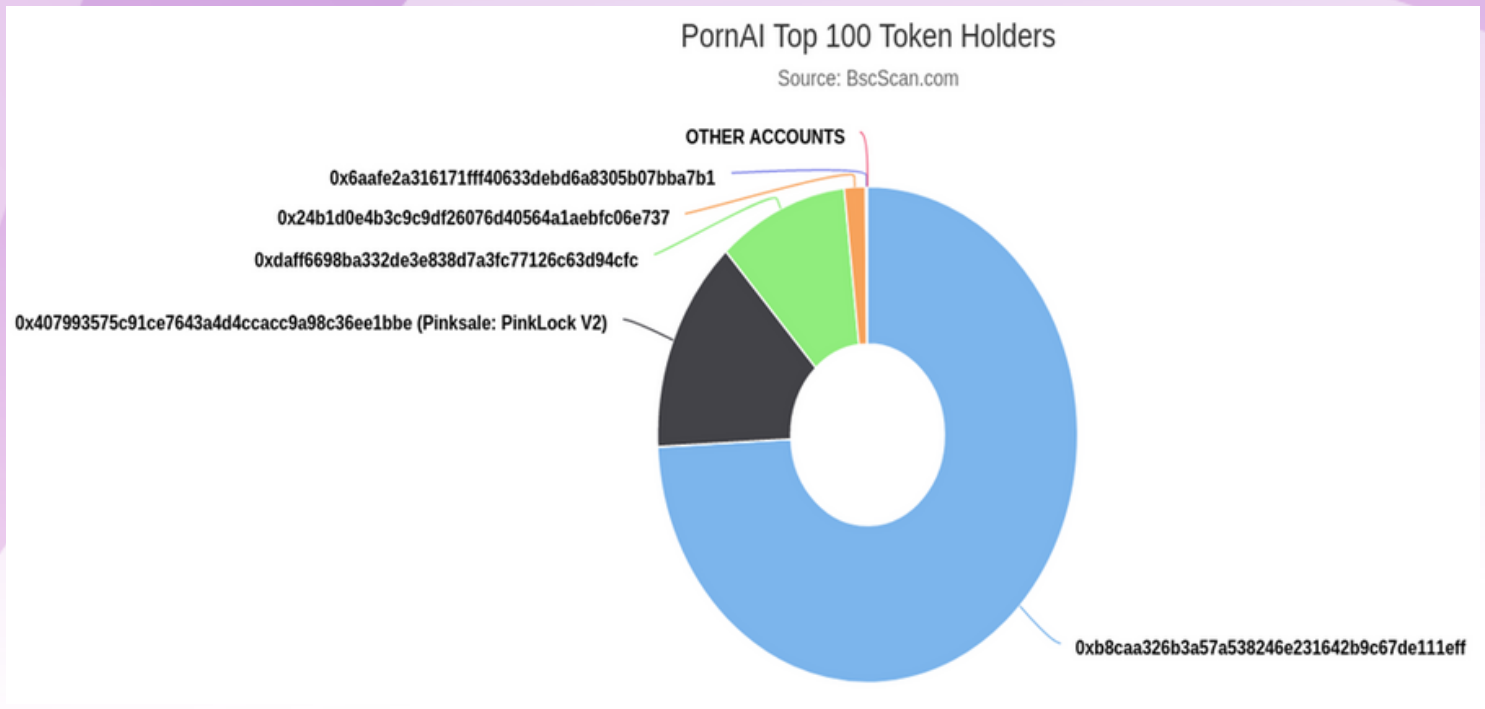
Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

TOKEN HOLDERS

Holders at time of audit

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xdea38f5674478f5a75571b76ef
302fbfdc71f470cb47bac10be31efad983c85a

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xa2435e61ad912c081c5d94cec62
890ebe74f18ba26db74e3e55102ebf3ff5339

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x2963cb39c03d00d4e6220f58c0
c05c9203120a4c4ec535b89c698b4d00acf94a

4- Transferring when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xfa73254f4a31b06682b8c89863
fc30a3471e4309e4c456dc3fd1e099fc8ffdd5

5- Buying when not excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x59f74de9a3861c2e53e3793b57
f9a033e65e1a368c003382efede9d6c9625563

6- Selling when not excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xbd34649c678b427fb7b3fc586a
0cfd583c3eb82c23f05c3effbfad096999822c

FUNCTIONAL TESTING

7- Transferring when not excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xb13f3714e68d2c096506d8ab69
ed8f8ebb24a7c1a96f0c58035fca044bb248c0

MANUAL TESTING

No Issues Found

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. 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 Auditace 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 Auditace 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 Auditace team disclaims any liability for the resulting losses.

ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.

https://auditace.tech/

https://t.me/Audit_Ace

https://twitter.com/auditace_

https://github.com/Audit-Ace