freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

HarryPotterObamaPacMan8XInu
$HPOP8XI

21/08/2023

TOKEN OVERVIEW

Fees

• Buy fees:	1%
• Sell fees:	1%

Fees privileges

- Can’t change / set fees

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can’t change max tx amount and max wallet amount

Blacklist

- Blacklist function not detected

Other privileges

- Contract owner has to call enableTrading function to enable trade
- Can burn tokens

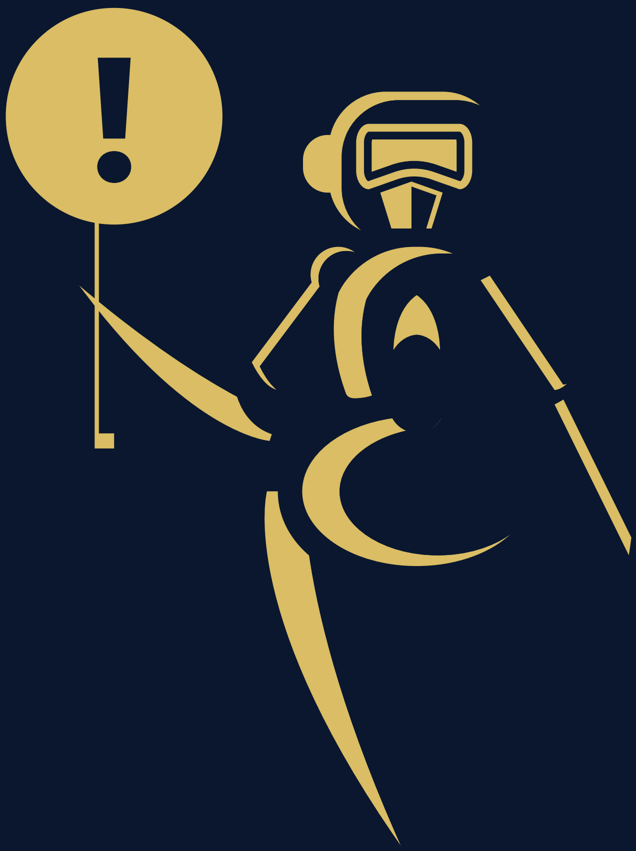
TABLE OF CONTENTS

DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website https://freshcoins.io
FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy (RUG or Honeypot etc)



INTRODUCTION

FreshCoins (Consultant) was contracted by HarryPotterObamaPacMan8XInu (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x223e6876E96E86e3487C6eE6ACe0798510f7e6Ba

Network: Ethereum (ETH)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 21/08/2023

WEBSITE DIAGNOSTIC

https://hpop8xi.com/

🟥 0-49

🟨 50-89

🟩 90-100

Performance	Accessibility	Best Practices	SEO	Progressive Web App
NA	NA	NA	NA	NA

Socials

Twitter

https://twitter.com/HPOP8XI

Telegram

https://t.me/HPOP8XI

AUDIT OVERVIEW

82

Security Score

98

Static Scan

Automatic scanning for common vulnerabilities

84

ERC Scan

Automatic checks for ERC's conformance

1	High
0	Medium
0	Low
0	Optimizations
0	Informational

No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Passed
4	Timestamp dependence	Passed
5	Integer Overflow and Underflow	Passed
6	Reverted DoS	Passed
7	DoS with block gas limit	Passed
8	Methods execution permissions	Passed
9	Exchange rate impact	Passed
10	Malicious Event	Passed
11	Scoping and Declarations	Passed
12	Uninitialized storage pointers	Passed
13	Design Logic	Passed
14	Safe Zeppelin module	Passed

OWNER PRIVILEGES

● Contract owner can't mint tokens after initial contract deploy

● Contract owner can't exclude an address from transactions

● Contract owner can whitelist presale address

```
function setPresaleAddress (address presale) external onlyOwner {
        require (presale != address(0), "zero address not allowed");
        presaleAddress = presale;
}
```

● Contract owner can burn tokens

```
function burn (uint256 amount) external {
        _burn(msg.sender, amount);
}

function _burn(address account, uint256 amount) internal virtual {
        require(account != address(0), "ERC20: burn from the zero address");

        _beforeTokenTransfer(account, address(0), amount);

        uint256 accountBalance = _balances[account];
        require(accountBalance >= amount, "ERC20: burn amount exceeds balance");
        unchecked {
            _balances[account] = accountBalance - amount;
            // Overflow not possible: amount <= accountBalance <= totalSupply.
            _totalSupply -= amount;
        }

        emit Transfer(account, address(0), amount);

        _afterTokenTransfer(account, address(0), amount);
}
```

● Contract owner has to call enableTrading function to enable trade

Note that owner can trade even if trading is disabled

```
function enableTrading() external onlyOwner{
        require(!tradingEnabled, "Trading is already enabled.");
        tradingEnabled = true;
        swapEnabled = true;
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        _transferOwnership(newOwner);
}

function _transferOwnership(address newOwner) internal virtual {
        address oldOwner = _owner;
        _owner = newOwner;
        emit OwnershipTransferred(oldOwner, newOwner);
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
        _transferOwnership(address(0));
}
```

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. The risk can be prevented by temporarily locking the contract or renouncing ownership.

CONCLUSION AND ANALYSIS

Smart Contracts within the scope were manually reviewed and analyzed with static tools.

Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.

Found 1 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	1%
Sell fees:	1%
Max TX:	N/A
Max Sell:	N/A

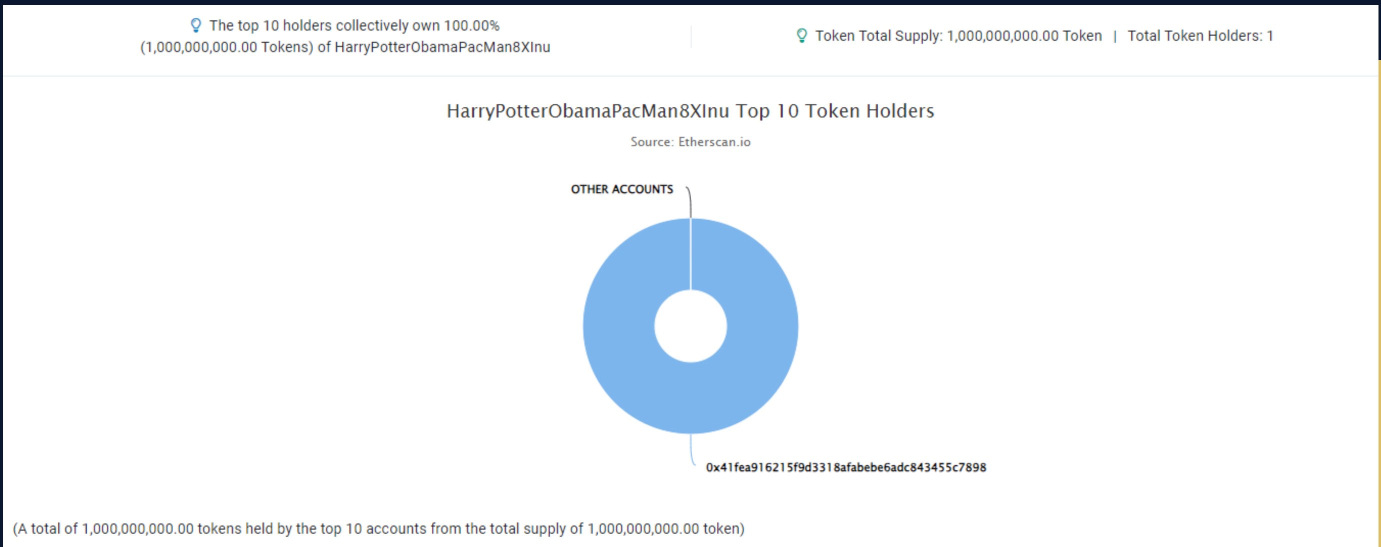
Honeypot Risk

Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	Not detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

Rug Pull Risk

Liquidity:	N/A
Holders:	100% unlocked tokens

HPOP8XI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



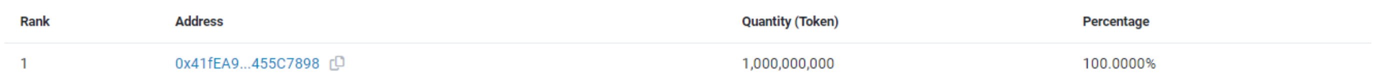
The top 10 holders collectively own 100.00% (1,000,000,000.00 Tokens) of HarryPotterObamaPacMan8XInu

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 1

HarryPotterObamaPacMan8XInu Top 10 Token Holders

Source: Etherscan.io

(A total of 1,000,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000.00 token)



Rank	Address	Quantity (Token)	Percentage
1	0x41fEA9...455C7898	1,000,000,000	100.0000%

TECHNICAL DISCLAIMER

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.