RUGFREECOINS

Drip X Token

RugfreeCoins Verified on March 07th, 2024

Overview

❌ Mint function found, the owner can mint tokens after initial deployment.

✅ The owner can't set a max transaction limit

✅ The owner can't pause trading once it's enabled

✅ The owner can't change fees.

❌ The owner can blacklist wallets.

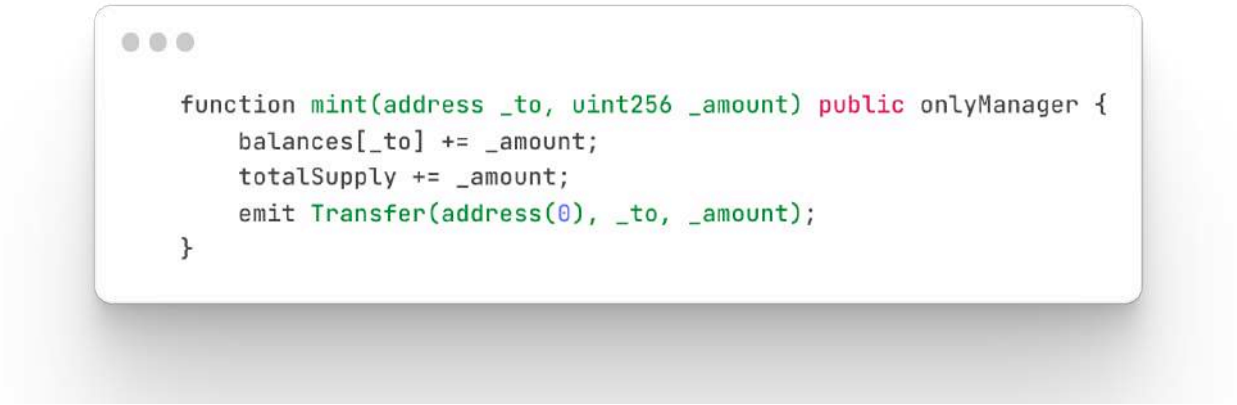
✅ The owner can't set a max wallet limit

❌ The owner can claim the contract's balance of its own token.

❗ HIGH SEVERITY ISSUES

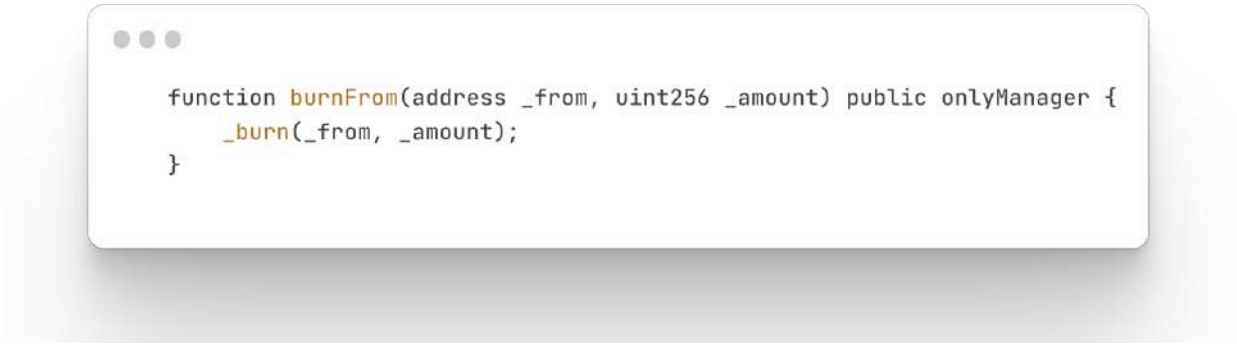
The contract is an upgradable contract, the owner can change the functions later

The owner can mint new tokens without a limit



```
function mint(address _to, uint256 _amount) public onlyManager {
    balances[_to] += _amount;
    totalSupply += _amount;
    emit Transfer(address(0), _to, _amount);
}
```

The owner can burn tokens from any wallet without their permissions



```
function burnFrom(address _from, uint256 _amount) public onlyManager {
    _burn(_from, _amount);
}
```

The owner can blacklist wallets

```
function setIsBlacklisted(address user, bool value) public onlyOwner {
    isBlacklisted[user] = value;
}
```

The owner can withdraw native tokens from the contract

```
function withdrawTokens() public onlyOwner {
    _transfer(address(this), owner(), balanceOf(address(this)));
}
```

Contents

Audit details

Audited project

DripX Token

Contract Address

0xeCbcE2c13d3A1248deBfD23DCc0558b495916756

Client contact

DripX Token Team

Blockchain

Binance Smart chain

Project website

https://www.dripx.win/

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – **please make sure to read it in full.**

Background

RugfreeCoins was commissioned by the **DripX Token Team** to perform an audit of the smart contract.

https://bscscan.com/address/0xeCbcE2c13d3A1248deBfD23DCc0558b495916756

This audit focuses on verifying that the smart contract is secure, resilient, and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, and long-term sustainability, and as a guide to improving the smart contract's security posture by remediating the identified issues.

Tokenomics

▲ 0% tax when buying & selling

Target market and the concept

- Anyone who’s interested in the Crypto space with long-term investment plans.
- Anyone who’s ready to earn a passive income by holding tokens.
- Anyone who’s interested in trading tokens.
- Anyone who’s interested in taking part in the DripX token ecosystem.
- Anyone who’s interested in taking part in the future plans of DripX Token.
- Anyone who’s interested in making financial transactions with any other party DripX Token as the currency.

Potential to grow with score points

⚡ Project efficiency	9 / 10
🌟 Project uniqueness	9 / 10
📊 Information quality	9 / 10
👌 Service quality	9 / 10
💻 System quality	9 / 10
🌍 Impact on the community	9 / 10
💼 Impact on the business	9 / 10
🔮 Preparing for the future	9 / 10
🔒 Smart contract security	7 / 10
🛠️ Smart contract functionality assessment	10 / 10
🏆 **Total Score**	**8.9** / 10

Contract details

Token contract details for 07th of March 2024

Contract name	**DripX Token**
Contract address	**0xeCbcE2c13d3A1248deBfD23DCc0558b495916756**
Token supply	**336,369,294,645.906426**
Token ticker	**DRIPX**
Decimals	**18**
Token holders	**4175**
Transaction count	**63379**
Contract deployer address	**0xe90C5C1D36aB80FfcCCca40C4989633026EF45Fa**

Contract code function details

№	Category	Item	Result
1	Coding conventions	ERC20 Token standards	PASS
		Compile errors	PASS
		Compiler version security	PASS
		Visibility specifiers	PASS
		Gas consumption	PASS
		SafeMath features	PASS
		Fallback usage	PASS
		tx.origin usage	PASS
		Deprecated items	PASS
		Redundant code	PASS
		Overriding variables	PASS
2	Function call audit	Authorization of function call	PASS
		Low level function (call/delegate call) security	PASS
		Returned value security	PASS
		Self destruct function security	PASS
3	Business security & centralisation	Access control of owners	HIGH
		Business logics	PASS
		Business implementation	PASS
4	Integer overflow/underflow		PASS
5	Reentrancy		PASS
6	Exceptional reachable state		PASS
7	Transaction ordering dependence		PASS
8	Block properties dependence		PASS
9	Pseudo random number generator (PRNG)		PASS
10	DoS (Denial of Service)		PASS
11	Token vesting implementation		PASS
12	Fake deposit		PASS
13	Event security		PASS

Contract description table

The below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions, and implementations with their visibility and mutability.

Contract	Type	Bases		
└	Function Name	Visibility	Mutability	Modifiers
Admin Upgradeability Proxy	**Implementation**	**TransparentUpgradeableProxy**		
└		Public ❗	💵	Transparent Upgradeable Proxy
TransparentUpgradeableProxy	**Implementation**	**ERC1967Proxy**		
└		Public ❗	💵	ERC1967Proxy
└	admin	External ❗	🛑	ifAdmin
└	implementation	External ❗	🛑	ifAdmin
└	changeAdmin	External ❗	🛑	ifAdmin
└	upgradeTo	External ❗	🛑	ifAdmin
└	upgradeToAndCall	External ❗	💵	ifAdmin
└	_admin	Internal 🔒		
└	_beforeFallback	Internal 🔒	🛑	
BeaconProxy	**Implementation**	**Proxy, ERC1967 Upgrade**		
└		Public ❗	💵	NO ❗
└	_beacon	Internal 🔒		
└	_implementation	Internal 🔒		
└	_setBeacon	Internal 🔒	🛑	

Upgradeable Beacon	**Implementation**	**IBeacon, Ownable**		
└		Public ❗	🛑	NO ❗
└	implementation	Public ❗		NO ❗
└	upgradeTo	Public ❗	🛑	onlyOwner
└	_setImplementation	Private 🔐	🛑	
ERC1967Proxy	**Implementation**	**Proxy, ERC1967 Upgrade**		
└		Public ❗	💵	NO ❗
└	_implementation	Internal 🔒		
ProxyAdmin	**Implementation**	**Ownable**		
└	getProxyImplementation	Public ❗		NO ❗
└	getProxyAdmin	Public ❗		NO ❗
└	changeProxyAdmin	Public ❗	🛑	onlyOwner
└	upgrade	Public ❗	🛑	onlyOwner
└	upgradeAndCall	Public ❗	💵	onlyOwner
IBeacon	**Interface**			
└	implementation	External ❗		NO ❗
Proxy	**Implementation**			
└	_delegate	Internal 🔒	🛑	
└	_implementation	Internal 🔒		
└	_fallback	Internal 🔒	🛑	
└		External ❗	💵	NO ❗
└		External ❗	💵	NO ❗
└	_beforeFallback	Internal 🔒	🛑	

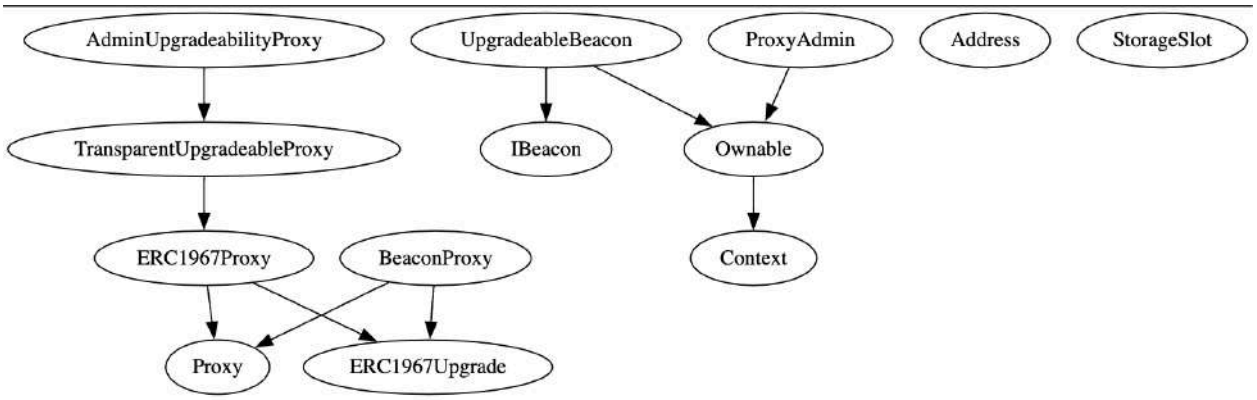
ERC1967 Upgrade	Implementation			
└	_getImplementation	Internal 🔒		
└	_setImplementation	Private 🔐	🛑	
└	_upgradeTo	Internal 🔒	🛑	
└	_upgradeToAndCall	Internal 🔒	🛑	
└	_upgradeToAndCallSecure	Internal 🔒	🛑	
└	_upgradeBeaconToAndCall	Internal 🔒	🛑	
└	_getAdmin	Internal 🔒		
└	_setAdmin	Private 🔐	🛑	
└	_changeAdmin	Internal 🔒	🛑	
└	_getBeacon	Internal 🔒		
└	_setBeacon	Private 🔐	🛑	
Address	**Library**			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	_verifyCallResult	Private 🔐		
StorageSlot	**Library**			
└	getAddressSlot	Internal 🔒		
└	getBooleanSlot	Internal 🔒		

└	getBytes32Slot	Internal 🔒		
└	getUint256Slot	Internal 🔒		
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Inheritance Hierarchy



Security issue checking status

❖ High severity issues

The contract is an upgradable contract, the owner can change the functions later

The owner can mint new tokens without a limit

```
function mint(address _to, uint256 _amount) public onlyManager {
    balances[_to] += _amount;
    totalSupply += _amount;
    emit Transfer(address(0), _to, _amount);
}
```

The owner can burn tokens from any wallet without their permissions

```
function burnFrom(address _from, uint256 _amount) public onlyManager {
    _burn(_from, _amount);
}
```

The owner can blacklist wallets

```
function setIsBlacklisted(address user, bool value) public onlyOwner {
    isBlacklisted[user] = value;
}
```

The owner can withdraw native tokens from the contract

```
function withdrawTokens() public onlyOwner {
    _transfer(address(this), owner(), balanceOf(address(this)));
}
```

- Medium severity issues

  No medium severity issues found

- Low severity issues

  No high severity issues found

Owner privileges

- Owners can mint new tokens without a maximum limit

```
function mint(address _to, uint256 _amount) public onlyManager {
    balances[_to] += _amount;
    totalSupply += _amount;
    emit Transfer(address(0), _to, _amount);
}
```

- Owners can burn tokens from any wallet

```
function burnFrom(address _from, uint256 _amount) public onlyManager {
    _burn(_from, _amount);
}
```

- Owner can blacklist/whitelist wallets

```
function setIsBlacklisted(address user, bool value) public onlyOwner {
    isBlacklisted[user] = value;
}
```

❖ Owner can withdraw native tokens from the contract

```
function withdrawTokens() public onlyOwner {
    _transfer(address(this), owner(), balanceOf(address(this)));
}
```

Audit conclusion

RugFreeCoins team has performed in-depth testing, line-by-line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status:	PASS
Smart contract security Status:	HIGH ISSUES
Number of risk issues:	04
Solidity code functional issue level:	PASS
Number of owner privileges:	04
Centralization risk correlated to the active owner:	HIGH
Smart contract active ownership:	ACTIVE