Cyberscope

Audit Report

Banana

November 2022

Type	BEP20
Network	BSC
Address	0x7604e590299221f34f7A79b90289f084E77cAa2e
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Banana
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x7604e590299221f34f7A79b90289f084E77cAa2e
Symbol	BANANA
Decimals	18

Audit Updates

Initial Audit	24th November 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/ERC20.sol	bce14c3fd3b1a668529e375f6b70ffdf9cef8c4e410ae99608be5964d98fa701
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
contracts/apeparadise/Banana.sol	54d57016bd4d79e3cc95616322dee2591856ae8ee8885549c72e32dfa9789e26

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Unresolved
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Unresolved
●	BT	Burns Tokens	Unresolved
●	BC	Blacklists Addresses	Passed

OTUT - Transfers User's Tokens

Criticality	critical
Location	contract.sol#L90,96
Status	Unresolved

Description

The “caveAddress” address has the authority to transfer the balance of a user’s contract to the “caveAddress” address. The “caveAddress” address may take advantage of it by calling the transferToCave function.

```
function transferToCave(address _from, uint256 _amount) external {
    require(caveAddress != address(0), "missing initial requirements");
    require(_msgSender() == caveAddress, "only the cave contract can call
transferToCave");
    _transfer(_from, caveAddress, _amount);
}
```

The “upgradeAddress” address has the authority to transfer the balance of a user’s contract to the “upgradeAddress” address. The “upgradeAddress” address may take advantage of it by calling the transferForUpgradesFees function.

```
function transferForUpgradesFees(address _from, uint256 _amount) external {
    require(upgradeAddress != address(0), "missing initial requirements");
    require(_msgSender() == upgradeAddress, "only the upgrade contract can call
transferForUpgradesFees");
    _transfer(_from, upgradeAddress, _amount);
}
```

Recommendation

The team should carefully manage the private keys. 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.

MT - Mints Tokens

Criticality	critical
Location	contract.sol#L73
Status	Unresolved

Description

The “forestAddress” address has the authority to mint tokens. The “forestAddress” address may take advantage of it by calling the mint function. As a result the contract tokens will be highly inflated.

```
function mint(address _to, uint256 _amount) external {
    require(forestAddress != address(0) && apeAddress != address(0) &&
caveAddress != address(0) && upgradeAddress != address(0), "missing initial
requirements");
    require(_msgSender() == forestAddress,"msgsender does not have
permission");
    _mint(_to, _amount);
}
```

The owner has the authority to mint tokens with three additional ways

```
function mintPromotionalBanana(address _to) external onlyOwner {}
function mintBnbLPBanana() external onlyOwner {}
function mintTreeLPBanana() external onlyOwner {}
```

Recommendation

The “forestAddress” address and owner should carefully manage the credentials. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

BT - Burns Tokens

Criticality	critical
Location	contract.sol#L79
Status	Unresolved

Description

The “apeAddress”, “caveAddress” and “upgradeAddress” has the authority to burn tokens from a specific address. They may take advantage of it by calling the burn function. As a result the targeted contract address will lose the corresponding tokens.

```
function burn(address _from, uint256 _amount) external {
    require(apeAddress != address(0) && caveAddress != address(0) &&
upgradeAddress != address(0), "missing initial requirements");
    require(
        _msgSender() == apeAddress
        || _msgSender() == caveAddress
        || _msgSender() == upgradeAddress,
        "msgsender does not have permission"
    );
    _burn(_from, _amount);
}
```

Recommendation

The “apeAddress”, “caveAddress” and “upgradeAddress” addresses should carefully manage the credentials of the owner’s account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Contract Diagnostics

🔴 Critical 🟠 Medium ⚪ Minor / Informative

Severity	Code	Description	Status
⚪	L04	Conformance to Solidity Naming Conventions	Unresolved

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/apeparadise/Banana.sol#L90,79,36,73,96,32,12,49,44,28,67
Status	Unresolved

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.

```
_amount
_from
_upgradeAddress
_to
_caveAddress
NUM_BANANA_BNB_LP
_apeAddress
_forestAddress
_numBananaBnbLp
...
```

Recommendation

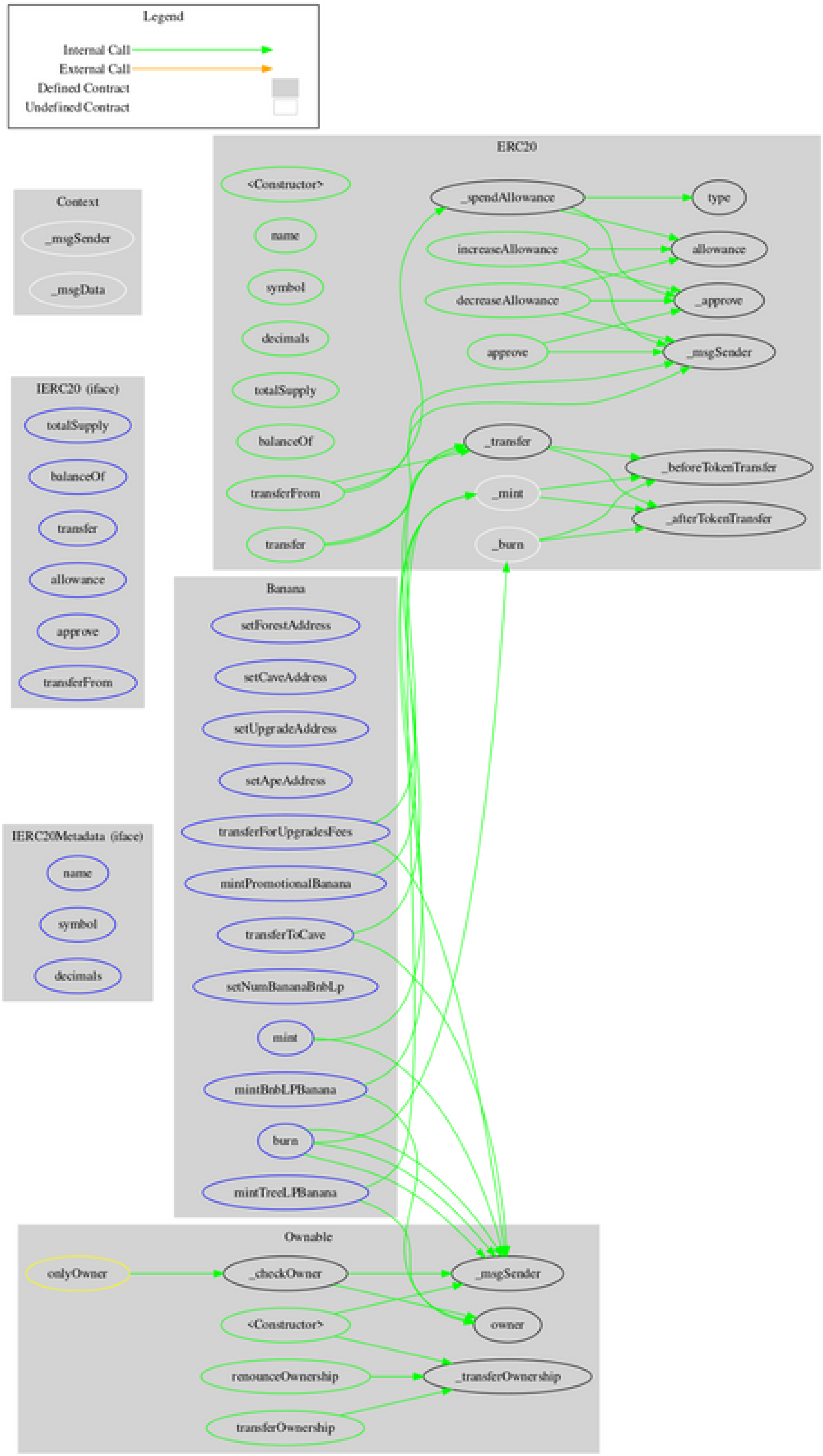
Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-conventions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

	_afterTokenTransfer	Internal	✓	
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Banana	Implementation	ERC20, Ownable		
	setForestAddress	External	✓	onlyOwner
	setCaveAddress	External	✓	onlyOwner
	setUpgradeAddress	External	✓	onlyOwner
	setApeAddress	External	✓	onlyOwner
	mintPromotionalBanana	External	✓	onlyOwner
	mintBnbLPBanana	External	✓	onlyOwner
	mintTreeLPBanana	External	✓	onlyOwner
	setNumBananaBnbLp	External	✓	onlyOwner
	mint	External	✓	-
	burn	External	✓	-
	transferToCave	External	✓	-
	transferForUpgradesFees	External	✓	-

Contract Flow



Summary

There are some functions that can be abused by the owner like transferring the user's tokens, minting tokens and burning tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. if the contract owner abuses the burn functionality, then the users could lost their tokens. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io