Cyberscope

Audit Report

BOBO coin

February 2025

Network BSC

Address 0x0F433fEbc93003b1e07826F9ee18479e905BC5A6

Audited by © cyberscope

Analysis

🔴 Critical 🟡 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🔴	ST	Stops Transactions	Unresolved
🔵	OTUT	Transfers User's Tokens	Passed
🔵	ELFM	Exceeds Fees Limit	Passed
🔵	MT	Mints Tokens	Passed
🔵	BT	Burns Tokens	Passed
🔵	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	UAR	Unexcluded Address Restrictions	Unresolved
●	UM	Unused Methods	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L19	Stable Compiler Version	Unresolved

Table of Contents

Risk Classification

The criticality of findings in Cyberscope’s smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
● Critical	Highly Likely / High Impact
● Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
● Minor / Informative	Unlikely / Low to no Impact

Review

Contract Name	SaleCompatibleToken
Compiler Version	v0.8.26+commit.8a97fa7a
Optimization	200 runs
Explorer	https://bscscan.com/address/0x0f433febc93003b1e07826f9ee18479e905bc5a6
Address	0x0f433febc93003b1e07826f9ee18479e905bc5a6
Network	BSC
Symbol	BOBO
Decimals	18
Total Supply	420.690.000.000.000

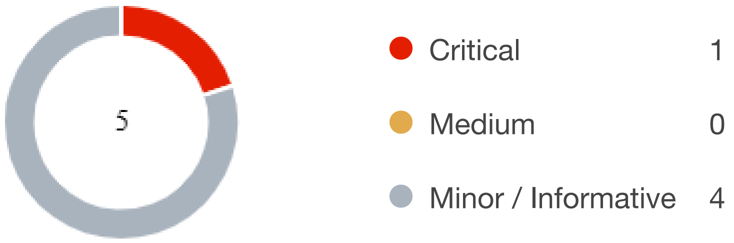
Audit Updates

Initial Audit	06 Feb 2025

Source Files

Filename	SHA256
contracts/token_nomint.sol	272e4f17e61905f83c91d5b374f02c9eef40cb968b146a3115d6d0da461ef689

Findings Breakdown



Severity	Unresolved	Acknowledged	Resolved	Other
● Critical	1	0	0	0
● Medium	0	0	0	0
● Minor / Informative	4	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	contracts/token_nomint.sol#L70
Status	Unresolved

Description

The contract owner has the authority to set the `maxTransferAmount` to a very small number (e.g. 1) for all users excluding the owner. As a result, the users will only be able to sell an extremely small amount of tokens not worth the gas they will spend for each transaction and the contract may operate as a honeypot.

```
bool isSaleTransfer = (from == saleContract || to ==
saleContract);

// Skip trading restrictions for sale contract transfers
if (!isSaleTransfer && from != owner() && to != owner()) {
    require(tradingEnabled, "Trading not enabled");
    require(
        amount <= maxTransferAmount,
        "Transfer amount exceeds limit"
    );
}
```

Recommendation

The contract could embody a check for not allowing setting the `maxTransferAmount` less than a reasonable amount. A suggested implementation could check that the minimum amount should be more than a fixed percentage of the total supply. 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.

Temporary Solutions: These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

UAR - Unexcluded Address Restrictions

Criticality	Minor / Informative
Location	contracts/token_nomint.sol#L19
Status	Unresolved

Description

The contract incorporates operational restrictions on transactions, which can hinder seamless interaction with decentralized applications (dApps) such as launchpads, presales, lockers, or staking platforms. In scenarios where an external contract, such as a launchpad factory, needs to integrate with the contract, it should be exempt from the limitations to ensure uninterrupted service and functionality. Failure to provide such exemptions can block the successful process and operation of services reliant on this contract.

In the contract's current configuration there is only a single address `saleContract` that can be excluded from such restrictions. This may cause issues since there is a possibility that multiple addresses need to be excluded.

```
address public saleContract;
```

Recommendation

It is advisable to modify the contract by incorporating functionality that enables the exclusion of designated addresses from transactional restrictions. This enhancement will allow specific addresses, such as those associated with decentralized applications (dApps) and service platforms, to operate without being hindered by the standard constraints imposed on other users. Implementing this feature will ensure smoother integration and functionality with external systems, thereby expanding the contract's versatility and effectiveness in diverse operational environments.

UM - Unused Methods

Criticality	Minor / Informative
Location	contracts/token_nomint.sol#L80
Status	Unresolved

Description

The contract includes a function named `isSaleContract` that checks if a given address matches the `saleContract` address and returns a boolean value. However, `saleContract` is a public variable meaning it can just be retrieved directly. Furthermore, `isSaleContract` is not utilized in the rest of the contract's functionality.

```
function isSaleContract(address _address) public view returns
(bool) {
    return _address == saleContract;
}
```

Recommendation

It is recommended to remove unused functionalities as they reduce code readability and increase deployment cost.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contracts/token_nomint.sol#L38,80
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
address _saleContract
address _address
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/stable/style-guide.html#naming-conventions.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/token_nomint.sol#L2
Status	Unresolved

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```
pragma solidity ^0.8.20;
```

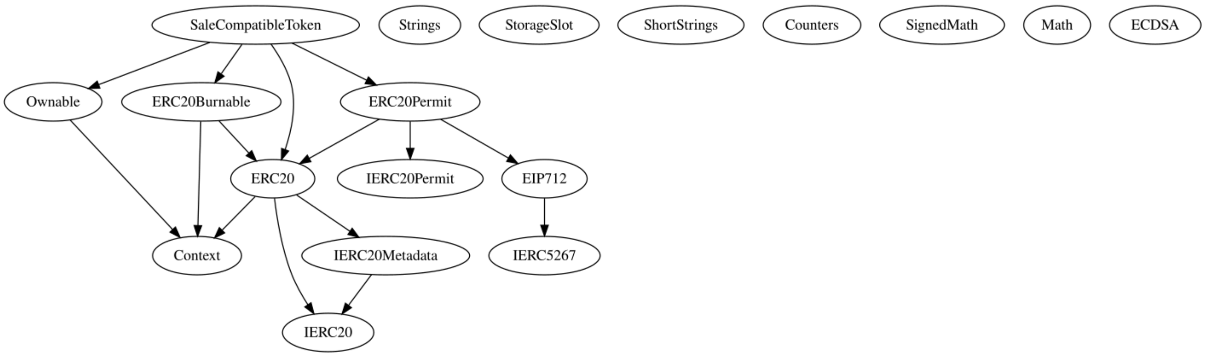
Recommendation

The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

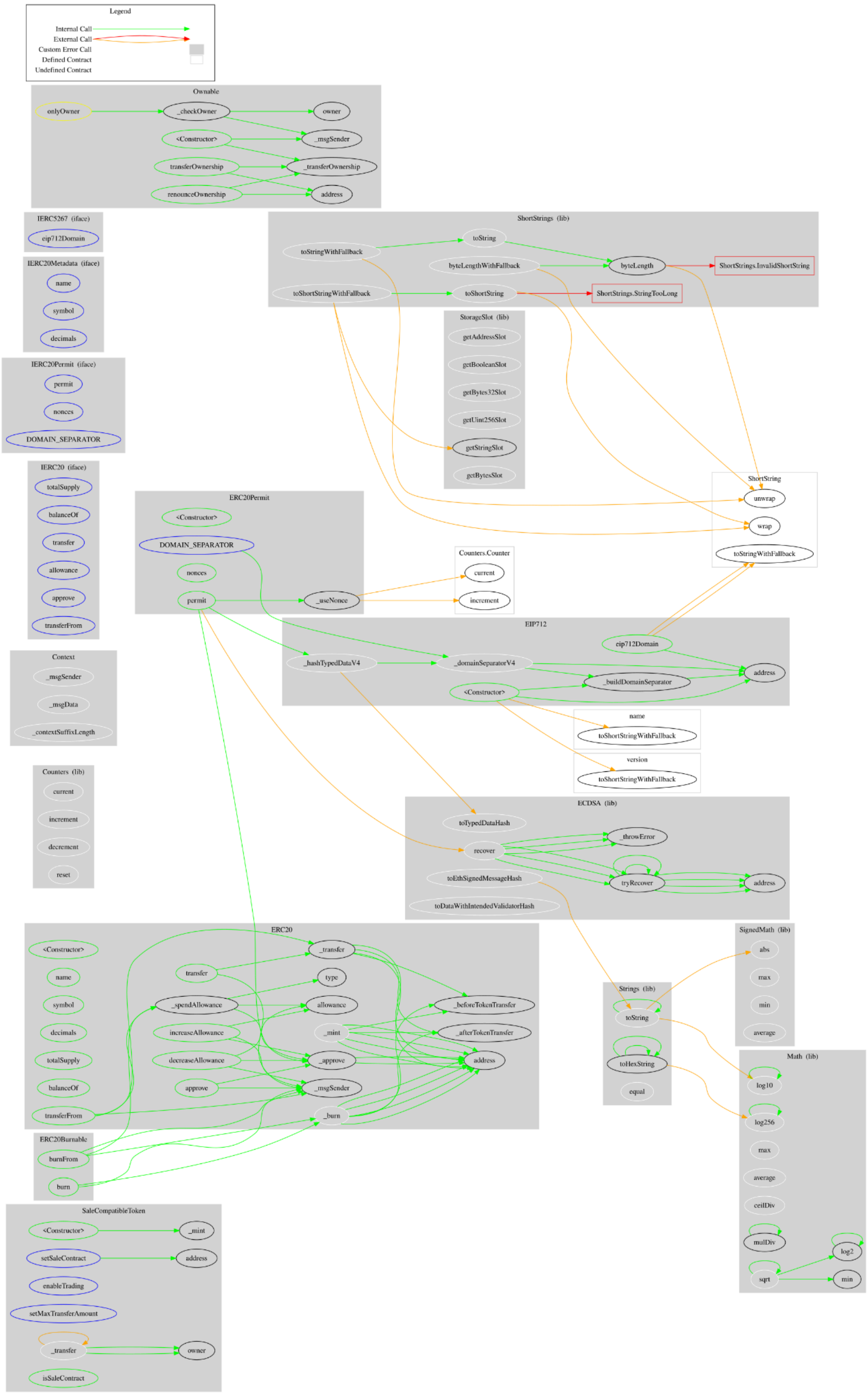
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SaleCompatibleToken	Implementation	ERC20, ERC20Burnable, ERC20Permit, Ownable		
		Public	✓	ERC20 ERC20Permit Ownable
	setSaleContract	External	✓	onlyOwner
	enableTrading	External	✓	onlyOwner
	setMaxTransferAmount	External	✓	onlyOwner
	_transfer	Internal	✓	
	isSaleContract	Public		-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
_checkOwner
owner
<Constructor>
_msgSender
transferOwnership
_transferOwnership
renounceOwnership
address
IERC5267 (iface)
eip712Domain
IERC20Metadata (iface)
name
symbol
decimals
IERC20Permit (iface)
permit
nonces
DOMAIN_SEPARATOR
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData
_contextSuffixLength
Counters (lib)
current
increment
decrement
reset
ShortStrings (lib)
toStringWithFallback
toString
byteLengthWithFallback
byteLength
ShortStrings.InvalidShortString
toShortStringWithFallback
toShortString
ShortStrings.StringTooLong
StorageSlot (lib)
getAddressSlot
getBooleanSlot
getBytes32Slot
getUint256Slot
getStringSlot
getBytesSlot
ShortString
unwrap
wrap
toStringWithFallback
ERC20Permit
<Constructor>
DOMAIN_SEPARATOR
nonces
permit
_useNonce
Counters.Counter
current
increment
EIP712
eip712Domain
_hashTypedDataV4
_domainSeparatorV4
address
_buildDomainSeparator
<Constructor>
name
toShortStringWithFallback
version
toShortStringWithFallback
ECDSA (lib)
toTypedDataHash
_throwError
recover
tryRecover
toEthSignedMessageHash
address
toDataWithIntendedValidatorHash
ERC20
<Constructor>
_transfer
name
transfer
type
symbol
_spendAllowance
allowance
_beforeTokenTransfer
decimals
increaseAllowance
_mint
_afterTokenTransfer
totalSupply
decreaseAllowance
_approve
address
balanceOf
approve
_msgSender
transferFrom
_burn
ERC20Burnable
burnFrom
burn
SaleCompatibleToken
<Constructor>
_mint
setSaleContract
address
enableTrading
setMaxTransferAmount
_transfer
owner
isSaleContract
Strings (lib)
toString
toHexString
equal
SignedMath (lib)
abs
max
min
average
Math (lib)
log10
log256
max
average
ceilDiv
mulDiv
log2
sqrt
min

Summary

BOBO coin contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. There are some functions that can be abused by the owner like stop transactions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. The contract does not implement any fees.

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

cyberscope.io