Cyberscope

Audit Report

eventflo

December 2024

Repository	https://github.com/eventfloHQ/FloCoin
Commit	2b41ddfdaa6ab2461c0436d1450a98e39cc8aafc
Audited by	© cyberscope

Analysis

🔴 Critical 🟠 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🔵	ST	Stops Transactions	Passed
🔵	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
●	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	FloCoin
Repository	https://github.com/eventfloHQ/FloCoin
Commit	2b41ddfdaa6ab2461c0436d1450a98e39cc8aafc
Testing Deploy Implementation	https://testnet.bscscan.com/address/0xb5a035a6630b55098140531c8305f97fdd377fd9
Testing Deploy Proxy	https://testnet.bscscan.com/address/0x7193853eE332e7a26D4c919F62c54AC2638E7658
Decimals	18
Badge Eligibility	Yes

Audit Updates

Initial Audit	22 Dec 2024

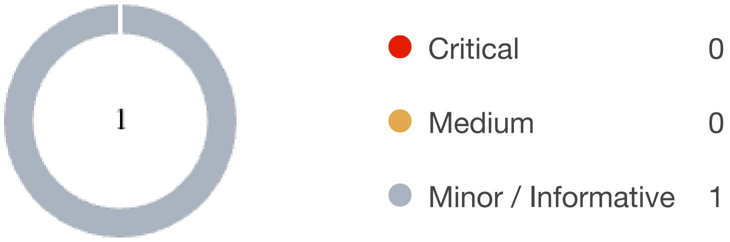
Source Files

Filename	SHA256
contracts/FloCoin.sol	f966c544139eefa9695a6d841fe399f357d053cebc7ae63e92d8670f9f8651e2
@openzeppelin/contracts-upgradeable/utils/NoncesUpgradeable.sol	d63abfbf20ca119bf162ec5bb343df4b189208d1d0a73657555f688536498cd8
@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	a08e16324da33a9d666dc07a22ae58031c242a3869f6808e55b4b82fc70cb209
@openzeppelin/contracts-upgradeable/utils/cryptography/EIP712Upgradeable.sol	e4921efd4791f39deaa0cd71fee74de4f2c8320a99f388b4189df504157981c8

@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol	aaabbd0bac5de418bfd3c3c6429b2e9dbed8fd61fdb1ac1c4ef4434f803eda88
@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20PermitUpgradeable.sol	c538cb63958c81d3d796fae44104d42f2d0fa877a15ae8f5477593869c6e2a30
@openzeppelin/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol	3c76952e97e1cacfec407359b9fc7ae41a739fab804458ba7c0806206949570f
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	a8b7eafa0fdc7cb5a644c8c61a8e4c51e031d5e1e6f268f72dbe18b768ead56e
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	9247b9ad7939d23990dbdc9274917c3762ffb37e5137ef7bbfcc2e2fba1b8dd2
@openzeppelin/contracts/utils/Strings.sol	27ab8a578913671cdda422a97e96c2239157be7a7c2cfb6d042ddd6963f200f9
@openzeppelin/contracts/utils/StorageSlot.sol	75704538dcb223239280c6726d9a31cf769a7816718517c997fc7d63bdb70778
@openzeppelin/contracts/utils/Panic.sol	270fc8401c1a13fae6a7a4a2dd6e381b95d658896701e51f0d3e2688acab3dec
@openzeppelin/contracts/utils/Errors.sol	0704b9d6c032cca8512a3bc3f30f49f86f1f03102d2896a3d23e794b82efea66
@openzeppelin/contracts/utils/Address.sol	8228692ef1ccb4cfe5b7cc58324cadd06045e10e386185f9e4d45173d6d6c633
@openzeppelin/contracts/utils/math/SignedMath.sol	1ed50b1056af886752f0fb48a0165d381e69bb4a4b18b893b066dc144a7e08d7
@openzeppelin/contracts/utils/math/SafeCast.sol	9769274bf53f26a7c7896c526ea1980dc9bea5bf5c2a5fd04870008c4afc1de9
@openzeppelin/contracts/utils/math/Math.sol	68cf79a637995d5ed243c4a5856b42a5e134ee8786a05034e24d75927fc40ebc
@openzeppelin/contracts/utils/cryptography/MessageHashUtils.sol	93e4c09f9c65d37a14d796601b67a67fc918e16957a6328261f8635e136adf76

@openzeppelin/contracts/utils/cryptography/ECDSA.sol	0964ddd02f4a7a8cf9ba130e3aeead588ca3d425d5bc13cc4221358c69108ed0
@openzeppelin/contracts/token/ERC20/IERC20.sol	30edf7394bab78d48b7db3a059248e1ea7c2c77d2ec0e37a13bb91415aafbe5a
@openzeppelin/contracts/token/ERC20/extensions/IERC20Permit.sol	026aca1c8ee4574eb9719dca7dfc33e3e57a618715ae702a675e8a8c9ea1e82d
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	9e7c70ec72d2f7d592e23ea84f3852b04f91f6f644ce57e0263493046b36afb9
@openzeppelin/contracts/proxy/beacon/IBeacon.sol	422eabc0e645e24c3a52898f6255b349323b013544a3ebdc4b2d3f7fc5bb7e9e
@openzeppelin/contracts/proxy/ERC1967/ERC1967Utils.sol	2ba1cb9e2c1a0518ac940fa1f10e62fa56f1ba2b13973f36a8b505de627e4119
@openzeppelin/contracts/interfaces/draft-IERC6093.sol	56380323009ef4a119d44550b910fde1bff9cedde8f7f4c690152c7629bc3338
@openzeppelin/contracts/interfaces/draft-IERC1822.sol	71190a8ee26dab908d3dad703ccfff09ebfc5850f2f405463e21df21a8643bbc
@openzeppelin/contracts/interfaces/IERC5267.sol	efd1ebd1e04b6ef9c3b8781a097588f83da954323f438d54a71dc06508e6c7b8
@openzeppelin/contracts/interfaces/IERC1967.sol	886b093d8f7c41f73af42b8e183314b3654531a9d5e11f07c41a5a7f11d3e006

Findings Breakdown



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

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/FloCoin.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 ^0.8.13;
```

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**
FloCoin	Implementation	ERC20Upgradeable, ERC20PermitUpgradeable, UUPSUpgradeable, OwnableUpgradeable		
		Public	✓	-
	initialize	Public	✓	initializer
	_authorizeUpgrade	Internal	✓	onlyOwner
NoncesUpgradeable	Implementation	Initializable		
	_getNoncesStorage	Private		
	__Nonces_init	Internal	✓	onlyInitializing
	__Nonces_init_unchained	Internal	✓	onlyInitializing
	nonces	Public		-
	_useNonce	Internal	✓	
	_useCheckedNonce	Internal	✓	
ContextUpgradeable	Implementation	Initializable		
	__Context_init	Internal	✓	onlyInitializing
	__Context_init_unchained	Internal	✓	onlyInitializing
	_msgSender	Internal		

	_msgData	Internal		
	_contextSuffixLength	Internal		
EIP712Upgradeable	Implementation	Initializable, IERC5267		
	_getEIP712Storage	Private		
	__EIP712_init	Internal	✓	onlyInitializing
	__EIP712_init_unchained	Internal	✓	onlyInitializing
	_domainSeparatorV4	Internal		
	_buildDomainSeparator	Private		
	_hashTypedDataV4	Internal		
	eip712Domain	Public		-
	_EIP712Name	Internal		
	_EIP712Version	Internal		
	_EIP712NameHash	Internal		
	_EIP712VersionHash	Internal		
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20, IERC20Metadata, IERC20Errors		
	_getERC20Storage	Private		
	__ERC20_init	Internal	✓	onlyInitializing
	__ERC20_init_unchained	Internal	✓	onlyInitializing
	name	Public		-
	symbol	Public		-

	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	_transfer	Internal	✓	
	_update	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
ERC20PermitUpgradeable	Implementation	Initializable, ERC20Upgradeable, IERC20Permit, EIP712Upgradeable, NoncesUpgradeable		
	__ERC20Permit_init	Internal	✓	onlyInitializing
	__ERC20Permit_init_unchained	Internal	✓	onlyInitializing
	permit	Public	✓	-
	nonces	Public		-
	DOMAIN_SEPARATOR	External		-

UUPSUpgradeable	Implementation	Initializable, IERC1822Proxiable		
	__UUPSUpgradeable_init	Internal	✓	onlyInitializing
	__UUPSUpgradeable_init_unchained	Internal	✓	onlyInitializing
	proxiableUUID	External		notDelegated
	upgradeToAndCall	Public	Payable	onlyProxy
	_checkProxy	Internal		
	_checkNotDelegated	Internal		
	_authorizeUpgrade	Internal	✓	
	_upgradeToAndCallUUPS	Private	✓	
Initializable	Implementation			
	_checkInitializing	Internal		
	_disableInitializers	Internal	✓	
	_getInitializedVersion	Internal		
	_isInitializing	Internal		
	_getInitializableStorage	Private		
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
	_getOwnableStorage	Private		
	__Ownable_init	Internal	✓	onlyInitializing
	__Ownable_init_unchained	Internal	✓	onlyInitializing
	owner	Public		-
	_checkOwner	Internal		

	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Strings	Library			
	toString	Internal		
	toStringSigned	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toChecksumHexString	Internal		
	equal	Internal		
StorageSlot	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
	getInt256Slot	Internal		
	getStringSlot	Internal		
	getStringSlot	Internal		
	getBytesSlot	Internal		
	getBytesSlot	Internal		
Panic	Library			

	panic	Internal		
Errors	Library			
Address	Library			
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	verifyCallResultFromTarget	Internal		
	verifyCallResult	Internal		
	_revert	Private		
SignedMath	Library			
	ternary	Internal		
	max	Internal		
	min	Internal		
	average	Internal		
	abs	Internal		
SafeCast	Library			
	toUint248	Internal		
	toUint240	Internal		
	toUint232	Internal		

	toUint224	Internal		
	toUint216	Internal		
	toUint208	Internal		
	toUint200	Internal		
	toUint192	Internal		
	toUint184	Internal		
	toUint176	Internal		
	toUint168	Internal		
	toUint160	Internal		
	toUint152	Internal		
	toUint144	Internal		
	toUint136	Internal		
	toUint128	Internal		
	toUint120	Internal		
	toUint112	Internal		
	toUint104	Internal		
	toUint96	Internal		
	toUint88	Internal		
	toUint80	Internal		
	toUint72	Internal		
	toUint64	Internal		
	toUint56	Internal		
	toUint48	Internal		
	toUint40	Internal		
	toUint32	Internal		

	toUint24	Internal		
	toUint16	Internal		
	toUint8	Internal		
	toUint256	Internal		
	toInt248	Internal		
	toInt240	Internal		
	toInt232	Internal		
	toInt224	Internal		
	toInt216	Internal		
	toInt208	Internal		
	toInt200	Internal		
	toInt192	Internal		
	toInt184	Internal		
	toInt176	Internal		
	toInt168	Internal		
	toInt160	Internal		
	toInt152	Internal		
	toInt144	Internal		
	toInt136	Internal		
	toInt128	Internal		
	toInt120	Internal		
	toInt112	Internal		
	toInt104	Internal		
	toInt96	Internal		
	toInt88	Internal		

	toInt80	Internal		
	toInt72	Internal		
	toInt64	Internal		
	toInt56	Internal		
	toInt48	Internal		
	toInt40	Internal		
	toInt32	Internal		
	toInt24	Internal		
	toInt16	Internal		
	toInt8	Internal		
	toInt256	Internal		
	toUint	Internal		
Math	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	ternary	Internal		
	max	Internal		
	min	Internal		
	average	Internal		
	ceilDiv	Internal		
	mulDiv	Internal		

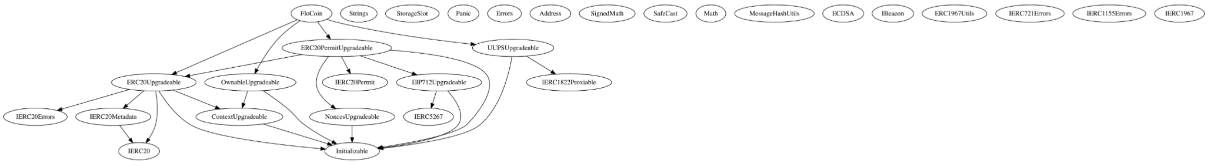
	mulDiv	Internal		
	invMod	Internal		
	invModPrime	Internal		
	modExp	Internal		
	tryModExp	Internal		
	modExp	Internal		
	tryModExp	Internal		
	_zeroBytes	Private		
	sqrt	Internal		
	sqrt	Internal		
	log2	Internal		
	log2	Internal		
	log10	Internal		
	log10	Internal		
	log256	Internal		
	log256	Internal		
	unsignedRoundsUp	Internal		
MessageHashUtils	Library			
	toEthSignedMessageHash	Internal		
	toEthSignedMessageHash	Internal		
	toDataWithIntendedValidatorHash	Internal		
	toTypedDataHash	Internal		

ECDSA	Library			
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	_throwError	Private		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
IERC20Metadata	Interface	IERC20		
	name	External		-

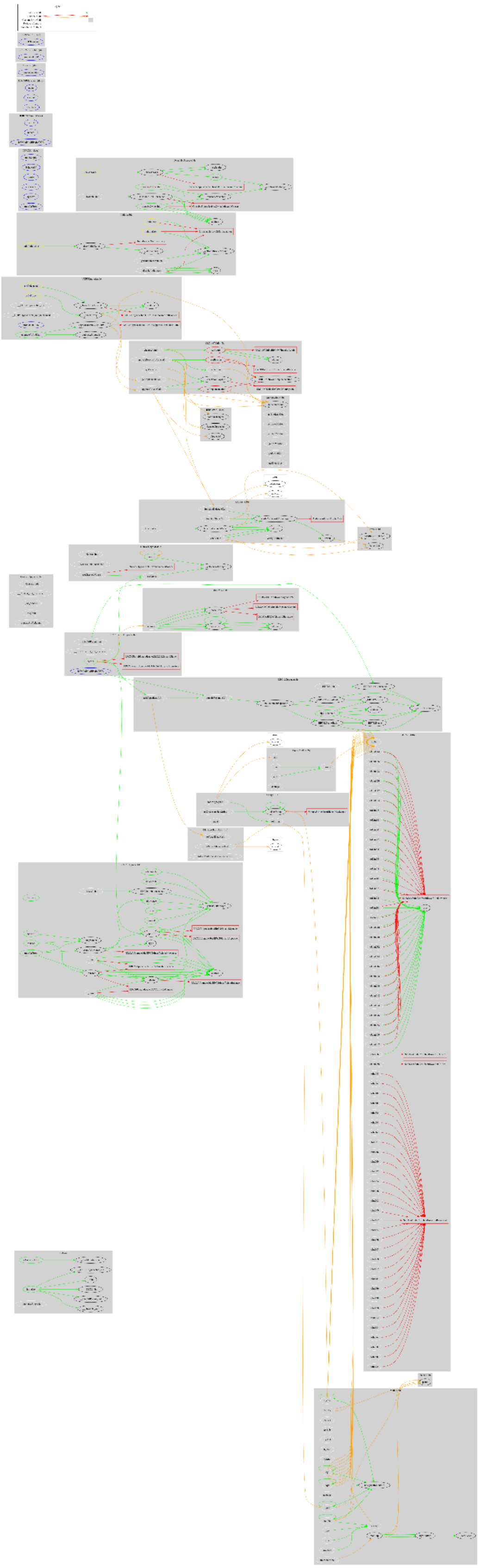
	symbol	External		-
	decimals	External		-
IBeacon	Interface			
	implementation	External		-
ERC1967Utils	Library			
	getImplementation	Internal		
	_setImplementation	Private	✓	
	upgradeToAndCall	Internal	✓	
	getAdmin	Internal		
	_setAdmin	Private	✓	
	changeAdmin	Internal	✓	
	getBeacon	Internal		
	_setBeacon	Private	✓	
	upgradeBeaconToAndCall	Internal	✓	
	_checkNonPayable	Private	✓	
IERC20Errors	Interface			
IERC721Errors	Interface			
IERC1155Errors	Interface			

IERC1822Proxiable	Interface			
	proxiableUUID	External		-
IERC5267	Interface			
	eip712Domain	External		-
IERC1967	Interface			

Inheritance Graph



Flow Graph



Summary

eventflo contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. eventflo is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

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