Cyberscope

Audit Report
Galaxy Fox

January 2024

Network	ETH
Address	0x8F1CecE048Cade6b8a05dFA2f90EE4025F4F2662
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
●	PVO	Potential Volume Overflow	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Acknowledged
●	L09	Dead Code Elimination	Acknowledged
●	L17	Usage of Solidity Assembly	Acknowledged

Table of Contents

Review

Contract Name	GalaxyFox
Compiler Version	v0.8.23+commit.f704f362
Optimization	200 runs
Explorer	https://etherscan.io/address/0x8f1cece048cade6b8a05dfa2f90ee4025f4f2662
Address	0x8f1cece048cade6b8a05dfa2f90ee4025f4f2662
Network	ETH
Symbol	GFOX
Decimals	18
Total Supply	5,000,000,000
Badge Eligibility	Yes

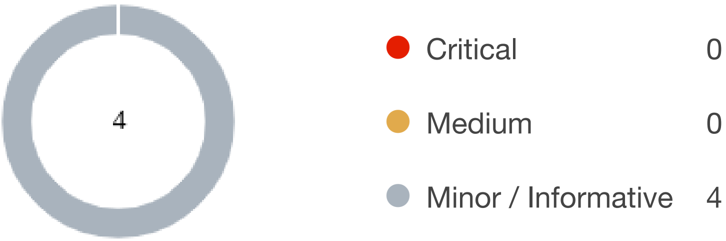
Audit Updates

Initial Audit	23 Jan 2024
Corrected Phase 2	27 Jan 2024

Source Files

Filename	SHA256
GalaxyFox.sol	0a3e380e9ee69325e5f2c7b3e02d82f55529d1fc2138cffc5673ac5a8274652c

Findings Breakdown



Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	1	3	0	0

PVO - Potential Volume Overflow

Criticality	Minor / Informative
Location	GalaxyFox.sol#L1499
Status	Unresolved

Description

The contract is designed to calculate the `period` variable representing the number of days passed since the original `block.timestamp`. This is achieved by dividing the current block timestamp by a constant representing one day (`DAY`). The contract then increments the `volume[sender][period]` mapping with the amount for each transaction. However, contrary to the contract's comments that suggest overflow is prevented when a user is within certain conditions, the current implementation does not adequately check the scenario where a user is flagged as `isExcludedFromDailyVolume`. In such cases, the volume `[sender][period]` can be repetitively increased for the same day without any restrictions. This could lead to an unintended behavior where the `volume` variable could be excessively incremented within a single day for the same `sender`. The use of the `unchecked` block in this context will bypass overflow checks, leading to incorrect calculations or vulnerabilities due to the volume variable overflowing.

```
       // DAY is constant so no division by 0
        // volume can't overflow if we reached this point (balance check)
        unchecked {
            // loss of precision is wanted here
            uint256 period = block.timestamp / DAY;

            volume[sender][period] += amount;
            require(
                volume[sender][period] <= maxDailyVolume ||
                    isExcludedFromDailyVolume[sender],
                "GalaxyFox: max daily volume exceeded"
            );
   }
```

Recommendation

It is recommended to revise the volume tracking logic to include overflow protection and consistent daily volume limits for all users, irrespective of their `isExcludedFromDailyVolume` status. This can be achieved by introducing additional checks or mechanisms that monitor and cap the daily transaction volume, ensuring it does not exceed predefined limits or risk overflow. Additionally, reevaluate the use of the unchecked block to ensure that overflow risks are adequately mitigated. Implementing these measures will enhance the contract's security and reliability, preventing potential vulnerabilities associated with unchecked arithmetic operations.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	GalaxyFox.sol#L572,999,1221,1223,1254,1570,1584,1598,1613,1614,1615,1633,1634,1635
Status	Acknowledged

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.

```
_marketingHolder
_liquidityHolder

...
```

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/v0.8.17/style-guide.html#naming-convention.

L09 - Dead Code Elimination

Criticality	Minor / Informative
Location	GalaxyFox.sol#L137,504,607,670,682,716,785,801,814,837,881
Status	Acknowledged

Description

In Solidity, dead code is code that is written in the contract, but is never executed or reached during normal contract execution. Dead code can occur for a variety of reasons, such as:

- Conditional statements that are always false.
- Functions that are never called.
- Unreachable code (e.g., code that follows a return statement).

Dead code can make a contract more difficult to understand and maintain, and can also increase the size of the contract and the cost of deploying and interacting with it.

```
function _contextSuffixLength() internal view virtual returns (uint256) {
    return 0;
}

...
```

Recommendation

To avoid creating dead code, it's important to carefully consider the logic and flow of the contract and to remove any code that is not needed or that is never executed. This can help improve the clarity and efficiency of the contract.

L17 - Usage of Solidity Assembly

Criticality	Minor / Informative
Location	GalaxyFox.sol#L735
Status	Acknowledged

Description

Using assembly can be useful for optimizing code, but it can also be error-prone. It's important to carefully test and debug assembly code to ensure that it is correct and does not contain any errors.

Some common types of errors that can occur when using assembly in Solidity include Syntax, Type, Out-of-bounds, Stack, and Revert.

```
assembly {
    let returndata_size := mload(returndata)
    revert(add(32, returndata), returndata_size)
}
```

Recommendation

It is recommended to use assembly sparingly and only when necessary, as it can be difficult to read and understand compared to Solidity code.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
	_contextSuffixLength	Internal		
IERC20Errors	Interface			

ERC20	Implementation	Context, IERC20, IERC20Metadata, IERC20Errors		
		Public	✓	-
	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	✓	
IERC20Permit	Interface			
	permit	External	✓	-

	nonces	External		-
	DOMAIN_SEPARATOR	External		-
Address	Library			
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	verifyCallResultFromTarget	Internal		
	verifyCallResult	Internal		
	_revert	Private		
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	forceApprove	Internal	✓	
	_callOptionalReturn	Private	✓	
	_callOptionalReturnBool	Private	✓	
Ownable	Implementation	Context		

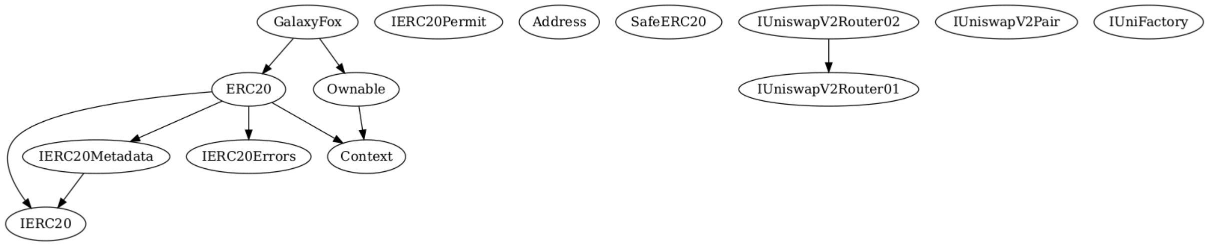
		Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-

	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-

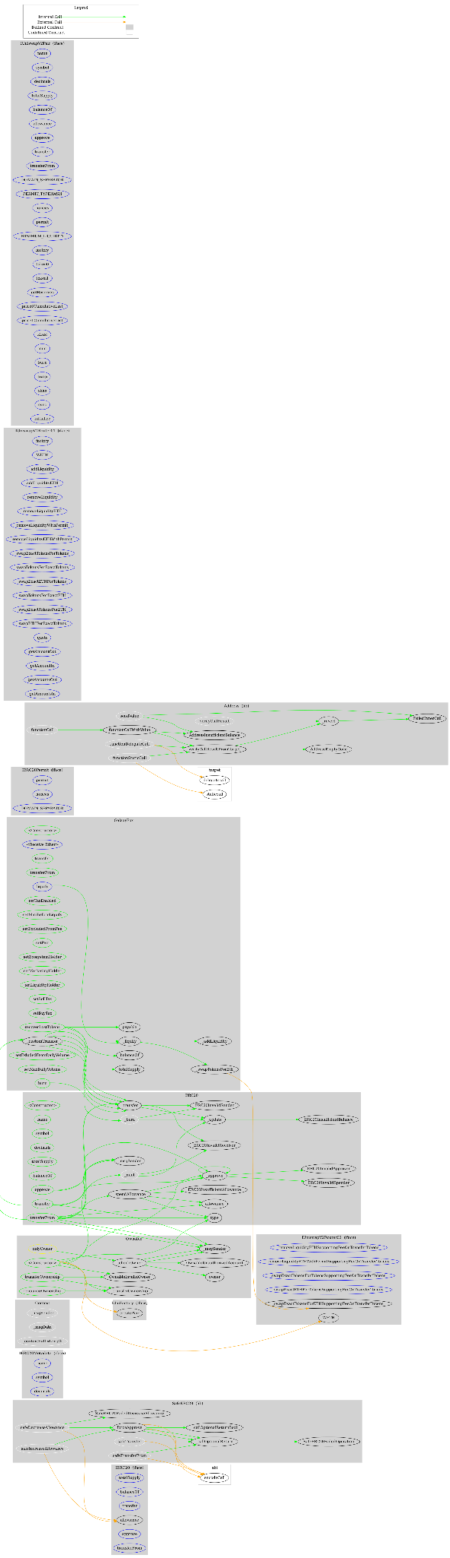
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniFactory	Interface			
	createPair	External	✓	-
GalaxyFox	Implementation	ERC20, Ownable		

		Public	✓	ERC20 Ownable
		External	Payable	-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_customTransfer	Internal	✓	
	setTaxEnabled	Public	✓	onlyOwner
	setMiniBeforeLiquify	Public	✓	onlyOwner
	setExcludedFromFee	Public	✓	onlyOwner
	setPair	Public	✓	onlyOwner
	setEcosystemHolder	Public	✓	onlyOwner
	setMarketingHolder	Public	✓	onlyOwner
	setLiquidityHolder	Public	✓	onlyOwner
	setSellTax	Public	✓	onlyOwner
	setBuyTax	Public	✓	onlyOwner
	burn	Public	✓	-
	recoverLostTokens	Public	✓	onlyOwner
	setExludedFromDailyVolume	Public	✓	onlyOwner
	setMaxDailyVolume	Public	✓	onlyOwner
	liquify	External	✓	onlyOwner
	_liquify	Private	✓	
	_swapTokensForEth	Internal	✓	
	_addLiquidity	Internal	✓	

Inheritance Graph



Flow Graph



Summary

Galaxy Fox contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Galaxy Fox 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. There is also a limit of max 20% buy and sell 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

https://www.cyberscope.io