Cyberscope

Audit Report

zkSwap Finance Token

January 2024

Network	zkSync
Address	0x31C2c031fDc9d33e974f327Ab0d9883Eae06cA4A
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	Acknowledged
●	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

Review

Explorer	https://explorer.zksync.io/address/0x31c2c031fdc9d33e974f327ab0d9883eae06ca4a

Audit Updates

Initial Audit	13 Jan 2024
Acknowledged Phase	24 Jan 2024

Source Files

Filename	SHA256
ZFToken.sol	b0923c28f1ce882e0b62bac42713667d54c49cb2cf768761c1c90b3e0bcc3f5d
@openzeppelin/contracts/utils/Strings.sol	0519199dbc635f98ce2e4537986604ee618bca665c65e9a1738702dfacf72010
@openzeppelin/contracts/utils/StorageSlot.sol	b4a5fb7ab93bfeda06509eafbd5f71fde0e0de84b6d9129553bd535a42166c15
@openzeppelin/contracts/utils/ShortStrings.sol	ddd52921d2996abf2e3d9c1c4f6d00194a3e3b278a164948f995862371444a55
@openzeppelin/contracts/utils/Nonces.sol	1c16c3cf8bb0679cbd47cddd8b141fea193e76966c94c858c5bccc94b8695030
@openzeppelin/contracts/utils/Context.sol	847fda5460fee70f56f4200f59b82ae622bb03c79c77e67af010e31b7e2cc5b6
@openzeppelin/contracts/utils/math/SignedMath.sol	768c28e3a33c3312e57ae8a1caaec2893bc89ac6e386621de018f85e9a2d6e99
@openzeppelin/contracts/utils/math/Math.sol	a6ee779fc42e6bf01b5e6a963065706e882b016affbedfd8be19a71ea48e6e15

@openzeppelin/contracts/utils/cryptography/MessageHashUtils.sol	2fd5c641cf452efd15f784827cb2835664970d7fbc166bf80824ed27011cc374
@openzeppelin/contracts/utils/cryptography/EIP712.sol	27dac0732a0154f432c0a7a1d1f067ab51116105e157d0e5d68d040fd83954d5
@openzeppelin/contracts/utils/cryptography/ECDSA.sol	37828cb50b47bcc51c7b770bde15d5885d871ef1e67028057a0b788c3568726e
@openzeppelin/contracts/token/ERC20/IERC20.sol	6f2faae462e286e24e091d7718575179644dc60e79936ef0c92e2d1ab3ca3cee
@openzeppelin/contracts/token/ERC20/ERC20.sol	ddff96777a834b51a08fec26c69bb6ca2d01d150a3142b3fdd8942e07921636a
@openzeppelin/contracts/token/ERC20/extensions/IERC20Permit.sol	912509e0e9bf74e0f8a8c92d031b5b26d2d35c6d4abf3f56251be1ea9ca946bf
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	1d079c20a192a135308e99fa5515c27acfbb071e6cdb0913b13634e630865939
@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol	677cb995a34f0cc937f3d77d4626c46fbf47cdef4c9cc0314c27672c0459cf80
@openzeppelin/contracts/token/ERC20/extensions/ERC20FlashMint.sol	58f4f4e5b759b5709a7ba705dfe60a26a60fb18154bff8cdf145a7c4e8c4c368
@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	2e6108a11184dd0caab3f3ef31bd15fed1bc7e4c781a55bc867ccedd8474565c
@openzeppelin/contracts/interfaces/draft-IERC6093.sol	4aea87243e6de38804bf8737bf86f750443d3b5e63dd0fd0b7ad92f77cdbd3e3
@openzeppelin/contracts/interfaces/IERC5267.sol	efd1ebd1e04b6ef9c3b8781a097588f83da954323f438d54a71dc06508e6c7b8
@openzeppelin/contracts/interfaces/IERC3156FlashLender.sol	3fb668ca6aaf756f5db9049abd2a18f638ff70307ca7ce59f85e772bae17380d

@openzeppelin/contracts/interfaces/IERC3156FlashBorrower.sol	06a759fc3607f87bfb716c95ac2f67c64b1485703bdd9467f1fea7ecc1180215
@openzeppelin/contracts/access/Ownable.sol	38578bd71c0a909840e67202db527cc6b4e6b437e0f39f0c909da32c1e30cb81

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	2

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

MT - Mints Tokens

Criticality	Minor / Informative
Location	ZFToken.sol#L27
Status	Acknowledged

Description

The role `onlyMinter`, which is assigned by the contract owner, has the authority to mint tokens. The `onlyMinter` 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) public onlyMinter
returns(bool) {
    _mint(to, amount);
    return true;
}
```

Recommendation

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.

Team Update

The team has acknowledged that this is not a security issue and states: *Currently, all minters are under either a 24/48-hour timelock or Multisigs Wallet. All information is transparent, public, and verifiable. More information can be found at: https://docs.zkswap.finance/contracts-and-audits/smart-contracts*

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	ZFToken.sol#L3
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 ^0.8.0;
```

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**
ZFToken	Implementation	ERC20, Ownable		
		Public	✓	-
	mint	Public	✓	onlyMinter
	burn	Public	✓	-
	addMinter	Public	✓	onlyOwner
	removeMinter	Public	✓	onlyOwner
	getMinterLength	Public		-
	isMinter	Public		-
	getMinter	Public		-
Strings	Library			
	toString	Internal		
	toStringSigned	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	equal	Internal		
StorageSlot	Library			

	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		
	getStringSlot	Internal		
	getStringSlot	Internal		
	getBytesSlot	Internal		
	getBytesSlot	Internal		
ShortStrings	Library			
	toShortString	Internal		
	toString	Internal		
	byteLength	Internal		
	toShortStringWithFallback	Internal	✓	
	toStringWithFallback	Internal		
	byteLengthWithFallback	Internal		
Nonces	Implementation			
	nonces	Public		-
	_useNonce	Internal	✓	
	_useCheckedNonce	Internal	✓	
Context	Implementation			

	_msgSender	Internal		
	_msgData	Internal		
	_contextSuffixLength	Internal		
SignedMath	Library			
	max	Internal		
	min	Internal		
	average	Internal		
	abs	Internal		
Math	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	max	Internal		
	min	Internal		
	average	Internal		
	ceilDiv	Internal		
	mulDiv	Internal		
	mulDiv	Internal		
	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		
EIP712	Implementation	IERC5267		
		Public	✓	-
	_domainSeparatorV4	Internal		
	_buildDomainSeparator	Private		
	_hashTypedDataV4	Internal		
	eip712Domain	Public		-
	_EIP712Name	Internal		
	_EIP712Version	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	✓	-
ERC20	Implementation	Context, IERC20, IERC20Metadata, IERC20Errors		
		Public	✓	-
	name	Public		-
	symbol	Public		-

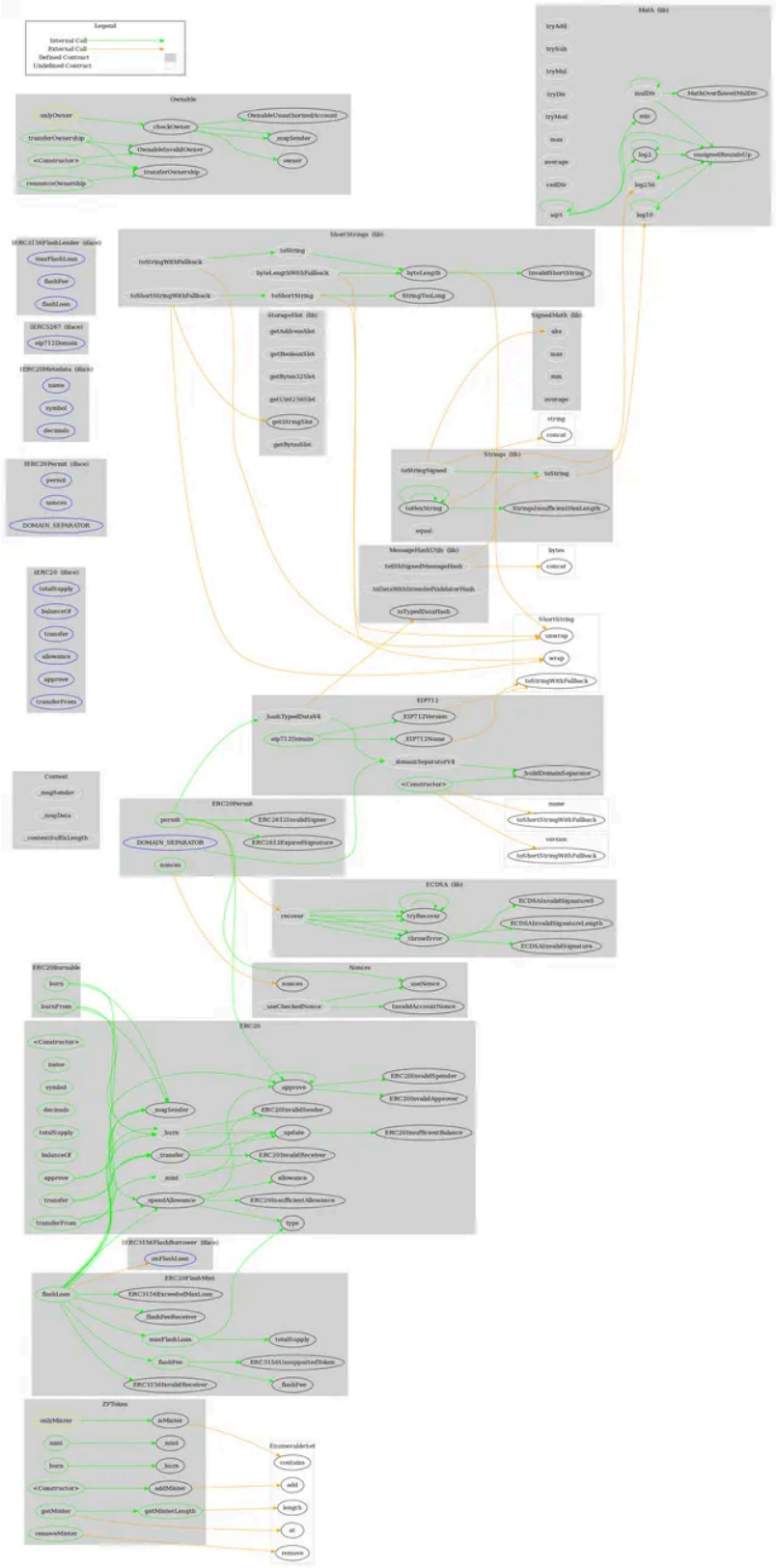
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	_transfer	Internal	✓	
	_update	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-

	decimals	External		-
ERC20Permit	Implementation	ERC20, IERC20Permit, EIP712, Nonces		
		Public	✓	EIP712
	permit	Public	✓	-
	nonces	Public		-
	DOMAIN_SEPARATOR	External		-
ERC20FlashMint	Implementation	ERC20, IERC3156FlashLender		
	maxFlashLoan	Public		-
	flashFee	Public		-
	_flashFee	Internal		
	_flashFeeReceiver	Internal		
	flashLoan	Public	✓	-
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-
	burnFrom	Public	✓	-
IERC20Errors	Interface			
IERC721Errors	Interface			

IERC1155Errors	Interface			
IERC5267	Interface			
	eip712Domain	External		-
IERC3156FlashLender	Interface			
	maxFlashLoan	External		-
	flashFee	External		-
	flashLoan	External	✓	-
IERC3156FlashBorrower	Interface			
	onFlashLoan	External	✓	-
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	

Inheritance Graph

Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
checkOwner
OwnableUnauthorizedAccount
transferOwnership
_msgSender
OwnableInvalidOwner
«Constructor»
owner
renounceOwnership
_transferOwnership
Math (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
max
average
ceilDiv
sqrt
mulDiv
MathOverflowedMulDiv
min
log2
unsignedRoundsUp
log256
log10
IERC3156FlashLender (iface)
maxFlashLoan
flashFee
flashLoan
IERC5267 (iface)
eip712Domain
IERC20Metadata (iface)
name
symbol
decimals
ShortStrings (lib)
toString
toStringWithFallback
byteLength
byteLengthWithFallback
toShortString
toShortStringWithFallback
InvalidShortString
StringTooLong
StorageSlot (lib)
getAddressSlot
getBooleanSlot
getBytes32Slot
getUint256Slot
getStringSlot
getBytesSlot
SignedMath (lib)
abs
max
min
average
string
concat
IERC20Permit (iface)
permit
nonces
DOMAIN_SEPARATOR
Strings (lib)
toStringSigned
toString
toHexString
StringsInsufficientHexLength
equal
MessageHashUtils (lib)
toEthSignedMessageHash
toDataWithIntendedValidatorHash
toTypedDataHash
bytes
concat
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
ShortString
unwrap
wrap
toStringWithFallback
EIP712
_hashTypedDataV4
eip712Domain
EIP712Version
EIP712Name
domainSeparatorV4
«Constructor»
_buildDomainSeparator
Context
_msgSender
_msgData
_contextSuffixLength
ERC20Permit
permit
ERC2612InvalidSigner
DOMAIN_SEPARATOR
ERC2612ExpiredSignature
nonces
name
toShortStringWithFallback
version
toShortStringWithFallback
ECDSA (lib)
recover
tryRecover
throwError
ECDSAInvalidSignature
ECDSAInvalidSignatureLength
ECDSAInvalidSignatureS
ERC20Burnable
burn
burnFrom
Nonces
nonces
useNonce
_useCheckedNonce
InvalidAccountNonce
ERC20
«Constructor»
name
symbol
decimals
totalSupply
balanceOf
approve
transfer
transferFrom
_msgSender
_burn
_transfer
_mint
_spendAllowance
approve
_update
allowance
type
ERC20InvalidSpender
ERC20InvalidApprover
ERC20InvalidSender
ERC20InsufficientBalance
ERC20InvalidReceiver
ERC20InsufficientAllowance
IERC3156FlashBorrower (iface)
onFlashLoan
ERC20FlashMint
flashLoan
ERC3156ExceededMaxLoan
flashFeeReceiver
maxFlashLoan
totalSupply
flashFee
ERC3156UnsupportedToken
ERC3156InvalidReceiver
flashFee
ZSToken
onlyMinter
isMinter
mint
_mint
burn
_burn
«Constructor»
addMinter
getMinter
getMinterLength
removeMinter
EnumerableSet
contains
add
length
at
remove

Summary

zkSwap Finance 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 mint tokens. If the minters abuse the mint functionality, then the contract will be highly inflated. The team has acknowledged the issue.

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