Cyberscope

Audit Report

eBit

February 2024

Network	ETH
Address	0x4f14ba78a51925ee934c373a2cf56b2d8da63f7f
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

Analysis 1
Diagnostics 2
Table of Contents 3
Review 4
Audit Updates 4
Source Files 4
Findings Breakdown 7
L19 - Stable Compiler Version 8
Description 8
Recommendation 8
Functions Analysis 9
Inheritance Graph 10
Flow Graph 11
Summary 12
Disclaimer 13
About Cyberscope 14

Review

Contract Name	EBit
Compiler Version	v0.8.19+commit.7dd6d404
Optimization	200 runs
Explorer	https://etherscan.io/address/0x4f14ba78a51925ee934c373a2cf56b2d8da63f7f
Address	0x4f14ba78a51925ee934c373a2cf56b2d8da63f7f
Network	ETH
Symbol	EBIT
Decimals	18
Total Supply	100,000,000
Badge Eligibility	Yes

Audit Updates

Initial Audit	24 Feb 2024

Source Files

Filename	SHA256
contracts/EBit.sol	8a423186c43618064440c2ca80e9dd2686756656b610f6f984031158b43d1c8c
@openzeppelin/contracts/utils/Strings.sol	cb2df477077a5963ab50a52768cb74ec6f32177177a78611ddbbe2c07e2d36de

@openzeppelin/contracts/utils/StorageSlot.sol	51c993d87ee6096af06f8701d8497272751c8cc9c9c2b31e67a3087d5f2302df
@openzeppelin/contracts/utils/ShortStrings.sol	0f1abe1770f7e80f0b9221aac780424a42e5c64b9814aa735d28115a0f549915
@openzeppelin/contracts/utils/Counters.sol	2fdcb1343e5621385b62e57b5c7775607c272122b6f2dc77da8f84828aa40cd0
@openzeppelin/contracts/utils/Context.sol	b2cfee351bcafd0f8f27c72d76c054df9b571b62cfac4781ed12c86354e2a56c
@openzeppelin/contracts/utils/math/SignedMath.sol	420a5a5d8d94611a04b39d6cf5f02492552ed4257ea82aba3c765b1ad52f77f6
@openzeppelin/contracts/utils/math/Math.sol	85a2caf3bd06579fb55236398c1321e15fd524a8fe140dff748c0f73d7a52345
@openzeppelin/contracts/utils/cryptography/EIP712.sol	6403cb83154d37ce7030e1b94f07766a6dd20835c4a691bcff87870810df15e2
@openzeppelin/contracts/utils/cryptography/ECDSA.sol	445963619903cee339e49aa2d7a0b07cfad90959529fff136394429c4a92d554
@openzeppelin/contracts/token/ERC20/IERC20.sol	7ebde70853ccafcf1876900dad458f46eb9444d591d39bfc58e952e2582f5587
@openzeppelin/contracts/token/ERC20/ERC20.sol	d20d52b4be98738b8aa52b5bb0f88943f62128969b33d654fbca731539a7fe0a
@openzeppelin/contracts/token/ERC20/extensions/IERC20Permit.sol	bbf4d26d660ce8f9c9887af3a46bf4eab5f4c15eb8413ad89b65ecdd05dee7a9
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol	89bfb6393150001a5e394e9347ba4addf460b3579450a6caa28ae7d1f24e427c
@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	0344809a1044e11ece2401b4f7288f414ea41fa9d1dad24143c84b737c9fc02e

@openzeppelin/contracts/interfaces/IERC5267.sol	72424d3a7a9b7d1c3584d4ec83bf116f30dfaf7180298a32669eefacb55ccb92
@openzeppelin/contracts/access/Ownable.sol	a8e4e1ae19d9bd3e8b0a6d46577eec098c01fbaffd3ec1252fd20d799e73393b

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	1

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/EBit.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 ^0.8.19;
```

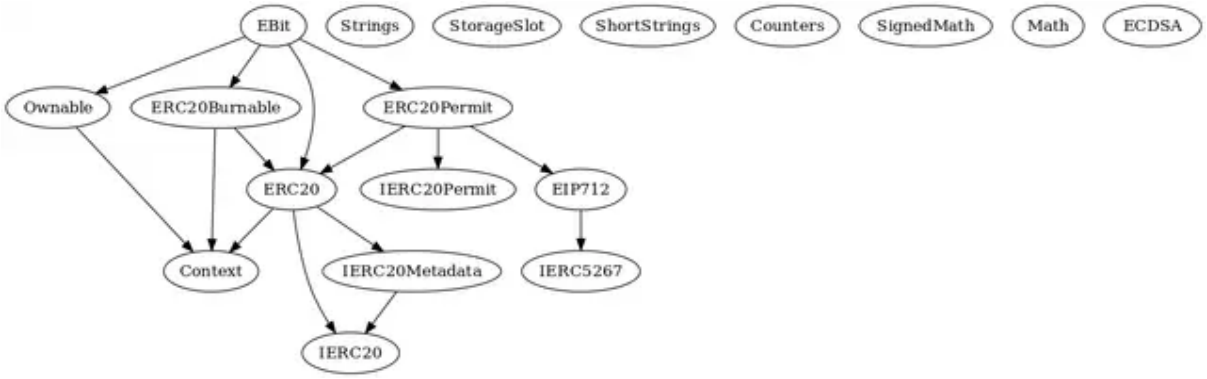
Recommendation

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.

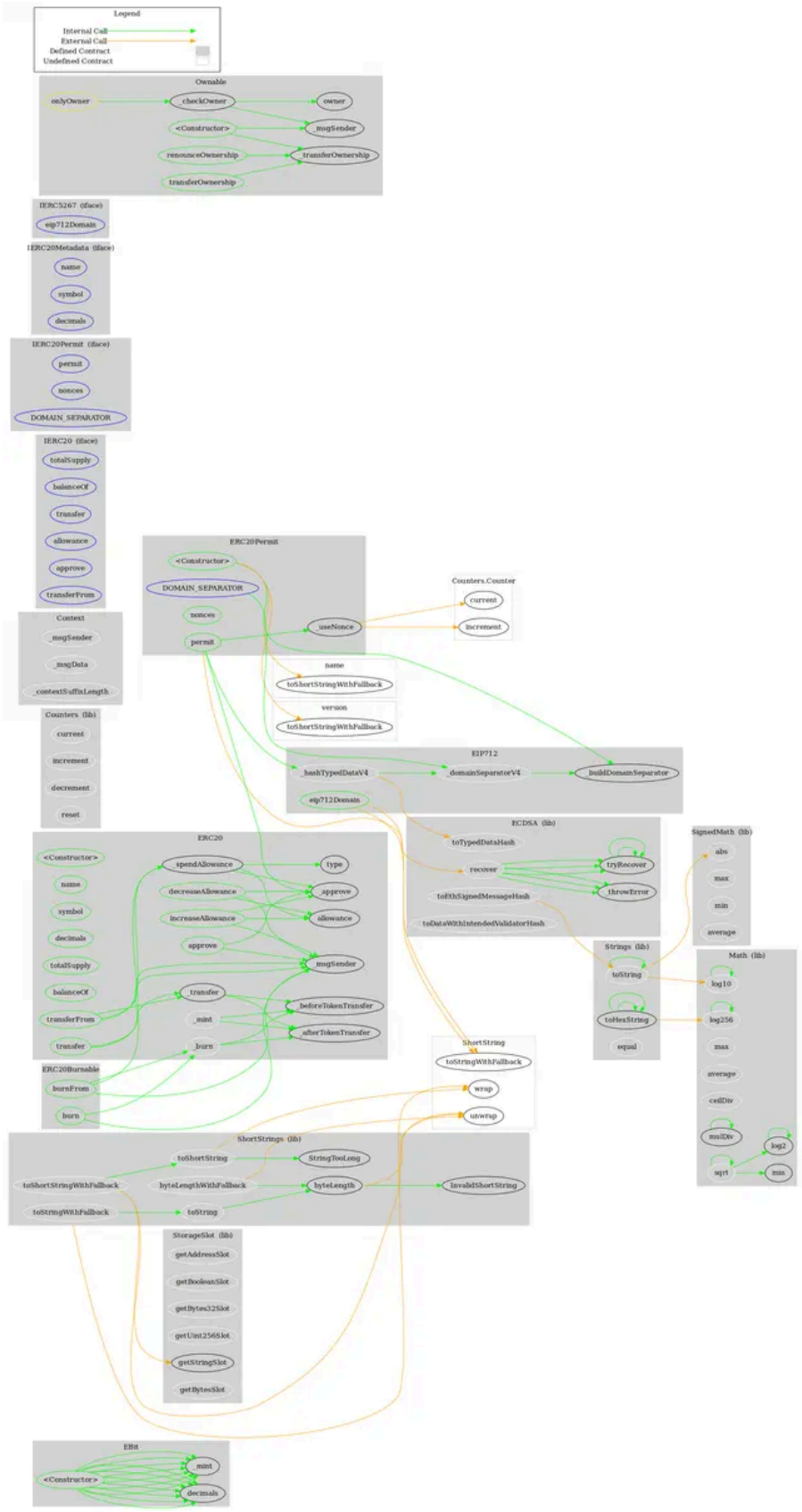
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
EBit	Implementation	ERC20, ERC20Burnable, Ownable, ERC20Permit		
		Public	✓	ERC20 ERC20Permit

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
checkOwner
owner
<Constructor>
msgSender
renounceOwnership
transferOwnership
transferOwnership
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
ERC20Permit
<Constructor>
DOMAIN_SEPARATOR
nonces
permit
useNonce
Counters.Counter
current
increment
name
toShortStringWithFallback
version
toShortStringWithFallback
EIP712
_hashTypedDataV4
domainSeparatorV4
buildDomainSeparator
eip712Domain
ECDSA (lib)
toTypedDataHash
recover
tryRecover
throwError
toEthSignedMessageHash
toDataWithIntendedValidatorHash
SignedMath (lib)
abs
max
min
average
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transferFrom
transfer
spendAllowance
decreaseAllowance
increaseAllowance
approve
transfer
_mint
_burn
type
_approve
allowance
msgSender
_beforeTokenTransfer
_afterTokenTransfer
Strings (lib)
toString
toHexString
equal
Math (lib)
log10
log256
max
average
ceilDiv
mulDiv
log2
sqrt
min
ERC20Burnable
burnFrom
burn
ShortString
toStringWithFallback
wrap
unwrap
ShortStrings (lib)
toShortString
StringTooLong
byteLength
InvalidShortString
toShortStringWithFallback
byteLengthWithFallback
toStringWithFallback
toString
StorageSlot (lib)
getAddressSlot
getBooleanSlot
getBytes32Slot
getUint256Slot
getStringSlot
getBytesSlot
EBit
<Constructor>
mint
decimals

Summary

eBit contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. eBit is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error 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

https://www.cyberscope.io