Cyberscope

A TAC Security Company

Audit Report

Bitelions Token

January 2026

Network	BSC
Address	0x7F206d9E6e1783f92c763C7E9900E6d07E0D561D
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
●	MVN	Misleading Variables Naming	Unresolved
●	MC	Missing Check	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	BITELIONS
Compiler Version	v0.8.30+commit.73712a01
Optimization	200 runs
Explorer	https://bscscan.com/address/0x7f206d9e6e1783f92c763c7e9900e6d07e0d561d
Address	0x7f206d9e6e1783f92c763c7e9900e6d07e0d561d
Network	BSC
Symbol	BTL
Decimals	18
Total Supply	100,000,000

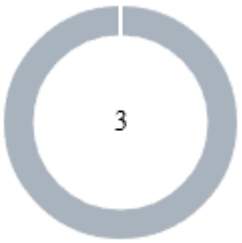
Audit Updates

Initial Audit	11 Jan 2026

Source Files

Filename	SHA256
BITELIONS.sol	41502cf4e0bc3fb4e10f91c94b6f51423d7aea9e14cc35554d6b0bc8174269d0

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	3

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

MVN - Misleading Variables Naming

Criticality	Minor / Informative
Location	BITELIONS.sol#L1456,1484
Status	Unresolved

Description

Variables can have misleading names if their names do not accurately reflect the value they contain or the purpose they serve. The contract uses some variable names that are too generic or do not clearly convey the information stored in the variable. Misleading variable names can lead to confusion, making the code more difficult to read and understand.

```
Shell
function enableTrading() external onlyAdmin {
        require(!tradingEnabled, "Already enabled");
        tradingEnabled = true;
        firstBlock = block.number;
        emit TradingEnabled();
....

if (tradingEnabled && block.number == firstBlock &&
sender == pairAddress) {
            revert("Anti-sniper: trading not allowed in
first block");

        }
```

Recommendation

It's always a good practice for the contract to contain variable names that are specific and descriptive. The team is advised to keep in mind the readability of the code.

MC - Missing Check

Criticality	Minor / Informative
Location	BITELIONS.sol#L1441,1446,1447,1451,
Status	Unresolved

Description

The contract is processing variables that have not been properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

Shell

```
function setFeeExempt(address wallet, bool exempt) external
onlyAdmin {
        isFeeExempt[wallet] = exempt;
        emit FeeExemptUpdated(wallet, exempt);

function lockWallet(address wallet, uint256
secondsDuration) external onlyAdmin {
        lockedUntil[wallet] = block.timestamp +
secondsDuration;
        emit WalletLocked(wallet, lockedUntil[wallet]);

function unlockWallet(address wallet) external onlyAdmin {
        lockedUntil[wallet] = 0;
        emit WalletUnlocked(wallet);
    }


lockedUntil[wallet] = block.timestamp + secondsDuration;
```

Recommendation

The team is advised to properly check the variables according to the required specifications.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	BITELIONS.sol#L1384
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.

```
Shell

pragma solidity ^0.8.20;
```

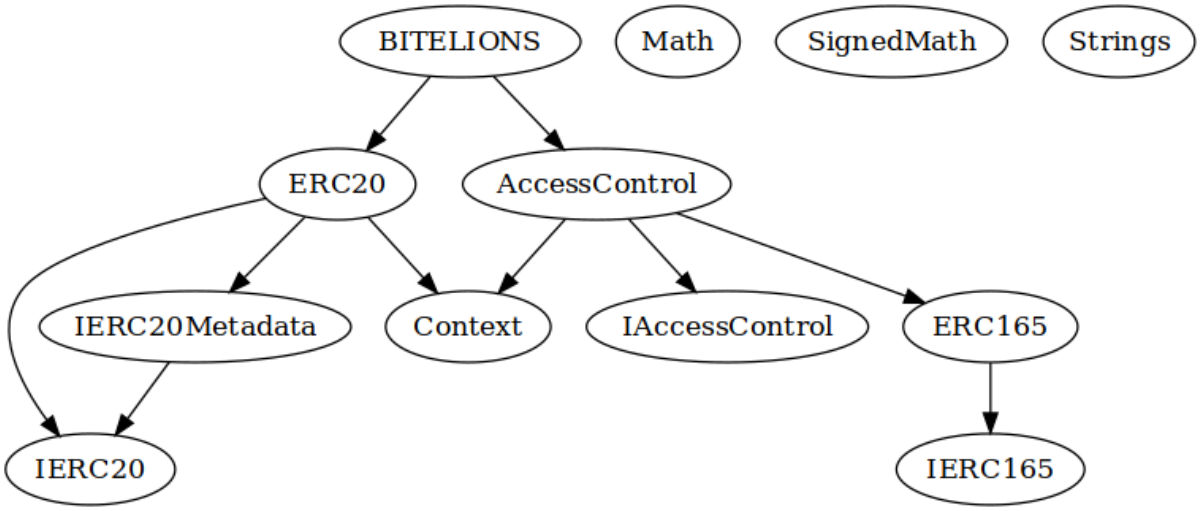
Recommendation

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.

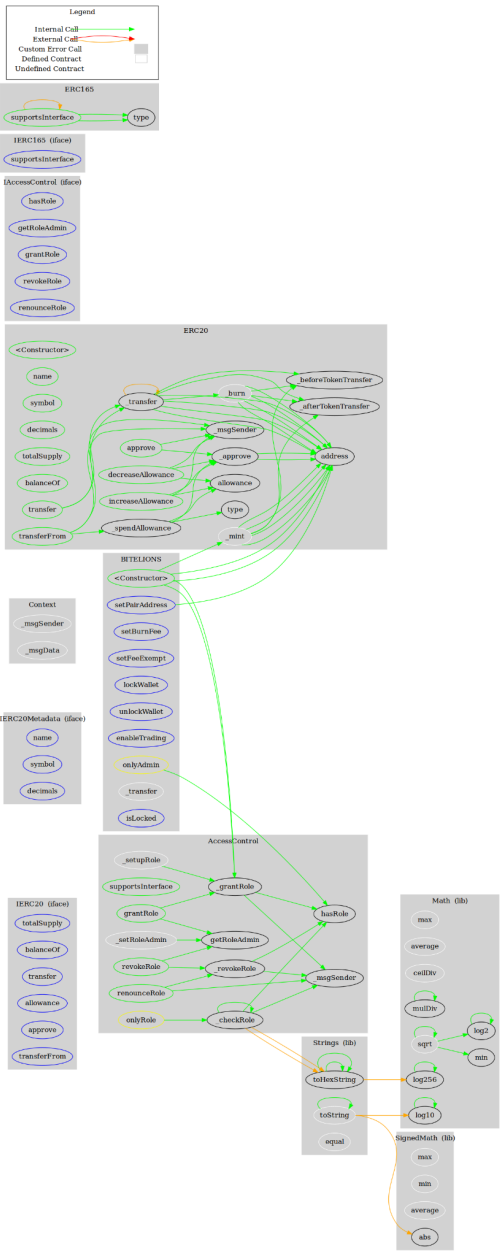
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
BITELIONS	Implementation	ERC20, AccessControl		
		Public	✓	ERC20
	setBurnFee	External	✓	onlyAdmin
	setFeeExempt	External	✓	onlyAdmin
	lockWallet	External	✓	onlyAdmin
	unlockWallet	External	✓	onlyAdmin
	enableTrading	External	✓	onlyAdmin
	setPairAddress	External	✓	onlyAdmin
	_transfer	Internal	✓	
	isLocked	External		-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
ERC165
supportsInterface
type
IERC165 (iface)
supportsInterface
IAccessControl (iface)
hasRole
getRoleAdmin
grantRole
revokeRole
renounceRole
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transfer
transferFrom
approve
decreaseAllowance
increaseAllowance
spendAllowance
_transfer
_burn
_mint
_approve
_beforeTokenTransfer
_afterTokenTransfer
_msgSender
address
allowance
type
BITELIONS
<Constructor>
setPairAddress
setBurnFee
setFeeExempt
lockWallet
unlockWallet
enableTrading
onlyAdmin
_transfer
isLocked
Context
_msgSender
_msgData
IERC20Metadata (iface)
name
symbol
decimals
AccessControl
_setupRole
supportsInterface
grantRole
_setRoleAdmin
revokeRole
renounceRole
onlyRole
_grantRole
hasRole
getRoleAdmin
_revokeRole
_msgSender
_checkRole
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Math (lib)
max
average
ceilDiv
mulDiv
sqrt
log2
min
log256
log10
Strings (lib)
toHexString
toString
equal
SignedMath (lib)
max
min
average
abs

Summary

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

The contract's ADMIN ROLE has been revoked. The information regarding the transaction can be accessed through the following link:

https://bscscan.com/tx/0x6c17c798a65edea69b20a2d93bdb79fb9c5861dbd9a96403253dae508d9684ef

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 TAC 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.

A **TAC Security** Company

The Cyberscope team

cyberscope.io