Cyberscope

Audit Report

PREME

January 2025

SHA256 73622cc067d0c04a5712af9e994fd320853b4d4434e2efb568c370aa9c928344

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
	MEA	Misleading Event Argument	Unresolved
	NCE	Non Compliant ERC20	Unresolved
	L11	Unnecessary Boolean equality	Unresolved
	L13	Divide before Multiply Operation	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	PREME
Compiler Version	v0.8.20+commit.a1b79de6
Optimization	200 runs
Test Deployment	https://testnet.bscscan.com/address/0x2b94BC2543daEc2D0850EBfd62d8A14c3e781718
Address	0x2b94BC2543daEc2D0850EBfd62d8A14c3e781718
Network	BSC_TESTNET
Symbol	PREME
Decimals	18
Badge Eligibility	Yes

Audit Updates

Initial Audit	11 Jan 2025

Source Files

Filename	SHA256
PREME.sol	73622cc067d0c04a5712af9e994fd320853b4d4434e2efb568c370aa9c928344

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	4

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	4	0	0	0

MEA - Misleading Event Argument

Criticality	Minor / Informative
Location	PREME.sol#L223
Status	Unresolved

Description

The `FailedToSwap` event is designed to log failures during token-to-ETH swaps, expecting the token's address and the amount involved as arguments. However, the contract incorrectly passes the `teamWallet` (a wallet address) as the first argument when emitting the event. This behavior is misleading and can result in confusion when analyzing emitted events, as the logged token address might incorrectly represent a wallet address instead of the token contract address involved in the failed swap. Additionally, this error could hinder accurate debugging and monitoring of swap failures.

```
event FailedToSwap(address indexed token, uint256 amount);
...
emit FailedToSwap(teamWallet, amount);
```

Recommendation

To mitigate this issue, the team could replace the `teamWallet` with the correct token contract address when emitting the `FailedToSwap` event. By addressing this issue, the contract will produce accurate event logs, improving traceability, monitoring, and debugging of swap failures.

NCE - Non Compliant ERC20

Criticality	Minor / Informative
Location	PREME.sol#L144
Status	Unresolved

Description

The `_update` function overrides OpenZeppelin's implementation and includes custom logic that alters the recipient address to `address(0)` if it matches `DEAD_WALLET` or `DEAD_WALLET_VB` . While this behavior effectively burns tokens, it violates the ERC20 specification, which expects transfers to respect the explicitly provided recipient address. This deviation from the standard can lead to unexpected behavior, compatibility issues with external tools, and confusion for users or applications interacting with the contract.

```
if (to == DEAD_WALLET || to == DEAD_WALLET_VB) to = address(0);
super._update(from, to, value);
```

Recommendation

The team is advised to allow transfers to `DEAD_WALLET` or `DEAD_WALLET_VB` without modifications, as tokens sent to these addresses are already considered irrecoverable and effectively burned. By addressing this issue, the contract will align with the ERC20 standard while maintaining its intended burn functionality.

L11 - Unnecessary Boolean equality

Criticality	Minor / Informative
Location	PREME.sol#L246
Status	Unresolved

Description

Boolean equality is unnecessary when comparing two boolean values. This is because a boolean value is either true or false, and there is no need to compare two values that are already known to be either true or false.

it's important to be aware of the types of variables and expressions that are being used in the contract's code, as this can affect the contract's behavior and performance. The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
require(
            value != true ||
            address(this) == Pair(pair).token0() ||
            address(this) == Pair(pair).token1(),
            "Address is not a pair of this token"
        )
```

Recommendation

Using the boolean value itself is clearer and more concise, and it is generally considered good practice to avoid unnecessary boolean equalities in Solidity code.

L13 - Divide before Multiply Operation

Criticality	Minor / Informative
Location	PREME.sol#L154,157
Status	Unresolved

Description

It is important to be aware of the order of operations when performing arithmetic calculations. This is especially important when working with large numbers, as the order of operations can affect the final result of the calculation. Performing divisions before multiplications may cause loss of prediction.

```
uint256 maxSwapAmount = balanceOf(swapPair) / 100
handledTokens = maxSwapAmount * totalTax / swapTax
```

Recommendation

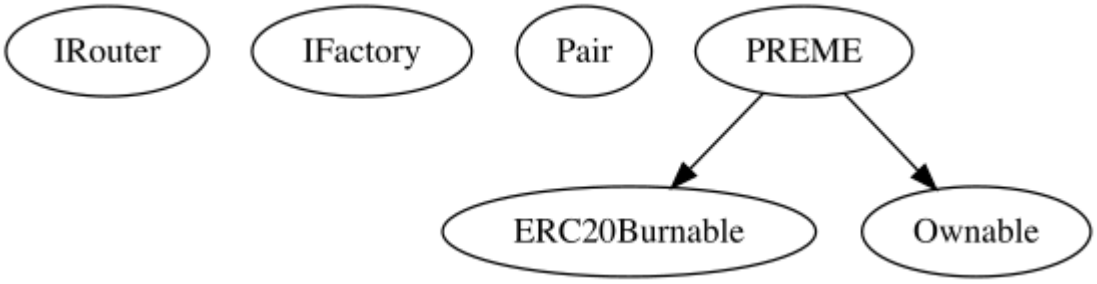
To avoid this issue, it is recommended to carefully consider the order of operations when performing arithmetic calculations in Solidity. It's generally a good idea to use parentheses to specify the order of operations. The basic rule is that the multiplications should be prior to the divisions.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IRouter	Interface			
	factory	External		-
	swapExactTokensForETH	External	✓	-
	swapExactETHForTokens	External	Payable	-
IFactory	Interface			
	getPair	External		-
Pair	Interface			
	token0	External		-
	token1	External		-
PREME	Implementation	ERC20Burnable, Ownable		
		Public	✓	ERC20 Ownable
	_update	Internal	✓	
	handleTax	Private	✓	
	buyAndBurnTokens	Private	✓	lockTheSwap
	swapTokensForETH	Private	✓	lockTheSwap
	setSwapPair	Private	✓	

	setPair	Public	✓	onlyOwner
	setSwapAtPercentage	Public	✓	onlyOwner
	setTax	Public	✓	onlyOwner
	setExcludedFromTaxStatus	Public	✓	onlyOwner
	setTeamWallet	Public	✓	onlyOwner
	setNftWallet	Public	✓	onlyOwner
	manualSwap	External	✓	onlyOwner
	manualBuyAndBurn	External	✓	onlyOwner
		External	Payable	-

Inheritance Graph



Summary

PREME Token contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. PREME 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 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 4% 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

cyberscope.io