Cyberscope

A TAC Security Company

Audit Report

OneVoice

October 2025

Network	BSC
Address	0x95fC8E3982729a76d607CB8F009d93EC24EAbeC8
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
●	ROF	Redundant Ownable Functionality	Unresolved
●	L19	Stable Compiler Version	Unresolved

Table of Contents

Analysis **2**
Diagnostics **3**
Table of Contents **4**
Risk Classification **5**
Review **6**
Audit Updates 6
Source Files 6
Findings Breakdown **8**
ROF - Redundant Ownable Functionality 9
Description 9
Recommendation 9
L19 - Stable Compiler Version 10
Description 10
Recommendation 10
Functions Analysis **11**
Inheritance Graph **12**
Flow Graph **13**
Summary **14**
Disclaimer **15**
About Cyberscope **16**

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	OneVoiceToken
Compiler Version	v0.8.20+commit.a1b79de6
Optimization	200 runs
Explorer	https://bscscan.com/address/0x95fc8e3982729a76d607cb8f009d93ec24eabec8
Address	0x95fc8e3982729a76d607cb8f009d93ec24eabec8
Network	BSC
Symbol	Voice
Decimals	18
Total Supply	1,000,000,000

Audit Updates

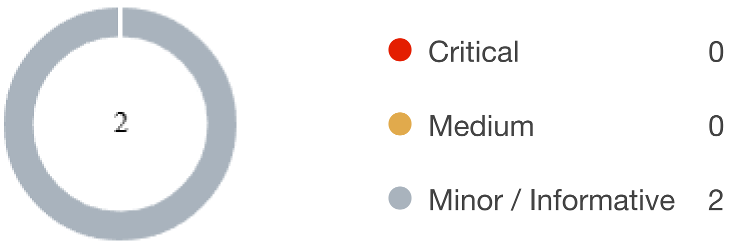
Initial Audit	09 Oct 2025

Source Files

Filename	SHA256
contracts/OneVoiceToken.sol	e1b554f3c7d04c8544239347b184006d7215d81e765de08feacd34a3507f076c
@openzeppelin/contracts/utils/Nonces.sol	9b4cbb85d1f5053c744e83302538eb643a713ffd14bc37665b224f1c66529339
@openzeppelin/contracts/utils/Context.sol	847fda5460fee70f56f4200f59b82ae622bb03c79c77e67af010e31b7e2cc5b6
@openzeppelin/contracts/utils/cryptography/EIP712.sol	75b837fe3868fd4217cc5e9a6ca89055b7277dc7a41b01db0fe6253ebe6aa95d
@openzeppelin/contracts/token/ERC20/IERC20.sol	30edf7394bab78d48b7db3a059248e1ea7c2c77d2ec0e37a13bb91415aafbe5a

@openzeppelin/contracts/token/ERC20/ERC20.sol	c08afc9ba498f2e0262075e565baccd4311db16a354ac63b3d14b930a5c69671
@openzeppelin/contracts/token/ERC20/extensions/IERC20Permit.sol	026aca1c8ee4574eb9719dca7dfc33e3e57a618715ae702a675e8a8c9ea1e82d
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	9e7c70ec72d2f7d592e23ea84f3852b04f91f6f644ce57e0263493046b36afb9
@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol	75f9f66db047b1413aa45538a53211e7b20479d74c3dd2657335bf4dc50b8811
@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	2e6108a11184dd0caab3f3ef31bd15fed1bc7e4c781a55bc867ccedd8474565c
@openzeppelin/contracts/interfaces/IERC5267.sol	efd1ebd1e04b6ef9c3b8781a097588f83da954323f438d54a71dc06508e6c7b8
@openzeppelin/contracts/access/Ownable.sol	38578bd71c0a909840e67202db527cc6b4e6b437e0f39f0c909da32c1e30cb81

Findings Breakdown



Severity	Unresolved	Acknowledged	Resolved	Other
● Critical	0	0	0	0
● Medium	0	0	0	0
● Minor / Informative	2	0	0	0

ROF - Redundant Ownable Functionality

Criticality	Minor / Informative
Location	OneVoiceToken.sol#L7,18
Status	Unresolved

Description

The `OneVoiceToken` contract inherits from the `Ownable` contract. This contract is typically used to implement access control by designating an owner account with exclusive privileges for executing restricted functions. However, in the current implementation, none of the contract's functions utilize `onlyOwner` or any ownership-related logic. As a result, the inheritance of `Ownable` is redundant and introduces unnecessary code complexity.

Shell

```
import {Ownable} from
"@openzeppelin/contracts/access/Ownable.sol";

contract OneVoiceToken is ERC20, ERC20Burnable,
ERC20Permit, Ownable
```

Recommendation

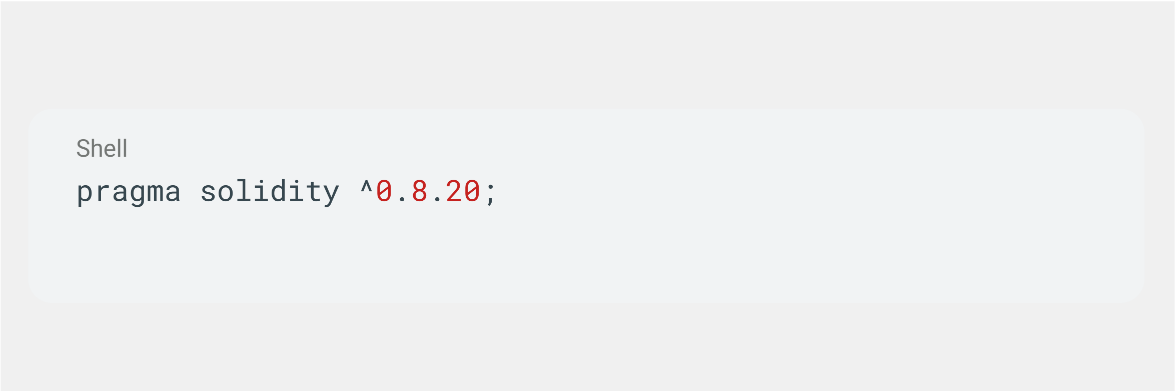
It is recommended to remove the unused `Ownable` inheritance to eliminate redundancy, improve code clarity, and reduce the overall contract size. This will enhance readability, maintainability, and gas efficiency.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	OneVoiceToken.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.



Shell

```
pragma solidity ^0.8.20;
```

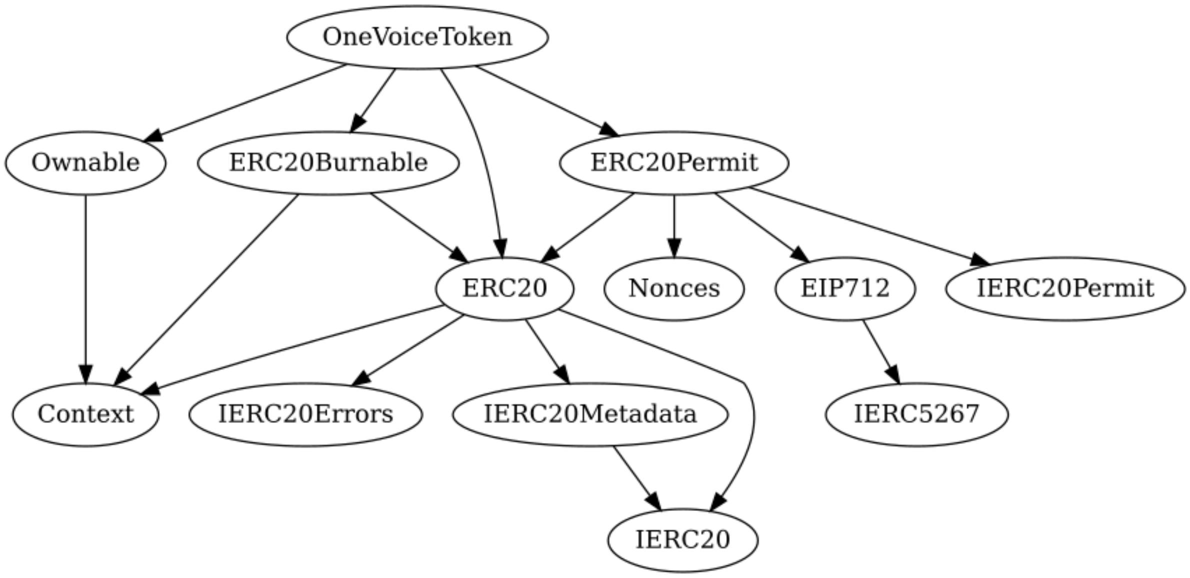
Recommendation

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.

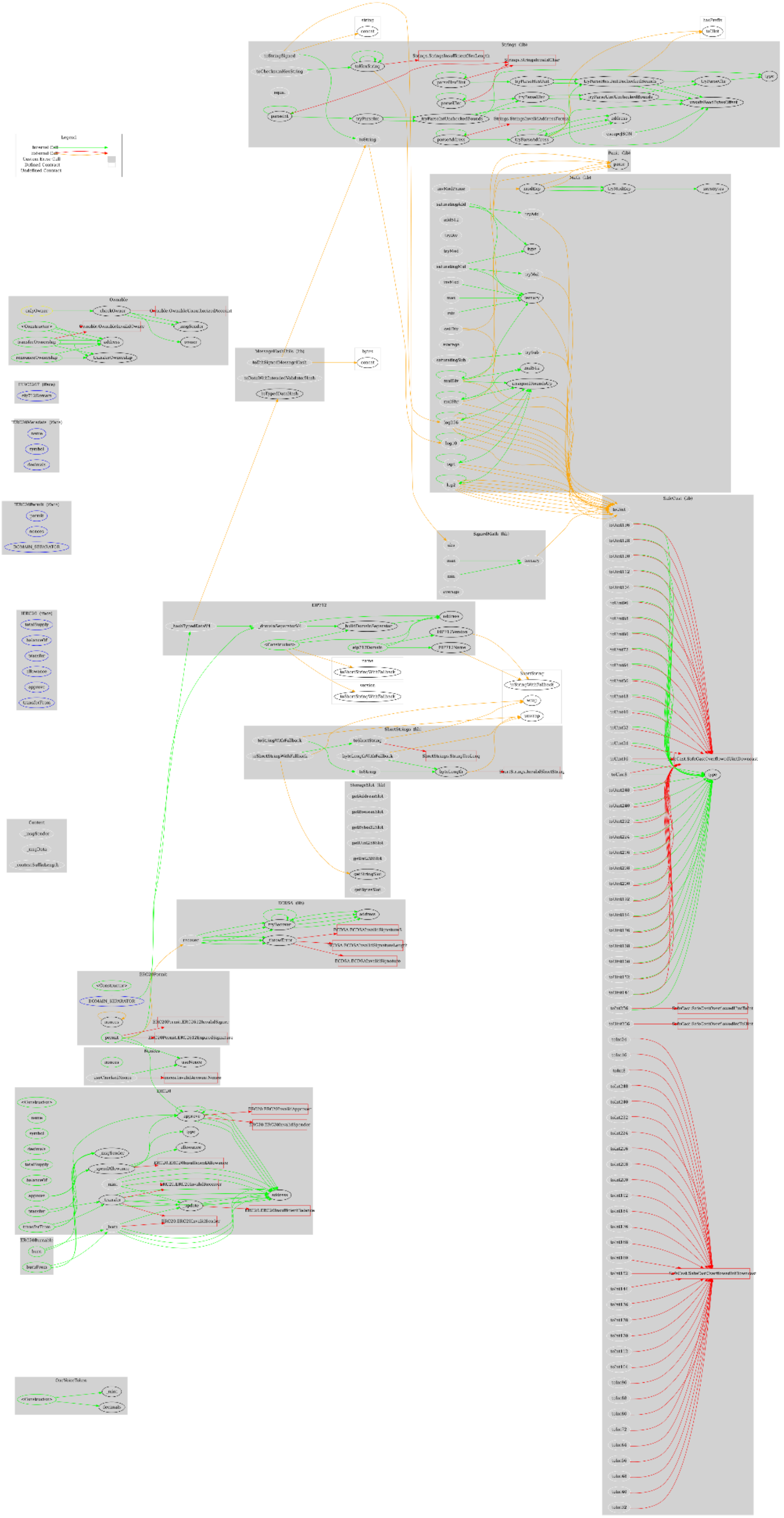
Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
OneVoiceToken	Implementation	ERC20, ERC20Burnable, ERC20Permit, Ownable		

Inheritance Graph



Flow Graph



Summary

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

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