Cyberscope

Audit Report

OneXchain

September 2024

Network	Address
ETH	0xaB7752c21CdE29D262C759046841E39f2d5CD654
BSC	0xaB7752c21CdE29D262C759046841E39f2d5CD654
MATIC	0xaB7752c21CdE29D262C759046841E39f2d5CD654
TRON	TPRCA6LBDvGAWfTECcqMZ1xDTzHYbXrUeM
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
●	L04	Conformance to Solidity Naming Conventions	Acknowledged
●	L15	Local Scope Variable Shadowing	Acknowledged
●	L18	Multiple Pragma Directives	Acknowledged
●	L19	Stable Compiler Version	Acknowledged

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	onexchain
Compiler Version	v0.8.19+commit.7dd6d404
Optimization	200 runs
Symbol	OXO
Decimals	18
Total Supply	2,000,000,000
Badge Eligibility	Yes

Deployments

Network	Explorer	Address
ETH	https://etherscan.io/address/0xaB7752c21CdE29D262C759046841E39f2d5CD654	0xaB7752c21CdE29D262C759046841E39f2d5CD654
BSC	https://bscscan.com/address/0xaB7752c21CdE29D262C759046841E39f2d5CD654	0xaB7752c21CdE29D262C759046841E39f2d5CD654
MATIC	https://polygonscan.com/address/0xaB7752c21CdE29D262C759046841E39f2d5CD654	0xaB7752c21CdE29D262C759046841E39f2d5CD654
TRON	hhttps://tronscan.org/#/contract/TPRCA6LBDvGAWfTECcqMZ1xDTzHYbXrUeM	TPRCA6LBDvGAWfTECcqMZ1xDTzHYbXrUeM

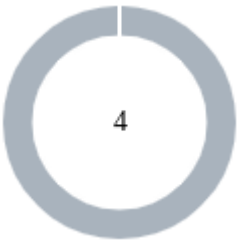
Audit Updates

Initial Audit	21 Mar 2024
Corrected Phase 2	25 Sep 2024

Source Files

Filename	SHA256
onexchain.sol	5b46642edaa7a55f50b67a57ee35392ff78cdb69dc4c5176fc1689c7625eacad

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

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	onexchain.sol#L642
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
contract onexchain is ERC20, Ownable {
    constructor (string memory name, string memory symbol, uint256
total) ERC20 (name, symbol) {
        _mint(msg.sender, total * 10 ** decimals());
    }

    function burn(address from, uint256 amount) public onlyOwner {
        _burn(from, amount);
    }
}
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/stable/style-guide.html#naming-conventions.

L15 - Local Scope Variable Shadowing

Criticality	Minor / Informative
Location	onexchain.sol#L643
Status	Unresolved

Description

Local scope variable shadowing occurs when a local variable with the same name as a variable in an outer scope is declared within a function or code block. When this happens, the local variable "shadows" the outer variable, meaning that it takes precedence over the outer variable within the scope in which it is declared.

```
string memory name,
string memory symbol,
```

Recommendation

It's important to be aware of shadowing when working with local variables, as it can lead to confusion and unintended consequences if not used correctly. It's generally a good idea to choose unique names for local variables to avoid shadowing outer variables and causing confusion.

L18 - Multiple Pragma Directives

Criticality	Minor / Informative
Location	onexchain.sol#L7,31,127,187,268,298,640
Status	Unresolved

Description

If the contract includes multiple conflicting pragma directives, it may produce unexpected errors. To avoid this, it's important to include the correct pragma directive at the top of the contract and to ensure that it is the only pragma directive included in the contract.

```
pragma solidity ^0.8.19;
```

Recommendation

It is important to include only one pragma directive at the top of the contract and to ensure that it accurately reflects the version of Solidity that the contract is written in.

By including all required compiler options and flags in a single pragma directive, the potential conflicts could be avoided and ensure that the contract can be compiled correctly.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	onexchain.sol#L7,31,127,187,268,298,640
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;

pragma solidity ^0.8.19;
```

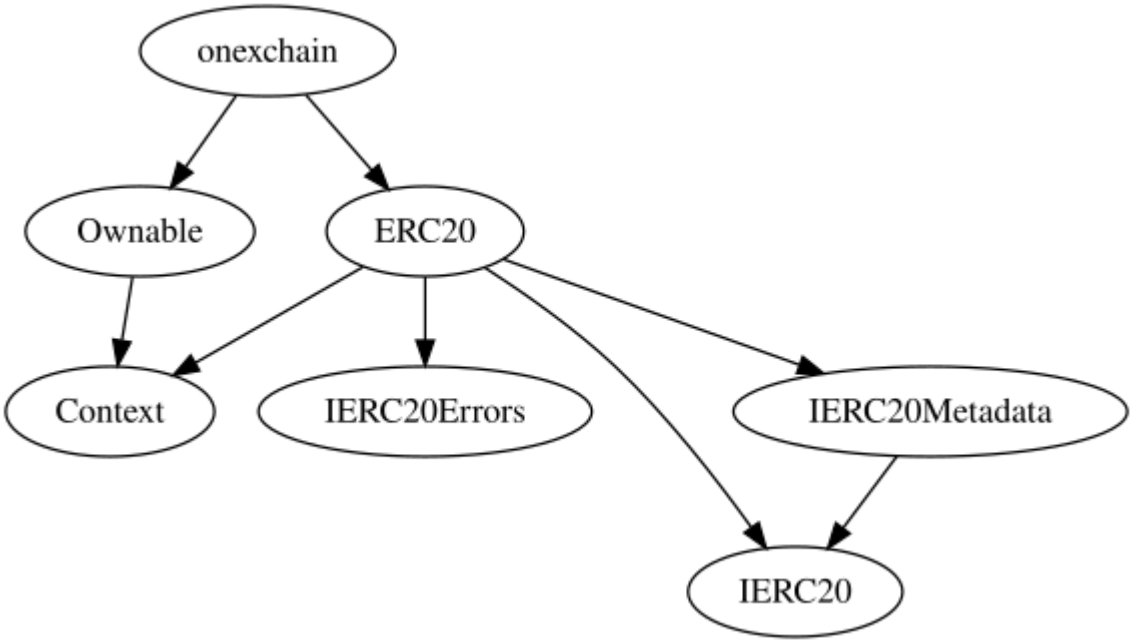
Recommendation

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.

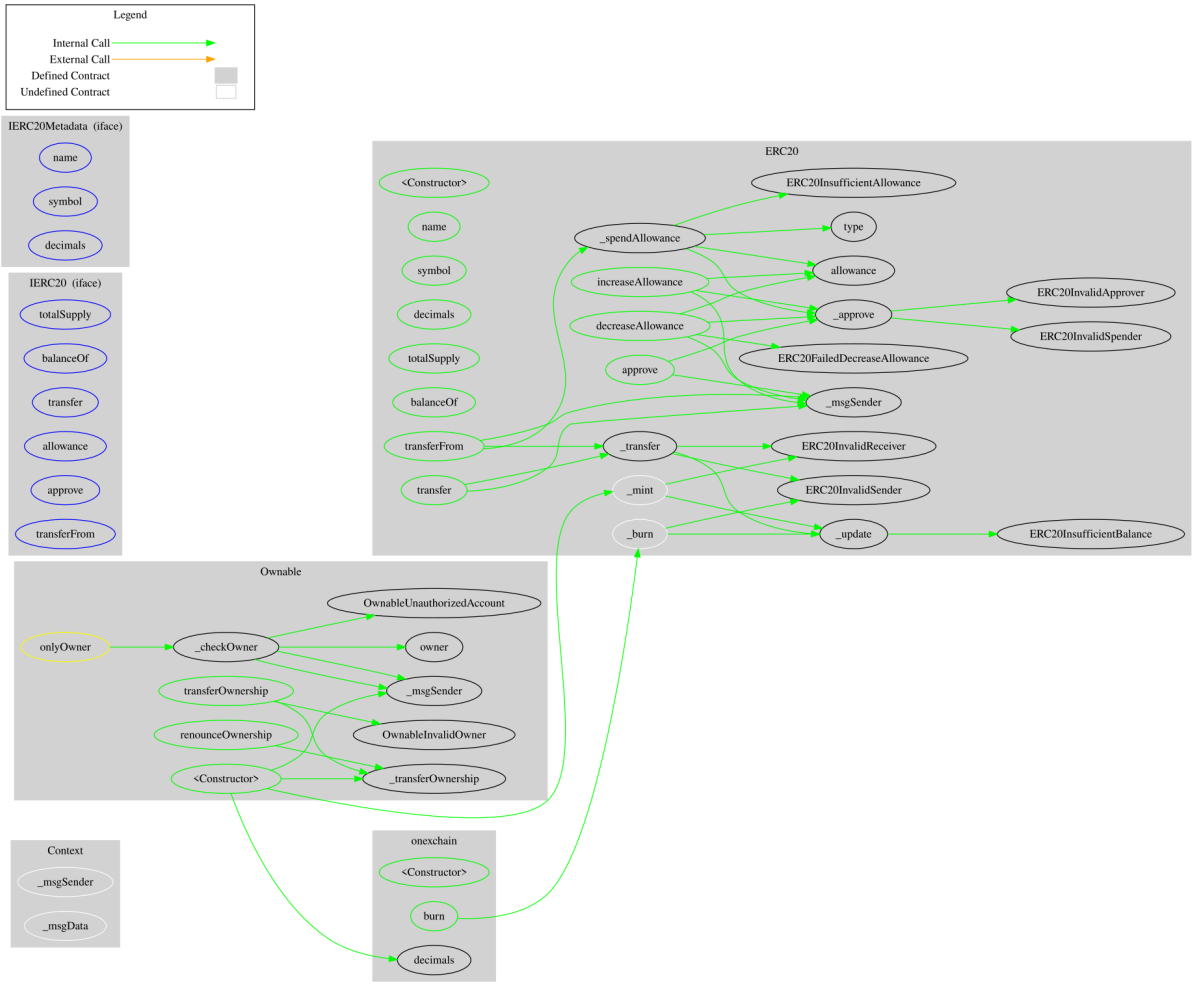
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
onexchain	Implementation	ERC20, Ownable		
		Public	✓	ERC20
	burn	Public	✓	onlyOwner

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transferFrom
transfer
_spendAllowance
increaseAllowance
decreaseAllowance
approve
_transfer
_mint
_burn
ERC20InsufficientAllowance
type
allowance
_approve
ERC20FailedDecreaseAllowance
_msgSender
ERC20InvalidReceiver
ERC20InvalidSender
_update
ERC20InvalidApprover
ERC20InvalidSpender
ERC20InsufficientBalance
Ownable
onlyOwner
_checkOwner
transferOwnership
renounceOwnership
<Constructor>
OwnableUnauthorizedAccount
owner
_msgSender
OwnableInvalidOwner
_transferOwnership
Context
_msgSender
_msgData
onexchain
<Constructor>
burn
decimals

Summary

OneXchain contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. OneXchain 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.

The ownership of the contracts have been renounced. The information regarding the transactions can be accessed through the following links:

Network	Renounce Transaction Hash
ETH	https://etherscan.io/tx/0xb71fa0987e718ace3b74346e32bce78ad36c36fb36347f44c9cee687c999dfdb
BSC	https://bscscan.com/tx/0x5b5e5795c5963fdf5a8d4b830890c05fbffa322110f47c2c21cd358bb7db9183
MATIC	https://polygonscan.com/tx/0x5f7ab380a7fa3278798c6dbccf2519021af2517de2d627fb2f6ab75f794ffdc7
TRON	https://tronscan.org/#/transaction/1c92c395ba62a00cda72091f59c31c578d516718822b108b8023784eed9e4903

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