Cyberscope

Audit Report

Infinix Chain

January 2025

SHA256

7238311dc9b238fd697e57c42dcf4468c41db3f5bb15cba2cb08fafc1c64adbe

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
●	MLI	Missing License Identifier	Unresolved
●	ROF	Redundant Ownable Functionality	Unresolved
●	UDO	Unnecessary Decimals Override	Unresolved
●	UVL	Unspecified Versions of Libraries	Unresolved
●	L02	State Variables could be Declared Constant	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

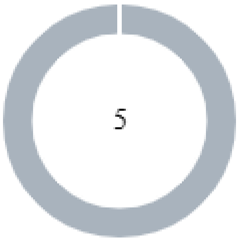
Audit Updates

Initial Audit	25 Jan 2025 https://github.com/cyberscope-io/audits/blob/main/3-fnx/v1/audit.pdf
Corrected Phase 2	28 Jan 2025

Source Files

Filename	SHA256
Infinixchain_Smart_Contract.sol	7238311dc9b238fd697e57c42dcf4468c41db3f5bb15cba2cb08fafc1c64adbe

Findings Breakdown



●	Critical	0
●	Medium	0
●	Minor / Informative	5

Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	5	0	0	0

MLI - Missing License Identifier

Criticality	Minor / Informative
Status	Unresolved

Description

The audited smart contract is missing an explicit SPDX license identifier, which is a crucial component for ensuring the legal clarity and compliance of the code. The license identifier specifies the terms under which the code can be used, modified, and redistributed, providing essential guidance to developers and end-users. Its absence creates ambiguity regarding the rights and obligations associated with the code, potentially leading to legal disputes or misuse.

Recommendation

The team is recommended to add an SPDX license identifier to the top of the smart contract to clearly specify the terms under which the code can be used, modified, and distributed.

ROF - Redundant Ownable Functionality

Criticality	Minor / Informative
Location	Infinixchain_Smart_Contract.sol#L6
Status	Unresolved

Description

The contract inherits from the Ownable abstract contract to define an owner. In smart contracts, an owner typically has elevated privileges to execute administrative functions. However, in this case, while the contract defines an owner, it does not include any administrative functionalities. Therefore, the inheritance of Ownable is redundant.

```
contract InfinixChain is ERC20, Ownable {
    constructor() ERC20("InfinixChain", "FNX")
Ownable(msg.sender) {/*...*/}
```

Recommendation

Eliminating redundancies will reduce code size and enhance readability. By removing the unnecessary inheritance, the contract becomes more efficient and aids in future maintainability.

UDO - Unnecessary Decimals Override

Criticality	Minor / Informative
Location	Infinixchain_Smart_Contract.sol#L13
Status	Unresolved

Description

The contract is currently implementing an override of the decimals function, which simply returns the value 18. This override is redundant since the extending token contract already specifies 18 decimals as its standard. In the context of ERC-20 tokens, 18 decimals is a common default, and overriding this function to return the same value adds unnecessary complexity to the contract. This redundancy does not contribute to the functionality of the contract and could potentially lead to confusion about the necessity of this override.

```
function decimals() public pure override returns (uint8) {
    return 18;
}
```

Recommendation

Since the inherited ERC-20 contract already defines the decimals number, maintaining an overriding function that merely repeats this value does not contribute to the contract's effectiveness. As a result, it is recommended to remove the redundant `decimals` function from the contract. Removing this function will simplify the contract, making it more straightforward to maintain without impacting its operational capabilities.

UVL - Unspecified Versions of Libraries

Criticality	Minor / Informative
Location	Infinixchain_Smart_Contract.sol#L1,2
Status	Unresolved

Description

The contract does not specify the versions of openzeppelin libraries that it is using.

```
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
```

Recommendation

It is recommended that the team specifies the version of the libraries that are used in the contract. Clear versioning ensures consistency, stability, and compatibility, reducing the likelihood of vulnerabilities or unexpected issues arising from library updates.

L02 - State Variables could be Declared Constant

Criticality	Minor / Informative
Location	Infinixchain_Smart_Contract.sol#L7
Status	Unresolved

Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
uint256 public initialSupply = 10_000_000_000 * 10**18
```

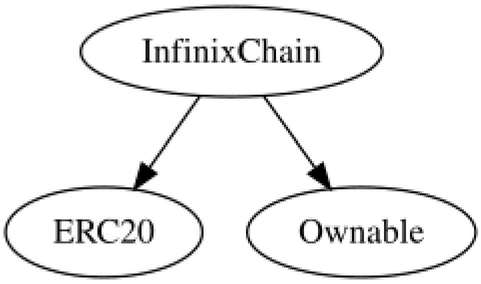
Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

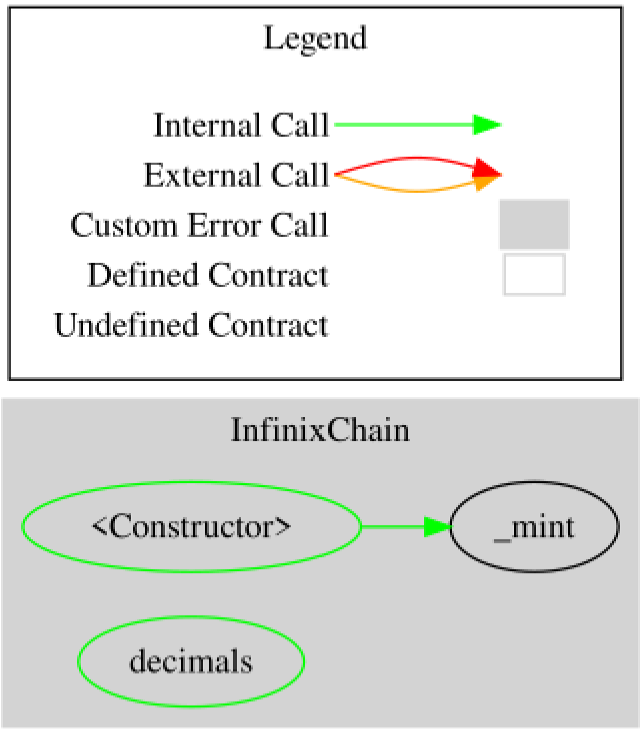
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
InfinixChain	Implementation	ERC20, Ownable		
		Public	✓	ERC20 Ownable
	decimals	Public		-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
InfinixChain
<Constructor>
_mint
decimals

Summary

Infinix Chain contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Infinix Chain 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 only access functions provided by the Ownable contract and does not have any additional functionalities. The contract does not implement any 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