慢雾科技
slow mist

Smart Contract Security Audit Report

The SlowMist Security Team received the GXChain team's application for smart contract security audit of the GXC Token on Aug 24, 2020. The following are the details and results of this smart contract security audit:

Token name :

GXC

The File Name and HASH(SHA256):

GXC.sol: 46fcd2aa104e6496ced40ab83275b0f6d75ba1f1e935aa2dd16d6fd17272eccf

The audit items and results :

(Other unknown security vulnerabilities are not included in the audit responsibility scope)

No.	Audit Items	Audit Subclass	Audit Subclass Result
1	Overflow Audit	-	Passed
2	Race Conditions Audit	-	Passed
3	Authority Control Audit	Permission vulnerability audit	Passed
		Excessive auditing authority	Passed
4	Safety Design Audit	Zeppelin module safe use	Passed
		Compiler version security	Passed
		Hard-coded address security	Passed
		Fallback function safe use	Passed
		Show coding security	Passed
		Function return value security	Passed
		Call function security	Passed
5	Denial of Service Audit	-	Passed
6	Gas Optimization Audit	-	Passed
7	Design Logic Audit	-	Passed
8	“False Deposit” vulnerability Audit	-	Passed
9	Malicious Event Log Audit	-	Passed
10	Scoping and Declarations Audit	-	Passed

11	Replay Attack Audit	ECDSA's Signature Replay Audit	Passed
12	Uninitialized Storage Pointers Audit	-	Passed
13	Arithmetic Accuracy Deviation Audit	-	Passed

Audit Result : **Passed**

Audit Number : 0X002008310004

Audit Date : Aug 31, 2020

Audit Team : SlowMist Security Team

Summary: This is a token contract that does not contain the tokenVault section. The comprehensive evaluation contract is no risk.

The source code:

```
// SPDX-License-Identifier: MIT
pragma solidity 0.6.2;
pragma experimental ABIEncoderV2;
import "@openzeppelin/contracts/presets/ERC20PresetMinterPauser.sol";
contract GXC is ERC20PresetMinterPauser {
    bytes32 public constant ADJUST_ROLE = keccak256("ADJUST_ROLE");
    bytes32 public constant DELIVER_ROLE = keccak256("DELIVER_ROLE");

    string[100] private txidArray;
    uint256 arrayLength = 100;
    uint256 private id;

    uint256 private _minDeliver = 50000;
    uint256 private _minBurn = 50000;

    uint8 private decimals_ = 5;
    event Deliver(address indexed to, uint256 amount, string from, string txid);

    event Burn(address indexed from, uint256 amount, string to);

    constructor(string memory name, string memory symbol)
        public
        ERC20PresetMinterPauser(name, symbol)
    {
        super._setupDecimals(decimals_);
        _setupRole(ADJUST_ROLE, _msgSender());
    }


    function deliver(
        address to,
        uint256 amount,
        string memory from,
        string memory txid
    ) public {
        require(
            amount >= _minDeliver,
            "The minimum value must be greater than minDeliver"
        );
        require(hasRole(DELIVER_ROLE, _msgSender()), "Must have deliver role to deliver");
        //SlowMist// txidArray is used only to prevent multiple commits of the same transaction over
a short period of time and is not used to record the transaction record itself
        for (uint256 i = 0; i < arrayLength; i++) {
            require(
                keccak256(abi.encodePacked(txidArray[i])) !=
                    keccak256(abi.encodePacked(txid)),
                "The txid has existed"
            );
        }
        uint256 id_number = id % arrayLength;
        txidArray[id_number] = txid;
        id++;
        transfer(to, amount);
        emit Deliver(to, amount, from, txid);
    }

    function burn(uint256 amount, string memory to) public {
        require(
            amount >= _minBurn,
            "The minimum value must be greater than minBurn"
        );
        super.burn(amount);
        emit Burn(msg.sender, amount, to);
    }

    function adjustParams(uint256 minDeliver , uint256 minBurn)
        public
    {
        require(hasRole(ADJUST_ROLE, _msgSender()), "Adjust role required");
        _minDeliver = minDeliver;
        _minBurn = minBurn;
    }

    function getParams() public returns (uint256 ,uint256){
        return (_minDeliver, _minBurn);
    }

    function getTxids() public view returns (string[100] memory) {
        return txidArray;
    }
}
```

慢雾科技
slow mist

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

WeChat Official Account