



# 智能合约安全审计报告



慢雾安全团队于 2019-08-24 日，收到 GXChain 团队对 GXC 项目智能合约安全审计申请。如下为本次智能合约安全审计细节及结果：

Token 名称：

GXC

文件名及哈希（SHA256）：

GXC.sol: 873db0972c23682140f08f4015564404a1741673a30967d5a12be4b311817691

本次审计项及结果：

（其他未知安全漏洞不包含在本次审计责任范围）

序号	审计大类	审计子类	审计结果
1	溢出审计	-	通过
2	条件竞争审计	-	通过
3	权限控制审计	权限漏洞审计	通过
		权限过大审计	通过
4	安全设计审计	Zeppelin 模块使用安全	通过
		编译器版本安全	通过
		硬编码地址安全	通过
		Fallback 函数使用安全	通过
		显现编码安全	通过
		函数返回值安全	通过
		call 调用安全	通过
5	拒绝服务审计	-	通过
6	Gas 优化审计	-	通过
7	设计逻辑审计	-	通过
8	“假充值”漏洞审计	-	通过
9	恶意 Event 事件日志审计	-	通过
10	变量声明及作用域审计	-	通过

11	重放攻击审计	ECDSA 签名重放审计	通过
12	未初始化的存储指针	-	通过
13	算术精度误差	-	通过

备注：审计意见及建议见代码注释 //SlowMist//.....

审计结果：**通过**

审计编号：0X002008310004

审计日期：2019 年 08 月 31 日

审计团队：慢雾安全团队

(**声明**：慢雾仅就本报告出具前已经发生或存在的事实出具本报告，并就此承担相应责任。对于出具以后发生或存在的事实，慢雾无法判断其智能合约安全状况，亦不对此承担责任。本报告所作的安全审计分析及其他内容，仅基于信息提供者截至本报告出具时向慢雾提供的文件和资料（简称“已提供资料”）。慢雾假设：已提供资料不存在缺失、被篡改、删减或隐瞒的情形。如已提供资料信息缺失、被篡改、删减、隐瞒或反映的情况与实际情况不符的，慢雾对由此而导致的损失和不利影响不承担任何责任。慢雾仅对该项目的安全情况进行约定内的安全审计并出具了本报告，慢雾不对该项目背景及其他情况进行负责。)

**总结：此为代币(token)合约，不包含锁仓(tokenVault)部分。综合评估（合约无风险）。**

合约源代码如下：

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.21 <0.8.0;
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[10] private txidArray;
    uint256 arrayLength = 10;
    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 仅用于预防短时间内同一笔交易多次提交，不用于记录交易记录本身

    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 returns (string[10] memory) {  
    return txidArray;  
}  
}
```



官方网址

[www.slowmist.com](http://www.slowmist.com)

电子邮箱

[team@slowmist.com](mailto:team@slowmist.com)

微信公众号

