慢雾科技
slow mist

智能合约安全审计报告

目录

1 前言 3

2. 审计方法 3

3. 项目背景 4

3.1 项目介绍 4

4. 代码概述 5

4.1 合约可见性分析 5

4.2 合约信息 14

4.3 代码审计 14

4.3.1 低危漏洞 14

4.3.2 增强建议 16

5. 审计结果 18

5.1 结论 18

6. 声明 18

1. 前言

慢雾安全团队于 2021 年 09 月 04 日，收到 WePiggy 团队对 WePiggy 第四期代码进行安全审计的申请，根据项目特点慢雾安全团队制定如下审计方案。

慢雾安全团队将采用“白盒为主，黑灰为辅”的策略，以最贴近真实攻击的方式，对项目进行安全审计。

慢雾科技 DeFi 项目测试方法：

黑盒测试	站在外部从攻击者角度进行安全测试。
灰盒测试	通过脚本工具对代码模块进行安全测试，观察内部运行状态，挖掘弱点。
白盒测试	基于项目的源代码，进行脆弱性分析和漏洞挖掘。

慢雾科技 DeFi 漏洞风险等级：

严重漏洞	严重漏洞会对项目的安全造成重大影响，强烈建议修复严重漏洞。
高危漏洞	高危漏洞会影响项目的正常运行，强烈建议修复高危漏洞。
中危漏洞	中危漏洞会影响项目的运行，建议修复中危漏洞。
低危漏洞	低危漏洞可能在特定场景中会影响项目的业务操作，建议项目方自行评估和考虑这些问题是否需要修复。
弱点	理论上存在安全隐患，但工程上极难复现。
增强建议	编码或架构存在更好的实践方法。

2. 审计方法

慢雾安全团队智能合约安全审计流程包含两个步骤：

- ◆ 使用开源或内部自动化分析的工具对合约代码中常见的安全漏洞进行扫描和测试。
- ◆ 人工审计代码的安全问题，通过人工分析合约代码，发现代码中潜在的安全问题。

如下是合约代码审计过程中慢雾安全团队会重点审查的漏洞列表：

（其他未知安全漏洞不包含在本次审计责任范围）

- 重入攻击
- 重放攻击
- 重排攻击
- 短地址攻击
- 拒绝服务攻击
- 交易顺序依赖
- 条件竞争攻击
- 权限控制攻击
- 整数上溢/下溢攻击
- 时间戳依赖攻击
- Gas 使用，Gas 限制和循环
- 冗余的回调函数
- 不安全的接口使用
- 函数状态变量的显式可见性
- 逻辑缺陷
- 未声明的存储指针
- 算术精度误差
- tx.origin 身份验证
- 假充值漏洞
- 变量覆盖

3. 项目背景

3.1 项目介绍

WePiggy 是一个开源，非托管的加密资产借贷市场协议。在 WePiggy 的市场上，用户可存入特定的加密资产赚取利息，也可以支付一定的利息借取某种加密资产。

项目官网地址:

https://wepiggy.com

审计版本代码:

https://github.com/WePiggy/wepiggy-contracts-arbitrum/tree/de97e5325562c537660cccb669735cc63d3c3896

不包含如下合约：

- contracts/lens/WePiggyLensV2.sol
- contracts/lens/WePiggyLens.sol
- contracts/governance/SidechainFundingManager.sol
- contracts/governance/MainnetFundingManager.sol

4. 代码概述

4.1 合约可见性分析

在审计过程中，慢雾安全团队对核心合约的可见性进行分析，结果如下:

PiggyBreeder			
Function Name	Visibility	Mutability	Modifiers
poolLength	External	-	-
usersLength	External	-	-
setDevAddr	Public	Can modify state	-
setMigrator	Public	Can modify state	onlyOwner
setEnableClaimBlock	Public	Can modify state	onlyOwner
setReduceIntervalBlock	Public	Can modify state	onlyOwner
setAllocPoint	Public	Can modify state	onlyOwner
setReduceRate	Public	Can modify state	onlyOwner
setDevMiningRate	Public	Can modify state	onlyOwner
replaceMigrate	Public	Can modify state	onlyOwner
migrate	Public	Can modify state	onlyOwner
safePiggyTransfer	Internal	Can modify state	-
getPiggyPerBlock	Public	-	-
getMultiplier	Public	-	-
allPendingPiggy	External	-	-
pendingPiggy	External	-	-
_pending	Internal	-	-
massUpdatePools	Public	Can modify state	-
updatePool	Public	Can modify state	-
add	Public	Can modify state	onlyOwner
stake	Public	Can modify state	-

unStake	Public	Can modify state	-
claim	Public	Can modify state	-
emergencyWithdraw	Public	Can modify state	-

FundingHolder			
Function Name	Visibility	Mutability	Modifiers
transfer	Public	Can modify state	onlyOwner

FundingManager			
Function Name	Visibility	Mutability	Modifiers
safePiggyTransfer	Internal	Can modify state	-
addFunding	Public	Can modify state	onlyOwner
setFunding	Public	Can modify state	onlyOwner
getPendingBalance	Public	-	-
claim	Public	Can modify state	-

WePiggyToken			
Function Name	Visibility	Mutability	Modifiers
mint	Public	Can modify state	-
_transfer	Internal	Can modify state	-
delegates	External	-	-
delegate	External	Can modify state	-
delegateBySig	External	Can modify state	-
getCurrentVotes	External	-	-
getPriorVotes	External	-	-
_delegate	Internal	Can modify state	-
_moveDelegates	Internal	Can modify state	-
_writeCheckpoint	Internal	Can modify state	-
safe32	Internal	-	-
getChainId	Internal	-	-

Timelock			
Function Name	Visibility	Mutability	Modifiers
setDelay	Public	Can modify state	-
acceptAdmin	Public	Can modify state	-
setPendingAdmin	Public	Can modify state	-

queueTransaction	Public	Can modify state	-
cancelTransaction	Public	Can modify state	-
executeTransaction	Public	Payable	-
getBlockTimestamp	Internal	-	-
receive()	External	Payable	-

AToken2PTokenMigrator			
Function Name	Visibility	Mutability	Modifiers
migrate	Public	Can modify state	-
_getTokenBalance	Internal	Can modify state	-
Receive	External	Payable	-
compareStrings	Internal	-	-

ATokenMigrator			
Function Name	Visibility	Mutability	Modifiers
replaceMigrate	External	Payable	-
migrate	External	Payable	-
receive	External	Payable	-

CErc20Migrator			
Function Name	Visibility	Mutability	Modifiers
replaceMigrate	External	Can modify state	-
migrate	External	Can modify state	-

CEthMigrator			
Function Name	Visibility	Mutability	Modifiers
replaceMigrate	External	Payable	-
migrate	External	Payable	-
receive	External	Payable	-

Comptroller			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	initializer
enterMarkets	Public	Can modify state	-

addToMarketInternal	Internal	Can modify state	-
exitMarket	External	Can modify state	-
getAssetsIn	External	-	-
checkMembership	External	-	-
mintAllowed	External	Can modify state	-
mintVerify	External	Can modify state	-
redeemAllowed	External	Can modify state	-
redeemAllowedInternal	Internal	-	-
redeemVerify	External	Can modify state	-
borrowAllowed	External	Can modify state	-
borrowVerify	External	Can modify state	-
repayBorrowAllowed	External	Can modify state	-
repayBorrowVerify	External	Can modify state	-
liquidateBorrowAllowed	External	Can modify state	-
liquidateBorrowVerify	External	Can modify state	-
seizeAllowed	External	Can modify state	-
seizeVerify	External	Can modify state	-
transferAllowed	External	Can modify state	-
transferVerify	External	Can modify state	-
getAccountLiquidity	Public	-	-
getAccountLiquidityInternal	Internal	-	-
getHypotheticalAccountLiquidity	Public	-	-
getHypotheticalAccountLiquidityInternal	Internal	-	-
liquidateCalculateSeizeTokens	External	-	-
_setPriceOracle	Public	Can modify state	onlyOwner
_setCloseFactor	External	Can modify state	onlyOwner
_setCollateralFactor	External	Can modify state	onlyOwner
_setMaxAssets	External	Can modify state	onlyOwner
_setLiquidationIncentive	External	Can modify state	onlyOwner
_supportMarket	External	Can modify state	onlyOwner
_addMarketInternal	Internal	Can modify state	onlyOwner
_setMarketBorrowCaps	External	Can modify state	-
_setBorrowCapGuardian	External	Can modify state	onlyOwner
_setPauseGuardian	Public	Can modify state	onlyOwner
_setMintPaused	Public	Can modify state	-
_setBorrowPaused	Public	Can modify state	-
_setTransferPaused	Public	Can modify state	-
_setSeizePaused	Public	Can modify state	-
_setDistributeWpcPaused	Public	Can modify state	-

_setPiggyDistribution	Public	Can modify state	onlyOwner
getAllMarkets	Public	-	-
isMarketMinted	Public	-	-
isMarketListed	Public	-	-
_setMarketMinted	Public	Can modify state	-
_setMarketMintCaps	external	Can modify state	onlyOwner

SimplePriceOracle			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	initializer
getUnderlyingPrice	Public	-	-
setUnderlyingPrice	Public	Can modify state	onlyOwner
setPrice	Public	Can modify state	onlyOwner
getPrice	External	-	-
get	External	-	-
compareStrings	Internal	-	-

WePiggyPriceOracleV1			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	initializer
getPrice	External	-	-
setPrice	External	Can modify state	onlyOwner
setTokenConfig	Public	Can modify state	onlyOwner

WePiggyPriceProviderV1			
Function Name	Visibility	Mutability	Modifiers
getUnderlyingPrice	External	-	-
_getUnderlyingPriceInternal	Internal	-	-
_getCustomerPriceInternal	Internal	-	-
_getCompoundPriceInternal	Internal	-	-
_getChainlinkPriceInternal	Internal	-	-
addTokenConfig	Public	Can modify state	onlyOwner
addOrUpdateTokenConfigSource	Public	Can modify state	onlyOwner
updateTokenConfigBaseUnit	Public	Can modify state	onlyOwner
updateTokenConfigFixedUsd	Public	Can modify state	onlyOwner
getOracleSourcePrice	Public	-	-
compareStrings	Internal	-	-

oracleLength	Public	-	-

PiggyDistribution

Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can Modify State	initializer
distributeMintWpc	Public	Can Modify State	-
distributeRedeemWpc	Public	Can Modify State	-
distributeBorrowWpc	Public	Can Modify State	-
distributeRepayBorrowWpc	Public	Can Modify State	-
distributeSeizeWpc	Public	Can Modify State	-
distributeTransferWpc	Public	Can Modify State	-
_stakeTokenToPiggyBreeder	Public	Can Modify State	onlyOwner
_claimWpcFromPiggyBreeder	Public	Can Modify State	onlyOwner
setWpcSpeedInternal	Internal	Can Modify State	-
updateWpcSupplyIndex	Internal	Can Modify State	-
updateWpcBorrowIndex	Internal	Can Modify State	-
distributeSupplierWpc	Internal	Can Modify State	-
distributeBorrowerWpc	Internal	Can Modify State	-
grantWpcInternal	Internal	Can Modify State	-
claimWpc	Public	Can Modify State	-
claimWpc	Public	Can Modify State	-
claimWpc	Public	Can Modify State	-
_setWpcSpeed	Public	Can Modify State	onlyOwner
_setEnableWpcClaim	Public	Can Modify State	onlyOwner
_setEnableDistributeMintWpc	Public	Can Modify State	onlyOwner
_setEnableDistributeRedeemWpc	Public	Can Modify State	onlyOwner
_setEnableDistributeBorrowWpc	Public	Can Modify State	onlyOwner
_setEnableDistributeRepayBorrowWpc	Public	Can Modify State	onlyOwner
_setEnableDistributeSeizeWpc	Public	Can Modify State	onlyOwner
_setEnableDistributeTransferWpc	Public	Can Modify State	onlyOwner
_setEnableAll	Public	Can Modify State	onlyOwner
_transferWpc	Public	Can Modify State	onlyOwner
_transferToken	Public	Can Modify State	onlyOwner
pendingWpcAccrued	Public	-	-
pendingWpcInternal	Internal	-	-
pendingWpcBorrowInternal	Internal	-	-
pendingWpcBorrowIndex	Internal	-	-
pendingWpcSupplyInternal	Internal	-	-

pendingWpcSupplyIndex	Internal	-	-

BaseJumpRateModel			
Function Name	Visibility	Mutability	Modifiers
updateJumpRateModel	External	Can modify state	-
utilizationRate	Public	-	-
getBorrowRateInternal	Internal	-	-
getBorrowRate	External	-	-
getSupplyRateInternal	Internal	-	-
getSupplyRate	External	-	-
updateJumpRateModelInternal	Internal	Can modify state	onlyOwner

DAIInterestRateModel			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	initializer
updateDAIJumpRateModel	External	Can modify state	-
getSupplyRate	External	-	-
dsrPerBlock	Public	-	-
poke	Public	Can modify state	-

JumpRateModel			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	initializer

PERC20			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	initializer
mint	External	Can modify state	-
mintForMigrate	External	Can modify state	-
redeem	External	Can modify state	-
redeemUnderlying	External	Can modify state	-
borrow	External	Can modify state	-
repayBorrow	External	Can modify state	-
repayBorrowBehalf	External	Can modify state	-
liquidateBorrow	External	Can modify state	-
_addReserves	External	Can modify state	-

getCashPrior	Internal	-	-
doTransferIn	Internal	Can modify state	-
doTransferOut	Internal	Can modify state	-
flashloan	external	Can modify state	nonReentrant
_setFlashloan	public	Can modify state	onlyOwner

PEther			
Function Name	Visibility	Mutability	Modifiers
initialize	Public	Can modify state	initializer
mint	External	Payable	-
mintForMigrate	External	Payable	-
redeem	External	Can modify state	-
redeemUnderlying	External	Can modify state	-
borrow	External	Can modify state	-
repayBorrow	External	Payable	-
repayBorrowBehalf	External	Payable	-
liquidateBorrow	External	Payable	-
	External	Payable	-
getCashPrior	Internal	-	-
doTransferIn	Internal	Can modify state	-
doTransferOut	Internal	Can modify state	-
require-Error	Internal	-	-
flashloan	external	Can modify state	nonReentrant
_setFlashloan	public	Can modify state	onlyOwner

PToken			
Function Name	Visibility	Mutability	Modifiers
init	Public	Can modify state	onlyOwner
transferTokens	Internal	Can modify state	-
transfer	External	Can modify state	nonReentrant
transferFrom	External	Can modify state	nonReentrant
approve	External	Can modify state	-
allowance	External	-	-
balanceOf	External	-	-
balanceOfUnderlying	External	Can modify state	-
getAccountSnapshot	External	-	-
getBlockNumber	Internal	-	-
borrowRatePerBlock	External	-	-

supplyRatePerBlock	External	-	-
totalBorrowsCurrent	External	Can modify state	nonReentrant
borrowBalanceCurrent	External	Can modify state	nonReentrant
borrowBalanceStored	Public	-	-
borrowBalanceStoredInternal	Internal	-	-
borrowInterestBalancePriorInternal	Internal	-	-
exchangeRateCurrent	Public	-	-
exchangeRateStored	Public	-	-
exchangeRateStoredInternal	Internal	-	-
getCash	External	-	-
accrueInterestSnapshot	Public	-	-
accrueInterest	Public	Can modify state	-
mintInternal	Internal	Can modify state	nonReentrant
mintInternalForMigrate	Internal	Can modify state	nonReentrant
mintFresh	Internal	Can modify state	-
redeemInternal	Internal	Can modify state	nonReentrant
redeemUnderlyingInternal	Internal	Can modify state	nonReentrant
redeemFresh	Internal	Can modify state	-
borrowInternal	Internal	Can modify state	nonReentrant
borrowFresh	Internal	Can modify state	-
repayBorrowInternal	Internal	Can modify state	nonReentrant
repayBorrowBehalfInternal	Internal	Can modify state	nonReentrant
repayBorrowFresh	Internal	Can modify state	-
liquidateBorrowInternal	Internal	Can modify state	nonReentrant
liquidateBorrowFresh	Internal	Can modify state	-
seize	External	Can modify state	nonReentrant
seizeInternal	Internal	Can modify state	-
_setComptroller	Public	Can modify state	onlyOwner
_setReserveFactor	External	Can modify state	nonReentrant
_setReserveFactorFresh	Internal	Can modify state	onlyOwner
_addReservesInternal	Internal	Can modify state	nonReentrant
_addReservesFresh	Internal	Can modify state	-
_reduceReserves	External	Can modify state	nonReentrant
_reduceReservesFresh	Internal	Can modify state	onlyOwner
_setInterestRateModel	Public	Can modify state	-
_setInterestRateModelFresh	Internal	Can modify state	onlyOwner
_setMigrator	Public	Can modify state	onlyOwner
_setMinInterestAccumulated	Public	Can modify state	onlyOwner
getCashPrior	Internal	-	-

doTransferIn	Internal	Can modify state	-
doTransferOut	Internal	Can modify state	-

4.2 合约信息

合约	主网地址(Arbitrum 主网)
WPC	0x7a603D06007fc09f896Fb75644365AB091A7b91a
WE_PIGGY_PRICE_ORACLE_V1(proxy)	0x896aecb9E73Bf21C50855B7874729596d0e511CB
WE_PIGGY_PRICE_ORACLE_V1(Implementation)	0xa43bf6193a89d28edb529ab5ca9ad7506798f9f1
WP_PIGGY_PRICE_PROVIDER_ARB	0x04d2944394b70d6e56fcf1CaD3aa6b5a43Ec8A5C
STABLECOIN_JUMP_RATE_MODEL(Proxy)	0x5676Eb997C30140606965CeBd4CA829Ab89A6CaC
BTC_ETH_JUMP_RATE_MODEL(Proxy)	0x0944eB1060cBD8a7923b1e7b7a10a17603261D2C
MAINSTREAM_JUMP_RATE_MODEL(Proxy)	0x6d4D85C417aabdD2923165d5C66D92BA2eC56104
JUMP_RATE_MODEL(Implementation)	0x3157e0bbdc7e5dea0f4c33a0ad7211b9a4ff19ee
COMPTROLLER(Proxy)	0xaa87715E858b482931eB2f6f92E504571588390b
COMPTROLLER(Implementation)	0xf0fe1cb691c4153bbcf7ef03cd26e1d85848042a
PIGGY_DISTRIBUTION(Proxy)	0x77401FF895BDe043d40aae58F98de5698682c12a
PIGGY_DISTRIBUTION(Implementation)	0x9a9b2bf1d1c96332c55d0b6acb8c2b441381116d
MAX_IMILLION	0x417FDfC74503d8008AeEB53248E5C0f1960c2C1d
P_ETH(Proxy)	0x17933112E9780aBd0F27f2B7d9ddA9E840D43159
P_WBTC(Proxy)	0x3393cD223f59F32CC0cC845DE938472595cA48a1
P_USDC(Proxy)	0x2Bf852e22C92Fd790f4AE54A76536c8C4217786b
P_LINK(Proxy)	0x8F87c9c6Efe9CA6997d6FEC8BC930C1fEd90ccC7
P_USDT(Proxy)	0xB65Ab7e1c6c1Ba202baed82d6FB71975D56F007C
PToken(Implementation)	0x22f934a1bb68ea7e7893ef8f76249afe904af6ae

4.3 代码审计

4.3.1 低危漏洞

4.3.1.1 权限过大问题

获取价格的方式采用了 chainlink 的 Oracle，compound 的 Oracle，以及中心化的方式，通过 Token 的配置来决定要使用哪个方式获取价格，Owner 可以配置 Token 的取价方式，存在权限过大的风险。

SimplePriceOracle 合约中的 Owner 可以任意设置价格，存在权限过大的风险。

- contracts/oracle/SimplePriceOracle.sol

```
    function setUnderlyingPrice(PToken pToken, uint price) public onlyOwner {
        address asset = _pETHUnderlying;
        if (!compareStrings(pToken.symbol(), "pETH")) {
            asset = address(PERC20(address(pToken)).underlying());
        }
        uint bt = block.timestamp;
        data[asset] = Datum(bt, price);
        emit PricePosted(asset, data[asset].price, price, price, bt);
    }
    function setPrice(address asset, uint price) public onlyOwner {
        uint bt = block.timestamp;
        emit PricePosted(asset, data[asset].price, price, price, bt);
        data[asset] = Datum(bt, price);
    }
```

WePiggyPriceProviderForArb 的 Owner 可以添加，更新 Token 的配置，存在权限过大的风险。

- contracts/oracle/WePiggyPriceProviderForArb.sol.sol

```
function addTokenConfig(address pToken, address underlying, string memory underlyingSymbol, uint256 baseUnit, bool
fixedUsd,
        address[] memory sources, PriceOracleType[] calldata sourceTypes) public onlyOwner {
        require(sources.length == sourceTypes.length, "sourceTypes.length must equal than sources.length");
        // add TokenConfig
        TokenConfig storage tokenConfig = tokenConfigs[pToken];
        require(tokenConfig.pToken == address(0), "bad params");
        tokenConfig.pToken = pToken;
        tokenConfig.underlying = underlying;
        tokenConfig.underlyingSymbol = underlyingSymbol;
        tokenConfig.baseUnit = baseUnit;
        tokenConfig.fixedUsd = fixedUsd;
        // add priceOracles
        require(oracles[pToken].length < 1, "bad params");
        for (uint i = 0; i < sources.length; i++) {
            PriceOracle[] storage list = oracles[pToken];
            list.push(PriceOracle({
            source : sources[i],
            sourceType : sourceTypes[i]
            }));
        }
        emit ConfigUpdated(pToken, underlying, underlyingSymbol, baseUnit, fixedUsd);
        emit PriceOracleUpdated(pToken, oracles[pToken]);
    }
```

```
function addOrUpdateTokenConfigSource(address pToken, uint256 index, address source, PriceOracleType _sourceType) public
onlyOwner {
        PriceOracle[] storage list = oracles[pToken];
        if (list.length > index) {//will update
            PriceOracle storage oracle = list[index];
            oracle.source = source;
            oracle.sourceType = _sourceType;
        } else {//will add
            list.push(PriceOracle({
            source : source,
            sourceType : _sourceType
            }));
        }
    }
function updateTokenConfigBaseUnit(address pToken, uint256 baseUnit) public onlyOwner {
        TokenConfig storage tokenConfig = tokenConfigs[pToken];
        require(tokenConfig.pToken != address(0), "bad params");
        tokenConfig.baseUnit = baseUnit;
        emit ConfigUpdated(pToken, tokenConfig.underlying, tokenConfig.underlyingSymbol, baseUnit, tokenConfig.fixedUsd);
    }
function updateTokenConfigFixedUsd(address pToken, bool fixedUsd) public onlyOwner {
        TokenConfig storage tokenConfig = tokenConfigs[pToken];
        require(tokenConfig.pToken != address(0), "bad params");
        tokenConfig.fixedUsd = fixedUsd;
        emit ConfigUpdated(pToken, tokenConfig.underlying, tokenConfig.underlyingSymbol, tokenConfig.baseUnit, fixedUsd);
    }
```

同理 PiggyDistribution，PToken 和 Comptroller 的 Owner 权限没有设置为 timelock 合约，建议将 PiggyDistribution，PToken 和 Comptroller 的 Owner 权限设置为 timelock 合约。

修复状态：暂未修复。

4.3.2 增强建议

4.3.2.1 签名重放问题

delegateBySig 函数 nonce 是由用户自己传入的参数进行签名的，当用户传了一个较大的 nonce 时，当前交易无法通过校验但是相关的签名数据仍会留在链上，导致此签名可能在未来某个时间段可用。建议参考

eip-2612 进行修复。

参考：https://github.com/ethereum/EIPs/blob/master/EIPS/eip-2612.md#implementation

- wepiggy-contracts/contracts/token/WePiggyToken.sol

```
function delegateBySig(
        address delegatee,
        uint nonce,
        uint expiry,
        uint8 v,
        bytes32 r,
        bytes32 s
    )
    external
    {
        bytes32 domainSeparator = keccak256(
            abi.encode(
                DOMAIN_TYPEHASH,
                keccak256(bytes(name())),
                getChainId(),
                address(this)
            )
        );

        bytes32 structHash = keccak256(
            abi.encode(
                DELEGATION_TYPEHASH,
                delegatee,
                nonce,
                expiry
            )
        );

        bytes32 digest = keccak256(
            abi.encodePacked(
                "\x19\x01",
                domainSeparator,
                structHash
            )
        );

        address signatory = ecrecover(digest, v, r, s);
        require(signatory != address(0), "WePiggyToken::delegateBySig: invalid signature");
        require(nonce == nonces[signatory]++, "WePiggyToken::delegateBySig: invalid nonce");
        require(now <= expiry, "WePiggyToken::delegateBySig: signature expired");
        return _delegate(signatory, delegatee);
```

修复状态: 这个问题由于不直接影响项目的安全性，属于增强点，在保证签名的 nonce 准确的情况不会有该问题，因此暂时忽略。

5. 审计结果

5.1 结论

审计结果: 低风险

审计编号: 0X002109050001

审计日期: 2021 年 09 月 05 日

审计团队: 慢雾安全团队

总结: 慢雾安全团队采用人工结合内部工具对代码进行分析，审计期间发现了 1 个低危漏洞，1 个增强建议。Owner 权限过大的问题经过沟通将后续将通过 timelock 机制进行缓解，目前 Owner 采用了多签合约进行管理，还未将权限移交给 timelock 合约，有 1 个增强建议暂时被忽略。

6. 声明

厦门慢雾科技有限公司(下文简称 ”慢雾 ”) 仅就本报告出具前项目方已经发生或存在的事实出具本报告，并就此承担相应责任。对于出具以后项目方发生或存在的未知漏洞及安全事件，慢雾无法判断其安全状况，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。

慢雾科技
slow mist

官方网址

www.slowmist.com

电子邮箱

team@slowmist.com

微信公众号