智能合约安全审计报告

[2021]

目录

1 前言

慢雾安全团队于2021.04.11，收到Hero团队对Three King Doms智能合约安全审计的申请，慢雾安全团队根据项目特点制定如下审计方案。

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。

慢雾科技项目测试方法：

测试方法	说明
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。

慢雾科技漏洞风险等级：

漏洞等级	说明
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:

（其他未知的安全漏洞及审计项不包含在本次审计责任范围）

- 重入漏洞
- 重放漏洞
- 重排漏洞
- 短地址漏洞
- 拒绝服务漏洞
- 交易顺序依赖漏洞
- 条件竞争漏洞
- 权限控制漏洞
- 整数上溢/下溢漏洞
- 时间戳依赖漏洞
- 未声明的存储指针漏洞
- 算术精度误差漏洞
- tx.origin身份验证漏洞
- 假充值漏洞
- 变量覆盖漏洞
- Gas优化审计
- 恶意 Event 事件审计
- 冗余的回调函数
- 不安全的外部调用审计

- 函数状态变量可见性审计
- 业务逻辑缺陷审计
- 变量声明及作用域审计

3 项目概要

3.1 项目介绍

NFT-STK.com是NFT-HERO孵化发行的GameFi应用，由“集卡挖矿”、“质押抽卡”、“抽卡商店”、“卡牌交易”、“占领抢矿”等多个模块组成。游戏规则为：通过收集三国主题的NFT卡牌获得战力（不同的卡牌组合策略下，战力会不同），根据战力获得STK（代币全称：Super Three Kingdoms）代币奖励。NFT-STK.com现已上线HECO火币生态链，可以通过登录支持HECO的钱包（如TokenPocket、CodeBank、火币钱包等）参与NFT-STK.com。

项目网站：

https://www.nft-stk.com/

审计版本代码：

ThreeKingdoms202104030105.zip(SHA256):

d2e5c97dd32c0804ad5c1e6070be88a4e5764949ebd70615b3238f8524fc78ba

修复版本代码：

SuperThreeKingdoms.zip(SHA256)

df06aefcea4813fdfddfa656cf658e919dfeaa62ee67fdf0cc0d6abd995363d5

3.2 漏洞信息

如下是本次审计发现的漏洞及漏洞的修复状态信息：

NO	标题	漏洞类型	漏洞等级	漏洞状态
N1	随机数操控风险	时间戳依赖攻击	低	已修复
N2	事件记录缺失	其它	建议	已忽略
N3	权限过大问题	权限控制攻击	低	已确认

4 审计详情

4.1 合约基础信息

如下是合约主网地址：

目前代码还未部署到主网。

4.2 函数可见性分析

在审计过程中，慢雾安全团队对核心合约的函数可见性进行分析，结果如下:

ERC20			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-

ERC20			
mint	External	Can Modify State	onlyContractOwner
_burn	Private	Can Modify State	-
burn	External	Can Modify State	-
burnFrom	External	Can Modify State	-
_transfer	Private	Can Modify State	-
transfer	External	Can Modify State	-
transferFrom	External	Can Modify State	-
approve	External	Can Modify State	-

Hero			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC721
setBurnLockDuration	External	Can Modify State	onlyPermit
burn	Public	Can Modify State	-

Card			
Function Name	Visibility	Mutability	Modifiers
mint	External	Can Modify State	checkAddressMap
burn	Public	Can Modify State	-
batchBurn	External	Can Modify State	-

ERC721Ex			
Function Name	Visibility	Mutability	Modifiers
_mint	Internal	Can Modify State	-
_burn	Internal	Can Modify State	-
safeBatchTransferFrom	External	Can Modify State	-
safeBatchTransferFrom	Public	Can Modify State	-
batchTransferFrom	Public	Can Modify State	-
setUriPrefix	External	Can Modify State	onlyPermit
tokenURI	External	-	-

ERC721			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
balanceOf	External	-	-
tokensOf	External	-	-
ownerOf	External	-	-
safeTransferFrom	External	Payable	-
safeTransferFrom	Public	Payable	-
transferFrom	External	Payable	-
_transferFrom	Internal	Can Modify State	-
_removeTokenFrom	Internal	Can Modify State	-

ERC721			
_addTokenTo	Internal	Can Modify State	-
approve	External	Payable	-
setApprovalForAll	External	Can Modify State	-
getApproved	External	-	-
isApprovedForAll	External	-	-
supportsInterface	External	-	-

Member			
Function Name	Visibility	Mutability	Modifiers
setManager	External	Can Modify State	onlyContractOwner

AddressMap			
Function Name	Visibility	Mutability	Modifiers
_addAddressMap	Internal	Can Modify State	-
addAddressMap	Public	Can Modify State	onlyContractOwner
_removeAddressMap	Internal	Can Modify State	-
removeAddressMap	Public	Can Modify State	onlyContractOwner
removeAddressMapAll	Public	Can Modify State	onlyContractOwner
getAddressMapLength	Public	-	-
getAddressMaps	Public	-	-
containsAddressMap	Public	-	-

AddressMap			
requireAddressMap	Public	-	-

ContractOwner			
Function Name	Visibility	Mutability	Modifiers

Manager			
Function Name	Visibility	Mutability	Modifiers
setMember	External	Can Modify State	onlyContractOwner
addPermit	External	Can Modify State	onlyContractOwner
removePermit	External	Can Modify State	onlyContractOwner
removePermitAll	External	Can Modify State	onlyContractOwner
getPermitLength	External	-	-
getPermitMaps	External	-	-
containsPermit	Public	-	-
requirePermit	Public	-	-
getTimestamp	External	-	-

MoneyTransfer			
Function Name	Visibility	Mutability	Modifiers
_transferMoney	Internal	Can Modify State	-
_receiveMoneyFrom	Internal	Can Modify State	-

MoneyTransfer			
_receiveMoney	Internal	Can Modify State	-
_receiveMoneyExceed	Internal	Can Modify State	-
_sendMoneyTo	Internal	Can Modify State	-
_sendMoney	Internal	Can Modify State	-

HeroPackage			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC721
mint	External	Can Modify State	onlyShop

Package			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
addShop	External	Can Modify State	onlyPermit
setShopTarget	External	Can Modify State	onlyPermit
setShopEnable	External	Can Modify State	onlyPermit
getShopLength	External	-	-
getShops	External	-	-
open	External	Can Modify State	-
tokenURI	External	-	-

Shop			
Function Name	Visibility	Mutability	Modifiers
setQuantityLimit	External	Can Modify State	onlyPermit
getQuantitiyLimit	External	-	-
getRarities	External	-	-
calcRarity	Public	-	-
addRarity	External	Can Modify State	onlyPermit
setRarity	External	Can Modify State	onlyPermit
removeRarity	External	Can Modify State	onlyPermit
getRarityWeights	Public	-	-
getStarWeights	Public	-	-
onOpenPackage	External	Can Modify State	-

Store			
Function Name	Visibility	Mutability	Modifiers
<Receive Ether>	External	Payable	-
setCodeAddress	External	Can Modify State	-

HeroUtil			
Function Name	Visibility	Mutability	Modifiers
setHeroConfig	External	Can Modify State	onlyPermit
setRarityBuffer	External	Can Modify State	onlyPermit

HeroUtil			
setStarBuffer	External	Can Modify State	onlyPermit
getHeroConfig	External	-	-
getFight	External	-	-
setExp	External	Can Modify State	onlyPermit
getExp	External	-	-
setRarityMax	External	Can Modify State	onlyPermit
setStarMax	External	Can Modify State	onlyPermit

ScoreBoard			
Function Name	Visibility	Mutability	Modifiers
getScores	External	-	-
_addScore	Internal	Can Modify State	-
_subScore	Internal	Can Modify State	-
addScore	Public	Can Modify State	checkAddressMap
subScore	Public	Can Modify State	checkAddressMap
addScores	External	Can Modify State	useTimes
subScores	External	Can Modify State	useTimes

Slot			
Function Name	Visibility	Mutability	Modifiers
setHeroLevelConfigs	External	Can Modify State	onlyPermit

Slot			
setCountryHeroMax	External	Can Modify State	onlyPermit
setCountryBuffer	External	Can Modify State	onlyPermit
setCountryChangeCost	External	Can Modify State	onlyPermit
setMineralInfo	External	Can Modify State	onlyPermit
getUserInfo	External	-	-
calcCountryChangedFight	External	-	-
onERC721Received	External	Can Modify State	-
onERC721ExReceived	External	Can Modify State	-
_onFightChanged	Internal	Can Modify State	-
_calcHeroFight	Internal	Can Modify State	-
_addHeroes	Internal	Can Modify State	-
removeHeroes	External	Can Modify State	-
removeHeroesAll	External	Can Modify State	-
_upgradeHeroes	Internal	Can Modify State	-
changeCountry	External	Can Modify State	-
useFight	External	Can Modify State	checkAddressMap
returnFight	External	Can Modify State	-
updateFight	External	Can Modify State	-
updateFight	External	Can Modify State	onlyPermit
_updateFight	Internal	Can Modify State	-

Slot			
withdrawReward	External	Can Modify State	-
withdrawRewardAll	External	Can Modify State	-
withdrawRelease	External	Can Modify State	-

HeroShop			
Function Name	Visibility	Mutability	Modifiers
_buy	Internal	Can Modify State	increaseQuantity
stopShop	External	Can Modify State	onlyPermit
getRarityWeights	Public	-	-
onOpenPackage	External	Can Modify State	-

HeroShopDrawLots			
Function Name	Visibility	Mutability	Modifiers
setMoneyIn	External	Can Modify State	onlyPermit
setTokenAmount	External	Can Modify State	onlyPermit
setConditions	External	Can Modify State	onlyPermit
getConditions	External	-	-
_checkLotBuy	Internal	-	-
_checkLotDraw	Internal	-	-
buyLot	External	Can Modify State	-
drawLot	External	Can Modify State	onlyPermit

HeroShopDrawLots			
pickLotCost	External	Can Modify State	-
pickLotLuck	External	Can Modify State	-

HeroShopMoney			
Function Name	Visibility	Mutability	Modifiers
setMoneyIn	External	Can Modify State	onlyPermit
setTokenAmount	External	Can Modify State	onlyPermit
buy	External	Payable	-

HeroShopMoneyMineral			
Function Name	Visibility	Mutability	Modifiers
buy	Public	Can Modify State	checkStarRandomToken
setMoneyIn	External	Can Modify State	onlyPermit
setMakeUpTokenBeforeEnd	External	Can Modify State	onlyPermit
setMineralInfo	External	Can Modify State	onlyPermit
addMineralAmount	Public	Can Modify State	-
subMineralAmount	Public	Can Modify State	-

HeroShopRandom			
Function Name	Visibility	Mutability	Modifiers
setStarConfig	External	Can Modify State	onlyPermit

HeroShopRandom			
getStarConfigs	External	-	-
getStarWeights	Public	-	-

HeroShopStarRandom			
Function Name	Visibility	Mutability	Modifiers
setStarConfig	External	Can Modify State	onlyPermit
getStarConfigs	External	-	-
getStarWeights	Public	-	-

HeroShopToken			
Function Name	Visibility	Mutability	Modifiers
buy	External	Can Modify State	checkStarRandomToken

4.3 漏洞详情

[N1] [低] 随机数操控风险

漏洞类型: 时间戳依赖攻击

详细内容

随机数使用的是 `bytes32 bh = blockhash(block.number - 1);` 进行计算的，该_drawLot函数仅 Admin 可以调用，Admin和矿工可以通过回滚攻击操纵随机数。

- heroShop/HeroShopDrawLots.sol

```
function drawLot(uint256 quantity) external onlyPermit("Admin") {
        quantity = _drawLot(quantity);
```

```
        address cashier = manager.members("cashier");
        _sendMoneyTo(moneyIn, cashier, price * quantity);
    }
```

- part/DrawLots.sol

```
function _drawLot(uint256 quantity) internal returns(uint256) {
        uint256 _now = block.timestamp;
        require(_now > lotDrawStartTime && _now < lotPickStartTime, "not in time");

        bytes32 bh = blockhash(block.number - 1);
        uint256 count = 0;

        for (uint256 i = 0; i < quantity && lots.length > 0; ++i) {
            uint256 index = Util.randomUint(abi.encode(bh, i), 0, lots.length - 1);
            LotInfo storage lotInfo = lots[index];

            if (_checkLotDraw(lotInfo)) {
                lucks.push(lotInfo);

                lotUsers[lotInfo.user].luckCount++;

                count++;
            }

            lots[index] = lots[lots.length - 1];
            lots.pop();
        }

        return count;
    }
```

解决方案

建议采用链上预言机获取随机数，避免因为依赖链上区块信息来计算随机数导致随机数可被操控。

漏洞状态

已修复；经过与项目方的沟通讨论反馈，项目方通过使用下一个区块的区块hash来计算随机数，通过这种方式来提高Admin和矿工操纵随机数的成本。

[N2] [建议] 事件记录缺失

漏洞类型: 其它

详细内容

如下函数没有事件记录，不利于项目方对敏感操作的监控和社区用户的审查。

- Manager.sol

```
function setMember(string memory name, address member)
        external onlyContractOwner {

        members[name] = member;
    }

    function addPermit(string memory permit, address account)
        external onlyContractOwner {

        require(permits[permit].add(account), "account existed");
    }

    function removePermit(string memory permit, address account)
        external onlyContractOwner {

        require(permits[permit].remove(account), "account not existed");
    }

    function removePermitAll(string memory permit)
        external onlyContractOwner {

        delete permits[permit];
    }
```

- HeroUtil.sol

```
    function setHeroConfig(uint256 hero, HeroConfig memory config)
        external onlyPermit("Config") {

        heroConfigs[hero] = config;
    }

    function setRarityBuffer(uint256 rarity, uint256 buffer)
        external onlyPermit("Config") {

        rarityBuffers[rarity] = buffer;
    }

    function setStarBuffer(uint256 star, uint256 buffer)
        external onlyPermit("Config") {

        starBuffers[star] = buffer;
    }
```

- part/Package.sol

```
function addShop(address target) external onlyPermit("Config") {
        require(target != address(0), "zero address");
        require(shopIndexes[target] == 0, "shop existed");

        shopIndexes[target] = shops.length;
        shops.push(ShopInfo({
            target: target,
            enabled: true
        }));
    }

    function setShopTarget(uint256 index, address target)
        external onlyPermit("Config") {
```

```
        require(target != address(0), "zero address");
        require(shopIndexes[target] == 0, "shop existed");

        ShopInfo storage shopInfo = shops[index];

        shopIndexes[shopInfo.target] = 0;
        shopIndexes[target] = index;

        shopInfo.target = target;
    }

    function setShopEnable(uint256 index, bool enable)
        external onlyPermit("Config") {

        shops[index].enabled = enable;
    }
```

- part/DrawLots.sol

```
function setLotTime(uint256 buyStart, uint256 drawStart, uint256 pickStart)
        external onlyPermit("Config") {

        lotBuyStartTime = buyStart;
        lotDrawStartTime = drawStart;
        lotPickStartTime = pickStart;
    }

    function setLotQuantityMax(uint256 max)
        external onlyPermit("Config") {

        lotQuantityMax = max;
    }
```

解决方案

建议对上述敏感函数添加事件记录，便于项目方对敏感操作的监控和社区用户对敏感操作进行审查。

漏洞状态

已忽略

[N3] [低] 权限过大问题

漏洞类型: 权限控制攻击

详细内容

项目中使用Manager合约对项目的权限进行管理，Manager合约中的ContractOwner角色有权利修改所有的权限控制。

存在权限过大问题。

- Manager.sol

```
contract Manager is ContractOwner {
    using AddressSet for AddressSet.Set;

    mapping(string => address) public members;

    mapping(string => AddressSet.Set) internal permits;

    modifier onlyPermit(string memory permit) {
        require(permits[permit].contains(msg.sender), "no permit");
        _;
    }

    function setMember(string memory name, address member)
        external onlyContractOwner {

        members[name] = member;
    }

    function addPermit(string memory permit, address account)
        external onlyContractOwner {

        require(permits[permit].add(account), "account existed");
    }
```

```
    function removePermit(string memory permit, address account)
        external onlyContractOwner {

        require(permits[permit].remove(account), "account not existed");
    }

    function removePermitAll(string memory permit)
        external onlyContractOwner {

        delete permits[permit];
    }

    function getPermitLength(string memory permit) external view returns(uint256) {
        return permits[permit].length();
    }

    // [startIndex, endIndex)
    function getPermitMaps(string memory permit, uint256 startIndex, uint256 endIndex
)
        external view returns(address[] memory) {

        return permits[permit].get(startIndex, endIndex);
    }

    function containsPermit(string memory permit, address account)
        public view returns(bool) {

        return permits[permit].contains(account);
    }

    function requirePermit(string memory permit, address account) public view {
        require(containsPermit(permit, account), "not permit");
    }

    function getTimestamp() external view returns(uint256) {
        return block.timestamp;
    }
```

解决方案

建议将Manager合约中的ContractOwner角色设置为timelock合约或治理合约

漏洞状态

已确认；经过与项目方的沟通反馈，Manager合约用于做内部的权限管理，项目方不会授权给外部地址，由项目方对项目的权限进行整体的管理。

5 审计结果

审计编号	审计团队	审计日期	审计结果
0X002104200002	SlowMist Security Team	2021.04.11 - 2021.04.20	低风险

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 2 个低危漏洞， 1 个增强建议。其中 1 个低危漏洞已确认； 1 个增强建议暂时被忽略；其它所有漏洞均已修复。 目前代码还未部署到主网。

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。

官方网址

www.slowmist.com

电子邮箱

team@slowmist.com

微信公众号