TMG35S NFT Auction
审计报告

版本：1.0.0

报告编号：2022052400011019

Fairyproof 发布

2022 年 5 月 24 日

01. 介绍

本报告包含了Fairyproof对TMG35S团队NFT发行和拍卖项目代码进行审计的结果。

审计开始时间：

2022年5月23日

审计结束时间：

2022年5月24日

项目通证名：

TMG35S

审计代码文件及目录结构：

审计结束时的代码文件名及对应的SHA-256哈希值为：

```
Auction.sol:  0x89e99e4f40364f29a6e3ce5630573f08e7a38da018b988ac984f78faa579f987
TMG35S.sol :  0x1173493039467530aaa704cd169dbd448ea217bdd2855614a58cc630d7b79942
```

审计代码的文件名及目录结构为：

```
contracts/
├── Auction.sol
└── TMG35S.sol

0 directories, 2 files
```

本次审计的目的是为了审阅TMG35S项目基于Solidity语言编写的NFT发行和拍卖平台应用，发现潜在的安全隐患，研究其设计、架构，并试图找到可能存在的漏洞。

我们全面阅读了TMG35S团队提交的上述代码，并仔细审阅了上述代码中可能出现问题的方方面面，对上述合约代码给出了全面、综合的改进意见及评审结果。

本次审计仅针对授权方指定版本的代码、安装包及其它授权方提供的素材展开，其结论仅对相应版本的应用适用，一旦相关代码、配置、运营环境等发生变化，相应结论将不再适用。

一 免责声明

截至本报告发布之日，本报告所阐述的内容仅反映审计团队对当前所审计的代码安全进展及状况的理解。任何人在接触或使用与本报告相关的服务、产品、协议、平台、或任何物品时，自行承担一切可能产生的冲突、损失、利益及风险，本报告的审计团队概不负责。

本审计不涉及审计代码的编译器及任何超出代码编程语言的领域。如果所审计的代码为智能合约，则合约由引用链下信息或资源所导致的风险及责任不在本审计覆盖的范围之内。

本审计无法详尽查看每一个细节，也无法穷尽每一种可能，因此本报告的审计团队鼓励本项目的开发团队及任何相关利益方对所审计代码进行任何后续的测试及审计。

对任何第三方使用本报告中所提及或涉及的软件、源码、软件库、产品、服务、信息等一切事物所产生的冲突、损失、利益及风险，本审计团队不保证、不承诺也不承担任何责任。

本报告的内容、获取方式、使用以及任何其所涉及的服务或资源都不能作为任何形式的投资、税务、法律、监管及建议等的依据，也不产生相关的责任。

— 审计方式

审计TMG35S项目的源代码是为了能清晰地理解该项目的实现方式及运行原理。审计团队对项目代码进行了深入的研究、分析和测试，并收集了详尽的数据。审计团队会在本报告中会详细列举所发现的每个问题、问题所在的源码位置、问题产生的根源以及对问题的描述，并对问题给出相应的改进建议。

Fairyproof审计的流程如下：

1. 背景研究。Fairyproof团队会阅读项目介绍、白皮书、合约源码等一切TMG35S团队所提供的相关材料及信息，以确保Fairyproof团队理解项目的规模、范围及功能。
2. 自动化检测。此步骤主要用自动化工具扫描源码，找到常见的潜在漏洞。
3. 人工审阅项目代码。此步骤由工程师逐行阅读代码，找到潜在的漏洞。
4. 逻辑校对。此步骤审计工程师将对代码的理解与TMG35S团队提供的材料及信息相比较，检查代码的实现是否符合项目的定义及白皮书等信息中的描述。
5. 测试用例检测。此步骤包括三部分：
   i. 测试用例设计。审计工程师将根据前述步骤对项目背景的理解及合约代码的理解，针对项目可能的执行逻辑及方式设计测试用例。

   ii. 测试范围分析。该步骤会详细检查所设计的测试用例是否覆盖了合约代码的所有逻辑分支，并判断测试用例执行后，合约代码的逻辑是否能得到充分的执行及检查

   iii. 符号执行。该步骤将运行测试用例以测试代码所有可能的执行路径。
6. 优化审查。该流程将根据合约的应用场景、调用方式及业界最新的研究成果从可维护性、安全性及可操作性等方面审查项目代码。

— 报告结构

本报告列举的每个问题都被设置了一个安全级别，这些安全级别根据其对项目的影响及安全隐患的大小而定。我们对每个问题都给出了相应的改进建议。为了便于读者阅读，我们分别按主题内容和安全级别这两种方式罗列了所有的问题，并提出了全面增强安全性的建议。

— 引用文档

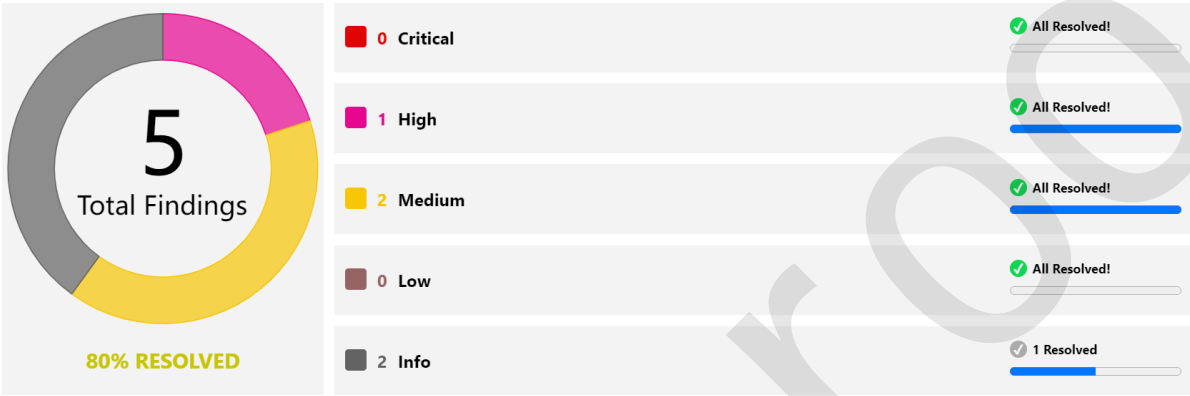
在审阅过程中，我们参考了与项目相关的文档以加深对项目逻辑、功能及应用的理解。本次报告参阅的文档资料如下：

https://tmg35s.d1verse.io/#/home

上述文档被视为本项目代码实现及功能的定义。当我们认为代码实现与文档定义有分歧时，我们及时咨询并与TMG35S团队进行了沟通和确认。

一 审计结果

报告编号	审计团队	审计时间	审计结果
2022052400011019	Fairyproof Security Team	2022.05.23 - 2022.05.24	低风险



审计结论：

Fairyproof团队综合使用了自动化工具和人工方式审计了本项目，在审计过程中发现1个高风险、2个中风险和2条信息。项目团队修复了1个高风险、2个中风险和1条信息，忽略了1条信息。

02. Fairyproof介绍

Fairyproof是一家领先的区块链技术公司，公司为行业企业提供安全审计和咨询方面的服务。Fairyproof研发了自己的一系列合约编写和安全审计标准，为众多客户提供了周到、严谨的服务。

03. 审计代码的主要功能

被审计代码的主要功能为NFT发行及拍卖平台。

NFT发行：

NFT名称：TMG35S

NFT符号：TMG35S

发行方式：项目方铸造

NFT拍卖平台：

用户可以在拍卖平台上进行上架，取消上架，竞拍操作。拍卖有时间限制，如果竞拍发生在最后十分钟之内，竞拍时间会自动延长十分钟。

竞拍结束时，最后一个竞拍者可以提取竞拍折NFT资产，平台从竞拍的价格中按比例提取一定手续费。

竞拍参数由拍卖者设定，其中竞拍资产可以为ETH资产或者ERC-20资产。

注意： 拍卖平台不支持转移过程中有损耗的通缩代币。

04. 风险种类

当前审计采用智能工具静态分析和人工审计相结合的方法，从以下多个风险种类方面对项目代码进行了全方位的审计。

- 重入攻击
- 重放攻击
- 重排攻击
- 矿工特权
- 回滚攻击
- 注入攻击
- 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
- 冗余的回调函数
- 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
- 参数检查
- 设计缺陷
- 潜在后门
- 通证发行

- 管理权限
- 代理升级
- 委托调用插槽共享
- 资金安全
- 迁移管理
- 代码优化
- 其它分类

05. 风险分级

本报告中的每个问题都被设置了一个安全等级，程度由高到低排列如下：

Critical/致命 风险及隐患需要立刻解决。

High/高危 风险及隐患将引发风险及问题，必须解决。

Medium/中 风险及隐患可能导致潜在风险，最终仍然需要解决。

Low/低 风险及隐患主要指各类处理不当或者会引发警告信息的细节，这类问题可以暂时搁置，但建议最终解决。

Informational/信息 一般不会引起风险问题，主要是代码的改进与优化。

06. 本审计关注的风险重点

根据所审计代码的功能及应用场景，我们着重审查了下列功能中可能潜藏的风险。

- 数值安全

我们检查了合约中的数值处理是否有算术溢出问题。常规的加减运算容易引起整数溢出，尤其是在处理通证金额或计算奖励金额时，本合约均使用了严谨的数学模块进行了处理。

经审查此功能暂未发现明显风险。

- 权限检查

我们检查了每一个能改变合约状态的函数是否具备合适的权限，重点检查那些必须管理员权限才能操作的函数。

经审查此功能暂未发现明显风险。

- 通证发行与交易

我们检查了通证发行与交易是否有不合规或者逻辑不正确的接口，以保护投资者的利益和系统的稳定运行。

经审查此功能发现一个中风险，不支持非标准的ERC-20转移。细节请参看“08. 问题详述”中的FP-3。

- 不安全的状态更改

我们检查了合约创建时初始化的一些关键状态变量。在很多情况下，这些变量只应该初始化一次，运行后再更改可能会给整个合约运行带来意料不到的风险。

经审查此功能暂未发现明显风险。

- 资金安全与后门

我们重点检查了入金和出金函数是否存在用户资金不受控制的风险、是否可能导致用户资金受损的问题。

经审查此功能发现了一个冗余的 `receive` 函数，细节请参看“08. 问题详述”中的FP-2。

- 其它

经审查其它功能发现了一个DDOS风险，细节请参看“08. 问题详述”中的FP-1。还发现了其它风险极低的问题，细节请参看“08. 问题详述”中的FP-4和FP-5。

07. 基于风险等级的问题列表

编号	标题	分类	等级	状态
FP-1	未采用提币模式	拒绝服务攻击	高风险	部分修复
FP-2	冗余的 `receive` 函数	资金安全	中风险	✓ 已解决
FP-3	未使用SafeTransfer	设计缺陷	中风险	✓ 已解决
FP-4	未验证零地址	参数检查	信息	✓ 已解决
FP-5	购买者可以是拍卖者	设计缺陷	信息	已忽略

08. 问题详述

[FP-1] 未采用提币模式 高风险 部分修复

风险分类：拒绝服务攻击

描述：

`Auction.sol` 中，当NFT的标价资产为ETH时，如果恶意合约参与竞拍，它可以不接受返还的ETH从而形成DDOS攻击，使得别人无法出更高的价格进行竞拍，最终以较低的价格获取拍卖品。

相关代码为：

```
if (collectible.currency == address(0)) {
      require(msg.value >= price, "Insufficient payment");
      _executeFundsTransfer(collectible.currency, lastBidder, paymentAmount);
} else {
    IERC20(collectible.currency).safeTransferFrom(_msgSender(), address(this),
price);
    IERC20(collectible.currency).safeTransferFrom(_msgSender(), lastBidder,
paymentAmount);
}
```

建议：

1、改为提币模式，用户自己提取以前质押的资产。

2、限制外部合约参与竞拍，降低拒绝服务攻击的风险

项目方反馈：

考虑到用户体验，项目方决定限制外部合约参与竞拍。在 `bid` 函数中增加了如下限制：

```
require(!Address.isContract(_msgSender()), "Invalid caller");
require(tx.origin == _msgSender(),"only eoa");
```

状态：

部分修复。考虑到竞拍时的标价资产是由用户设定的，在极端情况下（例如竞拍账号竞拍代币被锁定无法退还）仍然可能存在DDOS风险，但可能性已经大大降低，故综合评定为低风险。

[FP-2] 冗余的receive函数 中风险 ✓已解决

风险分类：资金安全

描述：

`Auction.sol`中，存在一个冗余的`receive`函数允许合约接收ETH，但合约业务逻辑中没有直接接收ETH的功能需求。当用户误向此合约地址发送ETH时，发送的ETH会永久丢失。

相关代码为：

```
receive() payable external {
}
```

建议：

移除该函数。

项目方反馈：

已经按照建议修改。

状态：

已修复。

[FP-3] 未使用SafeTransfer 中风险 ✓已解决

风险分类：设计缺陷

描述：

`Auction.sol`中，函数`_executeFundsTransfer`中，在发送ERC20资产时，未使用SafeTransfer进行发送，而是直接使用`transfer`。当代币为非标准的ERC20时，有可能无法发送代币造成资产死锁。

相关代码：

```
function _executeFundsTransfer(address currency, address to, uint256 amount)
private {
    if (currency == address(0)) {
        Address.sendValue(payable(to), amount);
    } else {
        IERC20(currency).transfer(to, amount);
    }
}
```

建议：

使用safeTransfer来替代transfer。

项目方反馈：

已经按照建议修改。

状态：

已修复。

[FP-4] 未验证零地址 信息 ✓ 已解决

风险分类：参数检查

描述：

`Auction.sol`中，`setPayee`函数中未验证`newPayee`参数是否为零地址，当误设置为零地址时，ETH手续费会丢失，ERC20代币手续费无法发送从而用户无法取回拍卖的产品。

建议：

增加`require(newPayee != address(0))`限制。

项目方反馈：

已经按照建议修改。

状态：

已修复。

[FP-5] 购买者可以是拍卖者 信息 已忽略

风险分类：设计缺陷

描述：

`Auction.sol`中，未对竞拍者是否拍卖者进行验证，用户可以自己竞拍自己的NFT产品。

建议：

增加此项验证。

项目方反馈：

项目方认为这不是个风险。

状态：

已忽略。

09. 增强建议

增加建议一般不涉及安全风险，通常是为了增加代码的健壮性和可读性。

1. 建议：增加一个函数用来获取每个NFT参与竞拍的人数，也就是`bidders`状态变量中每个NFT对应的竞拍者地址长度。

2. 在`withdraw`函数中，增加竞拍人数大于0的限制，否则调用时会数学溢出报错，不方便查看原因。

3. 事件`Bided`与`Withdrawn`中增加一个竞拍资产的参数，方便进行追踪。

4. 在拍卖ERC-1155资产时，如果竞拍的数量`collectible.amount`大于1，用户最终的出价为竞拍数量 * 竞拍价格，和ERC-721稍有区别，建议向用户说明。

5. 合约中存在两种管理员权限，均需妥善保管，权限丢失后可导致最终竞拍胜出者无法取回自己竞拍的NFT。

FAIRYPROOF

https://medium.com/@FairyproofT

https://twitter.com/FairyproofT

https://www.linkedin.com/company/fairyproof-tech

https://t.me/Fairyproof_tech

Reddit: https://www.reddit.com/user/FairyproofTech