



# Seraph 游戏分析报告

2023.11.24

Senna

DAMOCLES LABS

# 目录

- 概要(游戏安全性评分)
- 游戏背景
  - ◆ 游戏版本
  - ◆ 游戏类型&游戏引擎
  - ◆ 游戏玩法可能存在的问题
- 游戏安全分析
  - ◆ 游戏代码保护
  - ◆ 游戏基础反作弊
  - ◆ 游戏逻辑问题
  - ◆ 游戏协议分析
- Web3 安全分析
  - ◆ 代币合约安全
  - ◆ 游戏内经济系统安全
- 关于 Damocles

## 一、 概要（游戏安全性评分）

Seraph 于 2023 年 11 月 22 日开放三测。Damocles 团队于 11.24 号针对该游戏进行了安全分析与评估，但是评估结果不尽如人意。首先是项目方在代码中留存大量 Log 信息，并且可以从 Log 信息推断出项目方并非韩国团队，而是中国团队，且该游戏采用 Unity 加载 lua 的方式，并未对 Lua 代码进行保护，或者使用 lua jit 等提升逆向难度的手段进行源码保护，这就导致源码完全暴露，只需要 hook load 函数即可从内存中 dump 出游戏源码。但是该游戏属于 ARPG 游戏，该类游戏有天然的防作弊优势，即大部分数据均通过服务器同步，所以又在一定程度上缓解了游戏的安全问题。

安全评分：



## 二、 游戏背景

- 进行评估的游戏版本：v0.0.0.6
- 游戏类型&游戏引擎：ARPG，Unity
- 游戏玩法可能存在的问题：
  - 瞬移
  - 加速（加速移动，加速释放技能）
  - 自动挂机
  - 倍率修改
  - 无敌
  - Buff 修改（让角色可以持续存在增加灵魂晶石产出的 buff 或其它）

### 三、 游戏安全性分析

#### 游戏代码保护：

##### 分析过程：

1. 由于不同的引擎有不同的分析模式, 所以在获取到游戏 EXE 后首先需要确定游戏使用的引擎, 通过对游戏基础信息识别我们可以确定该游戏是使用 Unity 进行开发。

out	2023/11/29 16:22	文件夹	
out - 副本	2023/11/29 12:30	文件夹	
SeraphGamma_Data	2023/11/21 23:52	文件夹	
baselib.dll	2023/11/21 22:51	应用程序扩展	410 KB
GameAssembly.dll	2023/11/21 22:52	应用程序扩展	45,209 KB
out.zip	2023/11/25 18:26	WinRAR ZIP 压缩...	955 KB
SeraphGamma.exe	2023/11/21 22:51	应用程序	651 KB
UnityCrashHandler64.exe	2023/11/21 22:51	应用程序	1,089 KB
UnityPlayer.dll	2023/11/21 22:51	应用程序扩展	29,905 KB

2. 通过游戏目录中的 GameAssembly.dll 以及 global-metadata.dat 可以确定游戏采用的是 il2cpp 的编译模式, 于是通过 il2Cpdumper 进行源码还原。

Unity_Tools > Il2CppDumper-net6-v6.7.40 > seraph_output				在 seraph_outp
名称	修改日期	类型	大小	
DummyDll	2023/11/22 14:34	文件夹		
out	2023/11/24 1:18	文件夹		
dump.cs	2023/11/22 14:33	C# 源文件	21,784 KB	
il2cpp.h	2023/11/22 14:34	C Header 源文件	34,906 KB	
out.zip	2023/11/24 1:17	WinRAR ZIP 压缩...	955 KB	
script.json	2023/11/22 14:34	JSON 源文件	66,861 KB	
stringliteral.json	2023/11/22 14:34	JSON 源文件	1,135 KB	

```
F: > GameFi > Unity_Tools > Il2CppDumper-net6-v6.7.40 > seraph_output > dump.cs

1 // Image 0: mscorlib.dll - 0
2 // Image 1: UnityEngine.UIElementsModule.dll - 1901
3 // Image 2: System.dll - 2984
4 // Image 3: UnityEngine.CoreModule.dll - 3871
5 // Image 4: Unity.RenderPipelines.Universal.Runtime.dll - 4811
6 // Image 5: Unity.ThirdParty.dll - 5378
7 // Image 6: AstarPathfindingProject.dll - 5854
8 // Image 7: MagicaCloth.dll - 6256
9 // Image 8: com.rlabrecque.steamworks.net.dll - 6570
10 // Image 9: System.Core.dll - 7076
11 // Image 10: Unity.Mathematics.dll - 7702
12 // Image 11: ZFBrowser.dll - 7794
13 // Image 12: Unity.TextMeshPro.dll - 8563
14 // Image 13: RayFireAssembly.dll - 8690
15 // Image 14: UnityEngine.UI.dll - 8897
16 // Image 15: Unity.RenderPipelines.Core.Runtime.dll - 9104
17 // Image 16: UnityEngine.TextCoreTextEngineModule.dll - 9364
18 // Image 17: ThirdPartyPlugins.dll - 9432
19 // Image 18: Unity.VisualScripting.Core.dll - 9585
20 // Image 19: UnityEngine.IMGUIModule.dll - 9816
21 // Image 20: Pathfinding.Ionic.Zip.Reduced.dll - 9869
22 // Image 21: RuntimeInspector.Runtime.dll - 9956
23 // Image 22: Mono.Security.dll - 10065
24 // Image 23: Unity.Patch.dll - 10169
25 // Image 24: Unity.AutoLOD.dll - 10256
26 // Image 25: UnityEngine.ParticleSystemModule.dll - 10320
27 // Image 26: UnityEngine.PhysicsModule.dll - 10402
28 // Image 27: Unity.Collections.dll - 10448
29 // Image 28: UnityEngine.dll - 10599
30 // Image 29: Unity.Timeline.dll - 10600
31 // Image 30: UnityEngine.AnimationModule.dll - 10672
32 // Image 31: Whinarn.UnityMeshSimplifier.Runtime.dll - 10734
33 // Image 32: protobuf-net.dll - 10765
34 // Image 33: Assembly-CSharp.dll - 10803
35 // Image 34: UnityEngine.AndroidJNIModule.dll - 10871
36 // Image 35: UnityEngine.TerrainModule.dll - 10886
37 // Image 36: CSharp.dll - 10911
38 // Image 37: LitJson.dll - 10926
39 // Image 38: UnityEngine.PropertiesModule.dll - 10956
40 // Image 39: Pathfinding.ClipperLib.dll - 11043
41 // Image 40: UnityEngine.UnityWebRequestModule.dll - 11063
```

但是在 dump.cs 文件中并没有发现跟游戏相关性比较强的代码逻辑，于是猜

测该游戏并不是用 C#进行开发，而是通过 lua 进行加载的，于是通过代码

Hook 游戏 loadbuff 相关的函数，获取到了游戏真正的源码。

@AI_ai_utils.lua	2023/11/29 11:56	Lua 源文件	3 K
@AI_bt_action_nodes.lua	2023/11/29 11:56	Lua 源文件	39 K
@AI_bt_base_nodes.lua	2023/11/29 11:56	Lua 源文件	13 K
@AI_bt_condition_nodes.lua	2023/11/29 11:56	Lua 源文件	12 K
@AI_mgr_entity_aggro_cell.lua	2023/11/29 11:56	Lua 源文件	8 K
@AI_mgr_entity_ai_cell.lua	2023/11/29 11:56	Lua 源文件	202 K
@BT_bt94.lua	2023/11/29 12:06	Lua 源文件	3 K
@BT_bt20003.lua	2023/11/29 12:03	Lua 源文件	2 K
@BT_bt20004.lua	2023/11/29 12:06	Lua 源文件	1 K
@BT_bt21000.lua	2023/11/29 12:16	Lua 源文件	1 K
@BT_bt21015.lua	2023/11/29 12:15	Lua 源文件	1 K
@BT_bt23007.lua	2023/11/29 12:11	Lua 源文件	1 K
@BT_bt23020.lua	2023/11/29 12:08	Lua 源文件	2 K
@Core_Class.lua	2023/11/29 11:56	Lua 源文件	4 K
@Core_Entity.lua	2023/11/29 11:56	Lua 源文件	7 K
@Core_EventDispatcher.lua	2023/11/29 11:56	Lua 源文件	2 K
@Core_FrameTimerHeap.lua	2023/11/29 11:56	Lua 源文件	1 K
@Core_FunctionInvoker.lua	2023/11/29 11:56	Lua 源文件	1 K
@Core_GameWorld.lua	2023/11/29 11:56	Lua 源文件	52 K
@Core_LoggerHelper.lua	2023/11/29 11:56	Lua 源文件	3 K
@Core_LuaBehaviour.lua	2023/11/29 11:56	Lua 源文件	2 K
@Core_PrintTable.lua	2023/11/29 11:56	Lua 源文件	4 K
@Core_Queue.lua	2023/11/29 11:56	Lua 源文件	2 K
@Core_Stack.lua	2023/11/29 11:56	Lua 源文件	2 K
@Core_StringEx.lua	2023/11/29 11:56	Lua 源文件	4 K
@Core_TableEx.lua	2023/11/29 11:56	Lua 源文件	6 K
@Core_TimerHeap.lua	2023/11/29 11:56	Lua 源文件	4 K
@Core_XmlSimple.lua	2023/11/29 11:56	Lua 源文件	6 K
@CSFacade.lua	2023/11/29 11:56	Lua 源文件	25 K
@Dramas_drama_trigger_cg.lua	2023/11/29 11:56	Lua 源文件	1 K
@Dramas_drama_trigger_guide.lua	2023/11/29 11:56	Lua 源文件	3 K
@Entities_Avatar.lua	2023/11/29 16:19	Lua 源文件	16 K

\*/+\*-/

```
require "Core.GameWorld"
require "GameDataHelper.GameDataHelper"
require "GameManager.GameManager"
require "GameConfig.GameConfig"
require "GameUtil.GameUtil"
require "PlayerManager.PlayerManager"
require "CSFacade"
require "AI/bt_base_nodes"
require "AI/bt_action_nodes"
require "AI/bt_condition_nodes"

local Application = UnityEngine.Application
local OperationLogManager = GameManager.OperationLogManager
local SystemConfig = GameLoader.SystemConfig
local DataParseHelper = GameDataHelper.DataParseHelper

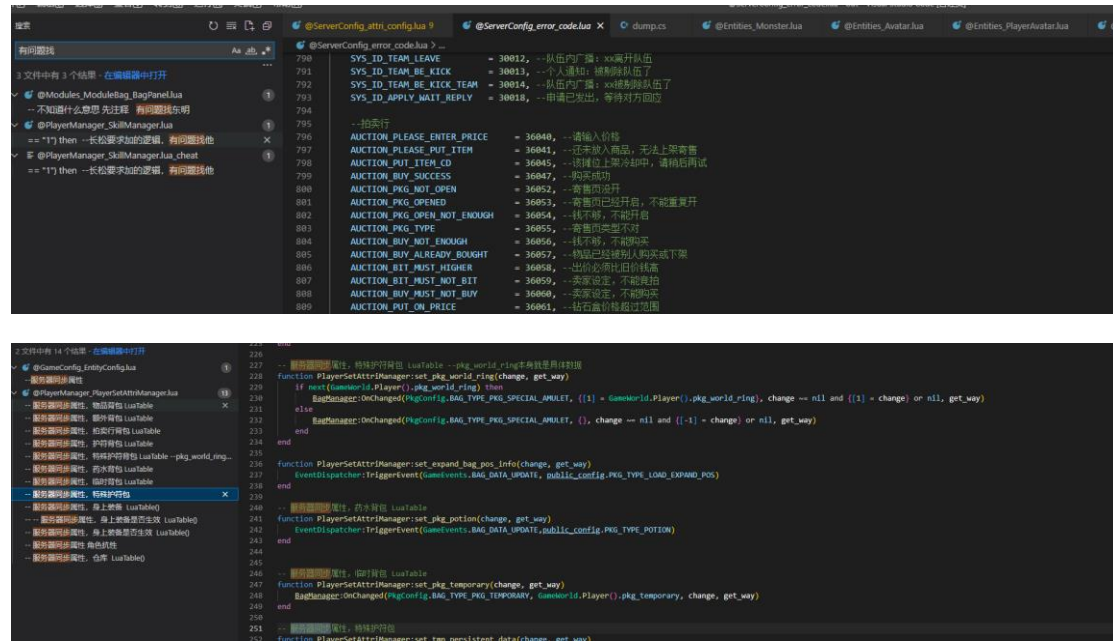
--主入口函数，从这里开始lua逻辑
function Main()
    GameManager.OperationLogManager.BaseLogData(GameWorld.public_config.OPERATION_LOG_14)
    --package.cpath = package.cpath .. ";C:/Users/admin/AppData/Roaming/JetBrains/IdeaIC2020.1/plugins/intellij-emmylua/classes/debugger/emmy/windows/x64/?..dll"
    --local dbg = require('emmy_core')
    --dbg.tcpListen('localhost', 9966)
    Init()
    Start()
end

function Init()
    local platformType = GameWorld.GetPlatformSetting()
    if platformType == "pc" then
        GameWorld.platformType = 1
    elseif platformType == "mobile" then
        GameWorld.platformType = 2
    else
        GameManager.BagItemManager:Init()
    end
    GameWorld.platformType = 3
end

GameConfig.Global.SetLang()
GameWorld.LoggerHelper:Init()
GameManager.XMLManager.CheckReloadXML()
GameManager.GUIManager:Init()
GameManager.ModuleManager:Init()
GameManager.LanguageManager:Init()
GameManager.CreateRoleManager:Init()
GameManager.DramaManager:Init()
GameManager.LoadingBarManager:Init()
RegisterEntities()
--GameManager.ModuleManager.TestF = function() end
--LoggerHelper.Log("Init1::",GameManager.XMLManager.map[1])
--LoggerHelper.Log("Init2::",GameManager.XMLManager.spell[21].group)
end
```

并且在游戏源码中发现一些有意思的注释：

```
@ServerConfig_attri_config.lua
798 SYS_ID_TEAM_LEAVE = 30012, --队伍内广播：xx离开队伍
799 SYS_ID_TEAM_BE_KICK = 30013, --个人通知：被剔除队伍了
800 SYS_ID_TEAM_BE_KICK_TEAM = 30014, --队伍内广播：xx被剔除队伍了
801 SYS_ID_APPLY_WAIT_REPLY = 30018, --申请已发出，等待对方回应
802
803 --拍卖行
804 AUCTION_PLEASE_ENTER_PRICE = 36040, --请输入价格
805 AUCTION_PLEASE_PUT_ITEM = 36041, --还未放入商品，无法上架寄售
806 AUCTION_PUT_ITEM_CD = 36045, --该摊位上架冷却中，请稍后再试
807 AUCTION_BUY_SUCCESS = 36047, --购买成功
808 AUCTION_PKG_NOT_OPEN = 36052, --寄售尚未开启
809 AUCTION_PKG_OPENED = 36053, --寄售尚未开启，不能重复开
810 AUCTION_PKG_OPEN_NOT_ENOUGH = 36054, --钱不够，不能开售
811 AUCTION_PKG_TYPE = 36055, --寄售类型不对
812 AUCTION_BUY_NOT_ENOUGH = 36056, --钱不够，不能购买
813 AUCTION_BUY_ALREADY_BOUGHT = 36057, --物品已经被别人购买或下架
814 AUCTION_BUY_MUST_HIGHER = 36058, --出价必须比旧价高
815 AUCTION_BUY_MUST_NOT_BUY = 36059, --卖家设定，不能竞拍
816 AUCTION_PUT_ON_PRICE = 36061, --钻石盒价格超过范围
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```



## 分析结论：

Seraph 在游戏代码保护方面得分为 0，毫无保护。在采用 Lua 开发的传统游戏中，往往会采用定制 lua 解释器，使用 LuaJit 进行一定程度的代码保护。由于 Seraph 并没有健全代码保护机制，导致恶意玩家分析代码的门槛与成本都很低，如果有外挂出现，对正常玩家来说是不公平的，且有一定可能会对游戏的经济模型造成影响。







## 分析结论：

1. Seraph 在反作弊能力方面得分为 0，如果存在恶意用户可以任意作弊。
2. 只测试将 Lua 重新加载到游戏的主要原始就是：该行为是进行 Lua 类游戏作弊的基础，如果该点都无法做好的话，其他方面的反作弊只会更差。

## 游戏逻辑问题

### 分析过程：

由于目前已经获取到游戏的源码，所以在分析过程中针对逻辑层我们进行了安全分析，并没有对协议层进行分析，在逻辑层方面我们主要是针对以下几点进行了安全测试，分别是角色初始化时的属性篡改：（发现该部分的敏感属性并不多，并不能提升收益）

```
function PlayerAvatar:OnEnterWorld()
    self.canPickUpStatus = 0
    self:AddListeners()
    GameManager.OperationLogManager.BaseLogData(GameWorld.public_config.OPERATION_LOG_10)
    self.cs.isLoadImmediately = true
    PlayerAvatar._base.OnEnterWorld(self)
    GameWorld.SetAccount(self.account_name)
    GameWorld.SetAvatarName(self.name)
    GameWorld.CheckFullUploadLog()
    --self._stateManager:SetDispatchEnterDead(true)
    -- self:CheckNewPlayer()
    -- self:SetPosToSlot()
    RedPointManager:OnEnterWorld()
    PlayerDataManager:OnPlayerEnterWorld()
    --EquipManager:OnPlayerEnterWorld();
    BagManager:OnPlayerEnterWorld()
    PlayerActionManager:OnPlayerEnterWorld()
    -- PlayerCommandManager:Init()
    PlayerCommandManager:OnPlayerEnterWorld()
    ResourceMappingManager:OnPlayerEnterWorld()
    CopperAvatarManager:OnPlayerEnterWorld()
    TrainManager:OnPlayerEnterWorld()
    self:LoadPlayerBleedingUIfx()
    self:SetPreloadSkill(true)
    self:SetLearnedSkillDict(self:GetEquipSkillIdList())
    self:PlayerEnterWorldLog()
    GameWorld.SetMaxScreenMonsterCount(GlobalParamsHelper.GetParamValue(916))
    if GameStateManager.GetState() ~= GAME_STATE.SCENE then
        --遮盖创角选角短暂穿帮问题
        --LoadingBarManager:ShowProgressBar(0.1)
    end
    --self:CheckLoadEmptyWeapon()
    self:SetFunctionOpenActive()
    self:SetServerInfo(LocalSetting.settingData.SelectedServerID, ServerListManager.SelectedServerName())

    --EventDispatcher:TriggerEvent(GameEvents._GameON_FUNCTIONPREVIEWPANEL) --进入游戏后刷新functionPreview
    if GUIManager.hasHandleEscape then
        self._clearStackTimer = TimerHeap.AddSecTimer(0, self._clearStackTime, 0, function()
            GUIManager.ClearClosePanelStack()
        end)
    end
    --GameWorld.ShowMainCamera()
    self:SetQualitySettingValue(0)

    if (GameLoader.SystemConfig.IsUsePlatformSdk and GameManager.LoginManager.accessToken) then
```

其次是主动攻击时的一些技能相关的篡改:(发现该部分只做展示并没有实际参与伤害校验)

```
function SkillManager:GetSkillShowDamage(data,level)
    --技能主要伤害
    local skillId = data.skillId
    local owner = data.owner--owner: 1=玩家,2=佣兵,3=其他玩家
    local cal = SkillDataHelper.GetDamageCal(skillId)
    if next(cal) then
        local param = self:DescParam(data,level)
        local minDamage
        local maxDamage
        local factor
        if owner == 1 then
            minDamage = AttributeManager.GetAttributeByAttrID(attri_config.ATTRI_ID_DMG_MIN)
            maxDamage = AttributeManager.GetAttributeByAttrID(attri_config.ATTRI_ID_DMG_MAX)
            factor = AttributeManager.GetAttributeByAttrID(attri_config.ATTRI_ID_NOW_WEAPON_ED)
        elseif owner == 2 then
            if data.mercenaryAtt then
                minDamage = data.mercenaryAtt[attri_config.ATTRI_ID_DMG_MIN] or 1
                maxDamage = data.mercenaryAtt[attri_config.ATTRI_ID_DMG_MAX] or 1
                factor = data.mercenaryAtt[attri_config.ATTRI_ID_NOW_WEAPON_ED] or 0
            else
                minDamage = 1
                maxDamage = 1
                factor = 0
            end
        elseif owner == 3 then
            local attData = OtherPlayerManager.GetAllAttributesData()
            minDamage = attData[attri_config.ATTRI_ID_DMG_MIN] or 1
            maxDamage = attData[attri_config.ATTRI_ID_DMG_MAX] or 1
            factor = attData[attri_config.ATTRI_ID_NOW_WEAPON_ED] or 0
        end
        local min = 0
        local max = 0
        for i,v in ipairs(cal) do
            if v == 0 then
                elseif v == 1 then
                    min = min + param[i][1]/100 * minDamage * (1 + factor/100)
                    max = max + param[i][1]/100 * maxDamage * (1 + factor/100)
                elseif v == 2 then
                    if #param[i] == 2 then
                        loggerHelper.Error("Config Error!!!!!!Excel:spell.xml ---- damage_cal ----id:".skillId)
                    else
                        min = min + param[i][1]
                        max = max + param[i][2]
                    end
                end
            end
        end
        min = min - min % 0.1
        max = max - max % 0.1
        max = max * 100
        min = min * 100
        return (min,max)
    end
end
```

最后是怪物被攻击时的逻辑修改（发现该点修改后并没有实际的意义，猜测该模块开发时的主要目的是触发事件做记录并不存在实际的计算参与）

```
--这个方法，目前都是技能伤害导致血量变化。
function Monster:OnChangeHp(casterId, oldHp, newHp, attackType)

    -- 修改，只要血量变化 就会杀怪
    -- LoggerHelper.Error("[Seraph]casterId: " .. tostring(casterId) .. "   oldHp: " .. tostring(oldHp) .. "   newHp: " .. tostr
    if casterId == GameWorld.Player().id and newHp <= 0 then
        EventDispatcher:TriggerEvent(GameEvents.KillMonster_Exp)
    end

    if casterId ~= GameWorld.Player().id then
        return
    end

    if newHp <= 0 then
        return
    end
    -- LoggerHelper.Error("casterId: " .. tostring(casterId) .. "   oldHp: " .. tostring(oldHp) .. "   newHp: " .. tostring(ne
    EventDispatcher:TriggerEvent(GameEvents.MONSTER_HP_CHANGE_EVENT, self.id)

    if self.behavior_state == 0 then
        local exit_action = MonsterDataHelper:GetExitAction(self.monster_id)
        if exit_action > 0 then
            local player = GameWorld.Player()
            player.server.set_monster_behavior_state(self.id, 1)
        end
    end

    if self:GetEntityAI() then
        self:GetEntityAI():on_change_hp(casterId, tonumber(newHp) - tonumber(oldHp), attackType)
    end
    --LoggerHelper.Error("[Seraph]事件二次触发casterId: " .. tostring(casterId).. "当前ID"..GameWorld.Player().id .. "   oldHp: " .. to
    casterId = GameWorld.Player().id
    newHp = 0
    attackType = 30
    if casterId == GameWorld.Player().id and newHp <= 0 then
        EventDispatcher:TriggerEvent(GameEvents.KillMonster_Exp)
        return
    end
end
end
```

## 分析结论：

1. Seraph 在我们随机篡改的三点上均未生效，证明其伤害计算与展示是分离进行，或者是由服务器进行计算其安全性还是有一定保障的，评分 3 分 0。
2. 但是其部分伤害判定存放在本地，作弊空间还是存在的。

# 游戏 RPC 分析

该游戏采用 *protobuf* 进行协议交互，其中 Web3 相关的交互也是使用使用该方案，，目

前针对该部分还未进行细致的测试，后期可能会对 *ProtoBuf* 部分进行细致测试。

```
protobuf
79 文件中有 1676 个结果，在编辑器中打开

WEAPON__NIGHTMAREUPGRADE_FIELD = protobuf.FieldDescriptor{
WEAPON__HELLUPGRADE_FIELD = protobuf.FieldDescriptor{
WEAPON__REPOINT_FIELD = protobuf.FieldDescriptor{
WEAPON__PICKUPSOUND_FIELD = protobuf.FieldDescriptor{
WEAPON__EFFECT_FIELD = protobuf.FieldDescriptor{
WEAPON__MAINTYPE_FIELD = protobuf.FieldDescriptor{
WEAPON__BASE_CRIT_FIELD = protobuf.FieldDescriptor{
WEAPON__BASE_SKILLCRIT_FIELD = protobuf.FieldDescriptor{
  weapon = protobuf.Message(WEAPON)
  local protobuf = require "protobuf.protobuf"
  local protobuf = require "protobuf.protobuf"
  protobuf = require "protobuf.protobuf"
  WORLD_BOSS__ID_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__SENCE_ID_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__LEVEL_LIMIT_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__REALM_LIMIT_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__REALM_SHOW_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_LEVEL_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_NAME_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_NAME_ICON_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_ICON_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_MONSTER_ID_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_DROPS_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__BOSS_POS_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__PEACE_SHOW_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__KILL_TIME_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__SCALE_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__POSUI_FIELD = protobuf.FieldDescriptor{
  WORLD_BOSS__MODE_FIELD = protobuf.FieldDescriptor{
  world_boss = protobuf.Message(WORLD_BOSS)
  local protobuf = require "protobuf.protobuf"
  local protobuf = require "protobuf.protobuf"
  local protobuf = require "protobuf.protobuf"
  WORLD_MAP__ID_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__BUTTON_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__BUTTON_NAME_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__TYPE_MAP_ICON_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__REGIONAL_ICON_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__SCENE_ID_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__DOWN_POSITION_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__UP_POSITION_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__MAP_LEVEL_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__SCALE_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__OFFSET_FIELD = protobuf.FieldDescriptor{
  WORLD_MAP__SCALE2_FIELD = protobuf.FieldDescriptor{
  world_map = protobuf.Message(WORLD_MAP)

@Protobuf.world_boss.pb.lua
119 WORLD_BOSS__BOSS_MONSTER_ID_FIELD.label = 2
120 WORLD_BOSS__BOSS_MONSTER_ID_FIELD.has_default_value = false
121 WORLD_BOSS__BOSS_MONSTER_ID_FIELD.default_value = 0
122 WORLD_BOSS__BOSS_MONSTER_ID_FIELD.type = 5
123 WORLD_BOSS__BOSS_MONSTER_ID_FIELD.cpp_type = 1
124
125 WORLD_BOSS__BOSS_DROPS_FIELD.name = "_boss_drops"
126 WORLD_BOSS__BOSS_DROPS_FIELD.full_name = ".world_boss__boss_drops"
127 WORLD_BOSS__BOSS_DROPS_FIELD.number = 11
128 WORLD_BOSS__BOSS_DROPS_FIELD.index = 10
129 WORLD_BOSS__BOSS_DROPS_FIELD.label = 2
130 WORLD_BOSS__BOSS_DROPS_FIELD.has_default_value = false
131 WORLD_BOSS__BOSS_DROPS_FIELD.default_value = ""
132 WORLD_BOSS__BOSS_DROPS_FIELD.type = 9
133 WORLD_BOSS__BOSS_DROPS_FIELD.cpp_type = 9
134
135 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.name = "_boss_extra_drops"
136 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.full_name = ".world_boss__boss_extra_drops"
137 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.number = 12
138 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.index = 11
139 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.label = 2
140 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.has_default_value = false
141 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.default_value = ""
142 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.type = 9
143 WORLD_BOSS__BOSS_EXTRA_DROPS_FIELD.cpp_type = 9
144
145 WORLD_BOSS__BOSS_POS_FIELD.name = "_boss_pos"
146 WORLD_BOSS__BOSS_POS_FIELD.full_name = ".world_boss__boss_pos"
147 WORLD_BOSS__BOSS_POS_FIELD.number = 13
148 WORLD_BOSS__BOSS_POS_FIELD.index = 12
149 WORLD_BOSS__BOSS_POS_FIELD.label = 2
150 WORLD_BOSS__BOSS_POS_FIELD.has_default_value = false
151 WORLD_BOSS__BOSS_POS_FIELD.default_value = ""
152 WORLD_BOSS__BOSS_POS_FIELD.type = 9
153 WORLD_BOSS__BOSS_POS_FIELD.cpp_type = 9
154
155 WORLD_BOSS__PEACE_SHOW_FIELD.name = "_peace_show"
156 WORLD_BOSS__PEACE_SHOW_FIELD.full_name = ".world_boss__peace_show"
157 WORLD_BOSS__PEACE_SHOW_FIELD.number = 14
158 WORLD_BOSS__PEACE_SHOW_FIELD.index = 13
159 WORLD_BOSS__PEACE_SHOW_FIELD.label = 2
160 WORLD_BOSS__PEACE_SHOW_FIELD.has_default_value = false
161 WORLD_BOSS__PEACE_SHOW_FIELD.default_value = 0
162 WORLD_BOSS__PEACE_SHOW_FIELD.type = 5
163 WORLD_BOSS__PEACE_SHOW_FIELD.cpp_type = 1
164
165 WORLD_BOSS__KILL_TIME_FIELD.name = "_kill_time"
166 WORLD_BOSS__KILL_TIME_FIELD.full_name = ".world_boss__kill_time"
167 WORLD_BOSS__KILL_TIME_FIELD.number = 15
168 WORLD_BOSS__KILL_TIME_FIELD.index = 14
169 WORLD_BOSS__KILL_TIME_FIELD.label = 2
170 WORLD_BOSS__KILL_TIME_FIELD.has_default_value = false
171 WORLD_BOSS__KILL_TIME_FIELD.default_value = 0
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```































## WEB3 安全分析：

### 概要：

目前 Seraph 并未发行代币, Mint 合约为使用代理合约的常规 NFT721 合约总供应量 3225, 并且不管是 Mint 还是跨链, 均有 Role 控制, 链上安全性可控。

ABI for the implementation contract at 0x4f20065affb0438a0a4703610917c2f72c52d3dd, using the EIP-1967 Transparent Proxy pattern.

[\[Expand all\]](#) [\[Reset\]](#)

1. ADMIN_MINTER_ROLE	 
2. BUY_MINTER_ROLE	 
3. CROSS_MINTER_ROLE	 
4. DEFAULT_ADMIN_ROLE	 
5. GAME_MINTER_ROLE	 
6. PAUSER_ROLE	 
7. balanceOf	 
8. getApproved	 
9. getContent	 
10. getRoleAdmin	 
11. getRoleMember	 
12. getRoleMemberCount	 
13. hasRole	 
14. isApprovedForAll	 
15. name	 

### 游戏内经济系统安全：

目前 Seraph 打金主要方式还是以灵魂晶石为主, 不管是打造灵魂之匣还是开起均由服务端进行判定, 客户端仅做请求发起, 安全性主要由服务端做控制, 故其安全性评估不在客户端安全评估范围内, 后期 Damocles 可能会对所有请求进行梳理并黑盒测试。

```
end
elseif act_id == action_config.CHAOS_GET_MONEY_REQ then --获取账号pkg_chaos的货币数量
    self.XGoldCount = args
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.XGoldCount,args)
    EventDispatcher:TriggerEvent(gameEvents.RefreshXGoldSum)
elseif act_id == action_config.CHAOS_ADD_MONEY_REQ then --游戏内注册pkg_chaos货币
    ChaosWarehouseManager:RequestXGoldCount()
elseif act_id == action_config.CHAOS_REMOVE_MONEY_REQ then --游戏内从pkg_chaos取货币
    ChaosWarehouseManager:RequestXGoldCount()
elseif act_id == action_config.CHAOS_ITEM_UPDATE_RESP then --web端更新了pkg_chaos
    ChaosWarehouseManager:RequestPKGChaos(self.OnePageCount)
elseif act_id == action_config.CHAOS_MONEY_UPDATE_RESP then --web端更新了货币
    ChaosWarehouseManager:RequestXGoldCount()
elseif act_id == action_config.CHAOS_GET_PACK_Y_CNT then --获取该魂一个灵魂之屋需要的y币数量
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.YCoinToSoulBoxExchangeProportion,GameWorld.Player().tmp_persistent_data[public_config.TMP_PERSISTENT_DATA_KEY_DAY_PACK_Y_CNT])
elseif act_id == action_config.CHAOS_PACK_Y_COIN then --该魂灵魂之屋
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.UpdateSoulPackingView)

    GUIManager:ShowText(11,LanguageDataHelper.CreateContent(3035,{"0"}-ChaosWarehouseManager.itemCountText))
elseif act_id == action_config.CHAOS_UNPACK_SOUL_BOX then --开始灵魂之屋
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.UpdateSoulPackingView)

    GUIManager:ShowText(11,LanguageDataHelper.CreateContent(3037,{"0"}-ChaosWarehouseManager.itemCountText))
elseif act_id == action_config.CHAOS_SRF_SMELT then --灵魂之屋冶炼SRF
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.UpdateSPFSmeltView)

    GUIManager:ShowText(11,LanguageDataHelper.CreateContent(3038,{"0"}-ChaosWarehouseManager.itemCountText))
elseif act_id == action_config.CHAOS_SRF_EXCHANGE then --SRF兑换灵魂之屋
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.UpdateSPFSmeltView)

    GUIManager:ShowText(11,LanguageDataHelper.CreateContent(3034,{"0"}-ChaosWarehouseManager.itemCountText))
elseif act_id == action_config.CHAOS_CHARGE_REQ then --充值
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.Recharge,args)
elseif act_id == action_config.CHAOS_WITHDRAW_REQ then --提现
    EventDispatcher:TriggerEvent(gameEvents.ChaosWarehouse.UpdateSPFSmeltView)
end
end
```

## 关于 Damocles

Damocles labs 是成立于 2023 年的安全团队,专注于 Web3 行业的安全,业务内容包括:

合约代码审计, 业务代码审计, 渗透测试, GameFi 代码审计, GameFi 漏洞挖掘, GameFi 外挂分析, GameFi 反作弊。

我们会在 Web3 安全行业持续发力, 并且尽可能多的输出分析报告, 提升项目方和用户对

GameFi 安全的感知度, 以及促进行业的安全发展。

Twitter: <https://twitter.com/DamoclesLabs>

Discord: <https://discord.gg/xd6H6eqFHz>