Damocles

Abyss World Security Analysis

2024.03.21

Senna

DAMOCLES LABS

Contents

一、 Summary

As an FPS game, NyanHeroes has a security level of 0 on its client side. Due to the high requirement for fairness in STG games, if fairness is compromised, it will disrupt the overall token output and negatively impact the gaming experience for regular users. Additionally, there are suspected vulnerabilities in the game's settlement logic. During its initial release, the game also experienced issues with PDB leakage and lacks any anti-cheat system integration. Therefore, Damocles has determined its security rating to be 1 star.

Security Rating： ★☆☆☆☆

二、 Game Background

- Game Version：Beta01
- Genres & Engine：FPS，UE5
- Possible Issues in Gameplay：
  - Unauthorized movement (modifying local character attributes for speed enhancement).
  - No recoil
  - Aimbot
  - Teleportation
  - Settlement replay attack
  - Modification of local character attributes, such as jumping
  - Wallhacks (ability to see through walls)

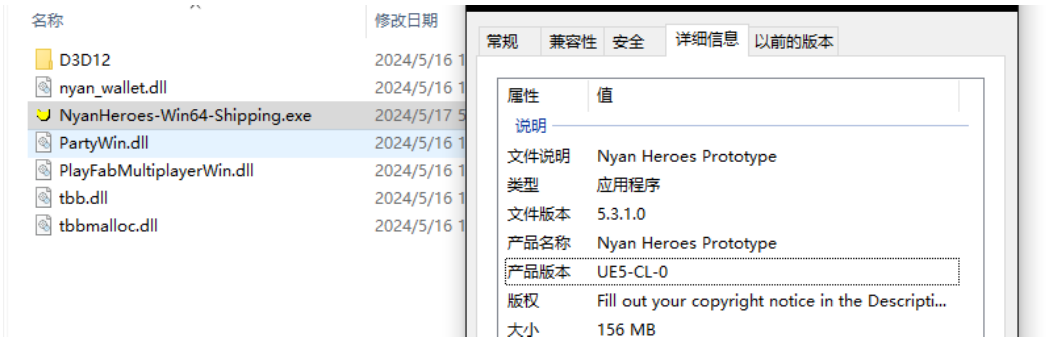
Damocles

三、 Game Security Analysis

Game Code Protection：

Analysis Process：

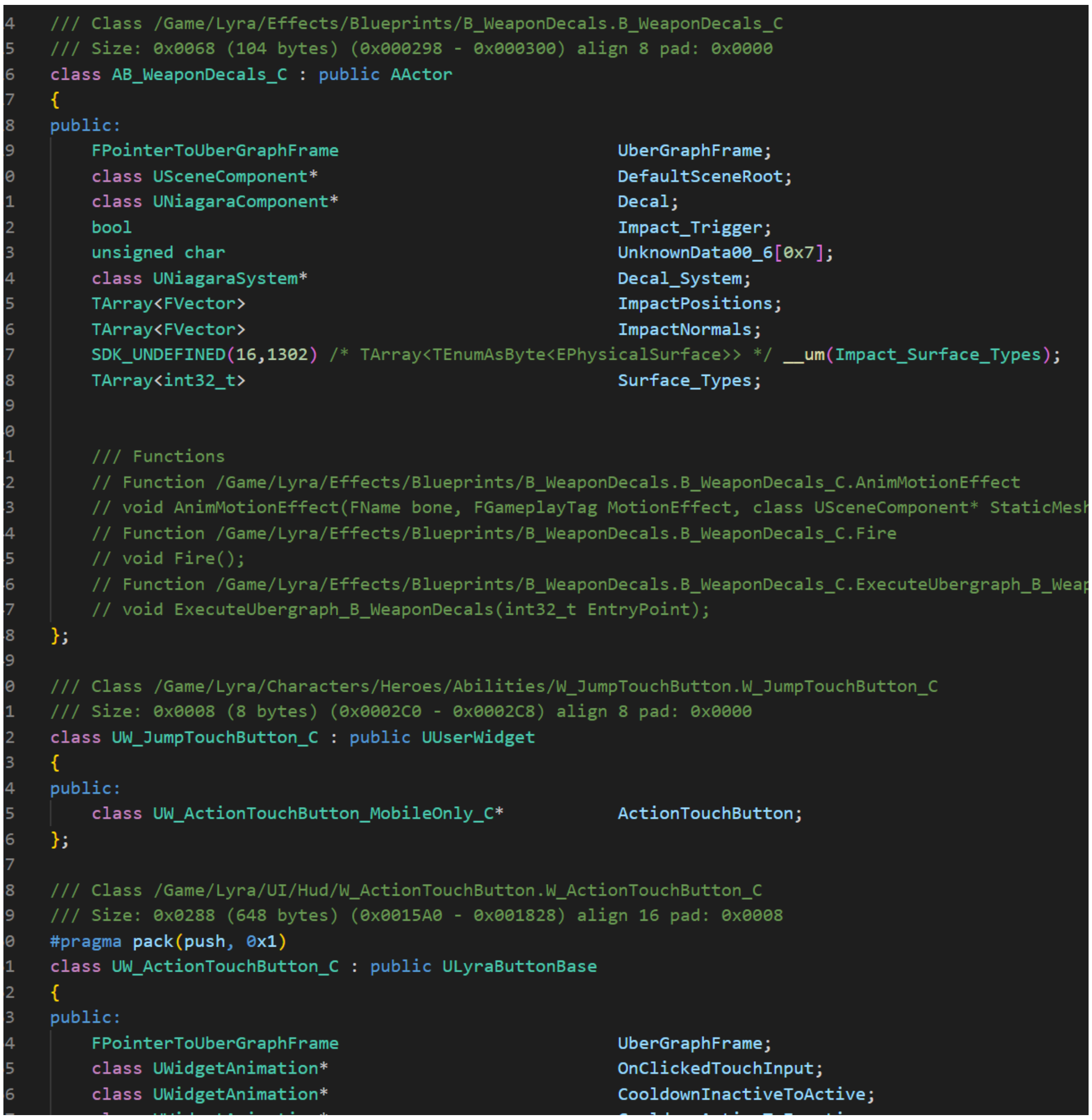
1. Since different engines have different analysis modes, it is important to determine the game engine used after obtaining the game EXE. By analyzing the basic information of the game, we can determine that this game was developed using UE5。



名称	修改日期
D3D12	2024/5/16 1
nyan_wallet.dll	2024/5/16 1
NyanHeroes-Win64-Shipping.exe	2024/5/17 5
PartyWin.dll	2024/5/16 1
PlayFabMultiplayerWin.dll	2024/5/16 1
tbb.dll	2024/5/16 1
tbbmalloc.dll	2024/5/16 1

属性	值
说明	
文件说明	Nyan Heroes Prototype
类型	应用程序
文件版本	5.3.1.0
产品名称	Nyan Heroes Prototype
产品版本	UE5-CL-0
版权	Fill out your copyright notice in the Descripti...
大小	156 MB

2. Using tools to dump the structure of UE (Unreal Engine) characters for fast positioning. Once located, indexing and modification can be done through UE's unique linked list structure.



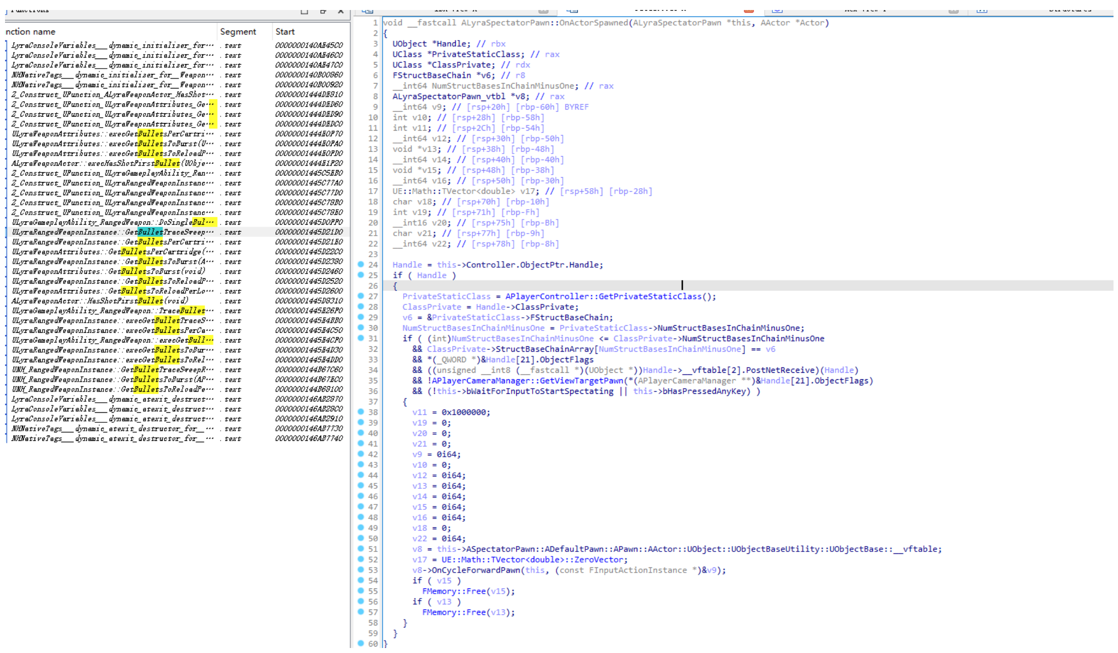
```
/// Class /Game/Lyra/Effects/Blueprints/B_WeaponDecals.B_WeaponDecals_C
/// Size: 0x0068 (104 bytes) (0x000298 - 0x000300) align 8 pad: 0x0000
class AB_WeaponDecals_C : public AActor
{
public:
    FPointerToUberGraphFrame                              UberGraphFrame;
    class USceneComponent*                                DefaultSceneRoot;
    class UNiagaraComponent*                              Decal;
    bool                                                  Impact_Trigger;
    unsigned char                                         UnknownData00_6[0x7];
    class UNiagaraSystem*                                 Decal_System;
    TArray<FVector>                                       ImpactPositions;
    TArray<FVector>                                       ImpactNormals;
    SDK_UNDEFINED(16,1302) /* TArray<TEnumAsByte<EPhysicalSurface>> */ __um(Impact_Surface_Types);
    TArray<int32_t>                                       Surface_Types;


    /// Functions
    // Function /Game/Lyra/Effects/Blueprints/B_WeaponDecals.B_WeaponDecals_C.AnimMotionEffect
    // void AnimMotionEffect(FName bone, FGameplayTag MotionEffect, class USceneComponent* StaticMesh
    // Function /Game/Lyra/Effects/Blueprints/B_WeaponDecals.B_WeaponDecals_C.Fire
    // void Fire();
    // Function /Game/Lyra/Effects/Blueprints/B_WeaponDecals.B_WeaponDecals_C.ExecuteUbergraph_B_Weap
    // void ExecuteUbergraph_B_WeaponDecals(int32_t EntryPoint);
};

/// Class /Game/Lyra/Characters/Heroes/Abilities/W_JumpTouchButton.W_JumpTouchButton_C
/// Size: 0x0008 (8 bytes) (0x0002C0 - 0x0002C8) align 8 pad: 0x0000
class UW_JumpTouchButton_C : public UUserWidget
{
public:
    class UW_ActionTouchButton_MobileOnly_C*           ActionTouchButton;
};

/// Class /Game/Lyra/UI/Hud/W_ActionTouchButton.W_ActionTouchButton_C
/// Size: 0x0288 (648 bytes) (0x0015A0 - 0x001828) align 16 pad: 0x0008
#pragma pack(push, 0x1)
class UW_ActionTouchButton_C : public ULyraButtonBase
{
public:
    FPointerToUberGraphFrame                              UberGraphFrame;
    class UWidgetAnimation*                               OnClickedTouchInput;
    class UWidgetAnimation*                               CooldownInactiveToActive;
```

Furthermore, during static code analysis using IDA, it was discovered that the game code structure is complete and lacks any protection measures.



```
nction name                                                  Segment  Start
LyraConsoleVariables___dynamic_initializer_for…              .text    0000000140AB45C0
LyraConsoleVariables___dynamic_initializer_for…              .text    0000000140AB46C0
LyraConsoleVariables___dynamic_initializer_for…              .text    0000000140AB47C0
NHNativeTags___dynamic_initializer_for__Weapon…              .text    0000000140B00860
NHNativeTags___dynamic_initializer_for__Weapon…              .text    0000000140B00920
Z_Construct_UFunction_ALyraWeaponActor_HasShot…              .text    00000001444DB810
Z_Construct_UFunction_ULyraWeaponAttributes_Ge…              .text    00000001444DBD60
Z_Construct_UFunction_ULyraWeaponAttributes_Ge…              .text    00000001444DBD90
Z_Construct_UFunction_ULyraWeaponAttributes_Ge…              .text    00000001444DBDC0
ULyraWeaponAttributes::execGetBulletsPerCartri…              .text    00000001444B0F70
ULyraWeaponAttributes::execGetBulletsToBurst(U…              .text    00000001444B0FA0
ULyraWeaponAttributes::execGetBulletsToReloadP…              .text    00000001444B0FD0
ALyraWeaponActor::execHasShotFirstBullet(UObje…              .text    00000001444B1F20
Z_Construct_UFunction_ULyraGameplayAbility_Ran…              .text    00000001445C5BB0
Z_Construct_UFunction_ULyraRangedWeaponInstanc…              .text    00000001445C77A0
Z_Construct_UFunction_ULyraRangedWeaponInstanc…              .text    00000001445C77D0
Z_Construct_UFunction_ULyraRangedWeaponInstanc…              .text    00000001445C78B0
Z_Construct_UFunction_ULyraRangedWeaponInstanc…              .text    00000001445C78E0
ULyraGameplayAbility_RangedWeapon::DoSingleBul…              .text    00000001445D0FF0
ULyraRangedWeaponInstance::GetBulletTraceSweep…              .text    00000001445D21D0
ULyraRangedWeaponInstance::GetBulletsPerCartri…              .text    00000001445D21E0
ULyraWeaponAttributes::GetBulletsPerCartridge(…              .text    00000001445D22C0
ULyraRangedWeaponInstance::GetBulletsToBurst(A…              .text    00000001445D2380
ULyraWeaponAttributes::GetBulletsToBurst(void)               .text    00000001445D2460
ULyraRangedWeaponInstance::GetBulletsToReloadP…              .text    00000001445D2520
ULyraWeaponAttributes::GetBulletsToReloadPerLo…              .text    00000001445D2600
ALyraWeaponActor::HasShotFirstBullet(void)                   .text    00000001445D8310
ULyraGameplayAbility_RangedWeapon::TraceBullet…              .text    00000001445E26F0
ULyraRangedWeaponInstance::execGetBulletTraceS…              .text    00000001445E4BB0
ULyraRangedWeaponInstance::execGetBulletsPerCa…              .text    00000001445E4C50
ULyraGameplayAbility_RangedWeapon::execGetBull…              .text    00000001445E4CF0
ULyraRangedWeaponInstance::execGetBulletsToBur…              .text    00000001445E4D30
ULyraRangedWeaponInstance::execGetBulletsToRel…              .text    00000001445E4DD0
UNH_RangedWeaponInstance::GetBulletTraceSweepR…              .text    0000000144B67C60
UNH_RangedWeaponInstance::GetBulletsToBurst(AP…              .text    0000000144B67BC0
UNH_RangedWeaponInstance::GetBulletsToReloadPe…              .text    0000000144B68100
LyraConsoleVariables___dynamic_atexit_destruct…              .text    0000000146AB2870
LyraConsoleVariables___dynamic_atexit_destruct…              .text    0000000146AB28C0
LyraConsoleVariables___dynamic_atexit_destruct…              .text    0000000146AB2910
NHNativeTags___dynamic_atexit_destructor_for__…              .text    0000000146AB7730
NHNativeTags___dynamic_atexit_destructor_for__…              .text    0000000146AB7740
```

```
void __fastcall ALyraSpectatorPawn::OnActorSpawned(ALyraSpectatorPawn *this, AActor *Actor)
{
  UObject *Handle; // rbx
  UClass *PrivateStaticClass; // rax
  UClass *ClassPrivate; // rdx
  FStructBaseChain *v6; // r8
  __int64 NumStructBasesInChainMinusOne; // rax
  ALyraSpectatorPawn_vtbl *v8; // rax
  __int64 v9; // [rsp+20h] [rbp-60h] BYREF
  int v10; // [rsp+28h] [rbp-58h]
  int v11; // [rsp+2Ch] [rbp-54h]
  __int64 v12; // [rsp+30h] [rbp-50h]
  void *v13; // [rsp+38h] [rbp-48h]
  __int64 v14; // [rsp+40h] [rbp-40h]
  void *v15; // [rsp+48h] [rbp-38h]
  __int64 v16; // [rsp+50h] [rbp-30h]
  UE::Math::TVector<double> v17; // [rsp+58h] [rbp-28h]
  char v18; // [rsp+70h] [rbp-10h]
  int v19; // [rsp+71h] [rbp-Fh]
  __int16 v20; // [rsp+75h] [rbp-Bh]
  char v21; // [rsp+77h] [rbp-9h]
  __int64 v22; // [rsp+78h] [rbp-8h]

  Handle = this->Controller.ObjectPtr.Handle;
  if ( Handle )
  {
    PrivateStaticClass = APlayerController::GetPrivateStaticClass();
    ClassPrivate = Handle->ClassPrivate;
    v6 = &PrivateStaticClass->FStructBaseChain;
    NumStructBasesInChainMinusOne = PrivateStaticClass->NumStructBasesInChainMinusOne;
    if ( (int)NumStructBasesInChainMinusOne <= ClassPrivate->NumStructBasesInChainMinusOne
      && ClassPrivate->StructBaseChainArray[NumStructBasesInChainMinusOne] == v6
      && *(_QWORD *)&Handle[21].ObjectFlags
      && ((unsigned __int8 (__fastcall *)(UObject *))Handle->__vftable[2].PostNetReceive)(Handle)
      && !APlayerCameraManager::GetViewTargetPawn(*(APlayerCameraManager **)&Handle[21].ObjectFlags)
      && (!this->bWaitForInputToStartSpectating || this->bHasPressedAnyKey) )
    {
      v11 = 0x1000000;
      v19 = 0;
      v20 = 0;
      v21 = 0;
      v9 = 0i64;
      v10 = 0;
      v12 = 0i64;
      v13 = 0i64;
      v14 = 0i64;
      v15 = 0i64;
      v16 = 0i64;
      v18 = 0;
      v22 = 0i64;
      v8 = this->ASpectatorPawn::ADefaultPawn::APawn::AActor::UObject::UObjectBaseUtility::UObjectBase::__vftable;
      v17 = UE::Math::TVector<double>::ZeroVector;
      v8->OnCycleForwardPawn(this, (const FInputActionInstance *)&v9);
      if ( v15 )
        FMemory::Free(v15);
      if ( v13 )
        FMemory::Free(v13);
    }
  }
}
```

Therefore, it is possible to gain a basic understanding of the game's code logic by combining the dumped UE data structures and using reverse engineering tools.

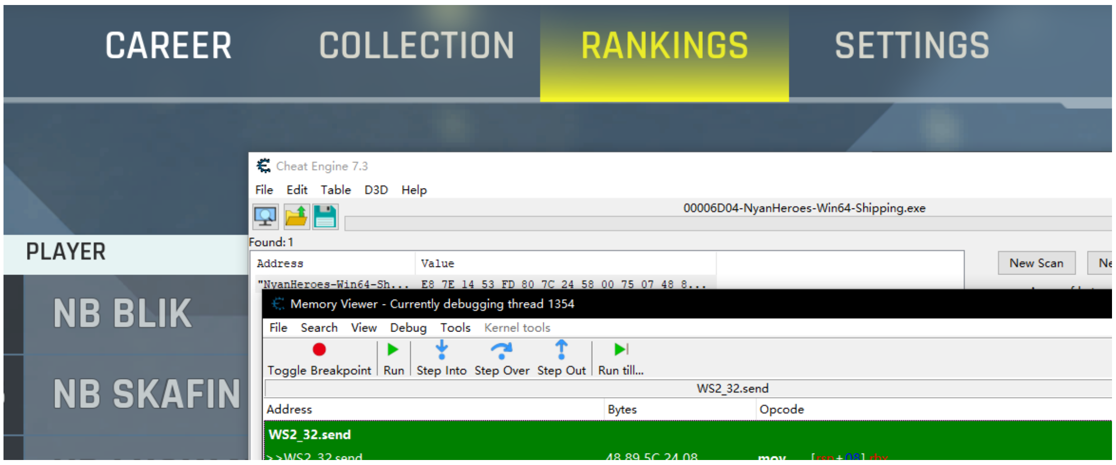
Analysis Conclusion:

Nyan Heroes scores 0 in terms of game code protection. The client code lacks any form of protection, and combined with PDB analysis, it can be determined to some extent as source code-level analysis. In other words, there are no barriers or measures in place to counteract malicious players. Additionally, the game has not integrated any anti-cheat systems, such as the free EAC (Easy Anti-Cheat) solution.

Game Basic Anti-Cheat:

Analysis Process:

1. In terms of basic anti-cheat detection, we primarily determine whether the game loads and executes external logic by replacing Lua files.
2. While attaching with Cheat Engine (CE) in the game's open state and setting breakpoints on common functions, it was observed that the game did not exit or provide any prompts..



00006D04-NyanHeroes-Win64-Shipping.exe

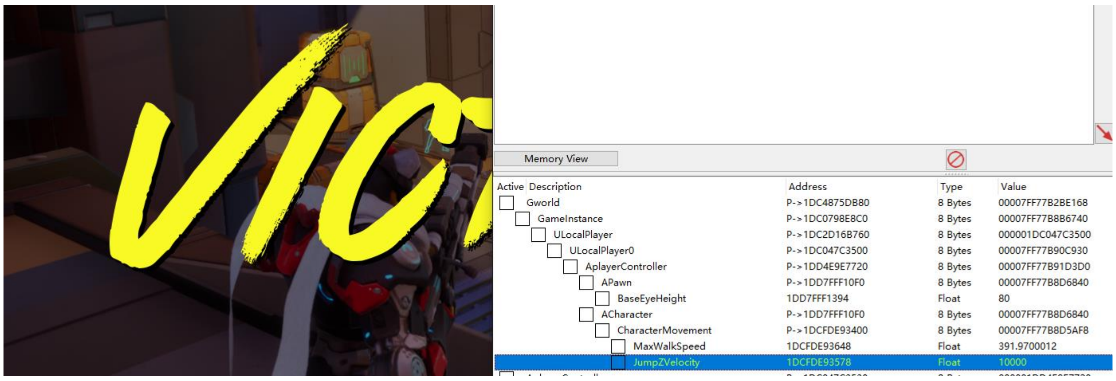
Found: 1

```
Address                  Value
"NyanHeroes-Win64-Sh...  E8 7E 14 53 FD 80 7C 24 58 00 75 07 48 8...
```

WS2_32.send

```
Address          Bytes          Opcode
WS2_32.send
>>WS2_32.send    48 89 5C 24 08  mov    [rsp+08],rbx
```

3. It is possible to directly modify the in-game character attributes, and the Game Server (GS) does not perform any kick-out operations.



Active	Description	Address	Type	Value
☐	Gworld	P->1DC4875DB80	8 Bytes	00007FF77B2BE168
☐	GameInstance	P->1DC0798E8C0	8 Bytes	00007FF77B8B6740
☐	ULocalPlayer	P->1DC2D16B760	8 Bytes	000001DC047C3500
☐	ULocalPlayer0	P->1DC047C3500	8 Bytes	00007FF77B90C930
☐	AplayerController	P->1DD4E9E7720	8 Bytes	00007FF77B91D3D0
☐	APawn	P->1DD7FFF10F0	8 Bytes	00007FF77B8D6840
☐	BaseEyeHeight	1DD7FFF1394	Float	80
☐	ACharacter	P->1DD7FFF10F0	8 Bytes	00007FF77B8D6840
☐	CharacterMovement	P->1DCFDE93400	8 Bytes	00007FF77B8D5AF8
☐	MaxWalkSpeed	1DCFDE93648	Float	391.9700012
☐	JumpZVelocity	1DCFDE93578	Float	10000

Analysis Conclusion:

1. NyanHeroes has a basic score of 0 in terms of anti-cheat protection. It

lacks countermeasures against dynamic debugging and dynamic analysis, making it easily exploitable by malicious players at a low cost. Additionally, it lacks the ability to detect players who are already cheating. The leakage of PDB files further exacerbates the issue, allowing targeted analysis of the game, such as determining the influencing factors of the raycast detection mechanism used in FPS games to determine bullet trace..

2. The reason for focusing only on anti-debugging and read/write protection testing is that for a cheat program, finding data and implementing desired functionalities can be achieved through debugging and memory manipulation. If the most fundamental protection measures in these two aspects are missing, other detection methods such as code injection and hooking become meaningless.

Game Logic Issues

Analysis Process:

As we analyzed the game, we discovered that NyanHeroes has issues with incomplete data synchronization. Additionally, during our analysis, we found the presence of cheats in the market. Based on this, we expanded our analysis of the game's logic and introduced an analysis of cheat principles.

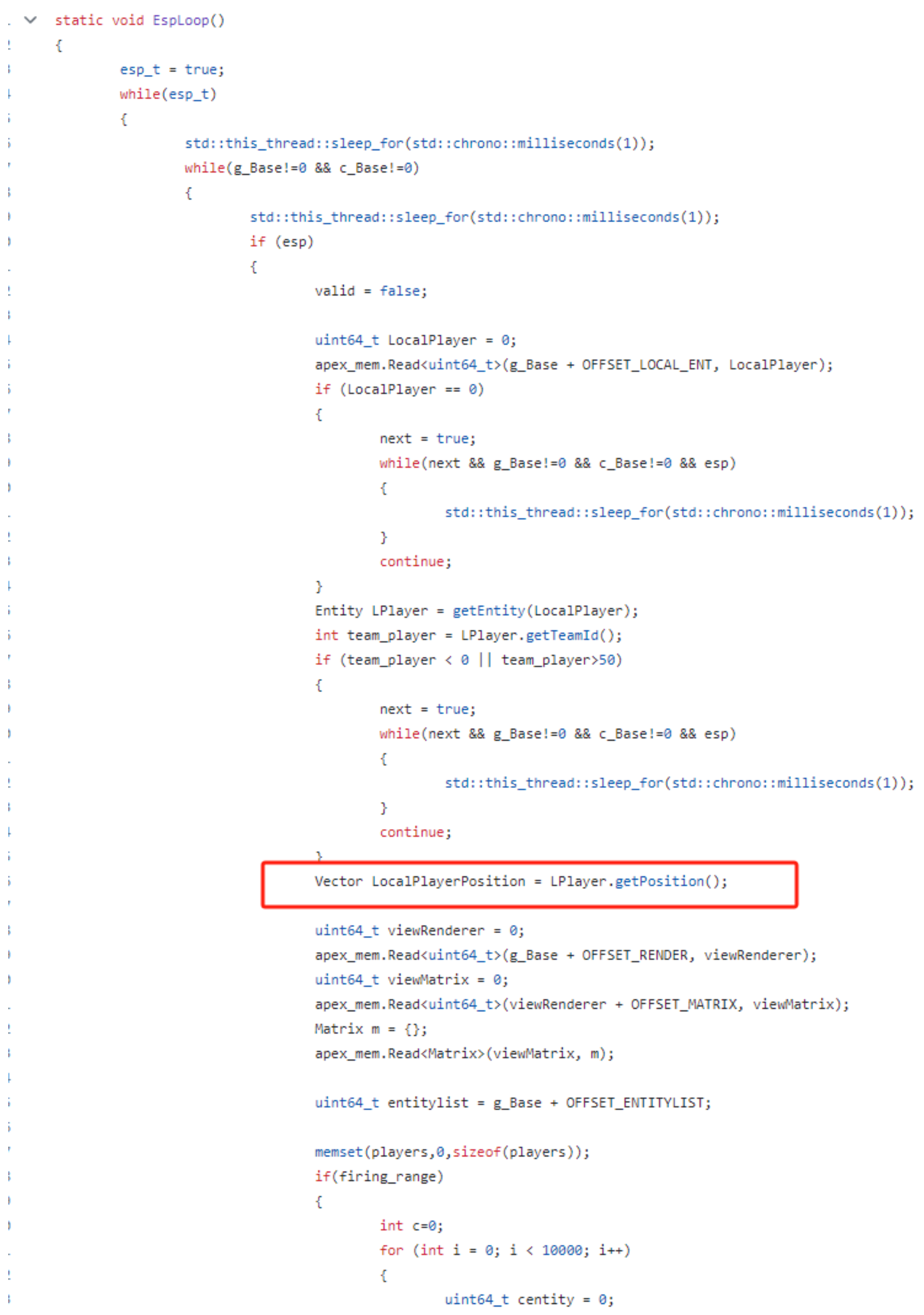
NyanHeroes utilizes the Unreal Engine, and the implementation logic for wallhacks (ESP) and aimbots can be templated. The implementation primarily relies on character-related attributes, specifically:

Attributes to modify for aimbot: APlayerController->Apawn->FRotation {Pitch, Yaw, Roll}

Attributes to retrieve for wallhacks: APlayerController->Apawn->FLocation {X, Y, Z}

Once the memory addresses for these data are obtained, the cheat program performs matrix transformations to convert the two-dimensional data into three-dimensional data. After calculations are performed by the cheat program, modifications are made in the game to achieve aimbot or wallhack functionality. It is worth noting that FPS games developed using the Unreal Engine often have templates for bullet tracking. Therefore, we recommend that the project team pays attention to detection in this aspect in the future.

Taking Apex, which uses the Source Engine, as an example, the logic is similar, as shown below:



```
static void EspLoop()
{
    esp_t = true;
    while(esp_t)
    {
        std::this_thread::sleep_for(std::chrono::milliseconds(1));
        while(g_Base!=0 && c_Base!=0)
        {
            std::this_thread::sleep_for(std::chrono::milliseconds(1));
            if (esp)
            {
                valid = false;

                uint64_t LocalPlayer = 0;
                apex_mem.Read<uint64_t>(g_Base + OFFSET_LOCAL_ENT, LocalPlayer);
                if (LocalPlayer == 0)
                {
                    next = true;
                    while(next && g_Base!=0 && c_Base!=0 && esp)
                    {
                        std::this_thread::sleep_for(std::chrono::milliseconds(1));
                    }
                    continue;
                }
                Entity LPlayer = getEntity(LocalPlayer);
                int team_player = LPlayer.getTeamId();
                if (team_player < 0 || team_player>50)
                {
                    next = true;
                    while(next && g_Base!=0 && c_Base!=0 && esp)
                    {
                        std::this_thread::sleep_for(std::chrono::milliseconds(1));
                    }
                    continue;
                }
                Vector LocalPlayerPosition = LPlayer.getPosition();

                uint64_t viewRenderer = 0;
                apex_mem.Read<uint64_t>(g_Base + OFFSET_RENDER, viewRenderer);
                uint64_t viewMatrix = 0;
                apex_mem.Read<uint64_t>(viewRenderer + OFFSET_MATRIX, viewMatrix);
                Matrix m = {};
                apex_mem.Read<Matrix>(viewMatrix, m);

                uint64_t entitylist = g_Base + OFFSET_ENTITYLIST;

                memset(players,0,sizeof(players));
                if(firing_range)
                {
                    int c=0;
                    for (int i = 0; i < 10000; i++)
                    {
                        uint64_t centity = 0;
```

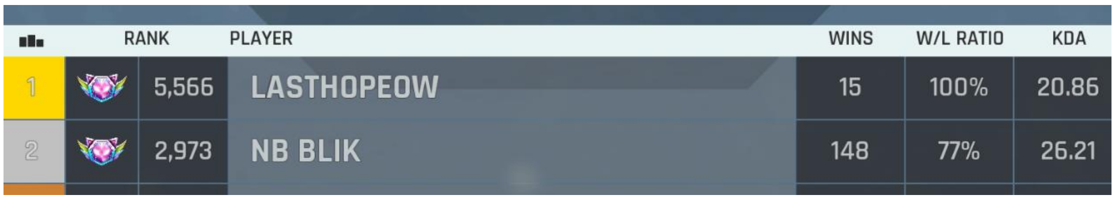
Analysis Conclusion:

1. For a game, the security of its local logic is closely related to the judgment and local security measures of the game server (GS). Based on the current performance of the game, the GS lacks control over synchronized data, and the synchronization of data is incomplete.

Additionally, there are missing user data reports. Therefore, the security rating in this aspect is 0.

Game Protocol & Server Security Analysis

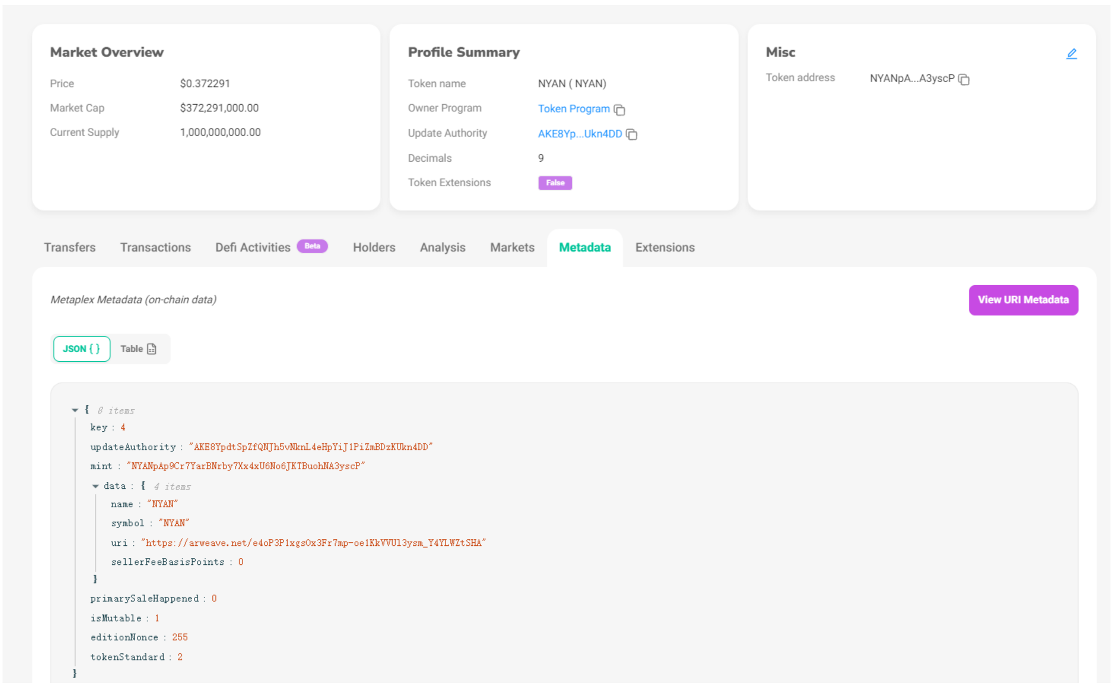
The current game protocol is limited, primarily focused on match settlement. Based on the ranking data, there appear to be some suspicious issues, such as having 15 wins but reaching a rank of 5566. The specific vulnerabilities will not be discussed here.



	RANK	PLAYER	WINS	W/L RATIO	KDA
1	5,566	LASTHOPEOW	15	100%	20.86
2	2,973	NB BLIK	148	77%	26.21

WEB3 Security Analysis：

Nyan token is an standard SPL Token issued on the Solana blockchain, developed using a standardized template. Therefore, the security aspect will not be discussed here.



Market Overview

Price	$0.372291
Market Cap	$372,291,000.00
Current Supply	1,000,000,000.00

Profile Summary

Token name	NYAN (NYAN)
Owner Program	Token Program
Update Authority	AKE8Yp...Ukn4DD
Decimals	9
Token Extensions	False

Misc

Token address	NYANpA...A3yscP

Metaplex Metadata (on-chain data)

```
{ 8 items
  key : 4
  updateAuthority : "AKE8YpdtSpZfQNJh5vNknL4eHpYiJ1PiZmBDzKUkn4DD"
  mint : "NYANpAp9Cr7YarBNrby7Xx4xU6No6JKTBuohNA3yscP"
  data : { 4 items
    name : "NYAN"
    symbol : "NYAN"
    uri : "https://arweave.net/e4oP3P1xgsOx3Fr7mp-oe1KkVVUl3ysm_Y4YLWZtSHA"
    sellerFeeBasisPoints : 0
  }
  primarySaleHappened : 0
  isMutable : 1
  editionNonce : 255
  tokenStandard : 2
}
```

About Damocles

Damocles Labs is a security team established in 2023, specializing in security for the Web3 industry. Their services include contract code auditing, business code auditing, penetration testing, GameFi code auditing, GameFi vulnerability discovery, GameFi cheat analysis, and GameFi anti-cheat measures. They are committed to making continuous efforts in the Web3 security industry, producing as many analysis reports as possible, raising awareness among project owners and users about GameFi security, and promoting the overall security development of the industry.。

Twitter: https://twitter.com/DamoclesLabs

Discord: https://discord.gg/xd6H6eqFHz