Damocles

MetalCore Security Analysis

2024.05.21

Senna

DAMOCLES LABS

Damocles

Contents

Summary

As an STG (Shooting Game) genre, MetalCore has zero security on its client-side. The lack of client protection makes it easy for attackers to perform malicious analysis, leading to imbalances in game mechanics. However, MetalCore exhibits relatively rigorous DS (Data Security) checks compared to other contemporary STG games. Therefore, Damocles rates its security score as 2 stars.

Security Rating: ★★☆☆☆

Game Background

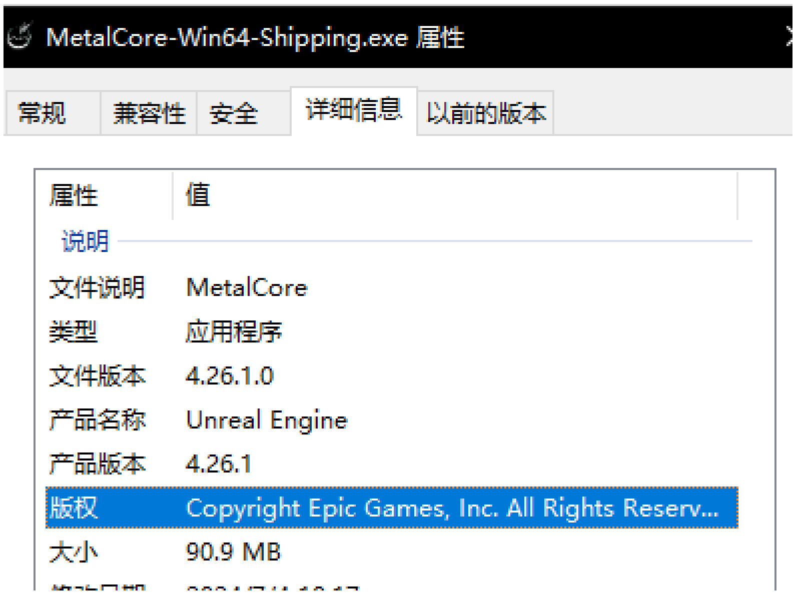
- Game Version Evaluated: 1.0.0.38727
- Game Type & Engine: STG, UE4.26
- Potential Gameplay Issues:
  - Speed hacking (modifying local time rate)
  - No recoil
  - Unlimited ammo
  - Item ESP (seeing items through walls)
  - Modifying local character attributes such as attack speed
  - Character ESP (seeing characters through walls)
  - Bullet tracking
  - Aim assist

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。

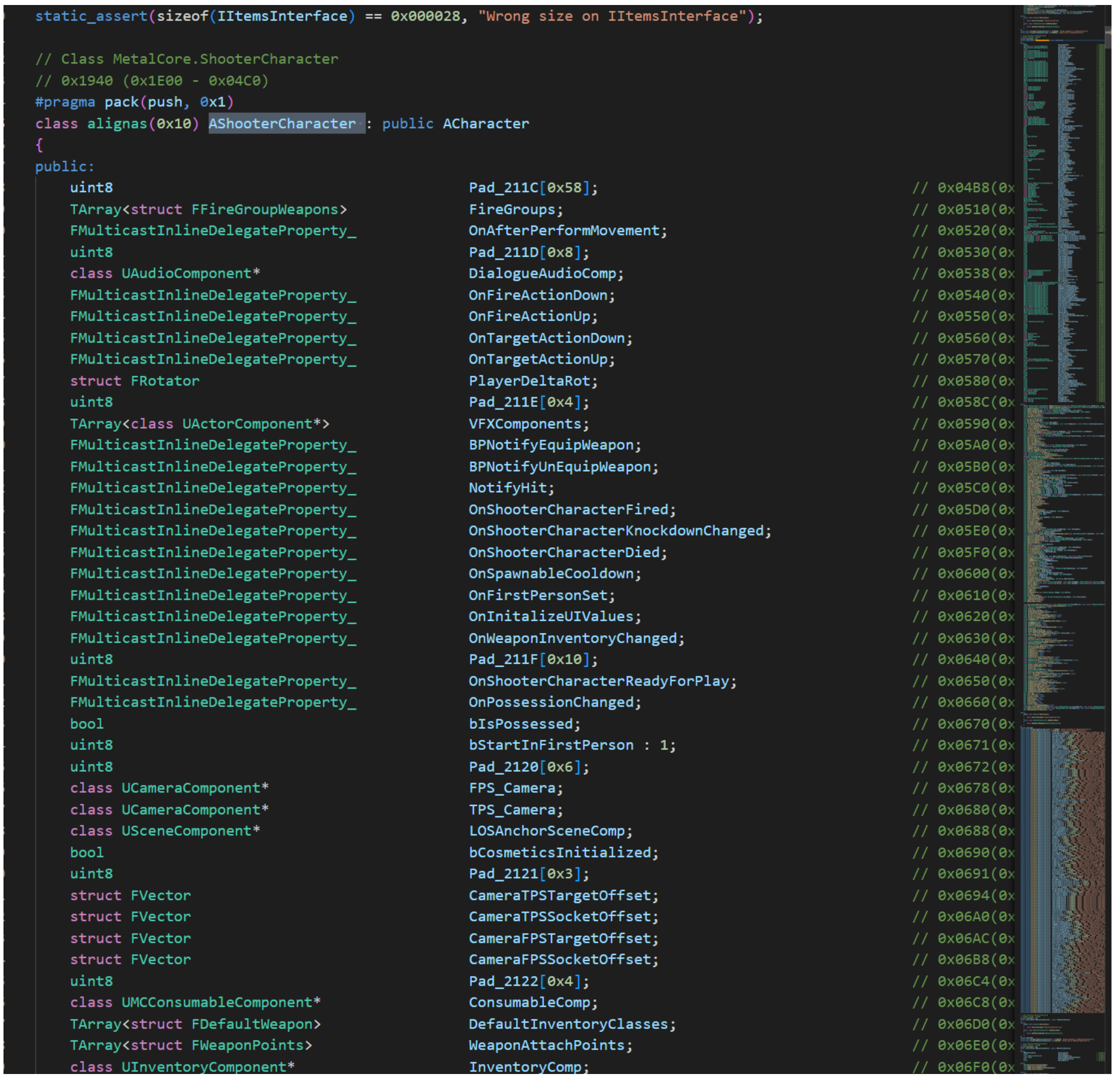


MetalCore-Win64-Shipping.exe 属性

常规 | 兼容性 | 安全 | 详细信息 | 以前的版本

属性	值
说明	
文件说明	MetalCore
类型	应用程序
文件版本	4.26.1.0
产品名称	Unreal Engine
产品版本	4.26.1
版权	Copyright Epic Games, Inc. All Rights Reserv...
大小	90.9 MB

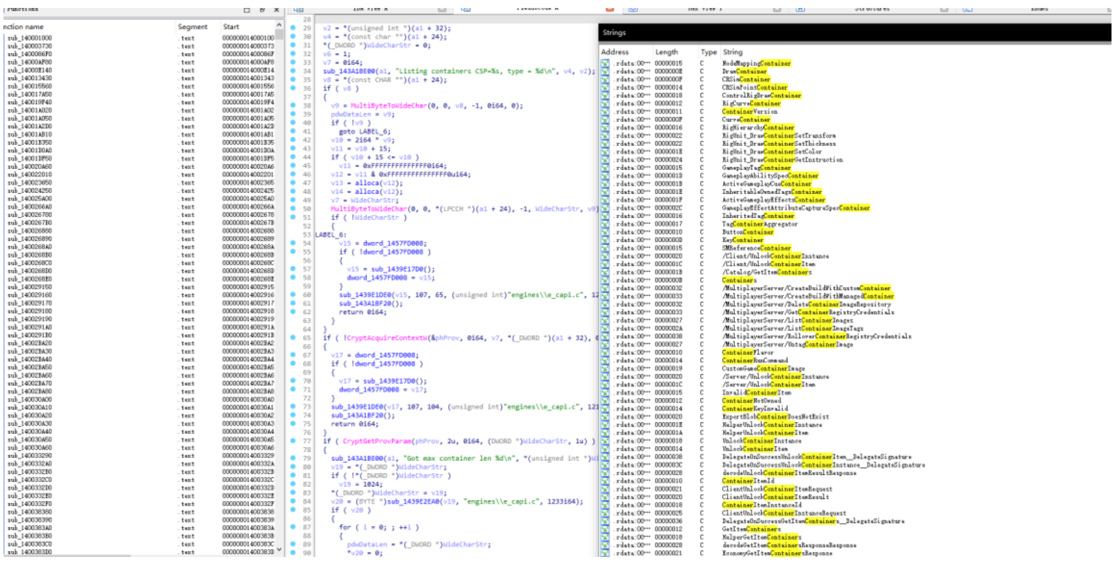
2. Using tools to dump UE character structures for quick location. After locating, we use UE's unique linked list structure for indexing and modification.



```
static_assert(sizeof(IItemsInterface) == 0x000028, "Wrong size on IItemsInterface");

// Class MetalCore.ShooterCharacter
// 0x1940 (0x1E00 - 0x04C0)
#pragma pack(push, 0x1)
class alignas(0x10) AShooterCharacter : public ACharacter
{
public:
    uint8                                         Pad_211C[0x58];                                   // 0x04B8(0x
    TArray<struct FFireGroupWeapons>              FireGroups;                                       // 0x0510(0x
    FMulticastInlineDelegateProperty_             OnAfterPerformMovement;                           // 0x0520(0x
    uint8                                         Pad_211D[0x8];                                    // 0x0530(0x
    class UAudioComponent*                        DialogueAudioComp;                                // 0x0538(0x
    FMulticastInlineDelegateProperty_             OnFireActionDown;                                 // 0x0540(0x
    FMulticastInlineDelegateProperty_             OnFireActionUp;                                   // 0x0550(0x
    FMulticastInlineDelegateProperty_             OnTargetActionDown;                               // 0x0560(0x
    FMulticastInlineDelegateProperty_             OnTargetActionUp;                                 // 0x0570(0x
    struct FRotator                               PlayerDeltaRot;                                   // 0x0580(0x
    uint8                                         Pad_211E[0x4];                                    // 0x058C(0x
    TArray<class UActorComponent*>                VFXComponents;                                    // 0x0590(0x
    FMulticastInlineDelegateProperty_             BPNotifyEquipWeapon;                              // 0x05A0(0x
    FMulticastInlineDelegateProperty_             BPNotifyUnEquipWeapon;                            // 0x05B0(0x
    FMulticastInlineDelegateProperty_             NotifyHit;                                        // 0x05C0(0x
    FMulticastInlineDelegateProperty_             OnShooterCharacterFired;                          // 0x05D0(0x
    FMulticastInlineDelegateProperty_             OnShooterCharacterKnockdownChanged;               // 0x05E0(0x
    FMulticastInlineDelegateProperty_             OnShooterCharacterDied;                           // 0x05F0(0x
    FMulticastInlineDelegateProperty_             OnSpawnableCooldown;                              // 0x0600(0x
    FMulticastInlineDelegateProperty_             OnFirstPersonSet;                                 // 0x0610(0x
    FMulticastInlineDelegateProperty_             OnInitalizeUIValues;                              // 0x0620(0x
    FMulticastInlineDelegateProperty_             OnWeaponInventoryChanged;                         // 0x0630(0x
    uint8                                         Pad_211F[0x10];                                   // 0x0640(0x
    FMulticastInlineDelegateProperty_             OnShooterCharacterReadyForPlay;                   // 0x0650(0x
    FMulticastInlineDelegateProperty_             OnPossessionChanged;                              // 0x0660(0x
    bool                                          bIsPossessed;                                     // 0x0670(0x
    uint8                                         bStartInFirstPerson : 1;                          // 0x0671(0x
    uint8                                         Pad_2120[0x6];                                    // 0x0672(0x
    class UCameraComponent*                       FPS_Camera;                                       // 0x0678(0x
    class UCameraComponent*                       TPS_Camera;                                       // 0x0680(0x
    class USceneComponent*                        LOSAnchorSceneComp;                               // 0x0688(0x
    bool                                          bCosmeticsInitialized;                            // 0x0690(0x
    uint8                                         Pad_2121[0x3];                                    // 0x0691(0x
    struct FVector                                CameraTPSTargetOffset;                            // 0x0694(0x
    struct FVector                                CameraTPSSocketOffset;                            // 0x06A0(0x
    struct FVector                                CameraFPSTargetOffset;                            // 0x06AC(0x
    struct FVector                                CameraFPSSocketOffset;                            // 0x06B8(0x
    uint8                                         Pad_2122[0x4];                                    // 0x06C4(0x
    class UMCConsumableComponent*                 ConsumableComp;                                   // 0x06C8(0x
    TArray<struct FDefaultWeapon>                 DefaultInventoryClasses;                          // 0x06D0(0x
    TArray<struct FWeaponPoints>                  WeaponAttachPoints;                               // 0x06E0(0x
    class UInventoryComponent*                    InventoryComp;                                    // 0x06F0(0x
```

Using IDA for static code analysis, we found that local strings are not encrypted and the code is not obfuscated.



```
Functions
nction name                    Segment   Start
sub_140001000                  .text     000000014000100
sub_140003730                  .text     000000014000373
sub_1400086F0                  .text     00000001400086F
sub_14000AF80                  .text     000000014000AF8
sub_14000E140                  .text     000000014000E14
sub_140013430                  .text     000000014001343
sub_140015560                  .text     000000014001556
sub_140017A50                  .text     0000000140017A5
sub_140019F40                  .text     0000000140019F4
sub_14001A020                  .text     000000014001A02
sub_14001A050                  .text     000000014001A05
sub_14001A2D0                  .text     000000014001A2D
sub_14001AB10                  .text     000000014001AB1
sub_14001B350                  .text     000000014001B35
sub_14001D0A0                  .text     000000014001D0A
sub_14001DF50                  .text     000000014001DF5
sub_140020A60                  .text     0000000140020A6
sub_140022010                  .text     000000014002201
sub_140023650                  .text     000000014002365
sub_140024250                  .text     000000014002425
sub_140025A00                  .text     0000000140025A0
sub_1400266A0                  .text     00000001400266A
sub_140026780                  .text     000000014002678
sub_1400267B0                  .text     00000001400267B
sub_140026880                  .text     000000014002688
sub_140026890                  .text     000000014002689
sub_1400268A0                  .text     00000001400268A
sub_1400268B0                  .text     00000001400268B
sub_1400268C0                  .text     00000001400268C
sub_1400268D0                  .text     00000001400268D
sub_1400268E0                  .text     00000001400268E
sub_140029150                  .text     000000014002915
sub_140029160                  .text     000000014002916
sub_140029170                  .text     000000014002917
sub_140029180                  .text     000000014002918
sub_140029190                  .text     000000014002919
sub_1400291A0                  .text     00000001400291A
sub_1400291B0                  .text     00000001400291B
sub_14002BA20                  .text     000000014002BA2
sub_14002BA30                  .text     000000014002BA3
sub_14002BA40                  .text     000000014002BA4
sub_14002BA50                  .text     000000014002BA5
sub_14002BA60                  .text     000000014002BA6
sub_14002BA70                  .text     000000014002BA7
sub_14002BA80                  .text     000000014002BA8
sub_140030A00                  .text     0000000140030A0
sub_140030A10                  .text     0000000140030A1
sub_140030A20                  .text     0000000140030A2
sub_140030A30                  .text     0000000140030A3
sub_140030A40                  .text     0000000140030A4
sub_140030A50                  .text     0000000140030A5
sub_140030A60                  .text     0000000140030A6
sub_140033290                  .text     000000014003329
sub_1400332A0                  .text     00000001400332A
sub_1400332B0                  .text     00000001400332B
sub_1400332C0                  .text     00000001400332C
sub_1400332D0                  .text     00000001400332D
sub_1400332E0                  .text     00000001400332E
sub_1400332F0                  .text     00000001400332F
sub_140038380                  .text     000000014003838
sub_140038390                  .text     000000014003839
sub_1400383A0                  .text     00000001400383A
sub_1400383B0                  .text     00000001400383B
sub_1400383C0                  .text     00000001400383C
sub_1400383D0                  .text     00000001400383D
```

```c

  v2 = *(unsigned int *)(a1 + 32);
  v4 = *(const char **)(a1 + 24);
  *(_DWORD *)WideCharStr = 0;
  v6 = 1;
  v7 = 0i64;
  sub_143A1BE00(a1, "Listing containers CSP=%s, type = %d\n", v4, v2);
  v8 = *(const CHAR **)(a1 + 24);
  if ( v8 )
  {
    v9 = MultiByteToWideChar(0, 0, v8, -1, 0i64, 0);
    pdwDataLen = v9;
    if ( !v9 )
      goto LABEL_6;
    v10 = 2i64 * v9;
    v11 = v10 + 15;
    if ( v10 + 15 <= v10 )
      v11 = 0xFFFFFFFFFFFFFFF0i64;
    v12 = v11 & 0xFFFFFFFFFFFFFFF0ui64;
    v13 = alloca(v12);
    v14 = alloca(v12);
    v7 = WideCharStr;
    MultiByteToWideChar(0, 0, *(LPCCH *)(a1 + 24), -1, WideCharStr, v9)
    if ( !WideCharStr )
    {
LABEL_6:
      v15 = dword_1457FD008;
      if ( !dword_1457FD008 )
      {
        v15 = sub_1439E17D0();
        dword_1457FD008 = v15;
      }
      sub_1439E1DE0(v15, 107, 65, (unsigned int)"engines\\e_capi.c", 12
      sub_143A1BF20();
      return 0i64;
    }
  }
  if ( !CryptAcquireContextW(&phProv, 0i64, v7, *(_DWORD *)(a1 + 32), 0
  {
    v17 = dword_1457FD008;
    if ( !dword_1457FD008 )
    {
      v17 = sub_1439E17D0();
      dword_1457FD008 = v17;
    }
    sub_1439E1DE0(v17, 107, 104, (unsigned int)"engines\\e_capi.c", 121
    sub_143A1BF20();
    return 0i64;
  }
  if ( CryptGetProvParam(phProv, 2u, 0i64, (DWORD *)WideCharStr, 1u) )
  {
    sub_143A1BE00(a1, "Got max container len %d\n", *(unsigned int *)Wi
    v19 = *(_DWORD *)WideCharStr;
    if ( !*(_DWORD *)WideCharStr )
      v19 = 1024;
    *(_DWORD *)WideCharStr = v19;
    v20 = (BYTE *)sub_1439E2EA0(v19, "engines\\e_capi.c", 1233i64);
    if ( v20 )
    {
      for ( i = 0; ; ++i )
      {
        pdwDataLen = *(_DWORD *)WideCharStr;
        *v20 = 0;
```

```
Strings
Address         Length    Type  String
.rdata:00···    00000015  C     NodeMappingContainer
.rdata:00···    0000000E  C     DrawContainer
.rdata:00···    0000000F  C     CRSimContainer
.rdata:00···    00000014  C     CRSimPointContainer
.rdata:00···    00000018  C     ControlRigDrawContainer
.rdata:00···    00000012  C     RigCurveContainer
.rdata:00···    00000011  C     ContainerVersion
.rdata:00···    0000000F  C     CurveContainer
.rdata:00···    00000016  C     RigHierarchyContainer
.rdata:00···    00000022  C     RigUnit_DrawContainerSetTransform
.rdata:00···    00000022  C     RigUnit_DrawContainerSetThickness
.rdata:00···    0000001E  C     RigUnit_DrawContainerSetColor
.rdata:00···    00000024  C     RigUnit_DrawContainerGetInstruction
.rdata:00···    00000015  C     GameplayTagContainer
.rdata:00···    0000001B  C     GameplayAbilitySpecContainer
.rdata:00···    0000001B  C     ActiveGameplayCueContainer
.rdata:00···    0000001E  C     InheritableOwnedTagsContainer
.rdata:00···    0000001F  C     ActiveGameplayEffectsContainer
.rdata:00···    0000002C  C     GameplayEffectAttributeCaptureSpecContainer
.rdata:00···    00000016  C     InheritedTagContainer
.rdata:00···    00000017  C     TagContainerAggregator
.rdata:00···    00000010  C     ButtonContainer
.rdata:00···    0000000D  C     KeyContainer
.rdata:00···    00000015  C     SMReferenceContainer
.rdata:00···    00000020  C     /Client/UnlockContainerInstance
.rdata:00···    0000001C  C     /Client/UnlockContainerItem
.rdata:00···    0000001B  C     /Catalog/GetItemContainers
.rdata:00···    0000000B  C     Containers
.rdata:00···    00000032  C     /MultiplayerServer/CreateBuildWithCustomContainer
.rdata:00···    00000033  C     /MultiplayerServer/CreateBuildWithManagedContainer
.rdata:00···    00000032  C     /MultiplayerServer/DeleteContainerImageRepository
.rdata:00···    00000033  C     /MultiplayerServer/GetContainerRegistryCredentials
.rdata:00···    00000027  C     /MultiplayerServer/ListContainerImages
.rdata:00···    0000002A  C     /MultiplayerServer/ListContainerImageTags
.rdata:00···    00000038  C     /MultiplayerServer/RolloverContainerRegistryCredentials
.rdata:00···    00000027  C     /MultiplayerServer/UntagContainerImage
.rdata:00···    00000010  C     ContainerFlavor
.rdata:00···    00000014  C     ContainerRunCommand
.rdata:00···    00000019  C     CustomGameContainerImage
.rdata:00···    00000020  C     /Server/UnlockContainerInstance
.rdata:00···    0000001C  C     /Server/UnlockContainerItem
.rdata:00···    00000015  C     InvalidContainerItem
.rdata:00···    00000012  C     ContainerNotOwned
.rdata:00···    00000014  C     ContainerKeyInvalid
.rdata:00···    00000020  C     ExportBlobContainerDoesNotExist
.rdata:00···    0000001E  C     HelperUnlockContainerInstance
.rdata:00···    0000001A  C     HelperUnlockContainerItem
.rdata:00···    00000018  C     UnlockContainerInstance
.rdata:00···    00000014  C     UnlockContainerItem
.rdata:00···    00000038  C     DelegateOnSuccessUnlockContainerItem__DelegateSignature
.rdata:00···    0000003C  C     DelegateOnSuccessUnlockContainerInstance__DelegateSignature
.rdata:00···    00000028  C     decodeUnlockContainerItemResultResponse
.rdata:00···    00000010  C     ContainerItemId
.rdata:00···    00000021  C     ClientUnlockContainerItemRequest
.rdata:00···    00000020  C     ClientUnlockContainerItemResult
.rdata:00···    00000018  C     ContainerItemInstanceId
.rdata:00···    00000025  C     ClientUnlockContainerInstanceRequest
.rdata:00···    00000036  C     DelegateOnSuccessGetItemContainers__DelegateSignature
.rdata:00···    00000012  C     GetItemContainers
.rdata:00···    00000018  C     HelperGetItemContainers
.rdata:00···    00000028  C     decodeGetItemContainersResponseResponse
.rdata:00···    00000021  C     EconomyGetItemContainersResponse
```

Therefore, using the dumped UE data structure, we can use decompilation tools to understand the basic logic of the game code.

Analysis Conclusion:

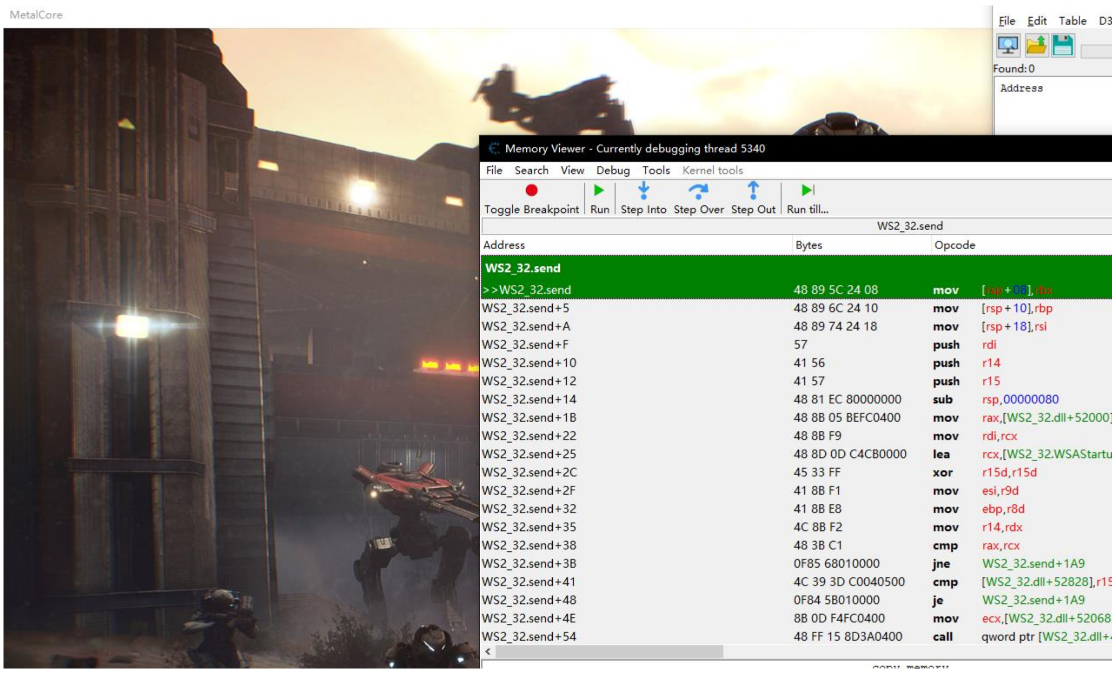
MetalCore scores 0 points in game code protection. Its client code has no protection, and strings are not encrypted, making it easy to analyze the game quickly using data structures and strings.

Damocles

Game Basic Anti-Cheat:

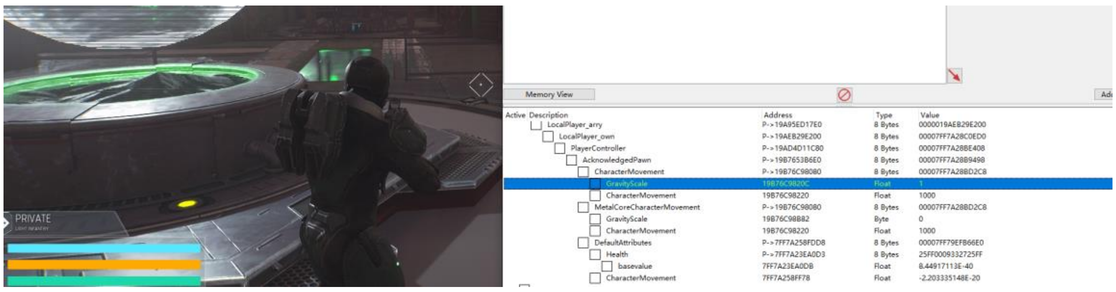
Analysis Process:

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..



```
WS2_32.send
Address                 Bytes                 Opcode
WS2_32.send
>>WS2_32.send           48 89 5C 24 08        mov   [rsp+08],rbx
WS2_32.send+5           48 89 6C 24 10        mov   [rsp+10],rbp
WS2_32.send+A           48 89 74 24 18        mov   [rsp+18],rsi
WS2_32.send+F           57                    push  rdi
WS2_32.send+10          41 56                 push  r14
WS2_32.send+12          41 57                 push  r15
WS2_32.send+14          48 81 EC 80000000     sub   rsp,00000080
WS2_32.send+1B          48 8B 05 BEFC0400     mov   rax,[WS2_32.dll+52000]
WS2_32.send+22          48 8B F9              mov   rdi,rcx
WS2_32.send+25          48 8D 0D C4CB0000     lea   rcx,[WS2_32.WSAStartu
WS2_32.send+2C          45 33 FF              xor   r15d,r15d
WS2_32.send+2F          41 8B F1              mov   esi,r9d
WS2_32.send+32          41 8B E8              mov   ebp,r8d
WS2_32.send+35          4C 8B F2              mov   r14,rdx
WS2_32.send+38          48 3B C1              cmp   rax,rcx
WS2_32.send+3B          0F85 68010000         jne   WS2_32.send+1A9
WS2_32.send+41          4C 39 3D C0040500     cmp   [WS2_32.dll+52828],r15
WS2_32.send+48          0F84 5B010000         je    WS2_32.send+1A9
WS2_32.send+4E          8B 0D F4FC0400        mov   ecx,[WS2_32.dll+52068
WS2_32.send+54          48 FF 15 8D3A0400     call  qword ptr [WS2_32.dll+
```

3. We could directly modify in-game character attributes like speed and jump height, and DS did not kick out the player, although speed modifications caused a rollback.



```
Active Description                          Address            Type     Value
  LocalPlayer_arry                          P->19A95ED17E0     8 Bytes  0000019AEB29E200
    LocalPlayer_own                         P->19AEB29E200     8 Bytes  00007FF7A28C0ED0
      PlayerController                      P->19AD4D11C80     8 Bytes  00007FF7A28BE408
        AcknowledgedPawn                    P->19B7653B6E0     8 Bytes  00007FF7A28B9498
          CharacterMovement                 P->19B76C98080     8 Bytes  00007FF7A28BD2C8
            GravityScale                    19B76C9820C        Float    1
            CharacterMovement               19B76C98220        Float    1000
          MetalCoreCharacterMovement        P->19B76C98080     8 Bytes  00007FF7A28BD2C8
            GravityScale                    19B76C98B82        Byte     0
            CharacterMovement               19B76C98220        Float    1000
          DefaultAttributes                 P->7FF7A258FDD8    8 Bytes  00007FF79EFB66E0
            Health                          P->7FF7A23EA0D3    8 Bytes  25FF0009332725FF
              basevalue                     7FF7A23EA0DB       Float    8.44917113E-40
            CharacterMovement               7FF7A258FF78       Float    -2.203335148E-20
```

4. Weapon modifications, however, were effective locally, enabling unlimited ammo and no recoil.

Analysis Conclusion:

1. MetalCore has almost zero protection against cheats, lacking dynamic debugging and analysis countermeasures. For malicious players, the cost is minimal, and the game lacks detection capabilities for cheats.
2. We only tested anti-debugging and read/write protection because, for creating cheats, finding data and implementing features only requires debugging and read/write capabilities. If these basic protections are missing, advanced protections like injection and hook detection are meaningless.

Game Logic Issues

Analysis Process:

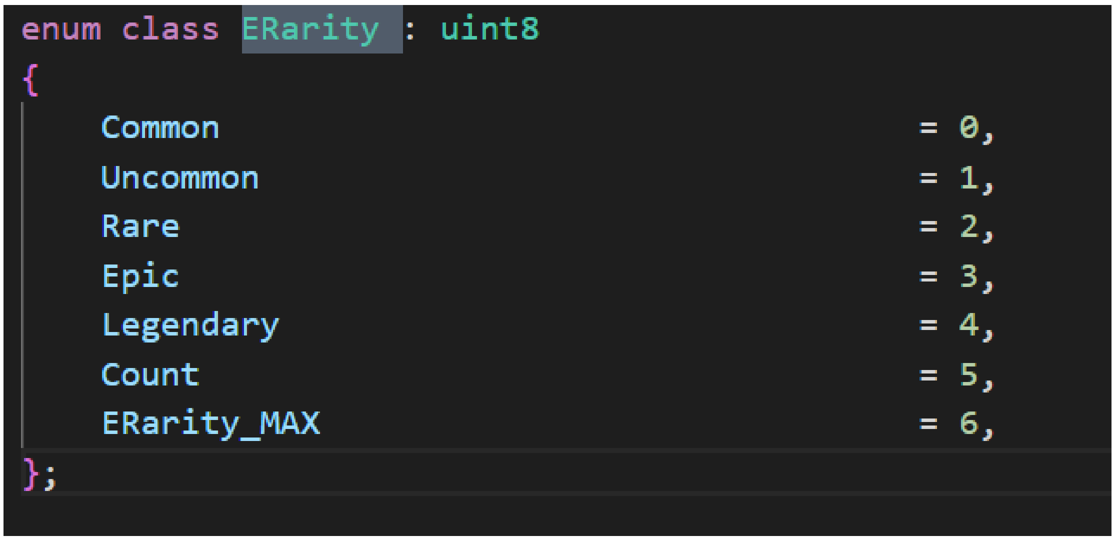
While analyzing MetalCore, we found that the main profit modes for players are:

- ✓ ***Completing tasks to earn airdrop tokens.***
- ✓ ***Searching for boxes on the map to collect materials for NFT minting.***

Cheat developers typically target these high-profit areas, resulting in:

- ✓ ***Aimbot, ESP, bullet tracking, and weapon attribute modifications for PVPVE.***
- ✓ ***High-value item location for quick profit.***

We also found that the game data structure exposed enumeration types for item rarity.



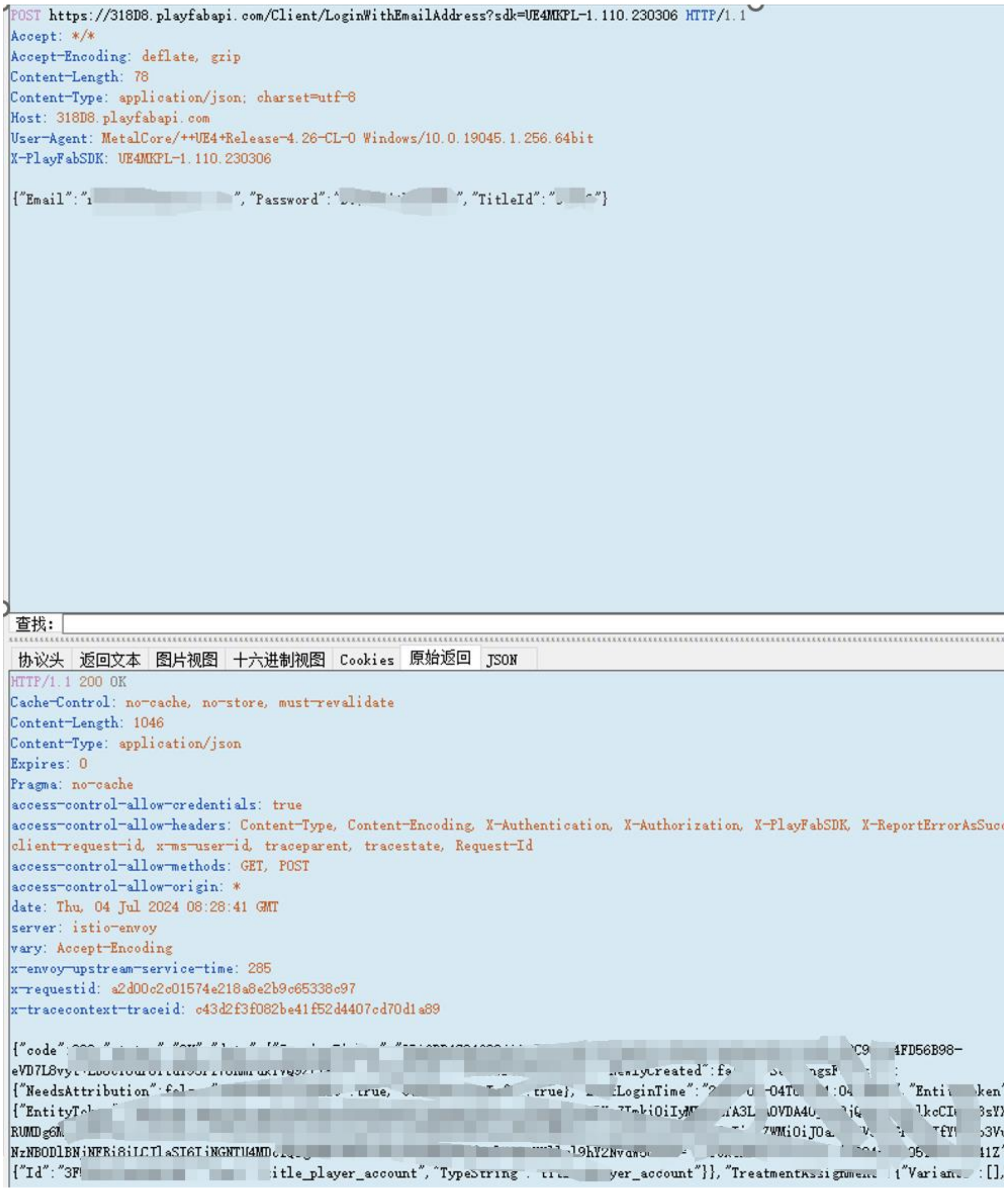
```
enum class ERarity : uint8
{
    Common                                          = 0,
    Uncommon                                        = 1,
    Rare                                            = 2,
    Epic                                            = 3,
    Legendary                                       = 4,
    Count                                           = 5,
    ERarity_MAX                                     = 6,
};
```

Analysis Conclusion:

1. Both methods that affect game/economic balance can be quickly implemented. Considering the current project lacks proactive security measures, we deem the cheat risk for the game's main logic as extremely high, giving it a security rating of 0.

Game Protocol & Server Security Analysis

The current game protocol is minimal and incomplete. We only briefly analyzed the login logic. During analysis, we found that the project transmits usernames and passwords in plaintext, which is inadvisable and should be corrected.



```
POST https://318D8.playfabapi.com/Client/LoginWithEmailAddress?sdk=UE4MKPL-1.110.230306 HTTP/1.1
Accept: */*
Accept-Encoding: deflate, gzip
Content-Length: 78
Content-Type: application/json; charset=utf-8
Host: 318D8.playfabapi.com
User-Agent: MetalCore/++UE4+Release-4.26-CL-0 Windows/10.0.19045.1.256.64bit
X-PlayFabSDK: UE4MKPL-1.110.230306

{"Email":"[redacted]","Password":"[redacted]","TitleId":"[redacted]"}
```

查找:

协议头 返回文本 图片视图 十六进制视图 Cookies 原始返回 JSON

```
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store, must-revalidate
Content-Length: 1046
Content-Type: application/json
Expires: 0
Pragma: no-cache
access-control-allow-credentials: true
access-control-allow-headers: Content-Type, Content-Encoding, X-Authentication, X-Authorization, X-PlayFabSDK, X-ReportErrorAsSucc
client-request-id, x-ms-user-id, traceparent, tracestate, Request-Id
access-control-allow-methods: GET, POST
access-control-allow-origin: *
date: Thu, 04 Jul 2024 08:28:41 GMT
server: istio-envoy
vary: Accept-Encoding
x-envoy-upstream-service-time: 285
x-requestid: a2d00c2c01574e218a8e2b9c65338c97
x-tracecontext-traceid: c43d2f3f082be41f52d4407cd70d1a89

{"code":[redacted]C9[redacted]4FD56B98-
eVD7L8vy[redacted]ewlyCreated":fa[redacted]
{"NeedsAttribution":[redacted]true,[redacted]true},[redacted]LoginTime":"[redacted]-04T[redacted]:04[redacted]"Enti[redacted]ken"
{"EntityT[redacted]
RUMDg6M[redacted]
NzNBODlBNjNFRi8iLCJlaSI6IjNGNTU4MD[redacted]
{"Id":"3F[redacted]itle_player_account","TypeString[redacted]yer_account"}},"TreatmentAssignment[redacted]{"Varian[redacted]:[],
```

WEB3 Security Analysis：

Symbol	MCG
TotalSupply	**3,000,000,000**
Address	0xc9E503562d0Db0A2629288a5D3b1c94Ea7741869

The standard ERC20 token contract uses Immutable wrapper for implementation with a fixed total supply.

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