UDC004.056

PROTECTION OF THE DATA OF THE APPLICATIONS DEVELOPED ON UNITY 3D

ARTSEMI KARNILOVICH, DMITRY PASTUHOV
Polotsk State University, Belarus

This article discusses how to obtain unauthorized access to thesaving data and resources of applications developed on Unity3d, as well as possible ways to solve this problem.

Unity3D is a cross-platform game engine from Unity Technologies. Unity allows you to create applications that run on more than 20 different operating systems. The engine supports many popular formats of models, sounds, materials and textures.

The project in Unity divided into scenes (levels) - separate files containing their game worlds with their own set of objects, scenarios, and settings. Scenes can contain both, in fact, objects (models) and empty game objects - objects that do not have a model. Objects, in turn, contain sets of components with which scripts interact.

All these data must be protected from copying, illegal use, pirated distribution, professional analysis and hacking.

When compiling a project Unity creates an executable game file, and in a separate folder - game data (including all game levels and dynamic link libraries).

Source code. For Windows builds, Unity compiles and saves the source code of all game scripts in the Managed directory. The code is in the libraries: Assembly-CSharp.dll, Assembly-CSharp-firstpass.dll and Assembly-UnityScript.dll. For decompiling and viewing managed code of .NET libraries there are quite convenient and at the same time free utilities: IlSpy and dotPeek.

The data approach is especially effective for our purposes: Unity very sparingly optimizes the source code of game scripts, practically without changing its structure, and does not hide the names of variables. This makes it easy to read and understand decompiled material.

In such cases, developers have to worry about the security of their code. For such purposes, usually use obfuscators.

The code obfuscation is a mechanism for hiding the original algorithm, data structures or the logic of the code, or to harden or protect the code from the unauthorized reverse engineering process. In general, code obfuscation involves hiding a program's implementation details from an adversary, i.e. transforming the program into a semantically equivalent (same computational effect) program, which is much harder to understand for an attacker. None of the current code obfuscation techniques satisfies all the obfuscation effectiveness criteria to resistance the reverse engineering attacks. Therefore, the researchers as well as the software industries are trying their best to apply newer and better obfuscation techniques over their intellectual property in a regular process. But unfortunately, software code is not safe, i.e. still it can be cracked. In especially unpleasant circumstances, obfuscator can make the program completely unsuitable for execution, in less severe cases new errors may appear in the program. Therefore, obfuscation should be applied with maximum caution.

In AssetStore there are many ready-made solutions, but most of them are paid. Free versions, as a rule, are limited or have low efficiency.

Project resources. Most Unity project resources packaged in proprietary format files with the .assets and .resources extensions. Despite the closeness of the formats, there are tools for unpacking such files. For example, Unity Assets Explorer is able to extract most textures and shaders from the game. The resulting textures will be in DDS format, which can be opened using the Windows Texture Viewer. Shaders extracted in the already compiled form and there are no solutions for their decompilation. However, this circumstance does not prevent the import and use of the resulting shaders in another Unity-project.

Three-dimensional models in the Unity assembly are located in various resources, and some of them can be generated at all during the game. You can get this data straight from the GPU memory. When the game is running, all the information about the textures and models visible on the screen is in the memory of the video card. Using the 3D Ripper DX utility, you can extract all this information and save it in a format understandable to 3D editors.

Stored data. PlayerPrefs is a class from the standard Unity library that allows you to save data in the long-term memory of the device. It is a pair of key - value. Developers to store various settings, achievements, player progress, and other information about the state of the game often use it. On Windows, this data is stored in the system registry. Other operating systems, data is stored on the device in the local folder of the application in a special file. In most cases, they can be easily accessed and modified using a text editor. For example, in Windows it is enough to use the built-in utility RegEdit to modify any values of PlayerPrefs, thereby changing the configuration and status of the game.

The easiest way to counteract is to encode the stored data, for example, base 64. This method is not very effective, but can provide initial protection against viewing. Checking whether the data was changed without the knowledge of the hash functions will help: by comparing the checksums of the stored data, we can make sure that nobody and nothing except our code has changed this data.

It is necessary to use encryption or various combinations of the listed methods for more reliable protection.

You can also implement your own save format. Thanks Mono Unity supports work with the file system. Thus, you can serialize all the necessary data, apply strong encryption and save in a safe place.

RAM. Cheat Engine - a well-known program for hacking memory. It finds the area of RAM that belongs to the process of the running application and allows you to arbitrarily change it. Over several iterations of screening, you can easily find the location of most game variables and arbitrarily change them.

This program takes advantage of the fact that developers rarely protect variable values. Protection from such programs is quite simple - you need to encrypt the values in the application's memory. Encrypt each time you write and decrypt each time you read. Since the operation is quite frequent, we need a very fast algorithm. For this, for example, an encryption algorithm based on XOR and base 64 may be suitable. Since there will be work with data in RAM, there is no need to save the key, for this you can use the session key generated just before the operation.

However, a more effective method would be to store critical values immediately in the long-term memory of the device. Playerprefs allows you to do this quite quickly and simply, but, again, you need to take care of the safety of this data.

Digital rights protection. Digital Rights Management or DRM is a scheme that controls access to copyrighted material using technological means. It may refer to the usage of proprietary software, hardware, or any type of content: music tracks, video files, ebooks, games, DVD movies, emails, documents, etc.The purpose of DRM is to prevent unauthorized redistribution of digital media and restrict the ways consumers can copy content they have purchased. DRM products were developed in response to the rapid increase in online piracy of commercially marketed material, which proliferated through the widespread use of peer-to-peer file exchange programs. Typically, DRM is implemented by embedding code that prevents copying, specifies a period in which the content can be accessed or limits the number of devices the media can be installed on.

Although copyright laws protect digital content, policing the Web and catching law-breakers is very difficult. DRM technology focuses on making it impossible to steal content in the first place, a more efficient approach to the problem than the hit-and-miss strategies aimed at apprehending online poachers after the fact.

Unfortunately, there are not so many ways to protect the game from hacking. Being installed on the user device, it actually reveals all the textures, models and source code. If someone wants to decompile the application and steal resources - it is only a matter of time.

Despite this, there are effective methods that will seriously complicate the lives of hackers.

REFERENCES

1. Guide to Storage Encryption Technologies for End User Devices / Karen Scarfone, Murugiah Souppaya, Matt Sexton. – November 2007. – 40 p.
2. Unity – Scripting API [Electronic resource]. –Mode of access: https://docs.unity3d.com/ScriptReference/. – Date of access: 22.01.2019.
3. Практическое руководство по взлому (и защите) игр на Unity [Электронныйресурс]. –Режим доступа: https://habr.com/ru/post/266345/. –Дата доступа: 22.01.2019.