Steam and freeware Japanese skill [Part 2].

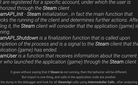
September 23, 2019

The second global stage will be a decrypting from **Steam**.
Let's open Ilsen during a **UPX Explorer** is not loud as the **Import Directory**.

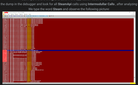
Specifically, we are interested in the **steam_api.dll** module.
Here, only 5 functions are called from it, the essence of which I will try to describe.

SteamAPI_RunCallbacks - a function that calls directly to the Steam client itself and receives a response from it (callback).
SteamAPIInit - a function that receives a list of applications (games) from the Steam client, checks the current user which the user is authorized through the Steam client.
SteamAPI_Init - Steam initialization, in fact the main function that checks the running of the client and determines further actions. After calling it, the Steam client will consider that the application (game) is running.
SteamAPI_Shutdown - is a finalization function that is called upon completion of the process and is a signal to the Steam client that the application (game) has ended.
SteamUser - is a function that receives information about the current user who launched the application (game) through the Steam client.

It goes without saying that if **Steam** is not running, then the behavior will be different.
But forget it for one thing, and calm it the usual way - another.
Let's put the mark in the debugger and look for all **Steam**api calls using **Immunedbe Calls**, after analyzing the code
we will see that **Steam** will return the functions.



Put already set soft breakpoints, put plus **F9** and watch the behavior.
SteamAPI_Init is called **1 time**,
SteamAPI_Shutdown is called **1 time**.



Now suppose that it happened, but at the same time it is worth adjusting the results of the research, in your review
Steam Stub is the most instructive and change its response **SteamAPI_Init** is called.

We continue to launch,
SteamAPI_RunCallbacks is called **1 time**.

We can't have any pain callbacks, we can do a lot of unnecessary launching,
to stop, it will be called about 5 times.

Let's wait until the time expires **SteamUser**.

The further fun is that we need to change the remission value **EAX**, otherwise the main call will cause, throwing a chain
of calls.
Move on to the next instructions and change the value of the **EAX** register to **0**.
Continuing the launch, we are again met by **RunCallbacks**, only it is called a much larger number of times.
We can't have any pain callbacks, we can do a lot of unnecessary launching,

Let's unscrewed, we will be the last screen, Ilsen, and the family comes out and we will have
the game.

We achieve the second ending here.
Yaoi, Ilsen is not a man, she like to drink on the night
and we go into the character selection tab.

Select any of them and press the green button below.

We are pressed one another, challenge, the theme is **Steamtown**.

We will have to stop this call, and add it some part of the procedure.
We put it **new origin** in the addition.

Ask why? because at normal starting
at the address **0140D2D8** a function from the **steamclient.dll** library will be called , and outside the **steam** it will break
further execution flow.

Let's release the free swimming game and wait for further events.
Nothing more to describe from cracks.
Let's go back to the main menu.
We have already put the key in advance using **Steamtown**.

If you use the **Steam** or **Steam Stub** in the **Steam** folder, then it will be in the **Steam**folder, **steam_api.dll**
steam_api64.dll.

The process is over, good luck.

From the point of view of cookies, something will hurt the map.

But there is one "but", namely the value of the **DllCharacteristics** field
instead of 0140 it has changed to 0100.
If you do not change this, then **UPX Explorer** will be crashing.

Let's go back to what we began.
Let's completely zero the field in advance, since **UPX Explorer** does not know how to interpret the first big dump, which
is big by distribution.
Open the **Optional Header** and insert the **00 00 hex** here.

Now we can save it and run the **PE-bear** utility.

Now everything looks to the

There are a couple of cells left to do.
We can hunt them in **UPX Explorer**.

Let's put the zero value, namely **7 and 8 in** the compiler stamp in **0000**.

Now we can proceed to the last step, we will change the **Checksum**, we will change it to 0.

Rebuild and save the file.

Let's check if it works, we do not need to change anything.

This completes the unpacking and the file is ready for further analysis.