

Imports Fixer Plugin Help File

We'll see in this section a detailed explanation about plugins and how they can be implemented in IF.

1- What is a plugin ?

A plugin is an external file (Dynamic Link Library) which can be used directly by the appropriate program "HOST". A plugin can be written in any available language and it is intended to bring extra features or to carry out additional tasks. To be recognised by the HOST, the plugin must respect a certain syntax which is imposed by the author of the HOST or even use a particular programming language. A plugin can contain only functions and procedures or it can include resources (like skins) or can be an independant DialogBox or sub-program into the big one (HOST).

2- Why not include directly plugins features in the HOST instead of using plugins ?

Plugins are very helpful and easy to use. Indeed, they are complementary to the HOST and can be used without installation. In a practical view, they make third-party coding possible and can be updated separately from the HOST. This way, the author doesn't have to update the whole release and lots of people will be contributing to the same program. Seeing that the author can't code everything in its HOST, it's an opportunity for other coders to implement new ideas and show their talent and creativity, the HOST will be then in permanent update.

3- Why plugins are used by IF ?

As you know, IF has its own algorithm to trace, find and resolve IAT entries. Thus IF can resolve IAT functions in protected apps without IAT redirection or elimination. Unfortunately, this isn't sufficient due to the variety of protectors and IAT redirection tricks. Consequently, plugins will be used to complete the work of IF and resolve the remaining invalid thunks left unresolved by IF. In further versions, other types of plugins might be supported, I'm talking about IAT Auto Search plugins (for specific protected targets which IF fails to detect correctly the IAT location) and eventually IF skins.

4- What language must be used to write IF plugins ?

You can use any language you're used to. The most important thing will be the syntax and the way the target is opened. Of course, some of you who have already written plugins for ImpRec will ask if their plugins are still operational in IF. Right now it's impossible because ImpRec and IF are using different approaches while dealing with targets. IF uses the "ReadProcessMemory" API in its interventions whereas ImpRec uses the "MapViewOfFile" API. Also the parameters passed to Resolve function we'll be discussing in next part are different from ImpRec's one.

5- What is then the syntax to follow to write a compatible IF plugin ?

You have to declare a function named "Resolve". You will have to respect the following syntax and parameters :

```
function  
Resolve(hProcess:DWORD;ImageBase:DWORD;Size:DWORD;VA:DWORD;Value:DWORD):D  
WORD;
```

The previous declaration is written using Delphi, but there is no difference if you use another language. For example C declaration would be :

```
DWORD Resolve(DWORD hProcess, DWORD ImageBase, DWORD Size, DWORD VA,
DWORD Value);
```

Now here's a little explanation of the parameters passed to this function :

- hProcess : It's the process handle of the target file. You will use the hProcess in ReadProcessMemory parameters.
- ImageBase : It's the ImageBase of the target. You will use this in ReadProcessMemory parameters.
- Size : It's the SizeOfImage in memory of the target. You will use this in ReadProcessMemory parameters.
- VA : It's the Virtual Address of the invalid thunk to resolve.
- Value : The value pointed by VA and needs to be corrected.

Why have VA and Value together in parameters if we know that VA is always pointing to Value? Well Value parameter has been added since version 1.6 to support fixing direct calls. You have to check in your plugin code whether VA is null or not. If VA=0 then the thunk is a direct call, otherwise it is a normal thunk.

If all is OK the function return value must be the valid VA of the API. Example : If the API is GetModuleHandleA the result should be its VA, in my case : 77E59F93. It's IF job to extract the API name from its VA. If there was an error during the resolve function the return value must be null (0).

Don't forget :

- To Allow virtual memory at the beginning and read the whole image of the target by using VirtualAlloc and ReadProcessMemory APIs.
- That you can get the size of a region of the target file by using the VirtualQueryEx API.
- That you can read any region of the target file even if it's outside the ImageBase+SizeOfImage range.
- To Free the allocated memory at the end by using the VirtualFree API.

For a concrete example, you can find the source code of teLock98 IF plugin in Samples subdirectory.

P.S : I have used an Assign function because I needed to alloc memory twice, the first one was for the whole memory image and the second was the region where IAT was redirected. Keep in mind that you aren't supposed to use such a function if you have to alloc memory once.

A summary for this part (Delphi example) :

```
library MyPlugin;
```

```
uses Windows;
```

```
function
```

```
Resolve(hProcess:DWORD;ImageBase:DWORD;Size:DWORD;RVA:DWORD):DWORD;StdCall
```

```
;
```

```
var hFile,NOBR:DWORD;
```

```
begin
```

```
hFile:=DWORD(VirtualAlloc(nil,Size,MEM_COMMIT,PAGE_EXECUTE_READWRITE));
```

```
ReadProcessMemory(hProcess,Ptr(ImageBase),Ptr(hFile),Size,NOBR);
```

```
// Your code here

// Code end
VirtualFree(Ptr(hFile),0,MEM_RELEASE);
end;

exports Resolve;

begin
end.
```

6- How can I contact you if I face a problem during my plugin coding ?

I hope that this small help file has clarified some points concerning plugins and, of course, nothing is perfect. Problems, suggestions and criticism are always welcome, you can :

- Visit the official support site of IF at : <http://forum.tuts4you.com/index.php?showforum=119>
- Write an e-mail to the author at : SuperCRacker3@hotmail.com

7- Greetingz

Thanks to all who supported me during IF programming especially Teddy Rogers and Lena from SnD tEaM for their tests and bug reports.

SuperCRacker 2010-2011
SnD tEaM.