Comment Viewer v.0.2

Developed by © bLaCk-eye/RET
October 2006

0. Purpose

Comment Viewer is a plug-in for Interactive Dissasembler (IDA) whose purpose is to provide an easy way for the security researcher to manage the comments in the database. It should prove to be useful on large analysis of binary code projects where keeping a good image of the executable actions is needed.

The plug-in supports a variety of options to be as efficient as possible in a variety of cases, while keeping much of its simplicity. For more information on the various options, what they mean and how to use them please read next chapters.

1. How to install

Comment Viewer plug-in supports IDA v.4.9 and v.5.0 and probably versions that will appear, in the future, as the authors stated that the SDK will be frozen from 4.9 upwards; so if you have another lower version please upgrade to one of the above.

To install the plug-in properly please follow the following steps:

i. Unzip/Unrar "**showcmt.plw**" file intro "**IDA\Plugins\ directory**"

ii. Edit "**plugins.cfg**" file by adding the following two lines:

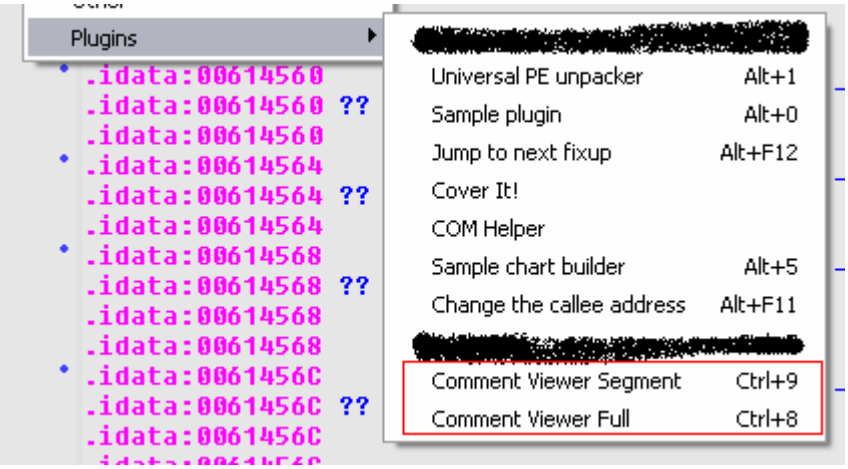
```
Comment_Viewer_Full       showcmt   Ctrl-8   0
Comment_Viewer_Segment    showcmt   Ctrl-9   1
```

Because IDA only scans for plug-ins at startup, after completing the previous steps you need to close all your IDA instances if you want to use the plug-in.

The plug-in can be called using the two defined hotkeys defined in **"plugins.cfg"**

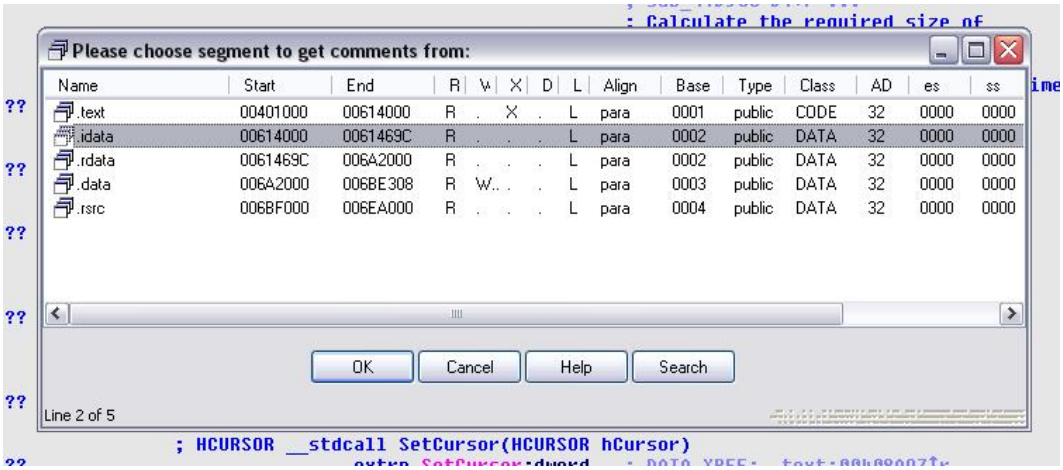
2. How to use

The plug-in can be called from the "**Edit\Plugins**" menu:



Or using the shortcut keys we defined in plugins.cfg (which are also shown in the edit menu shown previously).
What's the difference between the two functions:

- **Full** : this will scan all the database for comments based on the options you choose
- **Segment** (view picture): you can select a certain segment of the binary to scan for comments based on the options you choose. This is useful in a variety of cases when the researcher is looking for the comments attached to a particular section of the binary (e.g. a malware) without having to browse the rest of the comments as well (the default segment is chosen from the address of your cursor in the database);



-

After selecting the area you are interested in seeing comments, you will be presented with a couple of options meant to make the listing of comments as close as possible to want you need to analyze/check.

Let's get down to the details.

In IDA there are 2 types of comments: repeatable and non repeatable. Difference between repeatable and non repeatable is that: **"a repeatable comment will appear attached to the current item and all other items referencing it**" (from IDA help file).

So the first 2 options tell the plug-in what types of comments to search for, as a start. The other options act as filter so that the researcher gets rid of unwanted entries in the comment list.

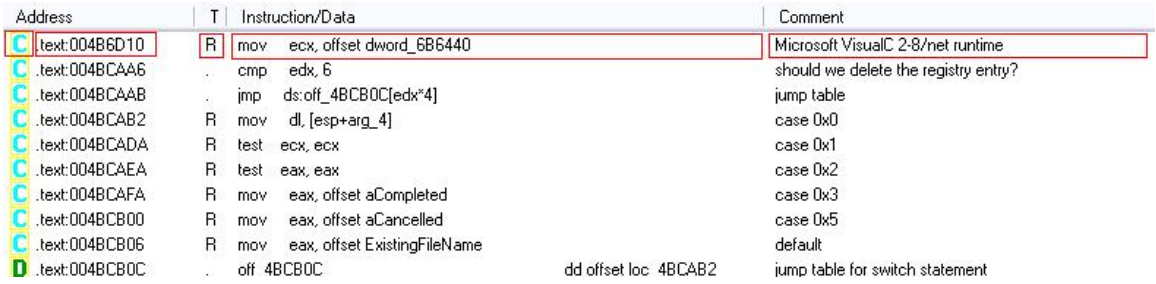
The other options:

- **Show code comments:** the plug-in will add to the list of comments, the ones associated with byte sequences marked as code by IDA or by the user. In the listing they can be easily recognized by a **"C"** letter icon.
- **Show data comments:** the plug-in will add to the list of comments, the ones associated with byte sequences marked as data by IDA or by the user. In the listing they can be easily recognized by a **"D"** letter icon.
- **Show function comments:** the plug-in will add to the list of comments, the ones associated with function comments. Function comments can only be repeatable comments. A repeatable comment is marked as a function comment if it is attached to the first instruction of the function. In the listing they can be easily recognized by an **"F"** letter icon.
- **Show unknown bytes comments:** the plug-in will add to the list of comments, the ones associated with byte sequences marked as unknown by IDA or by the user. In the listing they can be easily recognized by a **"U"** letter icon.

- **Show IDA' s default code comments:** IDA by default tries to make default comments on parts it considers to be informative for the researcher like for example the types of win32 API calls parameters. While they are useful, the disadvantage is that they are numerous and can make the study of the other, more interesting, comments quite hard (e.g. a binary, analyzed by IDA, no user intervention, with all the options on , so including this one had ~19.000 comments, after un-checking it they numbered 1200). If not checked, the plug-in will skip this kind of comments. By default this option is unchecked.
- **Show function declarations:** IDA supports type declarations on functions (default hotkey "Y"), to increase comprehension of the analyzed code and making the code clearer and easier to understand. The plug-in does the same thing: if this option is checked and a function has a type declaration and a comment of course, the listing will show the declaration instead of name of the function. If the function doesn't have a declaration, this option doesn't do anything, the name will appear inside the listing.

The listing is done in a MDI window attached to the IDA interface (via choose2 SDK function), so many of the options for a default chooser apply here also:

- **Delete** (hotkey: Del) : deletes the current comment from the listing and from the database
- **Edit** (hotkey: Ctrl-E): edits the current comment in the listing as well in the database. If no comment is set the comment is automatically deleted from the listing.
- **Refresh** (hotkey Ctrl-U): the plug-in will rescan the database for comments based on your initial options.
- **Copy** (hotkey Ctrl-Ins): the full comment listing is copied to the clipboard.
- **Search** (hotkey Alt-T): search inside the listing.
- **Search Again** (hotkey Ctrl-T): find next match for the search string.



Address	T	Instruction/Data		Comment
.text:004B6D10	R	mov ecx, offset dword_6B6440		Microsoft VisualC 2-8/net runtime
.text:004BCAA6	.	cmp edx, 6		should we delete the registry entry?
.text:004BCAAB	.	jmp ds:off_4BCB0C[edx*4]		jump table
.text:004BCAB2	R	mov dl, [esp+arg_4]		case 0x0
.text:004BCADA	R	test ecx, ecx		case 0x1
.text:004BCAEA	R	test eax, eax		case 0x2
.text:004BCAFA	R	mov eax, offset aCompleted		case 0x3
.text:004BCB00	R	mov eax, offset aCancelled		case 0x5
.text:004BCB06	R	mov eax, offset ExistingFileName		default
.text:004BCB0C	.	off_4BCB0C	dd offset loc_4BCAB2	jump table for switch statement

The listing has 4 columns:

- **Address** at which the item with comment is found.
- **T**ype of comment at that address ("R" = repeatable, "." = non-repeatable)
- **Instruction/Data** : the information found at that address (if it is an instruction it will be shown as one, if a procedure the name of the procedure will be shown or the declaration of the option is checked etc)
- **Comment** : the comment found at that address

The plug-in supports a large number of opened windows with different options, and it's only limited by the number of windows allowed by IDA. This means you can have 2 comment viewers for the same segment or the full database while having different options for each: for example in the first you are interested in the commented functions and in the second one you are interested in code and unknown byte comments.

Because of the way that IDA is built, you can't successfully synchronize (or at least I have not figure it out yet) two windows which share some of their comments. So if you delete a comment from one of the listing and the comment is listed in the other listing as well, then you will see that comment will now have a "X" like icon. This shows that the comment has been deleted from the database and you should not rely on it or you should just update your listing (Ctrl-U). The comments are only gathered for the current options at the start of the plug-in, so you need to update it if you want to be sure you have the latest comments shown.

3. Contact

If you have bugs please mail me with what do you consider the bug to be and I'll to my best to fix it.Thanks

Mailto:

bLaCk@reteam.org

PS: Thanks all of the beta testers and the people who helped me while coding this:
- Rolf Rolles – for putting up with my newbie IDA plug-in coding questions.
- Kao – my loyal beta tester ☺
- LibX – my second loyal tester ☺
- Zairon – for suggesting me what to add to make it more useful