Flier's Sky

Labels

debugging (3)
google chrome (1)
look'n'stop (1)
plugin (1)
security (3)
software (1)
virtualization (3)
wireshark (1)

Links

Flier's Sky - My Chinese Blog
Flier's Sky - My Chinese Tech Blog
PyDbgExt - Python extension for WinDbg
PyV8 - Python wrapper for V8 Javascript Engine
python-google-url/ - A python wrapper for google-url project

Blog Archive

- ► 2008 (3)
- ▼ 2007 (6)
  - ► August (1)
  - ► July (1)
  - ▼ May (4)
    - Write a debugger in 5 minutes with PyDbgEng
    - Access the kernel space with PyDbgEng
    - Dump Windows Service Table in WinDbg
    - An alternative open source virtualization solution...

Flier Lu

Beijing, Beijing, China

View my complete profile

LineBuzz

Saturday, May 26, 2007

Access the kernel space with PyDbgEng

One year ago, I wrote a Chinese article <How to use kd/windbg engine to access the kernel space>, now I port the implementation to the PyDbgExt project, so we can directly access the kernel space in python.

```
>>> from PyDbgEng import *
>>> c = DebugClient()
>>> c.AttachKernel()
>>> c.Control.WaitForEvent()
True
>>> c.Symbols.LoadedModules
{'nt': (Module nt @ ffffffff80800000)}
>>> c.Symbols.GetSymbols("nt!KiServiceTable")
{'KiServiceTable': ((Symbol nt!KiServiceTable), 0)}
>>> offset = c.Symbols.GetSymbols("nt!KiServiceTable").popitem()[1][0].Offset
>>> c.Symbols.GetSymbols(c.DataSpaces.Virtual.ReadPointers(offset)[0])
{'NtAcceptConnectPort': ((Symbol nt!NtAcceptConnectPort), 18446744071571636794L)}
```

To access the kernel mode, we must attach engine to the local kernel with AttachKernel() first, and begin wait a debug event process with WaitForEvent(). For the kernel mode, this function will return immediately. After this, we can use almost all the functions to access the kernel space, such as modules or symbols.

Under the hood, to support this feature in a standalone python module, I use some dirty hack method, because the debug engine and driver disallow it used outside kd.exe or windbg.exe.
So, before call IDebugClient::AttachKernel method to enter the kernel mode, we must first hook four system functions:

```
static DWORD WINAPI HookedGetModuleFileNameW(HMODULE hModule, LPWSTR lpFilename, DWORD nSize)
{
DWORD dwSize = s_fnGetModuleFileNameW(hModule, lpFilename, nSize);

if (!hModule)
{
wchar_t *pch = wcsrchr(lpFilename, L'\\');
wcscpy_s(pch ? pch+1 : lpFilename, pch ? (nSize - (pch - lpFilename)) : nSize, L"kd.exe");
dwSize = wcslen(lpFilename);
}

return dwSize;
}
```

- GetModuleFileNameW, debug engine use it to got the current executable filename, and check whether the filename end with "kd.exe" or "windbg.exe", but our filename maybe "python.exe"
- FindResourceW, SizeofResource, LoadResource: debug engine use those functions to find and export the driver, which implement some internal works. I extract it from windbg.exe and embedded into PyDbgEng.dll.

As the previous description mentioned, we embedded the driver "kldbgdrv.sys" to resource, which as type 0x7777 and id 0x4444, like

```
///////////////////////////////////////////////////
//////////////////////////
//
// RCDATA
//

30583 17476 "kldbgdrv.sys"
```

This tech can be used in any program which wants to access the kernel mode :)

Submit by Flier Lu @ 3:19:00 PM Lables: debugging

7 comments:

Yibam desde MAdrid said...

He, I try instance attachkernel.py with the next code:

```
from PyDbgEng import *
c = DebugClient()
c.AttachKernel()
c.Control.WaitForEvent()

c.Symbols.LoadedModules

c.Symbols.GetSymbols("nt!KiServiceTable")
```

and when trace from wing in 'debug pobre' in line c.Symbols.GetSymbols("nt!KiServiceTable") the python.exe process open the VS2008 debugger ... and finsh the script.

Can i execute scripts.py import your PyDbgEng ???

Could you show a example, pleas ?

Thanks !!!

Im spanish man sorry by my english !!!!
jejje

September 9, 2009 at 6:15 AM

+NCR/CRC! [ReVeRsEr] said...

Hi!,

i would like to know if there is some documentation for this tool. I think it is a good extension but we need more doc or examples.

Thanks in advanced!.

September 12, 2009 at 5:18 AM

Flier Lu said...

Hi Yibam,

What's OS you are using? MS made some limitation for kernel mode debugging in the latest OS or Service Pack.

Hi NCR/CRC!,

Sorry for the poor document, I haven't put too many effort on it, because most of PyDbgEng's basic concepts are just come from WinDbg debugging engine framework. So, I suggest you could read its architect before PyDbgEng. If you need more example, I think you could point out it, and I could provide it later :)

September 15, 2009 at 12:50 AM

+NCR/CRC! [ReVeRsEr] said...

Hey!, thanks for the answer! i'll let you know if i have some problem!.

Thanks in advanced!

September 16, 2009 at 10:41 AM

Yibam desde MAdrid said...

Hi Flier !!!

My OS is Win XP SP3.

I think that the problem is the version of Windbg (6.10).

Could you help us? (NCR and me).

Thanks !!!!

September 16, 2009 at 6:23 PM

Flier Lu said...

Hi Yibam,

Thanks for your comments, I think XP SP3 may introduce some new limitation, and I will try to find root cause later.

For you second question, you definitely could load PyDbgEng from your script, because that is why I design PyDbgEng besides PyDbgExt. You could check the simple example, Write a debugger in 5 minutes with PyDbgEng, http://flierlu.blogspot.com/2007/05/write-debugger-in-5-minutes-with.html

September 18, 2009 at 1:32 AM

Yibam desde MAdrid said...

Hi Flier !!!

First, thanks for your answer.

I hope that PyDbgExt work fine with XP SP3. Could I help you in the study for the Acces Violation?

What do you need ?

What about Vista and W2008 ?

Thanks !!!!!

September 21, 2009 at 7:49 PM

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)