Remote debugging of CrashMe with ntsd -d

Posted by Guillaume - 19 Jul 2011 - 19:01

____

Thanks for your application, I had problems setting up my windbg environment and starting for a clean set of source and PDB was very useful!

I have a DLL that is loaded by Winlogon.exe. To debug it, I established a kernel connexion and used Image File Execution options (IFEO) registry settings so that winlogon is started through ntsd -d.

```
ntsd -d -G -lines -x -y c:symcache;c:windowssystem32 -n -srcpath C:CrashMe
```

I successfull connect to it from windbg with like this :

```
windbg -n -k com:pipe,port=\.pipecom_1,reconnect -srcpath SRV*;C:CrashMe -y
c:windowssystem32;c:windowssymbols;C:CrashMedebug
```

I can see symbols, set breakpoints, but I can't see the source. If I let Winlogon run, use breakin , .reload /f mycode.dll then I can see the source.

I have copied CrashMe in C:CrashMe on both sides, I use the same version of windbg everywhere, etc.

Why is it that piping the ouput through ntsd makes me loose the link between source and symbols ?

ps : I asked the same question on stackoverflow.com.

==

Re: Remote debugging of CrashMe with ntsd -d

Posted by Robert Kuster - 19 Jul 2011 - 22:26

____

Guillaume, welcome.

My experience is that it is often not worth to debug user mode applications from a kernel mode debugger. True, the official docus propose to debug Winlogon just as you did. But hey, Winlogon is almost an ordinary user mode application and with a few simple tricks a user mode debugger will do it just fine.

First note that it is wise to debug Winlogon on a remote machine, because it is considered to be part of the OS. If Winlogon crashes or the debugger screws it up the whole system is taken down. Remote debugging is shortly described in WinDbg. From A to Z! - "Remote Debugging with WinDbg" at slide 87. Basically you have to copy dbgsrv.exe, dbgeng.dll and dbghelp.dll to the remote machine, run dbgsrv.exe on a given port, and connect to that port with WinDbg. The additional trick here is that dbgsrv.exe should run as a service so one can connect to it even before any user logs on. There are two wonderful applications, namely Srvany.exe and Instrsrv.exe, that help you to achieve just that. Just follow the steps described here: How To Create a User-Defined Service. Once you set everything up you should see something like this in the registry of your target machine:

..

```
"Application"="C:\dbgsrv\dbgsrv.exe -t tcp:port=1222"
```

When you restart that machine dbgsrv.exe will be up and running waiting for WinDbg connections. Then attaching to Winlogon will be just one more click away..
Bottom line: I would only use the officially proposed solution with a kernel debugger if debugging a user mode application early in the boot process. In all other scenarios the above solution should yield more satisfactory results.

I hope this helps,
Robert

==