## 2021/01/26\_调试器\_第2课\_调试器的编写

笔记本: 调试器

创建时间: 2021/1/26 星期二 10:38

作者: ileemi

## • 调试器断点原理

获取对方进程加载的dll时,第一次读取对方进程dll的地址,第二次通过这个地址获取加载dll的路径信息。

## 对调试器封装

" FUNCTION " (C标准): 输出函数名

## 调试器断点原理

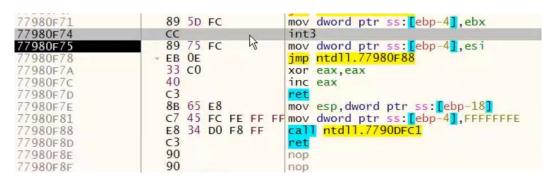
调试器不处理 "系统断点异常",程序不能正常运行(反之可以)。有些异常需要调试器去处理有些异常需要软件自己去处理。当一个程序运行起来的时候,就没有

机会单步执行程序的汇编代码(软件运行过程中没有机会下断点)。

可执行程序加载第一行汇编代码前让其停下来,等待进行调试。也就是说当创建一个调试进程的时候,操作系统会将当前程序的EIP修改为软件的入口地址前的一个地址(让软件第一行代码执行int3,使其产生异常,代码执行后再跳转回去),当不调试进程的时候,软件的EIP就为程序的入口地址。

调试进程时,操作系统会修改EIP的值,指向一块新的内存,进行暂停操作。程序的dll由操作系统去加载。

系统断点 (80000003H): 该异常由操作系统产生,其必须进行处理 (系统断点是操作系统为调试器做的异常),当该断点到来时,操作系统已经将程序需要加载的模块都加载完毕了。系统断点下一行执行的代码为真正的入口代码。这样就可以从入口处单步执行调试程序的汇编代码。



程序中的入口为程序的入口,程序执行入口代码前还需要执行一些初始化的代码(操作系统需要执行一些初始化操作,ntdll.dll模块)。**注意:系统断点,调试器只负责执行一次,下一次程序中再出现系统断点就应该交给程序自己去处理。** 

系统断点来的时候(接下来需要执行入口代码时)输入命令(单步,执行等),反汇编目标程序的代码时(代码在目标进程中),需要从入口代码处开始(EIP所指向的地址)。

获取目标进程的寄存器环境(**GetThreadContext**)并从目标进程中读取代码进行反汇编解析。

单步执行修改目标程序的TF标志,让程序继续执行,会触发单步异常,在处理单步异常里显示一行待执行的汇编代码,继续输入命令产生对应的异常并处理。单步调试利用的就是修改TF标志位并产生一个"80000004H"的异常(调试器执行"T"命令时,会执行一条汇编指令后并产生一个单步异常,并等待处理)。