逆向工程 黑客 (Hacker) 计算机科学

关注者 96 被浏览 10,138

ollydbg中的内存断点和硬件断点有什么区别？

看了一些书和资料，上面说都是当“访问那个内存地址时”中断，请问它们的具体区别在哪里呢？新手求指教

6 个回答

默认排序

旅人
(暴躁老哥) 软件开发|软件安全|x86底层|码农永远在学习中...

81 人赞同了该回答

不请自来，我竟然在知乎上看到人问od（见鬼的表情

首先是关于内存断点的一些前置知识:
1.在windows系统上，内存是一页一页的，也就是说并不是0x00000000~0x7FFFFFFF(0xFFFFFFFF)中任何一个地址都是可以访问的，不然每个程序都有真正的2g内存太可怕了。实际情况是，你程序申请了一块0x2000的内存，这时候windows内核经过一系列处理，告诉你，我在某个地址上(比如说0x00500000)给你设置了0x2000的内存，你可以拿去用了。这时候0x00500000~0x00502000这块内存你才可以访问，你要是尝试访问0x00502001依然会出错，因为这个地址根本不存在。

2.每个内存页都是有属性的，比如你申请时可以要求当前页面里的内容不可作为代码执行，因为你只是拿来存数据的，如果不这样设置的话可能会遭到恶意利用。(比如说缓冲区溢出攻击)

3.调试器可以先于被调试程序拿到一些系统下发的消息，比如说被调试程序中出现了异常。

现在可以说说内存断点都做了些什么了。

内存断点的实现方式是将你欲下断地址**所在的内存页**增加一个名为**PAGE_NOACCESS**的属性，这个属性会把当前内存页设为禁止任何形式的访问，如果进行访问会触发一个内存访问异常。在这同时，od开始捕获目标程序中出现的这个异常，并判断触发这个异常的位置是否跟你下断的地址相同，如果相同则内存断点触发，暂停被调试程序的运行，否则放行。

这就是内存断点的基本原理，补充一些相关的东西：
1.内存断点很消耗资源，因为**PAGE_NOACCESS**属性一设置就是一整个内存页无法访问，那么当程序访问该内存页中非断点地址的内容同样会触发异常，这时od收到异常后需要进行特殊处理，临时放行，非常消耗资源，甚至这使得内存断点在调试很多大型程序时慢到近乎不可用。
2.虽然内存断点的效率经常很不理想，但是因为仅仅是修改了一个内存属性，所以内存断点可以下数量非常多、单断点范围非常大。这是它的优势。
3.只在写入时断下的内存断点通常是将内存属性设为**PAGE_EXECUTE_READ**，也就是不可写来实现的。对这种属性的内存进行写操作将会触发异常。
4.关于内存属性相关的知识:msdn.microsoft.com/en-u...

接下来是硬件断点的前置知识。

1.现代cpu为程序调试提供了6个寄存器，名为DRx，仔细说的话就是DR0/DR1/DR2/DR3/DR6/DR7。没错就是没有4和5，我没写错。cpu内部对这6个寄存器作为硬件断点提供了支持。

相关问题

怎么让调试器当内存出现某些数据时中断？ 11 个回答

shellcode原理是啥？ 11 个回答

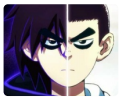
英雄联盟无法定位程序输入点SetupDiGetDevicePropertyW于动态链接库?如何解决？ 8 个回答

为什么程序比较难写、bug 比较难调呢？ 64 个回答

相关推荐


【软件安全】缓冲区溢出攻防 ★★★★★


【软件安全】环境变量与Set-UID攻防 ★★★★★


【软件安全】内核写时复制漏洞 ★★★★★

2.调试器可以轻易读写被调试程序的这6个寄存器，而被调试程序不容易读写也通常不需要读写。

3.DR0~DR3四个寄存器用来存放欲下断的地址，DR6和DR7用来控制断点的大小和触发断点的时机。(比如说大小一个byte，触发时机为写入时)

硬件断点就不需要od做太多事情了，它只需要把用户的需求转换一下格式，写入被调试程序的DRx系列的寄存器中，并等待系统发来的消息就行了。(我记不得这个消息是不是也是一个异常消息了,年代久远太久不碰)
当od收到了消息就暂停目标程序，你就知道程序断下了。

关于硬件断点：
1.寄存器数量的限制导致硬件断点最多只能同时存在4个，并且od在特定设置或者插件的影响下可能内部还会占用一两个用来辅助程序调试，导致可用数量十分有限。
2.不仅硬件断点数量不多，在32位程序中，每个硬件断点最大范围是4个字节，这也经常不太够用。
3.由于cpu的直接支持，硬件断点的效率是非常高的，给一个程序设置了硬件断点，在不触发的情况下，不会有肉眼可见的效率影响，毕竟只是写了个寄存器而已。

更新 16.11.24
根据评论提醒，我去测试了一下，确实原来对内存断点所设置的属性弄错了，文章已更新。抱歉各位，科普回答中出现这种错误是件挺糟的事情。
更新简述:

> 内存访问断点所设置的属性为PAGE_NOACCESS，而不是PAGE_GUARD，前者是禁止该端内存中的任何访问，后者是只在第一次访问时触发异常。
> 内存写入断点应该是PAGE_EXECUTE_READ，因为PAGE_READONLY会影响到当前地址作为代码执行。

编辑于 2016-11-24

VeroFess
360 安全工程师

4 人赞同了该回答

楼上几位说的很明白了，这里补一下具体内容...

断点都是通过ExceptionRecord->ExceptionCode判断

在进行内存断点时，要是使用添加PAGE_GUARD的方式进行保护，会触发
STATUS_GUARD_PAGE_VIOLATION(0x80000001)的异常
而通过取消相应权限的，则会触发EXCEPTION_ACCESS_VIOLATION(0xC0000005)异常
但是绝对不会触发
EXCEPTION_IN_PAGE_ERROR(0xC0000006)

而硬件断点(包括单步和DrX)都会触发STATUS_SINGLE_STEP(0x80000004)异常

而使用int 3进行中断则会触发EXCEPTION_BREAKPOINT(0x80000003)异常

这里就是不同之处了
搞定，吃饭去

编辑于 2016-11-17

许羡
神经质码农一枚。

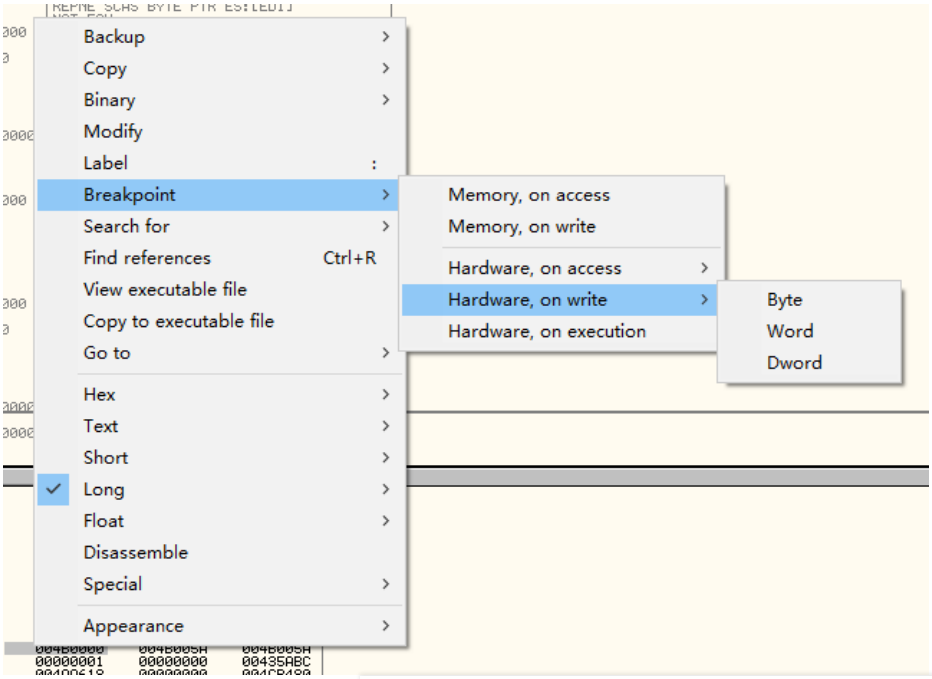
9 人赞同了该回答

内存断点设置内存属性而触发的。例如内存属性，有可读可写可执行。你修改了不可读，当程序读那内存的时候，就会触发内存错误异常，调试器会捕获此异常。
硬件断点是设置dr寄存器而触发的。这个是CPU层面的支持因而只能设置四个，因为只有dr0--dr3。这四个。CPU执行命令的时候，发现与dr寄存器符合的时候会触发异常。

发布于 2016-11-15

2 人赞同了该回答

内存断点和硬件断点是两个不同的概念,我感觉题主是搞混了.
内存断点的实现方式有两种,一种是通过设置该内存位置所在的页不可读/写,然后当程序中对该内存进行读/写的时候就会触发异常,调试器捕获到这个异常后范县这个不可读/写的异常是因为设置了内存断点,于是就断下来了;还有一种方式是使用硬件断点,是的,硬件断点可以用来实现内存断点.当你把某个地址上设置了硬件读/写断点后,程序对该内存进行读写时CPU会触发一个异常,调试器捕获该异常后发现是设置了硬件的读写断点后就断下来了.
题主如果仔细观察OD设置内存断点的菜单就会发现设置内存断点的菜单有内存读/写断点,还有硬件的读/写/执行断点.



想要详细了解这个可以看一下张银奎的<软件调试>一书.

编辑于 2016-11-16

▲ 赞同 2 ▼ 1 条评论 分享 收藏 喜欢

老董
读的英语，做着码农，在学算法...

因为我是玩过软件加解密的。我就讲实际的。

内存断点是通过把相应位置指令替换成int3来实现的。
硬件断点是通过设置CPU相应硬件寄存器来阻止程序继续运行的。

因为修改程序代码你想怎么改就怎么改。所以内存断点你可以设置很多个。而硬件寄存器数量有限，所以只能设置几个（目前大多数是4个）
因为修改的是程序的代码，所以内存断点很容易被程序自身检测到。而硬件断点则很难被发现。

实际使用过程中，内存的效率比较低，但可以下许多个。而硬件断点效率高，但是只能设置有限个数（大概三个，具体查intel手册）。
另外建议楼主去看雪查查就知道了，到知乎学不了这个。

发布于 2016-11-15

▲ 赞同 ▼ 1 条评论 分享 收藏 喜欢

写回答