**堆破坏**

所谓的堆破坏，是说没控制好自己的指针，把不属于你分配的那块内存给写覆盖了。这块内存可能是你程序的数据，也可能是堆的管理结构。

堆破坏较为理想的情况是被修改的数据会马上导致程序crash，最差的情况是你的堆数据莫名其妙在今天被改了，但明天才crash。这个时候在去分析crash，就如我们的警察叔叔现在接手一桩10年前的案子一般----无从下手。老外称之为heap corruption是很贴切的，有时候堆数据被意外篡改是无声无息的，你也许没法从界面甚至日志文件中看到它被篡改的一点迹象，当到某一个时刻，这种错误会暴露出来，然而这个时候查看堆信息也许会是毫无头绪。所以对于堆破坏，咱的策略是尽早发现我们的堆被篡改了，最好能够在堆数据被意外篡改的那一时刻诱发一个异常来提醒我们----堆被腐蚀了。

微软提供了一些方案，来帮助我们诊断堆破坏。一般来说，堆破坏往往都是写数据越界造成的，所以微软在堆分配上，给程序员门额外提供了2种堆分配模式--完全页堆（full page heap），准页堆(normal page heap)，用来检测堆被写越界的情况。

完全页堆：

当分配一块内存时，通过调整内存块的分配位置，使其结尾恰好与系统分页边界对齐，然后在边界处再多分配一个不可访问的页作为保护区域。这样，一旦出现内存读/写越界时，进程就会Crash，从而帮助及时检查内存越界。

因为每次分配的内存都要以这种形式布局，尤其对于小片的内存分配，即使分配一个字节，也要分配一个内存页，和一个保留的虚拟内存页（注意在目前的实现中，这个用作边界保护区域的页从来不会被提交）。这就需要大量的内存，到底一个进程需要多少内存，很难估算，因此在使用Page Heap前，至少保证你的机器至少设置了1G虚拟内存以上。

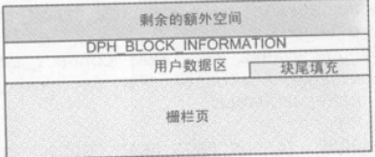
准页堆：

准页堆原理与CRT调试内存分配函数类似，通过分配少量的填充信息，在释放内存块时检查填充区域。来检测内存是否被损坏，此方法的优点是极大的减少了内存耗用量。缺点是只能在释放块时检测，不太好跟踪出错的代码位置。

**完全页堆（FULL PAGE HEAP）**

**检测原理**

完全页堆的检测基本思路是通过分配相邻的一个页，并将其设为不可访问属性，然后用户数据块会被分配到内存页的最末端，从而实现越界访问的检测。当我们对堆中分配的内存读写越界后便会访问到那个不可读的页，系统捕获到该次异常后会试图中断执行并将该异常上报给debugger，或者崩溃。具体的内存组织结构如下图



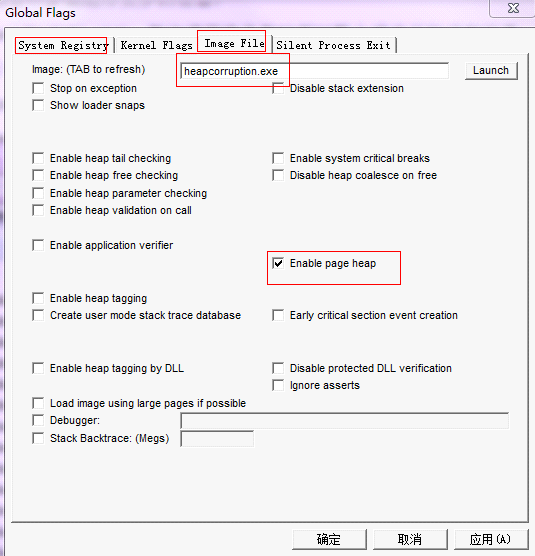
与普通堆不同的是，内存块前面的HEAP\_ENTRY结构被DPH\_BLOCK\_INFORMATION结构取代，这个结构内部记录了页堆模式下这个内存块的一些基本信息。如果用户数据区前面的数据，也就是DPH\_BLOCK\_INFORMATION结构被破坏了，那么在释放内存块的时候系统会报错，如果编程者对这块内存块读写越界了，当然，这里越界有几种情况：

* 读越界，但只是访问了块尾填充部分数据，那么系统不会报错
* 写越界，但只篡改了图中块尾填充的部分，那么在堆块释放的时候会报错
* 读越界，且超过了块尾填充的部分，访问到了栅栏页，那么系统会立即抛出一个异常并中断执行
* 写越界，且超过了块尾填充部分，写到了栅栏页，那么系统会立即抛出一个异常并中断执行

这里需要注意的还是块尾填充不一定存在，块尾填充是因为要满足堆内存的最小分配粒度，如果本身内存块的分配粒度就已经是最小分配粒度的倍数了，那么块尾填充就不存在了，比如堆内存分配粒度是是8 bytes，那么如果申请了14 bytes的话会有2 bytes的大徐小的块尾填充块，如果申请了24bytes，那么就没有块尾填充了，因为24正好是8的倍数。

示例

开启全页堆（用windbg目录下的gflags）



可以看到程序分为三个标签页：system Registry、Kernel flags 、Image File。

标签页System Registry用于设置针对整个系统的选项。设置之后需要重启系统才能生效。

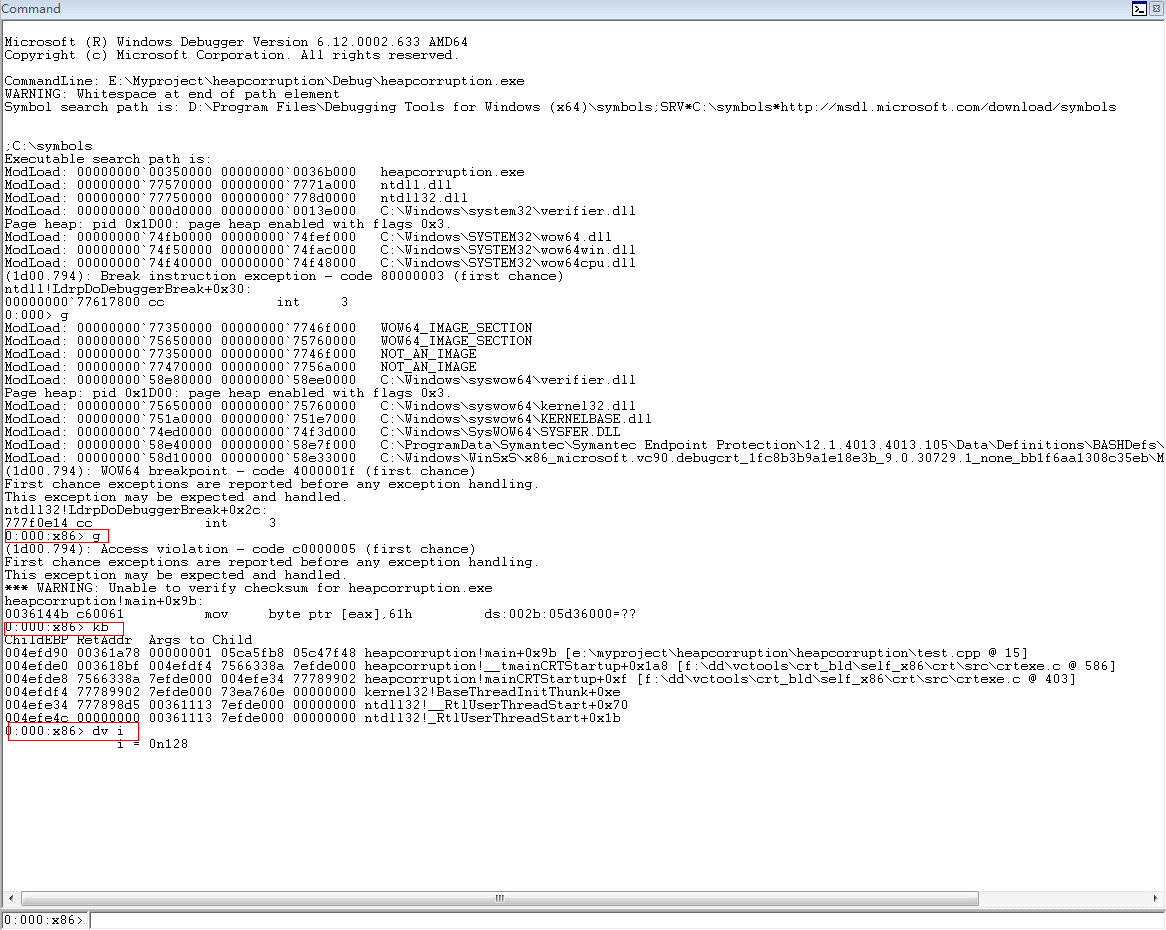
Image File用于设置针对单个进程的配置。设置之后重启进程后才会生效。

Kernel Flags用于设置只对内核产生影响的选项。

在gflags中包含了针对操作系统的各个方面的配置信息，这些信息是被保存在注册表相应的位置上。

单击Image File标签页，在Image：（Tab to Refresh）编辑框内输入要设置的进程名称。如calc.exe。设置完成后点击Tab进行刷新，刷新后下面的各个控件变为可用状态。在需要设置的调试选项后点勾，确认后即可。

Heapcorruption.exe的调试步骤：



在填充第128字节的时候，temp指针访问到了栅栏页，从而报出了一个内存违规的异常。

**结论：只要写越界，系统都能够检测出来，只不过如果写越界写到了栅栏页会理解触发异常中断，而写越界只写了块尾填充部分，那么系统在适当时机（比如堆被销毁，或者这块内存被重新分配等时机）会对块尾填充部分做完整性检测，如果发现被破坏了，就会报错。**