c语言vt指令,VT系列一：VT简述

转载

本文只是学习此视频后的一些总结 不当之处还请指出

视频作者：小宝来了

视频连接：http://bbs.pediy.com/showthread.php?t=211973

约定：

本文中出现的名词

虚拟机 客户机 GUEST 都是被监控的操作系统或应用程序

宿主机 HOST Hypervisor都是指监控虚拟机的“原”操作系统

VMM:当客户机发生退出事件时,进入的就是VMM

VM：当客户机正常运行时就是VM

VMM监控VM

1.什么是虚拟机？

自己的理解是：运行在Hypervisor监控下的系统为虚拟机

以下内容摘自看雪论坛两个大牛

首先要说一下VMM(VirtualMachineMonitor)，这个是虚拟机的监控器，监控着虚拟机的运行。比如虚拟机想执行一条指令：cpuid，这个时候被VMM捕捉到，然后VMM去模拟执行这条指令(模拟执行的意思是按我们自己的意愿去执行)，然后返回给虚拟机，完成了一次vmexit。因为VMM需要执行ring0的指令，所以VMM需要运行在ring0下。

而VT使得CPU进入了一个全新的特殊模式(VMX模式)，在这个模式下，CPU可以处于VMXroot状态或者VMXnon-root状态。处于VMXnon-root操作状态下的CPU行为受到了某些方面的限制，关键的共享资源必须运行于VMXroot操作状态的监控器的控制之下，并且，对于VMXnon-root状态中的任何CPU特权都有效(只要处于VMXnon-root状态，ring0-ring3都被监控)。因此，将VMM运行于VMXroot操作状态，可以轻松监控管理客户操作系统(就是安装在虚拟机里面的操作系统)和客户应用程序(虚拟机里面操作系统里面安装的软件)。

----以上资料摘自看雪论坛海风月影

VT技术方面,因为VT技术比较新,如果你对VT技术还不甚明白,请先查看intel文档.中文不大好可以配合newbluepill那本书看,VT技术方面,我使用它的下面几个特性.

重定向中断.对于1号中断,属于硬件中断,我会在windows中搜索0x20一下的中断号给1号中断使用.对于3号中断属于软中断,随便在IDT中搜索一个空的就OK了.

MSR寄存器保护,我在插件开启的时候会将MSR_IA32_SYSENTER_EIP0x176换掉,换成我们实现的KiFastCallEntry.在这里判断是否是我们需要出来的SSDT调用,根据不同的SSDT调用我们转到不同的内核模块中.(我们重定位的模块,或者是系统本身的模块)

VT技术还有很多特性,比如CR寄存器访问,DR寄存器访问.本来都想用上,但是发现,不行.因为在CR寄存器访问,和DR寄存器访问的时候,我没有办法确定线程的运行上下文在那个进程,线程环境中..所以针对Cr3的保护,和DRx寄存器保护没有做.

另外还有一个比较遗憾的地方是,本来我决定要给整一个无限断点的.后面因为项目搁置没有弄.另外一个是想写一个类似NewBluePill的内存隐藏.如果给台硬件调试器,可以试试.

----以上资料摘自看雪论坛Ddvp插件作者JoenChen

2.VT启动流程：

详细见：intel手册 31.5章

VMX退出原因指示器：记录了什么原因导致产生退出事件的(比如触发了CPUID)

VMCS:记录了导致退出事件时的一些信息(寄存器同上下文)

虚拟机状态保存区：记录了当发生退出事件时的信息 用于恢复虚拟机的状态或我们修改

宿主机状态保存区：略

虚拟机运行控制域：定义虚拟机什么情况下发生退出事件

其它省略

3.VMX相关汇编指令

VMREAD当虚拟机发生退出事件时,使用VMREAD读取发生退出事件时的上下文

VMWRITE当虚拟机发生退出事件时，如果需要按我们的意愿执行 则使用次指令修改相关参数

VMCALL一个让虚拟机主动发生退出事件的指令用于关闭VT时

VMLAUNCH：启动虚拟机 并将控制权交于虚拟机

VMRESUME：用于在宿主机中恢复虚拟机运行并将控制权交给虚拟机

VMXOFF：关闭虚拟机

VMXON：启用一些VMX 指令

下一章将将检测CPU是否支持虚拟化