剖析Windows系统服务调用机制(中) PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/140/2021_2022__E5_89_96_E6_9E_90Wind_c100_140142.htm 四> Windows 2000系统服务调用机制 Windows 2000的陷阱调度(Trap Dispatching)机制包括了：中断(Interrupt)，延迟过程调用(Deferred Procedure Call)，异步过程调用(Asynchronous Procedure Call)，异常调度(Exception Dispatching)和系统服务调用。在Intel x86的Windows 2000系统中，处理器执行int 0x2e指令来激活Windows系统服务调用；在Intel x86的Windows XP系统中处理器却是通过执行sysenter指令使系统陷入系统服务调用程序中；而在AMD的Windows XP中使用了指令syscall来实现同样的功能。我们暂时使用x86的Windows 2000为例来演示。我们先给出一个系统服务调用的模型： mov eax, ServiceId lea edx, ParameterTable int 2eh ret ParamTableBytes 其中，ServiceId清楚的说明了传递给系统服务调用的系统服务号，内核使用这个标识符来查找系统服务调度表(System Service Dispath Table)中的对应系统服务信息。在系统服务调度表中的每一项包含了一个指向系统服务程序的指针，我们Hook时就是修改这个指针使其指向我们自定义的系统服务的地址。ParameterTable是传递的参数，系统服务调用程序KiSystemService必须严格校验传递的每一个参数，并将其参数从线程的用户堆栈中复制到系统的核心堆栈以备使用。由于执行int指令会导致陷阱发生，所以在Windows 2000内的中断描述表(IDT = Interrupt Descriptor Table)中的0x2e项指向了系统服务调用程序。最后返回的ParamTableBytes是关于参数个数的信息。 现在我们已

经看得出来了，系统服务调用只是一个接口，它提供了将用户模式下的请求转发到Windows 2000内核的功能，并引发处理器模式的切换。在用户看来，系统服务调用接口就是Windows内核组件功能实现对外的一个界面。系统服务调用接口定义了Windows内核提供的大量服务。 五> Windows 2000系统服务调用类型 在Windows 2000中默认存在两个系统服务调度表，它们对应了两类不同的系统服务。这两个系统服务调度表分别是：KeServiceDescriptorTable和KeServiceDescriptorTableShadow。 Windows 2000执行程序服务对应于NTDLL.dll为我们提供的系统服务调用。子系统通过调用NTDLL.dll中的函数接口来实现它们需要的功能。系统服务调度表KeServiceDescriptorTable定义了在ntoskrln.exe中实现的系统服务，通常在kernel32.dll/advapi32.dll中提供的函数接口均是调用的这个系统服务调度表中。 同时存在于Windows 2000操作系统中还有在Win32k.sys中实现的相关Win32USER和GDI函数，它们是属于另一类系统服务调用。与之对应的系统服务调度表为KeServiceDescriptorTableShadow，它提供了内核模式实现的USER和GDI服务。函数KeAddSystemServiceTable允许Win32.sys和其他设备驱动程序添加系统服务表。除了Win32k.sys服务表外，使用KeAddSystemServiceTable添加的服务表会被同时复制到KeServiceDescriptorTable和KeServiceDescriptorTableShadow中去。 我们可以看出这两类函数实现在服务调度上的区别：Win32内核API经过Kernel32.dll/advapi32.dll进入NTDLL.dll后使用int 0x2e中断进入内核，最后在Ntoskrnl.exe中实现了真正的函数调用；Win32 USER/GDI API直接通

过User32.dll/Gdi32.dll进入了内核，最后却是在Win32k.sys中实现了真正的函数调用。在此我们只讨论与NTDLL.dll相关的函数，也就是我们例子中处理的函数。 六> Hook系统服务调用的作用 钩子(Hooking)是一种拦截/监听可执行代码在执行过程中相关信息的一种通用机制。它使我们了解系统内部结构，运作机制甚至修改系统行为的想法成为可能。在一个像M$存在的世界里，Windows的很多内部信息我们都是无法得知的，因为Windows不是Linux，但这并不意味着我们就此放弃！只要开动你的大脑，很多事情都会变成可能。 1. 事件追踪 你想知道Windows在什么时候会打开一个进程吗？你想知道Windows任务管理器中进程相关信息的获取调用了哪些函数吗？我们都可以使用Hook技术来实现这些你想要的信息。我们可以追踪ZwOpenProcess的执行情况，我们同样也可以追踪ZwQueryInformationProcess的执行情况，包括传递的参数和返回的结果。大家可以看到本文相关的程序T-ProcMon就是一个进程监视工具，它会追踪系统中与进程相关的各种信息。在某些我们期望的事件发生时，程序会通知用户发生了什么，这也是我们期望看到的结果。 2. 修改系统行为 操作系统为我们提供了一些通用的功能，如查询系统进程信息ZwQuerySystemInformation(SystemInformationClass == 5)，它会返回系统中当前所有进程/线程的相关信息。如果我们希望隐藏一些特殊的进程那该怎么办呢？那就是修改系统服务调用，也就是修改ZwQuerySystemInformation的行为。在查询系统进程时，系统会返回一个进程信息队列，每个单元对应一个进程，如果我们想隐藏其中的某个进程，只须修改队列中的某些数据，然后返回给上层函数，它们就不会发现Xxx.exe

进程存在于系统之中了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com