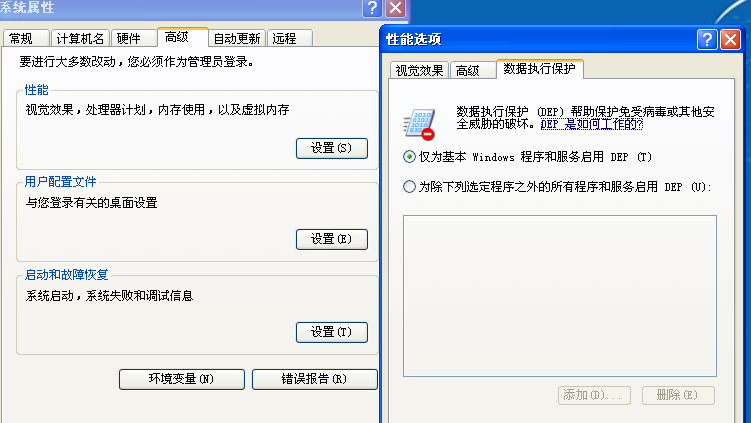
DEP

1. DEP

DEP(数据执行保护Data Exevution Prevention)：将数据所在内存页标识为不可执行，当程序溢出成功转入shellcode时，程序会尝试在数据页上执行命令，此时CPU就会抛出异常，而不会执行恶意代码。

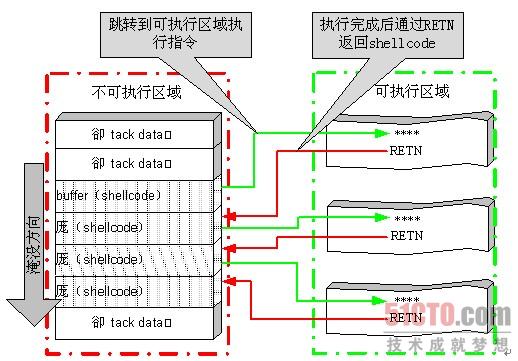
软件DEP：就是safeSEH，它的目的是阻止利用SEH攻击

硬件DEP：通过设置内存页的NX/XD属性标记，来指明不能从该内存执行代码



1. Ret2Libc

Ret2Libc是Return-to-libc的简写，由于DEP不允许我们直接到非可执行页执行指令，我们就需要在其他可执行的位置找到符合我们要求的指令，让这条指令来替我们工作，为了能够控制程序流程，在这条指令执行后，我们还需要一个返回指令，以便收回程序的控制权，然后继续下一步操作



为此，我们在继承这种思想的大前提下，介绍三种经过改进的、相对比较有效的绕过DEP的exploit方法。

（1）通过跳转到ZwSetInformationProcess函数将DEP关闭后再转入shellcode执行。

（2）通过跳转到VirtualProtect函数来将shellcode所在内存页设置为可执行状态，然后再转入shellcode执行。

（3）通过跳转到VIrtualAlloc函数开辟一段具有执行权限的内存空间，然后将shellcode复制到这段内存中执行。