

第 5 章 MS08-067 漏洞攻击实验

5.1 预备知识

内存攻击指的是黑客利用操作系统等软件的漏洞,构造恶意输入,导致软件在处理输入数据时出现非预期的错误,将输入数据写入内存中的某些特定敏感位置,从而劫持软件控制流,转而执行外部输入的非安全的指令代码,造成所在系统被获取远程控制或者被拒绝服务。内存攻击的表面原因是软件编写错误,例如过滤输入的条件设置缺陷、变量类型转换错误、逻辑判断错误、指针引用错误等,但究其原因是现代电子计算机在实现图灵机模型时没有在内存中严格地区分数据和指令,这就存在程序将外部输入数据作为指令代码而被执行的可能。任何操作系统级别的防护措施都不可能完全根除现代计算机体系结构上的这个弊端,而只是试图去阻止攻击者利用此弊端攻击计算机。

5.1.1 缓冲区溢出

通常情况下,缓冲区溢出的数据只会破坏程序数据,造成意外终止。但是如果有人精心构造溢出数据的内容,那么就有可能获得系统的控制权。

缓冲区在系统中的表现形式是多样的,高级语言定义的变量、数组、结构体等在运行时都是保存在缓冲区内的,因此,缓冲区可以更抽象地理解为一段可读写的内存区域,缓冲区攻击的最终目的就是希望系统能执行这块可读写内存中已经被蓄意设定好的恶意代码。按照冯·诺依曼存储程序原理,程序代码是作为二进制数据存储在内存的,同样程序的数据也在内存中,因此,直接从内存的二进制形式上是无法区分哪些是数据哪些是代码的,这也为缓冲区溢出攻击提供了可能。

一般根据缓冲区溢出的内存位置的不同,将缓冲区溢出又分为栈溢出和堆溢出。

5.1.2 栈溢出

栈作为一种数据结构,是一种只能在一端进行插入和删除操作的特殊线性表。它按照先进后出的原则存储数据,先进入的数据被压入栈底,最后的数据在栈顶,需要读数据的时候从栈顶开始弹出数据(最后一个数据被第一个读出来,即“后进先出”)。栈具有记忆作用,对栈的插入与删除操作中,不需要改变栈底指针。

在计算机系统中,栈是一个具有以上属性的动态内存区域。程序可以将数据压入栈中,也可以将数据从栈顶弹出。压栈的操作使得栈顶的地址减小,从栈顶弹出操作使得栈顶的地址增大。

栈在程序的运行中有着举足轻重的作用。最重要的是,栈保存了一个函数调用时所需要的维护信息,常称之为堆栈帧或者活动记录。

堆栈帧一般包含如下几方面的信息:

- 函数的返回地址和参数。

- 临时变量,包括函数的非静态局部变量以及编译器自动生成的其他临时变量。

栈溢出发生在程序向位于栈中的内存地址写数据,当写入的数据长度超过栈分配给缓冲区的空间时就会造成栈溢出。从栈溢出的原理出发,攻击者可以找到如下集中方式来利用这种类型的漏洞:

- 覆盖缓冲区附近的程序变量,改变程序的执行流程和结果,从而达到攻击者的目的。
- 覆盖栈中保存的函数返回地址,修改为攻击者指定的地址,当程序返回时,程序流程将跳转到攻击者指定的地址,理想情况下可以执行任何代码。
- 覆盖某个函数指针或者程序异常处理结构,只要溢出之后目标函数或者异常处理例程将被执行,同样可以让程序流程跳转到任意地址。

5.1.3 堆溢出

不同于栈,堆是程序运行时动态分配的内存,用户通过 malloc(C 语言)、new(Java 语言等)等函数申请内存,通过返回的起始地址指针对分配的内存进行操作,使用完成后要通过 free 等函数释放这部分内存,否则会造成内存泄漏。

堆的操作分为分配、释放和合并 3 种。因为堆在内存中位置不固定,大小比较自由,多次申请和释放后可能会更加混乱,系统从性能、空间利用率以及安全的角度考虑来管理堆。下面通过其中的空闲堆块操作进行简要介绍。

系统按照堆块大小不同维护一系列的堆块。而堆块又分为块首和数据区,其中空闲堆块数据区的前两个双字分别是双向链表的两个指针。通常同样大小的空闲堆块通过双向链表连接在一起,分配与释放堆,分别对应插入与删除双向链表节点的操作,而合并则会同时进行这两种操作。

空闲堆块中,两个指针 Previous block 和 Next block 分别指向双向链表中此堆块的前后两个空闲堆块的数据部分。分配一个堆块时,将分配堆块从空闲堆块双链表中删除。同一个堆中的堆块在内存中通常是连续的,因此很可能发生以下情况:在向一个已分配堆块中写入数据时,由于数据长度超过了所在堆块的大小,导致数据溢出覆盖了堆块后方相邻的空闲堆块,而包含的堆块的两个前后指针就会被覆盖或者部分覆盖。

假设有这样一个空闲堆块,它的前后堆块指针被覆盖。也就是说,本来应该指向该堆块的前一个堆块和后一个堆块的数据被改写,替换成了其他的数据。而如果紧接着这个空闲堆块被分配出去,需要将这个空闲堆块从空闲堆块的链表中删除。那么在分配的过程中 DeleteBlock 函数就会将该节点的下一个节点的前向指针指向该节点之前的空闲块的前向指针,从而知道,每个堆块指针指向的就是堆块的前向块。因此,这个动作相当于就是对该节点的解引用。也就是说,该节点的后向指针所对应的地址的内容,其实就是前向指针所在位置的数据。

5.2 MS08-067 漏洞攻击实验

实验器材

Back Track5 的镜像文件,1 套。

VMware 虚拟机软件,1 套。

Windows Server 2003 SP0 镜像文件,1套。

实验任务

通过本实验,掌握针对内存泄漏攻击的相关知识。

实验环境

一台安装了 VMware 虚拟机软件的 Windows 7 操作系统的计算机,BT5(Back Track five)系统,以及 Windows Server 2003 SP0 靶机系统。

预备知识

2008年10月24日凌晨,联想网御安全服务部攻防研究团队在监测系统安全状态过程中,发现 Windows Server 服务远程 RPC 栈溢出漏洞(MS08-067)。这是在 Windows 操作系统下的 Server 服务在处理 RPC 请求过程中存在的一个严重漏洞,远程攻击者可以通过发送恶意 RPC 请求触发这个溢出,导致完全入侵用户系统,并以 SYSTEM 权限执行任意指令并获取数据,造成系统失窃及系统崩溃等严重问题。

实验步骤

使用 Metasploit 框架中的 MS08-067 渗透攻击模块,对一台自己架设的还没有使用 DEP 与 ASLR 安全防护机制的 Windows Server 2003 SP0 靶机进行渗透实验。

1. Windows Server 2003 SP0 靶机架设

Windows Server 2003 是 Microsoft 公司基于 Windows XP/NT5.1 开发的服务器操作系统,于 2003 年 3 月 28 日发布,并在同年 4 月底上市。相对于 Windows 2000 做了很多改进。

Windows Server 2003 的官方支持已在 2015 年 7 月 14 日结束,Windows Server 2003 的安全性不再获得保障,此处是作为实验用软件进行安装。

(1) 从网上下载无任何 SP 补丁的 Windows Server 2003 镜像文件,保存到本地待接下来安装到 VMware 虚拟机中。

(2) 打开 VMware 虚拟机软件,出现安装向导窗口,单击“创建新的虚拟机”选项,出现图 5.2.1 所示的“新建虚拟机向导”界面,通过本向导来创建一个新的虚拟机。

(3) 在配置类型中,选择“自定义(高级)(C)”,并单击“下一步”按钮,出现如图 5.2.2 所示的“选择虚拟机硬件兼容性”界面。

(4) 在“选择虚拟机硬件兼容性”界面中,选择默认的硬件兼容性,即 Workstation 12.0 即可,单击“下一步”按钮。

(5) 在出现的如图 5.2.3 所示的“安装客户机操作系统”界面中,选择“安装程序光盘影像文件(iso)(M)”选项,通过“浏览”找到刚才下载好的系统镜像文件并添加,然后单击“下一步”按钮。

(6) 如图 5.2.4 所示,此时进入的是简易安装信息界面,需要输入一个系统的产品密钥,可以选择此时输入;也可以直接单击“下一步”按钮,即在虚拟机中安装系统的时候再输入。两种方法都可以,没有影响。

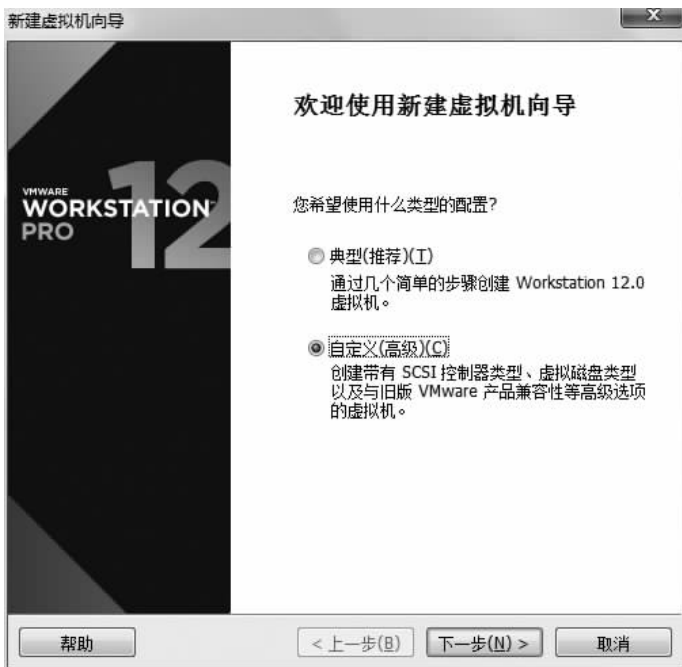


图 5.2.1 “新建虚拟机向导”窗口



图 5.2.2 “选择虚拟机硬件兼容性”界面



图 5.2.3 “安装客户机操作系统”界面



图 5.2.4 简易安装信息界面

(7) 在图 5.2.5“命名虚拟机”界面的“虚拟机名称(V)”选项中全部选择系统默认的设置,并单击“下一步”按钮。

(8) 在出现的如图 5.2.6 所示的处理器配置界面中,可以根据自己实验平台的硬件条件,自行决定“处理器数量”以及“每个处理器的核心数量(C)”的具体值。本次实验使用的是默认值,单击“下一步”按钮。



图 5.2.5 “命名虚拟机”界面



图 5.2.6 “处理器配置”界面

(9) 在如图 5.2.7 所示的“此虚拟机的内存(M)”界面中,同样可以根据自己实验平台的硬件条件,为虚拟机设置内存大小。本次实验选用的是 1024MB,单击“下一步”按钮。

(10) 在如图 5.2.8 所示的“网络类型”界面的“网络连接”选项中,为虚拟机选择“使用网络地址转换 NAT(E)”模式,单击“下一步”按钮。

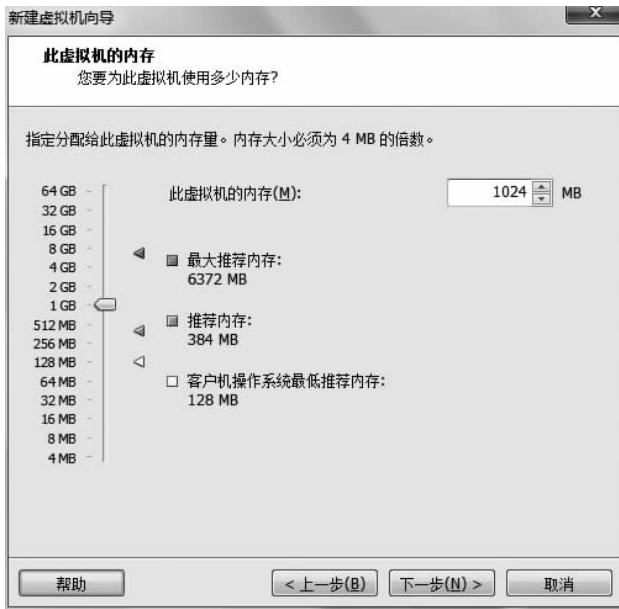


图 5.2.7 “此虚拟机的内存”界面

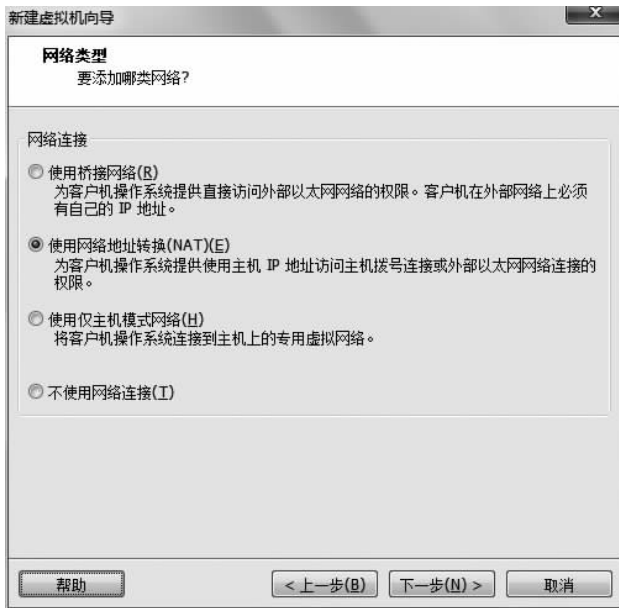


图 5.2.8 “网络配置”界面

(11) 在如图 5.2.9 所示的“选择 I/O 控制器类型”界面的“SCSI 控制器”选项中,选择软件推荐的 LST Logic(L)单选项,然后单击“下一步”按钮。

(12) 在如图 5.2.10 所示的“选择磁盘类型”界面的“虚拟磁盘类型”选项中,同样选择软件推荐的 SCSI(S)选项,然后单击“下一步”按钮。



图 5.2.9 “选择 I/O 控制器类型”界面



图 5.2.10 “选择磁盘类型”界面

(13) 在如图 5.2.11 所示的“选择磁盘”界面的“磁盘”选项中,选择“创建新虚拟磁盘(V)”模式,单击“下一步”按钮。

(14) 在如图 5.2.12 所示的“指定磁盘容量”界面的“最大磁盘大小(GB)(S)”选项中,同样使用软件建议的 40.0,当然,磁盘大小可以根据自己硬件条件进行调整。不建议勾选“立即分配所有磁盘空间”,因为根据使用大小再分配磁盘空间大小完全够用,并不会影响使用效果。接下来,勾选“将虚拟磁盘拆分成多个文件(M)”选项,单击“下一步”按钮。

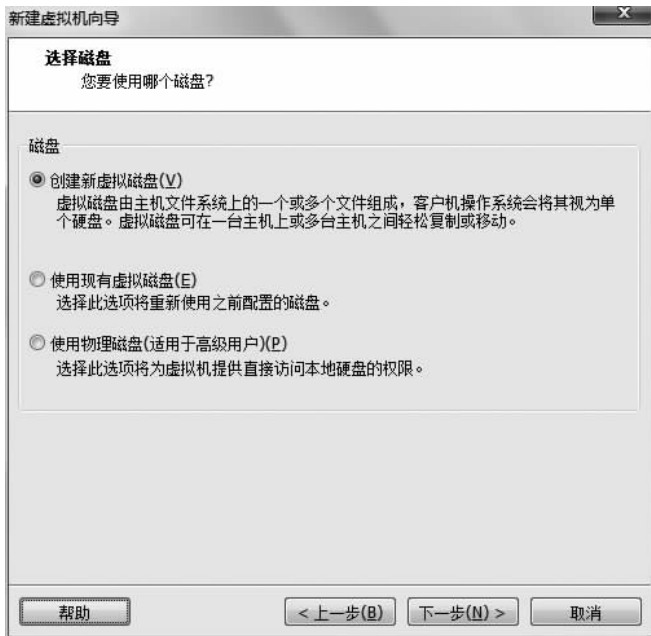


图 5.2.11 创建虚拟磁盘

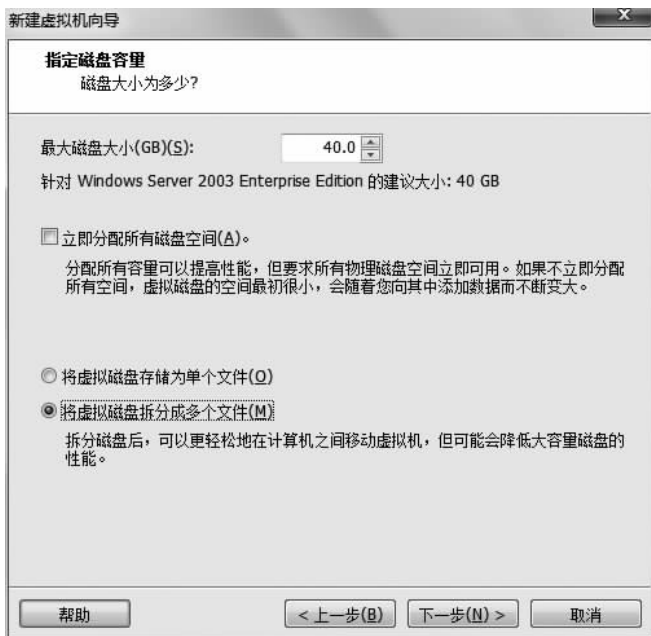


图 5.2.12 设置磁盘大小

(15) 在如图 5.2.13 所示的“指定磁盘文件”界面中同样选择软件默认的文件名称和磁盘文件存储地址,单击“下一步”按钮。

(16) 此时软件会提示已准备好创建虚拟机,单击“完成”按钮,系统会自动开启此虚拟机。



图 5.2.13 指定磁盘文件



图 5.2.14 安装完成

(17) 新建的虚拟机开启后会进入 Windows Setup 界面,如图 5.2.15 所示。然后,虚拟机自动安装好系统。

(18) 系统安装过程会比较长,不过基本不需要操作就会将 Windows Server 2003 系统安装到该虚拟机中,安装 Windows Server 2003 界面如图 5.2.16 所示。