|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名：崔文帅 | | 学号：2025060163 | 专业年级： 20网工 | 班级： 网三 |
| 分组： | | 实验室：402 | 指导教师：郭念 | 实验日期：2023-05-05 |
| **实验的准备阶段**  **(指导教师填写)** | **课程名称** | **网络协议分析与攻防技术** | | |
| **实验名称** | 实验十 缓冲区溢出攻击 | | |
| **实验目的** | 1) 了解程序调入内存中的分段情况  2) 掌握堆、栈、BSS段缓冲区溢出的原理  3) 了解缓冲区溢出常见的攻击手法  4) 掌握利用MS08067远程溢出工具发送特殊构造的数据包,使目标主机产生溢出，然后执行shellcode代码，,打开目标主机端口，使用telnet连接目标端口，获得系统权限。 | | |
| **实验内容** | 1) 学习溢出工具的代码构成。  2) 学习堆溢出的原理。  3）MS08-067远程溢出漏洞的原因是由于Windows系统中RPC存在缺陷造成的，Windows系统的Server服务在处理特制RPC请求时存在缓冲区溢出漏洞，远程攻击者可以通过发送恶意的RPC请求触发这个溢出，如果受影响的系统收到了特制伪造的RPC请求，可能允许远程执行代码，导致完全入侵用户系统，以SYSTEM权限执行任意指令并获取数据，并获取对该系统的控制权，造成系统失窃及系统崩溃等严重问题。 | | |
| **实验类型**  （打☑） | ☑验证性 □演示性 □设计性 □综合性 | | |
| **实验的重点、难点** | **缓冲区溢出攻击** | | |
| **实验环境** | 实验任务一  工具：C:\实验工具集\02\_主机安全\03\_远程缓冲区技术  实验工具：溢出演示工具  实验任务二  绘图1  目标机：192.168.1.3  工具: C:\toos\MS08067利用方法 | | |
| **实验的实施阶段** | **实验步骤及实验结果** | **实验任务一：Windows堆溢出**  **一、原理描述**  1.1 这里引入一个基于堆的缓冲区溢出的例子，首先介绍例子功能：在一个堆栈里边申请两块存储空间，处于低地址的buf1和处于高地址的buf2。在buf2当中，存储了一个名为myoutfile的字符串，用来存储文件名；buf1用来接收输入，同时将这些输入字符写入到buf2存储的文件名myoutfile所指向的文件中。如图1、图2所示  http://10.11.52.15/upload/init/initimage/1489403941194/image002.jpg  图 1  http://10.11.52.15/upload/init/initimage/1489403941194/image004.jpg  图2  通过malloc函数，在内存中申请了两个堆的存储空间。在这里要注意分配堆的存储空间时，存在一个顺序问题。buf2的申请命令虽然在buf1的申请命令之前，但是在运行过程中，内存空间中buf2是在高地址位，buf1是在低地址位。这个随操作系统和编译器的不同而不同。接着定义了diff变量，它记录了buf1和buf2之间的地址距离，也就是说buf1和buf2之间还有多少存储空间。fopen语句将buf2指向的文件打开，打开的形式是追加行，用了关键字“a”。即打开这个文件后，如果这个文件是以前存在的，那么写入的文件就添加到已有的内容之后；如果是以前不存在，用fprint语句将buf1中已经获得的语句写入到这个文件里，然后关闭文件。  1.2 进入堆溢出文件夹（C:\实验工具集\02\_主机安全\03\_远程缓冲区技术\第2节 Windows堆溢出\code1），双击运行sample.exe软件。**实验截图如下：**    从输出信息可以看到buf1和buf2都分配了内存地址，并且内存地址相差72字节，buf2内存储了字符串“myoutfile”，作为buf1变量接收字符串时存储文件的名称。  1.3 输入71个数字，因c语言默认会在字符串后面加入\0结束符来表示字符串输入结束，所以相当于占用了72个字节空间。那么buf1全部使用完毕，并存储在名称为myoutfile的文件里。**实验截图如下：**    1.4 从输出信息发现，软件没有出错。字符串正常写入文件。**实验截图如下：**    1.5 输入72个数字，因c语言默认会在字符串后面加入“\0”结束符来表示字符串输入结束。所以相当于占用了73个字节空间。那么buf1全部使用完毕，并且最后一个“\0”因buf1的内存空间用完，按内存地址增加的顺序存入下一个字节空间，即buf2的第一个字节空间存储了“\0”。当程序打开buf2变量所指定的文件名时，读到第一字节为“\0”,表示读取结束。程序读到值为空，出错。**实验截图如下：**    1.6 从输出信息中，可以看到buf2中没有数据，造成软件崩溃。如图7所示  http://10.11.52.15/upload/init/initimage/1489403941194/image014.jpg  图7  1.7 输入75个数字，因c语言默认会在字符串后面加入“\0”结束符来表示字符串输入结束。所以相当于占用了76个字节空间。那么buf1全部使用完毕，并且 “345\0”因buf1的内存空间用完，按内存地址增加的顺序存入下一个字节空间，即buf2的前四个字节空间存储了“345\0”。当程序打开buf2变量所指定的文件名时，读到为“345\0”,表示读取结束。程序buf2存储的文件名被覆盖为345。**实验截图如下：**    1.8 从输出信息可以看到buf2存储的文件名信息变成了345。如图9所示  http://10.11.52.15/upload/init/initimage/1489403941194/image018.jpg  图9     从内存分配来看，buf1填充了大于72个字节的字符串，余下的”345\0”就扩展到了buf2的空间之中。同样，字符串要以\0表示结束。但是原先的buf2中的内容也有一个\0表示字符串的结束，但是这个\0落在了”345\0”的后边，所以系统当看到”345”后边的\0时就认为字符串结束了，所以输出的是345。而读取buf1的内容时候，到存储空间结束也没有遇到\0，那么它就继续往下读，直到遇见了\0，所以它读取的长度已经超过了它本身分配的存储空间的长度。这样就构造了一个新的文件名覆盖了原先的内容，从而输出到一个我们定制的文件中，产生了基于堆的溢出。  **实验任务二：MS08067利用方法**  **一、运行MS08067远程溢出软件**  1.1 单击“开始”-〉“运行”，在弹出的对话框中输入“cmd”,打开命令行界面。**实验截图如下：**    1.2 在命令行界面输入”cd  C:\tools\MS08067利用方法”。**实验截图如下：**    1.3 输入命令”MS08067.exe”,显示工具使用方法。**实验截图如下：**    1.4 输入命令“MS08067.exe  192.168.1.3”,工具向远程主机192.168.1.3发送一串特殊构造的数据包，使目标主机产生内存溢出，程序跳转到shellcode代码处执行，shellcode代码在目标主机上开启4444端口，等待连接。**实验截图如下：**    **二、连接远程主机并添加用户**  2.1 输入命令“telnet 192.168.1.3 4444”。**实验截图如下：**    2.2 成功连接目标主机，输入命令“ipconfig”,查看远程主机IP信息。**实验截图如下：**    2.3 输入命令“net user aaa 123456 /add”在远程主机上添加帐户aaa，密码：123456。**实验截图如下：**    2.4 输入命令“net localgroup administrators aaa /add”把帐户aaa添加到管理员组。**实验截图如下：**    2.5 输入命令“netstat -an”查看远程主机开放端口，发现3389已经打开。**实验截图如下：**    2.6 输入命令“net user aaa”查看帐户aaa是否已经输入管理员组。**实验截图如下：**    2.7 单击“开始”-〉”运行“->输入命令”mstsc“,打开远程桌面窗口。**实验截图如下：**    2.8 输入账号“aaa“和密码”123456“，登录目标主机。**实验截图如下：**    2.9 使用aaa账号登录成功。**实验截图如下：** | | |
| **实验结果的处理阶段** | **实验结果的分析与总结** | 1. 堆喷射技术有什么含义？   利用堆的特性和漏洞，向内存中预先分配大量恶意代码，来提高攻击的成功率。  **2．**为什么溢出成功后，“telnet 目标IP 4444 “连接的是4444端口，可以是其它端口吗？  不仅仅可以是4444端口，选择目标系统开放的，未被占用的端口即可 | | |