缓冲区溢出原理浅析以及防护 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/264/2021_2022__E7_BC_93_E5_86_B2_E5_8C_BA_E6_c97_264477.htm 【摘要】本文从程序语言本身的缺陷,不够健壮的角度出发,详细分析了缓冲溢出攻击的基本原理,描述了黑客利用缓冲区漏洞进行系统攻击的一般过程,最后又简单讨论了几种防范溢出攻击的策略. 【关键字】缓冲溢出. 程序跳转. 长跳转缓冲区. 近些年来,黑客攻击事件频繁发生,尤其是缓冲区溢出漏洞攻击占据了网络远程攻击的绝大多数. 因为这类攻击可以使任何人获得系统主机的完全控制权,所以它代表了一类十分严重的攻击. 缓冲区溢出攻击之所以常见,是因为它太常见了,且易于实现, 这完全是软件发展史上不可避免的问题. 缓冲区漏洞是程序员在编写程序时未检查内存空间,导致内存泄漏而引起,以下我们先来简单了解一下它: 一、认识缓冲区溢出 缓冲溢出是一种系统攻击的手段,借着在程序缓冲区编写超出其长度的代码,造成溢出,从而破坏其堆栈,使程序执行攻击者在程序地址空间中早已安排好的代码,以达到其目的. 一般黑客攻击root程序,然后执行类似exec(sh)的代码获得root的shell. 它造成了两种严重的后果: 1. 覆盖堆栈的相邻单元. 使程序执行失败, 严重可导致系统崩溃. 2. 可执行认识指令代码,最后获得系统root特级权限. 现在很多人使用C或C 编写程序,但同时太多的人忽略了对其的数组边界检查和类型安全检查,所以现今的大多数溢出都和C语言有关, C语言中中有可能产生溢出的函数有:char s[n],strlen(s),strcpy(dst, src),p=malloc(n),strcat(s,suffix)等等,所以我们要尽可能地避免使用这些危险函数,即使使用,也一定要做

严格的检查.为容易理解,我们来看一个简单的程序: /* * example.c * written by Devil_Angel * gcc o example example.c */ void func(char * str) { char buf[8]. strcpy(buf, str). printf(“ %sn ” ,buf). } int main(int argc, char * argv[]) { If(argc >1) Func(argv[1]). }//end of main 该程序在输入时,并没有对str的大小进行检查便直接送入数组buf,一旦输入超出buf长度,就产生了最简单的溢出,当然象这样的溢出一般只会出现Segmentation fault错误,而不能达到攻击的目的. 这里并没有进一步深入分析,只是让大家对溢出有一个大概的概念,在以后将会对其做进一步的分析.