缓冲区溢出机理及攻击分析

刘永艳， 殷肖川， 邓 军
（空军工程大学 电讯工程学院，陕西 西安 710077）

摘 要：以缓冲区溢出攻击的基本机理为出发点，着重分析缓冲区溢出攻击的步骤及分类，并进行实验分析，最后给出几点预防缓冲区溢出的防范措施。

关键词：缓冲区溢出；机理；攻击分析
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-3516(2003)05-0071-04

计算机技术的不断发展和计算机的广泛应用，促进了社会的进步和繁荣，并为人类创造了巨大的财富。但是，随着计算机在国家政治、军事、经济和金融等社会各领域的使用，大量的机密数据资料存储在计算机系统的数据库中。而计算机系统自身的脆弱性和人为的恶意攻击严重地威胁着计算机系统的安全，特别是计算机互连网络应用的广泛普及，人们面临着越来越多的安全漏洞。在这些漏洞中，缓冲区溢出漏洞是最为常见的一种形式。

通过缓冲区溢出攻击计算机系统，最著名的例子是“蠕虫程序”，它于1988年利用缓冲区溢出感染了英特网中的数万台机器，从而引起了人们对于计算机病毒的高度重视。一个匿名的internet用户，通过缓冲区溢出攻击，可以得到一台正常运行主机的部分或全部的控制权，获得控制权的“黑客”便可进行各种破坏。所以，缓冲区溢出是“黑客”最常用的一种攻击手段，也是目前极为严重的一类安全威胁。

1 缓冲区溢出的机理

缓冲区是指在内存中开辟的一个连续的块，它是计算机运行程序时 缓冲数据的地方，它保存了给定类型的数据[1]。

在计算机系统中，为了节约内存空间，经常采用动态变量和缓冲区的动态分配，它是指在程序运行的过程中动态分配缓冲区。如果在分配过程中不进行相关检查，就有可能发生缓冲区溢出问题。当程序试图将数据放到计算机内存中的某一位置，而该位置没有足够空间时会发生溢出，或者当程序在动态分配的缓冲区中放入很长的数据时，也会发生溢出，溢出部分漏到了其它地方。仅有缓冲区溢出，并不一定产生安全问题，只有将溢出部分送到能够以root权限运行命令的区域才有可能威胁安全。如果一个缓冲区操作程序，将一段能够运行的指令放在了有root权限的核心内存中，且该段程序指令一旦运行，那么它就以root权限控制了计算机。

C语言中，strcpy(buffer,str)的功能是把str所指的内容copy到buffer中，如果str所指的字符串长度比buffer变量的空间大得多，就会造成buffer的溢出，使程序运行出错。如果这个缓冲区是在系统的核心空间中，那么多出来的字符将会覆盖关键字段，导致程序运行失败、损坏系统等后果。更为严重的是，可以利用它执行非授权指令，甚至可以取得系统特权，进而进行各种非法操作。存在类似问题的标准函数还有strcat()、sprintf()、vsprintf()、gets()、scanf()等。

上述错误不仅会在普通程序员编程中发生，而且在一些由UNIX、Windows、路由器、网关等联网设备构成的系统中也时有发生。总而言之，在系统设计时，略有不慎都会引发缓冲区溢出。因此，在进行软件开发

收稿日期：2003-06-19
作者简介：刘永艳(1979-)，女，江苏连云港人，硕士生，主要从事网络安全技术研究.

时,要尽量避免以下问题:

1)使用安全性较差的语言,且不进行数组边界检查和类型安全检查。如 C/C + + 允许开发人员创建非常接近硬件环境运行的程序,允许直接访问内存和寄存器,若不执行数组边界检查和类型安全检查,就可能会发生缓冲区溢出。

2)采用不安全的方式访问或复制缓冲区。如果应用程序从用户(或攻击者)那里获取数据,并将数据复制到应用程序所维护的缓冲区中,且未考虑目标缓冲区的大小,则可能造成缓冲区溢出。

3)将缓冲区放在内存中关键数据结构旁边或邻近的位置(重要的数据结构包括 C + + V 表、异常处理程序地址、函数指针等)。例如,当某个函数的缓冲区紧邻堆栈,该函数的返回地址紧靠在缓冲区之后。如果攻击者使该缓冲区发生溢出,覆盖函数的返回地址,那么在函数返回时,就可以返回到攻击者定义的地址,达到控制计算机的目的。

2 缓冲区溢出攻击分析

缓冲区溢出攻击的根本目的就是扰乱某些特权操作程序的功能,以便攻击者控制该程序,若此程序具有足够的权限,那么攻击者就可以控制整个计算机系统[2]。要实现这一目的,必须具备两个条件:一是攻击者要在程序的地址空间中插入其执行代码。二是通过适当的改变相关寄存器和内存单元的内容,让程序跳转到入侵者安排的地址空间执行。因此,缓冲区溢出攻击实际上是分为两步进行。

2.1 在地址空间插入代码

在地址空间插入攻击代码有两种方法:

1)植入字符法。

攻击者向被攻击的程序输入一个可运行的指令序列字符串,该程序会把这个字符串放到缓冲区中。缓冲区可能设在任何地方:堆栈(stack,自动变量)、堆(heap,动态分配的内存区)和静态资料区。

2)利用已有代码法。

攻击者所期望的代码已经在被攻击的程序中了,攻击者只需对代码传递一些参数。比如,在 UNIX 环境下,libc 库中的代码执行"exec (arg)",其中 arg 是一个指向字符串的指针参数,若攻击代码要求执行"exec ("/bin/sh")",那么攻击者只需把传入的参数指针改成指向"/bin/sh"。

2.2 将控制程序转移到攻击代码

攻击者将攻击代码植入程序后,就要寻求改变程序的执行流程,使之跳转到攻击代码。最基本的方法就是溢出一个没有边界检查或者存在其它弱点的缓冲区,扰乱程序的正常执行顺序。通过溢出一个缓冲区,攻击者可以直接跳过系统的检查,强制改写相邻的程序空间。

被攻击程序缓冲区的空间大小在原则上是任意的,攻击者就是利用这一点来强制改变程序指针。然而,在攻击的具体实施过程中,攻击者的切入点会随着程序空间和内存空间的定位不同而改变,主要有以下三种:

1)活动纪录(Activation Records)。

一个函数调用发生时,调用者会在堆栈中留下一个活动纪录,它包含了函数调用结束时的返回地址。攻击者通过溢出堆栈中的自动变量,使返回地址指向攻击代码。这样,当函数调用结束时,程序就会跳转到攻击者设定的地址,而不是原先的地址。这类攻击被称为堆栈溢出攻击(Stack Smashing Attack)[3],是目前最常见的缓冲区溢出攻击方式。

2)函数指针(Function Pointers)。

函数指针可以用来定位任何地址空间。例如:"void (* f)()"声明了一个返回值为 void 的函数指针变量 f。所以攻击者只需在函数指针附近找到一个能够溢出的缓冲区,然后溢出这个缓冲区来改变函数指针,当程序通过函数指针调用函数时,程序的流程就按攻击者的意图改变了。

3)长跳转缓冲区(Longjmp buffers)。

在 C 语言中包含了一个简单的检验/恢复系统(setjmp/longjmp)。意思是在检验点设定"setjmp(buffer)",用"longjmp(buffer)"来恢复检验点。如果攻击者能够进入缓冲区空间,那么"longjmp(buffer)"就可以跳转到攻击者的执行代码。象函数指针一样,longjmp 缓冲区能够指向任何地方,所以攻击者所要做的就是

找到一个可以溢出的缓冲区。一个典型的例子是 Perl 5.003 的缓冲区溢出漏洞,攻击者首先进入用来恢复缓冲区溢出的 longjmp 缓冲区,然后诱导被攻击程序进入恢复模式,这样就使 Perl 的解释器跳转执行攻击代码。

3　缓冲区溢出攻击实验分析

为了验证缓冲区溢出机理及攻击分析,我们对微软 IIS4.0 一个溢出漏洞进行了攻击实验。实验原理:IIS 将整个的 URL 地址传给后缀名为. ASP、IDC、HTR 的 DLL。如果 ISAPI DLL 不进行严格的边界检查,就会产生一个缓冲区溢出,通过 IIS(inetinfo. exe)可以执行远程计算机上的任意代码。根据这一原理,利用 eEye - Digital Security Team 开发的 Retina 安全扫描器逐一使用这些后缀,探测结果发现存在这样的漏洞。在发送"GET /[overflow]. htr HTTP/1.0"后,被攻击的实验服务器没有了反应。使用调试器分析,发现有 3K 的缓冲区。

溢出解释:这个溢出和. HTR 后缀有关。. HTR/ISM. DLL ISAPI 过滤器缺省安装在 IIS 服务器上,IIS 包含了允许 Windows NT 用户通过 web 目录/iisadmpwd/改变口令的功能,它是由一系列的. HTR 文件和 ISAPI 后缀文件 ISM. DLL 实现的。在将 URL 传递给 ISM. DLL 的某个地方时,没有进行边界检查,于是就发生了溢出。攻击方法:利用上述缺陷,eEye 编写了两个程序: iishack. exe 和 ncx99. exe。把 ncx99. exe 拷贝到攻击者的 web 服务器上。ncx99. exe 是一个改进 netcat. exe 的特洛伊木马程序,它将 -l -p 80 -t -e cmd. exe 作为一个固定的参数运行,始终将 cmd. exe 绑定在 99 端口上。

假设攻击者的 web server 是:www. sev. com,被攻击者的 IIS server 是 www. xxx. com 。运行下面的命令:iishack www. xxx. com 80 www. sev. com/ncx99. exe

等待足够多的时间,攻击者就可以利用这一漏洞在被攻击的实验服务器上留下后门,再用 Telnet 操作进行攻击:

Telnet www. xxx. com 99

这时攻击者已经顺利进入了被攻击的计算机,可以进行任何破坏性操作。如果想要退出,只需键入 exit。

以下系统中都存在这一漏洞:internet Information Server 4.0 (IIS4),Microsoft Windows NT 4.0 SP3 Option Pack 4,Microsoft Windows NT 4.0 SP4 Option Pack 4 和 Microsoft Windows NT 4.0 SP5 Option Pack 4。目前,Internet 上 90% 的 NT Web 服务器运行的都是上述系统,所以这一漏洞造成的后果相当巨大。

4　缓冲区溢出攻击防范措施

目前,有四种保护缓冲区溢出攻击的基本方法。

4.1　编写正确的代码

正确的代码是性能和正确性的统一,尤其是编写诸如 C 语言那种易出错的程序,更要严谨。目前可以采用一些技术和工具来帮助程序员进行漏洞检查,其中最简单的方法是用 grep 来搜索源代码中容易产生漏洞的库调用,如前例中的 strcpy。虽然有些程序能够通过代码安全检查,但可能仍有缓冲区溢出的问题。因此,人们开发了一些高级的查错工具,如 fault injection 等。使用这些工具,可以随机地产生一些缓冲区溢出来寻找代码的安全漏洞。还有一些静态分析工具用于侦测是否存在缓冲区溢出。

4.2　缓冲区的非执行技术

在计算机系统中,缓冲区的非执行技术,就是仅允许执行程序段地址空间代码,禁止执行数据段地址空间代码。由于缓冲区在数据段地址空间,既使攻击者把攻击代码殖入到程序输入缓冲区,它也是被禁止执行,不产生任何破坏作用。非执行堆栈技术,可以有效防范将攻击代码殖入自动变量缓冲区,并通过缓冲区溢出的攻击。事实上,很多老的 Unix 系统都是这样设计的,但是近来的 Unix 和 MS Windows 系统为了实现更好的性能和功能,在数据段中允许动态地放入可执行代码。因此,为了保持程序的高性能和兼容性,不能禁止执行所有程序的数据段地址空间代码。但是我们可以设定堆栈数据段不可执行,这样就可以最大限度地保持程序的高性能、兼容性和安全性。

4.3 数组边界检查

数组边界检查是对数组的读写操作进行严格的检查,确保对数组的操作是在正确的范围内[4-5]。目前有以下几种检查方法:Compaq C 编译器(由康柏公司开发),存储器存取检查(purify 工具)和选用类型-安全语言。

4.4 程序指针完整性检查

程序指针完整性检查和边界检查略有不同。程序指针完整性检查,就是在指针被引用之前检测它是否发生过改变,即使一个攻击者成功地改变了程序的指针,由于系统事先检测到了指针的改变,因此这个指针将不会被使用。程序指针完整性检查通常有三种方法:一是堆栈检测,它能通过检测 cpu 堆栈来确定缓冲区溢出;二是堆栈保护,这是一种提供程序指针完整性检查的编译器技术,通过检查函数活动纪录中的返回地址来实现。三是指针保护,通过在所有的代码指针之后放置附加字节来检验指针在被调用之前的合法性,如果检验失败,会发出报警信号,并退出执行程序。

5 结束语

综上所述,缓冲区溢出是目前许多操作系统和软件产品开发中的常见问题,针对缓冲区溢出的攻击,已经给人类社会带来了一次次的灾难,给计算机网络系统带来了极大的隐患,严重威胁着网络安全。如何安全地使用计算机,预防和打击计算机犯罪,在普及计算机知识和应用的今天,已成为十分紧迫的任务。

参考文献:

[1] 凌雨欣,常 红. 网络安全技术与反黑客[M]. 北京:冶金工业出版社,2000.
[2] 谭敏安. 网络攻击防护编码设计[M]. 北京:希望电子出版社,2002.
[3] Eric Cole. Hacker beware[M]. IN:. New Riders Publishing,2002.
[4] 汪立东,方滨兴. UNIA 缓冲区溢出攻击:技术原理、防范与检测[J]. 计算机工程与应用,2000,(2):12-14.
[5] 雷英杰,赵 晔,邓 宁. 面向大型网络的防火墙系统设计[J]. 空军工程大学学报(自然科学版),2001,2(6):34-36.

(编辑:田新华)

Mechanism and Attacking Analysis of Buffer Overflow

LIU Yong-yan, YIN Xiao-chuan, DENG Jun
(The Communication Engineering Institute, Air Force Engineering University, Xi'an, Shaanxi 710077, China)

Abstract: Based on the fundamental mechanism of buffer overflow attacking, the process and classification of buffer overflow attacking are analyzed in this paper, simultaneously the experimental analysis is made and several measures used to protect buffer overflow are listed in the end.

Key words: buffer overflow; attack

(上接第 70 页)

Analysis of Implementing Performance for Typical Parallel Algorithms

LEI Ying-jie[1], HUO Hong-wei[2]
(1. The Missile Institute, Air Force Engineering University, Sanyuan, Shaanxi 713800, China; 2. School of Computer Science and Engineering, Xidian University, Xi'an, Shaanxi 710071, China)

Abstract: This paper discusses and analyzes the programming problems with C++ description under Windows XP-based environments supported by message passing interface (MPI), which is a parallel programming tool, and implementing performance of several typical parallel algorithms and their variations for processing. Still it presents in detail the computed results in the parallel programs and analyzes a comparison of their computing performances and the effect of them on precision in calculation.

Key words: parallel computation; MPI; parallel algorithms; high performance computation