安全技术经典译丛

Buffer Overflow Attacks
Detect, Exploit, Prevent

缓冲区溢出攻击
——检测、剖析与预防

(美) James C. Foster 等著
蔡 勉 译

清华大学出版社

缓冲区溢出攻击

—— 检测、剖析与预防

(美) James C. Foster 等著

蔡 勉 译

清华大学出版社

北 京

James C. Foster et al
Buffer Overflow Attacks:Detect,Exploit,Prevent
EISBN：1-932266-67-4

北京市版权局著作权合同登记号　图字：01-2005-2882

图书在版编目(CIP)数据

缓冲区溢出攻击——检测、剖析与预防/(美)福斯特(Foster, J. C.)等著；蔡勉译.
—北京：清华大学出版社，2006.12
书名原名：Buffer Overflow Attacks:Detect,Exploit,Prevent
ISBN 7-302-13942-3
Ⅰ.缓… Ⅱ.①福…②蔡… Ⅲ.计算机网络—安全技术 Ⅳ.TP393.08

中国版本图书馆 CIP 数据核字(2006) 第 120372 号

责任编辑：曹　康　徐燕萍
装帧设计：孔祥丰
责任校对：成凤进
责任印制：杜　波
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：北京鑫丰华彩印有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：26.25　**字　数**：606 千字
版　　次：2006 年 12 月第 1 版　　**印　次**：2006 年 12 月第1 次印刷
书　　号：ISBN 7-302-13942-3/TP·8382
印　　数：1～4000
定　　价：49.00 元

序　言

对于缓冲区溢出，有三件事令我印象深刻：第一次成功地在 Linux 的 imapd 上利用一个缓冲区溢出；第一次在本地的 Linux 中独立发现和利用了缓冲区溢出；第一次通过编写缓冲区溢出成功进入别人的主机。

在读过 Aleph1 的关于缓冲区溢出的重要论文“Buffer overflows for fun and profit”后，大多数人想到的主要是由此带来的好处。因为精通编写缓冲区溢出程序的人在该行业任何一家大公司中做咨询师的年薪都可以达到 9～12 万美元。

但另一方面，许多人对获得这种技能有一个很大的误区，认为学会这种技能，就可以一劳永逸了。确实，在 IT 领域许多诀窍是可以学会的，一旦了解了这些诀窍，也就拥有这些诀窍了。但是编写缓冲区溢出程序却并不如此，从书本中或者速成班中可以学会基础知识，但是编程的环境总是在变化。一方面，黑客在不停地寻找新的方法来更好的利用漏洞，寻找发现新的漏洞方法；另一方面，微软公司每天都在给它的代码增加保护，如果 3 个月不写缓冲区溢出代码，原有的技巧也就过时了。编写缓冲区溢出中最难的就是要地根据不断变化的环境给出新对策。

用于编写溢出程序的工具也在不断变化着，以前编写溢出程序只需要一份 Softice 或 GDB 的拷贝，就可以由某个人单独完成。但是今天，即使是一个简单的缓冲区溢出，Immunity 公司也会有相当大的投入。例如，需要有专门的调试器来查询、脚本化正在运行的程序；专门的编译器来创建和调整克服弱点所需求的 shellcode；购买或产生用于专门解决各种不同问题的反向工程工具；用 Python 语言编写的完整的 mySQL 和 SSL 库。一个相对复杂的漏洞利用需要整个工作小组协调完成。

每种复杂的漏洞利用都会有相应的文章介绍，这些漏洞利用来自于整个团队在不同漏洞利用过程中所得到的经验。

最好的缓冲区溢出程序决不会是蠕虫。攻击者定制的漏洞利用会使被攻击者身陷其中。如果一个顶级黑客要攻击某个人，则会完全掌握目标主机的工作环境，目的只有一个，产生一个只使用一次的缓冲区溢出。

编写缓冲区溢出程序有几个阶段。本书作者 James Foster 介绍了一些基础的知识和技巧，对初学者进行基础的训练，确定专攻的方向，然后就可以独自编写缓冲区溢出程序。虽然本书不能使读者站到技术的最前沿，但是能够确信自己掌握了基础知识，能够做出正确的决定。也许读者可能投身于此项工作，致力于编写代码、提高技能。对于选择这一行的读者，请记住下面的座右铭：

- 永远不要害怕。微软的销售人员在不停地告诫人们发现微软的新软件的缓冲区溢出漏洞和编写相应的漏洞利用程序是件非常困难的事情。激励自己继续做下去的对策就是想像一旦漏洞利用成功，自己将会如何处理这个漏洞。编写漏洞利用程序需要掌握很多单调枯燥的技术，例如，HP-UX 所用的少见的扇形内存访问方

式；Irix 带有的笨拙的高速缓冲存储器。虽然编写缓冲区溢出程序需要数千条的汇编语句，学起来并不是一件容易的事情，但只要自己觉得能做到，那就一定能做到。

- 不要太把自己当回事。无论自己多么优秀，在遥远的地方可能还有一些十五岁的年轻人每天花 20 小时来争取做得更好。不要把编写缓冲区溢出程序当成一场竞争，否则你不久就会崩溃。
- 找一些伙伴。编写缓冲区溢出程序不是独自能不断进步的一个技能，需要别人的帮助，找出自己在哪方面还比较薄弱。
- 不管目标是什么，要把这本书当作工作表，而不是一本小说。要一边读一边在电脑上操作。一本缓冲区溢出的书不能造就出一个高明的黑客。在逐章学习的过程中，会发现在漏洞利用不起作用时，自己会去不断地尝试，会废寝忘食，会不惜花金钱来更准确地掌握所学到的知识。
- 我的观点是：漏洞利用是事实的复杂陈述，如果你赞同这个观点，将使编写缓冲区溢出程序变得更加美妙。

希望有一天能像欣赏艺术品一样欣赏你的代码。

—Dave Aitel

Immunity 公司创始人兼 CEO

目　录

第 I 部分　扩展缓冲区溢出

第 II 部分 缓冲区溢出解析

第III部分 查找缓冲区溢出

第Ⅰ部分

扩展缓冲区溢出

第 1 章

缓冲区溢出的基本概念

本章内容包括：

- 软件安全危机
- 缓冲区溢出的增加
- exploits 与缓冲区溢出的比较
- 定义

1.1 简介

当代信息技术的许多领域中，缓冲区溢出与漏洞(vulnerability)是同义的；一些情况下，也与 exploits 同义。这是一个令人惊慌的词，它不仅使人怀疑是否购买了最好的防火墙、是否正确地配置了基于主机的入侵防御系统并对整个系统环境进行了修补，而且这会比最好的 McAfee 反病毒软件或 Symantec 的最新成果更能引起人们对于安全防护的讨论。缓冲区溢出证明了计算机科学或软件编程组织仍然没有理解(或是真实地认识)如何设计、创建和实现安全代码。

不管怎样，所有缓冲区溢出都是结构较差的软件程序的产品。这种程序有多种不足，譬如堆栈溢出、堆腐烂(heap corruption)、格式化字符串缺陷和竞争条件(race condition)，前三种通常被作为简单的缓冲区溢出。缓冲区溢出可以简单到在百万行的程序中只出现一个放错的字符，也可以复杂到对多维字符数组的不合理处理。缓冲区溢出可以出现在本地程序中，例如，日历程序、计算器、游戏和 Microsoft 的办公软件；也可以出现在远程软件中，例如，E-mail 服务器、FTP、DNS，以及曾流行的 Internet Web 服务器。

黑客通常会攻击系统中最薄弱的环节，基于这样的思想，就可以想象出最流行的软件集合中包含着许多已经得到确认的漏洞。从另外一个角度看，因为最流行软件会得到更多的关注，所以也就增加了在流行软件发现更多缺陷的机会。

如果读者目标比较简单，只是希望对这个话题进行简单的讨论，那么阅读完第一章后就可以达到这个目标。然而，如果读者是富有抱负并热切希望迎接更大挑战的人，那么本章只是为读者作结构性介绍，后面会有更详细的内容。在阅读完这些之后，相信读者不会成为一个超级黑客(uber-hacker)或 exploit 编写者，但读者将会在得到很少或没有帮助的情况下使用后面得到的工具和知识来阅读、分析、修改和编写自己的缓冲区溢出。

1.2 软件安全危机

软件工程是一项相当困难的任务，在所有和软件制造相关的职业中，软件架构师的任务可能是最困难的。最初，软件架构师只负责软件产品的高级设计，通常包括协议的选择、第三方部件的评估和选择，以及通信介质的选择。毫无疑问，这些对于任何架构师来说都是很有价值和必需的目标，但是现在软件架构师的任务变得更加困难了，需要熟悉操作系统、软件语言和它们在不同平台下的内在优点和缺点。另外，软件架构师还面临着逐渐增加的压力，要设计能阻止精明黑客渗透的复杂软件，但这几乎是不可能的。

Gartner Research已经声明：软件和应用层的漏洞、入侵和入侵尝试的数目在各种环境中都在增加。由于只使用了少量的精密、自动应用程序软件扫描器和入侵检测系统，该声明及相关的统计是不准确的。基于软件的漏洞，特别是出现在Web上的漏洞是很难识别和检测的。当利用攻击包和系统响应来分析SQL攻击、认证强力技术(authentication brute-forcing technique)、目录游走(directory traversals)、cookie中毒、跨网站脚本(cross-site scripting)和逻辑缺陷时，它们与正常的或不怀恶意的HTTP请求有着惊人的相似。

今天，在对企业网络的攻击中，超过 70%来自于应用层，而非网络层或系统层。

——*The Garter Group*

如表 1-1 所示，对于非服务器(non-server)应用软件攻击的增加已经很久了。该表的建立使用了由政府资助的 Mitre 公司所提供的数据。Mitre 公司作为漏洞信息的文档化和目录化方面的世界领先者已经超过 5 年了。SecurityFocus 公司(由 Symantec 公司掌控)在漏洞信息的收藏和目录编制方面是 Mitre 公司惟一的竞争对手。尽管 SecurityFocus 公司比 Mitre 公司拥有更好的漏洞文件，但两个公司都拥有数以千计的漏洞的文件档案和索引。

表 1-1 漏洞数据表

分 布 比 例	2004	2003	2002	2001
操作系统	124 (15%)	163 (16%)	213 (16%)	248 (16%)
网络协议堆栈	6 (1%)	6 (1%)	18 (1%)	8 (1%)
非服务器应用程序	364 (45%)	384 (38%)	267 (20%)	309 (21%)
服务器应用程序	324 (40%)	440 (44%)	771 (59%)	886 (59%)
硬件	14 (2%)	27 (3%)	54 (4%)	43 (3%)
通信协议	28 (3%)	22 (2%)	2 (0%)	9 (1%)
加密模块	4 (0%)	5 (0%)	0 (0%)	6 (0%)
其他	5 (1%)	16 (2%)	27 (2%)	5 (0%)

非服务器应用程序包括Web应用程序、第三方组件、客户应用程序(比如FTP和Web客户机程序)，以及包括media播放器、控制台游戏(console games)的所有本地应用程序。读者可能想知道这些漏洞有多少是由于较差的架构、设计或实现产生的。

Oracle 的 Larry Ellis 对于 Oracle 的近乎神话的安全特性和无风险状态做了大量的声明，但每一项都被证明是错误的。尤其在其所提及的拥有“无漏洞”特性的 Oracle 8.x 软件中，这一点更是千真万确，因为后来在 Oracle 8.x 中发现了大量的缓冲区溢出、SQL 注入攻击(injection attack)和众多的接口安全问题。上面的例子说明：绝对安全是不可实现的目标。

更合理的建议是在开发、设计、实现软件时，采用几个小的且可以实现的具有特殊安全特性的步骤来逐步实现安全目标。希望在产品的发布版本上只发现几个漏洞是不现实的。如果产品经理或开发经理将此定为团队目标，这些经理是应该被解雇的。以下是一个更现实、更简洁且更好的目标。

- 开发那些用户在输入中无法提供漏洞的软件；
- 开发那些没有回避认证(authentication bypassing)漏洞的软件；
- 将测试版本向所有基于 URI 的漏洞开放；
- 开发不依赖第三方应用程序安全的软件(架构师的部分工作就是评估第三方组件的安全性和计划是否可靠)，因为第三方应用程序可能会产生漏洞。

Microsoft 软件也是有缺陷的

令人吃惊的是，不同的Microsoft应用软件已被证明含有不同的软件漏洞。在这里，作者不想加入抨击Microsoft的行列中。综合各种因素，应该说Microsoft公司已经深深领会了安全漏洞的意义，并在产品发布前对漏洞的修补做了出色的工作。作为一名在该领域研究漏洞与安全很多年的资深研究员，个人认为Microsoft公司的这种安全漏洞是最受欢迎的那类漏洞。Microsoft公司的产品在市场上处于主导地位且拥有庞大的用户群，基于这个事实，名称识别(name recognition)也伴随着寻找Microsoft漏洞而出现。与在拥有数千万用户的Windows XP中找出一个漏洞相比，在只拥有100个用户的Mike Spice CGI中找出一个漏洞是微不足道的事情，因为目标用户增加了数万倍。

接下来…

漏洞和远程代码执行

要想在安全界出名，最简单的方法是找出远程代码执行中的 Microsoft 软件的重大漏洞。可以通过将漏洞的详细细节公布在许多安全邮件列表和所知道的 BAM！上来实现。最难的部分是粘贴上您的姓名。可以通过发行物来扩大您的名声，例如，通过编写公开的源代码工具、在会议上发表演讲或者对最新的重大漏洞的信息穷追到底。如果您在一年里发现并发布了 10 个主要的漏洞，那么您就已经迈向可成名的大道上。

Microsoft 公司似乎每天都会确认和发布一种新的缓冲区溢出，但这种确认和发布过程已得到明显改进。Microsoft 公司每个月对漏洞发布一次，以减轻修补美利坚公司(corporate America)的痛苦。即使拥有所有可以自动完成和简化修补问题的新技巧，这仍然是一个问题。Citadel's Hercules、Patchlink、Shavlik，甚至 Microsoft 的修补服务器都被设计为只需按一下键就可修复漏洞。

图 1-1 所示的是一个典型的为一个严重漏洞而建立的 Microsoft 安全公告，该漏洞允许执行远程代码。10 个漏洞中有 9 个是这样的，但是 Microsoft 的一个远程代码执行漏洞仅仅是一个漏洞而已。在本书的后面，不仅会学习到如何利用缓冲区溢出漏洞，也要学习如何寻找它们，这样才会掌握最具经济利益的信息安全技巧。

远程代码执行漏洞会很快变体为具有自动能力的威胁，比如网络派生的病毒或著名的网络蠕虫。Sasser 蠕虫及其变体曾经是网络世界中最具破坏性和引起损失最大的蠕虫之一。它通过一个在许多 Microsoft 操作系统中存在的严重的缓冲区溢出进行扩散。通常，蠕虫及其变体是最让人感兴趣的公开代码。

网络蠕虫包括下面的 4 个主要部分：

- 漏洞扫描；
- 漏洞可利用；