<<黑客防线2009缓冲区溢出攻击>>

图书基本信息

书名：<<黑客防线2009缓冲区溢出攻击与防范专辑>>

13位ISBN编号：9787115204240

10位ISBN编号：7115204241

出版时间：2009-6

出版时间：人民邮电出版社

作者：《黑客防线》编辑部 编

页数：214

字数：379000

前言

Exploit、缓冲区溢出漏洞是什么？
它们有什么异同？
在网络安全技术飞速发展的今天，基于缓冲区溢出漏洞（以下简称漏洞）的研究已经越来越重要，无数网络安全爱好者开始关注网络安全技术的核心——漏洞发掘以及Exploit编写。
“黑客通过某某漏洞攻破某某大型系统”、“某某企业系统漏洞导致上亿元的商业损失”……见诸报端的各种新闻似乎都在强调一个问题：“漏洞”是黑客恶意攻击的核心，是黑客入侵的最重要的“门户”！
对于真正喜欢网络安全技术，并愿意深入研究技术的朋友来说，各种非专业信息当然是不准确的，我们需要了解它们的本质意义。
Exploit的英文本意是“利用”。
“漏洞”是在硬件、软件、协议的具体实现或系统安全策略上存在的缺陷，从而可以使攻击者能够在未授权的情况。
访问或破坏系统。
很多时候，许多网络安全专家都混淆了两者的概念——我们首先需要明确的是它们二者之间的关系：有漏洞不一定就有Exploit（利用），而有Exploit就肯定存在漏洞！
漏洞分析有何意义？
有何困难？
漏洞存在的本质原因是因为所有程序都是由人编写的，任何一种思维的惯性和不严谨性都可能造成致命漏洞的出现，而漏洞的存在意味着原以为安全的系统有了崩塌的可能。
当今网络的快速发展改变了人们的生活和工作习惯，人们已经习惯了将重要资料存放在计算机里，网络上也存在越来越多的秘密信息，这也是黑客们搜索漏洞并进行攻击的主要原因。
随着网络的发展，漏洞必将具有越来越大的危害性！
尽管网络安全技术防范人员和居心叵测的恶意攻击者，都知道漏洞的危害，然而真正拥有漏洞发掘和编写Exploit能力的人，却依然屈指可数，这涉及以下5个方面的核心问题。
漏洞的根本存在机理是什么？
漏洞的触发究竟需要什么条件？
Exploit的编码规范有什么不能违背的原则？
各种安全策略和安全审计的限制如何突破？
如何构造Exploit的核心shellCode？
本书解决的问题、本书的特点、本书的结构：常有网络安全爱好者抱怨发掘漏洞、构造Exploit是多么多么的困难，难以找到系统的、全面的、详尽而深入的漏洞发掘和Exploit编写教程，而想拥有一本代码经过严格测试，各种细节均由实战提炼得出，各种漏洞和相关Exploit都是完全可实际操作的书籍，却又是难上加难。
编纂本书的目的就是为了解决这些问题。
从经典的serv－U溢出，到迅雷、卡巴斯基漏洞存在机理分析，本书给出了30多个漏洞的触发机制、发掘过程，以及这些漏洞调试和Exploit编写的详尽技术精华，更有值得深入研究的每一个漏洞的核心ShellC0de的调试过程、实战构造。
考虑到网络安全爱好者的技术层次参差不齐，本书分为“初级篇”、“分析篇”和“Shellcode篇”，以详细分析过程附带完整源代码的方式，将每一步详细的分析过程呈现给读者，务求可操作性的完整体现，拥有本书的读者完全可以按照书中的内容自己完成相关漏洞的分析和Exploit的重塑！
本书所讲述的内容仅做学习之用。
切勿用于非法用途。
通过阅读本书.希望读者能够树立良好的网络安全意识，提高网络安全防御水平。

<<黑客防线2009缓冲区溢出攻击>>

内容概要

本书由国内最早创刊的网络安全类媒体之一《黑客防线》编纂。
全书秉承“在攻与防的对立统一中寻求突破”的核心理念，以30多个漏洞机理分析，触发机制，发掘过程，调试过程，ExPloit编写，核心ShellCode调试，实战构造为例，深入分析漏洞发掘的整个过程，详细解析Exploit的编写步骤。
本书分为“初级篇”、“分析篇”、“ShellCode篇”三篇，由浅入深地进行系统全面的缓冲区溢出攻击与防范的技术探讨，适合各层次的网络安全技术爱好者阅读。

<<黑客防线2009缓冲区溢出攻击>>

书籍目录

章节摘录

插图：system32＼services意外终止关闭提示框.根据以往的经验，我们可以看出该漏洞是存在于services进程中，这一点和work Station服务一样，因此归为后一类。
2.服务进程接收缓冲区长度这个问题集中体现在用于接收请求的函数（如recv、recvf rom等）第3个参数Ten上。
如果指定的Ten很小.我们是无法一次性传送功能复杂的很长的shell Code的.这时就必须采用其他传送的方式.例如把shell Code分段传送，这在10期一篇溢出文章（（编写分段传送的shell Code））中有介绍。接收缓冲区长度问题一般在服务进程自身绑定了端口的溢出漏洞中出现，因此PNP服务中可以暂不考虑。
3.异常发生的位置这个问题是指我们向服务程序提交超长的请求后，会覆盖问题函数的返回地址EIP等指针。
当函数返回的时候，跳转到我们覆盖数据指定的地址的时候可能发生异常：另外一种情况就是在该函数返回之前，由于我们覆盖了局部变量中其他指针，导致对其进行引用的时候发生异常。
简而言之，就是rel前与rel后发生异常的区别。
对于ret后的异常，我们可能需要通过KiUse rExceptionDispatcher和监视点的方法来联合定位溢出点.并且漏洞利用的方式有两种改写EIP为jmp esp地址或改写SEH指针而对于ret前的异常.我们一般通过KiUserExceptionDispatcher来定位溢出点就够了，利用的时候也只能改写SEH指针。
我们预先并不知道PNP溢出漏洞属于哪一类.不过等下面进一步调试后就明确了。
4.判断ret函数的调用约定这个问题是最容易被忽略的。
如果我们能够采用jmp esp的利用方式的话，可能很多人会在覆盖jmp esp位置的紧接4个字节开始的地方连接shell Code.问题就出在这里。

编辑推荐

《黑客防线2009缓冲区溢出攻击与防范专辑》深入分析漏洞存在机理及触发机制，完整呈现漏洞发掘过程，详细解析Exploit的编写与核心ShellCode的调试过程。

<<黑客防线2009缓冲区溢出攻击>>

版权说明