第 33 卷　第 24 期　　计　算　机　工　程　　2007 年 12 月
Vol.33　No.24　　Computer Engineering　　December 2007
· 安全技术 ·　　文章编号： 1000—3428(2007)24—0138—03　　文献标识码： A　　中图分类号： TP393.08

Cisco IOS 系统缓冲区溢出攻击研究

凤　丹，邹　敏
（江南计算技术研究所，无锡　214083）

摘　要：路由器安全在网络安全领域占有非常重要的地位，该文针对互联网中使用最为广泛的 Cisco 路由器，介绍 Cisco IOS 的基础特性，从缓冲区溢出的原理出发，阐述一种利用 IOS 缓冲区溢出漏洞远程攻击路由器的方法，提出针对该类攻击的防护措施。

关键词：网络安全；路由器；缓冲区溢出

Study on Cisco IOS Buffer Overflow Attack

FENG Dan, ZOU Min
(Jiangnan Institute of Computing Technology, Wuxi 214083)

【Abstract】Router security is very important in computer networks. This paper focuses on Cisco routers which are the broadest used, describes the basics of IOS which runs in Cisco routers, then based on the principle of buffer overflow, describes a method of buffer overflow attack in Cisco IOS, and presents the defense methods against this kind of attack.

【Key words】network security; router; buffer overflow

路由器是 Internet 网络中进行网间连接的关键设备，是网络间相互连接的枢纽。路由器系统构成了基于 TCP/IP 的 Internet 的主要框架，对路由器技术的研究和应用在整个互联网络技术的研究和应用中始终具有核心地位。因此，路由器的相关安全技术是网络安全的重要课题，路由器攻击方面的研究不管是对于网络的安全渗透性测试或是增强路由器的抗攻击性来说都是非常重要的[1]。Cisco 路由器在互联网中得到广泛使用，本文从 Cisco 路由器上 IOS 的基础和特性以及缓冲区溢出原理出发，阐述了一种利用 IOS 的堆溢出漏洞远程攻击路由器的方法，并提出了针对该类攻击的防护措施。

1 Cisco 路由器 IOS 相关基础

1.1 Cisco IOS 简介

Cisco IOS 是 Cisco 公司专门开发的用于其网络产品上的操作系统，目前 Cisco 公司出品的路由器、交换机、防火墙等网络产品上使用的基本都是 Cisco IOS 操作系统。

Cisco IOS 是一个基于单片机结构的嵌入式操作系统，其上并行运行多个进程，每个进程有且只有一个线程，有自己的内存栈空间、CPU 上下文 (例如寄存器值等)。为了保证高效、迅速地进行报文交换，Cisco 牺牲了一些稳定性和安全性，IOS 在很多方面没有采取其他操作系统的安全保护措施，例如，IOS 进程间没有存储保护机制，虽然每个进程都有各自的存储空间，但是其他的进程同样可以访问该空间[2]。

1.2 Cisco IOS 内存结构剖析

整个 IOS 内存被组织成一个完整连续的虚拟地址空间，通常根据物理内存的不同类型分成区 (region)，区又同时被分成多个子区。IOS 通过一系列存储池 (pool) 来管理可用的空闲存储区域，这些存储池也就是通常意义上的堆。每个池都是存储块 (block) 的集合，这些块根据需要可以被分配也可以被收回。

在存储池中，IOS 为了查询的高效和快速，使用了一种双向链表结构来组织所有的存储块。如图 1 所示。

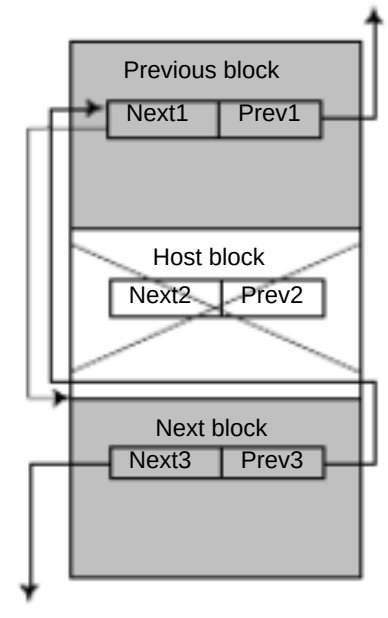

图 1 IOS 内存块双向链表结构

每个内存块除了其中存储的数据之外，还包含一个头部数据结构，维护了一些相关的基本信息，如内存块的大小等。内存块头部结构如图 2 所示。其中有两个指针：Next ptr 和 Prev ptr，这两个指针值是内存块头部的关键数据，分别指向相邻的前一个内存块和后一个内存块，串起了内存池中的所有内存块，组成了一个如图 1 所示的内存块双向链表。

同时，在内存池中，还有另一个双向链表，即空闲块双向链表，它将所有的空闲内存块组织起来。空闲内存块的数据结构如图 3 所示。系统在内存块头部的后面，又加上一个空闲块头部结构。其中也包含了两个指针：Next Free ptr 和 Prev Free ptr，分别指向最近的前一个空闲内存块和后一个空闲内存块，这样将内存中所有空闲内存块串联起来，组成了

基金项目：国家“863”计划基金资助项目“网络安全积极防御关键技术研究”(2003AA146010)
作者简介：凤　丹 (1983－)，女，硕士研究生，主研方向：网络安全，路由器；邹　敏，工程师
收稿日期：2007-01-21
E-mail：wsfd311@gmail.com

空闲块双向链表。

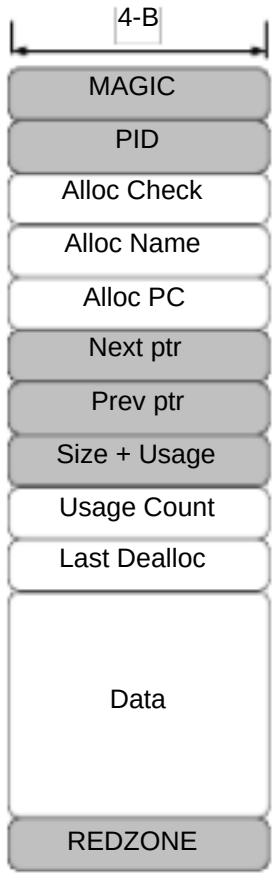

图 2 IOS 内存块数据结构

图 3 IOS 空闲内存块数据结构

1.3 IOS 中的内存保护机制

虽然 IOS 没有采取进程间的存储保护机制，但是为了系统的安全和稳定， IOS 也采取了一些其他机制来保证内存的正确性和完整性。 在 IOS 的进程中， 有一个进程一直在运行，就是“ Check Heaps ”进程，它定期检查 IOS 运行代码和存储堆结构的完整性，若发现异常，则强制系统重启[3]。

Check Heaps 进程每隔一分钟检查一次内存池中的内存块双向链表，检查每个内存块的头部数据结构中的相关字段是否正确，同时检查空闲内存块双向链表，检查空闲快头部结构中的相关字段正确性。被检查的相关字段见图 2 和图 3 中的阴影部分字段。

2 缓冲区溢出原理及利用分析

2.1 缓冲区溢出原理

通常，在路由器上所说的缓冲区溢出是指堆溢出。堆溢出是指当用户输入超出程序中 malloc( ) 函数预先分配的空间大小，而系统没有进行边界检查时，超出的数据部分就会覆盖掉这段空间之后的存储区域。

以下的具体漏洞利用方面说明都以 Cisco IOS 的 TFTP 服务长文件名远程缓冲区溢出漏洞为例， TFTP 服务器在处理长文件名时缺少正确的边界检查，如果请求的文件名超过 700 个字节就有可能导致路由器崩溃而重新启动。这是因为 IOS 的 TFTP 服务在收到请求时会向内存池管理程序请求一个内存块来存储文件名，若是发送一个 get AAAAAA …(700 个)命令，就会出现堆溢出，超出的部分文件名数据覆盖了内存块中的 REDZONE 字段和下一个内存块的头部。所以，系统的 Check Heaps 进程在检查内存双向链表的正确性时就会发现异常，从而强制系统重启。

2.2 堆溢出利用分析

向 Cisco IOS 发送一个超过 700 B 的 TFTP 文件名请求就会导致系统重启，但是为了达到更好的攻击效果，即在溢出之后能够使 IOS 执行特定的 Shellcode，从而控制路由器，要对 Cisco IOS 上的堆溢出利用方式进行研究。

系统在利用 free 函数释放一个内存块之前要检查内存块链表上相邻的内存块是否为空闲块，若为空闲块，则首先要进行空闲块的合并操作，然后再将待释放的内存块释放，将合并生成的新空闲块插入到空闲块链表中。用流程图表示 free( ) 函数的处理过程如图 4 所示。

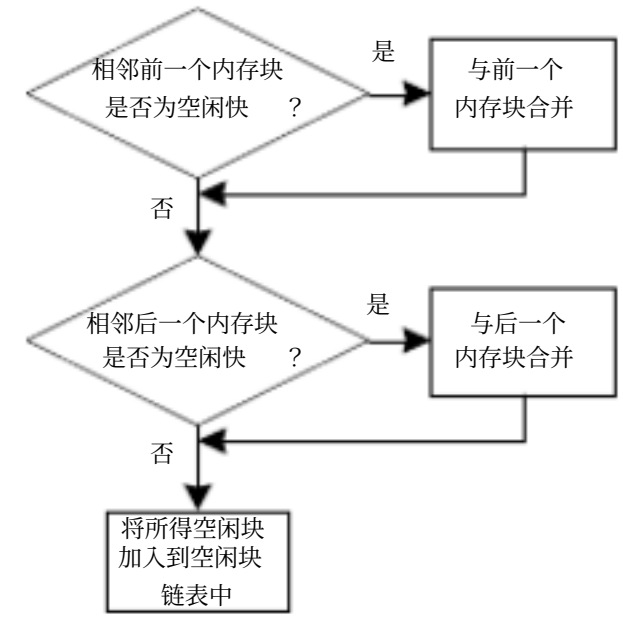

图 4 free( )函数流程

在进行空闲块合并操作时要修改空闲块链表，将前一个或后一个空闲内存块从空闲块双向链表上删除，还要修改前后内存块的相关指针值，假设待释放的内存块相邻的后一个内存块为空闲块， 则合并前后的内存空闲块链表如图 5 所示。

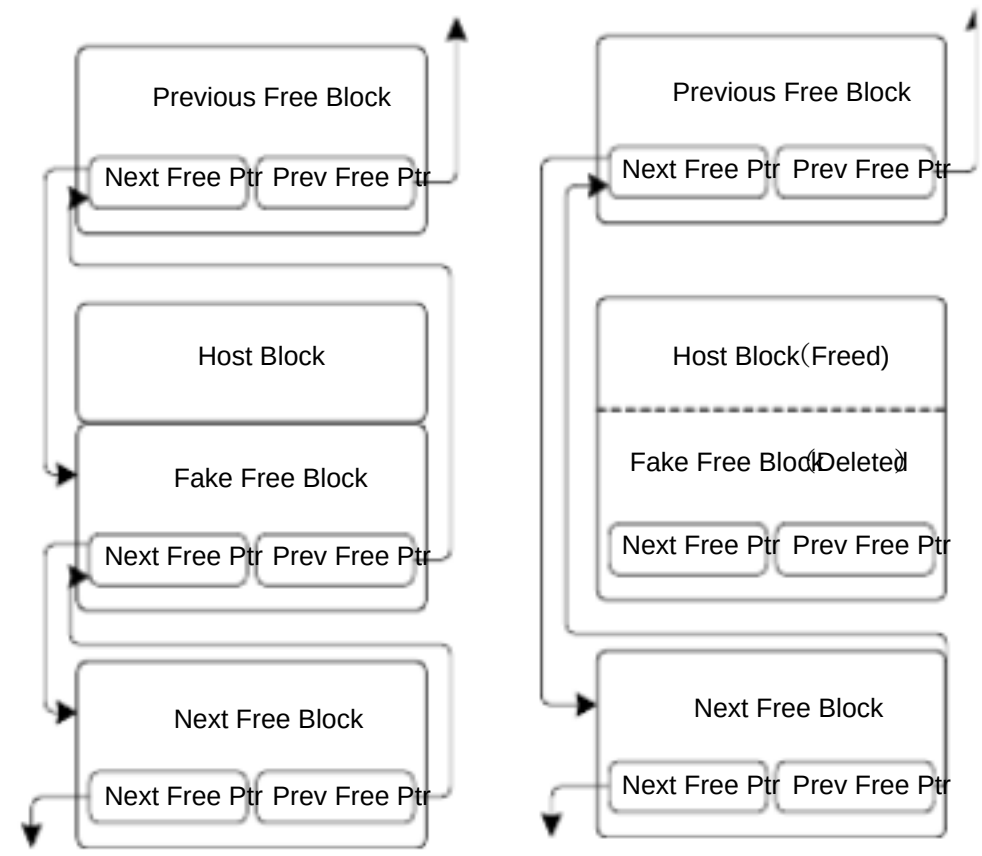

图 5 空闲块合并前后空闲块链表

用程序语言来描述这个过程如下：

*(Fake->Free_Prev) = Fake->Free_Next (1)

(Fake->Free_Next)->Free_Prev = Fake->Free_Prev (2)

可以利用这两条内存的读写操作，完成溢出后的程序流程跳转， 即用 Shellcode 地址覆盖某函数的返回地址， 从而使系统跳转执行它们自己定义的代码，达到控制功能。

具体实现时，采用构造假空闲块的方法，在发送的溢出代码之后构造一个假的空闲块，合理地填写块头部各个字段值以防止 Check Heaps 进程发现异常而使系统崩溃。同样以 TFTP 服务长文件名远程缓冲区溢出漏洞为例，构造发送的 TFTP 请求文件名如图 6 所示。

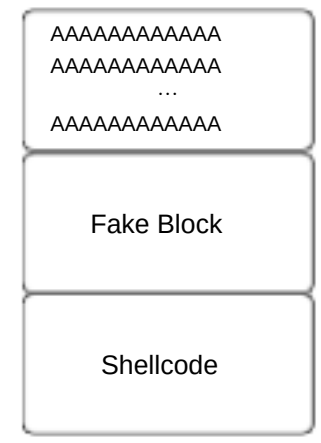

图 6 TFTP 文件名构造

如上文所述，在 Fake Block 中，着重需要注意构造的是 Free_Prev 和 Free_Prev 两个指针的值。Free_Prev 中填写的是一个系统进程返回地址所在内存中的位置，Free_Next 中填写的是 TFTP 文件名中 Shellcode 段所在内存的地址，则上文所述的式 (1)可以表述为

*(ReturnAddress) = &(Shellcode)

这样，在 TFTP 缓冲区内存块与 Fake Block 进行合并之后，就会将某一系统进程的返回地址改写为填写的 Shellcode 的地址，系统在执行完该系统进程之后返回时就会执行到 Shellcode 上，从而达到了控制整个系统流程的效果。

Shellcode 是笔者自己编写的可以被 IOS 执行的一段二进制代码，可以在 Shellcode 中实现很多功能以达到对路由器系统的控制，例如：重写路由器 NVRAM 中存储的配置文件、打开 TTY 控制台等。

该方法在实验室环境下，针对 Cisco2501 系列路由器的 IOS 11.2 版本，可以达到预期的攻击控制效果。

3 防护措施

从路由器自身安全策略上来说，存在该类可远程利用的堆溢出漏洞的 Cisco IOS 系统并没有根本的措施来抵御此类攻击。但是，用户仍然可以采取一些方法来防止此类攻击的发生。

(1)Cisco 公司以及一些安全组织非常重视 IOS 的安全性，不断致力于加强 IOS 系统安全性和稳定性的工作。随着 Cisco IOS 版本的更新，IOS 的安全性也在不断提高。针对各种 Cisco 网络产品的安全漏洞，Cisco 公司都会及时响应并采取措施，比如发布修补漏洞的 IOS 版本等。从用户角度来说，为了保证网络安全，除了要对路由器等网络设备进行完善的安全配置之外，还应该关注相关的安全动态，积极更新 Cisco 设备的 IOS 版本，从而抵御该类网络攻击。

(2) 由于该类攻击方法有一个明显的特征：在发送的数据包中必须包含一个假的 IOS 内存块头部数据结构。利用该特征，可以在路由器的前端放置一个包过滤防火墙，对将要发送到路由器上的数据包进行一次过滤，若是发现包含如 MAGIC, REDZONE 等显著特征数据的数据包将视为恶意攻击数据包，不允许其发送到路由器上，这种方法将在一定程度上防御该类攻击，保护路由器等网络设备的安全。

4 结束语

在 Cisco 发布的漏洞公告中，类似 TFTP 服务长文件名远程缓冲区溢出漏洞这样的堆溢出漏洞还有 OSPF 报文远程缓冲区溢出漏洞、系统定时器堆溢出漏洞等。在针对 Cisco 路由器 IOS 的攻击类型中，针对缓冲区溢出漏洞攻击是最有可能达到控制路由器效果的一类攻击。对这类攻击方式和原理的研究对于路由器安全来说具有非常重要的意义。

本文从 Cisco 路由器 IOS 一些基本特性出发，阐述了在 Cisco IOS 上进行远程缓冲区溢出的原理以及利用方法，并简要介绍了针对该类攻击的防范措施。目前国内对路由器操作系统缓冲区溢出领域的研究非常有限，更好地保证路由器的安全、建立更安全稳定的网络环境，还任重而道远。

参考文献

[1] Bollapragada V, Murphy C, White R. Cisco IOS 精髓 [M]. 北京：中国电力出版社, 2001.

[2] 张宏科，张思东，苏 伟. 路由器原理与技术 [M]. 北京：国防工业出版社, 2005.

[3] Vladimirov A A, Gavrilenko K V, Vizulis J N. Hacking Exposed Cisco Networks: Cisco Security Secrets & Solutions[M]. [S. l.]: Graw Hill, 2006.

[4] Anonymous. Once upon a free()[J]. Phrack Magazine, 2001, 11(57).

~~~~~~

(上接第 137 页)

表 1 不同窗口大小下的估计结果

分析窗口	含密数据包比率 /(%)										
	0	10	20	30	40	50	60	70	80	90	100
50	9.5	16.2	20.0	30.2	42.0	53.3	58.9	68.3	80.1	90.4	99.7
80	8.1	11.4	18.2	31.9	37.9	48.3	57.3	69.1	78.2	90.7	100
100	11.0	16.3	21.7	30.4	41.5	51.7	61.3	73.5	83.3	91.4	98.7
200	8.9	16.5	23.8	31.0	42.2	51.1	59.3	70.2	79.8	89.4	100
1 000	2.5	10.7	20.6	29.6	38.1	47.9	58.0	67.7	76.7	86.3	95.7

通过实验，发现嵌入率在 [0%,10%] 之间时，由于 $K_2 = 18n$ 不成立，因此估计效果不是很理想。但是对于较大的含密数据包比率，2.2 节给出的方法能获得比较精确的结果。

文献 [5] 提出的 IPIDs 隐写方案在统计上不安全的根本原因在于，其明文加密加密算法不能保证密文的直方图分布与正常数据流 IPIDs 的分布的相似性。为增强算法的安全性，可以考虑直接利用密钥控制产生一串 0，1 随机数，将随机数与秘密信息异或，得到均匀分布的密文，然后将密文代替数据流中的 IPIDs 高 8 位。这种方法不论是在算法复杂度还是抗统计攻击的安全性角度都较原方案为优。

4 结束语

针对文献 [5] 提出的一种基于 IP 标识位的协议隐写算法，本文从统计分析的角度给出了能准确检测出隐写存在的隐写检测方法和能估计出分析窗口内含密数据包比率的方法。由于文献 [5] 的方法采用了较为简单的数据加密策略，因此带来了明显的不安全性，文章最后提出了一个可能的改进的方案。由于网络数据流不同于多媒体信息所具有的较为明显的统计特性，针对网络协议隐写的分析相比更加困难，也更具有挑战性。因此，在进一步的研究中将考虑针对改进方案的隐写分析和针对更为一般的数据包协议隐写算法的分析方法。

参考文献

[1] Handel T, M Sandford. Hiding Data in the OSI Network Model[C]// Proceedings of the 1st International Workshop on Information Hiding. Cambridge, U.K: [s. n.], 1996.

[2] Giffin J, Greenstadt R, Litwack P, et. al. Covert Messaging Through TCP Timestamps[C]//Proceedings of 2nd International Workshop on Privacy Enhancing Technologies. San Francisco, CA, USA: [s. n.], 2003.

[3] Cauich E, Gardenas R G, Watanabe R. Data Hiding in Identification and Offset IP Fields[C]//Proceedings of 5th International Symposium on Advanced Distributed System, Guadalajara, Mexico: [s.n.], 2005.

[4] 唐作吟，杨宗凯，谭运猛，等. 互联网协议中信息隐藏技术的研究 [J]. 计算机应用研究, 2003, 20(8): 14-16, 24.

[5] Ahsan K, Kundur D. Practical Data Hiding in TCP/IP[C]// Proceedings of ACM Workshop on Multimedia Security. Juan-les-Pins, France: [s. n.], 2002.
~~~~~~