一种有效的 Cisco IOS 映像注入攻击分析方法*

曾 诚, 刘胜利, 肖 达, 陈立根
(解放军信息工程大学 数学工程与先进计算国家重点实验室, 郑州 450000)

摘 要: 针对 Cisco IOS 映像注入攻击提出了一种基于虚拟化的恶意代码分析方法。通过虚拟化技术的研究, 设计实现了虚拟化分析平台 CDAP(Cisco dynamic analysis platform),为 IOS 系统提供了运行环境,在此基础上, 采用指令信息截获与过滤技术、数据跟踪技术对注入到 IOS 映像中的恶意代码进行分析。使用该方法可对遭受 IOS 注入攻击的多型号多版本的 Cisco IOS 映像进行分析。实验结果证明了该方法的有效性。

关键词: Cisco IOS; 虚拟化技术; 代码分析; 指令截获; 指令过滤; 数据跟踪
中图分类号: TP311; TP393.08 **文献标志码**: A **文章编号**: 1001-3695(2013)12-3775-04
doi:10.3969/j.issn.1001-3695.2013.12.066

Effective method for analysis of Cisco IOS image injection attack

ZENG Cheng, LIU Sheng-li, XIAO Da, CHEN Li-gen
(*State Key Laboratory of Mathematical Engineering & Advanced Computing*, *PLA Information Engineering University*, *Zhengzhou* 450000, *China*)

Abstract: This paper proposed a method for analysis of malicious code to resist the injection attack of Cisco IOS image based on the virtualization. After the research of visualization, it designed and realized a platform called CDAP to run the IOS. On the basis of CDAP, it analyzed the malicious code injected in the IOS image, with data tracking and instruction interception and filtration technique. Using this method, it analyzed multiple models and versions of IOS images that suffered injection attack. The result of experiment demonstrates this method is effective.

Key words: Cisco IOS; virtualization; code analysis; instruction interception; instruction filtration; data track

随着人们对信息需求的不断扩大,计算机网络和通信技术得到快速的发展。为了更便捷、更安全地获取和输送信息,网络通信技术不断被创造和更新,但是无论使用何种网络技术手段进行信息的交流,都离不开网络基础设施的支持。路由器是互联网络中重要的基础设施,是网络互连的基础,其安全直接影响到整个网络的安全。作为全球互联网中部署最为广泛,技术和市场占有率占绝对优势的 Cisco 路由器,其安全性直接影响到全球网络的安全防护能力。由于 Cisco 公司生产的路由器和交换机都使用其自主开发的操作系统 Cisco IOS(internetwork operating system),且其代码为非公开的,同时基于 Cisco 路由器所使用的处理器芯片的类型,Cisco IOS 一般采用 MIPS 或者 Power PC 构架[1],与常见的 Intel 构架相去甚远,可借鉴的资料较少,这些使得对 Cisco IOS 的检测和分析变得更加困难。近年来一些网路安全研究者和研究机构通过对 Cisco IOS 的安全机制的长期研究,相继在 Cisco IOS 安全脆弱性方面有一定的发现,并依此提出了相关攻击方法[2,3]。其中就有一类针对 Cisco IOS 映像文件完整性保护机制的脆弱性而发起的 IOS 静态注入攻击[4,5],这对 Cisco 路由器安全有极大的影响。虽然 Cisco 公司随后为大部分在用的 IOS 映像提供了一个 MD5 的校验码,用于 IOS 映像的完整性校验[6],以防御这类攻击方法,但这种检验是需要管理员主动进行,并且在路由器运行的时候不能进行,使得这种安全防护方法并不能有效地保护 Cisco 路由器的安全。而现有可以借鉴且用于分析这类攻击的方法也比较有限,一般有 IDA pro[2]、gdb 调试技术以及 Muñiz 等人[7]提出的一种基于虚拟化的 Fuzzing 方法。IDA pro 和 gdb 调试技术能对一些型号的路由器系统结构和部分功能函数进行静态分析,但对经过处理的映像代码进行分析时比较困难,也很难对指定的进程进行跟踪调试。Muñiz 等人的基于虚拟化的 Fuzzing 方法能对路由器发生异常时的内存与寄存器数据进行记录,并对 MIPS64 与 PowerPC 下的指令进行单步调试,但它的分析过程依然依赖于 IDA pro 的分析结果,这使得分析的范围被限制,同时该方法支持的路由器型号也有限。现有的 Cisco IOS 分析技术都不能很好地对 Cisco IOS 静态注入攻击进行分析。

本文在研究虚拟化技术[8]的基础上,设计、实现一个动态分析平台 CDAP,并在该平台的基础上,研究出一种针对 Cisco IOS 静态注入攻击的分析方法。在得到注入代码于 IOS 映像中的位置后,利用分析平台 CDAP,通过指令流截获和分析技术、数据的污点传播技术[9],提取出指定进程的指令流和数据流,为注入代码的分析和 Cisco 路由器的安全维护提供支持。

1 动态分析平台的设计与实现

1.1 动态分析平台的设计

在路由器虚拟化的基础上,设计实现动态分析平台 CDAP。该分析平台能够对 IOS 执行流程进行监控,对寄存器和内存数据进行查看与修改,并对虚拟机系统外部输入/输出数据以及系统运行过程中的内存数据变化进行监视,同时依据对指令流控制规则的设置,对 IOS 指定执行过程的指令流进行

收稿日期: 2013-03-12; **修回日期**: 2013-05-09 **基金项目**: 郑州市科技创新团队项目(10CXTD150)

作者简介:曾诚(1986-),男,湖北咸宁人,硕士研究生,主要研究方向为信息安全(chengceng777@126.com);刘胜利(1973-),男,副教授,主要研究方向为信息安全;肖达(1981-),男,助教,主要研究方向为信息安全;陈立根(1988-),男,硕士研究生,主要研究方向为信息安全.

截获与处理。CDAP 以 Cisco 路由器虚拟机作为功能执行部件,通过 Hypervisor 与用户交互接口通信,执行用户动态分析指令。其总体设计如图 1 所示。

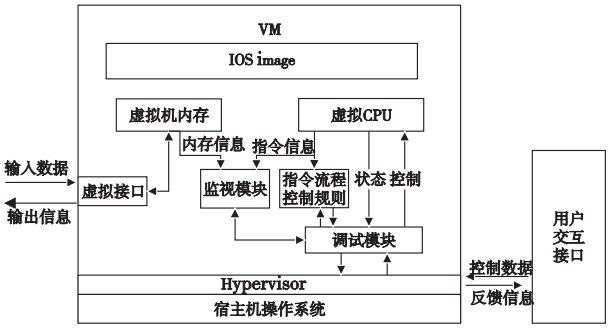

图1 CDAP平台构架

在这个模型中,通过 Hypervisor(虚拟机管理程序)模块与用户进行通信,使得用户能够对虚拟路由器的运行状态和运行信息进行跟踪和控制。虚拟机中调试模块与路由器主 CPU 并行运行,调试模块通过 Hypervisor 和监视模块的反馈信息,对 CPU 的运行进行控制,对内存数据和指令信息进行操作。将注入代码定位结果提交到监视模块中,监视模块负责对内存数据和当前运行指令信息的监视,当某个内存块被访问、操作,某条指令被调用时,监控模块就通知调试模块,进而对相关指令信息和内存数据进行截获、分析。在对注入代码的分析中,需要根据注入代码定位结果对监视模块进行相关数据的设置。指令流程控制规则主要在分析注入代码污染的 IOS 功能函数及运行数据的过程中用到,当注入攻击代码在运行后,对路由器内存或寄存器内容进行了一系列的操作,但是攻击代码本身却没有使用到这些被污染的数据,或是注入代码调用了 IOS 原有的一些功能函数,用于帮助实现其攻击目的,那么在对注入攻击代码进行分析后,通过对指令流程控制规则的设置,控制虚拟 CPU 中运行的指令流程,达到对污染函数和数据的分析,为分析注入攻击代码的功能提供帮助。

1.2 动态分析平台的实现

1.2.1 虚拟机模块化管理与调试模块

虚拟机设计中,虚拟设备采用先注册、后使用的机制。注册就是将虚拟硬部件设备管理结构挂接到虚拟机实例的设备管理链表中。不同的虚拟设备有不同的注册方法,提供不同的链表管理结构对设备进行管理。一般虚拟硬件使用设备管理结构 vdevice 来进行管理。该结构中,定义了设备名、设备在虚拟机内存空间中起始地址、地址范围和该设备映射至宿主机虚拟机进程的地址,提供了该设备的访问函数以及其他访问属性。在设备初始化时,使用绑定函数 vm_bind_device( )将设备绑定到虚拟机实例。虚拟机实例的管理可以分为本地模式与 Hypervisor 管理模式。本地模式是在虚拟机程序运行时,将虚拟机所有的配置写入启动命令,虚拟机进程启动后无法再进行重新配置;而 Hypervisor 模式则是在虚拟机进程启动时,加载必要的虚拟设备并进行设备初始化,在启动后,可以通过远程登录进行系统重新配置,如指定需要加载的 IOS 映像的路径、创建新的虚拟网卡、将其绑定到指定宿主机物理网卡等。可以在 Hypervisor 模式状态下使用用户命令进行虚拟硬件重构,同时还可以进行系统状态实时监控。

在 Hypervisor 模式下,设计加载调试模块,将其注册到 Hypervisor 模式的虚拟机实例硬件管理结构中,同时,注册调试模块相应的命令与控制函数。用户可以通过使用包含模块名、功能名和参数的命令来操作模块实现相应动态分析功能。调试模块主要使用控制命令结构数组将用户命令与命令解析执行函数对应起来。

通过定义 vm_cmd_array 数组来建立用户命令与解析执行函数的一一映射关系。在调试模块初始化时,将模块管理结构注册到 Hypervisor 管理结构中,并为该模块注册 vm_cmd_array 数组,当用户输入特定命令,根据命令名从管理结构中查找到该命令的解析执行函数地址,以命令参数字符串作为执行函数实参,调用执行函数执行用户命令。

1.2.2 虚拟 CPU 控制机制

虚拟机设计中,虚拟 CPU 的功能模拟主要使用虚拟 CPU 线程来实现。该线程启动后,每次从程序计数器指向的内存中读入一条指令,解析后进行模拟执行,将执行结果反馈到虚拟机设备中。为了使虚拟机支持动态分析,需要控制 CPU 的运行状态,当系统运行状态满足用户设置的条件时,暂停 CPU 的运行。

在虚拟 CPU 的结构定义中,一般定义三种 CPU 执行状态,即

```
CPU_STATE_RUNNING = 0   //运行状态
CPU_STATE_HALTED        //停止状态
CPU_STATE_SUSPENDED     //暂停状态
```

1.2.3 用户交互接口

在 CDAP 的设计实现中,还需要设计外部用户交互接口,用于接收用户输入数据,传输至虚拟机以监控 IOS 执行流程,接收虚拟机运行反馈状态信息。该部分的主要功能是与虚拟机 Hypervisor 模块进行通信,读取不同型号 Cisco 虚拟机的配置文件,发送至 Hypervisor,对系统进行虚拟硬件重构,启动虚拟机实例;其后,发送用户命令,接收用户反馈信息。其工作流程如图 2 所示。

CDAP 设计完成后,可以满足基本的进程动态跟踪分析的需求,用户可通过用户交互接口控制 IOS 的执行流程,分析 IOS 的运行状态,虚拟机自身还为用户提供多种管理接口,如 telnet、console 管理等。

2 注入代码动态分析技术

在 Cisco 路由器中,IOS 的进程与其他操作系统中的单线程基本相同,同时每一个 IOS 进程有且只有一个线程。IOS 采用带优先级的运行直至结束策略调度的可执行进程,一个进程可以被内核或其他运行的进程创建或终止。那么一个进程在运行的过程中可能会被中断打断,也可能会挂起等待某个资源,也可能主动创建某个进程实现某个功能。从 CPU 执行的指令流序列来看,可以得到图 3 这样一个指令流序列,其中进程 a 为指定进程。

所以要想得到注入代码实际的执行指令流,就需要对 CPU 中的指令流进行过滤分离,得到注入代码自身的指令流序列、区分开其所调用的功能函数的指令流序列。

2.1 信息截获与过滤

Cisco IOS 采用 MIPS 或者 Power PC 构架。表 1 为 ppc32 的指令集类型。

从表 1 中可以看出,要过滤出注入代码指令流序列,就需要以转移指令和系统调用/返回指令为依据对 CPU 指令流进行分割;要过滤出注入代码的数据控制流,就只需要对注入代码指令流执行过程中的数据控制流进行记录。注入代码指令流截获与过滤算法逻辑描述如图 4 所示。

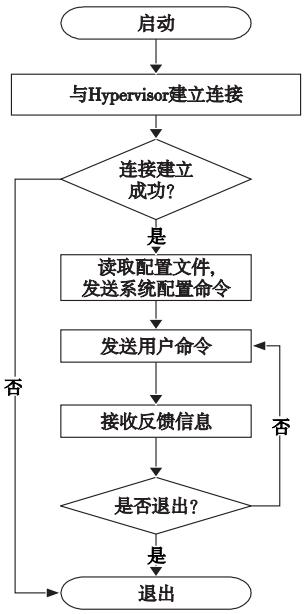

图2　用户交互接口工作流程

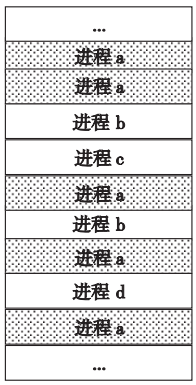

图3　CPU执行的指令流序列

表 1　ppc32 指令类型

指令功能	指令类型	指令功能	指令类型
整数存储	stbx、sthx、stw、stmw 等	数据运算	add、xor、cmp 等
整数加载	lbx、lhz、lwz、lmw 等	系统调用与返回	sc、rfi 等
特殊寄存器传送指令	mfx、mtx 等	转移指令	b、bl、bcx 等

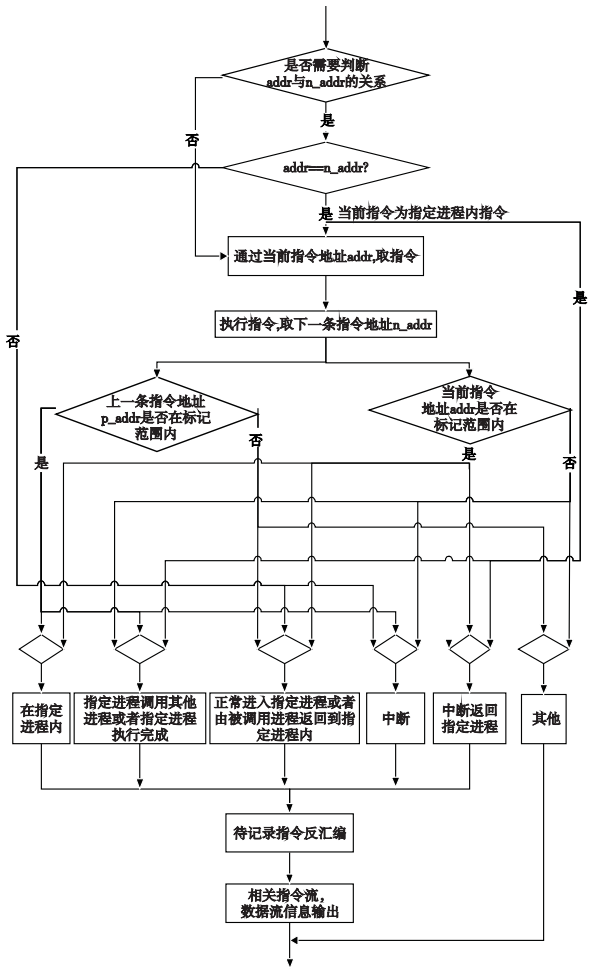

图4　指令流截获与过滤算法

依据注入代码在 IOS 映像中定位的结果,换算出它在 IOS 运行时的虚拟内存地址空间,并进行标记。当 CPU 执行的指令在被标记的范围时,即认为注入代码被执行,对 CPU 指令流进行记录。通过对当前即将执行指令 addr 和上一条指令 p_addr 是否在标记的内存范围内,以及当前即将执行指令 addr 是否与上一条指令 p_addr 执行完成后程序计数器的值 n_addr 相同的情况,确定当前执行的指令归属,进而对指定进程的执行指令流进行连续截获,屏蔽系统中断的影响,并在发生系统中断、中断返回、调用函数、调用返回等情况时给出提示,并记录对应入口地址。

2.2　污点数据跟踪

通过对表 1 中指令的指令格式进行分析,可以知道通过对数据操作指令的跟踪,得到数据的扩散特点。在 2.1 节中可以得到注入代码指令流序列,在该指令流序列执行的同时,通过对数据操作指令的过滤及指令内容的解析,可以建立一个数据扩散链表(图 5)来显示所跟踪代码动态执行过程中的数据流特点。

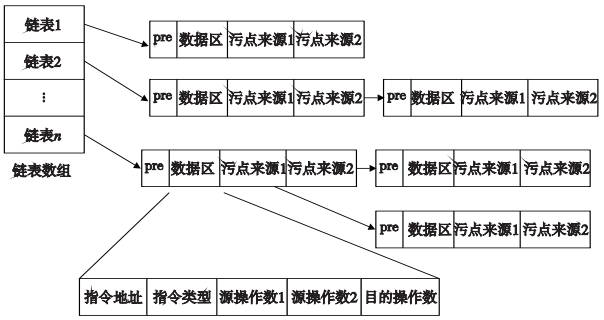

图5　数据链表

在单个链表节点中,pre 为本节点的上一个节点的地址指针,指令地址为要记录的数据操作指令地址,源操作数与目的操作数为数据的寄存器号、内存地址或立即数,污点来源为源操作数中标示为污点的数据产生的所在指令地址。例如在记录的数据信息中有下面两条按顺序记录的指令:

0x80a7114c　add　rD,rA,rB

0x80a71150　add　rE,rD,rC

其中:rA 已标示为污点数据地址,当执行 0x80a7114c 指令后,通过 add 操作;rD 为指令执行后数据的存放位置,故 rD 被标记为污点数据地址。在执行 0x80a71150 这条指令时,由于其源操作数 rD 被标记,该指令产生的链表节点中,其污点来源的值为 0x80a7114c。当一个新的数据操作指令被过滤下来时,对指令内容进行解析,得到源操作数地址、目的操作数地址等信息,构建链表节点,填充数据区内容,依源操作数内容在已建立的数据链表中搜索其污点来源,而后加入到链表中。通过数据扩散链表的建立,可以很清晰地展示被跟踪代码执行过程中数据的传播方式和特点,为代码分析提供有力支持。

在过滤和解析指令流、建立数据扩散链表的同时,也对被污染的 IOS 内存空间及特殊寄存器范围进行统计和监视。在对污染数据范围统计的基础上,对污染区进行监视,监测该区域内容是否被 CPU 执行,并将被执行的污染区域进行标记统计,并按分析者的实际需要,确定是否加入到代码跟踪范围内。通过对被污染内存空间及特殊寄存器范围的统计,能很好地反映所跟踪代码带来的影响;通过对污染区是否被执行情况进行检测,能很好地识别并提取出所跟踪代码中的隐蔽代码[10]。

注入代码的动态分析,主要是在 CDAP 平台的基础上,利用对虚拟硬件的控制,截获 CPU、寄存器等设备在运行过程中的信息。通过对指令内容的解析和过滤条件的设置,过滤出所需的信息。

3 实验测试

本章将对动态分析平台 CDAP 进行功能测试,验证平台加载多版本 Cisco IOS 的有效性,利用 CDAP 对遭受静态注入攻击的 Cisco 2600 IOS 样本进行注入代码动态分析。该样本在 Cisco IOS c2600-I-mz.122-8T10.bin 映像的基础上修改得到。该注入代码不影响系统的正常运行,代码功能为检测用户输入是否含有字符“a”,含“a”则跳转入 check heaps( )函数,不含则无操作。

3.1 测试环境

测试环境搭建拓扑如图 6 所示。

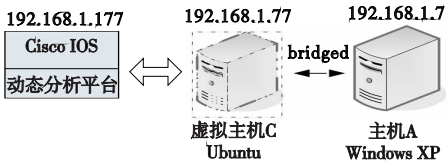

图6 CDAP平台测试环境

在主机 A 中运行 VMware 虚拟机软件,用于构建虚拟主机 C(Ubuntu 9.04 系统),在虚拟主机 C 上运行 CDAP 平台。

3.2 测试内容

3.2.1 平台加载多平台多版本 IOS 测试

运行构建好的 CDAP 平台,启动路由器虚拟平台,加载各项调试模块,在用户调试模式下,设置 Cisco 路由器参数,加载 IOS,使 IOS 正常运行于虚拟平台上,在用户能通过调试接口对 IOS 进行调试、分析的同时,也能通过路由器自身的各项接口对路由器进行管理和向路由器传输数据。图 7 所示为 CDAP 加载 Cisco 2600 IOS 映像过程、用户交互接口连接虚拟路由器及通过 console 口连接路由器。

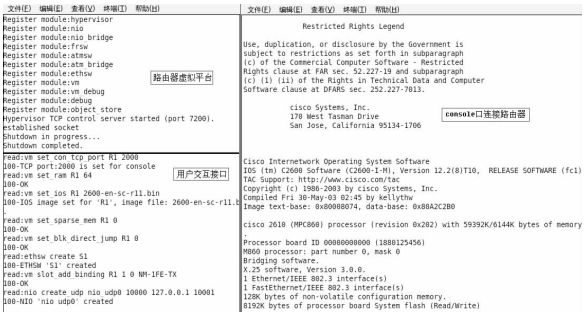

图7 CDAP平台的加载与运行

通过对多版本的 Cisco IOS 的测试,测试 CDAP 平台加载 IOS 映像,用户通过交互接口连接调试路由器、用户通过路由器 console 接口连接管理路由器。测试结果如表 2 所示。

表2 平台加载等功能测试表

硬件型号 & 处理器平台	IOS 主版本	IOS 版本数量	加载功能	console 连接
Cisco 1700 PowerPC	12.0/12.1/12.2/12.3/12.4	15	成功	成功
Cisco2600 PowerPC	11.3/12.0/12.1/12.2/12.3/12.4	18	成功	成功
Cisco 2691 MIPS64	12.2/12.3/12.4	10	成功	成功
Cisco 3725 MIPS64	12.3/12.4	9	成功	成功
Cisco 3745 MIPS64	12.3/12.4	7	成功	成功
Cisco 7200 MIPS64	12.0/12.1/12.2/12.3/12.4	15	成功	成功

3.2.2 CDAP 平台对注入代码分析

在定位注入代码的范围后,通过该结果,配置 CDAP 平台的监控参数。在触发注入代码后,对注入代码的执行指令流进行有序的记录,并对相关数据污染范围及污染数据是否可执行进行监控,据此进行后续分析工作。样本中注入代码为对用户输入指令进行识别,检测用户输入是否含有字符“a”。如图 8 所示,代码分析过程中,用户输入为除“a”以外的字符序列。调试记录信息显示:注入代码某一执行路径被截获及过滤出来,其指令流序列被实时显示及同步记录,其数据扩散信息被统计。统计信息表明,该部分代码未对内存进行修改,不含隐蔽代码。

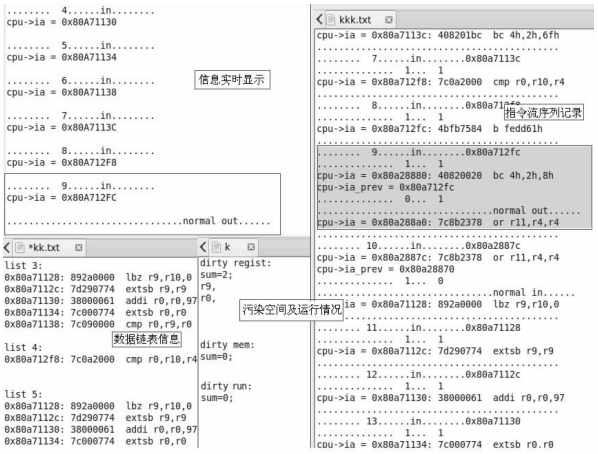

图8 信息截获与过滤

4 结束语

在虚拟化技术的基础上实现了 Cisco 路由器的虚拟化,并在添加了用户交互接口后,能够对路由器的硬件设备进行监控,从而能够对虚拟 CPU 执行指令的截获和过滤,进而得到指定进程的指令流序列,并得到相应的数据信息,为对注入代码的分析提供帮助。下一步将对如何提高代码分析的覆盖率及代码分析自动化程度进行研究,进一步提高代码分析的效率,为安全防护提供支持。

参考文献:

[1] FUTORANSKY A. Viral infections in Cisco IOS [EB/OL]. (2008-08-06). https://media.blackhat.com/bh-usa-08/video/bh-us-08-Futoransky/black-hat-usa-08-futoransky-viralCisco-hires.m4v.

[2] LYNN M. The holy grail: Cisco IOS shellcode and exploitation techniques [EB/OL]. (2010-11-21). http://securityvulns.com/Fnews57.html.

[3] CHAWDHARY G, VARUN U. Cisco shellcodes[EB/OL]. (2008-08-06). https://www.blackhat.com/presentations/bh-usa-08/Chawdhary_Uppal/BH_US_08_Chawdhary_Uppal_Cisco_IOS_Shellcodes.pdf.

[4] 风丹,邹梅. Cisco IOS 系统缓冲区溢出攻击研究[J]. 计算机工程,2007,33(24):138-140.

[5] FELIX L. Cisco IOS router exploitation[EB/OL]. (2009-06-22). http://www.blackhat.com/presentations/bh-usa-09/LINDNER/BHUSA09-Lindner-RouterExploit-PAPER.pdf.

[6] Rootkits on Cisco IOS devices[EB/OL]. (2011-01). http://www.cisco.com/warp/public/707/cisco-sr-20080516-rootkits.shtml.

[7] MUÑIZ S, ORTEGA A. Fuzzing and debugging Cisco IOS [EB/OL]. (2011-12-21). http://www.pdfpedia.com/download/13758/fuzzing-and-debugging-cisco-ios-blackhat-europe-2011-pdf.html.

[8] 怀进鹏,李沁,胡春明. 基于虚拟机的虚拟计算环境研究与设计[J]. 软件学报,2007,18(8):2016-2026.

[9] 陈恺,冯登国,苏璞睿. 基于有限约束满足问题的溢出漏洞动态检测方法[J]. 计算机学报,2012,35(5):898-909.

[10] 王祥根,司端峰,冯登国,等. 一种基于自修改代码技术的软件保护方法[J]. 中国科学院研究生院学报,2009,26(5):688-694.