第36卷 第21期 计 算 机 工 程 2010年11月
Vol.36 No.21 Computer Engineering November 2010
·博士论文· 文章编号：1000—3428(2010)21—0017—02 文献标识码：A 中图分类号：TP393

BIOS恶意代码实现及其检测系统设计

王晓箴[1,2]，刘宝旭[1]，潘 林[1,2]
(1. 中国科学院高能物理研究所计算中心，北京 100049；2. 中国科学院研究生院，北京 100049)

摘 要：根据基本输入输出系统(BIOS)恶意代码的植入方式，将其分为工业标准体系结构、高级配置和电源管理接口、外部设备互连模块恶意代码3类，分别对其实现过程进行研究。在此基础上，设计一种BIOS恶意代码检测系统，包括采样、模块分解、解压缩、恶意代码分析模块。应用结果表明，该系统能检测出BIOS镜像文件中植入的恶意代码，可有效增强BIOS的安全性。

关键词：基本输入输出系统；恶意代码；安全检测

BIOS Malicious Code Implementation and Its Detection System Design

WANG Xiao-zhen[1,2], LIU Bao-xu[1], PAN Lin[1,2]
(1. Computing Center, Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

【Abstract】 Based on the implantation method of Basic Input Output System(BIOS) malicious code, this paper divides the malicious code into Industry Standard Architecture(ISA), Advanced Configuration and Power management Interface(ACPI) and Peripheral Component Interconnect (PCI) module malicious code, and analyzes the implementation processes of three types of BIOS malicious code. It designs a BIOS malicious code detection system which includes the modules of sampling, module disassembling, decompressing and malicious code analyzing. Application results show that this system can detect the malicious code in BIOS image file, and it can effectively enhance the security of BIOS.

【Key words】 Basic Input Output System(BIOS); malicious code; security detection

1 概述

21世纪以来，伴随着互联网技术的高速发展，计算机应用已深入到社会生活的各个角落，使人们的生活工作更便捷，但随之而来的安全问题却越来越严重。木马、蠕虫、DDoS攻击、网络陷阱等各种威胁潜伏于互联网上，侵害用户权益。其中，基本输入输出系统(Basic Input/Output System, BIOS)恶意代码作为罕见但具毁灭性的一类近年来出现频繁，它是一组固化到计算机内主板的ROM芯片上的程序，保存计算机最重要的基本输入输出的程序、系统设置信息、开机上电自检程序和系统启动自举程序。计算机开机执行的第1条指令以及第1组程序均源自主板BIOS。因此，BIOS层的安全是很重要的，一旦BIOS被攻击破坏，会对整个计算机的可用性造成影响，如1998年的CIH病毒就是通过攻击计算机BIOS，删除其中信息使机器找不到系统盘从而导致机器崩溃，在世界范围内造成严重危害。本文主要工作是提出BIOS恶意代码的实现方式，并设计一种BIOS恶意代码检测系统。

2 BIOS恶意代码的实现方式

目前研究认为，BIOS安全隐患可分为BIOS功能障碍、BIOS设置漏洞、BIOS物理攻击、BIOS木马4类[1]。本文认为“BIOS木马”的表述方法不尽准确，因此以“BIOS恶意代码”来代表这类程序[2]。其实现方式是：攻击者利用BIOS代码只占用Flash芯片上部分空间这个特点，将恶意代码封装成合法的BIOS扩展模块，通过BIOS读写工具或驱动自动加载过程写入到BIOS芯片内。恶意代码在BIOS内成功加载后，可与远程主机进行通信，进行窃取、删除数据等各种破坏性的操作。

本文根据BIOS恶意代码的植入方式，将BIOS恶意代码分为工业标准体系结构(Industry Standard Architecture, ISA)模块恶意代码、高级配置和电源管理接口(Advanced Configuration and Power management Interface, ACPI)模块恶意代码、外部设备互连(Peripheral Component Interconnect, PCI)模块恶意代码3类，下文分别对其实现方式及特点进行阐述。

2.1 ISA模块恶意代码

ISA模块BIOS恶意代码是目前主流的实现方式，著名黑客Kaspersky K曾写过一个用于嵌入BIOS的ISA模块程序，IceLord的BIOS Rootkit实现也利用该技术。IceLord的核心代码是ISA模块文件leaving.bin，IceLord.exe在驱动文件的协助下，把ISA模块leaving.bin刷写进BIOS中，将恶意代码植入。

ISA模块恶意代码主要应用于早期BIOS为Award类型的计算机，近年来生产的计算机BIOS已禁止加载ISA模块的功能，因此该类恶意代码目前只能作为实验调试方法使用。

2.2 ACPI模块恶意代码

ACPI定义了允许操作系统控制电源管理和设备配置的机制，是BIOS现行的工作标准。由于ACPI具有规范的驱动

基金项目：国家科技支撑计划基金资助重点项目(2009BAH52B06)；北京市自然科学基金资助面上项目(4072010)
作者简介：王晓箴(1985－)，女，博士研究生，主研方向：网络安全；刘宝旭，副研究员；潘 林，博士研究生
收稿日期：2010-04-20 **E-mail：**wangxz@ihep.ac.cn

体系(如图 1 所示),并且可以轻松访问物理设备以及系统 I/O,因此便于攻击者利用其实现 BIOS 恶意代码,文献[3]于 2006 年做过这方面的阐述。

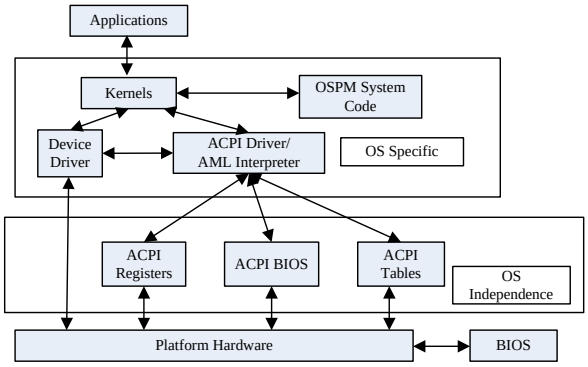

图 1　ACPI 体系结构

ACPI 体系中的核心层用于提供基本的 ACPI 服务,使用 ASL(ACPI Source Language) Code 进行编程。其中,Operations Region 函数用以定义程序访问硬件的接口,其用法如下:

OperationRegion(Name, Space, Offset, Length)

对于 Windows 操作系统,攻击者可先利用 SeAccesscheck 函数提升权限,再利用 OperationsRegion 改写 ACPI 加入恶意代码,最后利用 acpi.sys 此驱动进行解析。对于 Linux 操作系统,攻击者可通过使用 OperationsRegion,篡改这个未经使用的系统调用句柄 sys_ni_syscall(),再在 BIOS 内植入恶意代码。

2.3　PCI 模块恶意代码

2.2 节与 2.3 节中提到的 ISA 及 ACPI 恶意代码,均可以通过 CMOS 中设置禁止加载模块的方式规避,而通过 PCI 设备携带的扩展 ROM 芯片向 BIOS 中植入恶意代码,目前仅有携带可信赖平台模块(Trusted Platform Module, TPM)芯片的计算机能够在启动时检测出异常,因此,这种恶意代码具有广阔前景。计算机中可支持的 PCI 设备种类繁多,本节将提出一种利用预启动执行环境(Pre-boot Execution Environment, PXE)引导芯片的实现的 BIOS 恶意代码加载方式。

PXE 无需硬盘,利用固件(如 ROM 芯片)启动计算机的方法,支持网络启动。恶意代码无论实现什么功能,与远程主机通信是必不可少的部分,支持 PXE ROM 的 PCI 网卡或 BIOS 可以使计算机无需进入操作系统便可接入网络进行通信。对于一台带有 PXE ROM 芯片的计算机,网络启动功能可作为初始程序导入(Initial Program Load, IPL)。文献[4]中详细定义了将设备识别为 IPL 并加入 IPL Table,所需的 ROM 芯片的编程格式。

攻击者按照规范要求的格式对 ROM 芯片编程,编写一个针对中断 19h 的挂钩,使其在开机自检(POST)之后启动,并将当时的内存地址保存,以便恶意代码的工作完成后,能返回原处继续正常的启动序列。挂钩成功后,利用 PXE API 为计算机分配一个 IP 地址,与远程控制主机进行通信,并可通过 TFTP 协议传输数据。PXE API[5]的结构关系如图 2 所示。

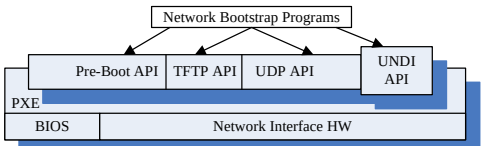

图 2　PXE API 的结构关系

3　BIOS 恶意代码的检测

一旦 BIOS 恶意代码植入计算机,会造成严重危害。因此,如何有效地检测计算机中是否存在 BIOS 恶意代码极为重要。

3.1　BIOS 采样

从计算机主板的 BIOS 芯片中读取 BIOS 镜像文件,是进行计算机 BIOS 安全检测的第 1 步。目前,市场上主流的 BIOS 共有 Award、AMI、Phoenix 3 种类型,其中,Award 已被 Phoenix 收购,其 BIOS 镜像文件的模块组成、遵循规范等也不尽相同,因此,开发出支持多种 BIOS 类型的采样工具具有一定难度。目前常见的 BIOS 采样软件有:winflash(Award 类型 BIOS),WinPhlash(Phoenix 类型 BIOS),amiflash(AMI 类型 BIOS)以及在 Dos 环境下支持多种 BIOS 采样的开源项目 uniflash 等。本节提出一种可用于采样多种 BIOS 镜像文件的设计方案,具体步骤如下:(1)在操作系统下进入 Ring0 层;(2)检测并识别出主板型号;(3)根据 PCI 总线标准,判断出 PCI 设备的设备号(DeviceId)及制造商号(VendorId);(4)查找主板上的 BIOS 芯片,定义一种数据结构 FlashInfo,用以储存芯片的基本信息,包括大小、扇区信息、页面信息等,对 BIOS 芯片进行读写。

3.2　镜像文件分析

获取到 BIOS 镜像文件后,需要对其二进制代码进行模块分解及分析。

(1)模块分解。无论哪个厂家的 BIOS 镜像文件,都由多个 BIOS 功能代码模块或数据模块按照固定头部结构封装并组合形成的。通过顺序查找模块头部的特征字,可顺利分解出 BIOS 镜像文件中各模块内容。以 AMI BIOS 为例,其包含的主要模块如表 1 所示。

表 1　AMI BIOS 主要模块用途说明

模块名称	ID 号码	模块用途
POST	00h	POST 程序代码
Setup Server	01h	AMI BIOS 设置服务器端
Runtime	02h	BIOS 功能函数/常驻的程序代码
DMI Data	06h	数据区
Interface	08h	BIOS 模块压缩/解压子程序模块
ROM-ID	0Ch	记录 ROM ID 与相关信息的模块
INT-13	0Dh	BIOS 软盘/硬盘/光驱读写形式
ACPI Table	0Fh	ACPI 规范控制表格区
Configuration	12h	系统组态显示模块
PCI AddOn Rom	20h	PCI 适配器 ROM 程序代码模块

(2)解压缩。由于 BIOS 芯片存储容量大小有限,大部分 BIOS 模块均通过压缩方式存储。当调用某模块时,需要将其解压缩到内存执行。压缩算法一般采用 LZSS 和 LZINT 算法,利用其对分解后的各模块代码进行解压缩。

(3)恶意代码分析。根据“2 BIOS 恶意代码的实现”,可针对恶意代码的检测,将 BIOS 的“高危模块”定义为 ACPI、PCI、ISA 3 种,其余模块均定义为“安全模块”。

在捕捉到高危模块后,将代码送入分析器中进行分析,具体步骤如下:(1)特征码检测。收集目前已有的 BIOS 恶意代码样本,建立样本库,确定该恶意代码对应特征码。对于每个高危模块,查找匹配样本库中的特征码,确定该模块是否存在已知恶意代码。若查找到匹配特征码,则报知用户处理。(2)恶意行为分析。若未查找到匹配特征码,则进一步对高危模块代码进行恶意行为分析。具体方法为:通过模型检验方式,将恶意代码的模式抽象为有穷状态机[6],对被检测

(下转第 21 页)