EFI OS Loader安全加固技术的研究与实现

吴伟民 陈东新 赖文鑫 苏 庆
(广东工业大学计算机学院 广州 510006)

摘 要 对统一可扩展固件接口(UEFI)的体系架构和执行流程进行安全性分析,发现Windows启动过程中EFI OS Loader的可信性校验存在安全漏洞,其可导致Windows启动流程被劫持。针对该安全漏洞,从文件分离保护、开机身份认证和系统关键区域防护3个层次出发,提出了一种基于USB Key启动、动态口令手机令牌和EFI安全防护软件的三层安全加固的方案。将EFI OS Loader文件存放在USB Key中并加密,实现对文件的保护;把动态口令认证服务端置于USB Key中,两者的有机结合实现了高强度的开机身份认证;设计并开发了遵循UEFI规范的EFI应用程序型安全防护软件,实现了对系统关键区域的保护。实验结果表明,该方案的双认证与安全防护机制弥补了相关安全漏洞,增强了计算机系统启动过程的安全性。

关键词 EFI OS Loader,可信性校验,安全加固,身份认证

中图法分类号 TP309.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.037

Research and Implementation of EFI OS Loader Security Reinforcement Technology

WU Wei-min CHEN Dong-xin LAI Wen-xin SU Qing
(School of Computers, Guangdong University of Technology, Guangzhou 510006, China)

Abstract By analyzing the safety of architecture and boot procedure of unified extensible firmware interface (UEFI), it is found that the credibility verification of EFI OS Loader has security risks, which can lead to the hijack of Windows startup process. To avoid the security risks, considering from the three layers of file isolation protection, boot authentication and system critical region protection, a three-layer security reinforcement plan based on USB Key, the dynamic password cell phone token and EFI antivirus software was proposed. Storing the EFI OS Loader file in the USB Key and encrypting it can achieve the file protection. The dynamic password authentication server is placed in the USB Key, and the combination of both mechanism can achieve a high intensity boot authentication. Designing and developing an EFI application security software following the UEFI specification can achieve the protection of the key region of system. The results show that the dual authentication and security mechanism of the program make up the relevant security vulnerabilities, and enhance the security of computer systems during startup.

Keywords EFI OS loader, Credibility verification, Security reinforcement, Identity authentication

1 前言

UEFI提供了一个操作系统与平台固件之间的接口规范[1],用于在初始化硬件后自动加载操作系统。

由于UEFI已逐渐成为主流,国内外对其安全性开展了一系列的研究,并针对一些已知的UEFI漏洞提出了几种UEFI攻击方法,如NVRAM的修改[2]、UEFI驱动的劫持[3]等。另外,国内有学者根据可信联盟倡导的可信度量安全思想[4]提出了几种EFI安全增强方案,如基于USB Key的EFI BIOS增强方案[5]、基于指纹加密的EFI BIOS技术研究[6]等。

在UEFI启动过程的最后一个阶段,会通过EFI OS Loader[7](OS Loader)加载操作系统。OS Loader以文件的形式存放在ESP(EFI System Partition)分区[8]中。其文件名在Windows操作系统中通常为bootmgfw.efi或bootx64.efi;在Linux操作系统中通常为grub.efi或grubx64.efi。OS Loader的设计原理以及执行机制在不同的操作系统中基本相同,所以不失一般性以及基于对桌面PC操作系统市场份额的考虑,本文以Windows操作系统为主要研究对象。

OS Loader的可信性校验漏洞是一个可导致操作系统启动过程被劫持的漏洞。目前没有发现针对该漏洞的安全加固技术研究。因此,本文提出了一个基于USB Key、动态口令令牌和EFI应用型安全防护软件(简称EFI安防软件)的三层安全防护加固方案。

UEFI的启动流程主要包括SEC(Security)、PEI(Pre-

到稿日期:2015-08-03 返修日期:2015-11-04 本文受广州市科技计划项目(2012Y2-00046,2013Y2-00043)资助。

吴伟民(1956-),男,教授,硕士生导师,CCF会员,主要研究方向为信息安全、可视计算、系统工具与平台;陈东新(1991-),男,硕士,主要研究方向为信息安全、系统介入技术,E-mail:dxchen_66@163.com;赖文鑫(1993-),男,主要研究方向为系统介入技术;苏 庆(1979-),男,副教授,主要研究方向为信息安全。

EFI)、DXE(Driver Extension Environment)和 BDS(Boot Device Select)等几个阶段[1]，如图 1 非阴影部分所示。其中，DXE 阶段是驱动执行阶段，用于轮询并加载所有的驱动程序；BDS 阶段是启动设备选择阶段；TSL 阶段是短暂系统载入阶段。在 TSL 阶段，UEFI 会调用 OS Loader 以加载操作系统。但是 UEFI 却没有对 OS Loader 本身进行可信性校验，而是只让 OS Loader 调用自身的校验函数完成校验。因此，只要能绕过校验函数，就可以劫持操作系统的启动过程。

针对 OS Loader 可信性校验安全漏洞，本文设计了一个包含系列安全防护措施的加固方案，如图 1 阴影部分所示。令 UEFI 在 DXE 阶段加载 USB Key 驱动，驱动加载成功与否会影响到在 BDS 阶段引导管理器识别 USB Key 硬件；在 BDS 阶段，引导管理器必须从 USB Key 启动；在 TSL 阶段，验证 USB Key 的 PIN 码，并加载 USB Key 中的 OS Loader 文件至内存，接着调用动态口令手机令牌认证服务端进行二次认证，最后打开 EFI 安防软件对关键磁盘分区进行安全防护扫描。该方案在不影响计算机正常启动的前提下，通过软硬件结合的方式，增强了计算机启动流程的安全性。

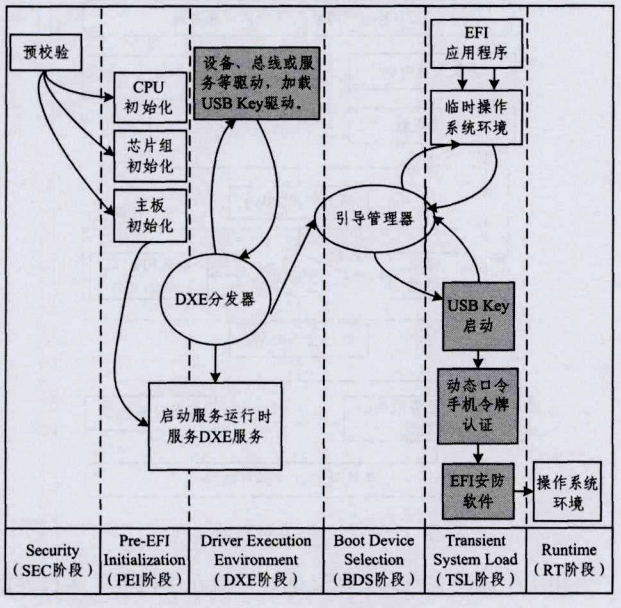

图 1　加固方案技术框架图

2　OS Loader 的安全漏洞分析及验证

2.1　安全漏洞分析

OS Loader 文件格式是 PE32+[9]。PE32+应用程序只能在 64 位操作系统中运行。

IDA Pro[10]是一款支持逆向分析 PE32+文件的静态反汇编工具；Arium ECM-XDP3[11]是一款支持 Intel 处理器调试的硬件仿真器，配合 Arium 的调试软件 SourcePoint，可以实时调试 Legacy BIOS 和 UEFI 等程序。结合 IDA Pro，使用 ECM-XDP3 对 UEFI 程序进行动态跟踪调试，发现 UEFI 并没有对 OS Loader 进行可信性校验，而是让其自身完成校验工作。以 Windows 8.0 为例，UEFI 在加载 OS Loader 时，OS Loader 会调用自身的 BmFwVerifySelfIntegrity（）函数来对自身进行可信性校验，如图 2 所示。如果校验失败，操作系统的启动过程会中断。因此，OS Loader 存在可被篡改的安全隐患，可通过修改其二进制文件绕过校验函数从而劫持启动流程，甚至可以使用其它含有恶意功能的 EFI 应用程序替换 OS Loader 的二进制文件进行攻击。

```
                cmp     byte ptr [rsp+128h+arg_10], dil
                jnz     short loc_1000133B
                lea     rcx, [rdi+38h]
                call    BmFwVerifySelfIntegrity
                cmp     eax, edi
                mov     ebx, eax
                jl      loc_100016FB

loc_1000133B:                           ; CODE XREF: BmMain+2FF↑j
                lea     rcx, [rsp+128h+var_E0] ; _int64
                call    BmResumeFromHibernate
                cmp     eax, edi
                mov     ebx, eax
                jl      loc_100016F6
```

图 2　OS Loader 自检函数

2.2　安全漏洞验证

根据 2.1 节 OS Loader 存在的安全漏洞，本文设计了以下实验方案进行验证。

(1)将原 OS Loader 文件 bootmgfw.efi 重命名为 load.efi。

(2)在 UDK(即 EDKII，一个跨平台的 UEFI 应用开发环境)[12]上开发一个新的 EFI 应用程序，其名字为bootmgfw.efi，将用于代替原OS Loader 文件进行操作系统的加载，以验证利用前述漏洞进行攻击的可行性。此程序主要实现了两个功能：

1)在屏幕上打印字符串后，等待键盘输入；

2)调用 load.efi 进入操作系统。

关键代码如下：

```
if(!BlkIo->Media->RemovableMedia) {
    DstDevicePath = FileDevicePath(HandleBuffer[i], L"\\EFI\\
    Microsoft\\Boot\\load.efi");
    Status=BS->LoadImage(TRUE, ImageHandle, DstDevice-
    Path, NULL, 0, &FileImageHandle);
    FreePool(DstDevicePath);
     if(!EFI_ERROR(Status)) {
          Print (L"Press any key to continue loading Windows…\r
          \n");
          Input (NULL, InputString, 20);
          Status = BS->StartImage (FileImageHandle, & Exit-
          DataSize, & Exit Data);
  }
  continue;
}
```

(3)ESP 分区是一个隐藏分区，在 Windows 系统中可以通过 DiskGenuins 等专用的硬盘分区管理程序获取和替换 OS Loader 文件。图 3 展示了使用 DiskGenius 将原 bootmgfw.efi 文件重命名为 load.efi 以及将步骤(2)中编写好的 bootmgfw.efi 复制到 ESP 分区的情况。

图 3　重命名原 bootmgfw.efi 文件和复制新的 bootmgfw.efi 文件

(4)重启计算机。在计算机启动过程中会调用 bootmgfw. efi 应用程序,如图 4 所示,在接收任意键盘输入后,会调用 load. efi 加载操作系统。

图 4　执行 EFI 应用程序

实验结果证明,可通过编写新的 OS Loader 文件替换原文件,从而绕过 OS Loader 的自检,实现对启动流程的劫持。

3　安全加固技术的设计与实现

3.1　加固方案的设计

本方案从 OS Loader 文件分离保护、开机身份认证和系统关键区域防护 3 方面进行安全性考虑,实现三者的有机结合,保障计算机的安全启动,如图 5 所示。

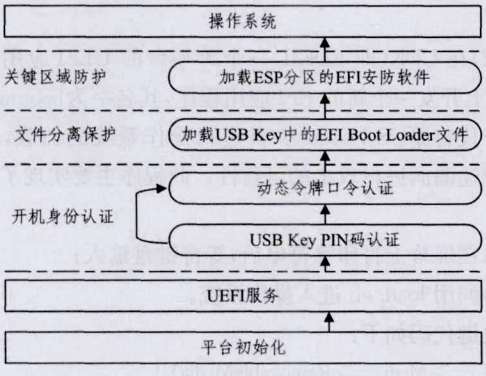

图 5　安全加固整体方案设计

本方案利用 USB Key 内置储存器和高强度加密算法等优势,实现 OS Loader 文件与操作系统的物理分离,将 OS Loader 文件放在 USB Key 中;利用动态口令令牌一次一密的优点[13],在加载 USB Key 中的 OS Loader 文件前需输入口令进行认证;认证成功后,ESP 分区内的 EFI 安防软件自动运行,对系统关键区域进行扫描。

3.2　加固技术的实现

3.2.1　USB Key 技术的实现

在 UEFI 启动流程的 DXE 阶段结合运用 USB Key 技术,将 OS Loader 文件加密存储在 USB Key 中,实现了对 OS Loader 文件的保护,如图 6 所示。

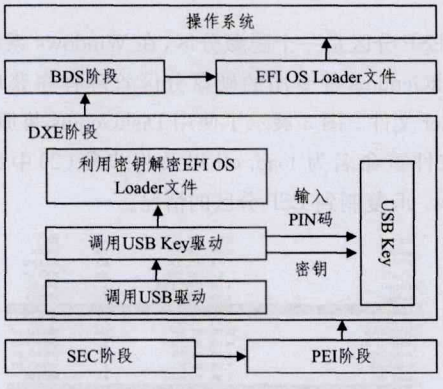

图 6　USB Key 启动过程

利用 EFI 启动项管理的功能调整 EFI 驱动的启动顺序,在 DXE 阶段加载 USB Key 驱动程序,通过 ConnectController()服务及控制传输模式与该设备通信,实现身份认证及其它控制操作。

3.2.2　动态口令手机令牌的具体实现

动态口令手机令牌认证服务端(以下简称服务端)是一个在 UDK 环境下开发的 EFI 应用程序,存放在 USB Key 中,用于计算机启动时认证用户身份。手机动态口令客户端(以下简称客户端)是一个用于生成开机密码的手机 App。

图 7 描述了计算机、客户端和服务端之间的协作认证流程。首先服务端认证程序获取系统时间,调用 srand()函数将其设置为随机种子,再调用 rand()函数生成 6 位随机数。这两个函数都位于 C 语言接口库的 stdilib. h 文件中。将该 6 位随机数作为动态因子,与预留私有密钥组成新的字符串。采用 MD5 散列算法,以新字符串为输入消息串生成 6 位动态密码 PSW1。与服务端类似,在客户端采用相同的方法生成 6 位动态密码 PSW2。在服务端中将 PSW1 与 PSW2 进行匹配,若匹配成功,则加载 OS Loader 文件;否则,输出错误提示。

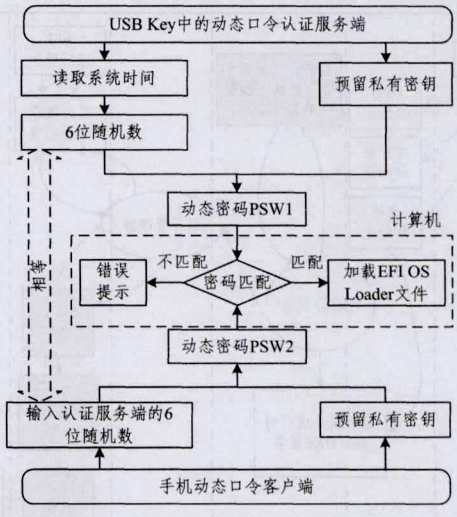

图 7　动态口令手机令牌实现

用于生成动态密码 PWS2 的客户端在手机上的运行界面如图 8 所示。

图 8　客户端生成动态密码

在认证过程中,服务端提供的 6 位随机数存在两种可能的失效情况:一种是有效期 60 秒过后立即失效;另外一种是用户输入的密码 PSW2 与服务端生成的密码 PSW1 匹配不成功。以上两种情况下服务端都会重新生成随机数。图 9 描述了用户输入错误密码后重新生成随机数的过程。在当前方案中,手机无需联网,也无需与计算机时间同步。

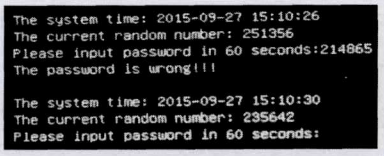

图 9　服务端程序界面

计算机登录日志记录所有登录认证结果，如图 10 所示。若同一用户密码认证连续失败 5 次，则自动锁定计算机，需联系管理员解锁。

图 10　计算机登录日志

3.2.3　EFI 安防软件的设计

EFI 安防软件是基于 UEFI 规范开发的 PE32＋的应用程序，其设计模型如图 11 所示。该软件存放在 EFI 系统分区中。在加载 OS Loader 文件进入操作系统前会调用 EFI 安防软件，EFI 安防软件通过挂载安全防护引擎对系统分区和 EFI 系统分区进行扫描，确保操作系统未被感染，保证计算机的安全启动。

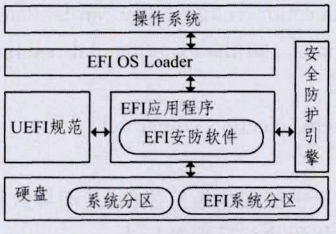

图 11　EFI 安防软件模型设计

为了实验方便，本次实验只对 ESP 分区进行安全防护扫描。如图 12 所示，EFI 安防软件在挂载安全防护引擎后对 ESP 分区进行扫描，在扫描过程中会提示所有的可疑文件。

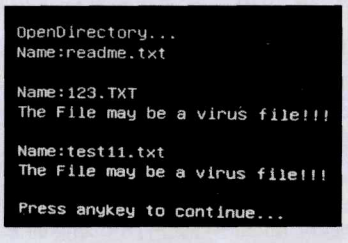

图 12　EFI 安防软件扫描 ESP 分区

结束语　针对 OS Loader 存在可被篡改而导致系统启动过程被劫持的可信性校验漏洞，本文设计并实现了基于 USB Key 启动、动态口令手机令牌和 EFI 安防软件的三层防护安全加固方案，该方案通过双认证与安全防护软件的结合提高了计算机系统启动过程的安全性。在下一步工作中，将在现有 EFI 安防软件技术的基础上继续研究与实现 EFI 云防护技术，进一步提高该技术的适应性和可靠性。

参 考 文 献

[1] UEFI Forum. Unified Extensible Firmware Interface Specification V2.3.1 [EB/OL]. [2012-07-27]. http://www.uefi.org

[2] Bashun V, Sergeev A, Minchenkov V, et al. Too young to be secure: Analysis of UEFI threats and vulnerabilities[C]// Conference of Open Innovations Association. IEEE, 2013: 16-24

[3] Tang Wen-bin, Zhu Yue-fei, Chen Jia-yong. Research on Attack Method of Unified Extensible Firmware Interface [J]. Computer Engineering, 2012, 38(13): 99-101(in Chinese)
唐文彬，祝跃飞，陈嘉勇. 统一可扩展固件接口攻击方法研究[J]. 计算机工程，2012，38(13)：99-101

[4] Chi Ya-ping, Wang Quan-min, Wu Li-jun. A Scheme of strengthening the security of the root of trust for measurement based on USBKey [J]. Information Security and Communications Privacy, 2007(12): 114-117(in Chinese)
池亚平，王全民，吴丽军. 一种基于 USBKey 的可信测量根安全增强设计方案[J]. 信息安全与通信保密，2007(12)：114-117

[5] Yang Shao-qian. Design and Implementation of Strengthing the EFI BIOS Security [D]. Xi'an: Xidian University, 2009(in Chinese)
杨少谦. EFI BIOS 安全增强方案设计与实现[D]. 西安：西安电子科技大学，2009

[6] Shi Jie. Key Technology of EFI BIOS Research and Design Based on Fingerprint Encryption [D]. Tianjin: Tianjin University, 2010(in Chinese)
史杰. 基于指纹加密的 EFI BIOS 关键技术研究及设计[D]. 天津：天津大学，2010

[7] Zimmer V, Rothman M, Marisetty S. Beyond BIOS: developing with the unified extensible firmware interface [M]. Intel Press, 2010

[8] Nikkel B J. Forensic analysis of GPT disks and GUID partition tables [J]. Digital Investigation, 2009, 6(1/2): 39-47

[9] Microsoft Corporation. Microsoft PE and COFF Specification [EB/OL]. https://msdn.microsoft.com/en-us/windows/hardware/gg463119.aspx

[10] Egale. The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler [M]. Beijing: Posts & Telecom Press, 2010(in Chinese)
伊格尔. IDA Pro 权威指南[M]. 北京：人民邮电出版社，2010

[11] Arium Corporation. ECM-XDP3 Intel JTAG Debugger [EB/OL]. [2012]. http://www.arium.com/product/55/ECM-XDP3-Intel-JTAG-Debugger.html

[12] Dai Zheng-hua. UEFI Principles and Programming [M]. China Machine Press, 2015(in Chinese)
戴正华. UEFI 原理与编程[M]. 机械工业出版社，2015

[13] Chen Li-zhi, Li Feng-hua. User Authentication Mechanism Based on Dynamic Password and Its Security Analysis [J]. Computer Engineering, 2002, 28(10): 48-49(in Chinese)
陈立志，李凤华. 基于动态口令的身份认证机制及其安全性分析[J]. 计算机工程，2002，28(10)：48-49