信息系统漏洞挖掘技术体系研究

张友春[1]，魏强[2]，刘增良[3]，周颖[4]

（1. 北京科技大学 信息工程学院，北京 100083；2. 解放军信息工程大学 信息工程学院，河南 郑州 450002；
3. 解放军国防大学 信指部，北京 100091；4. 北京市海淀区环境保护局，北京 100089）

摘 要：首先讨论漏洞挖掘相关的术语定义，分析漏洞挖掘目标对象特点，研究漏洞挖掘的一般流程，然后利用层次结构模型方法，创造性地提出了 5 层漏洞挖掘技术体系架构模型，并详细描述基础层、抽象层、挖掘层、分析层和利用层的内容、作用及其相关支撑技术。最后指出漏洞挖掘技术的发展方向是兼顾各层、协同发展。

关键词：信息系统；漏洞挖掘；目标对象；体系架构；支撑性技术

中图分类号 TP393.08 **文献标识码**：A **文章编号**：1000-436X(2011)02-0042-06

Architecture of vulnerability discovery technique for information systems

ZHANG You-chun[1], WEI Qiang[2], LIU Zeng-liang[3], ZHOU Ying[4]

(1. College of Information Engineering, University of Science and Technology, Beijing 100083, China;
2. College of Information Engineering, PLA Information Engineering University, Zhengzhou 450002, China;
3. Information Department, PLA National Defense University, Beijing 100091, China;
4. Beijing Haidian District Environmental Protection Agency, Beijing 100089, China)

Abstract: First, the definition of technical terms about vulnerability discovery were presented. The characters of the targets of vulnerability discovery were Analyzed. The general process of vulnerability discovery was studied. Then, by applying layer construction model way, proposed the architecture construction model of vulnerability discovery technique, which was divided into five layers, and also explained the contents，roles and key techniques of each layer. Finally, the future direction for the technique is that a comprehensive and coordinated method is used, with all the five layers taken into consideration.

Keywords: information system; vulnerability discovery; targets object; architecture construction; key technique

1 引言

为避免一些概念性分歧，参照国内外一些权威机构和专家的定义[1,2]，首先给出本文要使用的一些相关术语的定义。

信息系统是指根据规定程序而有组织地生成、收集、处理、存储、传送、接收、显示、分发或使用信息的系统或装置。

漏洞是指在一个信息系统的硬件、软件或固件的需求、设计、实现、配置、运行等过程中有意留下或无意中产生的一个或若干个缺陷，它会导致该信息系统处于风险之中[1]。

收稿日期：2010-03-20；**修回日期**：2010-05-20

基金项目：国家自然科学重点基金资助项目（90818025）；国家高技术研究发展计划（“863”计划）基金资助项目（2008AA01Z420）

Foundation Items: The National Natural Science Foundation of China (90818025); The National High Technology Research and Development Program of China (863 Program)(2008AA01Z420)

漏洞挖掘是指采用一定的信息技术方法去发现、分析和利用信息系统中漏洞的过程。

自软件系统出现以来，人们就开始关注和研究软件的可靠性和安全性问题，软件的测试、静态分析及形式化验证技术等随之逐步发展起来。2000 年前后，著名图灵奖获得者 Tony Hoare 提议将软件验证作为计算机科学中的一个重大挑战性问题，希望能像人类基因组计划那样，通过国际合作，在该方向取得重大进展。作为专注于挖掘软件系统中可利用脆弱性的漏洞挖掘技术，可以说是在这些技术的基础上发展起来的，但又不完全等同于这些技术，渐渐形成了具有自身特色的一个独立的研究分支。

传统漏洞挖掘的目标对象仅仅针对软件系统本身，但目前越来越多的人开始关注存在于电子设备硬件电路或固件系统中的漏洞，而不再局限于软件本身，因此有必要对漏洞挖掘对象的范畴做一个探讨。近些年来，出现了大量的漏洞挖掘系统，也涌现了种类繁多的挖掘技术及方法，但一直以来很少有人去探索和研究以下问题。① 这些技术彼此之间的关系，分析其共性和特殊性；② 挖掘技术的体系架构，这些技术、方法在漏洞挖掘中分别处于哪一层面，解决的技术问题；③ 在挖掘技术体系中，还有哪些挖掘技术有待解决，未来的研究方向。

2　漏洞挖掘相关研究

严格来说，目前并未看到有专门针对漏洞挖掘技术体系架构进行研究的相关论文。这里，本文主要就漏洞挖掘工作流程和挖掘系统框架 2 个相关方面做一介绍。

2.1　漏洞挖掘流程

M. Sutton 等[3]提出了模糊（fuzzing）测试的一般流程，他将模糊测试的流程划分为：识别目标、识别输入、生成模糊测试数据、执行模糊测试数据、监视异常、确定可利用性等 6 个流程，如图 1 左侧所示。2006 年，Funnywei[4]总结了 Fuzzing 测试的框架。2008 年，文献[3]提出的漏洞挖掘流程如图 1 右侧所示。

然而，这些针对挖掘流程的分析与探讨存在不足之处如下。

1) 多数只是针对基于模糊测试的方法进行了漏洞挖掘流程的探讨，不具有一般性。

2) 仅陈述了挖掘的一般流程，没有针对每个流程中涉及的关键技术进行探讨与研究，没有建立相应的层次关系。

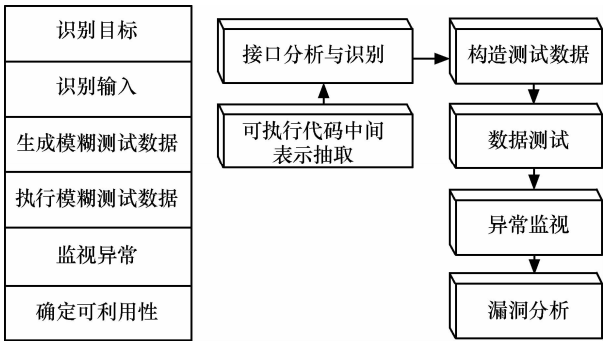


图 1　漏洞挖掘流程

3) 对于异常分析这个环节没有进行深入的探讨，而忽略了漏洞挖掘过程中很重要的分析、利用环节。

2.2　代表性漏洞挖掘框架

早期代表性的 Fuzzing 系统有 SPIKE[6]、FileFuzz[7]、COMRaider[8]等，自 2006 年以后陆续出现以 Peach[9]、Sulley[10,11]等为代表的 Fuzzing 测试系统。J. Demott[12]提出了 EFS 的框架，引入了基于演化的 Fuzzing 挖掘框架。除 Fuzzing 挖掘框架外，还有以 Archer[13]、BitBlaze[14]、SAGE[15]等为代表的符号执行系统，以 BOON[16]为代表的约束分析系统，以 MOPS、SMV、SPIN、SLAM 等为代表的模型检测框架等。

这些挖掘技术架构具有以下局限性。

1) 只能发现特定类型的漏洞，扩展性不好。

2) 提出的都是以某种手段发现某种类型的漏洞，不具有通用性。

3) 研究针对特定平台，假设前提是分析环境、代码、程序都提取好的。而现在的嵌入式系统、专用设备，这些前提可能是不存在的。只针对某种平台漏洞进行挖掘，假设前提针对嵌入式的研究较少。针对专用系统的研究较少。

4) 程序模型的建立都是具体而言，没有更进一步指出如何建模，不适用于动态挖掘本身。

通过上述分析可知，这些研究大多是零散的，针对挖掘技术某一个方面，缺乏关于漏洞挖掘技术方法系统性总结，不能够较好地概括漏洞挖掘整体情况。而漏洞挖掘技术体系架构的研究作为一项基础研究，意义重大。

3　漏洞挖掘目标对象分析

3.1　目标对象范畴

人们通常依据分析对象维度，把漏洞挖掘分为

基于源代码和基于可执行代码的挖掘，这就定义了一个相对狭义的漏洞挖掘对象范畴，即只针对软件系统的漏洞挖掘。但在信息技术飞速发展的今天，漏洞挖掘不应仅局限在这个范畴，而应该是更广义的，涵盖本文所定义的信息系统的挖掘，漏洞挖掘目标对象至少包括以下 4 类（依据挖掘目标对象在信息系统中所处的层次划分）。

1) 数据通信网络基础设施。包含各类通信网络的联网硬件设备和软件。这类网络包括如卫星通信网、光纤网络、移动通信网络、无线网络以及各种专用网络如国防信息网络等。

2) 基础服务对象。包括基于数据通信网络实现的各种基础服务，如 Web 服务、电子邮件服务、软件分发控制中心、国家大型计算中心提供的计算服务、云计算服务等。

3) 高级应用系统。包括使用基础服务所构建的应用系统，如医疗保障系统、电力管理系统、金融系统、后勤管理系统等，主要涉及一些重要部门联网运行的专用硬件及其软件。

4) 终端和接入设备目标对象。包括使用基础服务和高级应用的接入设备或终端硬件以及其上运行的软件。硬件设备如 PC 机终端、智能手机、专用通信终端，一些嵌入式的接入系统，甚至一个传感器。软件则包括桌面操作系统、文字处理软件、图像处理软件等。

3.2 目标对象特点

针对 3.1 节 4 类目标对象的漏洞挖掘研究，总体而言具备以下特点。

1) 设备的逆向剖析及代码抽取、仿真成为挖掘的基础。

漏洞挖掘重要的前提之一是要能够识别、提取、分析和运行设备中的代码（无论软、固件、电路形态），而这些代码在研究过程的测试床必须是一个能够满足漏洞挖掘需求的仿真环境或一个在线的真实环境。然而，生活中接触到大量的网络，如卫星通信网、光纤网络、移动通信网络、无线网络以及各种专用网络等，这就意味着支撑起这些网络含有大量的网络通信设施、基础服务和设备终端，换句话说就是漏洞挖掘必须面对种类繁多的硬件设备和软件系统。而这其中很多设备并非像日常见的 PC 终端、Mobile Phone、路由器那样具有良好通用的仿真、调试和分析环境。因此，对于设备的逆向剖析，并从中抽取代码，提供仿真、调试和分析环境成为漏洞挖掘必须要做的重要的基础工作。

2) 芯片各种各样，核心逻辑很多采用专用芯片，剖析、反编译工作面临巨大挑战。

很多设备的核心芯片并非采用 x86、ARM 这样通用的架构，研究者也就无法利用类似 IDA PRO 这样的专业反编译工具进行反编译和分析，更不能像近来很多的漏洞挖掘工具那样，以 IDA PRO 插件方式进行开发，从而忽略掉本属于挖掘体系中必须解决的前端的反编译分析处理技术。

现代电子设备电路复杂度越来越高、印制电路板的层数越来越多、装配集成度越来越高、结构越来越复杂、软件代码量越来越大，要想准确理清每个处理器系统的软硬件结构和各处理器间的协同关系，这都会是挖掘技术中遇到的一个巨大的难题。

关键电子设备（诸如路由器、交换机、加密网关、防火墙等）通常由单个或多个处理器为核心的硬件电路，以固件形态出现的操作系统、通信协议、安全算法以及应用程序等组成。为提高电子产品的集成度和抗逆向分析能力，大量电子产品中采用了专用芯片，尤其是加密算法的专用芯片。

3) 专用系统种类繁多，挖掘技术的非适用性，有的挖掘技术仅针对特定的系统。

应用需求的增长导致了越来越多嵌入式设备的产生，这里面有诸如 VxWorks，裁剪的 Linux 等系统。以日常使用手机设备为例阐述，主流的智能操作系统就达近 10 种，苹果的 iPhone、诺基亚的 Symbian、谷歌的 Android、微软的 Windows Mobile、黑莓的 Black Berry 等。这些专用系统由于其设备、系统设计的差异，使得漏洞的发现、分析、利用技术都不尽相同，目前也没有一种挖掘系统可以不做任何修改就能应用于这些设备。因此，降低了挖掘技术的普适性。

4) 安全防护机制越来越完善，漏洞挖掘及漏洞分析利用困难重重。

为了提高产品的抗逆向分析能力，电子产品在设计和实现上综合运用了多种安全防护措施：主要有大量使用安全微处理器、ASIC 芯片和可编程逻辑器件增加数据获取的难度，在 ASIC 芯片制造过程中使用抗物理解剖技术，在安全微处理器中使用抗旁路攻击技术，使用凝固性材料对部件或电路板进行灌装以屏蔽测试点，采用防开盖、防窜改、防跟踪、防移动、防探测、防电磁泄露等技术。

操作系统也采用多种安全保护机制，例如

Windows 操作系统自 Vista 开始使用 UAC（用户帐户控制）机制，采用完整性模型来保护系统，使得攻击者利用漏洞得到的权限较为有限。Windows 系统还采用了大量的内存保护技术，如 ASLR（地址加载空间随机化），DEP（数据执行保护），SafeSEH（安全异常处理结构链保护）等。

5) 非合作条件下的逆向剖析与挖掘挑战。

逆向剖析难点体现在：专用硬件、系统缺乏源代码和公开资料。在多数情况下，有些专用设备解剖工作几乎都是在零技术资料的情况下开展工作；另一方面在相当多的情况下，被解剖设备都是备配件缺失的设备，一旦在设备解剖过程的某个环节处置不当使设备受到损坏，将导致工作条件不可再现，使得后续逆向工作难以继续，得出的分析结论无从验证，因此对这类设备的逆向解剖必须保证在非破坏性条件下进行。Windows 操作系统、专用系统都是不开源系统。专用设备也没有相关资料说明。

4　漏洞挖掘技术体系架构

依据目标对象的特点和挖掘工作的一般流程，可以将挖掘技术体系架构划分为基础层、抽象层、挖掘层、分析层和利用层等 5 个层次（如图 2 所示）。

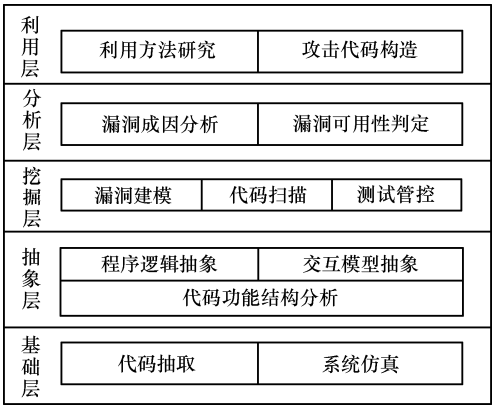


图 2　漏洞挖掘技术体系架构

4.1　基础层

基础层主要解决挖掘分析基础环境的构建及挖掘分析对象的提取。基础层的分析对象是硬件设备与底层代码。一些电子设备尤其是专用设备或特定数字终端，具有特殊的硬件结构和专用的系统软件，为了给软件漏洞挖掘提供调试分析的通用环境，需要对指定设备进行模拟仿真或构造在线分析环境，对特定芯片架构的机器码进行反汇编。基础层通过软硬件代码剥离、仿真、反汇编等技术构造一个可供上层测试分析的运行环境，作为挖掘平台的基础。该层的支撑性技术包括数据在线读取、固件代码还原、软硬件代码剥离、多源目标模拟、多处理器仿真等。

4.2　抽象层

抽象层位于基础层之上，该层主要解决：分析对象是什么问题。通过代码功能结构分析、程序逻辑抽象、交互模型抽象来划分代码的功能模块，理解程序的内在逻辑，解析接口与协议，建立程序与外部环境之间交互模型的描述，并为挖掘层分析和测试提供必要的统一的抽象表示。

代码功能结构分析主要完成代码的功能模块划分、数据结构分析，建立程序的执行状态模型，建立程序的逻辑抽象，识别程序的外部接口可能的接口数据格式，从而建立程序与外部的交互模型。这里面涉及的技术还包括：控制流分析、指向分析、数据流分析等，过程间调用关系图的构建等。其中，指向分析包含别名分析、指针分析、形态分析、逃逸分析等。

4.3　挖掘层

挖掘层主要解决采用什么样的方法对目标对象进行漏洞挖掘。具体来说，主要通过漏洞建模、代码扫描、测试管控等方法，对潜在的程序代码漏洞和系统安全机制漏洞进行挖掘。

该层支撑性技术包括漏洞模型构建、安全模型逆向抽取、测试过程控制、数据构造方法等。漏洞模型构建技术其出发点在于对不同漏洞建立相应合适的模型来发现，包含模型检测、类型系统、抽象解释、符号执行、约束分析技术等。代码扫描技术泛指常见的静态分析技术，典型的商业工具有 Fortify，Coverity Prevent 等。安全模型逆向抽取，其关键在于通过安全机制分析，逆向分析与抽取系统或软件访问控制模型，从而归结出其安全模型，定位其脆弱点。测试管控技术包含测试过程控制、测试数据生成技术等。其中测试过程控制用于保证测试过程是可控的，并朝着一个良好的方向演化，包含测试控制理论和测试覆盖评价理论[17]。测试数据生成包含静态和动态方法，其核心问题在于测试用例的约简、测试集合的合理构造，其他方法还包含蜕变测试构造[18]等。

4.4　分析层

分析层主要解决确认发现的疑似漏洞是不是

一个真正的漏洞。通过程序执行调试，上下文环境分析，定位并确认漏洞，记录漏洞发生的执行过程，便于进一步分析判定，对漏洞的可利用性做出判定。该层的支撑性技术主要包含可控数据追踪、执行控制分析、异态监管技术。

执行控制分析主要指如何发现和分析漏洞的执行控制流。包含数据回溯技术、指令追踪技术、控制流重定向分析技术。可控数据追踪技术主要实现注入点定位、污点传播分析、数据流分析等技术。

异态监管技术通过检测程序的运行状态、执行轨迹、输出特征来发现程序是否从正常状态进入了一个威胁状态，从而识别漏洞的真实有效性。异态监管方法包含从程序的正常输出中获取信息，插装代码（如 Valgrind[19]）来获取信息，系统平台接口获取信息、异常点捕获技术等。

4.5 利用层

利用层主要解决在确认漏洞之后，对其可利用性或危害性进行真实判别。通过攻击元构建、有效载荷组装、保护机制突绕技术实现对不同系统、不同设备的漏洞利用，并研究其稳定性和可靠性。该层的支撑性技术包括攻击代码建模、漏洞适应性利用、关键保护绕防技术等。ImmunitySec 公司开发 Canvas VisualSploit 插件[20]在 Shellcode 构造、攻击代码生成上，具有一定的自动化和较好的可视化效果。

5 结束语

本文通过分析漏洞挖掘目标对象的特点，结合漏洞挖掘的一般流程，提出了漏洞挖掘技术体系架构的 5 层模型，这 5 层模型重点针对漏洞挖掘中发现、分析、利用 3 个环节进行分层设计。

目前的研究活动更多地集中在挖掘层，试图解决某种漏洞挖掘技术问题，提高挖掘的速度和精度。未来漏洞挖掘研究则需要同时关注各挖掘层次的问题，漏洞挖掘技术的发展方向应当是兼顾各层、协同发展。其趋势在目前在业界已经有所反映。

1) 向底层走，加强抽象层和基础层的关键技术研究。例如，BitBlaze 等项目已经将研究的重点放在可执行文件的中间表示技术上，这是抽象层中重要的关键技术之一。近年来，针对手机、路由器漏洞挖掘研究活动也越来越多，电子设备的逆向研究、漏洞及后门挖掘也逐渐成为一个关注的重点，而这些研究都必须有基础层关键技术的支撑才能展开相关研究。

2) 向上层走，逐步开始研究漏洞的成因及可利用性判断自动化分析技术。2006 年，微软公司发布 WinDBG 调试器的!exploitable 插件[21]就是一个可以对 Fuzzing 异常进行可利用性判断的工具。2009 年，funnywei[22]在 XCON 大会上，也介绍了一款漏洞挖掘辅助分析工具。

3) 向前走，针对新型操作系统、新型安全机制、云计算服务等计算模型的新漏洞形态展开研究。

参考文献：

[1] ZHONG W S. Review and outlook of information security vulnerability analysis[J]. Journal of Tsinghua Univ (Science and Technology), 2009, 49(2): 2065-2072.

[2] National information assurance (IA) glossary, CNSS instruction No.4009[EB/OL]. http://www.cnss.gov/assets/pdf/cnssi_4009.pdf. 2010.

[3] SUTTON M, GREEN A, AMINI P. Fuzzing: Bruce Force Vulnerability Discovery[M]. Addison-Wesley Professional, 2007.8-15.

[4] FUNNY WEI. Vulnerability discovery's past, present and future[A]. XCON[C]. 2006. 4-8.

[5] WEI Q. Research on Static Analysis Technology of Executable Code Vulnerability Discovery[D]. PLA Information Engineering University, 2008.

[6] AITEL D. SPIKE[EB/OL]. http://www.immunitysec.com/downloads/SPIKE29.tgz.2002.

[7] SUTTON M. FileFuzz[EB/OL]. http://labs.idefense.com/software/fuzz ing.php，2005.

[8] SUTTON M.ComRaider[EB/OL]. http://labs.idefense.com/software/fuzzing.php#more_comraider,2005.

[9] Peach[EB/OL]. http://peachfuzzer.com/,2006.

[10] Sulley[EB/OL]. http://code.google.com/p/sulley/,2007.

[11] AMINI P.Sulley: fuzzing framework[EB/OL]. http://www.fuzzing.org/wp-content/SulleyManual.pdf,2007.

[12] DEMOTT J.Revolutionizing the field of gray box attack surface testing with evolutionary fuzzing[EB/OL]. http://www.defocon.org/ images/defcon-15/dc15-presentation/DeMott_Enbody_and_Punch/ Whitepaper/ dc-15-demott_enbody_and _punch-WP.pdf,2008.

[13] XIE Y, CHOU A, ENGLER D. An automated tool for detecting buffer access errors[A]. Proceedings of ESEC/FSE 2003[C]. Helsinki, Finland, 2003. 13-18.

[14] BitBlaze: binary analysis for computer security[EB/OL]. http://bitblaze.cs.berkeley.edu/,2008.

[15] GODEFROID P, LEVIN M Y, MOLNAR D. Automated whitebox

fuzz testing[A]. NDSS'08:Network and Distributed Systems Security[C]. 2008. 151-166.

[16] WAGNER D, FOSTER J S, BREWER E A. A first step towards automated detection of buffer overrun vulnerabilities[A]. Network and Distributed System Security Symposium[C]. San Diego, CA, USA, 2004.3-17.

[17] LIU L, MIAO H K. axiomatic assessment of logic coverage software testing criteria[J]. Journal of Software, 2004, (9):1301-1310.

[18] GUO D, NIE W, HAI C, *et al*. Effectively metamorphic testing based on program path analysis[J]. Chinese Journal of Computers, 2009, 32(5):1002-1013.

[19] Valgrind[EB/OL]. http://valgrind.org, 2006.

[20] Canvas visualspoit[EB/OL]. http://www.immunitysec.com/,2004.

[21] !exploitable crash analyzer, MSEC debugger extension[EB/OL]. http:/www.codeplex.com/msecdbg,2007.

[22] FUNNYWEI, An introduction to assistant vulnerability exploitability analysis tools[A]. XCON[C]. Beijing, 2009.

作者简介：

张友春（1963-），男，安徽长丰人，博士，高级工程师，主要研究方向为通信和信息安全。

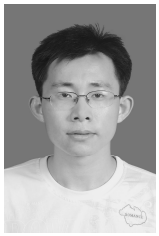
魏强（1979-），男，江西南昌人，解放军信息工程大学讲师，主要研究方向为网络信息安全。

刘增良（1958-），男，河北石家庄人，博士，解放军国防大学教授、博士生导师，主要研究方向为人工智能、信息战和网络信息安全等。

周颖（1967-），女，四川成都人，北京海淀区环保局工程师，主要研究方向为环保和网络技术。

（上接第 41 页）

作者简介：



王鼎（1982-），男，安徽芜湖人，解放军信息工程大学博士生，主要研究方向为阵列信号处理和无源定位。

潘苗（1982-），女，辽宁本溪人，江南计算技术研究所助理工程师，主要研究方向为网络信息处理和阵列信号处理。

吴瑛（1960-），女，河南郑州人，解放军信息工程大学教授、博士生导师，主要研究方向为数字信号处理，阵列信号处理及其 DSP 实现。