基于多核架构的安全漏洞分析平台研究

焦婉妮[1] 吴开贵[2]
(武汉数字工程研究所 武汉 430074)[1]
(重庆大学计算机学院 重庆 400030)[2]

摘 要 军工数字制造网络安全漏洞分析存在漏洞信息难以获取、分析和验证耗时长而且精度低等问题,传统的企业信息安全漏洞分析、测试验证技术不能够完全解决该问题,所以提出了基于多核架构的漏洞分析、验证、测补技术平台。将漏洞探测与分析技术、启发式漏洞渗透性测试技术、动态临时补丁生成和安装技术与基于多核处理器的软件架构相结合,实现漏洞分析、验证与测补的并行处理。原型系统实现与测试表明,该平台能够及时发现、完备验证、准确测试与有效补防军工企业数字化制造网络信息的安全漏洞,实现全生命周期功能平台与应用设备,为构建军工企业可信计算网络以及安全等级评估等工作提供有力的基础支撑。

关键词 漏洞分析,多核架构,系统安全
中图法分类号 TP309 **文献标识码** A

Security Vulnerabilities Analysis Technology Based on Multi-core Architecture

JIAO Wan-ni[1] WU Kai-gui[2]
(Wuhan Digital Engineering Institute (WDEI), Wuhan 430074, China)[1]
(College of Computer Science, Chongqing University, Chongqing 400030, China)[2]

Abstract While conducting vulnerabilities analysis and testing in military enterprise network, it is difficult to get the vulnerability information and the analyzing or verifying process will be accomplished in a long time with relatively low precision. While traditional vulnerabilities analysis and testing techniques couldn't solve the problem in a good way, this paper proposed a vulnerabilities analysis platform based on multi-core architecture. The schema of vulnerabilities detecting and analyzing, heuristic penetrating test, dynamic patches creation and installation were combined with the software architecture based on multi-core processors to implement the parallel processing of vulnerabilities analysis, test and control. The implementation and experiment of prototype system prove that this platform can discover, verify, test the hidden vulnerabilities and make perfect patch in time for digital manufacturing network of military enterprises. It also will build a solid base for trustable computing network and security level estimation.

Keywords Vulnerabilities analysis, Multi-core architecture, System security

1 引言

随着军工企业网络的发展,各种网络设备和应用程序呈指数级增长,网络漏洞的规模较为庞大,同时军工企业信息系统对安全性要求较高,对网络上数目巨大的漏洞进行分析、检测及验证测试需要花费极大的资源和时间[1]。传统的漏洞分析技术尽管采用了多线程技术来提高任务处理的速度,但当前处理器的主频已经接近了物理极限,主频对性能的提升空间已经有限,而采用多核多线程处理技术具有普通单核、单线程处理器所未有的性能优势[2]。

为了保障军工系统安全漏洞分析,本文提出了基于多核处理器的安全漏洞分析、验证、测补技术平台,亦即将传统的软件系统漏洞探测与分析技术、启发式漏洞渗透性测试技术、动态临时补丁生成与安装技术、测试用例库构建与维护技术和多核架构下的资源分配及任务调度技术相结合。多核处理器具有可编程、可扩展、高性能等优点,能够极大地满足漏洞分析、检测与验证系统对大流量、高处理性能的要求。建立军工特色漏洞库,能够较及时地发现军工企业数字化制造网络的信息安全漏洞,有效补防平台与应用设备的全生命周期功能,为构建军工企业可信计算网络以及安全等级评估等工作提供有力的基础支撑。

本文首先提出了多核架构下的安全漏洞分析、验证、测补平台,并对平台的特点和结构进行了分析,通过对称多处理方式对多核系统中的安全任务进行负载均衡。接着对该平台工作的各项流程进行了研究,重点阐述了启发式漏洞验证测试过程。最后通过与单核架构的对比实验验证了该平台能够有效分析并减少系统中的安全漏洞数量。

到稿日期:2010-11-03 返修日期:2011-03-04 本文受国家自然科学基金重点项目"大型分布式软件系统的行为监控与可信演化"(90818028)资助。

焦婉妮 女,助理工程师,主要研究方向为物流管理、信息安全;**吴开贵** 男,博士,副教授,主要研究方向为信息安全、可信服务计算。

2 基于多核架构的安全漏洞分析、验证与测补平台

2.1 安全漏洞分析、验证与测补平台框架

军工系统中的安全漏洞分析、验证、测试及补防平台从功能上需要实现“三个基准、两个环境、三项技术以及三种管理”，其中“三个基准”是构建军工企业补丁基准库、漏洞基准库以及用例基准库；“两个环境”主要是指形成多核计算环境和虚拟化测试环境；“三项技术”是指完成漏洞分析验证技术、渗透性测试技术以及临时补丁生成技术的研究；“三类管理”是实现全生命周期的漏洞管理、补丁管理以及测试环境管理功能。安全漏洞分析、验证、测试及补防技术研究的总体研究方案如图 1 所示。

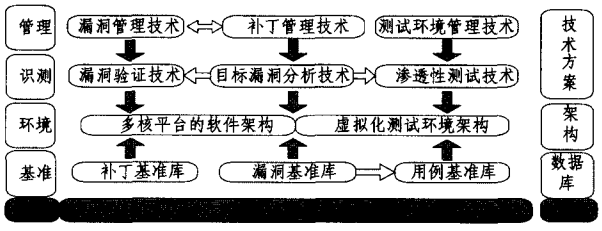

图 1 军工企业信息系统漏洞分析、验证、测补平台

军工企业信息系统漏洞分析、验证、测补平台从功能上分为基准研究、运行环境研究、识测技术研究、管理技术研究 4 部分，从形态上可以分为相关数据库及专家系统研究、软件及系统架构技术研究、应用技术方案研究。安全分析需要满足如下原则：1）漏洞扫描覆盖面广：扫描安全漏洞的范围应能覆盖常用漏洞列表所涵盖的大部分漏洞，同时支持多漏洞数据库的转换，并且能够通过漏洞挖掘技术“尽力”分析未知漏洞，为每个漏洞提供统一的标识号，做到漏洞的命名标准，形成一套标准的漏洞库。2）漏洞分析算法快速、高效：在对网络及应用系统进行漏洞分析的过程中，会占用一定的网络带宽并有可能影响系统的正常运行，因此应当采用一定的技术优化检测过程，以提高分析的速度和效率。3）深度分析、可持续性：漏洞分析、验证、渗透测试结束后，应根据测试结果对目标系统的潜在安全问题进行深度的分析，并给出分析报告。另外，需要对漏洞的全生命周期进行有效的管理，建立同一个目标系统漏洞信息档案，分析网络安全漏洞的走向和趋势，使军工企业网络及应用系统漏洞实现从“确定”、“评估”到“修复”、“报告”、“改进”、“监控”的闭环管理。4）可扩展性强：网络及应用系统漏洞分析技术是一个不断更新、不断发展的过程，因此在系统设计过程中应使其具有很强的扩展性。一方面可以很方便地将最新的安全漏洞信息添加到漏洞数据库中，从而发现各种最新的安全漏洞；另一方面可以不断扩展系统功能，使漏洞分析更全面、更准确。

2.2 基于多核处理器的平台软件架构

基于多核处理器的系统软件架构中最关键的是其资源调度及分配技术，本文将采用对称多处理（Symmetric Multi-Processing，SMP）方式的软件体系结构来解决这个问题。多核架构下的安全漏洞分析、验证与测补软件平台基本工作流程如图 2 所示。

军工企业信息系统漏洞分析、验证、测补过程主要分为漏洞分析、验证与测补 3 个步骤。漏洞检测、分析与验证过程的特点是并行度高、数据流量大、任务过程重复性高等，在系统架构过程中，关键的问题是如何进行合理的任务划分来提高系统的可并行性和负载均衡。本文首先研究了所要处理的任务的基本特点和多核并行处理环境下的并行算法，对整个任务集进行分割，并尽量维持各个分割块工作量均衡，同时利用 SMP 的特性对各个核的运行负荷进行监控、动态调度与分配，最后对处理的最终结果进行规约总结。

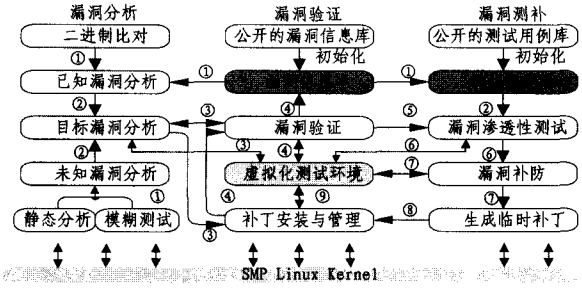

图 2 多核架构下的安全漏洞分析、验证与测补软件平台基本工作流程

3 安全漏洞分析平台工作流程研究

3.1 基于目标代码分析与 Fuzzing 测试技术的漏洞分析

软件平台采用的漏洞分析技术包括针对程序目标代码的静态测试和针对程序运行错误的 Fuzzing 测试。其中目标代码分析部分的设计中采用插件加载的方式增加系统的扩展性，包含模式匹配分析插件、数据流分析差件、词法语法分析插件等[3,4]；在采用基于 Fuzzing 测试技术的分析方法中，其核心分析模块同样以插件形式存在，包含端口 Fuzzing 测试插件、ActiveX Fuzzing 测试插件、文档 Fuzzing 测试插件等[5,6]。漏洞分析流程如图 3 所示。

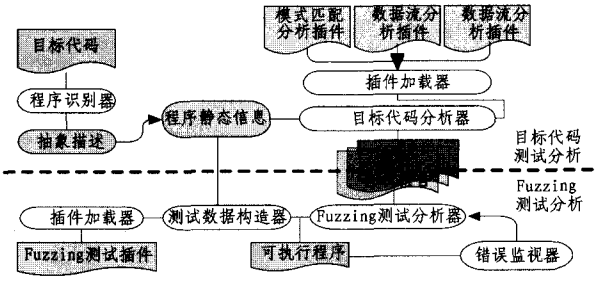

图 3 安全漏洞分析流程

图 3 中目标代码测试分析部分的输入为被测软件的目标代码，程序识别器完成从原始程序到抽象表示形式的转换，本质上是对程序某种粒度上的理解，目标代码分析器在分析插件的驱动下对程序的静态信息进行漏洞分析，最终得到漏洞分析报告；在基于 Fuzzing 测试的动态测试中，测试数据构造器根据插件脚本并结合程序静态信息生成测试模板，根据要插入程序中的正确文件，用随机数据替换该文件的某些部分，由此生成测试用的数据文件。用于自动化测试的大量数据文件，在构造数据时一般根据一定规则来生成，通常采用的规则按深度和广度以及按匹配方式进行数据替换。Fuzzing 测试分析器利用目标程序逐个打开构造的数据文件进行测试。错误监视器在使用目标程序逐个打开每一个构造好的数据文件时，利用钩子技术或系统提供的 API 功能监测系统的状态，监测并记录出现的异常状态及此时所用的数据文件。Fuzzing测试分析器的模式方法包括暴力测试、等价类测试、边界

值测试、组合字段测试等。在分析异常的基础上，可以对测试模式及数据构造方式进行调整，以有效地发现有价值的异常。

3.2 启发式漏洞渗透测试验证

渗透测试以漏洞验证的结果为输入，基于虚拟化测试环境，调用测试用例库中的攻击样本来测试目标，最终形成测试结果，同时向用例库反馈生效的攻击规则并更新测试用例库和漏洞信息库[7,8]。漏洞渗透测试过程如图 4 所示。

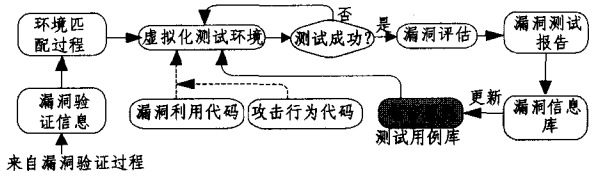

图 4 安全漏洞渗透测试流程

3.2.1 漏洞信息格式定义

由于渗透测试的输入来自于漏洞验证的输出，因此定义合适的漏洞验证信息格式和指标是很重要的，拟采用统一的漏洞验证信息格式，如表 1 所列。

表 1 安全漏洞验证信息格式定义

名称	英文名称	属性
漏洞标识	Vulnerability Identification (VID)	对漏洞的唯一编号，应和标准漏洞库一致
漏洞类型	Vulnerability Type(VT)	标识漏洞所属类型，如缓冲区溢出型漏洞等
载体信息	Container	漏洞所属软件的信息，如应用程序、操作系统的服务等
运行环境	Environment	漏洞能被利用的软件平台，以及对应的操作系统版本
端口信息	Port	远程入侵时所使用的端口
脆弱性函数	Vulnerable Function(VF)	软件中存在漏洞的函数
脆弱性参数	Vulnerable Parameter (VP)	软件中能被利用触发漏洞的具体函数参数

3.2.2 漏洞验证测试流程

测试模块将读取漏洞验证数据包，分析相关信息后动态搭建虚拟化测试环境，并从测试用例库中调用测试样本进行测试[9]。测试用例库将包括 3 个子库：漏洞利用代码(Exploit Code)子库、攻击行为代码(Shell Code)子库和组合规则(Combination Policy)子库。每一个测试用例都是用一个漏洞利用和一个攻击行为组合对目标软件和系统进行的测试，并不是任意漏洞利用和攻击行为都能生效，往往特定漏洞和系统都只能被特定的 Exploit Code 和 Shell Code 成功渗透，因此需要用特定的代码组合去匹配特定的测试目标。

测试用例辅助生成技术的基本模型如图 5 所示，图中漏洞库存储已知漏洞的基本特征及相关补丁信息，测试用例库包含现存的所有测试用例。推理机从漏洞库中获取漏洞的基本特征，从用例库中获取测试用例信息，构成测试用例的推导规则和解释，得到测试用例集。测试模块用于解析测试用例文件并执行渗透测试，若产生相应的预期结果，则表示漏洞存在并结束测试过程；若没有产生任何错误，则执行其它测试用例，直到覆盖整个测试用例集。

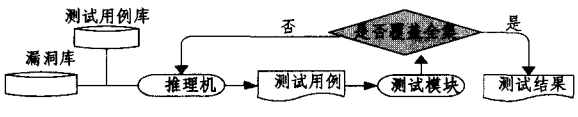

图 5 测试用例生成模型

3.2.3 启发式漏洞验证测试

随着测试用例库样本数的增加，如何高效调用测试样本以及如何提高有效样本命中率成为难点[11]。为解决传统顺序穷举式匹配技术的低效和慢速等不足，拟采用基于漏洞验证信息的自动过滤技术；组合规则子库的优先匹配模式和基于最近生效和最大生效概率的样本调度策略。

首先根据漏洞的验证信息能过滤掉大量明显不适用的测试样本；组合规则子库中存放被证实有效的组合规则，因此优先调用被组合规则支持的测试样本；若还未命中，则优先调用最近生效的样本和生效次数最多的样本。这些技术手段相结合，能极大提高测试用例样本的匹配效率。渗透成功后，如果该渗透测试的攻击组合不在组合规则中，则添加此条规则以更新组合规则子库。被测漏洞被证实能对系统造成安全威胁，对漏洞进行威胁严重程度评估后形成漏洞标签(Vulnerability Label)并输出漏洞测试报告。

通过对系统漏洞的分析和验证，得到系统存在漏洞的描述信息 S，然后从漏洞库中挑选具有相似描述的漏洞集 $V=\{V_1,V_2,\cdots\cdots V_m\}$，这主要是通过计算扫描信息和漏洞库中每个漏洞的特征信息 C 之间的相似度 R_s 来实现的，通过基于统计分析设定相似度阈值(threshold)，选择 m 个相似度 R_s 最高的漏洞。其中最关键的技术为计算扫描信息 S 与漏洞特征 C 之间的相似度，设定合适的相似度阈值(threshold)。对于相似度的计算，主要采用的办法是获得扫描信息和漏洞特征的摘要信息(如关键词、逻辑结构信息等)，通过模式匹配算法计算相似度；相似度阈值(threshold)主要是通过大量试验统计分析得到。

根据漏洞的特征信息关联相应的测试用例(脚本)，对于漏洞 V_i 关联相应的测试脚本 T_i 形成测试用例 TC_i，并按照各漏洞的相似度 R_s 进行排序、去重、简化得到最小覆盖测试用例集合 $TC=\{TC_1,TC_2,\cdots,TC_{\min}\}$，其中最关键的技术为将漏洞按照其特征信息与测试脚本关联起来，主要采用基于专家知识(经验)和动态维护来实现。对生成的测试用例集合进行简化、去重可以提高渗透测试的效率，求取最小覆盖集可以获得较高的效率，而最小覆盖集是 NP 难度的，因此采用近似计算逼近方法使得测试用例集足够小，且能保证测试用例的完备性。

3.3 临时补丁生成与安装

在漏洞分析、验证和测试阶段，可获取足够多的漏洞信息，并将漏洞段定位在函数级(Function Level)，由于目标程序源代码不可得，因此采用内存补丁机制代替正式补丁临时修复安全漏洞，使得只要在补丁装载时，攻击者就无法利用该漏洞[13]。内存补丁的基本工作机制和实现流程如图 6 所示。

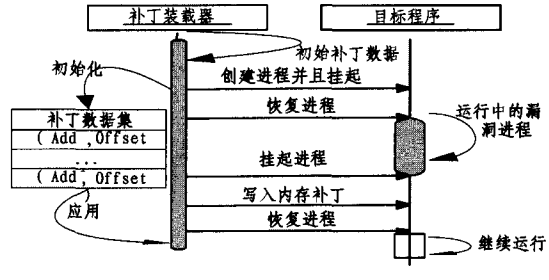

图 6 内存补丁工作机制

在装载临时内存补丁前，需要利用内核调试工具分析目

标软件的运行，根据渗透测试中的数据（主要是漏洞定位信息）确定漏洞段的偏移量和所占字节数，并计算目标程序的循环冗余校验码（CRC），基于这些信息编写内存补丁数据并指定补丁字节数，使补丁恰好覆盖漏洞段所在内存空间，最小化补丁对程序的影响。

临时补丁装载的基本流程为：

（1）补丁数据初始化。根据目标程序漏洞信息动态配置补丁数据。

（2）为目标程序创建进程。用补丁装载器开启目标程序进程是一种方式，也可以将临时补丁依附到目标程序所属进程来实现补丁的装载。

（3）挂起目标程序线程。此时目标进程已分配了内存地址空间，挂起线程以避免内存读写的冲突。

（4）写入内存补丁。根据预处理时所计算的漏洞段偏移量，将配置好的内存补丁数据精确写入漏洞段所在内存地址，以替换可能被利用的内存数据。

（5）恢复目标程序线程。使目标程序继续正常运行，由于预处理时进行了 CRC 计算，因此补丁装载器能保证带有检验机制的应用程序也能无缝运行。

此外，对于使用多线程技术的应用程序，也能通过设计更为复杂的补丁装载器来实现补丁的临时装载。

4 原型系统实现与测试

为了验证该安全漏洞分析、验证、测试和补防平台的架构与工作流程，本文实现了基本的安全漏洞分析、验证和测补平台原型并在虚拟网络环境中对其漏洞防护功能进行了实验。

Tilera Pro 64 多核处理器具有可编程、可扩展、高性能等优点，能够极大地满足漏洞分析、检测与验证系统对于大流量、高处理性能的要求，所以本原型系统基于 Tilera Pro 64 多核处理器构建，如图 7 所示。

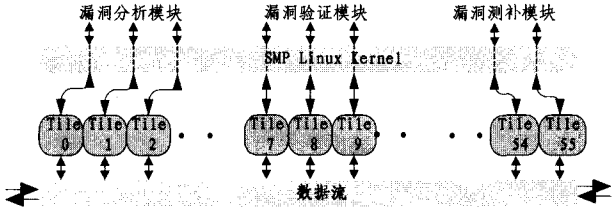

图 7 基于 Tilera Pro 64 多核处理器的原型系统实现图

在虚拟网络环境中对该原型系统进行了实验，首先逐步手动增加系统中的安全漏洞，记录网络中原有的安全漏洞与采用该平台后的剩余安全漏洞总数。经过 10 次测试，原有的安全漏洞数、采用单核漏洞分析机制后的系统剩余安全漏洞数与采用多核平台后的安全漏洞总数的对比如图 8 所示。

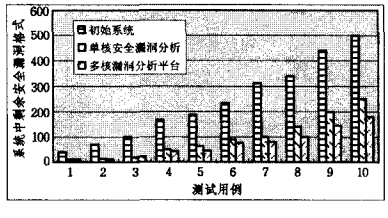

图 8 实验效果对比图

实验结果表明，对大部分测试用例，采用任一种漏洞分析与测补机制，系统中的漏洞数量均有明显降低；与单核漏洞分析机制相比，多核漏洞分析平台能够更有效地降低系统中的漏洞数目，尤其在系统中存在较多漏洞时，多核平台的效果更为明显。

结束语 军工企业数字化制造网络安全保障基础应该以“事前管内、积极防外”为基本思路，其中“管内”需要从自身信息系统的漏洞分析、测试验证入手，“管内”是“防外”的基础，也是系统“防外”的重要手段。本文提出面向军工数字制造业网络的信息漏洞分析、检测以及验证的多核软硬件平台，从而为各大型军工企业网络信息系统提供了生产制造全生命周期的漏洞分析、验证、测试和补防手段。军工网络化制造的各种信息传输是通过网络实现的，因此必须建立一个可信的网络环境，以确保制造企业中以及各制造企业间的各种信息和数据实现安全交换、安全可靠的传输及各制造企业的技术、数据、知识和专利不被非法窃取，为确保军工企业的安全生产、信息数据保密以及信息系统后期安全等级评估等重要的安全保障工作奠定了坚实的基础。本文研究的主要技术可以直接应用于各型武器网络信息平台中，为解决武器装备平台中信息系统漏洞探测、分析以及验证等急迫需求提供了重要手段。

参 考 文 献

[1] Denning D. Cryptography and Data Security[M]. Boston: Addison-Wesley, 1982

[2] Stefan F, Martin M, et al. Large-Scale Vulnerability Analysis [C]// SIGCOMM'06 Workshops. Pisa, Italy, September 2006

[3] Singh A. Identifying Malicious Code Through Reverse Engineering[M]. US: Springer, 2009

[4] Chess B V. Improving Computer Security using Extended Static Checking[C]// IEEE Symposium on Security and Privacy. 2002

[5] Massimiliano D P, Luigi C, Lerina A. The life and death of statically detected vulnerabilities: An empirical study[J]. Information and Software Technology, 2009, 51(10): 1469-1484

[6] Puchkov F M, Shapchenko K A. Static Analysis Method for Detecting Buffer Overflow Vulnerabilities[J]. Programming and Computer Software, 2005, 3(4): 179-189

[7] 梁彬，侯看看，石文昌，等. 一种基于安全状态跟踪检查的漏洞静态检测方法[J]. 计算机学报，2009，32(05)

[8] Brian C, Jacob W. Dynamic taint propagation: Finding vulnerabilities without attacking[J]. Information Security Technical Report, 2008, 13(1): 33-39

[9] Sutton M, Greene A, Fuzzing P A. Brute Force Vulnerability discovery[M]. Boston: Addison-Wesley, 2007

[10] Paul M. Perspectives on Penetration Testing — Black Box vs. White Box[J]. Network Security, 2002, 11: 10-12

[11] 张敏，冯登国，陈驰. 基于安全策略模型的安全功能测试用例生成方法[J]. 计算机研究与发展，2009(10)

[12] Sanjay B. Penetration Testing[M]. Computer and Information Security Handbook, 2009: 369-382

[13] Ozment A. Improving Vulnerability Discovery Models[C]// QoP'07. Alexandria, Virginia, USA, October 2007

[14] 巫茜，张栋，包坤. 基于角色的工作流平台访问控制安全模型[J]. 重庆理工大学学报：自然科学版，2011，25(3)：78-82