软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(4):883−897 [doi: 10.13328/j.cnki.jos.005192]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

拟态防御 Web 服务器设计与实现*

仝 青[1], 张 铮[1], 张为华[2], 邬江兴[3]

[1](数学工程与先进计算国家重点实验室,河南 郑州 450001)

[2](复旦大学 并行处理研究所,上海 201203)

[3](国家数字交换系统工程技术研究中心,河南 郑州 450002)

通讯作者: 张铮, E-mail:ponyzhang@126.com

摘 要: Web 服务器系统作为重要的服务承载和提供平台,面临的安全问题日益严重.已有的防御技术主要基于已知攻击方法或漏洞信息进行防御,导致难以很好地应对未知攻击的威胁,从而难以全面防护 Web 服务器系统的安全.首先提出了攻击链模型,对已有技术的问题和不足进行了深入的分析.在此基础上,提出了基于"动态异构冗余"结构的拟态防御模型,并描述了拟态防御模型的防御原理和特点.基于拟态防御模型构建了拟态防御 Web 服务器,介绍了其架构,分析了拟态原理在 Web 服务器上的实现.安全性和性能测试结果显示,拟态防御 Web 服务器能够在较小开销的前提下防御测试中的全部攻击类型.说明拟态防御 Web 服务器能够有效地提升系统安全性,验证了拟态防御技术的有效性和可行性.最后讨论了拟态防御技术今后的研究前景和挑战.

关键词: 拟态防御;Web 服务器系统;攻击链;系统安全;网络空间安全

中图法分类号: TP316

中文引用格式: 仝青,张铮,张为华,邬江兴.拟态防御 Web 服务器设计与实现.软件学报,2017,28(4):883−897. http://www.jos.org.cn/1000-9825/5192.htm

英文引用格式: Tong Q, Zhang Z, Zhang WH, Wu JX. Design and implementation of mimic defense Web server. Ruan Jian Xue Bao/Journal of Software, 2017,28(4):883−897 (in Chinese). http://www.jos.org.cn/1000-9825/5192.htm

Design and Implementation of Mimic Defense Web Server

TONG Qing[1], ZHANG Zheng[1], ZHANG Wei-Hua[2], WU Jiang-Xing[3]

[1](State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China)

[2](Parallel Processing Institute, Fudan University, Shanghai 201203, China)

[3](National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: The Web server system, being the most important platform of supporting and providing network services, is facing serious security problem. The existing defending technologies mainly deal with the known attacking methods or the known vulnerabilities, and therefore are not effective in case of the unknown threats and do not provide overall defense. This paper first proposes an attacking model to analyze the shortcomings of existing defending technologies. Next, a dynamic heterogeneous redundancy structure based mimic defending model is proposed, and its defending principles and the characteristics are interpreted. Then, the mimic defending Web server is designed on the mimic defending model, and the structure and the implementation principles in the Web server design are introduced. The results of security and performance tests show that the presented mimic defending Web server can defend against all kinds of attacks in

* 基金项目: 国家重点研发计划(2016YFB0800104); 国家自然科学基金(61572520); 上海市科学技术委员会科研计划(14DZ1105300)

Foundation item: National Key R&D Program of China (2016YFB0800104); National Natural Science Foundation of China (61572520); Science and Technology Committee of Shanghai Municipal Research Project (14DZ1105300)

收稿时间: 2016-06-19; 修改时间: 2016-09-08; 采用时间: 2016-11-26; jos 在线出版时间: 2017-01-24

CNKI 网络优先出版: 2017-02-20 13:51:10, http://www.cnki.net/kcms/detail/11.2560.TP.20170220.1351.009.html

the tests with little performance loss, which verifies the effectiveness and the practicability of the mimic defending technology. Finally a perspective of the future work and challenges of mimic defending technology is discussed.

Key words: mimic defense; Web server; attack chain; system security; cyberspace security

1 Web 服务器系统安全现状

网站是人们从互联网上获取信息和服务的主要来源,截止 2015 年底,中国的网站数量已达 426.7 万个[1].从网站上获取各类信息和服务已成为社会生活的重要组成部分.Web 服务器系统作为当前最重要的互联网网站、服务承载和提供平台,是政府、企业以及个人的在互联网上的虚拟代表.由于所存储的文档、所支持的业务以及对受害机构造成的有形损失与无形损失,Web 服务器系统已成为网络攻击的主要目标[2].篡改网页、植入后门、拒绝服务攻击等攻击手段层出不穷,攻击者利用这些手段瘫痪目标服务器的业务,窃取用户敏感信息,或控制相关设备和资源为其所用[1].Web 服务器系统的安全性已成为网络空间安全领域的焦点问题.

安全威胁产生的根本原因在于漏洞的存在,也称为计算机或网络系统的脆弱性[3].根据中国国家信息安全漏洞库的统计[4],漏洞总体数量呈线性增长趋势.近年来,每年新发布安全漏洞数量均在 4 000 个以上,高危及危急漏洞占 1/3 以上,如图 1 所示.

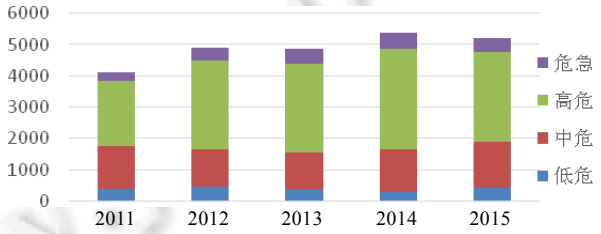


Fig.1 Vulnerability numbers from 2011 to 2015[4]

图 1 2011 年~2015 年漏洞数量[4]

随着安全漏洞的增加和高危漏洞的频繁爆出,近年来各种恶性安全事件频发.2015 年 3 月,美国第二大医疗保险服务商安森公司的信息系统被恶意攻破,近 8 000 万员工和客户资料泄露.同年,全球最大黑客组织“匿名者(anonymous)”攻击了包括湖南警察学院在内的多个中国政府及机构的网站,以声援香港“占中”人士.2016 年 1 月 4 日,由于黑客在乌克兰国家电网中植入了恶意软件,至少有 3 个区域的电力系统大规模被攻击,从而导致发电站意外关闭,伊万诺-弗兰科夫斯克地区近一半的家庭陷入黑暗当中.同年 3 月初,未知黑客入侵孟加拉国央行在纽约联邦储备银行的账户,盗走 8 100 万美元,是有史以来规模最大的网络盗窃案.这些安全事件不仅造成计算机系统服务、软件功能的异常,更严重的是危害了个人用户、企业、政府乃至国家的相关利益.

漏洞的本质是产品缺陷,是产品在设计与实现方面的缺陷和错误导致的“暗功能”,其中既有设计方法和工程实现上的科学技术问题,也有经验积淀和质量管理上的问题,更受限于设计者认识上的局限性.漏洞的成因,表明了漏洞的不可避免性.从一般意义上说,在现有的技术框架内,存在漏洞就存在着被攻击者发现并加以利用的可能性,通过注入、关联或协同等已知或未知攻击方式和手段威胁宿主的信息安全或系统安全.后门是一种特殊的漏洞,即,通过产品设计链、工具链、制造链、加工链、供应链、销售链、服务链等环节被植入产品中的隐蔽的恶意功能.后门被触发时,往往以合法的身份、合法的逻辑运行,很难被传统的入侵检测方法发现.

在网络攻防博弈中,攻击者只需找到系统的一个脆弱点就可以策划攻击,然而防御者则需要进行全面的防护.另一方面,防御者既难以准确预测攻击的策划或发生,也无法短时间内迅速定位漏洞并打上补丁,上述原因导致了防御者的被动不利地位.

已有的典型防御技术,如入侵隔离[5]、入侵检测[6]、入侵容忍[7]和移动目标[8],从不同的角度试图防御网络攻击.入侵隔离将恶意流量阻挡在系统外以防护系统,然而任何隔离技术都不是完全隔离,存在着被攻击者绕过

或穿过的可能.入侵检测技术基于已有攻击的特征进行匹配或检测系统的异常以识别入侵行为,然而存在误报的问题,也很难对新型攻击进行有效防御.入侵容忍通过冗余、多样化设计掩盖入侵造成的影响,达到维持系统的正常服务和功能的目的,但却存在代价高昂的缺点.移动目标防御改变了系统的静态特性,增大了攻击难度,从而实现系统的相对安全,但需要系统进行频繁的变化,给系统性能带来较大的损耗.上述几种防御技术只针对攻击的局部属性或片面的安全需求进行防御,主要的依据是已有漏洞或攻击的相关信息,因而难以达到全面、高效防御的目的.

针对传统防御技术的缺点,本文首先提出了一种攻击链分析模型,利用该模型分析了攻击过程的主要阶段,并指出了已有防御技术的问题和不足.在此基础上,依据攻击对平台具有依赖性这一本质特征,提出了基于"动态异构冗余"结构的拟态防御模型.该模型通过实现多层面的异构性和动态性,降低单步攻击成功的概率,扰乱攻击的反馈信息,增大系统的不确定性,从而达到增加攻击难度和提升系统安全性的目的.在此基础上,设计了基于拟态防御模型的 Web 服务器,实现了多层次的"动态异构冗余"结构,并通过安全性和性能测试验证拟态防御模型的有效性和可行性.安全性测试和性能测试结果显示:在包括扫描探测、Web 应用类攻击、系统突破、系统损毁、后门植入、后门触发等在内的所有模拟攻击测试中,拟态防御 Web 服务器都可以有效防御,同时没有明显降低系统性能.相比于已有的防御技术,拟态防御 Web 服务器能够在不明显降低系统性能的前提下,有效提升系统的防御能力.

本文第 2 节介绍已有的防御技术,提出攻击链模型,对已有的防御技术进行对比分析.第 3 节提出拟态防御模型,阐述模型的特点和优势,并分析防御特性和效果.第 4 节描述拟态防御 Web 服务器的架构和功能.第 5 节对拟态防御 Web 服务器进行安全性和性能评估.最后讨论今后可能的研究方向和前景,并总结全文.

2 已有防御技术

在网络攻防博弈中,防御技术是对抗网络攻击的直接手段.自信息系统安全技术发展以来,出现了许多防御技术,典型的,如入侵隔离、入侵检测、入侵容忍和移动目标等,这些技术有着不同的防御特点和防御对象.本节首先提出攻击链模型,继而依据该模型对 4 种防御技术进行介绍.

2.1 攻击链模型

将防御目标作为一个系统考虑,这个系统可以是一台主机、一个 Web 服务器系统或其他计算机系统或者一个局域网等.攻击者攻击系统的目的主要是对系统实施破坏,以使系统功能或服务受损,或通过入侵窃取系统的敏感信息,如用户信息、机密文件等.常见的攻击方式包括恶意钓鱼、病毒感染、植入木马或恶意软件、拒绝服务攻击、社会工程学攻击以及渗透攻击等.对于典型的攻击行为,攻击方法也不尽相同.虽然攻击方式和种类千差万别,但根据攻击者与系统的交互过程,攻击链可以分为扫描探测、漏洞挖掘、攻击植入和攻击维持这 4 个阶段,具体如图 2 所示.

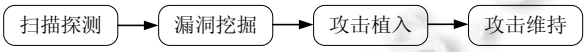


Fig.2 Attack chain model

图 2 攻击链模型

- 扫描探测:对于一个攻击目标,攻击者通常会通过扫描探测等方式获取系统的详细信息,如操作系统版本号、使用的各类软件版本信息等,进一步可通过释放探针等方法获取系统的组成结构等;
- 漏洞挖掘:基于已获取的信息,攻击者通过在线交互或离线的方式搜寻目标存在的可利用的漏洞,包括应用程序、协议、硬件等的各种漏洞;
- 攻击植入:攻击者通过利用漏洞将恶意代码或恶意数据传输给系统,试图破坏系统功能或获取敏感信息.该阶段可能只是植入后门或使攻击具备条件,不一定会使系统发生功能故障、服务受损或其他违反系统安全策略的异常行为;
- 攻击维持:攻击者达到入侵目的后,可能会埋下后门作为下一次的入口,或安装恶意程序使攻击持续

发作.

攻击链模型概括了攻击者的攻击过程,通过该过程,攻击者最终可以达到损毁系统服务或窃取敏感信息的目的.对于一个完整的攻击过程,通过阻断攻击链的任一环节,都可以达到防御攻击的目标.

2.2 已有防御技术分析

已有的防御技术通常是基于对已有攻击手段的了解,通过一定的方法作用于攻击链的部分环节,以期“斩断”攻击链,阻止攻击的进行或攻击目的的达成.防御技术的防御强度越强,对攻击的阻拦效果越好.然而,已有的防御技术很难在攻击的各个环节都有较强的防护强度.同时,由于已有防御技术主要基于已有攻击的特性进行防御,防御过程相对于新型手段具有滞后性.漏洞的不可避免性与攻击方式的演进,造成现有防御技术在特定环境下会失效.因而除了防御强度之外,防御面的大小也是影响防御能力的一个重要因素,如对攻击链的覆盖度,防御各种攻击类型的有效性.同时,安全性提升所需要的代价也是一个不可忽略的因素,如实现防御技术所需要的硬件资源、软件资源、性能代价等因素.在安全与代价之间权衡,也是选择防御技术的一个必须要考虑的因素.依据攻击链模型,可以从防御强度、防御面和安全代价这 3 个方面对已有安全技术进行分析,表 1 给出了入侵隔离、入侵检测、入侵容忍和移动目标这 4 种典型技术在攻击链上的防御阶段和能力强弱的分析结果.

Table 1 Defending parts and ability of the existing defense security practice in the attack chain

表 1 已有防御技术在攻击链上的防御阶段和防御能力

	扫描探测	漏洞挖掘	攻击植入	攻击维持
入侵隔离	弱	无	弱	弱
入侵检测	强	无	强	强
入侵容忍	无	无	强	强
移动目标	强	弱	强	弱

- 入侵隔离

入侵隔离技术是将攻击阻挡在系统以外,阻止入侵者对系统的访问[5].典型的入侵隔离技术,如防火墙、访问控制、VPN(virtual private network)、网络开关等,这些隔离技术通过限制访问来阻止可能的入侵.入侵隔离技术可以隔离攻击者与系统的交互,如拦截恶意流量或来自特定对象的流量,但对于漏洞挖掘没有防御能力.从防御面上看,隔离技术对攻击链的覆盖度较好,且对于各种类型的攻击均有一定的防御能力.然而已实现的隔离方法,如防火墙,很容易被绕过或借助合法身份穿过,其防御强度较弱.根据隔离所使用技术的不同,物理隔离的代价相比软件隔离要高,而软件隔离,如 VPN、防火墙等被透过的风险较大,因而,入侵隔离技术通常作为一种粗粒度的基础防御技术.

- 入侵检测

入侵检测是通过收集系统中关键节点的信息并对所收集的数据进行分析,从而发现违反系统安全策略或引发系统故障的行为[9].入侵检测的方法主要分为两种:异常检测和误用检测[10].基于这两种基本方法产生的入侵检测的技术繁多,实际应用非常广泛,如 Snort[11].除此之外,有多种检测算法综合了不同类型的入侵检测技术,实现误报率更低、检测更全面的入侵检测[12−14].

入侵检测技术的防御强度随检测粒度的大小而变化.检测粒度越细,防御强度越强.常见的入侵检测系统工具,如 Snort[11],NIPS[15],能够实现较高的检出率.入侵检测系统能够在扫描探测、攻击植入和攻击维持这 3 个阶段检测出相关攻击行为,如果策略完备,可以检测出已知的各种类型的攻击.机器学习算法的使用使得部分入侵检测系统具备了检测未知的新型攻击的能力,因而入侵检测技术的防御面是比较全面的.然而,目前实际应用的入侵检测仍然主要基于已有攻击的特征信息进行检测,防御上存在滞后性.同时,在分析过程中也存在误报与漏报的问题,影响了检测的正确性.

- 入侵容忍

入侵容忍技术是由容错技术发展而来的.DARPA(defense advanced research projects agency)的“有机保证和可生存信息系统计划”将入侵容忍系统定义为:系统在面临攻击时,仍然可以持续正确地工作且向用户及时提供

预期的服务[16].防御机理是掩盖由攻击导致的系统错误,消除入侵对系统服务造成的影响.典型的入侵容忍技术研究,如欧洲的 MAFTIA(malicious- and accidental-fault tolerance for Internet applications)项目[17]、SITAR(salable intrusion-tolerant architecture for distributed services)系统[16].数据碎片化技术和门限机制[16−19]也被用于入侵容忍系统的设计,以保证数据库的可用性和机密性[20].

根据其防御目的,入侵容忍旨在消除攻击对系统服务造成的影响,只追求高可用性,因而扫描探测阶段和漏洞挖掘阶段并不做特殊的防护工作,这也造成了入侵容忍的防御面小,防御阶段少.由于多数入侵容忍系统采用冗余设计,消耗的硬件或软件资源量大是其主要的防御代价.

- 移动目标防御

移动目标防御的构建、评价和部署机制及策略是多样的、不断变化的,限制脆弱性暴露和被攻击的机会,从而提高系统的弹性,增加攻击者的攻击难度及代价[21].移动目标防御已在多种层面上部署应用,包括网络层、平台层、运行环境层、软件层和数据层等[22−26].其他移动目标系统还包括变形网络(mutable network,简称 MUTE)和移动攻击面系统(moving attack surface,简称 MAS)[27]等.

移动目标防御技术能够模糊攻击者获取的信息,增大攻击者扫描探测和攻击植入的难度;目标频繁变化,能够在一定程度上限制攻击者利用后门或攻击持续发作的能力.移动目标是一种概率性防御,以较高的频率进行变化才具有较高的防御强度.无论变化频率高低如何,移动目标均存在一定的“防御间隙”[28].当攻击者掌握了系统的变化规律后,能够利用防御间隙进行攻击.

已有的防御技术侧重点不同,各有防护缺陷,难以实现全面的防御.为了应对多种攻击,实际应用的防御工具多是各种防御技术的叠加.然而技术的叠加经常带来冗余计算和资源消耗,有时,不同防御技术还可能相互冲突,在提升防御效果上只是事倍功半.同时,整合已有技术也不可避免地需要承担各种技术的实现代价,对于系统的性能影响较大.随着社会环境和网络环境的变化,攻击者的目的、手段在不断变化,已有的防御技术在应对新型的、未知的攻击行为时显得能力不足,需要一种新型的防御技术,从系统整体考虑,实现全面、高效的防御.

3 拟态防御

基于对已有的防御技术和网络攻击本质的分析,本文提出了基于“动态异构冗余”结构的拟态防御技术.拟态防御相比主流的防御技术有着更大的防御面,尤其能够防御基于未知漏洞的攻击.

网络攻击虽然类型繁多、手段各异,但依赖于特定的攻击环境.除了 DDoS 等耗尽资源型的攻击外,网络攻击通常针对特定的漏洞,策划相关的攻击步骤,通过与攻击目标的若干次交互,最终达到攻击的目的.相对于防御而言,攻击的发生是比较容易的,然而网络攻击本身并非易事,越是危害大的攻击,往往越需要精心构造攻击链,任何一个环节的失败,都有可能导致攻击的失败.

网络攻击通常依赖于具体系统的特定属性.不同的系统设计或实现往往使具有类似功能的系统具有完全不同的特性,如不同的 Web 服务软件,Apache,Nginx 和 Lighttpd 在稳定性、安全性、静态文件处理、反向代理等方面各有千秋;不同的操作系统,具有不同的优势和缺陷,如系统权限提升漏洞 CVE-2014-6324 仅存在于 Windows 系统上,而 Linux 系统根本不存在该漏洞.异构系统的这种差异性,也为安全防御提供了可能,如果使用不同的系统提供同一功能,并进行响应比较和动态切换,则原本存在的漏洞所引发的异常就会在比较中被纠正,某段时间存在的漏洞在系统切换后则将消失.如果对这种方法进行多层次组合,则可以进一步降低被攻击的概率.基于以上分析,本文提出拟态防御模型.拟态防御模型是一种“动态异构冗余”结构,破坏攻击对平台、环境的可依赖性;通过异构性设计扰乱攻击的反馈信息,通过动态性设计在时间维度上增大异构性并增大系统的不确定性,从而增加攻击难度,降低攻击成功的概率,防御已知和未知漏洞带来的威胁.

3.1 “动态异构冗余”结构

动态异构冗余(dynamic heterogeneous redundancy,简称 DHR)结构是拟态防御的基本原理,如图 3 所示.计算机系统的功能可以概括为“输入-处理-输出”,即,结构化设计中的 IPO(input-process-output).动态异构冗余结构在“处理”环节使用异构执行体集进行处理,将同一输入通过输入代理复制为 n 份,并分发给执行体集中的 n 个异

构执行体进行处理,将处理结果收集至表决器进行表决,得到唯一的相对正确的输出.异构元素组成异构构件,由动态选择算法选择异构构件组成在线的执行体集.根据运行时的反馈信息,动态选择算法会产生新的执行体集以替换当前集合.

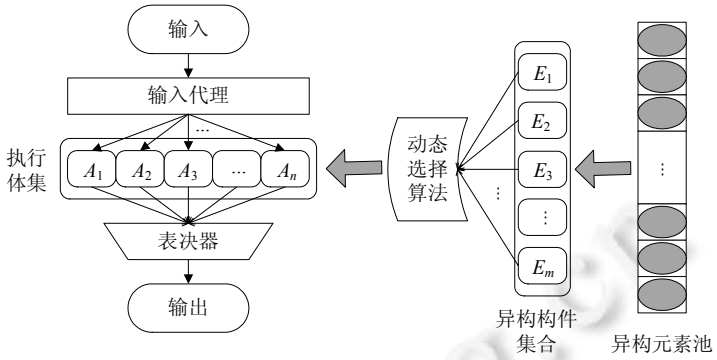


Fig.3 Dynamic heterogeneous redundancy structure

图 3 动态异构冗余结构

动态异构冗余结构的基础是异构性.执行体应尽可能地保证在各种属性或特征上的异构,才能避免相同漏洞的同时出现.相反地,当攻击针对非异构的属性时,动态异构冗余结构则失去了防护能力.异构的层面越多,即执行体异构的属性越多,能够防御的漏洞越多,攻击难度就越高.异构性可以基于多样性实现,应用的多样性、操作系统的多样性、程序设计语言的多样性均能带来一定的异构性.然而,多样性也存在“同源”的问题,即:表面上异构,而实质上是不同版本的更替,或基于同一代码框架的不同实现.因而,基于现有产品多样性实现的异构性并不理想,异构性的实现一方面需要异构化技术的研究,另一方面需要从更多角度提高异构性.

动态性作为异构性在时间维度上的增益,能够补救“同源”带来的异构性不足的问题.动态选择算法使动态异构冗余结构具有动态性.如果当前执行体集被攻击突破,动态选择算法在得到系统反馈后,就会重新生成执行体集替换当前执行体集,改变攻击所依赖的环境,使相同攻击难以维持或再现.另一方面,动态性的存在使系统在不同的时间段表现出不一样的特征,对攻击者呈现出不确定性,进一步增大了攻击难度.

冗余是指多执行体处理同一请求,对比不同执行体的处理结果,通过表决得到相对正确的响应返回给用户.冗余性与异构性相互配合,实现对攻击所依赖的单一环境的改变,增大了攻击难度,提高了系统的安全性.在模型具体化的过程中,执行体集内执行体的个数至少应为两个,才能维持异构冗余的特征,然而两个执行体处理结果不一致时,无法判断出相对正确的结果.3 模冗余[29]是大数表决算法需要的最少执行体个数,也是常见的冗余方式.随着执行体数量的增加,表决策略也需要有适应性的变化,才能得出唯一且相对正确的结果.然而异构执行体的数量的多少与表决结果的正确性并非正相关,相反地,表决执行体数量较多时,反而可能出现受攻击执行体占优势的情形.同时,执行体数量的增加,会导致成本的增长和表决算法复杂度的增大,因而 3 模冗余的实用性更强.

表决器的分布可以采用集中式,如在最终输出前进行一次表决;也可采用分布式,如对每一层的输出均进行表决,主要取决于输出的形式是否便于比较.表决次数的增加会带来响应时延的增加,因而应在提高安全性的同时,尽可能地降低表决的次数.表决在不同的层次进行,也会带来不同的效果:语义层次的表决粒度较难掌握,需要智能化算法的配合,有可能漏报,但能够降低误报率,适合于模糊比较表决;数据层次的表决比语义层次的表决更为精确,减少了漏报,但容易造成较高的误报率,主要用于精确比较表决中.两种表决各有优劣,需要根据不同的应用场景灵活选择.

3.2 防御特性分析

以 3 模冗余的拟态防御模型为例,如图 4 所示.

执行体集包含 3 个执行体,执行体上实现了服务器软件、文件系统和操作系统等多层次的异构化,也存在

未异构化的层次,如PHP脚本以及未在图中显示的硬件层.

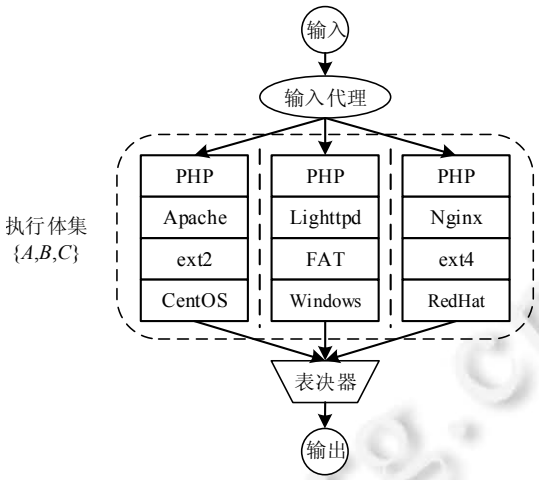


Fig.4 Mimic defending model of triple modular redundancy

图4 3模冗余的拟态防御模型

在某一次攻击交互中,攻击者向系统提交输入,并接收输出以判断攻击结果或策划进一步攻击,当输出显示攻击成功或对策划进一步攻击有益时,认为此次攻击交互成功.对于已知或未知的攻击,不同的攻击交互的难度或代价是不同的,可归结为攻击交互的成功概率的不同.不妨设对单一执行体 A(传统系统架构)第 i 次攻击交互成功的概率为 P_{A_i},则在执行体 B,C 上成功的概率分别记为 P_{B_i},P_{C_i};而在拟态防御结构中,此次攻击交互成功的概率可记为

$$P_i' = P_{A_i} \times P_{B_i} \times P_{C_i} \times V_i,$$

其中,V_i 是3个执行体的输出结果在表决时的一致性.对于特定的某一次交互,输出能否通过表决,其结果是确定的,即:

$$V_i = \begin{cases} 1, & \text{表决结果为一致} \\ 0, & \text{表决结果不一致} \end{cases}.$$

又由于 $0 \leqslant P_{A_i}, P_{B_i}, P_{C_i}, V_i \leqslant 1$,

因而容易推出 $P_i' \leqslant \min\{P_{A_i}, P_{B_i}, P_{C_i}\}$.

且若 V_i=0,则 $P_i' = 0$.

对 $P_{A_i}, P_{B_i}, P_{C_i}$ 的取值进行采样,在 V_i=1 的情况下,对比传统单一执行体和拟态防御结构一次攻击交互成功的概率 $\min\{P_{A_i}, P_{B_i}, P_{C_i}\}$ 和 P_i',计算拟态防御结构下攻击交互成功概率的下降幅度.

$$Dec = (\min\{P_{A_i}, P_{B_i}, P_{C_i}\} - P_i') / \min\{P_{A_i}, P_{B_i}, P_{C_i}\}.$$

统计结果如图5所示.图5的统计结果显示:即使攻击的响应结果均一致,即 V_i=1,拟态防御结构下攻击交互成功的概率也比单一执行体下的概率要低,且在半数以上的情况下攻击交互的成功概率下降幅度超过 60%,说明拟态防御从理论上能够有效降低一次攻击交互的成功概率.

再以利用“菜刀一句话”木马窃取文件为例,攻击者通过木马实现远程控制,将敏感文件回传到攻击端.基本攻击步骤可概括为:

1) 向系统上传“菜刀一句话”木马;
2) 通过“菜刀一句话”木马客户端连接木马;
3) 窃取敏感文件.

上传木马成功后,利用“菜刀一句话”木马客户端连接木马,此时木马在目标主机上运行,并返回所在主机的文件目录到客户端.然而在拟态防御结构下,由于异构的主机文件目录是不同的,因而在表决时发生不一致,表

决器根据表决策略可以输出默认响应,如返回 404 页面,该默认响应可在系统配置中预先设定.此时,在攻击者看来,没有接收到预期的反馈信息,误认为木马未成功上传或木马上传后无法运行,从而难以继续利用木马攻击.

上例说明:拟态防御能够扰乱攻击的反馈信息,从而增大攻击难度.

综合以上分析,拟态防御能够降低单步攻击成功的概率,并扰乱攻击的反馈信息,从而达到增大攻击难度的效果.

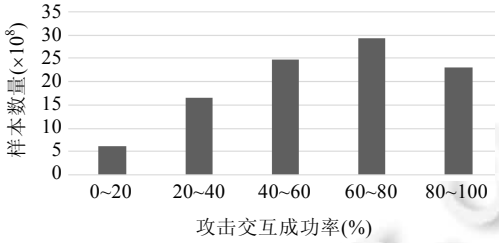


Fig.5 Success probability decrease distribution of *i*(th) attacking interaction

图 5 第 *i* 次攻击交互成功概率的下降幅度分布

3.3 与已有防御技术对比

与已有的防御技术相比,拟态防御技术具有较大的优势.结合攻击链模型分析可以发现,基于动态异构冗余结构的拟态防御技术可以在攻击的每个阶段起到防御效果,如图 6 所示.

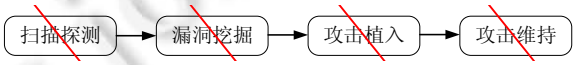


Fig.6 Defence position of mimic security in the attacking chain

图 6 拟态防御在攻击链模型上的防御点

在扫描探测阶段,由于拟态防御模型具有动态性,使得攻击者对系统进行扫描探测的难度加大.在漏洞挖掘阶段,拟态防御模型的输入输出代理隐匿了系统结构和特性,使系统内部结构难以被探测.同时,动态性又增加了系统的不确定性,使得漏洞挖掘难度较大.在攻击植入阶段,拟态防御模型能够通过多层的异构性使攻击无法在多个执行体上同时成功,并通过表决机制,扰乱攻击的反馈信息,降低攻击成功的概率,增大进一步攻击的难度.在攻击维持阶段,拟态防御模型通过动态选择执行体集,适应性地改变系统组成,使攻击的前期工作失效,相同攻击难以维持或再现.由此可见,拟态防御技术能够在攻击链的多个阶段提供较强的防护作用.拟态防御的作用不针对特定的攻击类型,而是从攻击的本质出发,由系统架构内生的特性提供防御能力,因而对已知攻击和未知攻击具有相同的防御原理,均能起到较强的防御效果.

防御代价上,根据不同的实现方法和部署,如异构化的层面不同、异构化的方式不同,采用的算法和模型不同,会带来不同的代价.通过技术的优化,可以将成本从硬件成本缩减为软件成本,并进一步降低性能开销.

总体而言,拟态防御模型能够在多个攻击阶段实施有效的防护,防御面较广,对于各种已知和未知的攻击均具有防御潜能,且成本可控,相比已有防御技术具有较强的防御优势.针对各类防御技术下具体系统的横向比较实验,将能更好地说明拟态防御的优势与特点.然而,由于多数系统不存在开源版本.另外,功能、性能相对均衡的条件下对比安全性的实验变量较难控制,因而该问题有待进一步研究实验.

相比已有的防御技术,拟态防御模型提出的是一种整体协同的防御技术,而不是简单的技术叠加,通过系统内部结构的改变而内生安全性,防御原理是基于攻击“环境依赖性”的本质特征,改变了传统的针对特定攻击类型检测过滤的防御方式,为网络安全防御技术提供了新的研究思路.

4 基于拟态防御的Web服务器

Web服务器是一种多层次的组成架构,自上而下可包括应用软件层、服务器软件层、数据库层、操作系统层以及各层中的子层等,这种结构也为攻击者提供了多重攻击目标,使得 Web 服务器历来防御难度较大.针对Web服务器的攻击过程,通常是一种由上及下的渗透攻击,虽然Web应用层中存在的漏洞数量巨大[2],但其危害性大多是通过底层漏洞发挥的,应用层的漏洞并非威胁的根本原因;存在于服务器软件层、操作系统层的漏洞危害程度高、范围广、危害性强,是Web服务器面临的真正威胁.因而,对Web服务器的防御工作不应停留在上层的应用,而应对Web服务器的各层面实施全面的防御.

基于 Web 服务器的层次结构,在物理操作系统层、虚拟化层、服务器软件层、应用脚本层和数据层实现拟态防御模型.根据 Web 服务器基于"请求-响应"的服务特性,以各层软件或实现的多样性作为异构的基础,实现3模动态异构冗余结构的拟态防御Web服务器,并通过测试验证拟态防御Web服务器的有效性.

基于拟态防御模型,构建拟态防御Web服务器的架构,如图7所示.

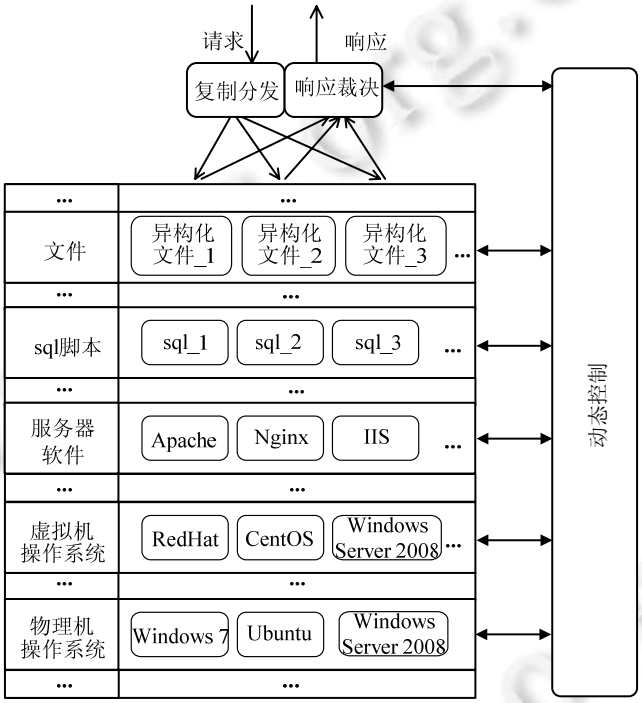


Fig.7 Layer framework of the mimic defending model based Web server prototype

图7 拟态防御Web服务器层次结构

拟态防御Web服务器主要在文件层、sql脚本层、服务器软件层、虚拟机操作系统层以及物理机操作系统层这 5 层实现拟态防御模型.每一层包含了多种可用的异构化的构件供选择,通过兼容性组合构成了多样的服务器软件栈,从而构成异构的服务器执行体.在服务入口和出口处,增设了请求分发和响应比较模块作为输入和输出代理.当请求到来时,分发器将请求复制为3份,分发至3个异构的服务器执行体,响应裁决模块收集比较3个响应的异同,通过表决算法得到唯一的响应输出,同时,将不一致信息发送到动态控制模块.依据Web服务器基于"请求-响应"的服务特性,在返回给用户最终响应之前进行一次集中表决,以缩减中间层表决可能带来的时延.另外,在服务器与数据库之间进行一次对异构sql语句的表决,以保证数据库接收到的查询语句的正确性、合法性.

拟态防御 Web 服务器的异构性主要通过两种方式获得:一种是利用软件、操作系统在实际应用中的多样性直接获得一定的异构性,如服务器软件层由 Apache,Nginx,IIS 等现有软件构成可选集,虚拟机操作系统和物

理机操作系统由 Redhat,Centos,Windows Server 2008 等操作系统构成可选集.另一种是人为异构化,对于 sql 查询语句、文件(包括文件目录)等不具备多样性的数据,采用了关键字标签化、文件标签化、目录随机化等方法,使执行体上的数据具备人为的异构性.基于多样性产生的异构性在一定程度上存在“同源”的问题,相对而言,人为异构化产生的异构性更强.人为异构化的方法多种多样,根据不同的对象形式和特点,通过较小的改变可以取得较大的异构性,异构化方法相对私密,破解难度较大,因而,人为异构化方法能够通过较小的代价实现较高的异构性,从而增强拟态防御模型的内生安全性.

动态控制模块对系统整体进行管理控制,包含了动态选择这一拟态防御模型的关键机制,主要功能包括改变比较表决策略、更改系统或软件的安全性配置、替换本层失效构件、控制执行体集的轮换、清洗下线执行体使其恢复到安全状态等.动态选择机制接收各层的异常反馈信息,如构件运行状态.同时,主要依据响应裁决模块的不一致信息反馈判断异常或失效的执行体,适应性地对异常的执行体进行替换,或依据系统受损程度对执行体集进行替换.除此之外,在无异常情况下进行执行体集的随机变化,增大系统的不确定性,增加攻击难度,在一定程度上抵御长期持续的探测或攻击,如 APT 攻击.

拟态 Web 服务器的原型实现中有两个表决器:一个是在响应返回给用户之前设置表决器进行表决,得到唯一响应,记为前端表决器.另一个是在服务器与数据库之间,对服务器输入数据库的 sql 查询语句进行一次表决,以过滤非法 sql 语句,记为后端表决器.前端表决是在语义层面进行表决,由于不同类型服务器的响应在数据层面上有细节的差异,如包头的 context-type 域、server 域的差异,这些差异属于非语义的差异;又如内容中包含的空格、“/”等符号的个数的差别.为了避免这些差异影响表决结果,前端表决在语义层面进行.后端表决的对象是从多个冗余服务器输入数据库的 sql 语句,sql 语句的异构化由拟态异构化机制实现,是一种不改变语义的数据层异构化,因而在该层面的表决是数据层次的表决.

5 实验评估

为了检验拟态防御 Web 服务器的有效性,对其进行了多方面的测试.以安全性测试为主,设置参照对象,进行模拟攻击对比测试.在此基础上,对拟态防御 Web 服务器进行性能测试,评估其性能损耗.

5.1 安全性测试

安全性测试主要从远程攻击和内网渗透两方面进行,模拟攻击覆盖了入侵链的大部分环节,对比了拟态防御 Web 服务器(device under test-1,简称 DUT-1)、具备典型安全防护的服务器(DUT-2)和无安全防护服务器(DUT-3)在同样攻击下的防御效果.

3 种实验对象,除了拟态防御 Web 服务器为实现多样性和异构性而同时具有多种虚拟操作系统外,其余主机操作系统、服务器软件、Web 应用脚本均一致,典型安全防护 Web 服务器在无安全防护服务器的基础上增加了卡巴斯基安全软件、安全狗等安全工具.3 种实验对象的组成配置见表 2.

Table 2 Configuration of the devices under test

表 2 测试对象配置

实验对象	主机操作系统	虚拟操作系统	服务器软件	脚本、数据	其他安全工具
DUT-1	Windows server 2008 R2, Centos 7	Windows server 2008 R2, Ubuntu 14.04, Centos 6.6, Windows server 2003, Windows XP SP3, Windows 7	Apache, Nginx, IIS, Lighttpd	异构化处理后的脚本和数据	无
DUT-2	Windows server 2008 R2	Windows XP	Apache2.0.63	未异构化处理的脚本和数据	卡巴斯基, 安全狗
DUT-3	Windows server 2008 R2	Windows XP	Apache2.0.63	未异构化处理的脚本和数据	无

在 3 种测试对象上分别安装相同的测试网站,安全配置完全一致.设计的测试案例覆盖了攻击链的各个阶段,共包含 15 项测试.

在扫描探测阶段,使用 Nikto,Nmap 等扫描工具,并利用绿盟远程安全评估系统进行多次扫描和探测,统计扫

描结果,验证拟态防御技术对系统信息呈现性质的影响.

在漏洞挖掘阶段,在已知系统内部含有 Apache 目录浏览漏洞的前提下,尝试利用该漏洞获取网站目录.另外,通过 Burpsuite 等工具截获并修改数据包,诱使访问出错,尝试通过显示的错误路径获取系统后台信息.通过以上测试评估拟态防御技术对漏洞挖掘阶段的防御效果.

在攻击植入阶段,通过利用系统局部的已知漏洞或故意制造漏洞,模拟漏洞利用、木马植入、病毒感染、sql 注入等攻击植入过程.需要特别说明的是,在测试中,虚拟机的 Windows 操作系统保留了 ms08_067 和 ms12_020 两个漏洞,模拟对防御者而言的未知漏洞,并通过预埋后门模拟对防御者而言未知的后门.通过以上测试,评估拟态防御技术对利用已知和未知漏洞后门的攻击的防御能力.另外,通过系统局部宕机评估拟态防御技术对除攻击以外的异常情况的应对能力.

在攻击维持阶段,预埋后门,并修改攻击的表决结果,故意使第 1 次利用后门的攻击成功,而后再次利用后门,检验攻击结果.通过这种测试,验证拟态防御技术对攻击维持阶段的防御效果.

以上测试案例的设计多采用人为预置漏洞或后门的方法,以保证攻击的直接性和破坏性.测试类型遍布攻击的各个阶段,覆盖不同类型的攻击方式,从而在最大范围内、最大程度地对拟态防御 Web 服务器的防御能力进行测试.测试案例及结果见表 3.

Table 3 Testing results of security

表 3 安全性测试结果

攻击阶段	测试项	DUT-1	DUT-2	DUT-3
扫描探测	1. Nikto,Nmap 扫描 2. 绿盟远程安全评估系统扫描	多次扫描探测 得到的信息不一致	多次扫描探测 得到的信息一致	多次扫描探测 得到的信息一致
漏洞挖掘	3. 尝试浏览 Apache 目录	失败	成功	成功
	4. 诱导访问出错,暴露路径	失败	成功	成功
攻击植入	5. 篡改网页	失败	成功	成功
	6. 触发“菜刀一句话”木马	失败	失败	成功
	7. 触发 Weevely PHP 木马	失败	失败	成功
	8. 触发预埋后门	失败	成功	成功
	9. 利用 ms12_020 漏洞	失败	成功	成功
	10. 利用 ms08_067 漏洞	失败	成功	成功
	11. Win32.Alcaul 等病毒感染	失败	失败	成功
	12. sql 注入	失败	成功	成功
	13. 物理主机宕机	失败	成功	成功
	14. Apache DoS	失败	成功	成功
攻击维持	15. 预埋能够攻击成功的后门,并再次触发	失败	成功	成功
总计	15 项	0 项成功	10 项成功	13 项成功

根据表 3 的测试结果可以发现,拟态防御 Web 服务器可以全面实现对攻击的防御目的.具体而言:在扫描探测阶段,拟态防御 Web 服务器(DUT-1)能够变换系统指纹信息,呈现不确定性;在漏洞挖掘阶段,使漏洞的出现具有不确定性,增大了漏洞利用的难度;在攻击植入阶段,无论是针对已知漏洞的攻击还是针对未知漏洞后门的攻击,均有强抵抗力.同时,在服务可靠性方面也强于 DUT-2 和 DUT-3;在攻击维持阶段,拟态防御 Web 服务器能够持续抵御利用未知后门实施的攻击.

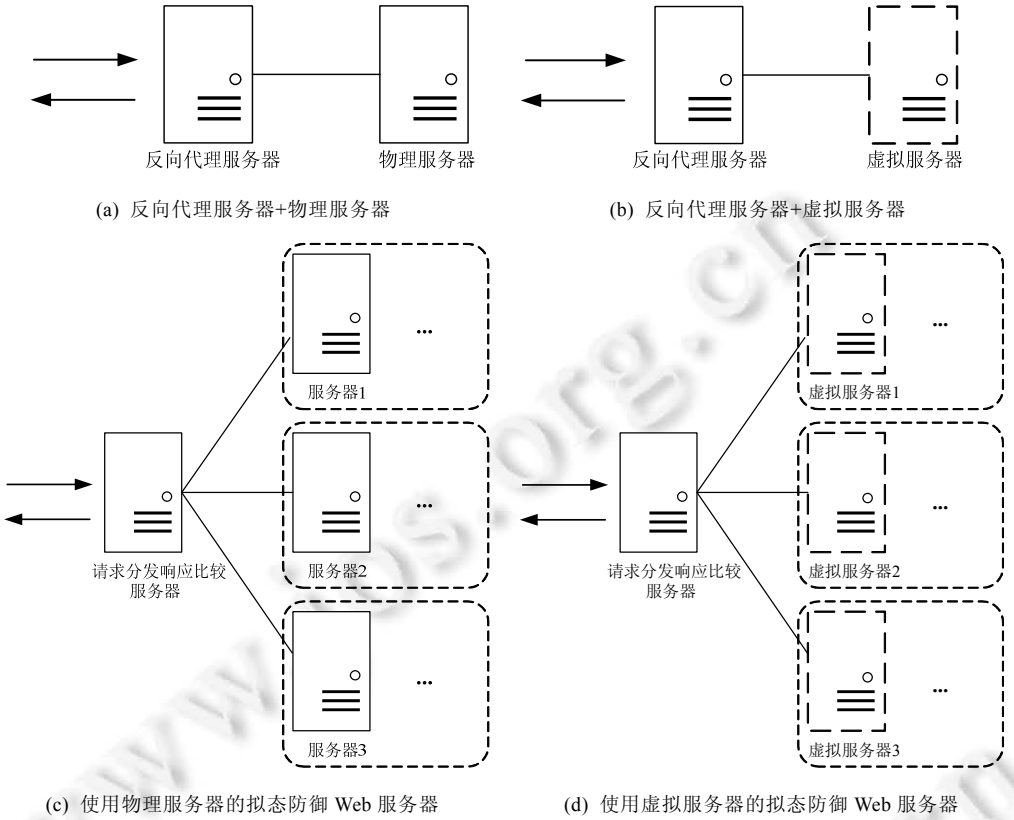
综合以上结果可以得出结论:相比典型的安全防护 Web 服务器和无防护 Web 服务器,拟态防御 Web 服务器在所有攻击案例中均防御成功,安全性最强.

5.2 性能测试

在安全性测试的基础上,测试拟态防御 Web 服务器的性能.将新建速率、并发数、吞吐量和响应时间这 4 项主要指标作为性能评价的考察因素.为了更好地说明性能损耗的原因,采用了对照测试的方法分析拟态防御 Web 服务器的性能代价.如图 8 所示,构建了包括拟态防御 Web 服务器在内的 4 种测试对象.

图 8(a)使用一台反向代理服务器连接物理服务器代表典型的服务器结构,作为基本参照;图 8(b)使用虚拟

机承载的服务器,测试虚拟机服务器的性能;图 8(c)使用物理服务器实现拟态防御 Web 服务器的功能;图 8(d)为拟态防御 Web 服务器的数据流示意图,省略了拟态防御 Web 服务器的感知变迁器等细节,主要用于介绍性能测试的结构.4 种测试对象除了结构不同以外,其余的主机型号、操作系统版本、服务器软件版本等配置对应相同.



(a) 反向代理服务器+物理服务器 (b) 反向代理服务器+虚拟服务器

(c) 使用物理服务器的拟态防御 Web 服务器 (d) 使用虚拟服务器的拟态防御 Web 服务器

Fig.8 Compositions of the four test objects

图 8 4 种测试对象组成

利用 SprientAvalanche 31000b 测试仪仿真大量 HTTP 请求发送至被测服务器,测试图 7 中 4 种测试对象的新建速率、并发数、吞吐量和响应时间,结果见表 4.

Table 4 Testing records of performance

表 4 性能测试记录

测试对象	新建速率(/s)	并发数	吞吐量	响应时间(ms)
(a) 物理机对照	23 643	69 716	984 000	0.004
(b) 虚拟机对照	3 941	65 320	387 000	1.533
(c) 物理机拟态防御 Web 服务器	18 527	65 177	982 000	0.231
(d) 虚拟机拟态防御 Web 服务器	3 880	64 002	301 496	1.704

根据表 4 的结果,有如下结论.

- 对照情形(a)、情形(b)的测试数据可以发现:虚拟机服务器相对于物理机服务器在新建速率和吞吐量上有明显的降低,并发数也有少量降低,同时延长了响应时间约 1.5ms,说明虚拟机服务器代替物理机服务器在较大程度上降低了服务器系统的性能;
- 对照情形(c)、情形(d)的测试数据可以发现,与情形(a)、情形(b)的对照结论基本相同,进一步说明虚拟机服务器对系统性能的影响较为强烈;
- 对照情形(a)、情形(c)的测试数据可以发现,使用物理服务器作为变迁对象的拟态防御 Web 服务器在

新建速率和响应时间上的影响较为明显,说明拟态防御机制的使用有一定的性能代价;但在新建速率、并发数和吞吐量的数量级上没有降低;

- 对照情形(a)、情形(d)的测试数据可以发现:相比于普通 Web 服务器,使用虚拟机作为变迁对象的拟态防御 Web 服务器性能损失较大;然而结合前两条结论分析,可以判断出性能损失的主要来源是虚拟服务器的使用,拟态防御机制造成的性能损失并不是主要原因;
- 对照情形(a)、情形(c)和情形(b)、情形(d)的响应时间差值可以发现,两组对照的响应时间均存在约 0.2ms 的差值,可以推测,该时延差值是分发表决机制带来的固有时延,是拟态防御机制不可避免的时延.

综合以上结论,性能测试的结果表明:新建速率、并发数和吞吐量这 3 项指标在数量级上没有降低,分发表决机制存在约 0.2ms 的固有时延,因而拟态防御 Web 服务器相比同类服务器在性能上有一定的损失.由于性能测试对服务器系统的安全性和功能等方面没有要求对等,因而从性能单方面来看,拟态防御 Web 服务器是降低的.但是拟态防御 Web 服务器目前处于初步研制阶段,存在较大的优化空间.从测试数据中可以看出:物理机代替虚拟机就能够实现性能较大幅度的提高;分发表决算法的优化能够降低拟态防御机制的固有时延.因而在下一步工作中,拟态防御 Web 服务器面临着较大的优化工作量.

5.3 性能优化分析

拟态防御 Web 服务器作为具有“内生安全性”的系统设备,在附加安全工具、系统维护方面的成本整体低于采用传统防御技术的 Web 服务器系统.另外,通过改进拟态防御 Web 服务器的实现技术、优化 Web 服务器的性能,拟态防御 Web 服务器的安全性、功能和性能能够得到较大程度的提升.拟态防御 Web 服务器的优化工作主要包括拟态防御机制相关的实现技术优化和 Web 服务器的运行特性的优化.

针对拟态防御机制在 Web 服务器上的当前实现、分发表决模块以及服务器池的实现存在较大的优化空间.对于前置请求分发-响应表决模块,相比单台 Web 服务器,时延主要来源于该模块,可以通过分发表决模块功能最小化进行优化,以缩短固有时延.通过模块硬件化设计,仅保留分发、比较和表决等关键功能,形成请求分发-响应表决专用组件,以硬件实现代替软件实现,从而尽可能地缩短时延,提高效率,进而降低拟态防御 Web 服务器系统的整体时延.同时,该专用组件的形成,能够为其他拟态防御系统的实现提供高效的解决方案.对于基于虚拟机的 Web 服务器池,可以采用容器技术进行轻量化设计.利用容器承载 Web 应用,使 Web 服务以进程的形式运行于物理载体上,从而将运行时平台、操作系统等基础设施的安全性问题归约为容器安全,同时大大降低软/硬件资源消耗、增强可扩展性并增强拟态防御机制部署的灵活性.

针对拟态防御 Web 服务器的运行特性,由于拟态 Web 服务器在原型实现上侧重于对拟态防御原理的实现和安全性的验证,在对 Web 服务器的选取上,选择了简单搭建的 Web 服务器,对于系统的整体运行特性优化工作有所欠缺.依据传统的 Web 服务器优化方案[30],从整体上可分为硬件优化和软件优化.

- 硬件上,影响因素主要是内存、处理器、网络环境、硬盘等,根据服务器的实际使用场景,采用容量合适、可扩展性强的硬件是较好的选择;
- 软件上的优化又可分为应用的优化和服务器软件配置的优化:应用的优化,如网页静态化设计、建立动态内容缓存等,通过优化应用的设计,避免过多的响应时延和资源消耗;服务器软件的优化包括中间件配置的优化、服务器负载均衡、设置 url 缓存等,这些优化方法均能在降低响应时延、提高资源利用率上产生一定的效果.

另外,鉴于 Web 服务器系统与数据的紧密关联性,数据库的性能优化设计也能为 Web 服务器的性能提升带来益处.基于传统的 Web 服务器运行特性的优化方案,拟态防御 Web 服务器既要优化每个 Web 服务承载主体(执行体),也要优化动态选择机制,保证服务器池中冗余工作的 3 个执行体性能均衡且较优越,以避免“短板效应”.

6 总结与展望

拟态防御基于动态异构冗余结构,从系统结构层面内生安全性.通过异构性扰乱攻击的反馈信息,降低单步

攻击成功的概率.同时,通过动态性在时间维度上增大异构性,增大系统的不确定性,从而加大攻击难度.

相比于已有的典型防御技术,拟态防御在攻击链的各个阶段均具有较强的防御能力,能够应对针对已知和未知漏洞的攻击.

拟态防御 Web 服务器验证了拟态模型的可行性,同时说明拟态防御模型具有较好的发展前景.通过使用不同的异构技术、动态选择机制以及优化拟态防御机制的实现技术,可以构建高性价比的拟态防御 Web 服务器.拟态防御具备灵活的部署方式,能够适应不同的应用场景,在云安全、数据安全、系统安全等领域具有较大的发展空间.

本文分析了系统安全现状,并基于攻击链模型对已有安全技术进行了分析和对比,基于已有防御技术的分析,提出了拟态防御模型,并基于该模型构建实现了拟态防御 Web 服务器.通过安全性和性能测试,验证了拟态防御的有效性和可行性.最后,提出了拟态防御的发展前景和推广价值.

References:

[1] Internet Society of China, CNCERT/CC. China network sites developing situation and security report (2016). 2016 (in Chinese). http://tech.163.com/16/0320/15/BIK212JA00094P25.html

[2] Fang SW, Portante A, Husain MI. Moving target defense mechanisms in cyber-physical systems. In: Securing Cyber-Physical Systems. CRC Press, 2015. 63. https://books.glgoo.com/books?hl=zh-CN&lr=&id=wB6vCgAAQBAJ&oi=fnd&pg=PA63&ots=bkQgsF0K0T&sig=NMqbYCLX0YGm329DhO-0zLmxSIc#v=onepage&q&f=false

[3] Subrahmanian VS, Ovelgonne M, Dumitras T, Prakash BA. The Global Cyber-Vulnerability Report. Springer Int'l Publishing, 2015. 33−64. [doi: 10.1007/978-3-319-25760-0]

[4] China Information Technology Security Evaluation Center. China national vulnerability database of information security. 2015 (in Chinese). http://www.cnnvd.org.cn/vulnerability/statistics

[5] Xu H, Chen X, Zhou J, Wang Z. Research on basic problems of cognitive network intrusion prevention. In: Proc. of the 9th Int'l Conf. on Computational Intelligence and Security (CIS). 2013. 514−517. [doi: 10.1109/CIS.2013.114]

[6] Chung CJ, Khatkar P, Xing T, Lee J, Huang D. NICE: Network intrusion detection and countermeasure selection in virtual network systems. IEEE Trans. on Dependable and Secure Computing, 2013,10(4):198−211. [doi: 10.1109/TDSC.2013.8]

[7] Madan BB, Goševa-Popstojanova K, Vaidyanathan K, Trivedi KS. A method for modeling and quantifying the security attributes of intrusion tolerant systems. Performance Evaluation, 2004,56(1-4):167−186. [doi: 10.1016/j.peva.2003.07.008]

[8] Okhravi H, Hobson T, Bigelow D, Streilein W. Finding focus in the blur of moving-target techniques. Security & Privacy, 2014,12(2):16−26. [doi: 10.1109/MSP.2013.137]

[9] Vasilomanolakis E, Karuppayah S, User M, Fischer M. Taxonomy and survey of collaborative intrusion detection. ACM Computing Surveys (CSUR), 2015,47(4):55. [doi: 10.1145/2716260]

[10] Liao HJ, Lin CHR, Lin YC, Tung KY. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 2013,36(1):16−24. [doi: 10.1016/j.jnca.2012.09.004]

[11] Whitea JS, Fitzsimmonsb T, Matthewsc JN. Quantitative analysis of intrusion detection systems: Snort and suricata. Proc. of the SPIE, 2013,8757:875704-1. [doi: 10.1117/12.2015616]

[12] Kenkre PS, Pai A, Colaco L. Real time intrusion detection and prevention system. In: Proc. of the 3rd Int'l Conf. on Frontiers of Intelligent Computing: Theory and Applications (FICTA 2014). Springer Int'l Publishing, 2015. 405−411. [doi: 10.1007/978-3-319-11933-5_44]

[13] Ho CY, Lai YC, Chen IW, Wang FY, Tai WH. Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems. IEEE Communications Magazine, 2012,50:146−154. [doi: 10.1109/MCOM.2012.6163595]

[14] Song J, Takakura H, Okabe Y, Nakao K. Toward a more practical unsupervised anomaly detection system. Information Sciences, 2013,231:4−14. [doi: 10.1016/j.ins.2011.08.011]

[15] Vaidya N, Godbole P. Hardware implementation of key functionalities of NIPS for high speed network. In: Proc. of the Computing and Network Communications. 2015. 892−897. [doi: 10.1109/CoCoNet.2015.7411296]

[16] Wang F, Uppalli R, Killian C. Analysis of techniques for building intrusion tolerant server systems. In: Proc. of the Military Communications Conf., Vol. 2. 2003. 729−734. [doi: 10.1109/MILCOM.2003.1290202]

[17] Powell D, Stroud R. Conceptual model and architecture of MAFTIA. Technical Report, University of Newcastle Upon Tyne Computing Science, 2003. 23−29.

[18] Wang F, Jou F, Gong F, Sargor C, Gosevapopstojanova K. SITAR: A scalable intrusion-tolerant architecture for distributed services. In: Proc. of the Workshop on Information Assurance and Security. 2003. 38−45. [doi: 10.1109/DISCEX.2003.1194957]

[19] Nguyen QL, Sood A. A comparison of intrusion-tolerant system architectures. IEEE Security & Privacy, 2011,9(4):24−31. [doi: 10.1109/MSP.2010.145]

[20] Yu J, Cheng XG, Li FG, Pan ZK, Kong FY, Hao R. Provably secure intrusion-resilient public-key encryption scheme in the standard model. Ruan Jian Xue Bao/Journal of Software, 2013,24(2):266−278 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4324.htm [doi: 10.3724/SP.J.1001.2013.04324]

[21] Zhang XY, Li ZB. Overview on moving target defense technology. Communications Technology, 2013,46(6):111−113 (in Chinese with English abstract).

[22] Antonatos S, Akritidis P, Markatos EP, Anagnostakis KG. Defending against hitlist worms using network address space randomization. Computer Networks, 2007,51(12):3471−3490. [doi: 10.1016/j.comnet.2007.02.006]

[23] Huang Y, Ghosh A. Automating intrusion response via virtualization for realizing uninterruptible Web services. In: Proc. of the Network Computing and Applications (NCA 2009). 2009. 114−117. [doi: 10.1109/NCA.2009.37]

[24] Shacham H, Page M, Pfaff B, Goh E-J, Modadugu N, Boneh D. On the effectiveness of address-space randomization. In: Proc. of the 11th ACM Conf. on Computer and Communications Security. 2004. 298−307. [doi: 10.1145/1030083.1030124]

[25] Salamat AG, Franz M. Reverse stack execution in a multivariant execution environment. In: Proc. of the Workshop Compiler and Architectural Techniques for Application Reliability and Security. 2008. 1−7. http://babaks.com/files/catars08.pdf

[26] Nguyentuong A, Evans D, Knight JC, Cox B, Davidson JW. Security through redundant data diversity. In: Proc. of the IEEE Int'l Conf. on Dependable Systems and Networks with FTCS and DCC (DSN 2008). 2008. 187−196. [doi: 10.1109/DSN.2008.4630087]

[27] Huang Y, Ghosh AK. Introducing diversity and uncertainty to create moving attack surfaces for Web services. In: Proc. of the Moving Target Defense. New York: Springer-Verlag, 2011. 131−159. [doi: 10.1007/978-1-4614-0977-9_8]

[28] Okhravi H, Rabe MA, Mayberry TJ, Leonard WG, Hobson TR, Bigelow D, Streilein WW. Survey of cyber moving targets. Technical Report, No. MIT/LL-TR-1166, Massachusetts Inst of Technology Lexington Lincoln Laboratory, 2013.

[29] Wang ZY, Yang XJ, Zhou Y. Scalable triple modular redundancy fault tolerance mechanism for MPI-oriented large scale parallel computing. Ruan Jian Xue Bao/Journal of Software, 2012,23(4):1022−1035 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4011.htm [doi: 10.3724/SP.J.1001.2012.04011]

[30] Wang YN, Wu HR, Huang F. Optimization analysis and research of high concurrency Web application system performance. Computer Engineering and Design, 2014,35(8):2976−2980 (in Chinese with English abstract).

附中文参考文献:

[1] 中国互联网站发展状况及其安全报告(2016).2016. http://tech.163.com/16/0320/15/BIK212JA00094P25.html

[4] 中国国家信息安全漏洞库.2015. http://www.cnnvd.org.cn/vulnerability/statistics

[20] 于佳,程相国,李发根,潘振宽,孔凡玉,郝蓉.标准模型下可证明安全的入侵容忍公钥加密方案.软件学报,2013,24(2):266−278. http://www.jos.org.cn/1000-9825/4324.htm [doi: 10.3724/SP.J.1001.2013.04324]

[21] 张晓玉,李振邦.移动目标防御技术综述.通信技术,2013,46(6):111−113.

[29] 王之元,杨学军,周云.大规模MPI并行计算的可扩展三模冗余容错机制.软件学报,2012,23(4):1022−1035. http://www.jos.org.cn/1000-9825/4011.htm [doi: 10.3724/SP.J.1001.2012.04011]

[30] 王亚楠,吴华瑞,黄锋.高并发Web应用系统的性能优化分析与研究.计算机工程与设计,2014,35(8):2976−2980.

仝青(1992－),女,河南郑州人,学士,主要研究领域为网络空间安全.

张为华(1974－),男,博士,副教授,CCF 专业会员,主要研究领域为计算机体系结构,软件纠错,编译器优化.

张铮(1976－),男,博士,副教授,主要研究领域为网络空间安全.

邬江兴(1953－),男,教授,博士生导师,主要研究领域为信息通信网络,网络空间安全.