计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140608
52(8):1873-1882,2015

面向云环境内部 DDoS 攻击检测的博弈论优化

王一川[1,2] 马建峰[2] 卢 笛[2] 张留美[2] 孟宪佳[2]
[1](西安理工大学计算机科学与工程学院 西安 710048)
[2](西安电子科技大学计算机学院 西安 710071)
(ctechsky@gmail.com)

Game Optimization for Internal DDoS Attack Detection in Cloud Computing

Wang Yichuan[1,2], Ma Jianfeng[2], Lu Di[2], Zhang Liumei[2], and Meng Xianjia[2]
[1](*Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an* 710048)
[2](*School of Computer Science and Technology, Xidian University, Xi'an* 710071)

Abstract A collaborative intrusion detection system (IDS) model, entitled virtual machine introspection & network-based IDS (VMI-N-IDS) is proposed, which is based on traditional introspection-based IDS and network-based IDS, for the defense of internal distributed denial of service (DDoS) attack threat of cloud cluster (e.g. cloud droplets freezing, CDF Attack). The CDF attack can exhaust the internal bandwidth of the cluster, the CPU and the memory resources of physical servers. Based on the game theory, IDS and attacker are treated as the two game parties in the VMI-N-IDS model. Utility functions of the two parties are supported, and it is proved that the game model is a non-cooperative and repeated game of incomplete information, and the subgame perfect Nash equilibrium is existent. Finally, the optimal defense strategy is proposed, which is the tradeoff between the false alarm rate and the malicious software size control, for solving the problem of dynamical adjustment strategy of internal intrude detection. The best strategy for the stages of IDS is to increase the threshold value β when the mathematical expectation of the suspicious value is greater than the load of server resources, and to reduce such value conversely. Experimental result shows that the proposed method can effectively defense the internal DDoS attack threat in the cloud environment.

Key words cloud computing; network security; intrusion detection; DDoS attack; game theory

摘 要 结合传统基于虚拟机内省(virtual machine introspection-based, VMI)和基于网络(network-based)入侵检测系统(intrusion detection system, IDS)的特点,提出一种部署在云服务器集群内部的协同入侵检测系统(virtual machine introspection & network-based IDS, VMI-N-IDS)来抵御云环境内部分布式拒绝服务攻击(distributed denial of service, DDoS)攻击威胁,比如"云滴冻结"攻击.将入侵检测系统和攻击者看作是博弈的双方,提出一种针对云服务器集群内部 DDoS 攻击与检测的博弈论模型;分别给出博弈双方的效用函数,并证明了该模型子博弈精炼纳什均衡;给出了权衡误报率和恶意软件规模控制的最佳防御策略,解决了动态调整云环境内部入侵检测策略的问题.实验表明,VMI-N-IDS 能够有效抵御云环境内部 DDoS 攻击威胁.

关键词 云计算;网络安全;入侵检测;DDoS 攻击;博弈论

中图法分类号 TP309

收稿日期:2014-07-07;**修回日期**:2014-10-22

基金项目:国家科技重大专项基金项目(2012ZX03002003);中央高校基本科研业务费专项资金项目(JY10000903001);西安理工大学博士启动金基金项目(112-256081504)

近几年,各种云计算(cloud computing)技术的不断涌现和推广为用户使用互联网资源提供了极大的便利[1].云计算的思想是采用一种普适的、便捷的、按需的方式,可配置地组织和共享各种资源(包括网络、服务器、存储、应用等)[2].其中,"多租户"(multi-tenancy technology)是云计算技术的重要特征.以虚拟机为代表的虚拟化(virtualization)技术成为了云环境下的关键技术之一.虚拟机的数据隔离、快照和动态迁移等相关技术确实在用户隐私保护、服务可生存性等方面发挥出了较强的优势,然而对某些网络攻击却也提供了更好的温床[3].

攻击者可在非常短的时间内,使用少量投资租用云服务商提供的大量虚拟机来实施攻击.以 Amazon (elastic compute cloud, EC2)为例,租用小型的 Linux/Unix 服务每小时仅需要 0.06 美元,租金最高的 8 倍超大型 Windows 服务每小时仅需要 4.931 美元.如果攻击者直接以云服务器集群为攻击目标,事情将变得更加糟糕.Arbor Networks 公司第 8 份世界基础设施年度安全报告指出,云计算服务和数据中心正在遭受越来越多的网络攻击[4].

本文作者先前的工作在文献[5]中描述了一种针对典型云环境内部的、实用的、新型的分布式拒绝服务攻击 DDoS 攻击——"云滴冻结"(cloud droplets freezing, CDF)攻击.通过攻击实验验证了该攻击不仅可以对云服务器集群的带宽产生严重的拥塞效果,而且能够极大地消耗物理主机的内存和 CPU 等资源.这使得本来能够分配给合法虚拟机的资源被非法占用,从而达到拒绝服务攻击的效果.通过分析攻击原理可知,传统的拒绝服务攻击防御系统和技术不能对该攻击产生较好的防御效果.

在抵御 CDF 攻击方法中,一个直接的思路是在 CDF 攻击刚刚发生,甚至是发生前就终止僵尸虚拟机的运行.那么,这就要求每一台物理服务器上存在一个部署在虚拟机监控器(virtual machine monitor, VMM)中的虚拟机入侵检测系统,该系统能够准确且迅速地分析并判断 VMM 上运行的虚拟机是否是僵尸虚拟机.该模块实时地运行在监控器中,对虚拟机之间、虚拟机与虚拟机监控器之间以及虚拟机与云用户之间网络通信的数据流进行分析,并以此分析甄别各个虚拟机之间的行为是否存在安全威胁.然而,由于该系统对虚拟机行为的分析依赖于对其虚拟网卡的数据包分析和资源使用状况的动态侦测,这种分析的准确性在很大程度上取决于现有虚拟机行为特征库的完备性.而事实上,文献[6]指出,目前没有任何一种入侵检测/保护系统(intrusion detection/prevention systems, IDS/IPS)能侦测出云平台下所有恶意的软件和服务.因此,研究如何在控制误报率的前提下尽可能地限制云集群中恶意软件的规模具有重要意义.

1 "云滴冻结"(CDF)攻击[5]

为了详细分析本文模型和工作,有必要对 CDF 攻击进行简要介绍.

1.1 网络模型

在典型云服务器集群部署拓扑环境中(如图 1

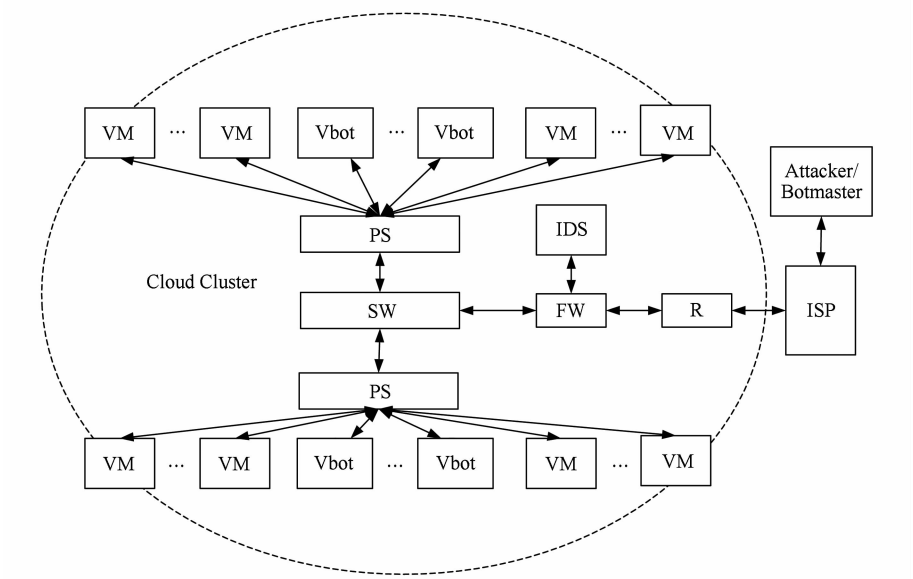


Fig. 1 Network model.

图 1 网络模型

所示),云服务器集群(cloud cluster)中存在 P 台物理服务器(physical server, PS),每台物理服务器上通过虚拟机监控器(VMM)运行和控制若干台虚拟机(virtual machine, VM)服务器,这些虚拟机服务器对因特网用户提供相应的云计算服务.其中某些虚拟机被僵尸网络所控制而成为僵尸虚拟机(virtual robot, Vbot).物理服务器通过云集群的核心交换机(switch, SW)相连.云集群通过防火墙(firewall, FW)和路由器(router, R)连接因特网服务提供商(Internet service provider, ISP)并为因特网用户提供云服务.防火墙中部署有 IDS/IPS,负责对服务器集群与因特网的网络连接进行分析、过滤和保护.僵尸网络控制者(botmaster)作为攻击者(attacker)通过因特网连接至云服务器集群,并对受控的僵尸虚拟机发布控制命令.

1.2 攻击目标和攻击方法

传统拒绝服务攻击(denial of service, DoS)攻击的目标是通过外部连接请求来耗尽集群核心交换机与路由器之间的网络带宽资源.然而,入侵检测系统一旦发现来自服务器集群外部的恶意连接则会更新防火墙策略,从而断开该恶意连接.随着入侵检测技术的不断升级,传统的 DoS 攻击效果受到很大限制.

CDF 攻击的目标和思路与传统 DoS 有很大差别,其攻击思路是消耗云服务器集群内部的物理服务器的计算资源和集群内部通信链路的带宽资源,从而降低甚至"冻结"物理服务器处理合法用户服务的能力,提高物理服务器与核心交换机的通信时延,最终达到拒绝服务攻击的效果.其攻击目标可以分为 3 种:1)云服务器集群中以核心交换机为中心的网络核心带宽资源;2)云服务器集群中僵尸虚拟机所在物理服务器的 CPU 资源;3)云服务器集群中僵尸虚拟机所在物理服务器的内存资源.

CDF 攻击者利用服务器集群中受控的僵尸虚拟机,并通过在僵尸虚拟机之间相互发送网络消息来拥塞网络和过载物理服务器的 CPU 负载.发动 CDF 攻击需要以下 3 个步骤:

Step1. 攻击者选择云服务器集群 P 作为攻击目标;

Step2. 在目标云服务器中租赁或者感染 N 台虚拟机作为僵尸虚拟机;

Step3. N 台僵尸虚拟机中两两互相发送网络数据包 $\{S_{Vbots[i]}, D_{Vbots[j]}, T, M \mid i, j < N, i \neq j\}$,以达到过载云服务器集群资源的目的.$Vbots$ 表示僵尸虚拟机的集合,S/D 表示网络消息的源/目的地址,T 表示网络协议,M 表示网络消息.网络消息可以是未加密的,但本文建议采用加密形式,因为这样可以尽可能地消耗物理机的 CPU 资源.

1.3 攻击原理

假设在云服务集群中共存在 N 台僵尸虚拟机,其分布并运行在 P 台物理服务器上.由于 CDF 攻击对于任意一条边建立独立的 socket 连接,因此僵尸虚拟机节点间形成有向完全图的通信连接.即在这 N 个虚拟机间存在 $N(N-1)$ 条通信连接.但是这些通信连接并不都需要占用物理链路的带宽.比如,运行在同一台物理服务器上的虚拟机之间可以通过共享内存(shared-memory communication channel)等方式来完成通信,并不需要占用真实物理链路带宽[7].因此,假设第 i 台物理服务器上存在 N_i 台僵尸虚拟机,则通过该物理服务器与核心交换机之间的物理链路的通信连接数为

$$C_i = 2N_i(N - N_i).$$

假设每条通信连接占用带宽资源约为 $\alpha > 0$,则每条物理链路被 CDF 攻击占用的资源为 $B_i = C_i\alpha$.该云服务器集群中内部网络带宽资源被 CDF 攻击占用总量为

$$B = \frac{1}{2}\alpha\sum_{i=1}^{P} C_i = \alpha\sum_{i=1}^{P}(N_iN - N_i^2),$$

其中,$N, P \geqslant 1$,$N, P \in \mathbb{N}$.为了提高虚拟机的执行效率,绝大多数现有的虚拟机监控器都采用分离式设备驱动模型[8],如 Xen,VirtualBox.在这种通信模型下,所有的数据通信路径都由于前端驱动(frontend driver)和后端驱动(backend driver)的信息交互而被延长了.对于一个全虚拟化的虚拟机而言,当应用程序要进行一次网络 I/O 时,为了读写网卡寄存器,需要首先陷入到虚拟机监控器,接着切换到虚拟机管理模块去读/写由快速模拟器(quick emulator, QEMU)模拟的寄存器.被延长的不仅仅是数据请求的路径,中断处理的路径也被延长了.在每一页大小的数据包传输的过程中,虚拟机都要发生一次软中断,而所有的虚拟中断必须由 QEMU 来处理.因此,物理机的软中断次数和 CPU 的软中断处理时间被增加了.假设第 i 台物理服务器上存在 N_i 台僵尸虚拟机,则通过该物理服务器 VMM 的通信连接数为

$$E_i = 2N_i(N - N_i) + N_i(N_i - 1).$$

假设每条通信连接占用 CPU 时间资源约为 $\xi > 0$,则第 i 台物理服务器被 CDF 攻击所占用的 CPU 资源为 $S_i = E_i\xi$.该云服务器集群内部所有物理服务器被 CDF 攻击占用的 CPU 资源为

$$S = (N^2 - N)\xi.$$

CDF 攻击对物理服务器内存消耗的攻击效果取决于物理服务器分配给僵尸虚拟机的内存所占物理机总内存的比例.僵尸虚拟机数目跟其物理服务器内存的消耗呈线性关系.

CDF 攻击对云平台最显著的资源消耗是集群内部的网络带宽.因此,本文特别地针对类似 CDF 攻击的云环境内部 DDoS 攻击这一特点提出云服务器集群内部的协同入侵检测系统(virtual machine introspection & network-based IDS, VMI-N-IDS),以及提高检测效率和降低误报率为目的的动态检测策略博弈论优化模型.

2 云环境内部入侵检测的博弈论优化模型

2.1 协同入侵检测系统(VMI-N-IDS)

根据 1.2 节和 1.3 节的分析,传统部署在云服务集群防火墙处的 IDS/IPS 是不能抵御 CDF 攻击的,因为只有将入侵检测系统部署于监控器当中才能捕捉到上述这些数据流,从而发现和限制僵尸虚拟机的运行.

基于虚拟机内省(virtual machine introspection-based, VMI)的入侵检测系统(VMI-IDS)是基于虚拟机监控器入侵检测系统的一个实例[9-10].IBM 研究中心正在推行这种虚拟机内省的方法来创建一个分层的安全服务集合,可以从云集群物理服务器内部保护运行在其上的多台虚拟机.VMI-IDS 不同于传统的基于主机的入侵检测系统(host-based IDS, HIDS),其位于虚拟机监控器的最底层,或者是硬件驱动与原来虚拟机监控器之间的一层.也正是由于 VMI-IDS 工作的位置更加底层,所以它能够实时地、直接地检测虚拟机各个虚拟硬件、事件和软件的工作和映射状态.此外,VMI-IDS 还能更方便地监控 VMM 本身的工作状态.因此,VMI-IDS 对虚拟机和物理服务器的运行状态能够实现较全面的监控和较好的工作鲁棒性.然而,这种方法面临的挑战有:1)实现机制复杂且缺乏实践验证[11];2)操作系统和虚拟机监控器的代码或执行库可能会被破坏或篡改[9];3)缺乏对多虚拟机之间恶意行为关联性的整体分析.此外,如何设计恶意服务和僵尸虚拟机的判断策略仍然是亟待研究的问题.

基于网络数据分析的入侵检测系统(network-based IDS,NIDS)具有较强的实时检测机制,其通过将当前探测到的网络行为与行为特征库中进行比较来判断当前网络行为是否为入侵行为.NIDS 通常的检测手法是分析网络层和传输层的每一个数据包的包头来发现入侵活动.一旦发现网络数据包的签名错误或者存在异常的网络数据特征,则判断存在入侵行为.因此,NIDS 通常被部署在服务器集群与外部网络经常发生直接交互的服务器上,以便检测虚拟机以及虚拟机监控器的恶意网络行为.

文献[12]提出了与虚拟机兼容的 NIDS 架构.其主要包含 2 个组件:入侵检测系统管理单元(IDS management unit)和入侵检测传感器(IDS sensor).入侵检测系统管理单元的主要模块包括:事件采集器、事件数据库、数据组件分析和远程控制.其中事件采集器收集被入侵检测传感器识别的恶意行为并将其存储到事件数据库中.这个方法用于在多租户环境下保护虚拟机以免被恶意破坏;但其缺点是需要在集群中部署多个入侵检测系统实例.在文献[13]提出的方法中,入侵检测系统部署的目的在于检测虚拟机与数据库交互的日志.通过分析该日志并实时验证数据签名,该入侵检测系统可以分析出已知的入侵攻击.一旦发现攻击行为,入侵检测系统会通知该攻击虚拟机的监控器和物理服务器,从而丢弃该攻击数据包并撤销对数据库的操作.如果攻击类型是 DDoS 攻击,物理服务器和虚拟机监控器会切断该攻击虚拟机的一切网络通信.同时,入侵检测系统会检查与该虚拟机进行通信的其他虚拟机.此外,数据中心和路由表等设备会立即进行相应的更新操作;云服务器集群的防火墙也会切断一切与攻击虚拟机相关的网络连接.这种方法确实可以有效地发现虚拟机与数据中心进行交互过程中的入侵行为,但如果分析虚拟机之间经过加密的通信则是非常困难的事情,因此仅仅采用这一种方法防御 CDF 攻击仍然是不够的.

综上所述,VMI-IDS 可以有效地分析和监控虚拟机对系统设备和资源的使用情况;而 NIDS 可以通过分析虚拟机的网络通信来判断入侵活动,同时通过监控虚拟机之间的通信可以判断集群中虚拟机行为的相关性.因此,研究一种部署在云服务集群内部的、结合 VMI-IDS 和 NIDS 优点的协同入侵检测系统(VMI-N-IDS)对抵御 CDF 攻击具有重要的意义.VMI-N-IDS 的拓扑结构如图 2 所示.

入侵检测传感器部署于每台 PS 的 VMM 中.每个入侵检测传感器负责监控与其对应的一台 VM 的行为和运行状态,并实时地将这些信息发送给入侵检测管理单元.虚拟机行为分析模块(VM behavior analysis module)对入侵检测传感器收集的 VM 信息进行分析,然后将其特征值与虚拟机行为特征库(VM behavioral characteristics database)

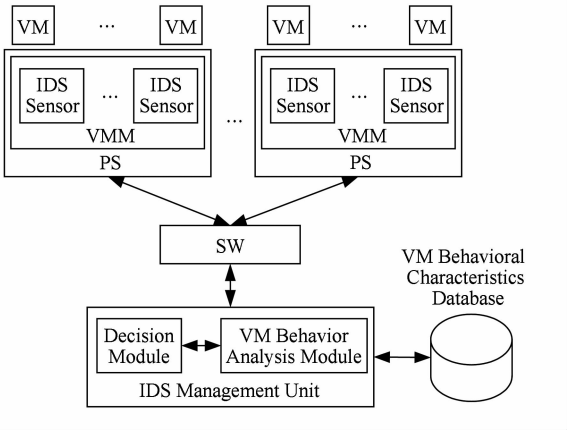


Fig. 2 VMI-N-IDS architecture.

图 2 VMI-N-IDS 结构

中数据进行比对,然后将虚拟机行为的可疑程度返回给决策模块(decision module).最终,决策模块对虚拟机行为给出决策结果,并通知入侵检测传感器终止或者忽略对该 VM 的行为操作.

本文将 VMI-N-IDS 看作是防御者,并与 CDF 攻击者看作是博弈的双方,提出一种针对云服务器集群内部恶意软件攻击与检测的博弈论模型.本文分别给出博弈双方的效用函数来评估其攻防过程中的成本和收益,以此来分析和权衡误报率及恶意软件规模的最佳策略.下面,本文分别从防御者和攻击者 2 个方面对其进行分析.

2.2 防御者

对于防御者而言,入侵检测管理单元通过部署在各个物理服务器 VMM 中的入侵检测传感器,得知每一台虚拟机的行为和运行状态.由于该系统对虚拟机行为的分析依赖于对其虚拟网卡的数据包分析和动态地资源使用状况的侦测.这种分析的准确性在很大程度上取决于现有虚拟机行为特征库的完备性.

实际上,正如前文所言,防御者并不能完全准确地确定某一台虚拟机的行为是否是恶意的.针对 CDF 攻击而言,虚拟机的行为主要体现是网络数据连接的状态.因此,防御者只能根据虚拟机行为特征库来对当前虚拟机网络数据包的特征进行判断其网络行为的可疑度.下面给出可疑度 π 的定义.

定义 1. 可疑度(suspicious).一个网络连接的可疑度 π 定义为,防御者主观认为的该网络连接是一个恶意的攻击行为的概率.

简单而言,决策模块得到各个虚拟机行为的可疑程度后与门限值 β 比较来判断是否断开该网络连接.然而虚拟机行为特征库并不完备,存在漏报和误报的可能.因此,本文定义恶意度 γ 的概念来描述某条连接是否真的具备恶意攻击目标服务器的可能.

定义 2. 恶意度(malicious intent).一个网络连接的恶意度 γ 定义为该连接具备恶意攻击目标服务器的概率.显然地,这是一个客观的度量指标,是一个关于入侵检测管理单元识别度 γ 和网络连接可疑度 π 的函数,表示为 $\gamma(\gamma,\pi)$.

当 $\gamma=0$,则所有网络连接的恶意度 $\gamma=\frac{1}{2}$.这表示入侵检测管理单元对于网络连接是否是恶意的识别能力为 0.因此,此时虚拟机行为分析模块判断出的某个网络连接 X 的可疑度 π_X 其实没能给出指导意见.当 $\gamma\to1$,则网络连接 X 的恶意度 $\gamma_X\to\pi_X$.这表示当前虚拟机行为分析模块的识别能力非常强,其主观判断的可疑度值跟实际上该网络连接的恶意度接近.当然,如何通过神经网络、机器学习等手段进行自学习来提高 γ 的值不在本文讨论范围之内,如感兴趣请自行查阅相关文献和资料.为了便于分析,本文简单假设 $\gamma(\gamma,\pi)$是一个依赖于 γ 的线性函数,其可以表示为

$$\gamma(\gamma,\pi)=(\pi-\frac{1}{2})\gamma+\frac{1}{2}.$$

因此,一个网络连接不是一个攻击连接的概率可以表示为概率函数 $1-\gamma(\gamma,\pi)$.那么,对于任意网络连接而言,防御者的判断与操作分布如表 1 所示:

Table 1 Judgment and Operation Distribution of Defender

表 1 防御者的判断与操作分布表

Judgment	Operation		
	Retention	Disconnection	Marginal Probability
Attack Connection	P_{AR}	P_{AD}	$\gamma(\gamma,\pi)$
Legal Connection	P_{LR}	P_{LD}	$1-\gamma(\gamma,\pi)$
Marginal Probability	$P\{\pi<\beta\}$	$P\{\pi\geqslant\beta\}$	

由于变量 γ 和 β 是互相独立的,因此本文得到其联合分布:

$$\begin{cases}P_{AR}=P\{\pi<\beta\}\left(\left(\pi-\frac{1}{2}\right)\gamma+\frac{1}{2}\right);\\ P_{AD}=P\{\pi\geqslant\beta\}\left(\left(\pi-\frac{1}{2}\right)\gamma+\frac{1}{2}\right);\\ P_{LR}=P\{\pi<\beta\}\left(\frac{1}{2}-\left(\pi-\frac{1}{2}\right)\gamma\right);\\ P_{LD}=P\{\pi\geqslant\beta\}\left(\frac{1}{2}-\left(\pi-\frac{1}{2}\right)\gamma\right).\end{cases}$$

对于网络连接 i,防御者防御操作的数学期望为

$$E_{\mathrm{D}}^{i}=P_{\mathrm{AR}}^{i}W_{\mathrm{AR}}+P_{\mathrm{AD}}^{i}W_{\mathrm{AD}}+P_{\mathrm{LR}}^{i}W_{\mathrm{LR}}+P_{\mathrm{LD}}^{i}W_{\mathrm{LD}}.$$

不妨假设攻击连接和正常的网络连接请求消耗类似的网络资源. $\mathcal{R}$ 表示云服务器集群网络资源总数,$\overline{R}_{\mathrm{C}}$ 表示平均每个连接的网络资源消耗. 那么,$N_{\mathrm{R}}=\dfrac{\mathcal{R}}{\overline{R}_{\mathrm{C}}}$表示云服务器集群所能容纳的最大网络连接数目. 当前的网络连接数目如果用 N_{C} 表示,则需要满足 $N_{\mathrm{C}}<N_{\mathrm{R}}$. 这样,本文得到防御者的效用函数:

$$U_{\mathrm{D}}=\frac{N_{\mathrm{C}}}{N_{\mathrm{R}}}W_{\mathrm{O}}+\left(1-\frac{N_{\mathrm{C}}}{N_{\mathrm{R}}}\right)W_{\mathrm{D}}\sum_{i=1}^{N_{\mathrm{C}}}E_{\mathrm{D}}^{i},\ N_{\mathrm{C}}\leqslant N_{\mathrm{R}}.$$

W_{O} 表示云服务器集群网络资源被完全耗尽的代价权重;W_{D} 表示云服务器集群网络资源没有被完全消耗的情况下,保留 N_{C} 个虚拟机的网络连接所带来收益的权重.

2.3 攻击者

攻击者希望通过租用(或者感染)尽可能少的僵尸虚拟机来发动有效的 CDF 攻击. 攻击者不确定入侵检测管理单元对当前攻击的抵御能力. 因此,其发起的攻击连接被断开的越多,说明防御者对当前攻击方式的防御能力越强,反之则越差. 本文假设 N_A 表示当前仍然被目标服务器防火墙保持的攻击连接所发起的僵尸虚拟机数目. N_t 表示从攻击开始到时刻 t 已经发送过攻击连接到目标服务器的僵尸虚拟机总的数目,显然,这是一个关于时间 t、以最小值为 0、最大值为当前攻击者所控制当前云服务器集群内部僵尸虚拟机总数 N_{B} 的非递减函数.

在某一时刻,攻击者可以采取的策略可分为是否继续增加僵尸虚拟机来发送攻击数据包. 显然,对于已经参与过攻击的僵尸虚拟机就已经暴露给防御者了,无论其在下次攻击中是否参与. 因此,攻击者的最佳策略就是参与过攻击的僵尸虚拟机继续发送攻击数据包直到整个攻击结束. 为了便于分析,本文将任一僵尸虚拟机开始发送攻击数据包到目标服务器的时间定义为该节点的门限时刻 T. 即对于某一僵尸虚拟机而言,设攻击者下达攻击命令后开始计时时刻 t,如果 $t\geqslant T$ 则该僵尸虚拟机开始发送攻击数据包;如果 $t<T$ 则该僵尸虚拟机保持沉默状态,不发送攻击数据包.

因此,不妨设函数 N_T 为攻击者所采取攻击策略的序列,$\dfrac{N_{\mathrm{A}}}{N_T}$表示入侵检测管理单元保留攻击连接的比例,那么 $1-\dfrac{N_{\mathrm{A}}}{N_T}$表示在时刻 T 如果继续按照目前的攻击方式进行攻击入侵检测管理单元断开某攻击连接的概率.

那么,对于所有僵尸虚拟机的任一个攻击连接,有攻击者的判断与操作分布如表 2 所示:

Table 2 Judgment and Operation Distribution of Attacker

表 2 攻击者的判断与操作分布表

Judgment	Operation		
	Retention	Disconnection	Marginal Probability
Defended	P_{DA}	P_{DH}	$1-N_{\mathrm{A}}/N_T$
Not Defended	P_{NA}	P_{NH}	N_{A}/N_T
Marginal Probability	$P\{t\geqslant T\}$	$P\{t<T\}$	

由于变量 N_{A} 和 T 是相互独立的,因此本文假设其概率分布也相互独立. 这样本文得到:

$$\begin{cases}P_{\mathrm{DA}}=P\{t\geqslant T\}\left(1-\dfrac{N_{\mathrm{A}}}{N_T}\right);\\ P_{\mathrm{DH}}=P\{t<T\}\left(1-\dfrac{N_{\mathrm{A}}}{N_T}\right);\\ P_{\mathrm{NA}}=P\{t\geqslant T\}\left(\dfrac{N_{\mathrm{A}}}{N_T}\right);\\ P_{\mathrm{NH}}=P\{t<T\}\left(\dfrac{N_{\mathrm{A}}}{N_T}\right).\end{cases}$$

设攻击者第 i 个攻击连接的攻击策略期望为 E_{A}^{i},则:

$$E_{\mathrm{A}}^{i}=P_{\mathrm{DA}}^{i}W_{\mathrm{DA}}+P_{\mathrm{DH}}^{i}W_{\mathrm{DH}}+P_{\mathrm{NA}}^{i}W_{\mathrm{NA}}+P_{\mathrm{NH}}^{i}W_{\mathrm{NH}}.$$

那么,攻击者的效用函数为

$$U_{\mathrm{A}}=\frac{N_T}{N_{\mathrm{B}}}W_{\mathrm{E}}+\left(1-\frac{N_T}{N_{\mathrm{B}}}\right)W_{\mathrm{A}}\sum_{i=1}^{N_{\mathrm{A}}}E_{\mathrm{A}}^{i}+\frac{N_{\mathrm{C}}}{N_{\mathrm{R}}}W_{\mathrm{S}},\ N_T\leqslant N_{\mathrm{B}}.$$

设攻击者所有僵尸虚拟机都暴露的代价权重为 W_{E};W_{A} 表示未使用的僵尸虚拟机给攻击者带来的收益权重;W_{S} 表示攻击者成功实现了 DDoS 攻击的收益权重.

3 模型分析

3.1 效用函数分析

考虑当 $W_{\mathrm{AR}}=W_{\mathrm{LD}}=-1$,并且 $W_{\mathrm{AD}}=W_{\mathrm{LR}}=1$ 时的场景. 图 3 显示了防御者的效用函数 U_{D} 在 π 不同分布时的变化.

实验 1. 参数设置为:$N_{\mathrm{R}}=10\ 000$,$\gamma=0.5$,$W_{\mathrm{O}}=-100$,$U_{\mathrm{D}}(Load,\beta)$,当 $W_{\mathrm{D}}=1$. 对于 $E(\pi)$ 的值,本文首先选择 $\pi_1'\sim N(30,1)$ 和 $\pi_2'\sim\mathcal{N}(80,1)$ 作为实际环境下的近似值. 同时,令 $\dfrac{\pi_1'}{100}$ 和 $\dfrac{\pi_2'}{100}$ 分别作为 π_1 和 π_2 的分布函数.

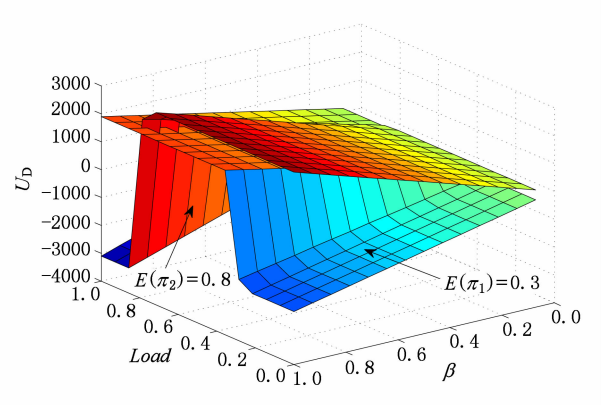


Fig. 3 Influence of π towards utility function of defender.

图 3 π 对防御者效用函数的影响

令服务器的网络资源负载表示为 $Load=\frac{N_C}{N_R}$,则这个例子表明,当 $Load \geqslant E(\pi)$ 时应当减少门限值 β. 这与实际是相符合的,如果服务器网络资源的负载很重或者当前接入服务器的网络连接请求的可疑度普遍很高,为了保证更倾向于正常网络通信的连接畅通和较高的服务级别用户的服务质量(quality of service, QoS),适当减少门限值以对现有网络连接进行严格筛选是很有必要的. 反之当 $Load < E(\pi)$,则应当适当增大 β. 这与实际也是相符合的,如果服务器资源的负载很轻或者当前接入服务器的网络连接请求的可疑度普遍比较低,则为了保证更多合法用户的接入,可以适当放宽对现有网络连接的筛选要求.

实验 2. 参数设置为 $N_R=10\,000$, $E(\pi)=0.6$, $W_O=-100$, $W_D=1$, $U_D(Load,\beta)$,当 γ 分别为 0.2 和 0.7. 对于 $E(\pi)$ 数值的生成方法与实验 1 类似. 图 4 显示了 γ 的改变对防御者效用函数的影响. 不难看出,γ 值越大,则 β 的改变策略对防御者效用函数的影响越明显.

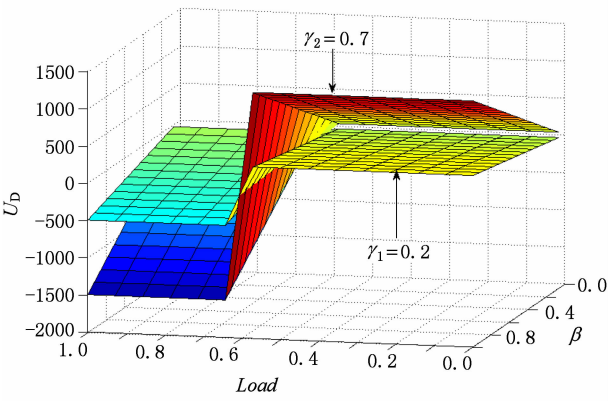


Fig. 4 Influence of γ towards utility function of defender.

图 4 γ 对防御者效用函数的影响

实验 3. 针对攻击者,考虑场景 $W_{DA}=W_{NH}=-1$ 且 $W_{DH}=W_{NA}=1$. 图 5 显示了 N_B 的不同分布对攻击者效用函数的影响. 实验参数设置为 $\lambda=0.3$, $N_R=10\,000$, $W_E=-100$, $W_A=1$, $W_S=100$, $Load=0.7$. N_B 的取值分别为 100 和 1 000. $\psi=\frac{N_T}{N_B}$, $\lambda=\frac{N_A}{N_T}$.

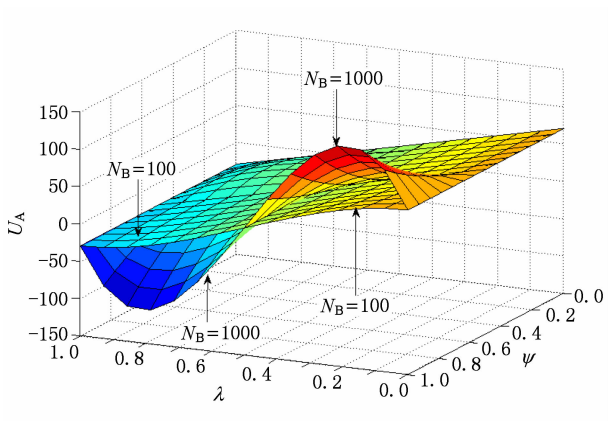


Fig. 5 Influence of N_B towards utility function of attacker.

图 5 N_B 对攻击者效用函数的影响

这个例子表明,当参与攻击的僵尸虚拟机超过攻击者所控制的僵尸虚拟机数目的一半时,即 $\lambda \geqslant \frac{1}{2}$,则攻击者应当减少攻击的强度 ψ,即减少新加入参与攻击的僵尸虚拟机的数目. 因为这样能减少僵尸网络中所有主机完全暴露的可能. 反之,如果参与攻击的僵尸虚拟机不足攻击者所控制的僵尸虚拟机数目的一半时,即 $\lambda < \frac{1}{2}$,为了加快 DDoS 攻击的效果,攻击者则应增加 ψ,即适当增加新加入参与攻击的僵尸虚拟机的数目.

3.2 动态博弈分析

3.1 节分别对防御者和攻击者的效用函数以及策略选择进行了分析,但实际上双方的博弈过程是重复博弈. 现在本文将双方的博弈过程进行动态的整体分析. 表 3 显示了双方的阶段博弈策略矩阵.

Table 3 Staged Game Strategy Matrix Between Defender and Attacker

表 3 防御者和攻击者之间的阶段博弈策略矩阵

Attacker Strategies	Defender Strategies	
	Increasing β	Decreasing β
Increasing the Number of New Vbots	$\boldsymbol{G}_1$	$\boldsymbol{G}_2$
Decreasing the Number of New Vbots	$\boldsymbol{G}_3$	$\boldsymbol{G}_4$

通过 2.2 节的分析可知,防御者可用的策略是增大或缩小 β 的值;攻击者可用的策略是增加或减少新攻击连接数目. 图 3 中"$\boldsymbol{G}_1$","$\boldsymbol{G}_2$","$\boldsymbol{G}_3$","$\boldsymbol{G}_4$"分别表示双方选择不同策略是所对应的场景联合策略二元组(X,Y).

定理 1. 纳什均衡(Nash equilibrium)在阶段博弈中存在且唯一. 如果防御者和攻击者都是理性的,那么在双方的阶段博弈策略矩阵中存在唯一的纳什均衡点 $\boldsymbol{s}^*(X^*,Y^*)\in\{\boldsymbol{G}_1,\boldsymbol{G}_2,\boldsymbol{G}_3,\boldsymbol{G}_4\}$.

证明. 对于防御者而言,在给定的时刻 T,被断开的攻击连接的数目与总攻击连接数目的比值 λ 是未知的. 因此,防御者也不能确定攻击者是否增加或者减少了新攻击连接的数目. 那么,防御者只能根据自身所知道的服务器的网络负载情况 $Load$ 以及自身主观检测到的 π 的数学期望 $E(\pi)$来选择最佳策略以调整增大或者减小 β 的值. 如前文所述,如果 $E(\pi)>Load$,则增大 β,反之减小 β. 类似地,对于攻击者而言,在给定的时刻 T,可疑度 π 的值是未知的. 这导致其数学期望 $E(\pi)$也是未知的. 因此,攻击者很难确定防御者是否增大或减小了门限值 β. 那么,攻击者也只能选择基于已知的 λ 值来选择自己的最佳策略. 正如本节所述,如果$\lambda<\frac{1}{2}$则增加新攻击连接数目,反之减少新攻击连接数目.

那么,在给定的时刻 T,防御者和攻击者分别根据 $E(\pi)$和 λ 来决定自己的策略. 最终结果会是上述阶段博弈策略矩阵表 3 中"$\boldsymbol{G}_1$","$\boldsymbol{G}_2$","$\boldsymbol{G}_3$","$\boldsymbol{G}_4$"中的 1 个. 此时,该阶段博弈达到纳什均衡点. 本文用 $\boldsymbol{s}^*(X^*,Y^*)$来表示该纳什均衡点,并且满足:

$$U_{\mathrm{A}}(\lambda,X^*)\geqslant U_{\mathrm{A}}(\lambda,X),$$

其中,X 表示是否增加新攻击连接数目,满足:

$$U_{\mathrm{D}}(E(\pi),Y^*)\geqslant U_{\mathrm{D}}(E(\pi),Y),$$

Y 表示是否增大或减少 β 的值. 博弈双方随后的阶段博弈过程可以参考该阶段博弈的结果. 防御者能够基于新的 β 值来得到策略矩阵中所对应的策略 y^* 的取值,从而决定保留或者断开每一个网络连接;攻击者能够基于策略 X^* 来得到策略 x^* 的取值,进而调整新攻击连接数目.

又有:

$$U_{\mathrm{A}}(\lambda,x_i^*)\geqslant U_{\mathrm{A}}(\lambda,x_i),1\leqslant i\leqslant N_{\mathrm{C}},\forall x_i;$$

$$U_{\mathrm{D}}(E(\pi),y_j^*)\geqslant U_{\mathrm{D}}(E(\pi),y_j),1\leqslant j\leqslant N_{\mathrm{B}},\forall y_j.$$

在每一次阶段博弈中,如果防御者和攻击者都是理性的,他们总是选择使得自身效用函数达到最大的策略. $(E(\pi)>Load)\cap(E(\pi)\leqslant Load)=\varnothing$且$\left(\lambda<\frac{1}{2}\right)\cap\left(\lambda\geqslant\frac{1}{2}\right)=\varnothing$,因此,在每一次阶段博弈过程中,"$\boldsymbol{G}_1$","$\boldsymbol{G}_2$","$\boldsymbol{G}_3$","$\boldsymbol{G}_4$"这 4 种情况有且只有 1 种情况会被选中. 因此在防御者和攻击者之间的阶段博弈策略矩阵中存在唯一的纳什均衡点.

证毕.

引理 1. 如果防御者和攻击者都是理性的,那么在双方的博弈模型中,其子博弈精炼纳什均衡存在.

证明. 根据定理 1,如果防御者和攻击者都是理性的,那么在阶段博弈矩阵中存在唯一的纳什均衡点 $\boldsymbol{s}^*(X^*,Y^*)\in\{\boldsymbol{G}_1,\boldsymbol{G}_2,\boldsymbol{G}_3,\boldsymbol{G}_4\}$. 对于阶段 i,设 $\boldsymbol{S}_i^*=\boldsymbol{s}^*$,可得到双方策略的二元组序列为 $\boldsymbol{S}^*=(\boldsymbol{S}_1^*,\cdots,\boldsymbol{S}_i^*,\cdots,\boldsymbol{S}_n^*)$. 根据子博弈精炼纳什均衡的定义,当参与人的策略在每一个子博弈中都构成纳什均衡叫做精炼纳什均衡. 因此,在本节的双方博弈模型中,其策略二元组序列 $\boldsymbol{S}^*=(\boldsymbol{S}_1^*,\cdots,\boldsymbol{S}_i^*,\cdots,\boldsymbol{S}_n^*)$是子博弈精炼纳什均衡的. 证毕.

4 仿真实验

本文实验采用广泛使用的、轻量、高效的快速模拟器——基于内核的虚拟机(quickemulator kernel-based virtual machine, QEMU-KVM)——作为 VMM 实例. 僵尸虚拟机之间发送消息的方式采用安全壳(secure shell, SSH)协议. SSH 是目前广泛采用且较为可靠的、专为远程登录会话或对其他网络服务提供安全保障的协议. 目前主流云计算平台,如 Openstack 和 Eucalyptus 均采用 SSH 作为安全通信协议. 实验配置清单见表 4 所示:

Table 4 Experiment Configuration

表 4 实验配置清单

Category	Configuration
PS CPU	Intel® Core™2 Quad CPU Q8400 @ 2.66 GHz
PS Ⅰ Memory	4 GB, DDR3 1333 MHz
PS Ⅱ Memory	2 GB, DDR3 1333 MHz
PS OS	Ubuntu 12.04
Network Bandwidth/Mbps	100
SW	ZTE ZXR10 3928
VMM	QEMU-KVM V 1.2.0
Vbot CPU	QEMU Virtual CPU V 1.2.0 1 Core 2.66 GHz
Vbot Memory/MB	512
Vbot OS	KNOPPIX 3.0.4

其中物理服务器Ⅰ运行有 3 台僵尸虚拟机,物理服务器Ⅱ运行有 2 台僵尸虚拟机.实验参数设置为:E_π 分别取 0.3 和 0.7;β 初始值设为 0.5,改变量为±0.05;CDF 攻击的开始时间为第 140 秒;资源使用情况记录结束时间为第 540 秒.采用本文策略前后的 CDF 攻击对比实验结果如图 6 所示:

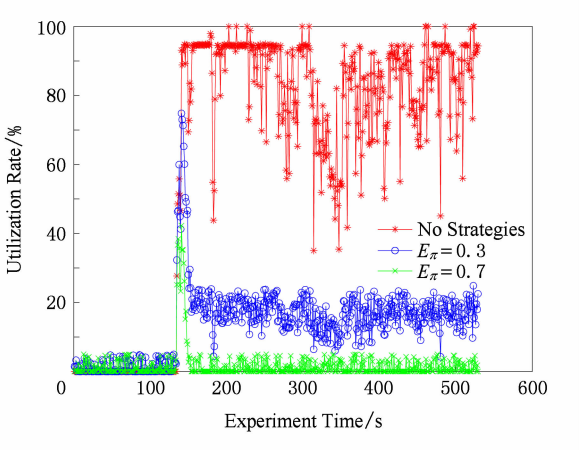


Fig. 6 CDF attack comparison.

图 6 采用本文策略的 CDF 攻击对比

实验结果表明,采用本文策略后,CDF 攻击所引起的云服务器集群内部网络带宽的消耗明显降低.本文策略对抵御 CDF 攻击效果显著.

5 结束语

本文结合传统基于虚拟机监控器和网络的特点,提出一种部署在云服务器集群内部的协同入侵检测系统(VMI-N-IDS)来抵御云环境内部攻击威胁;将 VMI-N-IDS 看作是防御者,并与 CDF 攻击者看作是博弈的双方,提出一种针对云服务器集群内部恶意软件攻击与检测的博弈论模型;分别给出博弈双方的效用函数来评估其攻防过程中的成本和收益,证明了本文模型的阶段博弈中博弈双方纳什均衡点的唯一存在性.通过对博弈双方的效用函数和动态博弈过程 2 方面的分析,证明了本文博弈模型子博弈精炼纳什均衡.由此,得到权衡误报率和恶意软件规模的最佳策略,解决了动态调整云环境内部入侵检测策略的问题.

实验表明,本文方法能够有效降低 CDF 攻击所引起的云服务器集群内部网络带宽的消耗,抵御该攻击的效果显著.最后,希望本文工作能够激励研究者的进一步研究,以更好地解决此类攻击的安全问题.

参 考 文 献

[1] Liu Zhengwei, Wen Zhongling, Zhang Haitao. Cloud computing and cloud data management technology [J]. Journal of Computer Research and Development, 2012, 49(1): 26-31 (in Chinese)
(刘正伟,文中领,张海涛. 云计算和云数据管理技术[J]. 计算机研究与发展, 2012, 49(1): 26-31)

[2] Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing v3.0 [OL]. 2011 [2014-06-20]. https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

[3] Modi C, Dhiren P, Bhavesh B, et al. A survey on security issues and solutions at different layers of cloud computing [J]. The Journal of Supercomputing, 2013, 63(2): 561-592

[4] Arbor Networks. The arbor networks 8th annual worldwide infrastructure security report finds DDoS has become part of advanced threat landscape [OL]. 2013 [2014-06-20]. http://www.arbornetworks.com/news-and-events/press-releases/recent-press-releases/4737-the-arbor-networks-8th-annual-worldwide-infrastructure-security-report-finds-ddos-has-become-part-of-advanced-threat-landscape

[5] Wang Yichuan, Ma Jianfeng, Lu Di, et al. Cloud droplets freezing attack in cloud computing [J]. Journal of Xidian University, 2014, 41(3): 116-122 (in Chinese)
(王一川,马建峰,卢笛,等. 云环境下的"云滴冻结"攻击[J]. 西安电子科技大学学报, 2014, 41(3): 116-122)

[6] Modi C, Patel D, Borisaniya B, et al. A survey of intrusion detection techniques in cloud [J]. Journal of Network and Computer Applications, 2013, 36(1): 42-57

[7] Dunlap G W, King S T, Cinar S, et al. ReVirt: Enabling intrusion analysis through virtual-machine logging and replay [J]. ACM SIGOPS Operating Systems Review, 2002, 36(SI): 211-224

[8] Liu Jiuxing, Wei Huang, Bulent A, et al. High performance VMM-bypass I/O in virtual machines [C] //Proc of the Annual Conf. Berkeley, CA: USENIX Association, 2006: 29-42

[9] Garfinkel T, Rosenblum M. A virtual machine introspection based architecture for intrusion detection [C] //Proc of the Network and Distributed Systems Security Symp. Reston, VA: Internet Society, 2003: 191-206

[10] Laureano M, Maziero C, Jamhour E. Intrusion detection in virtual machine environments [C] //Proc of the 30th Euromicro Conf. Los Alamitos, CA: IEEE Computer Society, 2004: 520-525

[11] Dhage S N, Meshram B B. Intrusion detection system in cloud computing environment [J]. International Journal of Cloud Computing, 2012, 1(2): 261-282

[12] Roschke S, Cheng F, Meinel C. An extensible and virtualization-compatible IDS management architecture [C] // Proc of the 5th Int Conf on Information Assurance and Security. Piscataway, NJ: IEEE, 2009: 130-134

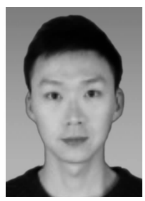
[13] Bakshi A, Yogesh B. Securing cloud from DDoS attacks using intrusion detection system in virtual machine [C] //Proc of the 2nd Int Conf on Communication Software and Networks. Piscataway, NJ: IEEE, 2010: 260-264

Wang Yichuan, born in 1983. Received his PhD degree in the School of Computer Science and Technology, Xidian University, China in 2014. Lecturer in Faculty of Computer Science and Engineering, Xi'an University of Technology. Student member of China Computer Federation. His research interests include key management and networks security.

Ma Jianfeng, born in 1963. Received his PhD degrees in computer software and communications engineering from Xidian University, China in 1995. Fellow of China Computer Federation and Senior member of IEEE. His current research interests include distributed systems, wireless and mobile computing systems, computer networks, and information and network security(jfma@mail. xidian. edu. cn).

Lu Di, born in 1983. Received his PhD degree in the School of Computer Science and Technology, Xidian University, China in 2014. Lecturer in School of Computer Science and Technology, Xidian University. Student member of China Computer Federation. His research interests include virtualization technology, operating system and storage technology(nijino2002@gmail. com).

Zhang Liumei, born in 1981. Received his MIT degree in database management from the School of Information Technology, University of Sydney, Sydney, Australia, in 2007. PhD candidate in computer architecture at Xidian University, Xi'an, Shaanxi, China. His research interests include data mining, wireless sensor networks, and intelligent computing(rikrun@gmail. com).



Meng Xianjia, born in 1985. Received his Bachelor degree in the School of Software, Xidian University, China in 2007. PhD candidate in the School of Computer Science and Technology at Xidian University, Xi'an, Shaanxi, China. His research interests include trust management and service computing (wddtsmxj8513@163. com).