文章编号：2095-2163(2019)03-0016-07 中图分类号：TP393.08 文献标志码：A

云计算网络中基于隔离边界的安全审计体系研究

葛思江[1,2]，王利明[1]，李兆璨[1,3]，马多贺[1]
（1 中国科学院 信息工程研究所，北京 100093；2 中国科学院大学 网络空间安全学院，北京 100049；
3 中国海洋大学 信息科学与工程学院，山东 青岛 266100）

摘 要：云计算市场规模逐年上升，其所面临的安全问题也越来越受到人们的关注。在云计算环境中，若网络隔离机制失效，恶意租户突破边界发起非法访问，将使云中数据资产和隐私面临巨大的安全风险。因此，本文提出一种面向云计算网络隔离边界的全周期安全审计体系，以期及时发现云中隔离失效的安全威胁，从而增强云平台安全能力。

关键词：云计算；网络隔离；安全审计

Security auditing for network isolation in cloud computing networks

GE Sijiang[1,2], WANG Liming[1], LI Zhaocan[1,3], MA Duohe[1]
(1 Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;
2 School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China;
3 College of Information Science and Engineering, Ocean University of China, Qingdao Shandong 266100, China)

【Abstract】 In cloud computing environment, if the network isolation mechanism fails and malicious tenants break through the border to initiate illegal access, data assets and privacy in the cloud will face enormous security risks. Therefore, this paper proposes a full-cycle security audit system oriented to isolation boundaries in cloud computing networks, in order to find out the security threats of isolation failure in the cloud in time, so as to enhance the security capabilities of cloud platforms.

【Key words】 cloud computing; network isolation; security auditing

0 引 言

云计算是信息技术服务模式的重大创新，通过资源（包括计算资源、网络资源、存储资源等）池化技术，使基础设施可以被多个租户共享使用。

然而，云环境中的多租户共享技术是一把双刃剑。一方面，实现了一种按需的服务方式，提供方便快捷的使用手段，提高了资源利用率，使云计算成为战略性新兴产业的重要组成部分[1]；另一方面，打破了物理设备之间的壁垒，导致安全边界模糊、弱化，其所面临的一系列安全问题也越来越受到人们的关注。

尤其在网络方面，基于云计算技术构建的数据中心与传统数据中心不同，遭受的攻击不仅来自外部，更有很大一部分来自内部恶意租户。若内部攻击者打破云计算网络中的虚拟隔离边界，发起非法访问，将使其他租户的数据资产和隐私面临巨大的安全风险。

CSA[2]将这一问题定义为共享环境中的隔离机制失效（isolation failure）。该威胁同时被 CSA、GARTNER、ENISA 等权威机构列为云内最大的安全风险来源之一[3]。这对云计算网络提出了更高的安全要求。确保云内隔离机制是否有效，也成为云服务被租户接受的前提[4]。

因此，本文提出了一套面向云计算网络隔离边界的安全审计体系，包含从审计数据采集、审计数据分析到审计数据管理的全周期安全审计流程，并实现全方位、多层次地对云中网络隔离边界进行分析。以期在保障云计算网络对租户的透明性的同时，及时发现云中隔离失效的安全威胁，从而增强云平台安全能力。

1 相关工作

为了解决共享环境中隔离失效问题，本文引入安全审计技术，以进一步研究云计算网络中基于隔离边界的安全审计体系。

基金项目：国家重点研发计划（2017YFB1010000）。

作者简介：葛思江（1994-），女，硕士研究生，主要研究方向：云安全；王利明（1978-），男，博士，正高级工程师，主要研究方向：云安全、大数据安全、网络安全等；李兆璨（1994-），女，硕士研究生，主要研究方向：云安全、大数据安全、隐私保护；马多贺（1982-），男，博士，副研究员，主要研究方向：移动目标防御、云安全、网络与系统安全等。

收稿日期：2019-03-10

安全审计是对计算机系统和计算机网络中的各种信息进行(实时)采集、分析,以查证是否发生安全事件的一种技术[5]。但云计算环境比传统 IT 信息系统复杂,其所特有的虚拟化、多租户、跨域共享等技术使得传统安全审计的具体方法不能直接迁移到云中使用。

本节首先依据审计数据来源不同,分类介绍了云安全审计的 4 个类型;然后针对其中的配置审计维度,描述现有基于云计算网络配置的安全审计相关研究工作。

1.1　审计数据来源

依据云中审计数据来源的不同,可将云环境中的审计机制分为存储审计、系统审计、网络流审计和配置审计四类。对此可做阐释解析如下。

(1)存储审计。指租户针对云端存储数据的完整性和可用性进行分析,检查数据是否被破坏或丢失[5]。

(2)系统审计。指对租户和云管理员的文件操作、进程调用等行为进行分析,检查是否存在越权、非法操作等异常行为。例如,针对云中用户操作,Majumdar 等人[6]提出利用持续监控的审计方法对云内用户认证域实施分析。

(3)网络流审计。指对云平台中的网络流量进行采集和分析,包括南北向流量和东西向流量。通过网络流审计也可分析云网络环境的安全隔离性,但该方法不属本文讨论范围。如文献[7]通过添加标记探测网络信息流,利用流量验证租户之间的隔离性;文献[8]将 Snort 入侵检测与网络审计功能模块结合;Shetty[9]监控云内网络流量,并利用机器学习自适应地调整检测阈值发现异常。

(4)配置审计。指对云平台中的网络结构配置、防火墙策略配置等进行分析。该方法可通过分析云内的网络机制是否与期望一致,查证云计算网络是否存在潜在的隔离失效安全风险。

1.2　网络配置审计

本文工作可归类为配置审计研究。针对网络配置数据源,对多租户环境下的网络结构和转发要素实施审计,分析云内网络中是否存在共享环境的隔离失效问题。

目前,由云服务提供商自身或可信第三方实施审计都已经比较常见了。文献[10-11]认为可以通过事后安全审计,在攻击发生后通过日志分析发现云内安全违规事件。Majumdar 等人[12]提出的方法支持事后追溯的审计。Madi 等人[13]将云管理平台(如 Openstack)划分为网络管理层和网络实现层,然后基于约束求解器 Sugar[14]分别对不同层面的安全隔离属性实施审计。

然而,这些工作的问题与不足就在于无法及时对云计算网络环境中的安全事件实施及时的处理。因此,近年来出现了基于持续监控实现运行时审计的研究工作。

在传统的非云网络环境中,VeriFlow[15]和 NetPlumber[16]提出了运行时的网络验证方法。通过监视网络事件的配置变化,在违规发生之前或发生时立即捕获网络违规事件。Libra[17]使用分治技术验证大型网络中是否存在可达性故障,并利用分布式并行技术提高了效率。

在云环境中,Bleikertz 等人[18-19]提出利用基于图的方法,检查云内配置变更事件,接近实时地发现云计算基础设施中的错误配置,增强安全合规能力。Majumdar 等人[20]提出利用依赖模型预先推断将发生的关键配置变更,通过主动检查以避免安全违规事件的发生,并进一步在文献[21]中提出针对文献[20]中依赖模型的改进措施,利用基于贝叶斯网络的主动学习技术自动提取事件之间的依赖,在文献[22]中进一步描述了该方法在 Openstack 中的具体实现。TenantGuard[23]对云平台三层网络进行建模,通过持续监控的审计方法实现了在运行时实施虚拟机级别的可达性验证,但该项工作没有考虑二层网络的隔离情况。

与上述相关工作的不同之处在于,本文明确了云中网络隔离边界划分和失效模式,然后基于隔离边界提出了针对不同网络层面、不同控制粒度的审计数据分析方法,从而全方位、多层次地对云计算虚拟网络进行持续分析。

2　云内网络隔离边界分析

2.1　边界失效模式

以主流开源云平台 Openstack 为例,设计采用的网络架构方案如图 1 所示,可划分为 2 个层次,分别是:网络管理层和网络实现层[13]。

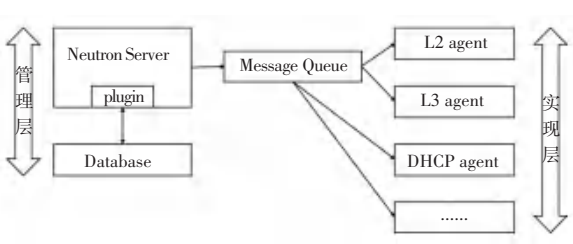

图 1　网络模型图

Fig. 1　Network model

其中,网络管理层通过控制节点上的持久化数据库维护云中网络状态,同时利用消息队列将网络配置信息下发给各个计算节点上的二层网络代理和三层网络代理实施,从而实现多租户网络的管理;网络实现层提供了云平台中的网络隔离机制的具体实现。

从网络管理层的角度看,恶意内部租户可能利用漏洞或错误配置,非法访问控制节点,从而获得云平台最高权限。另外,还可能出现恶意管理员或管理员被收买的情况。此时,攻击者可以任意地覆盖原有网络配置,植入其想要的任何网络配置规则,打破正常情况下的云平台网络隔离。

从网络实现层的角度看,云内恶意用户可对网络实现层发起攻击,篡改实现层网络配置。一方面,若攻击者获得计算节点上的 Hypervisor 权限,则可恶意篡改计算节点上的网络配置并发起非法访问;另一方面,若云平台管理软件(如 Openstack)存在漏洞,恶意租户可利用其存在的漏洞发起非法访问。例如,OpenStack Neutron OSSA-2014-008 中报告的漏洞允许租户在未授权的情况下在虚拟路由器上创建端口,连接到其他租户的网络中。

因此,针对云计算网络,应基于安全审计的流程,探讨分析其网络管理层和实现层在运行时的网络边界状态是否存在隔离失效威胁。另外,本文研究目标旨在解决隔离失效的安全问题,无法用于漏洞发现,研究提出方法当且仅当在云中破坏租户隔离机制的事件造成了配置数据变更的情况下生效。

2.2 边界划分模式

云中多租户网络隔离边界的定义依据为国内现有标准对云安全需求的抽象定义,对此可表述为:

依据 GB/T 31168-2014《云计算服务安全能力要求》归纳云环境中数据网络边界划分的基本要求:为云中虚拟网络资源上的虚拟机间的访问实施网络逻辑隔离,并提供访问控制手段。

依据 GA/T 1390.2-2017《网络安全等级保护基本要求 云计算安全扩展要求》所述内容,明确云环境中网络边界划分的详细要求,实现云租户的网络之间、安全区域之间、虚拟机之间的安全防护。

具体来说,将云中网络隔离边界粒度划分为 3 个层次。这里将展开研究论述如下。

(1)租户网络隔离边界。该边界旨在让每个租户拥有和其他租户完全隔离的一个虚拟网络,实现不同云租户之间的地址空间隔离和数据流量隔离,工业界将这样一个虚拟网络称为 VPC(Virtual private cloud)。其中,地址空间隔离指的是不同租户可以分配和使用相同的 IP 地址,各租户可自定义网段;数据流量隔离指的是云内各个租户无法嗅探或感知到其他虚拟网络内部的流量。

(2)安全区域隔离边界。在云计算网络环境中,租户内部可以规划子网,所以一般来说,租户通常采用访问控制手段,再根据所承载业务的安全保护等级来划分基础设施资源池,从而形成安全区域边界,使不同安全保护等级的区域之间实现完全隔离,无法相互通信。

(3)虚拟机隔离边界。与安全区域之间的隔离边界相比,虚拟机之间的隔离边界粒度更细。两者的区别在于,前者的安全防护措施在边界上实施,而后者是直接作用于云中虚拟机节点的。

以 Openstack 为例,不同物理节点上不同租户虚拟机进行通信时,网络实现层相关隔离边界设备如图 2 所示,包括虚拟交换机、虚拟路由器、虚拟防火墙和安全组。

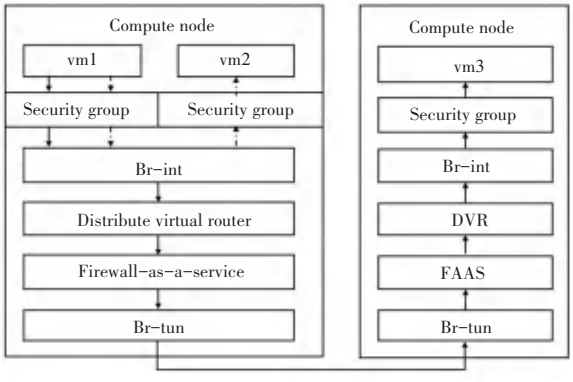

图 2 网络隔离设备

Fig. 2 Network isolation equipment

综上,本节根据现有安全标准中的抽象描述,明确细化了云中多租户网络的隔离边界,将其划分为 3 个级别,作为网络隔离边界审计的基准。

3 云内网络隔离边界审计体系

3.1 体系概述

在传统计算机系统安全审计领域,出现的第一个正式标准是 TCSEC(Trusted Computer System Evaluation Criteria),于 1970 年由美国国防科学委员会提出。在这之后,还有欧洲出台的 ITSEC 标准、加拿大出台的 CTCPEC 标准等随即也陆续涌现,直到 6 个国家(美、加、英、法、德、荷)共同起草了信息技术安全评价通用准则(The Common Criteria for Information Technology security Evaluation),简称 CC

标准。至此，则综合已有的信息安全的准则和标准，形成了一个全面的框架。

在云计算安全审计领域，本章节所描述的云平台网络隔离边界审计体系的主要参考依据来源于 CSA 云安全联盟发布的 Security，Trust & Assurance Registry(Star)[24] 标准。Star 标准定义了云内验证评估和审计的框架，划分了云安全审计层次。第一个层次是云服务提供商的自我评估；第二个层次是由可信第三方实施审计，确保云服务满足 CCM 云安全控制矩阵[25] 要求；第三个层次是利用持续监控(continue monitor)的方法实施审计。

本章叙述的网络隔离边界审计体系，遵循的是第三个层次持续监控要求。1.2 节中所提到的针对云内网络的动态验证的方法，即属于第三个层次。具体的持续监控方案，依据标准《云计算服务安全能力要求》制定。

云计算网络中基于隔离边界的安全审计体系架构如图 3 所示。该体系面向云计算网络，旨在在云平台运行过程中，全方位、全周期、多层次地对网络隔离边界实施审计。

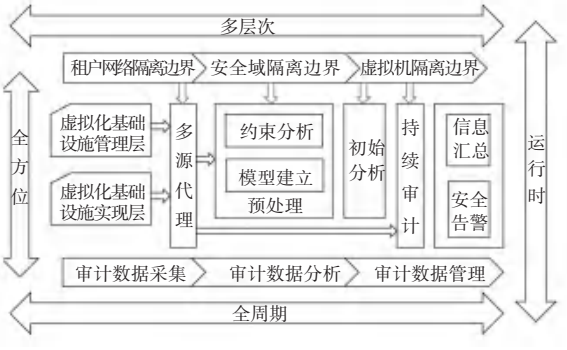

图 3　云内网络隔离边界审计体系

Fig. 3　Audit system for network isolation

依据不同分类维度，该体系主要特点可概论如下：

(1)全方位：从云平台网络架构的维度出发，分别对不同来源的审计数据，包括虚拟化基础设施的网络管理层配置数据和底层网络实现层配置数据，进行全方位的处理和分析。

(2)多层次：从网络隔离边界层次的维度出发，分别依据不同粒度级别的隔离边界模式，包括租户网络隔离边界、安全域隔离边界和虚拟机隔离边界，对审计数据进行多层次的分析。

(3)全周期：从功能维度出发，分别按照审计数据采集、审计数据分析和审计数据管理的基本功能流程，对多租户网络隔离边界进行全周期的审计。

(4)运行时：在云计算平台运行过程中，当配置状态不满足边界划分模式时，认为网络隔离边界被打破。此时，云网络环境中存在潜在的隔离失效威胁，可能被恶意租户利用发起非授权访问。

接下来，本文分别详细说明该体系中审计数据采集、审计数据分析和审计数据管理三个基本功能流程中采用的具体方法。

3.2　审计数据采集

针对网络隔离边界进行安全审计，所需数据来源于云平台的控制节点和各个计算节点，具有分布式的特征。因此，本文引入代理机制，持续获取分布在云平台各个节点的网络配置数据，并采用消息机制汇总到可信第三方。分布式代理运行机制如图 4 所示。对其中每一部分的设计功能可给出分析详述如下。

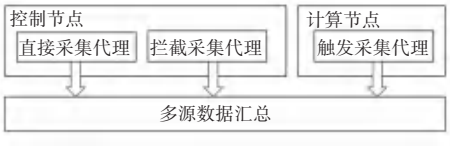

图 4　分布式代理机制

Fig. 4　Distributed agent mechanism

(1)直接采集代理。部署在控制节点上，对管理层网络配置信息进行初始数据采集，只在整套审计系统部署时运行一次，且运行时间比较长。要求在采集过程中没有发生配置变更，云平台网络状态稳定不变。

(2)拦截采集代理。对管理层配置变动操作进行截取。只有当可信第三方的分析结果显示没有违反网络隔离边界时，代理才将数据转发，写入云内持久化数据库，并下发到各个节点。否则，被拦截的配置变更操作直接被丢弃。

(3)触发采集代理。用于获取实现层配置变动。通过触发器机制对网络实现层上虚拟交换机、虚拟防火墙、虚拟路由器的配置数据项进行采集，从而避免因直接遍历带来的节省时间开销。采集频率由审计第三方指定。

3.3　审计数据分析

审计数据分析是实施云安全审计的核心。本文主要采用基于规则库的审计分析方法，通过专家经验预先定义规则集合，构成规则库，然后对采集数据进行处理，提取审计数据特征，与规则进行某种比较和匹配操作，发现具有明显特征的违规行为。

如图 5 所示，本文提出一种基于图的网络隔离边界审计分析方法。设计研究内容详见如下。

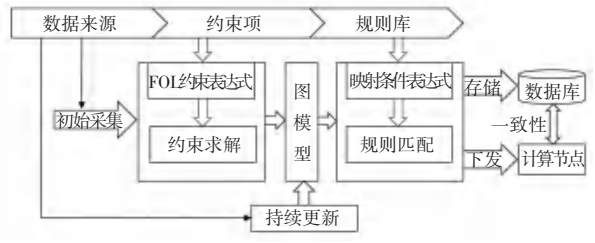

图 5 数据分析流程

Fig. 5 Data analysis for network isolation

3.3.1 预处理与图建模

约束分析的目标是在建立模型前,检测是否存在违背约束条件的配置。本文使用谓词表示法,根据采集的网络配置数据,建立个体之间的关系。研究中定义的基本状态谓词见表 1。

表 1 基本谓词定义

Tab. 1 Predicates definition

谓词	定义
BELONG(x,y)	x 属于 y
ALLOT(x,y)	x 分配给 y
RELATE(x,y)	x 关联到 y
CONNECT(x,y,z)	x 通过 y 连接到 z

以 Openstack 为例,涉及到的个体有租户 t、虚拟网络 vn、段号 sg、端口 vp、虚拟机实例 vm。将这些个体代入谓词中,并利用逻辑连词将其联结起来,转换为一阶逻辑(FOL, First-order logic)表达式描述云中网络约束条件的语义。对此可得研究表述如下。

(1)约束项一:在多租户网络中,虚拟机实例是属于云内各个租户的独享资源,不能有一个虚拟机实例同时属于 2 个或更多的租户。表达式记为:

$\forall vm \in VM, \forall t1, t2 \in TENANT$:

$$BELONG(vm,t1) \wedge BELONG(vm,t2) \rightarrow (t1 = t2). \quad (1)$$

(2)约束项二:租户网络隔离边界的核心是段号。段号可以有不同的实现方式,但其与租户必定是一对一对应关系。表达式记为:

$\forall t1, t2 \in NET, \forall seg1, seg2 \in SEG$:

$$RELATE(seg1,t1) \wedge RELATE(seg1,t2) \rightarrow (t1 = t2)$$

$$RELATE(seg1,t1) \wedge RELATE(seg2,t1) \rightarrow (seg1 = seg2). \quad (2)$$

(3)约束项三:进一步地,段号被分配给虚拟网络时,不能有一个段号被分配给不同的虚拟网络。表达式记为:

$\forall vn1, vn2 \in NET, \forall seg1, seg2 \in SEG$:

$$ALLOT(seg1, vn1) \wedge ALLOT(seg2, vn2) \wedge \neg (seg1 = seg2) \rightarrow \neg (vn1 = vn2)$$

$$ALLOT(seg1, vn1) \wedge ALLOT(seg2, vn2) \wedge \neg (seg1 = seg2) \rightarrow \neg (vn1 = vn2). \quad (3)$$

(4)约束项四:再进一步地,不能将不同的段号关联到同一个端口。表达式记为:

$\forall seg1, seg2 \in SEG, \forall vp \in PORT$:

$$ALLOT(seg1, vp) \wedge ALLOT(seg2, vp) \rightarrow (seg1 = seg2). \quad (4)$$

(5)约束项五:在云环境中,虚拟机实例通过关联到虚拟网卡的端口,拥有其自己的 MAC 地址,然后连入虚拟网络。分配给虚拟网络的段号和与虚拟端口关联的段号应当是一致的。表达式记为:

$\forall vm \in VM, \forall vn \in NET, \forall seg1, seg2 \in SEG, \forall vp \in PORT$:

$$CONNECT(vm, vn, vp) \wedge ALLOT(seg1, vn) \wedge RELATE(vp, seg2) \rightarrow (seg1 = seg2). \quad (5)$$

采用现有 SAT 求解器(如 Sugar[14])求解当前网络配置状态是否满足上述五项约束条件。如果输入数据满足约束条件(SAT),则 Sugar 求解器将提供所有解,否则将返回 UNSAT。利用这一特性,把以上五项 FOL 表达式取反(¬),再放入求解器,此时若有解,则意味着网络配置违规。

当验证约束条件满足,初始建立图模型。定义有向图模型 $G = (V,E,C)$。其中,V 表示虚拟机节点,集群中任一节点满足 $v_i \in V$;E 表示各节点之间的边,集群中的边 $e_{i,j}$ 即表示节点 v_i 可访问另一节点 v_j,$e_{i,j} \in E$;C 表示节点之间的连通关系,$c_{i,j}$ 表示节点 v_i 与 v_j 之间的连通关系,图 G 中初始默认 = 0,若 v_i 与 v_j 连通,则 $c_{i,j} = 1, c_{i,j} \in C$。将最终建立的初始状态图模型记为 G_{init}。

3.3.2 初始隔离分析

依据 2.2 节中所述边界划分模式,本文预先定义的规则包括如下 2 类:

(1)TI 规则集合。TI 规则集合定义映射 f,区分隶属于不同租户网络的虚拟机节点。若使 n 表示虚拟机节点的标识,$tenant_n$ 表示标识为 n 的虚拟机节点所属的租户,则可将 TI 规则集合中单个规则 ti 记为:

$$f(n) = tenant_n. \quad (6)$$

(2)ZI 规则集合。ZI 规则集合针对多租户云平台中属于同一租户的虚拟机,定义 L 为域等级。L 值为 uppr、normal 或 lower。其中,uppr 域可访问非 uppr 的所有域,不可被访问;normal 域可与 normal 域等级相等的域互访;lower 的域可被所有域访问,不可发起访问。对于域内的节点,若 n 表示节点的标识,$level_n$ 表示标识为 n 的节点的标记值,则计算

节点 n 的标记值 $level_n$ 定义为：

$$level_n = \begin{cases} 1, & L = \text{uppr}; \\ 0, & L = \text{lower}; \\ x, & L = \text{normal},\ x > 1. \end{cases} \tag{7}$$

若建立同一租户内各个安全域与虚拟机节点的正确映射关系 y，则可将 ZI 规则集合中单个规则 zi 记为：

$$y(n) = level_n, \tag{8}$$

针对 TI 规则集合，有映射规则 f 代表租户网络与云中虚拟机节点的正确映射关系。则 TI 规则集合初始分析过程中，遍历查询图 G_{init} 中虚拟机节点 v_i 与 v_j 之间的连通关系 $c_{i,j}$。若 $c_{i,j} = 1$，则图模型 G_{init} 在满足以下条件时合规：

$$f(i) = f(j), \tag{9}$$

若 $c_{i,j} = 1$，则图 G_{init} 在满足以下条件时合规：

$$f(i) \neq f(j), \tag{10}$$

反之则认为连通关系 $c_{i,j}$ 违背租户网络隔离边界要求。

针对 ZI 规则集合，初始分析过程中，遍历查询图 G_{init} 中虚拟机节点 v_i 与 v_j 之间的连通关系 $c_{i,j}$。若 $c_{i,j} = 1$，则当且仅当 $level_i$ 值与 $level_j$ 值满足以下规则时，图模型 G_{init} 中连通关系 $c_{i,j}$ 合规：

(1) $y(i) \neq 0 \wedge y(j) = 0$；

(2) $y(i) = 1 \wedge y(j) \neq 1$；

(3) $y(i) = y(j) = y(j) \wedge y(i) > 1 \wedge y(j) > 1$. (11)

若 $c_{i,j} = 0$，则当且仅当 $level_i$ 值与 $level_j$ 值满足以下规则时，图模型 G_{init} 中连通关系 $c_{i,j}$ 合规：

(1) $y(i) = 0$；

(2) $y(i) \neq 1$；

(3) $y(i) \neq y(j) \wedge y(i) > 1$. (12)

反之，则认为图模型 G_{init} 中连通关系 $c_{i,j}$ 违背安全域隔离边界与虚拟机隔离边界要求。

3.3.3　持续隔离分析

针对管理层网络，通过主动拦截管理操作，对待更改的配置数据进行提前审计分析。当云平台中的管理员或租户管理员实施创建实例、删除实例、创建网络、删除网络、创建子网、删除子网、创建安全策略、删除安全策略等操作时，增量更新图模型，记为 G_{time}，$time$ 表示当前时间戳。分析图模型 G_{time} 的增量部分是否满足网络边界隔离的安全需求。若符合边界安全需求，则接受并下发配置；若不符合边界安全需求，则跳转到审计管理流程，由审计管理员做出决定。

针对实现层网络，虚拟网络设备分布在云平台的各个节点，使得原来的可信边界被打破，导致云计算网络面临更多的安全风险。因此按照固定频率采集虚拟设备数据，同时对虚拟交换机流表、虚拟路由器路由表、虚拟防火墙和安全组规则等的配置状态进行计算和分析，得到底层实现网络中虚拟机连通状态，与管理层图模型 G 中连通度 C 进行比对。若不一致，则跳转到审计管理流程，发出安全告警。

3.4　审计数据管理

审计数据管理旨在使审计管理员能实时查看审计数据分析结果，了解云平台运行过程中多租户网络隔离情况。具体功能包括：查看审计分析结果、查询审计记录、安全告警等。

审计管理员以审计数据分析结果为依据，检查云平台中网络隔离边界的安全状态变化，并针对云平台中存在的潜在隔离失效威胁，下发安全策略，做出相关决策。

4　结束语

本文通过对云计算网络中的隔离边界进行分析，设计了一套基于隔离边界的安全审计体系。该体系包含从审计数据采集、审计数据分析到审计数据管理的全周期安全审计流程，并提出了全方位、多层次的隔离分析方法。本工作研究解决了云中多租户网络隔离边界被打破，导致共享环境中的网络隔离失效这一问题，从而有效防范云内潜在恶意租户发起的非法访问。

参考文献

[1] 工业和信息化部. 云计算综合标准化体系建设指南[J]. 电子政务，2015(11)：14.

[2] Cloud SecurityAlliance. Top threats to cloud computing[EB/OL]. [2018-08-08]. https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-deep-dive/.

[3] 中国信息通信研究院. 云计算发展白皮书[EB/OL]. [2018-08-14]. http://www.caict.ac.cn/xwdt/hyxw/201808/t20180808_181480.htm.

[4] 石勇，郭煜，刘吉强，等. 一种透明的可信云租户隔离机制研究[J]. 软件学报，2016，27(6)：1538-1548.

[5] 王文娟，杜学绘，王娜，等. 云计算安全审计技术研究综述[J]. 计算机科学，2017，44(7)：16-20，30.

[6] MAJUMDAR S，MADI T，WANG Yushun，et al. User-level runtime security auditing for the cloud[J]. IEEE Transactions on Information Forensics and Security，2018，13(5)：1185-1199.

[7] PRIEBE C，MUTHUKUMARAN D，O´KEEFFE D，et al. Cloudsafetynet：Detecting data leakage between cloud tenants[C]//Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security. Scottsdale，Arizona，USA：ACM，2014：117-128.

[8] 包捷，吕智慧，华锦芝，等. 私有云环境下安全审计系统的设计与实现[J]. 计算机工程与设计，2014，35(11)：3708-3711，

3729.
[9] SHETTY S. Auditing and analysis of network traffic in cloud environment[C]//Proceedings of 2013 IEEE Ninth World Congress on Services. Santa Clara, CA, USA: IEEE, 2013: 260-267.
[10] MADI T, MAJUMDAR S, WANG Yushun, et al. Auditing security compliance of the virtualized infrastructure in the cloud: Application to OpenStack [C]//Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy. New Orleans, LA, USA: ACM, 2016: 195-206.
[11] ULLAH K W, AHMED A S, YLITALO J. Towards building an automated security compliance tool for the cloud[C]//2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Melbourne, VIC, Australia: IEEE, 2013: 1587-1593.
[12] MAJUMDAR S, MADI T, WANG Yushun, et al. Security compliance auditing of identity and access management in the cloud: Application to OpenStack [C]//2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom). Vancouver, BC, Canada: IEEE, 2015: 58-65.
[13] MADI T, JARRAYA Y, ALIMOHAMMADIFAR A, et al. ISOTOP: Auditing virtual networks isolation across cloud layers in OpenStack [J]. ACM Transactions on Privacy and Security (TOPS), 2018, 22(1): 1.
[14] TAMURA N, BANBARA M. Sugar: A CSP to SAT translator based on order encoding [C]//Proceedings of the Second International CSP Solver Competition. [S.l.]: CSP, 2008: 65-69.
[15] KHURSHID A, ZHOU Wenxuan, CAESAR M, et al. Veriflow: Verifying network-wide invariants in real time[C]// Proceedings of the first workshop on Hot topics in software defined networks. Helsinki, Finland: ACM, 2012: 49-54.
[16] KAZEMIAN P, CHANG M, ZENG H, et al. Real time network policy checking using header space analysis[C]// the 10th USENIX Symposium on Networked Systems Design and Implementation. Lombard, IL: USENIX Association, 2013: 99-111.
[17] ZENG Hongyi, ZHANG Shidong, YE Fei, et al. Libra: Divide and conquer to verify forwarding tables in huge networks[C]// NSDI´14 Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation. Seattle, WA: ACM, 2014: 87-99.
[18] BLEIKERTZ S, VOGEL C, GROß T. Cloud radar: Near real-time detection of security failures in dynamic virtualized infrastructures [C]//Proceedings of the 30th Annual Computer Security Applications Conference. New Orleans, Louisiana, USA: ACM, 2014: 26-35.
[19] BLEIKERTZ S, VOGEL C, GROß T, et al. Proactive security analysis of changes in virtualized infrastructures[C]//Proceedings of the 31st Annual Computer Security Applications Conference. Los Angeles, CA, USA: ACM, 2015: 51-60.
[20] MAJUMDAR S, JARRAYA Y, MADI T, et al. Proactive verification of security compliance for clouds through pre-computation: Application to OpenStack [C]//European Symposium on Research in Computer Security. Cham: Springer, 2016: 47-66.
[21] MAJUMDAR S, JARRAYA Y, OQAILY M, et al. LeaPS: Learning-based proactive security auditing for clouds [M]// FOLEY S N, GOLLMANN D, SNEKKENES E. Esorics 2017. LNCS. Cham: Springer, 2017, 10493: 265-285.
[22] TABIBAN A, MAJUMDAR S, WANG Lingyu, et al. Permon: An openstack middleware for runtime security policy enforcement in clouds[C]//2018 IEEE Conference on Communications and Network Security (CNS). Beijing, China: IEEE, 2018: 1-7.
[23] WANG Yushun, MADI T, MAJUMDAR S, et al. Tenantguard: Scalable runtime verification of cloud-wide VM-level network isolation [C]// Network and Distributed System Security Symposium. San Diego, CA, USA: 2017 Internet Society, 2017: 1-15.
[24] Cloud SecurityAlliance. CSA STAR Program and Open Certification Framework in 2016 and Beyond[EB/OL]. [2016-12-04]. https://cloudsecurityalliance.org/artifacts/csa-star-program-open-certification-framework-in-2016-and-beyond/.
[25] Cloud SecurityAlliance. Cloud Control Matrix CCM v3.0.1[EB/OL]. [2014-09-16]. http://www.coursehero.com/file/14169827/CSA-CCM-v301-09-16-2014xlsx/.

(上接第15页)

[3] 唐晓东. 套牌机动车辆检测方法分析[J]. 中国人民公安大学学报(自然科学版), 2013, 19(2): 76-79.
[4] 黄银龙,王占斌,徐旭,等. 基于ISO/IEC18000-6B标准的RFID车卡防伪问题研究[J]. 中国电子商情(RFID技术与应用), 2008(5): 39-42.
[5] 杨博. 物联网ZigBee技术在套牌车监管中的应用研究[J]. 制造业自动化, 2012, 34(17): 41-43.
[6] 黄小龙. 基于卡口车辆特征信息表达方法及其在车辆图像智能分析系统中的应用实现[D]. 广州: 中山大学, 2016.
[7] 笪东旭. 基于车脸识别的套牌车检测方法研究[D]. 西安: 西安电子科技大学, 2013.
[8] 卢晓春,周欣,蒋欣荣,等. 基于网格化监控的套牌车检测系统[J]. 计算机应用, 2009, 29(10): 2847-2848.
[9] 俞东进, 平利强, 李万清,等. 一种基于Hadoop的套牌车识别方法: 中国, CN104035954A[P]. 2014-09-10.
[10] 王涛,王顺,沈益民. 交通流大数据中的套牌车并行检测算法[J]. 湖北工程学院学报, 2014, 34(6): 29-32.
[11] 李悦. 大规模数据集关联关系并行发现与优化方法研究[D]. 北京: 北方工业大学, 2016.
[12] 莫迪. 基于大数据分析的套牌实时检测系统研究与实现[D]. 上海: 东华大学, 2017.
[13] 姬倩倩. 公共交通大数据平台架构服务模式研究[D]. 西安: 西安电子科技大学, 2014.
[14] ITS. National intelligent transportation systems program plan: A ten-year vision[Z]. USA: The Intelligent Transportation Society of America and U.S. Department of Transportation, 2002.
[15] SUNDERAM V S. Current trends in high performance parallel and distributed computing [C]// 7th International Parallel and Distributed Processing Symposium (IPDPS 2003). Nice, France: dblp, 2003: 1.
[16] ZHU Tongyu, YU Jianjun, DU Bowen. RTIC-C: A cloud computing platform for history data mining of traffic information [C]// 2012 International Conference on Connected Vehicles and Expo (ICCVE). Beijing, China: IEEE, 2012: 282-283.