抗侧信道攻击的服务功能链部署方法

伊 鹏 谢记超* 张 震 谷允捷 赵 丹
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：侧信道攻击是当前云计算环境下多租户间信息泄露的主要途径，针对现有服务功能链(SFC)部署方法未充分考虑多租户环境下虚拟网络功能(VNF)面临的侧信道攻击问题，该文提出一种抗侧信道攻击的服务功能链部署方法。引入基于时间均值的租户分类策略以及结合历史信息的部署策略，在满足服务功能链资源约束条件下，以最小化租户所能覆盖的服务器数量为目标建立相应的优化模型，并设计了基于贪婪选择的部署算法。实验结果表明，与其他部署方法相比，该方法显著提高了恶意租户实现共存的难度与代价，降低了租户面临的侧信道攻击风险。

关键词：侧信道攻击；服务功能链；部署方法；租户分类；历史部署信息
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2019)11-2699-09
DOI: 10.11999/JEIT190127

A Service Function Chain Deployment Method Against Side Channel Attack

YI Peng XIE Jichao ZHANG Zhen GU Yunjie ZHAO Dan
(*National Digital Switching System Engineering & Technological Research Center*, *Zhengzhou* 450002, *China*)

Abstract: Side channel attack is the primary way to leak information between tenants in current cloud computing environment. However, existing Service Function Chain (SFC) deployment methods do not fully consider the side channel attack problem faced by the Virtual Network Function (VNF) in the multi-tenant environment. A SFC deployment method is proposed against side channel attack. A tenant classification strategy based on average time and a deployment strategy considering historical information are introduced. Under the resource constraints of the SFC, the optimization model is established with the goal of minimizing the number of servers that the tenant can cover. And a deployment algorithm is designed based on the greedy choice. The experimental results show that, compared with other deployment methods, this method can significantly improve the difficulty and cost of malicious tenant to realize co-residence, and reduces the risk of side channel attack faced by tenants.

Key words: Side channel attack; Service Function Chain(SFC); Deployment method; User classification; Historical deployment information

1 引言

端到端网络服务通常需要其流量经过一系列有序的网络功能和应用程序的处理，而这一组功能序列被称为服务功能链(Service Function Chain, SFC)[1]。然而传统的网络服务提供方式需要部署大量复杂的网络功能，面临着不断扩展基础设施以及维护复杂网络所产生的巨大成本[2]。网络功能与网络拓扑的紧耦合特性，也愈发难以满足新形势下网络服务对多样化、定制化和动态化网络功能的需求[3]。云计算、软件定义网络(Software Defined Network, SDN)以及网络功能虚拟化(Network Function Virtualization, NFV)技术的出现为网络的发展注入了极大活力，基于三者的SFC部署方案为解决上述问题提供了良好的思路[3]，利用云计算提供的按需索取的资源，借助NFV技术将各类网络功能以软件形式运行在通用硬件之上，即：虚拟网络功能(Virtual Network Function, VNF)，并基于SDN技术实现业务流量在VNFs间的灵活高效引

收稿日期：2019-03-01；改回日期：2019-06-11；网络出版：2019-06-20
*通信作者： 谢记超 912104210329@njust.edu.cn
基金项目：国家自然科学基金(61802429, 61872382, 61521003)，国家重点研发计划(2017YFB0803201, 2017YFB0803204)
Foundation Items: The National Science Foundation of China (61802429, 61872382, 61521003), The National Key R&D Program of China (2017YFB0803201, 2017YFB0803204)

导，这种新型的SFC部署方案为网络服务的部署提供了前所未有的灵活性。

然而这种新型的SFC部署方案在走向成熟的过程中仍面临着许多挑战，如：资源分配优化、动态服务映射、服务策略实施、服务可靠性、安全性等，现有SFC部署主要以端到端时延[4]、运营成本[5]、资源利用率[6,7]、可靠性[8]等作为优化目标，而在保证安全性方面考虑的较少。然而安全性始终是一个重要的焦点，NFV在落地实施的过程中也面临着众多的安全挑战，包括NFV特有的安全威胁、通用虚拟化安全威胁和通用网络威胁等[9]。本文则重点关注通用虚拟化安全威胁中VNF面临的侧信道攻击[10]问题。

在相关的虚拟机(Virtual Machine, VM)部署和虚拟网(Virtual Network, VN)映射领域，针对侧信道攻击的防御方法主要分为以下3类。

(1) 消除侧信道。如：修改调度程序、采用硬件隔离机制等，该类方法需要对硬件、VM或VM管理程序进行重大修改，难以很快被云服务提供商所采用，此外该类方法往往不能覆盖到未来不断被发掘出来的侧信道攻击手段[11]，需要更加通用的防御方法。

(2) 定期迁移[12,13]和触发迁移[14]。文献[12]对亚马逊数据中心进行了为期5个月的实验，深入分析了攻击者实现共存的效果，并提出了一套权衡共存时间和迁移代价的迁移指南。文献[13]针对定期动态迁移方法存在迁移算法收敛时间长、开销过大的问题，提出了一种基于安全等级的VM动态迁移方法，显著降低了VM迁移的数量和频率，但是该方法对关键VM的定义具有一定的租户主观性。文献[14]建议对侧通道攻击进行实时检测，并采用触发式的VM迁移来防止信息的泄漏，但是此类方法需要掌握相关侧信道攻击手段的特征，难以应对特征未知的侧信道攻击。现有定期迁移和触发迁移方法虽然合理地控制了迁移频率和迁移开销，但是该类方法在迁移过程中会造成一定时间的服务中断，对服务质量有较大影响，此外VM在迁移过程中也面临着诸多安全风险[15]。

(3) 提高共存难度。文献[16]为了解决VN中信息的机密性和完整性保护问题，引入了安全级别和安全性需求的概念，但是该方法使得恶意租户可通过高安全需求的资源请求实现与高安全需求租户的轻易共存。文献[17]比较了常用的VM部署策略在防御侧信道攻击时的安全性，并提出了基于博弈论的混合部署策略，并在之后的研究[18]中，利用聚类分析和半监督学习方法实现对租户的分类，并基于租户分类结果对VM进行分域部署，但是所设计的分类方法过于复杂，且并不能严格准确地区分出恶意租户与合法租户，但是相关租户分类策略具有很好的参考价值，本文的研究便是受到了该研究的启发。

针对现有SFC部署方法缺乏对侧信道攻击问题的考虑，本文在借鉴相关领域防御方法之上，从提高共存难度的角度出发，提出了一种抗侧信道攻击的服务功能链部署方法，引入基于时间均值的租户分类策略以及结合历史信息的部署策略，并以租户所能覆盖的服务器数量为优化目标建立了部署模型，设计了基于贪婪选择的部署算法，最后通过实验对所提方法的性能进行了深入评估。

2 问题描述与模型建立

2.1 侧信道攻击问题描述

云服务提供商可将VNF作为服务提供给租户，并借助虚拟化技术实现资源在多租户间的高效复用，租户可按需租用VNF。一个典型的租户SFC请求实例如图1上半部分所示，该请求包含4个VNF，图1描述了该SFC请求在网络中的部署情况。然而恶意租户可利用云服务提供商的资源分配缺陷，采用一定的SFC请求策略与目标租户以较大概率共存于同一服务器，图2简要描述了恶意租户实现共存的过程。恶意租户在成功实现与目标租户VNF共存后，可利用共享资源(如：CPU、内存、磁盘、网络)构建广泛的侧信道[10]，如图3所示，进而从共存的VNF中获取敏感信息，范围可从粗粒度的工作负载、流量速率到细粒度的加密密钥等，有些看似无害的信息有时也是极其有用的，如工作负载统计信息可用于识别系统何时最易受攻击[18]。如果不能很好地解决租户VNF面临的信道攻击风险，将直接影响这种新型网络服务提供方式的广泛应用。

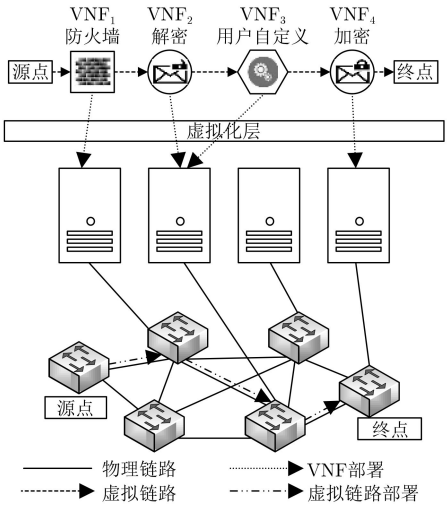

图 1 基于云环境的服务功能链部署示意图

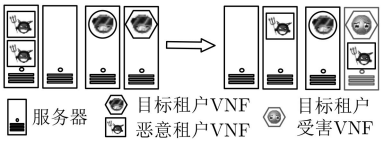

图 2 恶意租户实现共存过程

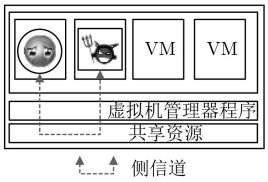

图 3 恶意租户实施侧信道攻击

2.2 系统模型

(1) 数据中心网路。可用一个无向图$\bar{G}=(\bar{N},\bar{S},\bar{L})$表示，其中$\bar{N}$, $\bar{S}$, $\bar{L}$分别表示服务器、交换机和物理链路集合，用n, s, l分别表示服务器、交换机和物理链路的总数量。定义$\boldsymbol{B}_{S\times S}$为交换机连接矩阵，元素$B_{i,j}\in R^{+}$表示交换机节点$i$到$j$的通信链路容量。用$V=\eta(u)=\{v|B_{u,v}>0\}\subseteq\bar{S},u,v\in\bar{S}$表示与交换机$u$直连的交换机集合。定义二值矩阵$\boldsymbol{H}_{n\times s}$为服务器与交换机连接矩阵，元素$H_{i,j}\in\{0,1\}$表示服务器节点$i$是否连接在交换机$j$上。定义$K$为服务器资源类型(如：CPU、内存、存储)集合，用k表示资源类型总数量。定义$\boldsymbol{C}_{n\times k}$为底层服务器资源容量矩阵，元素$C_{i,j}\in R^{+}$表示服务器节点$i$上可提供的第$j$类资源容量。定义矩阵$\boldsymbol{C}^{\mathrm{rem}}_{n\times k}$和$\boldsymbol{B}^{\mathrm{rem}}_{s\times s}$分别表示当前网络服务器资源和链路资源的剩余情况。

(2) 虚拟网络功能。定义P为VNF类型集合(如：Firewall, IDS等)，用p表示VNF类型总数量，定义$\boldsymbol{Q}_{p\times k}$为VNF资源需求系数矩阵，元素$Q_{i,j}$表示$i$类型VNF处理单位带宽流量所占用的$j$类资源数量。定义二值矩阵$\boldsymbol{S}_{n\times p}$，表示服务器节点可承载的VNF类型矩阵，元素$S_{i,j}\in\{0,1\}$表示服务器节点$i$是否支持$j$类型VNF的部署。

(3) 服务功能链请求。租户集合用Ω表示，SFC请求集合用R表示，每一个请求信息可用一个6元组$r=<\eta^r,\bar{u}^r,\bar{v}^r,\beta^r,\tau^r,\psi^r>$表示，其中$\eta^r\in\Omega$表示该请求所属的租户，$\bar{u}^r,\bar{v}^r$表示接入和出口交换机，$\beta^r$表示该服务链所需处理的流量大小，$\tau^r$表示该请求的生命周期，$\psi^r$表示处理流量所需的VNF序列，用$m$表示该请求中VNF的总数量，定义行向量$\boldsymbol{T}^r_{1\times m}$表示该请求的VNF组成类型向量，元素$T^r_{1,i}\in P$表示请求中第$i\in\{1,2,\cdots,m\}$个VNF的类型。每一个SFC请求均可用一个有向图$G^r=(N^r,L^r)$表示，其中$N^r$为节点(接入交换机，VNFs，出口交换机)集合，$L^r$为连接这些节点的虚拟链路集合。

(4) 服务功能链部署。服务提供商接收到租户的SFC请求后，根据请求信息r以及当前网络资源状态，按照一定策略完成SFC部署。SFC部署可描述为将请求拓扑G^r映射到物理网络拓扑$\bar{G}$之上，定义为$M:G^r\rightarrow\bar{G}^r\subseteq\bar{G}$，其中$\bar{G}^r$是$\bar{G}$的子集，若映射成功，则随后完成所需VNF的实例化以及转发路径的下发。

2.3 服务功能链部署模型

本文以最小化租户所能覆盖的服务器数量为目标函数，建立了服务功能链部署模型。定义二值矩阵$\boldsymbol{F}^r_{m\times n}$表示请求$r$中的$m$个VNF与$n$个服务器节点间的映射关系，元素$F^r_{i,j}\in\{0,1\}$，表示第$i$个虚拟网络功能$\mathrm{VNF}_i{}^r$是否部署在服务器节点$j$上。定义$n^r_i\in\bar{N}$，表示$\mathrm{VNF}_i{}^r$部署的服务器节点。定义$s^r_i\in\bar{S}$，表示$VNF_i{}^r$部署的服务器节点$n^r_i$所直连的交换机。当$i=0$时，令$s^r_0=\bar{u}^r$，当$i=m+1$，令$s^r_{m+1}=\bar{v}^r$，分别表示接入交换机和出口交换机。定义二值矩阵$\boldsymbol{E}^{s^r_i,s^r_{i+1}}_{s\times s}$表示$\mathrm{VNF}_i{}^r$与$\mathrm{VNF}^r_{i+1}$之间的虚拟链路$l^r_{i,i+1}\in L^r$与物理链路$\bar{l}_{u,v}\in\bar{L}$之间的映射关系，元素$E^{s^r_i,s^r_{i+1}}_{u,v}\in\{0,1\}$表示虚拟链路$l^r_{i,i+1}$是否部署在物理链路$\bar{l}_{u,v}$之上。此外用$B^r_{\cos t}$表示请求$r$的总的链路带宽资源消耗。

目标函数：最小化每个租户所能覆盖的服务器数量。

$$\min\sum_{r\in R}\sum_{i=1}^{m}\left[\left|\left\{n\left|\mathrm{n}\in\bar{N}^{\mathrm{XT}},F^r_{i,n}=1,\eta^r\odot\eta=1\right.\right\}\right|\right],\quad\forall\eta\in\Omega \tag{1}$$

约束条件

$$\sum_{j\in\bar{N}^{\mathrm{XT}}}F^r_{i,j}=1,\quad\forall i\in\{1,2,\cdots,m\} \tag{2}$$

$$S_{n^r_i,T^r_{1,i}}=1,\quad\forall i\in\{1,2,\cdots,m\},n^r_i\in\bar{N}^{\mathrm{XT}} \tag{3}$$

$$\sum_{i=1}^{m}Q_{T^r_{1,i},k}\times\beta^r\times F^r_{i,n}\le C^{\mathrm{rem}}_{n,k},\quad\forall k\in K,\forall n\in\bar{N}^{\mathrm{XT}} \tag{4}$$

$$\sum_{i=0}^{m}E^{s^r_i,s^r_{i+1}}_{u,v}\times\beta^r\le B^{\mathrm{rem}}_{u,v},\quad\forall u\in\bar{S},\forall v\in V=\eta(u) \tag{5}$$

$$B^r_{\cos t}=\sum_{i\in\{0,1,\cdots,m\}}\sum_{u\in\bar{S}}\sum_{v\in V=\eta(u)}E^{s^r_i,s^r_{i+1}}_{u,v}\beta^r \tag{6}$$

式(2)表示SFC请求r中的每一个VNF仅能部署在一个服务器节点上，式(3)表示用于承载$\mathrm{VNF}_i{}^r$的

服务器n_i^r必须支持该类型VNF的部署，式(4)表示部署请求r中全部的VNF所使用的各类资源数量不能超过每台服务器各类资源剩余量，式(5)表示部署请求r中全部的虚拟链路所占用的带宽不能超过各条物理链路剩余的带宽容量，式(6)表示请求r总的带宽资源消耗。

3 基于租户分类和历史信息的部署方法

3.1 租户分类策略

文献[13,18]采用的租户分类策略(Tenant Classification Strategy, TCS)，在应对侧信道攻击问题时均取得了较好的效果，但是分类方法存在的一定的复杂性和租户主观性，为此本文提出一种更为简洁高效的分类方法，分类的目的也并非严格准确地区分出恶意租户与合法租户，而是迫使恶意租户表现的与目标租户一致，否则恶意租户将无法与目标租户同区域部署。

关注每一个租户η的SFC请求总数量AMT_η以及相关请求的运行时间τ，并以此计算每个租户SFC的平均运行时间AVG_η。定义运算规则$\odot$，若$\eta=\eta^r, \eta\odot\eta^r=1$；若$\eta\neq\eta^r, \eta\odot\eta^r=0$，则

$$\mathrm{AMT}_\eta=\sum_{r\in R}\eta\odot\eta^r \tag{7}$$

$$\mathrm{AVG}_\eta=\frac{\sum_{r\in R}(\eta\odot\eta^r)\cdot\tau^r}{\mathrm{AMT}_\eta} \tag{8}$$

依据平均运行时间AVG_η将租户划分为潜在恶意租户(Potentially malicious Tenant, PT)、不确定租户(Uncertain Tenant, UT)和低风险租户(Low-risk Tenant, LT)3类，用XT表示租户所属分类。设定用于租户分类的低风险阈值(Low ThresHold, LTH)和高风险阈值(High ThresHold, HTH)，将$\mathrm{AVG}_\eta\geq\mathrm{LTH}$的租户$\eta$划分为低风险租户，将$\mathrm{LTH}>\mathrm{AVG}_\eta\geq\mathrm{HTH}$的租户$\eta$划分为不确定租户，将$\mathrm{AVG}_\eta<\mathrm{HTH}$租户$\eta$划分为潜在恶意租户，即

$$\mathrm{XT}=\begin{cases}\mathrm{LT}, & \mathrm{AVG}_\eta\geq\mathrm{LTH}\\ \mathrm{UT}, & \mathrm{LTH}>\mathrm{AVG}_\eta\geq\mathrm{HTH}\\ \mathrm{PT}, & \mathrm{AVG}_\eta<\mathrm{HTH}\end{cases} \tag{9}$$

定义$\bar{N}^{\mathrm{LT}}$, $\bar{N}^{\mathrm{UT}}$和$\bar{N}^{\mathrm{PT}}$分别表示低风险租户可部署服务器集合、不确定租户可部署服务器集合和潜在恶意租户可部署服务器集合。当SFC请求r到达时，对该请求所属租户η^r的分类信息进行更新，确定该租户所属分类$\mathrm{XT}\in\{\mathrm{LT},\mathrm{UT},\mathrm{PT}\}$，并依据租户当前的分类情况XT将其资源请求部署到相应服务器集合$\bar{N}^{\mathrm{XT}}$，其中$\bar{N}^{\mathrm{XT}}\in\{\bar{N}^{\mathrm{LT}},\bar{N}^{\mathrm{UT}},\bar{N}^{\mathrm{PT}}\}=\bar{N}$。

3.2 结合历史信息的部署策略

上述租户分类方法可显著提高恶意租户实现共存的成本，但是当恶意租户与目标租户被划分为同一类别时，恶意租户仍可通过新的资源请求不断尝试与同区域的目标租户共存，为了进一步提高共存的难度和代价，本文从限制租户VNF在域内扩散的角度出发，最小化租户所能占用的服务器数量。具体做法如下：在对租户的SFC请求进行部署时，首先根据3.1节的租户分类策略确定可部署的服务器集合$\bar{N}^{\mathrm{XT}}$，之后再结合该租户SFC请求的历史部署信息(Combine Historical Deployment Information, CHDI)，优先选择该租户η当前已经占用或近期占用过的服务器，定义该类服务器集合为$\bar{N}_\eta^{\mathrm{XT}}$，在保证服务器与VNF的相关约束条件下，最小化每个租户所能覆盖的服务器数量。此外，考虑到租户服务的可靠性问题，本文限制每台服务器可承载同一租户VNFs的最大数量为MNPT(Maximum Number of VNFs Per Tenant)，该约束条件可表示为

$$\sum_{r\in R}(\eta^r\odot\eta)\sum_{i=1}^{m}F_{i,n}^r\leq\mathrm{MNPT},\quad\forall n\in\bar{N}^{\mathrm{XT}} \tag{10}$$

4 算法设计

本文以最小化租户所能覆盖的服务器数量为目标，以式(2)—式(10)为约束条件，设计了一种基于贪婪选择的SFC部署算法，算法流程如表1所示，主要分为租户分类、VNF部署和虚拟链路部署3个阶段。

对上述算法的计算复杂度进行分析：租户分类过程的计算复杂度为$O(|R|)$，其中$|R|$为所有SFC请求的总数量；基于贪婪算法的VNF部署过程，其最大计算复杂度为$O(m(|\bar{N}_{\eta^r}^{\mathrm{XT}}|+|\bar{N}^{\mathrm{XT}}|))$；基于贪婪算法的虚拟链路部署过程，其最大计算复杂度为$O(|L^r||\bar{L}|)$。因此本算法的计算复杂度为$O(|R|+m(|\bar{N}_{\eta^r}^{\mathrm{XT}}|+|\bar{N}^{\mathrm{XT}}|)+|L^r||\bar{L}|)$，为多项式时间复杂度。

5 实验仿真

5.1 实验设置

本算法使用Python实现，采用与文献[19]一致的数据中心胖树拓扑网络结构，包含54个服务器、45个交换机和162条链路，服务器节点的计算资源为50，链路的带宽容量为50。实验中设置了8种不同类型的VNF，参考文献[20]对VNF资源需求系数进行设置，如表2所示，每个服务器节点随机选取6种作为可承载VNF。SFC请求所需处理的流量大

小从{1, 2, 3}中随机选取，所需的VNF从中随机选取4种。实验中设置了背景工作负载，使得稳定状态下服务器的平均资源使用率为50%。对恶意租户实现共存的过程做如下简化：待系统稳定后，随机在某一区域部署一条SFC请求作为目标租户，随后恶意租户采用一定的策略不断请求SFC尝试与目标租户共存，当恶意租户与目标租户有任何一个VNF共存于同一服务器时停止实验，此时判定恶意租户成功实现共存。每次实验重复100次，并对实验数据进行统计分析。

表 1　基于租户分类和历史信息的部署算法

输入：服务功能链请求信息r

输出：请求r的部署方案

(1) #**租户分类**

(2)计算平均运行时间AVG_η，确定请求所属租户η^r的分类XT；

(3)依据分类结果，确定可部署服务器集合$\bar{N}^{\text{XT}}$以及租户η^r在该区域已占用的服务器集合$\bar{N}^{\text{XT}}_{\eta^r}$；

(4) #**VNF部署**

(5) SFCdpsucc=0, nodedpsucc=0#设置部署成败状态标志；

(6) For each VNF^r_iin ψ^r#遍历SFC请求中所有的m个VNF；

(7)　筛选出$\bar{N}^{\text{XT}}, \bar{N}^{\text{XT}}_{\eta^r}$中支持该类型VNF且剩余资源足够的服务器集合$\bar{N}^{\text{XT}}_{\text{VNF}^r_i}, \bar{N}^{\text{XT}}_{\eta^r,\text{VNF}^r_i}$；

(8)　If $\bar{N}^{\text{XT}}_{\eta^r,\text{VNF}^r_i}$不为空，则从中选取剩余资源最多的服务器节点部署$\text{VNF}^r_i$；

(9)　If $\bar{N}^{\text{XT}}_{\eta^r,\text{VNF}^r_i}$为空，则从$\bar{N}^{\text{XT}}_{\text{VNF}^r_i}$中选取剩余资源最多的服务器节点部署$\text{VNF}^r_i$；

(10)　记录VNF^r_i所部属的服务器节点n^r_i，并对节点n^r_i资源余量和$\bar{N}^{\text{XT}}_{\eta^r}$进行预更新；

(11) If ψ^r中所有的VNF均找到可部署服务器节点；

(12)　nodesucc=1，并对相关服务器节点资源余量和$\bar{N}^{\text{XT}}_{\eta^r}$进行更新。

(13) #**虚拟链路部署**

(14) linkdpsucc=0#设置链路部署成败状态标志；

(15) If nodedpsucc==1；

(16)　For each $l^r_{i,i+1}$ in L^r#遍历该SFC请求中所有的虚拟链路；

(17)　　确定节点n^r_i与n^r_{i+1}之间带宽余量足够的可用链路集合$\bar{L}_{n^r_i,n^r_{i+1}}$；

(18)　　从中筛选出部署代价$B^r_{\cos t}$最小的链路集合#存在多条同等长度的链路；

(19)　　从中选取带宽资源余量最大的链路；

(20)　　记录所使用的链路，并对链路资源余量进行预更新；

(21)　If L^r中所有的虚拟链路找到可部署的物理链路；

(22)　　linkdpsucc=1，并对相关物理链路资源余量进行更新；

(23) If (nodedpsucc and linkdpsucc)==1；

(24)　SFCdpsucc=1#该SFC请求部署成功；

部分设置需要根据子实验需求进行设定，主要有：SFC请求最小生命周期(Minimum Life Cycle, MLC)和风险阈值HTH, LTH，其中HTH=$a\times$MLC, LTH=$b\times$MLC；恶意租户所能使用的账户数量情况，单账户(Single Account, SA)或多账户(Multiple Accounts, MA)；以及每个SFC请求的生命周期。用于进行对比实验的相关部署方法有：快速高效的随机部署方法Random，考虑负载均衡性的最多资源部署方法Most，基于租户分类的部署方法Most+TCS，基于历史部署信息的部署方法Most+CHDI，以及结合两者的部署方法Most+TCS+CHDI。

5.2　评价指标

从以下4个方面对比相关方法在防御共存攻击的性能。(1)覆盖的服务器数量：指的是随着恶意租户所请求SFC数量的增加，恶意租户所能占用服务器数量的情况；(2)实现共存所请求的SFC数量；(3)SFC请求累计租用时间平均值：指的是在100次重复实验下，恶意租户每次成功实现共存所请求的全部SFC生命周期时间之和的统计平均值；(4)成功实现共存的概率：指的是100次重复实验下成功实现共存所请求的SFC数量的累计分布函数。

5.3　实验结果分析

实验分为以下3个子实验，分别展示单账户攻击场景下不同方法的效果，多账户攻击策略对相关方法的影响，以及不同的时间参数所产生的影响。

(1) 单账户攻击场景下不同部署方法的效果

相关设置如下：令MLC=6, a=2, b=4, MNPT=4。恶意租户使用单账户执行攻击，在Random, Most和Most+CHDI方法下，恶意租户每个SFC请求采用最小生命周期6，以最小化攻击成本。而在Most+TCS和Most+TCS+CHDI两个采用租户分类策略的方法下，令恶意租户拥有3个属于不同租户类别的账户，为了保持账户所属分类不变以及最小化攻击成本，3个账户SFC请求的生命周期分别为6, 12和24，并依序对3个不同区域进行共存攻击尝试。

表 2　VNF资源需求系数

VNF类型	NAT	Firewall	Proxy	IDS	UD_1	UD_2	UD_3	UD_4
计算资源需求/(单位带宽)	1	2	2	6	1	2	3	4

图4展示了随租户请求的SFC数量增加，租户所能覆盖的服务器数量情况，可以看出使用CHDI策略的两种方法可显著降低租户覆盖服务器数量的增长趋势，这是由于对租户的SFC请求进行部署时，会优先选择租户已占用的或近期占用过的服务器。此外可以发现在Random方法下，租户无法覆盖全部的54个服务器，这是由于存在背景SFC负载将个别服务器资源用尽的现象，反映出Random部署算法在负载均衡性方面效果较差。图5展示了恶意租户实现共存所请求的SFC数量情况，可以看出采用CHDI策略的方法可提高恶意租户实现共存所请求的SFC平均数量，其中Most+CHDI方法最为优异。图6展示了100次重复试验下，成功实现共存的概率，可以看出在Random部署方法下，大约需要13条SFC请求一定可与目标租户实现共存，在Most和Most+TCS部署方法下，大约需要8条，而在Most+CHDI和Most+TCS+CHDI部署方法下，大约需要23条，由此可见本文所提Most+TCS+CHDI部署方法提高了恶意租户实现共存的难度和成本。

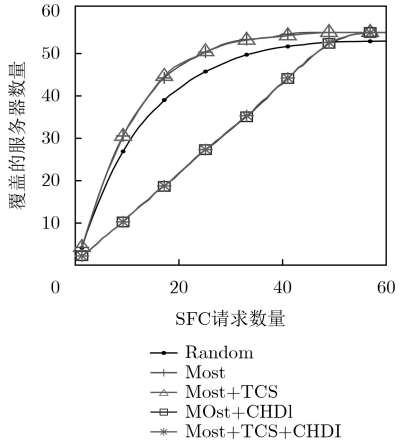

图 4 覆盖服务器数量的变化趋势

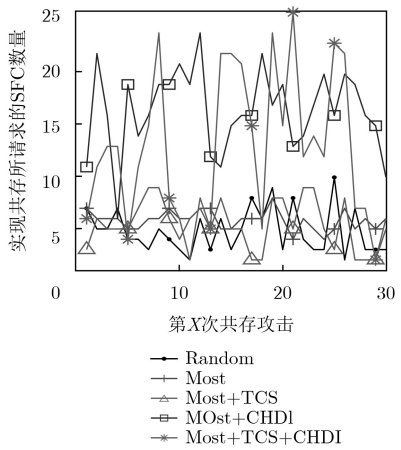

图 5 实现共存所请求的SFC数量情况

(2) 多账户攻击策略对相关部署方法的影响

为了验证多账户攻击策略的影响，采用子实验(1)的相关设置重新运行实验，参数MLC, HTH, LTH和MNPT均不变。恶意租户可采用单账户或多账户进行攻击，由于Random和Most方法平等的对待所有账户请求，多账户攻击策略对其无影响，不再列出。为了最大化多账户场景下的攻击效率和最小化攻击成本，在Most+CHDI方法下，恶意租户每个账户仅请求1条生命周期为6的SFC。而在Most+TCS和Most+TCS+CHDI方法下，由于租户分类策略的存在，每个新账户请求的第1条SFC只能部署在潜在恶意租户区域，第2条SFC请求才能根据分类结果部署到不确定租户区域或低风险租户区域。为了最小化共存攻击成本，恶意租户在潜在恶意租户区域无需进行额外的共存尝试，因此每个新账户请求的两条SFC生命周期为12或者24，攻击时先对不确定租户区域进行共存尝试，同时检查第1条SFC在潜在恶意租户区域是否实现共存，覆盖该区域全部服务器仍未实现共存时，再对低风险租户区域进行共存尝试。

图7展示了在单账户(SA)攻击策略和多账户(MA)攻击策略下，租户所能覆盖的服务器数量情况，可以看出MA对采用CHDI策略的部署方法影响很大，在MA攻击策略下，Most+CHDI和Most+TCS+CHDI的服务器覆盖趋势分别退化为Most和Most+TCS，这是由于CHDI策略无法分辨同一恶意租户的不同新账户。图8则展示了在MA攻击策略下，恶意租户实现共存所请求的SFC数量变化情况，MA攻击策略显著减少了恶意租户在Most+CHDI和Most+TCS+CHDI部署方法下需要请求的SFC数量。图9则展示了MA攻击策略对共存概率的影响，可以看出MA攻击策略显著降低了恶意租户在Most+CHDI和Most+TCS+CHDI部署方法下实

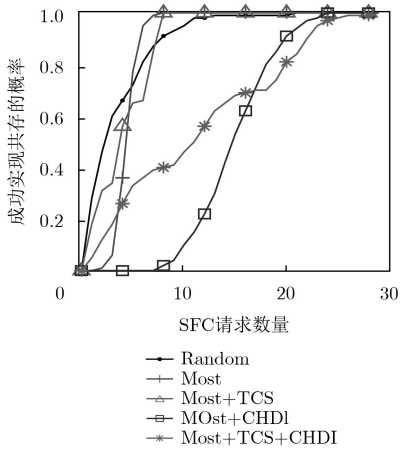

图 6 成功实现共存的概率

现共存的难度。图10展示了攻击者采用SA攻击策略时成功实现共存所租用资源累计时间的统计平均值，而恶意租户的成本正比于租用资源时间的长度，可以看出TCS和CHDI两个策略均能很好地提高恶意租户实现共存所付出的租用资源成本，将两者结合的Most+TCS+CHDI方法能显著提高恶意租户的攻击成本。图11则展示了MA攻击策略对恶意租户实现共存所租用资源累计时间的影响，MA攻击策略提高了Most+TCS部署方法下的攻击成本，这是由于每个新账户的第1条SFC请求只能部署在潜在恶意租户区域，然而显著降低了Most+CHDI和Most+TCS+CHDI部署方法下的攻击成本。从图7—图11可以看出多账户攻击策略对本文所提部署方法性能影响较大，为了保证Most+TCS+CHDI部署方法的有效性，建议云服务提供商严格审批租户新账户的申请，以避免潜在恶意租户掌握大量可用账户。

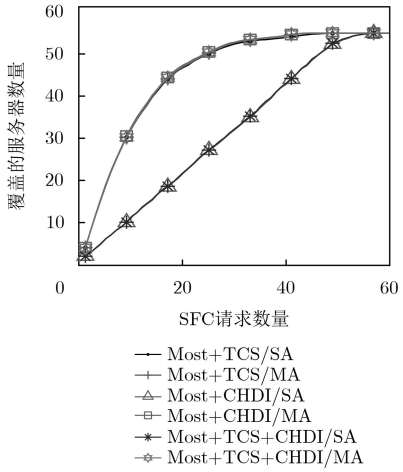

图 7 MA对覆盖服务器数量的影响

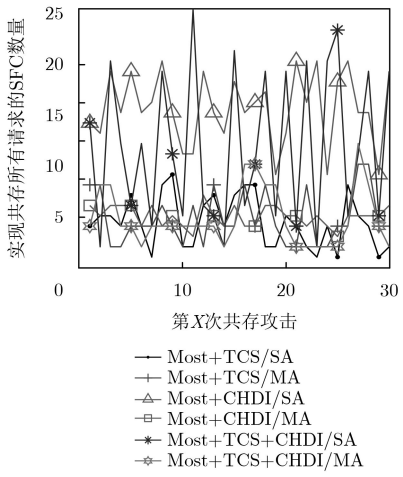

图 8 MA对实现共存所请求SFC数量的影响

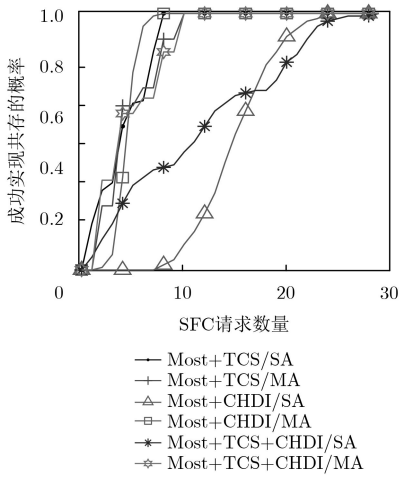

图 9 MA对共存概率的影响

(3) 不同的时间参数所产生的影响

为了验证相关时间参数对部署方法性能的影响，采用以下3组参数分别进行重复实验，第1组：MLC=3,a=2, b=4；第2组：MLC=6, a=2,b=4；第3组：MLC=6, a=4, b=8。此外MNPT=4，恶意租户仅使用单账户攻击策略。时间参数的改变仅对实现共存所租用资源累计时间的平均值有影响，而对其他3个评价指标无影响。从图12可以看出最小生命周

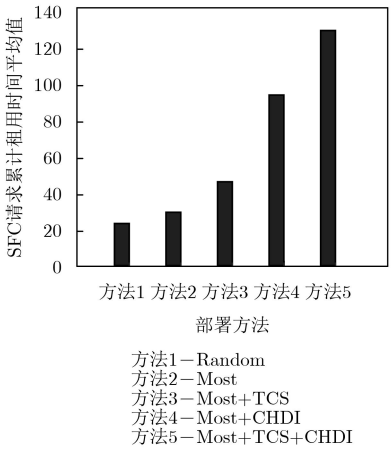

图 10 实现共存所租用资源累计时间的平均值图

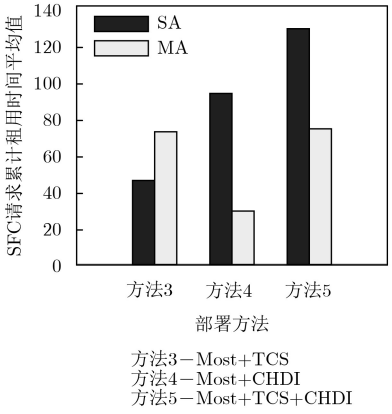

图 11 MA对租用资源累计时间平均值的影响

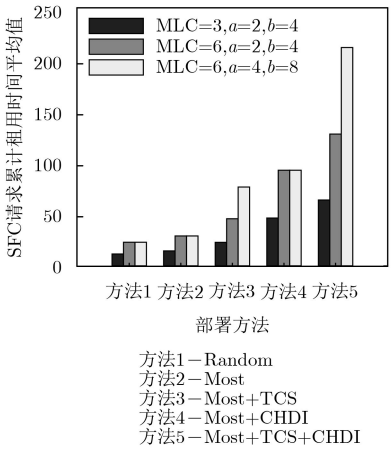

图 12 时间参数对租用资源累计时间平均值的影响

期MLC值的增大，可提高恶意租户在所有部署方法下实现共存所租用资源的成本，而风险阈值HTH和LTH的增大可进一步提高采用TCS策略部署方法下的攻击成本。

6 结论

本文分析了多租户环境下现有SFC部署方法在应对侧信道攻击时存在的不足，提出了一种抗侧信道攻击的服务功能链部署方法。引入基于时间均值的租户分类策略，实现对租户的分类和分域部署，结合历史部署信息，优先选择租户正在使用或近期使用过的服务器，并以最小化租户所能覆盖的服务器数量为优化目标建立了部署模型，设计了基于贪婪选择的部署算法。最后通过实验，验证了单账户攻击场景下本文所提方法在提高恶意租户共存难度和成本的优异性能，展示了多账户攻击策略对相关方法的影响，并提出了合理化的建议，此外展示了相关时间参数对所提方法防御效果的影响。

参 考 文 献

[1] MEDHAT A M, TALEB T, ELMANGOUSH A, *et al*. Service function chaining in next generation networks: State of the art and research challenges[J]. *IEEE Communications Magazine*, 2017, 55(2): 216–223. doi: 10.1109/MCOM.2016.1600219RP.

[2] 周伟林, 杨芫, 徐明伟. 网络功能虚拟化技术研究综述[J]. 计算机研究与发展, 2018, 55(4): 675–688. doi: 10.7544/issn1000-1239.2018.20170937.

ZHOU Weilin, YANG Yuan, and XU Mingwei. Network function virtualization technology research[J]. *Journal of Computer Research and Development*, 2018, 55(4): 675–688. doi: 10.7544/issn1000-1239.2018.20170937.

[3] BO Yi, WANG Xingwei, LI Keqin, *et al*. A comprehensive survey of Network Function Virtualization[J]. *Computer Networks*, 2018, 133: 212–262. doi: 10.1016/j.comnet.2018.01.021.

[4] 袁泉, 汤红波, 黄开枝, 等. 基于Q-learning算法的vEPC虚拟网络功能部署方法[J]. 通信学报, 2017, 38(8): 172–182. doi: 10.11959/j.issn.1000-436x.2017173.

YUAN Quan, TANG Hongbo, HUANG Kaizhi, *et al*. Deployment method for vEPC virtualized network function via Q-learning[J]. *Journal on Communications*, 2017, 38(8): 172–182. doi: 10.11959/j.issn.1000-436x.2017173.

[5] GHAZNAVI M, KHAN A, SHAHRIAR N, *et al*. Elastic virtual network function placement[C]. Proceedings of the IEEE 4th International Conference on Cloud Networking, Niagara Falls, Canada, 2015: 1–7. doi: 10.1109/CloudNet.2015.7335318.

[6] MIJUMBI R, HASIJA S, DAVY S, *et al*. Topology-aware prediction of virtual network function resource requirements[J]. *IEEE Transactions on Network and Service Management*, 2017, 14(1): 106–120. doi: 10.1109/TNSM.2017.2666781.

[7] 陈卓, 冯钢, 刘蓓, 等. 运营商网络中面向资源碎片优化的网络服务链构建策略[J]. 电子与信息学报, 2018, 40(4): 763–769. doi: 10.11999/JEIT170641.

CHEN Zhuo, FENG Gang, LIU Bei, *et al*. Construction policy of network service chain oriented to resource fragmentation optimization in operator network[J]. *Journal of Electronics & Information Technology*, 2018, 40(4): 763–769. doi: 10.11999/JEIT170641.

[8] QU Long, ASSI C, SHABAN K, *et al*. A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks[J]. *IEEE Transactions on Network and Service Management*, 2017, 14(3): 554–568. doi: 10.1109/TNSM.2017.2723090.

[9] FIROOZJAEI M D, JEONG J, KO H, *et al*. Security challenges with network functions virtualization[J]. *Future Generation Computer Systems*, 2017, 67: 315–324. doi: 10.1016/j.future.2016.07.002.

[10] 梁鑫, 桂小林, 戴慧珺, 等. 云环境中跨虚拟机的Cache侧信道攻击技术研究[J]. 计算机学报, 2017, 40(2): 317–336. doi: 10.11897/SP.J.1016.2017.00317.

LIANG Xin, GUI Xiaolin, DAI Huijun, *et al*. Cross-VM cache side channel attacks in cloud: A survey[J]. *Chinese Journal of Computers*, 2017, 40(2): 317–336. doi: 10.11897/SP.J.1016.2017.00317.

[11] ZHANG Xu, WANG Haining, and WU Zhenyu. A measurement study on co-residence threat inside the cloud[C]. Proceedings of the 24th USENIX Conference on Security Symposium, Washington, USA, 2015: 929–944.

[12] ATYA A O F, QIAN Zhiyun, KRISHNAMURTHY S V, *et al*. Malicious co-residency on the cloud: Attacks and defense[C]. Proceedings of IEEE Conference on Computer

Communications, Atlanta, USA, 2017: 1–9. doi: 10.1109/INFOCOM.2017.8056951.

[13] 赵硕, 季新生, 毛宇星, 等. 基于安全等级的虚拟机动态迁移方法[J]. 通信学报, 2017, 38(7): 165–174. doi: 10.11959/j.issn.1000-436x.2017091.
ZHAO Shuo, JI Xinsheng, MAO Yuxing, *et al*. Research on dynamic migration of virtual machine based on security level[J]. *Journal on Communications*, 2017, 38(7): 165–174. doi: 10.11959/j.issn.1000-436x.2017091.

[14] ZHANG Tianwei, ZHANG Yinqian, and LEE R B. CloudRadar: A real-time side-channel attack detection system in clouds[C]. Proceedings of 19th International Symposium on Research in Attacks, Intrusions, and Defenses, Paris, France, 2016: 118–140. doi: 10.1007/978-3-319-45719-2_6.

[15] NOSHY M, IBRAHIM A, and ALI H A. Optimization of live virtual machine migration in cloud computing: A survey and future directions[J]. *Journal of Network and Computer Applications*, 2018, 110: 1–10. doi: 10.1016/j.jnca.2018.03.002.

[16] LIU Shuhao, CAI Zhiping, XU Hong, *et al*. Towards security-aware virtual network embedding[J]. *Computer Networks*, 2015, 91: 151–163. doi: 10.1016/j.comnet.2015.08.014.

[17] HAN Yi, CHAN J, ALPCAN T, *et al*. Using virtual machine allocation policies to defend against co-resident attacks in cloud computing[J]. *IEEE Transactions on Dependable and Secure Computing*, 2017, 14(1): 95–108. doi: 10.1109/TDSC.2015.2429132.

[18] HAN Yi, ALPCAN T, CHAN J, *et al*. A game theoretical approach to defend against co-resident attacks in cloud computing: Preventing co-residence using semi-supervised learning[J]. *IEEE Transactions on Information Forensics and Security*, 2016, 11(3): 556–570. doi: 10.1109/TIFS.2015.2505680.

[19] LI Defang, HONG Peilin, XUE Kaiping, *et al*. Virtual network function placement considering resource optimization and SFC requests in cloud datacenter[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2018, 29(7): 1664–1677. doi: 10.1109/TPDS.2018.2802518.

[20] BARI F, CHOWDHURY S R, AHMED R, *et al*. Orchestrating virtualized network functions[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(4): 725–739. doi: 10.1109/TNSM.2016.2569020.

伊　鹏：男，1977年生，博士，研究员，研究方向为网络空间安全.
谢记超：男，1993年生，硕士生，研究方向为网络安全.
张　震：男，1985年生，博士，讲师，研究方向为新型网络体系结构.
谷允捷：男，1994年生，博士生，研究方向为网络功能虚拟化.
赵　丹：女，1992年生，助理工程师，研究方向为新一代信息通信网.