计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2018.20180447
55(10):2256-2268,2018

面向SDN的脆弱性扩散形式化建模与扩散因素分析

王 健[1] 赵国生[2] 赵中楠[1] 李 可[1]
[1](哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)
[2](哈尔滨师范大学计算机科学与信息工程学院 哈尔滨 150025)
(wangjianlydia@163.com)

Formal Modeling and Factor Analysis for Vulnerability Propagation Oriented to SDN

Wang Jian[1], Zhao Guosheng[2], Zhao Zhongnan[1], and Li Ke[1]
[1](*College of Computer Science and Technology, Harbin University of Science and Technology, Harbin* 150080)
[2](*College of Computer Science and Information Engineering, Harbin Normal University, Harbin* 150025)

Abstract Software defined network (SDN) is one of the most popular network technologies nowadays. SDN decouples the traditional control plane from the forwarding plane, resulting in many new security and management issues while performing centralized control. Meanwhile, the types of vulnerabilities are diverse in each layer and north-south trending interfaces of SDN, and the spread trend is quite different. Aiming at the effect of vulnerability propagation in/between layers of SDN as well as its suppression strategy, a formal model of vulnerability propagation for SDN based on Bio-PEPA is proposed in this paper. First of all, the basic syntax of Bio-PEPA is discussed, and its applicability to SDN with obvious hierarchical structure and the vulnerability propagation process with dynamic characteristic is illustrated. Then, the vulnerabilities existing in each layer of SDN are explored and modeled in terms of layers. Besides, by constructing a formal model for the process of vulnerability propagation in/between layers of SDN, the mechanism of vulnerability propagation is analyzed in two levels, horizontal (in layers) and vertical (between layers). In this way, the vulnerability propagation of SDN can be better suppressed. Finally, the simulation results show that the vulnerability propagation of SDN can be effectively retained by reducing the connection conversion rate, improving the detection conversion rate and repairing conversion rate. Our works provide a reference for the law of vulnerability propagation of SDN, so as to improve the security of SDN.

Key words software defined network (SDN); Bio-PEPA; formal modeling; vulnerability propagation; suppression strategy

摘 要 SDN将传统网络控制面与转发面解耦,在实施集中化管控的同时引入诸多新的安全和管理问题.脆弱点类型在SDN各层及南北向接口存在差异性,且传播趋势不同.针对脆弱性在SDN层内及

收稿日期:2018-06-07;**修回日期**:2018-08-03
基金项目:国家自然科学基金项目(61403109,61202458);高等学校博士学科点专项科研基金项目(20112303120007);黑龙江省自然科学基金项目(F2017021);黑龙江省教育厅科研基金项目(12541169);哈尔滨市科技创新人才研究专项资金项目(2016RAQXJ036)
This work was supported by the National Natural Science Foundation of China (61403109, 61202458), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20112303120007), the Natural Science Foundation of Heilongjiang Province (F2017021), the Scientific Research Fund of Heilongjiang Provincial Education Department (12541169), and the Specialized Research Fund for Scientific and Technological Innovation Talents of Harbin (2016RAQXJ036).

层间的扩散效果及抑制策略问题,提出了一种基于 Bio-PEPA 的 SDN 脆弱性扩散形式化模型.1)对 Bio-PEPA 基础语义进行讨论,阐明其适用于具有明显分层架构的 SDN 及具有动态性的脆弱性扩散过程;2)探讨 SDN 中各层存在的脆弱性问题,并对 SDN 中存在的脆弱性以层为单位进行建模,通过对 SDN 层内及层间脆弱性扩散过程构建形式化模型,进而分析 SDN 内脆弱性在水平(层内)及垂直(层间)2 个维度内的扩散机理,从而更好地抑制脆弱性在 SDN 内的扩散;3)通过仿真实验得出可以通过降低连接转化率、提升检测转化率及修复转化率来有效抑制 SDN 的脆弱性扩散.

关键词　软件定义网络;Bio-PEPA;形式化建模;脆弱性扩散;抑制策略

中图法分类号　TP393

脆弱性是引发网络安全问题的重要因素,始终是网络安全领域研究的热点之一.随着人们对网络需求的日益提高,网络架构也在发生着重大变化.一种新型网络架构——软件定义网络(software defined networking, SDN)的出现[1],更好地满足了人们的需求.它采用控制和转发相分离的体系架构,使网络管理员可以通过软件实现任意的网络控制逻辑,而不需要对网络设备本身进行修改,具备极强的灵活性与开放性.在 SDN 的灵活性与开放性为网络管理员及用户带来便利的同时,也使网络引入了更大的安全风险及脆弱性.从控制器通过南向接口传播蠕虫病毒给底层交换机[2],由交换机对控制器发起 DDoS 攻击[3],应用层恶意非法用户篡改身份占用 SDN 带宽[4]等,都会造成 SDN 的瘫痪故障.因此,SDN 的安全问题成为制约其普及应用的障碍.

目前,针对 SDN 脆弱性研究仍处于初级阶段.文献[5]基于 SDN 的可编程性提出了一种防火墙应用,该应用可以不借助任何硬件即可工作,解决了传统网络中不同防火墙间规则冲突等问题.文献[6]面向云计算,提出了一种入侵检测和控制器的联动机制相融合的入侵防御方法,该方法解决了传统的入侵防御系统由于串联在网络环境中且处理能力有限所造成的网络拥塞问题.文献[7-10]结合 SDN 集中控制与网络虚拟化的特性,利用 SDN 网络解决在传统网络中一直困扰网络管理者的 DDos 攻击问题.以上研究者的关注点在通过 SDN 解决某些在传统网络中不能妥善解决的问题,而没有关注到 SDN 自身的安全问题.文献[11]根据 STRIDE 威胁影响模型对 SDN 所面临的脆弱性问题及解决策略进行研究,但是并没有考虑到 SDN 内脆弱性的关系及扩散差异问题.文献[12]则研究 SDN 内交换机、控制器、通信信道等重要的网络元件所具有的脆弱性问题及解决对策,虽然该方法考虑到了 SDN 内各网络元件之间的差异性,但是并没有考虑脆弱性在 SDN 层内及层间的动态扩散问题.此外,一些研究者则从局部,比如以 SDN 架构某一层[13-16]或接口[17-18]为切入点来对 SDN 存在的脆弱性问题进行分析,还有一些研究者则从整体架构层面[19-22]对 SDN 存在的脆弱性问题进行分析.这些研究都是从局部或外部层面上对 SDN 面临的脆弱性问题及解决策略进行分析,并没有对脆弱性的产生及扩散的内部因素等展开深入研究,难以有效指导 SDN 的安全设计.文献[23]针对 SDN 流表更新一致性的问题及措施展开研究.文献[24]针对 SDN 开放性较高及安全问题出现概率较大的问题,对 SDN 的接入控制进行研究.上述研究面向 SDN 中的某一具体安全问题进行了探讨,但是这些研究结果因针对性较强,不能迁移到其他类型安全问题的分析,即上述研究并没有系统地分析安全性问题产生的内在原因或者说脆弱性扩散/传播的内在因素.

综上所述,研究者更多关注于基于 SDN 增强传统网络的安全性,虽然已有部分研究为引发 SDN 安全性问题的内因(脆弱性)及其预测抑制提供了重要的研究基础,但由于 SDN 分层架构、集中控制及开放灵活等特性而引入的脆弱性扩散分析仍具有一定局限性:1)已有研究仅面向 SDN 不同组件,已有安全模型仅针对某一类脆弱点引发的安全问题,各有其应用与限制范围,未突破面向 SDN 脆弱性分析的简单性和局限性;2)目前对于 SDN 脆弱性的分析,普遍基于三层两接口的 SDN 架构进行分类描述,但在实际 SDN 安全对抗中,往往是采用跨层模式,已有研究缺乏对 SDN 脆弱性及其扩散过程的完整描述与分析能力;3)虽然已有少量研究涉及 SDN 内脆弱性的关系及扩散差异性,但并没有考虑脆弱性在 SDN 层内及层间的动态扩散问题,对于脆弱性系统化地建模与动态扩散过程分析,尚无实际解决方案.

针对已有研究不足,本文借助形式化建模语言 Bio-PEPA,对 SDN 脆弱性扩散问题以层为单位进行建模,通过 SDN 层内及层间脆弱性扩散的形式化模型,研究 SDN 内脆弱性在水平(层内)及垂直(层间)2 个维度内的扩散本质,抑制脆弱性在 SDN 内的扩散. 本文的主要贡献有 5 个方面:

1) 系统分析了 SDN 内不同组件及 SDN 各层/接口存在的脆弱性及内在安全交互问题;

2) 基于 Bio-PEPA 的层次化特性,构建 SDN 脆弱性扩散过程的形式化模型,提高了对 SDN 组件间逻辑关系及动态演化过程的描述能力;

3) 考虑了脆弱性在 SDN 架构层内及层间 2 个维度上的扩散机理,适合于 SDN 脆弱性扩散的特点,以此为基础系统地分析 SDN 脆弱性的扩散规律;

4) 借助 Eclipse Bio-PEPA Plugin 工具,分别采用 ODEs 和 Gillespie's Stochastic 方法对 SDN 脆弱性模型进行分析,过程透明客观,验证其合理性;

5) 实现对 SDN 脆弱性的实时监控,进一步对扩散因素及趋势进行分析与预测,使面向 SDN 脆弱性的管理从被动变为主动,并制定有效的抑制策略,提升 SDN 安全性.

1 Bio-PEPA 基础语义

Bio-PEPA 是为生化系统的建模和分析而开发的一种形式化语言. 它主要包括描述物种行为的物种组件(S)和描述物种组件间交互的模型组件(P). 2 种组件描述为[25]:

$$S::=(\alpha,k)\ op\ S\mid S+S\mid C, \tag{1}$$

其中,$op=\downarrow\mid\uparrow\mid\oplus\mid\ominus\mid\odot$;

$$P::=P\underset{G}{\bowtie}P\mid S(x). \tag{2}$$

在前缀$(\alpha,k)\ op\ S$ 中,k 表示在执行动作 α 时物种 S 的化学计量数;前缀组合"op"代表反应中物种 S 所起的作用,具体包括反应物($\downarrow$)、生成物($\uparrow$)、促进剂($\oplus$)、抑制剂($\ominus$)和通用修改($\odot$);操作符"+"代表选择操作;C 代表常量. 操作符 $\bowtie$ 代表合作操作,G 是合作过程中必须同步执行的动作集合,$\underset{*}{\bowtie}$ 代表合作过程中所需要的全部动作必须同步;$S(x)$中的参数 $x\in\mathbb{R}$,代表物种的初始数量.

此外,还需对 Bio-PEPA 中的位置(locaction)进行说明,位置构成了静态层次结构,适用于 SDN 这种具有明显层次结构的性能分析. 符号 $S@L$ 表示由组件 S 描述的物种位于位置 L 中. 每个位置由"$L:s\ unit, kind$"表示. 其中 L 是位置的名称,是唯一的;s 表示大小;$unit$ 是与之相关的度量单位;$kind$ 代表种类.

为收集系统动态信息,Bio-PEPA 用函数速率 f_α 与动作 α 相关联. 这个函数代表了关联反应的动力学规律,即反应速率. 对于函数速率的定义,Bio-PEPA 考虑具有简单运算的数学表达式和包含常数参数和分量的运算符,所有动力学法则都可按照这种方式进行定义. Bio-PEPA 还包括一些预定义函数来表达最常用的动力学法则. 因此,函数速率可以通过数学操作符(sk)和预定义函数(sk_2)进行定义:

$$f\_rate::=f_\alpha(\bar{k},\bar{C})=sk\mid f_\alpha(\bar{k})=sk_2, \tag{3}$$

$$sk::=\mathrm{int}\mid\mathrm{float}\mid\mathrm{vachar}\mid sk+sk\mid sk\times sk\mid sk/sk\mid sk-sk\mid sk^{sk}\mid\exp(x)\mid\log(sk)\mid\sin(sk)\mid\cos(sk), \tag{4}$$

$$sk_2::=f_{MA}(sk)\mid f_{MM}(sk,sk)\mid f_H(sk,sk,\mathrm{int}). \tag{5}$$

本文涉及的函数速率为 mass-action 类型,即 $f\_rate::=f_{MA}(sk)$,该类型反应[①]的反应速率仅与反应物浓度相关,即对于任意反应 α,函数速率(即反应速率)$f_\alpha::=f_{MA}(r_\alpha)=r_\alpha\times$反应物浓度,其中 r 为反应转化率参数.

Bio-PEPA 可以通过多种方法进行求解,如随机模拟、连续时间 Markov 链(CTMC)和常微分方程(ODEs)的方式进行求解,同时 Bio-PEPA 语言有一整套自动处理 Bio-PEPA 模型的软件工具支持模型求解. 求解 Bio-PEPA 主要是为了获得各组件的近似稳态概率,继续对系统做进一步的分析. 所谓近似稳态概率是指模型稳定后,各种类组件的数量占所有组件总量的比例. 本文用这个性能参数来评价 SDN 网络的脆弱性扩散结果.

2 SDN 脆弱性分析

SDN 是一种集中控制的网络架构,其空间结构主要包括 3 个层次:基础设施层(infra-structure layer)、控制层(control layer)和应用层(application layer). 该架构将传统网络设备的数据平面和控制平面分离,底层硬件的复杂度得到大大简化. 主张通过集中式的控制器平台实现网络的控制,由于软件

① 文内不再区分反应与动作,因"动作"为 PEPA 中的概念,"反应"为生化网络环境下与 PEPA 中动作相等价的概念,Bio-PEPA 为 PEPA 基础上融合生化网络概念发展而来的新型形式化语言,因此在 Bio-PEPA 中二者等价,无区分的必要.

控制器的数量远远小于传统网络中网络设备的数量,配置和检查工作均被大大简化;另外,由于软件控制器一般由第三方实现,使得控制平面脱离了硬件厂商的限制,修改和添加新特性只需在软件控制器上修改或者添加应用即可,相比于传统网络简单很多.虽然 SDN 为网络引进了控制面和数据层分离,具有简化底层硬件实现、简化网络配置过程以及向上层应用提供网络全局视图等优点,但是,作为一个尚在起步阶段的体系结构,SDN 在简化网络管理、缩短创新周期的同时,也引入了不可低估的安全威胁.

基础设施层位于 SDN 网络架构的最底层,由路由器、物理/虚拟交换机、接入点等网络设备组成.该层主要负责数据的处理、转发和状态收集,底层硬件的复杂度得到大大简化.该层设备是可访问的,通过南向接口,由控制器管理,对控制器下发的流规则完全信任.该层存在的主要脆弱性问题包括:恶意/虚假流规则注入、DDoS/DoS 攻击、数据泄露、非法访问、身份假冒和交换机自身的配置缺陷等.此外,基础设施层还可能面临着由虚假控制器的无序控制指令导致的交换机流表混乱等威胁.

控制层包含一个特别的网络组件控制器,按照逻辑集中物理分散的原则进行分布.控制器实质上是一个软件平台,主要负责 SDN 网络内流规则的建立和终止.SDN 控制器为底层网络提供可编程接口,其整体管理功能完全托付在 SDN 控制器,这有利于整个网络的集中管理.同样,这也成为 SDN 的致命弱点.当攻击者入侵控制器,将有能力控制整个网络,进而将给 SDN 带来难以估量的危害.控制层的典型脆弱性问题是集中式管控带来的控制器单点故障问题,如针对控制器的 DoS/DDoS 攻击及控制器在逻辑/物理上遭到破坏而导致的拒绝服务等故障.此外,控制层还面临非法访问、身份假冒、恶意/虚假流规则注入以及控制器自身的配置缺陷等脆弱性问题.

应用层负责提供一系列的服务和应用,例如入侵检测系统、入侵防护系统、深度包检测、负载均衡、安全监测、接入控制等 Openflow 应用程序、安全服务类应用程序、第三方应用程序.一些流规则还将由上述应用程序制定,并由控制器下发至相关的交换机和网络设备.目前,针对应用程序自身的安全型保护机制并不健全,由于基础设施层的各种交换机和网络设备对控制器下发的流规则完全信任,且不假思索的执行,一旦这些参与制定流规则的应用受到篡改和攻击,将给 SDN 带来难以预估的危害.因此,应用层面临的脆弱性主要包括:应用程序隐含的恶意代码、应用程序代码的恶意篡改、身份假冒、非法访问以及应用程序自身的配置缺陷等.

针对以上 SDN 三层结构,将 SDN 结构及各层包含的脆弱性问题总结如图 1 所示:

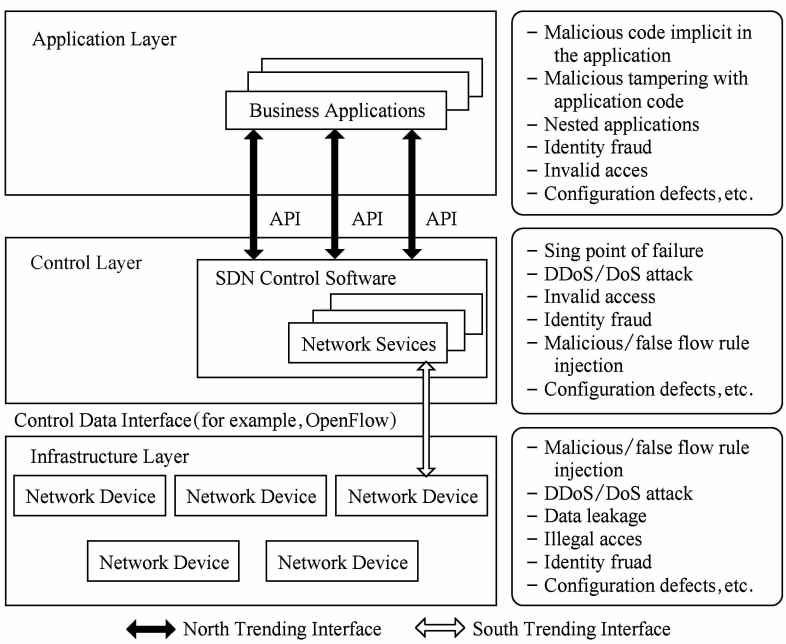


Fig. 1　SDN architecture and vulnerability analysis

图 1　SDN 架构及脆弱性分析

3 SDN 脆弱性扩散模型

3.1 问题描述

脆弱状态是指一个已授权状态,且由该状态经过已授权的转移方式可以到达未授权状态,而脆弱性是指脆弱状态区别于非脆弱状态的特征,通常将这种具有脆弱性的组件称为脆弱点.造成系统脆弱性的原因具有多种类型,传统上主要包括操作系统的设计缺陷、软件 bug 等.此外,连接性、可疑用户输入、不合理信任关系、间谍或者恶意软件等因素也会引入额外的脆弱点.

根据 SDN 结构及对 SDN 脆弱性的分析,我们分别以层为单位,通过对底层网络设备、控制器、上层应用 3 种类型组件在垂直及水平 2 个维度内的运行变化,来分析脆弱性层内和层间的扩散情况.为了描述脆弱性扩散的过程,需要定义组件的运行状态,因此将 SDN 组件运行状态抽象为以下 5 种.

P:该状态表示节点具有潜在的脆弱性,在其他节点的作用下,可能被渗透而具有脆弱性.

U:该状态表示节点已经表现出脆弱性,但尚未被检测出.

D:该状态表示节点已经表现出脆弱性,且已经被检测出.

F:该状态表示节点中的脆弱性未得到及时处理,节点出现故障或失效.

M:该状态表示经减缓、修复等操作,节点脆弱性得到减缓.

其中,{U,D,F}归类为脆弱态集合,网络的脆弱性主要由这 3 种状态的组件数量影响,因此本文用脆弱态组件的稳态概率(即脆弱态组件占所有组件的比例)来衡量 SDN 脆弱性的扩散结果.

Bio-PEPA 起初是针对生化网络建模提出的,SDN 网络与生化网络有相似之处,它的静态分层等特性在研究 SDN 脆弱性的扩散建模中仍然很具适用性,但需要针对应用环境进行相应变换.在 SDN 网络中,依据 Bio-PEPA 定义的语法语义,可进行基本抽象:

1) 将 SDN 中的 3 个层(Layer)抽象为 Bio-PEPA 模型中的位置,设置如下:

$$location_1: size=1, type=compartment;$$
$$location_2: size=1, type=compartment;$$
$$location_3: size=1, type=compartment.$$

2) 5 种 SDN 组件状态对应于 Bio-PEPA 模型中的物种组件(S),P,U,D,F,M 状态的节点,在 Bio-PEPA 模型中分别被称作潜在态组件、未检测态组件、检测态组件、故障态组件和修复态组件.

3) SDN 网络中,任意 2 种组件之间的一种交互变化,对应于 Bio-PEPA 模型中的一种动作类型 α,其动态性对应于 Bio-PEPA 模型中的函数速率 f_α.

SDN 组件之间的交互变化可抽象为 5 个类型的反应,简述为连接反应、检测反应、异常反应、修复反应和缺陷反应,具体内容将在 3.2,3.3 节详述.

$$Connetion^* = \{Connection1\_i, Connection2\_i, Connection3\_i, Connection1\_i\_j, Connection2\_i\_j, Connection3\_i\_j\};$$
$$Detection^* = \{Detection\_i\};$$
$$Abnormity^* = \{Abnormity1\_i, Abnormity2\_i\};$$
$$Mitigation^* = \{Mitigation1\_i, Mitigation2\_i\};$$
$$Defect^* = \{Defect\_i\}.$$

4) SDN 脆弱性扩散模型由 5 种组件间的合作操作进行定义.

若模型内有 U,D,F 三种状态的脆弱点存在,且各反应转化率参数不为 0,即会触发脆弱性扩散.连接反应转化率使得 U,D,F 状态下的脆弱点与 P 状态的脆弱点反应,使 P 状态组件表现出脆弱性,完成脆弱性由 U,D,F 向 P 的扩散.U,D 组件由于异常反应转化率不为 0,脆弱态会近一步演化为 F 状态.

3.2 层内脆弱性扩散形式化模型

层内扩散是指同层内组件间脆弱性的扩散.例如,在基础设施层,攻击者可能会尝试将一些恶意的流规则插入到流表中,导致整个安全机制受到损害.此外,大量无用的流规则即流表洪泛攻击,也会引发基础设施层主机无法存贮正常的流规则,导致脆弱性在基础设施层内进行扩散.在控制层,多控制器协同工作时,若网络中存在延迟且得不到妥善处理时,就难以保证转发规则的一致性,进而造成网络面临环路或者冲突.同时,东西向接口的不安全性,使得一些脆弱性在控制器间进行通信时得以迅速扩散.在应用层,由于缺乏身份认证授权等管理,使得非法应用可以接入到网络中,可以避开安全规则的审计.此外,多个应用共享数据流,同时处理一个数据流的时候就会造成规则的冲突,导致脆弱性在应用层内进行扩散.

为便于描述,将基础设施层、控制层、应用层分别记为:$Layer\_1, Layer\_2, Layer\_3$,即 $Layer\_i \in \{Layer\_1, Layer\_2, Layer\_3\}$.因此,基于 Bio-

PEPA 定义的层内脆弱性扩散规则描述如下:

[horizon]

1) ⟨*Connection*1_*i*⟩

U@*Layer_i*+P@*Layer_i*→U:P@*Layer_i*+U@*Layer_i*;

2) ⟨*Conversion*1_*i*⟩

U:P@*Layer_i*→U@*Layer_i*;

3) ⟨*Connection*2_*i*⟩

D@*Layer_i*+P@*Layer_i*→D:P@*Layer_i*+D@*Layer_i*;

4) ⟨*Conversion*2_*i*⟩

D:P@*Layer_i*→U@*Layer_i*;

5) ⟨*Connection*3_*i*⟩

F@*Layer_i*+P@*Layer_i*→F:P@*Layer_i*+F@*Layer_i*;

6) ⟨*Conversion*3_*i*⟩

F:P@*Layer_i*→U@*Layer_i*;

7) ⟨*Detection_i*⟩

U@*Layer_i*→D@*Layer_i*;

8) ⟨*Abnormity*1_*i*⟩

U@*Layer_i*→F@*Layer_i*;

9) ⟨*Abnormity*2_*i*⟩

D@*Layer_i*→F@*Layer_i*;

10) ⟨*Mitigation*1_*i*⟩

D@*Layer_i*→M@*Layer_i*;

11) ⟨*Mitigation*2_*i*⟩

F@*Layer_i*→M@*Layer_i*;

12) ⟨*Defect_i*⟩

M@*Layer_i*→P@*Layer_i*.

上述规则中,⟨*Connection*1_*i*⟩,⟨*Connection*2_*i*⟩,⟨*Connection*3_*i*⟩描述在基础设施层中,当攻击者攻击基础设施层主机而进入网络,使主机感染脆弱性并表现为 U@*Layer*_1,D@*Layer*_1 或者 F@*Layer*_1 状态,进而假冒主机的身份向与之相连的交换机发送数据包,请求与具有潜在脆弱性的表现为 P@*Layer*_1 状态的另一主机进行通信,交换机查询自身的流表,并按照流表内的信息进行转发完成被攻击主机与 P@*Layer*_1 状态主机的通信,同时脆弱性也通过被攻击主机→交换机→P@*Layer*_1 主机的连接完成了层内扩散过程,使 P@*Layer*_1 主机表现为 U@*Layer*_1 状态.在控制层中,存在多控制器协同工作的情况,控制器间通信信息对于协同工作十分重要.若一个控制器中引入脆弱性,在多控制器进行通信的过程中脆弱性会在控制层内进行扩散,使其他控制器表现为 U@*Layer*_2.在应用层中,对于嵌套应用而言,外部应用与嵌套应用嵌套连接,共享权限、隐私等信息.当外部应用由于多种原因,被攻击者攻击,而具有脆弱性,脆弱性就会通过共享信息侵入嵌套应用,使嵌套应用表现为 U@*Layer*_3,使脆弱性在应用层内扩散.但是,该扩散过程可能随时由于攻击停止、安全管理策略库的升级而终止.在⟨*Connection*2_*i*⟩与⟨*Connection*3_*i*⟩中,基于脆弱性扩散的过程性与阶段性,D,F 组件与 P 组件连接默认形成 U 组件,U 组件可进一步经过⟨*Detection_i*⟩与⟨*Abnormity*1_*i*⟩转换为 D 组件与 F 组件.

⟨*Detection_i*⟩描述在基础设施层,一些安全设备通过流量、IP 异常检测等方法及时发现假冒、篡改等脆弱性的存在,组件脆弱性由 U@*Layer*_1 转变为 D@*Layer*_1.在控制层,一些新型控制器可以对 SDN 中的不安全因素进行及时的检测与发现,如:FortNox 中的冲突检测算法可以及时检测出由于流规则的不一致性引发的脆弱性问题.基于 Netflow 的入侵检测方法可以有效地检测到 SDN 脆弱性,使组件脆弱性由 U@*Layer*_2 转变为 D@*Layer*_2.在应用层,一些 SDN 安全应用可以检测出部分安全问题,使组件脆弱性由 U@*Layer*_3 转变为 D@*Layer*_3.

⟨*Abnormity*1_*i*⟩,⟨*Abnormity*2_*i* 描述由于脆弱性的持续累积,部分组件出现故障,状态演变为 F@*Layer_i*.

⟨*Mitigation*1_*i*⟩,⟨*Mitigation*2_*i*⟩描述由于被检测或演化出故障的组件被发现后,系统会通过打补丁、重配置错误信息,增加异常流规则过滤装置、对异常流规则进行修改、对非法流规则进行丢弃等方式对脆弱组件进行减缓或修复,使组件转变为 M@*Layer_i*状态.

⟨*Defect_i*⟩描述由于修复操作的不彻底性,如协议自身的安全性、接口标准等不能完全修复因素的存在,导致部分被修复的组件内仍然存在脆弱性,因此,部分 M@*Layer_i* 组件仍会转化为 P@*Layer_i* 组件.

在层内脆弱性扩散模型中,设每种反应的转化率为 r_α,α 为层内扩散.用[P@*Layer_i*],[U@*Layer_i*],[D@*Layer_i*],[F@*Layer_i*]及[M@*Layer_i*]分别表示第 i 层中组件 P,U,D,F,M 的初始个数.由 Bio-PEPA 的 Mass-Action 语义规则,则函数速率分别满足:

$$f_{Connection1\_i}=r_{Connection1\_i}\times[\mathrm{U}@Layer\_i]\times[\mathrm{P}@Layer\_i];$$
$$f_{Conversion1\_i}=r_{Conversion1\_i}\times[\mathrm{U}:\mathrm{P}@Layer\_i];$$
$$f_{Connection2\_i}=r_{Connection2\_i}\times[\mathrm{D}@Layer\_i]\times[\mathrm{P}@Layer\_i];$$
$$f_{Conversion2\_i}=r_{Conversion2\_i}\times[\mathrm{D}:\mathrm{P}@Layer\_i];$$
$$f_{Connection3\_i}=r_{Connection3\_i}\times[\mathrm{F}@Layer\_i]\times[\mathrm{P}@Layer\_i];$$
$$f_{Conversion3\_i}=r_{Conversion3\_i}\times[\mathrm{F}:\mathrm{P}@Layer\_i];$$
$$f_{Detection\_i}=r_{Detection\_i}\times[\mathrm{U}@Layer\_i];$$
$$f_{Abnormity1\_i}=r_{Abnormity1\_i}\times[\mathrm{U}@Layer\_i];$$
$$f_{Abnormity2\_i}=r_{Abnormity2\_i}\times[\mathrm{D}@Layer\_i];$$
$$f_{Mitigation1\_i}=r_{Mitigation1\_i}\times[\mathrm{D}@Layer\_i];$$
$$f_{Mitigation2\_i}=r_{Mitigation2\_i}\times[\mathrm{F}@Layer\_i];$$
$$f_{Defect\_i}=r_{Defect\_i}\times[\mathrm{M}@Layer\_i].$$

3.3 层间脆弱性扩散形式化模型

脆弱性扩散也会发生在相邻 2 个垂直层之间，与层内扩散不同的是，层间扩散会将某层内的脆弱性通过南北向接口传播至相邻层的不同类型组件上. 同理，基于 Bio-PEPA 定义的层间脆弱性扩散规则描述如下：

[vertical]

1) ⟨*Connection*1_*i*_*j*⟩

P@*Layer_i*+U@*Layer_j*→U:P@*Layer_i*+U@*Layer_j*;

2) ⟨*Conversion*1_*i*_*j*⟩

U:P@*Layer_i*→U@*Layer_i*;

3) ⟨*Connection*2_*i*_*j*⟩

P@*Layer_i*+D@*Layer_j*→D:P@*Layer_i*+D@*Layer_j*;

4) ⟨*Conversion*2_*i*_*j*⟩

D:P@*Layer_i*→U@*Layer_i*;

5) ⟨*Connection*3_*i*_*j*⟩

P@*Layer_i*+F@*Layer_j*→F:P@*Layer_i*+F@*Layer_j*;

6) ⟨*Conversion*3_*i*_*j*⟩

F:P@*Layer_i*→U@*Layer_i*.

在 vertical 系列规则中，*Layer_j* 表示与 *Layer_i* 相邻的层. 因此，$j\in\{i+1,i-1\}\cap\{1,2,3\}$. 结合图 2 及 SDN 各层间脆弱性扩散原理，对 SDN 层间扩散规则 vertical 进行阐述.

1) 基础设施层→控制层

攻击者首先侵入 SDN 中的部分主机，使其表现为 U@*Layer_*1，D@*Layer_*1 或者 F@*Layer_*1 状态，并通过其向网络中输入恶意请求. 如：大量伪造的无效网络流量请求(DoS 攻击)，从而触发 SDN 交换机的转发流规则请求操作，导致大量 Packet-In 流信息同时涌向 P@*Layer_*2 状态的控制器. 使控制器服务能力达到上限，不能响应正常的流请求，使得控制器服务能力瘫痪，表现为 U@*Layer_*2 状态，使脆弱性由基础设施层通过 Packet-In 信息向控制层进行扩散.

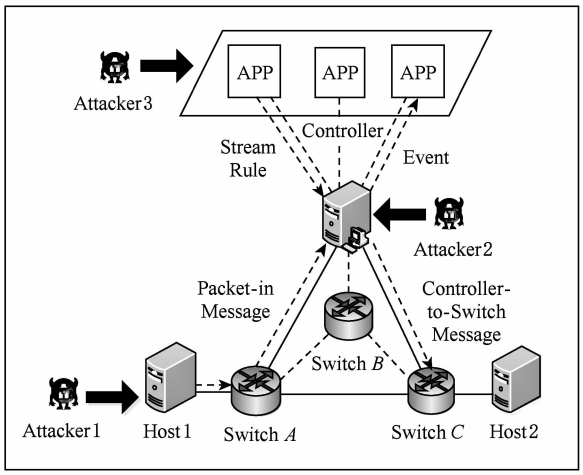


Fig. 2 Vulnerability propagation between layers

图 2 层间脆弱性扩散

2) 控制层→基础设施层

当攻击者侵入控制器并使其表现为 U@*Layer_*2，D@*Layer_*2 或者 F@*Layer_*2 状态时，控制器会产生一些携带脆弱性的非法流规则，并且通过 Controller-to-switch 信息下发到基础设施层的网络设备. 由于基础设施层的网络设备对控制器下发的流规则完全信任，因此，基础设施层的部分设备表现为 P@*Layer_*1 状态. 网络设备很容易因执行控制器下发的非法流规则而表现出脆弱性，使脆弱性从控制层通过 Controller-to-switch 信息传播至基础设施层.

3) 控制层→应用层

应用层的应用行为由来自控制器的事件或者外部输入所驱动. 当与应用相连的控制器被攻击而表现为 U@*Layer_*2，D@*Layer_*2 或者 F@*Layer_*2 状态时，P@*Layer_*3 状态的应用有可能由于控制器上传的非法事件而感染脆弱性，进而表现为 U@*Layer_*3 状态，脆弱性由控制层通过非法事件扩散到应用层.

4) 应用层→控制层

在应用层中，恶意的应用程序可以很容易地被开发，已授权的合法应用程序也可能被篡改，从而使某些应用表现为 U@*Layer_*3，D@*Layer_*3 或者

F@$Layer_3$状态. 当非法应用程序向控制层的控制器下发流表信息时,应用层的脆弱性就通过流规则与控制器相连,使控制器不能正常工作,将脆弱性由应用层通过非法流规则扩散至控制层.

在层间脆弱性扩散模型中,设每种反应的转化率参数分别为 $r_{Connection1\_i\_j}$, $r_{Connection2\_i\_j}$, $r_{Connection3\_i\_j}$. 用[P@$Layer\_i$],[U@$Layer\_j$],[D@$Layer\_j$],[F@$Layer\_j$]分别表示第 i 层中组件 P 及第 j 层中组件 U,D,F 的初始个数. 由 Bio-PEPA 的 Mass-Action 语义规则,则此时函数速率分别满足:

$$f_{Connection1\_i\_j}=r_{Connection1\_i\_j}\times[\mathrm{P@}Layer\_i]\times[\mathrm{U@}Layer\_j];$$

$$f_{Conversion1\_i\_j}=r_{Conversion1\_i\_j}\times[\mathrm{U:P@}Layer\_i];$$

$$f_{Connection2\_i\_j}=r_{Connection2\_i\_j}\times[\mathrm{P@}Layer\_i]\times[\mathrm{D@}Layer\_j];$$

$$f_{Conversion2\_i\_j}=r_{Conversion2\_i\_j}\times[\mathrm{D:P@}Layer\_i];$$

$$f_{Connection3\_i\_j}=r_{Connection3\_i\_j}\times[\mathrm{P@}Layer\_i]\times[\mathrm{F@}Layer\_j];$$

$$f_{Conversion3\_i\_j}=r_{Conversion3\_i\_j}\times[\mathrm{F:P@}Layer\_i].$$

3.4　脆弱性扩散模型描述

本文将 SDN 网络以层为单位,分析了脆弱性在水平(层内)及垂直(层间)2 个维度的扩散情况. 综合以上层内及层间分析,可将 SDN 脆弱性扩散状态转移图概括为图 3 所示:

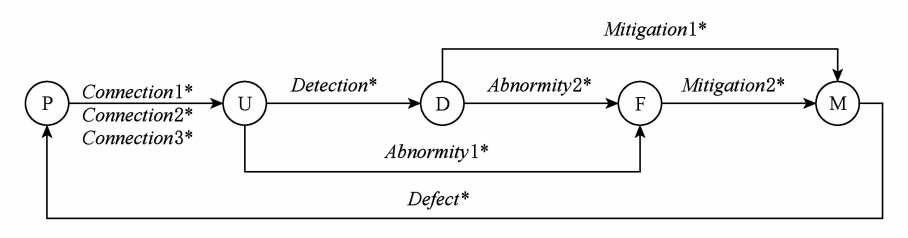


Fig. 3　State transition diagram of vulnerability propagation

图 3　脆弱性扩散状态转移图

基于此,可用 Bio-PEPA 对状态转换即 SDN 脆弱性扩散过程进行形式化建模,部分关键描述如下:

$$\mathrm{U:P}\overset{\mathrm{def}}{=}\sum_{i=1}^{3}(Connection1\_i,1)\uparrow$$

$$\mathrm{U:P@}Layer\_i+\sum_{i=1}^{3}(Conversion1\_i,1)\downarrow$$

$$\mathrm{U:P@}Layer\_i+\sum_{i=1}^{3}(Connection1\_i\_j,1)\uparrow$$

$$\mathrm{U:P@}Layer\_i+\sum_{i=1}^{3}(Conversion1\_i\_j,1)\downarrow$$

$$\mathrm{U:P@}Layer\_i;$$

$$\mathrm{D:P}\overset{\mathrm{def}}{=}\sum_{i=1}^{3}(Connection2\_i,1)\uparrow$$

$$\mathrm{D:P@}Layer\_i+\sum_{i=1}^{3}(Conversion2\_i,1)\downarrow$$

$$\mathrm{D:P@}Layer\_i+\sum_{i=1}^{3}(Connection2\_i\_j,1)\uparrow$$

$$\mathrm{D:P@}Layer\_i+\sum_{i=1}^{3}(Conversion2\_i\_j,1)\downarrow$$

$$\mathrm{D:P@}Layer\_i;$$

$$\mathrm{F:P}\overset{\mathrm{def}}{=}\sum_{i=1}^{3}(Connection3\_i,1)\uparrow$$

$$\mathrm{F:P@}Layer\_i+\sum_{i=1}^{3}(Conversion3\_i,1)\downarrow$$

$$\mathrm{F:P@}Layer\_i+\sum_{i=1}^{3}(Connection3\_i\_j,1)\uparrow$$

$$\mathrm{F:P@}Layer\_i+\sum_{i=1}^{3}(Conversion3\_i\_j,1)\downarrow$$

$$\mathrm{F:P@}Layer\_i.$$

在模型组件表述中,组件后面标注的数字代表对应组件的初始数量. 具体地,模型组件描述如下:

$$\mathrm{P}([\mathrm{P@}Layer\_1])\underset{*}{\bowtie}\mathrm{P}([\mathrm{P@}Layer\_2]\underset{*}{\bowtie}$$

$$\mathrm{P}([\mathrm{P@}Layer\_3]\underset{*}{\bowtie}\mathrm{U}([\mathrm{U@}Layer\_1])\underset{*}{\bowtie}$$

$$\mathrm{U}([\mathrm{U@}Layer\_2]\underset{*}{\bowtie}\mathrm{U}([\mathrm{U@}Layer\_3]\underset{*}{\bowtie}$$

$$\mathrm{D}([\mathrm{D@}Layer\_1])\underset{*}{\bowtie}\mathrm{D}([\mathrm{D@}Layer\_2]\underset{*}{\bowtie}$$

$$\mathrm{D}([\mathrm{D@}Layer\_3]\underset{*}{\bowtie}\mathrm{F}([\mathrm{F@}Layer\_1])\underset{*}{\bowtie}$$

$$\mathrm{F}([\mathrm{F@}Layer\_2]\underset{*}{\bowtie}\mathrm{F}([\mathrm{F@}Layer\_3]\underset{*}{\bowtie}$$

$$\mathrm{M}([\mathrm{M@}Layer\_1])\underset{*}{\bowtie}\mathrm{M}([\mathrm{M@}Layer\_2]\underset{*}{\bowtie}$$

$$\mathrm{M}([\mathrm{M@}Layer\_3]\underset{*}{\bowtie}\mathrm{U:P}([\mathrm{U:P@}Layer\_1])\underset{*}{\bowtie}$$

$$\mathrm{U:P}([\mathrm{U:P@}Layer\_2])\underset{*}{\bowtie}$$

$$\mathrm{U:P}([\mathrm{U:P@}Layer\_3])\underset{*}{\bowtie}$$

$$\mathrm{D:P}([\mathrm{D:P@}Layer\_1])\underset{*}{\bowtie}$$

$$\mathrm{D:P}([\mathrm{D:P@}Layer\_2])\underset{*}{\bowtie}$$

$$\mathrm{D:P}([\mathrm{U:P@}Layer\_3])\underset{*}{\bowtie}$$

$$\mathrm{F:P}([\mathrm{F:P@}Layer\_1])\underset{*}{\bowtie}$$

$$\mathrm{F:P}([\mathrm{F:P@}Layer\_2])\underset{*}{\bowtie}\mathrm{F:P}([\mathrm{F:P@}Layer\_3]).$$

利用该模型可以对 SDN 内部各层、层间脆弱性及其在水平、垂直 2 个维度内的扩散过程进行描述

和分析.需要说明的是,在节点有防护的情况下,可以通过增大检测反应转化率或者提高修复反应转化率来实现,进而降低脆弱性在网络中的扩散,表现为脆弱态组件数量减少.另一方面,由于内外条件变化等,脆弱性扩散后在 SDN 不同层内及层间对其他节点带来的影响具体表现不同,在模型中统一用脆弱态组件的数量进行衡量.

4 仿真实验与分析

4.1 模型分析

针对本文所构建的脆弱性扩散模型,我们分别对 SDN 架构中的 3 个层次进行了层内和层间脆弱性扩散行为的形式化建模与分析.为了进一步准确分析验证本文提出模型的合理性,本文选用 Eclipse Bio-PEPA Plugin 工具包,分别采用 ODEs 方法和 Gillespie's Stochastic 方法进行测试.本实验设定脆弱性组件总数为一个常数,由 3 000 个个体组成,3 个层内组件数量(P,U,D,F,M)的初始值分别为(1 000,0,0,0,0),(600,300,100,0,0),(1 000,0,0,0,0).观察时间取 24 个时间单位,模型的参数取值如表 1 所示,Gillespie's Stochastic 方法选取 97 个随机点.

Table 1 Main Parameter Values of Model

表 1 模型主要参数取值表

Parameter	Value	Parameter	Value
$r_{Connection1\_i}$	0.1	$r_{Mitigation1\_i}$	0.2
$r_{Connection2\_i}$	0.3	$r_{Mitigation2\_i}$	0.1
$r_{Connection3\_i}$	0.3	r_{Defect_i}	0.1
$r_{Detection\_i}$	0.3	$r_{Connection1\_i\_j}$	0.1
$r_{Abnormality1\_i}$	0.1	$r_{Connection2\_i\_j}$	0.3
$r_{Abnormality2\_i}$	0.2	$r_{Connection3\_i\_j}$	0.3

采用 Gillespie's Stochastic 算法的脆弱性扩散随机仿真实验和脆弱性扩散的 ODE 仿真结果分别如图 4 和图 5 所示.对比 2 图可以发现,2 种仿真方法脆弱性的扩散趋势及结果大致相同.从图 3 中可以看出,系统中 P 组件的数量在反应开始后迅速减少,U,D 组件数量迅速上升.这是由于此时系统中 P 组件数量最多,浓度最大,此时 Connection 系列反应占主导,迅速消耗 P 组件,产生 U,D,F 三种组件.对应于实际 SDN 网络中脆弱性组件 U,D 与潜在脆弱性组件 P 在连接反应的作用下,使脆弱性在 SDN 网络内迅速扩散.此外,从图 3 中还可以看出,U@*Layer*_1,U@*Layer*_3 从 0 开始增长,P@*Layer*_1,P@*Layer*_3 从 1 000 开始下降,说明此时发生了层间脆弱性扩散现象.脆弱性由 *Layer*_2 层(控制层)的 U@*Layer*_2,D@*Layer*_2,F@*Layer*_2 组件向 *Layer*_1 层(基础设施层)、*Layer*_3 层(应用层)进行扩散,部分 P 组件转化为 U 组件,进而在 *Layer*_1 层、*Layer*_3 层内进行层内扩散.U 组件数量在 0～2 个时间单位之间达到峰值并开始出现下降趋势,D 组件数量仍在上升.说明此时 U 组件浓度达到最大,此时〈*Detection*〉反应占据主导,消耗 U 组件,生成 D 组件.对应于扩散模型中的〈*Detection*〉过程,脆弱性组件 U 被检测到脆弱态,进而表现为 D 状态.D 组件数量在 2～4 个时间单位之间达到峰值并出现下降趋势,F 组件数量在 8～9 个时间单位之间达到峰值并出现下降趋势,而 M 组件仍处在上升阶段.这是由于在这 2 个时间点,D,F 组件浓度达到最大,〈*Mitigation*〉系列反应占据主导,消耗 D,F 组件,生成 M 组件.对应于扩散模型中的〈*Mitigation*〉过程,表现出脆弱性 D,F 的组件通过一系列修复手段,脆弱性得到缓解,转变为 M 状态.最终,P,U,D,F,M组件数量趋于稳定,达到一种扩散平衡态.因此,

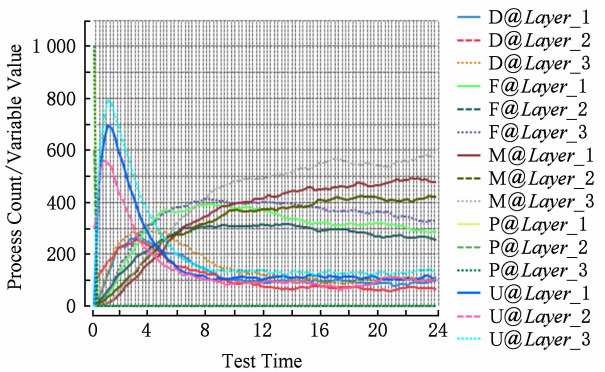


Fig. 4 Random simulation results for vulnerability propagation of SDN

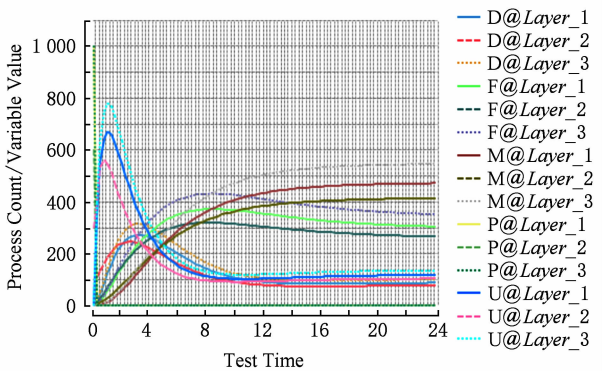


Fig. 5 ODE simulation results for vulnerability propagation of SDN

图 5 SDN 脆弱性扩散 ODE 仿真结果

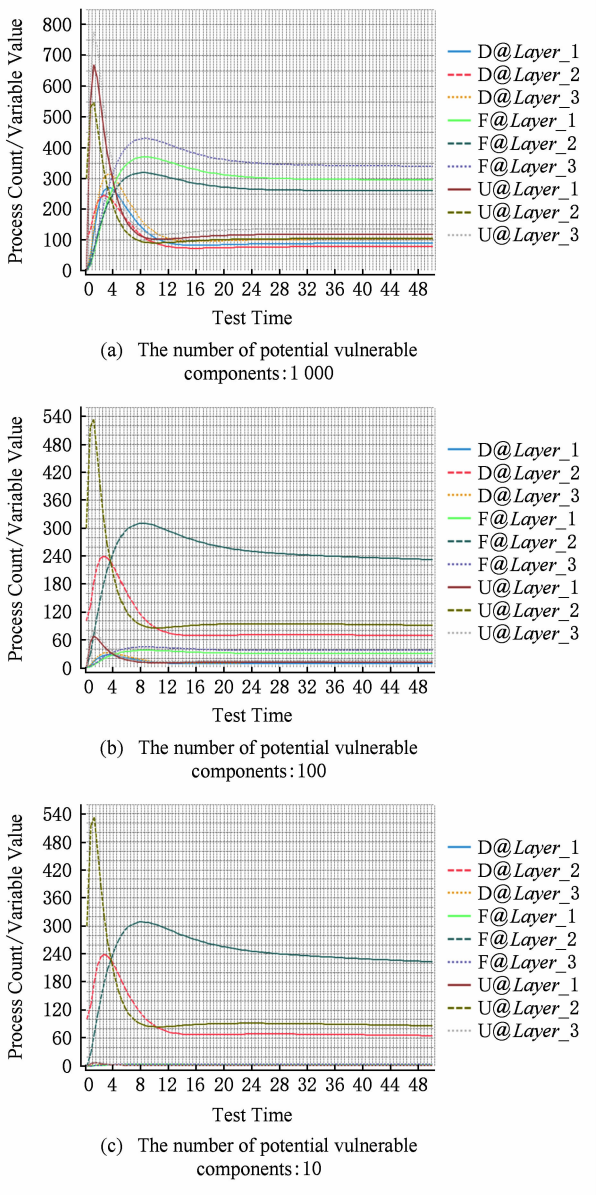
本文基于 Bio-PEPA 构建的 SDN 脆弱性扩散形式化模型可以正确反映 SDN 网络中的脆弱性扩散问题.

4.2　扩散因素分析

通过对扩散机制的分析,本文着重研究连接转化率参数 $r_{Connection^*}$、检测反应转化率参数 $r_{Detection^*}$、修复反应转化率参数 $r_{Mitigation^*}$ 三个因素对 SDN 网络脆弱性扩散的影响.

1) 潜在态组件数量对脆弱性扩散的影响

潜在态组件的存在为 SDN 内脆弱性的扩散提供了可能,脆弱态组件可通过连接反应,使潜在态组件感染脆弱性,使脆弱性在网络内快速扩散. 因此,潜在态组件的数量是影响 SDN 网络内脆弱性扩散的一个重要因素. 针对该因素对脆弱性扩散的影响,本部分实验采用改变基础设施层和应用层内潜在态组件数量的方式,以扩散平衡状态下脆弱态组件的稳态概率作为衡量指标进行实验. 试验中基础设施层和应用层内潜在态组件数量分别为 1 000,100,10,其他参数不变,运行时间为 50 个时间单位,SDN 网络内脆弱态组件变化情况如图 6 所示.



(a) The number of potential vulnerable components: 1 000

(b) The number of potential vulnerable components: 100

(c) The number of potential vulnerable components: 10

Fig. 6　Changes of vulnerable components in SDN

图 6　SDN 内脆弱态组件变化情况

此外,保持其他变量不变,使潜在态组件数量在 0 到 1 000 之间随机取 7 个值,计算扩散平衡状态下脆弱态组件的稳态概率,结果如表 2 所示. 由表 2 可以看出,在 SDN 内随着基础设施层和应用层潜在态组件的减少,SDN 脆弱性扩散平衡状态下脆弱态组件的稳态概率逐渐减小. 这是由于潜在态组件数量减少,脆弱态组件不能或者更难寻找到感染的受体,因而脆弱性的扩散受到抑制. 因此,可以通过减少潜在态组件的数量来抑制 SDN 网络内脆弱性的扩散.

Table 2　Steady-State Probability of Vulnerable Components

表 2　脆弱态组件稳态概率

Number of Potential Vulnerable Components	Steady-State Probability
1 000	0.507 535 822
500	0.488 351 413
250	0.459 524 55
100	0.421 011 957
50	0.400 051 301
25	0.389 101 363
10	0.378 132 346

2) 连接转化率对脆弱性扩散的影响

连接反应扩散方式是层内脆弱性扩散及层间脆弱性扩散的主要反应方式之一,主要由连接转化率参数 $r_{Connection^*}$ 进行控制. 层内连接转化率参数 $r_{Connection1\_i}$,$r_{Connection2\_i}$,$r_{Connection3\_i}$ 表示层内处于 U,D,F 状态的组件对 P 状态组件状态转化的影响,层间连接转化率参数 $r_{Connection1\_i\_j}$,$r_{Connection2\_i\_j}$,$r_{Connection1\_i\_j}$ 则表示 SDN 某层内处于 U,D,F 状态的组件对直接相邻层内 P 状态组件状态转化的影响. 为了分析连接转化率参数 $r_{Connection^*}$ 对脆弱性扩散的影响,下面将以稳态概率作为评估指标,在整个 SDN 网络层面,考察连接转化率参数 $r_{Connection^*}$ 取值水平对脆弱性扩散的影响. 分别对层内及层间连接转化参数从 0.1 到 1 之间 10 等分均匀取值,整个网络范围内稳态概率变化如图 7 所示.

从图 7 中可以看出,随着连接转化率参数逐渐增大,SDN 网络的脆弱性稳态概率值逐渐增大. 这是

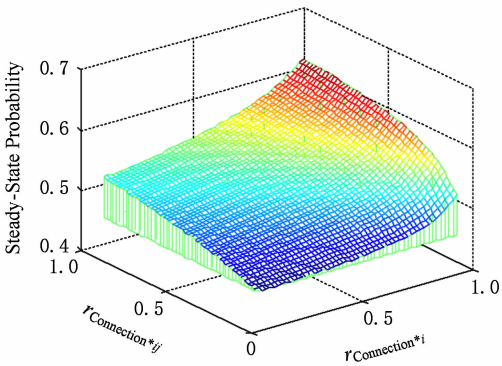


Fig. 7 The influence of connection conversion rate on SDN vulnerability

图 7 连接转化率参数对 SDN 脆弱性的影响

由于随着连接转化率参数的增大,层内及层间节点交互更为紧密,发生脆弱性传播的几率加大,导致在 SDN 网络达到平衡状态时,脆弱态组件数量增多. 因此,可以得出连接转化率参数与脆弱性扩散速率正相关,因而为了抑制 SDN 网络内脆弱性的传播需降低连接转化率参数.

3) 检测转化率参数对脆弱性扩散的影响

检测反应可以及时发现 SDN 网络中存在的 U 状态组件将其转化成 D 状态组件,并进行修复等操作,因此检测反应转化率参数对脆弱性扩散有一定影响. 本部分实验以检测参数取值 0.3 为基准,分别取基准值的 1/3,1/2,1,2,3 倍,即 0.1,0.15,0.3,0.6,0.9 进行对比试验,采用脆弱性稳态概率作为评估指标,衡量网络的脆弱状况. 不同检测参数取值下,SDN 网络内脆弱态集合内组件数量情况如图 8 所示,脆弱性稳态概率如图 9 所示.

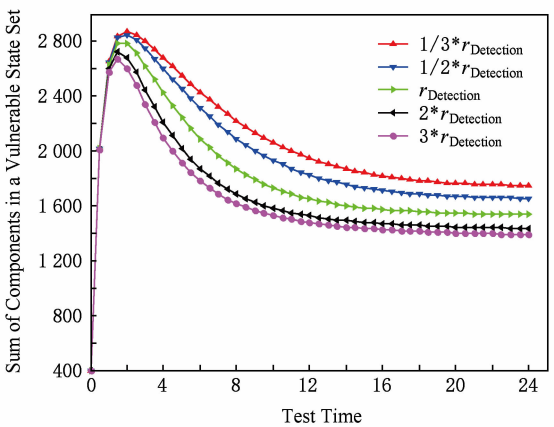


Fig. 8 The influence of detection conversion rate on the number of components in the vulnerable state set

图 8 检测转化率参数对脆弱态集合内组件数量的影响

结合两图可以看出,随着检测转化率参数的增大,脆弱态集合内组件数量和呈下降趋势. 这是由于随着检测转化率参数的增大,更多的 U 状态组件可被检测出脆弱性进而转化为 D 状态组件,D 状态组件又经 Mitigation 反应转变为 M 状态组件,也就是说组件状态经历了 U→D→M 的过程,进而迁移出脆弱态集合,使脆弱态集合内组件数量减少,使 SDN 脆弱性降低. 因而,可以通过增大检测转化率参数来抑制脆弱性扩散.

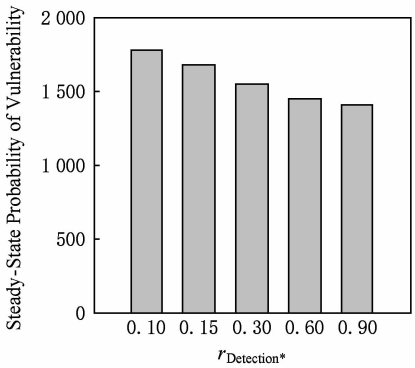


Fig. 9 The influence of detection conversion rate on the steady-state probability of vulnerability

图 9 检测转化率参数对脆弱性稳态概率的影响

4) 修复转化率参数对脆弱性扩散的影响

SDN 网络内部,修复能力的存在可以有效地降低系统内 D,F 组件的数量,缓解网络的脆弱性. 本部分实验以稳态概率作为 SDN 脆弱性的衡量指标,对修复转化率参数 $r_{Mitigation1*}$ 和 $r_{Mitigation2*}$ 在[0,1]之间每 0.1 间隔随机取值,其他参数完全相同. 通过对比实验,研究修复转化率参数对 SDN 网络内脆弱性扩散的影响,实验结果如图 10 所示:

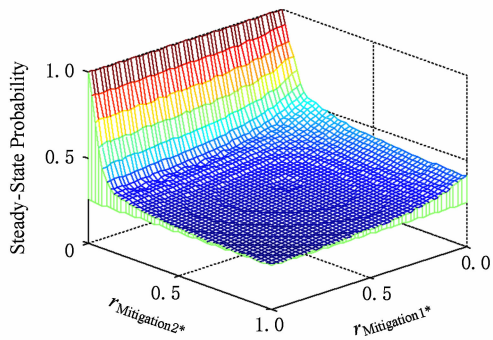


Fig. 10 The influence of repairing conversion rate on SDN vulnerability

图 10 修复转化率参数对 SDN 脆弱性的影响

从图 10 中可以看出,随着修复转化率参数的增大,稳态概率值逐渐变小,说明此时,网络内脆弱态集合内组件总数量逐渐变小,SDN 网络脆弱性降低.

同时,观察左右 2 个侧面图,我们可以发现 $r_{Mitigation2^*}$ 对网络脆弱性的扩散影响较大,即脆弱性在不同脆弱态组件间的扩散具有差异性.这是由于在脆弱性状态循环中,如图 3 所示,当 $r_{Mitigation1^*}$ 较小时,部分 D 状态组件还可以通过〈$Mitigation2^*$〉系列反应转化为 F 状态组件,F 状态组件经过 $Mitigation2^*$ 系列反应转变为 M 组件,从而迁出脆弱态集合;当 $r_{Mitigation2^*}$ 较小时,F 状态组件没有其他反应途径进行状态转化,导致大量的 F 组件滞留在脆弱态集合中,无法转化为非脆弱态的组件,使网络脆弱性较高.因此,可以得出修复转化率参数与脆弱性扩散速率负相关,因而为了抑制 SDN 网络内脆弱性的传播需提高修复转化率参数,尤其是 $Mitigation2^*$ 转化率参数.

综合以上分析,可以看出本文构建的形式化模型对 SDN 脆弱性扩散状态间的逻辑关系及动态变化具有较强的描述能力,且与仿真分析结果基本相符,能够正确地反映出不同条件下 SDN 脆弱性的变化趋势.可以通过降低连接转化率,提升检测转化率及修复转化率等措施抑制 SDN 脆弱性的扩散.具体地,针对 SDN 存在的可能脆弱点,可以通过设计面向 SDN 的可靠控制器,制定授权认证及安全约束,扩展安全通信协议,采用连接迁移、流量迁移、执行触发、主动流规则检测与分析、拟态防御、应用隔离及弹性策略、容错机制、探测防御、安全加固等技术与防御方法,多维度提高 SDN 安全性.

5　总　　结

本文采用 Bio-PEPA 对 SDN 网络的脆弱性扩散问题进行了形式化建模,针对 SDN 具有明显分层的网络架构,分别就水平垂直 2 个维度内的脆弱性的扩散的本质问题进行了研究.本文对构建的 Bio-PEPA 形式化模型分别采用 ODEs 方法和 Gillespie's Stochastic 方法进行了测试,实验结果基本相符,反映本文所构建 SDN 脆弱性模型的合理性.此外,本文还对影响 SDN 脆弱性扩散的因素进行了分析,通过仿真实验证明,潜在态组件数量、连接转化率参数、检测转化率参数、修复转化率参数都对 SDN 脆弱性的扩散具有影响,因此,本文对 SDN 网络的构建具有一定指导性意义.在下一步研究工作中,将面向 SDN 的可扩展性,对其脆弱性扩散模型展开深入研究.

参　考　文　献

[1] Feamster N, Rexford J, Zegura E. The road to SDN: An intellectual history of programmable networks [J]. ACM SIGCOMM Computer Communication Review, 2014, 44(2): 87-98

[2] Kreutz D, Ramos F, Verissimo P. Towards secure and dependable software-defined networks [C] //Proc of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. New York: ACM, 2013: 55-60

[3] Porras P, Cheung S, Fong M, et al. Securing the software-defined network control layer [C] //Proc of the Annual Network and Distributed System Security Symposium. San Diego: Internet Society, 2015: 1-15

[4] Gao Shang, Li Zecheng, Xiao Bin, et al. Security threats in the data plane of software-defined networks [J]. IEEE Network, 2018, 32(4): 108-113

[5] Suh M, Park S H, Lee B, et al. Building firewall over the software-defined network controller [C] //Proc of the 16th Int Conf on Advanced Communication Technology. Piscataway, NJ: IEEE, 2014: 744-748

[6] Chi Yaping, Jiang Tingting, Dai Chuping, et al. Design and implementation of cloud platform intrusion prevention system based on software defined network [J]. Journal of Computer Applications, 2017, 37(6): 1625-1629 (in Chinese)
(池亚平, 姜停停, 戴楚屏, 等. 基于软件定义网络的云平台入侵防御方案设计与实现[J]. 计算机应用, 2017, 37(6): 1625-1629)

[7] Wang Xiulei, Chen Ming, Xing Changyou, et al. Software defined security networking mechanism against DDoS attacks [J]. Journal of Software, 2016, 27(12): 3104-3119 (in Chinese)
(王秀磊, 陈鸣, 邢长友, 等. 一种防御 DDoS 攻击的软件定义安全网络机制[J]. 软件学报, 2016, 27(12): 3104-3119)

[8] He Heng, Hu Yan, Zheng Lianghan, et al. Efficient DDoS attack detection and prevention scheme based on SDN in cloud environment [J]. Journal on Communications, 2018, 39(4): 139-151 (in Chinese)
(何亨, 胡艳, 郑良汉, 等. 云环境中基于 SDN 的高效 DDoS 攻击检测与防御方案[J]. 通信学报, 2018, 39(4): 139-151)

[9] Lim S, Ha J, Kim H, et al. A SDN-oriented DDoS blocking scheme for botnet-based attacks [C] //Proc of the 6th Int Conf on Ubiquitous and Future Networks. Piscataway, NJ: IEEE, 2014: 63-68

[10] Wang Bing, Zheng Yao, Lou Wenjing, et al. DDoS attack protection in the era of cloud computing and software-defined networking [J]. Computer Networks, 2015, 81(C): 308-319

[11] Alsmadi I, Xu Dianxiang. Security of software defined networks: A survey [J]. Computers & Security, 2015, 53(C): 79-108

[12] Li Wenjuan, Meng Weizhi, Kwok L F. A survey on openflow-based software defined networks: Security challenges and countermeasures [J]. Journal of Network and Computer Applications, 2016, 68: 126-139

[13] Shin S, Gu Guofei. Attacking software-defined networks: A first feasibility study [C] //Proc of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. New York: ACM, 2013: 165-166

[14] Yang Yunfei. SDN flow table update method based on consistency of application layer policy conversion [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2015 (in Chinese)
(杨云飞. 基于应用层策略转换一致性的 SDN 流表更新方法 [D]. 南京: 南京邮电大学, 2015)

[15] Qin Kuangyu, Huang Chuanhe, Wang Caihua, et al. Balanced multiple controllers placement with latency and capacity bound in software-defined network [J]. Journal on Communications, 2016, 37(11): 90-103 (in Chinese)
(覃匡宇, 黄传河, 王才华, 等. SDN 网络中受时延和容量限制的多控制器均衡部署 [J]. 通信学报, 2016, 37(11): 90-103)

[16] Duan Jie, Gao Jiangming, Cheng Kefei, et al. Failure recovery algorithm based on flow type in SDN data plane [J]. Journal of Chongqing University of Posts and Telecommunications, 2018, 30(1): 134-140 (in Chinese)
(段洁, 高江明, 程克非, 等. 基于流类型的 SDN 数据平面故障恢复算法 [J]. 重庆邮电大学学报, 2018, 30(1): 134-140)

[17] Hu Fei, Hao Qi, Bao Ke. A survey on software-defined network and openflow: From concept to implementation [J]. IEEE Communications Surveys & Tutorials, 2014, 16(4): 2181-2206

[18] Oktian Y E, Lee S G, Lee H J, et al. Secure your northbound SDN API [C] //Proc of the 7th Int Conf on Ubiquitous and Future Networks. Piscataway, NJ: IEEE, 2015: 919-920

[19] Yeganeh S H, Tootoonchian A, Ganjali Y. On scalability of software-defined networking [J]. Communications Magazine, 2013, 51(2): 136-141

[20] Nunes B A A, Mendonca M, Nguyen X, et al. A survey of software-defined networking: Past, present, and future of programmable networks [J]. IEEE Communications Surveys & Tutorials, 2014, 16(3): 1617-1634

[21] Zhang Chaokun, Cui Yong, Tang Heyi, et al. State-of-the-art survey on software-defined networking (SDN) [J]. Journal of Software, 2015, 26(1): 62-81 (in Chinese)
(张朝昆, 崔勇, 唐翯祎, 等. 软件定义网络(SDN)研究进展 [J]. 软件学报, 2015, 26(1): 62-81)

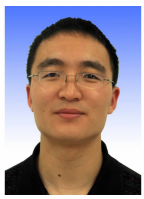
[22] Akhunzada A, Gani A, Anuar N B, et al. Secure and dependable software defined networks [J]. Journal of Network and Computer Applications, 2016, 61(C): 199-221

[23] Zhou Ye, Yang Xu, Li Yong, et al. Classification based consistent flow update scheme in software defined network [J]. Journal of Electronics & Information Technology, 2013, 35(7): 1746-1752 (in Chinese)
(周烨, 杨旭, 李勇, 等. 基于分类的软件定义网络流表更新一致性方案 [J]. 电子与信息学报, 2013, 35(7): 1746-1752)

[24] Matias J, Garay J, Mendiola A, et al. FlowNAC: flow-based network access control [C] //Proc of the 3rd European Workshop on Software Defined Networks. Piscataway, NJ: IEEE, 2014: 79-84

[25] Ciocchetta F, Hillston J. Bio-PEPA: A framework for the modeling and analysis of biological systems [J]. Theoretical Computer Science, 2009, 410(33): 3065-3084

Wang Jian, born in 1979. PhD, associate professor and master supervisor. Member of CCF. Her main research interests include SDN, survivability, cognitive network and crowd sensing.



Zhao Guosheng, born in 1977. PhD, professor, and master supervisor. Senior member of CCF. His main research interests include survivability, SDN, and trusted computing. (zgswj@163.com)

Zhao Zhongnan, born in 1978. PhD, lecturer. Member of CCF. His main research interests include security situation awareness, and fault-tolerance (piconet@126.com).

Li Ke, born in 1993. Master candidate. Student member of CCF. Her main research interests include security situation awareness and cognitive computing (1736819170@qq.com).