doi: 10.12068/j.issn.1005-3026.2020.09.001

SDN中DDoS攻击的高效联合检测和防御机制

曾荣飞[1], 高 原[2], 王兴伟[2], 张 榜[2]
(1. 东北大学 软件学院, 辽宁 沈阳 110169; 2. 东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

摘 要: 为解决软件定义网络(SDN,software-defined networking)控制器所面临的DDoS攻击问题,本文提出一个高效率的联合检测和防御机制. 联合检测部分采用改进自组织映射(SOM,self-organizing mapping)算法和多维条件熵算法相结合,通过对自组织映射算法的改进,与多维条件熵算法相互提供反馈信息,达到高效联合检测目的. 联合防御部分采用常规防御模块与快速防御模块相结合,通过调整优先级的方式针对不同的检测结果采取不同的防御策略. 大量实验表明,本文的联合检测机制可以达到95.2%的检测率;与单独的防御机制相比,联合防御机制中控制器的响应时间可以平均降低0.11 s.

关 键 词: 软件定义网络;分布式拒绝服务攻击;改进自组织映射算法;多维条件熵算法;优先级
中图分类号: TP 393.08 **文献标志码**: A **文章编号**: 1005-3026(2020)09-1217-06

Efficient Joint Detection and Defense Mechanism for DDoS Attack in SDN

ZENG Rong-fei[1], *GAO Yuan*[2], *WANG Xing-wei*[2], *ZHANG Bang*[2]
(1. School of Software, Northeastern University, Shenyang 110169, China; 2. School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China. Corresponding author: WANG Xing-wei, E-mail: wangxw@mail.neu.edu.cn)

Abstract: In order to defend against the DDoS attacks for SDN(software-defined networking) controller, this paper proposed an efficient joint detection and defense mechanism. The joint detection part adopted the combination of improved self-organizing mapping algorithm and multidimensional conditional entropy algorithm. By combining the two methods, the purpose of joint detection was achieved. The joint defense part includes a conventional defense module and a fast defense module, which adopts different defense strategies for different detection results by adjusting the priority. Extensive experimental results showed that the joint detection mechanism can achieve a detection rate of 95.2%, and the response time of the joint defense mechanism to the controller can be reduced by 0.11 s on average, compared with the single defense mechanism.
Key words: software-defined networking; distributed denial of service attack; improved self-organizing mapping algorithm; multidimensional conditional entropy algorithm; priority

随着计算机网络的高速发展,传统的TCP/IP网络在已有架构的基础上不断引入新技术,传统网络已不能满足当今网络需要,一种新型网络架构——SDN(software-defined networking)由此出现. 这种新型网络架构的最大特点是将数据平面与控制平面相分离,通过编程控制网络[1]. 目前,一些企业已成功将SDN部署到真实的网络环境. 例如,谷歌的数据中心[2]、无线传感器网络[3]、NTT边缘网关[4]等. 由于SDN拥有在传统网络中无法获得的灵活性、可编程性和可扩展性,一提出就受到广泛关注[5].

新型SDN网络在具有诸多优点的同时,还面临着控制器DDoS攻击等安全问题. 控制器是SDN的核心,攻击者一旦接入控制器,就可以控制整个网络,从而造成难以预料的危害.

针对上述问题,本文设计了一种高效的联合检

收稿日期: 2019-09-30
基金项目: 国家重点研发计划项目(2017YFB0801701); 国家自然科学基金资助项目(61872073).
作者简介: 曾荣飞(1983-),男,辽宁沈阳人,东北大学副教授; 王兴伟(1968-),男,辽宁盖州人,东北大学教授,博士生导师.

测和防御机制,利用改进的 SOM 算法和多维条件熵算法分别对流表项和控制器进行检测,基于这两种算法的检测模块互相为对方提供反馈信息. 本文针对不同的联合检测结果采取不同的防御策略. 该检测和防御机制对攻击流量更加敏感,从而能够更好地对 SDN 控制器中的 DDoS 攻击进行检测和防御.

1 相关工作

到目前为止,针对该工作使用的检测方法主要包括统计学和人工智能两大类.

Mousavi 等[6]根据 SDN 控制器使用资源的情况,提出了一种基于目的 IP 地址熵变化的轻量级解决方案来检测 SDN 中的 DDoS 攻击.

Li 等[7]、Jiang 等[8]和 Vokorokos 等[9]均采用自组织映射(self-organizing mapping,SOM)算法对流量进行聚类,但是传统的 SOM 算法其网络结构是固定的,网络没有很好的自适应能力. 本文针对以上局限性对传统 SOM 算法进行改进,增加了生长操作,增强了网络的自适应性.

2 架构设计

本文将面向 SDN 控制器的 DDoS 攻击分为两个阶段:第一个阶段是针对 SDN 中交换机的流表项;第二个阶段是针对大量被发送到控制器的数据包. 结合两个攻击阶段的特点和传统网络中的三种检测方式(源端、中间网络、目的端),提出了以改进 SOM 算法为主,多维条件熵算法为辅的联合检测机制. 另外,本文还根据不同的检测结果采取不同的防御策略,整体框架如图 1 所示.

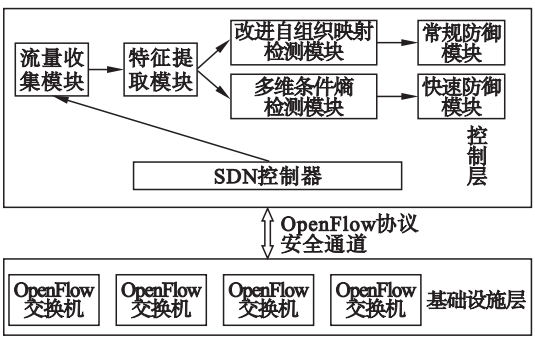

图 1 SDN 中 DDoS 攻击的联合检测和防御机制整体框架

Fig. 1 Overall framework of joint detection and defense mechanism for DDoS attacks in SDN

2.1 联合检测机制

2.1.1 改进自组织映射算法

本检测是针对攻击的第一阶段. 利用改进的 SOM 算法对 OpenFlow 交换机中流表项信息进行检测分类. 对判定为正常型的流表项直接放行,攻击型的流表项传入防御模块,可疑型的流表项传入多维条件熵检测模块. 根据多维条件熵检测模块的反馈结果来决定是否新增聚类中心.

传统的 SOM 算法主要包括初始化、采样、竞争、突触适应这几个步骤. 该算法存在诸如网络结构不能动态变化,网络在没有经过完整的学习前,不能增加新类别等不足. 为此,本文提出改进 SOM 网络,此种网络增加了动态生长操作,算法流程图如图 2 所示.

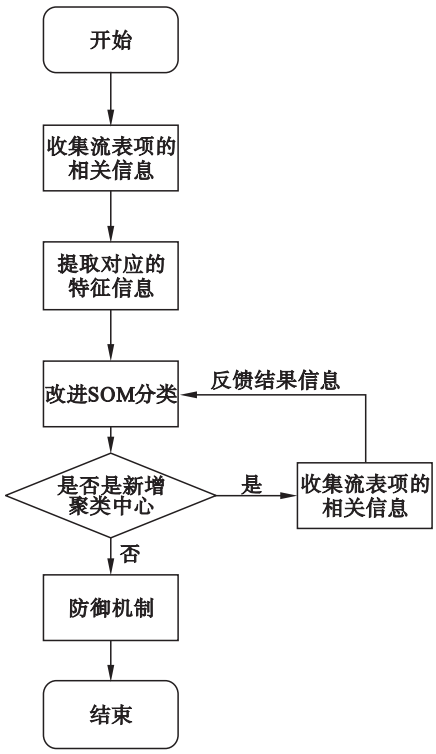

图 2 改进自组织映射算法流程图

Fig. 2 Flow chart of improved self-organizing mapping

该算法首先为每一个输入样本 x,找到其最佳匹配节点 b 及其邻域. 然后,调整与 b 相连以及 b 邻域中各节点的权值. 接着,计算 x 与 b 的累积误差 M,若 M 大于预先设定的生长阈值 GT(growth threshold),则在 b 的邻域中生成一个新节点,否则做权值调整操作,直到算法收敛. 改进的算法详细步骤如下:

1) 为使输入数据落在[0,1]区间,本文采用简单缩放中的 min - max 标准,公式为

$$x^* = \frac{x - \min}{\max - \min}. \tag{1}$$

2) 记第一个输入的向量为第一个聚类中心,根据需求计算生长阈值 GT,公式为

$$GT = D \times (1 - \beta)^n. \tag{2}$$

其中：D 是节点权重向量的维数；β 为调节因子；n 为聚类次数，满足 $0<\beta<1$，$n>1$.

3）找到最佳匹配节点 b，并记录当前神经元的最大差异（该神经元与其他神经元的最小距离），记为 dis，初始化为 0.

4）计算 W 的累积误差，记为 M，公式如下：

$$M=\sum_{k=1}^{D}(v_{x,k}-v_{b,k})^2. \tag{3}$$

式中：$\boldsymbol{v}$ 为节点权重向量；D 为 $\boldsymbol{v}$ 的维数.

5）若 $M>\mathrm{GT}$，按式（4）做生长操作，生成 b 的新的子节点 c；

$$\mathrm{LR}(i+1)=\mathrm{LR}(i)\times\sigma. \tag{4}$$

其中：$\mathrm{LR}(i)$ 为学习率；σ 为调节因子，满足 $0<\sigma<1$.

6）若 $M\leqslant\mathrm{GT}$，按式（5）做权值调整操作；

$$v_j(i+1)=\begin{cases}v_j(i), & j\notin N_{i+1};\\ v_j(i)+\mathrm{LR}(i)\times(v_i-v_j(i)), & j\in N_{i+1}.\end{cases} \tag{5}$$

式中：$v_j(i+1)$ 为 j 调整后的权值；N_{i+1} 为第 $i+1$ 次训练时 b 的邻域.

7）当有新聚类中心生成时，计算该聚类中心与其他聚类中心的最小欧氏距离，记为 S. 若 $S>\mathrm{dis}$，则更新 $\mathrm{dis}=S$.

8）重复以上操作，直到无节点生成.

9）计算每一个生成的聚类中心获胜的次数，将获胜次数过小的中心节点删除，并重新训练，直到网络不再有新节点生成.

2.1.2　多维条件熵算法

本检测是针对攻击的第二阶段. 利用该算法对已上传到控制器中的数据包和从改进 SOM 检测模块传来的可疑型流表项检测，将检测结果反馈给改进 SOM 检测模块，使改进 SOM 检测模块可以根据反馈信息决定是否增加新的聚类中心.

该算法首先是对控制器中的数据包进行特征提取，包括源 IP 地址 S_{ip}、目的 IP 地址 D_{ip} 和目的端口号 D_{port}. 由这三个特征可以得到 $H(S_{\mathrm{ip}}|D_{\mathrm{ip}})$，$H(D_{\mathrm{ip}}|S_{\mathrm{ip}})$ 等 6 个条件熵. 然后，由以上 6 个条件熵组成 1 个六维向量. 计算该六维向量与正常情况下流表项的六维向量的欧氏距离，结果与阈值进行比较，如果连续 5 次大于阈值，则认为攻击已经发生. 对正常情况下六维向量的值和阈值的选取可通过训练正常流量和攻击流量的实验获得.

此外，该检测模块需要将目的端检测结果反馈给改进 SOM 检测模块，并与改进 SOM 检测模块相结合，具体结合如下：

在改进 SOM 检测模块中，当 $S>\mathrm{dis}$ 时，新增聚类中心；当 $\mathrm{GT}<S<\mathrm{dis}$ 时，将该聚类中心设为待定聚类中心；当 S 比 dis 和 GT 均小时，不需要新增聚类中心.

1）当改进 SOM 检测模块新增聚类中心时，若多维条件熵检测模块的检测结果为正常，则将该聚类中心标记为正常；反之，标记为攻击.

2）当改进 SOM 检测模块的聚类中心设为待定聚类中心时，由目的端检测结果和距离此待定中心最近的聚类中心标签的结果共同决定是否新增此聚类中心. 若二者检测结果相同，则不增加此聚类中心，并将离其最近的聚类中心当作生长时期的获胜节点. 若二者检测结果不同，则新增此聚类中心并将其标记为目的端的检测结果.

3）当无新增聚类中心且多维条件熵检测模块检测到攻击时，对攻击和可疑数据包对应将要下发的流表项做新增聚类节点操作，并做出标记.

2.2　联合防御机制

对于改进 SOM 检测模块，采用限流的常规防御策略；对于多维条件熵检测模块，采用过滤的快速防御策略. 该机制流程图如图 3 所示.

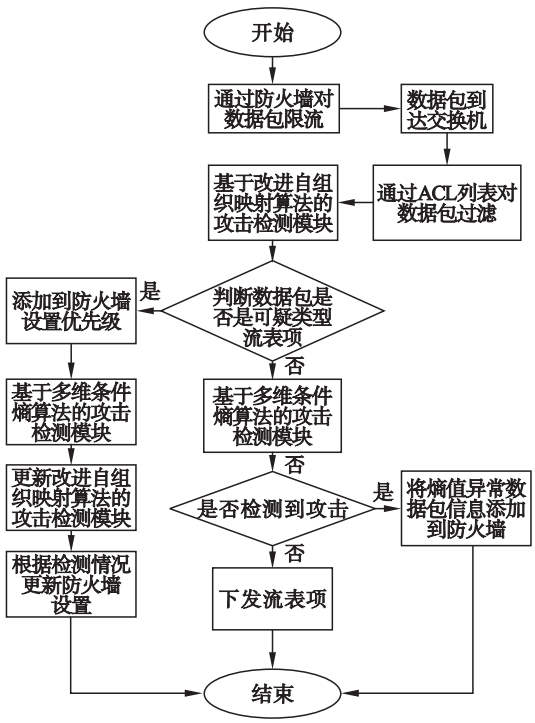

图 3　联合防御机制流程图

Fig. 3　Flowchart of joint defense mechanism

对于改进 SOM 检测模块，将判定为攻击型的流表项通过添加表 1 中的属性到 Floodlight 控制器的 ACL 中进行限流操作.

对于多维条件熵检测模块，结合该模块对目的端和改进 SOM 检测模块传入的可疑流表项的检测结果来调整各流表项在 Floodlight 控制器中

防火墙的优先级,优先级高的流表项采用过滤操作. 优先级达到 10 时,认为此节点无限接近正常节点;达到 0 时,认为无限接近攻击节点. 首先,将可疑节点的相关信息添加至防火墙条目中,优先级统一设置为 5. 然后,根据联合检测结果的四种情况提出如下四类防御策略.

表 1 ACL 控制列表中的属性

Table 1 Attributes in ACL control list

属性 1	属性 2	属性 3	属性 4	属性 5	属性 6	属性 7	属性 8	属性 9
ID	Source IP	Dest	Source IP	Mask	Dest IP	Port	Act	Delete

一是目的端未检测到攻击且可疑节点的熵值正常,此时认为可疑节点是正常节点,提高此节点在防火墙中的优先级;二是目的端未检测到攻击但可疑节点的熵值异常,此时不能确定可疑节点类型,不调整该节点在防火墙中的优先级;三是目的端检测到攻击但可疑节点的熵值正常,此时攻击已经发生,不能排除此节点是潜在的攻击节点,先降低此节点在防火墙中的优先级,并将此节点的分类标签修改为攻击. 另外,由于此时目的端检测到攻击,所以需要将熵值异常的节点信息反馈给改进 SOM 检测模块,并将此节点的分类标签标记为可疑,同时添加到防火墙中,设置优先级为 5;四是目的端检测到攻击且可疑节点的熵值异常,此时认为此节点无限趋近于攻击节点,将此节点的优先级降为 0,把信息反馈给改进 SOM 检测模块,并将此节点的分类标签设置为攻击. 另外,还需将其他熵值异常节点的相关信息反馈给改进 SOM 检测模块,并将分类标签设为可疑,同时将这些节点添加至防火墙中,将优先级设置为 5.

3 性能评价

3.1 仿真环境

本实验的拓扑参考来自 Topology Zoo 中的一个拓扑案例. 共有 4 台交换机,每个交换机下面依次有 6,7,9,2 台主机,如图 4 所示.

3.2 评价指标

1) 检测准确率. 针对分类算法的准确性,统一采用 $F1$ – Score 进行评价[10],计算公式如下:

$$\text{precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}, \quad (6)$$

$$\text{recall} = \frac{\text{TurePositive}}{\text{TruePositive} + \text{FalseNagetive}}, \quad (7)$$

$$F1_k = \frac{2 \times (\text{precision}_k \times \text{recall})}{\text{precision}_k + \text{recall}_k}, \quad (8)$$

$$\text{score} = \left(\frac{1}{n}\sum F1_k\right)^2. \quad (9)$$

其中:precision 为精确率;recall 为召回率;$F1_k$ 为每个类别下的 $F1$ – Score;TruePositive 为预测答案正确的标签数;FalsePositive 为错将其他类预测为本类的标签数;FalseNegative 为将本类预测为其他类的标签数.

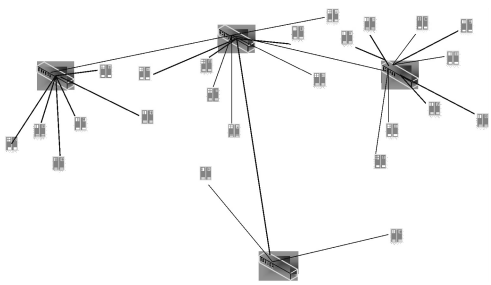

图 4 实验拓扑

Fig. 4 Experimental topology

2) 响应时间. 针对检测和防御算法,用响应时间作为评价指标. 本文对每十次消耗的时间做累积加和操作,对十次响应时间取平均值,得到平均响应时间.

3.3 攻击检测与防御机制性能评价

本文使用三种攻击速率(fast,faster,flood)以及三种流量(攻击、正常、混合)进行仿真实验,得到检测准确率和响应时间. 其中,fast,faster,flood 分别代表每秒发送 10 个,100 个和尽最快速度发送数据包且攻击流量占全部流量的比重低于 50% .

1) 检测准确率. 改进 SOM 算法中的流量收集时间窗口大小设为 5s,特征提取模块提取每个数据流的数据包数、字节数,持续匹配的时间以及端口的变化率,涉及的参数设置如表 2 所示. 训练时期,采用 TCP,UDP,ICMP 协议进行训练,具体比重参考文献[11]. 训练数据如表 3 所示,每个 OpenFlow 交换机的训练数据如表 4 所示.

多维条件熵算法中的流量收集时间窗口大小同设 5 s,选择不同的攻击速率,依次变换源 IP,目的 IP 和目的端口号. 使用正常流量和不同的攻击速率各训练 100 次,找到正常情况下六维向量的值(设为 2)和不同攻击速率下六维向量的值(阈

值 2.2). 然后,在已设定的网络拓扑下进行测试,每五次检测结果取一次平均值,得出的 $F1$ – Score 如图 5、图 6 所示,训练时期的检测准确率可达 94.9% . 最后,将其接入仿真环境中,取值方法同上,结果如图 7、图 8 所示,联合检测机制的准确率可达 95.2% .

表 2　检测模块参数设置

Table 2　Parameter setting of detection module

参数名称	参数值
邻域范围	与获胜节点直接相连的子节点
D	5
β	$0<\beta<1$
$n(t)$	第 t 次训练时网络节点数
训练时期的学习率	$1/k^{1/2}$
损失函数	Sigmoid 交叉熵

表 3　训练数据

Table 3　Training data

攻击类型	训练数据	测试阶段
TCP flood	505 000	860 000
UDP flood	78 000	156 000
ICMP flood	10 000	54 000

表 4　各个 OpenFlow 交换机的训练数据

Table 4　Training data for each OpenFlow switch

OpenFlow 交换机	训练阶段		测试阶段	
	攻击流量	合法流量	攻击流量	合法流量
交换机 1	148 500	200 000	267 000	300 000
交换机 2	148 500	200 000	267 000	300 000
交换机 3	147 500	200 000	268 000	300 000
交换机 4	147 500	200 000	268 000	300 000
总计	593 000	800 000	1 070 000	1 200 000

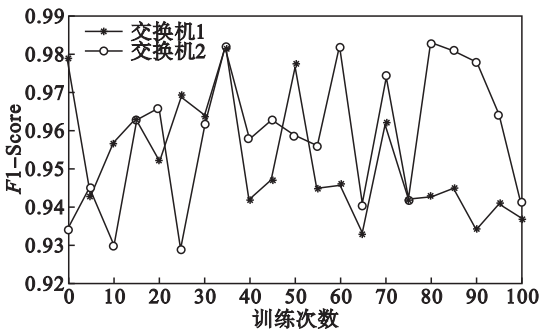

图 5　训练时期交换机 1 和 2 的 $F1$ – Score

Fig. 5　$F1$-Score of switches 1 and 2 during training period

2）响应时间. 依次使用三种攻击速率对常规、快速和联合防御模块各训练 100 次,得到结果如图 9 ~ 11 所示. 计算可知,与单独的常规防御机制和快速防御机制相比,本文设计的联合防御机制的控制器响应时间可分别减少 0.1,0.12 s.

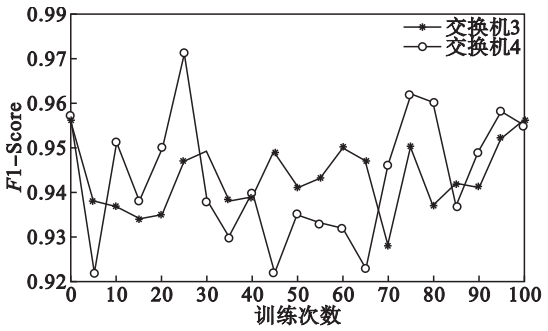

图 6　训练时期交换机 3 和 4 的 $F1$-Score

Fig. 6　$F1$-Score of switches 3 and 4 during training period

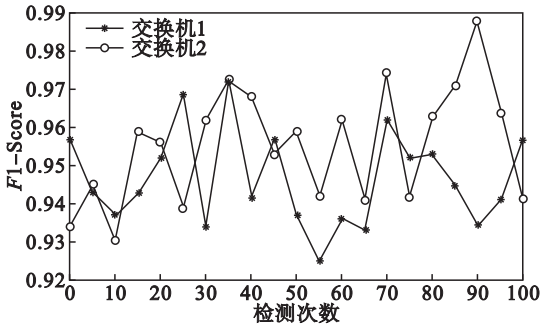

图 7　检测时期交换机 1 和 2 的 $F1$ – Score

Fig. 7　$F1$-Score of switches 1 and 2 during detection period

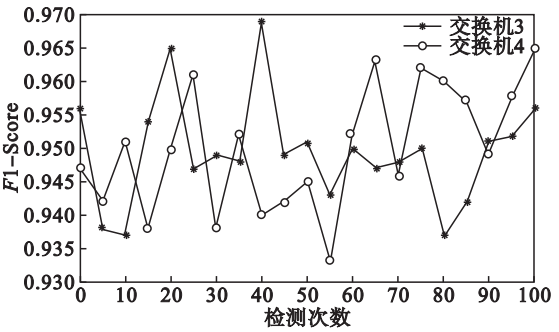

图 8　检测时期交换机 3 和 4 的 $F1$-Score

Fig. 8　$F1$-Score of switches 3 and 4 during detection period

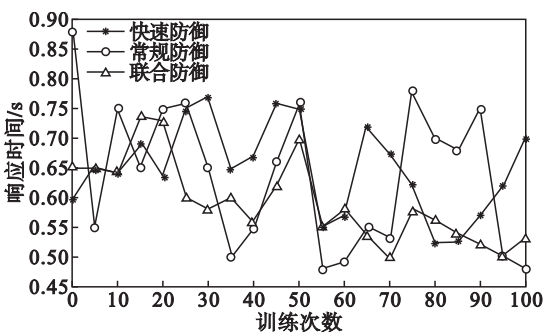

图 9　Fast 攻击速率下不同防御机制对控制器响应时间的影响

Fig. 9　Influence of different defense mechanisms at fast attack rates on response time of controller

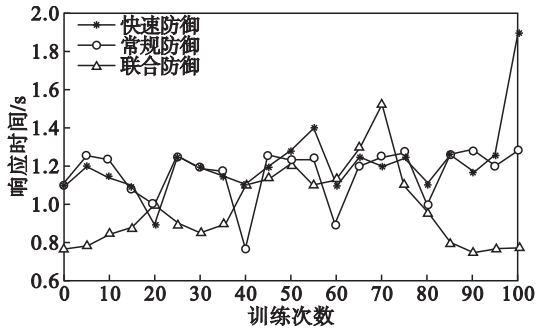

图 10 Faster 攻击速率下不同防御机制对控制器响应时间的影响

Fig. 10 Influence of different defense mechanisms at faster attack rates on response time of controller

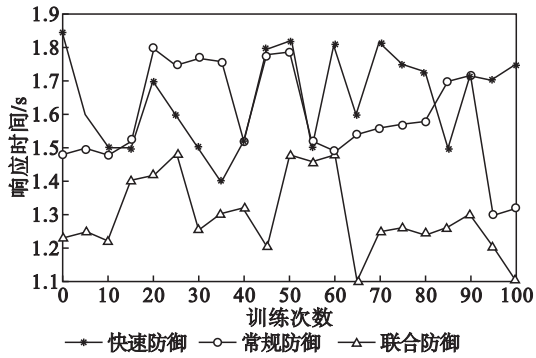

图 11 Flood 攻击速率下不同防御机制对控制器响应时间的影响

Fig. 11 Influence of different defense mechanisms at flood attack rates on response time of controller

4 结 语

本文以 SDN 架构为基础,对 SOM 算法进行改进,结合多维条件熵算法建立联合检测机制,将常规防御与快速防御相结合,建立联合防御机制. 改进的 SOM 算法增加了生长操作,并且能够从多维条件熵检测模块得到信息反馈,从而使得该联合检测机制能够达到 95.2% 的准确率. 另外,该机制与单独防御机制相比,控制器的响应时间能平均降低 0.11 s. 由于本文的仿真实验所设节点数目有限,今后会在更大规模上进行测试.

参考文献:

[1] Erel M, Teoman E, Özçevik Y, et al. Scalability analysis and flow admission control in mininet-based SDN environment [C]//Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN). San Francisco, CA, 2015: 18 – 19.

[2] Jain S, Kumar A, Mandal S, et al. B4: experience with a globally-deployed software defined WAN[C]//Proceedings of the ACM SIGCOMM Computer Communication Review. Hong Kong, 2013: 3 – 14.

[3] Luo T, Tan H P, Quek T Q S. Sensor OpenFlow: enabling software-defined wireless sensor networks [J]. *IEEE Communications Letters*, 2012, 16(11): 1896 – 1899.

[4] Natarajan S, Ramaiah A, Mathen M. A software defined cloud-gateway automation system using OpenFlow [C]// Proceedings of the 2013 IEEE the 2nd International Conference on Cloud Networking (CloudNet). San Francisco, CA, 2013: 219 – 226.

[5] Gao L L, Zheng L, Qiu Z L, et al. Research on model of five-level scheduling based on SDN [C]//Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS). Colmar, France, 2018: 1 – 5.

[6] Mousavi S M, St-Hilaire M. Early detection of DDoS attacks against SDN controllers [C]//Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC). Anaheim, CA, 2015: 77 – 81.

[7] Li M, Dongliang W. Anormaly intrusion detection based on SOM [C]//Proceedings of the 2009 WASE International Conference on Information Engineering. Taiyuan, 2009: 40 – 43.

[8] Jiang D, Yang Y, Xia M. Research on intrusion detection based on an improved SOM neural network [C]// Proceedings of the 2009 Fifth International Conference on Information Assurance and Security. Xi'an, 2009: 400 – 403.

[9] Vokorokos L, Balaz A, Chovanec M. Intrusion detection system using self organizing map[J]. *Acta Electrotechnica et Informatica*, 2006, 6(1): 1 – 6.

[10] Huang H, Xu H, Wang X, et al. Maximum F1-score discriminative training criterion for automatic mispronunciation detection[J]. *IEEE/ZACM Transactions on Audio, Speech, and Language Processing*, 2015, 23(4): 787 – 797.

[11] Borgnat P, Dewaele G, Fukuda K, et al. Seven years and one day: sketching the evolution of Internet traffic [C]// International Conference on Computer Communications. Rio de Janeiro, 2009: 711 – 719.