计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(8):133−139 [doi: 10.15888/j.cnki.csa.009206]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# SDN 环境下 DDoS 攻击检测和缓解系统①

沈浩桐, 魏松杰

(南京理工大学 计算机科学与工程学院, 南京 210094)
通信作者: 魏松杰, E-mail: swei@njust.edu.cn

**摘　要**: 分布式拒绝服务攻击 (distributed denial of service, DDoS) 是网络安全领域的一大威胁. 作为新型网络架构, 软件定义网络 (software defined networking, SDN) 的逻辑集中和可编程性为抵御 DDoS 攻击提供了新的思路. 本文设计并实现了一个轻量级的 SDN 环境下的 DDoS 攻击检测和缓解系统. 该系统使用熵值检测方法, 并通过动态阈值进行异常判断. 若异常, 系统将使用更精确的决策树模型进行检测. 最后, 控制器通过计算流的包对称率确定攻击源, 并下发阻塞流表项. 实验结果表明, 该系统能够及时响应 DDoS 攻击, 具有较高的检测成功率, 并能够有效遏制攻击.

**关键词**: 软件定义网络; 分布式拒绝服务攻击; 检测; 缓解; 决策树; 熵值

引用格式: 沈浩桐,魏松杰.SDN 环境下 DDoS 攻击检测和缓解系统.计算机系统应用,2023,32(8):133–139. http://www.c-s-a.org.cn/1003-3254/9206.html

## DDoS Attack Detection and Mitigation System in SDN Environment

SHEN Hao-Tong, WEI Song-Jie

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

**Abstract**: Distributed denial of service (DDoS) attack is a major threat in the field of network security. As a new type of network architecture, the logic centralization and programmability of software defined networking (SDN) provide new ideas for defending against DDoS attacks. This study designs and implements a lightweight DDoS attack detection and mitigation system in SDN. The system uses the entropy detection method and judges the abnormality through the dynamic threshold. If the dynamic threshold is abnormal, the system will use a more accurate decision tree model for detection. Finally, the controller determines the attack source by calculating the packet symmetry rate of the flow and delivers the blocking flow entry. The experimental results show that the system can respond to DDoS attacks in time. It has a high detection success rate and can effectively contain attacks.

**Key words**: software defined networking (SDN); distributed denial of service (DDoS); detection; mitigation; decision tree; entropy

DDoS 攻击易于实施, 利用有限的资源即可在网络中发起不对称攻击. 攻击者通过控制大量主机发送无用数据包来淹没受害者的带宽或攻击其可用资源, 从而阻止合法用户访问[1]. 在传统网络中, 有许多已知的防御 DDoS 攻击的方法. 然而, 由于传统网络的局限性, 这些防御方法仍存在着不足之处.

软件定义网络是一种以新颖的方式创建可编程和易于管理的网络模型[2]. 它由数据层、控制层和应用层组成[3]. 在 SDN 中, OpenFlow 是标准通信协议[4]. 当数据包进入交换机时, 交换机会查看流表项以确定数据

① 基金项目: 工信部 2020 年工业互联网创新发展工程 (TC200H01V); 国家自然科学基金 (61802186, 61472189)
收稿时间: 2023-02-06; 修改时间: 2023-03-08; 采用时间: 2023-03-22; csa 在线出版时间: 2023-06-09
CNKI 网络首发时间: 2023-06-09

包的转发规则. 如果不存在匹配的流表项, 交换机将发送 Packet-In 消息给控制器, 控制器决定数据包的转发规则. 这种决策与转发的分离使得 SDN 能够轻松检测和响应 DDoS 攻击. 然而, SDN 的网络结构也使得网络本身容易受到 DDoS 攻击. 例如 Packet-In 泛洪攻击、针对交换机流表的溢出攻击、针对南向接口的堵塞攻击等[5]. 这些攻击不仅会消耗受害设备的计算和存储资源, 还容易阻塞南向接口, 降低控制器的网络管理能力. 相较于传统网络, SDN 能够通过集中式控制器、全局可见的网络拓扑和按需创建流量转发规则来更有效地防御 DDoS 攻击[6].

为此, 本文利用 SDN 的可编程以及集中控制的特性, 提出了一种部署于控制器内的 DDoS 攻击检测和缓解系统.

## 1 相关工作

已有许多研究者针对 SDN 环境下的 DDoS 攻击进行了检测和缓解方法的研究.

Do 等人[7] 使用控制器收集的分组报头中的目的 IP 地址来计算目的 IP 地址熵, 并使用随时间变化的动态阈值来判断网络状态. Mishra 等人[8] 提出了一种基于熵值的检测方案, 以交换机中数据包速率作为触发标准, 并在异常交换机中添加阻塞流表项.

Makuavaza 等人[9] 提出了一种基于深度神经网络的检测方案, 可在 SDN 中实时检测 DDoS 攻击. 该方案在 CICIDS-2017 上的检测准确度是 96.67%, 精确度是 97.21%. 遗憾的是, 它并没有在 SDN 上进行仿真实验. Ye 等人[10] 通过控制器定期收集交换机流表的统计信息, 从中提取六元特征组, 并用 SVM 算法对这些特征组进行攻击分类. 另外, 贾锟等人[11] 提出了一种机制, 该机制在粗粒度模式下使用统计特征检测可疑行为, 在细粒度模式下使用熵值检测算法以及 SVM 检测算法. 实验表明, 该机制可以快速响应攻击.

攻击缓解是保护网络资源的重要防御措施. Hu 等人[12] 提出了一种基于白名单和流量迁移的攻击缓解机制. 当攻击发生时, 该机制将流量迁移到缓解代理, 并利用白名单标识流量. 然后, 缓解服务器对正常流量进行转发, 对攻击流量进行丢弃. 不过, 流量迁移会造成额外负担. 刘向举等人[13] 通过对节点流量特征进行聚类分析, 实现攻击溯源, 并根据溯源结果封禁攻击端口和过滤 Packet-In 消息. Wang 等人[14] 使用卷积神经网络和极限学习机的混合方法确定攻击流量, 并根据攻击流量特征进行 IP 回溯来定位攻击者, 最后, 控制器发送流表项来过滤攻击流量.

综合研究, 本文设计并实现一个 SDN 环境下轻量级的 DDoS 攻击检测和缓解系统. 轻量级保证检测机制不占用过多控制器的计算以及存储资源. 具体而言, 控制器通过提取 Packet-In 消息中的信息来计算熵值, 并使用动态阈值判断熵值是否异常. 若熵值异常, 则使用决策树模型进行二次检测. 最后, 控制器通过不对称分析过滤攻击流量.

## 2 系统设计

整个系统有 3 个模块, 分别是阈值警告模块、攻击检测模块和攻击缓解模块. 系统的组成和工作流程如图 1 所示.

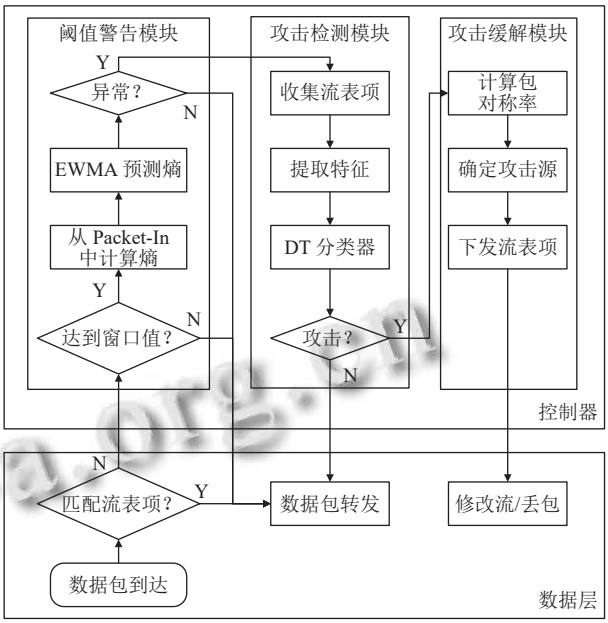


图 1 系统组成和工作流程图

### 2.1 阈值警告模块

常规的检测方案是通过控制器周期性地请求数据并进行攻击检测, 但调度周期很难确定. 若间隔时间较小, 控制器会频繁请求数据并进行检测, 对集中于控制器的检测机制而言, 这会增加对控制器以及南向接口的资源占用. 而间隔时间过长则可能导致检测不及时. 因此, 本文采用阈值警告模块并使用熵值检测方法来检测 DDoS 攻击, 有助于早期发现网络中的攻击. 此外, 它不需要周期性触发检测技术.

(1) 特征计算

当攻击者发起 DDoS 攻击时, 常进行源 IP 地址欺骗, 这会导致交换机不断触发 Table-Miss 选项, 并向控制器发送大量 Packet-In 消息. 控制器收集 Packet-In 消息, 并在消息数量达到预设窗口值时, 计算目的 IP 地址的熵值. 由于窗口中存在大量指向受害者地址的 Packet-In 消息, 因此目的 IP 地址的不确定性变小, 熵值也会显著降低. 值得注意的是, 交换机中的每条流表项都设有空闲时间和生存时间. 当流表项在空闲时间内未被命中, 或在生存时间结束后, 无论是否匹配, 均会被删除. 因此, 交换机会不断生成新的 Packet-In 消息发往控制器.

假设窗口容量为$N$, $W=\{y_1,y_2,\cdots,y_n\}$统计一个窗口中各目的 IP 地址的数量信息. 其中, $y_i$为第$i$个目的 IP 地址的数量, $n$为窗口中不同目的 IP 地址的数量. 目的 IP 地址的概率分布$P=\{p_1,p_2,\cdots,p_n\}$, 其中, $p_i$由式 (1) 计算所得:

$$p_i=\frac{y_i}{N} \tag{1}$$

熵值计算公式如式 (2) 所示:

$$E=-\sum_{i=1}^{n}p_i\log_2 p_i \tag{2}$$

(2) 阈值计算

静态阈值不能有效地根据网络流量的变化进行调整, 其选取需要较长时间的观察. 因此, 本文采用基于时间序列的指数加权移动平均 (exponentially weighted moving averages, EWMA) 模型计算动态阈值. EWMA 模型根据前一时刻的观测值和预测值计算当前时刻的预测值, 能够描述数值的变化趋势. 在网络正常情况下, 相邻窗口的熵值不会有太大的变化. 但在网络异常时, 熵值会发生显著变化, 导致 EWMA 模型预测失败. 因此, EWMA 模型可在小时间窗口内快速检测 DDoS 攻击. Aladaileh 等人[15] 已经证明了 EWMA 模型在 SDN 和 DDoS 检测中是有效的.

$\hat{y}_t$表示$t$时刻的窗口预测值, $y_t$是实际值, $\alpha$为平滑系数. EWMA 定义如式 (3) 所示:

$$\hat{y}_t=\begin{cases}\alpha y_{t-1}+(1-\alpha)\hat{y}_{t-1}, & t>2\\ y_1, & t=2\end{cases} \tag{3}$$

式 (3) 计算的预测值等于所有观测值的指数加权. 但是在实际操作中, 控制器不能记录所有的观测值. 为此, 本文设立一个滑动窗口, 大小为$w$. EWMA 模型将使用最接近当前时间的$w$个历史观测值来预测当前值. 那么当前预测值的计算公式将被如下定义:

$$\hat{y}_t=\alpha y_{t-1}+\alpha(1-\alpha)y_{t-2}+\cdots+\alpha(1-\alpha)^{w-1}y_{t-w} \tag{4}$$

动态阈值区域 (dynamic threshold, *DTH*) 计算公式如式 (5) 所示. 其中, $\sigma$是$t$时刻之前滑动窗口内样本的标准差, $C$是常量值.

$$\hat{y}_t-C\sigma\leqslant DTH\leqslant\hat{y}_t+C\sigma \tag{5}$$

正常网络流量的波动具有一定的规律性, 一般与前一段时间的流量波动具有相似性. 基于此, 本文使用标准差来描述网络本身的变化范围. 式 (6) 和式 (7) 为$t$时刻滑动窗口中所有样本的平均值以及标准差.

$$avg=\frac{1}{w}\sum_{i=t-w}^{t-1}y_i \tag{6}$$

$$\sigma=\sqrt{\frac{1}{w}\sum_{i=t-w}^{t-1}(y_i-avg)^2} \tag{7}$$

当目的 IP 地址的熵值在 *DTH* 内时, 滑动窗口将被更新. 反之, 阈值警告模块会发出警报, 并启动攻击检测模块. 熵值会因为大量突发性的合理请求或者网络链路故障而表现异常, 因此系统还实现了基于机器学习的攻击检测模块, 以提高检测精度并降低误报率.

### 2.2 攻击检测模块

一旦阈值警告模块触发警报, 防御系统将启动攻击检测模块以确定是否发生攻击. 本模块分为 3 个阶段: 流表信息收集、特征提取和攻击检测.

(1) 流表信息收集: 控制器以 $T$ 为周期发送 ofp_flow_stats_request 请求, 收集所有交换机的流表信息.

(2) 特征提取: 本文使用基于单位时间内流的统计特征来检测 DDoS 攻击. 选取的特征描述如下.

1) 流表项平均数据包量 (average packets of flow entries, *APF*)

在基于协议的 DDoS 攻击中, 攻击者将以较少的数据包量发送大量流量. 而针对大容量 DDoS 攻击, 每个流中的数据包将急剧增加, 以消耗网络带宽. 在这两种情况下, 每个流的平均数据包量要么太小, 要么太大, 这使其成为识别 DDoS 攻击的关键特征. 具体计算公式如式 (8) 所示:

$$APF=\frac{\sum_{i=1}^{N}P_i}{N} \tag{8}$$

其中, $P_i$是第$i$条流表项中的数据包数量, $N$为流总数.

2) 流表项平均字节数 (average bytes of flow entries, *ABF*)

与 *APF* 类似, 在攻击阶段, 每个流的平均字节数会过高或过低, 这使其成为检测 DDoS 攻击的一个重要特征.

$$ABF = \frac{\sum_{i=1}^{N} B_i}{N} \tag{9}$$

其中, $B_i$是第$i$条流表项中的字节数, $N$为流总数.

3) 流表项增长速率 (the growth rate of flow entries, *GFE*)

当发生 DDoS 攻击时, 每单位时间内增长的流表项数量显著高于正常值. 具体计算公式如式 (10) 所示:

$$GFE = \frac{S_{\text{flow}}}{T} \tag{10}$$

其中, $S_{\text{flow}}$是采样周期 $T$ 内增长的流表项数量.

4) 源 IP 地址的熵值 (entropy of source IP addresses, *ESA*)

由于 DDoS 攻击本质上是分布式的, 攻击者要么通过分布在网络中的程序, 要么使用伪造的 IP 地址攻击受害者. 这会导致在采样周期内, 源 IP 地址相对分散, 源 IP 地址的熵值变大. 具体计算公式如式 (11) 所示:

$$ESA = -\sum_{i} P_{sIP_i} \log_2 P_{sIP_i} \tag{11}$$

其中, $P_{sIP_i}$为源 IP 地址$sIP_i$的数量与源 IP 地址总数量的比值.

5) 目的 IP 地址熵值 (entropy of destination IP addresses, *EDA*)

攻击发生时, 流量往往集中在受害者. 因此目的 IP 地址的随机性会变低, 其熵值也会变小. 具体计算公式如下:

$$EDA = -\sum_{i} P_{dIP_i} \log_2 P_{dIP_i} \tag{12}$$

其中, $P_{dIP_i}$为目的 IP 地址是$dIP_i$的数量占总数量的比例.

6) 对流比 (ratio of pair flows, *RPF*)

当发生 DDoS 攻击并且攻击方伪造源 IP 地址时, 交换机中相交互的流表项数量会减少. 交互流是指交换机中既有源 IP 地址指向目的 IP 地址的流表项, 也存在目的 IP 地址响应源 IP 地址的流表项. 定义对流比计算公式:

$$RPF = \frac{F_{\text{pair}}}{F_{\text{sum}}} \tag{13}$$

其中, $F_{\text{pair}}$为交互流的数量, $F_{\text{sum}}$为总流表项数量.

(3) 攻击检测: 在机器学习领域中, DDoS 攻击检测可以被视为一个二值分类问题. 常见的分类算法包括朴素贝叶斯、逻辑回归、支持向量机和决策树等. 朴素贝叶斯算法假设特征独立性, 而逻辑回归算法对于非线性可分数据表现可能不佳. 支持向量机算法相对于决策树算法具有更高的复杂度. 因此, 本文选择决策树算法作为分类检测方法, 该算法具有良好的泛化能力、较高的检测精度以及较短的模型训练时间. 此外, 该算法几乎不需要数据预处理, 非常适合在线检测应用. 文献 [16,17] 使用 C4.5 决策树算法取得较好的检测效果. 因此, 本文选用 C4.5 决策树算法作为分类检测方法.

假设数据集$D$有$k$个不同的类别, $p_j$是类别发生的概率, $D$的信息熵计算公式如下:

$$E(D) = -\sum_{j=1}^{k} p_j \log_2 p_j \tag{14}$$

假设属性$A$有$n$种不同的值$\{a_1, a_2, \cdots, a_n\}$, 那么属性$A$根据取值将$D$划分为$n$个子集$\{D_1, D_2, \cdots, D_n\}$. $D$的信息增益公式如式 (15) 所示:

$$Gain(D|A) = E(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} E(D_i) \tag{15}$$

其中, $|D_i|$是属性$A$中值为$a_i$的样本数量, $|D|$是总样本数. 固有值$IV$的计算公式如式 (16) 所示:

$$IV(D,A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|} \tag{16}$$

信息增益率计算公式如式 (17) 所示:

$$GainRatio(D,A) = \frac{Gain(D,A)}{IV(D,A)} \tag{17}$$

信息增益率最高的属性作为当前节点的分裂属性. 重复步骤, 建立最优的决策树.

在攻击检测阶段, 控制器解析接收到的流表项信息, 并提取六元特征. 提取完毕后, 这些特征将被输入到 DT 模型中, 以判断网络流量是否异常. 如果检测结果为异常, 控制器将启动攻击缓解模块进行相应的防御措施. 反之, 若检测结果为正常, 则通知控制器无异常事件发生.

### 2.3 攻击缓解模块

攻击缓解模块的主要任务是利用各种技术手段,

阻止 DDoS 攻击流量并保护网络设施. 尽管攻击检测模块能够确定网络是否存在攻击流量, 但并不能确定攻击源和攻击目标. 为了避免切断合法用户的通信, 缓解模块不能简单地删除所有流量.

这里采用 Kreibich 等人[18]提出的包对称特性来过滤非法流量. 缓解模块聚合统计攻击检测模块获取的边缘交换机中的流表信息, 并计算主机发往目标主机的数据包数量$N_{tx}$以及目标主机发往该主机的数据包数量$N_{rx}$. 通过式 (18) 可以得到每条流的包对称率.

$$S = \ln\left(\frac{N_{tx}+1}{N_{rx}+1}\right) \tag{18}$$

对于 TCP 协议来说, $S > 4.5$则为攻击流量, 其他协议可以通过实验来确定合适的启发式阈值[19]. 如果流量表现出高度不对称性或包对称率超过某个阈值, 则被认为是攻击流量.

缓解模块分析边缘交换机的流表信息, 并通过包对称率识别可能的攻击流量. 控制器使用链路发现协议获取网络连接情况, 并通过判断攻击流量的输入端口是否连接到交换机来确定攻击源端口. 所有流表项分析结束后, 控制器将向交换机下发生存时间为 30 s 的阻塞流表项, 以防止进一步的攻击. 这些阻塞流表项包括交换机编号、输入端口、目的 IP 地址和具有高于其他流表项的优先级. 阻塞流表项的“outport”字段留空, 这意味着与该流表项匹配的数据包将被丢弃. 对于来自攻击端口但目的 IP 地址是受害者的良性流量, 控制器会修改正常流表项的优先级, 以使其优先级高于阻塞流表项, 来减少对正常访问者的影响.

## 3 实验

### 3.1 实验环境

Mininet[20]是一个用于仿真 SDN 网络的开源软件. 它支持创建带有虚拟主机, 交换机, 链路和控制器的 SDN 环境. 本实验使用开源控制器 Ryu, 并在 Ubuntu 16.04, i5 CPU 和 4 GB RAM 的计算机上完成. 由于缺乏可用的标准数据集, 本文使用图 2 的网络拓扑来生成数据集. 本文选用 Scapy 作为数据包生成工具, Scapy 是一款功能强大的交互式数据包操作工具, 可生成多种协议的数据包. Borgnat 等人[21]分析的数据表明, 在日本和美国之间的合法流量中, 85% 是 TCP, 10% 是 UDP, 5% 是 ICMP. 本文按照这个比例模拟背景流量, 并选$h_1$和$h_3$作为攻击主机, $h_8$作为受害服务器.

### 3.2 实验结果

根据 Fan 等人[22]的论述, 对于只有一个控制器和几百个主机的 SDN 来说, 窗口大小设为 50 是较为理想的. 此外, Oshima 等人[23]证明在熵值检测中, 有效检测的最小窗口值是 50. 因此本文选择 50 作为阈值警告模块的数据窗口大小, 并设置滑动窗口大小$w$为 50, 周期 $T$ 为 5 s.

根据图 3 所示, EWMA 模型的预测曲线与实际曲线的趋势较为一致, 表明该模型可以从历史数据中预测当前流量状况.

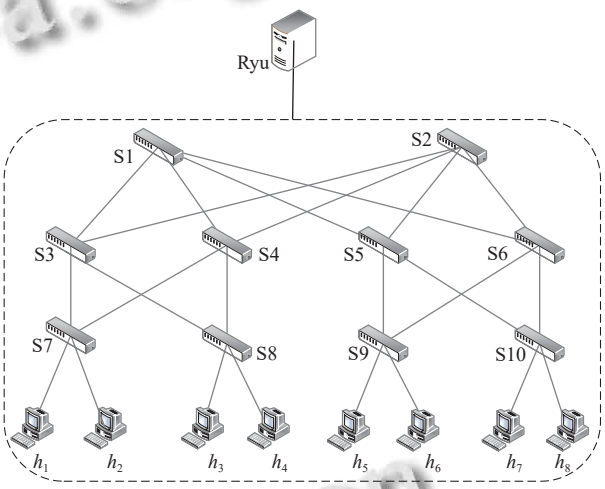


图 2 网络拓扑

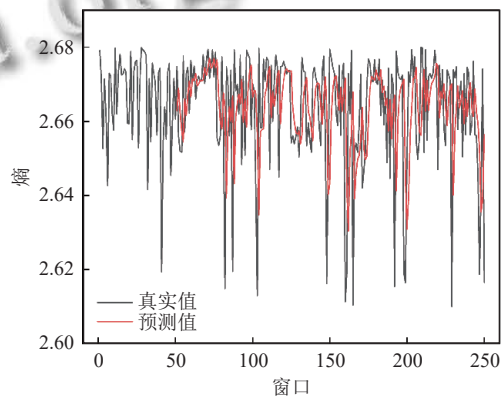


图 3 预测值与实际值的比较 ($\alpha = 0.5$)

在本研究中, 不同的平滑系数$\alpha$和常数$C$值的组合会影响 EWMA 模型的准确率、检测率以及误报率. 实验结果如表 1 所示, 在不同的参数组合下, EWMA 模型均有较好的攻击检测率, 但是有些组合的误报率较

高. 考虑到应用场景的不同, $C$可根据需求进行动态调整. 若 DDoS 需要严格监管, 则应降低$C$值; 若供应商希望首先降低误报率, 则可以松散定义$C$值. 在本实验中, $\alpha \in [0.5, 0.9]$, $C$为 5 时, 模型的准确率、检测率以及误报率均较好. 这表明阈值警告模块具有较高的可靠性.

表 1 不同参数组合的检测结果

| $\alpha$ | $C$ | 准确率 (%) | 检测率 (%) | 误报率 (%) |
|---|---|---|---|---|
| 0.3 | 3 | 94.56 | 95.24 | 14.89 |
| 0.3 | 4 | 94.56 | 92.86 | 13.33 |
| 0.3 | 5 | 97.58 | 92.86 | 2.50 |
| 0.5 | 3 | 94.56 | 95.24 | 14.89 |
| 0.5 | 4 | 94.56 | 92.86 | 13.33 |
| 0.5 | 5 | 97.89 | 92.86 | 1.27 |
| 0.7 | 3 | 95.17 | 95.24 | 13.04 |
| 0.7 | 4 | 94.56 | 92.86 | 13.33 |
| 0.7 | 5 | 97.89 | 92.86 | 1.27 |
| 0.9 | 3 | 95.17 | 95.24 | 13.04 |
| 0.9 | 4 | 95.17 | 95.24 | 13.04 |
| 0.9 | 5 | 97.89 | 92.86 | 1.27 |

本研究使用准确率, 检测率, 召回率和 $F1$ 值来评估系统中 DDoS 攻击检测模块的分类性能. 表 2 展示了在本文数据集下, C4.5、SVM、KNN 的分类结果. 结果显示, C4.5 算法的检测性能高于其他几种算法.

表 2 分类结果对比 (%)

| 算法 | 准确率 | 检测率 | 召回率 | $F1$值 |
|---|---|---|---|---|
| C4.5 | 98.89 | 99.08 | 99.38 | 99.23 |
| SVM | 96.22 | 95.03 | 100 | 97.45 |
| KNN | 98.44 | 98.18 | 99.69 | 98.93 |

图 4 展示了在有缓解模块和无缓解模块的情况下, 受害者遭受攻击时接收数据包数量的变化. 实线表示有缓解模块, 虚线表示无缓解模块. 数据由 Wireshark 抓包获得. 在正常情况下, 受害者接收数据包的速率约为 15 pkt/s. 在第 52 s 时, 网络遭受攻击, 攻击速率高达 92 pkt/s. 如果系统配备了缓解模块, 系统将向攻击源发送阻塞流表项. 受害者的数据包接收速率将在 65 s 时恢复到正常水平, 检测和缓解总用时约为 13 s. 如果没有缓解模块, 系统将在攻击停止后, 即 180 s 时才能恢复到正常水平.

为验证阈值警告模块的优势, 本文统计了在网络处于正常状态时, 使用和不使用阈值警告模块的控制器 CPU 平均使用率. 结果如表 3 所示, 使用阈值警告模块的控制器的 CPU 平均使用率低于不使用阈值警告模块的控制器. 因为不使用阈值警告模块的检测方案是周期性调用的. 这种方案需要周期性请求流表信息以及检测, 从而消耗控制器的 CPU 资源. 而使用阈值警告模块的检测方案是被动触发的, 不需要周期轮询数据以及检测. 随着网络规模和流表信息的增加, 控制器需要处理的数据也会变多, 对 CPU 的占用也会增加.

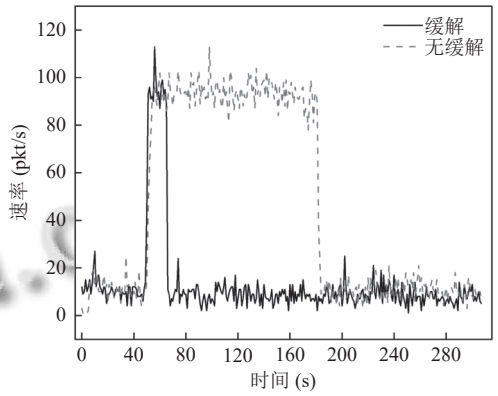


图 4 受害者接收数据量变化

表 3 控制器 CPU 平均使用率 (%)

| 有无警告 | CPU平均使用率 |
|---|---|
| 无警告 | 16.78 |
| 有警告 | 8.93 |

## 4 结论与展望

本文设计并实现了一种 SDN 中的轻量级 DDoS 攻击检测和缓解系统. 该系统使用被动触发的熵值检测方法, 不需要周期性轮询检测, 因此可以减少控制器 CPU 资源的消耗. 此外, 系统采用更加精确的决策树模型来判断网络是否存在攻击流量, 从而降低误报率. 最后, 通过不对称分析确定攻击源并下发阻塞流表项. 实验结果表明, 该系统具有较好的实时性、高检测率和显著的缓解效果. 另外, 有警告模块的系统与无警告模块的系统相比, 控制器 CPU 平均使用率减少了 7.85%. 由于本文仅考虑了单控制器的实验环境, 今后还需要进一步研究多控制器下的检测和缓解问题.

## 参考文献

1 Gaurav A, Gupta BB, Alhalabi W, *et al*. A comprehensive survey on DDoS attacks on various intelligent systems and its defense techniques. International Journal of Intelligent Systems, 2022, 37(12): 11407–11431. [doi: 10.1002/int.23048]

2 Patil J, Tokekar V, Rajan A, *et al*. Port scanning based model

to detect Malicious TCP traffic and mitigate its impact in SDN. Proceedings of the 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC). Jalandhar: IEEE, 2021. 365–370. [doi: 10.1109/ICSCCC51823.2021.9478150]

3 Ling Z, Luo JZ, Xu DN, *et al*. Novel and practical SDN-based traceback technique for malicious traffic over anonymous networks. Proceedings of the 2019 IEEE Conference on Computer Communications. Paris: IEEE, 2019. 1180–1188. [doi: 10.1109/INFOCOM.2019.8737586]

4 Peter LS, Kobo H, Srivastava VM. A comparative review analysis of OpenFlow and P4 protocols based on software defined networks. In: Jacob IJ, Shanmugam SK, Izonin I, eds. Proceedings of the 2022 Data Intelligence and Cognitive Informatics. Singapore: Springer, 2023. 699–711. [doi: 10.1007/978-981-19-6004-8_53]

5 Singh J, Behal S. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. Computer Science Review, 2020, 37: 100279. [doi: 10.1016/j.cosrev.2020.100279]

6 Ahmad I, Namal S, Ylianttila M, *et al*. Security in software defined networks: A survey. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2317–2346.

7 Do Van N, Huy LD, Truong CQ, *et al*. Applying dynamic threshold in SDN to detect DDoS attacks. Proceedings of the 2022 International Conference on Advanced Technologies for Communications (ATC). Ha Noi: IEEE, 2022. 344–349. [doi: 10.1109/ATC55345.2022.9943031]

8 Mishra A, Gupta N, Gupta BB. Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. Telecommunication Systems, 2021, 77(1): 47–62. [doi: 10.1007/s11235-020-00747-w]

9 Makuvaza A, Jat DS, Gamundani AM. Deep neural network (DNN) solution for real-time detection of distributed denial of service (DDoS) attacks in software defined networks (SDNs). SN Computer Science, 2021, 2(2): 107. [doi: 10.1007/s42979-021-00467-1]

10 Ye J, Cheng XY, Zhu J, *et al*. A DDoS attack detection method based on SVM in software defined network. Security and Communication Networks, 2018, 2018: 9804061.

11 贾锟, 王君楠, 刘峰. SDN 环境下的 DDoS 检测与缓解机制. 信息安全学报, 2021, 6(1): 17–31. [doi: 10.19363/J.cnki.cn10-1380/tn.2021.01.02]

12 Hu DW, Hong PL, Chen YX. FADM: DDoS flooding attack detection and mitigation system in software-defined networking. Proceedings of the 2017 IEEE Global Communications Conference. Singapore: IEEE, 2017. 1–7. [doi: 10.1109/GLOCOM.2017.8254023]

13 刘向举, 刘鹏程, 路小宝, 等. 基于 SD-IoT 的 DDoS 攻击防御方法. 计算机工程与设计, 2021, 42(11): 3001–3008. [doi: 10.16208/j.issn1000-7024.2021.11.001]

14 Wang J, Wang LP. SDN-defend: A lightweight online attack detection and mitigation system for DDoS attacks in SDN. Sensors, 2022, 22(21): 8287. [doi: 10.3390/s22218287]

15 Aladaileh MA, Anbar M, Hintaw AJ, *et al*. Renyi joint entropy-based dynamic threshold approach to detect DDoS attacks against SDN controller with various traffic rates. Applied Sciences, 2022, 12(12): 6127. [doi: 10.3390/app12126127]

16 刘俊杰, 王珺, 王梦林, 等. SDN 中基于 C4.5 决策树的 DDoS 攻击检测. 计算机工程与应用, 2019, 55(20): 84–88, 127.

17 Muthamil Sudar K, Deepalakshmi P. A two level security mechanism to detect a DDoS flooding attack in software-defined networks using entropy-based and C4.5 technique. Journal of High Speed Networks, 2020, 26(1): 55–76. [doi: 10.3233/JHS-200630]

18 Kreibich C, Warfield A, Crowcroft J, *et al*. Using packet symmetry to curtail malicious traffic. Proceedings of the 4th Workshop on Hot Topics in Networks. 2005. 1–6.

19 王睿. 面向软件定义物联网的信任管理及攻击防御机制研究 [博士学位论文]. 济南: 山东大学, 2018.

20 Gupta N, Maashi MS, Tanwar S, *et al*. A comparative study of software defined networking controllers using mininet. Electronics, 2022, 11(17): 2715. [doi: 10.3390/electronics11172715]

21 Borgnat P, Dewaele G, Fukuda K, *et al*. Seven years and one day: Sketching the evolution of internet traffic. Proceedings of the 2009 IEEE INFOCOM. Rio de Janeiro: IEEE, 2009. 711–719.

22 Fan C, Kaliyamurthy NM, Chen S, *et al*. Detection of DDoS attacks in software defined networking using entropy. Applied Sciences, 2021, 12(1): 370. [doi: 10.3390/app12010370]

23 Oshima S, Nakashima T, Sueyoshi T. Early DoS/DDoS detection method using short-term statistics. Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems. Krakow: IEEE, 2010. 168–173. [doi: 10.1109/CISIS.2010.53]

(校对责编: 孙君艳)