软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(8):3831−3857 [doi: 10.13328/j.cnki.jos.007249] [CSTR: 32375.14.jos.007249]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

可编程数据平面 DDoS 检测与防御机制*

武文浩[1], 张磊磊[1], 潘 恒[1], 李恩晗[1], 周建二[2], 李振宇[1]

[1](中国科学院 计算技术研究所, 北京 100190)

[2](南方科技大学, 广东 深圳 518055)

通信作者: 潘恒, E-mail: panheng@ict.ac.cn

摘 要: 传统的分布式拒绝服务攻击 (DDoS) 检测与防御机制需要对网络流量进行镜像、采集以及远程集中式的攻击特征分析, 这直接造成额外的性能开销, 无法满足高性能网络的实时安全防护需求. 随着可编程交换机等新型网络设备的发展, 可编程数据平面能力得到增强, 为直接在数据面进行高性能的 DDoS 攻击检测提供了实现基础. 然而, 当前已有的基于可编程数据面的 DDoS 攻击检测方法准确率低, 同时受限于编程约束, 难以在可编程交换机 (如 Intel Tofino) 中进行直接部署. 针对上述问题, 提出了一种基于可编程交换机的 DDoS 攻击检测与防御机制. 首先, 使用基于源目地址熵值差的攻击检测机制判断 DDoS 攻击是否发生. 在 DDoS 攻击发生时, 设计了一种基于源目地址计数值差的攻击流量过滤机制, 实现对 DDoS 攻击的实时防御. 实验结果表明, 该机制能够有效地检测并防御多种 DDoS 攻击. 相较于现有工作, 该机制在观察窗口级攻击检测中的准确率平均提升了 17.75%, 在数据包级攻击流量过滤中的准确率平均提升了 3.7%.

关键词: 分布式拒绝服务攻击; 可编程数据平面; 异常检测; P4; 网络安全

中图法分类号: TP309

中文引用格式: 武文浩, 张磊磊, 潘恒, 李恩晗, 周建二, 李振宇. 可编程数据平面DDoS检测与防御机制. 软件学报, 2025, 36(8): 3831–3857. http://www.jos.org.cn/1000-9825/7249.htm

英文引用格式: Wu WH, Zhang LL, Pan H, Li EH, Zhou JE, Li ZY. Detecting and Defending Mechanism Against DDoS Attacks in Programmable Data Plane. Ruan Jian Xue Bao/Journal of Software, 2025, 36(8): 3831–3857 (in Chinese). http://www.jos.org.cn/1000-9825/7249.htm

Detecting and Defending Mechanism Against DDoS Attacks in Programmable Data Plane

WU Wen-Hao[1], ZHANG Lei-Lei[1], PAN Heng[1], LI En-Han[1], ZHOU Jian-Er[2], LI Zhen-Yu[1]

[1](Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

[2](Southern University of Science and Technology, Shenzhen 518055, China)

Abstract: Traditional detection and defense mechanisms for distributed denial-of-service (DDoS) attacks require traffic mirroring, collection, and centralized remote analysis, which introduces extra performance overhead and fails to achieve real-time protection in high-performance networks. With the development of network devices such as programmable switches, the programmable data plane has emerged as a solid foundation for achieving high-performance DDoS attack detection. However, existing detection methods based on the programmable data plane cannot guarantee accuracy and are difficult to deploy directly in programmable switches (such as Intel Tofino) due to programming constraints. To this end, this paper proposes a programmable switch-based mechanism for detecting and defending against DDoS attacks. First, the mechanism uses the difference between the entropy of source and destination addresses to determine whether DDoS attacks occur. When DDoS attacks occur, a traffic filtration mechanism based on the difference in counts between source and destination address will defend against DDoS attacks in real time. Experimental results indicate that the proposed mechanism effectively identifies and defends against DDoS attacks. Compared with the benchmark method, the accuracy of this mechanism in window-

* 基金项目: 国家自然科学基金 (62002344, U20A20180, 62072437)
收稿时间: 2022-07-12; 修改时间: 2023-05-10, 2023-11-18, 2024-03-18, 2024-06-03; 采用时间: 2024-07-14; jos 在线出版时间: 2024-12-31
CNKI 网络首发时间: 2025-01-02

© 中国科学院软件研究所 http://www.jos.org.cn

level attack detection is increased by 17.75% on average, and the accuracy of packet-level attack filtration is increased by 3.7% on average.

Key words: distributed denial of service attack (DDoS); programmable data plane; anomaly detection; P4; cyber security

分布式拒绝服务攻击 (distributed denial of service, DDoS) 的检测与防御是网络安全人员重点关注的问题之一[1–3]. 随着网络技术的发展, DDoS 攻击的种类和数量都在不断增加, 根据 Telia Carrier 发布的 DDoS 威胁态势报告[4], 在 2020 年新冠疫情大流行开始后, DDoS 攻击的发送数量相较于 2019 年增长了 50%, 攻击流量的峰值跃升至 1.18 TB/s. 在 2021 年, 这一增长趋势仍未停止, 根据卡巴斯基 2021 年第四季度的 DDoS 攻击报告[5], 2021 年第四季度的 DDoS 攻击数量相较于 2020 年第四季度提高了约 365.22%. 日益泛滥的 DDoS 攻击已经成为现今互联网中最大的安全威胁之一. 因此, 如何实现高效而准确的 DDoS 攻击检测与防御对保证网络安全有着重要的意义.

传统 DDoS 攻击检测与防御机制的主要研究思路是通过对网络流量的镜像、采样、分析与建模, 获得流量的多维度特征, 并以此设计不同的 DDoS 攻击检测算法 (如基于随机森林的流量分类方法[6]、基于深度残差网络的异常流量检测算法[7]等), 从而实现对 DDoS 攻击流的精准识别. 然而, 网络流量的采集和集中式的流量分析必然会增加额外的性能开销, 无法实现实时、高效的网络安全防护.

随着可编程交换机、智能网卡等新型网络设备的快速发展, 可编程数据面[8]的能力得到不断增强, 可实现网络数据包的边传边算. 通常来讲, 可编程数据面被抽象成 Match+Action 的流水线模式, 利用 P4[9]等数据面编程语言定制化数据包的处理逻辑, 可实现网络流量及状态的实时监控、计算与更新等一系列网络任务. 可编程数据面直接面对网络数据包, 对其进行转发、处理与操作. 因此, 在可编程数据面实现在网的 DDoS 攻击检测, 可实现攻击流量的更早发现及防御策略的更早部署, 从而满足实时防护需求. 因此, 研究并实现基于可编程数据面的 DDoS 攻击检测与防御机制对保障高性能网络安全有着重要意义.

当前, 学术界已广泛开展相关研究, 但已有工作或需要网络控制面频繁参与决策, 从而导致整个 DDoS 攻击检测与防御性能的急剧降低, 或受限于硬件可编程交换机 (如 Intel Tofino) 严格的资源约束, 难以将 DDoS 攻击检测与防御机制部署在真实的可编程数据面. 例如, 文献 [10] 中提出的 Jaqen 在可编程交换机中利用 Sketch 与计数器等方法收集网络中的流量信息并上报至控制平面, 并在控制平面中实现多种类型的 DDoS 攻击检测. 这种方法导致 DDoS 攻击检测过程中数据面与控制面需要频繁交互, 降低性能. 为此, 文献 [11] 进一步提出了 Euclid. 这项工作是当前唯一一种能够完全部署在可编程数据面的 DDoS 攻击检测与防御机制, 可在不中断原有正常流量的情况下实现有效的网络安全防护. 然而, 这一工作所提出的检测算法在 DDoS 攻击流量占比较低时难以保证检测的准确率, 同时检测算法操作复杂, 仅能部署于虚拟可编程软件交换机 (behavioral model version 2, BMv2)[12]上. 导致上述部署受限问题的原因在于真实的可编程交换机对计算与存储资源的使用有着严格的限制, 大部分成熟的检测算法难以直接进行部署. 总结而言, 其关键挑战在于以下几个方面.

- 可编程交换机逐包处理模式. 相比于基于采样或直接镜像的流量采集方法, 可编程交换机无法在一次计算中同时获得多个数据包的信息, 仅允许在每个数据包到达时进行处理和状态更新.
- 可编程交换机计算能力受限. 可编程交换机仅支持简单的整数算术运算, 不支持对数、熵值等复杂运算类型及浮点数的计算.
- 可编程交换机流水线 stage 资源受限. 在数据面编程中, 具有相互依赖的网络功能处理逻辑需要被编译部署到不同的交换机流水线阶段 (stage) 之上. 因此, 复杂网络功能需要消耗大量的交换机流水线 stage, 难以在当前流水线 stage 资源受限 (如 12 个 stage) 可编程交换机上进行实现部署.

为了解决现有工作中检测准确率不足、检测机制难以真实部署的问题, 本文设计并实现了一种可在真实可编程交换机完整部署的 DDoS 攻击检测与防御机制, 在达到攻击检测高性能的同时满足高检测准确率. 针对检测准确率低的问题, 本文提出了一种基于熵差的 DDoS 攻击检测机制以及基于源目地址计数值差的攻击防御机制, 能够实现在网的 DDoS 攻击检测以及攻击数据包的快速过滤, 可显著提高检测准确率. 针对在真实可编程交换机中

© 中国科学院软件研究所 http://www.jos.org.cn

部署难的问题, 本文提出了一系列关键技术以应对上述关键挑战.

● 针对可编程交换机对逐包处理模式的限制, 本文采用了一种逐包累加的方法, 通过原始的单次计算转换成单个时间窗口的数据包熵值增量的累加.

● 针对可编程交换机计算能力受限的挑战, 本文提出了一种基于查找转发的计算方法, 将计算结果以表格的形式预先存储在可编程交换机上, 最后通过操作数的查表获得计算结果.

● 针对可编程交换机中流水线资源限制问题, 本文提出了一种多组 Sketch 轮换的新型数据结构, 能够在有限的流水线阶段中实现对网络中多个观察窗口的状态保存.

本文基于 Intel Tofino 可编程交换机实现了上述 DDoS 检测与防御机制, 并通过 BOUN DDoS 数据集[13]进行了实验验证. 实验结果表明, 与已有工作 Euclid 相比, 观察窗口级攻击检测中的准确率平均提升了 17.75%, 在数据包级攻击流量过滤中的准确率平均提升了 3.7%.

本文第 1 节介绍本文相关技术背景, 包括 P4 语言、可编程交换机和 Sketch 数据结构. 第 2 节介绍可编程数据平面应用于 DDoS 攻击检测的相关工作. 第 3 节介绍本文所实现的 DDoS 攻击检测与防御机制的基本原理与设计. 第 4 节介绍基于可编程交换机的 DDoS 攻击检测与防御的实现机制. 第 5 节通过对比实验来验证所提模型的有效性. 第 6 节总结全文.

1 背景知识

本文基于可编程交换机设计了一种 DDoS 攻击检测与防御机制. 下面将对相关的概念和基本知识予以介绍, 包括 P4 语言与可编程交换机和 Count-Min Sketch 数据结构.

1.1 P4 语言与可编程交换机

P4 语言是一种协议无关的数据平面编程语言[9]. 与原有的 SDN 和 OpenFlow[14]等网络技术相比, P4 语言和协议无关的交换机架构使得网络管理人员能够实现对数据包的自定义转发和处理. 图 1 展示了使用 P4 程序实现数据平面可编程的过程. 用户首先需要基于特定的交换机架构编写 P4 程序, 这一部分中主要包括一组基于有限状态机的解析器以及基于匹配-动作流水线的控制流程. P4 编译器执行的工作主要分为两部分, 首先其将 P4 代码转化为可编程交换机硬件中可以识别的二进制配置文件; 同时, 编译器会根据用户在数据面中所定义的控制逻辑, 为控制面生成可用的 API. 可编程交换机在运行 P4 程序时, 可以通过控制面程序对交换机中的匹配-动作表, 计数器, 寄存器等结构进行配置, 从而实现控制面与数据面的交互.

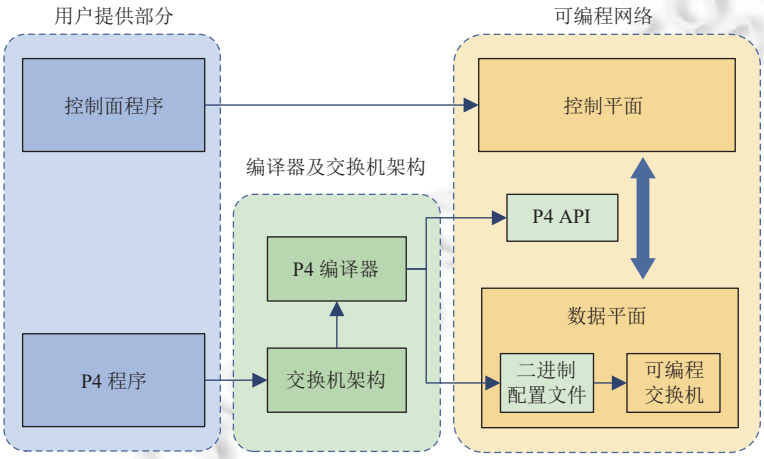


图 1 使用 P4 进行数据平面编程

在数据平面编程中所涉及的交换机架构是 P4 语言的编程模型, 其作为 P4 源代码与可编程设备之间的中间层, 为开发者提供了 P4 语言运行流程的逻辑视图. 交换机架构中将会为开发者提供 P4 基本操作以外的附加功能,

© 中国科学院软件研究所 http://www.jos.org.cn

例如校验和、哈希、随机数、寄存器等. 目前有多种不同的交换机架构以及对应的可编程设备, 例如支持 V1model 的 BMv2 软件交换机, 支持协议无关交换机架构 (protocol independent switching architecture, PISA) 的 Tofino[15]交换芯片. 其中, PISA 架构主要由一个可编程的解析器、一个可编程的逆解析器和一个由多个阶段组成的可编程匹配动作流水线组成. 其中, 解析器主要负责数据包头部信息的提取, 这一模块中允许开发人员通过有限状态机实现自定义的数据包头解析过程. 可编程匹配动作流水线由多个匹配动作单元组成, 每个单元包含一个或多个匹配动作表 (match-action table, MAT), 由开发者编写的 P4 程序将在编译器中转化为表的形式并在匹配动作流水线中执行. 在最后的逆解析器中, 数据及其头部信息以指定的方式重新序列化并离开交换机.

Tofino 交换芯片是第 1 个支持 PISA 架构的以太网交换 ASIC, 其架构如图 2 所示. 在 Tofino 的架构中包括固定的零阶段与内在元数据模块 (phase-0 and intrinsic metadata, PIM)、分组缓冲与复制模块 (packet buffer and replication engine, PRE) 以及队列缓冲模块 (buffer queuing engine, BQE). Tofino 中提供基于 PISA 架构的可编程入口模块与可编程出口模块, 这两个模块中分别包含一组完整的解析器、匹配动作流水线和逆解析器.

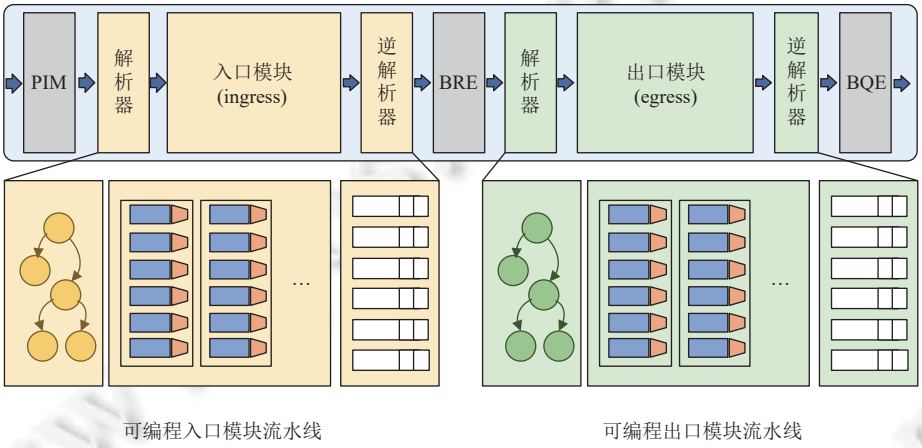


图 2 Tofino 架构示意图

1.2 Count-Min Sketch 数据结构

Count-Min Sketch[16]是一种由多个哈希表组成的数据结构, 这一结构用于统计在一组数据中各个不同元素的出现频次. 这一数据结构在原有基于哈希表的简单计数方法上, 通过同时使用多组哈希表来缓解由于哈希冲突产生的计数误差. 图 3 展示了 Count-Min Sketch 的工作原理, 该结构首先将会使用多组互不相关的哈希函数计算元素关键值的哈希结果, 之后在每个哈希表的对应位置将计数值加 1. 当需要进行查询时, Count-Min Sketch 将会依次查询每一个哈希表中对应位置的计数值, 并输出其中最小的一个作为查询结果.

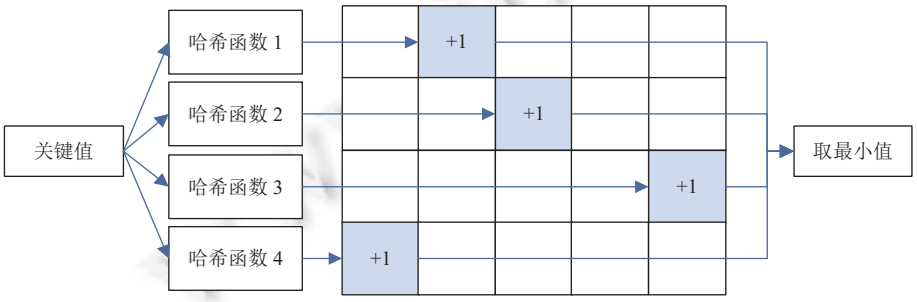


图 3 Count-Min Sketch 工作方式

© 中国科学院软件研究所 http://www.jos.org.cn

2 相关研究工作

2.1 传统的 DDoS 攻击检测与防御方法

传统的 DDoS 攻击检测机制需要收集网络中的流量信息并进行集中式的攻击特征分析, 例如采用 sFlow[17]进行数据包采样或使用 NetFlow[18]获取流量统计信息. 在此类方法中, 相关工作基于已有的网络流量数据集重点研究 DDoS 攻击检测中的具体检测算法, 而较少关注 DDoS 攻击检测与防御机制的部署方式以及数据采集过程在数据和控制平面之间产生的额外开销. 根据检测方法的不同, 现有的研究工作可以分为如下的两类.

(1) 基于统计特征的检测方法. 此类检测方法中计算当前网络流量的一个或多个统计特征, 并与网络正常工作时的状态进行比较, 若统计特征发生了超过阈值的变化则可以判定 DDoS 攻击的发生. DDoS 攻击检测中常用的统计特征包括相关性、熵、协方差、标准差等[19]. Tao 等人[20]使用数据流的熵值检测局域网中的 DDoS 攻击, 若该熵值在短时间内急剧下降, 则会发出 DDoS 攻击警报. Fortunati 等人[21]利用协方差矩阵判断当前流量与正常流量之间是否存在差异, 从而实现对网络中异常流量行为的检测.

(2) 基于机器学习的检测方法. 此类检测方法中将 DDoS 攻击检测问题视为一个区分正常流量与恶意流量的二分类问题, 首先需要训练样本对模型进行训练, 并使用训练完成的分类器对当前网络中的流量进行识别. 此类机制中常用的分类方法包括支持向量机[22]、决策树[23]、神经网络[24]等. Hou 等人[25]利用 NetFlow 对流量特征进行提取, 并设计了一种基于随机森林的 DDoS 攻击检测器. Christopher 等人[26]针对物联网场景设计了一种基于双向长短期记忆的递归神经网络的攻击检测机制, 用于检测由大规模僵尸网络发起的多种 DDoS 攻击.

2.2 基于可编程数据面的 DDoS 攻击检测与防御方法

可编程数据面的出现为 DDoS 攻击检测与防御机制的设计提供了新的思路, 在现有的部分研究工作中, 可编程数据平面已经被应用到原有的检测机制中, 用于对部分模块进行改进或者承担一部分数据采集与统计的工作. Grigoryan 等人[27]针对 DDoS 攻击提供了一种 DDoS 攻击防御机制. 在数据包进入网路时, 可编程交换机会向包中添加唯一标识符, 数据包到达目标主机前最后一个 P4 交换机将存储标识符与源地址的映射. 如果上层程序中检测到某一 IP 地址疑似 DDoS 的攻击者, 将会通知交换机并丢弃带有该 IP 对应标识的所有包. Kuka 等人[28]利用可编程数据平面实现数据包中特定信息的提取并将其转发给集中的攻击检测模块 (例如具备 DDoS 攻击检测功能的 SDN 控制器), 攻击检测模块中将会基于阈值判别占比最大的前 N 条流并将其过滤. 文献 [29−31] 中使用机器学习方法实现 DDoS 攻击检测机制. 其利用 P4 交换机实现数据收集, 并在控制平面上使用决策树、支持向量机等模型对收集到的数据流进行分类. 文献 [32] 中提供了一种基于可编程数据平面的网络测量系统. 此类方法在交换机中利用 Sketch 与计数器等方法收集网络中流量的信息并上报至控制平面, 同时, 为用户提供了多种可用的原语, 通过不同原语的组合使用实现对指定数据包的过滤. 以上的工作使用可编程数据面完成 DDoS 攻击检测中的数据采集工作, 但其核心的 DDoS 检测算法仍在控制面或远程服务器上独立执行, 难以实现实时的攻击检测与安全防护.

为了能够将更多的攻击检测与防御功能卸到可编程数据平面上来完成, 文献 [11,33] 中在可编程数据平面中通过计算流量的地址熵实现对恶意流量的检测, 一旦源地址熵产生突增或目的地址熵产生突降, 该算法将会认为该窗口中 DDoS 攻击发生. 文献 [34] 中将上述基于地址熵的检测作为一种初检方法, 在可编程交换机检测到 DDoS 攻击时将异常流量输入到控制器中使用深度神经网络进行复检. 这一类工作中, 数据平面已经初步具有 DDoS 攻击检测与防御能力, 但未完全考虑到真实可编程交换机的限制, 仅在可编程软件交换机模型中实现, 难以适用于真实环境.

我们在表 1 中详细对比了现有基于可编程数据平面的 DDoS 攻击检测机制存在的不足. 现有的同类型 DDoS 攻击检测机制仅能部署在模拟环境中, 无法适应真实可编程交换机的计算类型与计算资源限制. 具体而言, 文献 [33] 提出的 P4DDoS 重点关注可编程数据平面中的对数运算与熵值运算, 该方法中所提出的熵值计算方法超过了 Tofino 交换机的资源限制, 无法完全部署于可编程交换机硬件中. 同时, 该方法仅使用熵值对网络状态进行判断,

© 中国科学院软件研究所 http://www.jos.org.cn

并未设计完整的 DDoS 攻击检测与防御流程. 文献 [11] 中提出的 Euclid 基于可编程数据平面, 使用地址熵的方式实现了 DDoS 攻击检测, 在记录当前观察窗口的流量信息的同时保存了安全状态下窗口的流量信息, 从而用于 DDoS 受害者的识别及 DDoS 数据包过滤, Euclid 使用过多的计算与存储资源, 仍无法满足可编程交换机的硬件资源限制. 为了保存安全窗口的流量信息, Euclid 中使用了不符合可编程交换机流水线要求的计算流程. 如图 4 所示, Euclid 使用多组 Sketch 进行流量信息的记录, 其窗口变化时计算流程为读取当前窗口 Sketch_curr 的记录值并记录在上一窗口 Sketch_last 中, 之后处理当前进入的数据包, 在处理完成后将更新后的流量信息写入当前窗口 Sketch_curr 中. 在这一流程中两个不同的流水线阶段均访问了同一组数据结构, 这使得该机制无法在真实可编程交换机上运行.

表 1　基于可编程数据面的检测方法对比

DDoS攻击检测方法	是否P4实现	数据面是否部署完整的检测与防御机制	是否满足硬件资源限制
P4DDoS[33]	是 (软件模拟环境)	否	否
文献[34]	是 (软件模拟环境)	否	否
Euclid[11]	是 (软件模拟环境)	是	否
本文机制	是 (真实交换机环境)	是	是

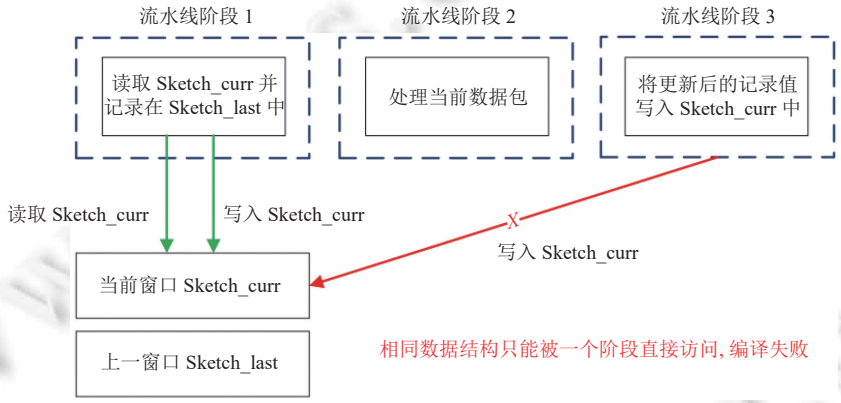


图 4　Euclid 部分运行流程

针对现有工作的不足, 本文提出了一种完全部署于可编程数据平面的 DDoS 攻击检测与防御机制, 使用地址熵的方式实现有效的 DDoS 攻击检测与防护. 与原有工作相比, 本文机制在检测效果上有明显的提升并能够完全部署于真实可编程交换机硬件上.

3　DDoS 攻击检测与防御方法

本节首先介绍 DDoS 攻击下的流量特征模式. 然后, 以此为基础提出了基于 IP 地址熵差的窗口级 DDoS 攻击检测方法, 以有效地解决异常流量占比相对较小时的误检率高的问题. 最后, 设计了基于源目地址计数值差的攻击数据包过滤机制, 以有效地实现 DDoS 攻击防御.

3.1　DDoS 攻击下的流量与 IP 地址变化特征

DDoS 攻击中, 攻击发起者通过大规模的僵尸网络向一个目标主机发动攻击, 利用大量的数据占用目标主机的网络与系统资源, 从而使得被攻击者无法为正常用户提供服务[35]. 正常流量与 DDoS 攻击流量在行为上的差异会导致网络中流量统计特性的变化[36].

图 5(a) 展示了在未受到攻击时, 合法用户中的正常流量经过可编程交换机与外部网络进行通信. 这一情况下,

© 中国科学院软件研究所 http://www.jos.org.cn

经过可编程交换机的流量在源目地址 IP 的分布情况上相似, 并且流量的各项统计特征在短时间内不会产生较大的变化. 图 5(b) 表示 DDoS 攻击出现时网络中的新增流量变化, 由攻击发起者控制的多台僵尸主机将向可编程交换机另一侧的特定被攻击者发送大量数据. 与正常状态相比, 经过交换机的数据包的源地址的基数将会增大且会变得更加分散, 而目的地址的分布将会集中于被攻击主机.

(a) 正常流量 (b) DDoS 攻击流量

图 5 正常流量与 DDoS 攻击流量

3.2 基于 IP 地址熵差的窗口级 DDoS 攻击检测

如第 3.1 节中所述, DDoS 攻击发生时流量的显著特征是在源目地址的分布上产生了不同的变化. 针对这一类问题, 通常可以使用熵度量状态的多样性与分散程度. 熵的概念由德国物理学家克劳修斯[37]于 1865 年提出, 并最初用于描述一个系统内在性质的改变, 这一概念在统计物理学领域有着广泛的应用. 香农[38]于 1948 年将这一概念拓展到通信领域, 并开创了信息论这一学科. 香农熵用于描述信源中一个消息所提供的平均信息量, 表示为对一个随机变量的不确定度的度量. 假设 X 为一个离散随机变量, 其样本空间为 $X=\{x_1,x_2,x_3,\ldots,x_n\}$, 则香农熵 S 的定义可以表示为公式 (1) 所示形式.

$$S=-\sum_{i=0}^{n}p(x_i)\log_2(p(x_i)) \tag{1}$$

当随机变量的样本空间越大, 概率分布越分散时, 随机变量的熵值越大. 网络中经过可编程交换机的数据包的 IP 地址可以视为一个离散随机变量, 通过估计数据包 IP 地址的熵值可以直观地判断网络中流量的分布是否发生了变化. 以源 IP 地址为例, 首先按顺序将到达可编程交换机的每 m 个数据包划分为一个观察窗口, 之后会统计每一种源 IP 在这一观察窗口中出现的次数, 并以频次估计每一种样本的出现概率, 则此时源 IP 地址熵定义如公式 (2) 所示.

$$S_{\text{src}}=-\sum\nolimits_{i=0}^{n}\frac{k_i}{m}\log_2\left(\frac{k_i}{m}\right) \tag{2}$$

其中, m 表示一个观察窗口中的数据包数, n 表示当前窗口中共有 n 种不同的 IP 地址, k_i 表示第 i 种 IP 地址在当前观察窗口中的出现频次. 地址熵越大, 表示当前窗口中的 IP 地址分布越分散.

在正常状态下, 源地址熵与目的地址熵随观察窗口的变化趋势如图 6(a) 所示, 在较短的时间范围内, 源目地址的熵值都在一定范围中上下波动. 同时, 源地址熵与目的地址熵两者有着明显的相关性, 源地址熵与目的地址熵在每一个观察窗口中的计算结果都相近, 并且在发生波动时有着相同的变化趋势. 图 6(b) 为 DDoS 攻击状态下源地址熵和目的地址熵随观察窗口的变化情况. 在观察窗口序号 52 后的网络中发生了 UDP 洪泛攻击, 这一事件的发生导致源地址的熵有着明显的升高, 而目的地址的熵值则会下降. 同时, 两者之间的熵值的差异相对于正常状态显著提升.

基于上述的特征, 分析熵值的变化可以有效地判断网络中是否出现了 DDoS 攻击检测. 现有的基于熵值的流量异常检测方法分别检测源地址与目的地址是否超过或低于某一阈值, 从而判断是否发生异常[18,19]. 这一类方法在 DDoS 攻击流量占比较大 (如超过 20%) 时具有较好的效果, 但在异常流量占比相对较小时将会产生较大的误

© 中国科学院软件研究所 http://www.jos.org.cn

检率. 这一缺陷产生的原因在于此类方法仅关注源目地址自身的变化, 未能利用源地址与目的地址熵的相关性, 从而容易被网络中正常的熵值波动所影响.

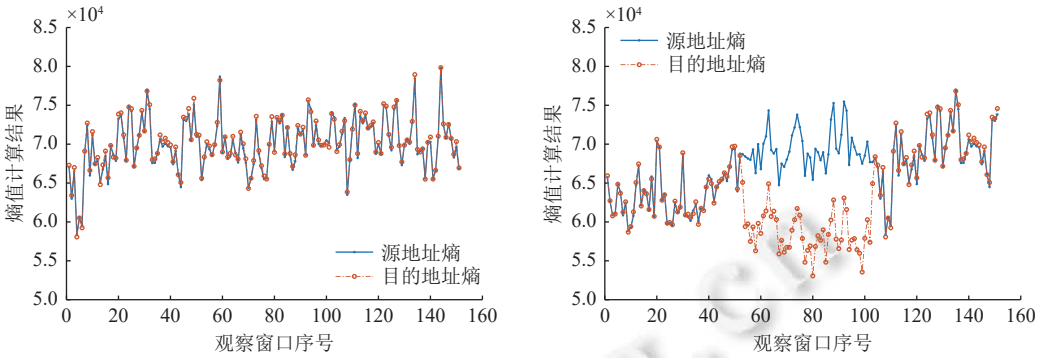


图 6 安全状态与 DDoS 攻击状态

为了解决这一问题, 本文采用基于熵差的 DDoS 检测机制. 首先将计算得到的源地址熵与目的地址熵记为 S_{src} 和 S_{dst}, 通过分析源目地址的熵值之差, 可以判断当前窗口中是否含有 DDoS 攻击流量, 若之差超过预先设定的阈值, 则认为当前的观察窗口中可能包含 DDoS 攻击流量, 这一过程可以由公式 (3) 表示.

$$\begin{cases} S_{\text{src}} - S_{\text{dst}} > T_S, & \text{DDoS} \\ S_{\text{src}} - S_{\text{dst}} \leqslant T_S, & \text{Safe} \end{cases} \tag{3}$$

其中, T_S 表示一个预先设定的固定阈值. 在此判别方式下, 熵值差仍可能会由于正常波动, 使得在个别时间点中正常状态下的熵值差超过阈值或在受到攻击的状态下熵值差小于阈值. 为了避免这一现象所导致的假阳性与假阴性错误, 仅当连续 a 个观察窗口中的熵差均高于阈值时, 观察窗口才会被认定为受到 DDoS 攻击. 同理, 当该机制目前认为网络正发生 DDoS 攻击时, 仅当连续 a 个观察窗口中的熵差均低于阈值时, 观察窗口才会被认定 DDoS 攻击结束, 回到正常状态, 否则将会保持原有的状态不变. 其中, 参数 a 的值可以根据具体网络状况由用户定义.

3.3 基于源目地址计数值差的攻击数据包过滤

基于熵差的 DDoS 攻击检测机制仅能实现观察窗口级的攻击检测, 无法查找到具体的被攻击主机 IP 地址, 从而难以进行有效的防御. 为解决这一问题, 在 DDoS 攻击检测机制检测到观察窗口发生 DDoS 攻击后, 利用网络中的流量特征实现一种恶意流量分离机制, 从而实现对 DDoS 攻击的防御. 从总体上看, 本文机制所采用的是一种两段式的 DDoS 攻击检测与防御, 仅当基于熵差的检测机制报告 DDoS 攻击发生时才会启动恶意流量的分离. 相比于原有的直接基于阈值的过滤方法[39], 通过两段式的配合避免了在 DDoS 攻击未发生时对合法流量产生的误检.

对于被攻击者的识别以及攻击流量的过滤问题, 采用了一种基于源目地址计数值差的攻击数据包过滤机制. 在 DDoS 攻击发生时网络中将新增大量僵尸主机产生的攻击流量, 此时可编程交换机中记录的源目 IP 地址计数值将会发生不对称的变化, 对于源地址, 新增的攻击流量在源 IP 地址上的计数值会被大量攻击者分散, 每个攻击者的源 IP 地址计数值不会产生大幅上升; 对于目的地址, 被攻击者 IP 地址的计数值会因为攻击流量的集中流入而显著提升, 从而产生了源目地址计数不对称变化的现象. 为了衡量上述不对称变化特征, 针对某一对源目 IP 地址的出现频次, 使用公式 (4) 衡量其不对称特征.

$$D = k_{\text{dst}} - k_{\text{src}} \tag{4}$$

其中, k_{dst} 是该目的 IP 地址在当前观察窗口中的计数值, k_{src} 是当前窗口中源地址的计数值, D 是两者之差. 与正常状态下相比, 当目的 IP 地址受到 DDoS 攻击时, 其计数值会显著上升, 但对于源地址而言, 由于僵尸网络中大量主机对攻击任务的分摊, 其计数值上升并不明显. 因此, 二者的计数值差在发生 DDoS 攻击时将会升高. 根据这一原理, 设计了如公式 (5) 所示的规则来判断目的 IP 地址是否受到攻击.

© 中国科学院软件研究所 http://www.jos.org.cn

$$\begin{cases} D_{\text{now}} - D_{\text{safe}} > T_D, & \text{目标IP受到DDoS} \\ D_{\text{now}} - D_{\text{safe}} \leqslant T_D, & \text{目标IP未受攻击} \end{cases} \tag{5}$$

其中, D_{now} 表示当前观察窗口中源目地址计数值之差, D_{safe} 表示在 DDoS 攻击未发生的情况下对应源目地址的计数值之差, T_D 为一个预先设定的阈值. 如果当前窗口中源目地址计数值之差相比于正常窗口下的增大幅度超过阈值, 则认为这一 IP 受到了攻击, 并且在 DDoS 攻击状态结束之前始终过滤掉目的地址为该 IP 的数据包, 从而实现对 DDoS 攻击的防御.

4 基于可编程交换机的 DDoS 攻击检测与防御机制设计

本节首先介绍整个 DDoS 攻击检测与防御系统的整体架构, 然后分别从地址频率统计与状态保存、IP 地址熵值在网计算以及 DDoS 攻击状态判定和准确防御等关键技术出发, 介绍其在可编程交换机上的具体实现细节.

4.1 系统工作流程与关键技术

为了实现基于熵的 DDoS 攻击检测以及基于源目地址计数值差的攻击数据包过滤机制, 需要完成的流程如图 7 所示, 具体的流程如下. (1) 当数据包到达可编程交换机后, 按顺序将到达可编程交换机的数据包划分为大小为固定数量的观察窗口, 可编程交换机首先统计每一种源 IP 在这一观察窗口的出现次数. (2) 获取对应 IP 在上一安全观察窗口中出现的次数. (3) 之后, 根据公式 (2) 所示的熵值计算方法计算当前窗口的源目地址熵. (4) 根据计算得到的熵值判断是否发生 DDoS 攻击. (5) 基于源目地址计数值差判断当前数据包类别. (6) 最后, 保存被判断为受到 DDoS 攻击的 IP 地址, 并且在 DDoS 攻击状态结束之前过滤掉目的地址为该 IP 的数据包. 这一操作流程能够在常规的计算方式 (如远程服务器或控制平面程序) 中通过简单的代码实现, 但由于可编程交换机对于计算与存储资源的严格限制, 在真实的可编程交换机中实现以上的全部流程面临着以下的难点.

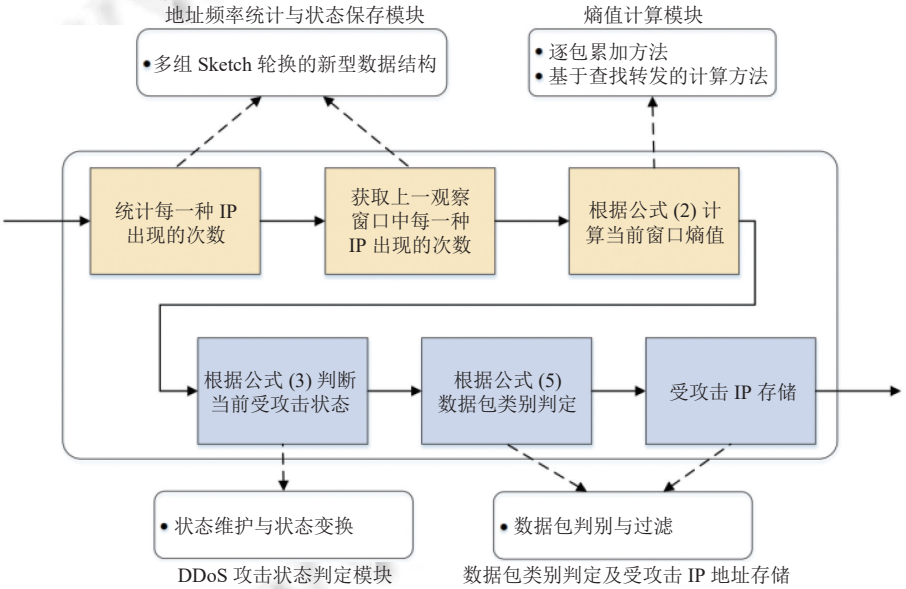


图 7 DDoS 攻击检测与防御机制流程与关键技术

● 难以在统计当前观察窗口 IP 出现次数的同时保存上一安全窗口的统计信息. 由于可编程交换机流水线 stage 资源受限, 同时保留多个观察窗口的数据意味着需要多组相互依赖的网络功能处理逻辑. 现有的技术难以在有限的流水线 stage 中实现复杂的 DDoS 攻击检测与防御功能的同时完成多组网络统计信息的保存.

© 中国科学院软件研究所 http://www.jos.org.cn

● 难以在可编程数据面上完成熵值计算. 由于可编程交换机仅支持简单的整数算术运算, 不支持对数、熵值等复杂运算类型及浮点数的计算, 并且无法在一次计算中同时获得多个数据包的信息, 这导致难以直接对所有数据包统计信息进行熵值计算.

为了解决以上的问题, 本文中针对可编程数据面的特性及计算与存储限制, 对上述流程中的步骤设计了适应可编程交换机的计算模块, 其中包含一种多组 Sketch 轮换的新型数据结构用于实现对多个窗口统计信息的记录与保存, 同时, 使用了逐包累加的熵值计算方式以及基于查找转发的计算方法, 实现对熵值的计算. 各模块功能与关键技术如下所示.

● 地址频率统计与状态保存 (第 4.2.1 节). 使用多组 Sketch 轮换的新型数据结构, 完成多个观察窗口统计数据的实现对观察窗口内 IP 地址频率的估计和保存, 同时完成观察窗口结束时所需的自动刷新.

● 熵值计算模块 (第 4.2.2 节). 使用逐包累加的熵值计算方法, 将原始的单次计算转换成单个时间窗口的数据包熵值增量的累加, 在每个数据包到达时对熵值进行更新, 无须获取其他数据包信息; 同时, 使用基于查找转发的计算方法, 完成计算中所涉及的对数、熵值等复杂运算.

● DDoS 攻击状态判定模块 (第 4.2.3 节). 通过多个计数器实现包含安全状态与 DDoS 攻击状态的有限状态机, 实现对当前窗口的 DDoS 攻击检测及当前系统所处状态的保存.

● 数据包类别判定及受攻击 IP 地址存储 (第 4.2.4 节). 使用来自此前模块中的源目地址计数值及状态信息判断当前数据包类别, 并将判断为 DDoS 攻击类型的数据包的目的 IP 存入哈希表.

4.2 关键模块工作原理

4.2.1 地址频率统计与状态保存模块

在本文所述机制中, 实现熵值计算和 DDoS 攻击防御需要统计并保存观察窗口中各 IP 地址的出现频次. 针对这一问题, 采用一种拓展的 Count-Min Sketch 结构实现对观察窗口内 IP 地址频率的估计, 同时完成观察窗口结束时所需的自动刷新. 与普通的 Count-Min Sketch 结构不同, 本次设计为每一个观察窗口进行编号, Count-Min Sketch 中将会同时记录该键值的计数与这一计数值对应的观察窗口序号. 在进行更新时, 如果 Count-Min Sketch 内部记录的窗口号与当前窗口一致, 则该位置计数值增加. 若与当前窗口序号不一致, 则说明观察窗口发生了变化, 此时将会把被选中位置的计数值置为 1 以实现自动的刷新. 为了同时计算源目地址的频率值并保存正常状态的信息, 交换机中将会同时维护 4 个上述数据结构, 其中每两个 Count-Min Sketch 为一组, 记录同一个观察窗口中的源 IP 地址和目的 IP 地址的出现频率. 两组数据结构将分别处于更新状态和查询状态并在观察窗口结束时根据当前网络状态进行轮换, 其中处于更新状态的 Count-Min Sketch 将会记录当前窗口中各 IP 的计数值, 而处于查询状态的 Count-Min Sketch 则会保持原有计数值不变.

如图 8 所示, 数据包到达后首先会查询一个标志寄存器, 并在该寄存器所指定的 Count-Min Sketch 上进行更新. 例如, 图中该寄存器的值为 0, 会更新上方两个编号为 0 的 Count-Min Sketch, 而另一组则不更新计数值, 仅进行查询操作. 此时, 下方编号为 1 的 Count-Min Sketch 中所保留的即为上一观察窗口中的各 IP 地址的计数值.

通过在每个窗口结束翻转标识寄存器的值, 两组数据结构将会轮换地进行更新, 从而实现在完成当前计数的同时保留上一个窗口中的计数状态. 为了给后续的 DDoS 攻击防御机制提供网络在正常状态下的计数值, 需要保证用于保存之前窗口计数值的一组 Count-Min Sketch 中的计数状态是在没有发生 DDoS 攻击的情况下得到的. 因此, 在多组数据结构进行轮换时会检查上一观察窗口中是否发生了 DDoS 攻击, 若上一窗口中发生了 DDoS 攻击, 则标志寄存器不会翻转. 算法 1 演示了两组数据结构在观察窗口结束时的轮换. 首先判断观察窗口是否结束, 并更新窗口序号 (第 1、2 行), 在安全状态的观察窗口结束后标志寄存器翻转 (第 3、4 行), 两组 Sketch 将会轮换工作. 当 DDoS 攻击发生时, 标识寄存器不会进行翻转 (第 5、6 行), 仍然会使用与上一个窗口中相同的 Sketch 进行计数, 此时另一组 Sketch 中的安全状态计数值将会得到保留. 最终这一模块将会为接下来的计算提供当前数据包的源目 IP 在本观察窗口中的计数值以及在正常状态下的计数值.

© 中国科学院软件研究所 http://www.jos.org.cn

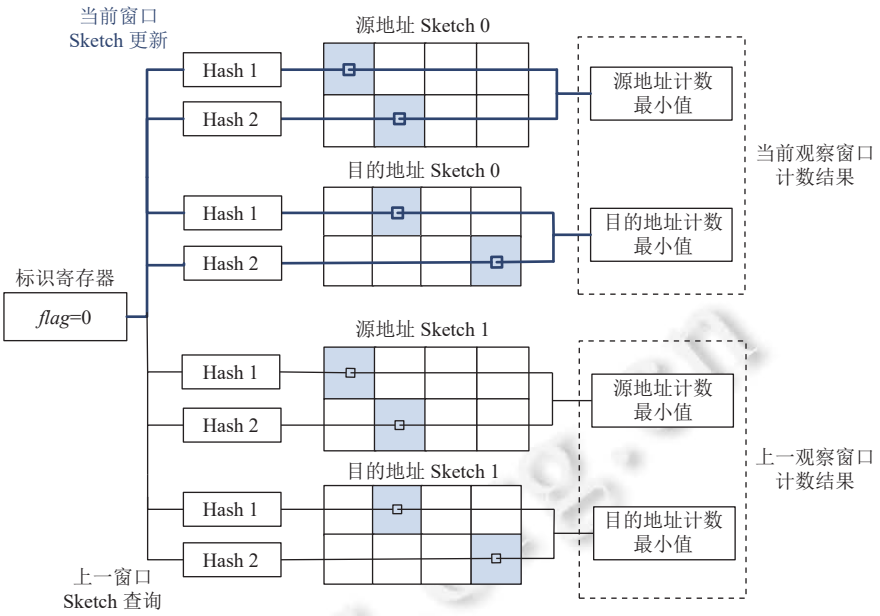


图 8 Sketch 轮换工作与副本保存原理

算法 1. 标识寄存器更新.

输入: 数据包 *p*;

输出: 标识寄存器值 *flag*.

1. **if** *p.resubmit_flag* = 1 // 数据包的重发标识 *resubmit_flag* 为 1, 上一观察窗口结束
2. *Wid* = *Wid* +1; // 观察窗口序号 *Wid* 更新
3. **if** *p.r.ddos_flag* = 1 // 上一窗口中未发生 DDoS 攻击, 重发包头 *r* 中的标识位为 0
4. *flag* = ~*flag*; // 标识寄存器翻转
5. **else**
6. *flag* = *flag*; // 标识寄存器保持不变
7. **end**
8. **end**

4.2.2 熵值计算模块

熵值计算模块使用前文中多组 Sketch 得到的当前观察窗口源目地址计数值计算当前的源目地址熵. 由于可编程交换机的限制, 数据平面中的单次处理仅能使用当前数据包的头部信息, 且数据平面中无法直接支持循环操作及浮点数运算. 为了解决以上问题, 本模块中采用了一种逐数据包累加的方式进行熵值更新, 并将实际的熵值扩大一定的倍数后近似为整数, 同时, 将复杂计算操作的计算结果预先配置在查找转发表中, 使用查找转发的计算方法进行对数等计算.

(1) 逐数据包累加的计算方法

原有的熵值计算方法中需要同时计算所有类别 IP 地址的计数信息, 由于可编程数据面中仅能处理当前数据包的头部信息, 难以同时获取所有 IP 的计数值, 因此本文实现了一种逐数据包累加的计算方法, 每次计算时仅需要使用单个数据包的头部信息, 如图 9 所示.

在这一计算方法的具体实现中, IP 地址熵值计算的公式可以改写为如公式 (6) 所示的形式.

© 中国科学院软件研究所 http://www.jos.org.cn

$$S' = m\log_2(m) - \sum_{i=0}^{n}\sum_{x=1}^{k_i}(x\log_2(x) - (x-1)\log_2(x)) \tag{6}$$

其中, S' 为实际熵值的 m 倍, m 为一个观察窗口中数据包的数量. 当得到一个 IP 地址对应的计数值 x 后, 通过添加新熵值与原有熵值之间的差以进行更新. 若交换机中所记录的当前熵值为 S'_{now}, 则新熵值 S'_{next} 的计算方式如公式 (7) 所示.

$$S'_{\text{next}} = S'_{\text{now}} - (x\log_2(x) - (x-1)\log_2(x)) \tag{7}$$

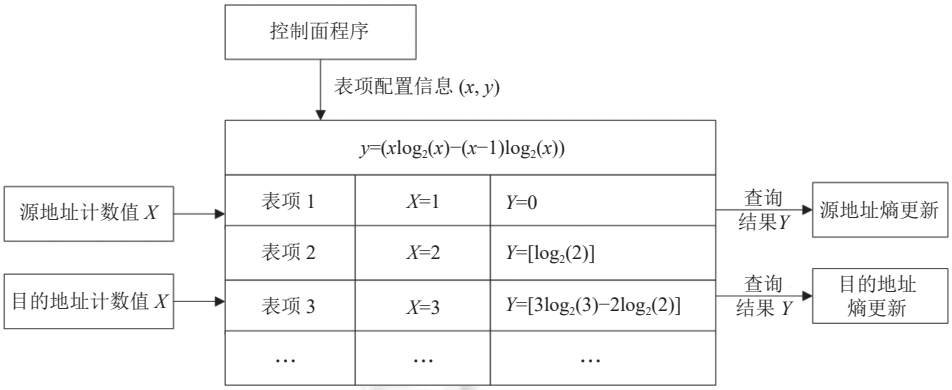


图 9 基于查找转发 IP 地址熵计算

当一个新的数据包到达交换机时, 只需要根据观察窗口中源地址及目的地址现有的计数值计算熵值的更新量 y, y 与 IP 地址计数值 x 的对应关系如公式 (8) 所示.

$$y = x\log_2(x) - (x-1)\log_2(x) \tag{8}$$

在得到上述更新值后, 可编程交换机将依照公式 (7) 更新当前所存储的源目地址熵. 当一个观察窗口结束时, 可编程交换机中所存储的累计计算结果即为该观察窗口源目地址熵的计算结果, 可编程交换机在完成后续的判别操作后将会重置熵值并开始下一轮计算, 这一过程如公式 (9) 所示.

$$S' = m\log_2(m) \tag{9}$$

使用以上的逐数据包累加的计算方法, 可以实现每次计算仅使用当前数据包的头部信息, 并且在一个观察窗口结束时能够输出正确的熵值计算结果.

(2) 查找转发的复杂函数计算方法

针对可编程交换机中无法完成的复杂计算, 本文机制中采用表的方式, 利用控制面程序将所有可能出现的计算结果预先配置在可编程交换机内. 该模块的工作原理如图 10 所示, 熵值的更新量会预先计算并近似为整数, 通过控制面程序配置入可编程交换机的表中. 由于观察窗口的大小为 m, 表中只需要维护 m 条表项即可覆盖 IP 计数值所有可能的取值. 完成计算和更新后, 该模块将会输出当前窗口中的源地址熵和目的地址熵.

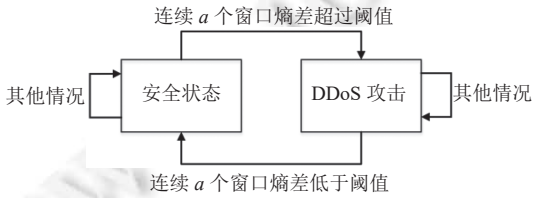


图 10 攻击检测机制状态转换

4.2.3 DDoS 攻击状态判定模块

DDoS 攻击状态判定模块根据公式 (3) 的判别原理判断当前是否处于 DDoS 攻击状态. 该模块中维护如图 10

© 中国科学院软件研究所 http://www.jos.org.cn

所示的具有两个状态的有限状态机, 在安全状态时, 若连续 a 个观察窗口中的熵差均高于阈值, 观察窗口被认定为受到 DDoS 攻击, 系统跳转至 DDoS 攻击状态; 当该机制目前认为网络正发生 DDoS 攻击时, 仅当连续 a 个观察窗口中的熵差均低于阈值时, 观察窗口认定 DDoS 攻击结束, 回到正常状态.

DDoS 攻击判定模块中通过维护两个计数器实现所需的状态转换方法, 这一流程如算法 2 所示. 在一个观察窗口结束时, 首先根据会上一模块中的源目地址熵计算结果判断熵差是否超过固定阈值. 若超过阈值则将正常状态窗口计数器归零, 并将 DDoS 攻击状态窗口计数器加 1; 若未超过固定阈值则会进行相反操作 (第 1–7 行). 在该模块结束时将检查两个计数器中的值, 若 DDoS 攻击状态窗口计数值超过设定的参数 a, 则认为这一窗口受到了 DDoS 攻击 (第 8、9 行); 若正常状态窗口计数值超过 a, 则认为该窗口中没有受到 DDoS 攻击 (第 10、11 行). 若二者均在 a 以下, 则保持状态寄存器中的值不变, 认为这一窗口与上一个窗口的状态相同 (第 12–14 行).

算法 2. DDoS 攻击状态判定.

输入: 当前观察窗口源地址熵 S_{src}, 目的地址熵 S_{dst};
输出: DDoS 攻击判定结果 *ddos_flag*.

1. **if** $S_{\text{src}} - S_{\text{dst}} > T_S$ // 源目地址熵值差超过固定阈值 T_S
2. $C_{\text{safe}} = 0$; // 正常状态窗口计数器 C_{safe} 置 0
3. $C_{\text{ddos}} = C_{\text{ddos}} + 1$; // DDoS 状态窗口计数器 C_{ddos} 加 1
4. **else** // 源目地址熵值差未超过阈值
5. $C_{\text{ddos}} = 0$; // 正常状态窗口计数器 C_{ddos} 置 0
6. $C_{\text{safe}} = C_{\text{safe}} + 1$; // DDoS 状态窗口计数器 C_{safe} 加 1
7. **end** // 标识寄存器保持不变
8. **if** $C_{\text{ddos}} > a$ // 正常状态窗口计数器超过 a, 连续 a 个窗口熵值差超过阈值
9. *ddos_flag* = 1; // 这一窗口被判定为 DDoS 攻击状态
10. **else if** $C_{\text{safe}} > a$ // 连续 a 个窗口熵值差低于阈值
11. *ddos_flag* = 0; // 这一窗口被判定为正常状态
12. **else** // 两个计数器均小于 a
13. *ddos_flag* = *ddos_flag*; // 当前窗口状态与上一窗口状态相同
14. **end**

4.2.4 数据包类别判定模块及受攻击 IP 地址存储模块

数据包在入口流水线中经过 DDoS 攻击检测流程后, 在桥接包头中携带检测结果与源目 IP 地址的计数值进入出口流水线. 出口流水线中将利用桥接包头中所携带的计算结果判断该数据包的目的 IP 是否受到了 DDoS 攻击. 各模块的工作流程如后文图 11 所示, 首先将会读取当前窗口中的 DDoS 攻击状态, 若 DDoS 攻击发生, 则根据公式 (5) 的原理判断源目 IP 地址之差的变化是否超过阈值 T_D. 当超过固定阈值时, 认为该数据包指向的 IP 地址受到 DDoS 攻击. 此时会计算这一 IP 的哈希值并将其存储在哈希表中的对应位置. 以上流程结束后, 交换机会检查该 IP 在哈希表中的对应位置是否有值, 若发现对应位置非零, 则会将该数据包丢弃.

4.3 基于 Tofino 架构的 DDoS 攻击检测与防御流程

4.3.1 基于 Tofino 架构的机制部署

基于 Tofino 架构实现所述的攻击检测与防御机制的总体设计与流程如图 12 所示, 主要包含控制平面和基于 Tofino 架构的可编程数据平面. 其中, 控制平面负责可编程交换的表项配置和状态查询, 在交换机启动后, 控制平面程序将通过插入表项的方式完成可编程交换机所需的配置, 之后将会循环查询当前网络的状态以及出口模块中

© 中国科学院软件研究所 http://www.jos.org.cn

被标记为疑似被攻击的 IP 地址, 并将其显示给网络管理人员. 控制平面并不参与 DDoS 攻击检测及防御过程, 这使得检测机制能够彻底回避由于控制平面与数据平面交互过程产生的时延.

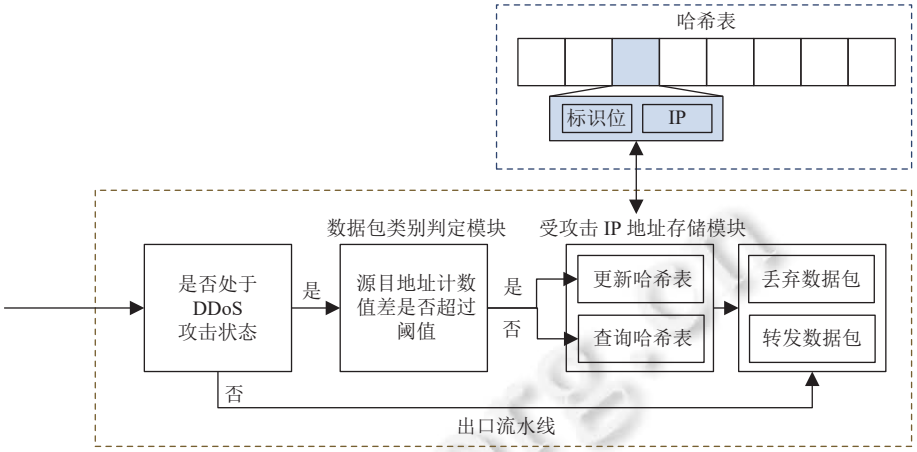


图 11 DDoS 攻击防御

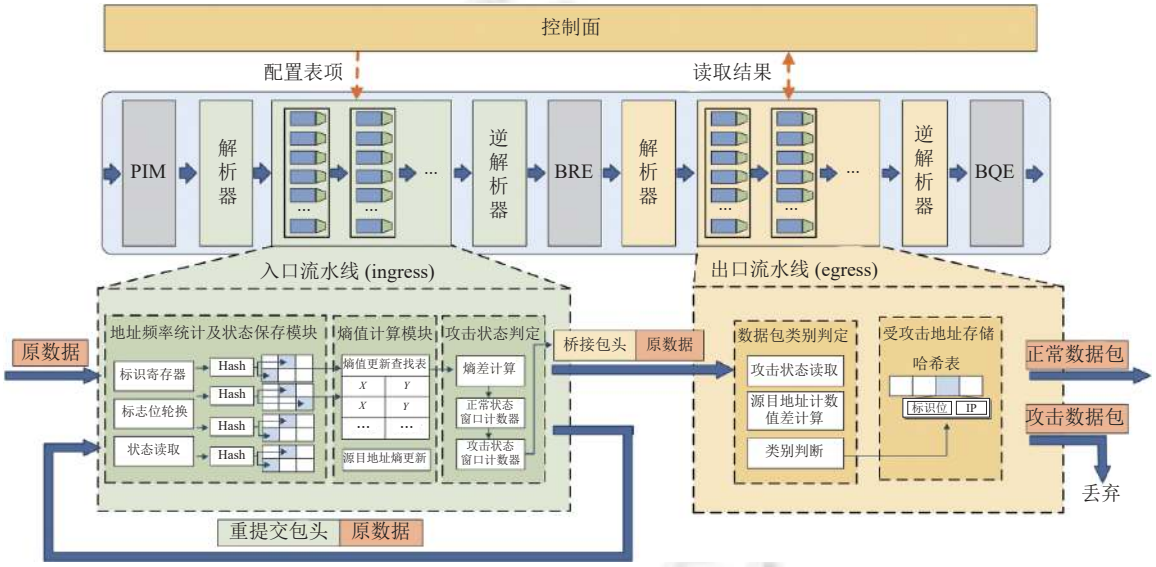


图 12 DDoS 攻击检测预防与机制的数据面实现

数据平面中的 DDoS 检测与防御机制分别部署到可编程交换机的入口与出口流水线, 采用这一设计方式的原因在于 Intel Tofino 可编程交换机单条流水线长度固定, 通过将检测与防御分开部署可实现资源物理隔离, 降低资源冲突. 其中, 入口流水线包含地址频率统计及状态保存模块、熵值计算模块、攻击状态判定模块; 出口流水线包含数据包类别判定及受攻击 IP 存储模块.

(1) 入口流水线 DDoS 攻击检测. 本文在可编程交换机入口流水线 (ingress) 中实现基于熵差的 DDoS 攻击检测机制, 其主要包括地址频率统计及状态保存模块、熵值计算模块、攻击状态判定模块这 3 个子模块. 其工作流程如下. 1) 首先通过在地址频率统计及状态保存模块中维护两组基于 Count-Min Sketch 的计数结构, 通过两组 Sketch 的轮换工作保存每个源目 IP 地址在当前观察窗口以及上一个安全状态的观察窗口内的出现频次. 2) 其次, 在熵值计算模块中根据当前窗口的计数值更新源地址熵和目的地址熵. 3) 之后, 在攻击状态判定模块判断源目地

© 中国科学院软件研究所 http://www.jos.org.cn

址熵值之差是否超过固定阈值. 4) 最后, 在入口流水线的末尾会将 DDoS 攻击检测结果和 IP 计数值信息封装在桥接包头中发送至出口流水线. 同时, 若当前观察窗口结束, 则使用可编程交换机提供的重发机制修改存储在上一级流水线中的 DDoS 攻击状态.

(2) 出口流水线 DDoS 攻击防御. 本文在可编程交换机出口流水线 (egress) 中实现基于源目地址计数值差的 DDoS 攻击防御机制, 主要包括数据包类别判定模块和受攻击 IP 地址存储模块这两个子模块. 其具体工作流程如下. 1) 首先, 数据包类别判定模块根据桥接包头中所封装的 DDoS 攻击检测结果判断当前观察窗口的状态, 若当前观察窗口中存在 DDoS 攻击, 则依据公式 (5) 判别该数据包的类别. 2) 之后, 若当前数据包的目的 IP 被判定为被攻击者, 则会在受攻击 IP 存储模块中使用哈希表存储该 IP, 并将后续流量中丢弃指向该 IP 的数据包, 从而实现 DDoS 攻击的实时防御.

4.3.2 基于数据包重提交机制的逆流水线信息传递

在入口流水线结束时, 采用重提交机制更新地址频率统计及状态保存模块中的标识位. 该操作仅在每观察窗口结束时进行一次. 本文所述机制在多组 Sketch 结构轮换中标识寄存器需要根据上一窗口的最终检测结果进行转换, 这一需求导致了两个无法放置在同一级流水线上的操作需要读取或更改同一个寄存器中的值. 而在 Tofino 架构中, 无法在某一阶段中访问另一阶段流水线中的寄存器. 我们采用 Tofino 交换芯片所提供的重发机制解决这一问题: 在入口流水线的逆解析器中可以选择对数据包进行重提交操作, 该数据包将会重新发送, 进入入口流水线. 如图 13 所示, 每当一个观察窗口结束时, 本文机制将当前窗口的检测结果封装至数据包的头部并进行重发. 在数据包进入入口流水线后, 首先将会检测这一数据包的 resubmit_flag 标识, 若检测为重发数据包, 可编程交换机将会解析封装在数据包头部的上一窗口计数结果, 并判断是否翻转 Sketch 计数模块前的标识寄存器.

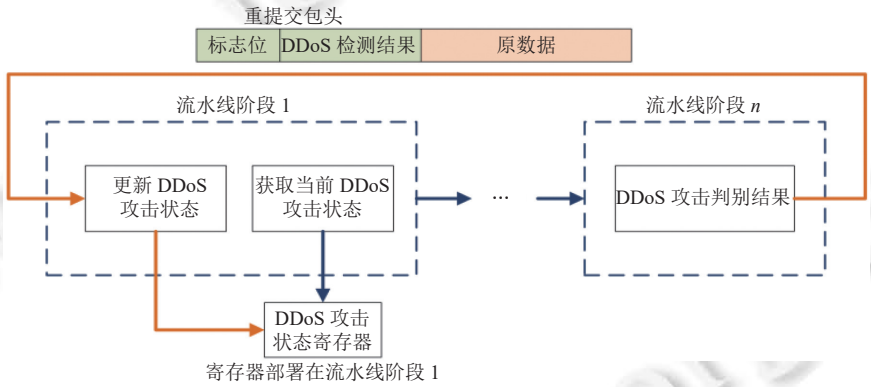


图 13 利用重提交机制实现信息传递

4.3.3 基于桥接数据包的出入口流水线信息传递

在入口流水线结束时, 采用桥接数据包头将入口流水线的计算结果传递至出口流水线. 在 $P4_{16}$ 标准下, 入口与出口流水线中不存在能够全局访问的用户自定义元数据 (metadata), 因此如果需要实现入口流水线与出口流水线之间的信息传递, 则需要定义额外的数据包头部进行桥接. 如后文图 14 所示为本文机制所定义的桥接头部以及信息传递流程, 在出口流水线中将会把关键的计算结果封装为桥接头部并解析. 桥接头部中包含的信息有: 当前窗口的 DDoS 检测结果, 当前数据包的源地址及目的地址 IP 在本观察窗口的总计数值, 以及在上一个安全状态观察窗口中的总计数值.

5 实验分析

5.1 系统部署

基于本文中所提出的架构, 使用 P4 语言实现了该 DDoS 攻击检测与防御机制, 并将其部署在 Intel-Tofino 可

© 中国科学院软件研究所 http://www.jos.org.cn

编程交换机中. 在数据平面的实现中包含了在第 4 节中所涉及的全部 DDoS 攻击检测与防御模块, 基于 P4 语言和 Tofino 交换机架构实现的 DDoS 攻击检测与防御机制被抽象成多个匹配-动作表的流水线模式, 其中 DDoS 攻击检测机制部署于可编程交换机入口流水线, 包括地址频率统计及状态保存模块、熵值计算模块、攻击状态判定模块; DDoS 攻击防御机制部署于可编程交换机的出口流水线, 包括数据包类别判定模块和受攻击 IP 地址存储模块, 通过将检测与防御分开部署可有效地实现资源物理隔离, 降低资源冲突. 在控制平面的实现中, 使用 Python 语言实现了该 DDoS 攻击检测与防御机制的控制功能. 控制平面的主要功能包括向数据平面中的查找表中预先配置熵值计算中所需要的表项, 同时通过 API 查询数据平面中对应寄存器获取数据平面状态. 当 DDoS 攻击检测与防御机制启动时, 需要先将 P4 代码部署于可编程交换机的数据平面, 然后运行控制面 Python 代码完成数据面表项配置, 之后, 交换机中部署的 DDoS 攻击检测与防御机制将会自动检测通过交换机的所有流量中是否存在 DDoS 攻击.

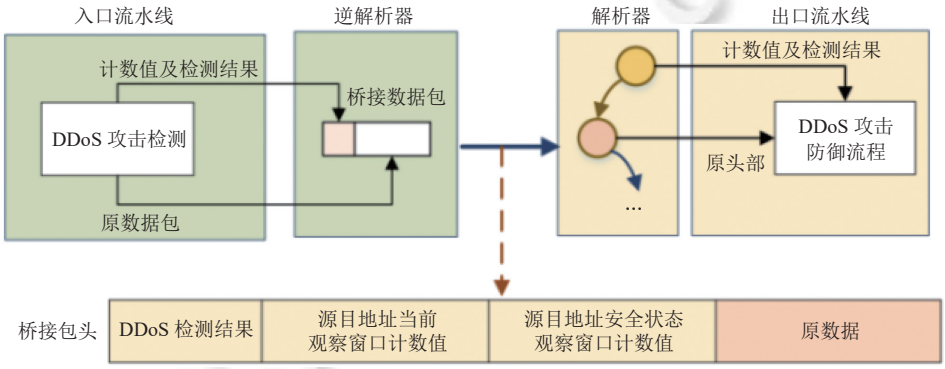


图 14 利用桥接数据包实现出入口流水线信息传递

本文的测试工作中, 在实验室使用 Intel-Tofino 可编程交换机搭建了如图 15 所示的实验网络. 该实验网络中多台用户主机产生的流量将会通过一台 Tofino 可编程交换机流向另一侧的目的主机. 实验中所使用的背景流量来自 BOUN DDoS 数据集, 使用 tcpreplay 对该流量记录进行修改与重放, 含有 TCP-SYN 洪泛攻击、UDP 洪泛攻击、DNS 反射放大攻击、ICMP 洪泛攻击的流量将会穿过可编程交换机到达被攻击者. 在固定参数的设定上, 观察窗口大小设置为 10 000, 熵差的固定阈值 T_S 设定为 512, 连续窗口数量 a 设定为 3, DDoS 防御机制中计数值差变化量的阈值 T_D 设定为 300.

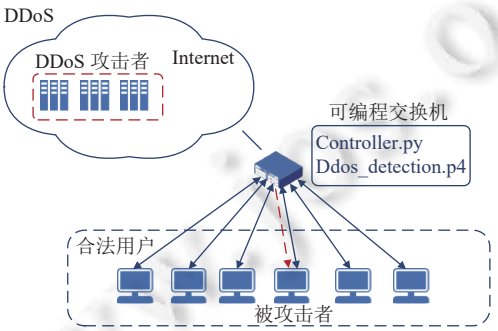


图 15 原型系统

5.2 实验数据

实验中使用 BOUN DDoS 数据集[13]对本文中所提出的 DDoS 攻击检测机制进行检验. 这一数据集包含两组以 csv 格式记录的网络流量, 两组网络流量数据中分别发生了 TCP-SYN 洪泛攻击和 UDP 洪泛攻击. 在每一组数

© 中国科学院软件研究所 http://www.jos.org.cn

据中, DDoS 攻击源将分 4 次对目标主机进行攻击, 且 4 次攻击中的攻击速率分别为每秒 1 000、1 500、2 000 和 2 500 个数据包. 被攻击者的 IP 地址为 10.50.199.86, 通过数据包的目的 IP 地址可以区分攻击数据包和合法数据包. TCP-SYN 洪泛攻击和 UDP 洪泛攻击数据集各有 4 个攻击周期. 并且 4 次攻击的强度依次增大. 其中, TCP-SYN 洪泛攻击的攻击数据包约占对应时期总数据包的 4.88%–9.77%, UDP 洪泛攻击数据包约占对应时期总数据包的 12.16%–18.76%.

本文基于 BOUN DDoS 数据集中 UDP 洪泛攻击数据集的背景流量, 进一步使用 DNS 反射放大攻击和 ICMP 洪泛攻击拓展数据集中包含的 DDoS 攻击方式. 我们将原有数据集中的 UDP 洪泛攻击流量全部剔除, 并在对应时刻以不同比例向背景流量中混合入 DNS 反射放大攻击流量与 ICMP 洪泛攻击流量. 其中, DNS 反射放大攻击数据包约占对应时期总数据包的 6.25%–12.5%, ICMP 洪泛攻击数据包约占对应时期总数据包的 2.5%–12.5%. 如表 2 所示为 4 组 DDoS 攻击的详细数据.

表 2　DDoS 攻击数据集

数据集分组	攻击轮次	攻击开始序号	攻击结束序号	攻击数据包数量	数据包总量	攻击流量占比
TCP-SYN洪泛攻击	1	1945608	2335362	19035	389781	0.0488
	2	3784781	4240070	27121	455289	0.0596
	3	5571097	5959329	35936	388232	0.0926
	4	7440584	7885602	43465	445018	0.0977
UDP洪泛攻击	1	1048852	1354950	37216	306098	0.1216
	2	2539197	2931244	55029	392065	0.1404
	3	4234356	4702829	75023	468473	0.1601
	4	6015917	6513625	93378	497708	0.1876
DNS反射放大攻击	1	1040001	1379985	21249	339984	0.0625
	2	2510001	2959997	37499	449996	0.0833
	3	4250001	4749991	49999	499990	0.1000
	4	6030001	6539993	63749	509990	0.1250
ICMP洪泛攻击	1	1040001	1379961	8501	339960	0.0250
	2	2510001	2959989	22499	449988	0.0500
	3	4250001	4749991	49999	499990	0.1000
	4	6030001	6539993	63751	509990	0.1250

5.3　评价指标

本文所提出的 DDoS 攻击检测机制能够有效地在可编程交换机计算与存储能力限制实现准确的 DDoS 攻击检测与防御. 针对本文所提出的方法, 将从 DDoS 攻击检测有效性和检测效率两个方面设计对应的评价指标.

(1) DDoS 检测准确性

DDoS 攻击检测与防御机制中观察窗口的攻击检测和数据包的过滤问题都可以视为一个二分类问题, 根据数据包的实际情况以及检测值, 将会产生真阳性 (true positive, *TP*)、假阳性 (false positive, *FP*)、真阴性 (true negative, *TN*)、假阴性 (false negative, *FN*) 这 4 种情况. 本文中使用精确率 (*Precision*)、召回率 (*Recall*) 和 *F*1 值评价 DDoS 攻击检测算法的有效性. 其中, 精确率表示被预测为 DDoS 攻击的样本中真实发生 DDoS 攻击的样本占比. 这一指标的值越高, 说明 DDoS 攻击检测中被误报为 DDoS 攻击的安全流量越少. 召回率表示全部 DDoS 攻击的样本中被检测为 DDoS 攻击的样本占比. *F*1 值综合反映了这两项指标的大小. 以上 3 项指标的计算方法如公式 (10) 所示.

$$\begin{cases} Precision = TP/(TP+FP) \\ Recall = TP/(TP+FN) \\ F1 = 2\cdot Precision\cdot Recall/(Precision+Recall) \end{cases} \tag{10}$$

(2) DDoS 检测性能

相比于原有的部署方式, 本文所提机制能够将 DDoS 攻击检测机制完全的部署于可编程数据面上. 这一部署

© 中国科学院软件研究所 http://www.jos.org.cn

方式能够显著地提高 DDoS 攻击检测效率, 更适用于高性能网络. 我们使用 DDoS 攻击检测延时对不同的 DDoS 部署方法的检测性能进行评估. 我们将 DDoS 攻击检测延时定义为在单次 DDoS 攻击中, 网络中第 1 个包含 DDoS 攻击数据包的观察窗口出现至 DDoS 攻击防御机制部署到数据面的时间. DDoS 攻击检测延时的值越低, 说明检测机制的检测性能越高.

5.4 对比实验方法

本文中重点关注能够完全部署于可编程数据平面上的 DDoS 攻击检测与防御机制. 在实验中, 本文所设计的机制将与现有的研究工作进行比较. 选取的对比机制包括 Jaqen[10]和 Euclid[11]. 同时, 为了评价可编程数据面部署的优势, 我们使用传统的检测方法进行流量采集, 即将所有数据包的头部信息采集至控制面或远程服务器, 并在控制面上使用本文所提出的检测机制进行 DDoS 攻击检测. 这 4 种方法的对比如表 3 所示.

表 3 DDoS 攻击检测与防御机制对比

名称	是否可以在Tofino交换机中实现	是否无须针对特定攻击类型进行配置	是否数据平面无须与控制平面交互	DDoS检测机制部署方式
本文机制	是	是	是	可编程数据面
Jaqen	是	否	否	可编程数据面 + 控制面查询接口
Euclid	否	是	是	可编程数据面 (BMv2模拟)
流量采集	否	是	否	控制面程序

其中, Jaqen 在可编程交换机中利用 Sketch 与计数器等方法收集网络中流量的信息并上报至控制平面. 同时, 在控制平面和前端页面中为用户提供了多种可用的原语, 从而在交换机上部署不同的测量任务. 通过不同原语的组合使用可以实现对多种类型 DDoS 攻击的检测与防御. 针对本文中所需的实验, 使用 Jaqen 提供的原语实现了 4 种基于阈值的 DDoS 攻击检测机制, 所检测的 DDoS 攻击类型分别为 TCP SYN 洪泛攻击、UDP 洪泛攻击、DNS 反射放大攻击和 ICMP 洪泛攻击. Jaqen 需要预先根据所需检测的 DDoS 攻击类型, 使用不同的控制原语.

Euclid 能够完全在数据平面实现细粒度、低开销、低延迟的 DDoS 攻击检测和防御. Euclid 单独考虑源目 IP 地址熵的变化情况, 并利用基于指数加权移动平均 (EWMA) 的自适应阈值实现 DDoS 攻击检测, 在受到 DDoS 攻击时将会过滤攻击数据包. 由于检测算法操作复杂且缺乏针对真实可编程交换机硬件的部署技术, Euclid 仅能部署于虚拟可编程软件交换机 BMv2 上, 无法在真实的可编程交换机环境中部署.

在流量采集方法中, 本文将通过交换机的数据包的头部信息全部采集至控制面, 并在控制面使用第 3 节中所提出的检测机制进行 DDoS 攻击检测. 在这一对比方法中, 检测机制在控制平面通过高级语言实现, 不再受到可编程交换机计算及存储资源的限制, 但在流量采集过程中引入了额外的通信开销与检测延迟.

5.5 DDoS 攻击检测结果与分析

5.5.1 总体检测结果

为了评估本文所设计的 DDoS 攻击检测与防御机制的有效性, 使用第 5.1 节中所述的数据集进行实验, 并与文献 [10,11] 中所提出的 Jaqen 机制和 Euclid 机制进行对比. 图 16 所示为网络流量在熵值上的变化, 其中展示的结果为可编程交换机中的计算值, 即实际熵值的 10 000 倍. 图 16(a) 为 TCP-SYN 洪泛攻击下对应的源目地址熵值变化, 在这一组数据中, DDoS 攻击将分为 4 轮, 依次在观察窗口的第 194–234 号、第 378–425 号、第 557–596 号、第 744–789 号出现. 该网络中的背景流量变化速度较快, 且熵值前后波动较大, 在第 100 号观察窗口中的背景流量的分布产生突变. 此类突变会显著地影响部分基于单个熵值阈值进行判别的算法, 例如对比算法 Euclid 在这一数据集中会将背景流量熵值的正常波动误报为 DDoS 攻击. 图 16(b) 为 UDP 洪泛攻击部分的网络流量在熵值上的变化, 在这一数据集中, DDoS 攻击依次在观察窗口的第 105–136 号、第 254–294 号、第 424–471 号、第 602–652 号出现. 在 DDoS 攻击期间, 源目地址的熵值产生了明显的差异, 并且能够观察到源地址熵的激增以及目

© 中国科学院软件研究所 http://www.jos.org.cn

的地址熵的突减. 在图 16(c) 与图 16(d) 中, DNS 反射放大攻击与 ICMP 洪泛攻击均在观察窗口的第 103–137 号、第 250–295 号、第 424–474 号、第 602–653 号出现. 攻击发生时, 源目地址的熵值仍会有差异性的变化, 但在攻击流量占比较低时, 熵的变化幅度较小, 并且从单个熵值的变化趋势上难以直接区分此类 DDoS 攻击状态与正常网络状态.

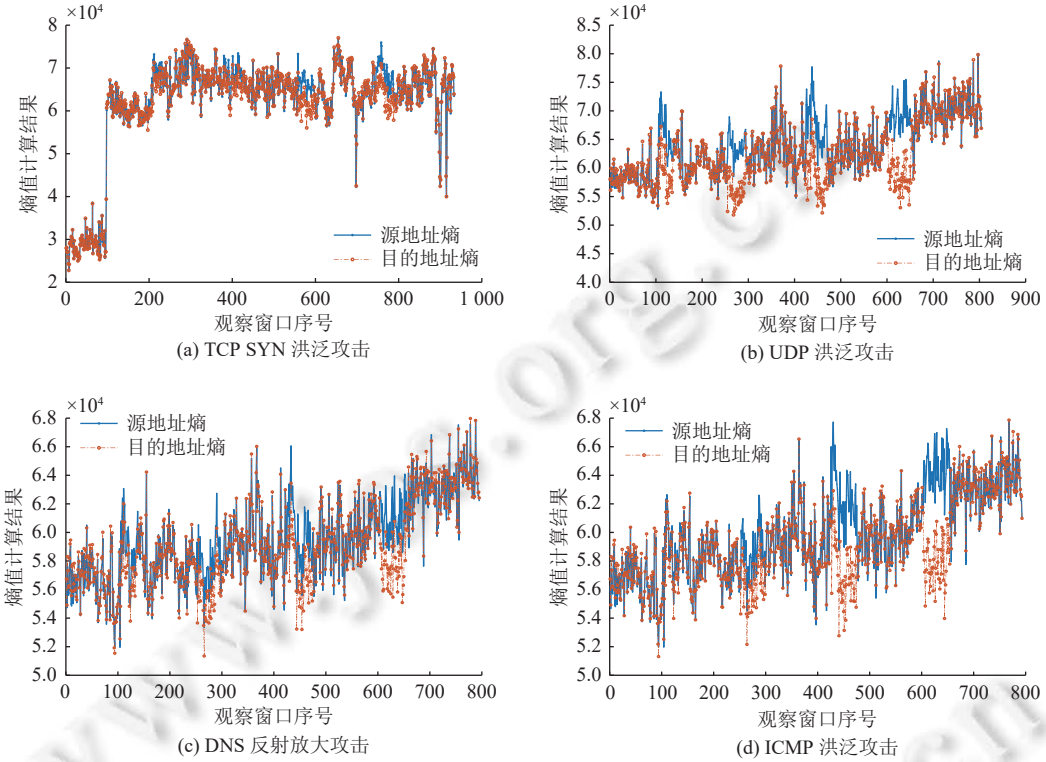


图 16 源目地址熵值计算结果

表 4 展示了 4 种 DDoS 攻击检测与防御机制在 TCP-SYN 洪泛攻击和 UDP 洪泛攻击中的详细实验结果. 数据共分为 4 部分, 分别为 TCP-SYN 洪泛攻击、UDP 洪泛攻击、DNS 反射放大、ICMP 洪泛攻击下的测试结果. 测试结果中包含 DDoS 攻击检测机制对当前观察窗口状态的检测结果, 以观察窗口为单位; DDoS 攻击数据包过滤机制对具体数据包进行的过滤情况, 以数据包为单位.

从测试结果来看, 本文所设计的 DDoS 攻击检测与防御机制能够有效地检测并防御 DDoS 攻击流量, 在 4 种攻击方式下, 观察窗口的检测准确率分别达到 97.6%、97.0%、97.2%、95.7%, 数据包的检测准确率分别达到 99.6%、99.8%、99.5%、99.6%; 同时, 在精确率与召回率上均有良好的表现.

与对比机制 Jaqen 相比, Jaqen 在 TCP SYN 洪泛攻击、DNS 反射放大攻击、ICMP 洪泛攻击这 3 种攻击类型中能够取得较高的检测准确率, 但在 UDP 洪泛攻击中, 本文所提机制比 Jaqen 有明显提升. 产生这一结果的原因在于, Jaqen 中基于阈值的检测策略能够有效地检测出网络中依赖特定协议实现的网络攻击, 而 UDP 洪泛攻击不依赖特定协议实现, 且网络中存在较大比例的合法 UDP 流量, 此时, 基于阈值的检测策略将会有较高的误检率. 同时, Jaqen 对于 DDoS 攻击的检测与防御需要获取数据包的具体协议类型, 依赖控制面原语, 即在检测进行之前, 需要针对性地配置相应的检测机制, 所实现的检测机制部署完成后只能对预期的攻击类型生效. 而本文所实现的机制利用攻击流量的网络特征完成 DDoS 攻击检测与防御, 无须预先了解将会发生的攻击类型, 可以检测任意能够改变网络中源目地址熵的 DDoS 攻击类型.

© 中国科学院软件研究所 http://www.jos.org.cn

与对比机制 Euclid 相比, 本文机制在检测结果的各方面指标上均有明显的提升. 这一提升的根本原因在于, 本文所提出的攻击检测方法深入地考虑了源目地址熵之间的关系, 能够有效地区分 DDoS 攻击与正常的网络流量变化. 而 Euclid 分别考虑源目地址熵是否产生突增或突减, 易受到背景流量变化的影响, 当 DDoS 攻击流量占比较低熵值或合法网络流量产生突变时, 会产生较大的误检与漏检现象.

表 4 实验结果

DDoS攻击种类	检测级别	检测方法	总样本	DDoS样本	准确率	精确率	召回率	*F*1值
TCP-SYN洪泛攻击	观察窗口状态检测	本文机制	932	172	0.976	0.921	0.953	0.937
		流量采集	932	172	0.990	0.966	0.982	0.973
		Euclid	932	172	0.790	0.419	0.380	0.399
		Jaqen	932	172	0.996	1.000	0.982	0.991
	DDoS攻击数据包过滤	本文机制	9329393	125557	0.996	0.812	0.867	0.839
		流量采集	9329393	125557	0.996	0.783	0.916	0.844
		Euclid	9329393	125557	0.969	0.171	0.346	0.229
		Jaqen	9329393	125557	0.999	1.000	0.999	0.999
UDP洪泛攻击	观察窗口状态检测	本文机制	804	172	0.970	0.925	0.936	0.931
		流量采集	804	172	0.978	0.937	0.965	0.951
		Euclid	804	172	0.801	0.528	0.663	0.588
		Jaqen	804	172	0.806	0.524	0.959	0.678
	DDoS攻击数据包过滤	本文机制	8047335	260646	0.998	0.973	0.977	0.976
		流量采集	8047335	260646	0.997	0.961	0.931	0.946
		Euclid	8047335	260646	0.951	0.375	0.689	0.486
		Jaqen	8047335	260646	0.970	0.517	0.983	0.678
DNS反射放大攻击	观察窗口状态检测	本文机制	795	180	0.972	0.925	0.956	0.940
		流量采集	795	180	0.979	0.922	0.989	0.954
		Euclid	795	180	0.733	0.373	0.261	0.307
		Jaqen	795	180	0.994	1.000	0.978	0.989
	DDoS攻击数据包过滤	本文机制	7950000	172496	0.995	0.897	0.851	0.873
		流量采集	7950000	172496	0.996	0.906	0.898	0.902
		Euclid	7950000	172496	0.951	0.240	0.579	0.340
		Jaqen	7950000	172496	0.999	1.000	0.998	0.999
ICMP洪泛攻击	观察窗口状态检测	本文机制	793	184	0.957	0.993	0.821	0.899
		流量采集	793	184	0.969	0.968	0.918	0.931
		Euclid	793	184	0.815	0.598	0.609	0.603
		Jaqen	793	184	0.994	1.000	0.978	0.989
	DDoS攻击数据包过滤	本文机制	7930083	144750	0.996	0.860	0.913	0.886
		流量采集	7930083	144750	0.996	0.883	0.925	0.903
		Euclid	7930083	144750	0.949	0.165	0.443	0.242
		Jaqen	7930083	144750	0.999	1.000	0.998	0.999

流量采集方法在控制面中使用了与本文机制所设计 DDoS 攻击检测方法. 这一方法在计算过程中不涉及可编程交换机计算与存储限制, 对浮点数与对数等计算相较于本文机制的真实可编程交换机硬件, 其实现更加准确. 在实验结果中, 此方法的攻击检测效果与本文机制接近, 可以说明本文所提出的检测与防御算法在硬件与软件中的不同实现方式均有良好的效果. 同时, 硬件实现方法在性能上有更好的表现 (见第 5.6 节).

5.5.2 检测原理分析

为了对实验结果产生的原因进行深层次的分析, 图 17 展示了在 UDP 洪泛攻击中, 本文机制和对比算法在各观察窗口的详细 DDoS 攻击检测结果. 其中, 每个观察窗口的检测结果按观察窗口序号 (即时间顺序) 依次排列, 分别由不同的颜色进行标识, 表示该观察窗口在对应 DDoS 攻击检测机制的检测过程中详细的所属状态 (真阳性、

© 中国科学院软件研究所 http://www.jos.org.cn

真阴性、假阳性、假阴性). 从检测结果来看, 本文所提出的检测机制在 DDoS 攻击检测中的表现相比于对比机制均有明显提升, 在完整的 DDoS 攻击流程中, 假阳性样本与假阴性样本数量明显少于 Jaqen 与 Euclid.

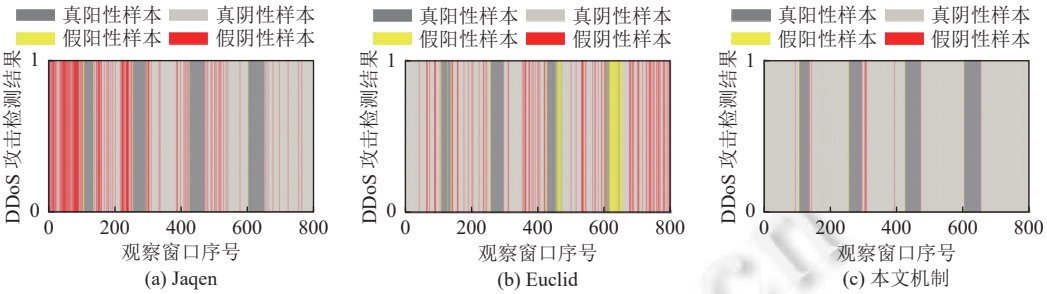


图 17 各检测机制检测结果对比

具体而言, 图 17(a) 展示了 Jaqen 机制的 UDP 洪泛攻击检测结果. Jaqen 机制基于流量阈值进行 DDoS 攻击检测. 因此, 在 DDoS 攻击流量占比较小 (即序号为 105–136 的观察窗口及此前的观察窗口) 时, 为了保证 DDoS 攻击检测的召回率, 较低的阈值设置会导致 Jaqen 将网络中合法的大流量错误地识别为 DDoS 攻击, 从而产生假阳性样本. 与 Jaqen 算法相比, 本文所提出的机制在序号为 0–100 的观察窗口中能够通过熵值差的变化准确地判断当前是否处于 DDoS 攻击状态, 有效地减少假阳性样本数量, 如图 17(c) 所示. 图 17(b) 展示了 Euclid 机制的 UDP 洪泛攻击检测结果. 作为一种基于熵值的检测机制, Euclid 在攻击流量占比较小时, 假阳性样本数量明显小于 Jaqen 机制. 而相比于本文机制, 在序号为 600 及后续的观察窗口中, 出现了大量的假阴性样本与假阳性样本. 产生这一显现的原因是, Euclid 分别考虑源目地址熵是否产生突增或突减, 当合法的背景网络流量产生突变时, 会产生误检与漏检现象. 图 18 进一步描述了产生这一现象的原因.

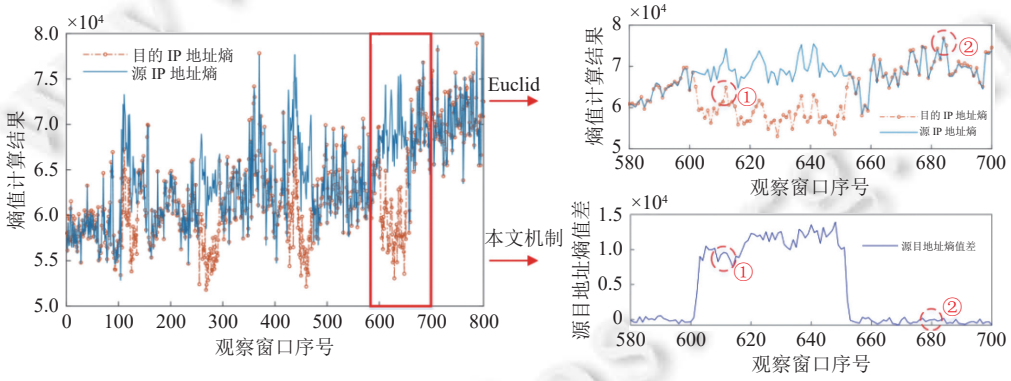


图 18 本文机制与 Euclid 检测现象对比

图 18 展示了本文机制与 Euclid 机制具体的检测细节. 图 18(a) 为 UDP 洪泛攻击中完整熵值计算结果, 针对其中检测错误情况出现较多的观察窗口 (序号 580–700), 在右侧展示了更加详细的熵值计算细节. Euclid 机制中分别考虑源地址熵与目的地址熵, 当源地址熵突增或目的地址熵突降时, Euclid 判定此时发生了 DDoS 攻击; 而本文机制计算源目地址熵值差, 并依据熵值差是否超过阈值进行 DDoS 攻击检测. Euclid 产生的误检情况主要包括: ① 因网络中背景流量波动产生的假阴性样本; ② 因网络中背景流量波动产生的假阳性样本.

在实验过程中, 从序号为 601 的观察窗口开始, 本文机制与 Euclid 均能够有效地识别网络中 DDoS 攻击状态的变化, 具体而言, Euclid 中发现目的地址熵突降, 而本文机制中检测到源目地址熵值差突增. 在图 18 中序号①所示的序号为 620 的观察窗口中, 由于网络中背景流量变化, 源目地址熵值均增大, 此时, Euclid 机制认为目的地址

© 中国科学院软件研究所 http://www.jos.org.cn

熵恢复正常状态, 且源地址熵的突增未超过阈值, 因此判定 DDoS 攻击已经结束, 产生假阴性样本; 在序号②所示的观察窗口中, 由于网络背景流量变化导致源地址熵产生超过阈值的突增, 此时, Euclid 认为 DDoS 攻击发生, 产生假阳性样本. 而本文所提出的机制中所计算的源目地址熵值差则能稳定且准确地反映当前真实的 DDoS 攻击状态.

以上的结果说明本文所设计的 DDoS 攻击检测与防御机制相对于现有的研究工作在检测原理与检测效果上均有明显的优势.

5.5.3 检测参数分析

为了进一步探究相关参数设置对本文所提出的 DDoS 攻击检测机制的影响, 使用第 5.2 节中所述的 TCP_SYN 攻击数据集在不同的熵差固定阈值 T_S 和不同的观察窗口大小下进行测试, 测试结果如图 19 所示. 其中, 图 19(a) 展示了 DDoS 攻击检测结果与熵差固定阈值 T_S 的关系, 当阈值设定过小时, 由于部分正常状态的观察窗口被误判为含有 DDoS 攻击的观察窗口, 导致精确率较低; 当阈值设定过大时, 由于部分含有 DDoS 攻击的观察窗口被误判为正常状态的观察窗口, 导致召回率较低. 当熵差固定阈值范围为 500–1 400 时, 本文所提出的检测机制均能取得较好的检测效果, 可见, 本文机制在较大的阈值范围内均能正常工作, 无需严格的参数配置. 图 19(b) 展示了 DDoS 攻击检测结果与观察窗口大小的关系, 当窗口大小设定小于 5 000 时, 由于检测精确率较低, 导致最终检测结果不佳; 当窗口大小设定大于 60 000 时, 由于召回率较低, 导致最终检测结果下降. 在观察窗口大小范围为 10 000–60 000 时, 本文所提出的检测机制均能取得较好的检测效果.

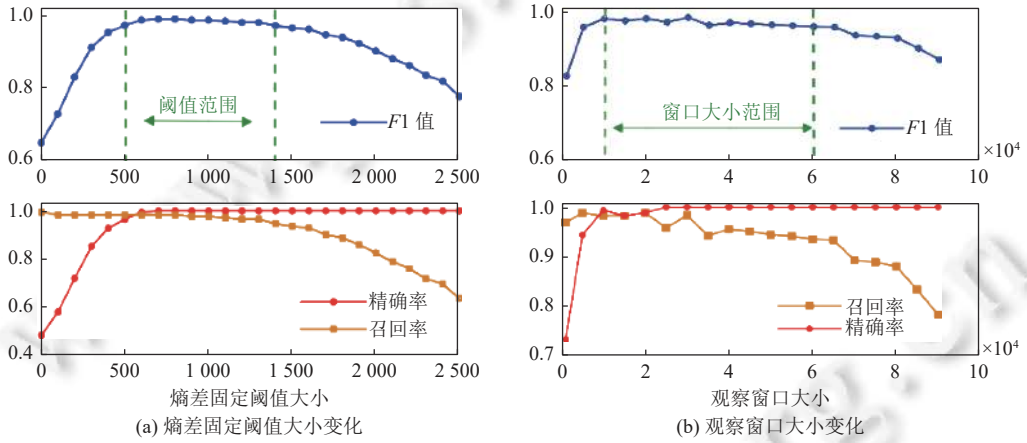


图 19 参数设置变化对检测效果的影响

为了进一步探究观察窗口大小设置与检测效果的关系, 后文图 20 展示了在攻击检测过程中观察窗口大小分别设定为 10 000、1 000、90 000 时, 对应计算得到的源目地址熵值差. 由图 20 中结果可得, 在不同的窗口大小设定下都能够明显地观察到 4 次 DDoS 攻击发生时的源目地址熵值差变化. 当窗口大小为 1 000 时, 检测过程中窗口数量较多且熵值差有更高频率的变化, 导致部分正常状态下的观察窗口被误判为包含 DDoS 的观察窗口, 即过小的窗口设定会导致较低的精确率. 当窗口大小为 90 000 时, 由于每次熵值计算间隔过长, 导致部分窗口中的 DDoS 数据包无法对熵差特征产生显著的改变, 导致部分含有 DDoS 攻击的观察窗口被误判为正常状态, 即过大的窗口设定会导致较低的召回率.

以上的结果说明本文所设计的 DDoS 攻击检测与防御机制相对于现有的研究工作具有较大的优势, 并且能够保证在真实可编程交换机中的可行性, 具有一定的指导意义和实践价值.

5.6 DDoS 攻击检测性能与开销

使用可编程数据平面进行 DDoS 攻击检测的优势在于能够在为网络带来较小开销的情况下实现实时的 DDoS 攻击检测与防御. 如表 5 所示, 实验中在第 5.1 节的参数设定下测试了本文所提出方法所带来的处理时间开

© 中国科学院软件研究所 http://www.jos.org.cn

销及 DDoS 攻击检测与防御机制的响应时间, 同时, 与 Jaqen 机制和基于远程服务器的流量采集方法进行对比. 实验结果中还提供了当可编程数据面仅作转发功能时的数据包处理时间作为参考. 对于单个数据包的处理时延, 本文所提出机制在入口流水线中占用 225 个时钟周期, 在出口流水线中占用了 163 个时钟周期, 相比于 Jaqen 机制, 在数据面中引入了 60 个时钟周期的额外处理时间; 相比于仅作转发的 P4 程序, 引入了 163 个时钟周期的额外处理时间. 在频率为 150 MHz 的可编程交换机中, 单个数据包的处理时间增加了约 0.41–1.09 μs. 由于 Tofino 可编程交换机中采用流水线实现所编写逻辑, 当不发生回环时, 无论所实现功能如何, 均能以线速处理数据包, 本文中所提出机制的部署不会对可编程交换机的吞吐率产生影响.

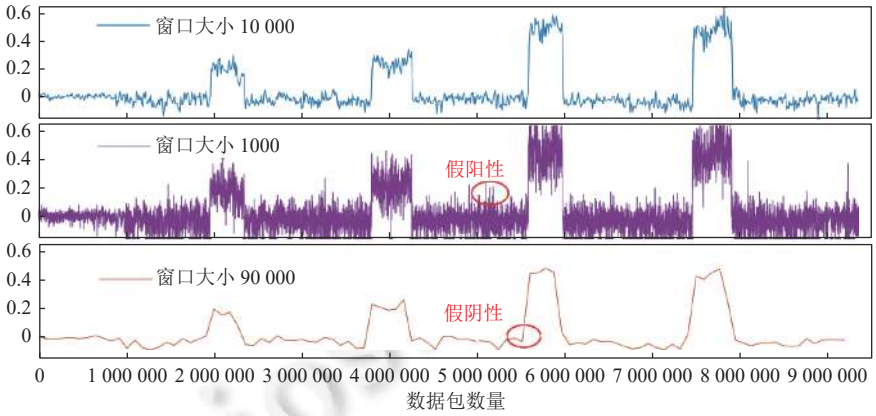


图 20 不同观察窗口大小对应的源目地址熵值差

表 5 DDoS 攻击检测开销

资源类型	入口流水线处理周期 (cycles)	出口流水线 处理周期 (cycles)	单个数据包处理时间 (μs)	控制面处理时间 (ms)	DDoS检测延时 (447 Kpps) (ms)	DDoS检测延时 (150 Mpps) (ms)	每窗口通信量 (KB)
本文机制	225	163	2.59	—	67	0.20	0
Jaqen	160	168	2.18	116	138	116	0.1
流量采集	57	168	1.50	162	183	162	340
仅转发	57	168	1.50	—	—	—	—

在 DDoS 攻击检测速度的评估上, 网络中第 1 个包含 DDoS 攻击数据包的窗口出现至 DDoS 攻击防御机制部署到数据面的时间视为 DDoS 检测延时. 本文中所提出的机制完全基于可编程数据平面实现, 无须控制面参与 DDoS 攻击检测流程, 其 DDoS 检测延时即为 3 个观察窗口通过交换机的时间, 与数据包的通过速率有关, 当数据包通过可编程交换机的速率为 447 Kpps 时, DDoS 检测延时约为 67 ms. 在 Jaqen 机制中, 完成一个观察窗口的数据包检测后需要使用控制面程序查询相应计数器完成攻击检测, 其中控制面处理时间约为 116 ms, 当数据包通过可编程交换机的速率为 447 Kpps 时, DDoS 检测延时约为 138 ms. 在基于流量采集并在远程服务器进行攻击检测的实现方案中, 控制面处理时间约为 162 ms, 当数据包通过可编程交换机的速率为 447 Kpps 时, DDoS 检测延时约为 183 ms. 同时, 由于将数据包的头部信息采集至数据平面, 这种部署方式在检测过程中, 每窗口均引入了较大的数据平面与控制平面之间的通信开销. 随着网络中数据包发送速率的增大, 数据平面中一个观察窗口的通过时间会进一步减小, 而 Jaqen 和流量采集方法由于依赖控制面程序参与检测机制, 其控制面处理时间将会成为检测时间瓶颈. 如表 5 所示, 当数据包发送速率达到 150 Mpps 时, 本文中提出的检测机制的检测延时相较于对比机制更加明显.

与对比机制 Jaqen 相比, 本文所提出的机制具有更小的 DDoS 检测延时和更小的每窗口通信量. 当网络吞吐量达到 150 Mpps 时, 一个观察窗口通过交换机的时间约为 0.07 ms, 由于本文所述机制完全在可编程数据平面上部署, 能够实现与网络吞吐量完全匹配的检测性能, 可以在 3 个时间窗口的时间内 (0.20 ms) 检测 DDoS 是否发生.

© 中国科学院软件研究所 http://www.jos.org.cn

而 Jaqen 机制是一种依赖控制平面的 DDoS 检测系统, 数据平面仅负责网络流量信息的采集与存储, 当一个观察窗口通过后, Jaqen 需要将所记录的信息上传至控制平面并做进一步的处理, 此时的数据-控制平面交互过程约为 116 ms, 这一交互时延远高于控制平面数据包的处理时延, 从而导致了 DDoS 攻击检测的性能受限.

与流量采集方法相比, 本文机制在可编程数据面上的完全部署能够实现更高的检测性能与更小的通信开销. 产生这一现象的原因是在传统的将流量信息转发至控制面或特定服务器进行处理的方式中, 数据-控制平面交互过程产生了较大的延迟与通信开销. 同时, 该方法需要在每观察窗口结束后批量地处理数据包, 由于没有对流量信息进行压缩和预处理, 该方法中的每窗口通信量与控制面计算时间明显大于其余方法, 从而导致 DDoS 攻击检测性能受限.

为了进一步说明检测性能与观察窗口大小的关系, 我们在表 6 中列出了在数据包发送速率达到 150 Mpps 时, 不同观察窗口大小下的检测延时与通信量, 相比于对比方法, 本文的机制基于可编程数据平面的实现能够有效地减小检测延时与通信开销. 与对比机制 Jaqen 相比, 本文机制在任意观察窗口大小中均有着明显的性能优势, 这一现象的产生原因是, 在高吞吐量的网络中, DDoS 攻击检测时延的瓶颈在于控制面-数据面之间的交互时间, 而本文机制能够完全部署在可编程数据面上, 避免了这一交互过程. Jaqen 中依赖控制面执行攻击检测, 虽然在高吞吐的实验环境中数据面处理时间较小, 但控制面-数据面交互流程 (约 116 ms) 无法避免; 在流量采集方法中, 将流量信息采集至控制面集中处理的方法也无法避免控制面-数据面交互流程, 同时, 由于没有对流量信息进行压缩和预处理, 当需要处理的数据包数量增大时, 其攻击检测所需的计算时间随观察窗口大小明显增大.

表 6　观察窗口大小与 DDoS 攻击检测开销

窗口大小	观察窗口大小1k (150 Mpps)		观察窗口大小10k (150 Mpps)		观察窗口大小90k (150 Mpps)	
	DDoS检测延时 (ms)	每窗口通信量 (KB)	DDoS检测延时 (ms)	每窗口通信量 (KB)	DDoS检测延时 (ms)	每窗口通信量 (KB)
本文机制	0.02	0	0.20	0	1.8	0
Jaqen	116	0.1	116	0.1	117	0.1
流量采集	17	34	162	340	1 630	3.06

以上结果说明本文所述的 DDoS 攻击检测与防御机制相对于现有的研究工作具有更好的实时性, 并且更适用于高性能网络.

5.7　资源占用情况及分析

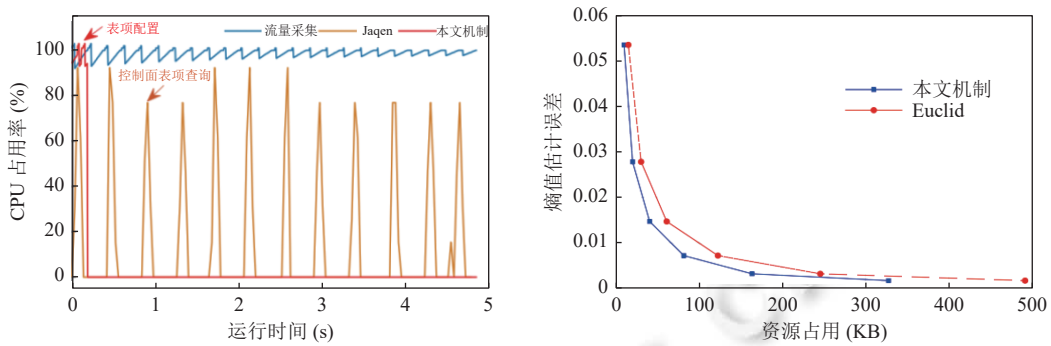
对于不同的 DDoS 攻击检测机制部署方式, 图 21(a) 记录了对应的控制面程序在控制面中的 CPU 资源占用情况. 其中, 基于流量采集的 DDoS 攻击检测完全在控制平面部署; Jaqen 基于可编程数据面记录流量信息, 并使用控制面进行攻击检测; 本文机制完全部署于可编程数据平面. 在进行攻击检测时, 基于流量采集的方法需要持续占用控制面的计算资源对采集到的流量信息进行处理. Jaqen 控制平面定期向数据平面发起表项查询, 对控制面的 CPU 资源占用率周期性波动. 本文所述机制能够完全在可编程硬件上进行部署, 在启动时完成表项部署后无须控制平面参与后续的 DDoS 攻击检测与防御, 能够有效地减少控制平面计算资源的占用.

与对比机制 Euclid 相比, 本文所提出的 DDoS 攻击检测机制能够适应可编程交换机硬件的计算与存储资源限制. Euclid 与本文机制均使用 Sketch 算法用于熵值的计算与保存. 图 21(b) 针对熵值计算模块, 对比了本文机制与 Euclid 中 Sketch 的资源占用情况. 实验结果中展示了在不同存储资源占用下的熵值估计误差. 本文机制相比于 Euclid, 能够使用更少的存储资源, 保证相同的熵值估计误差. 同时, 本文所提出的 Sketch 轮换工作与副本保存能够有效地适应可编程硬件交换机的计算模式, 而 Euclid 中的对应模块无法部署于真实的可编程交换机硬件上.

表 7 中展示了本文机制在可编程交换机中详细的资源消耗情况. 所述 DDoS 攻击检测与防御机制使用 P4 语言实现, 并部署于基于 Intel-Tofino 交换芯片的可编程交换机上, 所设计的机制共占用可编程交换机中 11 级流水线, 产生了 95 个表, 其中, 72 个表部署于入口流水线, 23 个表部署于出口流水线. 由于计算复杂且流程较长, 本文所设计的机制需要占用较多流水线阶段, 但对于每一个流水线阶段, 该机制所占用的可编程交换机资源占比较小.

© 中国科学院软件研究所 http://www.jos.org.cn

其中, Meter ALU 平均资源占用率为 39.6%, Logical Table ID 平均资源占用率为 25.0%, 其余各类资源平均占用率均在 20% 以下, 能够为其他应用程序保留大量的存储与计算资源.



(a) 控制面 CPU 占用情况 (b) Sketch 资源占用与熵值估计误差

图 21 资源占用情况

表 7 可编程交换机资源占用情况

资源类型	占用数量	占用比例 (%)
SRAM	64	6.7
TCAM	0	0
Meter ALU	19	39.6
Logical Table ID	48	25

6 总 结

当前, 依赖远程服务器和控制平面的 DDoS 攻击检测与防御机制在控制面与数据面交互的过程中会产生不可忽视的延迟, 难以满足现今高速网络对检测实验和吞吐量的要求. 针对上述问题, 本文基于可编程交换机提出了一种两段式的 DDoS 攻击与检测机制. 首先将数据包划分为观察窗口, 并设计了基于熵差的观察窗口级 DDoS 攻击检测机制, 之后在攻击发生时采用一种基于源目地址计数值变化的攻击数据包过滤机制实现对 DDoS 攻击的实时防御. 这一机制能够在无须控制面干预的情况下实现自动的 DDoS 攻击与检测, 同时, 通过多组 Count-Min Sketch 轮换、熵值累加更新等方法使这一机制能够满足真实环境中可编程交换机的计算与存储限制. 实验结果表明, 本文所设计的机制在方法和实现上与现有工作相比有着较大的进步, 在多种类型的 DDoS 攻击下有更高的准确性.

References:

[1] Chadd A. DDoS attacks: Past, present and future. Network Security, 2018, 2018(7): 13–15. [doi: 10.1016/S1353-4858(18)30069-2]

[2] Kaur S, Kumar K, Aggarwal N, Singh G. A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions. Computers & Security, 2021, 110: 102423. [doi: 10.1016/j.cose.2021.102423]

[3] Vishwakarma R, Jain AK. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommunication Systems, 2020, 73(1): 3–25. [doi: 10.1007/s11235-019-00599-z]

[4] Arelion. The 2021 DDoS threat landscape report. 2021. https://www.arelion.com/knowledge-hub/white-papers/ddos-threat-landscape-report-2021.html

[5] Gutnikov A, Kupreev O, Shmelev V. DDoS attacks in Q4 2021. 2022. https://securelist.com/ddos-attacks-in-q4-2021/105784/

[6] Chen LG, Zhang YD, Zhao Q, Geng GG, Yan ZW. Detection of DNS DDoS attacks with random forest algorithm on Spark. Procedia Computer Science, 2018, 134: 310–315. [doi: 10.1016/j.procs.2018.07.177]

[7] Hussain F, Abbas SG, Husnain M, Fayyaz UU, Shahzad F, Shah GA. IoT DoS and DDoS attack detection using ResNet. In: Proc. of the 23rd IEEE Int'l Multitopic Conf. (INMIC). Bahawalpur: IEEE, 2020. 1–6. [doi: 10.1109/INMIC50486.2020.9318216]

[8] Hauser F, Häberle M, Merling D, Lindner S, Gurevich C, Zeiger F, Frank R, Menth M. A survey on data plane programming with P4:

© 中国科学院软件研究所 http://www.jos.org.cn

Fundamentals, advances, and applied research. Journal of Network and Computer Applications, 2023, 212: 103561. [doi: 10.1016/j.jnca.2022.103561]

[9] Bosshart P, Daly D, Gibb C, Izzard M, McKeown N, Rexford J, Schlesinger C, Talayco D, Vahdat A, Varghese G, Walker D. P4: Programming protocol-independent packet processors. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87–95. [doi: 10.1145/2656877.2656890]

[10] Liu ZX, Namkung H, Nikolaidis G, Lee J, Kim C, Jin X, Braverman V, Yu ML, Sekar V. Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric DDoS attacks with programmable switches. In: Proc. of the 30th USENIX Security Symp. (USENIX Security 2021). Vancouver: USENIX Association, 2021. 3829–3846.

[11] da Silveira Ilha A, Lapolli ÂC, Marques JA, Gaspary LP. Euclid: A fully in-network, p4-based approach for real-time DDoS attack detection and mitigation. IEEE Trans. on Network and Service Management, 2021, 18(3): 3121–3139. [doi: 10.1109/TNSM.2020.3048265]

[12] GitHub: Behavioral model (BMv2). 2024. https://github.com/p4lang/behavioral-model

[13] Erhan D, Anarım E. Boğaziçi University distributed denial of service dataset. Data in Brief, 2020, 32: 106187. [doi: 10.1016/j.dib.2020.106187]

[14] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69–74. [doi: 10.1145/1355734.1355746]

[15] Open Tofino. 2024. https://github.com/barefootnetworks/open-tofino

[16] Cormode G. Count-Min Sketch. In: Kao MY, ed. Encyclopedia of Algorithms. New York: Springer, 2016. 464–468. [doi: 10.1007/978-1-4939-2864-4_579]

[17] Phaal P, Panchen S, McKee N. InMon corporation's sFlow: A method for monitoring traffic in switched and routed networks. RFC 3186, Internet Engineering Task Force, 2001.

[18] Claise B. Cisco systems NetFlow services export version 9. RFC 3954. Internet Engineering Task Force, 2004.

[19] Hoque N, Kashyap H, Bhattacharyya DK. Real-time DDoS attack detection using FPGA. Computer Communications, 2017, 110: 48–58. [doi: 10.1016/j.comcom.2017.05.015]

[20] Tao Y, Yu S. DDoS attack detection at local area networks using information theoretical metrics. In: Proc. of the 12th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications. Melbourne: IEEE, 2013. 233–240. [doi: 10.1109/TrustCom.2013.32]

[21] Fortunati S, Gini F, Greco MS, Farina A, Graziano A, Giompapa S. An improvement of the state-of-the-art covariance-based methods for statistical anomaly detection algorithms. Signal, Image and Video Processing, 2016, 10(4): 687–694. [doi: 10.1007/s11760-015-0796-y]

[22] Ye J, Cheng XY, Zhu J, Feng LT, Song L. A DDoS attack detection method based on SVM in software defined network. Security and Communication Networks, 2018, 2018: 1–8. [doi: 10.1155/2018/9804061]

[23] Lakshminarasimman S, Ruswin S, Sundarakantham K. Detecting DDoS attacks using decision tree algorithm. In: Proc. of the 4th Int'l Conf. on Signal Processing, Communication and Networking (ICSCN 2017). Chennai: IEEE, 2017. 1–6. [doi: 10.1109/ICSCN.2017.8085703]

[24] Saied A, Overill RE, Radzik T. Detection of known and unknown DDoS attacks using Artificial Neural Networks. Neurocomputing, 2016, 172: 385–393. [doi: 10.1016/j.neucom.2015.04.101]

[25] Hou JP, Fu PP, Cao ZG, Xu AL. Machine learning based DDoS detection through NetFlow analysis. In: Proc. of the 2018 IEEE Military Communications Conf. (MILCOM 2018). Angeles: IEEE, 2018. 1–6. [doi: 10.1109/MILCOM.2018.8599738]

[26] McDermott CD, Majdani F, Petrovski AV. Botnet detection in the internet of things using deep learning approaches. In: Proc. of the 2018 Int'l Joint Conf. on Neural Networks (IJCNN). Rio de Janeiro: IEEE, 2018. 1–8. [doi: 10.1109/IJCNN.2018.8489489]

[27] Grigoryan G, Liu YQ. LAMP: Prompt layer 7 attack mitigation with programmable data planes. In: Proc. of the 17th IEEE Int'l Symp. on Network Computing and Applications (NCA). Cambridge: IEEE, 2018. 1–4. [doi: 10.1109/NCA.2018.8548136]

[28] Kuka M, Vojanec K, Kučera J, Benáček P. Accelerated DDoS attacks mitigation using programmable data plane. In: Proc. of the 2019 ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS). Cambridge: IEEE, 2019. 1–3. [doi: 10.1109/ANCS.2019.8901882]

[29] Mi Y, Wang A. ML-pushback: Machine learning based pushback defense against DDoS. In: Proc. of the 15th Int'l Conf. on Emerging Networking Experiments and Technologies. Orlando: Association for Computing Machinery, 2019. 80–81. [doi: 10.1145/3360468.3368188]

[30] Kokila RT, Thamarai Selvi S, Govindarajan K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In: Proc. of the 6th Int'l Conf. on Advanced Computing (ICoAC). Chennai: IEEE, 2014. 205–210. [doi: 10.1109/ICoAC.2014.7229711]

© 中国科学院软件研究所 http://www.jos.org.cn

[31] Musumeci F, Ionata V, Paolucci F, Cugini F, Tornatore M. Machine-learning-assisted DDoS attack detection with P4 language. In: Proc. of the 2020 IEEE Int'l Conf. on Communications (ICC 2020). Dublin: IEEE, 2020: 1–6. [doi: 10.1109/ICC40277.2020.9149043]

[32] Gupta A, Harrison R, Canini M, Feamster N, Rexford J, Willinger W. Sonata: Query-driven streaming network telemetry. In: Proc. of the 2018 Conf. of the ACM Special Interest Group on Data Communication. Budapest: Association for Computing Machinery, 2018. 357–371. [doi: 10.1145/3230543.3230555]

[33] Ding DM, Savi M, Siracusa D. Tracking normalized network traffic entropy to detect DDoS attacks in P4. IEEE Trans. on Dependable and Secure Computing, 2022, 19(6): 4019–4031. [doi: 10.1109/TDSC.2021.3116345]

[34] Liu XJ, Shang LS, Fang XJ, Lu XB. Distributed denial of service attack detection based on programming protocol-independent packet processors. Application Research of Computers, 2022, 39(7): 2149–2155 (in Chinese with English abstract). [doi: 10.19734/j.issn.1001-3695.2021.12.0661]

[35] Hoque N, Bhattacharyya DK, Kalita JK. Botnet in DDoS attacks: Trends and challenges. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2242–2270. [doi: 10.1109/COMST.2015.2457491]

[36] Nooribakhsh M, Mollamotalebi M. A review on statistical approaches for anomaly detection in DDoS attacks. Information Security Journal: A Global Perspective, 2020, 29(3): 118–133. [doi: 10.1080/19393555.2020.1717019]

[37] Clausius R. Ueber verschiedene für die Anwendung bequeme Formen der Hauptgleichungen der mechanischen Wärmetheorie. Annalen der Physik, 1865, 201(7): 353–400.

[38] Shannon CE. A mathematical theory of communication. The Bell System Technical Journal, 1948, 27(3): 379–423. [doi: 10.1002/j.1538-7305.1948.tb01338.x]

[39] Ding DM, Savi M, Pederzolli F, Campanella M, Siracusa D. In-network volumetric DDoS victim identification using programmable commodity switches. IEEE Trans. on Network and Service Management, 2021, 18(2): 1191–1202. [doi: 10.1109/TNSM.2021.3073597]

附中文参考文献:

[34] 刘向举, 尚林松, 方贤进, 路小宝. 基于可编程协议无关报文处理的分布式拒绝服务攻击检测. 计算机应用研究, 2022, 39(7): 2149–2155. [doi: 10.19734/j.issn.1001-3695.2021.12.0661]

武文浩(2000－), 男, 硕士生, CCF 学生会员, 主要研究领域为网络安全, 可编程网络与系统.

李恩晗(1998－), 女, 硕士, 主要研究领域为网络测量, 全网联合测量.

张磊磊(1997－), 男, 硕士, 主要研究领域为网络测量, 可编程网络.

周建二(1986－), 男, 博士, 副研究员, 主要研究领域为网络测量, 网络性能优化, 云计算虚拟网络.

潘恒(1990－), 男, 博士, 副研究员, 主要研究领域为可编程网络与系统.

李振宇(1980－), 男, 博士, 研究员, 博士生导师, CCF 专业会员, 主要研究领域为网络传输, 网络测量, 网络人工智能.

© 中国科学院软件研究所 http://www.jos.org.cn