FlowTrApp: An SDN Based Architecture for DDoS Attack Detection and Mitigation in Data Centers

FlowTrApp：基于SDN的架构，用于数据中心中的DDoS攻击检测和缓解

Chaitanya Buragohain, Nabajyoti Medhi

Dept. of CSE, NIT Meghalaya

Shillong, India

chaitanyanitm@gmail.com, nabajyoti.medhi@nitm.ac.in

（1）两个流量参数

在本文中，我们为名为FlowTrApp的数据中心提出了一个SDN框架，该框架使用基于每个流的两个流量参数（即，流的流速和流持续时间）的一些界限来执行DDoS检测和缓解。

它尝试使用由基于sflow的流分析引擎sflow-rt和openflow控制器组成的sdn引擎检测从低速率到高速率、长寿命到短寿命的攻击流量。

（2）flowtrapp算法前提:

L7约束：程序管理员认为设置

require:l7约束允许一次从一个IP地址向应用程序发送一个请求，即一个IP地址在任何时间段内都不能向应用程序发送多个请求。

FlowTrapp根据应用程序管理员设置的特定于应用程序的L7约束进行DDOS检测和缓解，

对于来自特定IP地址的HTTP请求，可能只有一个应用程序会话。

这一约束在L7中被应用作为flowtrapp算法的前提条件

（3）获取合法的流速和流持续时间

FlowTrapp算法以合法的绑定信息为输入，并根据这些信息将流量分类为攻击流量或合法流量。

初始手动调查由一组合法用户完成，这些用户使用不同类型的设备对应用程序执行有效操作，以获得合法的持续时间。流量元组是通过观察从合法用户的使用中获得的数据集的流量率和持续时间的最小值和最大值来准备的。

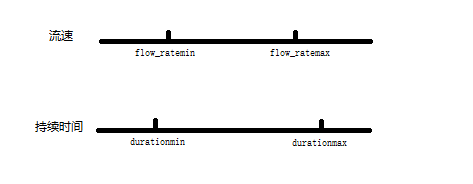
（4）定义ftt

流量和流量持续时间的阈值表示为流量元组（ftt），可以为以下模式中的任何应用程序定义该值：

<flow\_ratemin, flow\_ratemax, durationmin, durationmax>

这里，流速率定义流的数据速率，持续时间定义流的持续时间或生命周期。根据与流量元组值的比较，传入流被分类为合法流或攻击流。流量和流量持续时间在应用程序的ftt范围内的传入流称为该应用程序的合法流。

1. 恶意流分类



基于设备的ftt，恶意流可以分为以下任何重要攻击类别：

▪ 流量速率高于FTT上限但持续时间小于流量持续时间下限的流量称为高速率尖峰（HRS）攻击流量

▪ 流量速率高于FTT上限但流量持续时间小于流量持续时间上限的流量称为短流量高速率（SLHR）攻击流量。

▪ 具有流量速率和流量持续时间超过设备的FTT设置的上限的流被称为长期高速率（LLHR）攻击流。

▪ 流量速率位于FTT的合法范围内但持续时间长于流量持续时间上限的流被称为空闲用户（IS）攻击流。

▪ 流量速率低于FTT下限但持续时间长于流量持续时间上限的流量称为长期低速率（LLLR）攻击流量。

▪ 不能实时检测不属于上述任何类别的攻击流的特征。

（6）FlowTrApp架构中的操作

建议的FlowTrApp架构具有以下主要操作部分：

▪ 使用流量统计信息收集器（如sFlow-RT [24]）从一组合法用户的流量行为中定义合法FTT的阈值，

▪ 使用sFlow-RT识别前面提到的五种攻击类别中的任何一种中的恶意流并以低速率传递流。

▪ 如果发现源经常发送攻击流量，则在启用OpenFlow的交换机中编写OpenFlow规则以在特定时间内反应性地阻止恶意源。

当新的流量到达连接目标应用服务器的聚合交换机时，该流量被发送到SDN引擎的sflow-rt进行监视。

（7）缓解方法：

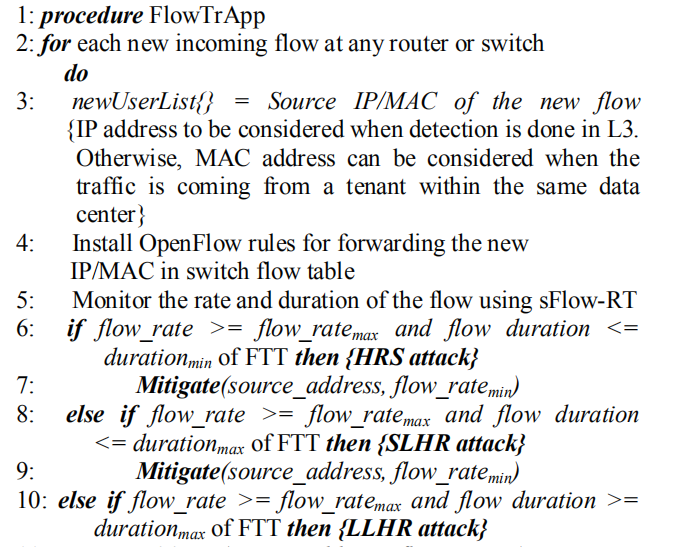
该算法评估源地址尝试攻击的次数。如果源尝试发送的流的数量超过违反FTT的随机合法计数器值，则它将被阻塞持续时间最长时间，以确保安全的无拥塞网络。

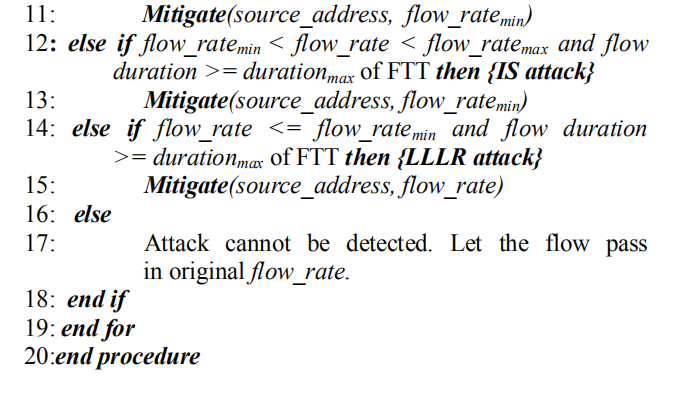
否则，来自该源地址的流可以在不阻塞的情况下传递，但如果流速率>最小流速率，则以最小流速率传递，否则以流速率传递，以避免由于高速流量而导致的拥塞。

（8）算法FlowTrApp描述

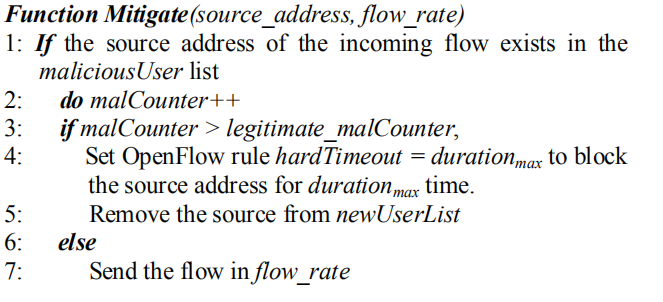
require:l7约束允许一次从一个IP地址向应用程序发送一个请求，即一个IP地址在任何时间段内都不能向应用程序发送多个请求。根据一组合法用户的流量行为，准备一个流量元组（FTT）。

让我们初始化一个空白列表maliciousUser={}，其中包含违反ftt的源地址，而其malcounter值初始化为0。malcounter表示特定源地址表现为恶意的次数。指定一个随机数作为legitimate\_malCounter，这表示特定源地址的malcounter上限。





（9）缓解函数



该算法的可扩展性取决于控制器的性能，本文试图通过使用sflow进行交换机统计监测，并保留一个全局控制器来进行正常的转发决策，从而使控制器的负载最小化。

如上一节所述，上述算法完成了体系结构的三个主要操作部分。通过以下几个步骤，它可以安装在目标应用程序服务器的网络中：

1. 从一组合法用户手动收集合法流量模式。
2. 使用步骤1中获得的流量和持续时间的阈值准备流量元组。
3. 在网络的核心层和访问层安装混合OpenFlow交换机，包括启用OpenFlow的边缘路由器。
4. 运行流统计收集器（如sflow-rt）以及保存在控制器设备中的OpenFlow控制器，并将它们连接到OpenFlow交换机。
5. 在控制器设备中运行flowtrapp算法脚本。