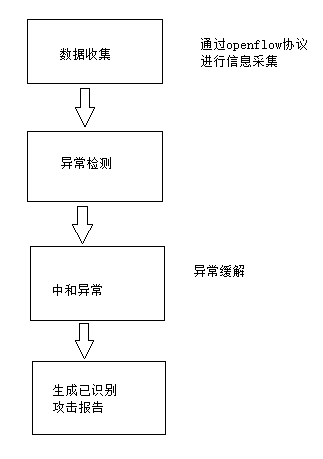
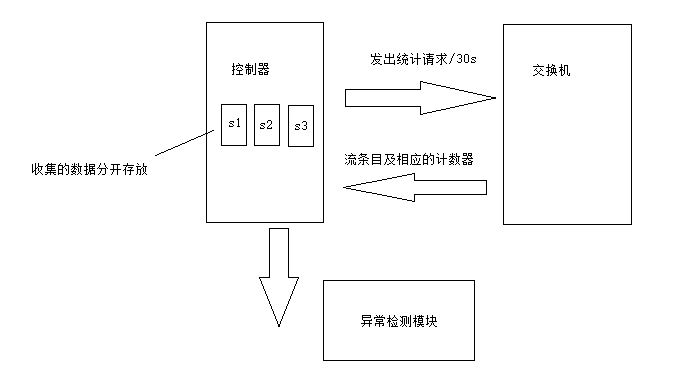
A Novel Anomaly Detection System to Assist Network Management in SDN Environment

IEEE ICC 2017



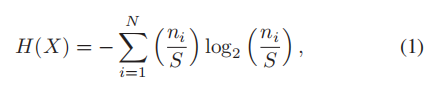
1. 数据收集模块
2. 

控制器定期发送统计请求，以收集交换机中的所有流条目及其相应的计数器。在接收到这个请求后，每个交换机使用OpenFlow消息用流表内容的块响应请求控制器。

每个交换机上收集的数据与从另一个交换机上获得的信息分开处理。因此，识别哪个OpenFlow交换机受到异常影响的主要挑战很简单。

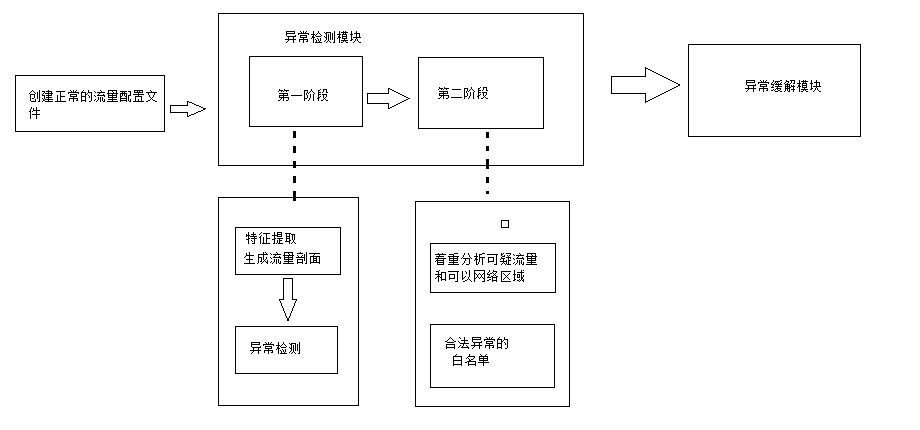
我们采用了30秒的时间窗，从交换机中请求统计数据，并将其发送到异常检测模块，确保在接近实时的情况下触发对策。

对于IP地址和端口特性，我们使用香农熵使它们能够用于流量分析。为了计算它，需要从每个流量测量的样本中创建一个直方图。给定属性 X= {n1，…，ni，…，nN}，其中ni表示样本i的出现次数，定义X的熵H为:



其中是直方图中出现的所有事件的总和。

二、异常检测模块：



1.创建配置文件

通过监测到的流量相对于正常预期的配置文件的偏差来检测可疑事件。

在[13]中，它观察到当前的流量是由具有特定网络使用特性的突发事件组成的循环组成的

2.第一阶段

在第一个模块中，对每个OpenFlow交换机执行流量特性描述，以便提取历史流量中呈现的常见事件。该过程对采集到的6个特征全部完成，其结果为一个正常的流量剖面，用矩阵表示，其中n为一天的总时间窗，a为数据维数，即流量特征个数。因此，表示第i个时间窗的第j个特征的期望值。

Primarily, the analyzed attributes are provided on a six-tuple composed of bits and packets counters followed by entropy calculated for source IP, destination IP, source and destination port, respectively.

当观察到表1中显示的异常流量或意外行为与当前时间窗口中的至少四个流量特征相关时，可能会触发异常情况。如果异常产生的行为与系统识别的特征码之间不存在对应关系，则认为该特征码未知，并将其特征码存储起来，以备以后检测

TABLE I

异常的例子及其对流量特征的影响。

|  |  |
| --- | --- |
| 异常 | 流量特性的影响 |
| 蠕虫 | Source address, destination address and  destination port |
| DDos | Source address, destination address,  destination port and some volume features |
| 端口扫描 | Destination address, destination port and  source address |

3.第二阶段

当识别到异常时，系统会对可疑流量和可疑网络区域进行更积极的分析。它由第二级异常检测模块执行，在多个时间窗口中检查组成异常场景的IP地址和端口。这些主机发送和接收的数据包和位的总数以及它们建立的连接数都受到监视。如果这些值导致偏离预期的交换机行为，则会采取对策使网络恢复正常运行。

4.异常白名单

但是有一些合法的网络异常会以与恶意异常相同的方式影响流量特性。为了避免良性流量事件的中断，我们实现了一个白名单来存储IP地址和端口，这些地址和端口的异常行为可能被视为无异常流量。因此，在应用缓解例程之前，它将检查与异常相关的IP地址和端口是否在白名单中。

1. 缓解模块

网络行为分为四种不同的状态：正常、移动模式、异常和正在进行的缓解

1.应用策略规则

应用策略规则来消除危害网络正常运行的流量，以应对不同的攻击者

一旦检测模块识别出异常，它就会将检测到的事件的相关信息传递给缓解模块，例如攻击者的类型、时间戳、IP地址以及恶意通信所涉及的端口。

将包含在异常检测模块中学习到的IP地址和端口的流条目连同必须对匹配这些流的新包执行的操作插入到交换机的转发表中。主要操作包括转发、删除或修改数据包字段[15]。我们将转发操作附加到每个良性流条目上，而drop操作用于阻止恶意流量。设计了相应的对策后，通过网络控制器将策略推送到交换机。这可以使用OpenFlow协议来实现，该协议直接作用于网络设备的转发逻辑。

1.1缓解规则条目：

与转发表中的其他活动条目相比，与缓解规则对应的流条目具有更高的执行优先级。此外，它还被分配了一定的时间(空闲超时)，以便当不再发生匹配时这些流仍然处于活动状态，从而允许在异常发生时进行缓解。

实验背景材料：

Mininet[16]是一个模拟器，它创建了一个由虚拟主机、交换机、控制器和链接组成的网络。在Mininet上开发的代码可以转移到真实的环境中，只需对真实世界的部署进行最小的更改。

本文所假设的方法基于前几周的历史，为每一个分析日创建一个正常的网络概要文件。因此，我们使用Scapy[17]，这是一个可编程的工具，它发送带有任意协议字段值(例如IP地址、端口和内容)的数据包序列来生成四天的历史流量。我们的测试场景是为了模拟真实的环境而构建的，通过考虑[4]提出的建议，尽可能地真实可靠。因此，我们的分析包括混合不同类型的合法流量和不同的攻击流量参数。测试期间的良性流量结合了大约85%的TCP流量、10%的UDP和5%的ICMP。

在确保所有收集到的流量特征都用量化值表示后，开始提取正常行为。在本文中，我们建立在我们之前的[18]流量分析技术的基础上，使用它来为异常检测提供基础。首先，将分析的属性提供在一个由比特和包计数器组成的六元组上，然后分别计算源IP、目标IP、源和目标端口的熵。然后，我们使用数据聚类方法来发现每个历史流量时间窗口的合理组织，以便识别和量化它们之间的相似性。在传统蚁群算法的基础上，采用蚁群算法对数字签名进行聚类。使用ACODS的优点是集群过程是一种无监督的学习技术，允许系统以自主的方式工作。此外，它可以构造非局部最优解，这是一些聚类算法中存在的问题