一种基于端口的网络流量特征熵的异常检测方法

王力 [1)] 丁伟 [2)] 吴琪 [3)] 夏震 [4)]

[1),2),3),4)](东南大学 计算机科学与工程学院, 南京市 211189)

摘 要 论文提出了一种基于端口的网络流量特征熵的异常检测方法。该方法首先对给定网络边界实测流量的端口流量分布信息熵进行了正态分布的检验。在此基础上，以正态分布 $N(\mu,\sigma^2)$的随机变量在区间$(\mu-n\sigma, \mu+n\sigma)$内的概率对任意给定 μ, σ 和 n 均相等的数学原理为思路，设计了一个异常检测算法。随后将算法作用到同一网络边界，并对在 14 天的运行过程中检测到的异常流量进行了分析，所有的异常均可以准确定位到特定的事件。这个结果表明该异常检测方法的有效性。

关键词 异常检测，流量特征分布，信息熵，正态分布，流记录

中图法分类号 TP393

An Anomaly Detection Method Based on Port Network Traffic Feature Entropy

WANG Li[1)] DING Wei[2)] WU Qi[3)] XIA Zhen[4)]

[1),2),3),4)](School of Computer Science and Engineering, Southeast University, Nanjing 211189)

Abstract The paper presents an anomaly detection method based on port network traffic feature entropy. This method first tests the normal distribution of the port distribution information entropy for a given measured flow in network boundary. On this basis, an anomaly detection algorithm is designed for the mathematical principle that the probability of random variables of normaly distribution $N(\mu,\sigma^2)$ in interval $(\mu-n\sigma, \mu+n\sigma)$ is the same for any given μ, σ and n. The algorithm is then applied to the same network boundary, and the abnormal traffic detected during the 14-day operation is analyzed, and all exceptions can be accurately positioned for a specific event. This result indicates the effectiveness of the anomaly detection method.

Key Word anomaly detection，traffic characteristics distribution，information entropy，normal distribution，flow records

1 引言

网络流量的变化规律能够在很大程度上反映网络运行的状况。大部分的网络异常都会在流量形态上有所体现。最常见的异常是如图 1 所示的尖峰流量。在类似这样的异常流量形态出现的时候，网络中到底发生了什么呢？这是一个迄今为止，并没有得到很好解决的问题。因此异常流量检测一直以来都是网络管理领域的一个研究热点。具有实用价值的研究工作需要从两个层面展开，首先是在海量的网络流量中检测出发生异常的位置，在此基础上需要完成对导致异常的流量的分析，从中找出导致异常发生的原因并形成安全事件进行响应处理。

异常流量行为的检测分析是面向测度的，最简单的测度是 bps(bits per second)和 pps(packets per second)等带宽相关测度。很多文献[1-5]尝试通过对源宿 IP 地址、源宿端口、流的大小和流的持续时间等流量特征的取值在一定时间间隔的网络流量中的分布特性进行统计分析，并以此为基础形成测度检测异常。在这种基于分布特征的测度形成方法上，文献[6]的研究认为绝大多数的异常都会引起报文头部流量特征的分布变化，并选择用熵来描述流量特

作者信息
姓名：王力
E-mail：332613292@qq.com
联系地址：南京市江宁区东南大学路 2 号计算机楼
电话：15850660770

征的分布情况。熵可以将随机变量的一组样本分布变成一个实数，因此是这类分析测度构成的主要思路。

端口是 TCP(transmission control protocol)和 UDP(user datagram protocol)协议中的重要字段。很多研究[1-6]表明，网络流量中的端口在分布上存在明显的重尾性。异常流量行为的发生会引起流量在端口分布上的变化，例如 DDoS(Distributed Denial of Service)攻击、扫描和巨流等。当这些异常发生时，原有的规律会被打破。从这样的角度分析，通过追踪端口流量分布的变化应该能够发现异常。

常见的异常流量行为的检测方法有小波分析和时间序列分析两类。前者有较好的检测效果，但计算过程复杂，有相当于滑动窗口大小的延迟，因此不适合在线检测；相对而言后者在该领域有更广泛的应用，常用的模型有 ARMA(Auto-Regressive and Moving Average)模型、ARIMA(Autoregressive Integrated Moving Averag)模型和指数平滑模型等，使用时间序列模型需要有一个学习的过程。在基于信息熵进行的相关研究中，文献[7]根据协议和宿端口号对报文进行分类并基于最大熵原理生成一个基准分布，通过计算网络当前状态与基准分布的相对熵(KL 距离)发现异常。文献[8,9]利用典型攻击案例，研究了选择不同特征和基于不同形式的熵对检测效果的影响，流量特征包括源宿 IP 地址、源宿端口等，熵的形式包括常见的香浓熵以及 Tsalliis 熵、Renyi 熵。文献[10]中提出了一种新颖的入侵检测系统，通过研究与网络流量相关联的熵的变化来执行异常检测。这些研究为新的检测方法提供了思路。

相对于异常检测，异常数据分析领域并没有非常有效的方法可以支持对检测出的异常流量进行分析，并定位其中可能的安全事件。所有的异常检测方法，均采用实测流量进行有效性验证。

2013 年 CERNET(China Education and Research Network)主干网升级后，所有 38 个主节点的网络边界均可以获得由华为 NE40E 路由器提供的流记录数据。我们基于经典的季节模型，对常规的 bps 和 pps 测度进行了异常检测。在这个实践过程中，我们发现无论怎样调整参数，都无法避免大量的误报的出现。为此，我们将思路转向基于流记录可以获取的单位时间内网络流量中的端口分布熵测度，尝试寻找面向这个测度的异常检测方法。

图 1 被管网一天出方向 bps

2 基于端口的流量特征熵

信息熵一直作为一种有效的检测度量被引入到网络流量领域支持异常检测。在信息熵的基础上，把流量数据当作离散信息源，将网络流量数据中的某个特征看作是一组随机事件，就可以得到特征熵。

2.1 特征熵的定义

定义 1. 特征熵。随机地对网络流量数据特征 X 进行观察，观察的样本总数为 S，样本特征值的取值个数为 N，其中特征 i 出现次数为 n_i 次，那么特征熵的定义如下：

$$H(X) = -\sum_{i=1}^{N}\left(\frac{n_i}{S}\right)\log_2\left(\frac{n_i}{S}\right) \qquad (1)$$

其中，$S = \sum_{i=1}^{N} n_i$。流量特征 X 可以是观测期间内的源宿 IP 地址和源宿端口等。

在具体计算时，又可以按流量的方向、源/宿端口以及字节数和报文数分别计算。例如，在计算一个观测周期入方向宿端口字节分布熵时，公式(1)中的 S 为该周期内入方向字节总数，n_i 为其中宿端口为 i 的字节数量，i=1,2...65535。

2.2 对端口网络流量特征熵的观测

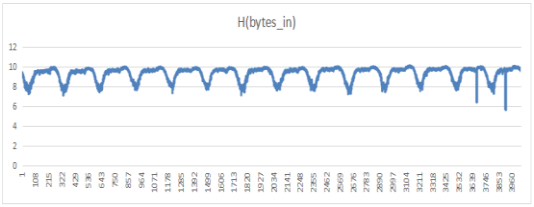
作为研究工作的起点，我们在 CERNET 南京主节点利用边界路由器提供

的流记录，对基于上述公式计算出的端口流量特征熵进行了 14 天的观测。由于是在网络的边界，因此只选择了入方向宿端口和出方向源端口作为观测对象，用于特征熵计算的观测周期是 5 分钟（以下简称为时间粒度）。每个观测对象的特征熵，又可以按字节数和报文数分别计算。这样每个时间粒度可以按公式(1)计算出 4 个特征熵：$H(X_{bytes\_in})$，$H(X_{bytes\_out})$，$H(X_{pkts_in})$和$H(X_{pkts_out})$。还是以入方向宿端口(*bytes_in*)为例，$H(X_{bytes_in})$的计算公式如下：

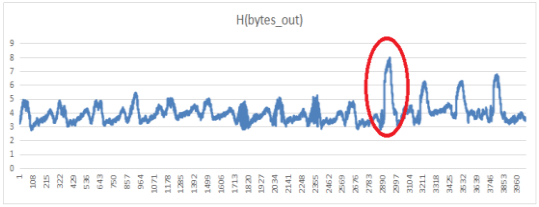
$$H\left(X_{bytes\_in}\right)=-\sum_{i=1}^{N}\left(\frac{m_i}{M}\right)\log_2\left(\frac{m_i}{M}\right) \qquad (2)$$

其中，入方向流量宿端口字节数统计集 $X_{bytes\_in}=\{m_i, i=1,2,...,65535\}$，$N=65535$，$M$ 为该时间粒度入方向流量字节总数。类似地，可以得到 $H(X_{bytes\_out})$、$H(X_{pkts\_in})$、$H(X_{pkts_out})$的计算公式。

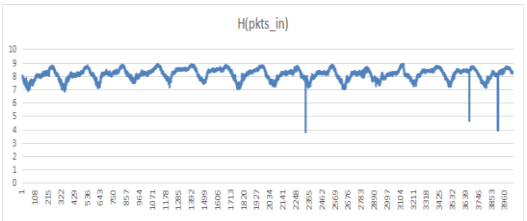
图 2 给出了上述 4 种特征熵在连续 14 天的观测结果。由于观测周期（时间粒度）是 5 分钟，这样 14 天的观测一共有 288*14=4034 个样本值，这也是每张图中横轴上点的数量。直接观察图 2，可以发现以下 2 个明显的现象：1) 4 种特征熵的时间序列以“天”为单位呈现明显的周期性变化规律。以入方向端口流量熵（$H(X_{bytes_in})$和 $H(X_{pkts\_in})$）更为明显。因为网络流量中宿端口的伪造没有意义，因此使用宿端口数据计算的 $H(X_{bytes\_in})$ 和 $H(X_{pkts_in})$相对更为真实，这个规律更应该被认可；2) 图 2(b)中标注出的 $H(X_{bytes\_out})$ 明显偏大，并且之后几天都出现了类似的现象。这个时间粒度是 2016 年 11 月 11 日 02:20。在这之前，2:05、2:10、2:15 所在时间粒度 $H(X_{bytes_out})$分别为 3.74、3.86、4.18，而在 2:20 的 $H(X_{bytes_out})$为 6.23。这四个时间粒度出方向总字节数(*bytes_out*)分别为 203G、173G、182G、203G，并无明显变化。但 2:20 的端口分布情况明显有异于另外的三个时间粒度。这说明端口流量熵能够发现从总流量无法反映异常的现象。是什么样的原因导致了这个现象的发生？由于在这 14 天的观测期内没有保存保存原始流记录，所以我们无法进行进一步的分析。在图 2 中，还有多处明显的异常点，随后的研究将以异常的定位和分析作为目标。首先是设计一个面向端口流量特征熵的异常检测算法，在定位异常的同时保存相关的原始流记录；第二步是通过对原始流记录进行分析，分析成因。



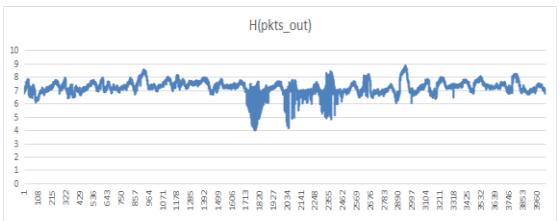
(a)$H(X_{bytes_in})$的时间序列



(b)$H(X_{bytes_out})$的时间序列



(c)$H(X_{pkts_in})$的时间序列



(d)$H(X_{pkts_out})$的时间序列

图 2　四种端口流量熵的时间序列图（x-label:时间粒度序号, y-label:端口流量熵）

3　端口流量分布异常检测算法

由于 14 天的流量观测显示了端口流量特征熵以天为周期呈现出明显的规律，这意味着在每天相同的时间粒度，特定端口熵的取值会比较接近。这是比较典型的正态分布特征，因此我们尝试用正态分布建立模型。

3.1 端口流量特征熵的随机过程模型

本文所关注的特征熵有 4 个 $H(X_{bytes\_in})$，$H(X_{bytes\_out})$，$H(X_{pkts\_in})$ 和 $H(X_{pkts\_out})$，它们在每个时间粒度（5 分钟）可以分别获得一个取值，每天有 288 个时间粒度，可以将其看成 288*4=1152 个随机变量。这样上述 4 个特征熵可以用随机过程 $H(i,t)$ 表示，i={0,1,2,3}，分别对应 $H(X_{pkts\_in})$，$H(X_{pkts\_out})$，$H(X_{bytes\_in})$ 和 $H(X_{bytes_out})$四种不同的端口流量特征熵，t={0,1,...286,287}，分别对应每天从 0:00,0:05,...23:50,23:55 为开始时间的粒度。给定 i_0 和 t_0，$PH(i_0,t_0)$为 t_0 时间粒度的 i_0 特征熵随机变量。例如 H(0,0)就表示每天 0:00-0:05 这个时刻特征熵 $H(X_{pkts_in})$的取值所构成的随机变量。

3.2 端口流量特征熵的正态分布检验

本小节的工作是希望完成对时间离散、状态连续的特征熵随机过程 $H(i,t)$的正态分布检验。采用的方法是通过在 CERNET 南京主节点网络边界获取足够的样本数据，对 $PH(i,t)$所有的 $H(i_0,t_0)$，共 1152 个随机变量进行正态分布检验。在实践过程中，观测的时间是连续的 76 天，这样每个待检验的随机变量均获得了 76 个样本。考虑其中可能存在异常值，对检验结果有一定的影响，因此去除每组样本中最大的两个数据和最小的两个数据，实际使用 72 个样本，使用 SPSS 软件支持的 K-S 检验方法，通过观测 K-S 统计量显著性水平值 P 是否大于临界值 0.05，完成对样本数据的正态分布检验。在 1152 个随机变量中，除了 *pkts_out* 组有 1 个未能通过，其余的 1151 个随机变量全部通过了检验。

3.3 基于正态分布的基本检测原理

正态分布是一种连续型随机变量的概率分布，在医学、社会学和心理学等大量领域被广泛用于异常检测。正态分布有两个参数：均值 μ 和均方差 σ，对应的随机变量 X 通常记作 $N(\mu,\sigma^2)$。遵从正态分布的随机变量 X 的概率规律是取值越靠近 μ 概率越大；σ 越小，分布越集中在 μ 附近。任何一个正态分布，无论其参数的取值如何，均服从下面的分布规律：

$$\begin{aligned} &P\{\mu-3*\sigma<X<\mu+3*\sigma\}=99.74\% \\ &P\{\mu-4*\sigma<X<\mu+4*\sigma\}=99.9936\% \\ &P\{\mu-4.8*\sigma<X<\mu+4.8*\sigma\}=99.9999\% \end{aligned} \quad (3)$$

这意味着对于一个遵从正态分布的随机变量，只要其参数 μ 和 σ 已知，就可以很方便地确定它的一个取值的概率范围。利用这个范围建立置信区间，就可以对端口流量特征熵 $H(i,t)$进行异常检测。参数 μ 和 σ 的获取，可以通过最大似然估计，基于获取的样本用以下公式计算：

$$\hat{\mu}=\bar{X} \quad (4)$$

$$\hat{\sigma}^2=\frac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X})^2=\frac{1}{n}\sum_{i=1}^{n}X_i^2-(\frac{1}{n}\sum_{i=1}^{n}X_i)^2 \quad (5)$$

3.4 检测算法

因为样本的端口流量特征熵通过了正态分布检验，因此正态分布的相关理论可以帮助确定端口流量熵的置信区间。基于这个基本原理，本小节提出一种基于正态分布的端口流量分布特征熵异常检测算法。为了能够对端口熵异常的原因作深入分析，在检测到异常时，实时保存该时间粒度的原始流记录。

算法 1. 基于正态分布的端口流量特征熵异常检测算法

输入：当前时间粒度的端口流量数据 $X(t)=\{m_i,\ i=1,2,...,65535\}$，*t=bytes_in, bytes_out, pkts_in* 或 *pkts_out*

输出：异常测度值

说明：1）算法临时保存当前时间粒度所有的原始流记录，如果该时间粒度被检测出熵异常，则对应的流记录会永久保存用于随后的成因分析；2）算法使用前 x 天相同时间粒度的特征熵值用于按公式(4)和(5)计算参数；3）参数 k 表示在计算参数时剔除的最大和最小异常值数量；4）a 置信区间因子，即公式(3)中 σ 前的系数。

算法描述

Step1：基于当前时间粒度的端口流量数据 $X(t)$，计算当前 4 种特征熵值；

Step2：利用保存的历史数据根据公式(4)和(5)计算 μ_0 和 σ_0；

Step3：根据 μ_0、σ_0、参数 a 和公式(3)，判断当前特征熵值是否异常，如果是，则记录该熵值，并保存原始流记录；

Step4：用当前特征熵值对历史数据进行维护，转 step1。

4 实验与结果分析

上述算法实现后，在 CERNET 南京主节点网络边界从 2017 年 1 月 5 日到 1 月 16 日，试运行了 11 天，下面对运行结果进行介绍和分析。在研究初期，为了异常检测的准确性，运行时算法设置参数 k=3 和 a=5，而设置参数 x=66 实际上意味着算法在开始异常检测之前已经进行了连续 66 天的运行以获得参数计算所需要数据。

4.1 实验结果

在 11 天的异常检测试运行期间，共检测到 29 次端口流量熵异常，均为偏小，出现在 17 个时间粒度，具体情况如表 1 所示。表中时间指各时间粒度的开始时间，下文中提到的时间也均为此义。

图 3 是观测期间四种端口流量熵的时间序列图，图中方框标注了检测到的异常的位置，序号对应表 1 中的异常序号。

表 1 端口流量熵异常检测结果

异常序号	时间	异常的熵测度
1	2017/1/6 9:30	$H(X_{pkts\_in})$
2	2017/1/6 15:45	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
3	2017/1/6 22:00	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
4	2017/1/7 9:35	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
5	2017/1/7 9:45	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
6	2017/1/7 9:50	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
7	2017/1/7 11:55	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
8	2017/1/7 12:00	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
9	2017/1/7 13:35	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
10	2017/1/8 4:40	$H(X_{bytes\_in})$
11	2017/1/8 5:10	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
12	2017/1/8 5:40	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
13	2017/1/8 6:00	$H(X_{bytes\_in})$
14	2017/1/10 11:25	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
15	2017/1/10 11:45	$H(X_{pkts\_in})$、$H(X_{bytes\_in})$
16	2017/1/14 16:45	$H(X_{bytes\_in})$
17	2017/1/14 4:20	$H(X_{pkts\_out})$

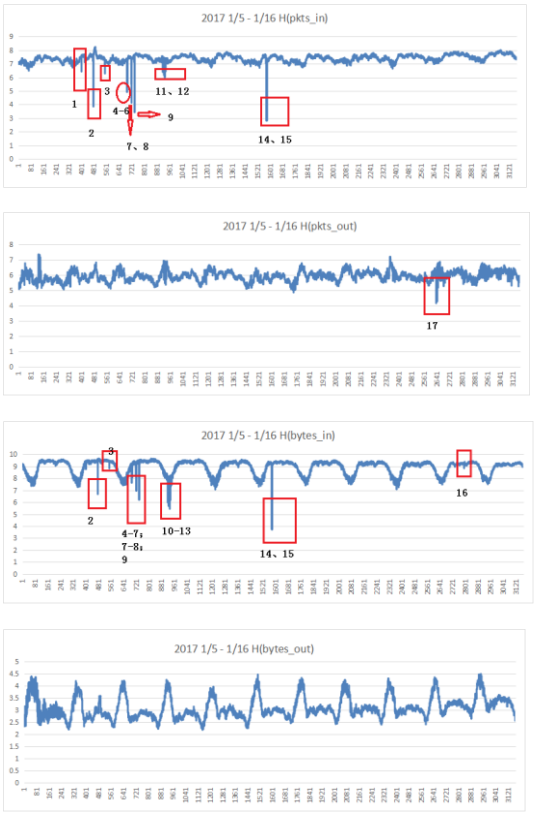


图 3 2017/1/5-2017/1/16 四种熵时间序列图
（x-label:时间粒度序号, y-label:端口流量熵）

4.2 检测结果分析

因为算法保存了所有存在异常的原始流记录，本小节将通过对这些原始流记录进行分析，尝试寻找导致异常的原因。

因为检测出的异常端口流量熵值均偏小，这说明其端口流量分布较正常情况更集中，即是由某些端口流量剧增造成的，为此采取以下方法进行分析：

1)根据异常的测度（bytes/pkts），列出该时间粒度端口流量 top 10，确定可疑端口。

2)找出所有可疑端口的流记录，并定位引起异常的流，通过观察这些流两个 IP 地址之间的交互情况，分析导致异常的原因。

基于这个步骤对全部 29 次熵异常所在的 17 个时间粒度的原始流记录进行分析的结果表明导致这 17 个时间粒度端口特征熵异常的原因可以分为 5 个事件，如表 2 所示，所有的异常都找到了原因，没有任何误报。其中巨流的定义如下：

定义 2. 巨流。巨流是超出正常水平的长流，它会占据整个网络绝大部分的流量，对网络的性能造成很大的影响，发现这些巨流能够帮助网络管理者更好了解网络的运行状态。

表 2 端口流量熵异常分析结果

序号	异常序号	异常原因	异常分析
1	1-9	DRDos	异常现象：导致异常的端口 80；所有对端端口 1900；异常报文使用的协议：UDP 原因分析：网内主机遭受 UDP 反射攻击
2	10-13	巨流	异常现象：这四个时间粒度的所有熵异常由同一条流导致，该流占据了对应时段全网报文数的 20%和字节数 30%，平均报文长度达到 1513，使用协议为 TCP，服务器端使用 80 端口；存在反向流，报文数双向较为平衡 原因分析：网内的一台主机在 4:40-6:00 期间从其他运营商的一台主机下载了大量的数据。
3	14-15	DRDoS	异常现象：排名靠前的若干端口的宿地址为同一网内地址，所有对端端口均为 123，使用协议为 UDP 原因分析：网内主机遭受 UDP 反射攻击
4	16	巨流	异常现象：该时间粒度的异常由一条流导致，网外地址是美国的一个医学数据库，网外主机端口 33001 是 Aspera Server(高速 FTP 服务)端口 原因分析：网内高校用户在下载医学数据。
5	17	UDP flood	异常现象：两条流的报文数均超过该时间粒度全网流量的 24%，合计约 49%，平均报文长度很短(60 字节)，不存在反向流，使用协议 UDP 原因分析：UDP Flood 攻击。

4.3 异常案例分析

本小节选取表 2 中事件 5，给出利用流记录进行深入分析的详细过程。

这个事件的特征熵异常发生在 1 月 14 日 4:20，异常的测度是 H(X_{pkts_out})偏小。对 4:20 出方向流量的源端口按报文数统计 top10，如图 4 所示。结果显示端口号 60520 和 48874 为源端口的出方向流量非常大，尽管分别只有 5 条流和 4 条流，但报文数占比分别达到了 24.8%和 24.3%，合计超过 49%；

```
[nbos@nbos136 data]$ /usr/local/nfdump/bin/nfdump -r /home/qwu/data/nfcapd.201701140420  -n 10 -s srcport/packets ' (out if 7 or o
ut if 12 or out if 13 or out if  38 or out if 45  ) '
Top 10 Src Port ordered by packets:
Date first seen          Duration Proto      Src Port     Flows(%)     Packets(%)       Bytes(%)     pps      bps   bpp
2017-01-14 03:52:23.000  1813.000 any         60520       5( 0.0)    442794(24.8)    26.6 M( 5.1)   244   117233    60
2017-01-14 03:51:55.000  1912.000 any         48874       4( 0.0)    433016(24.3)    26.0 M( 5.0)   226   108707    60
2017-01-14 03:50:05.000  2093.000 any            80   30416(12.6)    112404( 6.3)   147.4 M(28.5)    53   563431  1311
2017-01-14 04:09:29.000   899.000 any           123     221( 0.1)     75957( 4.3)    35.5 M( 6.9)    84   315832   467
2017-01-14 03:49:59.000  2089.000 any           600     748( 0.3)     68859( 3.9)    39.6 M( 7.7)    32   151739   575
2017-01-14 03:51:21.000  2013.000 any            22    1508( 0.6)     30718( 1.7)     5.6 M( 1.1)    15    22406   183
2017-01-14 04:18:30.000   358.000 any          8000     469( 0.2)     21543( 1.2)     4.7 M( 0.9)    60   105201   218
2017-01-14 04:19:19.000   137.000 any         28635   16387( 6.8)     16387( 0.9)     1.0 M( 0.2)   119    59323    61
2017-01-14 04:02:39.000  1309.000 any             0   11068( 4.6)     14196( 0.8)     2.5 M( 0.5)    10    15071   173
2017-01-14 03:51:17.000  1991.000 any            53    9601( 4.0)     12932( 0.7)     9.8 M( 1.9)     6    39529   760
```

图 4　2017/1/14 4:20 *pkts_out* 中 top10 端口

图 5 进一步展示了源端口是 60520 和 48874 的 9 条出方向流。发现异常来自其中两条，它们的报文数分别达到 433012 和 442789，而其它流只有 1-2 个报文。

```
[nbos@nbos136 data]$ /usr/local/nfdump/bin/nfdump -r /home/qwu/data/nfcapd.201701140420 '((src port=60520 )or (src port=48874)) an
d (out if 7 or out if 12 or out if 13 or out if  38 or out if 45  ) '
Date flow start          Duration Proto      Src IP Addr:Port           Dst IP Addr:Port   Packets    Bytes Flows    Input Output
2017-01-14 04:19:54.000     0.000 UDP     222.[redacted].186.18:60520 ->     61.[redacted].141.12:53         1       89     1 [     8    45 ]
2017-01-14 04:21:02.000     0.000 UDP      202.[redacted].45.21:60520 ->    122.[redacted].42.57:32813       1       82     1 [     6    45 ]
2017-01-14 03:51:55.000  1802.000 UDP     121.[redacted].60.133:48874 ->     91.[redacted].137.3:80     433012   26.0 M     1 [     6    38 ]
2017-01-14 04:21:26.000     0.000 TCP     222.[redacted].184.53:60520 ->    114.[redacted].82.23:8682        1       385     1 [     6    13 ]
2017-01-14 04:21:28.000     0.000 UDP     222.[redacted].186.17:48874 ->    140.[redacted].228.61:53         1       117     1 [     8    45 ]
2017-01-14 03:52:23.000  1802.000 UDP     121.[redacted].60.133:60520 ->     91.[redacted].137.3:21     442789   26.6 M     1 [     6    45 ]
2017-01-14 04:22:36.000     0.000 TCP      202.[redacted].48.85:60520 ->     58.[redacted].220.41:80          2       156     1 [     6    13 ]
2017-01-14 04:23:32.000     0.000 TCP     121.[redacted].25.139:48874 ->   211.[redacted].112.148:80          1        74     1 [     6    38 ]
2017-01-14 04:23:37.000    10.000 TCP     202.[redacted].46.189:48874 ->    202.[redacted].48.96:20513        2       132     1 [     6    12 ]
Summary: total flows: 9, total bytes: 52.5 M, total packets: 875810, avg bps: 219870, avg pps: 458, avg bpp: 60
Time window: 2017-01-14 03:51:55 - 2017-01-14 04:23:47
```

图 5 源端口为 60520 和 48874 的出方向流

因此认为这两个流引起了异常：

UDP　121.*.60.133:60520 -> 91.*.137.3:21

UDP　121.*.60.133:48874 -> 91.*.137.3:80

流中平均报文长度仅为 60 字节，且在同一时间粒度反方向对称流不存在，因此判定这是一次网内主机 121.*.60.133 向网

外主机 91.*.137.3 发起的 UDP Flood 攻击。

被攻击主机 91.*.137.3 归属于保加利亚，其 80 端口上提供 http 服务，存在正常的可访问的 web 页面。对网内主机 121.*.60.133 进行的常规服务器检测发现其无 DNS 响应；无 SMTP 响应；80 端口开放，可响应 TCP 连接请求，间断响应 http 请求。在 25 端口上有 TCP 流量。

对该粒度流记录的进一步分析发现：

(1)在该时间粒度内，两个 IP 间一共 1537 条流。除去上面两条，其余 1535 条都为以下的形式：

UDP 121.*.60.133:50435->91.*.137.3:x 1 1514

其中，x 表示随机高端端口，即 121.*.60.133 通过 50435 端口向 91.*.137.3 的 1535 个不同端口各发送了 1 个长度为 1514 的 UDP 报文。考虑到抽样比的存在，实际上发送该报文的端口数量会更多。

(2) 检索该时间粒度内网内地址 121.*.60.133 与网外主机的全部交互情况，结果如图 6 所示，宿地址为 91.*.137.3 的流报文数占到 99.9%。剩余的 0.1%主要发往同一网段的 91.*.137.10。所有这些流量均不是正常的流量状态。基于上述分析可以判定 1 月 14 日 4:20 引发 $H(X_{pkts_out})$偏小的原因：网内地址 121.*.60.133 用源端口 60520 和 48874 向网外地址 91.*.137.3 发起一个 UDP Flood 攻击。鉴于这样的攻击行为基本由僵尸进程导致，且对其流量行为的分析还表明该主机还存在其他的非正常行为，因此可进一步认定主机 121.*.60.133 在很大程度上可能存在着僵尸进程。

```
[nbos@nbos136 data]$ /usr/local/nfdump/bin/nfdump -r /home/qwu/data/nfcapd.201701140420 -o extended -s dstip/packets '(src ip 121.
[redacted].60.133) and (out if 7 or out if 12 or out if 13 or out if  38 or out if 45  ) '
Top 10 Dst IP Addr ordered by packets:
Date first seen          Duration Proto      Dst IP Addr   Flows(%)      Packets(%)        Bytes(%)          pps      bps   bpp
2017-01-14 03:51:55.000  1952.000 any        91.[redacted].137.3  1537(58.5)   877336(99.9)   54.9 M(97.7)   449   224885    62
2017-01-14 04:21:29.000   177.000 any        91.[redacted].217.10  844(32.1)    844( 0.1)    1.3 M( 2.3)     4    57754  1514
2017-01-14 04:23:10.000     3.000 any        44.[redacted].68.14     1( 0.0)      2( 0.0)     148( 0.0)     0      394    74
2017-01-14 04:19:37.000   243.000 any        104.[redacted].176.113  2( 0.1)      2( 0.0)     132( 0.0)     0        4    66
2017-01-14 04:20:11.000     0.000 any        198.[redacted].255.36   1( 0.0)      1( 0.0)      74( 0.0)     0        0    74
2017-01-14 04:20:11.000     0.000 any        198.[redacted].255.5    1( 0.0)      1( 0.0)      74( 0.0)     0        0    74
2017-01-14 04:20:11.000     0.000 any        198.[redacted].254.244  1( 0.0)      1( 0.0)      74( 0.0)     0        0    74
2017-01-14 04:20:08.000     0.000 any        198.[redacted].254.216  1( 0.0)      1( 0.0)      74( 0.0)     0        0    74
2017-01-14 04:20:09.000     0.000 any        198.[redacted].254.56   1( 0.0)      1( 0.0)      74( 0.0)     0        0    74
2017-01-14 04:20:06.000     0.000 any        198.[redacted]253.173   1( 0.0)      1( 0.0)      74( 0.0)     0        0    74
```

图 6　121.*.60.133 出方向流对宿地址 top 10

5　结论

本文基于 CERNET 南京主节点网络边界获取的流记录，定义并计算了四个测度的端口流量分布特征熵，通过建立面向 5 分钟长度的观测窗口并以 24 小时为周期建立的随机过程在连续的 76 天中获取的样本数据通过了正态分布检验。因此利用正态分布的 $n\sigma$ 原理构造了一个面向端口流量分布特征熵的异常检测算法。该算法在同一网络边界连续 11 天的运行过程中，共实时检测出 29 次端口流量分布特征熵异常，随后的分析表明这些异常由 5 个异常事件导致的，没有误判。这说明这个简单算法的有效性和准确性。本文的研究工作存在两点不足，一是算法没有实现对整个分析周期的正态分布检验，也没有考虑在无法通过正态分布检验的特定窗口的处理方案；二是异常事件的定位的过程还需要在人工的支持下进行。后续的研究工作将在完善这两点不足的基础上展开。

参 考 文 献

[1] 朱应武，杨家海，张金祥．基于流量信息结构的异常检测[J]．软件学报，2010，10（22）：2573-2583.

[2] KOHLER E, LI J, PAXSON V, et al. Observed structure of addresses in IP traffic[J]. IEEE/ACM Transactions on Networking, 2006,14(6):1207-1218.

[3] DUFFIELD N,LUND C,THORUP M. Estimating flow distributions from sampled flow statistics[J].IEEE/ACM Transactions on Networking, 2005,13(5):933-946.

[4] KUMAR A, SUNG M, XU J, et al. Data streaming algorithms for efficient and accurate estimation of flow size distribution [A]. Proceedings of ACM SIGMETRICS[C]. 2004.177-188.

[5] YANG L, MICHAILIDIS G. Sampled based estimation of network traffic flow characteristics[A].INFOCOM 2007 the 26th IEEE International Conference on Computer Communications[C]. 2007.

[6] Lakhina A,Crovella M, Diot C. Mining anomalies using traffic feature distributions. In Proc of the 2005 Conf on

Applications,Technologies, Architectures, and Protocols for Computer Communications.Pennsylvania,2005.217−228.

[7] Yu Gu, Andrew M,Don T.Detecting anomalies in network traffic using maximum entropy estimation[C]//Proec of the 5th ACM SIGCOMM Conf on Internet Measurement(IMC). New York：ACM，2005:345-350.

[8] Bereziński P, Pawelec J, Małowidzki M, et al. Entropy-Based Internet Traffic Anomaly Detection: A Case Study[C]// International Conference on Dependability and Complex Systems Depcos-Relcomex. 2014:47-58.

[9] Bereziński, Przemysław, Jasiul B, Szpyrka M. An Entropy-Based Network Anomaly Detection Method[J]. Entropy, 2015, 17(4):2367-2408.

[10] Callegari C, Giordano S, Pagano M. Entropy-based network anomaly Detection[C]// International Conference on Computing, NETWORKING and Communications. IEEE, 2017:334-340.

Wang Li, born in 1993, master candidate. His main research interests include network measurement and network behavior.

DING Wei, born in 1962, Ph.D., professor, Ph.D. supervisor. Her main research interests include computer integrated manufacturing, general search engine, PKI certificate system, remote education under network environment and network behavior.

Wu Qi, born in 1992, master.Her main research interests include network measurement and network behavior.

Xia Zhen, born in 1976, master. His main research interests include network measurement and network behavior.

Backgroud

The change of network traffic can reflect the situation of network operation to a large extent. Most of the network anomalies will be reflected in the flow patterns. In the event of anomaly traffic, what happened in the network? This is a problem that has so far not been well resolved. Therefore, anomaly traffic detection has always been a hotspot in the field of network management. Practical research work needs to start from two levels, the first is to locate the anomaly in the massive network traffic, the second is to analyze the reason why the anomaly happen and generate the relative responses.

The detection of anomaly traffic behavior is based on measures, and the simplest measure is bandwidth-related, such as bps and pps. Many related work attempts to analyze the distribution characteristics of the traffic characteristics in the network traffic at a certain time interval by analyzing the distribution of the flow characteristics such as the source and destination IP address, the source and destination port, the size of the stream, and the duration of the flow. Information entropy is a good way to integrate these measures, so many researchers choose it as a tool of research.

Port is an important field in TCP and UDP, which is set up to facilitate the simultaneous execution of multiple web applications with the same host, and is typically used to identify different applications. By observing changes in port traffic distribution, we could be able to find exceptions.

Common anomalous flow behavior detection methods include wavelet analysis and time series analysis. The former has a better detection effect, but the calculation process is complex, there is a sliding window size equivalent to the delay, it is not suitable for on-line detection; relatively speaking, the latter have a wider range of applications.

Relative to the anomaly detection, the exception data analysis area is not very effective method can support the detection of anomaly traffic analysis, and locate the possible security events. All the anomaly detection methods are validated by the measured flow rate.

Our group presents an anomaly detection method based on port network traffic feature entropy. It worked in the actual network environment, and it is efficient to complete the anomaly detection mission. By analyzing the detection results, we found all exceptions could be accurately positioned for the specific event. This indicates the effectiveness of the anomaly detection method.