基于特征熵的异常流识别技术

许 倩 程东年 张建辉 程国振
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要 多数识别技术通过建立流特征的正常模型来识别偏离的流,但流特征有较强的可变性,建立这样精微的模型非常困难。异常的发生通常会引起流量地址或端口在分布上的变化,分布的分散或集中程度可用特征熵来衡量。因此提出基于特征熵的异常流识别技术(Entropy of Characteristics based Anomaly Traffic Identification,ECATI),即利用特征熵依据流量特征参数的分布变化检测异常,通过分析异常间隔的流量迭代地排除类似正常的流,从而识别根源流。经过手动标记和人工注入异常的仿真实验证实,所提算法能精确地识别出异常流,在平均识别率89.5%的情况下几乎没有丢失流。识别算法能精确地诊断网络扫描、DDoS攻击和链路失败等多种异常类型。

关键词 特征熵,指数平滑法,分割缩减,异常流识别
中图法分类号 TP393.08 **文献标识码** A

Entropy of Characteristics Based Anomaly Traffic Identification Technique

XU Qian CHENG Dong-nian ZHANG Jian-hui CHENG Guo-zhen
(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract The existing methods build a model describing normal flow characteristics which is used to identify deviating flows. However, building such a microscopic model is challenging due to the wide variability of flow characteristics. The distributions of packet features (IP addresses and ports) observed in traces which can be described by entropy reveal the presence and the structure of a wide range of anomalies. A novel method named Entropy of Characteristics based Anomaly Traffic Identification (ECATI) was proposed. It utilizes entropy of characteristics to detect anomalies and analyzes traffic in anomalous time bins of which detector iteratively removes flows that seem normal. We measured the accuracy of ECATI algorithm using manually labeled anomalies and anomaly injection. The results show that ECATI accurately isolates the anomalous traffic with only few or zero missed flows under over 89.5% of average identification rate.

Keywords Entropy of characteristics, Exponentially weighted moving average, Partition reduction, Anomaly traffic identification

1 引言

研究人员提出了大量异常检测技术,其中部分技术已投入商业应用,其共同点是能把那些对网络管理中心(Network Operations Center,NOC)重要的事件标记为告警,包括流量滥用、洪泛和路由断供[1]。一旦触发告警,就要对异常流量进行根源分析,以探究引起检测器告警的原因。根源分析通常分为异常流提取与根源事件分类两个步骤。图1给出了异常与其根源的关系:1)发生/影响—根源事件影响网络流量的子集;2)触发—总体流量发生相应的改变并在异常检测器中触发告警;3)识别—提取出引发异常的流;4)分类—依据识别的流特征对根源事件分类。根源分析是试图恢复因果链的过程,从告警到异常流,再从异常流分析到根源事件。关键点在于正确识别出引发异常的根源流,即异常流识别[2]。异常流识别除了可以获得更多的异常信息,还能够提高异常检测的准确率或建立较为准确的检测模型。因此,异常流识别是确保网络安全的重要研究课题,对根源分析、攻击缓解和异常检测器的测试都是必不可少的。

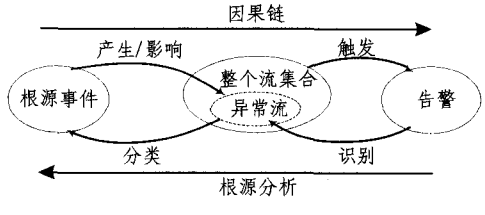

图1 异常与其根源事件之间的关系

目前提出的异常流识别技术的研究工作还很少。B. Krishnamurthy等人[3]通过建立描述正常流特征的模型来检测异常,再利用模型识别出偏离的流。然而,由于流特征具有较强的可变性,建立这样精微的模型非常具有挑战[4]。异常的发生通常会引起流量地址或端口在分布上的变化,因此,从特征参数的分布变化对网络流量进行分析能够有效识别出异常流[5],分布的分散或集中程度可用熵来衡量。信息熵[6]是

到稿日期:2012-02-05 返修日期:2012-06-07 本文受国家高技术研究发展计划(2012CB315901,(2012CB315905)资助。
许 倩(1987—),女,硕士生,主要研究方向为宽带信息网络、网络安全,E-mail:xq-8774@163.com;程东年 男,教授,硕士生导师,主要研究方向为宽带信息网络、网络体系结构、路由协议;张建辉 男,博士,讲师,主要研究方向为宽带信息网络、网络安全;程国振 男,博士生,主要研究方向为宽带信息网络、网络安全。

香农于1948年提出的，是对测量事件平均不确定度的描述。本文提出基于特征熵的异常流识别算法(Entropy of Characteristic based Anomaly Traffic Identification，ECATI)，即在源IP、目的IP、源端口和目的端口等流量特征参数下产生聚合流，在流聚合标准下用特征熵来衡量流量特征参数的分布变化，通过一次指数平滑法对周期内的正常流量做预测并判别异常的发生，在异常的观测周期内分析观测间隔内的流量，重复地排除类似正常的流，从而识别出异常的根源流。

2 基于特征熵的异常流识别算法

基于特征熵的异常流识别算法有两个步骤：(1)基于特征熵的异常检测(Entropy of Characteristics based Anomaly Detection，ECAD)，即利用流特征的分布来分析观测间隔内的流量；(2)流量分割缩减(Traffic Partition Reduction，TPR)，即利用特征参数识别引发异常的根源流。

2.1 基于特征熵的异常检测

信息熵表示所含信息量的多少，是对系统不确定程度的描述。网络流量测量数据来自NetFlow的信息采集，包括源IP、目的IP、源端口、目的端口、分组数和字节数等属性。把测量数据当作离散信源，把测量数据的各个特征属性看作是一组随机事件，就可以对其信息熵进行分析。信息熵能有效地表现出同一属性对应数据的集中和分散程度。在大规模网络流量中，数据越集中，其熵值越小；数据越分散，其熵值就越大。

定义1(特征熵) 给定 m 个一维统计特征集 $D_i=\{d_{ij}\}$，d_{ij} 表示特征参数 i 的第 j 个值的流个数($i=1,\cdots,m;j=1,\cdots,y_i$)，$Y=\{y_i\}$ 为 m 个特征的统计数集，y_i 为统计的时间间隔内不重复的特征 i 的总数。定义参数 i 的特征熵为：

$$H(D_i)=-\sum_{j=1}^{y_i}\frac{d_{ij}}{S_i}\ln\left(\frac{d_{ij}}{S_i}\right) \tag{1}$$

式中，$S_i=\sum_{j=1}^{y_i}d_{ij}$ 表示统计的时间间隔内特征参数 i 相应的总的流个数。

令 $m=4$，$D_i(i=1,\cdots,4)$分别代表4个常用的统计特性：源IP、目的IP、源端口与目的端口。例如，$d_{45}=1000$ 表示特征参数4“目的端口”的第5个值为“80”的流有1000条，$y_4=250$ 表示在统计的时间间隔内共有250个源端口。通过式(1)计算得到 $H(D_i)$ 表示监测所需的源IP地址熵 H(srcIP)、目的IP地址熵 H(dstIP)、源端口熵 H(srcPort)与目的端口熵 H(dstPort)，即统计的时间间隔内4个流特征参数分布的集中或分散情况。

指数平滑法[7]又称指数加权平均法，实际是加权的移动平均法，它是选取各时期权重数值为递减指数数列的均值方法。指数平滑法克服了移动平均法需要 k 个观测值和不考虑 $t-k$ 前时期数据的缺点，通过某种平均方式，以消除历史统计序列中的随机波动，并找出其中的主要发展趋势。一次指数平滑法适用于历史观测呈水平波动、无明显上升或下降趋势情况下的预测，它以该时刻的指数平滑值作为下一时刻的预测值，在计算预测值时对历史数据的观测值给予不同的权重，其预测公式为：

$$y_t=ax_{t-1}+(1-a)y_{t-1} \tag{2}$$

式中，y_t 与 y_{t-1} 分别是 t 时刻与 $t-1$ 时刻的预测值；a 是平滑系数，又称加权因子，取值范围为[0,1]；x_{t-1} 是历史数据序列在 $t-1$ 时刻的观测值。

因此检测算法采用一次指数平滑法对短时间周期内正常流量特征参数进行预测。令 H_{it} 为第 t 个时间间隔内特征参数 i 的特征熵；$\hat{H}_{it}$ 为第 t 个时间间隔内特征参数 i 的特征熵预测值。用一次指数平滑法计算 $\hat{H}_{it}$ 可得：

$$\hat{H}_{it}=aH_{t-1,i}+(1-a)\hat{H}_{t-1,i} \tag{3}$$

把 $\hat{H}_{t-1,i},\cdots,\hat{H}_{t-k,i}$ 的表达式依次代入 $\hat{H}_{it}$ 中，展开整理可得：

$$\hat{H}_{it}=aH_{t-1,i}+a(1-a)H_{t-2,i}+\cdots+a(1-a)^{k-1}H_{t-k,i}+(1-a)^k\hat{H}_{t-k,i} \tag{4}$$

式中，k 表示选取 k 个统计的时间间隔作为一个观测周期。基于线性模型的检测器有无限内存，但是最近 k 个时间间隔对于异常的贡献随着 k 的减小呈指数衰减，较近的时间间隔内的流更像是引发告警的流(即选取较小的 k 值)。因此触发异常的流集合符合最近 k 个时间间隔内在链路中测量的流，若采样间隔为5分钟，观测周期为1小时，则 $k=12$。

$\hat{H}_{t-k,i}$ 为该观测周期内的初始平滑值，一般情况下，可用前一周期最后几个观测值的平均值来代替，或直接用该时刻的观测值。在这里，选用该时刻观测值来代替 $\hat{H}_{t-k,i}$。在实际应用中，a 值是根据时间序列的变化特性来选取的。正常流量时间序列在一个观测周期内波动不大，比较平稳，因此 a 应取小一些。

令 σ_{it} 为第 t 个时间间隔内特征参数 i 的特征熵标准差；T_{it} 表示第 t 个时间间隔内特征参数 i 的特征熵的检测阈值，则：

$$\sigma_{it}=\sqrt{\frac{1}{k-1}\sum_{j=t-k}^{t-1}(H_{ij}-\hat{H}_{ij})^2} \tag{5}$$

$$T_{it}=\hat{H}_{it}\pm 3\sigma_{it} \tag{6}$$

式中，$i=1,\cdots,4$。

因此，第 t 个统计时间间隔内特征参数 i 的熵值的正常取值范围是 $[\hat{H}_{it}-3\sigma_{it},\hat{H}_{it}+3\sigma_{it}]$。当 $|H_{it}-\hat{H}_{it}|\geqslant 3\sigma_{it}$ 时，检测器产生告警。在ECATI算法中把异常检测过程看作一个“黑匣子”，对其形式不做任何假设，用函数 $F(\cdot)$ 来表示。图2为异常检测函数的伪码描述。

```
F(C,k,i)
1. Input: C={x1,x2,…,xn} //the set of flow records
2.        k: the number of reference time bins
3.        i: the flow feature selected
4. Output: ΔH //the anomaly score
5.   {dj}=E{xi}//Feature Extraction
6.   Si=Σ_{j=1}^{yi} dij//the sum of packets with feature i
7.   Hi=-Σ_{j=1}^{yi} (dij/Si)ln(dij/Si)// the entropy of the feature i
8.   compute Ĥi//use the function EWMA()
9.   ΔH=|Hi-Ĥi|
10. return ΔH
```

图2 异常检测函数伪码描述

假定链路中的流量被存储成固定的时间间隔，在每个间隔内分享相同五元组信息(即源/目的IP、源/目的端口号和协议类型)的分组被汇聚成流[8]。在 t 个连续的时间间隔内，

用 $S_1,\cdots,S_t$ 表示至今观测到的每个间隔内的流集合。函数 $F(\cdot)$ 表示异常检测器，把最近 $k+1$ 个间隔内的流 $S=\{S_{t-k},\cdots,S_t\}$ 作为输入，输出一个非负值表示第 t 个时间间隔的异常情况，记作异常得分。F_A 作为异常的判决门限来标记异常间隔内流量的异常得分能否使得检测函数告警，即当且仅当 $F(S)>F_A$ 时，检测器告警。

2.2 流量分割缩减

在识别过程中，有效地缩小异常流的选择范围是一个挑战[2]。假设在给定的检测器告警的时间间隔内，链路中有 S 条流。若对引发异常的事件不加任何假设，则要检查 2^S-1 个流的非空子集，这在拥有上千条流的链路中是不可行的。因此，采用流量分割缩减算法(TPR)来更新异常候选集 C，令 S 为 $k+1$ 个连续时间间隔内链路上的流集合，F_A 为判决门限。当引发报警时，有 $F(S)>F_A$，异常流识别算法的任务是识别涉及异常的流子集 $C\subseteq S$。

在识别过程中 ECATI 算法使用来自 ECAD 检测器的两个输入：1)检测器用来计算告警的流集合 S；2)异常检测器函数 $F(\cdot)$。由于告警是 S 的函数，因此这个流集合一定包含根源事件的流。识别算法从流集合 S 开始，检测器用来计算报警，丢弃对检测器函数 $F(\cdot)$ 来说看似正常的流，通过迭代把流集合 S 减少到更小的子集上。关键在于利用检测器得到的反馈来确定哪些更像是未被异常影响的流。算法只关注那些所有流对某些特征共享一个值的子集(即聚合流[7])。S 中每个流用特征来刻画，用 d_{fi} 表示流 f 的特征 i 的值。例如，一条给定的流，其特征“源端口”的值可能为“80/TCP”。

ECATI 算法每次迭代执行一次流量分割缩减算法(TPR)，具体步骤如下：

(1)输入流特征 i 和一个包含异常的候选流集合 C；

(2)根据 d_{fi} 的不同值把 C 分割开；

(3)返回一个可能仍然包含整个异常的 C 的子集。

定义 2(异常值) 假定包含异常的流集合 S，用异常检测函数计算其函数值，定义为异常值，用 F_S 表示，即 $F_S=F(S)$。

定义 3(补集异常值) 假定有包含异常的流集合 C，根据给定流特征 i 把 C 分割成特征值 d_{fi} 等于 d 的流子集 C_d，$S\backslash C_d$ 表示每个子集 C_d 关于 S 的补集。对于每个特征值 d，用异常检测函数 $F(\cdot)$ 计算补集 $S\backslash C_d$ 的函数值，将其定义为补集异常值，用 F_d^c 表示，即 $F_d^c=F(S\backslash C_d)$。

定义 4(完备异常候选集) 假设单位时间间隔内链路的流集合为 $S=C_1\cup C_2\cup\cdots\cup C_m$，若该间隔内链路中任意一个异常流 $f_A\in C_i$ 且 $f_A\notin C_j(j=1,\cdots,m$ 且 $j\neq i)$，则称 C_i 为完备异常候选集。

令 F_N 为常态门限。

定理 假设在告警的时间间隔内特征参数 i 有 m 个特征值，用异常检测函数 $F(\cdot)$ 计算得到 m 个补集异常值，按其大小排列为 $F_{d1}^c\leqslant F_{d2}^c\leqslant\cdots\leqslant F_{dm}^c$。若满足下述条件：

(a) $F_{d1}^c\leqslant F_N$，已知 $F_N\leqslant F_A$，

(b) $F_{d2}^c>F_A$，

则异常流的特征 i 的值为 d_1，完备异常候选集为 C_{d1}。

证明：若当前时间间隔内发生告警，给定的特征参数 i 有 m 个特征值，链路中的流集合为 $S=C_{d1}\cup C_{d2}\cup\cdots\cup C_{dm}$，通过异常检测函数 $F(\cdot)$ 计算得到 m 个补集异常值 $F_{d1}^c\leqslant F_{d2}^c\leqslant\cdots\leqslant F_{dm}^c$。令 f_A 表示异常流。

根据条件(a)，有 $F_{d1}^c\leqslant F_N\leqslant F_A$ 且 $F_{v1}^c=F(F\backslash C_{d1})$，则对于每个 $f\in C_{d2}\cup C_{d3}\cup\cdots\cup C_{dm}$，$f$ 为正常流。

又因为 $F(S)>F_A$，$S=C_{d1}\cup C_{d2}\cup\cdots\cup C_{dm}$，所以 $\forall f_A\in C_{d1}$，即任意一个异常流 $f_A\in C_{d1}$，则异常的特征 i 的值为 d_1。

根据条件(b)，有 $F_{d2}^c>F_A$，即 $F_{d2}^c=F(S\backslash C_{d2})>F_A$。因此 $C_{d1}\cup C_{d3}\cup\cdots\cup C_{dm}$ 包含全部异常流，即 $\forall f_A\notin C_{d2}$，$f_A\in C_{d1}\cup C_{d3}\cup\cdots\cup C_{dm}$。

又因为 $F_{dm}^c\geqslant\cdots\geqslant F_{d2}^c>F_A$，同理可得 $f_A\notin C_{d3},\cdots,f_A\notin C_{dm}$，即 $f_A\notin C_{dj}(j=2,\cdots,m)$。

综上所述，任意一个异常流 $f_A\in C_{d1}$ 且 $f_A\notin C_{dj}(j=2,\cdots,m)$，由定义 3 可知，$C_{d1}$ 为完备异常候选集。

常态门限 F_N 表示补集异常值为多少时其流集合才被认为是正常的。若 F_N 设置太高(接近于 F_A)，则算法的候选集缩减太快，可能丢弃某些异常流；若 F_N 设置太低(接近于 0)，则算法太过保守，可能在输出的流集合中包含正常流。在仿真实验中，可根据实际情况来调节 F_N 的大小。

定理中条件(a)表示从链路中去除 C_{d1} 后，剩余流量对于异常检测器来说是正常的；条件(b)表示去除该特征的其它任何特征值子集后剩余的流都不影响异常，即其它特征值子集均为正常流集合。当两个条件同时满足时定理成立，判定异常流的特征 i 的值为 d_1，并把原始的完备异常候选集 C 缩减为其子集 C_{d1}。若分割缩减定理的条件式(a)或(b)有一个不满足，则分裂缩减程序跳过该缩减步骤，输出 C 为候选集。然后，利用其它特征参数执行分割缩减算法来筛选完备异常候选集，直到遍历所有特征。图 3 给出了 ECATI 算法分割缩减程序的伪码描述。

```
ECATI 分割缩减算法(S,C,p)
1. 输入：S：整个流集合
2.       C：候选流集合
3.       i：流特征
4. 输出：新的候选集 //C 的子集
5.       D←{d_fi | f∈C}
6.   for all d∈D do
7.       C_d←{f | f∈C and d_fi=d}//划分子集
8.       F_d^c=F(S\C_d)
9.   end for
10.      {Let F_d1^c≤F_d2^c≤…≤F_dm^c}
11.        //补集异常值按大小排列
12.  if F_d1^c≤F_N and F_d2^c>F_A then //判别
13.      return C_d1
14.  else
15.      return C
16. end if
```

图 3 ECATI 的分割缩减算法

2.3 算法复杂度

ECATI 算法首先计算特征熵，每个特征的算法复杂度为 $O(y_i)$(y_i 是特征 i 的不同取值的个数)。采用一次指数平滑法进行预测的过程需要使用 $k(k=12)$ 个历史间隔，因此算法复杂度为 $O(ky_i)=O(12y_i)=O(y_i)$。统计间隔内的流量根据每个流特征的 y_i 个取值，可以分为 y_i 个子集，因此通过异

常检测函数来计算异常得分，需要计算 y_i 次，其算法复杂度为 $O(y_i^2)$。考虑最差情况，即每个特征的不同取值个数 y_i 都取 n，则算法复杂度为 $O(n^2)$。根据 m 个流特征进行流量分割缩减，迭代 m 次后得到输出结果，因此总的算法复杂度为 $O(n^2 m)$。

3 仿真实验

分别采用已标记异常（数据集 A、B）和人工注入异常（数据集 C）来度量 ECATI 算法异常流识别的准确性。

3.1 测试数据集

实验中采用的数据集如表 1 所列。数据集 A 是 2007 年 11 月在连接 Tier-1 ISP 与 GEANT2 的链路上搜集的，包含以 1/1000 的采样率搜集到的流记录。GEANT2 连接了欧洲国家研究教育网（National Research and Education Networks，NRENs），并提供连接到商业网和其他全球研究网的入口。数据集 A 和 B 均含有异常，文献[9]已对数据集 A 中的异常进行手动标记。数据集 B 是 MIT 林肯实验室从本地网络获取的九周原始 TCP 存储数据，这些数据被处理成几百万条连接记录，每条记录包含在一个时间间隔内传输的 TCP 包序列，随机提取大小为 2000 条记录的子集，在数据采集过程中统计了数据集中用于检测与识别的特征集（见表 2），运用 ECAD 检测器来计算统计间隔内流量的异常值，判断其是否超出了异常门限。ECAD 有两个参数：1）平滑系数 a；2）用于计算预测值的时间间隔的个数 k。数据集 C 是从某大型企业网上采集的流量，采用隧道协议进行传输，流量比较干净，几乎不含异常，因此可作为背景流量可控地注入异常。

表 1 实验数据

标号	来源	收集间隔	异常状况	标记状况
A	GEANT2	5min	存在异常	已标记
B	教育网	1min	存在异常	已标记
C	企业网	5min	无异常	* *

表 2 识别的流量特征集

符号	描述
D_1	不重复源 IP 的流个数集
D_2	不重复目的 IP 的流个数集
D_3	不重复源端口的流个数集
D_4	不重复目的端口的流个数集

3.2 实验结果

对每个异常，采用两个度量标准：1）被 ECATI 漏掉的异常流（或分组），即丢失流（Missed Traffic）；2）被 ECATI 识别出却不属于异常集的流量部分，即额外流（Extra Traffic）；3）正确识别为异常的样本占所有流量样本的比例，即准确率（Accuracy）。

3.2.1 手动标记数据仿真

用数据集 A 来验证 ECATI 算法的识别性能，图 4 给出了所有异常的丢失流和额外流（用流和分组表示）的 CDF。ECATI 算法在识别率为 38%的情况下几乎没有丢失流，识别率达到 96%的情况下也仅丢失少于 12%的流，且几乎不会引入额外流。经过分析发现在至少有一条丢失流的情况下，94%是由于短时测量误差引起的异常，说明这类短时测量误差造成的异常很难被识别出。

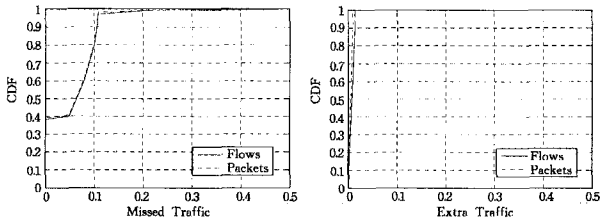

图 4 ECATI 算法在数据集 A 中的识别性能

文献[2]通过关联规则来识别异常流，用 AR 表示该识别算法。ROC 曲线表示异常流识别的准确率与误报率之间的权衡，用来评估识别算法的性能。其中，准确率：正确识别为异常的样本占所有流量样本的比例；误报率：额外流（正常被识别为异常的流样本数）占总流量的比例。图 5 给出了 ECATI 与 AR 算法的 ROC 曲线对比。从图 5 可以看出，在相同误报率水平下 ECATI 算法要优于 AR 算法，且 ECATI 算法在误报率为 5%的条件下识别的准确率能达到 96%。

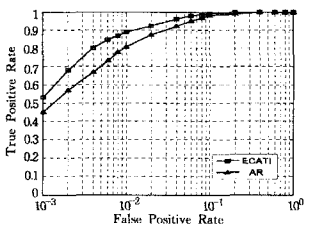

图 5 ECATI 与 AR 算法的识别性能

3.2.2 已标记数据仿真分析

在数据集 B 上运用 ECAD 检测器来计算异常值。设置 ECAD 的参数：观测的时间间隔 $k=12$（一个观测周期），平滑系数 $a=0.3$。然后运行 ECATI 算法的分割缩减程序，在实验中，设置 $F_N=0.5F_A$，即设置常态门限为异常判决门限的 1/2，此门限值在手动标记的异常数据集 A 中运行 ECATI 算法有很好的实验效果。图 6 给出了分割缩减步骤在数据集 B 中利用不同流特征来识别异常流的 4 次迭代。每个图说明在包含异常的 1 小时（一个观测周期）的窗口内每分钟的流个数，且给出了每次迭代后完备异常候选集中的流。

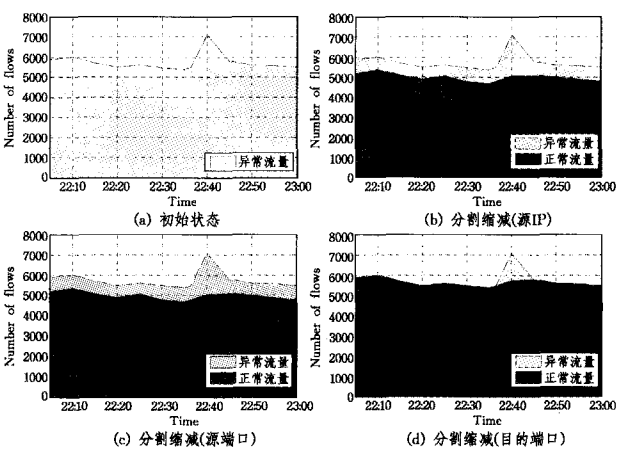

图 6 ECATI 算法的分割缩减过程

图 6(a)表示所有流都是完备异常候选集，即 ECATI 算法的初始状态；图 6(b)表示在识别出源 IP 地址后，完备异常候选集得到很大程度的缩减；图 6(c)为 ECATI 算法采用源端口来缩减候选集，其缩减结果与图 6(b)变化不大，说明源

（下转第 69 页）

ding a Programmable Access Device Having Distributed Service Control[P]. US:7499458B2,2009-03-03

[2] Frias-Martinez V,Sherrick J,Stolfo S J,et al. A Network Access Control Mechanism Based on Behavior Profiles[C]// Annual Computer Security Application Conference. ACSAC'09. Honolulu,2009:3-12

[3] TCG. Trusted Computing Group Timeline [EB/OL]. http://www.trustedcomputinggroup.org/files/resource_files,2011-02

[4] 颜菲,任江春,戴葵,等.基于 TNC 的安全认证协议设计与实现[J].计算机工程,2007,33(12):160-162

[5] 王佳慧.可信网络连接全生命周期接入与授权模型设计[D].西安:陕西师范大学,2010

[6] 邱罡,王玉磊,周利华.一种基于可信计算的 VPN 接入认证方案[J].计算机科学,2009,36(7):76-78

[7] Bell D E,LaPadula L J. Secure Computer System:Unified Exposition and Multics Interpretation[R]. Bedford,MA:The MITRE Corporation,1976

[8] 张俊伟,马建峰,文相在.通用可组合的可信网络连接模型和 IF-T 中的 EAP-TNC 协议[J].中国科学 E 辑,2010,40(2):200-215

(上接第 41 页)

端口对异常的影响不大;在缩减目的端口后,由异常引起的毛刺已经清晰地从正常流中隔离出来(见图 6(d))。

3.2.3 人工注入异常的识别准确性

为了评估 ECATI 流识别算法的准确率,采用异常注入的方式完成受控实验,其优点在于能学习算法的敏感度(如注入不同强度的异常、不同类型的异常)。表 3 给出了数据集 C 中注入的 3 种异常类型,数据集 C 本身是干净的(不包含异常),对每个检测的时间间隔(即 5 分钟),模拟每个间隔内异常对流的影响。若 ECAD 算法检测到注入异常,则运行 ECATI 识别算法。异常情况:1)链路失败,即通过去除给定持续时间内跨越链路的所有流量来仿真[10],设定 20 秒的链路失败时间;2)恶意流量为 DDoS 攻击和网络扫描时,为了使终端主机异常更为真实,可根据相同链路中发现的相似异常流的特征(IP 地址、端口和 AS 号等)来模拟产生流。

表 3 注入异常的类型

异常类型	异常描述
链路失败	去除间隔内流经被监测链路的所有流
DDoS	增加来自不同网络多个源 IP 地址到单一目的 IP 地址或单一目的端口的流
网络扫描	增加从单个源 IP 地址到单个 IP 目的地址的多个目的端口的流

图 7 给出了链路失败与恶意攻击的丢失流和额外流的 CDF。在 20 秒的链路失败中 ECATI 算法的识别错误率少于 2%,且几乎没有引入额外流。对于 DDoS 攻击和网络扫描,算法在识别率超过 82%的情况下能够精确地输出触发的异常流的特征值及完备异常候选集,丢失的流仅为 1%。经分析发现,这些注入的异常至少有一个终端主机有单一的流特征的值。因此,一旦算法识别出唯一的受害机(如 DDoS 和端口扫描)或使用单一的易受攻击的端口(如网络扫描),在间隔内大多数拥有这些特定特征值的流都可能是异常的一部分。

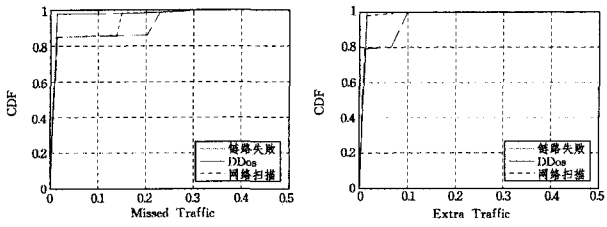

图 7 ECATI 算法对人工注入异常的识别性能

结束语 异常流识别是确保网络安全的重要研究课题,对根源分析、攻击缓解和异常检测器的测试都是必不可少的。ECATI 是一种自动异常流识别算法,算法采用特征熵来衡量流量特征参数的分布变化,用一次指数平滑法对正常流量进行预测来检测异常,在异常间隔内通过分析间隔内的流量来重复地排除类似正常的流,从而识别引发异常的根源流,关键在于利用检测器得到的反馈来确定哪些更像未被异常影响的流。经过真实异常数据集与人工注入异常数据集的实验评估,证实 ECATI 算法能够精确隔离出异常流,在平均识别率 89.5%的情况下识别的丢失流非常少甚至没有。ECATI 算法能有效减少网络管理员的工作,其识别结果能够为异常的正确分类、根源分析做准备。已证实大多数 ECATI 算法的误差是由小型路由事件引起的。尽管本文采用离线数据集来完成仿真实验,但 ECATI 算法能够足够快速地处理数据集以满足在线部署。例如,即使在普通 PC 机的常用硬件条件下,搜集时间为 3 个月的数据集 B 能够使用 ECATI 算法在 4 小时之内完成全部处理。算法依赖于异常检测器的输入,但不使用任何假设,可以不加修改地与任何异常检测器(如 Kalman[9]、PCA[10]与小波等[11])一起使用。本文提出的异常流量检测及识别技术能够用于监测网络,为网络的安全提供保障。

参 考 文 献

[1] Chandolav, Banerjeea, Kumarv. Anomaly Detection: A Survey [J]. ACM Computing Surveys,2009,41(3):1-58

[2] Brauckhoff D,Dimitropoulos X,Wagner A,et al. Anomaly Extraction in Backbone Networks using Association Rules [C]// IMC'09. November. Chicago,Illinois,USA,2009

[3] Krishnamurthy B,Sen S,Zhang Y,et al. Sketch-based change detection:methods,evaluation,and applications[C]// IMC'03:Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement. NewYork,NY,USA,2003:234-247

[4] Lakhina A,Crovella M,Diot C. Diagnosing network-wide traffic anomalies[C]//Proceedings of SIGCOMM. 2004:219-230

[5] Lakhina A,Crovella M,Diot C. Mining Anomalies Using Traffic Feature Distributions[C]//Proceedings of SIGCOMM. August 2005:217-228

[6] Nychis G,Sekar V,Andersen D G. An Empirical Evaluation of Entropy-based Anomaly Detection[C]// Proceedings of IMC. Vouliagmeni,Greece,2008:151-156

[7] Guo R-S, Chen J-J. An EWMA-based process mean estimator with dynamic turning capability[J]. IIE Transactions,2002

[8] Thatte G,Mitra U,Heidemann J. Parametric Methods for Anomaly Detection in Aggregate Traffic[J]. IEEE/ACM Transactions on Networking,2011,19(2)

[9] Silveira F,Diot C,Taft N,et al. ASTUTE:Detecting a Different Class of Traffic Anomalies[C]// Proceedings of ACM SIGCOMM. New Delhi,India,2010

[10] Kandula S,Katabi D,Vasseur J. Shrink:A tool for failure diagnosis in IP networks[C]// Proceedings of ACM SIGCOMM MineNet Workshop. August 2005