基于数据流的异常入侵检测*)

俞 研 郭山清 黄 皓

(南京大学软件新技术国家重点实验室 南京 210093) (南京大学计算机科学与技术系 南京 210093)

摘 要 目前,基于机器学习的异常入侵检测算法通常建立在对整个历史数据集进行等同的学习基础之上,学习到的网络行为轮廓过于依赖历史数据,难以准确反映当前网络通信量的行为特征。同时,算法的时间和空间复杂度较高,难以对网络中持续快速到达的大规模数据报文进行存储与维护。本文提出,一种基于数据流聚类的两阶段异常入侵检测方法,首先在线生成网络数据的统计信息,并利用最能反映当前网络行为的统计信息检测入侵行为。实验结果表明,其检测性能优于基于所有历史数据进行入侵检测的结果,并克服了内存等系统资源不足的问题,增加了系统的灵活性与并行性。

关键词 入侵检测,数据流处理,聚类分析

Anomaly Intrusion Detection Based on Data Stream

YU Yan GUO Shan-Qing HUANG Hao

(National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract Existing anomaly intrusion detection algorithms based on machine learning are usually founded on the equivalent learning of all historical dataset. Therefore, the learned network behavior profiles depend on the historical data heavily, thus behavior characteristics of current network traffic can not be represented exactly. At the same time, the network packets which arrive persistently with high speed and large volume can not be stored and maintained in time because of the high time and space complexity of the anomaly intrusion detection algorithms. So, a kind of two-phase intrusion detection method based on data stream clustering is presented. In the method, the statistical information of the network traffic are collected and generated on line firstly. Then the statistical information which can represent current network situation nicely are used to detect the intrusions. Accordingly, the influence of historical data can be reduced. The empirical results manifest that such a two-phase intrusion detection method has better detection performance than that based on all historical data, as well as resolves the problems of insufficient system resources, such as memory, etc., to improve the flexibility and concurrency of system.

Keywords Intrusion detection, Data stream process, Clustering analysis

1 引言

由于计算机网络的快速发展与日益普及,越来越多的信息通过网络来传输与存储,网络安全变得更加重要,同时也成为备受关注的研究领域。入侵检测(Intrusion Detection)技术作为信息安全领域的关键技术,对保护网络与系统的安全起到了至关重要的作用。

入侵检测的目标是通过监视系统或网络流量、系统审计记录等来发现与识别对网络和系统的入侵行为与入侵企图,是建立在入侵行为与网络或系统的正常行为不同这一假设基础上的[1]。入侵检测方法一般分为两类:误用检测(Misuse Detection)与异常检测(Anomaly Detection)。误用检测方法将已知攻击的攻击特征存储在特征库中,利用模式匹配的方式检测攻击。而异常检测是建立正常行为模式,以网络或系统行为是否偏离正常行为模式为依据来检测攻击。误用检测对攻击行为进行建模,检测已知攻击类型;异常检测对系统的正常行为进行建模,可以对新攻击类型进行检测。

由于基于特征签名的入侵检测方法需要已知攻击的特征,无法对新攻击类型与未知特征的攻击进行检测,因此采用机器学习的方法来建立入侵检测模型在入侵检测领域扮演了越来越重要的角色,得到了广泛而深入的研究[2~4,6~9]。目前,在入侵检测领域中采用的各种机器学习方法,目标都是尽可能建立系统或网络精确的行为轮廓模型,来标识网络或系统行为的异常。

当前的基于机器学习的入侵检测方法,通常是建立在对整个数据集进行等同学习的基础上的,检测结果受历史数据的影响较大,难以真实反映当前网络数据的行为特征。而检测网络入侵行为是否发生,则需要根据最近的网络行为就可以做出判断,并不依赖于整个历史数据集。另外,现存入侵检测算法的时间、空间复杂性较高,且受内存等系统资源的限制,难于对持续、快速到达的大规模原始网络数据进行处理,不适合进行在线检测。

针对以上在异常入侵检测领域存在的不足,本文对基于数据流的异常入侵检测模型进行了研究与探讨。数据流模型源自对网络路由、传感器网络所产生流数据的分析与管理需求,是当前数据库研究领域的一个热点问题[10~13]。由于网络通信量符合流数据所具有的数据持续到达、规模大及到达顺序相互独立等特点,采用数据流模型描述实际的网络通信量,

*)国家 863 计划(2003AA142010);江苏省高技术计划(BG2004030)。俞 研 博士研究生,主要研究方向为网络安全、数据挖掘等;郭山清 博士研究生,主要研究方向为网络安全、机器学习等;黄 皓 教授、博士生导师,主要研究方向计算机软件、信息安全等。

来解决当前入侵检测模型存在的不足是恰当的。

本文提出的基于数据流的异常入侵检测模型将对数据的处理分为统计信息生成部分与入侵检测部分，不但能够存储与维护海量原始网络数据的统计信息，克服了内存等系统资源不足的问题，而且利用最能反映近期网络数据情况的统计信息对最近发生的入侵行为进行检测，减少历史数据对检测结果的影响，提高入侵检测精度。同时，这种两阶段的入侵检测模型，既可以减轻入侵检测系统的负荷，又能够提高系统的灵活性与并行性。

本文的组织如下：第 2 节首先讨论了相关的异常入侵检测与数据流研究工作；第 3 节描述了基于数据流的入侵检测模型，分别给出了统计信息生成与入侵检测算法；第 4 节给出了实验方法与结果，并对结果进行了分析；最后对本文进行了总结。

2 相关工作

自 1987 年 D. E. Denning 提出入侵检测概念[9]以来，入侵检测尤其是异常入侵检测方法得到了深入的研究，采用机器学习的方法[2~4,6~8]来建立入侵检测模型在入侵检测领域发挥了日益重要的作用，各种算法被应用到异常入侵检测领域。Wenke Lee[2]采用数据挖掘算法从系统审计数据中抽取活动模式及特征，并根据获得的特征定义从审计数据中生成入侵检测规则；James Cannady[3]应用神经网络来标识与分类异常的网络行为；S. Mukkamala [4]采用支持向量机(SVM)来抽取网络数据的特征，建立入侵检测模型等。

在以上各种基于机器学习的入侵检测算法中，需要包含正常数据的训练集来对检测模型进行训练。如果训练集中混杂了入侵实例，则训练出来的模型很难在未来检测出相应的入侵行为。同时，在现实中想要获得完全标记或正常的训练集也是非常困难的，因此不依赖于有标记训练集的各种基于聚类分析的异常检测算法也得到了深入的研究。聚类分析[5]是机器学习领域中的一种非监督学习方法，能够针对给定的数据集，根据数据之间的相似性进行分组，使得同组数据之间的相似性最大而组间数据的相似性最小。Markus M. Breunig[6]采用基于密度的聚类算法，通过数据点间的距离来计算对象与其周围邻近对象的隔离程度来确定孤立点；Leonid Portnoy[7]则采用基于划分的聚类算法，根据入侵数据在网络数据总体里的分布情况来检测攻击行为；Q. Wang[8]提出了一种基于模糊连通度的聚类算法，用于检测已知攻击及其变体等。虽然，采用聚类算法来检测异常行为是基于入侵行为本质上不同于正常行为轮廓的假设，但这种假设与网络的实际情况相似，能够实现对网络数据的自然分类。

由于当前的异常入侵检测算法通常针对包含全部历史数据的整个数据集进行等同学习，难以真实描述网络数据随时间变化的特点，检测结果受历史数据影响，不能准确反映网络的行为特征。同时，网络数据具有海量、数据持续到达等特点，使得检测算法受到内存等系统资源的限制。因此，采用数据流模型来描述随时间变化的网络数据并对其进行处理也得到了广泛的研究。数据流处理模型[10]是建立在对到达数据的单次扫描基础上的。Venkatesh Ganti[11]通过维护增量的数据挖掘模型来存储数据集的当前子集，以检测数据流的进化情况；Shai Ben-David[12]通过数据点的距离度量来检测到达数据概率分布的变化等。考虑到网络的复杂性决定了网络的行为随着时间的变化较大，而近期的网络数据报文更能蕴涵网络的近期正常行为轮廓的特点，因此本文建立了一种基于数据流的异常入侵检测模型，利用最能反映网络数据情况的统计信息对入侵行为进行检测，减少历史数据的影响，提高检测效果，同时对原始网络数据的概要处理，也克服了系统资源的限制，提高了算法的并行性。

3 基于数据流的异常入侵检测模型

3.1 入侵检测模型

当前的入侵检测算法通常需要在整个数据集上进行计算[13]。然而，由于网络数据具有海量与快速到达等特点，导致现有的入侵检测算法易受系统资源的限制，效率不高。同时，网络数据的行为特征随时间不断地变化，对当前入侵行为的检测往往更加依赖于近期的网络数据作出判断。而基于整个数据集进行入侵检测的算法，受历史数据的影响，不利于准确地检测入侵行为。因此，本文采用了一种两阶段的入侵检测模型，将数据流入侵检测过程划分为统计信息生成与入侵检测两个阶段。基于数据流的入侵检测模型如图 1 所示。

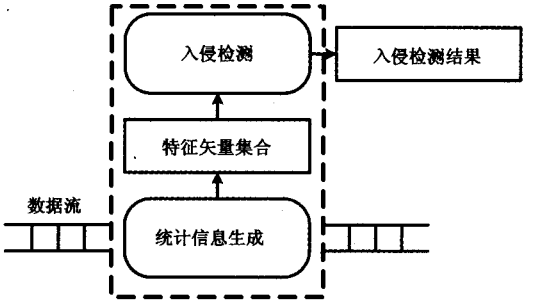

图 1 基于数据流的入侵检测模型

在统计信息生成阶段，通过对不断到达的网络数据流进行单次扫描聚类，生成并维护描述原始网络数据的统计信息。该统计信息描述了数据流聚类所形成的子簇信息，并在文[14]所描述的针对聚类特征矢量(CF, Cluster Feature Vector)的处理方式的基础上，根据数据流的特点，对其进行了时间维上的扩展。同时，定义了针对聚类特征矢量 CF 的加法与减法操作，利用这两种操作符，可以不断地更新并维护检测模型所保存的网络数据信息，而且能够为下一阶段针对当前的入侵检测提供必要的输入信息。

在入侵检测阶段，利用 CF 的减法操作，只针对近期的统计信息进行聚类分析，既减少了入侵检测系统的负荷，提高了系统的并行性，又使之可以只对检测当前攻击行为所必需的网络数据统计信息进行计算，有利于产生尽可能好的检测结果。

3.2 统计信息生成算法

为了准确刻画网络数据的时间特性，本文在文[14]所提出的模型基础上进行了时间维的扩展，并给出了更新与维护网络数据统计信息所需的 CF 加法定义。

定义 1(聚类特征矢量) 异常入侵检测模型中的聚类特征矢量(CF, Clustering Feature)定义为描述包含 d 维数据集合 $\{\cdots,\vec{X}_{i-1},\vec{X}_i,\vec{X}_{i+1},\cdots\}$ 的聚类信息的$(2d+2)$元组，即设给定一个子簇中的 d 维数据集合$\{\vec{X}_i \mid 0<i\leqslant n\}$，则有聚类特征矢量

$$\vec{CF}=(\vec{S},\vec{D},n,t)$$

其中，$\vec{S}=[\sum_{i=1}^{n}x_i^1,\sum_{i=1}^{n}x_i^2,\cdots,\sum_{i=1}^{n}x_i^d]$；

$\vec{D}=[\sum_{i=1}^{n}(x_i^1)^2,\sum_{i=1}^{n}(x_i^2)^2,\cdots,\sum_{i=1}^{n}(x_i^d)^2]$；

n 为子簇中数据的数量；t 为该特征矢量的存储时刻。

在统计信息生成与维护算法中，利用 CF 加法实现将新的网络数据加入现有子簇的操作以及子簇合并的操作。

定义 2(CF 加法) 设异常入侵检测模型中存在聚类特征矢量$\overrightarrow{CF_1}=(\vec{S}_1,\vec{D}_1,n_1,t_1)$与$\overrightarrow{CF_2}=(\vec{S}_2,\vec{D}_2,n_2,t_2)$分别描述了从起始时刻开始至 t_1 与 t_2 时刻得到的两个子簇 C_1 与 C_2 的信息，且 $C_1\cap C_2=\Phi$，则当两个子簇合并时，描述合并后子簇信息的 CF 特征矢量为

$$\overrightarrow{CF_\Sigma}=\overrightarrow{CF_1}+\overrightarrow{CF_2}=(\vec{S}_1+\vec{S}_2,\vec{D}_1+\vec{D}_2,n_1+n_2,t)$$

其中 $t=\max(t_1,t_2)$。

CF 加法操作不但可以实现两个子簇的合并，当新的网络数据加入时同样适用。通过加法操作，可以对描述数据流信息的概要数据结构进行增量计算，使之得到准确的存储和维护，同时该信息可以满足后续阶段聚类分析的计算要求。

统计信息生成算法以持续到达的数据流作为输入，对描述网络连接的数据记录进行聚类，输出的是描述数据统计信息的各子簇的 CF 特征矢量集合。算法的目标是在单次扫描数据流的前提下，以足够细的粒度维护并更新数据的统计信息，使之能满足下一阶段入侵检测部分对攻击检测的要求，同时尽可能保证同一子簇中只包含相同类型的数据点。

k-Means 聚类算法[5]根据相似度距离迭代地更新集合的聚类中心。但标准的 k-Means 算法需要多次对数据集进行迭代计算，不能够满足对数据流进行在线处理的要求。因此，为了从输入的数据流中创建子簇，对 k-Means 算法进行了一定的修改，使之适合于计算持续到达的数据流。算法由空的子簇集合开始，随着数据的到达生成不同的子簇。对于每一个新到达的数据点，计算该数据点到各子簇中心的距离，以获得距离最短的子簇。若该距离小于预先定义的子簇半径的阈值，则将该数据点加入该子簇中，否则判断现有子簇的数目是否超过最大子簇数目。若小于最大子簇数目，则生成新的子簇。否则，合并已有的距离最近的子簇，并为新到达的数据点创建新的子簇。

目前，针对聚类分析中数据点之间距离的度量，已经定义了许多距离函数，如 Euclidean 距离函数、Minkowski 距离函数等。本文采用使用最为广泛的 Euclidean 距离函数来计算数据点之间的距离。算法中各距离的计算公式如下：

设 x 为数据点 o_t 的属性值，y 为子簇 C_i 中心的属性值，且 $y=\frac{1}{|C_i|}\sum_{k=1}^{|C_i|}z_k, z_k\in C_i$，$d$ 表示数据的维数，则数据点 o_t 至子簇 C_i 中心的距离为

$$dist(o_t,C_i)=\sqrt{\sum_{j=1}^{d}(x_j-y_j)^2} \tag{1}$$

设 x 为子簇中各数据点的属性值，y 为子簇 C_i 中心的属性值，$y=\frac{1}{N}\sum_{k=1}^{N}x_k, x_k\in C_i$，$d$ 表示数据的维数，N 表示子簇中数据点的数目，则计算子簇中各数据点到子簇中心的平均距离作为子簇的半径：

$$R'=\sqrt{\frac{1}{N}\sum_{i=1}^{d}\sum_{j=1}^{N}(x_{ji}-y_{ji})^2} \tag{2}$$

将式(2)中等号右边展开，得

$$R'=\sqrt{\frac{1}{N}\sum_{i=1}^{d}(\sum_{j=1}^{N}x_j^2+Ny_i^2-2y_i\sum_{j=1}^{N}x_j)} \tag{3}$$

可以看出，在式(1)与式(3)中，均值 y 可由 CF 聚类特征矢量中的$\vec{S}$求得，而$\sum x_j^2$ 与$\sum x_j$ 则分别可由 CF 聚类特征矢量中的$\vec{D}$与$\vec{S}$求得。

入侵检测模型中的统计信息生成算法如图 2 所示。

算法：统计信息生成算法

输入：代表网络连接记录的数据流 $\{\ldots,o_{t-1},o_t,o_{t+1},\ldots\}$，$o_t$ 为时刻 t 到达的数据点；预先定义的子簇半径的阈值 R，算法所能存储的子簇的最大数目 num。

输出：时刻 T 生成的特征矢量集合。

```
初始化子簇集合 C 为空集;
读入数据点 o_t;
while (o_t 不为空)
    if (C 为空集)
        为 o_t 创建新的子簇 C_i, o_t ∈ C_i;
    else
        计算 o_t 与所有已存在子簇 C_i 的距离 dist(o_t,C_i), 确定最小距离 dist(o_t,C_j);
        if (dist(o_t,C_j) ≤ R)
            将 o_t 加入到子簇 C_j 中, o_t ∈ C_j;
        else
            if (当前已生成的子簇数目 < num)
                为 o_t 创建新的子簇 C_k, o_t ∈ C_k;
            else
                合并距离最小的子簇, 并为 o_t 创建新的子簇 C_k, o_t ∈ C_k;
                计算合并后子簇的半径 R'; if (R' > R) R = R';
    if (时刻 T)
        存储子簇集合 C 作为时刻 T 的特征矢量集合;
    t ← t+1;
    读入数据点 o_t;
```

图 2 统计信息生成算法

3.3 入侵检测算法

入侵检测部分以描述数据统计信息的 CF 特征矢量集合作为输入，可以对不同时间段内代表网络连接记录的数据流的 CF 特征矢量集进行聚类分析，实现对网络连接数据的分类。为了对特定时间窗口的 CF 特征矢量集合进行计算，以反映相应时间段内网络数据的实际情况，需要进行 CF 减法操作。

定义 3(CF 减法) 设入侵检测模型中的聚类特征矢量 $\vec{CF}_1=(\vec{S}_1,\vec{D}_1,n_1,t_1)$ 与 $\vec{CF}_2=(\vec{S}_2,\vec{D}_2,n_2,t_2)$ 分别描述了从起始时刻开始至 t_1 与 t_2 时刻得到的两个子簇 C_1 与 C_2 的信息，且 $t_1<t_2$，$C_1\subseteq C_2$，则当对 $(t_1, t_2]$ 时间窗口内的网络数据进行计算时，对于两个子簇的 CF 特征矢量有

$$\vec{CF}_\Delta=\vec{CF}_2-\vec{CF}_1=(\vec{S}_2-\vec{S}_1,\vec{D}_2-\vec{D}_1,n_2-n_1,t)$$

其中 $t=t_2-t_1$。

聚类特征矢量的减法操作是针对入侵检测中数据到达的选择时间窗口而言的，可以实现对给定时刻之前某一特定时间窗口的网络数据情况进行分析，而不考虑以前的历史数据，这样更能准确反映当前网络的通信量状况，检测对网络或系统的攻击行为。在本文中，我们主要针对近期的网络数据信息进行计算，以检测当前网络的攻击行为。

在基于距离的聚类分析中，计算的是数据对象到相应簇中心的距离。但在本文算法的聚类分析中，输入的数据对象是描述各子簇统计信息的 CF 特征矢量，是对多个数据点集合的描述。简单地以子簇中数据点的平均值作为簇中心，应用现有的距离函数进行计算，并不能获得精确的距离度量。同时，考虑到在入侵检测中，攻击行为多表现为孤立点，选择对孤立点具有更好健壮性的 k-Medoids 算法[5]的修改版本具有更好的效果。k-Medoids 聚类算法的基本策略是通过首先任意为每个聚类找到一个代表对象(medoid)，其他对象根据最小距离原则迭代地加入到各相应的聚类中。在入侵检测算法中，采用了一种利用所有数据点信息的簇间距离计算方法，直接针对子簇内所有的数据点进行计算，充分利用了现有的 CF 特征矢量信息：

设 $SubC_k$ 为输入的子簇，$CenC_l$ 为代表簇 C_l 中心的子簇，N_1 与 N_2 分别为 $SubC_k$ 与 $CenC_l$ 中的数据点数目，$\vec{X}_i$ 与 $\vec{Y}_j$ 分别表示 $SubC_k$ 与 $CenC_l$ 中的 d 维数据点，$i\in[1,N_1]$，$j\in[1,N_2]$，则子簇 $SubC_k$ 至簇中心 $CenC_l$ 的距离为

$$dist(SubC_k,CenC_l)=\sqrt{\frac{1}{N_1N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(\vec{X}_i-\vec{Y}_j)^2} \tag{4}$$

将式(4)中等号右边展开，可得

$$dist(SubC_k,CenC_l)=\sqrt{\frac{1}{N_1N_2}\left[N_2\left(\sum_{i=1}^{N_1}\vec{X}_i^2\right)+N_1\left(\sum_{j=1}^{N_2}\vec{Y}_j^2\right)-2\left(\sum_{i=1}^{N_1}\vec{X}_i\right)\left(\sum_{j=1}^{N_2}\vec{Y}_j\right)\right]} \tag{5}$$

式(5)中，$\sum\vec{X}_i^2$ 与 $\sum\vec{Y}_j^2$ 分别为描述两个子簇的 CF 特征矢量的 $\vec{D}$，而 $\sum\vec{X}_i$ 与 $\sum\vec{Y}_j$ 分别为两个子簇的 CF 特征矢量的 $\vec{S}$，因此上式完全可以由 CF 特征矢量计算得到。

入侵检测算法如图 3 所示。

算法：入侵检测算法

输入：T_1 与 T_2 时刻分别生成的各子簇 $SubC_i$ 的 CF 特征矢量 $\{\vec{CF}_{1i}\mid i\in[1,num]\}$ 与 $\{\vec{CF}_{2i}\mid i\in[1,num]\}$，$num$ 为算法所能存储的子簇的最大数目；预先设置的簇的数目 k。

输出：入侵检测结果。

根据 CF 特征矢量减法计算 $(T_1,T_2]$ 时间窗口内各子簇的 CF 特征矢量 $\{\vec{CF}_{\Delta i}\mid i\in[1,num]\}$；

计算子簇 $SubC_i$ 中数据点数量在所有数据点中的比例 p_i；

以概率 p_i 选择 k 个子簇 $SubC_i$ 作为初始的簇 C_j 中心，$SubC_i\in C_j$，$j\in[1,k]$；

while (簇的中心点发生改变)

　读入新的子簇 $SubC_i$，$SubC_i\notin C_j$，$j\in[1,k]$；

　while ($SubC_i$ 不为空)

　　根据 $\vec{CF}_{\Delta i}$ 计算 $SubC_i$ 与所有簇 C_j 的中心 $CenC_j$ 的距离 $dist(SubC_i,CenC_j)$，确定最小距离 $dist(SubC_k,CenC_l),l\in[1,k]$；

　　将子簇 $SubC_i$ 加入到簇 C_l 中，$SubC_i\in C_l$；

　　$i\leftarrow i+1$；

　　读入新的子簇 $SubC_i$，$SubC_i\notin C_j$，$j\in[1,k]$；

　计算所有簇 C_j 中代表簇中心的子簇 $CenC_j$，使得簇 C_j 中各子簇到其相应的簇中心 $CenC_j$ 的距离平方和最小；

　判断各簇的中心点是否发生改变；

对最终生成的包含不同类型连接记录的各簇，主动标识各簇数据的类型（正常或攻击）；

图 3 入侵检测算法

4 实验与分析

4.1 数据集描述

基于数据流的入侵检测中，所采用的数据集应体现出随时间变化的特点，以便能够更准确地对入侵检测模型进行评价，因此本文选择了 KDD CUP 1999 数据集[15]作为测试集。KDD CUP 1999 数据集是在入侵检测领域广泛使用的训练测试数据集，采用 1998 DARPA 入侵检测数据集来构造连接记录及提取特征[16]。DARPA 数据集包含 7 个星期网络流量的 tcpdump 数据，经过处理大约有 500 万条连接记录。数据中包含 4 种主要的攻击类型：(1)DoS，拒绝服务攻击；(2)R2L，对远程主机的未授权的访问；(3)U2R，对本地超级用户权限的未授权的访问；(4)Probe，扫描与探测行为。数据集中每条连接记录包含了 41 种属性。

由于KDD原始数据集过于庞大，为了对系统进行验证，本文选择了其中具有代表性的10%数据集作为测试集。

4.2 评价指标

基于数据流的入侵检测模型，分为统计信息生成与入侵检测两个阶段。由于这两阶段实现的功能与运行的环境不同，因此评价的标准也不尽相同。对于统计信息生成阶段，需要生成尽可能准确描述数据流信息的数据结构，此时产生的数据统计信息的质量由某一时刻 CF 特征矢量集合中的距离平方和(SSQ)来评价，目标是在内存确定的条件下子簇半径的阈值尽量适中，以保证相同类型的数据点包含在相同的子簇中，同时，也尽量不要生成过多的子簇。此处，$SSQ=\sum_{n}[dist(o_i,C_{o_i})]^2$，其中 o_i 为网络数据，C_{o_i} 为数据 o_i 所属的子簇的中心，$dist(o_i,C_{o_i})$为两点间的距离，n 为数据的数量。另外，为了保证不会因为网络数据的持续到达而导致算法性能的下降，还验证了算法的可扩展性。

对于入侵检测阶段，我们更关心的是评价入侵检测结果的检测率与误报率：

检测率(detection rate)＝检测到的入侵数据样本数/入侵样本总数；

误报率(false positive rate)＝被误报为入侵的正常数据样本数/正常样本总数。

4.3 参数估计

在评价算法的性能之前，首先要确定统计信息生成阶段子簇的半径 R，R 值应使得生成的子簇具有最大的分类正确率，即子簇中应尽可能包含相同类型的数据点。为了尽可能产生准确的结果，本文采用 SSQ 值来确定子簇的半径，选择具有最小 SSQ 值的 R 值作为子簇半径的阈值。采用10%数据集来对 SSQ 值进行测试，所得到的结果应用于随后的实验中。表1列出了不同的 R 值所对应的 SSQ 值。

表1 子簇半径 R

R	SSQ
0.4	11374
0.5	10635
0.6	10682
0.7	9122
0.8	10645
0.9	9958
1.0	10041

从表1中可以看出，当 $R=0.7$ 时，SSQ 的值最小。这表明此时的 R 值足够小以正确检测出代表新的子簇或异常点的数据，同时也不会太小以至于产生过多的子簇。在以下的实验中，均取 $R=0.7$。

4.4 针对数据集总体的实验与分析

为了测试采用两阶段数据处理的入侵检测算法的检测精度，首先以10%数据集作为测试集计算统计信息，然后对统计信息进行聚类分析以检测攻击。此处我们将该数据集作为从起始时刻到当前时刻接收到的数据流，数据匀速到达。

在实验过程中，分别对各种攻击的检测率与误报率进行评价，在表3中给出了实验结果，并将其与Wenke Lee方法[2]的检测结果进行了比较。其中，DR代表检测率，FPR代表误报率。

通过对比，可以发现：(1)对DoS攻击，基于数据流的方法明显优于Wenke Lee方法，而对于Probe与R2L&U2R攻击，基于数据流的方法只是略低于Wenke Lee方法，但在本文算法中，直接对未标记的数据进行处理，而不需要Wenke Lee方法中训练所需的标记数据，因而更具有实用价值；(2)在各类攻击中，算法对DoS攻击的检测效果最好，而Probe与R2L&U2R攻击的检测效果略差，这是因为在整个10%数据集中，高达79%的数据为DoS攻击，由DoS攻击组成的簇在分类过程中占据压倒多数，使得部分Probe攻击与R2L&U2R攻击数据混杂在DoS簇中，造成了分类错误，而正常的网络数据中，DoS攻击与Probe攻击大致均衡，R2L&U2R攻击所占据的比例也会有相应提高，算法对各类型攻击的检测结果也会有一定程度的改善。

表2 入侵检测结果

Attack Type	Method based on data stream		Method of Wenke Lee	
	DR	FPR	DR	FPR
DoS	97.86%	2.21%	79.9%	—
Probe	77.64%	0.02%	97.0%	—
R2L&U2R	55.52%	0.18%	60.0%	—

4.5 针对当前选择时间窗口的实验与分析

基于数据流的入侵检测算法的特点是在数据流环境中能够对给定时间段内的攻击进行检测，尤其是能够发现当前网络通信量的异常。因此，本节对不同时刻的入侵检测能力进行测试，并将检测结果与对整个数据集进行分析的入侵检测结果进行了比较。本文将10%数据集划分为五个子集。令 $index(x)$ 表示数据记录 x 在数据集中的序号，将10%数据集划分为$\{x_1 \mid index(x_1)\in[0,100000]\}$，$\{x_2 \mid index(x_2)\in[0,200000]\}$，$\{x_3 \mid index(x_3)\in[0,300000]\}$，$\{x_4 \mid index(x_4)\in[0,400000]\}$与$\{x_5 \mid index(x_5)\in[0,494020]\}$5个子集，分别表示在5个时刻对统计信息进行存储。这里将数据集划分为以上5个子集，首先是假设数据匀速到达，可以用数据数量的度量来代替时间度量；其次，是考虑到了实验的简便性，在实际应用中可以根据需要来确定存储统计信息的时刻。

实验中，令 $t_1=100,000$、$t_2=200,000$、$t_3=300,000$、$t_4=400,000$ 及 $t_5=494,020$，分别对以上5个数据子集测试从时刻0开始的入侵检测精度，然后在利用在不同时刻存储的统计信息测试在以下两种情况下的入侵检测结果：

Case1：对时间段$[0,t_1]$、$[0,t_2]$、$[0,t_3]$、$[0,t_4]$、$[0,t_5]$内的攻击进行检测，这些测试代表了对从时刻0开始到指定时刻持续到达的数据进行攻击检测；

Case2：对时间段$[0,t_1]$、$(t_1,t_2]$、$(t_2,t_3]$、$(t_3,t_4]$、$(t_4,t_5]$的攻击进行检测，这些测试代表了在指定时刻对近期到达的数据进行攻击检测，此时不需要对指定时间段的起始时间之前的数据进行计算。产生的检测结果即为本文算法的检测结果。

图4至图9分别对以上两种情况下DoS、Probe及R2L&U2R攻击的检测率与误报率进行了对比。

图4与图5为DoS攻击的检测率与误报率，图中表明Case2情况下对各时刻DoS攻击的检测率均好于Case1情况，且Case2的误报率明显低于Case1，特别地，在$(t_1,t_2]$阶段Case2的检测率为100%，误报率为0%，与10%数据集中的实际情况完全吻合，该时间段到达的数据均为DoS攻击数据。

图6与图7分别为Probe攻击的检测率与误报率，可以看出Case2情况下的检测率整体高于Case1，误报率虽稍高于Case1，但即使是在误报率最高的$(t_3,t_4]$阶段，其误报率也仅为1.33%，仍在可接受的范围之内。

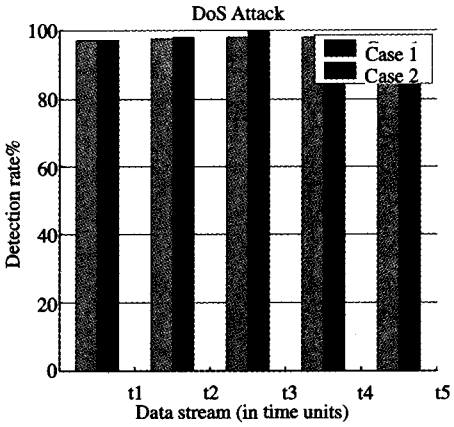

图 4　DoS 攻击检测率

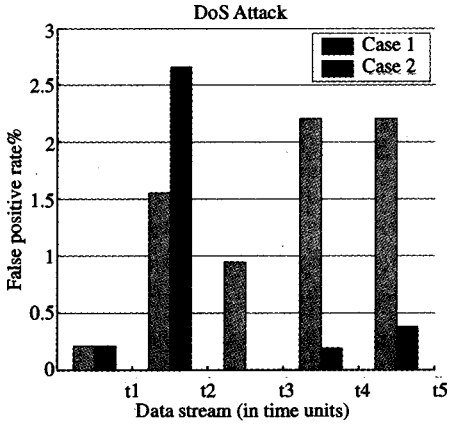

图 5　DoS 攻击误报率

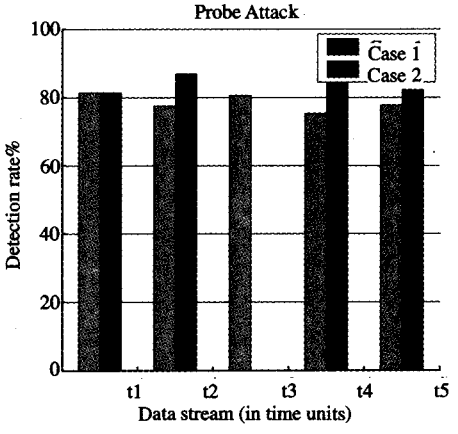

图 6　Probe 攻击检测率

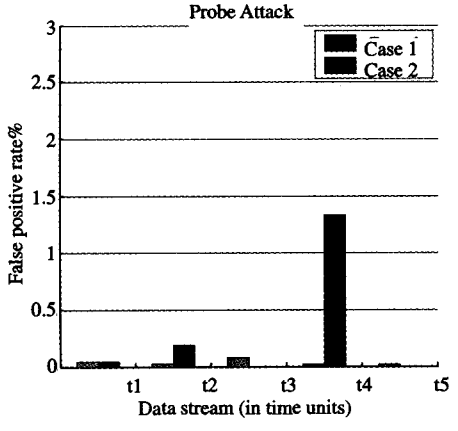

图 7　Probe 攻击误报率

图 8 与图 9 分别显示 R2L&U2R 攻击的检测率与误报率。由于在 10%数据集中，R2L&U2R 攻击的数目极少，且在时间分布上极不均衡，因此在 R2L&U2R 攻击数据相对较多的$[0,t_1]$、$(t_1,t_2]$时间段，Case2 情况下的检测率与误报率均好于 Case1 时的情况，而在 R2L&U2R 攻击极少甚至没有的$(t_2,t_3]$、$(t_3,t_4]$及$(t_4,t_5]$时间段，R2L&U2R 攻击混杂在 DoS 或 Probe 攻击中，难以检测，Case2 的检测率与误报率虽然趋近于 0，但也比较真实地反映了攻击的情况，即该时间段 R2L&U2R 攻击极少或没有出现。

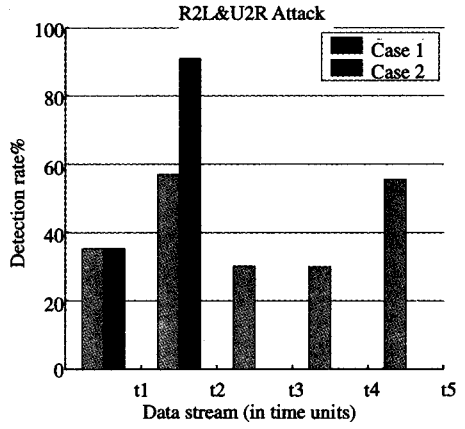

图 8　R2L&U2R 攻击检测率

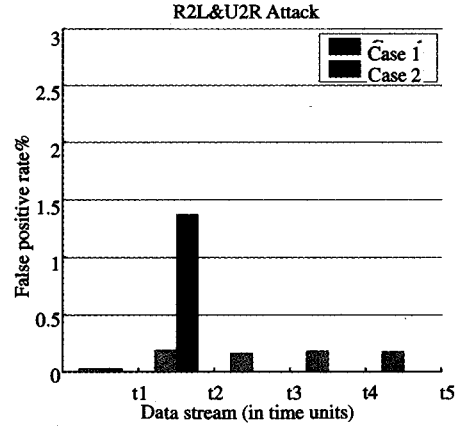

图 9　R2L&U2R 攻击误报率

由 Case1 与 Case2 两种情况下，分别对 DoS、Probe 与 R2L&U2R 攻击的检测结果进行了比较，可以看出，Case2 情况下得到的结果均好于 Case1 的结果。这说明对利用当前指定时间段的网络数据对入侵进行检测，其入侵检测结果与基于所有历史数据进行检测所得到的结果相比，可以更准确地反映当前网络数据的实际情况，具有更好的检测精度。

4.6　扩展性

由于数据流持续到达，而且数据量未知，因此需要快速有效地进行处理。在本文算法采用的两阶段入侵检测模型中，需要对统计信息生成算法的可扩展性进行测试，以保证不会因为数据的不断到达而导致算法性能下降。

在统计信息生成算法中，最频繁的操作是当新的数据到达时发现与该数据距离最近的子簇，以及子簇的合并操作。显然，由于所维护的子簇数量是由系统内存决定的，是固定的，因此发现距离最近子簇的操作时间也是固定的。但是，子

（下转第 114 页）

点(PDP)来处理 XACML 请求。系统中采用文件的形式作为策略库,使用 sunxacml 中的 API,策略很容易地被载入 SimplePDP 中。SimplePDP 中的公开接口是

evaluate(RequestCtx request)

SAMLReceiver(PEP)通过调用该接口获得 ResponseCtx 对象。ResponseCtx 类描述了 XACML 响应,SAMLReceiver(PEP)依据 ResponseCtx 对象来决定是否将当前的 SOAP 请求消息传递到实际的 Web 服务。

实验中,我们通过客户端向目标服务发送了多个请求,每个 SOAP 请求中都嵌入了不同的 SAML Token,但只有满足条件的请求才能真正调用 Web 服务,其他请求都被 PEP 拒绝。结果表明,PEP 和 PDP 在服务器端协同合作,能够对 SOAP 请求进行有效的过滤,从而防止相关资源被非法访问。

总结 Web 服务的安全性是制约其发展的关键因素,有效的访问控制机制对 Web 服务至关重要。相比传统模型,基于属性的访问控制(ABAC)具有高度的动态性和灵活性,能够细粒度地进行授权,有效地保护系统资源。同时,该模型与 SAML 和 XACML 标准结合,具有更强的互操作性。

参 考 文 献

1 Wonohoesodo R, Tari Z. A role based access control for Web services. In: IEEE International Conference on Services Computing(SCC 2004), 2004. 49~56
2 Bertino E, Squicciarini A C, Mevi D. A fine-grained access control model for Web services. In: IEEE International Conference on Services Computing(SCC 2004), 2004. 33~40
3 Bhatti R, Bertino E, Ghafoor A. A trust-based context-aware access control model for Web-services. In: IEEE International Conference on Web Services(ICWS'04) Proceedings, 2004. 184~191
4 The Security Assertions Markup Language (SAML) OASIS TC Homepage. http://www.oasisopen.org/committees/tc_home.php? wg_abbrev=security
5 The XML Access Control Markup Language (XACML) OASIS TC Homepage, http://www.oasisopen.org/committees/tc_home.php? wg_abbrev=xacml
6 Web Services Security (WSS) OASIS TC Homepage. http://www.oasisopen.org/committees/tc_home.php? wg_abbrev=wss
7 Axis Architecture Guide. http://ws.apache.org/axis/java/architecture-guide.html
8 Galbraith B, Hankison W, et al. Web 服务安全性高级编程. 北京:清华大学出版, 2003

(上接第 71 页)

簇的合并操作与数据的偏差程度有关,而与新数据到达的数量无关,过多的子簇合并操作会影响到算法的处理速度。图 10 显示了随着数据流的到达算法的处理速度。

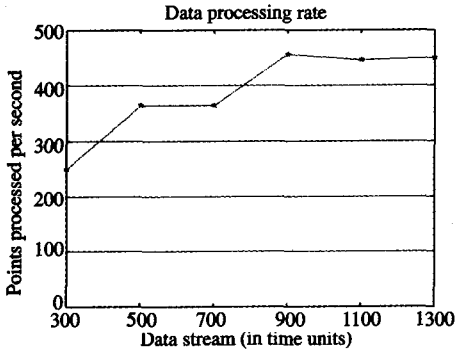

图 10 数据流处理速度

图 10 中可以看出,在开始阶段,由于子簇半径的阈值设置较小,子簇的合并操作较多,处理速度相对较低。随着数据的持续到达,内存中维护的子簇已经能够充分代表网络数据中的绝大部分数据分类,相应地,新到达的数据将被分配到相应的子簇中,此时的操作主要是将新数据加入相应子簇的操作,而非子簇合并操作。因此,当数据流的处理达到一定阶段时,算法的处理速度趋于稳定。

由以上的实验结果可以看出,两阶段的基于数据流的异常入侵检测算法既能够有效地解决网络数据快速到达与检测设备处理能力相对较慢之间的矛盾,又能够充分利用近期的网络数据,更能准确反映当前网络行为的特点,其检测性能优于基于所有历史数据进行入侵检测的算法。

结论 本文提出了一种两阶段的基于数据流的网络异常入侵检测方法,该方法能够在有限的系统资源情况下,以足够快的速度与细粒度保存与维护网络数据的统计信息,并利用该统计信息对不同阶段的网络入侵行为进行检测。理论分析与实验验证的结果显示,这种两阶段的在入侵检测算法,既克服了高速网络环境下内存等系统资源不足对入侵检测算法的影响,又能够准确地反映出现阶段网络数据的行为特征,其检测性能优于基于所有历史数据进行入侵检测的结果,并可以增加系统的灵活性与并行性。同时,算法直接利用未标记的网络数据进行分析计算,也增加了入侵检测方法的实用性。

参 考 文 献

1 Endorf C, Schultz E, Mellander J. Intrusion Detection & Prevention. McGraw-Hill, 2004
2 Lee Wenke, Stolfo S J, Mok K W. A data mining framework for building intrusion detection models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, 1999
3 Cannady J, Mahaffey J. The Application of Artificial Neural Networks to Misuse Detection: Initial Results. In: Proceedings of the 1st International Workshop on Recent Advances in Intrusion Detection (RAID 1998), 1998
4 Mukkamala S, Sung A H. Feature Ranking and Selection for Intrusion detection Systems. In: Proceedings of International Conference on Information and Knowledge Engineering, 2002. 503~509
5 Han Jiawei, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001
6 Breunig M M, Kriegel H P, Ng R T, et al. LOF: Identifying density-based local outliers. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Dallas, 2000. 93~104
7 Portnoy L, Eskin E, Stolfo S J. Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), Philadelphia, 2001
8 Wang Q, Megalooikonomou V. A clustering algorithm for intrusion detection. In: SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005. Orlando, Florida, USA, Mar. 2005
9 Denning D E. An Intrusion Detection Model. IEEE Transaction on Software Engineering, 1987, SE-13: 222~232
10 Babcock B, Babu S, Datar M, et al. Models and issues in data streams. In: Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Madison: ACM Press, 2002. 1~16
11 Ganti V, Gehrke J, Ramakrishnan R. Mining Data Streams Under Block Evolution. SIGKDD Explorations, 2002, 3(2): 1~11
12 Ben-David S, Gehrke J, Kifer D. Detecting Change in Data Streams. In: Proceedings of VLDB, 2004
13 Muthukrishnan S. Data streams: algorithms and applications. In: Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms, 2003
14 Zhang Tian, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Montreal, 1996. 103~114
15 KDD99. KDD99 cup dataset. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999
16 Lee Wenke, Stolfo S J. A Framework for Constructing Features and Models for Intrusion Detection Systems. ACM Transactions on Information and System Security, 2000, 3(4): 227~261