文章编号：1001－9081(2010)03－0699－03

聚类和时间序列分析在入侵检测中的应用

王令剑，滕少华
(广东工业大学 计算机学院，广州 510006)
(the13thking@163.com)

摘 要：入侵检测通过收集各种网络数据，从中分析和发现可能的入侵攻击行为。聚类算法是一种无监督分类方法，能够很好地用于入侵检测。提出一种基于聚类分析和时间序列模型的异常入侵检测方法，该方法不需要手动标示的训练数据集就可以探测到很多不同类型的入侵行为。实验结果表明，该方法用于入侵检测具有较高的检测率和较低的误报率。

关键词：入侵检测；数据挖掘；聚类；时间序列
中图分类号：TP18　　文献标志码：A

Application of clustering and time-based sequence analysis in intrusion detection

WANG Ling-jian, TENG Shao-hua
(Faculty of Computer, Guangdong University of Technology, Guangzhou Guangdong 510006, China)

Abstract: Intrusion detection system can discover potential intrusion behavior by collecting and analyzing various network data. Clustering algorithm is an unsupervised machine learning method well applied in intrusion detection. In this paper, an algorithm of intrusion detection was explored based on clustering analysis and time-based sequence analysis. It is able to detect many different types of intrusion without manually classified data for training. The experimental results show that the algorithm is feasible and effective. It has higher detection rate and a lower false positive rate.

Key words: intrusion detection; data mining; clustering; time-based sequence

0 引言

入侵检测系统(Intrusion Detection System, IDS)是一种主动的安全防护措施，它从系统内部和各种网络资源中主动采集信息，分析可能的入侵攻击行为。从检测技术可以将入侵检测分为两类：滥用检测和异常检测[1]。滥用检测是根据对攻击的先验知识建立入侵模式，利用已知的攻击模式来发现攻击，它对已知攻击的探测非常有效，但无法探测未知攻击。异常检测是寻找网络中与正常网络行为偏离的行为，利用已经建立的正常用户和系统的行为特征检测出新的和未知的入侵攻击行为，它可以检测出未知攻击，但是误报率高。

异常检测的思想是根据对历史数据的学习，建立单独网络环境下的正常行为判别模型，据此区分异常事件。然而，相对确定的攻击特征，正常行为的特征定义要更加抽象和模糊不清。为了建立一个正常行为模型，只有用给定的没有任何攻击的，纯净的数据集进行训练，而这样的数据集实际上很难获取。聚类分析可以缓解这个问题，聚类算法是一种无监督学习算法，它不依赖于预先定义的类和类标号的训练实例，它可以在未标记的数据集上进行训练，通过离群点来判别异常行为。

基于聚类的入侵检测方法是现阶段入侵检测的主要研究方法之一。Portoy等人[2]较早提出了利用基于距离度量的聚类算法进行入侵检测。Lazarevic等人[3]提出了使用孤立点分析来检测入侵的方法。国内的如中国科学院向继等人[4]、罗敏等人[5]都对无监督聚类算法用于入侵检测进行了实验研究分析，取得了很好的效果，证实了可行性。传统聚类算法如k-means算法存在许多问题，如容易陷入局部最优解，只能检测类间大小近似相等的数据集，在实际应用中初始聚类个数难以确定，常产生空的聚类等。本文为解决这些问题，提出一种新的基于聚类和时间序列分析的入侵检测算法，经实验证明具有较好的检测效果。

1 聚类分析算法

1.1 聚类分析

聚类分析应用在入侵检测中是基于以下两个基本的假设[2]：1)正常数据的数量远远大于攻击的数据量；2)攻击数据在某些属性的取值上明显偏离正常的取值范围。聚类分析法是在训练时根据相似度将数据集合划分成若干个称为类的子集，使同一个类中的对象具有尽可能大的相似性，而不同类中的对象具有尽可能大的差异性，并为这些类加以标记表明它们是正常还是异常，然后在检测时将从网络上采集到的数据记录划分到各个类中，根据类的标记来判断网络数据是否异常。

1.2 聚类算法

聚类分析算法主要分为：基于划分的方法、基于层次的方法、基于密度的方法、基于网格的方法和基于模型的方法。基于划分的k-means算法[6]具有简单、计算复杂性小和快速收敛等优点，用于异常入侵检测十分有效，是最典型的常用的聚类算法。它的基本思想是：首先从N个数据对象中随机选择k个对象，每个对象初始地代表了一个类的平均值，即为初始

收稿日期：2009－07－01；修回日期：2009－09－04。
基金项目：广东省自然科学基金资助项目(06021484)；广东省科技计划项目(2008A060201011)。
作者简介：王令剑(1984－)，男，江西赣州人，硕士研究生，主要研究方向：数据挖掘、网络安全；　滕少华(1962－)，男，江西南昌人，教授，主要研究方向：网络安全、数据挖掘、软件开发、软件工程。

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

聚类中心，然后将剩余的每个对象根据与这些聚类中心的相似度，分别赋予与其最相似（也就是距离最近）的聚类；再重新计算每个所获新聚类的聚类中心（即该聚类所有对象的平均值），不断重复迭代，直到聚类中心值不再变化为止。

聚类算法通过设定阈值，将算法生成的类划分成正常和异常类，但某些接近阈值的类，其内部可能既有正常数据，也有异常数据，且两种数据相差并不太大，划到正常类影响检测率，使漏报率偏高，划到异常类会增加误报。对这些归属模糊的类，可以将其单独划出，用时间序列分析法对落入该类的记录进行二次检测。

因此，本文对聚类算法加以改进，并用时间序列模型的检测方法作为补充，得到一种新的基于聚类和时间序列分析的入侵检测算法。

2 时间序列分析

时间序列是按照时间顺序收集到的一系列观测值。记第 q时刻的观测值为 $w(q)$，则 $w(1)$，$w(2)$，$w(3)$，…，$w(q)$，…就是一个时间序列。

入侵检测的时间序列分析[7]是将事件计数和/或资源耗用情况根据时间排成序列，如果在某特定时间内发生概率较低的某个事件发生了，则该事件可能是入侵事件。

本文的用于入侵检测的时间序列模型如表 1所示。

表 1 用于入侵检测的时间序列模型

属性	第 1次	第 2次	…	第 K次	均值	第 K+1次	相对偏差	阈值
第 1属性	$w(1,1)$	$w(2,1)$	…	$w(K,1)$	$avg(1)$	$w(K+1,1)$	$re(1)$	$th(1)$
第 2属性	$w(1,2)$	$w(2,2)$	…	$w(K,2)$	$avg(2)$	$w(K+1,2)$	$re(2)$	$th(2)$
⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮
第 L属性	$w(1,L)$	$w(2,L)$	…	$w(K,L)$	$avg(L)$	$w(K+1,L)$	$re(L)$	$th(L)$

设已对网络的 K个数据连接进行了检测，每个连接分为 L个特征属性，用 $w(,i\ j)$代表第 i个数据连接的第 j个特征值，$avg(j)$表示前 K个数据连接的第 j个特征值的平均值：

$$avg(j)=\frac{1}{K}\sum_{i=1}^{K}w(,i\ j),\quad j=1,2,\cdots,L \tag{1}$$

对第 K+1个数据连接进行检测时，以 $avg(j)$为该数据第 j个特征的期望值，$w(K+1,j)$为实际值，$re(j)$为相对偏差，

$$re(j)=\frac{|w(K+1,j)-avg(j)|}{avg(j)} \tag{2}$$

为数据连接的每一个特征属性设立一个阈值 $th(j)$，如果 $re(j)>th(j)$，则认为第 K+1个数据连接发生异常。

3 入侵检测算法

本文的入侵检测算法分为 4个模块：数据预处理模块、聚类算法模块、标类与时间序列训练算法模块和检测算法模块。

3.1 数据预处理模块

3.1.1 数据标准化

在对检测数据进行聚类之前要先对数据进行标准化，由于数据各属性采用不同的度量标准，如果不进行预处理可能产生大数吃小数的问题。为了消除因值域差别和度量不同对距离的影响，需将属性值标准化，从原来的度量值转化成无单位的值。

首先计算平均绝对偏差和平均值：

$$S_f=\frac{1}{n-1}\sum_{i=1}^{n}(x_{if}-m_f)^2 \tag{3}$$

$$m_f=\frac{1}{n}\sum_{i=1}^{n}x_{if} \tag{4}$$

其中：m_f为第 f个属性的平均值；S_f为第 f个属性的平均绝对误差；x_{if}表示第 i条记录的第 f个属性。然后计算标准化度量值，Z_f为标准化的第 f个属性值，其表示为：

$$Z_f=\frac{x_{if}-m_f}{S_f} \tag{5}$$

3.1.2 相似性度量

数据标准化处理以后就可以进行数据属性值的相似性度量。本文使用的 k均值算法是通过计算对象之间的距离来划分相似的类。在入侵检测记录中包括两种属性：连续型数值属性和离散型符号属性。记录中连续型数值属性之间的距离直接可以采用欧几里得距离来度量。离散型符号属性之间的距离可以用两条记录间含有不同符号特征的离散属性的个数来度量。例如某离散属性有 N种类型，N种不同类型的离散属性值就对应于 0到 N-1的不同数值。对于两个记录对象 x_i和 x_j设有 p个连续型数值属性，m个离散型符号属性，两条记录之间的距离 $d(x_i,x_j)$表示为：

$$d(x_i,x_j)=d_p(x_i,x_j)+\partial d_m(x_i,x_j) \tag{6}$$

$$d_p(x_i,x_j)=\sqrt{\sum_{t=1}^{p}(x_{it}-x_{jt})^2} \tag{7}$$

其中：$d_p(x_i,x_j)$是记录 x_i和 x_j的连续型数值属性之间的欧几里得距离；$d_m(x_i,x_j)$是记录 x_i和 x_j的离散型符号属性之间的距离；∂是权重因子，表示离散型符号属性之间的距离在度量记录间距离中所占的权重。

3.2 聚类算法模块

此模块用聚类算法，将数据集中的记录按特征属性的相似度划分成不同的类别，使正常数据与异常数据分离。聚类算法使用 k均值的改进算法。方法是：选取一个常数 R作为聚类半径，计算当前记录与所有聚类中心距离的最小值，若发现此最小值大于聚类半径 R，则将这个记录作为新的聚类中心，否则将记录加入最近的聚类。当聚类结果中出现空聚类时，采用的方法是将离聚类中心最远的对象移出所在的聚类，来产生新的聚类中心，用新产生的聚类来替代这些空聚类。采用“最大最小值”算法[8]确定初始聚类中心。算法具体描述如下：

步骤 1 用最大最小值法选取 k个初始聚类中心 $\{m_1,m_2,\cdots,m_k\}$，每个中心 m_i对应一个聚类 C_i；

步骤 2 对未被初始化的每一个数据记录 x_i（$i\in\{1,2,\cdots,n\}$），计算数据记录 x_i到聚类中心的距离的最小值 Min；

步骤 3 若 Min< R，将 x_i分配给最近的聚类，否则，产生一个新聚类，将 x_i作为新聚类的中心；

步骤 4 转到步骤 2直到所有数据记录都完成；

步骤 5 以每个聚类的平均值替代原来的聚类中心；

步骤 6 如果出现空聚类，则移出离聚类中心最远的点，产生新的聚类中心，用新产生的聚类来替代这些空聚类；

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

步骤 7　直到聚类中心点的值不再变化为止。

3.3　标类与时间序列训练算法模块

此模块对聚类算法生成的聚类进行标记,将包含数据量大于正常阈值的类标记为正常类,把包含数据量小于异常阈值的类标记为异常类,介于两个阈值之间的则标记为 unclear类。之后以正常类里的数据作为时间序列的训练数据,生成时间序列模型。假设 C_j($j=1,2,\cdots,num\_cluster$)为已生成的聚类,η_1为正常阈值,η_2为异常阈值,n为总记录数,算法描述如下:

步骤 1　$j=1$;

步骤 2　若 C_j记录数 $> n\times\eta_1$,则将 C_j标记为正常类;

步骤 3　若 C_j记录数 $< n\times\eta_2$,则将 C_j标记为异常入侵类;

步骤 4　否则将 C_j标记为 unclear类;

步骤 5　$j++$;

步骤 6　重复步骤 2~4直到 $j> num\_cluster$;

步骤 7　选定标记为正常类中记录数最大的聚类 C_i;

步骤 8　对 C_i中的每个数据记录 x_i提取其特征属性 w_i,w_{ij}为 x_i第 j个属性的特征值;

步骤 9　对 C_i中的所有数据,按式(1)求出 $avg(j)$。

3.4　检测算法模块

当从训练数据中得到聚类算法生成的聚类,用标类算法标好类,再得到训练完的时间序列模型后,就可以用它们检测网络入侵行为了,检测算法描述如下:

步骤 1　获取当前用户某一时刻使用网络情况的原始数据 x;

步骤 2　利用在预处理算法中得到的统计数据将 x标准化,即 $x\rightarrow x'$;

步骤 3　$j=1$;

步骤 4　计算 C_j的中心 m'_j与 x'的距离,即 $dist(m'_j,x')$;

步骤 5　$j++$;

步骤 6　重复步骤 4~5直到 $j> num\_cluster$;

步骤 7　找到最小的 $dist(m'_j,x')$,并得到 m'_{min}所属的类的类标 label;

步骤 8　若 label是正常,则 x是正常网络连接,若 label是异常,则 x是入侵网络连接;

步骤 9　若 label是 unclear,提取 x的特征属性,视其为表 1的第 $K+1$次检测值,按式(2)计算 x对于训练算法求出的 $avg(j)$的相对偏差 $re(j)$;

步骤 10　比较 $re(j)$与阈值 $th(j)$,如 $re(j)> th(j)$则 x是异常入侵连接,否则是正常网络连接。

4　实验及结果分析

4.1　实验数据集的选择

本文选用的数据集是 KDD Cup 99数据[9]中的“kddcup data_10 percent”。攻击数据分为 4大类型:DoS(Denial of Service)、U2R(User to Root)、R2L(Remote to Local)和 Probe。通过分析发现 DoS、Probe两种类型的攻击数据比较满足第二个假设:攻击数据在某些属性的取值上明显偏离正常的取值范围。所以分别对这两种攻击进行检测。首先选取了包含 DoS拒绝服务攻击的记录共 20 774条,其中 DoS攻击记录 382条,包括 smurf、neptune、back、teardrop、pod和 land等攻击。另外选取了包含 Probe攻击的记录共 20 586条,其中 Probe攻击记录 367条,包括 portsweep、ipsweep和 Satan等攻击。数据集中的每个连接记录有 41个特征属性,只选取其中 13个关键属性,以其中的 6个关键数值属性进行聚类,以另外 7个属性作为时间序列模型的观测值。

4.2　实验结果分析

实验是在 CPU 1.73 GHz、内存:1 024 MB、OS:Windows Vista、开发环境:VC++ 6.0下进行的。设初始聚类个数 $m=3$,异常数据阈值为 2%,正常阈值 2.5%。用攻击检测率和误报率来衡量检测结果。表 2、3分别是对 DoS和 Probe攻击的检测实验结果。

表 2　对 DoS攻击的检测结果

聚类半径 R	聚类个数	检测率/%	误报率/%
7	47	94.3	4.5
10	40	92.5	2.2
15	31	85.4	2.5
20	23	82.5	1.9
25	17	70.8	1.4

表 3　对 Probe攻击的检测结果

聚类半径 R	聚类个数	检测率/%	误报率/%
7	51	91.2	4.1
10	38	94.5	2.1
15	26	80.0	1.5
20	24	75.1	1.7
25	20	64.6	1.2

从结果可以看出,通常情况下初始聚类中心个数 m和异常数据所占比例都固定时,随着聚类半径 R的减小,检测率逐渐增大,虽然误报率也有所上升,但总体控制在一个较低的水平。该入侵检测算法在检测率较高的情况下保持了相对低的误检率,有较好的检测效果。

再用经典的聚类算法 k-means聚类算法对相同数据集进行检测,比较结果。表 4、5分别是基于 k-means聚类的入侵检测方法对 DoS和 Probe攻击的检测实验结果。

表 4　k-means算法对 DoS攻击的检测结果

聚类个数	检测率/%	误报率/%
32	90.8	14.5
28	89.1	13.8
24	85.2	7.9
16	80.1	2.8
8	49.5	0

表 5　k-means算法对 Probe攻击的检测结果

聚类个数	检测率/%	误报率/%
32	87.9	13.1
28	78.2	9.8
24	81.3	8.8
16	68.8	3.5
8	50.1	1.3

从表 2、3与表 4、5的对比可以看出,本文提出的聚类算法在 KDD Cup 99数据集上的测试结果明显优于基于 k-means聚类的入侵检测方法。

(下转第 714页)

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

中的对象如下：

1)M = ((((((FS1[ ] FS2)[| aD1 |] D1(f1)) [| aD2 |] D2(filP))[| aD3 |] D3(inode))[| aD4 |] D4(locks))[| aD5 |] D5(blocks)；

2)U = FS1；V = FS2；A = D3(inode) ∪ D4(locks) ∪ D5(blocks)；B = D1(f1) ∪ D2(filP)；

3)R1(M′,U′,V′) = L(M)[α(M)] Com[α(M′)] L(M′) \ α(U′) \α(A′) \α(B′)。

将 M及 FS1、FS2、D1(f1)、D2(filP)、D3(inode)、D4(blocks)、D5(locks)、R1、COM1、Spec的 CSP描述输入 FDR2，执行验证断言 AssertSpec[F= R1 即可验证上述信道控制策略，结果如图 3所示。

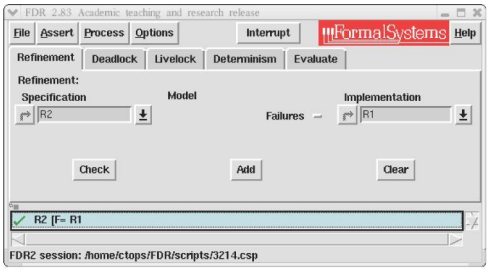

图 3 FDR2信道控制策略验证结果

显然，上述方案可以应用在复杂信息系统的信息流策略分析中。本文所定义的信道控制策略刻画了各个模块之间安全的信息交互的设计需求，而 2.2节给出的验证方案则可以用于验证系统设计能否满足根据上述安全需求所定义的信道控制策略。

4 结语

本文基于不干扰理论分析了系统中信息域之间的直接或间接的交互关系。而用于在两个不直接交互信息域之间传递数据的中间域就是信道。本文通过研究信道与那些向其输入信息或从其获得信息的信息域之间直接或间接的干扰关系来定义和描述信道。信息普遍存在于复杂系统中的任两个功能模块或进程之间，明确描述和严格控制系统模块和进程之间的信息通道，有利于最大限度地保障模块或进程的完整性和可控性。本文提出的信道控制策略正是用于基于上述目的信道描述。针对信道控制策略复杂而不便于手工验证的特点，本文提出了基于 CSP的系统和策略描述以及基于 FDR2的策略自动化验证方法。该方法能够在少量人工参与的情况下有效地分析信道控制策略，发现大部分存储隐蔽通道。

参考文献：

[1] GOGUEN J, MESEGUER J. Security policies and security models[C]// Proceedings of the 1982 IEEE Symposium on Research in Security and Privacy. Los Alamitos: IEEE Computer Society, 1982: 11-20.

[2] HAIGH J, YOUNG W. Extending the non-interference model of MLS for SAT[C]// Proceedings of the 1986 Symposium on Security and Privacy. Oakland, CA: IEEE Computer Society, 1986: 232-239.

[3] ROSCOE A W, WOODCOCK J C P, WULF L. Non-interference through determinism[J]. Journal of Computer Security, 1996, 4(1): 27-54.

[4] ROSCOE A W. CSP and determinism in security modeling[C]// Proceedings of the 1995 IEEE Symposium on Security and Privacy. Washington, DC: IEEE Computer Society, 1995: 114-127.

[5] Formal Systems (Europe) Ltd. FDR 2 user manual[EB/OL]. [2009-07-20]. http://www.fsel.com/documentation/fdr2/html/index.html.

[6] ROSCOE A W, GOLDSMITH M H. What is intransitive noninterference?[C]// Proceedings of the 12th Computer Security Foundations Workshop. Mordano, Italy: IEEE Computer Society, 1999: 228-238.

[7] RUSHBY J. Noninterference, transitivity, and channel-control security policies[R]. Menlo Park: Stanford Research Institute, 1992.

[8] HOARE C A R. Communicating sequential processes[J]. Communications of the ACM, 1978, 21(8): 666-677.

[9] ROSCOE A W. The theory and practice of concurrency[M]. Upper Saddle River, NJ: Prentice-Hall, 1997.

（上接第 701页）

5 结语

本文提出了一种基于聚类和时间序列模型的入侵检测算法，可以有效地将正常的数据和攻击数据区分开来，且具有很好的准确性。利用 KDD Cup 99数据集的实验表明，这一算法在具有较高的检测率的同时保持了相对较低的误警率，特别是对于拒绝服务攻击更是如此。

参考文献：

[1] 杨智君，田地，马骏骁，等. 入侵检测技术研究综述[J]. 计算机工程与设计，2006，27(12)：2119-2123.

[2] PORTNOY L, ESKIN E, STOLFO S J. Intrusion detection with unlabeled data using clustering[C]// DMSA 2001: Proceedings of 2001 ACM CSS Workshop on Data Mining Applied to Security. Philadelphia, PA: ACM Press, 2001: 5-8.

[3] LAZAREVIC A, ERTOZ L, KUMAR V, et al. A comparative study of anomaly detection schemes in network intrusion detection[C]// Proceedings of the 3rd SIAM Conference on Data Mining. New York: ACM Press, 2003: 801-813.

[4] 向继，高能，荆继武. 聚类算法在网络入侵检测中的应用[J]. 计算机工程，2003，29(16)：48-185.

[5] 罗敏，王丽娜，张焕国. 基于无监督聚类的入侵检测方法[J]. 电子学报，2003，31(1)：1713-1716.

[6] 李卫平. k-means聚类算法研究[J]. 中西部科技，2008，7(8)：52-53.

[7] 赵铁山，李增智，高波，等. 时间序列模型在入侵检测中的应用研究[J]. 计算机工程与设计，2005，26(5)：1128-1129.

[8] 陈铁梅，黄道平，陆顾新，等. 模式聚类在数据预处理中的应用研究[J]. 计算机与应用化学，2003，20(3)：241-243.

[9] University of California KDD Cup 1999 DATASETS[EB/OL]. [2009-04-20]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[10] LEE W. A data mining framework for building intrusion models[C]// Proceedings of the 1999 IEEE Symposium on Security and Privacy. Washington, DC: IEEE Computer Society, 1999: 120-132.

[11] HAN JIAWEI, KAMBER M. 数据挖掘：概念与技术[M]. 北京：机械工业出版社，2003.

[12] TENG SHAO-HUA, ZHANG WEI, ZHU ZHO-HUI, et al. DDoS attack detection and defense based on feature and data fusion[J]. System and Information Sciences Notes, 2007, 1(4): 390-395.

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net