计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
49(9): 1982-1990, 2012

基于 Shell 命令和共生矩阵的用户行为异常检测方法

李 超[1] 田新广[2] 肖 喜[3] 段淼毅[1,2]
[1](北京航空航天大学计算机学院 北京 100083)
[2](中国科学院计算技术研究所 北京 100190)
[3](中国科学院信息安全国家重点实验室 北京 100039)
(super_lcm@hotmail.com)

Anomaly Detection of User Behavior Based on Shell Commands and Co-Occurrence Matrix

Li Chao[1], Tian Xinguang[2], Xiao Xi[3], and Duan Miyi[1,2]
[1](*School of Computer Science and Engineering, Beihang University, Beijing* 100083)
[2](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)
[3](*State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing* 100039)

Abstract Anomaly detection of user behavior is now one of the major concerns of system security research. Anomaly detection systems establish the normal behavior profile of a subject (e.g. user), and compare the observed behavior of the subject with the profile and signal intrusions when the subject's observed behavior differs significantly from the profile. One problem with anomaly detection is that it is likely to raise many false alarms. Unusual but legitimate use may sometimes be considered anomalous. This paper proposes a novel method for anomaly detection of user behavior, which is applicable to host-based intrusion detection systems using shell commands as audit data. Considering the property and the uncertainty of user behavior, the method obtains an event sequence with less variety of events after hierarchically merging shell command tokens into sets and then profiles the user's normal behavior with a partly normalized co-occurrence matrix. In the detection stage, for event current sequence, a normalized co-occurrence matrix is constructed. Then the distances between these matrixes and the profile matrix are calculated according to the second matrix norm. Finally they are filtered with sliding windows and used to determine whether the monitored user's behavior is normal or anomalous. The experiment results on datasets of Purdue University and SEA show that the proposed method can achieve higher detection accuracy, require less memory and take shorter time than the other traditional methods.

Key words intrusion detection; anomaly detection; shell command; co-occurrence matrix; user behavior

摘 要 用户行为异常检测是当前网络安全领域研究的热点内容.提出一种新的基于共生矩阵的用户行为异常检测方法,主要用于 Unix 或 Linux 平台上以 shell 命令为审计数据的入侵检测系统.该方法在训练阶段充分考虑了用户行为复杂多变的特点和审计数据的时序相关属性,依据 shell 命令的出现频率并利用阶梯式的数据归并方法来确定事件,然后构建模型矩阵来刻画用户的正常行为.在检测阶段,首先为每一个当前事件序列构建一个部分正则化共生矩阵,然后根据矩阵 2 范数计算这些矩阵与模型矩阵的距离,得到距离流,最后通过平滑滤噪处理距离流来判决用户行为.在 Purdue 大学实验数据和

收稿日期:2011-07-07;修回日期:2011-12-30

SEA 实验数据上的两组实验结果表明,该方法具有很高的检测性能,其可操作性也优于同类方法.

关键词　入侵检测;异常检测;shell 命令;共生矩阵;用户行为

中图法分类号　TP393.08

目前,入侵检测技术主要有两种类型,分别是误用检测和异常检测. 误用检测通过监视目标系统的特定行为与已知的入侵模式是否匹配来检测入侵行为;而异常检测则是事先建立被监视目标在正常情况下的行为模式,通过检测当前行为是否显著偏离了相应的正常模式来进行入侵检测. 异常检测不需要过多专业知识就能够检测出未知的攻击类型,并且有较强的适应性[1]. 近年来,用户行为异常检测作为入侵检测的一个重要分支得到了广泛研究,并在网络安全工程中发挥着越来越大的作用.

用户行为异常检测面临的主要困难是用户行为具有多变性和复杂性,即用户行为会随着工作内容、用户兴趣、工作时间和其他不确定性因素的变化而变化[2-3]. 其具体实现过程一般分为训练和检测两个阶段. 训练阶段主要是在用户界面层建立系统中一个(或一组)合法用户的正常行为轮廓. 检测阶段则通过比较该合法用户的当前行为和此正常行为轮廓来识别异常行为:如果该合法用户的当前行为较大程度地偏离了其历史上的正常行为轮廓,则认为发生了异常,这种异常可能是该合法用户本身进行了非授权操作,也可能是系统中其他合法用户或外部入侵者(非法用户)冒充该合法用户进行了非法操作.

用户行为异常检测的审计数据一般具有 3 种属性:转移属性、频率属性和相关属性. 现有的工作大部分是基于转移属性,如文献[2,4-8],或频率属性,如文献[9-14]. 相关属性被前人工作所忽略,它是指用户的动态行为不仅可以通过相邻的事件来刻画,也可以通过不相邻的事件来刻画. Oka 等人[3]考虑到 shell 命令之间的相关属性,提出了特征共生矩阵(eigen co-occurrence matrix, ECM)的检测方法. 他们通过构建共生矩阵并利用主成分分析(principal component analysis, PCA)方法提取重要的特征,来建立分层网络以对用户行为进行分析. 因为他们把不同的 shell 命令当作不同的事件,这样不可免避地产生了高维矩阵,需要消耗很大的存储空间和计算成本.

考虑到用户行为的复杂多变性和审计数据 shell 命令序列的相关属性,本文提出一种新的用户行为异常检测方法. 该方法在训练阶段首先通过阶梯式的数据归并来划分事件,再在此基础上构建一个部分正则化的共生矩阵作为模型矩阵对合法用户的正常行为建模. 在检测阶段,为每个当前事件序列创建一个部分正则化的共生矩阵,然后基于矩阵 2 范数计算此矩阵与模型矩阵之间的距离,形成一个距离流. 最后对此距离流加窗均值滤噪处理,得到判决值来识别用户行为中的异常. 我们在两组用户行为异常检测的标准实验数据(Purdue 大学的实验数据和 SEA 实验数据)上对本文方法进行了性能测试. 结果表明,同 4 种其他相关方法相比,本文方法在减少存储成本和计算成本的同时,提高了检测准确率,改善了系统的整体性能,具有较强的实用性和可操作性,特别适用于在线检测.

1　相关工作

基于 shell 命令的用户行为异常检测在最近 10 年受到了较多研究者的关注. Lane 等人[2,4]开展了基于实例学习和隐 Markov 模型(hidden Markov model, HMM)的两种用户行为异常检测方法的研究. 基于实例学习的方法用特定的相似度函数刻画当前行为与正常行为模式之间的相似性,原理较为简单,有较强的适应能力,但在检测阶段没有考虑行为模式在训练数据中的出现频率和不同行为模式之间的相关性,因此检测准确率较低. 基于 HMM 的方法虽然准确率高,但是训练和工作中所需要的计算量比较大,检测效率较低. 孙宏伟等人[5]在基于实例学习方法的基础上改进了对用户行为模式的表示方式,以 shell 命令为单位进行相似度赋值,改善了检测性能. Schonlau 等人[15]研究了基于统计理论的异常检测方法,综合分析了 6 种不同方法的优势和局限性. Maxion 等人[9]对 Schonlau 的方法进行了改进,假设用户操作命令以固定的频率出现,与它前面的命令无关,由此引入了贝叶斯分类算法,提高了检测准确率. 最近,Dash 等人[12]提出延迟检测概念(deferred detection concept),运用适应性朴素贝叶斯方法进行用户行为异常检测,使检测性能得到进一步提高. Tian 等人[8]提出了基于 Markov 链模型

的检测方法,有良好的检测性能,但存在状态数目过多、计算复杂度大及泛化能力不强等缺点.

此外,Szymanski 等人[11]和 Wu 等人[16]研究了数据挖掘的方法;Kim 等人[17]提出了基于常见命令和投票引擎(common commands and voting engine)的支持向量机方法,这些方法具有较高的检测效率,但对用户行为变化的适应性不强. Wang 等人[13]研究了非负正定矩阵分解算法在检测中的应用;Coull 等人[7]把生物信息学里的序列比对(sequence alignment)算法应用于用户行为异常检测;Wang 等人[10]提出两种单分类训练方法(one-class training),达到了与多分类方法相同的检测效果.

2 共生矩阵

在用户行为异常检测中,审计数据的事件序列没有完备的语法规则,但事件序列里的两个事件之间有前后相关联系. 共生矩阵能够提取隐藏在事件序列里的相关属性. 共生矩阵通过关联一个事件和其后一定间隔距离内的事件来对事件序列进行建模,根据两个相关事件之间的距离和它们的出现频率来定义事件之间的关联强度. 即两个事件的距离越近或者它们的出现频率越大,则它们的关联强度越大. 为简单起见,我们只考虑在一定间隔距离内两个事件的出现频率,通过统计每个事件对在一定间隔距离内的出现次数来构建共生矩阵. 这样共生矩阵记录了相邻事件和不相邻事件的特征,它能够适应用户行为的复杂多变性[3].

设事件流(事件序列)为 $\boldsymbol{v}=(v_1,v_2,\cdots,v_r)$,事件集合为$\{1,2,\cdots,N\}$,间隔距离为 k. 基于事件流 $\boldsymbol{v}=(v_1,v_2,\cdots,v_r)$产生共生矩阵 $\boldsymbol{P}=(p_{ij})_{N\times N}$的算法(C 语言伪代码)如图 1 所示:

```
for(i=1;i≤N;i++)
  for(j=1;j≤N;j++)
    p_ij=0;
for(i=1;i≤r-k;i++)
    { a=v_i;
      for(j=1;j≤k;j++)
        { b=v_i+j;
          p_ab=p_ab+1;  }    }
```

Fig. 1 The generation algorithm of co-occurrence matrix.

图 1 共生矩阵产生算法

上述算法的核心思想是在事件流(事件序列)$\boldsymbol{v}=(v_1,v_2,\cdots,v_r)$中,通过滑动窗的方式统计在间隔距离为 k 的范围内两个事件共同出现的频率,进而得出共生矩阵 $\boldsymbol{P}=(p_{ij})_{N\times N}$,在计算过程中滑动窗的长度为 k,每次滑动的步进为 1. 共生矩阵反映了事件集合$\{1,2,\cdots,N\}$中各个事件之间的关联强度,事件的数量决定了共生矩阵的维数. 在实际的用户行为异常检测中,合理确定事件的数量对提高检测准确度、降低数据存储量具有重要意义.

目前,同类检测方法主要采用行为模式匹配和转移概率计算的方式来识别用户行为中的异常,行为模式匹配的结果和转移概率的大小反映了用户当前行为同历史行为轮廓的偏移程度. 这些方法的主要区别是行为模式的定义、模式库的构建和匹配存在差异,或者是转移概率的计算和平滑滤噪方面有所不同,但本质上在用户行为建模方面主要考虑了两类信息,一是行为模式的出现频率,二是行为模式之间的转移情况和相互距离. 本文方法的特点是基于共生矩阵来刻画用户正常行为轮廓,而且在构建共生矩阵时依据 shell 命令的出现频率并利用阶梯式的数据归并方法来确定事件,该方法在行为建模中不仅考虑了上述两类信息,而且还考虑了事件在一定间隔距离内的关联(共生强度),刻画了相邻事件和不相邻事件的特征,因而在检测准确度上有较大程度的提高.

3 训 练

在训练阶段,本文方法利用共生矩阵对用户正常行为进行建模. Oka 等人[3]把不同的 shell 命令当作不同的事件,存在事件数目过多的问题. 他们利用 PCA 方法来提取重要的特征. 本文采用阶梯式数据归并的方法来定义事件,大幅减少了事件数目. 本文方法训练阶段的具体过程如下:

1) 获取该合法用户的正常行为的训练数据

设正常行为训练数据为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_r)$,它是一个长度为 r 的 shell 命令流,其中 x_j 表示按时间顺序排列的第 j 个 shell 命令符号.

2) 定义事件

我们预先设定不同的事件个数为 N,事件集合为$\{1,2,\cdots,N\}$,基于 shell 命令流 $\boldsymbol{x}=(x_1,x_2,\cdots,x_r)$定义事件的主要步骤如下:

步骤 1. 提取出 shell 命令流 x 中互不相同的 shell 命令符号并将它们按出现频率降序排列.

设 x 中互不相同的 shell 命令符号共有 W 个$(W\leqslant r)$,shell 命令符号 $x_{\#}$ # 在 shell 命令流 x 中

出现的次数为 $C_\sharp$,则 $x_\sharp$ 在 x 中出现的频率 $F_\sharp$ 定义为

$$F_\sharp = C_\sharp / r,\ 1 \leqslant i \leqslant W, \tag{1}$$

将这 W 个互不相同的 shell 命令符号按其在 x 中出现的频率从大到小排序. 排序后记为 $x_1^*, x_2^*, \cdots, x_W^*$,设 F_j^* 为 x_j^* 在 x 中出现的频率($1 \leqslant j \leqslant W$),则有 $F_1^* \geqslant F_2^* \geqslant \cdots \geqslant F_W^*$.

步骤 2. 根据 shell 命令符号的出现频率进行阶梯式数据归并.

把排序后的 shell 命令符号按频率从大到小阶梯式归并成 $N-1$ 个集合. 设 b 是不大于 $W/(N-1)$ 的最大整数,$c=b+1$,$h=W-(N-1)b$,排序后的前 hc 个符号按 c 个一组归并成 1 个集合,剩下的符号按 b 个一组归并成 1 个集合,即第 1 个集合 $\Lambda_1=\{x_1^*,\cdots,x_c^*\},\cdots$;第 h 个集合 $\Lambda_h=\{x_{(h-1)c+1}^*,\cdots,x_{hc}^*\}$;第 $h+1$ 个集合 $\Lambda_{h+1}=\{x_{hc+1}^*,\cdots,x_{hc+b}^*\},\cdots$;第 $N-1$ 个集合 $\Lambda_{N-1}=\{x_{hc+(N-h-2)b+1}^*,\cdots,x_W^*\}$($W=hc+(N-h-1)b$).

步骤 3. 定义 shell 命令符号 $x_\sharp$ 的事件 $v_\sharp$. 如果存在 $1 \leqslant i \leqslant N-1$,使 $x_\sharp \in \Lambda_i$,则 $v_\sharp = i$;否则,$v_\sharp = N$. 在实际操作中为节约时间,可利用频率优先匹配方法确定 $x_\sharp$ 的事件 $v_\sharp$:依次在集合 Λ_1,Λ_2,$\Lambda_3\cdots$中查找 $x_\sharp$. 如果在第 $i(1 \leqslant i \leqslant N-1)$ 个集合 Λ_i 中查找到 $x_\sharp$,则 $v_\sharp = i$;如果在所有 $N-1$ 个集合中都查找不到 $x_\sharp$,则 $v_\sharp = N$.

通过以上 3 个步骤我们可以定义训练数据 $\boldsymbol{x}=(x_1,x_2,\cdots,x_r)$ 中每一个 shell 命令符号的事件,把这些事件按时间先后顺序排列,得到训练数据的事件流 $\boldsymbol{v}=(v_1,v_2,\cdots,v_r)$.

3) 基于事件流构造部分正则化共生矩阵

我们基于事件流 $\boldsymbol{v}=(v_1,v_2,\cdots,v_r)$ 利用第 2 节共生矩阵产生的算法,可以得到共生矩阵 $\boldsymbol{P}=(p_{ij})_{N\times N}$,然后对共生矩阵 $\boldsymbol{P}=(p_{ij})_{N\times N}$ 的子矩阵 $(p_{ij})_{(N-1)\times(N-1)}$ 的每行进行归一化操作,算法如图 2 所示:

```
for(i=1;i<N;i++)
  {  s=p_i1+p_i2+…+p_i(N-1);
     if(s!=0)
     {  for(j=1;j<N;j++)
           p_ij=p_ij/s;   } }
```

Fig. 2　Normalization algorithm.

图 2　归一化算法

经过以上操作我们可以得到 $\boldsymbol{v}=(v_1,v_2,\cdots,v_r)$ 的部分正则化共生矩阵 $\boldsymbol{P}=(p_{ij})_{N\times N}$,至此对合法用户的正常行为建模过程完成. 我们称由训练数据产生的部分正则化共生矩阵 $\boldsymbol{P}=(p_{ij})_{N\times N}$ 为模型矩阵.

4　检　　测

在检测阶段,我们为每一个当前事件序列构造一个当前部分正则化共生矩阵,通过矩阵的第 2 范数计算当前部分正则化共生矩阵与模型矩阵的距离,获取距离流,再对距离流平滑滤噪得到判决值来判决用户行为. 具体步骤如下:

1) 得到被监测用户在被监测的时间内执行的 shell 命令数据 $\boldsymbol{c}=(c_1,c_2,\cdots,c_r')$,它是一个长度为 r' 的 shell 命令流.

2) 利用第 3 节 2)的方法基于 shell 命令流 $\boldsymbol{x}=(x_1,x_2,\cdots,x_r)$ 为 shell 命令流 $c=(c_1,c_2,\cdots,c_r')$ 的每一个 shell 命令符号定义一个事件,得到事件流 $\boldsymbol{u}=(u_1,u_2,\cdots,u_r')$.

3) 构造当前部分正则化共生矩阵并获取距离流.

设定当前事件序列的长度为 h. 我们先利用第 3 节 3)的方法为每一个当前事件序列 $\bar{u}_i=(u_i,u_{i+1},\cdots,u_{i+h-1})$,构建一个部分正则化共生矩阵 $\boldsymbol{P}_i$($i=1,2,\cdots,r'-h+1$)(符号说明:$\bar{y}_i=(y_i,y_{i+1},\cdots,y_{i+h-1})$ 是字符流 $y=(y_1,y_2,\cdots,y_m)$ 的第 i 个(长度为 h)的序列)然后用式(2)计算当前矩阵 $\boldsymbol{P}_i$ 与模型矩阵 $\boldsymbol{P}$ 之间的距离:

$$dis(\boldsymbol{P}_i) = \|\boldsymbol{P}_i - \boldsymbol{P}\|_2. \tag{2}$$

设矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$,$\|\boldsymbol{A}\|_2$ 指 $\boldsymbol{A}$ 的最大奇异值,即:

$$\|\boldsymbol{A}\|_2 = \sqrt{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}. \tag{3}$$

通过计算每个当前部分正则化共生矩阵 $\boldsymbol{P}_i$ 与模型矩阵 $\boldsymbol{P}$ 的距离($i=1,2,\cdots,r'-h+1$),我们得到距离流($dis(\boldsymbol{P}_1),dis(\boldsymbol{P}_2),\cdots,dis(\boldsymbol{P}_{r'-h+1})$).

4) 平滑滤噪处理距离流计算判决值.

因为用户在短时间内的行为可能会偏离其历史行为,我们不直接利用距离对用户行为进行判决,而是先对距离流($dis(\boldsymbol{P}_1),dis(\boldsymbol{P}_2),\cdots,dis(\boldsymbol{P}_{r'-h+1})$)进行加窗平滑滤噪处理来计算判决值,公式如下:

$$D(n) = \frac{1}{w}\sum_{m=n-w+1}^{n} dis(\boldsymbol{P}_m), \tag{4}$$

其中,$D(n)$ 是矩阵 $\boldsymbol{P}_n$($w \leqslant n \leqslant r'-h+1$)对应的判决值,$w$ 为窗长度,n 的增长步长为 1. shell 命令流

$\boldsymbol{c}=(c_1,c_2,\cdots,c'_r)$的第 w 个 shell 命令序列 $\bar{c}_w=(c_w,c_{w+1},\cdots,c_{w+h-1})$及其后面的每个 shell 命令序列都分别对应一个判决值.

5) 根据判决值和预先设定的判决门限对用户行为进行判决.

设判决门限为 d,根据判决值 $D(n)$和判决门限 d 对被监测用户的"当前行为"作出判决. 判决方法为:如果 $D(n)<d$,将被监测用户的"当前行为"判为正常行为;否则,将其判为异常行为. 这里,"当前行为"是相对于矩阵 $\boldsymbol{P}_n$,也即相对于 shell 命令序列 $\bar{c}_n$ 而言的,它是指被监测用户执行的以 shell 命令序列 $\bar{c}_n$ 为终点的 w 个 shell 命令序列 $\bar{c}_{n-w+1}$, $\bar{c}_{n-w+2},\cdots,\bar{c}_n$,也是指被监测用户执行的以 shell 命令符号 c_{n+h-1}为终点的 $w+h-1$ 个 shell 命令符号 $c_{n-w+1},c_{n-w+2},\cdots,c_{n+h-1}$.

需要指出,在在线检测的情况下,被监测用户所执行的 shell 命令数据的获取、事件流的得到和距离流的获得、判决值的计算以及对用户行为的判决都是同步进行的. 当被监测用户执行完 $\boldsymbol{c}=(c_1,c_2,\cdots,c'_r)$中的第 $w+h-1$ 个 shell 命令之后,该用户每再执行完一个 shell 命令,检测系统就可以以此 shell 命令为终点截取一个新的长度为 $w+h-1$ 的 shell 命令序列,同时计算相应的判决值,进而对该用户的"当前行为"作出一次判决.

5 实验结果与分析

5.1 实验数据

与文献[2,4-9,11-12,15,17-18]中的用户行为异常检测实验相同,本文的实验采用 Unix 平台上的 shell 命令作为审计数据. 主要是考虑到:1) shell 命令容易收集,形式简单,便于分析;2) 在 Unix 平台上,shell 是终端用户与操作系统之间最主要的界面,能反映用户的行为,且大部分用户活动都是利用 shell 完成的.

国际上通用的用户行为异常检测实验数据主要有两组:Purdue 大学实验数据和 AT&T Shannon 实验室实验数据(简称 SEA 实验数据). 本文方法在训练和检测阶段处理的数据都是对用户的原始 shell 命令数据进行预处理后的数据. Purdue 大学实验数据预处理时滤除 shell 命令中的主机名、网址等信息,保留 shell 命令的名称及参数;各命令符号按照在 shell 会话中的出现次序进行排列,不同的 shell 会话按照时间顺序进行连接,每个会话开始和结束的时间点上插入了标识符号(如文献[2,4,6,8]所述). 例如,图 3 描述了某用户的一个 shell 会话数据:

```
>cd~/private/docs
>ls-laF|more
>cat foot. txt ball. txt sjb. txt>~/xiao/amusement
>mailx xiaoxi_ac@163. com
>exit
```

Fig. 3 Example of user shell session.

图 3 用户 shell 会话数据实例

经预处理后成为如下 shell 命令序列:(*SOF*, cd,〈1〉, ls,-laF, |, more, cat,〈3〉,>,〈1〉, mailx,〈1〉, exit, *EOF*),其中 *SOF* 和 *EOF* 分别是会话开始和结束的标识符号,〈1〉,〈3〉为目录名(地址)符号. 相对于前面的预处理方式,AT&T Shannon 实验室实验数据所采用的预处理方式较为简洁:只保留 shell 命令的名称,滤除命令参数和时间等信息(如文献[9,15]所述). 经过不同方式的预处理后的原始 shell 命令数据表现形式都是 shell 命令流(按时序排列的若干个 shell 命令符号).

5.2 实验设计

本文作者分别利用以上两组用户行为异常检测实验数据对本文方法的性能进行了两组实验. 在每组实验中,我们都只利用其中 *user*1, *user*2, *user*3 和 *user*4 的实验数据(shell 命令流),都把 *user*3 设为合法用户,*user*1, *user*2, *user*4 设为非法用户. 两组实验的参数设置一样,都设为 $N=4,k=3,h=9,w=91$.

5.2.1 在 Purdue 大学实验数据上的实验

Purdue 大学实验数据预处理后每个用户的实验数据中有 15 000 个 shell 命令. *user*3 的前 10 000 个命令作为正常行为训练数据用于正常行为的模型矩阵的建立,后 5 000 个命令作为正常行为测试数据用于检测性能(主要是虚警概率)的测试. *user*1, *user*2, *user*4 每个用户的实验数据的后面 5 000 个 shell 命令均作为异常行为测试数据用于检测概率的测试. 实验时,正常行为训练数据中互不相同的 shell 命令符号共有 200 个,正常行为的模型矩阵为

$$\boldsymbol{P}=\begin{pmatrix}0.218 & 0.202 & 0.580 & 0\\ 0.170 & 0.332 & 0.498 & 0\\ 0.192 & 0.171 & 0.637 & 0\\ 0 & 0 & 0 & 0\end{pmatrix}.$$

5.2.1.1 检测准确度分析

图 4 示出了由式(4)计算出的判决值曲线. 图 4

中下方的实线是合法用户 *user*3 的测试数据对应的判决值曲线,上方的 3 条虚线分别是非法用户 *user*4,*user*1,*user*2 的测试数据对应的判决值曲线.可见,图 4 中的判决值曲线具有很好的可分性.

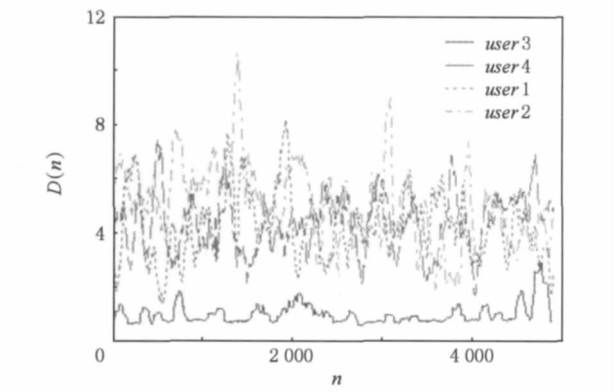

Fig. 4　Output curves of our method.

图 4　本文方法输出的判决值曲线

ROC 曲线是判断 IDS 检测准确度的通用标准,描述了虚警概率与检测概率之间变化关系.为进一步分析检测准确度,图 5 示出了本文方法和文献[3-5,8]方法的 ROC 曲线.图 5 中各种方法的参数是在保证平均检测时间基本相同前提下设置的.在虚警概率为 0%时,本文方法检测概率超过了 90%,而其他 4 种方法的检测概率都小于 73%,因此本文方法比已有文献中的 4 种方法的检测准确度均有大幅度的提高(在虚警概率为 0%时,检测概率至少提高了 17%).

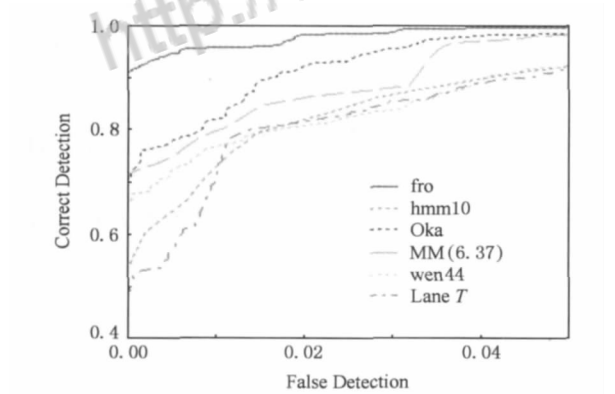

Fig. 5　ROC curves of 5 methods.

图 5　5 种方法的 ROC 曲线

5.2.1.2　事件个数和当前事件序列长度对检测性能的影响

图 6 示出了本文方法的事件个数 N 变化时的 ROC 曲线,其中 N 分别为 2,3,4,5,9,17,33,65.从图 6 可以看出,当 $N=2$ 时检测准确度最好;而 $N=65$ 时检测准确度最差.$N=9$ 和 $N=17$ 的 ROC 曲线基本重合.因为本文方法在 $N=2$ 时检测准确度最好,而由图 2 可知,当 $N=4$ 时,本文方法检测准确度已超过其他 4 种方法,所以本文方法取事件个数 $N=2$ 时,检测准确度会大大超过其他 4 种方法.

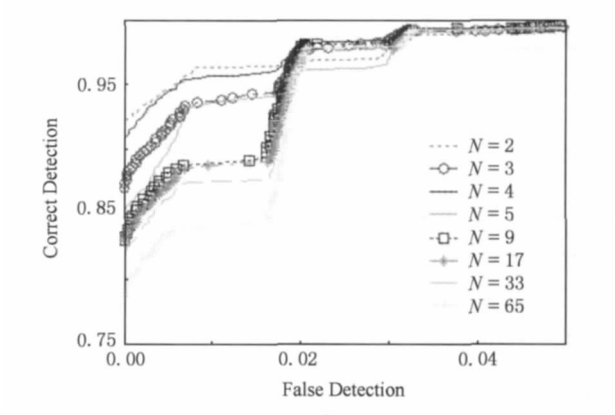

Fig. 6　ROC curves of various N.

图 6　不同的 N 对应的 ROC 曲线

图 7 示出了本文方法的当前事件序列长度 h 变化时的 ROC 曲线,其中 h 的取值为 7,8,9,10,14,22,38,70.从图 7 可以看出,当 $h=8$ 时检测准确度最好;而 $h=7$ 时检测准确度最差.$h=9$ 和 $h=10$ 的 ROC 曲线基本重合.在虚警概率较小的情况下(接近 0%时)呈现出一个总体趋势:检测准确度随当前事件序列长度 h 的增加得到了提高.在以上检测准确度最差的情况下,即 $h=7$ 时本文方法在虚警概率为 0%时检测概率大于 84%,而从图 5 可知其他 4 种方法在相同虚警概率下的检测概率都小于 73%,因而此时本文方法检测准确度也优于其他 4 种方法.考虑到存储空间和计算复杂度及方法的普遍性,本文实验中取 $N=4$,$h=9$.

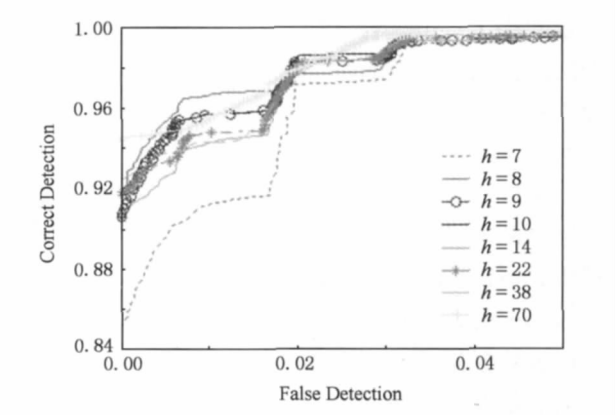

Fig. 7　ROC curves of various h.

图 7　不同的 h 对应的 ROC 曲线

5.2.1.3　存储空间和实验时间分析

文献[4-5]分别提出了两种不同的机器学习方法,文献[8]提出了基于 Markov 链模型的方法,文

献[3]提出了 ECM 方法. 表 1 列出了本文方法和以上 4 种方法的存储空间和实验时间(相同条件下测量):

Table 1 Experimental Results of 5 Methods

表 1 5 种方法的实验结果

Parameters	Method				Our Method
	In Ref[4]	In Ref[5]	In Ref[3]	In Ref[8]	
Memory Unit	2 544	3 512	29 148 336	40 601	216
Experimental Time/s	1 818.6	29.192	1 759.2	19.208	37.656

存储空间:从表 1 可看出,本文方法存储空间是所有方法中最少的,是文献[4]方法的 216/2 544 = 8.49%,减少了 1 个数量级;是文献[8]方法的 216/40 601 = 0.53%,减少了 3 个数量级. 从理论上讲,文献[4-5]的方法需要存储合法用户的行为模式库,行为模式库的数据量会随着用户行为的变化和训练数据的增加而增加;文献[8]的方法不仅要存储 shell 命令模式库,还要存储齐次 Markov 链的状态转移概率矩阵,而本文的方法主要通过构建模型矩阵来刻画用户的正常行为,模型矩阵的数据量不会因为训练数据的增加(用户行为的变化)而增加,因而在存储空间方面具有优势.

实验时间:实验时间是指实验中进行训练和检测所需要的时间,它与检测方法的计算成本成正比,并在一定程度上反映了检测的实时性. 本文方法实验时间远少于文献[3]和文献[4]方法的实验时间,是文献[4]方法的 37.656/1 818.6 = 2.07%,减少了 2 个数量级.

本文方法的检测准确度高于以上 4 种方法,存储成本小于以上 4 种方法,计算成本低于文献[3-4]方法. 综合这 3 方面,本文方法的整体性能优于已有的 4 种方法.

5.2.2 在 SEA 实验数据上的实验

AT&T Shannon 实验室实验数据预处理后每个用户的实验数据有 15 000 个 shell 命令,我们只利用 *user*1, *user*2, *user*3, *user*4 的前 5 000 个 shell 命令. *user*3 的 5 000 个 shell 命令的前 4 000 个命令作为训练数据用于正常行为建模,后 1 000 个命令作为测试数据用于测试虚警概率; *user*4, *user*1, *user*2 的 5 000 个 shell 命令均作为测试数据用于测试检测概率.

图 8 示出了由式(4)计算出的判决值曲线. 可见,图 8 中的判决值曲线具有很好的可分性. 图 9 示出了本文方法和文献[3-5,8]方法的 ROC 曲线. 由图 9 可知,本文方法在 AT&T Shannon 实验室实验数据上也比其他 4 种方法在检测准确度方面有大幅度的提高.

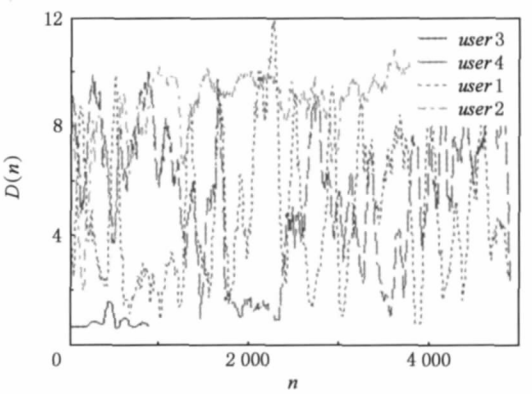

Fig. 8 Output curves of our method on the dataset of AT&T Shannon Lab.

图 8 本文方法在 AT&T Shannon 实验室数据上输出的判决值曲线

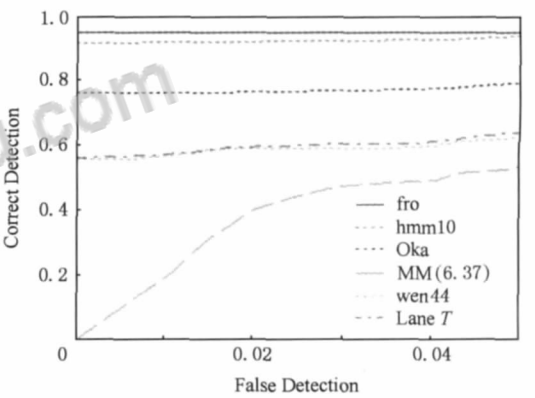

Fig. 9 ROC curves of 5 methods on the dataset of AT&T Shannon Lab.

图 9 5 种方法在 AT&T Shannon 实验室数据上的 ROC 曲线

6 结束语

本文提出一种高效的基于 shell 命令和共生矩阵的用户行为异常检测方法. 该方法在训练阶段,通过阶梯式数据归并来确定事件,利用部分正则化共生矩阵对用户正常行为进行建模;在检测阶段,通过计算当前事件序列的部分正则化共生矩阵与模型矩阵的第 2 范数得到距离流,然后对距离流加窗滤噪处理来判决用户行为. 实验表明,同已有的 4 种典型检测方法相比,本文基于矩阵第 2 范数的判决准则方法在减少存储成本和计算成本的同时,提高了检

测准确率,改善了系统的整体性能,具有很强的实用性和可操作性,特别适用于在线检测.此外,shell 命令和网络流量数据都属时间离散序列数据,共生矩阵方法应当也适用于网络流量异常检测,这是作者下一步要研究的内容.由于网络流量远大于主机(服务器)shell 命令数据流量,而且网络流量异常检测对计算效率的要求要高得多,因此共生矩阵在流量异常检测中的具体应用方式还有待进一步研究.

参　考　文　献

[1] Tian Xinguang, Duan Miyi, Cheng Xueqi. Masquerade detection based on shell commands and multiple behavior pattern mining [J]. Chinese Journal of Computers, 2010, 33(4): 697-705 (in Chinese)
(田新广,段洣毅,程学旗. 基于 shell 命令和多重行为模式挖掘的用户伪装攻击检测[J]. 计算机学报, 2010, 33(4): 697-705)

[2] Lane T. Machine learning techniques for the computer security domain of anomaly detection [D]. West Lafayette, Indiana: Purdue University, 2000

[3] Oka M, Oyama Y, Abe H, et al. Anomaly detection using layered networks based on eigen co-occurrence matrix [G] // LNCS 3224. Berlin: Springer, 2004: 223-237

[4] Lane T, Brodley C E. An empirical study of two approaches to sequence learning for anomaly detection [J]. Machine Learning, 2003, 51(1): 73-107

[5] Sun Hongwei, Tian Xinguang, Li Xuechun, et al. An improved anomaly detection model for IDS [J]. Chinese Journal of Computers, 2003, 26(11): 1450-1455 (in Chinese)
(孙宏伟,田新广,李学春,等. 一种改进的 IDS 异常检测模型[J]. 计算机学报, 2003, 26(11): 1450-1455)

[6] Tian Xinguang, Gao Lizhi, Sun Chunlai, et al. A method for anomaly detection of user behaviors based on machine learning [J]. The Journal of China Universities of Post and Telecommunications, 2006, 13(2): 61-65,78

[7] Coull S E, Branch J W, Szymanski B K, et al. Sequence alignment for masquerade detection [J]. Computational Statistics & Data Analysis, 2008, 52(8): 4116-4131

[8] Tian Xinguang, Duan Miyi, Li Wenfa, et al. Anomaly detection of user behavior based on shell commands and homogeneous Markov chains [J]. Chinese Journal of Electronics, 2008, 17(2): 231-236

[9] Maxion R A, Townsend T N. Masquerade detection using truncated command lines [C] //Proc of the Int Conf on Dependable Systems and Networks(DSN-02). Los Alamitos, CA: IEEE Computer Society, 2002: 219-228

[10] Wang K, Stolfo S J. One class training for masquerade detection [C] //Proc of the 3rd IEEE Conf Data Mining Workshop on Data Mining for Computer Security. Piscataway, NJ: IEEE, 2003: 1-10

[11] Szymanski B K, Zhang Y Q. Recursive data mining for masquerade detection and author identification [C] //Proc of the 5th IEEE System, Man and Cybernetics Information Assurance Workshop. Piscataway, NJ: IEEE, 2004: 424-431

[12] Dash S K, Reddy K S, Pujari A K. Adaptive naive Bayes method for masquerade detection [J]. Security and Communications Networks, 2011, 4(4): 410-417

[13] Wang W, Guan X, Zhang X. Profiling program and user behaviors for anomaly intrusion detection based on non-negative matrix factorization [C] //Proc of the 43rd IEEE Conf on Decision and Control (CDC'04). Piscataway, NJ: IEEE, 2004: 99-104

[14] Wan M D, Wu H C, Kuo Y W, et al. Detecting masqueraders using high frequency commands as signatures [C] //Proc of the Int Symp on Frontiers in Networking with Applications (FINA). Los Alamitos, CA: IEEE Computer Society, 2008: 596-601

[15] Schonlau M, DuMouchel W, Ju W H, et al. Computer intrusion: Detecting masquerades [J]. Statistical Science, 2001, 16(1): 58-74

[16] Wu H C, Huang S H S. Masquerade detection using command prediction and association rules mining [C] //Proc of the 2009 Int Conf on Advanced Information Networking and Applications. Los Alamitos, CA: IEEE Computer Society, 2009: 552-559

[17] Kim H S, Cha S D. Empirical evaluation of SVM-based masquerade detection using UNIX command [J]. Computers & Security, 2005, 24(2): 160-168

[18] Shim C Y, Kim J Y, Gantenbein R E. Practical user identification for masquerade detection [C] //Proc of the World Congress on Engineering and Computer Science. Los Alamitos, CA: IEEE Computer Secity, 2008: 47-51

Li Chao, born in 1976. PhD candidate of Beihang University. His current research interests include intrusion detection, anonymous communication and network security.

Tian Xinguang, born in 1976. PhD and associate professor. Post-doctoral fellow at the Institute of Computing Technology, Chinese Academy of Sciences. Member of China Computer Federation (CCF). His current research interests include intrusion detection, network security, risk evaluation, and signal processing.

Xiao Xi, born in 1979. PhD candidate at the State Key Laboratory of Information Security, Chinese Academy of Sciences. Member of China Computer Federation. His current research interests include information security and intrusion detection.

Duan Miyi, born in 1953. Professor and PhD supervisor. Senior member of China Computer Federation. His main research interests include computer network and information security.

阅读此文的还阅读了：

1. 一种基于选择性协同学习的网络用户异常行为检测方法
2. 谈实验教学安全管理命令方法
3. VBS的妙用
4. 一种基于 ASM 和灰度共生矩阵的光头定位检测方法
5. 基于Shell命令和共生矩阵的用户行为异常检测方法
6. 基于色彩共生矩阵的CBIR方法
7. 一种新的基于Markov链模型的用户行为异常检测方法
8. 留意你的DNS:从一些DNS/AD故障案例中吸取教训
9. 计算机使用技巧:doskey宏的使用
10. 移动视频监控及异常场景检测实现
11. 德国研究人员开发出眼控电脑软件系统
12. 动词第二人称命令式的用法小结
13. 基于shen命令和Markov链模型的用户行为异常检测
14. Biliary reflux detection in anomalous union of the pancreatico- biliary duct patients
15. 基于支持向量机的Web用户行为异常检测方法
16. A data driven approach for detection and isolation of anomalies in a group of UAVs

17. 判断文件夹的内容大小

18. 解决用户“痛点” 实现“弯道超车”——以钱江晚报的微信矩阵移动策略为例

19. 基于模糊类别共生矩阵的纹理疵点检测方法

20. 一键搞定Word中的网页文档

21. 基于多示例学习的异常行为检测方法

22. 基于shell命令和多重行为模式挖掘的用户伪装攻击检测

23. 基于模式挖掘的用户行为异常检测算法

24. 基于异常的入侵检测方法分析

25. 弹性波检测技术在水利工程施工中的应用

26. 基于shell命令和Markov链模型的用户行为异常检测

27. 基于隐马尔科夫模型的用户行为异常检测方法

28. 基于行为模型的IP Forwarding异常检测方法

29. 基于多示例学习的异常行为检测方法

30. 改变自己和改变世界

31. 在Windows Vista中，如何能够找到某个用户具有访问权限的所有文件？

32. 奶牛的发情鉴定及异常发情表现

33. 英国牛津大学检测异常系统行为项目（2013—105-英国-064）

34. 基于kNN算法的异常行为检测方法研究

35. 基于异常的入侵检测方法分析

36. 用户行为异常检测模型

37. 质监部门执法维权小集锦

38. 基于用户浏览行为的HTTP Flood检测方法

39. 要搜索就要彻底

40. 基于灰度-梯度共生矩阵的大米加工精度的机器视觉检测方法

41. 快速浏览功能键的用途

42. 基于灰度共生矩阵纹理特征的SAR影像变化检测方法研究

43. 孩子喝饮料过多易患多动症（1）

44. 是共生而不只是共赢

45. 摊铺机:专家之眼用户之见:采纳专家之高见——中国摊铺机发展现状

46. 基于视觉单词共生矩阵的图像分类方法

47. 儿童异常行为与心理治疗的探讨

48. Homeland Security Research:截至2016年,生物识别监控及行为异常探测市场总值将达到32亿美元

49. 基于共生矩阵的SAR图像村庄识别方法

50. 基于灰度共生矩阵的图像分割方法研究