基于 shell 命令和 Markov 链模型的用户行为异常检测

田新广[①②]　孙春来[①]　段洣毅[①]
[①](北京交通大学计算技术研究所　北京　100029)
[②](国防科技大学电子科学与工程学院　长沙　410073)

摘　要：异常检测是目前入侵检测系统(IDS)研究的主要方向。该文提出一种基于 shell 命令和 Markov 链模型的用户行为异常检测方法，该方法利用一阶齐次 Markov 链对网络系统中合法用户的正常行为进行建模，将 Markov 链的状态与用户执行的 shell 命令联系在一起，并引入一个附加状态；Markov 链参数的计算中采用了运算量较小的命令匹配方法；在检测阶段，基于状态序列的出现概率对被监测用户当前行为的异常程度进行分析，并提供了两种可选的判决方案。文中提出的方法已在实际入侵检测系统中得到应用，并表现出良好的检测性能。

关键词：入侵检测；shell 命令；Markov 链；异常检测；行为轮廓
中图分类号：TP393　　**文献标识码**：A　　**文章编号**：1009-5896(2007)11-2580-05

Anomaly Detection of User Behaviors Based on Shell Commands and Markov Chain Models

Tian Xin-guang[①②]　Sun Chun-lai[①]　Duan mi-yi[①]
[①](*Research Institute of Computing Technology, Beijing Jiaotong University, Beijing* 100029, *China*)
[②](*College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha* 410073, *China*)

Abstract: Anomaly detection acts as one of the important directions of research on Intrusion Detection Systems(IDSs).This paper presents a new method for anomaly detection of user behaviors based on shell commands and Markov chain models. The method constructs a one-order Markov chain model to represent the normal behavior profile of a network user, and associates shell commands with the states of the Markov chain. The parameters of the Markov chain model are estimated by a command matching algorithm which is computationally efficient. At the detection stage, the probabilities of the state sequences of the Markov chain is firstly computed, and two different schemes can be used to determine whether the monitored user's behaviors are normal or anomalous while the particularity of user behaviors is taken into account. The application of the method in practical intrusion detection systems shows that it can achieve high detection performance.

Key words: Intrusion detection; Shell command; Markov chain; Anomaly detection; Behavior profile

1　引言

网络入侵检测技术主要有异常检测和误用检测两种基本类型。目前，异常检测是入侵检测研究的主要方向，这种检测技术建立系统、用户或程序的正常行为轮廓，通过被监测对象的实际行为与正常行为轮廓之间的比较和匹配来检测入侵[1]，其优点是不需要过多有关系统缺陷的知识，并且能够检测出未知的入侵模式。近年来，针对用户和程序行为的异常检测得到了较多的研究和应用[1−10]。文献[1，3]提出了基于隐Markov模型(HMM)的用户行为异常检测方法，这种方法的主要优点是检测准确率高，但是HMM训练和工作中所需要的计算量比较大，检测效率较低。文献[5]提出一种基于模式挖掘的用户行为异常检测方法，该方法利用数据挖掘中的关联分析和序列挖掘技术对用户行为进行模式挖掘，并采用基于递归式相关函数的模式比较算法对用户历史正常行为和当前行为进行比较。文献[6，7]分别提出了基于HMM的程序行为异常检测方法，这两种方法是HMM在二元分类问题中较为典型的用法，在训练数据充足的情况下均能够获得比较高的检测准确率。文献[9]研究了基于实例学习的用户行为异常检测方法，该方法的优点是原理较为简单，可操作性强，但它没有考虑用户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，因而检测准确率相对较低[3]。文献[11]在文献[9]提出的检测方法的基础上，改进了对用户行为模式和行为轮廓的表示方式，并采用了新的相似度赋值方法，提高了检测准确度。

在以上研究工作的基础上，本文提出一种新的基于shell命令和Markov链模型的用户行为异常检测方法，该方法兼顾了计算成本和检测准确率，具有较强的实用性。该方法的计算成本低于文献[1，3]中基于HMM的检测方法，在检测准确率方面则优于文献[9]中基于实例学习的检测方法。目前，该

2006-04-03 收到，2006-09-26 改回
国家 863 高技术研究发展基金(863-307-7-5)和北京首信集团科研基金(050203)资助课题

方法已应用于国防科技大学和北京首信集团联合研制的入侵检测系统[12]，并表现出良好的检测性能。

2　Markov 链的概念与定义

Markov 链是状态和时间参数都离散的 Markov 随机过程，其定义如下：

定义 1　对于随机序列 X_n，在任一时刻 n，它可以处在状态 $\theta_1, \theta_2, \cdots, \theta_N$ 上(其状态集合为 $\Omega_q = \{\theta_1, \theta_2, \cdots, \theta_N\}$)，如果该随机序列在 $m+k$ 时刻所处的状态为 q_{m+k} 的概率，只与它在 m 时刻的状态 q_m 有关，而与 m 时刻之前它所处的状态无关，即

$$P\left(X_{m+k} = q_{m+k} / X_m = q_m, X_{m-1} = q_{m-1}, \cdots, X_1 = q_1\right) = P\left(X_{m+k} = q_{m+k} / X_m = q_m\right) \quad (1)$$

则称 X_n 为 Markov 链。式(1)中 $q_1, q_2, \cdots, q_m, q_{m+k} \in \Omega_q = \{\theta_1, \theta_2, \cdots, \theta_N\}$。

定义 2　对于 N 个状态的 Markov 链 X_n，称

$$P_{ij}(m,\ m+k) = P\left(q_{m+k} = \theta_j / q_m = \theta_i\right),\ 1 \le i, j \le N \quad (2)$$

为 k 步转移概率；如果 $P_{ij}(m,\ m+k)$ 与 m 无关时，称这个 Markov 链为齐次 Markov 链，此时 $P_{ij}(m, m+k) = P_{ij}(k)$。当 $k=1$ 时，$P_{ij}(1)$ 称为一步转移概率，简称为转移概率，记为 a_{ij}，并记 $\pi_i = P(q_1 = \theta_i)$。显然，$0 \le a_{ij} \le 1$，且 $\sum_{i=1}^{N} \pi_i = 1$。

定义 3　对于 N 个状态的齐次 Markov 链 X_n，称矩阵 $\boldsymbol{A} = (a_{ij})_{N \times N}$ 为状态转移概率矩阵，并称矢量 $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_N)$ 初始状态概率矢量。

3　基于 shell 命令和 Markov 链模型的用户行为异常检测新方法

本文提出的用户行为检测方法主要用于Unix(或Linux)平台上以shell命令为审计数据的入侵检测系统。在一个实际的计算机网络系统中，一般会有多个合法用户，这些合法用户通常具有不同的操作权限，而且不同的合法用户具有不同的行为特点。在很多情况下，我们需要对系统中一些关键合法用户的行为进行监视，检测其行为中的异常，以防止其他用户(包括非法用户)冒用这些关键合法用户的账号进行非法操作，或者防止这些关键合法用户进行非授权操作[12]。本文提出的检测方法在用户界面层建立网络系统中一个(或一组)关键合法用户的正常行为轮廓，并在检测中通过关键合法用户的当前行为与该正常行为轮廓之间的比较来识别异常行为；如果关键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，即认为发生了异常，这种异常可能是关键合法用户进行了非授权操作，也可能是系统中其他合法用户或外部入侵者冒充关键合法用户进行了非法操作。

本文的检测方法采用shell会话中用户执行的shell命令行作为原始审计数据，其原因在于：(1)在Unix平台上，shell是终端用户与操作系统之间最主要的界面，很大比例的用户活动都是利用shell完成的；(2)与其它审计数据(如CPU使用量、内存占用率)相比，shell命令能够更直接地反映用户的行为；(3)shell命令比较容易收集，也便于处理和分析。与文献[1，3，13]中的检测方法相同，本文的检测方法对用户在shell会话中所执行的原始shell命令进行了预处理，具体操作如下：(1)提取出shell命令的名称及参数，将shell命令行中的主机名、网址等信息用统一格式的标识符号来代替；(2)将各命令符号按照在shell会话中的出现次序进行排列；(3)把不同的shell会话按照时间顺序进行连接；(4)在每个会话开始和结束的时间点上插入标识符号[1,3]。经预处理后，原始的shell命令行数据在形式上成为shell命令流，即一系列按时间顺序排列的shell命令符号。对原始shell命令行进行预处理的详细方法以及shell命令流的具体形式可参见文献[3]或文献[13]。

3.1　Markov 链状态的确定

本文的检测方法采用一阶齐次 Markov 链描述网络系统中一个合法用户的正常行为轮廓；该 Markov 链的状态与合法用户执行的 shell 命令相对应。确定该 Markov 链状态的步骤有以下几个：

(1)获取该合法用户的正常行为训练数据。设正常行为训练数据为 $R = (s_1, s_2, \cdots, s_r)$，它是对该合法用户在历史上正常操作时所执行的 shell 命令行进行预处理所得到的 shell 命令流(其长度为 r)，其中 s_i 表示按时间顺序排列的第 i 个 shell 命令。

(2)将训练数据 R 中互不相同的 shell 命令提取出来，并计算这些 shell 命令在 R 中的出现频率。设 R 中互不相同的 shell 命令共有 M 个($M \le r$)，分别记为 $\hat{s}_1$，$\hat{s}_2, \cdots, \hat{s}_M$，并设第 i 个 shell 命令 $\hat{s}_i$ 在 R 中的出现频率为 F_i，则有

$$F_i = C_i / r,\ 1 \le i \le M \quad (3)$$

式中 C_i 为 $\hat{s}_i$ 在 R 中的出现次数。

(3)设定一个频率门限 η，将 $\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_M$ 中出现频率大于或等于频率门限 η 的 shell 命令提取出来。设在 R 中出现频率大于或等于频率门限 η 的 shell 命令共有 W 个($W \le M$)，分别记为 $s_1^*, s_2^*, \cdots, s_W^*$，并将它们在 R 中的出现频率分别记为 $F_1^*, F_2^*, \cdots, F_W^*$，其中 F_j^* 为 s_j^* 在 R 中的出现频率($1 \le j \le W$)。为以下分析方便，设 shell 命令集合 $\Omega_S = \{s_1^*, s_2^*, \cdots, s_W^*\}$。

(4)将 Markov 链的状态个数确定为 $W+1$，状态集合确定为 $\Omega_q = \{1, 2, \cdots, W+1\}$，其中状态 j 与 shell 命令 s_j^* 对应($1 \le j \le W$)；状态 $W+1$ 为附加状态，它对应于除 $\Omega_S = \{s_1^*, s_2^*, \cdots, s_W^*\}$ 中命令之外的其它 shell 命令。

3.2 Markov 链参数的计算

参数计算是 Markov 链在用户行为异常检测中应用的关键问题。在该方法中，Markov 链的初始状态概率矢量 $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_{W+1})$ 用于描述合法用户正常操作时各个 shell

命令在初始时刻出现的概率，状态转移概率矩阵 $\boldsymbol{A}=(a_{ij})_{(W+1)\times(W+1)}$ 则用于描述各个 shell 命令之间的时序相关性。 $\boldsymbol{\pi}=(\pi_1, \pi_2,\cdots, \pi_{W+1})$ 的计算方法为

$$\pi_j=\begin{cases}F_j^*, & 1\le j\le W\\ 1-\sum_{i=1}^{W}\pi_i, & j=W+1\end{cases} \tag{4}$$

$\boldsymbol{A}=(a_{ij})_{(W+1)\times(W+1)}$ 的计算可分以下几个步骤进行：

(1)设定初值 $\boldsymbol{A}=(a_{ij})_{(W+1)\times(W+1)}:=0$， $m:=1$。

(2)将训练数据 $R=(s_1, s_2,\cdots, s_r)$ 中的 shell 命令 s_m 与 shell 命令集合 $\Omega_S=\{s_1^*, s_2^*,\cdots, s_W^*\}$ 中的命令进行匹配。如果 s_m 与 Ω_S 中的某个 shell 命令相同(即 $s_m\in\Omega_S$)，则根据该 shell 命令确定 s_m 对应的状态 q_m；设 s_m 与 Ω_S 中的第 n 个 shell 命令 s_n^* 相同(即 $s_m=s_n^*$，其中 $1\le n\le W$)，则 $q_m:=n$。如果 s_m 与 Ω_S 中的任何一个 shell 命令都不相同(即 $s_m\notin\Omega_S$)，则 s_m 对应的状态 $q_m:=W+1$。

(3) $m:=m+1$。如果 $m\le r$，返回执行步骤(2)；如果 $m=r+1$，执行步骤(4)。

(4) $k:=1$。

(5) $i:=q_k$， $j:=q_{k+1}$； $a_{ij}:=a_{ij}+1$。

(6) $k:=k+1$。如果 $k\le r-1$，返回执行步骤(5)。如果 $k=r$，执行步骤(7)。

(7)对于 $1\le i,j\le W+1$，如果 $\sum_{j=1}^{W+1}a_{ij}>0$，则 $a_{ij}:=a_{ij}/\sum_{j=1}^{W+1}a_{ij}$。至此，状态转移概率矩阵的计算结束，此时的 $\boldsymbol{A}=(a_{ij})_{(W+1)\times(W+1)}$ 即为最后结果。

以上计算状态转移概率矩阵的方法是根据 shell 命令集合 $\Omega_S=\{s_1^*, s_2^*,\cdots, s_W^*\}$，通过命令匹配得到正常行为训练数据 $R=(s_1, s_2,\cdots, s_r)$ 所对应的状态序列 $q=(q_1, q_2,\cdots, q_r)$ (其中 q_m 是 shell 命令 s_m 对应的状态)，然后对该状态序列中各个状态之间的转移次数进行统计，进而计算出各个状态转移概率。在以上的参数计算中，本文假设了 $W+1$ 个状态的 Markov 链是一个各态历经随机过程。

3.3 检测

在检测阶段，利用以上计算出的 Markov 链参数，基于状态序列出现概率对该合法用户的当前行为进行判决。检测阶段的工作包括以下几部分：

(1)获取该合法用户在被监测的时间内执行的shell命令行，并将shell命令行处理成shell命令流的形式[12]。设该shell命令流为 $\overline{R}=(\overline{s}_1, \overline{s}_2,\cdots, \overline{s}_{\overline{r}})$，其中 $\overline{s}_j$ 表示按时间顺序排列的第 j 个shell命令， $\overline{r}$ 为该命令流的长度。在实时检测(在线检测)的情况下， $\overline{R}$ 中的各个shell命令是按照时间顺序依次得到的。

(2)利用以上参数计算中的命令匹配方法，将 $\overline{R}=(\overline{s}_1,\overline{s}_2,\cdots,\overline{s}_{\overline{r}})$ 中的每个 shell 命令依次同 shell 命令集合 $\Omega_S=\{s_1^*,s_2^*,\cdots, s_W^*\}$ 中的 shell 命令进行匹配，得到 $\overline{R}=(\overline{s}_1, \overline{s}_2,\cdots, \overline{s}_{\overline{r}})$ 对应的状态序列 $\overline{q}=(\overline{q}_1, \overline{q}_2,\cdots, \overline{q}_{\overline{r}})$，其中 $\overline{q}_m$ 是 $\overline{R}$ 中的第 m 个 shell 命令 $\overline{s}_m$ 对应的状态。若 $\overline{s}_m=s_n^*$，则 $\overline{q}_m=n$；若 $\overline{s}_m\notin\Omega_S$，则 $\overline{q}_m=W+1$。这里， $1\le m\le\overline{r}$， $1\le n\le W$。

(3)用滑动窗在状态序列 $\overline{q}=(\overline{q}_1, \overline{q}_2,\cdots, \overline{q}_{\overline{r}})$ 中依次截取短序列，得到状态短序列流 $\mathrm{Sq}=(\mathrm{Sq}_1, \mathrm{Sq}_2,\cdots, \mathrm{Sq}_{\overline{r}-u+1})$，其中第 i 个状态短序列 $\mathrm{Sq}_i=(\overline{q}_i, \overline{q}_{i+1}, \cdots, \overline{q}_{i+u-1})$，且 $u<\overline{r}$， $1\le i\le\overline{r}-u+1$。 u 为状态短序列的长度。(通过在 $\overline{q}$ 中截取短序列，可以实时地对用户行为进行判别。)

(4)根据 Markov 链的参数 $\boldsymbol{\pi}$ 和 $\boldsymbol{A}$，计算 Sq 中每个状态短序列的出现概率。设该用户的当前行为是正常行为的情况下 Sq 中第 i 个状态短序列 $\mathrm{Sq}_i=(\overline{q}_i, \overline{q}_{i+1},\cdots, \overline{q}_{i+u-1})$ 的出现概率为 $P(\mathrm{Sq}_i)$，则有

$$\begin{aligned}P(\mathrm{Sq}_i)&=P(\overline{q}_i)P(\overline{q}_{i+1}/\overline{q}_i)P(\overline{q}_{i+2}/\overline{q}_{i+1})\cdots P(\overline{q}_{i+u-1}/\overline{q}_{i+u-2})\\&=P(\overline{q}_i)\prod_{k=i}^{i+u-2}P(\overline{q}_{k+1}/\overline{q}_k)\end{aligned} \tag{5}$$

式中 $P(\overline{q}_i)$ 表示该用户的当前行为是正常行为的情况下状态 $\overline{q}_i$ 的出现概率， $P(\overline{q}_{i+1}/\overline{q}_i)$ 则表示从状态 $\overline{q}_i$ 到 $\overline{q}_{i+1}$ 的转移概率。设 $\overline{q}_i=m$， $\overline{q}_{i+1}=n$，则 $P(\overline{q}_i)=\pi_m$， $P(\overline{q}_{i+1}/\overline{q}_i)=a_{m\,n}$ (这里 $1\le m,n\le W+1$)。经过上述计算，可以得到状态短序列流 Sq 对应的概率序列 $P=(P(\mathrm{Sq}_1), P(\mathrm{Sq}_2),\cdots, P(\mathrm{Sq}_{\overline{r}-u+1}))$。

(5)对概率序列 $P=(P(\mathrm{Sq}_1), P(\mathrm{Sq}_2),\cdots, P(\mathrm{Sq}_{\overline{r}-u+1}))$进行加窗和处理，得到判决值，并将判决值与判决门限比较，进而对该用户的行为作出判决。(考虑到用户在短时间内的行为可能会偏离其历史行为，检测中并不直接利用 $P(\mathrm{Sq}_i)$ 对用户行为进行判决，而是对概率序列 P 进行加窗处理来得到判决值)。在对概率序列 P 进行加窗处理并对该用户行为进行判决时，有以下两种方案可以选择。

第 1 种方案 设定一个窗长度 w，然后对概率序列 P 进行加窗和处理，得到如下判决值：

$$\begin{aligned}D(n)&=\frac{1}{w}\sum_{i=n-w+1}^{n}\mathrm{sgn}[P(\mathrm{Sq}_i)-a]\\&=\frac{1}{w}\sum_{i=n-w+1}^{n}\mathrm{sgn}[P(\overline{q}_i)\prod_{k=i}^{i+u-2}P(\overline{q}_{k+1}/\overline{q}_k)-a]\end{aligned} \tag{6}$$

式中 $D(n)$ 表示状态短序列 Sq_n 对应的判决值，a 为预先设定的概率门限，w 为窗长度，且 $w\le n\le\overline{r}-u+1$， n 的增长步长为 1。状态短序列流 $\mathrm{Sq}=(\mathrm{Sq}_1, \mathrm{Sq}_2,\cdots, \mathrm{Sq}_{\overline{r}-u+1})$ 中第 w 个状态短序列 Sq_w 及其后面的每个状态短序列都分别对应一个判决值。除式(6)之外，还可以按照以下公式计算判决值：

$$\begin{aligned}D(n)&=\frac{1}{w}\sum_{i=n-w+1}^{n}\lg[P(\mathrm{Sq}_i)+e]\\&=\frac{1}{w}\sum_{i=n-w+1}^{n}\lg[P(\overline{q}_i)\prod_{k=i}^{i+u-2}P(\overline{q}_{k+1}/\overline{q}_k)+e]\end{aligned} \tag{7}$$

式中 e 是一个预先设定的大于或等于 0 的常数。在计算出判决值 $D(n)$ 后，就可以根据 $D(n)$ 对该用户(被监测用户)的当

前行为做出判决。判决方法为：设定一个判决门限 λ，如果 $D(n)$ 大于判决门限 λ，将该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，该用户的当前行为是相对于状态短序列 Sq_n 而言的，它是指以 Sq_n 为终点的 w 个状态短序列 Sq_{n-w+1}, Sq_{n-w+2},…, Sq_n 对应的行为，亦即该用户执行的以 $\overline{s}_{n+u-1}$ 为终点的 $w+u-1$ 个 shell 命令 $\overline{s}_{n-w+1}$,$\overline{s}_{n-w+2}$, …, $\overline{s}_{n+u-1}$ 对应的行为。

第 2 种方案　设定 V 个窗长度 $w(1),w(2),\cdots,w(V)$ 且 $w(1)<w(2)<\cdots<w(V)$；设定 V 个判决上限 $u(1),u(2),\cdots,u(V)$ 和 V 个判决下限 $d(1),d(2),\cdots,d(V)$，其中 $u(k)$ 和 $d(k)$ 是第 k 个窗长度 $w(k)$ 对应的判决上限和判决下限($1\le k\le V$)，且 $u(1)>u(2)>\cdots>u(V-1)>u(V)=d(V)>d(V-1)>\cdots>d(2)>d(1)$，然后按照以下方法计算状态短序列 Sq_n 对应的判决值并对该用户的当前行为作出判决：

步骤 1　设定 $k:=1$。

步骤 2 将 n 与 $w(k)$ 进行比较。如果 $n<w(k)$，则不计算判决值，也不对该用户的当前行为进行判决，并不再执行下面的步骤。如果 $n\ge w(k)$，执行步骤 3。

步骤 3　计算 Sq_n 对应的判决值 $D(n,k)$：

$$D(n,k)=\frac{1}{w(k)}\sum_{i=n-w(k)+1}^{n}\mathrm{sgn}[P(\mathrm{Sq}_i)-a\,]$$
$$=\frac{1}{w(k)}\sum_{i=n-w(k)+1}^{n}\mathrm{sgn}[P(\overline{q}_i)\prod_{k=i}^{i+u-2}P(\overline{q}_{k+1}/\overline{q}_k)-a\,]\quad(8)$$

或者按式(9)计算 $D(n,k)$：

$$D(n,k)=\frac{1}{w(k)}\sum_{i=n-w(k)+1}^{n}\lg\,[P(\mathrm{Sq}_i)+e]$$
$$=\frac{1}{w(k)}\sum_{i=n-w(k)+1}^{n}\lg\,[P(\overline{q}_i)\prod_{k=i}^{i+u-2}P(\overline{q}_{k+1}/\overline{q}_k)+e]\quad(9)$$

式(8)和式(9)中，a 为预先设定的概率门限，e 是一个预先设定的大于或等于 0 的常数。

步骤 4　判断是否满足判决条件：$D(n,k)>u(k)$。如果满足该条件，则将该用户的当前行为判为正常行为；至此，对状态短序列 Sq_n 对应的用户当前行为的判决结束，不再执行下面的步骤。如果不满足 $D(n,k)>u(k)$，执行步骤 5。

步骤 5　判断是否满足判决条件：$D(n,k)\le d(k)$。如果满足该条件，则将该用户的当前行为判为异常行为；至此，对状态短序列 Sq_n 对应的用户当前行为的判决结束。如果不满足 $D(n,k)\le d(k)$，执行步骤 6。

步骤 6　$k:=k+1$，即 k 的值增加 1，并返回执行步骤 2。

需要指出，在实时检测(在线检测)的情况下，被监测用户所执行的 shell 命令行的获取和预处理，shell 命令的匹配，状态短序列出现概率的计算，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 $\overline{R}=(\overline{s}_1,\overline{s}_2,\cdots,\overline{s}_\tau)$ 中的第 $w+u-1$ 个 shell 命令之后，该用户每再执行完一个 shell 命令，检测系统就可以通过命令匹配得到该 shell 命令对应的状态，并以此状态为终点组成(截取)一个新的状态短序列，同时计算该状态短序列的出现概率和相应的判决值，进而对该用户的当前行为做出一次判决。

4　实验设计与结果分析

本文对以上检测方法的性能进行了实验，实验中采用了普渡大学公开发布的shell命令实验数据[3]，该数据包含 8 个用户在两年时间内的活动记录。本文使用了其中 4 个用户 user1，user2，user3，user4 的数据，并将user2 设为合法用户，将user1，user3，user4 设为非法用户。user2 的实验数据(shell命令流)中有 15000 个shell命令，其中前 10000 个命令作为正常行为训练数据用于Markov链状态的确定和参数计算，后 5000 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。user1，user3，user4 的实验数据(shell命令流)各包含 5000 个shell命令，这些命令均作为异常行为测试数据用于检测概率的测试。实验中的检测阶段采用第一种方案计算判决值并对用户行为进行判决，实验的参数设置为 $\eta=0$，$u=2$，$w=91$，$a=10^{-4}$，$e=10^{-20}$。实验时，正常行为训练数据中互不相同的shell命令(符号)共有 224 个，Markov链的状态个数为 225。图 1 和图 2 分别示出了Markov链的初始状态概率和状态转移概率。

图 3 和图 4 分别示出了由式(6)和式(7)计算出的判决值曲线。图中上方的实线是合法用户 user2 的测试数据对应的判决值曲线，下方的 3 条虚线分别是非法用户 user1，user3，user4 的测试数据对应的判决值曲线。可见，图中的两种判决值曲线具有良好的可分性。

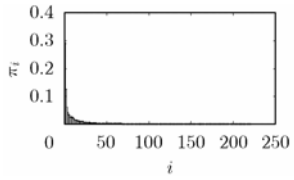

图 1　Markov 链的初始状态概率图

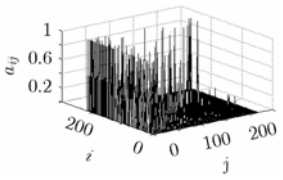

图 2　Markov 链的状态转移概率图

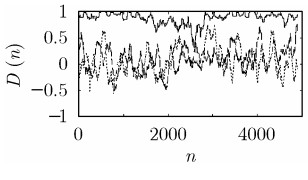

图 3 式(6)对应的判决值曲线

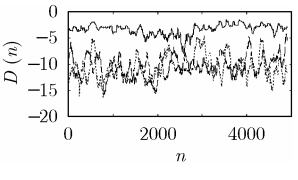

图 4 式(7)对应的判决值曲线

在实验的检测阶段，通过调整判决门限可以得到不同虚警概率条件下对 3 个非法用户的异常行为的平均检测概率；表 1 给出了式(6)和式(7)两种判决值计算公式对应的实验结果。

由表 1 的实验结果可见，采用式(6)和式(7)计算判决值均可以获得很高的检测准确率；而且，两者对应的检测准确率比较接近，这说明基于状态序列出现概率的判决值计算方法是

表 1 两种判决值计算公式对应的实验结果

虚警概率	0	0.001	0.005	0.010	0.050
式(6)对应的平均检测概率	0.808	0.829	0.834	0.912	0.961
式(7)对应的平均检测概率	0.864	0.865	0.867	0.932	0.995

一种性能稳健的方法。此外，根据实验结果，当参数设置不同时，检测性能会有一定的变化，因而，根据具体用户的行为特点选择合适的参数是实际应用中提高检测性能的重要途径。

5 结束语

本文提出一种新的基于 shell 命令和 Markov 链模型的用户行为异常检测方法，主要用于 Unix 平台上以 shell 命令为审计数据的入侵检测系统。该方法具有较低的计算成本和较高的检测准确率，已应用于实际的入侵检测系统。由于 shell 命令和系统调用在数据形式上具有一些共性，因而该方法的检测思想也可用于以系统调用为审计数据的程序行为异常检测，但具体的检测性能还有待进一步研究和实验。

参 考 文 献

[1] Lane T and Carla E B. An empirical study of two approaches to sequence learning for anomaly detection. *Machine Learning*, 2003, 51(1): 73–107.

[2] Ye N, Zhang Y, and Borror C M. Robustness of the Markov chain model for cyber attack detection. *IEEE Trans. on Reliability*, 2003, 52(3): 122–138.

[3] Lane T. Machine learning techniques for the computer security domain of anomaly detection [Ph.D.Thesis]. Purdue University, 2000.

[4] Mukkamala S, Sung A H, and Abraham A. Intrusion detection using an ensemble of intelligent paradigms. *Journal of Network and Computer Application*, 2005, 28(2): 167–182.

[5] 连一峰，戴英侠，王航. 基于模式挖掘的用户行为异常检测. 计算机学报, 2002, 25(3): 325–330.

[6] Yan Qiao, Xie Wei-Xin, and Yang Bin, *et al.*. An anomaly intrusion detection method based on HMM. *Electronics Letters*, 2002, 38(13): 663–664.

[7] Warrender C, Forrest S, and Pearlmutter B. Detecting intrusions using system calls: alternative data models. Proc. of The 1999 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 1999: 133–145.

[8] Maxion R A and Townsend T N. Masquerade detection using truncated command lines. Proc. of International Conference on Dependable Systems and Networks, Washington, DC, USA, 2002: 219–228.

[9] Lane T and Brodley C E. Temporal sequence learning and data reduction for anomaly detection. *ACM Trans. on Information and System Security*, 1999, 2(3): 295–331.

[10] Schonlau M and DuMouchel W, *et al.*. Computer intrusion: Detecting masquerades. *Statistical Science*, 2001, 16(1): 58–74.

[11] 孙宏伟，田新广，李学春，张尔扬. 一种改进的 IDS 异常检测模型. 计算机学报, 2003, 26(11): 1450–1455.

[12] 田新广. 基于主机的入侵检测方法研究. [博士论文]. 长沙: 国防科技大学, 2005.

[13] 田新广，高立志，张尔扬. 新的基于机器学习的入侵检测方法. 通信学报, 2006, 27(6): 108–114.

田新广： 男，1976 年生，博士后，参与完成多项国防重大科研项目，发表论文 30 余篇，发明专利 7 项，主要研究方向为信号与信息处理、网络安全、入侵检测.

孙春来： 女，1962 年生，高级工程师，主要研究方向为计算机应用、网络安全.

段洣毅： 男，1953 年生，研究员，博士生导师，中国计算机学会理事，中国电子学会高级会员，享受政府特殊津贴，主要研究方向为计算机应用、信息处理.