基于 shell 命令和多重行为模式挖掘的用户伪装攻击检测

田新广 段淼毅 程学旗

(中国科学院计算技术研究所网络科学与技术重点实验室 北京 100190)

摘 要 伪装攻击是指非授权用户通过伪装成合法用户来获得访问关键数据或更高层访问权限的行为.近年来,伪装攻击检测在保障网络信息安全中发挥着越来越大的作用.文中提出一种新的用户伪装攻击检测方法.同现有的典型检测方法相比,该方法在训练阶段改进了对用户行为模式的表示方式,通过合理选择用户行为特征并基于阶梯式的序列模式支持度来建立合法用户的正常行为轮廓,提高了用户行为描述的准确性和对不同类型用户的适应性;在充分考虑 shell 命令审计数据时序特征的基础上,针对伪装攻击行为复杂多变的特点,提出基于多重行为模式并行挖掘和多门限联合判决的检测模型,并通过交叉验证和等量迭代逼近方法确定最佳门限参数,克服了单一序列模式检测模型在性能稳定性和容错能力方面的不足,在不明显增加计算成本的条件下大幅度提高了检测准确度.文中提出的方法已应用于实际检测系统,并表现出良好的检测性能.

关键词 网络安全;伪装攻击;入侵检测;shell 命令;异常检测

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2010.00697

Masquerade Detection Based on Shell Commands and Multiple Behavior Pattern Mining

TIAN Xin-Guang DUAN Mi-Yi CHENG Xue-Qi

(*Key Laboratory of Network Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

Abstract Masquerade attacks are attempts by unauthorized users to gain access to confidential data or greater access privileges, while pretending to be legitimate users. Masquerade detection is now one of the major concerns of system security research. This paper proposes a novel method to distinguish legitimate users from masqueraders based on shell commands and multiple behavior pattern mining. In the method, behavioral patterns of legitimate users are characterized by shell command sequences of different lengths, and hierarchical sequence supports are employed to construct the normal behavior profiles of legitimate users, which improve the precision and adaptability of user profiling. In the detection stage, a model based on multiple sequence pattern parallel mining and multiple threshold joint decision is used to determine whether the monitored user's behavior is normal or anomalous. The model gives attention to both detection accuracy and computational efficiency, and is especially applicable for on-line detection. This study empirically demonstrated the promising performance of the method, and it has succeeded in getting application in practical host-based intrusion detection systems.

Keywords network security; masquerade attack; intrusion detection; shell command; anomaly detection

---

收稿日期:2009-02-10;最终修改稿收到日期:2009-11-17.本课题得到国家"八六三"高技术研究发展计划项目基金(2006AA01Z452)、国家 242 信息安全计划项目基金(2005C39)和国家"九七三"重点基础研究发展规划项目基金(2004CB318109)资助.**田新广**,男,1976 年生,博士,副研究员,主要研究方向为网络安全、入侵检测、智能信息处理. E-mail: tianxinguang@163.com. **段淼毅**,男,1953 年生,研究员,博士生导师,主要研究领域为计算机应用、入侵检测. **程学旗**,男,1971 年生,博士,研究员,博士生导师,主要研究领域为信息安全、互联网挖掘与搜索、舆情计算.

1 引 言

入侵者假冒或伪装成合法用户进入信息系统的入侵行为称为伪装(masquerade)[1-3]. 近年来,伪装攻击在网络信息安全事件中的比例不断增长[3-5],成为对系统破坏最为严重的攻击方式之一. 伪装攻击检测是当前网络安全领域关注的热点. 由于合法用户的行为本身是变化的,且伪装用户的行为可能看起来是正常的,这种不确定性使得伪装攻击检测比传统的网络入侵检测更为困难[4]. 目前的伪装攻击检测系统大多采用异常检测技术,这种技术对合法用户的正常行为进行建模,通过被监测用户的实际行为与合法用户正常行为之间的比较或匹配来检测入侵(攻击);其优点是不需要过多有关入侵行为的先验知识,且能够检测出未知的攻击类型[5].

近年来,国内外已经开展了数据挖掘、机器学习、支撑向量机、神经网络等技术在用户伪装攻击检测中的应用研究[2-9],研究目标主要是提高检测的准确性、实时性、高效性以及自适应性,其中一些研究成果已经接近或达到了实用化水平. Schonlau 等人研究了基于统计理论的伪装攻击检测方法[7],并利用 AT&T Shannon 实验室的 shell 命令数据对不同类型的统计方法进行了实验和综合对比. Maxion 等人对 Schonlau 的检测方法进行了改进[8],引入了贝叶斯分类算法,较大程度地提高了检测准确度. Lane 等人开展了基于机器学习的伪装攻击检测研究[9],利用特定的相似度函数计算行为模式之间的相似度,并将加窗平滑后的相似度曲线作为检测用户异常行为的依据. Tian 等人在 Lane 的基础上改进了相似度赋值方式[10],并采用可变窗长度进行相似度平滑滤噪,进一步改善了检测性能. 连一峰等人提出一种基于模式挖掘的用户行为异常检测方法[11],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,并采用了基于相关函数的模式比较算法. 此外,Kim 等人研究了基于支撑向量机的伪装攻击检测方法[4],Szymanski 等人提出了基于递归数据挖掘的检测方法[5],这两种方法均具有较高的检测效率,但对用户行为变化的适应性不强[2],仅适用于训练数据较为充分的场合.

在以上工作的基础上,本文提出一种新的用户伪装攻击检测方法. 该方法利用 shell 会话中用户执行的 shell 命令作为原始审计数据,基于实际检测系统[12]的实例分析和已有研究结论[2-9],改进了对用户行为模式的表示方式,通过合理选择用户行为特征并基于阶梯式的序列模式支持度来建立合法用户的正常行为轮廓,提高了用户行为描述的准确性和对不同类型用户的适应性;在检测阶段,充分考虑了 shell 命令审计数据的时序特征以及伪装攻击行为复杂多变的特点,通过多重行为模式并行挖掘来得到多个参考判决值,采用多门限联合判决的方式对被监测用户当前行为的异常程度进行分析,并通过交叉验证和等量迭代逼近方法确定最佳门限参数,克服了基于单一序列模式的检测模型[5,7-9]在检测性能稳定性和容错能力方面的不足. 实验表明,同现有的典型检测方法相比,本文的方法在不明显增加计算成本的条件下大幅度提高了检测准确度,具有很强的实用性. 该方法已应用于文献[12]所述发明专利的检测系统,并表现出良好的检测性能.

2 审计数据的分析及预处理

目前,基于主机数据的入侵检测研究有两个主要的分支[3-5,13],分别是以系统调用为审计数据的程序行为异常检测和以 shell 命令为审计数据的用户行为异常检测. 相对用户行为而言,程序行为比较单一,利用系统调用可以对程序行为特别是特权程序的行为建立较为稳定简洁的模型[14],但是,系统调用来自系统的核心层,获取过程比较复杂,耗用的系统资源较多,而且,基于系统调用的程序行为异常检测不能直接检测出用户帐号假冒等系统内部的攻击行为[13]. 由于用户行为复杂多变,使得用户正常行为模型的建立变得较为困难;国内外大都倾向于使用比较完整的审计事件来对用户行为进行建模. 在 Unix 系统环境下,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;shell 命令在系统用户层比较容易获取,而且能够直接反映出用户的行为模式. 因此,基于 shell 命令的用户伪装攻击检测在近些年得到了较多的研究[2,4-5,7-9].

与文献[4-5,7-9]中的伪装攻击检测方法相同,本文的检测方法采用 Unix 平台上的 shell 命令作为审计数据. 在训练和检测阶段,需要对用户的原始 shell 命令数据进行预处理. 预处理的方式主要有两种,第 1 种方式如文献[7-8]所述,预处理时只保留 shell 命令的名称,滤除命令参数和时间等信息. 第 2 种方式如文献[2,9]所述,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. 例如，某用户的一个 shell 会话数据：

```
>cd ~/private/docs
>ls -laF|more
>cat foo.txt bar.txt zlg.txt >~/tian/zly
>mailx tianxg@sina.com
>exit
```

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. 经过以上两种方式预处理后的原始 shell 命令数据在形式上均表现为 shell 命令序列(按时序排列的若干个 shell 命令符号). 相对于第 1 种方式，第 2 种方式预处理后的数据中互不相同的 shell 命令符号明显增多，但能够更加精确地反映用户行为.

3　训　练

用户伪装攻击检测的具体实现过程可分为训练和检测两个阶段[4-5]. 训练阶段的主要工作是根据训练数据对我们所关心的合法用户的正常行为进行建模. 同程序行为相比，用户行为具有一些不同的特点，尤其是在行为的稳定性方面. 因此，在程序行为异常检测中具有良好性能的训练和检测模型并不一定适用于用户行为异常检测[15]. 在用户行为异常检测中，由于用户行为的多变性，我们得到的训练数据往往不够充分，这就要求训练和检测模型应当具有一定的容错性、泛化能力以及对用户行为变化的适应性[2]. 如何利用相对不够充分的训练数据来尽可能精确地描述用户的正常行为轮廓，以及如何利用该正常行为轮廓进行异常检测是我们研究的重点. 在现有的基于机器学习和基于数据挖掘的检测方法中[5,9,11]，一般都采用单一长度的 shell 命令短序列来表示用户的行为模式(行为模式是指用户操作过程中体现出的某种规律性). 但实际中，不同用户所具有的行为模式存在差异，同一用户在完成不同行为模式时所执行的 shell 命令个数也不尽相同，因而，采用单一长度的 shell 命令序列往往难以全面准确地反映用户的行为轮廓，而且，这种方法在实际应用中也不容易估算针对具体用户的最佳序列长度[7,9]；例如，文献[9]主要采用实验方法来确定最佳序列长度，所需的计算量很大，而且其性能缺乏稳定性. 我们针对以上不足进行了改进和修正，采用多种长度不同的 shell 命令短序列来表示用户行为模式，通过合理选择用户行为特征并基于序列模式的支持度来建立合法用户的正常行为轮廓，提高了用户行为描述的准确性和对不同类型用户的适应性. 训练阶段的工作主要有以下几部分：

(1) 获取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原始 shell 命令数据进行预处理所得到的 shell 命令序列(其长度为 r)，其中 s_j 表示按时间顺序排列的第 j 个 shell 命令符号.

(2) 定义 W 种长度不同的 shell 命令短序列，用于表示该合法用户的各种行为模式.

设定义的短序列长度的集合为 $C=\{l(1),l(2),\cdots,l(W)\}$，其中 $l(i)$ 表示第 i 种 shell 命令短序列的长度，且 $l(1)<l(2)<\cdots<l(W)$. 在 W 确定的情况下，C 可有不同的选择. 例如 $W=3$ 时，C 可以为{1,2,3}(即 3 种序列的长度分别为 1,2,3)，也可以为{3,6,9}或其它组合. W 和 C 对检测性能有直接影响，在选择它们时，除了要充分考虑合法用户的行为特征之外，还需考虑检测系统的复杂度及检测效率(W 和 $l(i)$ 越大，检测系统的存储量和工作中的运算量也会越大).

(3) 由训练数据 R 生成 W 个短序列长度分别为 $l(1),l(2),\cdots,l(W)$ 的 shell 命令短序列流.

这里，分别用 $S^1,S^2,\cdots,S^W$ 表示由 R 生成的短序列长度分别为 $l(1),l(2),\cdots,l(W)$ 的 W 个 shell 命令短序列流，其中 S^i 是短序列长度为 $l(i)$($1\le i\le W$)的 shell 命令短序列流，它包含 $r-l(i)+1$ 个 shell 命令短序列；$S^i=(S^i_1,S^i_2,\cdots,S^i_{r-l(i)+1})$，其中 $S^i_j=(s_j,s_{j+1},\cdots,s_{j+l(i)-1})$，$S^i_j$ 是以 shell 命令 s_j 为起点的长度为 $l(i)$ 的短序列($1\le j\le r-l(i)+1$).

(4) 对于 $1\le i\le W$，计算 S^i 中每个 shell 命令短序列在 S^i 中的支持度.

定义 1.　一个长度为 $l(i)$ 的 shell 命令短序列 S^i_+ 在 shell 命令短序列流 S^i 中的支持度等于该序列在 S^i 中的出现次数除以 S^i 中的序列总数($1\le i\le W$)，即

$$support(S^i_+)=number(S^i_+)/(r-l(i)+1) \quad (1)$$

式中 $number(S^i_+)$ 表示短序列 S^i_+ 在 S^i 的 $r-l(i)+1$ 个 shell 命令短序列中的出现次数，$support(S^i_+)$ 表

示短序列 S^i_+ 在 S^i 中的支持度. $support(S^i_+)$ 描述了短序列 S^i_+ 在短序列流 S^i 中的出现概率.

(5)设置 W 个最小支持度

$$minsup(1), minsup(2), \cdots, minsup(W),$$

其中 $minsup(i)$ 是针对 shell 命令短序列流 S^i 中长度为 $l(i)$ 的短序列而设置的($1 \leq i \leq W$). 考虑到长序列模式对短序列模式的相容性,我们采用了阶梯式支持度,要求 $minsup(1) \geq minsup(2) \geq \cdots \geq minsup(W)$.

(6)从 W 个 shell 命令短序列流中,分别按照序列的支持度提取正常行为模式序列.

对于 $1 \leq i \leq W$,将 S^i 中支持度大于或等于 $minsup(i)$ 的短序列提取出来,构成满足支持度要求的序列库 $L(i)$. 设 S^i 中支持度大于或等于 $minsup(i)$ 的短序列共有 $K(i)$ 个,分别记为 $S^i_{1+}, S^i_{2+}, \cdots, S^i_{K(i)+}$(这里 $K(i) \leq r-l(i)+1$),则满足支持度要求的序列库 $L(i)=\{S^i_{1+}, S^i_{2+}, \cdots, S^i_{K(i)+}\}$.

(7)建立 W 个序列库来描述该合法用户的正常行为轮廓.

将 W 个满足支持度要求的序列库 $L(1), L(2), \cdots, L(W)$ 存储起来,建立用于异常检测的正常序列库 L. 这里,$L=\{L(1), L(2), \cdots, L(W)\}$,它用于描述该用户历史上的正常行为轮廓.

4 检 测

检测阶段的工作是根据训练阶段所建立的合法用户正常行为轮廓,利用特定的检测模型来识别被监测用户当前行为中的异常(伪装或假冒行为). 在检测阶段,我们充分考虑了 shell 命令审计数据的时序特征和短时相关性,针对伪装攻击行为复杂多变的特点,提出了基于多重行为模式并行挖掘和多门限联合判决的检测模型,并通过交叉验证和等量迭代逼近方法确定最佳门限参数,与文献[4-5,9]中的检测模型相比,在不明显增加计算成本的条件下大幅度提高了检测准确度,并且增强了检测性能的稳定性和容错能力. 检测阶段的工作包括以下几部分:

(1)获取用于检测的审计数据.

获取该用户在被监测的时间内执行的原始 shell 命令审计数据,并将其预处理成 shell 命令序列的形式. 设预处理后得到的 shell 命令序列为 $\overline{R}=(\bar{s}_1, \bar{s}_2, \cdots, \bar{s}_{\bar{r}})$,其中 $\bar{s}_j$ 表示按时间顺序排列的第 j 个 shell 命令符号,$\bar{r}$ 为该命令序列的长度. 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)由审计数据 $\overline{R}$ 生成 W 个短序列长度分别为 $l(1), l(2), \cdots, l(W)$ 的 shell 命令短序列流.

分别用 $\overline{S}^1, \overline{S}^2, \cdots, \overline{S}^W$ 表示由 $\overline{R}$ 生成的短序列长度分别为 $l(1), l(2), \cdots, l(W)$ 的 W 个 shell 命令短序列流,其中 $\overline{S}^i$ 是短序列长度为 $l(i)$ ($1 \leq i \leq W$) 的 shell 命令短序列流;$\overline{S}^i=(\overline{S}^i_{l(i)}, \overline{S}^i_{l(i)+1}, \cdots, \overline{S}^i_{\bar{r}})$,其中 $\overline{S}^i_j=(\bar{s}_{j-l(i)+1}, \bar{s}_{j-l(i)+2}, \cdots, \bar{s}_j)$,$\overline{S}^i_j$ 是以 shell 命令 $\bar{s}_j$ 为终点、长度为 $l(i)$ 的短序列($l(i) \leq j \leq \bar{r}$). $\overline{S}^i$ 中共有 $\bar{r}-l(i)+1$ 个 shell 命令短序列. 需要指出,在训练阶段利用训练数据生成 shell 命令短序列流时,采用了滑动窗前向截取短序列的方式;而在检测阶段利用审计数据生成 shell 命令短序列流时,采用了滑动窗后向截取短序列的方式,这样便于实时检测.

(3)对 W 个 shell 命令短序列流进行多重行为模式匹配与挖掘.

对于 $1 \leq i \leq W$,将 shell 命令短序列流 $\overline{S}^i=(\overline{S}^i_{l(i)}, \overline{S}^i_{l(i)+1}, \cdots, \overline{S}^i_{\bar{r}})$ 中的每个 shell 命令短序列 $\overline{S}^i_j=(\bar{s}_{j-l(i)+1}, \bar{s}_{j-l(i)+2}, \cdots, \bar{s}_j)$ 同相应的序列库 $L(i)=\{S^i_{1+}, S^i_{2+}, \cdots, S^i_{K(i)+}\}$ 中的短序列进行匹配,如短序列 $\overline{S}^i_j$ 与序列库 $L(i)$ 中的某个短序列相同(即 $\overline{S}^i_j \in L(i)$),则将短序列 $\overline{S}^i_j$ 视为正常行为模式序列,并记 $class(\overline{S}^i_j)=1$. 如果 $\overline{S}^i_j$ 与序列库 $L(i)$ 中的任何一个短序列都不相同(即 $\overline{S}^i_j \notin L(i)$),则将短序列 $\overline{S}^i_j$ 视为异常行为模式序列,并记 $class(\overline{S}^i_j)=0$. 经过以上序列匹配,对于每一个 shell 命令短序列流 $\overline{S}^i=(\overline{S}^i_{l(i)}, \overline{S}^i_{l(i)+1}, \cdots, \overline{S}^i_{\bar{r}})$,可得到序列 $(class(\overline{S}^i_{l(i)}), class(\overline{S}^i_{l(i)+1}), \cdots, class(\overline{S}^i_{\bar{r}}))$.

(4)计算判决值.

根据国内外现有的研究结论,进行伪装攻击的用户在短时间内的行为可能与合法用户的正常行为没有太大差别,但在较长时间段内表现出的行为特征通常会较大程度地偏离合法用户的正常行为轮廓. 考虑到用户伪装攻击的以上特点,我们并不直接利用 $class(\overline{S}^i_j)$ 进行判决,而是通过加窗平滑来得到参考判决值,进而对被监测用户的行为作出判决. 在对每个 shell 命令短序列流 $\overline{S}^i$ 进行模式匹配与挖掘之后,可对相应的序列 $(class(\overline{S}^i_{l(i)}), class(\overline{S}^i_{l(i)+1}), \cdots, class(\overline{S}^i_{\bar{r}}))$ 进行加窗平滑处理,得到如下判决值:

$$D^i(j)=\frac{1}{e}\sum_{n=j-e+1}^{j} class(\overline{S}^i_n) \tag{2}$$

式中,e 为窗长度,$D^i(j)$ 表示 shell 命令短序列 $\overline{S}^i_j$ 对应的判决值,且 $e+l(i)-1 \leq j \leq \bar{r}$. $D^i(j)$ 反映了以

$\overline{S}_j^i$ 为终点的 e 个 shell 命令短序列中正常行为模式序列的比例. shell 命令短序列流 $\overline{S}^i=(\overline{S}_{l(i)}^i,\overline{S}_{l(i)+1}^i,\cdots,\overline{S}_{\bar{r}}^i)$ 中第 $e+l(i)-1$ 个短序列及其后面的每个短序列都分别对应一个判决值. 对于 W 个 shell 命令短序列流 $\overline{S}^1,\overline{S}^2,\cdots,\overline{S}^W$，我们可以并行计算出判决值 $D^1(j),D^2(j),\cdots,D^W(j)$，其中 j 的增长步长为 1.

(5) 利用判决值对被监测用户的当前行为进行判决.

在利用上述判决值对被监测用户的行为进行判决时，有以下两种方案可供选择：

第 1 种判决方案. 根据合法用户的行为特点，在判决值 $D^1(j),D^2(j),\cdots,D^W(j)$ 中选取一个 $D^k(j)$ 作为最终判决值(这里 $1\leq k\leq W$). 对于一次以 $\overline{R}=(\bar{s}_1,\bar{s}_2,\cdots,\bar{s}_{\bar{r}})$ 为审计数据的检测过程，当 $D^k(j)$ 中的 j 变化(增大)时，k 是固定不变的. 在利用 $D^k(j)$ 进行判决时，首先设定一个判决门限 a. 如果 $D^k(j)>a$，将被监测用户的当前行为判为正常行为；如果 $D^k(j)\leq a$，将被监测用户的当前行为判为异常行为(这里，被监测用户的“当前行为”是相对于 $\overline{R}=(\bar{s}_1,\bar{s}_2,\cdots,\bar{s}_{\bar{r}})$ 中 shell 命令 $\bar{s}_j$ 对应的时间点而言的，它是指以 $\bar{s}_j$ 为终点的 $e+l(k)-1$ 个 shell 命令所对应的行为，即 $\bar{s}_{j-e-l(k)+2},\bar{s}_{j-e-l(k)+3},\cdots,\bar{s}_j$ 所对应的行为). 在该方案中，$j\geq e+l(k)-1$；也就是说，被监测用户执行完第 $e+l(k)-1$ 个 shell 命令后才能对其行为做第一次判决.

第 2 种判决方案. 设定 W 个判决门限 $a_1,a_2,\cdots,a_W$，在并行计算出判决值 $D^1(j),D^2(j),\cdots,D^W(j)$ 之后，按照以下步骤对被监测用户的当前行为进行判决.

1. 设定 $k:=1$.

2. 如果 $D^k(j)\leq a_k$，则将被监测用户的当前行为判为异常行为，并不再执行以下步骤.

3. 如果 $k<W$，则 $k:=k+1$(k 的值增加 1)，并返回执行步 2. 如果 $k=W$，则执行步 4.

4. 如果 $D^k(j)\leq a_k$，则将被监测用户的当前行为判为异常行为；否则，将被监测用户的当前行为判为正常行为.

在第 2 种判决方案中，如果对于 $1\leq k\leq W$，存在一个 $D^k(j)$ 小于或等于 a_k，则将被监测用户的当前行为判为异常行为；如果对于 $1\leq k\leq W$，所有的 $D^k(j)$ 均大于 a_k，则将监测用户的当前行为判为正常行为(被监测用户的“当前行为”是相对于 $\overline{R}=(\bar{s}_1,\bar{s}_2,\cdots,\bar{s}_{\bar{r}})$ 中 shell 命令 $\bar{s}_j$ 对应的时间点而言的，它是指以 $\bar{s}_j$ 为终点的若干个 shell 命令所对应的行为). 在该方案中，$j\geq e+l(W)-1$. j 的值每增加 1，都要执行以上步骤来对被监测用户的当前行为做一次判决.

对于第 1 种判决方案，在窗长度确定的情况下，虚警概率和检测概率不仅受判决门限的影响，而且还取决于最终判决值 $D^k(j)$，如何从 W 个参考判决值 $D^1(j),D^2(j),\cdots,D^W(j)$ 中选取最佳的最终判决值 $D^k(j)$ 是第 1 种判决方案的关键. 根据现有的研究成果和我们的检测实例分析，对于不同的合法用户，由于行为特点不尽相同，最佳的判决值 $D^k(j)$ 往往是不同的，而且，对于同一合法用户，针对不同伪装用户异常行为的最佳判决值 $D^k(j)$ 也是不同的，所以，在实际应用中不太容易确定最佳的 $D^k(j)$，特别是在对伪装(异常)行为先验知识缺少了解的情况下. 相比之下，第 2 种判决方案对不同合法用户和不同伪装(异常)行为具有较强的适应性，多模式并行挖掘和多门限联合判决保证了在用户行为复杂多变的情况下也能够获得很高的检测准确率，其代价是一定程度上增加了检测阶段的计算成本.

5　参数确定与特点分析

对于检测阶段的两种判决方案，如何选择判决门限是实际应用中的关键问题. 我们可参照文献[7]中交叉验证的方法来确定第 1 种判决方案中的判决门限 a，将获得的正常行为训练数据反复交叉地按固定比例分成两部分，一部分用于训练，另一部分用于测试虚警概率和判决门限，在测试中通过调整判决门限来得到不同虚警概率与不同判决门限的对应关系，并将期望虚警概率所对应的判决门限作为实际检测时的门限.

第 2 种判决方案中的各个判决门限 $a_1,a_2,\cdots,a_W$ 的选择相对复杂，可基于第 1 种判决方案中交叉验证的方法并通过等量迭代逼近的方式确定. 具体步骤如下：

1. 设定 $k:=1$，并设定期望虚警概率的上限 q.

2. 使用 $D^k(j)$ 作为第 1 种判决方案的最终判决值，并令 $q(k)=q/W$(这里 $q(k)$ 表示 $D^k(j)$ 作为最终判决值时此种判决方案所容忍的最大虚警概率)，然后利用获得的训练数据通过交叉验证的方法来得到第 1 种判决方案下期望虚警概率 $q(k)$ 所对应的判决门限 $p(k)$.

3. $k:=k+1$(k 的值增加 1). 如果 $k\leq W$，返回执行步 2. 如果 $k>W$，执行步 4.

4. 设定 $i:=1$. 对于 $1\leq k\leq W$，令 $a_k(i)=p(k)$.

5. 设定判决门限的增量 b.

6. $i:=i+1$. 对于 $1\leq k\leq W$，令 $a_k(i)=p(k)+b$.

7. 将 $a_1(i),a_2(i),\cdots,a_W(i)$作为第 2 种判决方案中的 W 个判决门限(其中 $a_k(i)$与 $D^k(j)$相对应),然后利用训练数据通过交叉验证的方法测试第 2 种判决方案的虚警概率 $u(i)$.

8. 如果 $u(i)>q$,则对于 $1\leq k\leq W$,令 $a_k=a_k(i-1)$. 如果 $u(i)=q$,则对于 $1\leq k\leq W$,令 $a_k=a_k(i)$. 如果 $u(i)<q$,则返回执行步 6.

基于以上方法确定两种判决方案的判决门限,我们可以对虚警概率进行控制(训练数据越充分,对虚警概率的控制就越精确),而实际检测中对伪装(异常)行为的检测概率则受多种因素影响. 此外,窗长度 e 也是一个重要参数,e 越大,检测准确率往往越高,但判决延迟也越大;根据实例分析和同类检测方法的实验结论,一般将 e 控制在 50～150 之间为宜.

需要指出,在实时检测(在线检测)的情况下,检测阶段中被监测用户所执行的 shell 命令的获取和预处理、shell 命令短序列流的生成、行为模式匹配与挖掘、判决值的计算以及对用户行为的判决都是同步进行的. 当获得审计数据 $\bar{R}=(\bar{s}_1,\bar{s}_2,\cdots,\bar{s}_r)$中的第 j 个 shell 命令 $\bar{s}_j$之后,就可以生成以 $\bar{s}_j$为终点的长度分别为 $l(1),l(2),\cdots,l(W)$的 W 个 shell 命令短序列 $\bar{S}_j^1,\bar{S}_j^2,\cdots,\bar{S}_j^W$,然后进行上述的序列匹配,计算出判决值 $D^1(j),D^2(j),\cdots,D^W(j)$,进而对被监测用户的当前行为做一次判决.

6 实验设计与结果分析

作者利用普度大学和 AT&T Shannon 实验室的两组 shell 命令实验数据分别对以上方法的性能进行了实验. 普度大学的数据包含 8 个 Unix 用户在两年时间内的活动记录(实验数据的详细说明见文献[9]或文献[2]). 实验中采用了其中的 4 个用户 user1、user2、user3、user4 的数据,并且将 user1、user2、user3 设为伪装用户,将 user4 设为合法用户. 每个用户的 shell 命令流中各有 15000 个命令,user4 的前 10000 个命令作为训练数据用于正常行为建模,而每个用户的后 5000 个命令作为测试数据用于性能测试. 实验的参数设置为 $W=3,C=\{1,2,3\}$,$minsup(1)=minsup(2)=minsup(3)=0.0002$,窗长度 $e=151$. 在采用第 1 种判决方案时,使用 $D^3(j)$作为最终判决值,判决门限 $a=0.5$. 在采用第 2 种判决方案时,3 个判决门限分别为 $a_1=0.86,a_2=0.64,a_3=0.48$. 为了方便与同类方法进行检测准确度的比较,实验中将上述两种判决方案的期望虚警概率分别设定为 0.06%和 0%,在此基础上确定相应的判决门限,然后根据判决值分别计算两种判决方案的检测概率.

在 AT&T Shannon 实验室的 shell 命令实验数据中(该数据的详细说明见文献[5]或文献[7]),我们选择前 4 个用户 user1、user2、user3、user4 的数据进行实验,每个用户有 5000 个 shell 命令,实验时将其中 user4 设为合法用户,该用户的前 4000 个命令作为训练数据用于正常行为建模,后 1000 个命令作为测试数据用于测试虚警概率;其他 3 个用户设为伪装用户,其中每个用户的 5000 个 shell 命令均作为测试数据用于测试检测概率. 实验的参数设置为 $W=3,C=\{1,2,3\}$,$minsup(1)=minsup(2)=minsup(3)=0.0005$,窗长度 $e=100$. 在采用第 1 种判决方案时,使用 $D^3(j)$作为最终判决值,判决门限 $a=0.35$. 在采用第 2 种判决方案时,3 个判决门限分别为 $a_1=0.62,a_2=0.37,a_3=0.34$.

图 1 给出了采用普度大学数据进行实验时判决值 $D^3(j)$的曲线,图中上方的细实线为合法用户 user4 对应的曲线,下方的两条虚线和粗实线分别是伪装用户 user1、user2、user3 对应的曲线. 图 2 给出了采用 AT&T Shannon 实验室数据进行实验时判决值 $D^3(j)$的曲线. 可以看出,两个图中合法用户对应的判决值曲线同伪装用户对应的判决值曲线具有良好的可分性. 在图 1 中,合法用户 user4 对应的判决值全部大于 0.48,而伪装用户 user3 对应的判决值全部小于 0.57,只有 user1、user2 对应的少量判决值在 0.57 之上. 在图 2 中,合法用户 user4 的判决值曲线能够同伪装用户 user2、user3 的判决值曲线很好地区分开来(仅有很少量的判决值出现交叠),但同 user1 判决值曲线的可分性稍差一些,说明两者的某些行为具有较高的相似度.

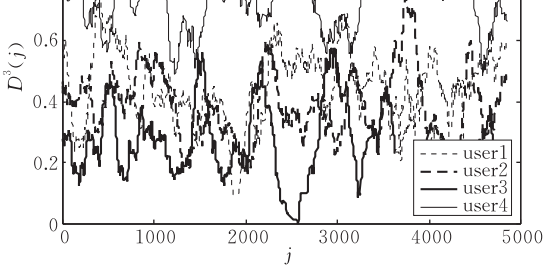

图 1 普度大学实验数据对应的判决值曲线

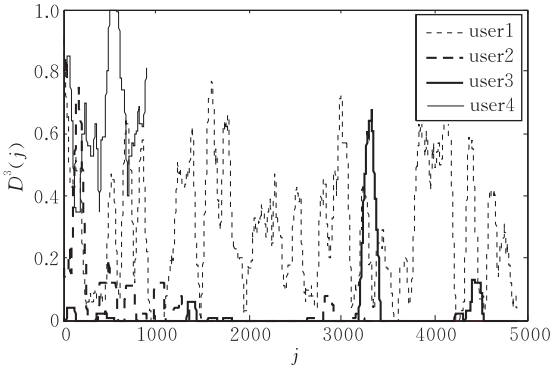

图 2　AT&T Shannon 实验室数据对应的判决值曲线

同时，作者还利用以上两组 shell 命令实验数据分别对文献[4]中基于支撑向量机的检测方法、文献[9]中基于机器学习的检测方法以及文献[2]中基于齐次 Markov 链模型的检测方法进行了实验，并同本文的方法在检测准确率和计算成本两个方面进行了对比. 表 1 给出了利用普度大学 shell 命令数据所做实验的结果.

表 1　普度大学数据对应的实验结果

性能指标	虚警概率/%	检测概率/%	实验时间/s
本文第 1 种判决方案的实验结果	0.06	80.53	587
本文第 2 种判决方案的实验结果	0	83.02	711
文献[4]中检测方法的实验结果	0.16	74.08	510
文献[9]中检测方法的实验结果	0.52	75.81	3146
文献[2]中检测方法的实验结果	0.33	75.93	465

表 2 给出了利用 AT&T Shannon 实验室 shell 命令数据所做实验的结果.

表 2　AT&T Shannon 实验室数据对应的实验结果

性能指标	虚警概率/%	检测概率/%	实验时间/s
本文第 1 种判决方案的实验结果	0.04	85.26	135
本文第 2 种判决方案的实验结果	0	84.91	173
文献[4]中检测方法的实验结果	0.32	80.87	120
文献[9]中检测方法的实验结果	0	83.60	934
文献[2]中检测方法的实验结果	0.19	81.92	108

根据表 1 和表 2 的实验结果，本文所提出方法的检测准确率明显高于文献[2,4,9]中的检测方法，而第 2 种判决方案的检测结果又优于第 1 种判决方案. 表中的实验时间是指实验中进行训练(正常行为建模)和检测所需要的时间，该指标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. 由实验结果可见，采用第 2 种判决方案时本文方法的计算成本略高于文献[2,4]中的检测方法，但远低于文献[9]中的检测方法. 可见，本文中检测方法的综合性能优于文献[2,4,9]中的检测方法. 此外，我们还利用普度大学和 AT&T Shannon 实验室两组数据中不同用户的 shell 命令数据进行了反复交叉实验，实验结果表明，当采用第 1 种判决方案时，最佳的判决值 $D^k(j)$ 是与具体用户相关的，因此实际检测中用于行为判决的最佳 $D^k(j)$ 主要基于已有的训练数据并通过预先实验来确定；而第 2 种判决方案对不同用户具有更好的适应性，总体上也有更高的检测性能. 为了进一步验证第 2 种判决方案的检测准确率，我们在上述实验的基础上对判决门限进行了调整，得到不同虚警概率条件下的检测概率. 图 3 给出了采用普度大学数据进行实验时反映虚警概率与检测概率对应关系的 ROC 曲线，图 4 给出了采用 AT&T Shannon 实验室数据进行实验时的 ROC 曲线. 从两组实验数据对应的 ROC 曲线可以看出，相对于文献[2,4,9]中的检测方法，本文方法中第 2 种判决方案的检测准确率具有明显的优势.

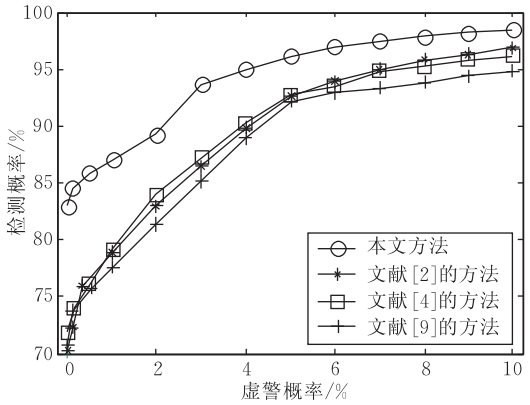

图 3　普度大学实验数据对应的 ROC 曲线

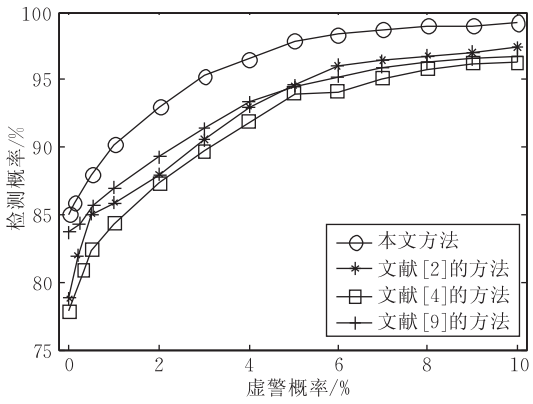

图 4　AT&T Shannon 实验室数据对应的 ROC 曲线

7　结束语

目前的伪装攻击检测方法存在的主要问题是对用户行为变化缺乏适应性，检测性能的稳定性和容

错能力不够强,检测准确度也有待提高.本文提出一种新的基于 shell 命令和多重行为模式挖掘的伪装攻击检测方法,该方法基于实际检测系统的实例分析和已有研究成果,充分考虑了审计数据和用户行为的特点,改进了用户行为模式和行为轮廓的表示方式,提出了基于多重行为模式并行挖掘和多门限联合判决的检测模型,在不明显增加计算成本的条件下大幅度提高了检测准确度,并且增强了检测性能的稳定性和对用户行为变化的适应性.需要指出,在该方法的实际应用中,通过优化行为模式序列的表示及存储方式,还可以进一步提高其检测效率.

参 考 文 献

[1] Tian Xin-Guang, Duan Mi-Yi, Sun Chun-Lai, Li Wen-Fa. Intrusion detection based on system calls and homogeneous Markov chains. Journal of Systems Engineering and Electronics, 2008, 19(3): 598-605

[2] Tian Xin-Guang, Duan Mi-Yi, Li Wen-Fa, Sun Chun-Lai. Anomaly detection of user behavior based on shell commands and homogeneous Markov chains. Chinese Journal of Electronics, 2008, 17(2): 231-236

[3] Gao D, Retier M K, Song D. Behavioral distance measurement using hidden markov models//Proceedings of the Conference on Recent Advanced in Intrusion Detection. Hamburg, Germany, 2006: 19-40

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

[5] Szymanski B K, Zhang Y Q. Recursive data mining for masquerade detection and author identification//Proceedings of the 5th IEEE System, Man and Cybernetics Information Assurance Workshop. West Point, NY, USA, 2004: 424-431

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

[7] Schonlau M, Mouchel W. Computer intrusion: Detecting masquerades. Statistical Science, 2001, 16(1): 58-74

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

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

[10] Tian Xin-Guang, Gao Li-Zhi, Sun Chun-Lai, Duan Mi-Yi, Zhang Er-Yang. A method for anomaly detection of user behaviors based on machine learning. The Journal of China Universities of Post and Telecommunications, 2006, 13(2): 61-65,78

[11] Lian Yi-Feng, Dai Ying-Xia, Wang Hang. Anomaly detection of user behaviors based on profile mining. Chinese Journal of Computers, 2002, 25(3): 325-330(in Chinese)
(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)

[12] Tian Xin-Guang, Sui Jin-Guo, Li Xue-Chun. A system and its method for anomaly detection of user behavior based on machine learning . Chinese Patent, ZL200510056934. 2005-03-23(in Chinese)
(田新广, 隋进国, 李学春. 基于机器学习的用户行为异常检测系统与方法. 中国专利, ZL200510056934. 2005-03-23)

[13] Ye N, Emran S M, Chen Q. Multivariate statistical analysis of audit trails for host-based intrusion detection. IEEE Transactions on Computers, 2002, 51(7): 810-820

[14] Tian Xin-Guang, Gao Li-Zhi, Sun Chun-Lai, Zhang Er-Yang. Anomaly detection of program behavior based on system calls and Markov chains. Journal of Computer Research and Development, 2007, 44(9): 1538-1544(in Chinese)
(田新广, 高立志, 孙春来, 张尔扬. 基于系统调用和齐次 Markov 链模型的程序行为异常检测. 计算机研究与发展, 2007, 44(9): 1538-1544)

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

TIAN Xin-Guang, born in 1976, Ph. D., associate researcher. His research interests include network security, intrusion detection, and intelligent information processing.

DUAN Mi-Yi, born in 1953, Ph. D., professor, Ph. D. supervisor. His main research interests include computer application and intrusion detection.

CHENG Xue-Qi, born in 1971, Ph. D., professor, Ph. D. supervisor. His main research interests include information security, network information retrieval, and P2P computing.

Background

The research of this paper is supported by the National High Technology Research and Development Program (863 Program) of China under grant No. 2006AA01Z452, National Information Security 242 Program of China under grant No. 2005C39 and National Grand Fundamental Research 973 Program of China under grant No. 2004CB318109. These programs aim to develop a distributed intrusion detection system for confidential networks to detect attacks and suspicious activities both at the network level and at the host level, and concentrate on anomaly detection of masquerade attacks. The authors have focused their work on the development of the distributed intrusion detection system for a long while, and have published several papers in international conferences and journals. In recent years, many computationally sophisticated methods for anomaly detection of masquerade attacks have been developed, but there are few well-accepted methods in widespread use. This paper presents a novel method to distinguish legitimate users from masqueraders. The method characterizes behavioral patterns of legitimate users with shell command sequences of different lengths, and employs hierarchical sequence supports to construct the normal behavior profiles of legitimate users. In the detection stage, a model based on multiple sequence pattern parallel mining and multiple threshold joint decision is used to determine whether the monitored user's behavior is normal or anomalous. The model gives attention to both detection accuracy and computational efficiency, and is especially applicable for on-line detection. The novel method has been applied to practical hosted-based intrusion detection systems in practical networks, and achieved high detection performance.