一种新型的入侵检测模型的研究与实现*)

The Study and Implement of a Novel Intrusion Detection Model

秦 拯[1] 吴中福[1] 廖晓峰[1] 李 华[1] 王 康[1] 吴李瀚[2]

(重庆大学计算机学院 重庆400044)[1] (重庆市车管所)[2]

Abstract To overcome some of the limitations in the existing models on intrusion detection, a new approach using a neural network based on the backpropagation learning algorithm is proposed for intrusion detection, and the results obtained on preliminary testing are displayed.

Keywords Neural network, Intrusion detection, BP learning algorithm

1 引言

入侵(Intrusion)是指有关试图破坏资源的完整性、机密性及可用性的活动集合[1]。入侵检测(Intrusion Detection,简称ID)是指识别谁在未经授权情况下使用计算机系统;哪些人在滥用他们的特权对系统进行非法的访问[2]。入侵检测系统(IDS)是一个计算机系统(可能由软件、硬件构成),它试图履行入侵检测功能,正如刚才所定义的,大多数是实时的[3]。

对于IDS的评估,主要的性能指标有:①可靠性,系统具有容错能力和可连续运行;②可用性,系统开销要最小,不会严重降低网络系统性能;③可测试,通过攻击可以检测系统运行;④适应性,对系统来说必须是易于开发的,可添加新的功能,能随时适应系统环境的改变;⑤实时性,系统能尽快地察觉入侵企图以便制止和限制破坏;⑥准确性,检测系统具有低的误警率和漏警率;⑦安全性,检测系统必须难于被欺骗和能够保护自身安全;⑧可集成性,检测系统必须易于同其他安全产品或网络系统进行集成,还可能需要在不同IDS之间相互合作。

传统检测方法和手段存在许多不足[4],难以满足IDS所需要的实时性、适用性、可用性、可靠性、准确性和集成性等方面的需求。研究表明,神经网络(Neural Network,以下简称NN)方法适合于入侵检测,首先,神经计算是一种数值计算,能满足实时性需要,若附加分接器(Tap),则可较好地解决可用性;神经网络具有容错性、鲁棒性、对噪声的容忍特性,勿需统计假设,还可利用其学习能力,使得它能满足IDS所需要的可靠性、准确性、适应性,集成性则通过考虑标准化来解决。本文提出一个利用NN进行入侵检测的新方法,以克服现存ID模型的一些不足。

2 IDS系统的形式化描述

通用的IDS系统可以形式化描述为由以下几部分组成:①用户集:$U=\{u_1,u_2,\cdots,u_m\}$;②被用户执行的命令集:$C=\{c_1,c_2,\cdots,c_n\}$;③由可能发生的入侵组成的有限集合:$I=\{i_1,i_2,\cdots,i_p\}$;④已识别的入侵序列:$R=\{r_1,r_2,\cdots,r_q\}$;⑤尚未识别的入侵序列:$N=\{n_1,n_2,\cdots,n_s\}$。

用户命令集中的命令来源于审计记录或网络管理软件。可将命令集进一步划分为2类,第一类命令集C_1中的命令由单个用户执行就可导致入侵,描述如下:

$$C_1=\{u_i(C)\}$$

式中$u_i(C)$是用户u_i执行的命令集,这些命令来自于C。

第二类命令集C_2中的命令由协作用户执行而导致入侵。这些命令收集在SMIB(the Security Management Information Base)中。显然,$C_2\subset C$。

任意入侵要么属于R,要么属于N,于是有

$$I=R\cup N$$

换句话说,

$$I=\{g(\{c_k\})\mid((\{c_k\}\subset C_1)\vee(\{c_k\}\subset C_2))\wedge((g(\{c_k\})\in R)\vee(g(\{c_k\})\in N))\}$$

式中$g(\cdot)$是命令集$\{c_k\}$与入侵集的关联函数.

3 方法

通过对已知入侵进行研究,分析特定环境下入侵的特点,尤其是分析得到构成入侵的命令事件。本文提出的ID模型的基本思想就是通过分析命令事件集来判断是否发生入侵、发生何种入侵。实时捕获或搜索来

*)本课题受重庆市科技攻关项目支持。

的命令存放在SMIB中，送入NN中进行快速分析。

对所提出的基于NN的ID模型分两步进行研究。第一步，离线获得专家知识，形成一系列规则。第二步，基于归纳法原则，允许系统在线工作，学习任何可能的未曾遇到过的入侵。事件发生顺序在基于NN的系统中没有什么意义。这样，由协作用户导致的事件，与由单个用户在不同时间触发的事件，可以一起处理，以预测入侵。

4 神经网络模型

基于NN的ID模型采用前向神经网络，并基于成熟且已得以广泛应用的BP算法。NN模型如图1所示。

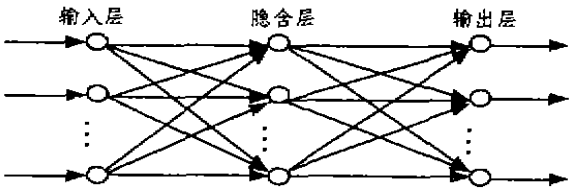

图1 基于NN的ID模型

该NN有一个输入层，接受二进制输入信号。这些二进制输入信号对应于已经出现在SMIB中的相关事件(对应用户集$\{u_k\}$的命令集$\{c_i\}$)。

NN的输出层用来指示可能的入侵。

NN可以有一个或多个隐含层，这要根据问题的复杂性(即相关事件的数量、规则数量和入侵数量)。隐含层神经元的数目则取决于训练用的样本数以及经验积累。

NN的每一层由一个或多个神经元构成，上一层的输出作为下一层的输入，每层的神经元同下一层的神经元相连，并赋以合适的权值。

对神经元i的网络输入x_i为：

$$x_i=\sum_j(w_{ji}*y_j)$$

式中，w_{ji}：连接神经元j(位于前一层)到i(位于当前层)之间连接权值；y_j：来自神经元j的输入。

位于前一层偏置神经元的输入置为-1，而权值可以调节，实质上，位于前一层偏置边上的权值表示在当前层上每个神经元的门槛值。

每个神经元的输出由该神经元的网络输入通过Sigmoid函数(又称为激活函数)$f(\cdot)$变换得到。

我们采用以下激活函数：

$$f(x)=(1+e^{-x})^{-1}$$

神经网络的训练有两个过程：前向过程和反向过程。前向过程是指对于给定的输入和目前的权值来估计神经网络的输出值。在反向过程中，将求得的神经网络输出值与期望输出值相比较，并将比较所得差别作为误差输回到神经网络中，以调整神经网络的权值。

在反向过程中，假设对于输出层K上的某一神经元i，其期望的输出值为d_i^K，求得的输出值为y_i^K，则反向传播误差δ_i^K为：

$$\delta_i^K=f'(x_i^K)[d_i^K-y_i^K]$$

式中x_i^K：第L层上第i个神经元的网络输入；$f'(\cdot)$：激活函数$f(\cdot)$的导数。

对于层$l=1,\cdots,(K-1)$，l层上的每个神经元i，误差值δ_i^l由下式求得：

$$\delta_i^l=f'(x_i^l)\sum_j(w_{ji}^{l+1}*\delta_j^{l+1})$$

K层上的神经元i和$(l-1)$层上的神经元j之间的连接权的修正值由下式确定：

$$w_{ji}^l=w_{ji}^l+\eta*\delta_i^l*y_j^{l-1}$$

式中，η：学习速率，依赖于神经网络的期望收敛速度。

该神经网络算法实质上是采用梯度下降法使神经网络的输出与期望输出之间的均方误差达到最小。

当满足下列条件时，神经网络可由离线模式转到在线模式：

①对于所有训练模式，当期望的输出与神经网络的实际输出相同时；

②当神经网络的实际输出与期望输出值之间的均方误差小于指定的门槛值(例如，当每单元输出值的均方误差小于0.01时)；

③当梯度(递减率，gradient)足够小时(根据定义，梯度最小值为0)。

入侵检测算法：

Begin

①构造由$1,2,\cdots,L$层组成的神经网络，并对每一层选择与问题相适应的神经元数目。

②将每一层上的偏置神经元设为-1.0。

③$w_{ij}=\delta$，对于任意i,j，δ是在$(0,1)$之间的随机值。

④若IDS是离线模式，则从样本集(序列)中选择一种输入模式；反之，若是在线模式，则从SMIB中选择相关事件作为输入。

⑤通过神经网络将信号传播到输出层。

⑥对输出层K上的每个神经元i，计算反向传播误差δ_i^K：

$$\delta_i^l=f'(x_i^l)\sum_j(w_{ji}^{l+1}*\delta_j^{l+1})$$

⑦对于层$l=1,\cdots,(K-1)$，其上的每个神经元i的反向传播误差δ_i^l由下式计算得到：

$$\delta_i^K=f'(x_i^K)[d_i^K-y_i^K]$$

⑧使用下式修正权值：

$$w_{ji}^l=w_{ji}^l+\eta*\delta_i^l*y_j^{l-1}$$

上式中，η为学习速率。

⑨若输出层的误差 $\sum_i [d^k - y^k]^2$ 在一个允许的范围内，则从离线操作模式转到在线操作模式。

⑩转至步骤4，直到 IDS 变得灵敏。

End

5 IDS 实例

下面基于 LINUX 系统，以实例来说明所提出的 ID 模型的特点。假设模型所需事件来自以下三种情况：①用户登录时；②访问网络资源；③访问跨域资源。

在本例中，IDS 考虑以下一些事件。若这些事件出现了，则第1层相应神经元的输入为"1"，否则为"0"。

①对远程连接，rlogin/telnet/ftp 被拒绝。

②rlogin/telnet/ftp 被拒绝，原因是，根据/etc/hosts.deny 规定，不允许访问远程主机。

③rlogin/telnet/ftp 被拒绝，在这里，远程主机可能允许被访问，但是远程主机上的用户入侵企图失败了。

④(在根帐户下)cp/bin/sh usr/root 被执行。(root shell 被拷贝到用户区)

⑤(在根帐户下)chmod 4755 usr/root 被执行。(通过执行该命令，允许用户使用 root shell)。

⑥ usr/root 被执行(用户正在使用 root shell)。

⑦ 用命令 cd 将目录改到不相关的组。

⑧ 试图删除其他用户区的文件。

⑨ 执行 su 命令成功。

⑩执行 su 命令失败。

⑪print/fax 拒绝对用户服务。

⑫print/fax 拒绝对越权用户提供服务。

相应的入侵有：

①远程主机企图建立一个连接。

②非授权系统的非授权用户试图登录。

③root shell 被拷贝。

④root shell 被拷贝且被执行。

⑤使用代理 root shell，目录被更改。

⑥使用代理 root shell，文件被删。

⑦su 成功，跟踪后续命令。

⑧su 失败，确定发出 su 命令的用户及其所在终端。

⑨用户被拒绝访问 print/fax。

⑩用户越过 print/fax 权限，采取适当的措施。

表1为 IDS 的初始化知识，用12条规则来表示，表中描述了 IDS 的16个事件和10种入侵。

在表1中，对应与协作用户入侵的命令是样本4～6，这些命令是从 SMIB 中收集来的，而不是来自每个用户的记录。其他样本(1—3，7—12)依据用户方式进行收集，并作为输入模式送进 NN 进行计算，产生相应输出。

表1 IDS 的初始化知识

序 号	事 件	入 侵
1	1110000000000000	1100000000000000
2	1100000000000000	1000000000000000
3	1000000000000000	1000000000000000
4	0001000000000000	0010000000000000
5	0001100000000000	0010000000000000
6	0001110000000000	0001000000000000
7	0001111000000000	0001000000000000
8	0001110100000000	0001010000000000
9	0000000010000000	0000001000000000
10	0000000001000000	0000000100000000
11	0000000000100000	0000000010000000
12	0000000000010000	0000000001000000

表2 新规则

序 号	事 件	入 侵
13	0000000000011000	0000000000010000

6 实例说明

对于表1中样本 No.6，事件④、⑤、⑥，即(cp/bin/sh usr/root)、(chmod 4755 usr/root)和(usr/root)被设为"1"，而其他事件相应值被设为"0"。此时会产生入侵④：root shell 被拷贝且被执行。进一步分析发现，在样本 No.7和样本 No.8中，都会产生多于1种的入侵。对于样本 No.7，事件④、⑤、⑥、⑦会产生入侵④和⑤；而样本 No.8中，事件④、⑤、⑥、⑧会导致入侵④和⑥。

6.1 规则的增添与删除

假设 IDS 碰到了某事件集，需要建立新的入侵集。由于 NN 是在在线方式下训练的，如果这种事件经常发生，这种事件——入侵组可以作为新的规则。同样，如果一条现有规则不再有用，超过规定时间，NN 将不再学习它。

NN 模型具有冗余的输出和输入神经元，能够适应新的事件——入侵的出现。如表1所示，尽管有12个事件和10种入侵，但还是让 NN 具有16个输入神经元、15个输出神经元和6个隐含神经元(由实验确定)，表2给出了增添新规则的示例。

新规则中，新的事件⑬和新的入侵⑪也被相应加入。

事件⑬：print/fax 命令发送太多。

入侵⑪：跟踪用户。

6.2 实验结果

NN 的输入层和输出层分别有16个和15个神经元。在 Intel Celeron 550微机上用 C++语言开发 NN 模型，工作站的操作系统为 LINUX。

建立NN模型主要需要考虑两方面的问题：一是确定网络的拓扑结构，二是学习参数的调整。对于一个特定的分类问题并没有一个规范的模式来构造网络结构，通常通过实验方法来得到一个适合的网络结构。Lippman在1987年已经证明，对于任意复杂的判别函数，一个三层的BP网络（即包含一个隐含层）就足够了。隐含层的神经元数目的确定较为复杂，并无确定的法则，只能通过一些经验法则，借助于实验来确定。一般来说，可考虑的经验法则有：①隐含层节点数不能是各层中节点数最少的，也不是最多的；②隐含层节点数 n_h 的确定有一些经验公式，本文采用经验公式 $n_h=\sqrt{n_i-n_o}+\beta$，式中 n_i 为输入节点数，n_o 为输出节点数，常数 β 为正整数，一般 $1\leqslant\beta\leqslant10$；③隐含层节点数的理论上限由其训练样本数据所限定。因此，采用试值法测试隐含层节点数对网络性能的影响。另外，学习参数的调整也只能采用试值法。

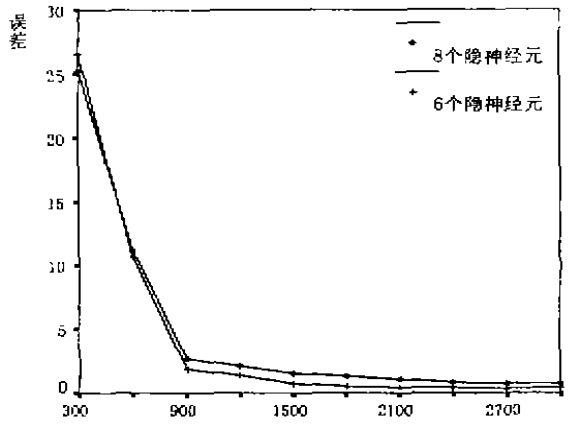

图2 $\eta=0.1$，隐含神经元数量分别为6和8时的误差情况

我们选择以下参数来考察NN的收敛性：①隐含层神经元分别采用6个和8个；②学习率 η 分别采用0.1和0.2。

对于上述每种情况，我们均采用表2给出的样本，对NN进行反复训练，当迭代次数为300的倍数时，计算出NN的误差情况。计算出的误差为，当迭代次数为300的倍数时，用12种规则训练得到的输出层所有神经元的输出值与期望值的误差平方和。实验表明，当迭代次数约大于800次时，NN开始收敛。

如图2所示，当学习率 $\eta=0.1$、迭代次数大于500时，隐含层神经元数目为6的NN模型优于隐含层神经元数目为8的NN模型，即误差要小。图3描述了当隐含层神经元数目为6，学习率分别为 $\eta=0.2$、$\eta=0.1$ 时的误差情况。图中曲线表明，$\eta=0.2$ 时，NN的收敛速度要快一些；迭代次数大于2700时，计算得到的误差（用12种规则训练得到的输出层所有神经元的输出值与期望值的误差平方和）约为0.03。

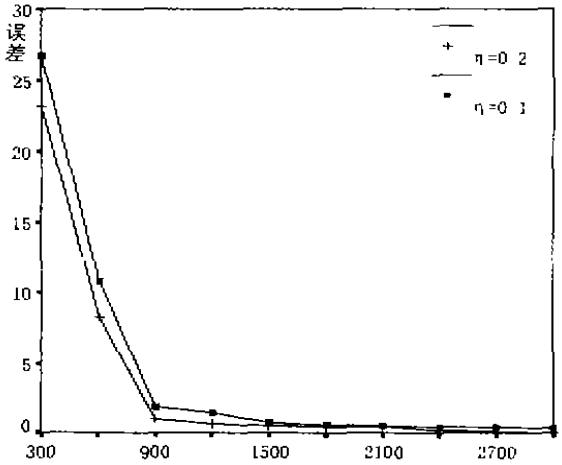

图3 隐含神经元数量为6，η 分别为0.1和0.2时的误差情况

结论 为克服传统ID模型的一些不足，本文提出了一种新的ID模型，采用神经网络方法。实验结果表明，该模型具有很好的收敛性，运算速度快，误差小，适于实时检测。可以通过增删"事件-入侵对"，方便地定制出适于特定环境的ID模型，从而形成系列化的IDS产品。对于某特定环境，可以结合入侵的分类和安全策略，找出各类别中由单用户命令或协作用户命令造成的入侵，再采用本文的方法，即可定制相适应的IDS。另外，在设计神经网络的拓扑结构时，保留了冗余的神经元，以扩展模型的检测范围。

参考文献

1 Spafford E. Crisis and aftermath. Communications of the ACM, 1989, 32(6): 678～687

2 刘美兰，姚京松. 审计跟踪与入侵检测 计算机工程与应用，1999(7)

3 http://all.net/books/dca/top.html

4 Lippmann R P, Cunningham. Improving intrusion detection performance using keyword selection and neural network. Computer Networks, 2000, 34