基于蜜罐的入侵检测系统的设计与实现 *

汪 洁[1], 杨 柳[2]
(1. 中南大学 信息科学与工程学院, 长沙 410083; 2. 湖南大学 软件学院, 长沙 410082)

摘 要: 传统的入侵检测系统无法识别未知的攻击,提出在入侵检测系统中引入蜜罐技术来弥补其不足,并设计和实现了一个基于人工神经网络的入侵检测系统 HoneypotIDS。该系统应用感知器学习方法构建 FDM 检测模型和 SDM 检测模型两阶段检测模型来对入侵行为进行检测。其中,FDM 检测模型用于划分正常类和攻击类,SDM 检测模型则在此基础上对一些具体的攻击类型进行识别。最后,设计实验对 HoneypotIDS 的检测能力进行了测试。实验结果表明,HoneypotIDS 对被监控网络中的入侵行为具有较好的检测率和较低的误报率。

关键词: 入侵检测; 蜜罐; 感知器
中图分类号: TP391.4 **文献标志码**: A **文章编号**: 1001-3695(2012)02-0667-05
doi:10.3969/j.issn.1001-3695.2012.02.071

Design and implementation of intrusion detection system based on honeypot

WANG Jie[1], YANG Liu[2]
(1. *School of Information Science & Engineering, Central South University, Changsha* 410083, *China*; 2. *School of Software, Hunan University, Changsha* 410082, *China*)

Abstract: The traditional IDS (intrusion detection system) can not identify the unknown attacks. Therefore, this paper introduced honeypot technique into the IDS. It desigaed a intrusion detection system based on ANN (artificial neural network). It constructed the system contained FDM detection model and SDM detection model by using perceptron learning method. FDM was used to distinguish the attack class from the normal class, while the other focused on detecting some main types of attacks. At last, an experiment was to test detection ability of HoneypotIDS. The results of the experiment show that HoneypotIDS has a better detection rate and a lower false positive rate for the intrusion activities in the monitored network.

Key words: intrusion detection; honeypot; perception

目前黑客开发攻击工具越来越容易,且破坏能力越来越强[1],这给网络安全带来了巨大的危害。网络的安全性已经成为阻碍 Internet 发展的重要因素之一。为此,人们不断提出各种安全防御技术来保护网络的安全。传统的静态安全防御技术包括防火墙、漏洞扫描、加密和认证等,它们从各种角度保障系统的安全,但是对网络环境下日新月异的攻击手段缺乏主动的反应。因此,人们又提出了以入侵检测技术为核心的动态主动防御技术。入侵检测系统通过收集操作系统、系统程序、应用程序、网络包等信息,发现系统中违背安全策略或危及系统安全的行为[2]。IDS 作为防火墙之后的第二道安全防线,能够对网络和操作系统所遭受的外部攻击、内部攻击以及误操作进行全面的检测。其主要功能是监视并分析用户和系统的行为、评估重要系统和数据文件的完整性、识别已知攻击、对异常行为模式进行统计分析、对操作系统进行审计跟踪管理等[3]。

传统的 IDS 极大地提高了系统和网络的安全性,但是它们仍然存在其固有的缺点,如误报率和漏报率较高、缺乏对未知攻击的检测能力。在 IDS 中引入蜜罐可以较好地解决其存在的缺陷。蜜罐[4]技术是一种主动的网络保护技术,其研究如何设计一个严格的欺骗环境(受控网络、主机或软件模拟的网络和主机),诱骗入侵者对其进行攻击或在其遭受攻击后作出预警,从而保护实际运行的网络和系统。蜜罐可以检测到未知的攻击,并且收集入侵信息,借以观察入侵者行为,记录其活动,以便分析入侵者的水平、目的、所用工具和入侵手段等[5]。蜜罐技术和入侵检测技术的结合,不但有可能减少传统入侵检测系统的漏报和误报,而且还能识别未知的攻击,极大地增强了检测能力。因此,研究和实现基于蜜罐的入侵检测技术具有非常重要的意义。

本文设计并实现一个基于蜜罐的入侵检测系统 HoneypotIDS。该系统充分利用蜜罐可用于检测的特点,对入侵行为具有较好的检测能力。HoneypotIDS 包含一个由人工神经网络构成的入侵检测模型 IDSM。并在被监控网络中配置多个蜜罐,从这些蜜罐中收集数据,然后使用 IDSM 分析一段时间内蜜罐中大量的全局信息来识别入侵行为。

1 HoneypotIDS 的总体设计

HoneypotIDS 主要是通过在被监控网络中配置多个蜜罐并从中收集数据,然后使用检测模块来识别入侵行为。这些检测模型是通过分析一段时间内蜜罐中大量全局信息来进行检测的。HoneypotIDS 有三个模块,分别是数据收集模块、数据预处理模块和入侵检测模块。其结构如图 1 所示。

收稿日期: 2011-07-25; **修回日期**: 2011-08-31 **基金项目**: 中南大学自由探索计划资助项目(2011QNZT035)

作者简介: 汪洁(1980-),女,湖南桃江人,博士,主要研究方向为网络安全(jwang@mail.csu.edu.cn);杨柳(1979-),女,博士,主要研究方向为数字图像处理和网络通信.

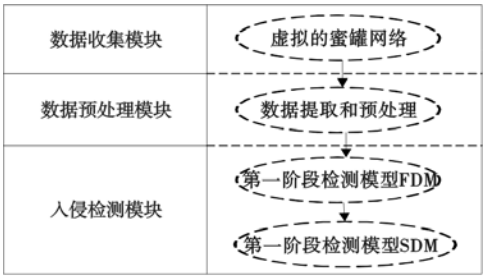

图1 HoneypotIDS结构

1)数据收集模块

数据收集模块是一个由蜜罐组成的网络,它的主要目的是利用蜜罐获取大量原始数据。为了捕获可疑的攻击数据,一般在被监控网络中配置若干蜜罐,并将这些单独的蜜罐配置成为一个蜜罐网络。

2)数据预处理模块

由数据收集模块得到的原始数据并不适合直接提供给入侵检测模块进行分析,所以必须由数据预处理模块对这些原始数据进行挖掘和标准化,然后再将处理过的标准化数据交由入侵检测模块进行检测。

因为数据预处理模块的数据来源是蜜罐中的事件,所以首先描述 HoneypotIDS 中的事件。当一台主机通过发送报文或者连接来访问蜜罐时,就产生了一个事件。所有的事件具有下列主要属性:时间、源 IP、目的 IP、协议、目的端口,以及在一个 TCP/UDP 连接中的数据包有效负载长度或者消息平均长度。

数据预处理模块收集给定时间内的事件并计算某些统计属性。本文给定时间假设为 1min,即数据预处理模块对每分钟内所有事件的某些属性进行统计。每分钟内得到的统计数据可以称之为一个样本,每个样本的属性如表 1 所示。在表 1 中,每个样本包含四个属性,分别用 A_1、A_2、A_3、A_4 来表示。这四个属性中,前三个是属于网络层的属性,最后一个是属于应用层的属性,由于目前大量的访问是针对 Web 服务器和 Telnet 服务器的,所以,此处采用 Telnet 服务器的访问次数作为属性之一。

表 1 样本的四个属性

属性	名称	描述
A_1	数据流量/Byte	该属性表示给定时间蜜罐系统中数据的总体情况,该属性值用 a_1 表示
A_2	平均数据包长度/Byte	该属性测量蜜罐所接收数据包的平均长度,该属性值用 a_2 表示
A_3	数据包进出总数/个	该属性统计进出蜜罐系统的 TCP、UDP 和 ICMP 数据包的总数,该属性值用 a_3 表示
A_4	Telnet 服务器的访问次数/次	该属性统计在给定时间对 Telnet 服务器的访问次数,该属性值用 a_4 表示

经过数据预处理后的样本保存在特定的文本文件当中,由入侵检测模块进行读取并分析。

3)入侵检测模块

入侵检测模块从数据预处理模块获得标准化数据,然后判断是否存在入侵事件。

在理论上可以认为,任何与蜜罐的交互极有可能是一次未授权的或者恶意的行为,任何与蜜罐建立连接的尝试都极有可能是一次探测、攻击或者攻陷。这就是说,通向蜜罐的流量都是高度可疑的。但是,并不是所有与蜜罐的交互都是真正的攻击,如在偶然情况下普通用户对蜜罐系统的无意访问就不属于攻击。因此,本文在设计入侵检测模块时,采用两阶段检测模型。第一阶段的检测模型用于将异常事件划分为正常类和攻击类。第二阶段检测模型在第一阶段检测模型的基础上进行进一步检测,确定其属于口令破解、端口扫描、缓冲区溢出、DoS 和未知攻击中的哪一类。本文将在第 2 章介绍如何建立以上两个检测模型。

2 入侵检测模块的设计

第一阶段的检测模型利用感知器学习方法建立,被称为 FDM 检测模型。该模型是一个单层感知器,其目标是将从数据预处理模块获得的事件划分为正常类和攻击类。该检测模型可以降低模型的设计复杂性,减少数据的计算量。第二阶段是采用感知器学习方法建立的 SDM 检测模型。其根据第一阶段检测模型的输出结果(即已经被划分为攻击类的异常事件)进行进一步检测,从而产生最终的检测结果。两阶段检测模型流程如图 2 所示。

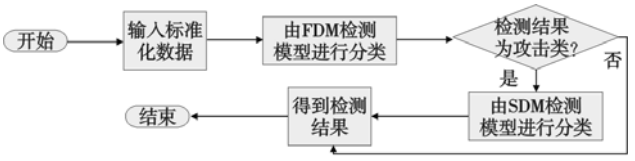

图2 两阶段检测模型的流程

2.1 第一阶段检测模型 FDM

FDM 检测模型是一个具有 5 个输入和 1 个输出的单层感知器,这个网络结构是根据所要解决的实际问题的性质所确定的。由于 FDM 检测模型的主要目标是用于区分正常类和攻击类,因此输出层只要 1 个神经元就可以解决两个类的分类问题。在数据预处理模块中,一个样本包含第 1 章所描述的四个属性(A_1,A_2,A_3,A_4),这四个属性上的取值(a_1,a_2,a_3,a_4)就是输入层上的一个输入。

此外,本文把感知器结构中的偏差 b 并入网络权值$(w_1,w_2,\cdots,w_n)$中。同时,输入样本也相应增加一个属性值 a_0,故令 $w_0=-b$,$a_0=1$。这样对每一个样本进行了扩展,在求总和时,不必再包含一个额外的常量元素。加上扩展属性,输入层一共有 5 个输入节点。FDM 检测模型的结构如图 3 所示。

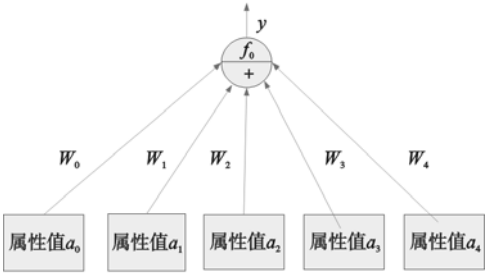

图3 FDM检测模型的结构

依据感知器的学习规则,FDM 检测模型的数学模型可表示为

$$y=f(\sum_{i=0}^{4}w_ia_i)$$

FDM 检测模型对输入样本的分类如下:

$$y=\begin{cases}1 & \text{攻击类}\\ 0 & \text{正常类}\end{cases}$$

当 FDM 检测模型的输出为 1 时,输入样本属于攻击类;输出为 0 时,输入样本属于正常类。

FDM 检测模型的学习规则在于寻找恰当的权系数 $w=(w_0,w_1,w_2,w_3,w_4)$,使模型对一个特定的样本 $a=(a_0,a_1,a_2,a_3,a_4)$能产生期望输出 t。当 a 分类为攻击类时,期望值 $t=1$;当 a 为正常类时,$t=0$。

感知器学习规则的实质是权值的变化量等于正负输入,即

$$\text{新的 } w_i=\text{旧的 } w_i+(t-y)\times a_i$$

因此,FDM 检测模型用以调整网络权值的学习规则可表示为

$$\text{新的 } w_i=\begin{cases}\text{旧的 } w_i & \text{当 } t=y\\ \text{旧的 } w_i+a_i & \text{当 } t=1,y=0\\ \text{旧的 } w_i-a_i & t=0,y=1\end{cases}$$

其中,$i=0,1,2,3,4$。

2.2　第二阶段检测模型 SDM

本文利用感知器学习方法建立 SDM 检测模型,它用于划分五个类,包括口令破解类、端口扫描类、缓冲区溢出类、DoS 类和未知攻击类。只有当 FDM 检测模型将一个样本划分为攻击类时,该样本才会被送入 SDM 检测模型进行进一步的检查,以确定它属于哪个具体的攻击类,或者属于未知攻击类。

SDM 检测模型是一个具有 5 个输入和 4 个输出的感知器,这个网络结构也是根据所要解决的实际问题的性质所确定的。由于 SDM 检测模型的主要目标是用于区分口令破解类、端口扫描类、缓冲区溢出类、DoS 类和未知攻击类这五个类,因此输出层需要 4 个神经元。这里需要 4 个而不是 5 个神经元的原因是,本文假设某个样本只要不属于口令破解类、端口扫描类、缓冲区溢出类或 DoS 类之中的任何一类,那么它就属于未知攻击类。与 FDM 检测模型的输入层一样,加上扩展属性,SDM 检测模型的输入层一共有 5 个输入节点。SDM 检测模型的网络结构如图 4 所示。

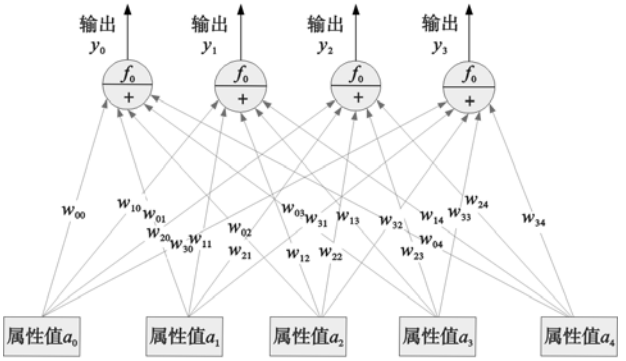

图4　SDM检测模型的网络结构

SDM 检测模型的数学模型可表示为

$$y_j=f\left(\sum_{i=0}^{4}w_{ji}a_i\right)$$

其中,$i=0,1,2,3,4$ 且 $j=0,1,2,3$。

SDM 检测模型对输入样本的分类如下:

$$Y=(y_0,y_1,y_2,y_3)\begin{cases}(1,0,0,0) & \text{口令破解类}\\ (0,1,0,0) & \text{端口扫描类}\\ (0,0,1,0) & \text{缓冲区溢出类}\\ (0,0,0,1) & \text{DoS 类}\\ \text{其他} & \text{未知攻击类}\end{cases}$$

SDM 检测模型的学习规则在于寻找恰当的网络权值 w_{ji}(其中,$i=0,1,2,3,4$,对应模型的 5 个输入;$j=0,1,2,3$,对应模型的 4 个神经元;w_{ji}表示第 j 个神经元与第 i 个输入间的连接权值),使模型对一个特定的样本 $a=(a_0,a_1,a_2,a_3,a_4)$能产生期望输出 $T=(t_0,t_1,t_2,t_3)$。当 a 分类为口令破解类时,期望值 $T=(1,0,0,0)$;当 a 为端口扫描类时,期望值 $T=(0,1,0,0)$;当 a 分类为缓冲区溢出类时,期望值 $T=(0,0,1,0)$;当 a 分类为 DoS 类时,期望值 $T=(0,0,0,1)$。

SDM 检测模型用以调整网络权值的学习规则可表示为

$$\text{新的 } w_{ji}=\text{旧的 } w_{ji}+(t_j-y_j)\times a_i$$

其中,$i=0,1,2,3,4$ 且 $j=0,1,2,3$。也就是说,对于一个特定的样本 $a=(a_0,a_1,a_2,a_3,a_4)$,用它的实际输出 $Y=(y_0,y_1,y_2,y_3)$与期望输出 $T=(t_0,t_1,t_2,t_3)$进行比较。如果实际输出和期望输出在第 j 个神经元上的值相等,则与该神经元相连接的网络权值不变;反之,与该神经元相连的所有网络权值都必须根据学习规则进行相应的修改。

3　HoneypotIDS 系统的配置与实现

3.1　HoneypotIDS 的配置

HoneypotIDS 包括 HoneypotIDS 代理和 HoneypotIDS 中心控制台两个部分。HoneypotIDS 的网络配置如图 5 所示。

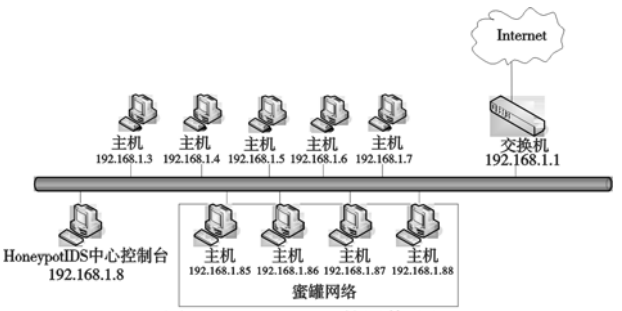

图5　HoneypotIDS的网络配置

HoneypotIDS 代理实现了 HoneypotIDS 体系结构中的 Honeynet 模块。在本次设计中,用虚拟机产生了 4 个蜜罐,每一个蜜罐就是一个 HoneypotIDS 代理,它们组成了一个蜜罐网络。HoneypotIDS 代理上具有没有安装补丁的 Windows XP 操作系统,开放了常用的 TCP/UDP 端口,并安装了 Telnet 服务器和 IIS 服务器。最后,还为每个代理配置了在被监控网络中未使用的 IP。HoneypotIDS 被伪装成具有一些特定服务和存在各种安全漏洞的诱饵,引诱入侵者对其进行攻击。

HoneypotIDS 中心控制台被安装在 Windows XP 平台上,它实现了 HoneypotIDS 体系结构中的数据预处理模块和入侵检测模块。由于中心控制台不停地监视着被监控网络,所以当一个 HoneypotIDS 代理被访问时,数据预处理模块就会统计事件并将数据规格化,然后交由入侵检测模块进行分析。入侵检测模块中检测模型得到的结果将显示在中心控制台上。除了数据预处理和检测功能外,HoneypotIDS 中心控制台还具有额外的功能,包括界面、日志管理,保存和导入检测模型的网络权值。

3.2　数据预处理模块的实现

HoneypotIDS 的数据预处理模块在获取数据并进行分析时,采用了第三方软件,即 Ethereal 和 Honeysnap。Ethereal 用于获取蜜罐网络中的数据报文,而 Honeysnap 则用于对 Ethereal 所获取的数据报文进行详细分析。

Ethereal 是一款功能强大的嗅探器[6],它可以从网络接口

上捕获实时数据包并以非常详细的协议方式显示,还可以打开或者存贮捕获的数据包。数据预处理模块利用 Ethereal 从蜜罐中获取大量原始数据。本设计主要使用 Ethereal 实时捕获数据包的功能。通过设置,Ethereal 可以捕获进出特定蜜罐主机的数据报文,还可以以特定的频率把所捕获的信息保存到某种格式的文件中去。例如,设计 Ethereal 使其能够对 IP 地址为 192.168.1.77 的蜜罐进行数据报文捕获,并每分钟产生一个 pcap 文件。Ethereal 生成的 pcap 文件则由 Honeysnap 软件来进行详细的分析。

捕获到原始数据并保存到 pcap 文件后,数据预处理模块利用 Honeysnap 对其进行分析和处理。Honeysnap 是一个用于从 pcap 文件中提取和分析数据的基本工具,它可以分析单个或者多个 pcap 数据文件,并针对被处理数据中的重要事件产生分析文件。例如,它可以统计 pcap 文件中 TCP、UDP 或 ICMP 包等数据包的总数;也可以分别统计 DNS、Http、Telnet 或 Ftp 等具体服务的进出数据包的总数;还可以得到一些全局的数据,如数据流量和平均数据包长度等。这不但为使用提供了高价值的预处理数据,还节省了大量时间。Honeysnap 是一个在命令行下运行的软件,它具有很多不同的命令。为了方便起见,可以将这些命令写成配置文件 honeynet.cfg。

通过使用第三方软件 Ethereal 和 Honeysnap,数据预处理模块为入侵检测模块提供用于检测的样本文件。

3.3　入侵检测模块的实现

对于本文建立的 FDM 检测模型和 SDM 检测模型,需要通过训练来确定它们的参数。本节讨论了训练这些检测模型的方法。训练环境和前面介绍的配置环境一样,包含 5 台主机和 7 个具有普通操作系统(主要是 Windows 操作系统)和一些特定服务的蜜罐。其中一台主机被选择用来完成训练案例。

本文使用了 6 组训练案例,它们被划分为攻击类和正常类两种类型。对于攻击类,采用一些著名的攻击工具来模拟入侵;对于正常类,模拟合法用户的行为,通过不同的协议来访问诱饵。表 2 为这些训练案例。

表 2　训练案例

类型	名字	攻击分类
攻击	Brutus AET2	口令破解
	Superscan	端口扫描
	RPC GUI v2-r3L4x	缓冲区溢出
	HGod	拒绝服务
正常	Web	其他
	Telnet	其他

1)FDM 检测模型的实现

对 FDM 检测模型进行训练的目标,就是计算它的网络权值。利用这些网络权值,可以将输入检测模型的样本进行正确分类。FDM 检测模型网络权值的确定不是通过计算,而是通过网络的自身训练来完成的,这也是人工神经网络在解决问题的方式上与其他方法的最大不同点。借助于计算机的帮助,几百次甚至上千次的网络权值的训练与调整过程能够在很短的时间内完成,而且不需要人工干预。由表 2 所述的训练案例可以得到一个适用于 FDM 检测模型的训练样本集,如图 6 所示。

经过 8001 次循环后,所有攻击训练样本和正常训练样本的输出类标志都与它们的期望类标志一致,即进行了正确分类。证明该样本集是线性可分的。此时,FDM 检测模型网络权值分别为:$w_0 = -10521.27$, $w_1 = 12998.05$, $w_2 = -507.98212$, $w_3 = 4507.4062$ 和 $w_4 = -8765.792$。

```
30 .58399 65 .49036 467 253 Attack
30 .59600 65 .51606 467 253 Attack
10 .25099 72 .19014 142 75 Attack
7 .59199 80 .76596 94 42 Attack
5 .65799 72 .53846 78 42 Attack
5 .65800 72 .53846 78 42 Attack
474 .09299 58 .00721 8171 4 Attack
473 .66100 58 .01115 8165 5 Attack
474 .03000 58 .00660 8170 4 Attack
473 .88799 57 .98923 8170 3 Attack
2 .67799 178 .53333 15 0 Attack
3 .75799 129 .58620 27 0 Attack
2 .65300 165 .81250 14 0 Attack
2 .73899 161 .11765 17 0 Attack
216 .73600 1042 .00000 208 0 Attack
227 .15599 1042 .00000 218 0 Attack
239 .65999 1042 .00000 230 0 Attack
222 .18899 1038 .26636 214 0 Attack
3 .64999 60 .83333 60 36 Normal
3 .92399 60 .36923 65 39 Normal
4 .06599 59 .79412 66 39 Normal
0 .91899 83 .54545 9 0 Normal
0 .67600 67 .60000 8 0 Normal
0 .59200 74 .00000 8 0 Normal
```

图6　FDM检测模型的训练样本集

2)SDM 检测模型的实现

训练 SDM 检测模型也是为了确定它的网络权值。由表 2 所述的攻击训练案例可以得到一个适用于 SDM 检测模型的训练样本集,如图 7 所示。

```
30 .58399 65 .49036 467 253 GuessPwd
30 .59600 65 .51606 467 253 GuessPwd
10 .25099 72 .19014 142 75 GuessPwd
7 .59199 80 .76596 94 42 GuessPwd
5 .65799 72 .53846 78 42 GuessPwd
5 .65800 72 .53846 78 42 GuessPwd
474 .09299 58 .00721 8171 4 PortScan
473 .66100 58 .01115 8165 5 PortScan
474 .03000 58 .00660 8170 4 PortScan
473 .88799 57 .98923 8170 3 PortScan
2 .67799 178 .53333 15 0 BufferOverflow
3 .75799 129 .58620 27 0 BufferOverflow
2 .65300 165 .81250 14 0 BufferOverflow
2 .73899 161 .11765 17 0 BufferOverflow
216 .73600 1042 .00000 208 0 DoS
227 .15599 1042 .00000 218 0 DoS
239 .65999 1042 .00000 230 0 DoS
222 .18899 1038 .26636 214 0 DoS
```

图7　SDM检测模型的训练样本集

经过 165 次循环后,所有攻击训练样本输出类标志都与它们的期望类标志一致,即每个样本都已经被划分为它所期望的类。此时,SDM 检测模型的网络权值 $W[4][5]$ 如图 8 所示。

94.73088	-1009.5615	554.87726	-6757.5938	14418.208
-95.96376	-1462.8273	-14997.491	8116.9688	-12150.289
66.00612	-2633.627	7091.24	-33176.84	-722.0601
-460.44592	16521.002	1703.578	-23754.062	-722.51294

图 8　SDM 检测模型的网络权值

在图 8 中,一行表示一个特定的神经元与 5 个输入间的连接权值。例如,第一行表示了图 4 中的(w_{01}, w_{11}, w_{21}, w_{31}, w_{41}),其他以此类推。至此,完成对 SDM 检测模型的训练。

3.4　HoneypotIDS 的测试

本文设计实验测试了 HoneypotIDS 的检测能力。对于 HoneypotIDS 中的所有检测模型,关注的都是它们是否能够对测试样本进行正确的分类。区别在于,FDM 检测模型区分正常类和攻击类,而 SDM 检测模型区分口令破解、端口扫描、缓冲

区溢出、DoS 和未知攻击五个类。在实际测试时,测试样本包括了正常类(2 个)和五类攻击(2 个口令破解、3 个端口扫描、3 个缓冲区溢出、2 个 DoS 和 2 个未知攻击)共 14 个,其中未知攻击的测试样本通过实施不属于其他 4 个攻击类的攻击(如远程控制)来获得。表 3 描述了实验中使用的测试案例。测试环境与前面介绍的配置环境一致,其中一台 IP 地址为 192.168.1.3 的主机 A 被选择来完成测试。

表 3　测试案例

类型	名字	攻击分类
攻击	Brutus AET2	口令破解
	X-scan	端口扫描
	Webdavx3	缓冲区溢出
	傀儡僵尸 DDoS 攻击集合	拒绝服务
	RemoteWhere	未知攻击(远程控制)
正常	Web	其他
	Telnet	其他

对 FDM 检测模型进行测试的方法是,由主机 A 使用表 3 中所描述的测试案例对任意一台 HoneypotIDS 代理进行正常访问或攻击,得到测试样本。观察 FDM 检测模型是否能将测试样本正确地划分到正常类或攻击类中。

对 SDM 检测模型进行测试的方法是,对于已经被 FDM 检测模型检测为攻击类的测试样本,观察 SDM 检测模型是否能将其归为正确的攻击类。

1)FDM 检测模型的测试结果

通过测试,FDM 检测模型对正常类和攻击类都具有 100% 的检测率,它能将样本正确地划分为正常类和攻击类。但实际上,FDM 检测模型不可能具有这么高的检测率。这主要是由于实验条件的限制导致测试样本集过小造成的。如果测试样本达到成百上千个,对 FDM 检测模型的检测率将会有更实际的测试效果。

2)SDM 检测模型的测试结果

被 FDM 检测模型划分为攻击类的测试样本,将进入 SDM 检测模型以进行进一步的检测。对于口令破解、端口扫描、缓冲区溢出、DoS 和未知攻击这五类,SDM 检测模型的检测率也非常高,分别为 100%、100%、100%、100% 和 92%。在所有的测试样本中,仅有 1 个本应属于未知攻击类的样本错分到了端口扫描类中。

从 FDM 检测模型和 SDM 检测模型的测试结果来看,HoneypotIDS 的总体检测效果还是比较好的。从上面的分析可以看出,HoneypotIDS 对于攻击的漏报率较低,很好地体现了蜜罐的检测功能。同时,HoneypotIDS 能够比较好地对口令破解、端口扫描、缓冲区溢出和 DoS 这几类攻击进行分类,但是会将未知攻击归为它们其中的一类,产生一定的误报率。这是可以理解的,因为未知攻击有可能包含了上述几种攻击类型的特征。

4　结束语

本文提出了 HoneypotIDS 检测系统,该系统是一个基于人工神经网络和蜜罐的入侵检测系统。它的基本设计思想是在被监控网络中配置多个蜜罐,从这些蜜罐中收集数据,然后使用由人工神经网络中的感知器学习方法建立起来的检测模型对入侵行为进行检测。对 HoneypotIDS 的测试表明,它具有很好的检测能力,能够比较好地对口令破解、端口扫描、缓冲区溢出和 DoS 这几类攻击进行分类。由于实验条件的限制,训练样本集较小,下一步将构造较为完备的训练样本,对模型进行训练。

参考文献:

[1] SUN Wen-chen, CHEN Yi-ming. A rough set approach for automatic key attributes identification of zero-day polymorphic worm[J]. Expert Systems with Applications,2009,36(3):4672-4679.

[2] 张新宇, 卿斯汉, 李琦. 一种基于本地网络的蠕虫协同检测方法[J]. 软件学报, 2007,18(2): 412-421.

[3] 肖枫涛, 胡华平, 刘波. HPBR: 用于蠕虫检测的主机报文行为评级模型[J]. 通信学报,2008,29(10):108-116.

[4] PERDISCI R, DAGON D, LEE W, *et al*. Misleading worm signature generators using deliberate noise injection[C]//Proc of 2006 IEEE Symposium on Security and Privacy. Washington DC: IEEE Computer Society, 2006: 17-31.

[5] 陈博,方滨兴,云晓春. 分布式蠕虫检测和遏制方法的研究[J]. 通信学报, 2007,28(2):9-16.

[6] WANG Jie, WANG Jian-xin, CHEN Jian-er, *et al*. An automated signature generation approach for polymorphic worm based on color coding[C]//Proc of IEEE International Conference on Communications. Washington DC: IEEE Computer Society, 2009: 1-6.

[7] 向继,高能,荆继武. 一种基于门限签名的可靠蠕虫特征产生系统[J]. 计算机学报,2009,32(5):930-939.

[8] STANIFORD S, PAXSON V, WEAVER N. How to own the Internet in your spare time[C]//Proc of the 11th USENIX Security Symposium. Berkeley: USENIX, 2002:149-167.

[9] MANNA P K, CHEN S, RANKA S. Inside the permutation-scanning worms: propagation modeling and analysis[J]. IEEE/ACM Trans on Networking,2010,8(3):858-870.

[10] STEPHENSON B, SIKDAR B. A quasi-species approach for modeling the dynamics of polymorphic worm[C]//Proc of the 25th IEEE International Conference on Computer Communications. Washington DC:IEEE Computer Society, 2006:1-12.

(上接第 666 页)

[15] GAO Tie-gang,CHEN Zeng-qiang. Image encryption based on a new total shuffling algorithm[J]. Chaos, Solitons & Fractals,2008,38(1):213-220.

[16] ALVAREZ G,LI Shu-jun. Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption[J]. Communications in Nonlinear Science and Numerical Simulation,2009,14(11):3743-3749.

[17] 李张铮. 基于混沌的图像加密方法研究[D]. 大连:大连理工大学,2009.

[18] XU Shu-jiang, WANG Ying-long, GUO Yu-cui, *et al*. A novel chaos-based image encryption scheme[C]//Proc of Information Engineering and Computer Science Conference. [S.l.]:IEEE Computer Society, 2009: 1-4.

[19] WONG K W, KWOK B S H, LAW W S. A fast image encryption scheme based on chaotic standard map[J]. Physics Letters A, 2008,372(15):2645-2652.

[20] 佟晓筠,崔明根. 基于扰动的复合混沌序列密码的图像反馈加密算法[J]. 中国科学 F 辑:信息科学,2009,39(6):588-597.

[21] ZHU Gui-liang, ZHANG Xiao-qiang. Mixed image element encryption algorithm based on an elliptic curve cryptosystem[J]. Journal Electronic Imaging,2008,17(2):1-5.