虚拟化环境中基于神经网络专家系统的Rootkit检测方法研究

赵志远 朱智强 孙 磊 马可欣
(解放军信息工程大学三院 郑州 450000)

摘 要 针对现有虚拟化环境客户操作系统中对Rootkit检测存在误判率高、无法检测未知Rootkit等问题,提出了一种基于神经网络专家系统的Rootkit检测方法(QPSO_BP_ES)。该方法将神经网络与专家系统相结合,利用其各自的优势构成检测系统。在实际检测时,首先捕获事先选取出来的Rootkit典型特征行为,然后通过训练好的神经网络专家系统来检测客户操作系统中是否存在Rootkit。最后通过实验表明,QPSO_BP_ES检测系统模型可以降低误判率,有效地检测已知和未知的Rootkit。

关键词 虚拟化,量子粒子群,神经网络,专家系统,Rootkit
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.8.037

Research on Rootkit Detection Method Based on Neural Network Expert System in Virtualized Environment

ZHAO Zhi-yuan ZHU Zhi-qiang SUN Lei MA Ke-xin
(The Third Institute, PLA Information Engineering University, Zhengzhou 450000, China)

Abstract In order to solve the problems about the high misjudgment ratio of Rootkit detection and undetectable unknown Rootkit in the virtualization guest operating system, a Rootkit detection method(QPSO_BP_ES) based on neural network expert system was proposed. The detection system combines neural network with expert system, which can take advantage of them. In the actual detection, QPSO_BP_ES firstly captures the previously selected Rootkit's typical characteristic behaviors. And then, the trained system detects the presence of Rootkit. The experimental results show that QPSO_BP_ES can effectively reduce the misjudgment ratio and detect both known and unknown Rootkit.

Keywords Virtualization, QPSO, Neural network, Expert system, Rootkit

1 引言

云计算作为一种新兴的计算模式,给虚拟化技术带来了全新的发展契机[1]。虚拟化技术作为云计算的重要支撑技术之一,其安全问题备受关注。同传统计算机安全威胁类似,恶意代码也在虚拟化技术安全威胁中占有突出位置[2]。

Rootkit作为一种特殊的恶意代码,一般具有系统最高权限,能够持久可靠地存在系统中,主要威胁是隐藏进程、注册表、文件以及其它恶意代码等。在应对虚拟化环境下客户操作系统中检测Rootkit的挑战时,美国乔治亚理工大学用开源虚拟化技术Xen开发出恶意代码行为分析系统Ether[3],其分为粗粒度和细粒度跟踪两种模式,可以有效欺骗恶意代码反检测,保证其透明性,但是对内存的监控只能精确到页面的粒度,无法抵挡内存重定向攻击。美国威斯康星大学利用开源虚拟化技术Xen开发出Rootkit检测程序Lycosid[4],其只信任虚拟机监视器VMM,通过交叉视图对比检测隐藏的进程从而判别是否存在Rootkit,但是有较高的误报率。潘剑锋等人提出一种基于专家系统的恶意代码检测方法[5],其能够准确检测出已经将恶意行为信息存储于知识库中的恶意代码和静态特征码改变的恶意代码,但是该系统需要扩展恶意行为信息,专家个性化知识具有片面性,难以刻画,导致恶意行为信息这种知识的获取是一个"瓶颈",并且自适应能力较差。上述方法虽然在一定程度上可以检测出Rootkit,但是面对恶意代码快速发展的态势,需要一种更加灵活、自适应强的启发式检测方法。

受文献[5,6]启发,即利用专家系统开发的Rootkit检测系统可以提高检测Rootkit的准确性,并且可以检测出通过加壳等规避技术的Rootkit,但是知识获取能力是该系统的一个"瓶颈",并且其自适应性有待提升,所以本文需要寻求一种可以有效获取恶意代码行为信息这种知识,同时又可以自适应学习检测未知恶意代码的方法。

将神经网络与专家系统结合起来可以避免传统的专家系统在获取知识方面的"瓶颈",同时由于神经网络在自适应、自学习等方面有特殊优势[7],因此可以很好地解决上述问题。

到稿日期:2014-08-12 返修日期:2014-10-26 本文受国家863计划基金项目(2008AA01Z404),国防预研基金项目(910A26010306JB5201)资助。

赵志远(1989-),男,硕士生,主要研究方向为虚拟化技术、信息安全,E-mail:zzy_taurus@foxmail.com;**朱智强**(1961-),男,教授,主要研究方向为云计算、信息安全;**孙 磊**(1973-),男,博士,副研究员,主要研究方向为云计算基础设施可信增强、可信虚拟化技术;**马可欣**(1990-),女,硕士生,主要研究方向为云计算、可信虚拟化技术。

神经网络是对人脑神经元网络的简化和模仿，由大量神经元以拓扑结构相连接[8]。利用神经网络的高效性和自学习能力，仅仅需用专家判断 Rootkit 恶意代码行为的知识来训练神经网络，得出与专家一致的判断即可，该方法有效地解决了专家系统在提取知识方面的“瓶颈”；同时利用神经网络的自适应、自学习能力，可以不断更新知识库，从而判断未知的 Rootkit 恶意代码，打破目前检测方法对未知的 Rootkit 毫无应对之策的局面。

本文提出一种虚拟化环境下基于神经网络专家系统的 Rootkit 启发式检测方法。针对现有方法无法检测 Rootkit 恶意软件的变形、未知 Rootkit、误报率高等问题，通过结合神经网络的自适应、自学习能力和专家系统解决问题的能力结合来应对上述问题，同时引入量子粒子群算法来提高神经网络的训练精度，避免神经网络收敛速度慢、容易陷入局部极小等传统问题，以提高对 Rootkit 恶意代码的识别能力，增强虚拟机的安全性。

2 神经网络专家系统模型设计

将神经网络与专家系统相结合，充分利用二者的性能优势来增加检测系统获取知识和推理问题的能力是本文的主要设计思想。需要解决的重点问题是知识获取、知识表达，以及根据知识库进行推理。

基于行为的 Rootkit 检测模型一般都是通过待测程序的行为特征与特征知识库进行匹配来判断 Rootkit 是否存在[9]。本文基于人工智能方法建立的 QPSO_BP_ES 检测系统模型可以动态地建立行为特征库，并且可以在后续检测过程中更新知识库。该系统模型主要包括行为特征捕获模块、量化模块、工作存储器、BP_ES 模块、解释器、显示界面，如图 1 所示。

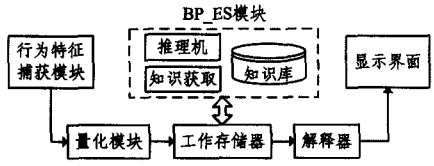


图 1 Rootkit 检测模型

行为特征捕获模块可靠地捕获被检测系统中与 Rootkit 有关的、有代表性的并且尽可能详细准确的信息，例如软件执行过程中的行为数据、文件注册表项、软件执行后的遗留痕迹等。

量化模块将每一个行为特征在检测 Rootkit 时的贡献值大小量化成 0－1 之间的不同数值，并将该数值作为 BP_ES 模块的输入值。

工作存储器存储量化模块提供的初始事实、推理机的推理结果等内容，并且处于不断变化之中。

BP_ES 模块是系统模型的核心模块，主要完成知识的获取、知识库的构建，以及根据知识库规则完成推理等工作。

解释器对系统推理结果提供解释。

显示界面显示最终的结果。

该系统模型部署在特权域中，虚拟机的强隔离性保证了系统模型免受 Rootkit 的攻击，同时又可以完成对 Rootkit 的检测。其工作流程可以描述为：行为特征捕获模块捕获被检测系统的行为特征，量化模块完成对行为特征的数值化，通过神经网络学习确定阈值权重等参数并存储在神经网络中构成专家系统知识库，神经网络专家系统完成推理过程后将推理结果存储于工作存储器中，推理结果经过解释器显示在显示界面中。

3 关键模块及其实现

本文设计的检测系统模型基于神经网络专家系统完成对 Rootkit 的智能检测。相比于其他检测方法，其优点在于可以自适应地学习及识别未知 Rootkit。而该系统中知识的获取、表达和构建占有非常重要的地位，所以本节主要介绍量化模块和 BP_ES 模块。

3.1 量化模块

对于神经网络专家系统检测模型，编码就是把逻辑思维形式的规则转化成数值机器的形式，并存放于神经网络之中。解码就是将数值化的形式转换成用户易理解的自然语言。本节根据 Rootkit 行为特征在检测时的贡献值进行量化编码。

Rootkit 的行为特征可以用一个集合表示为：

$$U=\{u_1,u_2,\cdots,u_n\}$$

其中，u_i 表示 Rootkit 的一种行为特征，n 为行为特征库中行为特征的总数。

一种行为特征可以出现在恶意或者正常的程序之中。因此，为更加准确表示某种特征对正确检测 Rootkit 的贡献，该检测模型记下每一个行为特征出现在 Rootkit 恶意代码和正常普通程序中的次数，然后用二元组表示每个行为特征：

$$u_i=\langle R_i,N_i\rangle$$

其中，R_i 表示该行为特征在 Rootkit 中表现的次数，N_i 表示该行为特征在正常程序中表现的次数。

一个行为特征在恶意代码中出现的次数远超于其在正常程序中出现的次数，则说明其是恶意代码的一种经典特征，在检测 Rootkit 时发挥的作用更大。这里对各行为特征赋予贡献值，以提高对 Rootkit 行为的检测，减少对正常程序的误报。对应于 U，将行为特征贡献值归一化，其向量形式表示为：

$$V=\{v_1,v_2,\cdots,v_n\}$$

这里定义行为特征 u_i 对检测 Rootkit 的贡献值为：

$$Con(u_i)=\frac{n\times R_i}{\sum R_i+\sum N_i}\times\left(1-\frac{n\times N_i}{\sum R_i+\sum N_i}\right)$$

将行为 u_i 的贡献值归一化后作为其神经网络的输入，即：

$$v_i=\frac{Con(u_i)}{\sum Con(u_i)}$$

其中，R_i 和 N_i 同上，n 为行为特征库中行为特征的总数。这样，程序行为特征 u_i 就可以通过量化表示出来。

3.2 BP_ES 模块

BP_ES 模块包括知识获取、知识库和推理机 3 个子模块，是该神经网络专家系统模型的核心组成部分。传统的专家系统一般采用规则、框架、模型、语义网络等标准的形式来显式地表达知识，这种方法难以表示领域专家的经验等个性化知识；然而神经网络专家系统将知识隐式地表示成网络结构中的权值和阈值，间接表示出专家的个性化知识。三层 BP 神经网络的结构如图 2 所示。

神经网络的结构、权值、阈值构成了专家系统的知识库，存储在神经网络中；将知识量化为 V 来获取知识；利用神经网络的结构来实现推理机制。

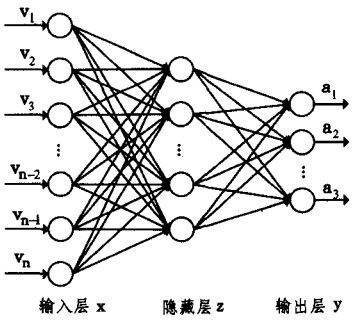


图 2　三层 BP 神经网络结构

3.3　神经网络专家系统的知识获取

利用样本程序来训练神经网络专家系统完成知识库的建立，使该系统能够获得与程序事实相符的输出，判断是否存入 Rootkit。针对 BP 神经网络收敛速度慢以及容易陷入局部极小的问题，本文将用量子粒子群算法优化神经网络来实现对该系统的训练，完成知识库的建立和推理机制的形成。

将量子粒子群算法(QPSO)与 BP 神经网络相结合来寻优网络的结构、权值和阈值的方法是：训练时，首先用量子粒子群算法优化网络的结构、参数，使搜索范围缩小；然后利用 BP 算法进行精确求解。这种方法同时兼具了二者的优势，利用它们之间这种互补的优势避免了收敛速度慢和局部极小的问题，从而使训练效果得以极大地改善。

(1)基于 QPSO 算法寻优

在 QPSO 中，粒子的进化公式如下：

$$P=\frac{\varphi_1 P_i+\varphi_2 P_g}{\varphi_1+\varphi_2},\varphi_1,\varphi_2=Rand$$

$$m_{best}=\sum_{i=1}^{M}P_i/M=(\sum_{i=1}^{M}P_{i1}/M,\sum_{i=1}^{M}P_{i2}/M,\cdots,\sum_{i=1}^{M}P_{iD}/M)$$

$$X(t+1)=P\pm\phi\times|m_{best}-X(t)|\times\ln\frac{1}{q},q=Rand$$

其中，M 为粒子数目；D 为粒子维数；m_{best} 为粒子群中粒子的平均最佳位置；P_i 代表粒子 i 的最佳位置；P_g 代表粒子种群的全局最佳位置；φ 为收缩扩张系数。

QPSO 寻优 BP 神经网络参数的流程如图 3 所示。

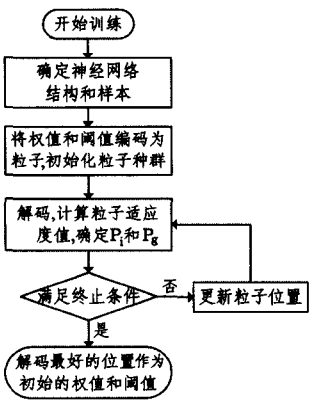


图 3　QPSO 算法流程

具体步骤如下：

步骤 1　确定 QPSO_BP_ES 系统结构，并给出训练样本集。

步骤 2　将神经网络的训练参数编码成实数码串表示的粒子，包括神经网络的结构、权值、阈值，则每个粒子代表一组神经网络的参数。根据粒子群的规模，依据上文粒子结构随机形成第一代种群，并初始化 P_i 和 P_g。

步骤 3　对每一个粒子码串进行解码来获得神经网络对应的参数，并将样本输入该参数构成的神经网络中，获得对应的输出，然后计算每个粒子的适应度值，确定 P_i 和 P_g。

步骤 4　判别是否满足算法的中断条件，若满足则转到步骤 6；否则继续步骤 5。

步骤 5　根据上述 3 个公式更新每一个粒子的位置，并转到步骤 3。

步骤 6　解码群体经历的最好位置并将其作为网络的初始化参数。

(2)基于 BP 算法精确求解

在上述寻优的基础上采用 BP 算法继续优化，得到最终的优化网络。设神经网络专家系统输入层为 n 个节点，隐含层为 k 个节点，输出层为 m 个节点。隐含层节点的输出函数为：

$$z_r=f(\sum_{i=1}^{n}W_{ir}x_i+T_r),r=1,2,\cdots,m$$

输出层节点的输出函数为：

$$y_j=f(\sum_{r=1}^{m}W'_{rj}z_r+\theta_j),j=1,2,\cdots,k$$

式中，$f(\cdot)$采用 S 型函数，即 $f(x)=(1+e^{-x})^{-1}$。其中，W_{ir}、W'_{rj}分别为 x_i 到 z_r 和 z_r 到 y_j 之间的权值，T_r、θ_j 分别为 z 和输出层的阈值。

神经网络在训练过程中，把输出层节点的误差反向逐层传播至各连接点，求出各连接点的误差，然后依据误差对各参数进行优化，使网络能够得到预期的输出，完成模式对 $X^{(l)}\to Y^{(l)}(l=1,2,\cdots,p)$的映射。其中 $X^l=(x_1^{(l)},x_2^{(l)},\cdots,x_n^{(l)})$，$Y^l=(y_1^{(l)},y_2^{(l)},\cdots,y_n^{(l)})$，$x_i^{(l)}\in R$，$y_j^{(l)}\in R$(R 为实数域)。

误差反向传播算法步骤如下：

步骤 1　将用量子粒子群算法获得的参数赋予神经网络。

步骤 2　对每一个样本模式对 $X^{(l)}\to Y^{(l)}(l=1,2,\cdots,p)$，按照下列方法操作：

(a)将 $x_i^{(l)}$ 的值输入到输入层，按下列公式计算输出值：

$$z_r=f(\sum_{i=1}^{n}W_{ir}x_i+T_r),r=1,2,\cdots,m$$

$$y_j=f(\sum_{r=1}^{m}W'_{rj}z_r+\theta_j),j=1,2,\cdots,k$$

(b)计算误差函数：

$$E_j=\sum(y_j^{(l)}-y_j)^2/2$$

(c)计算实际输出值 y_j 和期望输出值 $y_j^{(l)}$ 的差值 d_j：

$$d_j=y_j(1-y_j)(y_j^{(l)}-y_j)$$

(d)向隐含层节点反向分配误差 e_r：

$$e_r=z_r(1-z_r)(\sum_{j=1}^{m}W'_{rj}d_j)$$

(e)调整各层之间的连接权值：

$$\Delta W'_{rj}(t+1)=\lambda z_r d_j$$

$$\Delta W_{ir}(t+1)=\beta x_i e_r$$

式中，λ、β为学习率，一般在[0,1]范围内取值。

(f)调整输出层和隐含层单元的阈值：

$$\Delta\theta_j(t+1)=\lambda d_j$$

$$\Delta T_r(t+1)=\beta e_r$$

步骤 3　计算全局误差函数：$E=\sum E_j$，如果 $E\geqslant\varepsilon$ 则重复步骤 2；否则转到步骤 4。

步骤 4　学习结束。

该神经网络专家系统经过 QPSO 算法和 BP 算法学习训

练得到的权值和阈值存储在网络结构中，构成了系统模型的知识库。这种方法同时拥有二者的优势，避免了收敛速度慢和局部极小问题。

4 实验结果与分析

4.1 QPSO_BP_ES 检测模型部署

本文是在虚拟化环境下检测客户操作系统中是否存在 Rootkit，因此在虚拟机中如何部署该检测模型至关重要。在 Xen 虚拟化环境中，虚拟机监视器(Virtual Machine Monitor，VMM)具有最高特权级，特权域作为 VMM 的助手直接管理其他非特权域。虚拟化架构具有隔离性强、可信计算基小等优点。为使问题简单，假设 VMM 和特权域都是安全可信、不可被攻击的。基于此，将检测模型部署在特权域中，这样既可以保证检测模型与被检测系统相隔离以及检测模型的安全可靠，又不需要修改 VMM 的代码，如图 4 所示。

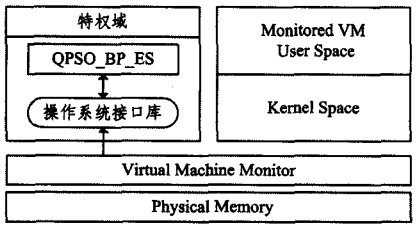


图 4 虚拟化环境系统部署

在特权域中增加操作系统接口库来保证专家系统的输入信息能达到操作系统级别，更加利于检测模型中特征行为的扩展。

4.2 Rootkit 特征行为描述

Rootkit 是一段程序或代码，能够长期隐藏在计算机中而不被检测程序发现[10]。其一般使用钩子改变函数执行流程到 Rootkit 钩子函数，使用直接内核对象操作(DKOM)技术将自身进程和内核模块等资源隐藏，使用远程线程注入技术操纵其他进程[11]。同时 Rootkit 可以植入系统内核，将自身静态信息隐藏，影响全系统的程序，致使目前安全病毒软件等检测程序难以检测。然而 Rootkit 要想运行发挥自己的功能，就不可避免产生多种软件行为和遗留的痕迹，如果可以将这些行为和痕迹捕捉并且进行必要的剖析，就有希望发现系统中的 Rootkit。在检测过程中，行为特征捕获模块捕获系统运行时的行为数据，经过选择后将其输入到系统模型计算出输出值。为使计算量更小，响应速度更快，同时又不能失去准确率，本文选取既具有代表性又能有效描述 Rootkit 特征行为的 14 种行为作为输入值，如表 1 所列。

表 1 Rootkit 典型行为特征

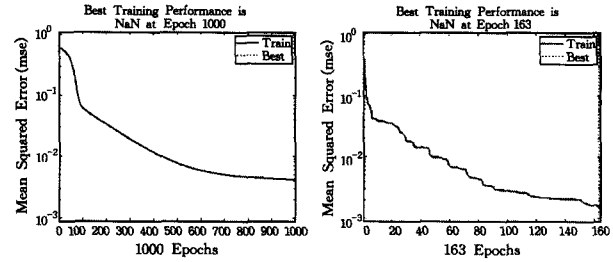
序号	特征行为描述
1	非系统进程 A 写入另一进程 B 的内存，但 A 不是 B 的父进程
2	创建进程的行为
3	非系统进程向其他进程注入远程线程的行为
4	创建服务、驱动等关键注册表键的行为
5	设置关键注册表值的行为
6	创建驱动文件等敏感文件的行为
7	修改可执行程序的导入表或者导出表来改变函数执行流程
8	非系统进程做出跨进程分配内存的行为
9	扫描出将系统服务派发表重定位的行为痕迹
10	扫描出 MSR Hook 的行为痕迹
11	扫描出 Inline Hook 的行为痕迹
12	扫描出导出表 Hook 的行为痕迹
13	扫描到隐藏进程的行为痕迹
14	扫描到隐藏驱动的行为痕迹

4.3 网络结构确定及训练

如何确定神经网络专家系统的结构及参数对于检测结果的准确性非常关键。依据 Kolmogorov 定理[12]，只需要一个隐含层，即选用 3 层结构即可。通过综合分析，以选出的 14 种典型 Rootkit 特征行为经过量化作为神经网络专家系统的输入值。根据输出结果判断其是否是 Rootkit，即两种情况：Rootkit 或者正常程序，因此输出层只需要一个神经元即可。而隐含层由公式 $N_{hidden}=\sqrt{N_{in}\cdot N_{out}}$[13] 确定为 4。综上分析，确定神经网络专家系统的网络结构为 14-4-1 型。

选取典型的 600 个程序(包括 Rootkit 和正常程序)，其中 550 个程序用于训练网络以最终确定权值和阈值，完成专家系统库的建立；再通过 50 个程序测试该系统模型的准确性。输入值在训练前已经通过量化模块进行了归一化处理，在此直接作为输入即可。训练过程中设定迭代次数为 1000，$\varepsilon=0.01$。

图 5(a)为传统的 BP 算法训练之后的结果误差，图 5(b)为经过本文方法改进之后训练的结果误差。



(a)BP 算法训练后的误差　　(b)改进方法训练后的误差

图 5

对比图 5(a)和图 5(b)后发现，相比 BP 算法，经过 QPSO_BP 训练的专家系统仅需较少的迭代次数就能达到很好的效果，且误差相对较小。

4.4 检测结果与分析

实验包括对正常系统和植入 Rootkit 后系统的检测，每次重复实验前均将系统恢复至纯净状态，设定输出层阈值为 0.1。正常系统运行时，每次实验持续 30s，输出层输出值均在 0 左右变化，如图 6 中带点线段，重复 10 次实验均没有误报；选取网上知名 Rootkit，如 hxdef、futo、ntrootkit、futo_enhanced、badrkdemo 等，在 T 时刻向系统中植入 Rootkit，然后观察输出层输出值的变化。如图 6 中平滑线段，在 19s 左右植入 Rootkit，输出层输出值变化为 1 左右。安装 Rootkit 后，行为数据中就会包括大量的隐藏信息、钩挂信息，从而输出层输出值趋近于 1，与正常系统有显著的区别，因此该系统能够有效地发现 Rootkit。

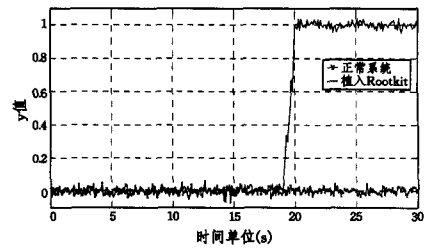


图 6 检测系统的检测结果

4.5 性能分析

为验证本文方法在检测 Rootkit 时的有效性，选取 QQ、

Office 等 30 个正常程序，同时选取 Hook Defender、HackerDefender、FURootkit、FUTo、NewRootkit11 等加壳或加密保护工具处理过的共 27 个恶意 Rootkit 程序，一共 57 个程序作为测试样本集。然后使用 Ether[3]、Lycosid[4]、潘剑锋专家系统 PJF_ES[5]、XenPHD[14]和本文方法来检测样本集，检测结果如表 2 所列。其中：

误判率＝(误判断的程序/样本集总数)×100%

漏报率＝(误判断的 Rootkit/Rootkit 样本集)×100%

检测率＝(正常检测的程序/样本集总数)×100%

＝1－误报率

表 2　检测结果对比

	Ether	Lycosid	PJF_ES	XenPHD	本系统
正常程序	30	29	30	30	30
恶意程序	24	21	25	24	26
误判率	5.3%	12.3%	3.5%	5.3%	1.8%
漏报率	11.1%	22.2%	7.4%	11.1%	3.7%
检测率	94.7%	87.7%	96.5%	94.7%	98.2%

由表 2 可以看出，本文系统的检测效果要优于其它检测系统。

为测试本系统运行时对整体性能的影响，分别选用 getpid 测试程序和文件拷贝对系统性能进行测试。其中，getpid 测试程序的唯一操作就是执行 getpid 系统调用，本文通过重复执行 getpid 程序 1500 次来模拟最坏情况；文件拷贝是 I/O 密集型操作，包含大量的读写作业。然后分别对 Ether、Lycosid、潘剑锋专家系统 PJF_ES、XenPHD 和本文方法进行性能测试。测试结果以没有任何检测系统为基准，计算各系统的性能损失的百分比，如图 7 所示。

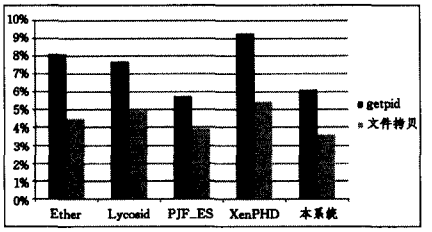


图 7　性能测试对比

由图 7 可以看出，本文系统在 getpid 测试中的性能损失略高于潘剑锋的专家系统检测方法，在文件拷贝测试中的性能损失最小，因此本文方法在性能损失方面表现很好。

结束语　有效检测 Rootkit 恶意代码对于虚拟化安全乃至云计算的发展至关重要。本文提出了一种虚拟化环境下基于神经网络专家系统的对客户操作系统中 Rootkit 的启发式检测系统，通过利用神经网络和专家系统解决此问题，同时引入量子粒子群算法提高神经网络的训练精度，避免了神经网络收敛速度慢、容易陷入局部极小的问题，提高了检测 Rootki 的能力，增强了虚拟机的安全性。最后通过实验表明该检测系统可以有效准确地发现 Rootkit，证实了本文方法的可行性。

参 考 文 献

[1] 冯登国，张敏，张妍等. 云计算安全研究[J]. 软件学报，2011，22(1)：71-83
Feng Deng-guo, Zhang Min, Zhang Yan, et al. Study on Cloud Computing Security[J]. Journal of Software, 2011, 22(1): 71-83

[2] 王怀习，陈建熊，王晨，等. 云计算中虚拟化技术的安全威胁[J]. 华中科技大学学报(自然科学版)，2012(S1)：153-156
Wang Huai-xi, Chen Jian-xiong, Wang Chen, et al. Security threats of virtualization in cloud computing[J]. Journal Huazhong University of Science and Technology (Natural Science Edition), 2012(S1): 153-156

[3] Dinaburg A, Royal P, Sharif M, et al. Ether: malware analysis via hardware virtualization extensions[C]//CCS08. 2008

[4] Jones S T, Arpaci-Dusseau A C, Arpaci-Dusseau R H, et al. VMM-based hidden process detection and identification using Lycosid[C]//VEE'08. 2008: 91-100

[5] 潘剑锋. 主机恶意代码检测系统的设计与实现[D]. 合肥：中国科学技术大学，2009
Pan Jian-feng. Design and Implemetation of Host-Based Malcode Detection System[D]. Hefei: University of Science and Technology of China, 2009

[6] 王蕊，冯登国，杨轶，等. 基于语义的恶意代码行为特征提取及检测方法[J]. 软件学报，2012，23(2)：378-393
Wang Rui, Feng Deng-guo, Yang Yi, et al. Semantics-Based Malware Behavior Signature Extraction and Detection Method[J]. Journal of Software, 2012, 23(2): 378-393

[7] 高刃，唐龙，伍爵博. 基于神经网络的无线传感器网络数据预测应用研究[J]. 计算机科学，2012，39(5)：44-47
Gao Ren, Tang Long, Wu Jue-bo. Application Research of Data Prediction in Wireless Sensor Network Based on Neural Network[J]. Computer Science, 2012, 39(5): 44-47

[8] 韩敏. 基于微粒群的神经网络预测控制理论及应用[M]. 北京：中国水利水电出版社，2013
Han Min. Theory and Application of Neural Network Predictive and Control Based on Particle Swarm[M]. Beijing: China Water-Power Press, 2013

[9] 冯帆，罗森林. 基于 VMM 的 Rootkit 检测技术及模型分析[J]. 信息网络安全，2013(6)：35-39
Feng Fan, Luo Sen-lin. The Analysis of VMM based Rootkit Detecting Technology and Model[J]. Information Network Security, 2013(6): 35-39

[10] 韩奕. 基于行为分析的恶意代码检测与评估研究[D]. 北京：北京交通大学，2014
Han Yi. A Research of Malware Detection and Evaluation Based on Behavior Analysis[D]. Beijng: Beijing Jiaotong University, 2014

[11] 刘婷婷. 面向云计算的数据安全保护关键技术研究[D]. 郑州：解放军信息工程大学，2013
Liu Ting-ting. Research on Key Technologies of Data Security towards Cloud Computing[D]. Zhengzhou: PLA Information Engineering University, 2013

[12] Kolmogorov A N. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers[J]. Dokl. Akad. Nauk SSSR, ,1941, 30(4): 299-303

[13] 李剑. 入侵检测技术[M]. 北京：高等教育出版社，2012
Li Jian. Intrusion Detection Technology [M]. Beijing: Higher Education Press, 2012

[14] 王丽娜，高汉军，刘炜，等. 利用虚拟机监视器检测及管理隐藏进程[J]. 计算机研究与发展，2011，48(8)：1534-1541
Wang Li-na, Gao Han-jun, Liu Wei, et al. Detecting and Managing Hidden Process via Hypervisor[J]. Journal of Computer Research and Development, 2011, 48(8): 1534-1541