基于混合式聚类算法的离群点挖掘在异常检测中的应用研究

尹 娜[1] 张 琳[1,2]
(南京邮电大学计算机学院 南京 210003)[1] (江苏省无线传感网高技术研究重点实验室 南京 210003)[2]

摘 要 为了提高异常检测系统的检测率,降低误警率,解决现有异常检测所存在的问题,将离群点挖掘技术应用到异常检测中,提出了一种基于混合式聚类算法的异常检测方法(NADHC)。该方法将基于距离的聚类算法与基于密度的聚类算法相结合从而形成新的混合聚类算法,通过k-中心点算法找出簇中心,进而去除隐蔽性较高的少量攻击行为样本,再将重复增加样本的方法结合基于密度的聚类算法计算出异常度,从而判断出异常行为。最后在KDD CUP 99数据集上进行实验仿真,验证了所提算法的可行性和有效性。

关键词 异常检测,离群点挖掘,NADHC
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.05.021

Research on Application of Outlier Mining Based on Hybrid Clustering Algorithm in Anomaly Detection

YIN Na[1] ZHANG Lin[1,2]
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)[1]
(Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China)[2]

Abstract In order to improve the detection rate of anomaly detection system, reduce the false alarm rate, and solve the problems existing in the current anomaly detection, outlier mining techniques were applied to anomaly detection, and this paper presented a network anomaly detection method based on hybrid clustering algorithm (NADHC). In the method, the clustering algorithm based on distance is combined with the density clustering algorithm to form a new hybrid clustering algorithm. The method is based on the k-medoids algorithm to find out the cluster centers. Next, NADHC removes a small amount of attack behavior samples which has obvious characteristics of high concealment, then calculates the abnormal degree by the repeated increasing samples combined with density-based clustering method to determine the abnormal behavior. NADHC algorithm was validated on KDD CUP 99 dataset. The experimental results show its feasibility and effectiveness.

Keywords Anomaly detection, Outlier mining, NADHC

1 引言

离群点是一个数据对象,其是显著偏离其余对象的数据对象,好像它是由不同的机制产生的一样。离群点的检测方法共分为4类:统计学方法、基于邻近性的方法、基于聚类的方法和基于分类的局部离群点方法。而入侵检测是对入侵行为的检测,它通过收集和分析网络行为、安全日志、审计数据、其他网络上可以获得的信息以及计算机系统中若干关键点的信息,检查网络或系统中是否存在违反安全策略的行为和被攻击的迹象。在入侵检测领域中的入侵行为通常被假设与系统的正常行为有着很大不同,根据这一特点,离群点挖掘技术能够用来解决入侵检测问题[1],此时入侵将被视为与正常行为数据不同的离群数据。

当前用于异常检测的方法有很多,例如张玲等人提出的基于粗糙集和人工免疫的集成入侵检测模型[2],通过运用一个"非自体检测器"来识别已知攻击,然后使用一个"自体检测器"来检测未知攻击;高赟等人提出了CADM算法进行异常检测[3],使用正常的日志和待检测的日志之间的相对熵作为异常程度的标识,充分利用了相对熵和文法压缩编码大小之间的关系来计算相对熵。除此之外,国外也有很多学者对离群点挖掘方法进行了深入的研究,例如Breunig等人提出了著名的基于密度的局部离群系数方法[4],该方法引入局部异常因子LOF来描述每个对象,从而达到网络异常检测的目的;Lee等人提出了一种基于在线过采样主成分分析的异常检测方法[5],该方法通过减少样本的策略对数据进行量化,从而实现异常流量的检测与识别。

虽然近年来出版的文献中提出了许多异常检测方法,但是上述的算法中都或多或少存在以下问题:计算复杂度高;检

到稿日期:2016-04-13 返修日期:2016-05-30 本文受国家自然科学基金(61402241,61572260,61373017,61572261,61472192),江苏省科技支撑计划(BE2015702),江苏省普通高校研究生科研创新计划(CXLX12_0482),南京邮电大学校级科研基金(NY217050)资助。
尹 娜(1994—),女,硕士生,主要研究方向为信任安全、数据挖掘等;张 琳(1980—),女,博士后,副教授,硕士生导师,CCF会员,主要研究方向为云计算、网络安全、信任、可信计算等,E-mail:zhangl@njupt.edu.cn。

测结果的精度和重复计算的次数依赖于用户给定的参数;需要处理大量的网络流量数据。针对以上问题,本文提出了一种基于混合式聚类算法的异常检测方法(Network Anomaly Detection method Based on Hybrid Clustering algorithm, NADHC)。将基于距离的聚类算法与基于密度的聚类算法相结合形成新的聚类算法,该算法通过 k-中心点算法找出簇中心,再根据簇中心的位置变化,通过新型的基于密度的聚类算法求解出异常度找出异常行为。通过与其他现有的异常检测算法相比,最终证明所提算法的异常检测性能优越。

2　离群点挖掘在异常检测中应用的基本框架

异常检测属于入侵检测的一种,入侵检测可分为异常检测和误用检测两类。异常检测是指将用户正常的习惯行为特征存储在数据库中,然后将用户当前的行为特征与特征数据库中的特征进行比较,如果两者的偏差足够大,则说明发生了异常[6]。从上述引言中可以发现,很多学者对基于离群点挖掘的异常检测方法有深入的研究,对一些经典的文献总结分析可以得出离群点挖掘在异常检测中应用的基本框架,如图 1 所示。

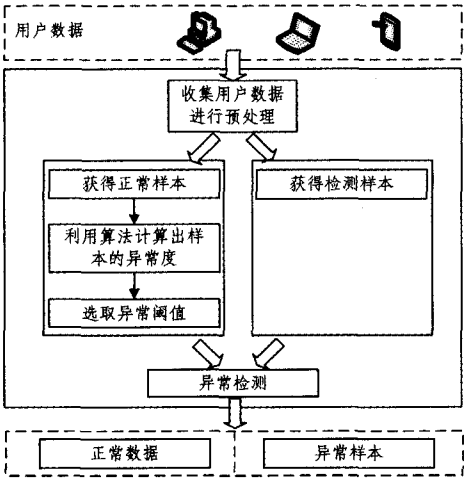


图 1　离群点挖掘在异常检测中应用的基本框架

通过分析上述基本框架,可以得出基于离群点挖掘的异常检测的一般过程。

Step1　首先对网络中的用户数据进行收集,此步骤是异常检测的初始准备阶段。对从网络上收集到的数据包、主机日志数据以及其他一些信息源数据通过传感器进行数据收集,在收集数据的过程中要注意收集对异常检测有用的数据,抛弃那些冗长而无关的数据,例如关于主机的数据和网络数据,只需要收集那些体现网络特征的信息(即 IP 数据包包头的信息),与用户行为无关的数据直接舍弃。在本文实验中使用了公共数据集 KDD CUP 99。

Step2　对 Step1 获得的数据进行预处理,并进行标准化处理,从而获得实验所需数据。Step1 采集到的数据一般具有冗长性、不一致性,以及不完整等特性,需要通过预处理来进行完善,数据的预处理主要包括数据清理、特征提取、格式转换、数据标准归一化等步骤。

Step3　将数据分成两个部分,一部分用于建立正常行为轮廓,另一部分用于最后的检测。

Step4　选取一个合适的衡量离群点离群的程度值,本文将该值定义为异常度。

Step5　根据异常度来进行异常检测,区分正常数据与异常数据。

3　基于混合式聚类算法的异常检测方法

本文提出了一种基于混合式聚类算法的异常检测方法 NADHC。该方法属于基于离群点挖掘的异常检测方法,其通过 k-中心点算法[7]找出簇中心,再通过新型的基于密度的聚类算法求解出异常度,找出异常行为。

3.1　簇中心的选择

假设现有正常样本集合为 A,样本的个数为 n,每个样本由 r 维的特征向量表示,利用 k-中心点算法将 A 中的对象分配到 k 个簇 $O_1,\cdots,O_k$,每个簇的中心为 $o_1,\cdots,o_k$。

在入侵检测中,使用相同手段攻击系统的黑客可能是类似的,并且形成一个簇,给入侵检测造成一定的干扰。为了解决上述问题,本文引入了"大簇"和"小簇"的概念,使用 α 区别二者,任何至少包含数据集中百分之 α 数据点的簇被视为大簇,而其余的簇被看作小簇。如图 2 所示,O_1,O_3 为正常样本簇,O_2 为异常样本簇,如果不考虑小簇的影响,很容易把类似 O_2 的小簇判断成正常样本,影响检测效果。为了避免小簇的干扰,本文只选取大簇作为参考对象。按照该标准从 $O_1,\cdots,O_k$ 中选择 $l(l\leqslant k)$ 个簇中心 $o_1,\cdots,o_l$ 作为正常样本集的参考样本。

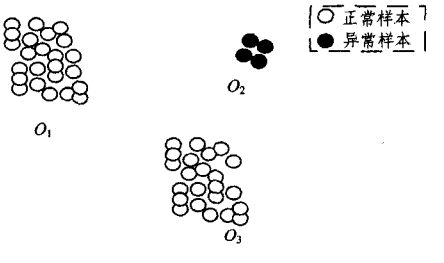


图 2　小簇中的异常样本

3.2　异常度的计算

本节将会用到曼哈顿度量算法,设 $i=(x_{i1},x_{i2},\cdots,x_{ip})$ 和 $j=(x_{j1},x_{j2},\cdots,x_{jp})$ 是两个被 p 个数值属性描述的对象。对象 i 和 j 之间的曼哈顿度量的定义如式(1)所示:

$$d(i,j)=|x_{i1}-x_{j1}|+\cdots+|x_{ip}-x_{jp}| \tag{1}$$

根据 k-中心点算法可知,少数异常样本会对簇中心产生较大的影响,而正常样本对簇中心的影响较小,因此可以通过增加若干个样本值观察簇中心的前后变化来衡量样本的性质[13]。图 3 展示了簇在增加若干个相同样本值后簇中心的前后变化情况。

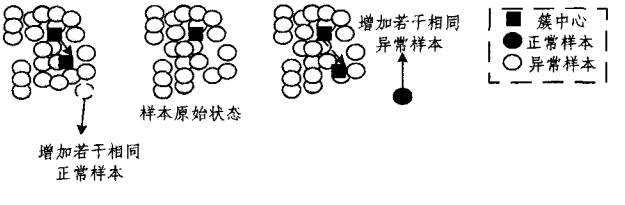


图 3　簇中心的变化情况

根据上述原理,假定 a 为 A 中的一个样本,根据式(1)通过曼哈顿度量方法计算 a 与 l 个簇中心 $o_1,\cdots,o_l$ 的距离值,

找出距离最近的簇中心 o_m，将该簇选为 a 的参考样本，然后增加 λ 个与 a 相同的样本，设增加后簇中心变为 o_m'。

通过密度值求出 o_m'，利用式(2)求出增加样本后每个样本相对于其他样本的密度。

$$F_i=\sum_{j=1}^{n+\lambda} e^{-\frac{\|a_i-a_j\|^2}{\delta}} \tag{2}$$

其中，$i=1,\cdots,n+\lambda$，δ 表示领域有效半径，将其取为 $n+\lambda$ 个样本间距离均方根的一半，即

$$\delta=\frac{1}{2}\sqrt{\frac{1}{(n+\lambda)(n+\lambda-1)}\sum_{i=1}^{n+\lambda}\sum_{j=1}^{n+\lambda}\|a_i-a_j\|^2} \tag{3}$$

式(2)与式(3)是针对传统的 k-means 算法进行的修改，由于传统的 k-means 算法初始化的聚类中心点的选取采用了随机选取的方式，该方式不能反映聚类本身的真实情况，因此该方式不适合用于本文选取新的中心点，为了能够避免这种随机选取方法的不稳定性，式(2)和式(3)综合考虑密度和距离两种方式，计算出每个样本相对于其他样本的密度，进而选出新的中心点。

比较各样本的密度，选择密度最大的样本作为簇中心，变化后的簇中心用 o_m' 表示，即

$$o_m'=\max\{F_1,\cdots,F_{n+\lambda}\} \tag{4}$$

计算出新的簇中心后，通过曼哈顿度量来衡量增加样本前后簇的变化，并引入新的变量异常度 $EOF(a)$ 来衡量样本 a 的异常情况，即

$$EOF(a)=dist(o_m,o_m') \tag{5}$$

式(5)中，本文选用曼哈顿度量距离来衡量由上述样本 a 增加前与增加后簇中心的相对距离，如式(1)所示。式中将相对距离作为异常度，能够直接衡量样本 a 相对于其距离最近的簇中所有样本的离群程度，因此若 $EOF(a)$ 值越大，表明 a 越有可能是异常样本。

3.3 异常阈值的选取

求解出异常度后，需要用一个异常阈值 ϵ 来判断样本是正常的还是异常的。对于一个网络异常检测方法，不同的异常阈值往往会对其检测性能产生较大的影响。一般而言，异常阈值设定得过高，异常检测方法的检测率会很低；异常阈值设定得过低，则其误报率会很高。本文将所有样本的异常度从大到小排列，并按顺序依次存储到数组 $b[i](i=1,\cdots,n)$ 中，将阈值设定为前几个数据的平均数，则阈值如式(6)所示：

$$\epsilon=\frac{\sum_{i=1}^{\Omega}b[i]}{\Omega}(1\leqslant\Omega\leqslant n) \tag{6}$$

通过上述方法求出阈值 ϵ 后，规定当 $EOF(a)\geqslant\epsilon$ 时，将样本 a 判定为异常对象，反之则判定为正常对象。本文提出 NADHC 算法的具体实现过程如图 4 所示。

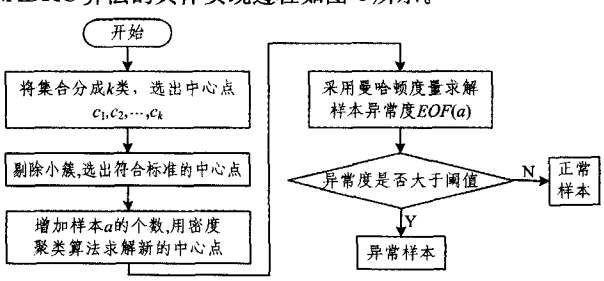


图 4 NADHC 算法模块流程

4 仿真实验

4.1 仿真数据集

为了证明本文算法的有效性，采用美国林肯实验室提供的 KDD 99 数据集进行实验仿真，KDD 99 数据包括 41 维向量特征，其中 38 个为数值型特征，其余 3 个为字符型特征。KDD CUPs 99 数据集中有 4 种入侵类型：Dos，U2R，R2L，Probe。本文为了降低仿真过程的复杂度，选取了 KDD CUP 99 数据集的“kddcup. data-10-percent”作为数据集。在实验仿真过程中，将训练集分成两个部分，其中 70%作为测试数据集，其余 30%用于训练以获取决策规则。在仿真中选择了 41 个特征项中的 18 个特征进行频繁集挖掘。在实验中为了消除数据集属性之间的差异性，对样本中的数据做标准归一化处理，方法如下：

$$s=\frac{1}{n}\sum_{i=1}^{n}(x_i-m) \tag{7}$$

其中，x_i 为数据值，m 为均值。由式(8)可得：

$$m=\frac{1}{n}\sum_{i=1}^{n}x_i \tag{8}$$

最终由式(9)可得标准数据集：

$$Y_i=\frac{x_i-m}{S} \tag{9}$$

本文的仿真环境为操作系统 Windows 8. 1，VC++6. 0，MATLAB7. 0，处理器为 AMD A4-3300M APU with Radeon (tm) HD Graphics 1. 90GHz，内存为 2GB。

4.2 评价指标的设定

为了较直观地验证算法对异常数据的识别性能，本文引用 3 个评价指标对算法的性能进行分析：检测率 DR(Detection Rate)、误警率 FAR(False Alarm Rate)和准确率 PR(Precision)。

检测率 DR 是指被正确检测到的异常数目 R 与检测数据集中总的异常数目 EN 之比，即：

$$DR=\frac{R}{EN} \tag{10}$$

误警率 FAR 是指没有被正确判断出所属类别的数目 E 与检测数据集中总的正常数目 RN 之比，即：

$$FAR=\frac{E}{RN} \tag{11}$$

准确率 PR 是指被正确判断出所属类别的数目 RA 与总的数目 N 之比，即：

$$PR=\frac{RA}{N} \tag{12}$$

漏检率 LR 是指正常样本被判断为异常的数目与检测数据集中总的异常数目之比，即：

$$LR=1-DR \tag{13}$$

4.3 参数的设定

在 NADHC 算法的实现过程中，有 3 个非常重要的参数，分别是 k 值、Ω 值(决定不同的阈值)以及重复样本的个数 λ。以上 3 个参数的不同对异常检测的效果可能会产生一定影响，因此选出 3 个参数的最优值从而改善算法的检测效果。具体的算法流程如算法 1 所示。方法中以误警率作为衡量标准，从 4，6，8，10 中选择出 k，从 500，1500，2500，3500 中选择

出 Ω,从 50,100,500,1000 中选择出 λ。将 30%的数据集代入算法中,最终发现当(k,Ω,λ)选择(6,2500,1000)时误警率最低。

算法 1　参数最优化算法流程

1. FOR k=(4,6,8,10)
2. FOR 为设定异常阈值的 Ω=(500,1500,2500,3500)
3. FOR 重复样本的个数 λ=(50,100,500,1000)
4. 设定 k 值,决定阈值的 Ω 以及重复样本的个数 λ
5. 将数据集带入运行每一种组合的情况
6. 计算出每种情况的误警率
7. 选择误警率最低的 k,Ω,λ

4.4　实验的结果与讨论

为了衡量本文算法的效果,将同样基于聚类算法的 CA&FCM 算法与本文算法在相同的实验环境下进行比较,所有的实验数据来自于美国麻省理工林克实验室提供的 KDD CUP 数据集,最终得到如表 1 所列的比较结果。

表 1　CA&FCM 与 NADHC 的检测性能比较/%

算法	检测率	误警率	准确率	漏检率
NADHC	97.9	2.65	92.62	2.1
CA&FCM	97.28	9.95	92.89	2.14

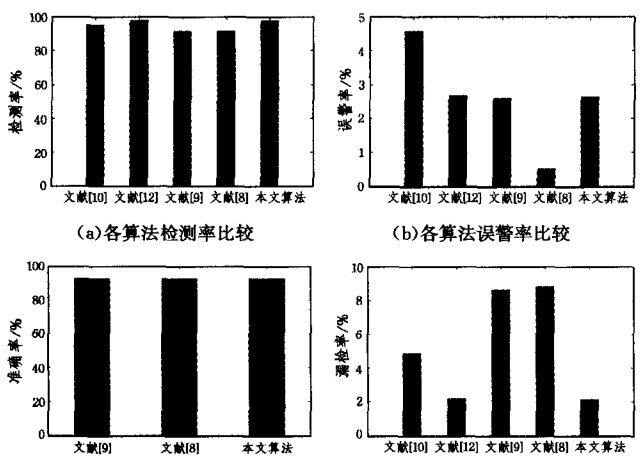
从表 1 中可以看出,虽然 CA&FCM 算法的准确率较高,但是该算法的误警率已高达 9.95%,而本文算法不仅有较高的检测率、准确率,而且其误警率和漏检率比 CA&FCM 算法低,从表中可以很明显地看出本文算法的优越性。

为了进一步凸显 NADHC 算法检测性能的优越性,本文对比了一些近年来的入侵检测方法。为了使比较具有意义,本文所选择的比较算法全部以 KDD CUP 99 作为测试集。表 2 列出了几种入侵检测方法的检测性能,除了文献[8]中的算法属于误用检测方法,其余 3 种都是异常检测的方法。

表 2　几种入侵检测方法的检测性能/%

评价指标	遗传优化 BP 网络算法[10]	RSAI-IID[12]	Strangeness[9]	多类模式分类算法[8]
检测率	95.18	97.86	91.40	91.20
误警率	4.57	2.68	2.60	0.50
准确率	NULL	NULL	92.50	92.87
漏检率	4.82	2.14	8.60	8.80

注:NULL 表示文献未给出指标。



(a)各算法检测率比较　(b)各算法误警率比较

(c)各算法准确率比较　(d)各算法漏检率比较

图 5　几种入侵检测算法检测性能的比较

结合表 1 与表 2 中的检测性能数值,绘制出图 5。从图 5(b)可以明显看出,文献[8]中的误用检测模型的误警率非常低,这主要是因为误用检测模型采用特征匹配的检测方式,使其能够以较低的误警率检测已知攻击。在其余的 4 种异常检测算法中,本文算法与 RSAI-IID 明显优于其余两种异常检测算法,虽然误警率不如文献[8],但检测率和准确率都明显高于文献[8]。总体来说,NADHC 模型提供了一个具有较强竞争力的异常检测性能。

结束语　针对现有异常检测算法计算量大、检测效率低等问题,本文将离群点挖掘技术用于异常检测中,提出了一种基于混合式聚类算法的异常检测方法,该方法通过基于距离的 k-中心点算法找出簇中心,进而通过增加若干重复样本的策略,结合基于密度的聚类算法求解出异常度,在求解过程中还考虑了"大簇"、"小簇"的影响,能够适应混入少量噪声数据的正常样本训练集。最后用 KDD CUP 99 数据集进行仿真实验,实验结果表明该算法具有较高的检测率和准确率,以及较低的误警率和漏检率,检测效率高。

参 考 文 献

[1] ANGIULLI F, BASTA S, PIZZUTI C. Detection and prediction of distance-based outliers[C]// Proceedings of the 2005 ACM Symposium on Applied Computing. ACM, 2005: 537-542.

[2] ZHU Y W, YANG J H, ZHANG J X. Anomaly detection based on traffic information structure[J]. Journal of Software, 2010, 21(10): 2573-2583. (in Chinese)
朱应武,杨家海,张金祥. 基于流量信息结构的异常检测[J]. 软件学报,2010,21(10):2573-2583.

[3] GAO Y, ZHOU W, HAN J Z, et al. An online log anomaly detection method based on grammar compression[J]. Chinese Journal of Computers, 2014, 37(1): 73-86. (in Chinese)
高赟,周薇,韩冀中,等. 一种基于文法压缩的日志异常检测算法[J]. 计算机学报,2014,37(1):73-86.

[4] BREUNIG M M, KRIEGEL H P, NG R T, et al. LOF: identifying density-basedlocal outliers[J]. ACM Sigmod Record, ACM, 2000, 29(2): 93-104.

[5] LEE Y J, YEH Y R, WANG Y C F. Anomaly Detection via Online Oversampling Principal Component Analysis[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(7): 1460-1470.

[6] CHANDOLA V, BANERJEE A, KUMAR V. Anomaly detection: A survey[J]. ACM Computing Surveys (CSUR), 2009, 41(3): 1-58.

[7] PARDESHI B, TOSHNIWAL D. Improved K-medoids clustering based on cluster validity index and object density[C]// Proc of the 2nd IEEE International Advance Computing Conference. 2010: 379-384.

[8] BADRAN K, ROCKETT P. Multi-class pattern classification using single, multi-dimensional feature-space feature extraction evolved by multi-objective genetic programming and its application to network intrusion detection[J]. Genetic Programming and Evolvable Machines, 2012, 13(1): 33-63.

(下转第 140 页)

参 考 文 献

[1] 李鸿培,于旸,忽朝俭,等. 2013 工业控制系统及其安全性研究报告[EB/OL]. (2013-06-24) [2016-03-11]. http://www.nsfocus.com.cn/content/details_62_881.html.

[2] CLAYTON M. Stuxnet malware is weapon out to destroy Iran's Bushehr Nuclear Plant[R]. Christian Science Monitor, 2010.

[3] BENCSÁTH B, PÉK G, BUTTYÁN L, et al. Duqu: A Stuxnet-like malware found in the wild[R]. CrySyS Lab Technical Report, 2011.

[4] Wikipedia Flame[EB/OL]. [2016-03-11]. http://en.wikipedia.org/wiki/Flame_(malware).

[5] 李鸿培,王晓鹏,王洋. 2014 绿盟科技工业控制系统安全态势报告[EB/OL]. (2014-09-15) [2016-03-11]. http://www.nsfocus.com.cn/content/details_62_887.html.

[6] 乌克兰电网遭黑客攻击[EB/OL]. (2016-01-06)[2016-02-17]. http://www.ftchinese.com/story/001065610.

[7] TANKARD C. Advanced Persistent threats and how to monitor and deter them[J]. Network security, 2011, 2011(8): 16-19.

[8] 王清. 0day 安全:软件漏洞分析技术[M]. 北京:电子工业出版社,2008.

[9] TAKANEN A, DEMOTT J D, MILLER C. Fuzzing for software security testing and quality assurance[M]. Artech House, 2008.

[10] SHAPIRO R, BRATUS S, ROGERS E, et al. Identifying vulnerabilities in SCADA systems via fuzz-testing[M]// Critical Infrastructure Protection V. Springer Berlin Heidelberg, 2011: 57-72.

[11] DEVARAJAN G. Unraveling SCADA Protocols: Using Sulley-Fuzzer[C]// Defon Conference. 2007.

[12] HOU Y, HONG Z, PAN F, et al. Model Based Automatic Fuzzing Script Generation[J]. Computer Science, 2013, 40(3): 206-209. (in Chinese)
侯莹,洪征,潘璠,等. 基于模型的 Fuzzing 测试脚本自动化生成[J]. 计算机科学,2013,40(3):206-209.

[13] 匡恩网络. 工控网络安全漏洞挖掘检测平台[EB/OL]. [2016-03-11]. http://www.kuangn.com/product/vulnerability.

[14] LI H. From "0" to "1": China's Industrial control protocol communictaion robustness test platform based on our independent research and development[J]. Automation Exhibition, 2015(9): 70-73. (in China)
李航. 从"0"到"1":我国自主研发工控协议通讯健壮性测试平台[J]. 自动化博览,2015(9):70-73.

[15] DEVARAJAN G. Unraveling SCADA protocols: Using Sulley-fuzzer, presented at the DefCon 15 Hacking Conference[EB/OL]. [2016-03-11]. http://www.defcon.org/html/defcon-15/dc15-speakers.html.

[16] KOCH R. Profuzz. [EB/OL]. [2016-03-11]. https://github.com/HSASec/ProFuzz.

[17] BYRES E J, KUBE N, DAN H. On Shaky Ground-A Study of Security Vulnerabilities in Control Protocols[C]// 5th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology, American Nuclear Society. 2006.

[18] General Purpose Fuzzer[EB/OL]. [2016-03-11]. http://www.vdalabs.com/tools/efs_gpf.html.

[19] BRATUS S, HANSEN A, SHUBINA A. LZFuzz: a fast compression-based fuzzer for poorly documented protocols[R]. Dartmouth Computer Science Technical Report TR 2008-634, 2008.

[20] DigitalBond. ICCPSic assessment tool set released, sunrise[EB/OL]. [2016-03-11]. http://www.digitalbond.com/2007/08/28/iccpsic-assessment-toolset-released.

[21] DYNAMICS M. Mu test suite[EB/OL]. [2016-03-11]. http://http://www.mudynamics.com/products/mu-test-suite.html.

[22] Wurldtech. Achilles test platform[EB/OL]. [2016-03-11]. http://www.wurldtech.com.

[23] Codenomicon. Defensics test platform[EB/OL]. [2016-03-11]. http://www.codenomicon.com/products /protocols.shtml.

[24] BeSTORM Software Testing Framework[EB/OL]. [2016-03-11]. www.beyondsecurity.com/black-box-testing.html.

[25] Peach[EB/OL]. [2016-03-11]. http://www.peachFuzzer.com.

[26] BOSSERT G, GUIHÉRY F, HIET G. Netzob: un outil pour la rétro-conception de protocoles de communication[C]// Actes Du Symposium Sur La Sécurité Des Technologies De Linformation Et Des Communications. 2012.

[27] ZHANG Y F, HONG Z, WU L F, et al. From-syntax based fuzzing method for industrial control protocols[J]. Application Research of Computers, 2016, 33(8): 2433-2439. (in Chinese)
张亚丰,洪征,吴礼发,等. 基于范式语法的工控协议 Fuzzing 测试技术[J]. 计算机应用研究,2016,33(8):2433-2439.

[28] Siemens. SIMATIC WinCC[EB/OL]. [2016-03-11]. http://www.wincc.com.cn.

(上接第 119 页)

[9] KUANG L, ZULEMINE M. An anomaly intrusion detection method using the csi-knn algorithm[C]// Proceedings of the 2008 ACM Symposium on Applied Computing. ACM, 2008: 921-926.

[10] HU M X. Intrusion detection algorithm based on BP neural network[J]. Chinese Journal of Computers, 2012, 38(6): 148-150. (in Chinese)
胡明霞. 基于 BP 神经网络的入侵检测算法[J]. 计算机工程,2012,38(6):148-150.

[11] HUANG M M, LIN B G. Fuzzy clustering method based on genetic algorithm in intrusion detection study[J]. Journal on Communications, 2009, 30(11): 140-145. (in Chinese)
黄敏明,林柏钢. 基于遗传算法的模糊聚类入侵检测研究[J]. 通信学报,2009,30(11):140-145.

[12] ZHANG L, BAI Z Y, LUO S S, et al. Integrated intrusion detection model based on rough set and artificial immune[J]. Journal on Communications, 2013(9): 166-176. (in Chinese)
张玲,白中英,罗守山,等 基于粗糙集和人工免疫的集成入侵检测模型[J]. 通信学报,2013(9):166-176.

[13] TSAI C F, CHENG K C. Simple instance selection for bankruptcy prediction[J]. Knowledge-Based Systems, 2012, 27(3): 333-342.