遗传算法在基于网络异常的入侵检测中的应用

张凤斌[1,2],杨永田[1],江子扬[2]
(1. 哈尔滨工程大学计算机科学与技术学院,黑龙江哈尔滨 150001;
2. 哈尔滨理工大学计算机与控制学院,黑龙江哈尔滨 150080)

摘 要: 考虑到基于误用的 IDS 不能有效检测未知入侵行为,而基于统计的异常检测法在建模时忽略了多变量在一段时间内的关系,提出了一种异常检测算法. 用滑动窗口将系统各属性表示为特征向量,从而将系统正常状态分布在 n 维空间中,并使用遗传算法进化检测规则集来覆盖异常空间. 经实验证明该方法提高了检测率.

关键词: 网络安全; 入侵检测系统; 遗传算法; 检测率
中图分类号: TP393.08 **文献标识码**: A **文章编号**: 0372-2112 (2004) 05-0875-03

Genetic Algorithms in Intrusion Detection Based on Network Anomaly

ZHANG Feng-bin[1,2], YANG Yong-tian[1], JIANG Zi-yang[2]
(1. *College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang* 150001, *China*;
2. *Computer and Control College, Harbin University of Science and Technology, Harbin, Heilongjiang* 150080, *China*)

Abstract: Intrusion detection system (IDS) based on misuse couldn't detect undefined intrusion behavior effectively and anomaly detection methods based on statistic ignored multivariable correlations of variables in amount of time. A novel anomaly detection algorithm was proposed. Feature vectors represent the system's attributes with slide window and distribute the normal system sate in n dimension space in this method. The method use rules set evolved with genetic algorithms to cover the abnormal space. Experiment shows that the algorithm improves the detection rate.

Key words: network security; intrusion detection system; genetic algorithms; detection rate

1 引言

目前的商业 IDS 大部分都是基于误用的,但是基于误用的 IDS 仅能检测出那些在知识库中事先定义好的入侵行为,因此此类 IDS 对未定义的攻击其检测率很低. 所以,对基于异常的 IDS 的研究就十分重要. 而传统的基于异常的算法主要是使用统计的方法. 统计方法针对代表系统状态的不同变量分别进行模型化,并对每个受监视的系统变量定义一个阀值,如果此变量超出了这个范围,就认为是异常[1]. 此方法的一个缺点是难以确定合适的阀值. 另一个缺点就是有些异常行为难以用纯统计的方法来建模,并且其在建模时忽略了多变量在一段时间内的它们之间的相互关系[2,3]. 这也是异常检测算法漏报率、误报率高的原因之一. 由于遗传算法其自学习、自适应和不需要其他辅助知识的特点可以克服其统计方法阀值和检测模型难以建立的缺点. 我们在表示系统状态时使用滑动窗口来表示一个系统变量在时间上的先后关系,并使用特征向量的方式来表示多变量之间的关系.

2 正常状态空间和检测规则的表示

设系统有 n 个属性,分别为 $A_1, A_2, \cdots, A_n$,由于每个属性在不同的时刻其值是不同的,所以在 t 时刻起的 w 个时刻内第 i 个属性 A_i 的值可以表示为 $A_i = (a_i^t, a_i^{t+1}, \cdots, a_i^{t+w-1})$,则 n 个属性相当于几个时间序列. 由于系统某一时刻的状态取决于多个属性之间的关系,因此在 t 时刻我们可得到向量 $x = (a_1^t, a_2^t, \cdots, a_3^t) \in [0.0, 1.0]^n$,此向量的每个分量就是属性值,其值可以映射到区间 $[0.0, 1.0]$ 上. 同时由于系统状态还和时间有关系,因此我们以滑动窗口的形式对每个分量再进行扩充,即, $x = (a_1^t, a_1^{t+1}, \cdots, a_1^{t+w-1}, a_2^t, a_2^{t+1}, \cdots, a_2^{t+w-1}, \cdots, a_n^t, a_n^{t+1}, \cdots, a_n^{t+w-1})$,并将 w 称为窗口大小. 由此系统各状态空间可以通过特征向量集合 $S \subseteq [0.0, 1.0]^n$ 来表示. 它包括了与系统所有可能状态相关的特征向量. 我们用特征向量的子集合 $Normal \subseteq S$ 代表系统的正常状态集合. 它的补集称作异常集合,记为 $Abnormal$,定义为 $Abnormal = S - Normal$. 下面为了表示方便,我们把特征向量简记为 $x = (x_1, \cdots, x_n)$.

由于此算法是基于网络的,我们将网络数据流经过处理得到提取出数据包量、时长、传输字节量等属性作为特征向量的分量来描述系统的状态. 所采用的属性越多,其判断的准确率越高,当然时空复杂度就越大[4]. 我们先使用最初的无入侵攻击的数据对系统进行训练,就可以得到正常向量集合 S,使用这个正常向量集合就可以初步检测异常了. 但是我们不能

收稿日期:2003-09-18;修回日期:2004-03-22
基金项目:黑龙江省自然科学基金(No. F9814);哈尔滨市学科后备带头人基金(No. 2003AFXXJ013)

认为凡是没有在正常空间中出现的点都是异常向量,因此使用一个偏离程度 v 来表示异常的严重程度,只要某个被检测的特征向量其偏离程度在我们可以允许的范围内,就认为此特征向量还是正常的. 如果以特征向量的分量为各方向轴形成 n 维空间,然后以各个正常特征向量为球心,v 为半径作超球体,把所有的超球体所包容的空间,称作正常空间,正常空间的外部空间称为异常空间. 由此可依据正常空间对网络数据进行检测,使用产生式的形式来检测,产生式的规则集定义如下

$$R^1: \text{If} \quad Cond_1, \quad \text{then } Abnormal$$
$$\cdots \qquad \cdots \qquad \cdots$$
$$R^m: \text{If} \quad Cond_m, \quad \text{then } Abnormal$$

其中

$Cond_i = x_1 \in [low_1^i, high_1^i]$ and $\cdots$ and $x_n \in [low_n^i, high_n^i]$;

$(x_1, \cdots, x_n)$ 为特征向量

$[low_j^i, high_j^i]$ 为在第 i 条规则 R^i 的条件下,分量 x_j 的上下限值.

第 i 条规则 R^i 定义了一个超矩形,如果被检测的数据(例如向量 x)在这个超矩形内(记为 $x \in R^i$),则认为产生了异常,进行报警. 而多条规则可以形成多个超矩形对正常空间的外部空间(异常空间)进行覆盖. 我们所要做的就是使用遗传算法来进化出这样一个规则集.

3 使用遗传算法进化检测规则集

3.1 适应值函数的确定

一条规则的好坏可以按此规则覆盖异常空间的体积大小,包含正常向量的个数来衡量.

规则 R 所包含的正常向量的数量如下式所示:

$$num\_elements(R) = |\{x \in S \mid x \in R\}| \tag{1}$$

规则 R 所产生的超矩形的体积如下式所示:

$$volume(R) = \prod_{i=1}^{n}(high_i - low_i) \tag{2}$$

$fitness(R) = volume(R) - C \cdot num\_elements(R)$ 即为适应值函数,其中 C 是惩罚系数. 如果一条规则覆盖正常例子的话,C 确定了此规则应受的惩罚. 系数越大,惩罚值越大.

由于我们要用多条规则来覆盖异常空间,为了保证进化出不同的规则,我们使用小生境算法,其思想就是通过一个判断个体之间的相似程度的函数来调整群体中每个个体的适应度,从而维护群体的多样性[5]. 这个相似判断函数通过判断两个超矩形的交集的体积来计算两条规则的相似程度,其伪代码如下:

$fitness = raw\_fitness_R$

for $(each\ R^j \in ruleSet)\{$

$fitness_R = raw\_fitness_R - volume(R \cap R^j)$

$\}$

3.2 遗传算子的选择

3.2.1 编码 规则中的上下限值是实数值,考虑到算法性能,我们进行规则进化的时候采取二进制编码方法,然后在将二进制编码映射到实数区间上. 以网络速率为例,十兆网络其最大速度为 1Mb/s,其区间为 [0,1000],我们以 0.1k 为变化的基准,即精确到小数点后一位,则将区间分为 1×10^4 等分,因为 $8192 = 2^{13} < 1 \times 10^4 < 2^{14} = 16384$,因此就使用 14 位二进制编码.

3.2.2 选择算子 选择算子采取排序选择法,排序选择法的个体选择概率为:

$$p(x^j) = \frac{1}{n}\left(\eta^+ - \frac{\eta^+ - \eta^-}{n-1}(j-1)\right), \quad j = 1, 2, \cdots, n$$

其中 η^+ 为适应函数值最好的个体的在选择操作后的期望数量,η^- 适应函数值最差的个体的在选择操作后的期望数量,其他个体的期望数量按等差序列排列.

3.2.3 交叉算子 采取多点交叉. 对于选定的两个个体位串,随机选择多个交叉点,构成交叉点集合:

$d_1, d_2, \cdots, d_K \in \{1, 2, \cdots, L-1\}$, $d_k \leq d_{k+1}$, $k = 1, 2, \cdots, K-1$

将 L 个基因位划分为 $K+1$ 个基因位集合:

$Q = \{l_k, l_k + 1, \cdots, l_{k+1} - 1\}$, $k = 1, 2, \cdots, K+1$, $l_1 = 1$, $l_{K+2} = L+1$

其算子形式为

$$O(p_c, k): \quad x'_{1i} = \begin{cases} x_{2i}, & \text{若 } i \in Q, k \text{ 为偶数} \\ x_{1i}, & \text{否则} \end{cases}$$
$$x'_{2i} = \begin{cases} x_{1i}, & \text{若 } i \in Q, k \text{ 为偶数} \\ x_{2i}, & \text{否则} \end{cases}$$

按照泊松分布选择交叉点数:

$$P(x) = \frac{\lambda^x}{x!}e^{-\lambda}, \quad E(x) = D(x) = \lambda = g(L) > 0$$

其中,x 为交叉点数,其均值 $E(x)$ 和方差 $D(x)$ 为位串长度的函数.

3.2.4 变异算子 对于给定的个体 $x = a_1 a_2 \cdots a_L$,其具体操作如下:

$$O(p_m, x): a'_i = \begin{cases} ran(a_i), & \text{若 } y_i \leq p_m \\ a_i, & \text{否则} \end{cases}, \text{其中 } i \in \{1, 2, \cdots, L\}$$

其中 $ran(x)$ 为随机函数,随机生成一个字符集中的值. y_i 为第 i 位基因发生变异的均匀随机变量. p_m 为设定的变异概率. 由于变异的概率较小,如果针对每个个体的每个基因都做此测试的话,将造成资源浪费[6],因此,做如下变通. 先计算个体发生变异的概率:$p_m(x^j) = 1 - (1 - p_m)^L$, $j = 1, 2, \cdots, n$ 对个体 x^j,随机的生成一个数 y(随机变量),若 $y \leq p_m(x^j)$,则对此个体进行变异. 再计算发生变异的个体上的基因变异的概率:

$$p'_m = \frac{p_m}{p_m(x^j)} = \frac{p_m}{1 - (1 - p_m)^L}$$

3.2.5 算法流程图(如图 1)

4 实验和结论

我们使用麻省理工学院林肯实验室提供的 1999 年的从模拟网络中搜集的网络攻击数据进行测试. 我们选取五个参数作为我们检测的属性,它们是字节数/分钟、包个数/分钟、连接持续时间、源地址字节数、目的地址字节数,这些属性每分种采样一次. 使用第一周的不包含入侵行为数据作为训练数据,第二周的数据作为测试数据. 表 1 为不同窗口,不同级

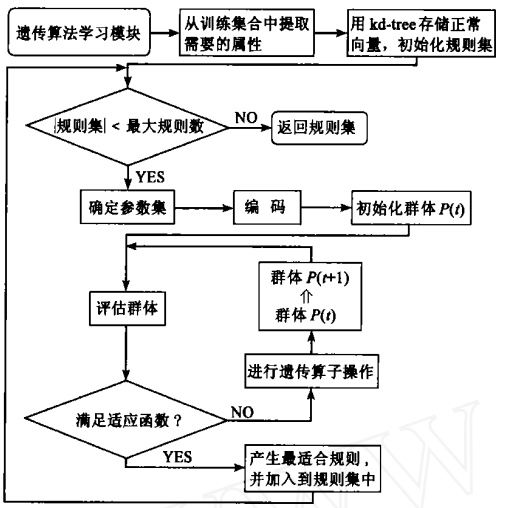

图 1　基于遗传算法的检测算法流程图

别下检测到的入侵次数. 图 2 和图 3 分别为在窗口大小为 3 和 1 的情况下的检测图.

表 1　不同窗口的检测结果

	窗口大小为 1	窗口大小为 3
检测率	83.5 %	93 %
误报率	3 %	1 %
平均规则数	2.1	49

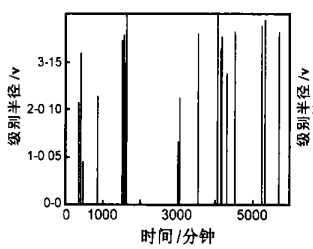

图 2　窗口大小为 3 情况下的检测图

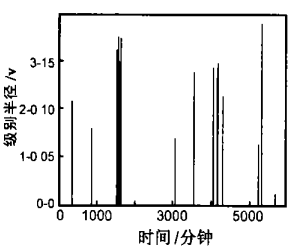

图 3　窗口大小为 1 情况下的检测图

我们设定遗传算法参数如下:种群大小 100,经历 1500 代,变异率 0.1,交叉率 0.2,经实验证明在窗口大小为 3 情况下,最好检测率为 93 %,最大误报率为 1 %. 此算法避免了传统统计方法的困难,算法自适应性强,提高了对未知入侵行为的检测率. 对算法的一些参数的最优选取和规则的最优覆盖方式还需要作更深入的研究.

参考文献:

[1] E Eskin. Anomaly detection over noisy data using learned probability distributions[A]. Proceedings of the 17th International Conference on Machine Learning[C]. San Mateo, CA: Morgan Kaufmann, 2000. 255 - 262.

[2] T Lane, C Brodley. Temporal sequence learning and data reduction for anomaly detection[J]. ACM Trans Info System Security, 1999, 2: 295 - 331.

[3] T Lane, C E Brodley. Data reduction techniques for instancebased learning from human/computer interface data[A]. Proceedings of the 17th International Conference on Machine Learning[C]. San Mateo, CA: Morgan Kaufmann, 2000. 519 - 526.

[4] D Dasgupta, F Gonzalez. An immunity-based technique to characterize intrusions in computer networks[J]. IEEE Transactions on Evolutionary Computation, 2002, 3(6): 281 - 291.

[5] E Zitzler, L Thiele. Multi-objective evolutionary algorithms: comparative case study and the strength pareto approach[J]. IEEE Trans of Evolutionary Computation, 1999, 3(4): 257 - 271.

[6] M Srinivas, M Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Trans on Systems, Man, and Cybernetics, 1993, 24(4): 656 - 667.

作者简介:

张凤斌　男, 1965 年生于哈尔滨市, 哈尔滨工程大学博士研究生, 哈尔滨理工大学副教授, 研究方向为网络安全与保密.

杨永田　男, 1940 年生于哈尔滨市, 哈尔滨工程大学教授, 博士生导师, 主要研究方向为计算机网络安全.