第34卷 第24期 计 算 机 工 程 2008年12月
Vol.34 No.24 Computer Engineering December 2008
·安全技术· 文章编号：1000—3428(2008)24—0172—03 文献标识码：A 中图分类号：TP309.5

基于关联规则的未知恶意程序检测技术

章 文，郑 烇，帅建梅，陈 超
(中国科学技术大学自动化系，合肥 230027)

摘 要：针对当前基于特征码病毒检测技术不能检测出未知病毒的缺点，通过研究某些病毒及其变种版本在执行过程中应用程序接口(API)调用序列的规律，提出一种基于数据挖掘的检测技术，采用 Apriori 算法从已知病毒的 API 调用序列中提取有价值的关联规则，用于指导病毒检测。实验结果表明该方法对未知病毒检测有良好的效果。

关键词：关联规则；未知恶意程序；应用程序接口

New Malicious Executables Detection Based on Association Rules

ZHANG Wen, ZHENG Quan, SHUAI Jian-mei, CHEN Chao
(Department of Automation, University of Science & Technology of China, Hefei 230027)

【Abstract】 In order to improve the current malicious detection technology based on signature, this paper presents a method based on data mining. By researching the rules of API calling sequences during executing viruses, the method uses Apriori algorithms to extract some valuable related rules which hide out in a lot of API calling sequences of viruses. These rules can be used to detect Viruses. Experimental results validate its effection.

【Key words】 association rules; new malicious executables; API

近年来，病毒模糊技术的发展给反病毒工作带来了极大的挑战。传统的基于特征码的病毒检测技术对各种变形病毒无能为力。有一些病毒经过变形之后，能够逃脱McAfee, Kaspersky等主流反病毒软件的检测[1]。本文利用数据挖掘算法从大量已知病毒的应用程序接口(API)调用序列中提取有价值的关联规则，用于指导对变形病毒的识别与检测。

1 基于关联规则的病毒检测方法

1.1 病毒检测模型的生成

首先将收集到的大量病毒样本输入该模型。每次输入的样本都是某一家族的病毒样本，这主要基于以下 2 个原因：(1)降低样本集的大小，提高算法的速度；(2)屏蔽来自其他家族的病毒样本的干扰，使提取的规则更可靠。由于样本都是 PE 格式的文件，经特定工具提取后可以形成 DLL 特征集和 API 特征集，根据一些原则进行聚类处理，将这 2 个特征集分别分为几个子集。之后将关联规则挖掘的 Apriori 算法应用于这些子集，生成一些规则集，这些规则集集中在一起就形成了规则库，可以用来指导未知病毒的检测，具体过程如图 1 所示。

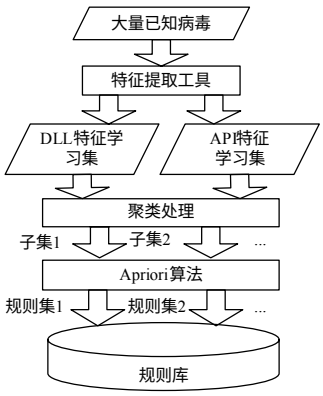

图 1 关联规则生成流程

1.2 未知恶意程序检测流程

当一个未知程序需要检测时，先利用图 1 中的特征提取工具将该程序的特征提取出，包括其使用的动态链接库(DLL)和系统函数调用。

检测引擎接收未知程序的特征，并在 DLL 规则库中搜索，如果发现有与之匹配的规则，则继续搜索 API 规则库，否则认为该程序为正常程序。若在 API 规则库中也找到与之匹配的规则，则可以认定其为恶意程序，否则，认为该程序为正常程序，具体过程如图 2 所示。

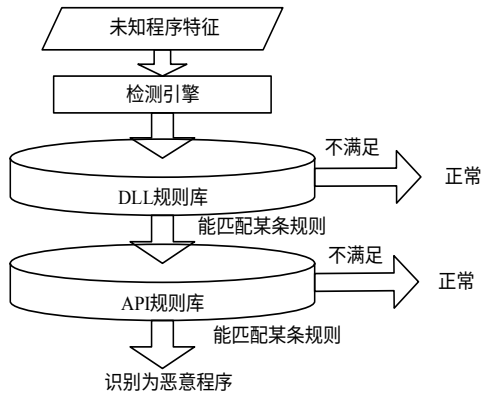

图 2 未知恶意程序检测流程

2 聚类处理

即使是同一家族的病毒，由于经过多次变种，它们所调用的API个数可能相差很多。表 1 列出了Win32.SdBot家族中一些病毒API的个数。可以看出，该家族同时存在仅调用 8

基金项目：国家“863”计划基金资助项目(2006AA01Z449)
作者简介：章 文(1986－)，男，硕士研究生，主研方向：信息安全；郑 烇，副教授；帅建梅，高级工程师；陈 超，硕士研究生
收稿日期：2008-01-30 **E-mail：**green@mail.ustc.edu.cn

个API和调用 86 个API的病毒变种，并且这个病毒样本集可以根据API调用个数的不同分成 2 个部分(例如Win32.SdBot.04.f和Win32.SdBot.02 就分别属于 2 个部分)。事实上，可以应用聚类算法将病毒样本集分成几个部分分别研究，这利用了Savasere划分[3]的思想。

表 1　一些 SdBot 病毒变种 API 个数

病毒名称	调用 API 个数
Win32.SdBot.02▲	8
Win32.SdBot.04.c▲	11
Win32.SdBot.04.d▲	8
Win32.SdBot.04.f★	74
Win32.SdBot.h★	74
Win32.SdBot.ig★	86
Win32.SdBot.05.d▲	8
Win32.SdBot.05.w▲	8
Win32.SdBot.kd▲	8
Win32.SdBot.kp▲	13

本文采用C-均值算法[2]对病毒样本进行聚类处理。从病毒特征集中选择每个样本的API个数作为C-均值算法的特征集：$X=\{x_1, x_2, \cdots, x_N\}$。算法的基本思想是：根据观察取定需要区分的类数$C$并从$X$中选择$C$个初始聚类重心。之后按照最小距离原则将$X$中各项分配到$C$类中的每一类，重新计算各类重心，并调整$X$中各项的类别，如此循环，直到各类中的项不再变化。

根据 C-均值算法，表 1 中的病毒样本集可以分成 2 类，分别用▲和★区分。

病毒样本特征集经过该算法被分为一个个小的子集，对这些子集分别应用关联规则算法即可得到最后的关联规则。需要指出的是，该方法在病毒样本数目比较庞大时才有意义，因为关联规则挖掘是一种从大规模数据中挖掘知识的方法，样本数目若较小则不推荐用本方法预处理样本集。

3　关联规则算法

本文采用Apriori算法进行关联规则的挖掘。Apriori算法[4]的基本思路为：遍历整个数据库，在事先指定的最小支持度下，获得数据库中所有频繁 1-项集，然后对这些 1-项集进行自连接扩展，得出候选 2-项集。再次扫描数据库，对候选集进行挑选，得到满足最小支持度的频繁 2-项集，依此类推，直到无法发现新的频繁项集为止。

对于一个给定的频繁项集$Y=\{I_1,I_2,\cdots,I_k,k\geq 1\}$，可以导出如下规则：$X\rightarrow I_j$，其中，$X=\{I_1,I_2,\cdots,I_{j-1},I_{j+1},\cdots,I_k\}$,$1\leq \forall j\leq k$。再计算每条规则的信任度：

$$confidence\cdot(X\Rightarrow I_j)=P(I_j\mid X)=\frac{count(X\cup I_j)}{count(X)}$$

其中，$count(X\cup I_j)$为数据库当中包含项集$X\cup I_j$的记录数目；$count(X)$为包含项集X的记录数目。显然，这样的规则一共有k条，计算每条规则的信任度，并与指定的最小信任度作比较，保留大于最小信任度的规则，丢弃剩余的规则。最后得到的规则即算法的最终结果。

4　基于关联规则的病毒检测

4.1　数据预处理

在对病毒样本进行特征提取、聚类分析并形成一些小的子集后，还必须对得到的数据进行一些预处理。对某些家族的病毒来说，一些DLL和API出现的概率等于 1。例如，在随机抽取的Specrem病毒样本中，MSVBVM60.dll出现的概率为100%；而在Antilam家族，API函数ImageList_Add出现的概率也为 100%。因此，在自连接扩展生成候选集时可以先不考虑这样的必然事件，生成最后的频繁项集时再将这些DLL和API直接添加进去。这样有助于减少每次自连接形成的候选集数目[5]：假设频繁 1-项集的数目为 41，那么进行　2 次、3 次、4 次扩展生成的候选集数目最大可以达到C_{41}^{2}=820, C_{41}^{3}=10 660, C_{41}^{4}=101 270，假设经过数据预处理后，频繁 1-项集的数目减少了 2，那么连接数目则分别为C_{39}^{2}=741, C_{39}^{3}=9 139, C_{39}^{4}=82 251。可以看出，仅将基数减少了 2，自连接生成的候选集数目就有了数量级的变化。这对算法性能的提高无疑是至关重要的。

4.2　关联规则

在应用 Apriori 算法得到频繁项集后，可以由此项集推导出最后的关联规则：

(1)产生每个频繁项集 l 的所有非空子集。

(2)对于每个 l 的非空子集 s，若

$$\frac{count(l)}{count(s)}\geq min\_conf$$

成立，则产生一个关联规则 $s\Rightarrow(l-s)$。其中，$count(l)$和$count(s)$分别指特征学习集中包含 l 和 s 的数目；min_conf 指事先指定的最小信任度阈值。

表 2 给出了 4 条关于病毒家族 SdBot 的关联规则。其中，前 2 条是关于 DLL 的关联规则；后 2 条是关于 API 的规则。

表 2　SdBot 关联规则示例

前项	后项	支持度/(%)	信任度/(%)
kernel32,advapi32,shell32,wininet,crtdll	wsock32	60	82
kernel32,advapi32,shell32,wininet,mpr	user32	20	100
LoadLibraryA,GetProcAddress,atoi, ExitProcess,RegCloseKey	ShellExecuteA	40	100
LoadLibraryA,GetProcAddress, InternetOpenA,ExitProcess	RegCloseKey, ShellExecuteA	40	100

如对于第 1 条表示规则：

$$\text{kernel32}\wedge\text{advapi32}\wedge\text{shell32}\wedge\text{wininet}\wedge\text{crtdll}\Rightarrow\text{wsock32}$$

支持度 60%表示在样本集中，kernel32, advapi32, shell32, wininet, crtdll, wsock32 这 6 个动态链接库同时出现在 SdBot 病毒中的概率为 60%。信任度 82%则表示当 SdBot 病毒程序中出现 kernel32, advapi32, Shell32, wininet, crtdll 这 5 个动态链接库时，wsock32 也出现在该程序中的概率为 82%。这些规则最终存储在 DLL 规则库和 API 规则库中，以便病毒检测时查询。

4.3　判决方法

在对一个未知程序进行检测时，先用特征提取工具提取出其 DLL 和 API 等信息并将其输入检测引擎。这里主要介绍 API 特征的匹配法，DLL 特征匹配可参考该方法。

该引擎从规则库的第 1 条规则开始，在 API 特征中寻找该规则的前项，如不能够找到，则认为此规则不适用于此程序的特征信息，转而查询规则库下一条规则，倘若查询完毕也找不到一条适用规则，则不对该程序进行认定(不认定其为

合法程序也不认定其为恶意程序)；如果找到一条适用规则，再寻找当前规则的后项，如寻找成功，则认为匹配成功，判定该程序为恶意程序；如不成功，则判定其为正常程序。

5 实验结果与分析

基于上述思想实现了一个基于关联规则的未知程序检测平台 SBAR(Scaner Based on Association Rules)。测试内容包括与目前主流的反病毒软件在未知病毒检测上作比较和对该平台误报率和漏报率进行观察。该平台以 4 120 个病毒、25 个病毒家族为样本训练规则库。

目前病毒一般采用多态和变形 2 种迷惑方式逃避检测[7]。前者一般采用如下 3 种方式：(1)插入无效代码，如在代码中插入一些nop指令。(2)代码变换，如改变代码原来的顺序，同时加入一些跳转语句以保持原来的语义。(3)对寄存器进行重分配操作。后者则采用加密技术对代码的恶意行为进行加密，并在代码执行的过程中进行相应的解密。

文献[6]提出了基于多态的变形技术的实现，本文利用该技术对 SdBot, HackTack, Ptakks, Littlewitch 这 4 类病毒进行了模糊化，用瑞星、江民、McAfee、Nod32 以及检测平台 SBAR 分别对这些变形病毒进行扫描，结果如表 3 所示。可以看出，在检测变形病毒上，基于关联规则的扫描程序检出率略高于 McAfee 和 NOD32，比瑞星、江民等更具优势。

表 3 各种反病毒平台对恶意代码的扫描结果

	瑞星	江民	NOD32	McAfee	SBAR
SdBot	√	√	√	√	√
SdBot.A	×	×	√	√	√
SdBot.B	×	×	√	×	√
HackTack	√	√	√	√	√
HackTack.A	√	×	√	√	√
HackTack.B	×	×	×	×	√
Ptakks	√	√	√	√	√
Ptakks.A	√	×	√	√	√
Ptakks.B	×	×	√	×	√
Littlewitch	√	√	√	√	√
Littlewitch.A	×	√	√	√	√
Littlewitch.B	×	×	×	√	×

另外各用 412 个病毒和正常程序测试 SBAR 的误报率和漏报率。实验中将平台的学习机制中的最小支持度阈值始终定为 5%，而对最小信任度阈值进行了一些调整，结果如表 4 所示。

表 4 信任度对误报率、漏报率的影响 (%)

Min_Conf	误报率	漏报率
20	17.49	5.28
40	10.17	7.46
60	8.13	9.79
80	6.80	14.23

可以看出，当最小信任度阈值增加后，误报率逐渐减小，漏报率逐渐增加，因为信任度阈值增加后，生成的关联减少，造成未知程序的匹配概率减小，这就使得正常程序被判为恶意的概率减小(误报率减少)，恶意程序被判为正常程序的概率增加(漏报率增加)。因此，权衡两者得出，置信度在 40%~60%之间为理想值。

6 结束语

本文介绍了一种基于关联规则的未知恶意程序检测技术，给出实现的细节，并进行了实验。根据实验结果，本文实现的 SBAR 能够检测出一些基于病毒特征码的反病毒软件所不能检测出的病毒，但是在误报率和漏报率方面仍需提高。对正常程序的特征进行学习对于降低误报率起到比较重要的作用，因此，今后将对其作进一步的研究。

参考文献

[1] Xu Jianyun, Sung A H, Chavez P. Polymorphic Malicious Executable Scanner by API Sequence Analysis[C]//Proc. of the 4th Int'l Conference on Hybrid Intelligent Systems. Washington, USA: [s. n.], 2004.

[2] 孙即祥. 现代模式识别[M]. 长沙: 国防科技大学出版社, 2001.

[3] Kamber M, Han Jiawei, Chiang Jenny. Metarule-guided Mining of Multi-dimensional Association Rules Using Data Cubes[C]//Proc. of the 3rd Int'l Conf. on Knowledge Discovery and Data Mining. California, USA: [s. n.], 1997: 207-210.

[4] Agrawal R. Mining Association Rules Between Sets of Items in Large Database[C]//Proc. of the ACM SIGMOD Conference on Management of Data. Washington, USA: ACM Press, 1993.

[5] 朱秋萍, 毛平平, 罗 俊. 基于关联规则的入侵检测系统[J]. 计算机工程与应用, 2004, 40(9): 160-162.

[6] Sung A H, Xu Jianyun, Chavez P. Analyzer for Vicious Executables[C]//Proc. of the 20th Annual Computer Security Applications Conference. Washington, USA: [s. n.], 2004: 326-340.

[7] Christodorescu M, Jha S, Seshia S A, et al. Semantics Aware Malware Detection[C]//Proceedings of the 2005 IEEE Symposium on Security and Privacy. Oakland, USA: IEEE Press, 2005.

(上接第 171 页)

[4] Sun Hung-Min. Threshold Proxy Signatures[J]. IEEE Proceedings of Computers and Digital Techniques, 1999, 146(5): 259-263.

[5] 王晓明, 张 震, 符方伟. 一个安全的门限代理签名方案[J]. 电子与信息学报, 2006, 28(7): 1308-1311.

[6] Hsu C L, Wu T S, Wu T C. New Nonrepudiable Threshold Proxy Signature Scheme with Known Signers[J]. Journal of Systems and Software, 2001, 58(9): 119-124.

[7] Hwang Min-Shiang, Lu Eric Jui-Lin, Hwang Iuon-Chang. A Practical (t, n) Threshold Proxy Signature Scheme Based on the RSA Cryptosystem[J]. Knowledge and Data Engineering, 2003, 15(6): 1552-1560.

[8] 范恒英, 何大可, 卿 铭. 公钥密码新方向: 椭圆曲线密码学[J]. 通信技术, 2002, 27(7): 82-84.

[9] 秦志光. 密码算法的现状和发展研究[J]. 计算机应用, 2004, 24(2): 124.