信息安全研究 第2卷 第1期 2016年1月
Journal of Information Security Research Vol.2 No.1 Jan. 2016

基于数据挖掘和机器学习的恶意代码检测方法

廖国辉　　刘嘉勇
(四川大学电子信息学院　成都　610065)
(sculiaoguohui@yeah.net)

A Malicious Code Detection Method Based on Data Mining and Machine Learning

Liao Guohui and Liu Jiayong
(*College of Electronics and Information Engineering, Sichuan University, Chengdu* 610065)

Abstract　In recent years, malicious code uses flower instructions and packers and other methods to bypass the detection of antivirus software, while the identification of existing methods for variants of malicious code can not be accurate. In the view of threat of malicious code on computer security and features of fast spread and wide variety, this paper uses the data mining and machine learning method to recognize and detect malicious code. Firstly, it proposes a malicious code detection framework based on data mining and machine learning, and extracts the code features from text structure layer, byte layer and code layer respectively. Secondly, it adapts the principal component analysis to reduce the dimension of combined feature matrix. Finally, it recognizes and classifies the malicious code using various classification methods. The result shows that the accuracy rate of every classification method based on combined feature matrix is higher than 90%, and among them, the method of decision tree gets the best. It is able to achieve effective detection of variants of malicious code, and provide a very effective method for malware killing to detect the variants of malicious code.

Key words　malicious code; multidimensional feature; data mining; machine learning; code detection

摘　要　近年来,恶意代码采用花指令以及加壳等方法来绕过杀毒软件的检测,而现有的方法对于变种恶意代码无法准确的识别.鉴于恶意代码对计算机安全性的威胁以及恶意代码传播速度快、种类繁多的特点,采用数据挖掘和机器学习的方法对恶意代码进行识别与检测.首先,提出了一种基于数据挖掘和机器学习的恶意代码检测框架,并分别从文本结构层、字节层、代码层3个角度提取了代码特征;然后采用主成分分析的方法对3种层次的组合特征进行特征降维;最后采用不同的分类方法对恶意代码进行识别与分类.分类结果表明:基于组合特征的不同分类方法对恶意代码的识别准确率都在90%以上,能够实现对变种恶意代码的有效检测,为恶意代码查杀提供了一种十分有效的方法,其中决策树分类方法的识别准确率最优.

关键词　恶意代码;多维特征;数据挖掘;机器学习;代码检测

中图法分类号　TP309

收稿日期:2015-12-25

恶意代码是一种引起计算机故障、信息外泄、破坏计算机数据、影响计算机系统正常使用的程序代码，是威胁计算机安全的重要形式之一，具有非授权性和破坏性 2 种特征形式[1-2]. 近年来，由于恶意代码在网络上的广泛传播而引起的网络安全事件数量正在逐年增加，据相关统计资料显示，从 20 世纪 90 年代至今，由恶意代码造成的网络安全事件数量每年增幅达 50%以上[3-4]. 这些网络安全事件的产生不仅反映了系统和网络安全的脆弱性，而且给目前以因特网为基础设施的经济发展造成巨大损失[5].

由于当前恶意代码的数量非常巨大，新的恶意代码出现的速度也越来越快[6-7]，传统的检测技术由于其检测速度、效率等问题已经无法应对当前恶意代码检测的需求，鉴于上述问题，本文采用数据挖掘和机器学习的方法，从恶意代码提取的各个特征出发，自动学习隐含在恶意代码中稳定的模式与规律，并利用该模式与规律进行检测，相对传统检测方法而言，分析速度更快，检测效率更高并具有很好的对未知恶意代码的检测能力[8-9].

1 关键技术

1.1 恶意程序

目前恶意程序分为 6 类[10]：病毒、蠕虫、木马、僵尸程序、间谍程序和流氓软件. 这 6 类恶意程序都会给用户带来巨大的损失. 而且，为了对抗反病毒程序，这些恶意程序具有反调试技术、反虚拟机技术、加壳技术、对抗安全软件的技术. 但是，总体来说，采用恶意程序的动态特征和静态特征就能反映恶意程序在计算机中的活动变化. 其中，动态特征包括恶意代码动态调用序列及其调用参数特征、系统调用关系图、控制依赖关系图、数据依赖关系图、恶意代码对系统资源（文件、注册表、进程、网络）的操作情况等. 静态特征包括恶意代码文件结构特征、字节序列特征、指令序列特征、函数调用关系图、系统调用关系图、控制流图、数据流图等.

为了综合考虑效率与成本的开销，本文在特征提取时只采用静态特征提取方案，同时为了更加全面地描述恶意代码，充分发挥静态特征的优势，本文从恶意代码的多个静态层次上提取所需的多维特征，包括文件结构层的结构特征、字节层的字节序列特征、指令层指令序列特征.

1.2 基于数据挖掘和机器学习的恶意代码检测架构

依据上述思想，本文提出的检测算法的基本框架如图 1 所示，从图 1 可以看出，检测过程主要分为 2 个阶段：训练阶段与测试阶段. 训练阶段主要完成样本的训练，包括样本静态反汇编、特征提取与选择、集成分类器的构建过程，其中静态反汇编主要完成判断恶意代码是否加壳并依据壳的类型选择相应的脱壳程序进行正确脱壳的过程. 特征提取主要完成实验样本的结构特征、字节序列、指令序列、基于语义的静态调用序列特征的特征提取过程，为了便于后续分类算法的学习，对于维度很高的特征必须进行特征降维与冗余特征约简处理. 集成分类器的构建主要完成多分类器的训练、选择和集成过程. 测试阶段主要完成样本的测试.

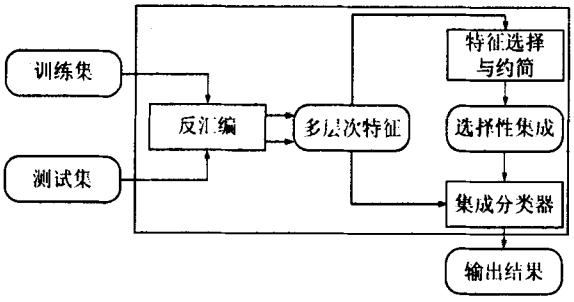

图 1 恶意代码检测架构

2 代码数据预处理

2.1 实验样本获取

本文所研究的源数据采用 VX Heavens 中的样本数据，从该网站下载 PE 格式的恶意程序样本 25 585 个，其中正常程序数量为 4 730 个，每个 PE 格式的恶意程序样本都由 DOS 文件头、DOS 块、PE 文件头、节表、代码、数据和资源节等几部分组成.

2.2 实验样本划分

获取的实验样本中包含木马、病毒、蠕虫、Rootkit、下载者、正常 6 类恶意代码. 在实验样本中各种类别的恶意代码如表 1 所示. 由表 1 可知，下载的数据集中每种类别的代码数量并不均衡，

容易在后续的恶意代码分类与识别过程中出现因类别样本不均衡而导致分类率下降的情况. 鉴于此,本文以 Rootkit 的数量为基准,在其他类别的代码中随机选取 2 768 个代码样本,采用交叉验证的方法,对恶意代码进行分类与识别.

表 1 实验样本类别及其数量

类别	数量
木马	3 534
病毒	6 001
蠕虫	3 225
Rootkit	2 768
下载者	5 327
其他	4 730

3 代码特征抽取

3.1 文本结构层特征

代码文件结构层特征指的是代码 PE 文件的静态结构信息,包括入口点是否正常、PE 头部的字节熵、标准节数目、非标准节的数目、节名是否异常和可执行节的数目等. 本文通过对恶意代码的 PE 文件结构进行分析,得到一个 19 维的代码特征向量,每维特征的值域采用数值或布尔值表示. 由于篇幅所限,本文只给出部分代码的特征含义与值域的对应关系,关系如表 2 所示:

表 2 部分代码特征及值域含义

编号	代码特征	值域
1	入口点是否正常	布尔值
2	PE 头部的字节熵	实数值
3	标准节数目	整数值
4	非标准节数目	整数值
5	节名是否异常	布尔值

3.2 字节层特征

恶意代码字节层特征表示代码在计算机中的二进制存储序列中出现的规律特征,在一定程度上表示恶意代码的指纹信息. 本文首先采用 Hexview 工具将每个代码样本转化为 16 进制的字节序列,然后采用 n-gram 滑动窗口的方法获取代码字节序列的字节特征. 以 2 作为 n-gram 窗口的滑动长度,为了避免抽取的字节层特征维数过多而造成系统内容溢出的问题,本文将字节层的特征维数上限设置为 10 万,即当算法在代码样本中识别出 100 000 B 序列组合特征后,算法便不再搜索其他新的字节序列,按照已有的字节序列进行特征统计.

3.3 指令层特征

代码指令层特征系指代码在计算机中执行机器指令过程中操作码和操作数的序列特征. 为了形象反映出代码、操作码和操作数之间的关系,本文对代码程序和操作码序列定义如下:

定义 1. 程序. 令代码程序 P 是一个指令序列的集合,$P=\{I_1, I_2, \cdots, I_i, \cdots, I_n\}$,其中 I_i 表示机器指令,由操作码和操作数组成,n 表示程序中包含的机器指令的个数.

定义 2. 操作码序列. 令操作码序列 $O=\{o_1, o_2, \cdots, o_i, \cdots o_m\}$,其中 o_i 表示一个操作码,m 为操作码个数. 操作码序列 O 可以被认为是代码程序 P 的一个子集,由 P 中包含的操作码组成.

本文通过反汇编工具 IDA Pro6. 1 对每个代码样本进行反向编译,采用 n-gram 算法对编译结果进行操作,获得编译结果中包含的 MOV, PUSH, SUB, CMP 等 13 个常用指令的序列特征. 本文以 2 作为 n-gram 窗口的滑动长度,从代码样本中抽取出 5 112 维特征,并得到了不同特征在样本指令中出现顺序的频次.

4 代码特征选择

由上文可知,本文分别从文本结构层、字节层和代码层抽取了 19 维、10 万维和 5 112 维代码特征. 由此可知,除了文本结构层代码特征较少之外,由字节层和代码层组成的代码特征矩阵或者由单一代码特征组成的特征矩阵都是一个高维并且稀疏的特征空间,并且特征与特征之间会由于相互关联而导致多重共线性问题. 鉴于此,为了提高分类效率,找到最有效表征恶意代码特点的数据特征,本文采用主成分分析方法分别对文本结构层、字节层以及代码层 3 种层次组合而成的特征矩阵进行降维,将降维后得到的特征矩阵作为

分类器的输入，进而评价特征组合对恶意代码的分类与识别结果.

主成分分析方法主要原理在于：通过一个由多维特征向量组成的投影矩阵将高维矩阵 X 转换成低维矩阵 Y，进而达到给高维矩阵 X 降维的目的. 其中，高维矩阵 X 表示由文本结构层、字节层以及代码层 3 种层次组合而成的特征矩阵. 该方法计算矩阵 X 的协方差矩阵进而获得协方差矩阵的特征值. 协方差矩阵的计算公式如式(2)所示，其中 $\bar{x}$ 表示每维特征向量的平均值，x_i 表示特征向量，N 表示特征向量维数，Cor 表示计算得到的协方差矩阵.

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{1}$$

$$Cor = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}}. \tag{2}$$

式(3)中的 λ 表示协方差矩阵的特征值向量，式(4)中的 u_i 表示根据特征值的大小选取的协方差矩阵的不同维向量，U 表示投影矩阵. 式(5)中的 Y 表示高维矩阵 X 经过投影矩阵转换而得的降维矩阵.

$$\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_i), \tag{3}$$

$$U = (u_1, u_2, \cdots, u_m), \tag{4}$$

$$Y = U^{\mathrm{T}}X. \tag{5}$$

由上所知，基于主成分分析方法计算由文本结构层、字节层以及代码层 3 种层次组合而成的特征矩阵的协方差矩阵，根据协方差矩阵的特征值和不同特征值所累加的贡献率，从协方差矩阵中选取不同的特征维组成投影矩阵，协方差特征值、贡献率及由不同特征值贡献率所累加的累计贡献率如表 3 所示. 由于篇幅所限，本文不列出所有特征值.

表 3 协方差矩阵的特征分析结果

编号	主成分	特征值	贡献率	累计贡献率
1	远程线程	8.36	0.06022	0.06022
2	写入服务	7.61	0.05481	0.11503
3	设置钩子	7.552	0.05440	0.16943
4	文件下载	7.47	0.05381	0.22323
5	文件隐藏	6.246	0.04499	0.26822
6	释放文件	5.783	0.04165	0.30988
7	注册表修改	5.78	0.04163	0.35151
⋮	⋮	⋮	⋮	⋮
41	异常表大小	1.113	0.00802	0.4327
42	计数器	1.112	0.00801	0.44071
⋮	⋮	⋮	⋮	⋮
109	时间戳	0.224	0.00161	0.60201

5 恶意代码分类与识别

5.1 恶意代码分类性能指标

基于主成分分析而得到的降维矩阵，本节采用多种分类器试图找出最有效的恶意代码分类方法. 为了衡量不同分类方法的性能，本文采用准确度、灵敏度和特异度 3 种评价指标评价分类器对恶意代码的分类与识别效果.

准确率反映的是分类器对整个样本集的判定能力，即将是恶意代码的测试样本判断为恶意代码，将不是恶意代码的测试样本判断为非恶意代码. 准确率计算公式如式(6)所示：

$$Accurcay = (TP + TN)/(TP + FN + FP + TN), \tag{6}$$

其中 TP，FP，FN，TN 分别表示被分类器识别为正的正样本、被分类器识别为正的负样本、被分类器识别为负的正样本、被分类器识别为负的负样本.

灵敏度反映的是分类器将正样本预测为正样本的能力. 即分类器将正常代码识别为正常代码

的能力.灵敏度计算公式如式(7)所示:

$$Sensitivity = TP/(TP + FN). \quad (7)$$

特异度反映的是分类器将负样本预测为负样本的能力.即分类器将恶意代码识别为恶意代码的能力.特异度计算公式如式(8)所示:

$$Specificity = TN/(TN + FP). \quad (8)$$

5.2 恶意代码分类与识别结果

为了对不同分类方法进行评价并找出最优的分类方法,本文基于 WEKA 数据挖掘平台,采用 NavieBayes, J48 (decision trees), JRip (rule learners), SVM (support vector machine), KNN (k-nearest neighbor)5 种分类方法对恶意代码进行识别与分类.此外,本文还分别从文本结构层、字节层、代码层以及 3 种代码特征组合 4 个角度,评价基于不同代码特征的分类器分类效果.由 2.2 节可知,本文共采用 16 608 个代码样本作为分类器的实验样本,其中每类代码的样本个数为 2768,采用 10 折交叉验证方法计算 3 种分类器评价指标平均值并作为最终的分类器效果的评价依据.基于不同特征的不同分类器的分类结果如表 4 所示:

表 4 不同特征的不同分类器的分类结果

分类算法	评价指标	结构特征	字节特征	指令特征	组合特征
NavieBayes	准确度	0.736	0.976	0.851	0.938
	灵敏度	0.754	0.981	0.876	0.943
	特异度	0.715	0.921	0.834	0.893
J48	准确度	0.675	0.923	0.934	0.946
	灵敏度	0.706	0.931	0.946	0.951
	特异度	0.656	0.899	0.864	0.907
JRip	准确度	0.671	0.896	0.936	0.936
	灵敏度	0.681	0.904	0.945	0.938
	特异度	0.652	0.876	0.896	0.908
SVM	准确度	0.667	0.974	0.981	0.925
	灵敏度	0.684	0.985	0.982	0.936
	特异度	0.613	0.923	0.914	0.886
KNN	准确度	0.737	0.807	0.911	0.941
	灵敏度	0.738	0.853	0.921	0.953
	特异度	0.696	0.792	0.876	0.918

根据表 4 不同特征的不同分类器的分类结果可知,在基于单一代码特征的分类实验中,基于结构特征的恶意代码识别准确率最低,而基于指令特征的恶意代码识别准确率最高,说明与其他代码特征相比,代码的指令特征更能表达恶意代码与正常代码之间的差异.在基于多种特征的分类实验中,基于组合特征的恶意代码识别获得了最优的分类结果,说明 3 种特征都分别从不同方面反映出恶意代码与正常代码之间的差异,基于多种特征的恶意代码识别方法相对可靠.

6 结 语

本文基于网上下载的代码实验样本,采用数据挖掘和机器学习的方法,从文本结构层、字节层、代码层 3 个角度提取了实验样本特征,通过主成分分析的方法对 3 种特征组合进行降维,基于得到的降维矩阵,采用贝叶斯、决策树、规则学习、支持向量机和最邻近算法 5 种分类方法对恶意代码进行分类,并分别与基于不同代码层次特征的

分类结果进行比较. 实验结果表明,基于数据挖掘和机器学习的恶意代码识别与分类方法在代码字节层和组合特征层都具有较好的准确度、灵敏度和特异度,其中又以基于组合特征的分类结果最优.

参 考 文 献

[1] Wang Z, Nascimento M, MacGregor M H. A multidisciplinary approach for online detection of X86 malicious executables [C] //Proc of Communication Networks and Services Research Conf (CNSR). Piscataway, NJ: IEEE, 2010: 160-167

[2] Patel S, Patel V, Jinwala D. Privacy preserving distributed k-means clustering in malicious model using zero knowledge proof [M] //Distributed Computing and Internet Technology. Berlin: Springer, 2013: 420-431

[3] Fu L, Zhang T, Zhang H, et al. A fuzzy classification method based on feature selection algorithm in malicious script code detection [C] //Proc of 2011 IEEE Int Conf on System Science, Engineering Design and Manufacturing Informatization (ICSEM). Piscataway, NJ: IEEE, 2011: 218-221

[4] Hsiao H W, Chen D N, Wu T J. Detecting hiding malicious website using network traffic mining approach [C] //Proc of the 2nd Int Conf on Education Technology and Computer (ICETC). Piscataway, NJ: IEEE, 2010: V5-276-V5-280

[5] Thuraisingham B M. Data mining for security applications [M] //Intelligence and Security Informatics. Berlin: Springer, 2006: 1-3

[6] 黄聪会, 陈靖, 龚水清, 等. 一种基于危险理论的恶意代码检测方法[J]. 中南大学学报: 自然科学版, 2014, 45(9): 3055-3060

[7] Lee T, Kim D, Jeong H, et al. Risk prediction of malicious code-infected websites by mining vulnerability features [J]. International Journal of Security and Its Applications, 2014, 8(1): 291-294

[8] Ramani R G, Kumar S S, Jacob S G. Rootkit (malicious code) prediction through data mining methods and techniques [C] //Proc of 2013 IEEE Int Conf on Computational Intelligence and Computing Research (ICCIC). Piscataway, NJ: IEEE, 2013: 1-5

[9] Li X, Dong X, Wang Y. Malicious code forensics based on data mining [C] //Proc of the 10th Int Conf on Fuzzy Systems and Knowledge Discovery (FSKD). Piscataway, NJ: IEEE, 2013: 978-983

[10] Li Y, Ma R, Jiao R. A hybrid malicious code detection method based on deep learning [J]. Methods, 2015, 9(5): 205-216

廖国辉

硕士研究生,主要研究方向为恶意代码检测、网络信息处理与信息安全.

sculiaoguohui@yeah.net

刘嘉勇

教授,博士生导师,主要研究方向为信息安全理论与应用、网络信息处理与信息安全.

ljy@scu.edu.cn

大数据相关词条(续)

◆数据结构(data structure):各种数据之间的逻辑关系,用来支持特定的数据处理功能,比如树、列表和链接表.

◆数据可视化(data visualization):关于数据的视觉表现形式的研究,是一种多维度数据通过图形的方式来做的展现,这种数据的视觉表现形式被定义为一种以某种概要形式抽取出来的信息,包括相应信息单位的各种属性和变量.

◆数据挖掘(data mining):从存放在数据库、数据仓库或其他信息库中的大量的数据中获取有效的、新颖的、潜在有用的、最终可理解的模式的过程.

◆数据仓库:是决策支持系统(DSS)和联机分析应用数据源的结构化数据环境.数据仓库研究和解决从数据库中获取信息的问题.数据仓库的特征在于面向主题、集成性、稳定性和时变性.

◆数据清洗(data cleaning):过滤那些不符合要求的数据,将过滤的结果交给业务主管部门,确认是否过滤掉还是由业务单位修正之后再进行抽取.