一种基于网络行为分析的 HTTP 木马检测模型

易军凯 刘健民 万 静*
(北京化工大学 信息科学与技术学院, 北京 100029)

摘 要: 基于 HTTP 协议进行网络通信的木马能够躲避部分网络安全监控系统的检测,是互联网安全的一个重大威胁。通过对该类木马样本和普通程序样本网络行为的对比分析,得到该类木马的 6 个网络行为特征,综合利用层级聚类、Davies-Bouldin 指数和 *k*-means 聚类方法提出了一种木马检测模型,实现了 HTTP 木马检测。结果表明,该 HTTP 木马检测模型准确率较高,误报率较低。

关键词: 木马检测;网络行为;HTTP
中图分类号: TP393.08

引言

随着互联网的普及,网络犯罪的数量也在不断的增多,恶意软件成为互联网安全的重大威胁。木马是一种流传广泛的恶意软件,与病毒和蠕虫等恶意软件相比,木马的主要危害在于它隐藏在受害者的主机内,监视受害者主机的活动,窃取受害者的隐私数据,造成的危害很大。

已有的木马检测方法主要分为两类:基于静态的分析法和基于动态的分析法。基于静态的分析将重点放在程序静态特征的提取上,依赖于强大的病毒库,但难以检测出已知木马的变种及新的木马。基于动态的分析主要是分析木马的行为,基于行为的分析与单纯的静态分析相比,能够提高对木马变种的识别率。虽然木马可以通过加密、打包等方法改变它的静态特征[1],但它们在运行时通常表现出相似的行为特征。基于行为分析方面,主要有基于系统的行为分析和基于网络的行为分析,但系统行为获取难度较大。在网络行为研究方面,Li 等[2]通过提取程序的网络出入流量、通信间隔、主从连接等特征,使用 *k*-means 聚类方法检测木马;Nari 等[3]提取恶意软件使用的应用层协议(如 DNS、HTTP、SMTP 等)来构建网络行为图,使用 J48 决策树来对恶意软件进行分类。有很多木马使用 HTTP 协议进行网络通信, Rossow 等[4]在对 10 万多个不同恶意软件样本的分析中发现,有 58.6% 的样本使用了 HTTP 协议。在基于网络行为分析检测 HTTP 木马方面,Ding 等[5]基于 C4.5 算法提取数据流的 21 个统计特征检测 HTTP 隧道,孙海涛等[6]采用 C4.5 算法通过操作行为分析检测 HTTP 隧道木马,Perdisci 等[7]使用层次聚类方法通过提取网络行为特征对基于 HTTP 协议的恶意软件进行网络行为签名进而进行检测。然而以上这些方法存在检测对象较为单一,或由于检测对象范围过大而缺乏针对性等问题。

本文采用基于网络行为分析的方法研究 HTTP 木马,通过分析基于 HTTP 协议木马的网络行为,采用网络行为特征和 *k*-means 聚类方法建立木马检测模型,利用该模型对木马的网络行为进行标注,实现木马检测。

1 HTTP 木马网络行为分析

基于 HTTP 协议的木马可使用 GET 方法从控制端获取信息,或者以参数的形式将木马端的简单信息告知控制端。同时,使用 POST 方法可将木马端的执行结果及所需数据发送到控制端,以这样的形式实现对木马端的控制。

本文从以下几个方面分析得到木马网络行为特征。

1)不重复页面数量。图 1 是基于 HTTP 协议的木马和普通程序运行 60 s 过程中木马与普通程序 GET 方法请求的不重复页面数量对比图。对比发现,HTTP 木马请求的一般为相同的几个页面,而普

收稿日期: 2013-06-28
基金项目:中央高校基本科研业务费(zz1311)
第一作者:男,1972 年生,教授
* 通讯联系人
E-mail: wanj@mail.buct.edu.cn

通程序请求的不重复页面较多。

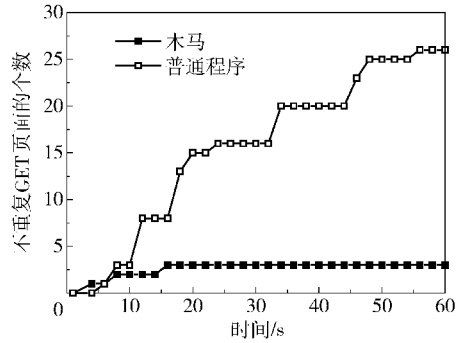

图 1　木马与普通程序 GET 请求的不重复页面数量对比图

Fig. 1　Comparison of the number of Trojan and normal program's different GET pages

2)出入流量。图 2 为 180 s 内 HTTP 木马的出入流量统计图,正数表示出流量,负数表示入流量。对比发现木马的出流量远大于入流量,这与基于 HTTP 协议的普通程序恰恰相反。图 2 也体现了木马网络流中的心跳包,心跳包是网络数据流中一种自定义协议、固定信息、循环发送的数据包,在各种网络应用中作为在线状态检测、状态汇报、网络同步或其他定时机制的应用而普遍存在[8]。木马使用心跳包进行在线状态检测。

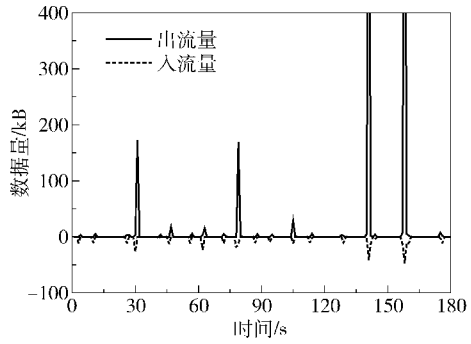

图 2　木马出入流量统计图

Fig. 2　Statistics of Trojan out-in traffic

3)POST 请求平均数据量。图 3 将一定时间内基于 HTTP 协议的木马样本的 POST 请求平均数据量与普通程序比较发现,木马每次发送的 POST 数据量较大,而普通程序的 POST 请求每次只发送少量数据。

2　检测模型

本文中木马样本特征的提取是基于两个 IP 之间在一个时间 T 内产生的 HTTP 流量,对木马样本

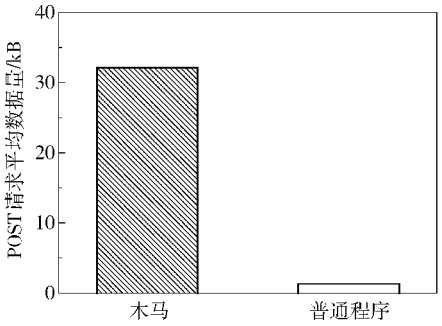

图 3　POST 请求平均字节数对比图

Fig. 3　Comparison of POST's average bytes

通信行为的提取也就是对 1 个 IP 对的通信行为的检测。

定义 1　$\boldsymbol{I}_i = \{I_{i,j} \mid 1 \leqslant j \leqslant n\}$。一个 IP 对的通信实例 $\boldsymbol{I}_i$ 由一组特征向量来描述,$\boldsymbol{I}_i$ 表示第 i 个通信实例,n 为每个通信实例的特征数,$I_{i,j}$ 表示第 i 个通信实例的第 j 个特征。

定义 2　$I = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \cdots, \boldsymbol{I}_m\}$。集合 I 表示所有 IP 对的通信实例的集合,其中 m 表示通信实例的总数。

定义 3　$C = \{C_1, C_2, \cdots, C_k\}$。集合 C 表示检测模型中簇的集合,其中 k 为模型中簇的个数。

2.1　模型概述

本文提出的木马检测模型主要步骤如下。

(1)特征提取。从每个样本产生的网络流量中提取出 HTTP 协议的流量,处理所有的 HTTP 流量,对于每一个通信实例提取所需特征(如数据流出流入比、POST 平均数据量等)构成 $\boldsymbol{I}_i$,同时标注出每个通信实例是木马实例还是普通实例,构成通信实例集合 I。

(2)部分样本聚类。从通信实例集合 I 中随机选取少量通信实例构成集合 $I' = \{\boldsymbol{I}'_1, \boldsymbol{I}'_2, \cdots, \boldsymbol{I}'_r\}$,其中 $I' \in I$ 且 $r < m$,使用层次聚类方法对 I' 进行聚类得到初始簇集合 $C' = \{C'_1, C'_2, \cdots, C'_k\}$,在聚类过程中使用 Davies-Bouldin(DB)指数确定簇数 k[9]。

(3)全部样本聚类。簇数 k 以及初始簇集合 C' 作为输入,使用 k-means 算法对全部通信实例的集合 I 进行聚类,得到最终的簇集合 C。

(4)木马检测。根据最初标注的每个通信实例的标签对每个簇 C_i 进行判断,判断该簇是木马簇、普通簇还是未知簇。对于待检测通信实例,计算它与每个簇中心点的距离,使用距离它最近的簇的标

签进行标注。

2.2 特征提取

通过对基于 HTTP 协议木马的网络行为进行分析,总结出以下几个特征用于对通信实例进行聚类。

1) *DifGetPages* 1 个 IP 对在通信过程中出现的 GET 请求的不重复页面数量。

2) *DifGetParas* 1 个 IP 对在通信过程中出现的不重复参数的数量。

3) *DifPostPages* 1 个 IP 对在通信过程中出现的 POST 请求的不重复页面数量。

4) *HeartBeat* 1 个 IP 对在通信过程中是否存在心跳包,如果存在则记为 1,不存在则记为 0。

5) *OIRatio* 1 个 IP 对在通信过程中的输出字节数与输入字节数的比。

6) *PostAvgBytes* 1 个 IP 对在通信过程中所有 POST 请求的平均字节数。

由于不同特征的取值范围不同,因此需要对特征进行归一化,将所有特征的取值映射至 0 到 1 的区间内,采用公式(1)对特征进行归一化

$$v_i^* = \frac{v_i - v_{\min}}{v_{\max} - v_{\min}} \tag{1}$$

其中 v_i^* 表示第 i 个通信实例的特征 v 归一化后的值, v_i 表示特征 v 原始的值,$v_{\min}$ 表示所有通信实例中特征 v 的最小值,$v_{\max}$ 表示所有通信实例中特征 v 的最大值。经过归一化后得到带有特征的通信实例的集合 I。

2.3 部分样本聚类

使用 k-means 算法对通信实例集合 I 聚类,但 k-means 算法中簇的个数 k 以及初始中心点难以确定,在本文的模型中,先从 I 中随机选取少量通信实例构成 I',使用层次聚类算法对 I' 聚类,并使用 DB 指数来确定 k。

计算距离使用欧氏距离公式(式(2))

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{l=1}^{n} (x_{i,l} - x_{j,l})^2} \tag{2}$$

其中 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 表示两个不同的通信实例,$x_{i,l}$ 表示 $\boldsymbol{x}_i$ 的第 l 个特征的值,n 为每个实例的特征数。选用层次聚类中凝聚的方法,即使用自底向上的策略,从令每个通信实例形成自己的簇开始,迭代把整个簇合并成越来越大的簇,直到所有的通信实例都在一个簇中,或者满足某个终止条件。层次聚类算法的时间复杂度较高,适合小型数据集的分类。

DB 指数是用来评估聚类算法的一个度量,它使用固有的数量和特征来验证聚类的效果,DB 指数可以用公式(3)表示

$$D(k) = \frac{1}{k} \sum_{i=1}^{k} \max_{j=1\cdots k, j\neq i} \left(\frac{S_i + S_j}{M_{i,j}} \right) \tag{3}$$

其中 S_i 和 S_j 是表示簇分散性的度量,计算公式如式(4)所示

$$S_i = \sqrt{\frac{1}{T_i} \sum_{j=1}^{T_i} | \boldsymbol{X}_j - \boldsymbol{A}_i |^2} \tag{4}$$

其中 $\boldsymbol{X}_j$ 表示簇 C_i 的一个 n 维特征向量,$\boldsymbol{A}_i$ 是 C_i 的中心点,T_i 是簇 C_i 的大小。公式(3)中的 $M_{i,j}$ 表示簇 C_i 和簇 C_j 的距离,计算公式如式(5)所示

$$M_{i,j} = \| \boldsymbol{A}_i - \boldsymbol{A}_j \|_2 = \sqrt{\sum_{l=1}^{n} | a_{i,l} - a_{j,l} |^2} \tag{5}$$

其中 $a_{i,l}$ 表示第 i 个簇的中心点 $\boldsymbol{A}_i$ 的第 l 个特征,每个中心点有 n 个特征。DB 指数越小表示 k 选择的越合理,根据这一评判准则来确定 k。在这一步中通过单链接层次聚类和 DB 指数确定了初始的簇集合 $C' = \{C'_1, C'_2, \cdots, C'_k\}$ 以及簇的个数 k。

2.4 全部样本聚类

对全部样本的聚类采用 k-means 算法,它能对大型数据集进行高效分类。时间复杂度为 $O(tknm)$,其中 t 为迭代次数,k 为聚类数,n 为特征数,m 为待分类的对象数,通常 $k, n, t \ll m$,在对大型数据集聚类时,k-means 算法比层次聚类算法快得多[10]。k 值及初始的簇集合 $C' = \{C'_1, C'_2, \cdots, C'_k\}$ 由 2.3 节得到。这里使用的 k-means 算法,根据 C' 计算初始的中心点集合 $A = \{\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_k\}$,通过计算 $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 把 $\boldsymbol{I}_i$ 分配到离它最近的 C_i,并重新计算 A,重复这些计算直到簇不再发生变化或达到最大迭代次数 t。

2.5 木马检测

对聚类算法生成的簇集合 C 进行标注,标注出第 i 个簇 C_i 为 trojan、normal 或者 unknown,分别表示该簇为木马簇、普通程序簇或者未知程序簇。对簇 C_i 统计该簇中木马和普通程序的个数,分别记为 $n_{i,\text{trojan}}$ 和 $n_{i,\text{normal}}$,则标注时使用的决策函数为公式(6)。

$$C_i = \begin{cases} \text{trojan}, n_{i,\text{trojan}} > n_{i,\text{normal}} \\ \text{normal}, n_{i,\text{trojan}} < n_{i,\text{normal}} \\ \text{unknown}, n_{i,\text{trojan}} = n_{i,\text{normal}} \end{cases} \tag{6}$$

对于一个待检测的通信实例 $\boldsymbol{I}^*$,计算它与所有 $\boldsymbol{A}_i$ 的距离,将 $\boldsymbol{I}^*$ 归类到距离它最近的 C_i 中,即 $\boldsymbol{I}^* \in$

C_i，满足 $d(\boldsymbol{I}^*, \boldsymbol{A}_i) = \min\{d(\boldsymbol{I}^*, \boldsymbol{A}_j), j = 1, 2, \cdots k\}$，最后使用 C_i 的标签对 $\boldsymbol{I}^*$ 进行标记。

3　木马检测实例

3.1　数据集

本文实验中的木马样本来自 Clean MX 数据库（http：// support. clean-mx. de/clean-mx/viruses. php），该数据库实时更新病毒样本，定期从中下载病毒样本，并在 VirusTotal（https：// www. virustotal. com/en/）上进行扫描，对于一个病毒样本，当扫描结果中有 7 个以上的杀毒软件将其判定为 Trojan 时，即判断为木马样本。将获取到的木马样本在虚拟机环境下运行 15 min，使用 Wireshark（http：// www. wireshark. org/）抓取木马样本的网络流量，生成 pcap 格式的文件，过滤出基于 HTTP 协议的木马样本集。抓取基于 HTTP 协议的普通软件的网络流量，生成 pcap 格式的文件作为普通程序样本集。实验中，将木马样本集和普通程序样本集分别分为两部分，一部分作为训练样本集 S_{train}，另一部分作为测试样本集 S_{test}，再将 S_{train} 和 S_{test} 分别分为 5 部分，在模型中进行实验。

3.2　实验结果及分析

实验中使用 S_{train} 进行训练，生成带有标注的簇集合 C，使用 C 对 S_{test} 进行检测，其中在 2.3 节对部分样本进行聚类时随机选取总样本的 10% 来确定 k 及初始簇 C'。实验结果如表 1 所示。这里使用准确率和误报率两个评估度量评估检测效果。准确率表示正确检测的木马通信实例数量占测试集中木马通信实例总数的比例，误报率表示被模型错误判断为木马的通信实例在所有被判断为木马的通信实例中的比例。

表 1　实验结果

Table 1　Experimental results

训练集	训练样本数	簇数				测试集	测试样本数	准确率/%	误报率/%
		trojan	normal	unknown	总数				
$S_{\text{train},1}$	1000	19	21	5	45	$S_{\text{test},1}$	500	82.37	1.45
$S_{\text{train},2}$	1000	18	18	3	39	$S_{\text{test},2}$	500	80.65	2.16
$S_{\text{train},3}$	1000	15	20	3	38	$S_{\text{test},3}$	500	83.23	1.86
$S_{\text{train},4}$	1000	16	23	3	42	$S_{\text{test},4}$	500	78.94	2.95
$S_{\text{train},5}$	1000	14	15	2	31	$S_{\text{test},5}$	500	81.79	1.96

如表 1 所示，实验结果表明，本文提出的木马检测模型对于 3.1 节构造的数据集的检测准确率都在 80% 左右，并且误报率较低。但一些网络行为隐藏做的比较好的木马还是能逃过检测模型的检测。实验中每个训练集的大小 m 均为 1000，部分样本数 r 均为 100，这限制了由该方法得到的 k 不会超过 100，因此当总样本数量 m 增加时，r 也需随之增加。实验中每个训练集的木马样本和普通程序样本的数量是相等的，因此 trojan 簇和 normal 簇的分布比较平均。通过对部分实验结果的人工分析，还发现该检测模型能够检测出训练集中已有的木马样本的变种。

如表 2 所示，将本文提出的方法与文献[6]的方法进行比较发现，本文提出的方法准确率比文献[6]略高，误报率更低。文献[6]使用了会话包总数、会话总数据量、会话时长、会话上传数据量、会话上传数据量和下载数据量之比、会话平均上传速度等 6 种特征，与本文使用的特征比较可以看出，文献[6]使用的特征注重 HTTP 会话的流量信息，而本文使用的特征更全面，更容易区分普通程序和木马程序，因此具有更好的检测性能。

表 2　不同方法实验结果对比

Table 2　Comparison of different methods

训练集	测试集	准确率/%		误报率/%	
		本文方法	文献[6]方法	本文方法	文献[6]方法
$S_{\text{train},1}$	$S_{\text{test},1}$	82.37	78.24	1.45	2.28
$S_{\text{train},2}$	$S_{\text{test},2}$	80.65	76.14	2.16	4.68
$S_{\text{train},3}$	$S_{\text{test},3}$	83.23	80.25	1.86	3.05
$S_{\text{train},4}$	$S_{\text{test},4}$	78.94	74.57	2.95	6.56
$S_{\text{train},5}$	$S_{\text{test},5}$	81.79	79.86	1.96	4.13

4　结束语

本文在对基于 HTTP 协议的木马行为分析的基

础上,提取了用于对网络行为建模的 6 个特征,综合利用层级聚类、Davies – Bouldin 指数和 k-means 聚类,对基于 HTTP 协议的木马和普通程序的网络行为建模。在实际环境中使用该模型进行实验,实验结果验证了本文选取的特征能够较好的表现 HTTP 木马的网络行为,与文献[6]的方法相比准确率更高,误报率更低,具有更好的检测效果。

参考文献:

[1] Guo F, Ferrie P, Chiueh T C. A study of the packer problem and its solutions[C]//Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection, Cambridge, USA, 2008: 98–115.

[2] Li S C, Yun X C, Zhang Y Z, et al. A general framework of trojan communication detection based on network traces[C]//7th International Conference on Networking, Architecture, and Storage (NAS), Xiamen, Fujian, 2012: 49–58.

[3] Nari S, Ghorbani A A. Automated malware classification based on network behavior[C]//International Conference on Computing, Networking and Communications (ICNC), San Diego, USA, 2013: 642–647.

[4] Rossow C, Dietrich C J, Bos H, et al. Sandnet: Network traffic analysis of malicious software[C]//Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, Salzburg, Austria, 2011: 78–88.

[5] Ding Y J, Cai W D. A method for HTTP-tunnel detection based on statistical features of traffic[C]//3rd International Conference on Communication Software and Networks (ICCSN), Xi'an, 2011: 247–250.

[6] 孙海涛, 刘胜利, 陈嘉勇, 等. 基于操作行为的隧道木马检测方法[J]. 计算机工程, 2011, 37(20): 123–126.
Sun H T, Liu S L, Chen J Y, et al. Tunnel trojan detection method based on operation behavior[J]. Computer Engineering, 2011, 37(20): 123–126. (in Chinese)

[7] Perdisci R, Ariu D, Giacinto G. Scalable fine-grained behavioral clustering of HTTP-based malware[J]. Computer Networks, 2013, 57(2): 487–500.

[8] 易军凯, 陈利, 孙建伟. 网络心跳包序列的数据流分簇检测方法[J]. 计算机工程, 2011, 37(24): 61–63.
Yi J K, Chen L, Sun J W. Data flow clustering detection approach of network heartbeat packet sequence[J]. Computer Engineering, 2011, 37(24): 61–63. (in Chinese)

[9] Davies D L, Bouldin D W. A cluster separation measure[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1979 (2): 224–227.

[10] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1): 48–61.
Sun J G, Liu J, Zhao L Y. Clustering algorithms research[J]. Journal of Software, 2008, 19(1): 48–61. (in Chinese)

A model of an HTTP-based Trojan detection based on network behavior analysis

YI JunKai LIU JianMin WAN Jing

(College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: HTTP-based Trojans which can avoid detection by a network security monitoring system are a major threat to internet security. In this paper we obtain six characteristics that can represent the network behavior of such Trojans through analyzing and comparing the network behavior of HTTP-based Trojan and normal program samples. We propose a model for Trojan detection that utilizes a single-linkage hierarchical clustering algorithm, the Davies – Bouldin index and a k-means clustering algorithm. The results show that the model of Trojan detection is suitable for detecting Trojans with high accuracy and low false positive ratios.

Key words: Trojan detection; network behavior; HTTP