doi :10.3969/j.issn.1671-1122.2018.08.001

基于机器学习的恶意文档识别工具设计与实现

文伟平[1]，吴勃志[1]，焦英楠[2]，何永强[1]

（1. 北京大学软件与微电子学院，北京 102600；2. 国家计算机网络应急技术处理协调中心，北京 100029）

摘　要：随着网络化、信息化的程度进一步提高，高级持续性威胁（Advanced Persistent Threat，APT）事件不断增多，给国家、企业的安全发展带来了严重威胁和巨大经济损失。APT 攻击通过定向情报收集、单点攻击突破、控制通道构建、内部横向渗透和数据收集上传等一系列步骤对特定目标进行长期持续的网络攻击。而在单点攻击突破阶段，最常用的网络攻击技术手段是采用植入远程木马的恶意文档，所以有效检测和识别恶意文档十分必要。文章在对现状进行充分调研后，提出一种基于机器学习的恶意文档检测方法。通过结合虚拟沙箱对未知文档进行动态行为分析，设计并实现了一种恶意文档识别工具。实验证明，该工具基于机器学习方式，可以高效处理和识别大规模的恶意文档文件。

关键词：恶意文档；机器学习；特征向量；虚拟沙箱

中图分类号：TP309　**文献标识码：**A　**文章编号：**1671-1122（2018）08-0001-07

中文引用格式：文伟平，吴勃志，焦英楠，等 . 基于机器学习的恶意文档识别工具设计与实现 [J]. 信息网络安全，2018，18（8）：1-7.

英文引用格式：WEN Weiping，WU Bozhi，JIAO Yingnan，et al. Design and Implementation on Malicious Documents Detection Tool Based on Machine Learning[J]. Netinfo Security, 2018, 18 (8): 1-7.

Design and Implementation on Malicious Documents Detection Tool Based on Machine Learning

WEN Weiping[1], WU Bozhi[1], JIAO Yingnan[2], HE Yongqiang[1]

(1. *School of Electronics Engineering and Computer Science*, *Peking University*, *Beijing* 102600, *China*; 2. *National Computer Network Emergency Response Technical Team / Coordination Center*, *Beijing* 100029, *China*)

Abstract: With the further improvement of the degree of network and information, the advanced persistent threat (APT) events are increasing, which brings serious threat to the security development of the state and huge economic losses to enterprises. APT attack carries out a long-term continuous network attack on specific target by using a series of steps which include targeted intelligence collection, single point attack breakthrough, control channel construction, internal horizontal penetration and data collection and upload and so on. In the single point attack breakthrough stage, the most commonly used technology of network attack is to use malicious documents implanted

收稿日期：2018-4-9

基金项目：国家自然科学联合基金 [U1736218]

作者简介：文伟平（1976—），男，湖南，教授，博士，主要研究方向为网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等；吴勃志（1990—），男，广东，硕士研究生，主要研究方向为漏洞分析和漏洞挖掘；焦英楠（1983—），女，辽宁，工程师，硕士，主要研究方向为软件工程、信息安全等；何永强（1984—），男，四川，硕士研究生，主要研究方向为软件工程。

通信作者：文伟平 weipingwen@pku.edu.cn

remote Trojans, so it is necessary to detect and identify malicious documents. After fully investigating the status quo, this paper proposes a malicious document detection method based on machine learning. By analyzing dynamic behaviors of unknown documents combining with virtual sandbox, a malicious document recognition tool is designed and implemented. Experiments show that the tool can efficiently process and identify large-scale malicious documents based on machine learning.

Key words: malicious document; machine learning; feature vector; virtual sandbox

0 引言

近几年来，高级持续性威胁（Advanced Persistent Threat，APT）攻击已经成为社会关注的热点。Symantec 2016 年发布的网络威胁报告 [1] 指出，随着攻击成本的降低，APT 攻击的主要目标已经逐渐由大型企业转变为中小型企业，这意味着有越来越多的公司面临此类攻击威胁。随着网络攻防态势的发展，远程类漏洞越来越少，恶意文档开始被频繁用于网络攻击。由于恶意文档比一般可执行程序更容易受到信任，因此在 APT 攻击中发挥了重要作用。

针对 APT 攻击，国内外学者进行了很多相关研究。其中发现，恶意文档是 APT 攻击单点突破阶段最常用的技术手段，对整体攻击的成功率起到重大作用。如果在邮件钓鱼等环节提前捕获并识别出恶意文档，那将能有效预防和对抗 APT 攻击。然而，识别恶意文档的方法大多不尽人意，要么存在准确率低、误报率高的问题，要么无法对加密文档进行检测。为此，本文提出一种基于机器学习的恶意文档检测方法，该方法将恶意文档攻击链的关键点作为特征，将检测样本置于虚拟沙箱子系统中动态运行，结合机器学习方法训练恶意文档分类模型。实验结果显示，本文提出的恶意文档检测方法具有很高的准确率，其中 office 文件准确率为 95.75%，pdf 文件准确率为 96.34%，swf 文件准确率为 88.41%。

1 相关工作

1.1 恶意文档当前攻击链分析

为了研究恶意文档检测技术，需要对恶意文档的攻击链进行详细了解和分析。利用恶意文档进行攻击时，主要分为 4 个关键阶段（如图 1 所示）：任意代码执行、绕过系统安全缓解策略、反安全产品检测和执行功能。下面重点介绍 4 个阶段中的关键技术。

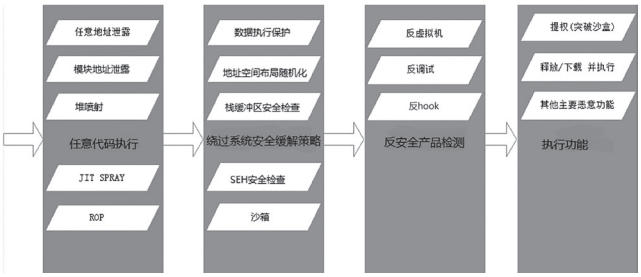

图 1 恶意文档攻击链示例图

1）突破 DEP/ASLR 保护技术。DEP（数据保护执行）和 ASLR（地址空间布局随机化）技术就是利用操作系统中异常处理机制和代码模块入口点随机分布这两项技术来阻止恶意代码的执行 [2]。

2）突破沙箱保护技术。沙箱技术简单来说就是让程序运行在其所需最小权限的受限环境中。例如，浏览网页的进程只负责图像渲染就可以了，没有必要拥有执行系统命令的权限。在沙箱保护模式下运行的程序即使攻击成功，获得的权限也是非常低的，无法正常读写文件。

3）任意内存地址读写漏洞利用。任意内存地址读写漏洞触发成功后即获得全内存区域读写权限，获取该权限后相当于拥有了一个程序运行时的内存补丁生成器，可以向程序打任意补丁，获取程序完全控制权。

4）模块地址泄露技术。ASLR 的最大缺陷就是虽然代码模块入口地址是随机分配的，但是代码模块内相对偏移是固定的 [3]。如果能找到漏洞将代码模块内任意地址泄露出来，那么整个代码模块的入口地址就可以根据固定偏移计算得来，从而使 ASLR 防护彻

底失效。

5）反调试技术。Payload（有效载荷）执行后，通过相关技术判断当前程序是否正在被调试，若是，则不再执行功能代码或执行干扰调试代码以防被进一步分析。

6）反 hook 挂钩技术。安全软件为了监视程序的可疑行为，会 hook 一些敏感 API 函数 [4]，而在漏洞利用过程中，通常会调用这些 API 函数完成一些特殊功能。因此为了绕过安全软件的检测，需要在 Payload 中加入反 hook 技术，从而突破安全软件的部分限制。

7）反虚拟执行检测。Payload 执行后，通过相关技术判断当前程序是否处在虚拟环境中运行，若是，则直接退出执行，以避免被分析。

1.2 国内外恶意文档检测技术

有许多相关技术可用于识别未知恶意代码及其变种，主要有以下 3 种。

1.2.1 静态启发式扫描技术

该技术指在代码没有动态运行的状态下，通过分析可疑文档中的特征指令序列（如 API 序列）来识别恶意代码。文献 [5] 对文档中的字段进行频谱分析，通过提取频谱特征属性来进行恶意文档的识别；文献 [6] 基于文献 [5] 提出应用 JRip 算法提取文件频谱图的特征来检测识别恶意 pdf 文档；文献 [7] 通过提取 pdf 文档中的 JavaScript 代码并进行特征分析来识别恶意 pdf 文档。这些方法在非加密变形的恶意文档方面取得了较好的识别效果，对于加密文档的识别效果并不显著。

1.2.2 基于代码的动态行为分析技术

该技术要点在于监控代码在运行过程中的运行行为，这些行为包括文件操作、网络连接行为等。HUANG[8] 等人提出通过观察可疑文档动态运行时系统注册表的键值、网络连接行为以及文件 IO 操作的异常变化来识别是否有恶意行为。该方法的优点是简单、高效，可以一定程度检测出未知恶意文档，而不用担心恶意代码是否被加密；缺点是误报率和漏报率较高。

1.2.3 基于机器学习的恶意文档检测技术

文献 [9] 提出一种改进的静态检测方案，通过学习一定量的样本来建立模型 O，计算由未知样本提取出的向量和模型 O 中原点之间的欧式距离，设置一定的权值 R，通过比较欧氏距离和权值 R 的大小来确定 pdf 文档是否为恶意。文献 [10] 提出利用半监督学习的方法，使用大量标记好的 pdf 文档进行分类模型的训练。以上两种方法虽然都有效提高了准确率，但也仅适用于检测识别基于 JavaScript 攻击的恶意 pdf 文档，同样具有局限性。

2 恶意文档检测方案

2.1 模型选择与方案设计

借助机器学习的方法，使用大规模训练样本集搭建智能模型，可以高效准确地解决目前大规模攻击导致的安全问题。然而不同的算法所擅长处理的领域有很大的区别，选择合适的机器学习算法非常重要。以下对机器学习算法的优缺点进行比较：

1）KNN 算法。KNN 算法理论简单、容易实现，但当样本容量大，数据集计算量比较大，样本不平衡时，预测偏差比较大。

2）支持向量机（SVM）。优点是擅长解决非线性及小规模样本条件下机器学习问题，无局部极小值问题，可以很好地处理高维数据集，泛化能力比较强。缺点是对于核函数尤其是径向基函数的高维映射解释力不强，且对缺失数据敏感。

3）AdaBoost 算法。优点是可以采用不同的分类算法作为弱分类器，并将多个弱分类器级联起来，构成一个强分类器，使得分类具有很高的精度。缺点是 AdaBoost 迭代次数也就是弱分类器数目不好设定，且数据不平衡导致分类精度下降。

4）决策树。优点是易于理解和解释，能可视化分析，容易提取出规则且可以同时处理标称型和数值型数据。测试数据集时运行速度比较快。决策树可以很好地扩展到大型数据库中，且它的大小独立于数据

库的大小。缺点是容易出现过拟合问题[11]（为了得到一致假设而使假设变得过度严格称为过拟合）。

5）朴素贝叶斯算法。优点是对大规模的训练和查询具有较高的速度，支持增量式运算。缺点是样本属性需要相互独立，当存在关联时效果不好。

6）人工神经网络。优点是分类准确率高、学习能力强、有联想能力、能逼近任意非线性关系、对噪声数据鲁棒性和容错性较强。缺点是神经网络参数较多，如权值和阈值，且黑盒过程不能观察中间结果。

由于恶意文档更新及变种速度极快，为了准确且快速地进行恶意文档识别，对算法的主要需求为：分类规则易于理解，可以结合虚拟沙箱模块进行扩展和改动，识别准确率高且生成的模型可以可视化显示，可以自动识别具有类似行为的新类别的恶意文档。

在分析、研究恶意文档的攻击链以及机器学习算法的基础上，本文提出一个检测方法，以恶意文档攻击链的动态行为作为特征向量获取的依据，生成决策树，从而实现对各种文件类型的恶意文档检测。本文设计采用 QEMU 虚拟沙箱，将训练文档投入其中模拟运行，通过 minifilter 过滤驱动框架[12]来定点监控训练文档的动态特征行为，采用 C4.5 决策树分类算法作为机器学习模型，以动态特征行为作为特征向量，训练该机器学习模型。利用这种方法得到一个分类器，后续对样本进行检测时，只需将样本放进虚拟沙箱，利用该分类器分析其动态特征行为即可判断该样本是否属于恶意文档。

2.2 特征向量的提取及分类

特征向量的提取是影响决策树识别结果的重要因素。本文方法基于当前恶意文档攻击链行为特征进行特征向量的提取。特征向量提取主要分为两个方面：静态特征向量提取和动态行为特征向量提取。

1）静态特征向量提取。针对恶意文档中经常出现的特殊字符串、特殊函数以及特殊关键字进行特征向量的提取。表 1 给出了从一些 pdf 格式的恶意文档中提取出的 JavaScript 代码的关键静态特征向量[13]。

表 1 pdf 恶意文档 JavaScript 代码的关键静态特征向量提取

特殊字符	特殊函数	特殊关键字
（1）含有参数的字符串个数 （2）长度大于 40 的字符串个数 （3）字符串中包含“Iframe”的数量 （4）可疑字符串的个数 （5）十六进制字符出现的个数 ……	（1）eval()的数量 （2）escape()和 unescape()的数量 （3）DOM()的数量 （4）CreateObject()的数量 （5）fromCharCode()和 parseInt()的数量 （6）setTimeOut()的数量 （7）ActiveXObject()的数量 ……	（1）for 出现的次数 （2）while 出现的次数 （3）classId 的数量 ……

2）动态行为特征向量提取。动态行为特征向量分别从任意代码执行、绕过系统安全缓解策略、反安全产品检测、执行功能这 4 个阶段进行提取，分析这 4 个阶段的触发过程和行为特点，主要包含网络、文件、内存、线程、进程、系统和注册表 7 种类别。

（1）任意代码执行行为特征向量

①线程行为。栈溢出时可能造成的内存破坏（SEH 链完整性破坏）。

②内存行为。通过堆喷射部署内存占领可预测内存发生的默认堆空间、堆块异常分配。

（2）绕过系统安全缓解策略行为特征向量

①系统行为。敏感 API 的异常环境调用。

②内存行为。绕过 DEP 和 ASLR 的 ROP 链操作，如 Kernel 32 模块的导出表或 MZ 头的异常访问。

（3）反安全产品检测行为特征向量

如反 hook、反调试、反虚拟机等。

（4）执行功能行为特征向量

①网络行为。建立网络链接、下载数据流量。

②文件行为。创建可执行文件。

③进程行为。创建进程。

④系统方面。调用获取系统环境信息的敏感 API。

⑤注册表方面。创建服务或设置自启动。

2.3 C4.5 决策树的生成与识别

C4.5 决策树的生成由以下步骤构成：

1）收集训练样本集，即恶意文档样本及非恶意文档样本。

2）将训练样本集中的文档样本由调度模块输入虚拟沙箱。

3）在虚拟沙箱中运行文档样本，捕获文档样本的运行行为。

4）根据文档样本的运行行为进行分类，包括网络行为、文件行为、内存行为、线程行为、进程行为、系统行为和注册表行为 7 种类别。

5）统计每一类行为的数量并根据该数量计算其行为特征值，将 7 种类别的行为特征值组合为该样本的动态行为特征向量。

6）提取训练样本集中所有样本的动态行为特征向量，计算特征向量的信息增益率 [14]，选取信息增益率最大的特征向量，将由该特征向量得到的行为特征作为分类特征，按照该特征划分建立子节点。信息增益率的计算公式如下

$$Gain\text{–}ratio=Gain(S,A)/I \quad (1)$$

其中，

$$Gain(S,A)=E(S)-E(S,A) \quad (2)$$

$$E(S)=\sum_{i=1}^{c} p_i \log_2 p_i \quad (3)$$

$$E(S,A)=\sum_{v\in V(A)} \frac{|S_v|}{|S|} E(S_v) \quad (4)$$

公式（1）~ 公式（4）中，*Gain-ratio* 为信息增益率，S 为样本集合，A 为特征，$Gain(S,A)$ 为信息增益，I 为分裂信息度量（用来衡量属性分裂数据的广度和均匀度）；$E(S)$ 为 S 的熵，p_i 是变量 i 的概率，c 为 S 的变量数量；$E(S,A)$ 为 A 的熵，$V(A)$ 是 A 的值域，S_v 是 S 中在 A 上值等于 v 的样本集合。

7）对子节点递归调用以上步骤建立决策树。

基于决策树的识别方法与决策树的生成方法类似：将待识别的未知样本输入虚拟沙箱以捕获动态行为；将动态行为传递到任务调度子系统以对它们进行分类，计算它们的行为特征值，组成行为向量；将行为向量带入决策树从根节点开始搜索，最终输出叶子节点，该叶子节点即为识别出的样本。基于决策树算法的识别流程如图 2 所示。

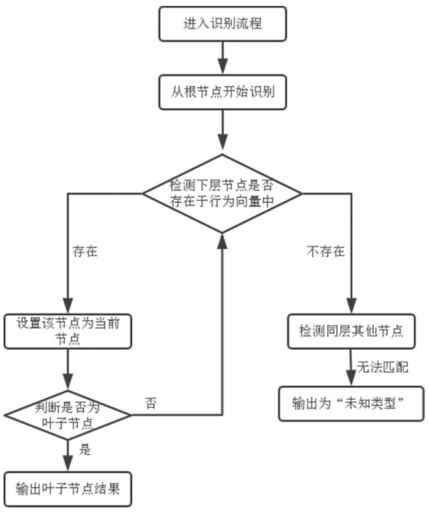

图 2 基于决策树算法的识别流程

2.4 恶意文档检测工具的具体实现

基于以上描述，本文设计了一个基于机器学习的恶意文档检测工具。该工具采用经典的 MVC 模型进行设计，主要包括以下 4 个子系统：前端可视化子系统、任务调度子系统、虚拟沙箱行为捕获子系统和基于机器学习识别子系统。

前端可视化子系统提供简单的用户登录登出操作，并提供用户上传检测样本文件接口，能将样本文件的检测结果简单友好地展示出来。

任务调度子系统完成文件格式识别和任务分发两项功能。文件格式识别基于文件 magic 进行识别，magic 匹配采用当前开源的 YARA[15] 规则匹配。

当目标进程存在过多行为时，传统沙箱严重影响主机效率 [16]，因此采用虚拟沙箱捕获样本动态运行时行为。选用 QEMU 进行设计和实现，通过 libvirt 进行统一管理 [17]。QEMU 是采用 GPL 许可证分发源码的模拟处理器 [18]，支持多种架构，仿真速度快，安装配置快捷简单，能够满足本文系统对虚拟化软件平台的基本需求。

基于机器学习识别子系统主要对已知恶意文档训练，提取行为特征并构建决策树规则，基于训练的决策树规则对未知恶意样本进行匹配和识别。

基于机器学习的恶意文档识别工具处理流程为：

1）被检测样本文件通过前端可视化子系统的上传接口上传到后台。

2）任务调度子系统接收被检测样本文件，通过 magic 方式识别出被检测文件格式。

3）根据识别出的文件格式确认样本打开执行方式，打包成 ISO 格式。

4）恢复虚拟机快照，将被检测样本通过 CD 挂载到虚拟机中并执行。

5）虚拟沙箱行为捕获子系统将捕获的动态行为发送到基于机器学习识别子系统。

6）基于机器学习识别子系统基于机器学习算法对行为进行分类清洗处理，送入决策树，通过决策树算法计算出最终的检测结果并写入数据库。

7）最终的检测结果通过前端可视化子系统进行统一展示。

基于机器学习的恶意文档识别工具系统架构如图 3 所示。

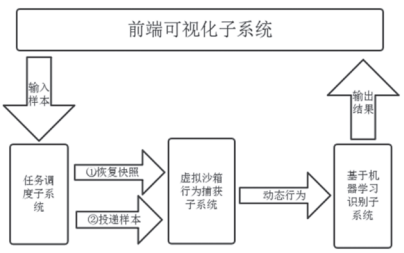

图 3 基于机器学习的恶意文档识别工具系统架构

3 实验测试与评估

3.1 实验准备

1）实验环境

主机环境采用 CentOS 7.2 操作系统，中央处理器为 Intel Xeon E5–2620 v2，内存为 32 GB。虚拟沙箱环境基于虚拟化平台 QEMU 2.8，其操作系统为 Windows 7 32，其软件环境包括 office、adobe reader 等常见文档软件。

2）训练数据集

收集 50 个恶意文档和 50 个正常文档组成样本训练集。用虚拟沙箱子系统捕获动态行为特征，根据捕获的动态行为特征组合成动态行为特征向量，以此建立决策树。

3）测试数据集

为了评估系统性能，实验主要考虑两方面性能：（1）测试系统对未知样本的识别准确率和误报率；（2）测试系统在大批量样本输入情况下的稳定性。从 virustotal 下载 245 个（其中，office 文件 112 个，pdf 文件 96 个，swf 文件 37 个）恶意样本和从本地收集整理 200 个（其中 ,office 文件 100 个，pdf 文件 68 个，swf 文件 32 个）正常样本作为系统准确率和误报率测试的实验数据。另外，随机收集 3000 个样本数据作为系统稳定性测试的实验数据。

3.2 准确率和误报率实验

利用本文系统对从 virustotal 下载的 245 个恶意样本和从本地收集整理的 200 个正常样本进行检测，测试结果如表 2 所示。

表 2 测试结果

文档样本总数/个	正确识别数/个	错误识别数/个	识别失败数/个	准确率/%	误报率/%
212（office 文件）	203	3	6	95.75	1.42
164（pdf 文件）	158	1	5	96.34	0.61
69（swf 文件）	61	2	6	88.41	2.90

由表 2 可知，本文设计可以有效捕获各类文档样本的动态行为,且具有较高的识别准确率和较低的误报率。但根据检测结果可以发现，系统还存在以下两大问题：

1）样本训练集不全。由检测结果可以发现，样本识别失败的个数超过识别错误的个数，说明构建的决策树样本训练集不够完全，后续需要增加更多的样本进行训练和学习，提高系统对恶意代码的识别率。

2）样本行为模拟不完善。通过对检测结果分析可以发现，系统对 swf 文件的识别准确率只有 88.41%，远低于 office 文件的 95.75% 和 pdf 文件的 96.34%。

此外，由对样本文件的深入分析发现，当 swf 样本存在输入参数或者内部代码加密时，系统对其处理不够完善，很难完整捕获该类型样本的所有行为，后续需要进一步研究如何能完整还原样本的真实运行场景，全面、准确地捕获样本运行行为。

3.3 稳定性实验

采用单例模式，并随机选择 3000 个样本对系统进行持续检测。测试发现，本文系统能够稳定持续运行 24×7 小时以上，具有良好的稳定性。

4 结束语

本文系统基于机器学习的方式，结合虚拟沙箱对未知文档进行动态行为分析，可以高效处理识别大规模的文档文件。在真实文档样本检测中，达到了较高的准确率及较低的误报率。但本文系统依然需要大规模的训练样本集来解决恶意文档识别过程中部分未知样本识别失败的情况。下一步将针对样本训练集不全的问题和 swf 文件行为模拟不完善的问题，通过收集更多的训练样本，研究如何更全面、准确捕获样本运行行为，以进一步提升系统性能。（责编 马珂）

参考文献：

[1] Symantec. Attackers Target Both Large and Small Businesses[EB/OL]. https://www.symantec.com/content/dam/symantec/docs/infographics/istr-attackers-strike-large-business-en.pdf,2018-3-20.

[2] MAO Ningxiang, WEN Weiping, FU Jun. Analysis of Anti-attack Key Technologies in IE Browser [J]. Netinfo Security, 2011, 11(7): 26-29.
毛宁祥，文伟平，傅军 . IE 浏览器防攻击关键技术分析 [J]. 信息网络安全 , 2011, 11(7): 26-29.

[3] CHEN Yue. The Survey of Address Space Layout Randomization (ASLR) Enhancement [J]. China Education Network, 2016(8):36-37.
CHEN Yue. 地址空间布局随机化 (ASLR) 增强研究综述 [J]. 中国教育网络 , 2016(8):36-37.

[4] YANG Chunhui, YAN Chenghua. The Analysis of the Security Strategy Based on Process Management[J]. Netinfo Security, 2014, 14(8): 61-66.
杨春晖 , 严承华 . 基于进程管理的安全策略分析 [J]. 信息网络安全 , 2014, 14(8): 61-66.

[5] LIU Lin, ZHAO Xianghui, YAO Yuangang, et al. Composite Document Malicious Code Detection Based on Spectrum Analysis[J]. Journal of Tsinghua University：Science and Technology, 2013, 53(12):1713-1718.
刘林，赵向辉，姚原岗，等 . 基于频谱分析的复合文档恶意代码检测 [J]. 清华大学学报：自然科学版 , 2013, 53(12):1713-1718.

[6] HAO Chenxi, FANG Yong. The Method of Malicious Code Detection for PDF Files Based on Spectrum Analysis[J]. Journal of Information Security Research, 2016, 2(2):166-171.
郝晨曦，方勇 . 基于频谱分析的 PDF 文件恶意代码检测方法 [J]. 信息安全研究 , 2016, 2(2):166-171.

[7] HU Jiang, ZHOU Anmin. Research on Malicious PDF Document Detection Technology for JavaScript Attack[J]. Modern Computer. 2016(1):36-40.
胡江 , 周安民 . 针对 JavaScript 攻击的恶意 PDF 文档检测技术研究 [J]. 现代计算机 , 2016(1):36-40.

[8] HUANG H D, CHUANG T Y, TSAI Y L, et al. Ontology-based Intelligent System for Malware Behavioral Analysis[EB/OL].http: / /twman.nchc.org.tw /wpcontent/uploads /2011/ 02/ss, 2010-4-16/2011-2-18.

[9] SUN Benyang, WANG Yijun, XUE Zhi. An Improved Static Monitoring Scheme for Malicious PDF Documents[J]. Computer Applications And Software, 2016, 33(3):308-313.
孙本阳，王轶骏，薛质 . 一种改进的恶意 PDF 文档静态监测方案 [J]. 计算机应用与软件 , 2016, 33(3):308-313.

[10] FENG Di, YU Min, WANG Yongjian, et al. Detecting Malicious PDF Files Using Semi-Supervised Learning Method[EB/OL]. http://www.clausiuspress.com/conferences/ACSS/ACSAT%202017/GACS25.pdf,2018-3-20.

[11] Tom Mitchell. Machine Learning[M]. ZENG Huajun, ZHANG Yinkui. Beijing: China Machine Press, 2008.
Tom Mitchell. 机器学习 [M]. 曾华军，张银奎，译 . 北京 : 机械工业出版社，2008.

[12] WEN Weiping, ZHANG Puhan, Xu Youfu, et al. Software Security Vulnerability Mining Method Based on Reference Security Patch Comparison[J]. Journal of Tsinghua University：Science and Technology, 2011(10): 1264-1268.
文伟平，张普含，徐有福，等 . 参考安全补丁比对的软件安全漏洞挖掘方法 [J]. 清华大学学报：自然科学版，2011(10):1264-1268.

[13] Adobe. PDF Reference[EB/OL]. http://www.adobe.com/devnet/pdf/pdf_reference.html,2018-3-20.

[14] ZHANG Xiaokang, SHUAI Jianmei, SHI Lin. Malicious Code Detection Method Based on Weighted Information Gain[J]. Computer Engineering, 2010(6):149-151.
张小康 , 帅建梅 , 史林 . 基于加权信息增益的恶意代码检测方法 [J]. 计算机工程，2010(6) :149-151.

[15] Microsoft. File System Minifilter Drivers[EB/OL]. https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/file-system-minifilter-drivers,2017-4-20.

[16] CHENG Sanjun, WANG Yu. Analysis of APT Attack Principle and Protection Technology[J]. Netinfo Security, 2016,16(9): 118-123.
程三军，王宇 . APT 攻击原理及防护技术分析 [J]. 信息网络安全，2016,16(9): 118-123.

[17] JIN Xin, CHEN Xingshu, ZHAO Cheng, et al. Trusted Attestation Architecture on an Infrastructure-as-a-service[J]. Journal of Tsinghua University, 2017, 22(5):469-477.
金鑫，陈兴蜀，赵成，等 . 基于基础设施即服务的可信认证架构 [J]. 清华大学学报 ,2017,22(5):469-477.

[18] YAO Huachao, WANG Zhenyu. Construction of Virtualization Resource Pool Based on KVM-QEMU and Libvirt[J]. Computer and Modernization, 2013(7):26-33.
姚华超，王振宇 . 基于 KVM-QEMU 与 Libvirt 的虚拟化资源池构建 [J]. 计算机与现代化，2013(7):26-33.