恶意 PDF 文档检测技术研究进展*

林杨东，杜学绘，孙 奕
（信息工程大学 河南省信息安全重点实验室，郑州 450004）

摘 要：针对 PDF 的漏洞及相应攻击手段日新月异，传统的恶意 PDF 文档检测技术难以应对各种新型威胁。目前针对恶意 PDF 文档检测的研究已取得一定成果，为了更深入地解决该技术存在的不足，采用文献分析方法，首先讨论了必要性、简述了其相关概念和检测基本框架；其次针对其分析技术的不同将现有方案进行分类，从适用范围、检测效果、检测效率等多个方面进行对比分析。最后归纳了该领域当前的热点和发展前景。

关键词：PDF；文档检测；静态分析；动态分析
中图分类号：TP309.2 **文献标志码**：A **文章编号**：1001-3695(2018)08-2251-05
doi:10.3969/j.issn.1001-3695.2018.08.003

Survey of malicious PDF documents detection

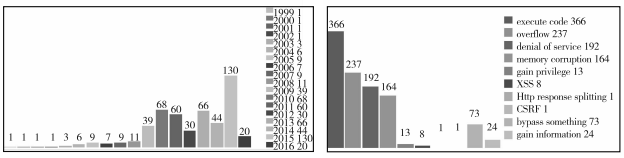
Lin Yangdong, Du Xuehui, Sun Yi
(*Henan Provincial Key Laboratory of Information Security*, *Information Engineering University*, *Zhengzhou* 450004, *China*)

Abstract: The vulnerability of PDF and targeted attacks using malicious PDF, it made a great threat to the network office environment of the government, enterprises, and important organizations, so malicious PDF document detection technology has gradually become the hot spot in the study of network security in recent years. Although the malicious PDF document detection technology has made some achievements, this paper was to find deficiencies of existing schemes. Firstly, it discussed the necessity and briefly introduced its related concepts and basic framework of detection. Secondly, according to the differences of its analysis technology, it divided the existing schemes into several categories and concluded the schemes from the aspects of application scope, detection effect and detection efficiency. Finally, it pointed out the existing problems and development prospects so as to provide reference for further research.

Key words: PDF; document detection; static analysis; dynamic analysis

0 引言

PDF 文件格式[1]以其高效性、稳定性和交互性在文档交互方面得到了广泛的应用。然而，近几年随着非可执行文件攻击技术和 APT 等攻击手段的兴起与发展，其安全性受到极大的挑战[2]。如图 1 所示，CVE 漏洞库统计的数据显示，针对 PDF 的漏洞呈逐年增长趋势，仅 2015 年一年，就曝出了 120 个 PDF 相关漏洞。而近年来，利用钓鱼邮件嵌入恶意 PDF 文档，结合社会工程学攻击政府、组织和企业的案例也屡见不鲜，其主要攻击意图在于窃取机密信息，监视和破坏政府、组织或企业的活动。



(a)vulnerabilities by year (b)vulnerabilities by type

图1 CVE公布的1999—2016年以来PDF漏洞情况

2014 年 1 月 15 日，发生了针对以色列国防部的攻击事件，根据媒体报道，其攻击过程是利用电子邮件，在其附件中添加恶意 PDF 文档并将其伪装为以色列国防部文件，打开时会自动安装木马使攻击者能完全控制计算机[3]。2014 年，黑客通过在钓鱼邮件附件中使用恶意 PDF 文件，成功入侵了索尼加利福尼亚公司内网并下载了上千份敏感文件和机密邮件[4]；2016 年 6 月，美国民主党委员会网络攻击和信息泄露事件曝出，对美国大选造成了极其关键的影响，有种种迹象和证据表明攻击者为俄罗斯黑客组织 APT28、APT29，而其利用的重要手段便是钓鱼邮件结合恶意文档[5]；2016 年 9 月，PaloAlto 研究中心的安全研究人员首先发现了针对苹果系统（OSx）的新型钓鱼邮件攻击，其技术核心在于钓鱼邮件中携带有木马病毒的恶意 PDF 文档[6]。种种案例表明，现有的针对恶意 PDF 文件的检测方案在检测和防止此种类型的攻击方面还存在一定的不足，还需要进一步的提升改进。知名安全公司 Recorded Future 调查表明，在美国的电脑系统中，有 85% 的用户安装有 Java 和 Adobe PDF 软件，而其存在着大量漏洞可被攻击者利用。可以预见的是，随着 PDF 格式的文档在我国的广泛推广，此类问题危害性与日俱增，将严重威胁到政府、大型组织、大企业等的办公安全。

PDF 文档的安全问题主要集中体现在三个方面：a）PDF 文件格式是一种非常灵活的中间载体[7]，因而可以被嵌入许多不同类型的攻击，如通过精妙设计的恶意 JavaScript 和 ActionScript 代码、恶意的可执行文件、恶意 PDF 文件[8,9]等；b）PDF 结构标准中的加密、压缩等技术给攻击者隐藏其攻击代码及行为提供了便利；c）外部的相关应用如 Adobe 阅读器、福昕阅读器应用程序以及 Adobe Flash 应用的某些特定漏洞可以被攻击者利用，从而将其在 PDF 文件中触发以进行攻击[10]。

目前大部分杀毒软件采用的基于启发式或字符串匹配采用的基于签名的方法无法应对新的攻击，并且已经被证明在应对多态攻击方面存在一定的不足。为了解决这些不足，最近的研究工作主要集中在两个不同的方面：a）通过静态和动态分析，着重检测 PDF 文件中的 JavaScript 代码；b）利用 PDF 文件的结构和元数据信息来检测恶意 PDF 文件，而不分析其携带的攻击代码或漏洞。第二种方法已经被证明比第一种方法更有效，因为相比第一种方法，它们能检测非 JavaScript 攻击，并

收稿日期：2017-06-23；**修回日期**：2017-08-11 **基金项目**：国家“863”计划资助项目（2015AA016006）；国家自然科学基金资助项目（61502531，61702550）

作者简介：林杨东（1992-），男，广东汕头人，硕士研究生，主要研究方向为信息安全、数据安全（xd_lyd@163.com）；杜学绘（1968-），女，教授，博士，主要研究方向为多级安全、云安全；孙奕（1979-），女，讲师，博士，主要研究方向为云计算、数据安全、安全交换.

且效率更高。然而进一步研究表明,第二种方法极易受到逃逸攻击。因此,研究再次重点关注恶意文档中 JavaScript 代码的检测,并且通过采用沙箱来强化其安全性。

本文尝试对恶意 PDF 文档识别的思路和技术方法进行归纳和总结,介绍恶意 PDF 文档检测的整体框架,详细分析恶意文档检测的研究现状并探讨面临的问题与挑战,为下一步研究提供参考。

1 基本概念及框架

PDF[11](portable document format)是由 Adobe Systems 在1993 年给出的用于文件交换的格式。它的优点在于跨平台、能保留文件原有格式、开放标准,能免版税自由开发 PDF 相容软体,是一个开放标准,2007 年 12 月成为 ISO 32000 国际标准。自 PDF 文件格式提出以来,便以其高效性、稳定性和交互性在文档交互方面得到了广泛的应用,然而其格式结构的灵活性也为其带来了一系列安全性问题[12]。

所谓恶意 PDF 文档,指的是通过嵌入并执行恶意代码或利用其结构特性,以窃取敏感信息、监视和破坏用户正常活动等恶意行为为目的的 PDF 文档。恶意 PDF 文档检测,即通过对 PDF 文档的结构与内容进行分析,确定其是否存在恶意性并进行进一步处理,从而防止用户因使用恶意 PDF 文档而造成敏感信息泄露等安全风险。当前大部分杀毒软件都有查杀 PDF 文档的功能,但主要的检测方法仍然是传统的基于标签和严格的启发式规则的方法[13]。PDF 文档检测框架如图 2 所示,其核心在于特征抽取和文档判别。

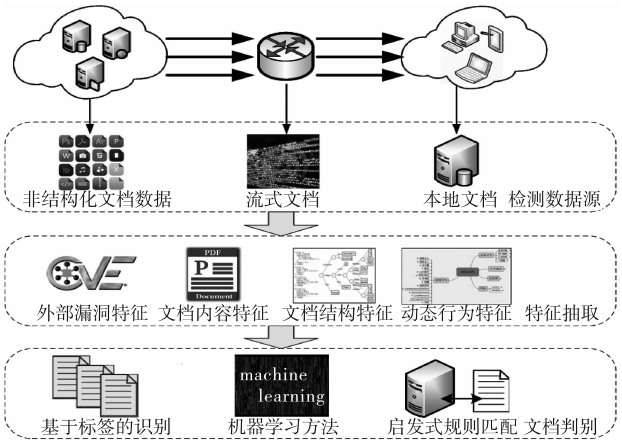


图2 恶意PDF文档检测基本框架

1)特征抽取

特征抽取是指根据判别方法以及不同类型恶意文档的实际情况,抽取出可作为判别恶意性依据的信息。这些特征主要包括 JavaScript 特征[14]、元数据特征、动态执行特征等。如图 3 所示,针对恶意 PDF 文档的特征主要分为静态特征[15]和动态特征两类。静态特征主要来源于文档的静态信息,其抽取过程一般较为简单,常见的静态特征主要包括结构特征(结构路径、敏感对象数量等)和内容特征(JavaScript、压缩数据等)。而动态特征主要来自于文档阅读器打开或执行文档时的动态行为,其抽取过程一般较为复杂。常见的动态特征主要有系统调用特征、缓冲区数据特征、跳转关系等,这些特征一般利用虚拟执行环境来执行并在执行过程中经过复杂的分析、判别获得。

2)文档判别

目前恶意 PDF 文档识别的方法主要有基于标签、基于启发式规则以及基于机器学习的识别方法[16]。传统的基于标签和启发式规则的识别方法的主要问题在于无法有效地应对未知类型的恶意 PDF 文档,且维护成本高,需要大量的人工分析来完成规则库及特征库的更新[17]。近几年,随着机器学习技术的深入发展和完善,基于机器学习的识别技术得到了广泛的认可,可弥补传统方法的不足。

基于机器学习的识别方法[18,19]一般包括学习过程和测试过程两部分。在学习阶段首先选取一批已知标签的样本作为训练集,并通过特征抽取、特征数据预处理等步骤得到特征集,最后利用机器学习算法进行训练得到分类模型;测试阶段将待分类样本通过特征提取及特征数据预处理后输入分类模型进行测试,最终得到相应的检测结果。

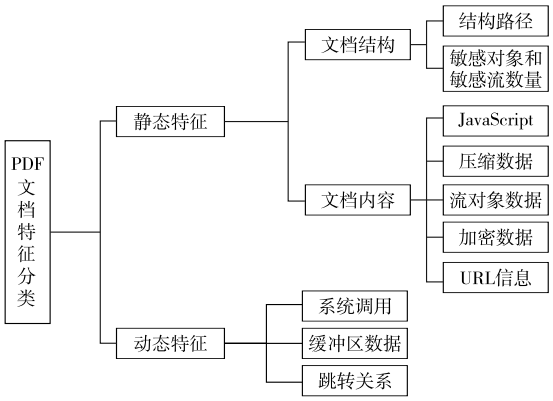


图3 识别恶意PDF文档的特征分类

2 典型技术分析

目前,针对恶意 PDF 文件的检测技术根据其所针对的特征不同,一般可以分为三类:基于静态分析的检测技术、基于动态分析的检测技术和动静态混合分析的检测技术[20]。

2.1 基于静态分析的恶意 PDF 文件检测技术

在 PDF 文件分析的起步阶段,主要使用的是完全静态分析的方法。早期静态分析的主要方法是使用类似于特征码扫描的方法,很容易被攻击者绕过[21]。

随着机器学习技术的发展,Li 和 Shafiq 等人[22,23]首先提出了“格式不可知论”,提出一种利用机器学习结合字节级的 *n*-gram 分析来检测恶意文件内容,然而早期的在 *n*-gram 分析的基础上完成的静态分析方法还未在 PDF 检测方面上尝试过,其主要原因在于受编码、压缩和加密等的影响,无法定位到 PDF 格式中的主要特性。此后,基于静态分析的 PDF 文件分析方法主要可分为两类,第一类的分析重点在于 PDF 文件中的 JavaScript 代码,另一类则着重于分析 PDF 文件中的元数据。

2011 年,Laskov 等人[24]设计了 PJScan,利用单类支持向量机(OCSVM)技术,通过对 PDF 文件中的 JavaScript 代码进行静态分析进而检测恶意 PDF 文件,其检测率能达到 85.17%,误报率为 17.35%。其主要不足在于,PJScan 为了提高效率,其 JavaScript 抽取器的查找目标只限定在 PDF 文档标准中规定的可能出现 JavaScript 代码的位置,这就可能导致攻击者将可以通过 PDF 的相关 API 函数将 JavaScript 代码放在任意的可达位置,并且通过类似于 eval()的函数调用将其重新恢复。

Vatamanu 等人[25]于 2012 年提出一种针对 JavaScript 的 PDF 文件聚类方法,利用分层聚类和哈希表聚类技术,根据 PDF 文件中经过混淆的 JavaScript 代码进行聚类并形成“指纹”。该实验得出一个重要的结论是,93% 的恶意 PDF 文件包括有 JavaScript,而正常 PDF 文件中仅有 5% 含有 JavaScript,这说明 JavaScript 可以作为分析恶意 PDF 文件的一个重要关注点。然而,在 PDF 文件分析中,JavaScript 代码的定位、抽取及解析始终具有较大难度,因此针对元数据的分析方法应运而生。

Maiorca 等人[26]于 2012 年提出了 PDF malware slayer,通过基于模式识别的方法进行 PDF 文件关键字特征的抽取,并根据文件特征利用随机森林算法进行分类。此方法关注的重要关键字主要有/JS、/JavaScript、/Encrypt、filter、stream 等。其缺点在于所抽取出来的关键字特征是有限的,且无法解决处理隐藏在数据流中的经过多样化处理的目标。此外,攻击者还可

以通过学习影响正常 PDF 的关键词并引入到恶意 PDF 中从而绕过检测。同年,Smutz 等人[27]提出了另一种基于内容元数据的检测系统 PDFRate,并设计了一种特殊的 PDF 解析器从而更好地提取出其所需要的特征,其关注的元数据特征包括字体对象的数量、流对象长度等共 202 个特征。在多种分类算法中,随机森林分类效果最好,其主要不足在于无法抽取出一个对象中的流数据,而流数据又很容易被用来对 PDF 文档的特征进行隐藏,从而避过检测器的检测。

随后,Šrndic 等人[28]提出了结构路径的概念,并设计了一种基于结构化路径的恶意 PDF 检测方法,将结构路径作为检测特征,分别在支持向量机(SVM)和决策树两种算法下进行了训练与测试,其检测效果较好并且具有较强的应对未知安全威胁的能力,然而其健壮性较弱。Maiorca 等人[29]针对此方案提出了基于结构检测的绕过方法 reverse mimicry,并通过测试证明了此类攻击的可行性和简易性。

为解决针对基于结构检测的绕过问题,Maiorca 等人[30]提出了一种基于结构和内容的检测方案,该方案使用 Peepdf 工具和 Origami 工具对 PDF 文件进行解析,从而得到文件的一般结构、对象的结构以及文件内容信息,并利用机器学习算法进行学习分类,从而得到检测模型。其主要不足是模型侧重于关注样本的有效特征,因此对样本数据的质量要求较高,其需要对函数的参数进行人工优化。

另一种独特的 PDF 文件的分析方法是 Pareek 等人[31]于 2013 年提出的,基于熵和 *n*-gram 分析的 PDF 分析方法,其最大的特点在于无须解析 PDF 内容,而是直接对整个 PDF 文件进行分析。此方法将 PDF 文件转换为一个字节序列集合,并计算其熵值(通过实验得到结论:恶意 PDF 熵值比正常 PDF 低),并将熵值较低的文件作为可疑文件进行 *n*-gram 分析。尽管其实验结果良好,但无法进行具体的解释分析,因此该方法更适用于检测前的过滤或预检测。

为解决传统的检测方案针对新型安全威胁能力不足以及需要大量人工干预的问题,Nissim 等人[32,33]提出了一个基于主动学习的恶意 PDF 文件检测方案,依次对 PDF 文件进行启发式教学检测、可用性检测,并使用 SVM-margin 积极学习算法构建基于结构的检测模型并进行检测和再学习。其主要优势在于降低了人工分析 PDF 文档的工作量,但检测过程较为复杂,效率较低。

总的来说,基于静态分析的检测方法无须执行 PDF 文件中的代码,占用计算资源少、分析速度较快,但难以针对经过混淆的代码,并且易于被绕过。

2.2　基于动态分析的恶意 PDF 文件检测技术

早期的动态分析的主要思想是 2002 年 Toth 等人[34]提出的,其主要思路是把一个可疑的目标用虚拟执行器执行从而进行安全分析。随后 Akritidis 和 Polychronakis 等人[35,36]提出了软件仿真的方法,即利用软件环境对可疑目标进行执行和仿真,从而得到相应的结果。其主要问题在于软件仿真无法完全覆盖指令集,因而容易被检测出来从而绕过检测。

2007 年 Willems 等人[37]提出了自动化动态二进制分析的方法,并设计了 CWSandbox,它在一个仿真的运行环境启动 Adobe Reader 软件逐字加载 PDF 文档样本,然后通过监测系统的调用和修改来检测恶意行为。其主要问题是攻击只能在特定的系统环境下被检测到,并且在将沙箱恢复到原始状态时需要消耗额外的资源。2009 年,Engelberth 等人[38]在 CWSandbox 的基础上设计了 MalOffice 系统,它使用 PDFtk 提取 JavaScript 代码,然后利用 CWSandbox 通过一组规则来进行代码分类从而分析代码行为。

为了解决软件仿真的问题、提高检测的可扩展性, Snow 等人[39]于 2011 年提出了一种轻量级的操作系统内核 Shellos,通过使用虚拟化的硬件来代替仿真,从而监控 shellcode 的执行过程,有效地检测应用分配的缓冲区中的 shellcode。其检测率能达到 80.24%,检测效率为每个文件 7 400 ~ 25 460 ms。其缺点在于成本较高、效率较低,当 ShellOS 被应用于检测网络级的攻击时,即使有良好的带宽,也会出现瓶颈问题,因为每一个检测过程要占用内存缓冲区,且需要在分析前就分配给各个应用。

另一类动态分析方法主要关注于在执行 JavaScript 代码过程中的恶意行为。2010 年 Cova 等人[40]设计了 JSand,使用了 10 个通过精确设计的启发式特征,通过训练得到正常 JavaScript 代码的模型,一旦 JavaScript 代码与此模型偏差较大,则认为检测到攻击(恶意软件)。2012 年,Overveldt 等人[41]将类似的方法成功地应用于检测 ActionScript。针对 JavaScript 的动态分析方法是在检测 shellcode 的方法上进行改进优化的,在保证高检测精度和极低误报率的基础上,检测效率能达到每个文件数百毫秒。

随着网络数据量的增大和计算机计算能力的提高,云计算也被应用于恶意 PDF 的监测与保护。2014 年,Maass 等人[42]提出了 in-Nimbo sandboxing 系统,该系统将对恶意 PDF 文件漏洞利用行为和恶意感染行为的分析放置在远程云计算环境中实例化的虚拟机,以此保证用户主机不会被感染。

总的来说,基于动态分析的检测方法一般需要对文档或其恶意内容进行执行,可直观地发现攻击行为及其攻击目的,健壮性较强;但占用资源较大,检测效率较低。

2.3　基于动静态混合分析的恶意 PDF 文件检测技术

动静态混合分析的方法结合了动态和静态方法的优点,并逐渐成为恶意 PDF 分析方法的主流。2010 年, Rieck 等人[43]提出了 Cujo,此模型结合静态分析(语法分析)和建立在 ADSandbox 沙盒[44]之上的动态分析,对恶意 JavaScript 代码进行检测,并且能通过自动学习得到影响 JavaScript 编译器的状态事件序列。

Tzermias 等人[45]在 2011 年设计了 MDScan 系统,它首先对 PDF 文档的结构进行静态分析,并从中解析出所有对象,着重关注包含 JavaScript 的对象以及在交叉引用表中不存在记录的潜在恶意对象,并从对象中抽取出相应的 JavaScript。这些被抽取出来的 JavaScript 会被放到 SpiderMonkey 的仿真 JavaScript 引擎中执行,并监控其执行过程中使用的所有内存缓冲区,一旦某些恶意 shellcode 出现在这些地址空间中,则认为该文件为恶意文件。通过实验,此方法的检测率为 89.34%。其主要问题是,无论利用静态分析对代码的提取还是动态执行 JavaScript 均是在模拟环境下,缺乏在正常的 Adobe 环境中一些专有功能。因此,这可能会导致一些意外的结果,如 Adobe Reader 突然终止运行等。

紧随之后 Curtsinger 等人[46]又提出了 ZOZZLE 系统用以分析 JavaScript 代码, 在 ZOZZLE 中,其动态和静态部分的功能则是相反的。ZOZZLE 中利用动态分析技术来提取 JavaScript,在 JavaScript 代码执行时将其从 IE 的 JavaScript 引擎中抽取出来,由此解决 JavaScript 的混淆问题,并采用基于语法分析的贝叶斯分类来检测 JavaScript 代码。其主要问题是模型复杂,且性能开销较大。

2012 年,Schmitt 等人[47]提出了 PDF Scrutinizer 工具,该方法利用静态分析技术提取 JavaScript 的操作,并利用动静态混合分析技术,结合启发式检测对 PDF 文件进行检测,可以有效抵制代码混淆技术、漏洞利用、堆喷射和内嵌恶意文件等攻击手段。但存在攻击者利用不同检测环境之间的差异来进行检测绕过的问题。

2013 年,Lu 等人[48]提出了 MPScan 系统,该方法结合动态 JavaScript 代码反混淆技术和静态恶意软件检测技术,通过挂接 AdobeReader 的本地 JavaScript 引擎,在执行过程中,利用动态代码提取模块将内嵌代码提取出来,并利用多级静态恶意软件检测模块对恶意文件进行检测,其主要特点在于首次提出利用 AdobeReader 本地的 JavaScript 引擎来提取 JavaScript 从而避免了代码混淆的干扰,相比基于动态分析的方案极大地减少了时间开销。

2014年,Liu等人[49]利用动静态混合分析的方法在JavaScript代码执行过程中检测其潜在的恶意行为。该方法首先抽取出一组静态特征,并在PDF文档中嵌入环境监控代码,用以在JavaScript代码执行过程中进行监控并提取行为特征,并通过静态特征和动态行为特征对文档进行检测。此方案具有较高的检测率,但其局限性在于仅对使用JavaScript类型的恶意文档有效。

随后,Corona等人[50]提出了Lux0R,该方法利用API调用信息来检测恶意PDF文件,利用动态分析技术将PDF文件中的JavaScript代码翻译为相应的API调用信息,并利用通过学习正常PDF文件和恶意PDF文件的API调用信息来生成检测模型。其优点在于模型轻便高效、可部署在终端用于实时检测;不足之处在于受API调用信息的抽取效果影响较大,且攻击者可能利用梯度下降攻击来自动找到恶意API调用模式从而绕过检测。

总的来说,基于动静态混合分析的检测方法较好地利用了静态分析和动态分析各自的优势,其检测内容更多地关注于文档中的JavaScript特征,能较好地应对代码混淆等手段。其主要不足是分析过程较为复杂,模型建立困难。

3 面临的挑战与前景展望

经过研究人员多年的不懈努力,恶意PDF文档技术逐步发展起来,并取得了一定的成果。

通过对这些相关工作的对比,将近年来恶意PDF文档相关检测技术及系统总结如表1所示,主要包括系统的数据规模、检测效率、准确率、误报率等重要评价参数。

表1 恶意PDF文档检测技术

系统名称	类型	恶意样本量	正常样本量	准确率/%	误报率/%
PjScan	静态	906	30 157	71.94	0.11
ShellOS	动态	105	179	80.24	–
MDScan	动静态	197	2 000	89.34	0
MPScan	动静态	207	500	98	–
PDFMS	静态	11 157	9 989	99.55	2.51
PDF Scrutinizer	动静态	11 278	6 054	90	0
PDFRate	静态	5 000	100 000	99 +	0.244
entropy and *n*-gram	静态	65 536	46 933	99.22	0.6
structural paths	静态	82 142	576 621	99.88	0.1
JavaScript clustering	静态	997 615	1 333 420	–	–
Lux0R	动静态	12 548	5 234	99.27	0.05
structural and content	静态	11 138	9 890	99.80	0.068

基于静态分析的检测技术关注的重点在于PDF文档中的内容特征、结构特征等,其主要优点在于检测过程简单轻便,检测效率高;但对于内容特征的提取分析而言,存在定位难、代码混淆等问题,而针对结构特征的分析则易于被攻击者绕过,因此其健壮性较差。

基于动态分析的检测技术的关注点主要在于PDF文档的行为特征,如系统调用与修改、shellcode执行过程[51]等,其主要优点在于健壮性强,不容易被绕过;但其占用资源较多,耗费时间长且一般需要特定的软硬件支持,检测效率较低。

基于动静态混合分析的检测技术的主要关注点在于文档中的JavaScript代码,其核心在于充分利用静态分析的轻便高效以及动态分析的高健壮性,利用动态分析技术来解决JavaScript的定位和混淆问题,提高健壮性,利用静态分析技术对JavaScript进行分类,保证检测率与检测效率。其优点在于检测率较高,健壮性强,且相较于动态分析在占用资源和耗时上有了较大的提升;其主要不足在于不关注PDF文档的元数据特征,仅对携带恶意代码类型的恶意PDF文件敏感。

通过对近几年文献的阅读分析发现,当前针对恶意PDF文档检测关注的特征点主要有内容特征、结构特征和行为特征三类[52]。其中内容特征主要关注JavaScript代码及其他元数据等,结构特征主要关注文件结构等,行为特征主要关注执行过程的API调用及内存信息等。在构建恶意PDF文档分类模型方面,所采用的机器学习算法主要有贝叶斯算法、决策树、支持向量机等,所得到的分类模型检测效果较为良好。近几年随着计算能力的提升,数据挖掘技术[53]以及深度学习算法等在恶意代码检测等方面也取得了一定的进展[54,55],但仍未被应用于文档的检测。

仔细分析不难发现,现有的检测技术在检测准确率上已经较好,其主要的问题是健壮性较差。因此后续的恶意PDF文档检测的研究热点及趋势也主要在于检测的健壮性。

a)特征选取方面,已有的特征选取方式比较单一,每种特征都有其存在的弊端及相应的攻击绕过方法,如JavaScript特征存在定位难、代码混淆等问题,结构特征易于被窜改绕过、行为特征由于需要执行,往往在占用资源和耗时上不占优等。所以研究如何对特征的选取进行优化与组合,可以从根本上提高检测模型的检测率和健壮性。

b)决策算法方面,当前文档检测领域主流的决策算法在构建模型上存在着过度依赖于初始学习数据的问题,即不同的初始学习数据,所建立的模型在检测效果上可能存在极大差异,并且为了构造一个良好的检测模型,往往还需要在学习过程中手动调整参数,模型适应性较差。因此,研究模型的决策算法可以有效降低模型对初始学习数据的依赖程度,有助于提高模型的健壮性。

c)检测效率方面,现有的检测模型大多数为离线检测,检测效率难以支撑实时检测的需要,尤其是当前PDF文档的浏览器上的显示大多为边加载边显示,这种情况下,针对此种类型的检测效率就需要有较大的提高。因此,如何提高检测模型的检测效率,使其适用于在线实时检测,也是当前的研究热点以及难点。

4 结束语

近年来随着PDF相关漏洞及安全事件的频发,恶意PDF文档检测技术已成为研究的热点并得到了广泛的应用。针对这一问题,本文首先梳理了PDF文件的安全性问题,然后介绍了恶意PDF文档检测框架,进而对不同类别的检测方案进行分析与比较,分别指出它们的优势、不足以及适用场景。最后,讨论了现有检测技术面临的挑战并介绍了未来可能的研究方向。

参考文献:

[1] PDF格式[EB/OL]. https://baike.baidu.com/item/pdf格式/8426775?fr=aladdin.

[2] 周可政,施勇,薛质.基于恶意PDF文档的APT检测[J].信息安全与通信保密,2016(1):131-136.

[3] Roychoudhury A, Liu Yang. A systems approach to cyber security [C]// Proc of the 2nd Singapore Cyber-Security R&D Conference. Amsterdam:IOS Press,2017.

[4] Rechange.真相只有一个:入侵索尼影视的居然是俄罗斯黑客?[EB/OL].(2015-02-07).http://www.freebuf.com/news/58552.html.

[5] Clouds.揭秘:俄罗斯APT漏洞利用工具包[EB/OL].(2016-08-12).http://www.freebuf.com/articles/network/111490.html.

[6] Clouds.盘点2016年针对苹果Mac系统恶意软件[EB/OL].(2017-01-12).http://www.freebuf.com/articles/system/124728.html.

[7] 武雪峰.恶意PDF文档的分析[D].济南:山东大学,2012.

[8] Blonce A, Filiol E. Portable document format (PDF) security analysis and malware threats[J]. Images Paediatr Cardiol, 2008, 10(2):1-3.

[9] Itabashi K. Portable document format malware[EB/OL].(2011-01-13)[2017-06].https://www.symantec.com/.

[10] Ulucenk C, Varadharajan V, Balakrishnan V, *et al*. Techniques for analysing PDF malware[C]//Proc of Asia-Pacific Software Engineering Conference. Washington DC: IEEE Computer Society,2011:41-48.

[11] PDF reference: version 1.7[R/OL].(2010-09-29).https://www.loc.gov/preservation/digitall formats/fdd/fdd000277.shtml.

[12] Stevens D. Malicious PDF documents explained[J]. IEEE Security & Privacy,2011,9(1):80-82.

[13] 陈亮,陈性元,孙奕,等. 基于结构路径的恶意 PDF 文档检测[J]. 计算机科学,2015,42(2):90-94.
[14] 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.
[15] 丁晓煌. 恶意 PDF 文档的静态检测技术研究[D]. 西安:西安电子科技大学,2014.
[16] Gandotra E, Bansal D, Sofat S. Malware analysis and classification: a survey[J]. Journal of Information Security,2016,5(2):56-64.
[17] Baccas P. Finding rules for heuristic detection of malicious PDFS: with analysis of embedded exploit code[C]//Proc of Virus Bulletin Conference. 2010.
[18] Shabtai A, Moskovitch R, Elovici Y, *et al*. Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey[J]. Information Security Technical Report, 2009,14(1):16-29.
[19] Šrndić N, Laskov P. Hidost: a static machine-learning-based detector of malicious files[J]. EURASIP Journal on Information Security, 2016,2016(1):22-41.
[20] Policicchio S. Bulk analysis of malicious PDF documents [D]. Pittsburgh:University of Pittsburgh,2015.
[21] 孙本阳. PDF 文档的安全性检测技术研究[D]. 上海:上海交通大学, 2015.
[22] Li W J, Stolfo S, Stavrou A, *et al*. A study of malcode-bearing documents [C]//Proc of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin: Springer-Verlag, 2007: 231-250.
[23] Shafiq M Z, Khayam S A, Farooq M. Embedded malware detection using Markov n-grams [C]//Proc of International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin: Springer, 2008: 88-107.
[24] Laskov P, Šrndić N. Static detection of malicious JavaScript-bearing PDF documents [C]//Proc of the 27th Computer Security Applications Conference. New York:ACM Press,2011: 373-382.
[25] Vatamanu C, Gavrilut D, Benchea R. A practical approach on clustering malicious PDF documents[J]. Journal in Computer Virology,2012,8(4):151-163.
[26] Maiorca D, Giacinto G, Corona I. A pattern recognition system for malicious PDF files detection [C]//Proc of International Conference on Machine Learning and Data Mining in Pattern Recognition. Berlin: Springer-Verlag,2012:510-524.
[27] Smutz C, Stavrou A. Malicious PDF detection using metadata and structural features[C]//Proc of the 28th Annual Computer Security Applications Conference. New York:ACM Press,2012:239-248.
[28] Šrndić N, Laskov P. Detection of malicious PDF files based on hierarchical document structure [C]//Proc of the 20th Annual Network & Distributed System Security Symposium. 2013.
[29] Maiorca D, Corona I, Giacinto G. Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection[C]//Proc of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. New York: ACM Press, 2013: 119-130.
[30] Maiorca D, Ariu D, Corona I, *et al*. A structural and content-based approach for a precise and robust detection of malicious PDF files [C]//Proc of International Conference on Information Systems Security and Privacy. Piscataway,NJ:IEEE Press,2015:27-36.
[31] Pareek H, Eswari P R L, Babu S C. Entropy and *n*-gram analysis of malicious PDF documents[J]. International Journal of Engineering Research & Technology,2013,2(2):1-4.
[32] Nissim N, Cohen A, Moskovitch R, *et al*. ALPD: active learning framework for enhancing the detection of malicious PDF files[C]// Proc of IEEE Joint Intelligence and Security Informatics Conference. Piscataway,NJ:IEEE Press,2014:91-98.
[33] Nissim N, Cohen A, Moskovitch R, *et al*. Keeping pace with the creation of new malicious PDF files using an active-learning based detection framework[J]. Security Informatics,2016,5(1):1-20.
[34] Toth T, Kruegel C. Accurate buffer overflow detection via abstract pay load execution[C]//Proc of the 5th International Conference on Recent Advances in Intrusion Detection. Berlin:Springer,2002:274-291.
[35] Akritidis P, Markatos E P, Polychronakis M, *et al*. STRIDE: polymorphic detection through instruction sequence analysis[C]//Proc of IFIP International Conference on Information Security and Privacy in the Age of Ubiquitous Computing. Boston:Springer,2005:375-392.
[36] Polychronakis M, Anagnostakis K G, Markatos E P. Comprehensive shellcode detection using runtime heuristics[C]//Proc of the 26th Computer Security Applications Conference. New York:ACM Press, 2010:287-296.
[37] Willems C, Holz T, Freiling F. Toward automated dynamic malware analysis using CWSandbox[J]. IEEE Security & Privacy,2007,5(2):32-39.
[38] Engelberth M, Willems C, Holz T. MalOffice: detecting malicious documents with combined static and dynamic analysis[C]//Proc of Virus Bulletin International Conference. 2009.
[39] Snow K Z, Krishnan S, Provos N, *et al*. Shellos: enabling fast detection and forensic analysis of code injection attacks[C]//Proc of the 20th USENIX Conference on Security. Berkeley: USENIX Association,2011:9.
[40] Cova M, Kruegel C, Vigna G. Detection and analysis of drive-by-download attacks and malicious JavaScript code[C]//Proc of International Conference on World Wide Web. New York: ACM Press, 2010:281-290.
[41] Van Overveldt T, Kruegel C, Vigna G. FlashDetect: actionScript 3 malware detection [C]//Proc of the 15th International Workshop on Research in Attacks, Intrusion and Defenses. Berlin: Springer, 2012:274-293.
[42] Maass M, Scherlis W L, Aldrich J. In-Nimbo sandboxing[C]//Proc of Symposium and Bootcamp on the Science of Security. New York: ACM Press,2014:Article No 1.
[43] Rieck K, Krueger T, Dewald A. Cujo: efficient detection and prevention of drive-by-download attacks[C]//Proc of the 26th Computer Security Applications Conference. New York:ACM Press,2010:31-39.
[44] Dewald A, Holz T, Freiling F C. ADSandbox: sandboxing JavaScript to fight malicious websites[C]//Proc of ACM Symposium on Applied Computing. New York: ACM Press,2010:1859-1864.
[45] Tzermias Z, Sykiotakis G, Polychronakis M, *et al*. Combining static and dynamic analysis for the detection of malicious documents [C]// Proc of the 4th European Workshop on System Security. New York: ACM Press,2011:ArticleN0 4.
[46] Curtsinger C, Livshits B, Zorn B, *et al*. ZOZZLE: fast and precise in-browser JavaScript malware detection[C]//Proc of the 20th USENIX Conference on Security. Berkeley:USENIX Association,2011:3.
[47] Schmitt F, Gassen J, Gerhards-Padilla E. PDF Scrutinizer: detecting JavaScript-based attacks in PDF documents[C]//Proc of the 10th International Conference on Privacy, Security and Trust. Washington DC:IEEE Computer Society,2012:104-111.
[48] Lu Xun, Zhuge Jianwei, Wang Ruoyu, *et al*. De-obfuscation and detection of malicious PDF files with high accuracy[C]//Proc of the 46th Hawaii International Conference on System Sciences. Washington DC:IEEE Computer Society,2013:4890-4899.
[49] Liu Daiping, Wang Haining, Stavrou A. Detecting malicious JavaScript in PDF through document instrumentation[C]//Proc of IEEE/IFIP International Conference on Dependable Systems and Networks. Washington DC:IEEE Computer Society,2014:100-111.
[50] Corona I, Maiorca D, Ariu D, *et al*. Lux0R: detection of malicious PDF-embedded JavaScript code through discriminant analysis of API references[C]//Proc of Workshop on Artificial Intelligent and Security. New York:ACM Press,2014:47-57.
[51] 白鹏,胡影,戴方芳. 基于 shellcode 检测的恶意文档检测[C]//第 19 届全国青年通信学术年会论文集. 2015:141-146.
[52] Nissim N, Cohen A, Glezer C, *et al*. Detection of malicious PDF files and directions for enhancements: a state-of-the art survey[J]. Computers & Security,2015,48(2):246-266.
[53] 黄海新,张路,邓丽. 基于数据挖掘的恶意代码检测综述[J]. 计算机科学,2016,43(7):13-18,56.
[54] Li Yuancheng, Ma Rong, Jiao Runhai. A hybrid malicious code detection method based on deep learning[J]. International Journal of Software Engineering & Its Applications,2015,9(5):205-216.
[55] Wang Yao, Cai Wandong, Wei Pengcheng. A deep learning approach for detecting malicious JavaScript code[J]. Security & Communication Networks,2016,51(8):28656-28667.