基于全局行为特征的未知恶意文档检测

陈 祥[1], 伊 鹏[1], 白 冰[2], 韩伟涛[1]

[1]战略支援部队信息工程大学信息技术研究所 郑州 中国 450002

[2]之江实验室 杭州 中国 311121

摘要 相比于基于宏的恶意办公文档, 基于漏洞利用的恶意办公文档在攻击过程中往往不需要目标交互, 能在目标无感的情况下完成攻击, 已经成为 APT 攻击的重要手段, 因此检测基于漏洞利用特别是未知漏洞利用的恶意文档对于发现 APT 攻击具有重要作用。当前的恶意文档检测方法主要围绕 PDF 文档展开, 分为静态检测和动态检测两类, 静态检测方法容易被攻击者规避, 且无法发现基于远程载荷触发的漏洞利用, 动态检测方法仅考虑 PDF 中 JavaScript 脚本或文档阅读器进程的行为特征, 忽视了针对系统其他进程程序的间接攻击, 存在检测盲区。针对上述问题, 本文分析了恶意办公文档的攻击面, 提出恶意文档威胁模型, 并进一步实现一种基于全局行为特征的未知恶意文档检测方法, 在文档处理过程中提取全系统行为特征, 仅训练良性文档样本形成行为特征库用于恶意文档检测, 并引入敏感行为特征用于降低检测误报率。本文在包含 DOCX、RTF、DOC 三种类型共计 522 个良性文档上进行训练获取行为特征库, 然后在 2088 个良性文档样本和 211 个恶意文档样本上进行了测试, 其中 10 个恶意样本为手动构造用于模拟几种典型的攻击场景。实验结果表明该方法在极低误报率(0.14%)的情况下能够检测出所有的恶意样本, 具备检测利用未知漏洞的恶意文档的能力, 进一步实验表明该方法也能够用于检测针对 WPS Office 软件进行漏洞利用的恶意文档。

关键词 恶意文档检测; 行为特征; 威胁模型; 漏洞利用; 未知威胁

中图法分类号 TP393.08 DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.09.07

Unknown Malicious Document Detection Based on Global Behavior Feature

CHEN Xiang[1], YI Peng[1], BAI Bing[2], HAN Weitao[1]

[1] Institute of Information Technology, PLA Strategic Force Information Engineering University, Zhengzhou 450002, China

[2] ZheJiang Lab, Hangzhou 311121, China

Abstract Compared with malicious office documents based on macros, malicious office documents based on vulnerability exploitation often do not need target interaction in the attack process, and can complete the attack without target perception. It has become an important means of Advanced Persistent Threat (APT) attack. Therefore, detecting malicious documents based on vulnerability exploitation, especially unknown vulnerability exploitation, plays an important role in discovering APT attacks. The current malicious document detection methods mainly focus on PDF documents. It is mainly divided into two categories: static analysis and dynamic analysis. Static analysis is easy to be evaded by hackers, and can not discovery exploits triggered by remote payload. Dynamic analysis only considers the behaviors of the JavaScript in PDF or document reader's process, ignoring the indirect attacks against other processes of the system, leads to a detection blind spot. To solve the above problems, we analyze the attack surface of malicious Office documents, come up with a threat model and implement an unknown malicious document detection method based on global behavior feature. In the process of document processing, the whole system behavior features are extracted, and only benign document samples are trained to form a behavior feature database for malicious document detection. In order to reduce false alarm rate, we introduce sensitive behavioral feature in detection. In this paper, 522 benign documents including DOCX, RTF and DOC are trained to obtain the behavior feature database, and then 2088 benign document samples and 211 malicious document samples are tested. Of these, 10 malicious samples are manually crafted to simulate several typical attack scenarios. The experimental results show that this method can detect all malicious samples with a very low false positive rate (0.14%) and is able to detect malicious documents that exploit unknown vulnerabilities. Further experiments show that this method can also be used to detect malicious documents exploiting WPS office software.

Key words malicious document detection; behavior feature; threat model; vulnerability exploitation; unknown threat

通讯作者: 陈祥, 助理研究员, Email: chenxndsc@163.com。

本课题得到国家自然科学基金(No. 62176264)资助。

收稿日期: 2022-01-11; 修改日期: 2022-04-22; 定稿日期: 2023-06-12

1 引言

电子文档如 Office 文档、PDF 文档等由于其便捷高效的优点, 在人们日常生活和工作中被广泛使用。与此同时, 电子文档带来的安全问题日益凸显, 一方面, 相比于 PE(Portable Execute, PE)类可执行文件, 人们往往倾向于认为电子文档是便捷安全的信息交互载体, 对文档攻击的网络安全防范意识较弱; 另一方面, 由于文档处理软件如 Microsoft Office、Adobe Reader 等功能越来越复杂, 代码的复杂性增加了漏洞存在的可能性, 不断有新的软件漏洞被爆出, 这些都导致恶意文档攻击已经成为网络攻击特别是高级持续性威胁(Advanced Persistent Threat, APT)攻击的重要手段。360 威胁情报中心发布的《2017 年鱼叉攻击邮件研究报告》指出, 攻击者在钓鱼邮件中最常使用的文档类型为 Office 文档, 占比高达 65.4%, 其次为富文本格式(Rich Text Format, RTF)文档, 占比达到 27.3%。卡巴斯基 2018 年年度安全报告显示, Office 和 PDF 应用在软件漏洞利用统计中占比高达 70%, 遥遥领先其他软件[1]。360 发布的《2020 全球高级持续性威胁 APT 研究报告》中披露了多起利用恶意文档针对我国进行的 APT 攻击活动[2]。

针对恶意文档攻击带来的安全威胁, 这些年学术界针对恶意文档检测提出了许多检测方法和技术, 主要分成两大类: 静态检测和动态检测。静态检测通过解析文档的结构或者内容发现文档中的恶意元素从而检测恶意文档, 而动态检测通过文档处理软件实际打开处理文档过程中所触发的行为来判断文档是否恶意。恶意文档静态检测方法虽然在一定程度上提高了文档检测的普适性, 但由于提取的检测特征只是表象特征, 没有触及恶意文档的本质特征, 容易被攻击者混淆绕过, 对模仿攻击的检测能力不足[3]。此外, 一些恶意文档样本可以通过恶意负载远程加载等攻击手法绕过静态检测。例如, 许多利用 Microsoft Office 软件 CVE(Common Vulnerabilities and Exposures, CVE)漏洞的恶意文档本身不包含恶意负载, 需要在文档打开处理过程中从远程服务器获得恶意负载才能进一步触发软件中的漏洞, 而单从文档的内容或者结构上难以发现异常[4-5]。恶意文档动态检测相关研究绝大部分围绕 PDF 文档中的恶意 JavaScript 的动态行为展开, 并不适用于恶意 Office 文档检测。其他少数关注文档处理软件进程系统行为的动态检测方法[6-8]中, 均忽视了以文档为载体的针对其他进程程序实施的间接攻击, 导致存在检测盲区。

针对上述问题, 本文在分析 Office 文档威胁模型的基础上实现了一种基于全局行为特征的未知恶意文档检测方法, 通过学习良性样本的行为特征形成良性行为特征库, 检测时依据待检测文档的行为特征与行为特征库的匹配情况及是否包含敏感行为特征判定文档是否为恶意文档。该检测方法不依赖恶意行为特征, 能够检测逻辑漏洞利用攻击, 能够检测针对 Office 进程之外的其他进程程序的间接攻击, 从而支持对未知恶意文档的检测发现。我们在包含 DOCX、RTF、DOC 三种类型共计 522 个良性文档上进行训练, 然后在 2088 个良性文档样本和 211 个恶意文档样本上进行了测试, 发现模型在设置合适的阈值时, 能够检测出全部的恶意文档, 并且对良性文档样本的误报率仅为 0.14%, 充分表明了检测方法的有效性。

总结来说, 本文的贡献在于:

1) 提出了针对恶意办公文档的威胁模型, 用于指导恶意文档检测技术研究。

2) 针对 Office 软件实现了基于全局行为特征的未知恶意文档检测方法, 在全局行为特征库匹配的基础上进一步引入敏感行为特征检测, 实验表明该方法检测能力强、误报率低, 具备实际应用于捕获未知文档攻击的能力。

3) 针对 WPS Office 软件也开展了相关实验, 在国产化替代的背景下为捕获针对 WPS Office 软件的未知恶意文档攻击提供了可行解决方案。

本文结构如下: 第二部分简述恶意文档检测的研究现状; 第三部分介绍针对 Office 的恶意文档攻击分类和攻击模型; 第四部分介绍本文提出了基于全局行为特征的未知恶意文档检测方法; 第五部分介绍针对本文提出检测方法的实验结果; 第六部分对全文进行总结。

2 相关工作

2.1 静态检测

静态检测方法中, 根据检测模型中提取和使用的文档特征的不同, 可以进一步区分为基于字节序列特征、内容特征、结构特征以及多种特征综合的检测方法。

字节序列特征是将文档看作一个连续的字节序列, 并从中提取检测特征。Li 等人[9]采用统计分析的思路, 分别为良性文档样本和恶意文档样本的 Word 文档按字节流构建 *n*-gram 特征库, 在检测文档时根据文档字节序列分别与良性 *n*-gram 特征库和恶意

n-gram 特征库的相似度判断文档安全性。Liu 等人[10]将文档的字节序列熵作为关注点, 从序列熵信号中提取全局特征(如均值、方差)、能谱特征和局部特征训练机器学习模型用于恶意文档检测。

内容特征是指从文档内容中提取的用于区分良性或者恶意文档的特征, 主要关注 shellcode、PDF 文档中的 JavaScript 和 Office 文档中的 VBA (Visual Basic for Applications, VBA)宏。李伟等[11]提出了一种空间向量检测法检测文档中的 shellcode, 通过将文档中数据片段逐个与已知的 shellcode 进行余弦相似度计算, 发现超过特定阈值就判定文档为恶意。白鹏等人[12]对各类文档进行解析, 获得可能存在 shellcode 的部分然后反汇编, 通过检测是否加栽动态链接库、自定位当前地址、自修改解码行为来决定是否为恶意 shellcode。PJScan[13]通过提取和解析 PDF 文档中 JavaScript 生成词法标记, 并将特定词法标记如变量名、括号、操作符的个数等作为特征进行训练, 并依据恶意样本特征训练得到 OCSVM (One Class Support Vector Machine)模型进行恶意 JavaScript 代码检测。Vatamanu[14]采用类似的方法完成词法标记, 并依据词频特征向量计算两个 PDF 文档之间的距离, 以此为基础研究如何实现准确、高效的文档聚类方法, 将良性/恶意文档划分到合适的类别中。Corona 等人[15]以恶意 JavaScript 代码和良性 JavaScript 代码中应用编程接口(Application Programming Interface, API)访问模式不同为出发点, 通过兼容 JavaScript 的解析器获得 API 访问序列, 并以不同类型 API 出现次数为特征输入评估了多种分类器的有效性。MPScan[16]通过拦截 PDF 阅读器的方式获得去混淆后 JavaScript 源代码和二进制代码, 然后对 JavaScript 源代码进行 shellcode 和堆喷(Heap Spraying)特征检测, 对二进制代码进行恶意操作码签名检测, 综合实现恶意 JavaScript 代码检测。

结构特征是指从文档的层次结构等元数据信息中提取鉴别文档的特征。Srndic 等人[17]基于相似文档具有相似的文档层次结构的认识, 将 PDF 文档元数据信息中包含的各种结构化路径的频率作为区分恶意文档和良性文档的特征, 并评估了支持向量机(Support Vector Machine, SVM)和决策树两种不同分类器的检测效果。Maiorca 等人[18]基于文档结构中的关键字频率以及文件大小、包含间接对象数量、流数量等通用结构特征作为识别恶意 PDF 文档的特征。Cohen 等人[19]采用与 Srndic 等人相同的思路, 通过提取文档中的结构路径特征用于检测恶意 DOCX 文档。Lu 等人[20]通过综合 VBA 函数关键字、OLE (Object Linking and Embedding)文件对象格式、结构路径和文档规范错误四个方面的特征实现恶意 Office 文档检测, 并将检测对象扩展到 PDF、图片等多种文件类型[21]。

2.2 动态检测

动态检测方法中, 大部分研究关注 PDF 文档内部 JavaScript 脚本的动态行为, 只有少数研究工作关注整个文档打开处理过程中的进程动态行为, 我们重点分析该部分研究的不足之处。

MDSCAN[22]提取 PDF 文档中的 JavaScript 代码并在修改后的 SpiderMonkey 解释器中执行, 通过定期扫描解释器的内存空间发现已知的 shellcode 或者操作码序列。PDF Scrutinizer[23]采用类似的方式拦截 Rhino 解释器, 并扫描堆喷、shellcode、受漏洞影响的方法调用等恶意代码模式。ShellOS[24]是一个用来执行 JavaScript 的轻量级操作系统, 它能够记录内存访问模式。在执行过程中, 如果内存访问模式和已知的恶意模式相一致, 如面向返回的编程(Return- Oriented Programming, ROP)关键的系统调用或函数调用等, 则该 JavaScript 被判定为恶意。

Li 等人[9]为了弥补静态检测能力的不足, 在结合静态检测结果的基础上, 进一步通过改变动态链接库(Dynamic Link Library, DLL)加载顺序以检测是否奔溃、是否存在区别于良性操作行为特征库的异常注册表行为、文档打开时弹框检测等措施增强对包含恶意代码的 Word 文档的检测能力, 其问题是考虑的动态行为特征和场景较为单一, 只能作为静态检测的补充。Scofield 等人[6]采用的思路与本文相似, 提取 PDF 文件打开过程中文档阅读器进程的文件操作、注册表操作、进程操作等关键行为特征, 并基于特征的编辑距离进行特征合并, 得到良性行为特征集, 基于行为特征集进行恶意 PDF 文档检测。但其仅关注文档阅读器进程行为忽视了以文档为载体针对其他进程程序的攻击的可能性, 并且采用的基于行为距离的特征合并方法容易被攻击者绕过。Xu 等人[7]基于良性文档在不同系统上表现一致恶意文档攻击在不同系统上表现不同的想法, 通过对比 PDF 文档在不同操作系统的 Adobe 阅读器上的内部和外部行为差异检测恶意文档, 提供了一种不依赖恶意行为先验知识的检测思路, 但其需要针对不同系统上的 PDF 阅读器开发对应的插件以获取内部行为, 不具备文档检测通用性, 另外, 不同系统之间本身存在的行为差异也会对检测结果产生较大的干扰。Jiang 等人[8]通过 cuckoo 沙箱获取办公文档处理过程

中的分析报告, 并从分析报告中提取高频合法字符串作为特征, 并通过大量良性样本和恶意样本训练TextCNN(Text Convolutional Neural Networks)模型用于文档检测并获得较好的检测效果, 但 cuckoo 沙箱分析文档时同样只关注 Office 进程行为, 因此无法发现针对其他进程程序的间接攻击, 且作者采用了大量来自 VirusShare 的恶意文档样本进行训练和测试, 实际上我们发现恶意文档样本绝大部分都是包含恶意宏的文档, 基于漏洞的恶意文档只是极少数, 在基于宏的恶意办公文档上提取特征并训练的模型难以有效检测基于漏洞利用的恶意文档特别是未知恶意文档。

3 恶意 Office 文档攻击类型和攻击模型

3.1 攻击类型

以 Office 办公文档为载体的恶意文档攻击主要包括两种攻击方式, 一种是社会工程学攻击, 利用人们网络安全防范意识薄弱, 攻击者通常需要诱导目标用户配合完成特定操作以达到攻击目的。另一种是漏洞利用攻击, 攻击者利用Office办公软件或者其他系统组件存在的漏洞实施攻击, 往往在用户无感的情况下完成攻击。由于基于漏洞的攻击方式更加隐蔽难以发现, 因此更受到 APT 攻击组织的青睐。

3.1.1 社会工程学攻击

嵌入恶意文件

Office办公文档作为一种复合文档, 允许嵌入各种类型的文件, 如文档、图片、音视频、可执行文件等。攻击者可以将恶意的可执行程序嵌入到文档中, 通过诱导用户点击运行的方式执行恶意程序实施攻击。但用户点击打开恶意软件运行时会出现告警弹框, 因此, 这种攻击方式容易被用户发现, 攻击的成功率不高。

恶意 VBA 宏

VBA 宏[25]是 Office 自带的一种高级脚本特性, 可以在 Office 中去完成某项特定的任务, 而不必再重复相同的动作, 目的是让用户文档中的一些任务自动化。与 PDF 文档中 JavaScript 不同, Office 宏中的 VBA 脚本功能强大, 能够支持远程下载、本地执行、注册表修改等多种类型操作, 攻击者也往往通过在文档中嵌入恶意宏实施攻击, 并采用各种混淆手段隐藏其攻击手法和意图[26], 防止被杀毒软件检测发现。

在 Microsoft Office 默认的宏安全设置下, 当打开带有宏的文档时, 宏不会自动运行, 需要用户决定是否运行宏。由于利用宏实施攻击便捷、稳定, 普通用户对宏的安全性认识不足, 基于宏的恶意文档攻击已经成为恶意文档攻击中广泛使用的攻击手段。

恶意 DDE 命令

动态数据交换(Dynamic Data Exchange, DDE)[27]主要是微软用来允许两个正在使用的应用程序共享数据的方式, 由于 DDE 本身是 Microsoft Office 系列软件的合法功能, 所以绝大多数安全软件不会阻止 DDE 字段文档运行。攻击者可以创建包含恶意 DDE 字段的 Word 文档, 用户打开文档并允许运行 DDE 功能后, 文档里内嵌的代码会运行, 如下载远程恶意代码并执行。例如迈克菲研究人员已经监测到借助 DDE 功能感染并下载的新网络间谍软件的攻击[28]。

3.1.2 漏洞利用攻击

由于 Microsoft Office 软件设计实现复杂, 会与系统中的许多功能组件交互, 且文档中嵌入的各种类型的对象也需要由相应的功能组件初始化。当 Office 软件本身或者其依赖的其他组件存在漏洞时, 用户打开攻击者精心制作的恶意文档就会触发对Office软件或相关组件的漏洞利用, 在用户无感的情况下实现恶意代码执行, 达成攻击目的。

我们梳理了 2017 年以来披露出的和 Microsoft Office 软件相关的重要漏洞, 如下表 1 所示。可以看出, 漏洞以逻辑漏洞和内存破坏漏洞为主, 并且涉及 Office 软件之外的多个系统功能组件。

表 1 2017 年以来披露出的和 Microsoft Office 软件相关的重要漏洞

Table 1 Important vulnerabilities related to Microsoft office software disclosed since 2017

漏洞类型	漏洞位置	漏洞 CVE 编号	说明
逻辑漏洞	Office 软件	CVE-2017-0199	用 URL Moniker 加载本地或远程 HTA 文件执行
逻辑漏洞	Office 软件	CVE-2017-8570	用 CompositeMoniker、FileMoniker、NewMoniker、scriptletfile 加载本地或远程脚本执行
逻辑漏洞	.NET 组件	CVE-2017-8759	利用.Net 组件加载解析远程 SOAP 配置文件生成 dll, 实现远程代码执行
类型混淆	Office 软件	CVE-2017-11826	没有正确地验证标签对象是否闭合造成类型混淆, 可实现任意代码执行
内存破坏漏洞	公式编辑器组件	CVE-2017-11882	公式编辑器栈缓冲区溢出

续表

漏洞类型	漏洞位置	漏洞 CVE 编号	说明
内存破坏漏洞	公式编辑器组件	CVE-2018-0802	公式编辑器栈缓冲区溢出
内存破坏漏洞	公式编辑器组件	CVE-2018-0798	公式编辑器栈缓冲区溢出
内存破坏漏洞	VBScript 组件	CVE-2018-8174	VBScript 脚本执行引擎存在 UAF 漏洞, 在文档中加载远程 html 文件利用漏洞, 可实现任意代码执行
内存破坏漏洞	JavaScript 组件	CVE-2020-0674	JScript 脚本执行引擎存在 UAF 漏洞, 在文档中加载远程 html 文件利用漏洞, 可实现任意代码执行
逻辑漏洞	MSHTML 组件	CVE-2021-40444	MSHTML 组件存在路径遍历漏洞, 在文档中加载远程 html 文件利用漏洞后可实现 dll 加载执行

逻辑漏洞往往是因为在设计软件时逻辑不严密导致的, 在漏洞利用上也不具备通用的攻击手法或特征, 因此难以检测和防范。对于内存破坏漏洞, 根据具体的漏洞类型及目标采取的安全防护措施, 攻击者往往采用不同的利用技术, 如针对栈缓冲区溢出漏洞通常采取返回地址修改、异常链劫持等手段实现控制流劫持; 针对堆内存破坏漏洞攻击者通常采用 ROP、堆喷等方式, 通过修改返回地址、函数指针、虚表指针等实现控制流劫持[29]。

3.2 攻击模型

本文所要检测的恶意文档是指基于漏洞利用攻击的恶意文档, 3.1 部分介绍的基于社会工程学手段实施攻击的恶意文档不在本文提出检测方法的检测范围内。

在对近年来 Office 软件相关漏洞利用分析的基础上, 我们发现: 1)恶意文档的攻击面广, 不仅仅局限于 Office 程序本身, 还包括所依赖的系统功能组件、程序。如 CVE-2018-8174, 攻击者在文档中加载远程 html 文件时利用系统 VBScript 脚本执行引擎存在的释放后重用(Use After Free, UAF)漏洞, 实现任意代码执行。2)攻击的上下文环境不仅仅局限于 Office 进程, 还包括系统服务进程等其他进程。如 CVE-2017-11882、CVE-2018-0802, 攻击者通过在文档中插入恶意公式, 使系统启动公式编辑器服务程序处理公式对象, 并利用公式编辑器服务程序中存在的栈缓冲区溢出漏洞实施攻击。理论上, 只要 Office 进程与其他进程存在数据交互, 就为针对其他进程程序的间接攻击提供了通道, 扩展了恶意文档的攻击表面。3)漏洞类型多样, 既有大量内存破坏漏洞, 也有不少逻辑漏洞。

Li 等人[30]针对 OLE 对象分析了 Microsoft Office 软件的攻击表面和可行的攻击向量, 这里, 我们从更一般的角度来分析恶意文档的攻击模型。上图 1 显示了基于漏洞利用的恶意文档攻击模型, 标识了漏洞利用的对象和所处的上下文环境。从攻击者的角度看, 攻击者可以通过构造恶意文档, 利用 Office 程序或者其加载的系统组件中的逻辑漏洞或者内存破坏漏洞, 在 Office 进程上下文环境下实施漏洞利用攻击; 也可以利用与 Office 进程存在数据交互的服务进程程序/组件中的逻辑漏洞或内存破坏漏洞, 在服务进程上下文环境下实施漏洞利用攻击。对于防御者而言, 在设计恶意文档检测方法时, 必须考虑到攻击者所有的攻击路径。

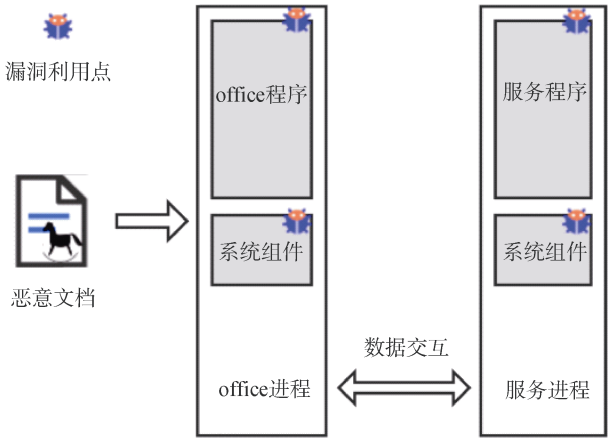

图 1 基于漏洞利用的恶意文档攻击模型

Figure 1 Malicious document attack model based on vulnerability exploitation

基于上述模型, 综合考虑上面三个方面的威胁因素, 我们提出了基于全局行为特征的未知恶意文档检测方法, 检测恶意文档对文档处理进程之外其他进程程序的间接攻击, 以增强检测能力。

4 基于全局行为特征的恶意文档检测

这一部分将详细描述本文提出和实现的基于全局行为特征的未知恶意文档检测方法。

下图 2 描述了基于全局行为特征的未知恶意文档检测方法的工作流程, 分为两个阶段: 训练阶段和测试阶段。在训练阶段, 仅采用良性文档作为训练样本, 经过全局行为特征提取和预处理后形成行为

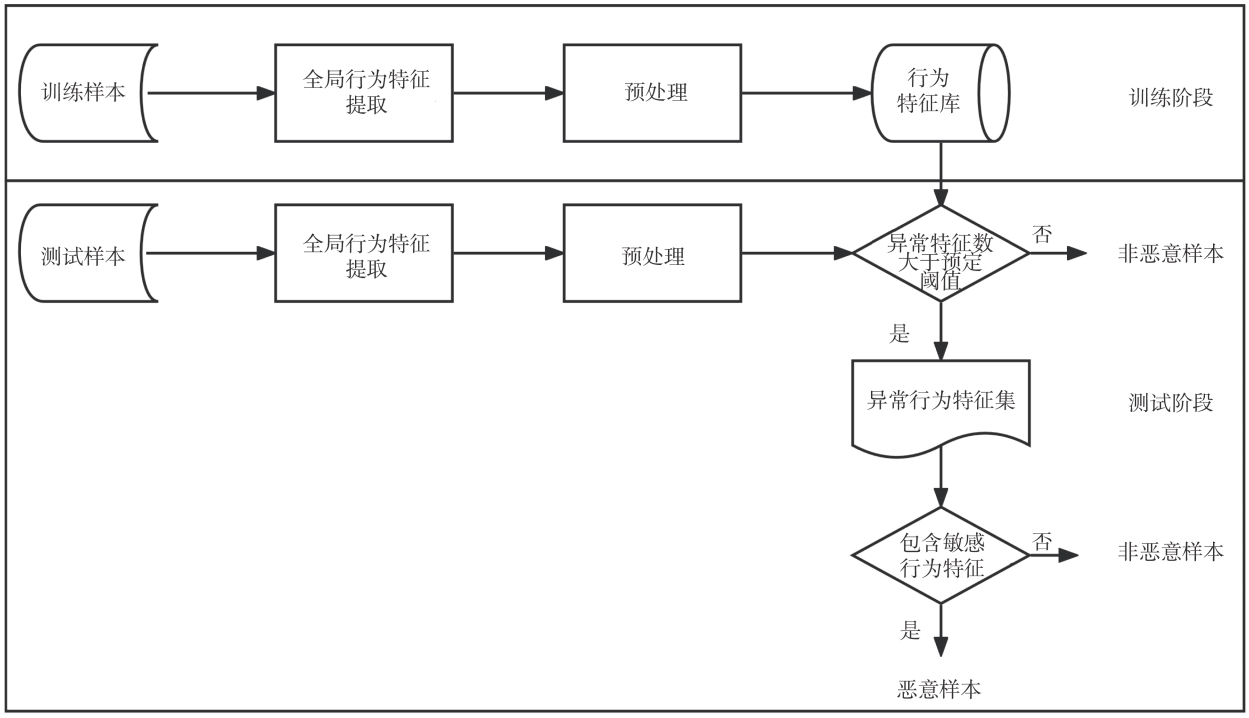

图 2　恶意文档检测工作流程

Figure 2　Workflow of malicious document detection

特征库; 在测试阶段, 依据待检测文档的行为特征与行为特征库的匹配情况及是否包含敏感行为特征, 来判定文档是恶意文档还是良性文档。

4.1　全局行为特征提取

在本文前面的威胁模型分析可知, 仅仅针对目标进程及其子进程的行为监控无法发现间接利用其他进程漏洞的攻击, 因此需要监测样本处理过程中系统所有进程的行为。

理论上, 系统调用作为操作系统为用户态程序提供的接口, 能够较好的反应程序在系统上的各类行为操作, 因此, 在系统调用层面抓取进程行为是一个非常合适的切入点。但由于 windows 系统中很多系统调用并不公开, 并且样本处理过程中采集全系统的系统调用数据量大, 处理时间开销和存储开销大, 不利于后续的进一步分析和处理, 因此我们将在 windows 系统 DLL 提供给开发人员的应用程序接口(API)处拦截并观测进程行为。行为特征用三元组表示如下:

<进程名, 操作名称, 操作对象>

进程名表示执行 API 操作的进程名称, 操作名称表示API操作的类型, 通常系统中提供的API操作类型较多, 我们仅需关注攻击者实施有效攻击所涉及的各类 API 操作, 包括文件类操作, 如打开文件、读写文件、删除文件等, 注册表操作, 如打开注册表、创建注册表, 读写注册表键值等, 网络类操作, 如创建连接, 发送/接收数据包, 关闭连接等, 进程类操作, 如创建进程、创建线程等, 此外还包括一些特殊操作类型, 如文件系统控制、设备 IO 控制、写回文件缓存等。操作对象是指该进程 API 操作的目标对象, 如文件、注册表键值、可执行文件镜像。下表 2 例举了一些典型的行为特征。

4.2　预处理

在获得全局行为特征后, 首先筛选出具有代表性的行为特征, 然后需要对行为特征中的操作对象进行归一化处理, 避免系统随机性带来特征抖动。

表 2　行为特征举例

Table 2　Examples of behavior features

序号	行为特征	说明
1	winword.exe, CreateFileMapping, C:\Windows\System32\wtsapi32.dll	创建文件映射
2	winword.exe, RegSetValue, HKCU\Software\Microsoft\Office\15.0\Word\MTTT	设置注册表值
3	winword.exe, Connect, test-PC:53775 -> 52.109.112.104:https	建立 TCP 连接
4	svchost.exe, CreateProcessA , C:\Windows\system32\DllHost.exe	创建新进程

1) 操作类型过滤

为了减少行为特征的规模同时不降低对恶意行为操作的观测能力, 我们需要在所有的操作类型中过滤掉不实质改变系统状态的操作, 以查询类操作为主, 如查询目录(QueryDirectory)、查询注册表键值(RegQueryValue)等。实验过程中发现该步骤能够明显缩减特征数目。

2) 操作对象归一化

文件

在处理样本时进程通常会在临时目录下释放一些临时文件, 这些临时文件按照某一格式采用随机数命名, 为了降低不同样本之间系统随机性带来的影响, 需要对临时文件名称进行归一化处理, 如以下两个文件类行为特征:

winword.exe, CreateFile, C:\Users\cx\AppData\Local\Temp\mso3C75.tmp

winword.exe, CreateFile, C:\Users\cx\AppData\Local\Temp\CVR2D12.tmp

可以按照正则表达式的方式将文件名称归一化为:

winword.exe, CreateFile, C:\Users\cx\AppData\Local\Temp\[a-zA-Z0-9]{7}.tmp

注册表

同上面针对临时文件的预处理类似, 一些针对临时注册表项的行为操作也需要屏蔽随机性带来的干扰。如以下两个注册表类行为特征:

winword.exe, RegCreateKey, HKCU\Software\Microsoft\Office\15.0\Word\Resiliency\DocumentRecovery\E2E3234

winword.exe, RegCreateKey, HKCU\Software\Microsoft\Office\15.0\Word\Resiliency\DocumentRecovery\E2E3DBD

可以归一化为:

winword.exe, RegCreateKey, HKCU\Software\Microsoft\Office\15.0\Word\Resiliency\DocumentRecovery\[a-zA-Z0-9]{7}$

网络操作

在处理样本时进程可能会有外部网络访问请求, 如获取远程模板文件、访问远程服务器以检查更新等, 为了防止办公软件产商采用 CDN(Content DeliveryNetwork)服务导致访问域名的外部 IP 地址发生变化, 需要将网络行为特征中的外部 IP 地址转化为域名; 另外, 主机本地打开端口属于临时使用的动态端口, 不属于行为特征的关键要素, 可以忽略。例如, 以下网络行为特征:

winword.exe, Connect, test-PC:53775 -> 52.109.124.116:https

可以标准化为:

winword.exe, Connect, test-PC -> office15client.microsoft.com:https

4.3 分类

基于良性行为特征库进行恶意文档检测分类始终面临一个问题: 对于检测中发现的不在特征库中的异常行为, 既可能是漏洞利用成功后攻击者实施的恶意行为, 也可能源于训练阶段的样本集不够全面, 待检测文档触发 Office 软件新的程序执行路径导致新行为的出现。如果分类时简单地认为出现异常行为就判定为恶意文档则会产生较多的误报。

为了降低误报的出现, 我们在文档检测时引入判定阈值, 以容忍少量异常行为的出现, 低于该阈值则认为是良性文档。事实上, 即使异常行为数超过了给定阈值, 也可能并不包含恶意行为。针对这种情况, 我们引入敏感特征的概念, 对于前面提到的文件、注册表、网络、进程等类型行为操作, 如果某种操作有可能影响系统的安全属性, 包括机密性、完整性和可用性, 则认为该操作为敏感操作, 对应的行为特征为敏感特征。例如, 读文件和获取文件属性操作是非敏感操作, 写文件和修改文件属性操作为敏感操作。我们通过对所有监测的 API 操作进行分析得到常用的敏感操作类型如下表 3 所示。

当被检测文档的异常行为中不含有敏感操作时, 即使异常行为数超过了判定阈值, 也应该认为该文档为良性文档。例如, 如果异常行为中仅仅包含读文件的行为, 由于读文件的操作并不损害系统的安全属性, 可以认为异常行为中没有恶意行为; 如果异常行为中不仅仅包含读文件的行为, 还有发送网络数据包的行为, 则有可能是系统在向外泄露敏感数据, 应该认为异常行为中包含可疑恶意行为。在现实攻击中, 攻击者也可能结合表中多种敏感行为操作实施攻击。引入敏感操作检测的目的是为了进一步降低误报, 对于没有对系统安全属性产生实质性影响的文档我们可以认为其是良性文档。

这里需要说明为什么不在预处理阶段过滤掉所有的非敏感操作, 只保留敏感操作行为用于后续检测? 这是因为在真实的攻击场景中, 每个敏感操作行为都会伴随大量的非敏感操作行为, 例如在向外发送网络数据包泄露敏感数据前, 需要打开和读取对应的敏感文件, 这样就会“放大”异常行为, 使得恶意行为更容易被凸显出来, 增加检测模型对恶意文档与良性文档的区分度。

在测试阶段对文档分类具体流程如下: 首先将样本的行为特征与已有的行为特征库进行匹配, 若不匹配的特征数小于给定阈值 T, 则认为该文档为良

表 3 敏感操作类型

Table 3 Sensitive operation type

序号	常用敏感操作类型	举例说明
1	增加、修改、删除注册表项	攻击者通过增加或修改自启动程序注册表项实现持久化, 破坏系统的完整性; 攻击者通过删除系统关键注册表破坏系统的可用性。
2	写文件、删除文件、修改文件属性	攻击者通过写文件在系统目录下释放恶意 dll 文件实现 dll 劫持破坏系统完整性; 攻击者删除用户关键文件破坏系统可用性。
3	发送网络数据包	攻击者读取系统敏感信息后通过网络数据包向外发送数据, 破坏系统的机密性。

性文档; 若不匹配的特征数大于给定阈值 T, 则进一步判断不匹配的行为特征集中是否包含敏感特征, 如包含其中任意 1 种类型, 则认为文档为恶意文档, 反之认为与是良性文档。

5 实验与结果

在本部分, 我们将会评估本文所提出检测方法的效果。

5.1 实验环境

实验环境如下表所示。为了确保能够得到尽量多的针对 Office 软件漏洞攻击生效的恶意样本, 我们选择的 Microsoft Office 版本为 2013 专业版。在实验时为了避免文档中的宏执行带来的影响, 我们将 Microsoft Office 软件的宏安全性设置为默认禁用所有宏。在行为特征抓取上选择 Process Monitor 工具, 抓取系统中除了该工具进程之外所有进程在文件、注册表、网络、进程四个方面共计 59 种行为操作。

考虑到通常攻击者在恶意文档打开后立即实现漏洞利用并触发恶意动作, 实验时我们设置默认行为抓取时间为文件打开后 10 秒。虽然攻击者可以延迟一段时间再触发恶意动作, 但由于用户也会关闭打开的文档提前结束文档进程从而导致攻击失效。

表 4 实验环境

Table 4 Experimental environment

宿主机	Huawei 2488V5 服务器 CPU: Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz X2 内存: 64 GB 操作系统: CentOS 8
虚拟机	操作系统: Win7 X64 SP1 内存: 8 GB Office 版本: Microsoft Office 专业版 2013 行为抓取工具: Process Monitor V3.5.3

5.2 实验数据集

由于针对 Office 的恶意文档检测并没有公开可用的数据集, 我们从 Bing 搜索引擎上检索并获取了 1044 个办公文档, 包括 RTF、DOC、DOCX 三种文档类型, 我们随机将其中一半文档划分为训练集 TD0, 另一半划分为测试集 TD1。样本中各种类型文档数量见表 5。为了进一步测试检测方法对正常文档的误报情况, 我们扩大了良性样本的测试数量, 从 Baidu 文库单独获取了 1566 个 DOC 类型的办公文档作为测试集 TD2。需要说明的是, 获取的上述样本经过某杀毒软件检测和某安全产商的沙箱检测, 均未标记为恶意文档, 我们以此为依据将其认定为良性文档样本。

表 5 办公文档数据集

Table 5 Office document dataset

数据集类型	训练集	测试集		
数据集标签	TD0 (良性样本)	TD1 (良性样本)	TD2 (良性样本)	TD3 (恶意样本)
包含文档类型与数量	DOCX 147, RTF 190, DOC 180	DOCX 159, RTF 197, DOC 166	DOC 1566	RTF 90, DOCX 88, DOC 33
总计	522	522	1566	211

由于当前公开可用的恶意文档集主要是基于宏的恶意文档, 基于漏洞利用的恶意文档极少, 因此我们自行构建了恶意样本测试集。恶意样本测试集 TD3 包含 211 个恶意文档样本, 其中 10 个恶意样本为手动构造, 从攻击模型的角度来模拟三种典型的攻击场景, 以评估检测方法对不同攻击场景的检测能力, 包括: 1)在 Office 进程上下文环境下针对 Office 程序的漏洞利用攻击, 2)在 Office 进程上下文环境下针对系统组件的漏洞利用攻击, 3)服务进程上下文环境下针对服务程序的漏洞利用攻击, 具体如下表 6 所示。需要重点指出的是, 其中一些攻击场景只触发了漏洞甚至没有实际的恶意操作行为, 用来模拟具有高度隐蔽性的攻击手法。另外 201 个漏洞利用恶意文档样本来自 Virustotal, 用于检验模型对实际恶意文档的检测能力。

5.3 实验评估指标

本文采用检测率 (Detection Rate, DR) 和误报率 (False Alarm Rate, FAR)作为实验评估指标。检测率能够反应检测方法对恶意文档样本的检测能力,

表 6 构建的不同攻击场景

Table 6 Types of crafted attack

CVE 编号	利用进程上下文	漏洞位置	恶意行为	数量
CVE-2017-8570	Office 进程	Office 程序	在临时文件夹下释放可执行文件并执行	3
CVE-2017-8570	Office 进程	Office 程序	修改注册表实现系统启动后从远程服务器下载恶意脚本执行	1
CVE-2017-8570	Office 进程	Office 程序	触发漏洞但不执行恶意操作	1
CVE-2017-11882	公式编辑器进程	公式编辑器组件	触发漏洞但不执行恶意操作	1
CVE-2017-11882	公式编辑器进程	公式编辑器组件	访问远程服务器	1
CVE-2017-11882	公式编辑器进程	公式编辑器组件	执行 shellcode 启动本地程序	1
CVE-2018-8174	Office 进程	VBScript 组件	获取远程 html 页面, 进程奔溃	1

误报率能够反应检测方法对良性文档样本的误报情况, 只有同时满足检测率高、误报率低的检测方法才具有实际应用价值。

$$DR = \frac{TR}{TP + FN}$$

$$FAR = \frac{FP}{TN + FP}$$

其中, *TP* (True Positive)是指将异常样本正确分类为异常样本的数量, *TN* (True Negative)是指将正常样本正确分类为正常样本的数量, *FP* (False Positive)是指将正常样本错误分类为异常样本的数量, *FN* (False Negative)是指将异常样本错误分类为正常样本的数量。

5.4 实验结果

在良性样本训练阶段, 我们评估了学习到的行为特征库的数目随训练样本数的变化情况, 如下图 3 所示。可以看出, 当训练的样本数量达到 218 以后行为特征库中的特征数接近饱和, 说明对良性样本行为特征的学习已经达到稳定状态, 最终特征库仅包含 3417 条特征。当学习的样本会触发大量不同于已经学习到的行为特征时, 便会看到图中出现的特征库规模突然急剧增加的情况。分析发现图中特征库数目饱和前的急剧增加来源于样本文档加载了 C:\Windows\Fonts\目录下多种不同类型的字体文件。

在完成训练生成良性行为特征库后, 我们将检测阈值设置为 50, 用于对良性样本和恶意样本进行检测。

首先基于该特征库对良性样本测试集 TD1 进行检测, 检验已有的特征库是否能够覆盖新的良性文档产生的行为特征, 实验结果如下图 4 所示。为了表述方便, 我们称不在特征库中的行为特征为异常特征, 可以看出, 绝大部分良性样本的异常特征数为 0, 在所有 522 个良性测试样本中只有 2 个样本的异常特征数超过设置的阈值, 由于样本异常特征中包含敏感特征, 文档被误判为恶意文档, 因此, 检测模型在 TD1 上的误报率为 0.38%。实验充分表明正常文档对办公软件功能特性的使用往往比较集中, 训练得到的行为特征库已经能够涵盖文档常用功能所涉及的行为特征。

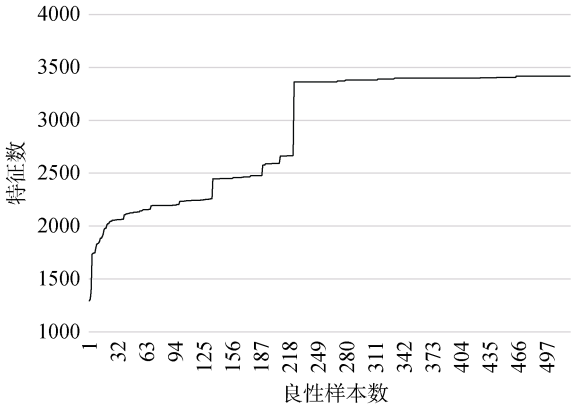

图 3 训练阶段行为特征数增长情况

Figure 3 Growth of behavioral feature in training stage

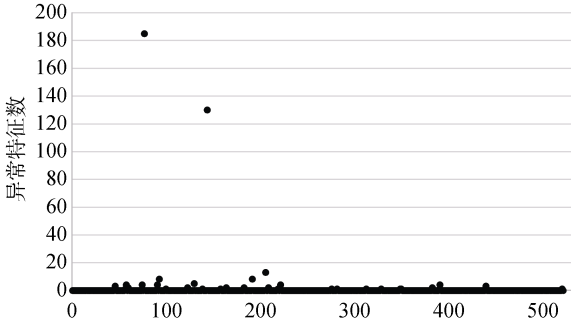

图 4 TD1 产生的异常特征数

Figure 4 Abnormal feature numbers of TD1

随后, 我们对恶意样本测试集 TD3 进行了测试, 以获得恶意样本产生的异常特征情况。实验结果如下图 5 所示, 图中对我们构造的恶意文档样本和从 Virustotal 获取的恶意文档样本进行了区分。另外, 为了更直观显示异常特征数较小的样本, 对于异常特征数超过 500 的样本在图中仅标记为 500。

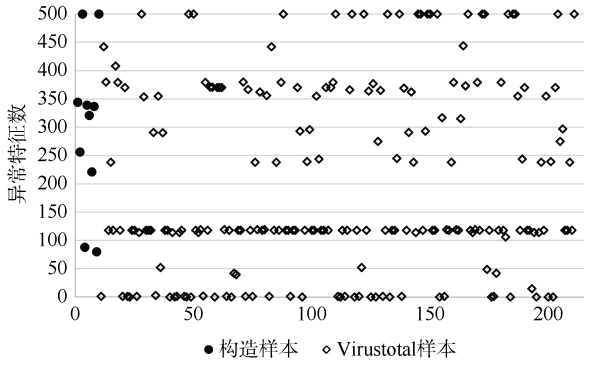

图 5 TD3 产生的异常特征数

Figure 5 Abnormal feature numbers of TD3

从实验结果可以看出, 我们构造的模拟不同攻击场景的 10 个恶意文档产生的异常特征数均超过阈值, 被成功检测出。构造的恶意文档中部分样本触发漏洞后并没有实际的恶意行为, 但仍然产生了大量异常特征, 我们对其异常特征进一步分析, 发现恶意文档利用过程中会在本进程或其他新进程中打开、加载多种 dll 文件, 并引起一些系统进程出现新的文件、注册表操作, 导致异常特征数较大。例如, 在利用逻辑漏洞 CVE-2017-8570 的恶意样本中, 为了执行在临时文件夹下释放的 vb 脚本文件, winword 进程需要先后打开、读取、加载 comsvcs.dll、scrobj.dll、vbscript.dll 等多个 dll 类型文件。又例如, 在利用内存漏洞 CVE-2018-8174 的恶意样本中, 由于漏洞利用实现了 winword 进程的控制流劫持, 导致进程奔溃, 从而启动了 DW20.exe、dwwin.exe 错误报告进程并产生大量不同于行为特征库的行为操作。

对于来自 Virustotal 的 201 个恶意文档, 模型能够检测发现其中 157 个恶意文档。由于我们考虑了全系统的行为操作, 对于恶意样本测试集 TD3 中大量利用 CVE-2017-11882 漏洞的恶意样本, 我们也能够发现公式编辑器进程上的大量异常行为特征, 这在传统的仅关注 Office 进程行为的动态检测方法中是无法发现的。

我们对这些未被检测出的 44 个恶意文档进行进一步分析发现, 其中 21 个文档被 Virustotal 标记为比我们实验采用的 Office 版本更早的 CVE 编号, 包括 CVE-2012-0158、CVE-2010-3333 和 CVE-2002-1623, 4 个文档被标记为宏, 由于我们禁用宏的设置, 导致这些文档在检测环境中并不能成功利用漏洞或者执行宏, 没有产生异常行为特征。对于剩下的 19 个恶意文档, 其中只有 1 个文档表现出了恶意行为, 该恶意文档被 Virustotal 标记为 CVE-2017-0199, 我们检测到 winword.exe 进程试图连接 192.158.2.178 地址 8080 端口的异常行为, 但由于连接失败导致无法下载恶意的 hta 脚本执行, 因此没有进一步的恶意行为特征。其余的恶意文档中 2 个文档打开失败, 3 个文档未被正确解析, 13 个文档未能在检测环境下表现出恶意行为, 我们猜测可能是恶意文档缺少成功利用所需的环境或者文档本身未被正确构造。

为了进一步验证基于良性行为特征库的检测方法带来的误报情况, 我们增加了对测试集 TD2 的测试(测试集 TD2:训练集 TD0=3:1), 实验结果如下图 6 所示。TD2 所有的样本中仅有 2 个样本产生的异常特征数超过特征检测阈值, 并且其中 1 个样本(图中已标记)的异常行为不含前面定义的敏感操作, 仅包含打开文件、读取文件和加载文件映射操作类型, 根据前面的检测流程, 该文档仍被判定为良性文档。因此, 在测试集 TD2 上仅有 1 个良性文档被检测为恶意文档, 误报率不足 0.1%。

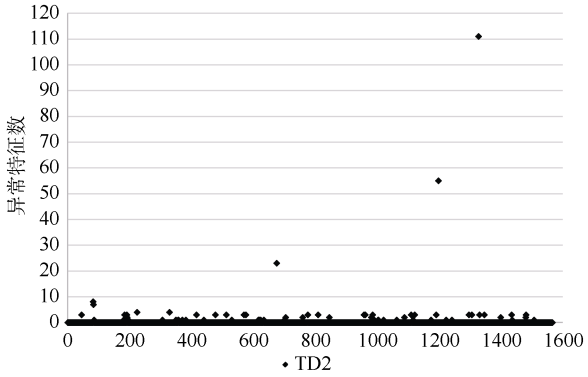

图 6 TD2 产生的异常特征数

Figure 6 Abnormal feature numbers of TD2

在上述测试完成后, 我们修改检测阈值, 综合良性样本集 TD1 和 TD2 的测试数据, 得到本文所采用的检测方法在其他不同的特征检测阈值下的检测率和误报率。由于检测模型关注的是行为特征, 在计算检测率时我们仅将 Virustotal 样本中实际表现出恶意行为的文档样本作为真实恶意样本, TD3 中最终共有 158 个有效恶意样本。我们选取了 5 个不同的特征检测阈值, 得到的检测结果如下表所示。可以看到, 当特征检测阈值设置为 40 时, 恶意文档的检测率为 100%, 此时良性样本的误报率仅为 0.14%。特征检测阈值越小, 此检测方法对未知的恶意样本检测越灵敏, 但良性样本误报的概率也会增加, 需要根据实际的应用场景去选择合适的特征检测阈值。

由于本文属于专门针对基于漏洞利用的 Office 文档检测进行研究, 与针对宏的恶意办公文档检测

表 7 检测结果

Table 7 Detection Result (%)

检测阈值	20	40	60	80	100
DR	100	100	98.1	98.1	96.8
FAR	0.19	0.14	0.14	0.14	0.14

我们进一步对文档检测的时间开销进行了分析, 实验发现将文档在虚拟机环境中的处理时间设置为 10s 时, 文档检测的平均时间开销为 22.88s, 其中动态分析开销 22.62s, 检测开销 0.26s, 具体如下表 8 所示。动态分析开销包括虚拟机快照恢复、待检测文档上传、文档在虚拟机中处理以及抓取的行为数据回传等产生的时间开销, 检测开销主要包括行为数据文件的预处理及特征库匹配开销, 可以看出, 动态分析开销才是影响文档检测速度的主要原因。

表 8 时间开销

Table 8 Time Consumption

	动态分析开销(s)	检测开销(s)
平均值	22.62	0.26
标准差	0.49	0.03

上述实验表明了基于良性行为特征的异常检测方法在检测针对 Microsoft Office 软件漏洞的恶意文档攻击具有很好的效果, 但是该方式是否也能够用于检测针对其他办公应用软件的恶意文档攻击呢?

为此, 我们进一步研究了国产办公软件 WPS Office 2016 在上述训练集 TD0 和测试集 TD1、TD2 上的实验情况, 分别如下图 7 和图 8 所示。实验可以看出, 通过少量的良性样本行为特征学习就能使行为特征集达到饱和状态, 并且以此行为特征集为基础进行大量良性样本测试, 只是极少数样本产生较多的异常特征, 绝大部分的样本行为特征与已有特征库相一致。由于难以获得针对 WPS Office 2016 的恶意样本进行测试, 通过前面的分析可以发现恶意样本产生的异常特征数与其要实现的恶意行为或者攻击手法强相关, 因此, 有理由认为针对 WPS Office 的恶意样本攻击同样会产生较多的异常特征, 这表明该检测方法针对办公文档类应用的恶意文档攻击检测具有一定的通用性。

5.5 讨论

虽然本文提出的基于全局行为特征的检测方法仅仅针对恶意文档检测进行了实验, 但该方法实质上是一种通用的异常检测方法, 只要能够通过适当的训练样本较好地学习到程序的良性行为特征, 实现行为特征库对多数常见样本行为的覆盖, 就可以用该方法开展入侵检测。当然, 由于监测系统全局行为特征会产生一定的时间和内存开销, 因此这种检测方式在应用场景上有一定的限制, 更适用于离线检测的场景, 在一些资源有限或者检测实时性要求较高的场景下适用性有限。

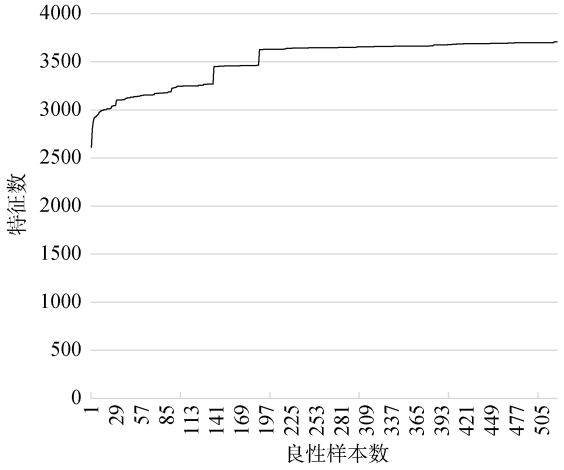

图 7 训练阶段行为特征数增长情况(WPS Office)

Figure 7 Growth of behavioral feature in training stage(WPS Office)

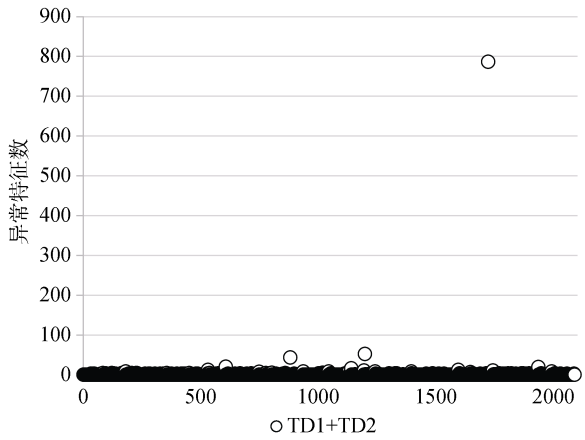

图 8 TD2 产生的异常特征数(WPS Office)

Figure 8 Abnormal feature numbers of TD2(WPS Office)

6 总结与展望

基于良性行为特征库进行恶意文档检测是一种朴素的检测方法, 但此前研究者仅关注文档阅读器进程及其子进程的行为, 未考虑利用文档进行间接攻击的可能性。我们在分析 Office 办公文档攻击模型的基础上, 提出了基于全局行为特征的未知恶意文档检测方法, 实验结果表明该方法在极低误报率(0.14%)的情况下能够发现各种不同攻击手法的恶意文档攻击。由于我们只基于良性样本构建行为特征库, 因此该方法能够检测包括逻辑漏洞在内的基于

0-day 漏洞的恶意文档攻击, 在钓鱼邮件攻击、隔离网络渗透等多种场景捕获基于文档的 0-day 攻击具有重要的应用价值。

在实验过程中我们也发现一些良性办公文档分别在 Microsoft Office 软件和 WPS Office 软件处理时表现出一致的异常特征, 考虑到 Microsoft Office 软件和 WPS Office 软件拥有公共漏洞的概率极低, 恶意文档很难在不同的软件上同时利用生效, 下一步我们将探索结合软件的异构性去消减检测过程中良性的异常特征, 进一步降低良性文档的误报率。

参考文献

[1] Kaspersky Security Bulletin 2018 STATISTICS.https:. go. kaspersky.com/rs/802-IJN-240/images/KSB_statistics_2018_eng_final.pdf. 2019.

[2] 2020 全球高级持续性威胁 APT 研究报告. http:. pub1-bjyt.s3.360.cn/bcms/2020 全球高级持续性威胁 APT 研究报告.pdf.2021.

[3] Yu M, Jiang J G, Li G, et al. A Survey of Research on Malicious Document Detection[J]. *Journal of Cyber Security*, 2021, 6(3): 54-76.
(喻民, 姜建国, 李罡, 等. 恶意文档检测研究综述[J]. *信息安全学报*, 2021, 6(3): 54-76.)

[4] CVE-2021-40444 漏洞深入分析. https:. cloud.tencent.com/ developer/article/1883461. 2021.

[5] CVE-2018-8174 双杀漏洞分析复现及防御. https:. www.freebuf.com/vuls/224379.html.2020.

[6] Scofield D, Miles C, Kuhn S. Fast Model Learning for the Detection of Malicious Digital Documents[C]. *The 7th Software Security, Protection, and Reverse Engineering / Software Security and Protection Workshop*, 2017: 1-8.

[7] Xu M, Kim T. PLATPAL: Detecting Malicious Documents with Platform Diversity[C]. *The 26th USENIX Conference on Security Symposium*, 2017: 271-287.

[8] Jiang J G, Wang C H, Yu M, et al. NFDD: A Dynamic Malicious Document Detection Method without Manual Feature Dictionary[C]. *International Conference on Wireless Algorithms, Systems, and Applications*, 2021: 147-159.

[9] Li W J, Stolfo S, Stavrou A, et al. A Study of Malcode-Bearing Documents[M]. Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 231-250.

[10] Liu L P, He X H, Liu L, et al. Capturing the Symptoms of Malicious Code in Electronic Documents by File's Entropy Signal Combined with Machine Learning[J]. *Applied Soft Computing*, 2019, 82: 105598.

[11] Li W, Su P R, Shi Y F. A Technique for Detecting Malicious Documents Based on Calculation of Vector Spaces[J]. *Journal of the Graduate School of the Chinese Academy of Sciences*, 2010, 27(2): 267-274.
(李伟, 苏璞睿, 时云峰. 基于空间向量计算的恶意文档检测技术[J]. *中国科学院研究生院学报*, 2010, 27(2): 267-274.)

[12] Bai Peng, Hu Ying, Dai Fangfang. Malicious Document Detection Based on Shellcode Detection[C]. *The 19th National Youth Communication Academic Annual Conference*. 2014.
(白鹏, 胡影, 戴方芳. 基于 shellcode 检测的恶意文档检测[C]. *第十九届全国青年通信学术年会论文集*. 2014.)

[13] Laskov P, Šrndić N. Static Detection of Malicious JavaScript-Bearing PDF Documents[C]. *The 27th Annual Computer Security Applications Conference*, 2011: 373-382.

[14] Vatamanu C, Gavriluț D, Benchea R. A Practical Approach on Clustering Malicious PDF Documents[J]. *Journal in Computer Virology*, 2012, 8(4): 151-163.

[15] Corona I, Maiorca D, Ariu D, et al. Lux0R: Detection of Malicious PDF-Embedded JavaScript Code through Discriminant Analysis of API References[C]. *The 2014 Workshop on Artificial Intelligent and Security Workshop*, 2014: 47-57.

[16] Lu X, Zhuge J W, Wang R Y, et al. De-Obfuscation and Detection of Malicious PDF Files with High Accuracy[C]. *2013 46th Hawaii International Conference on System Sciences*, 2013: 4890-4899.

[17] Srndic, Nedim and Pavel Laskov. Detection of Malicious PDF Files Based on Hierarchical Document Structure[C]. *The Network and Distributed System Security Symposium*. 2013.

[18] Maiorca, Davide Ariu, Igino Corona, et al. A structural and content-based approach for a precise and robust detection of malicious PDF files[C]. *2015 International Conference on Information Systems Security and Privacy*. 2015: 27-36.

[19] Cohen A, Nissim N, Rokach L, et al. SFEM: Structural Feature Extraction Methodology for the Detection of Malicious Office Documents Using Machine Learning Methods[J]. *Expert Systems with Applications*, 2016, 63: 324-343.

[20] Lu X F, Wang F, Shu Z F. Malicious Word Document Detection Based on Multi-View Features Learning[C]. *2019 28th International Conference on Computer Communication and Networks*, 2019: 1-6.

[21] Lu X F, Wang F, Jiang C, et al. A Universal Malicious Documents Static Detection Framework Based on Feature Generalization[J]. *Applied Sciences*, 2021, 11(24): 12134.

[22] Tzermias Z, Sykiotakis G, Polychronakis M, et al. Combining Static and Dynamic Analysis for the Detection of Malicious Documents[C]. *The Fourth European Workshop on System Security*, 2011: 1-6.

[23] Schmitt F, Gassen J, Gerhards-Padilla E. PDF Scrutinizer: Detecting JavaScript-Based Attacks in PDF Documents[C]. *2012 Tenth Annual International Conference on Privacy, Security and Trust*, 2012: 104-111.

[24] Snow K Z, Krishnan S, Monrose F, et al. SHELLOS: Enabling Fast Detection and Forensic Analysis of Code Injection Attacks[C]. *The 20th USENIX conference on Security*, 2011: 9.

[25] Office VBA 参考.https:. docs.microsoft.com/zh-cn/office/vba/api/overview. 2021.

[26] Kim S, Hong S, Oh J, et al. Obfuscated VBA Macro Detection Using Machine Learning[C]. *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2018: 490-501.

[27] DDE function. https:. support.microsoft.com/en-us/office/dde-function-79e8b21c-2054-4b48-9ceb-d2cf38dc17f9. 2021

[28] McAfee Labs Threat Advisory. https:. kc.mcafee.com/resources/sites/MCAFEE/content/live/CORP_KNOWLEDGEBASE/91000/KB91851/en_US/McAfee_Labs_Threat_Advisory-W97MMacroLess.pdf. 2017.

[29] Trends, challenges, and strategic shifts in the software vulnerability mitigation landscape. https:. github.com/Microsoft/MSRC-Security-Research/blob/master/presentations/2019_02_BlueHatIL/2019_01%20-%20BlueHatIL%20-%20Trends%2C%20challenge%2C%20and%20shifts%20in%20software%20vulnerability%20mitigation.pdf. 2019.

[30] https:. www.blackhat.com/docs/us-15/materials/us-15-Li-Attacking-Interoperability-An-OLE-Edition.pdf. 2015.

陈祥 于 2014 年在国防科技大学计算机专业获得硕士学位。现任战略支援部队信息工程大学信息技术研究所助理研究员。研究领域为网络内生安全。研究兴趣包括: 恶意文档检测、异常检测。Email: chenxndsc@163.com

伊鹏 战略支援部队信息工程大学信息技术研究所研究员, 河南省网络空间拟态防御重点实验室主任。研究领域为网络内生安全。研究兴趣包括: 新型网络体系结构、网络安全管控和主动防御技术研究。Email: 15238363586@139.com

白冰 于 2007 年于国防科技大学计算机专业取得硕士学位, 现任之江实验室助理研究员, 研究领域为网络空间安全, 研究兴趣包括内生安全、AI 安全等。Email: baibing@zhejianglab.com

韩伟涛 于 2020 年在战略支援部队信息工程大学信息与通信工程专业获得博士学位。现任战略支援部队信息工程大学副研究员。研究领域为网络结构安全、复杂网络鲁棒性等。Email: weitaohanchn@163.com