**基于机器学习的恶意文档检测与对抗性学习研究[[1]](#footnote-2)**

柯宗贵，王凤娇，江纬，杨育斌

（蓝盾信息安全技术股份有限公司，广东广州 510000）

**摘 要：**当前基于文档的攻击通常具有针对性，加上其文件结构的多样性，攻击手段变得丰富且隐蔽，因此更容易成功。文章提出了一个基于AI的恶意文档分类器原型，收集良性和恶意样本共20万个，经过对恶意样本静态解析，抽取具有显著分类能力的特征，训练生成AI模型。实验数据表明，模型的准确率达到99.82%，误报率只有0.01%。本文进一步研究了部分对抗性学习的方法，并用实验数据证明模型具有良好的抗逃逸能力和鲁棒性。

**关键词：**AI；机器学习；恶意文档检测；对抗性学习

**0 引言**

随着时间的推移，PDF规格和样式变得更加丰富，新版本增加脚本的功能使文档与可执行文件几乎能以相同的方式工作，如连接到Internet、运行进程及与其他程序进行交互等。这种复杂性的增长为攻击者提供了更多的武器发动攻击，并且能更灵活地隐藏恶意有效载荷逃逸检测。由于企业和个人普遍对此安全意识不足，导致大量系统未对版本进行更新，最终使这些攻击取得成功。

针对近年的多种基于文档的攻击，传统的PDF恶意文件检测方法如基于Shellcode的检测[1]、基于签名的检测[2]等，均存在无法及时更新的问题，导致无法检测到新的攻击。基于机器学习的方法在网页分类技术[3]、入侵检测[4]等方面取得了很好的效果，许多研究者也使用机器学习技术对PDF文件进行分析[5,6]，基于内容和结构提取文件的静态特征[7]，或者基于元数据与结构提取文件特征[8]对文件进行分类，经过AI算法调优后，可以达到很好的效果。研究者关注的不止是AI模型的预测能力[9]，更多的是AI模型的抗逃逸与鲁棒性，他们针对AI模型的逃逸提出了几种假设和方法[10]，最后以高概率成功逃逸分类器。本文使用4种方法来验证模型抗逃逸的能力，通过对模型进行改进，可以检测更多成功逃逸的样本。

**1 相关工作**

目前PDF恶意文档检测技术大约可分为两大类：静态分析和动态分析。两种方法在业界均有大量成功应用案例，更高级的解决方案是将静态和动态分析相结合[11]。表1是相关研究方法的对比。

表1 相关研究方法对比

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 方法 | 分析重点 | 检测技术 | 时间 | 外置  解析器 | ML | 可检测的  不同性 |
| 静态分析 | JavaScript | Lexical分析[5] | 2011 | Y | Y | Y |
| JavaScript | Token聚类[12] | 2012 | Y | Y | Y |
| JavaScript | API调用分类[13] | 2014 | Y | Y | Y |
| JavaScript | Shellcode and Opcode签名[2] | 2013 | N | N | N |
| Metadata | Linearized Object Path[14] | 2012 | Y | Y | Y |
| Metadata | 分层结构检测[15] | 2013 | Y | Y | Y |
| Metadata | 基于结构和Metadata[8] | 2012 | Y | Y | Y |
| Both | 基于结构和内容解析[7] | 2015 | Y | Y | Y |
| Both | 结合上述技术解析分类[11] | 2016 | Y | Y | Y |
| 动态分析 | JavaScript | Shellcode and Opcode签名检测[1] | 2011 | Y | N | N |
| JavaScript | 已知的攻击模式[16] | 2012 | Y | N | N |
| JavaScript | 内存访问模式[17] | 2011 | Y | N | Y |
| JavaScript | 常见maldoc行为分析[18] | 2014 | N | N | Y |
| JavaScript | 独立平台的tap point标识技术[19] | 2016 | N | N | Y |
| 文档类型 | 异常内存访问约束变量[20] | 2012 | N | N | N |
|  | 平台多样性 | 系统平台多样性利用[21] | 2017 | Y | N | Y |

由表1可知，静态分析技术一般聚焦于JavaScript[2]或Metadata[11]；动态分析技术一般聚焦于提取嵌入在PDF文档中的JavaScript代码[19]，再通过运行这些代码片段PlatPal[21]判定其恶意行为。其中80%的工作使用外置PDF解析器，这使得外置PDF解析器的健壮性成为研究焦点。几乎所有的静态分析工作都使用了机器学习技术，如PDFrate[8]。

Xu et al[22]构建了一个能自动生成恶意样本变种的框架，经过遗传算法将良性PDF对象加入到恶意样本的文件结构中混淆分类器，这种专门针对分类器的攻击及其框架被称为分类器逃逸攻击。

对于外置PDF解析器，现有的攻击手段是解析器混淆攻击；对于机器学习模型，现有的攻击手段是自动化分类器逃逸攻击。这些攻击手段对模型本身的安全提出了很大挑战，因此不仅需要生成准确度高的模型，并且在模型健壮性方面也需要有所突破[23,24]。

# 2 恶意文档检测器的设计与实现

本文于设计了一个基于机器学习的恶意文档检测框架，实验数据样本大小约为20万，主要对这些文件的内容和结构进行解析，选取具有良好分类效果的特征，然后对提取的特征通过机器学习进行分类。本文主要关注通用机器学习分类框架、数据集、特征工程、分类算法4个方面。

## **2.1 通用机器学习分类框架**

通用机器学习的分类器框架如图1所示。本文的目标是训练一个健壮性强的模型用于恶意文档分类。首先，在数据收集阶段抓取大量有标签的恶意/良性文档；然后，在特征工程阶段通过对比恶意样本与良性样本的差异，手工提取上百个文档的典型且有效的特征；最后，通过机器学习训练模型。本文的训练是离线的，并且需要相对较长的时间和较高的内存占用。预测过程是在线的，模型以服务的形式对外输出预测能力。当新样本提交到模型时，模型在短时间内可返回一个预测分数，分数越高表明模型越有信心预测其为恶意样本[25,26]。

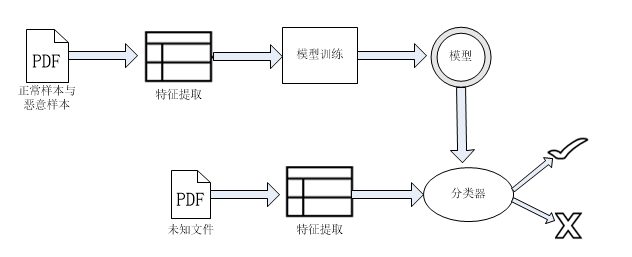


图1 分类器框架

## **2.2 数据集**

目前使用的数据集共有数据样本201368个，分为良性（28332）和恶意（173036）两大类。本文收集到的文件数据有167061个，其中156035个是从VirusShare下载得到，大小约6.8 GB；有9000个良性样本来自于Contagio；2026个良性数据集是在搜狗和百度上通过爬虫抓取的。本文还使用了mimicus[10]数据集中的2万平衡样本。使用VirusShare的数据样本作为源，使用攻击算法生成7000个对抗样本进行对抗性学习研究。

图2是数据样本的时间分布图。数据集中大部分样本来自2014年到2017年，这也也说明近几年恶意文档的产生在快速增多，要求检测引擎对未知恶意文档具有较强的检测能力。

图2 数据集的时间分布

## 2.3 特征工程

有效的特征提取方法主要基于结构、元数据、内容和Javascript实现，实验数据表明基于结构的特征具有很好的分类能力。本文通过计算数据集中每一个文档特征的平均值，发现良性样本与恶意样本的特征均值在某些特征中存在明显差异，如表2所示。

特征count\_font与count\_box：良性样本中有很多关于font、box等对象，而恶意文档一般不将展示信息作为其首要功能，通常直接将JS恶意代码嵌入文档中运行。在统计同一个页面中obj对象个数时，良性样本和恶意样本存在约1倍的差距，如果obj在同一个页面中突然增多，此文件为恶意文件的概率将增大。

特征count\_endobj 与count\_endstream：良性样本在每个对象结束时都有一个endobj，但恶意PDF文件为了混淆解析器会尽可能少地使用endobj和endstream，这就导致解析器在解析恶意PDF文件时不能完整获取整个对象，或导致整个PDF文件解析失败，使恶意PDF文件成功逃逸。这是恶意文件逃逸解析器最常使用的方法。

表2 良性样本与恶意样本特征均值对比

|  |  |  |
| --- | --- | --- |
| 特征 | 良性样本 | 恶意样本 |
| count\_font | 14.64 | 0.55 |
| count\_acroform\_obj | 700 | 1400 |
| count\_box\_a4 | 12001 | 200 |
| count\_box\_legal | 395040 | 0 |
| count\_endobj | 95.80 | 9.68 |
| count\_endstream | 30.43 | 3.78 |
| count\_page\_obj | 8001 | 16003 |
| count\_image\_small | 915892 | 12002 |
| count\_js | 0.71 | 1.01 |
| count\_obj | 100.96 | 12.01 |
| count\_objstm | 1.57 | 0.15 |

## 2.4 分类算法

AI引擎的重要组成部分之一是算法，本文对SVM算法、决策树算法、随机森林算法进行比较，经过多次训练与分类得到结果如表3所示。由表3可知，随机森林算法具有准确率高、误报率低、低延时、可解析性等优势，故本文使用随机森林作为模型的算法。随机森林算法是一个集成分类器，使用袋装训练数据随机选择特征子集，一个特征子集表示一棵树，利用这些特征数据确定每棵树的最佳分割点（每棵树相对独立），最后通过每棵树的投票确定最终结果。随机森林算法已被广泛应用于信息安全领域，如入侵检测算法研究[23]、网络舆情文本信息分类方法研究[26]等。

表3 不同分类算法比较

|  |  |  |  |
| --- | --- | --- | --- |
| 算法 | 检测率 | 训练时间/s | 检测时间/ms |
| SVM | 75.23% | 3489 | 1.2 |
| 决策树 | 82.41% | 4 | 0.1 |
| 随机森林 | 99.64% | 56 | 0.1 |

# 3 对抗性学习

对抗性学习是机器学习与计算机安全交叉的研究领域[25]。机器学习技术正越来越多地被用于各种恶意数据的检测[4]。然而，如果模型部署在线上，攻击者可以通过操纵数据进行逃逸。

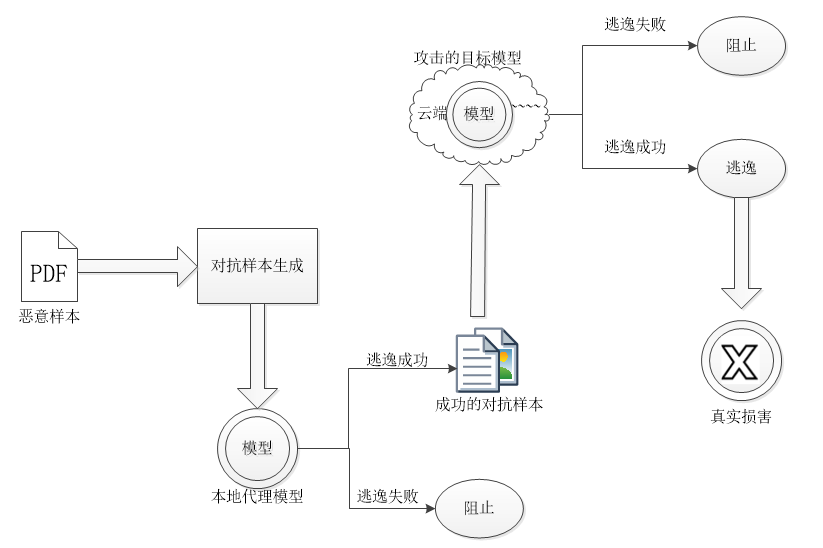


图3 模型逃逸框架

本文以部署在线上的Model2作为逃逸对象研究对抗样本的有效性。如图3所示，如果攻击者的目标是逃逸云端模型，首先会在本地部署一个代理模型（此模型应与逃逸模型相似），并选取一组具有代表性的恶意样本，利用不同的算法（攻击算法）生成恶意样本，改变文件本身、特征空间等，使其与良性样本相似但维持恶意行为；进而使用人工生成的有效的对抗样本尝试逃逸部署在云端的模型（假设是对抗样本的可迁移性）。云端模型若与本地代理模型相似度高（训练集、特征集、模型参数与算法等），对抗样本则能大概率逃逸此模型，对用户系统造成实在损害。本文重点关注对抗样本生成、案例分析（对抗样本迭代原理）、有效防御技术、模型性能等方面。

## 3.1 对抗样本生成

假设攻击者已知模型的一些附加信息，如模型提取的特征、模型算法等，如果攻击者道所有分类器组件的详细信息，则有最好的机会逃逸检测。这种情况下，可以在离线状态完全复制在线分类器，只有找到足够好的对抗样本时才提交攻击结果，如果逃逸了离线分类器，那么也有很大的可能逃逸在线分类器。

本文主要参考文献[10]中的几种攻击场景对模型进行对抗性学习。设F、T、C三个字母分别对应特征集、训练数据集和分类器算法。分别以攻击场景名表示4种攻击场景对应的信息：F（Feature）表示只有特征集可用于攻击者；FT（Feature and Training）表示除了已知特征外，攻击者还可以利用目标分类器训练数据集的知识；FC（Feature and Classifier）表示攻击者知道特征集及分类器的附加信息，如模型类型、参数、具体实现等；FTC（All Above）表示知道特征集、训练集和模型的全部细节，攻击者可以在线下通过训练完全重现在线分类器，只有找到足够好的对抗样本时才提交攻击结果。

## 3.2 案例分析

在变异过程中挑选一些典型的样本进行案例分析。选择一个包含有恶意代码的PDF文件，该文件可以利用漏洞（CVE-2013-0641）远程执行任意代码，通过以上4种攻击场景对选取的样本进行变异，然后分别查看样本的VT报告。样本变异前，VT报告中有33个检测引擎将其判断为恶意文件，变异后，识别为恶意文件的引擎减少为22个。样本经过变异后的VT检测结果如表4所示。

表4 样本经过变异后的VT检测结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| File\_HASH | 变异前 | F | FC | FT | FTC |
| 00ba5c43b1cec186c634c24ac21982d3  cve-2013-0641 | 33/61 | 22/60 | 23/60 | 22/60 | 22/60 |

由于大多数PDF文件检测器是基于结构和内容的，所以只要对文件结构和内容做一些改变，如添加良性样本的一些对象或者改变文件大小等，就可以成功逃逸分类器。于是将变异后的文件特征与变异前的文件特征进行比较，如表5所示，变异后样本依然保持其恶意代码，但新增10个分类器不能检测出文件的恶意代码。变异主要修改了文件的metadata大小和内容，增加作者信息（author\_lc从0到6，author\_len从0到14），增加count\_javascript的数量（从1到6），升级了版本（从4到7）。

表5 样本变异前后特征对比

|  |  |  |
| --- | --- | --- |
| 特征 | 变异前 | 变异后 |
| author\_lc | 0 | 6 |
| author\_len | 0 | 14 |
| count\_javascript | 1 | 6 |
| createdate\_ts | -1 | 650616173 |
| moddate\_ts | -1 | 482083775 |
| keywords\_len | 0 | 7 |
| producer\_lc | 0 | 8 |
| version | 4 | 7 |

## 3.3 有效防御技术

通过分类器找出评分较高的2000个恶意样本作为种子，使用上述的4种攻击场景生成对抗样本测试Model2。由表6（理想实验环境下FC攻击场景对抗样本数应为2000，但使用240也可得到同样结果）可知，4种攻击场景对Model2均有较大影响，其中，在FC的攻击场景下，Model2对对抗样本的检出率只有2.92%，这使我们需要对模型继续研究。

表6 不同攻击场景下模型的准确率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 攻击场景 | F | FC | FT | FTC |
| 对抗样本数目/个 | 2000 | 240 | 2000 | 2000 |
| Model2 | 71.18% | 2.92% | 84.25% | 15.83% |
| Model2.1 | 96.71% | 12.50% | 96.76% | 18.71% |
| Model2.2 | 89.43% | 81.95% | 98.58% | 86.21% |
| Model2.3 | 71.03% | 2.85% | 85.89% | 36.82 |
| Model3 | 98.65% | 88.89% | 98.98% | 92.83% |

1）Model2.1通过增大训练数据集（从10万到20万）达到增强模型健壮性的目的。由实验数据可知，训练样本数据越多，模型的健壮性越好（在F与FT情况下，模型的对抗样本检出率达到96.7%），同时有研究者认为集合多个分类器可以使模型更稳定[24]。

2）Model2.2在训练过程中添加大量生成的对抗样本，在机器学习中，该方法称为对抗性训练。本文将约7000个对抗样本加入Model2.2训练中，实验结果表明，Model2.2准确率有很大提升（尤其在FC与FTC两种攻击场景下模型表现出众），主要原因是该模型在训练阶段已具有对对抗样本进行识别的能力，故只要训练数据集和测试数据集具有相似的数据分布和样本特征，该模型在对抗性学习的测试就会有良好的性能。

3）Model2.3对阈值进行调整。如果仍然使用默认阈值（0.5），由于对抗样本的平均概率已经大幅降低，大多数对抗样本将能成功逃逸，会对计算环境造成潜在损害。通过降低阈值（如设定为0.4）可增加模型灵敏度，能对更多可疑文件进行更加全面的分析。

4）Model3是Model2.1、Model2.2和Model2.3的结合，该模型由一个更大的数据集（包括大量的对抗样本）训练，并调整阈值。显而易见，Model3是目前最好模型。

本文还研究了防御机制的有效性，果特征集已被攻击者利用，可通过改变特征集如修改权值或删除重要特征等操作重新训练模型。图4是Model3按照特征权重排序的前30个特征，可以看到count\_font、count\_javascript、size、count\_obj、count\_endobj等特征占有较大权重，同时也很容易被攻击者利用实现解析器和分类器逃逸。因此，可以在训练时删除这些已暴露特征，重新训练模型。

图4 Model3前30个重要特征排序

表7是对Model3前5个特征依次删除后的模型准确率。当分类器使用全部特征进行训练时，模型准确率达99.82%，当将第一重要特征删除时，检测率基本没有太大波动，当删减到前5个特征时，模型准确率依旧没有明显变化，维持在高位。

表7 依次删除前5个特征后模型准确率

|  |  |
| --- | --- |
| 删除特征 | 准确率 |
| 无删除 | 99.82% |
| 1 | 99.52% |
| 1,2 | 99.52% |
| 1,2,3 | 99.64% |
| 1,2,3,4 | 99.64% |
| 1,2,3,4,5 | 99.64% |

本文针对模型健壮性问题提出了5种防御技术：1）增加多样化和新样本进行训练（Model2.1）；2）对抗性训练（Model2.2）；3）调整模型阈值（Model2.3）；4）及时删除被对手利用的特征并重新训练模型；5）隐藏分类器的相关参数，如特征、阈值、算法等。

## 3.4 模型性能

本文通过对特征有效性进行研究来评估模型的健壮性。图5是特征在不断自删除时所对应的准确率曲线。由图5可知，当特征减少至100个时，重新训练后的模型准确率依然高达90%，这说明：1）当删除权重高特征时模型准确率会下降，但影响不大；2）中等权重特征能使模型持续有效，抵消删除高权重特征造成的缺失。

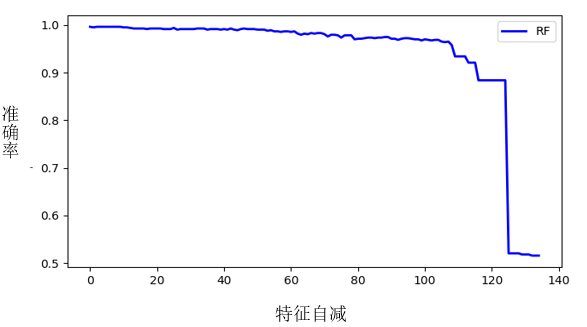


图5 特征自删除后的检测率（Model3）

为了评估模型的预测性能，将数据集随机分为训练（90％）和测试（10％）两部分，并采用10-Fold交叉验证评估模型有效性。图6为ROC曲线图，曲线下面积约为1，表明模型具有良好的预测性能。



图6 ROC曲线图

**4 结束语**

本文详细介绍了基于AI的PDF恶意文档分类器的设计与实现。实验数据表明，在20万数据集中本文模型有大于99%的准确率和小于0.01%的误报率。本文还提出了5种增强模型健壮性的防御技术，并通过实验证明其有效性。基于AI的恶意文档检测是社会工程、病毒分析等领域的重要研究课题，未来还会尝试使用动静态分析结合的方法检测恶意文档，并同时扩展文件检测类型，如docx、pptx等。

# 参考文献：

1. TZERMIAS Z, SYKIOTAKIS G, POLYCHRONAKIS M, et al. Markatos Combining Static and Dynamic Analysis for the Detection of Malicious Documents[C]//ACM. 4th European Workshop on System Security, April 10, 2011, Salzburg, Austria. New York: ACM: 1-6.
2. LU Xun, ZHUGE Jianwei, WANG Ruoyu, et al. De-obfuscation and Detection of Malicious PDF Files with High Accuracy[C]//IEEE. 46th Hawaii International Conference on System Sciences, January 7-10, 2013, Wailea, Maui, HI, USA. New Jersey:IEEE, 2013: 4890-4899.
3. 孙靖超. 一种基于机器学习的网页分类技术[J]. 信息网络安全，2017，17（9）：45-48.
4. 和湘, 刘晟, 姜吉国. 基于机器学习的入侵检测方法对比研究[J]. 信息网络安全，2018，18（5）：1-11.
5. Laskov Pavel and Srndic Nedim.Static Detection of Malicious JavaScript-Bearing PDF Documents[C]//ACSAC. In Proceedings of the Annual Computer Security Applications Conference, December 05 - 09, 2011. Orlando, Florida, USA . NY, USA. ACM .2011 . 373-382.
6. Šrndić N, Laskov P. Hidost: a static machine-learning-based detector of malicious files[J]. EURASIP Journal on Information Security, 2016, 2016(1): 22.
7. Maiorca Davide, Ariu Davide, Corona Igino,et al. A Structural and Content-based Approach for a Precise and Robust Detection of Malicious PDF Files[C]//ICISSP. In Proceedings of the International Conference on Information Systems Security and Privacy, 9-10 February,2015. Olivier Camp . France.IEEE. 14 July 2016: 27-36.
8. Smutz Charles and Stavrou Angelos. Malicious PDF Detection using Metadata and Structural Features[C]// ACSAC .In Proceedings of the Annual Computer Security Applications Conference. December 03 - 07, 2012. Orlando, Florida, USA. New York. ACM. 2012:239-248.
9. Biggio B, Fumera G and Roli F. Design of robust classifiers for adversarial environments[C]//IEEE, 2011 IEEE International Conference on Systems, Man, and Cybernetics. 9-12 October 2011. Anchorage, AK, USA. Anchorage. IEEE. 2011:977-982
10. Srndic Nedim and Laskov Pavel. Practical Evasion of a Learning- Based Classifier: A Case Study[C]//IEEE, .In Proceedings of the 35th IEEE Symposium on Security and Privacy (Oakland). 18-21 May 2014. San Jose, CA, USA. CA, USA. IEEE. 2014:197-211.
11. Maiorca D, Ariu D, Corona I, et al. An Evasion Resilient Approach to the Detection of Malicious PDF Files[C]//International Conference on Information Systems Security and Privacy. Springer, Cham, 2015: 68-85.
12. Vatamanu C, Gavriluţ D, Benchea R. A practical approach on clustering malicious PDF documents[J]. Journal in Computer Virology, 2012, 8(4): 151-163.
13. Corona I, Maiorca D, Ariu D, et al. Lux0r: Detection of malicious pdf-embedded javascript code through discriminant analysis of api references[C]//Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop. ACM, 2014: 47-57.
14. Maiorca D, Giacinto G, Corona I. A pattern recognition system for malicious pdf files detection[C]//International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer, Berlin, Heidelberg, 2012: 510-524.
15. Šrndic N, Laskov P. Detection of malicious pdf files based on hierarchical document structure[C]//Proceedings of the 20th Annual Network & Distributed System Security Symposium. 2013: 1-16.
16. 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. IEEE, 2012: 104-111.
17. Snow K Z, Krishnan S, Monrose F, et al. SHELLOS: Enabling Fast Detection and Forensic Analysis of Code Injection Attacks[C]//USENIX Security Symposium. 2011: 183-200.
18. Liu D, Wang H, Stavrou A. Detecting malicious javascript in pdf through document instrumentation[C]//Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on. IEEE, 2014: 100-111.
19. Carmony C, Hu X, Yin H, et al. Extract Me If You Can: Abusing PDF Parsers in Malware Detectors[C]//NDSS .In Proceedings of the 2016 Annual Network and Distributed System Security Symposium San Diego, 2016.
20. Willems C, Freiling F C, Holz T. Using memory management to detect and extract illegitimate code for malware analysis[C]// ACSAC. Proceedings of the 28th Annual Computer Security Applications Conference. ACM, 2012: 179-188.
21. Xu Meng and Kim Taesoo. PlatPal: Detecting Malicious Documents with Platform Diversity[C]//USENIX, USENIX(26th). August 16–18, 2017, Vancouver, BC, Canada. 2017:0-20
22. Xu W, Qi Y, Evans D. Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers[C]// NDSS.In Proceedings of the 2016 Annual Network and Distributed System Security Symposium San Diego, CA, February 2016. http:evademl.org, 2016.
23. 林伟宁, 陈明志, 詹云清, 刘川葆. 一种基于PCA和随机森林分类的入侵检测算法研究[J]. 信息网络安全,2017,17（11）：50-54
24. Smutz C, Stavrou A . When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors[C]//NDSS. In Proceedings of the 2016 Annual Network and Distributed System Security Symposium San Diego, 2016.
25. 顾巧云，孙玉龙，高丰.基于博弈论的网络攻防对抗模型及应用研究[J].信息网络安全, 2013, 13(1):0-0
26. 吴坚，沙晶.基于随机森林算法的网络舆情文本信息分类方法研究[J].信息网络安全, 2014, 14(11):36-43

作者简介：作者1：柯宗贵（1969－至今），男，中国，副董事长兼总经理，学士学位，研究方向：信息安全，广州市天河区科韵路16号信息港A栋20-21楼，邮编：510500，电话：13802736488，kzg@chinabluedon.cn ;

作者2：王凤娇（1992－至今），女，中国云南·大理，研究员，学士学位，研究方向：信息安全，广州信息港，邮编：510500，电话：15602209397，[yonahwang@foxmail.com](mailto:yonahwang@foxmail.com);

作者3：江纬（1986－至今 ），男，广东·广州，研究员，硕士学位，研究方向：信息安全，广州信息港，手机：18802014980，[weijiang2009@gmail.com](mailto:weijiang2009@gmail.com)；

作者4：杨育斌（1974－至今），男，广东·广州，高级工程师，博士，研究方向：信息安全、云计算、移动互联网、网络应急体系，广州信息港，手机：139 2601 3338，电话：（86-20）85526663，Email: yyb@chinabluedon.cn。

1. 收稿日期：2018-7-17

   作者简介：柯宗贵（1969—），男，**中国·广东**，本科，主要研究方向为信息安全；王凤娇（1992—），女，云南，研究员，本科，主要研究方向为信息安全；江纬（1986—），男，广东，研究员，硕士，主要研究方向为信息安全；杨育斌（1974—），男，广东，高级工程师，博士，主要研究方向为信息安全、云计算、移动互联网、网络应急体系。

   通信作者：王凤娇 yonahwang@foxmail.com [↑](#footnote-ref-2)