# 关于机器学习之PDF文件检测与分类器逃逸

实验报告

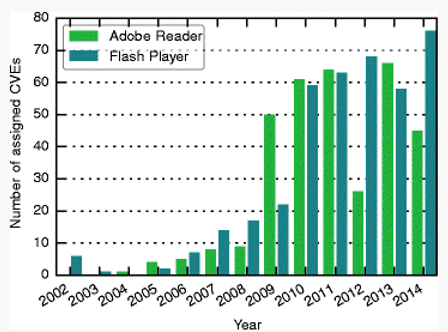
## 摘要

## 1.简介

非可执行文件作为有*针对性的攻击*手段尤其受欢迎。近年来，针对政府和工业界的一系列高调针对性攻击，并且越来越普遍且越来越隐蔽。针对欧洲政府机构的Miniduke针对性攻击活动使用了利用Adobe Reader零日漏洞的复杂PDF文件。

攻击者使用的最喜欢的格式是PDF（针对ADOBE READER），在2012年，开创性的漏洞攻击工具Blackhole专门针对Java，PDF和Flash文件，其继任者继续这种做法[ [3](https://link.springer.com/article/10.1186/s13635-016-0045-0#CR3) ]。2013年，通过网络传输的不可执行恶意软件主要是针对Adobe Reader和Microsoft Office应用程序的PDF和Flash文件[ [2](https://link.springer.com/article/10.1186/s13635-016-0045-0#CR2) ]。近期Flash已经广泛部署恶意广告，即通过广告网络在合法网站上放置恶意软件。甚至一些最着名的网站也成为这种攻击的受害者[ [3]](https://link.springer.com/article/10.1186/s13635-016-0045-0#CR3)]

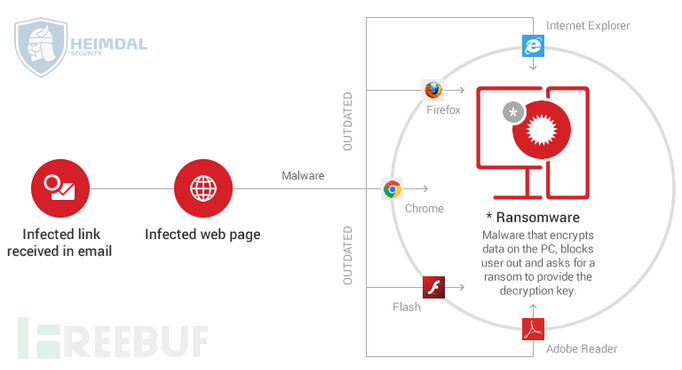
在2014年发现的24个0天中，有16个是针对Adobe Reader和Flash Player的（参见图[1](https://link.springer.com/article/10.1186/s13635-016-0045-0" \l "Fig1)）



## 无处不在的勒索软件

勒索软件是一种恶意软件，感染后它会阻止你访问文件系统，以此来要挟你支付赎金，这类的软件通常要求使用比特币支付，要是你在一定的时间内没有进行支付，它会自动将赎金加倍。而且在你支付之前，这个循环会无限滚雪球下去。这种攻击的强度和频率会持续增加，而它给受害者造成的损失也会一路狂飙。

部分数据统计“勒索软件受害者的数量在上升，在2015年4月到2016年3月之间有大概718536位受害者，这相当于2014-2015年同期的5.5倍。”



## 2.PDF文件分析

*可移植文档格式*（PDF）是一个开放标准，发布为ISO 32000-1：2008 [ [9](https://link.springer.com/article/10.1186/s13635-016-0045-0#CR9) ]。PDF文件格式包含以下4个部分：

**文件头——指明了该文件所遵从的PDF规范的版本号，它出现在PDF文件的第一行。**

**文件体——又称对象集合，PDF文件的主要部分，由一系列对象组成。**

**交叉引用表——对对象进行随机存取而设立的一个间接对象的地址索引表。（实际以偏移+索引的方式储存对象地址，下文会提及）**

**文件尾——声明了交叉引用表的地址，即指明了文件体的根对象（Catalog），从而能够找到PDF文件中各个对象体的位置，达到随机访问。另外还保存了PDF文件的加密等安全信息。**

**PDF文件格式图示：**



本段主要介绍PDF文件体的读取方式。

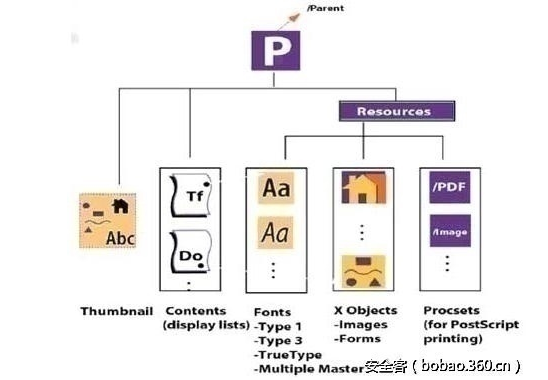
作为一种结构化的文件格式，一个PDF文档是由一些称为“对象”的模块组成的。每个对象都有数字标号，这样的话可以这些对象就可以被其他的对象所引用。这些对象不需要按照顺序出现在PDF文档里面，出现的顺序可以是任意的，比如一个PDF文件有3页，第3页可以出现在第1页以前，对象按照顺序出现唯一的好处就是能够增加文件的可读性，对象的信息以偏移+索引的形式保存在交叉引用表内。

文件尾说明了根对象的对象号，并且说明交叉引用表的位置，通过对交叉引用表的查询可以找到目录对象(Catalog)。这个目录对象是该PDF文档的根对象，包含PDF文档的大纲(outline)和页面组对象（pages）引用。大纲对象是指PDF文件的书签树；页面组对象（pages）包含该文件的页面数，各个页面对象(page)的对象号。因此，PDF阅读器通常从文件末尾开始阅读PDF以提高效率

页面（page）对象为PDF中最重要的对象，包含如何显示该页面的信息，例如使用的字体，包含的内容（文字，图片等），页面的大小。里面的信息可以直接给出，当然里面的子项更多的是对其他对象的引用，真正的信息存放在其他对象里面。页面中包含的信息是包含在一个称为流（stream）的对象里，这个流的长度（字节数）必须直接给出或指向另外一个对象（包含一个整数值，表明这个流的长度）。

可见stream流对象我们恶代分析需要获取的重点。

页面信息图示：



## 3.PDF特征选择

理解了上面的内容之后，我们可以得出针对恶代分析的PDF文件的大致解析思路：

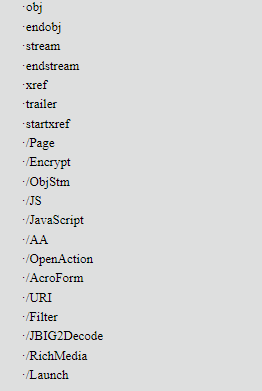
文件去除混淆

查找关键字

根据关键字获得可能存在恶意代码的stream 或者buffer

解码stream ，获取恶意代码

每一个PDF文件都包含有前7个字段，也有可能不包含strea和endstream。据说有一些ODF文件没有xref或则trailer，但是这种情况比较少见。如果一个PDF文件没有xref或者trailer关键字段，那么可以确定它不是恶意的PDF文件。



/xref 交叉引用表，描述每个间接对象的编号、版本和绝对的文件位置。PDF文档中的第一个索引必须从版本65535的0号对象开始，标识符/xref后面的第一个数字是第一个间接对象（即0号对象）的编号，第二个数字是/xref（交叉引用表）的大小。  
/Page指明PDF文件的页数，大多数恶意PDF文件仅仅只有一页  
/Encrypt指明PDF文件有数字水印或者是被加密过的。  
/ObjStm是object streams的数量。object streams是一个可以包含其他Object对象的数据流对象。  
/JS与/JavaScript指明PDF文件中含嵌有JavaScript代码。通常恶意的PDF文件都嵌套有JavaScript代码，这里一般都是利用JavaScript的解析漏洞或者使用JavaScript来实现堆喷射（heap spray），也有很多正常的PDF文件里会含有JavaScript代码  
/AA、/OpenAction和/AcroForm指明当查看PDF文件或者PDF的某页时会有动作随其执行，几乎所有嵌有JavaScript代码的恶意PDF文件都有自动执行JavaScript代码的动作(action)。如果一个PDF文件包含有/AA或/OpenAction自动执行动作的关键字段，而且含有JavaScript代码，那么这个PDF文件就极有可能是恶意的PDF文件  
/URI 如果你要在PDF文件中执行打开网页的动作就需要这个关键字段  
/Filter 一般为FlateDecode则是使用了Zlib压缩解压缩算法。  
/JBIG2Decode指明PDF文件使用了JBIG2压缩。虽然JBIG2压缩本身可能会有漏洞（CVE-2010-1297）。但/JBIG2Decode关键字段并不能说明PDF文件是否可疑  
/RichMedia Flash文件  
/Launch执行动作(action)数量

## 4.分类评估方法

所提出的用于基于结构的检测恶意PDF文档的方法包括以下两个步骤，如图2所示：

1.     *提取结构特征。*作为基本的预处理步骤，PDF文档的内容被解析并转换成特殊的形式，*包 - 路径*，以一种定义明确的方式表征文档结构。

2.     *学习和分类。*检测过程是由恶意和良性的PDF doc

Benign file

Mal\_file

Choose 80% of training at random

Model

Classification

Unknown file file

Prediction

Feature Extraction

## 5.模型训练与测试

## 6.分类器对抗逃逸分析

## 7.结论