题目：PDF 文件格式分析
Date：2010.10.31
Author：Cryin'
Link：https://github.com/Cryin/PDFTear
一、概述：

结构化的文档格式 PDF(Portable Document Format)是由美国排版与图像处理软件公司 Adobe 于 1993 年首次提出的。Adobe Reader 这款 pdf 阅读器软件相信大家并不陌生，人们熟知它的原因是因为它的应用相当普及，可能接触过计算机的人中没有几个会不知道它，但是相信也有一部分人注意到 Adobe 这款软件是因为它频频爆出漏洞的缘故，号称漏洞之王的 Adobe 似乎有报不完的漏洞，时不时就会给人以惊喜，其潜力真是不容置疑！这样说来，如果想对 Adobe 的漏洞原理进行分析。了解 PDF 文件的格式就变得尤为重要了！

二、PDF 文件结构：

PDF 的结构可以从文件结构和逻辑结构两个方面来理解。PDF 的文件结构指的是其文件物理组织方式，逻辑结构则指的是其内容的逻辑组织方式[1]。

1、数据对象类型：

PDF 文件的基本元素是 PDF 对象(PDF Object)，PDF 对象包括直接对象(Direct Object)和间接对象(Indirect Object)；其中直接对象如下几种基本类型：布尔型(Boolean)、数值型(Number)、字符串型(String)、名字型(Name)、数组型(Array)、字典型(Dictionary)、流对象(Stream)以及空对象(Null)；间接对象是一种标识了的 PDF 对象，这个标识叫作间接对象的 ID。标识的目的是为了让别的 PDF 对象引用。任何 PDF 对象标识后都变成了间接对象。

2、PDF 文件结构：

PDF 的文件结构(即物理结构)包括四个部分：文件头(Header)、文件体(Body)、交叉引用表(Cross-reference Table)和文件尾(Trailer),如图-1 所示：

图-1 PDF 文件结构

文件头(Header)指明了该文件所遵从 PDF 规范的版本号，它出现在 PDF 文件的第一行。如％PDF-1.6 表示该文件格式符合 PDF1.6 规范。

文件体(Body)由一系列的 PDF 间接对象组成。这些间接对象构成了 PDF 文件的具体内容如字体、页面、图像等等。

交叉引用表(Cross-reference Table)则是为了能对间接对象进行随机存取而设立的一个间接对象地址索引表。

文件尾(Trailer)声明了交叉引用表的地址，指明文件体的根对象(Catalog)，还保存了加密等安全信息。根据文件尾提供的信息，PDF 的应用程序可以找到交叉引用表和整个 PDF 文件的根对象，从而控制整个 PDF 文件。

3、PDF 文档结构：

PDF 的文档结构反映了文件体中间接对象间的等级层次关系。PDF 的文档结构是一种树型结构如图-2 所示。树的根节点就是 PDF 文件的目录对象(Catalog)。这个目录对象是 PDF 文档的根对象，包含 PDF 文档的大纲(Outlines)和页面组对象(Pages)。根节点下有四个子树：页面树(Pages Tree)、书签树(Outline Tree)、线索树(Article Threads)、名字树(Named Destination)。

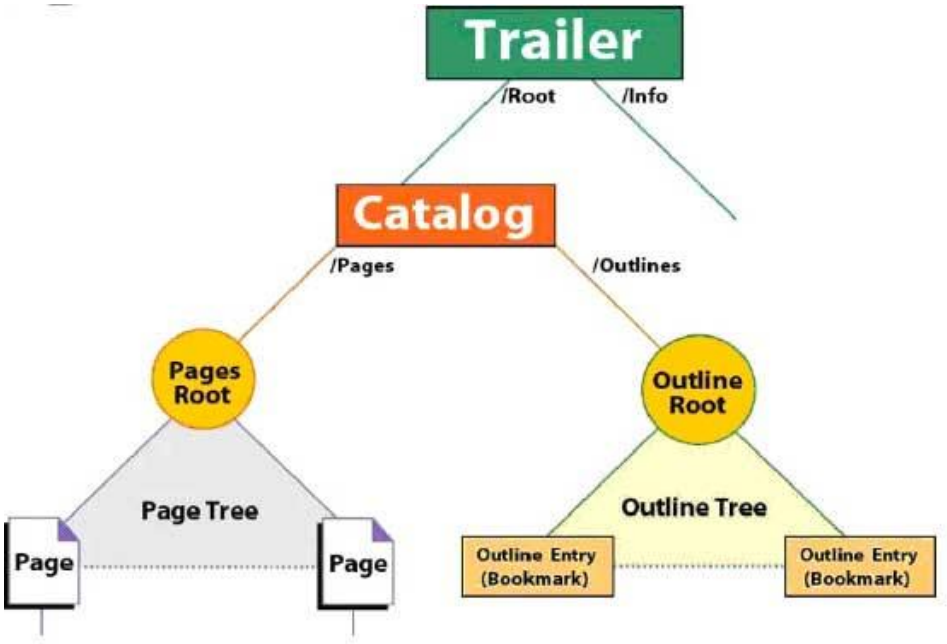

图-2 PDF 文档结构

4、PDF 中的资源：

PDF 文件中的内容(如文字、图形、图像)都保存在页面对象的 Contents 关键字对应的流对象(Stream)中。内容流(Content Stream)中用到了很多基本对象如数字、字符串，这些都是用直接对象(Direct Object)表示的。但还有其他一些对象如字体(Font)，本身就是用字典对象(Dictionary)或流对象(Stream)来表示的，无法用直接对象表示，而内容流中又不能出现任何间接对象，于是就将这些对象命名，并在内容流中用相应的名字来表示它们。这些用名字来表示的对象就称作命名资源(Named Resources)。

在页面对象中有一个资源项(Resources Key)，该项列出了内容流中用到的所有资源，并建立了一个资源名字与资源对象本身的映射表。

PDF 中的命名资源有：指令集(ProcSet)、字体(Font)、色彩空间(Color space)、外部对象(XObject)、扩展的图形状态(Extended graphics state)、底纹(Pattern)、用户扩展标记列表(Property list)。非命名资源有：Encoding、Font descriptor、Halftone、Function、CMap。由于非命名资源都是被隐含引用的，因此没有命名的需要。

5、PDF 页面描述命令：

PDF 共有 60 个页面描述指令。这 60 个页面描述指令描述了页面上的一系列图形对象。这些图形对象可分为四类：路径对象（Path Object）,文本对象（Text Object）,图像对象（Image Object）,外部对象（XObject）。

三、PDF 文件分析：

PDF 文件是一种文本和二进制混排的格式，但是 Adobe 更愿意让人把它当成二进制的文件，所以在里面建议当文件里面的文本太多的时候，可以加一些二进制的注释，好让现有的一些编译器把它当成二进制文件。里面的文本主要是对文件结构的一种描述，二进制的内容来自于三个方面：1、图片；2、字体；3、压缩后的 Post Script[2]。

下面使用只有一句话的一个 PDF 文件进行分析，使用 UltraEdit 打开 PDF 文件，然后选择以十六进制编辑就能看到类似下面的信息，我将着重挑选部分信息进行介绍，并使用#进行注释并在后面做相关解释。

```
%PDF-1.6            #文件头，说明符合 PDF1.6 规范
%忏嫌               #下面就是很多的 Object 对象
2 0 obj             #Object 对象，其中 2 是 Obj 顺序号，0 是 Obj 的版本号
<<                  # <<>>之间为 Object 对象的内容
[/ICCBased 3 0 R]
>>
Endobj              #Object 结束关键字

7 0 obj
<<
/Filter
/FlateDecode        #流对象的压缩方式为 zip 的压缩算法
/Length 148         #流对象的长度
>>
Stream              #流对象
PDF 文件格式分析  Author：Cryin'   #文件内容信息，注：此处为直观从而手动填写的
Endstream           #流对象结束标志
Endobj

8 0 obj
<<
/Contents 7 0 R             #页面内容对象的对象号为 7
/MediaBox [0 0 595.2 841.68]      #页面显示大小，以像素为单位
/PageIndex 1
/Parent 1 0 R       #其父对象号为 1 以及 Pages 对象
/Resources          #该页包含的资源
<</Font <</F4 4 0 R >>       #字体的类型
  /Shading <<>>
  /XObject <<>>             #外部对象
/ColorSpace <</CS1 2 0 R>> >>        #色彩空间
/Type /Page
>>
Endobj

1 0 obj
<<
/Count 1            #页码数量为 1
```

```
/Kids [8 0 R ]        #kids 对象说明它的子页对象为 8
/Type /Pages
>>
Endobj

13 0 obj
<<
/Author (? Cryin')
/CreationDate (D:20100926145832+08'00')
/Title (? PDF 文件格式分析)
>>
endobj

Xref                  #表示交叉引用表开始
0 14                  #0 表明引用表描述的对象从 0 开始，8 说明共有 8 个对象
0000000000 65536 f    #一般 pdf 都是以这行开始交叉引用表的，起始地址 0 和产生号
0000003195 00000 n    #表示对象 1，就是 catalog，3195 为偏移地址 n 表示对象在使用
0000000018 00000 n
0000000051 00000 n
0000003464 00000 n
0000000000 00000 f
0000004282 00000 n
0000002728 00000 n
0000002992 00000 n
0000003256 00000 n
0000003892 00000 n
0000003620 00000 n
0000008660 00000 n
0000008712 00000 n
Trailer               #说明文件尾对象开始
<</Size 14            #14 说明 PDF 文件对象数目
/Root 12 0 R          #说明跟对象号为 12
/Info 13 0 R>>
startxref
8980                  #8980 为交叉引用表的偏移地址，此处为十进制表示
%%EOF                 #文件结束标志
```

结束语：

PDF 文件基本的格式就分析到这里了，当然这里并没有谈到对 PDF 漏洞文件的分析，本例分析的 PDF 并没有嵌套 JavaScrpit 语句，但 PDF 漏洞分析的着手点就在于文件嵌套的 JavaScrpit 或者是 flash 文件。PDF 文件漏洞一般是利用 JavaScript 来实现堆喷射完成溢出[3]。这里就必然会在 PDF 文件中嵌套 JavaScript 语句了，PDF 中嵌套的 JavaScript 可以通过 Obj 对象里面的/OpenAction 这项定位其具体位置，不过一般都是经过 FlateDecode 编码过的，总之在分析 PDF 漏洞时先找出 JS 语句就可以找到其中的 shellcode，也可以自己修改 JavaScript。

这样对漏洞的分析就能更顺利的开展，当然具体分析还牵扯一些调试的过程，这方面本人也正在学习阶段，难免表述有所错误，还请见谅。总之，知识就是不断的总结和积累，也希望这篇简短的文章能给大家带来些许帮助！

参考：

[1]、《面向对象的中文 PDF 阅读器的设计与实现》作者：杨道良
[2]、《完整剖析 Acrobat Reader - Collab getIcon universal exploiter 之路》 作者：snowdbg
[3]、《Heap Feng Shui in JavaScript》 作者：Alexander Sotirov
[4]、《PDF Reference sixth edition》 作者： Adobe Systems Incorporated