PDF Examination

Investigating PDF Metadata

There are mutliple tools available to investigate the metadata attachded to PDF files. Of course, you can just use windows and show the dialog with the file properties & metadata. But this is limited as it doesn't show all metadata and not feasible for a forensic investigation as you want to document your findings. For this command line applications are better suited.

One such tool is pdfinfo which comes with the xpdfreader suite. If it isn't yet installed it can be done under kali-linux using *sudo apt install -y xpdfreader*. To show the pdfs metadata just run the command *pdfinfo <filename>* and have a look at all metadata.

As an alternative exiftool can be used. It can show some more metadata, like protections on the file, the language or the xmp toolkit used for the metadata. I suggest using arguments *-a* and *-G1* to print duplicate metadata tags too and print the tags group names.

So the command is exiftool -a -G1 <filename>.

```
┌──(joblo㉿PC-J0B10)-[/mnt/d/OneDrive/THU/S4 SS22/DIFO]
└─$ exiftool -a -G1 Handbook.pdf
[ExifTool]      ExifTool Version Number         : 12.41
[System]        File Name                       : Handbook.pdf
[System]        Directory                       : .
[System]        File Size                       : 878 KiB
[System]        File Modification Date/Time     : 2022:05:09 22:11:13+02:00
[System]        File Access Date/Time           : 2022:05:10 10:01:18+02:00
[System]        File Inode Change Date/Time     : 2022:05:10 10:01:18+02:00
[System]        File Permissions                : -rwxrwxrwx
[File]          File Type                       : PDF
[File]          File Type Extension             : pdf
[File]          MIME Type                       : application/pdf
[PDF]           PDF Version                     : 1.7
[PDF]           Linearized                      : No
[PDF]           Author                          : Jonas Blocher
[PDF]           Create Date                     : 2022:05:09 22:09:51+02:00
[PDF]           Creator                         : Microsoft® Word für Microsoft 365
[PDF]           Modify Date                     : 2022:05:09 22:11:12+02:00
[PDF]           Producer                        : Microsoft® Word für Microsoft 365
[PDF]           Has XFA                         : No
[PDF]           Language                        : de-DE
[PDF]           Tagged PDF                      : Yes
[PDF]           Page Count                      : 13
[PDF]           Signing Location                : Bissingen an der Teck
[PDF]           Signing Date                    : 2022:05:09 22:11:09+02:00
[PDF]           Signing Authority               : Jonas Blocher
[PDF]           Signing Reason                  : Ich bin der Autor des Dokuments
[PDF]           Modification Permissions        : No changes permitted
[XMP-x]         XMP Toolkit                     : XMP Core 5.5.0
[XMP-pdf]       Producer                        : Microsoft® Word für Microsoft 365
[XMP-dc]        Creator                         : Jonas Blocher
[XMP-xmp]       Creator Tool                    : Microsoft® Word für Microsoft 365
[XMP-xmp]       Create Date                     : 2022:05:09 22:09:51+02:00
[XMP-xmp]       Modify Date                     : 2022:05:09 22:11:12+02:00
[XMP-xmpMM]     Document ID                     : uuid:79A64C2B-8ED1-4256-ACCA-5AABBCB211A2
[XMP-xmpMM]     Instance ID                     : uuid:79A64C2B-8ED1-4256-ACCA-5AABBCB211A2
```

Detecting manipulations

Metadata may contain valuable information that help to qualify the importance of a PDF document for your investigation, like if it was created by a suspect or at the time of a crime. But Metadata is even more valuable to quickly identify manipulations to a scanned document. Most Programs that create PDF files set the **Producer** metadata tag. By this it can be quickly identified if the document was created by a scanner or by a PDF Editing Software with which the file could have been modified. Also, you can check if creation and modification dates differ.

This indicates that a file was modified but doesn't necessarily mean that it has been modified. You can use it tough as a quick indicator that some further investigations on the file should be conducted.

```
[PDF]          PDF Version                     : 1.2
[PDF]          Linearized                      : No
[PDF]          Modify Date                     : 2022:05:10 10:13:28+02:00
[PDF]          Producer                        : PDF XChange Pro
[PDF]          Has XFA                         : No
[PDF]          Page Count                      : 1
[XMP-x]        XMP Toolkit                     : Image::ExifTool 12.41
[XMP-dc]       Format                          : application/pdf
[XMP-xmp]      Create Date                     : 2022:05:09 09:07:25+02:00
[XMP-xmp]      Modify Date                     : 2022:05:10 10:13:28+02:00
[XMP-xmpMM]    Document ID                     : uuid:9baa2a6d-3b7e-4038-ac0e-35b5bd3569fe
[XMP-xmpMM]    Instance ID                     : uuid:3eaba34d-0b27-44eb-9f3a-92b5dcd42a24
```

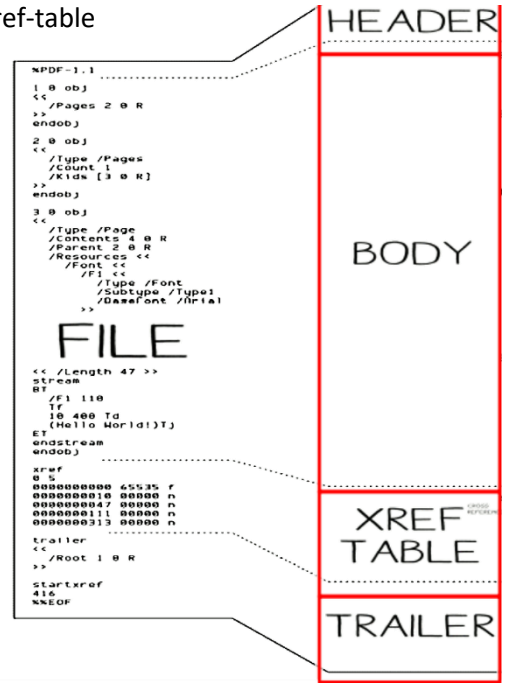
It should be noted that also metadata can be manipulated to hide modifications, so they are not a valid mean to verify the integrity of a file reliable.

Manipulating metadata with exiftools also leaves traces though: Exiftools uses it's own XMP toolkit which will also show up when metadata is investigated.

The PDF Format

To understand attacks over pdf files one must first have a basic understanding of the PDF Format.

Every PDF file consists of four parts; a header, a body, an xref-table and a trailer.



The header of PDFs is comparatively short, it starts with *%PDF-* followed by the version of pdf standard used by the file.

The body contains the content (and metadata) of the PDF as a list of objects. More on these objects later.

The xref-table, also called cross-reference table, lists the offset of each of the objects inside the file.

At the end of the file the trailer is located. It states where the root object (the first object rendered when opening the file) is located. It is identified by the name */Root*. Furthermore, the trailer contains the offset where the xref table starts so the reader can easily locate it.

Technically the PDF file is built from a graph of objects that instruct the PDF reader application what operations it must do to display the file content to the user. The following 8 object types are supported by the PDF standard:

1. Boolean – either *true* or *false*
2. Numeric – decimal or integer
3. String – sequence of characters between *()* or their hexadecimal values between <>
4. Name – sequence of characters with a */* before them
5. Array – sequence of objects between *[ ]*
6. Dictionary – map of key-value pairs, enclosed by *<< >>*
7. Stream – special object consisting of a dictionary and a sequence of data (typically, compressed text or images), introduced by keyword *stream*

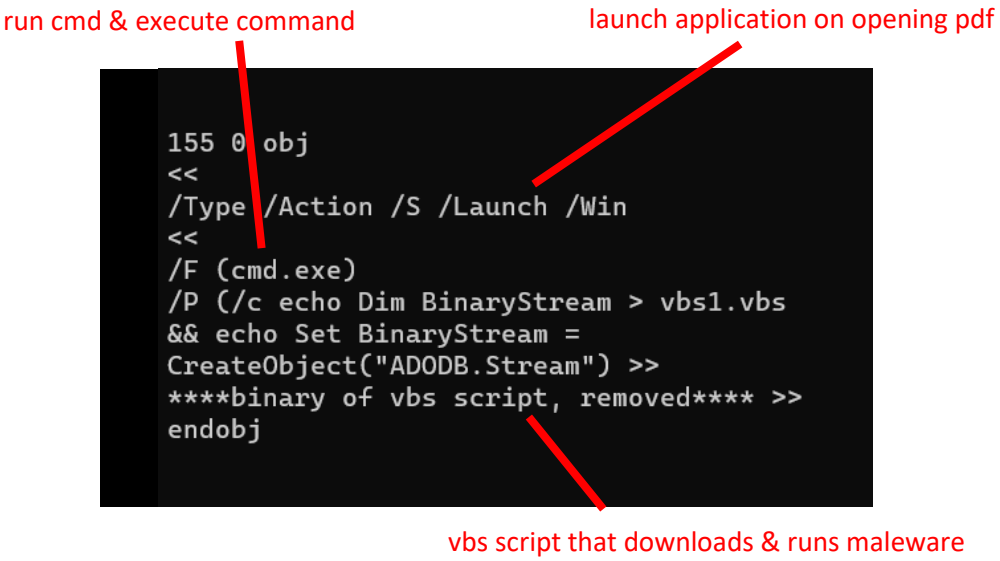
As you can see stream objects can contain all kinds of data, they aren't just limited to text and images. This is what makes PDF files so convenient for sharing all kinds of documents & information.
But it also is what opens up attackers a very large surface for their attacks. They mostly try to exploit vulnerabilities in a specific PDF reader.
When looking for such exploits in PDF files, investigators first search the PDF file for object names that are often used to exploit such vulnerabilities. This includes but is not limited to:

- **Embedded Javascript**: /JS, /JavaScript, /MacroForm and /XFA
- (**Embedded Flash**: */RichMedia* (Flash supports action script)) *support dropped!*
- **Launching:** */AA*, */Launch* and */OpenAction* (action upon opening the PDF)
- **Internet access:** */URI* and */SubmitForm*
- Embedded file: */EmbeddedFiles*

Looking at this list, a general rule of thumb is that any embedded scripts or files may be suspicious.
In fact malicious javascript code is a very common attack amongst PDF files. Another attack surface is exposed by the possibility to launch other applications on the PC.

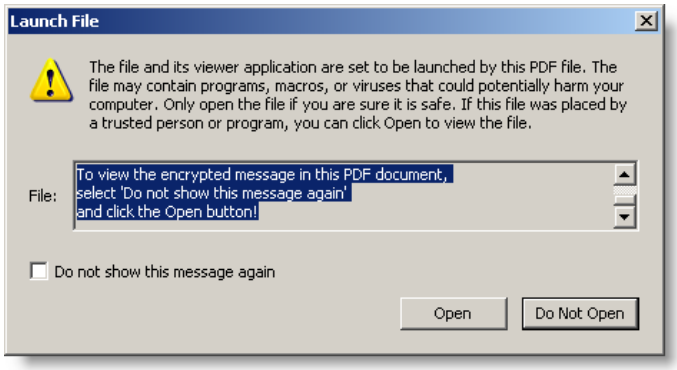
Example Attack

Let's have a look at a very simple but effective attack. It embeds the following object in a PDF file:



We can see a stream object. The names in the dictionary indicate an action that launches an application when the PDF is opened. In fact, the application that will be launched is the command shell. Upon opening the cmd, the cirtual basic script provided by the stream contained in the object is decoded and executed. In this attack scenario the script did download the Zeus trojan, a male ware running undetected in the background that spies for banking information and login data. It is also enabling the attacker to download and install other male ware without further notice by the user.

In general PDF readers show a dialog, asking the user for permission, before opening any file or running any script. But attackers were able to manipulate this dialogue and sneak in a message to manipulate the user's behavior:



More details on this attack can be found under CVE-2010-1240.

Phishing

As PDF files are cross platform and allow to engage with users, they are an enticing phishing vector for attackers. Most people think of emails and websites when they think about phishing but PDF's are just as vulnerable while being often even more convincing than a text based email with a link.

PDF files use many different schemes to trick users into clicking on links and buttons:

The 2020s phishing trends featured fake captchas embedded as image in the PDF. When clicked they redirect the user to an attacker-controlled website. These sites often disguise themselves through Homoglyphic Attacks and utilize cloned websites to lure users into logging in with their credentials. Often they also try to enable push notifications to trick the user to visiting the site again in the future. This way, users who are susceptible to such attacks can be attacked again and again, creating a steady revenue.

Other common phishing attacks in PDF files utilize the same teqchnique but with a different disguise. It can be a dropbox file embedded in the pdf that the user must open externally (requiring a login with dropbox credentials). Or just a video in the PDF that does redirect to a youtube clone ("please log in to your google account").

The possibilities are endless, and many users do not expect this from pdf sites.

A protective measure by many PDF readers is to not directly forward to the URL and instead ask the user for confirmation, showing him the url. But often PDFs are just read in the web browsers preview which mostly doesn't display such a dialog (tested it under firefox, chromium and edge).

PDF Scanning

The two most valuable tools for PDF scanning are pdfid and pdf-parser, both coming with kali-linux.

As they have already been covered by Halil Berisha in his talk *Forensics Tools and Forensics vs Pentesting* I'll not cover them again. His report can be found in moodle, see page 13ff.

Sources

Exiftool documentation: https://exiftool.org/

Pdfinfo documentation: https://www.xpdfreader.com/pdfinfo-man.html

2020 Phishing Trends with PDF Files:
https://fs.hs-ulm.de/public/schaefft/DIFO-Digital Forensics/ForensicsHacks/PDF Forensics

Davide Maiorca - Digital Investigation of PDF Files:
https://fs.hs-ulm.de/public/schaefft/DIFO-Digital Forensics/ForensicsHacks/PDF Forensics

Tho Le – PDF Forensiscs Introduction:
https://fs.hs-ulm.de/public/schaefft/DIFO-Digital Forensics/ForensicsHacks/PDF Forensics