PDF Analysis of Lokibot malware

July 25, 2022

Muhammad Hasan Ali

Malware Analysis learner

4 minute read

As-salamu Alaykum

Introducion

This sample is from Lokibot trojan which steals the credential information from web browsser, FTP server, SMTP server. This sample is a **PDF** file and our purpose of this blog is how to analyze a PDF file.

About PDF

Ability of a PDF file

A PDF file can impelemnt droppers, downloader, or exploit PDF reader application’s vulnerabilities.

PDF sttucture

- PDF header: Contains info about the version of the PDF such as %PDF-1.6
- Body:
  - Streams: a sequence of bytes such as images or data, which comes in encoded data.
  - Objects: How to render documanets which can include text or javascript.
  - Others such as names, dictionaries, strings, and arrays.
- Cross-reference table: contains the offsets of file’s objects.
- Trailer: contains the offset of xref table, and number of objects, /Root.

Dictionary entry is an item between « » and starts with slash / such as /Root which is the first object will be processed after loading the PDF file, /Root could be found in the Trailer section.

Suspecious keywords found when analyzing and their indications:

- /Js, /JavaScript: To execute embedded javascript
- /Launch, /EmbeddedFiles: To launch exeternal or embedded files
- /URI: To interact with URLs
- /OpenAction, /AA: To open an action
- /FlateDecode: uses the zlib/deflate decompression method.

A comment in PDF starts with %

About objects:

```
obj 1 0: % first number is ID, second number is version

type: catalog % catalog is an example, type can be empty.

Referencing: 3 0 R % object 1 0 refernces to 3 0, R indicates of referencing

........ % content of the object

endobj % the object ends with
```

For more info aboud PDF see this.

Methodology

use pdfid.py or peepdf.py:

> to perform an initial assessment by summarizing risky aspects

pdf-parser.py:

- to locate objects in file.pdf that include JavaScript
- to examine the contents of objects
- to decode the stream embedded from object
- to extract only the list of URL
- Follow object referencing to find the goal.

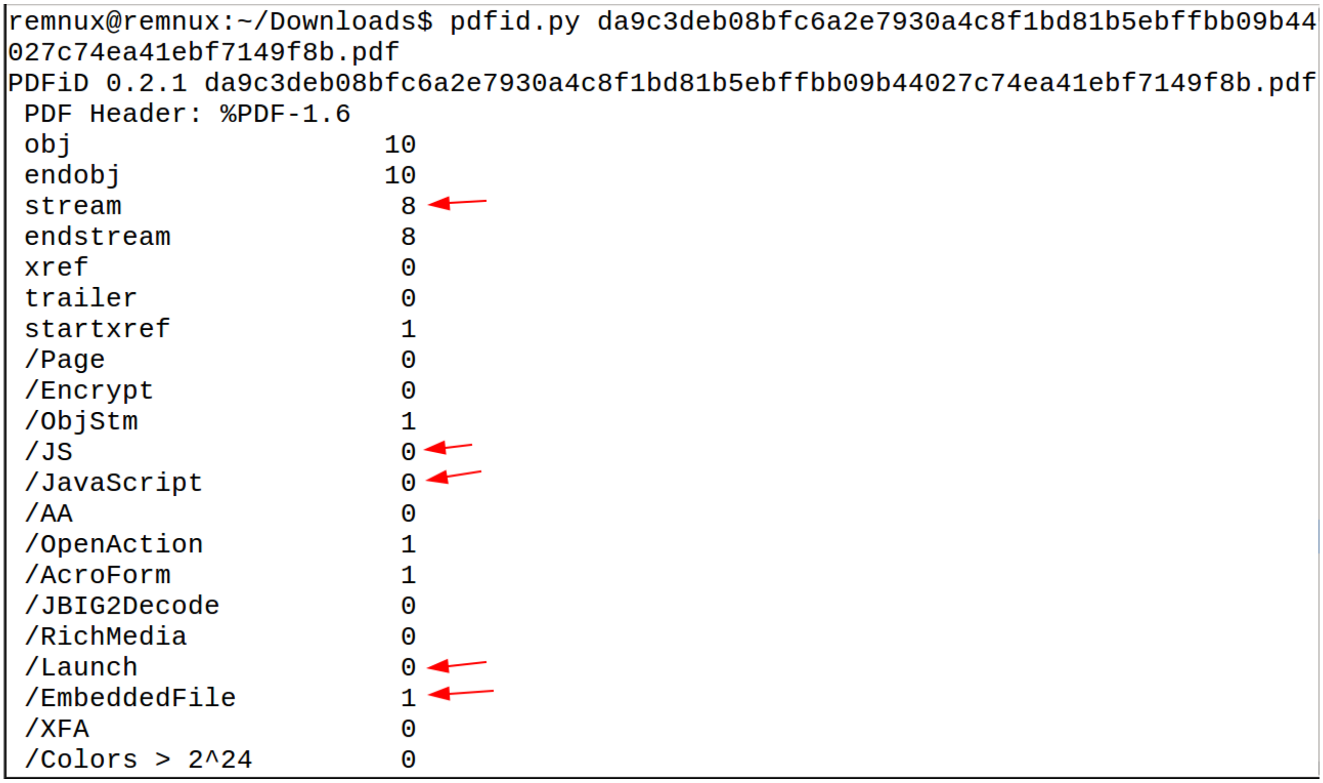
If you use peepdf.py and found that it has /EmbeddedFiles, start analyzing the object where is /EmbeddedFiles belongs to.

If you find /FlateDecode, go and try to analyze it which decodes stream.

PDF analysis

In this sample, We received a malicious PDF file which downloads Lokibot malware. So we need to start our analysis quickly using REMnux.

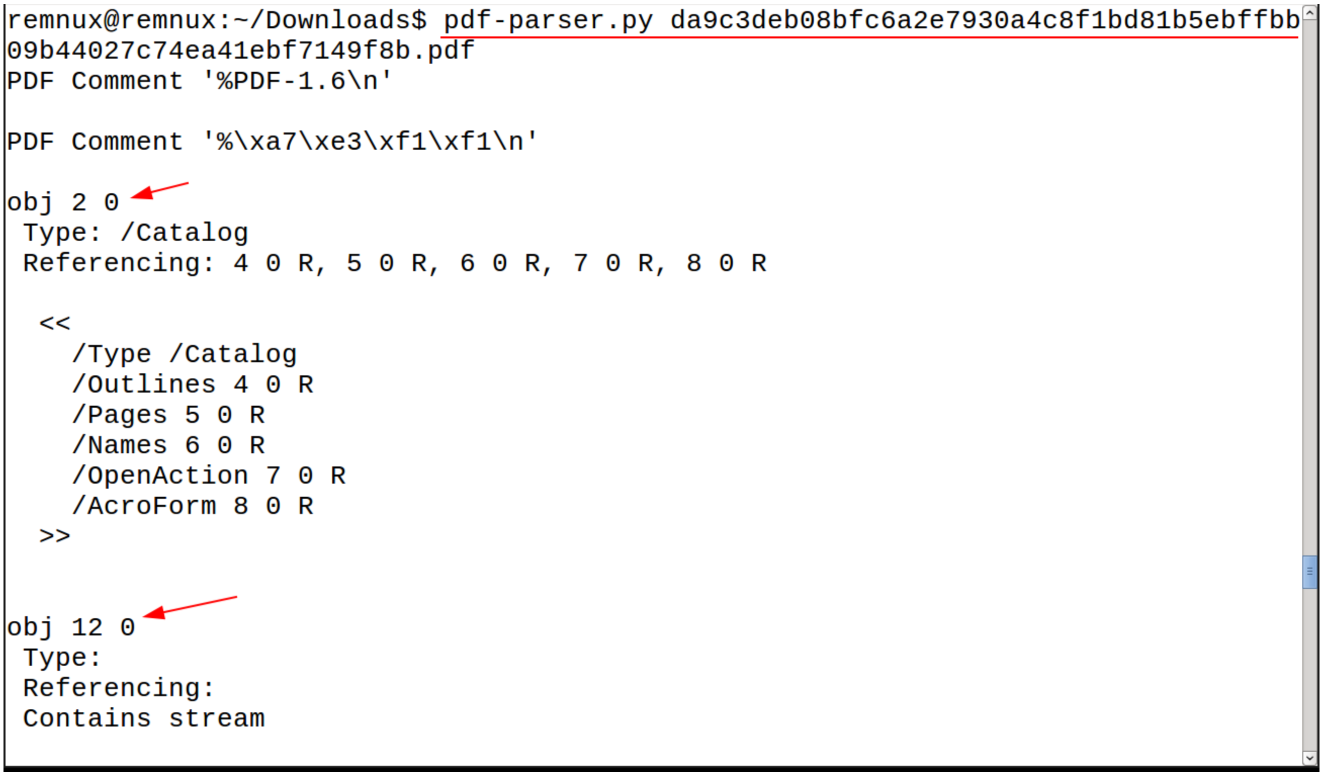
We first use `pdfid.py` to get info about the PDF and what is there. As we see, it has 8 streams and 1 /EmbeddedFiles and 0 javascript files. We can use `peepdf.py` to get which object contains the /EmbeddedFiles but an error occured running.



```
remnux@remnux:~/Downloads$ pdfid.py da9c3deb08bfc6a2e7930a4c8f1bd81b5ebffbb09b44
027c74ea41ebf7149f8b.pdf
PDFiD 0.2.1 da9c3deb08bfc6a2e7930a4c8f1bd81b5ebffbb09b44027c74ea41ebf7149f8b.pdf
 PDF Header: %PDF-1.6
 obj                    10
 endobj                 10
 stream                  8
 endstream               8
 xref                    0
 trailer                 0
 startxref               1
 /Page                   0
 /Encrypt                0
 /ObjStm                 1
 /JS                     0
 /JavaScript             0
 /AA                     0
 /OpenAction             1
 /AcroForm               1
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /EmbeddedFile           1
 /XFA                    0
 /Colors > 2^24          0
```

Figure(1): pdfid.py output

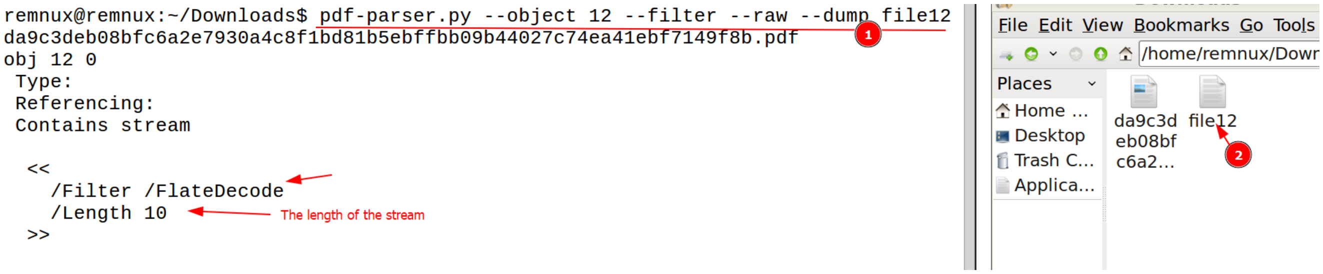
So we will use `pdf-parser.py` and to get our embedded file. We see many objects, Then start with objects which contains /FlateDecode and if we found /EmbeddedFiles go for it.



```
remnux@remnux:~/Downloads$ pdf-parser.py da9c3deb08bfc6a2e7930a4c8f1bd81b5ebffbb
09b44027c74ea41ebf7149f8b.pdf
PDF Comment '%PDF-1.6\n'

PDF Comment '%\xa7\xe3\xf1\xf1\n'

obj 2 0
 Type: /Catalog
 Referencing: 4 0 R, 5 0 R, 6 0 R, 7 0 R, 8 0 R

  <<
    /Type /Catalog
    /Outlines 4 0 R
    /Pages 5 0 R
    /Names 6 0 R
    /OpenAction 7 0 R
    /AcroForm 8 0 R
  >>


obj 12 0
 Type:
 Referencing:
 Contains stream
```

Figure(2): pdf-parser.py output

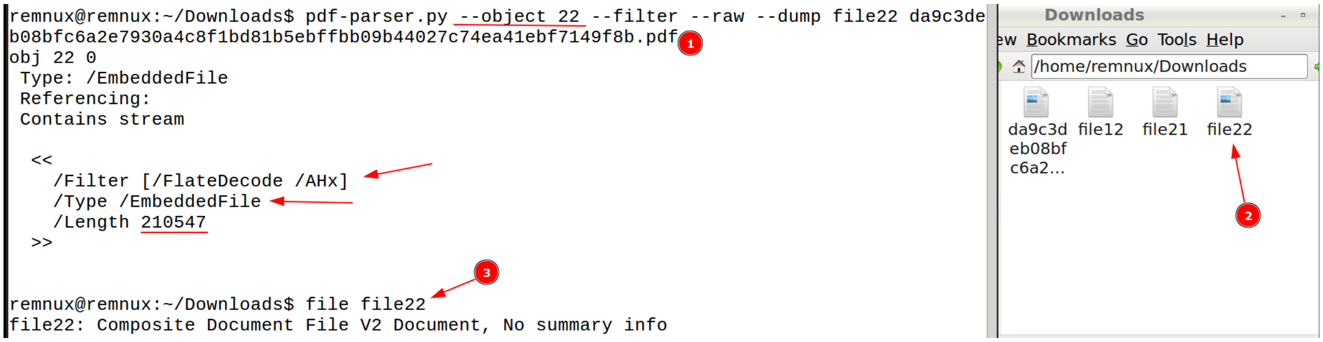
After scrolling down, we see object 12 conatins /FlateDecode. We try to decode it and dumping using



```
remnux@remnux:~/Downloads$ pdf-parser.py --object 12 --filter --raw --dump file12
da9c3deb08bfc6a2e7930a4c8f1bd81b5ebffbb09b44027c74ea41ebf7149f8b.pdf
obj 12 0
 Type:
 Referencing:
 Contains stream

  <<
    /Filter /FlateDecode
    /Length 10
  >>
```

Figure(3): After dumping the object 12

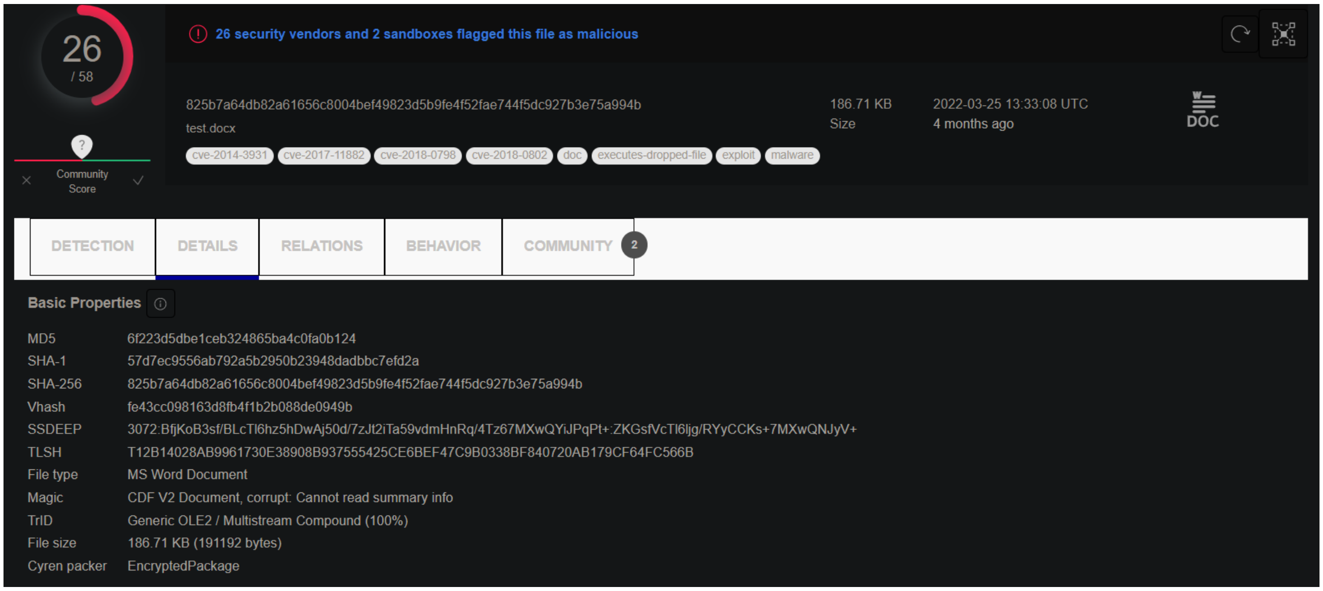
If we use `file` command to see its type, it's an ASCII text. Then we open `file12` using `scite` we it's useless. Some objects are useles, it takes time to find the peyload. We examine another object. When we get to object `22`, we our /EmbeddedFiles which is an indicator to that the PDF launches embedded file which has big length. Dump it to `file22` to see it's content and its type. After that we use file command, we notice that it's **Composite Document File V2 Document** CFBF is a compound document file format for storing numerous files and streams within a single file on a disk. In our case, this PDF stores an XLS file.



```
remnux@remnux:~/Downloads$ pdf-parser.py --object 22 --filter --raw --dump file22 da9c3de
b08bfc6a2e7930a4c8f1bd81b5ebffbb09b44027c74ea41ebf7149f8b.pdf
obj 22 0
 Type: /EmbeddedFile
 Referencing:
 Contains stream

  <<
    /Filter [/FlateDecode /AHx]
    /Type /EmbeddedFile
    /Length 210547
  >>

remnux@remnux:~/Downloads$ file file22
file22: Composite Document File V2 Document, No summary info
```

Figure(4): After dumping the object 22 and it's an xls excel spreadsheet

If we uploaded file22 to Virustotal we will find it already uploaded and it's malicous. Our purpose is to get the main payload and that's it.



26 security vendors and 2 sandboxes flagged this file as malicious

825b7a64db82a61656c8004bef49823d5b9fe4f52fae744f5dc927b3e75a994b
test.docx
cve-2014-3931 cve-2017-11882 cve-2018-0798 cve-2018-0802 doc executes-dropped-file exploit malware

186.71 KB
Size

2022-03-25 13:33:08 UTC
4 months ago

Basic Properties

MD5	6f223d5dbe1ceb324865ba4c0fa0b124
SHA-1	57d7ec9556ab792a5b2950b23948dadbbc7efd2a
SHA-256	825b7a64db82a61656c8004bef49823d5b9fe4f52fae744f5dc927b3e75a994b
Vhash	fe43cc098163d8fb4f1b2b088de0949b
SSDEEP	3072:BfjKoB3sf/BLcTl6hz5hDwAj50d/7zJt2iTa59vdmHnRq/4Tz67MXwQYiJPqPt+:ZKGsfVcTl6ljg/RYyCCKs+7MXwQNJyV+
TLSH	T12B14028AB9961730E38908B937555425CE6BEF47C9B0338BF840720AB179CF64FC566B
File type	MS Word Document
Magic	CDF V2 Document, corrupt: Cannot read summary info
TrID	Generic OLE2 / Multistream Compound (100%)
File size	186.71 KB (191192 bytes)
Cyren packer	EncryptedPackage

Figure(5): Virustotal analysis of xls dumped from PDF

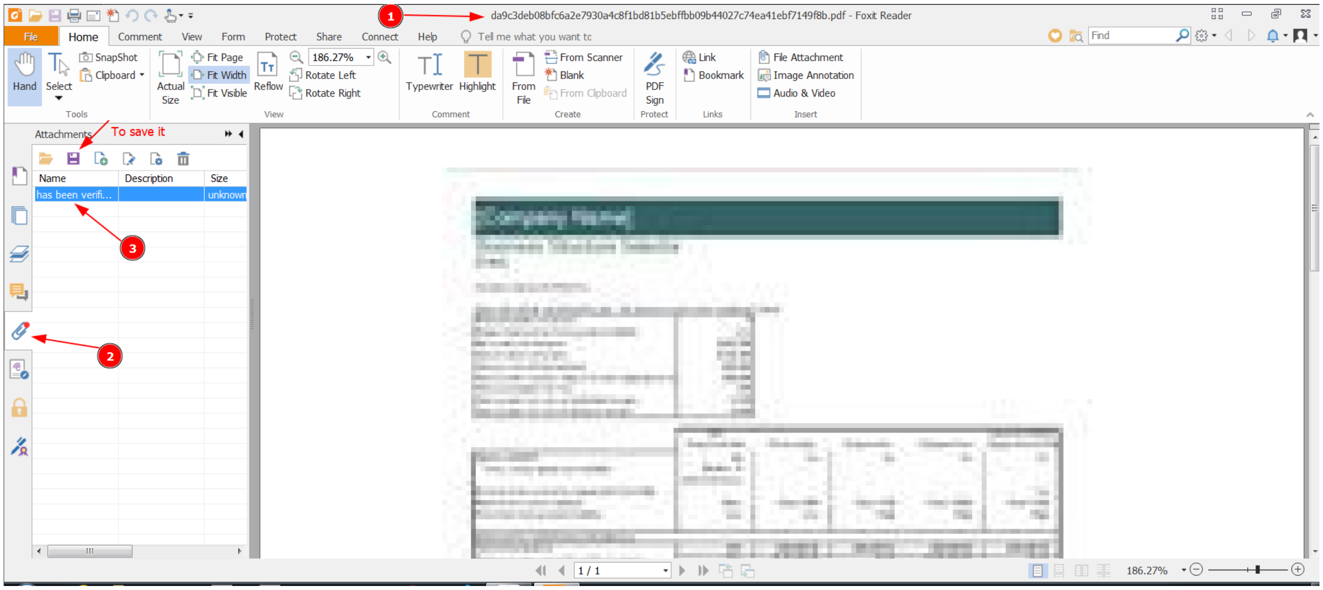
Dynamic analysis

We will open FlareVM which has our installed tools. We need to install PDF reader such as Foxit reader, and Microsoft office.

First, open `fakenet-ng` , if the malicious PDF tries to connect and download from internet, this PDF sample opens an xls spredsheet.

Then open the PDF. In foxit reader, disable safe mode and run the malicious PDF in privilage mode.

We open `Attachments` , we see there’s an attachment which will be our `xls` spredsheet file. You can open it manually. Double click on it and allow to open it. Then it will open an xls excel spredsheet. **Save** this attachment on your `Desktop` from foxit reader as shown.



Figure(6): When opening the PDF

IoCs

PDF file: `da9c3deb08bfc6a2e7930a4c8f1bd81b5ebffbb09b44027c74ea41ebf7149f8b`

xls sheet: `825b7a64db82a61656c8004bef49823d5b9fe4f52fae744f5dc927b3e75a994b`

Article quote

> إلهي ، ماذا وجد من فقدك وما الذي فقد من وجدك

REF

- hybrid-analysis