Digital Forensics: Assignment 1

Part 1

Use text editor to see the pdf file.

The pdf has JavaScript that is called to export a file name “eicar-dropper.doc”.

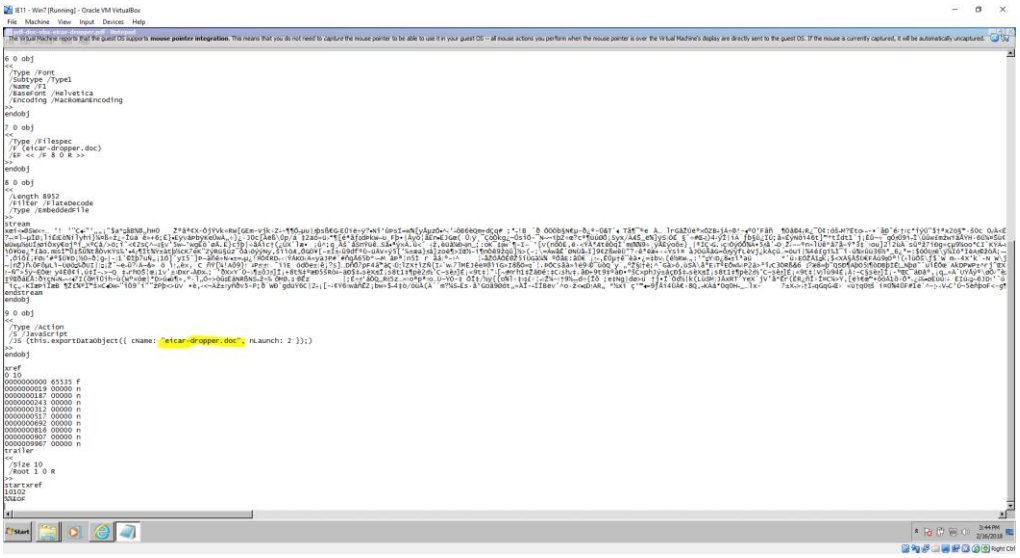

Fig 1.1 Investigate pdf with text editor

In the doc file, use the text editor to see it content. The VBA script part show the function “AutoOpen” which means the script will run automatically. In side the AutoOpen, I found function MsgBox with contain “EICAR test ..”. It means when the file is open it will automatically show the message box with “EICAR test” as a message.

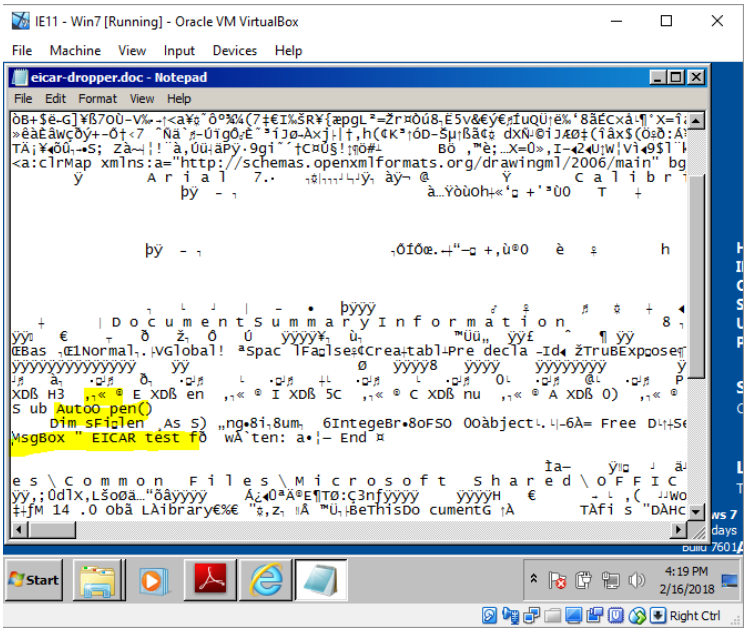

Fig 1.2 Investigate doc file with text editor

Part 3

Analyze malicious file from blackboard system
(Malicious Sample Files\Assignment 1 Malicious PDF File\Happy Valentine’s Day PDF File)

This analysis ran on REMnux virtual machine.

Q1 Report the number of objects in the file.

By using pdfid.py, I found 6 objects are in the pdf file.

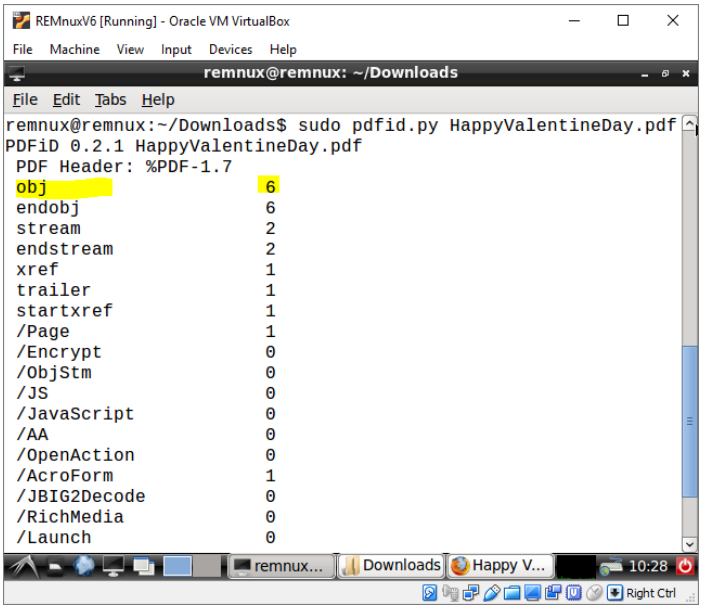

Fig.3.1 pdfid.py shows number of objects

Q2 Determine whether the file is compressed or not.

It has 2 fl decode in the object one. So it is compressed.

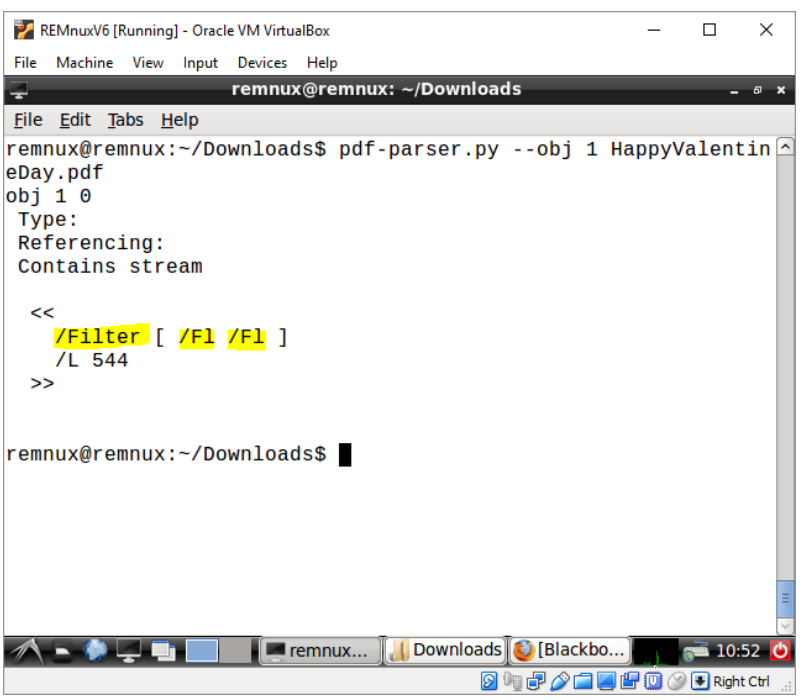

Fig.3.2 pdf-parser.py shows that it used Filter

Q3 Determine whether the file is obfuscated or not.

By using peepdf the result shown in Fig 3.3. The object [1] is Javascript and had been encoded (obfuscated).

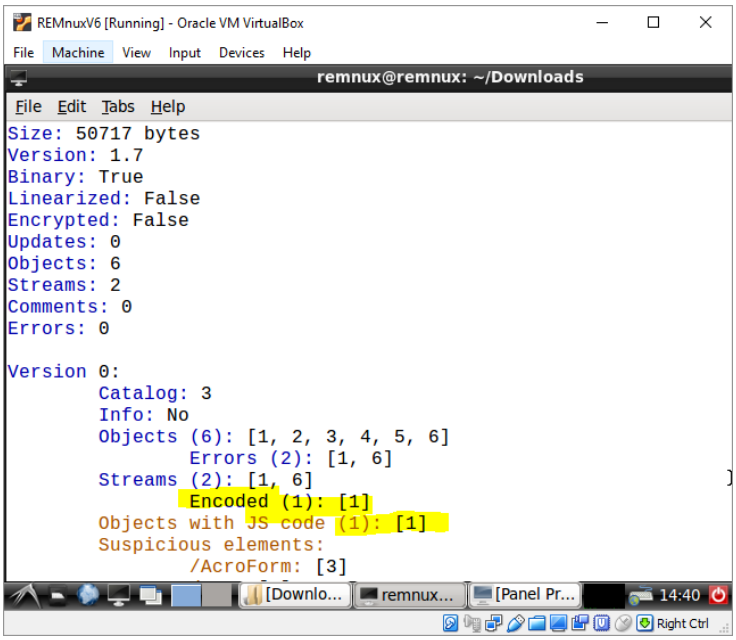

Fig. 3.3 The file is encoded (obfuscated)

Q4 Find and Extract Javascript

As shown in Fig. 3.4, I used pdf-parser to extract object [1] which is suspected and contain Javascript.

After look though the extracted file, the javascript code was found as shown in Fig. 3.5

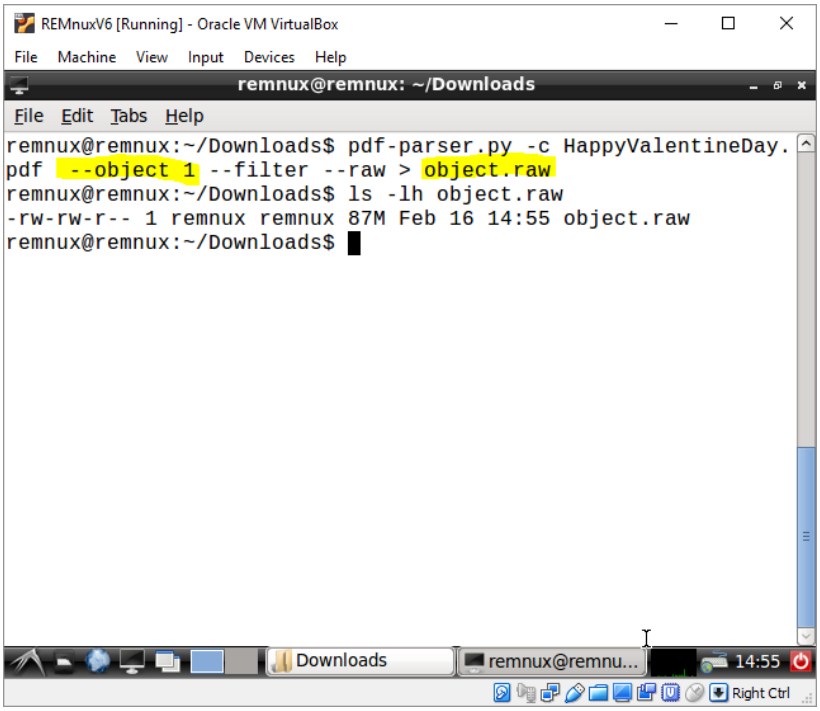

Fig. 3.4 Using pdf-parser to extract object [1]

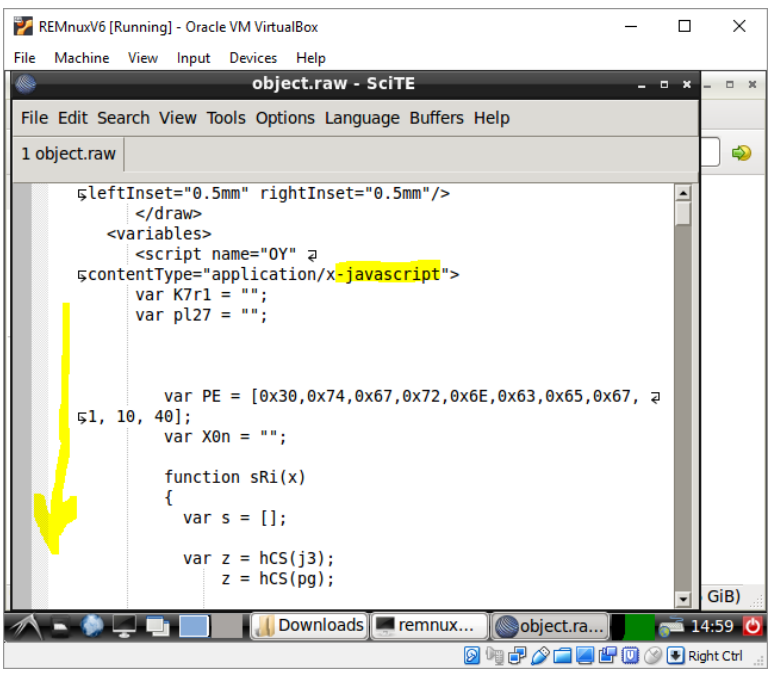

Fig. 3.5 Javascript found in extracted object 1

Q5 De-obfuscate JavaScript

The Javascript are separated obfuscate in to 4 part. I tried to put every part together.

The tool that used to de-obfuscate this JavaScript is SpiderMonkey. However it require manual code fixing, for example changing '$lt;' to '<' and '$gt;' to >. After manually fixing the code was still cause errors and SpiderMonkey can not compile the JavaScript.

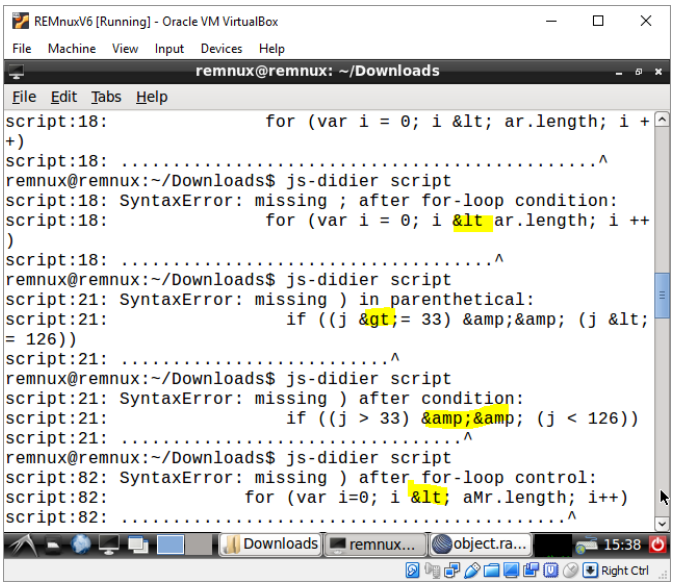

Fig. 3.6 Manual edit is required on obfuscate Javascript

Q6 – Q7

The file is the same as provided on

https://countuponsecurity.com/2014/09/22/malicious-documents-pdf-analysis-in-5-steps/

The author did some manual work to edit the JavaScript code but didn't provide the detail.

I tried to fix the code but it still can't run on SpiderMonkey and can't generate the SHELL code.

Q8 Analyze what it does

From the result of peepdf, it shows suspicious elements. One of them is BMP/RLE heap corruption (CVE-2013-2719) which allow attackers to execute arbitrary code or cause a denial of service (memory corruption) via unspecified vectors.

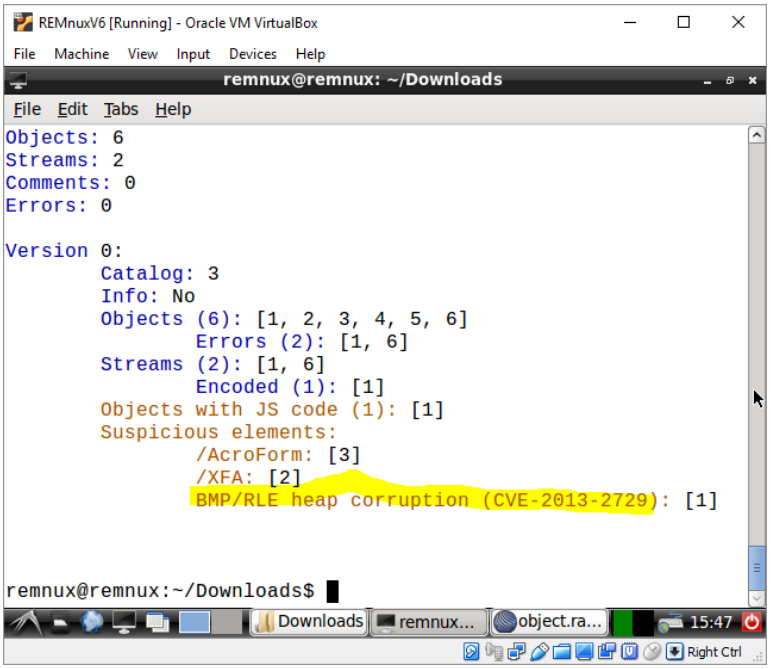

Fig. 3.7 peepdf shows suspicious elements