

Malicious PDF File Analysis - No. 2

# PDF analysis

1. Report the number of objects in the file - 18
  2. Determine whether the file is compressed or not. - NO
  3. Determine whether the file is obfuscated or not. - YES
  4. Find and Extract JavaScript.

**14 and 15 Objects have js**, JavaScript extracted by the peepdf.py extract command is given below

```
this.exportDataObject({  
    cName: "sample",  
    nLaunch: 0  
});
```

## 5. De-obfuscate JavaScript.

By using command –raw –filter the obfuscated js is stored in shellcode.js

The screenshot shows a REMnux terminal window titled "remnux@remnux: ~/Downloads/peepdf-master". The user is executing a command to parse a PDF file and generate shellcode.js:

```
remnux@remnux:~/Downloads/peepdf-master$ ./peepdf.py --object 14 --raw --filter DigitalForensicsHW1.pdf > shellcode.js
```

The terminal output displays the generated shellcode.js file, which is extremely long and contains many null bytes (0x00). The code is designed to be executed in a DOS mode environment, as indicated by the error message:

```
This program cannot be run in DOS mode. Run it in MS-DOS mode.
```

The bottom of the terminal shows the file size and transfer progress:

```
remnux@remnux:~/Downloads/peepdf-master [Send Anywhere - File transfer --] 1 / 2
```

## 6. Extract the shell code.

The process of extracting shellcode is via following steps

## 1. Convert to Unicode format

```
cat object.raw | tr '\' '%' > object.unicode
```

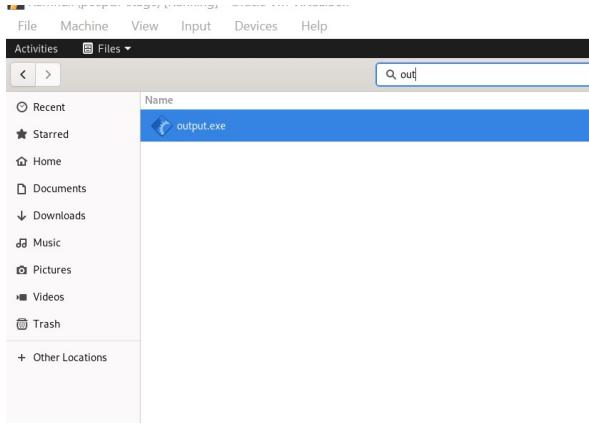
## 2. Convert the Unicode to hex

`unicode2hex-escaped < object.unicode > object.hex`

## 7. Create a shell code executable

Convert Hex file from the previous step to shell code

```
shellcode2exe -s object.hex
```

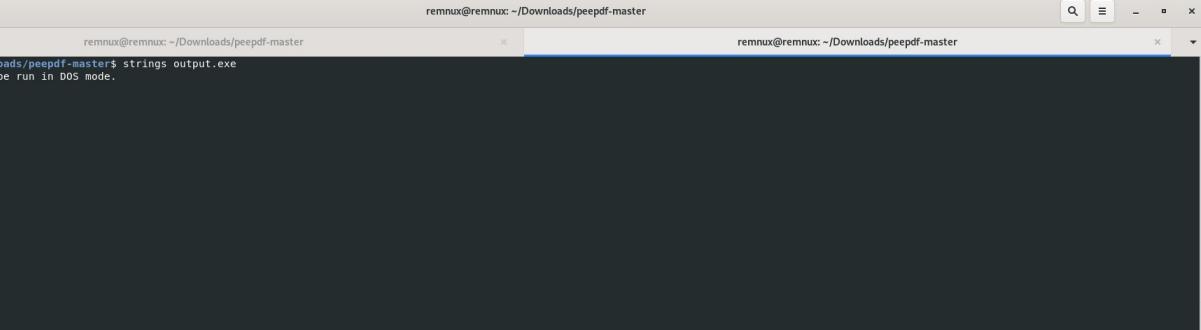


8. Analyze shell code and determine what it does or even execute it using sctest or spider monkey.

## 1. Running PDF stream dumper

Screenshot below of the analysis

2. Screenshot below of the hex code converted into exe files  
And a strings were extracted to print out strings in the exe file



remnux@remnux: ~/Downloads/peepdf-master\$ strings output.exe  
!This program cannot be run in DOS mode.  
0^C  
.text  
.idata  
#zx5ffM  
fFE34e  
k3fe0C  
Af55\$  
[ITD  
e4L\_4  
EE78lu  
EE96  
fBE%  
2d6m

3. After Printing out the strings we check for emulation of the exe
  4. Running scdb test to emulate shell commands and found below signatures  
*generic.hll.prolog.1 and generic.hll.prolog.3*  
And no memory monitor logs were found.

**Screenshot below**

REMnux (peepdf-stage) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Oct 7 16:00

remnux@remnux: ~/Downloads/peepdf-master

remnux@remnux: ~/Downloads/peepdf-master

```
Stepcount 1616893

Analysis report:
    Uses peb.InMemoryOrder List

Signatures Found:
    40101c  hasher.harmony

Memory Monitor Log:
    *PEB (f53b) accessed at 0x40100b
    peb.InMemoryOrderModuleList accessed at 0x401012

Z:\opt\scdbg>Loaded 199f bytes from file Z:\home\remnux\DOWN-NTG\DIGI-CR1.SC
Memory monitor enabled...
Initialization Complete...
Interactive Hooks enabled
Max Steps: 2000000
Using base offset: 0x401000

0      opcode 6f not supported

0  ???? No memory At Address           step: 0  foffset: 0
eax=0          ecx=0     edx=0          ebx=0
esp=12fe00    ebp=12ffff0   esi=0          edi=0      EFL 0

$Stepcount 0

Analysis report:
Signatures Found:
    404cc1  generic.dll.prolog.1
    405247  generic.dll.prolog.3

Memory Monitor Log:

Z:\opt\scdbg>
```

9. What is the secret code? - SKY IS RED

Secret code is embedded via Metasploit launch message and the exploit must be  
**exploit/windows/fileformat/adobe\_pdf\_embedded\_exe**

```
15 0 obj
<</S>/JavaScript/J(S(this.exportDataObject({ cName: "sample
endobj
16 0 obj
<</S>Launch/Type/Action/Win<</F(cmd.exe)/D(c:\\windows'
```

```
SKY IS RED)>>>
endobj
```