

CS 6345 – Digital Forensics
Assignment-1
Stage-2

Sai Meghana Akula-R11773118
Ravi Sankar Gogineni-R117788968
Venkata Veera Siva Dasari-R11716467

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:

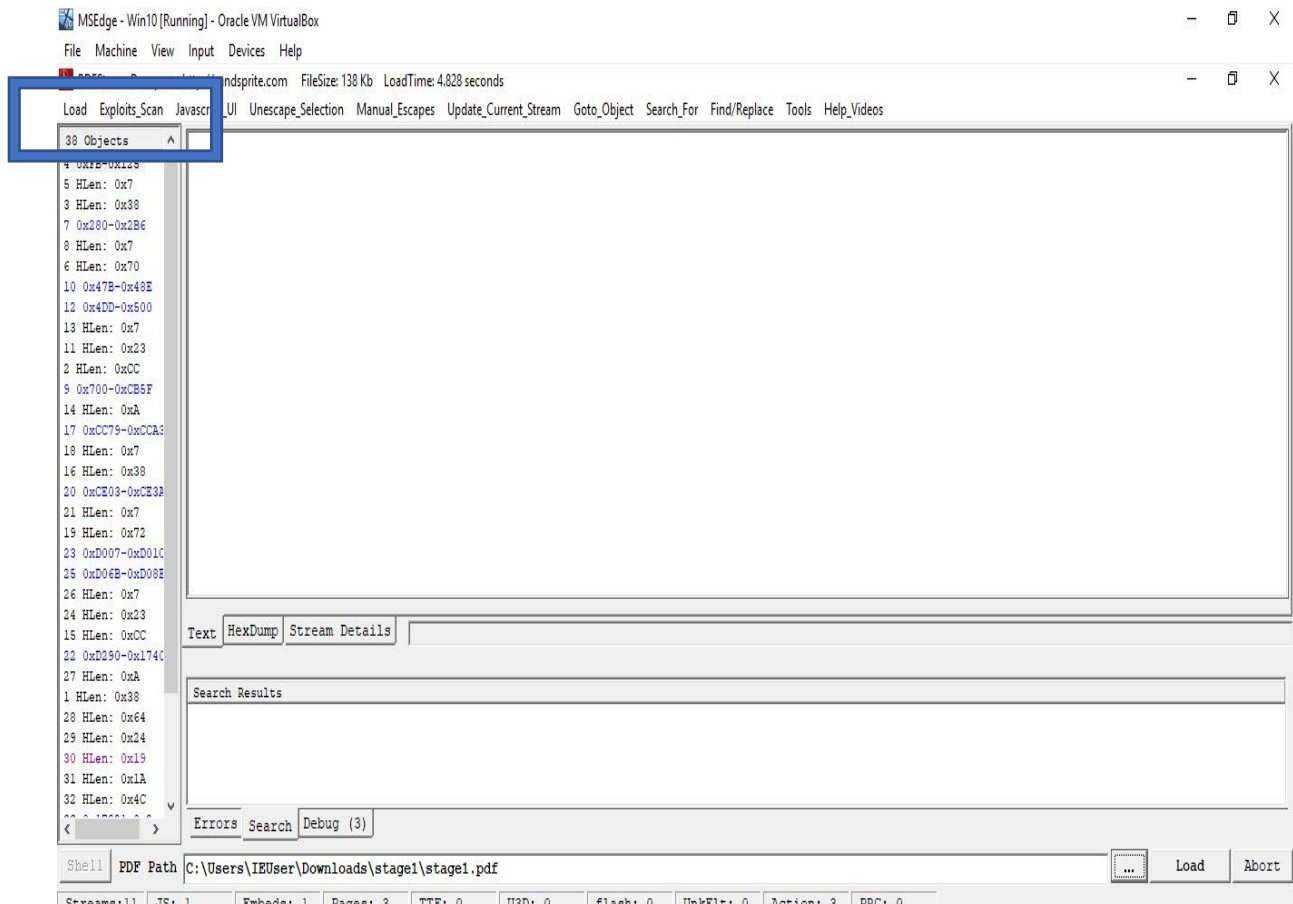
1. Report the number of objects in the file.
2. Determine whether the file is compressed or not.
3. Determine whether the file is obfuscated or not.
4. Find and Extract JavaScript.
5. De-obfuscate JavaScript.
6. Extract the shell code.
7. Create a shell code executable
8. Analyze shell code and determine what it does or even execute it using sctest or spider monkey.
9. What is the secret code?

Answers:

The stage 2 of the assignment is executed using pdf stream dumper.

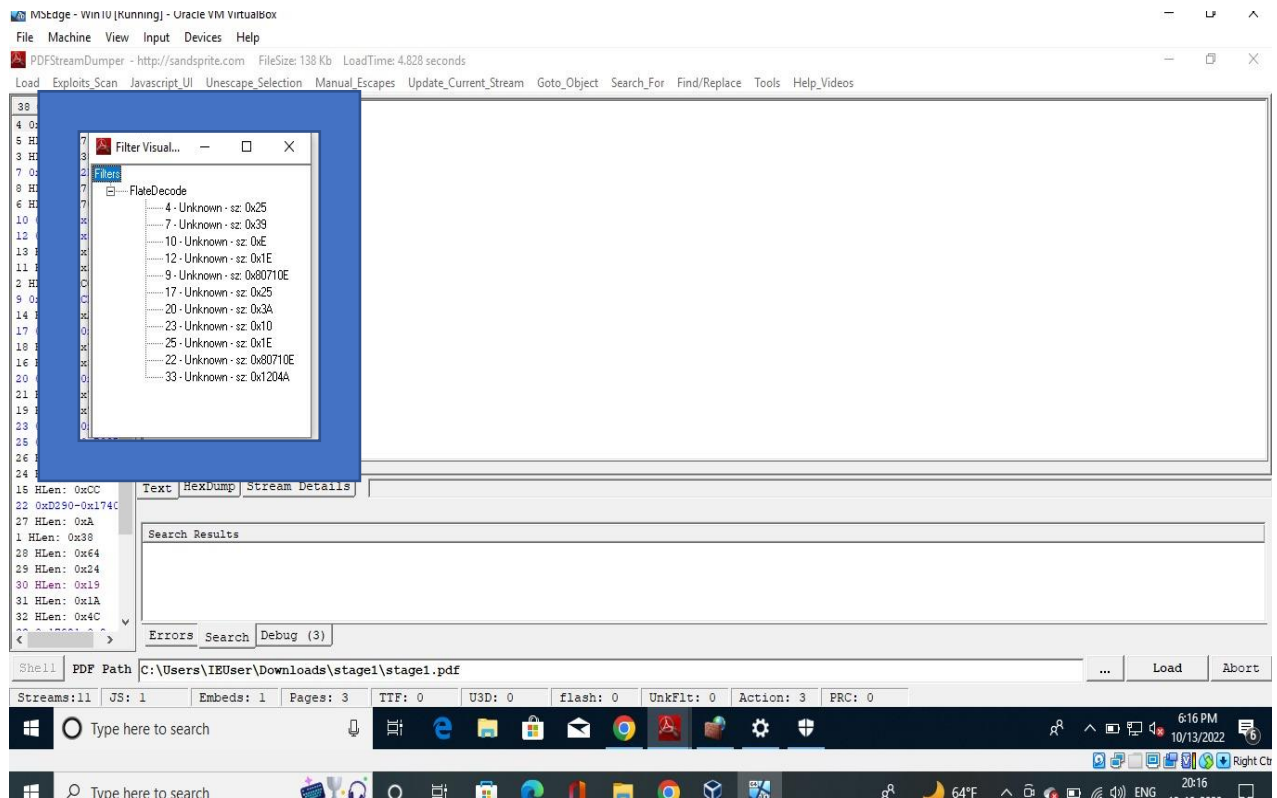
1. Report the number of objects in the file.

Answer: The number of objects in the file are **38**



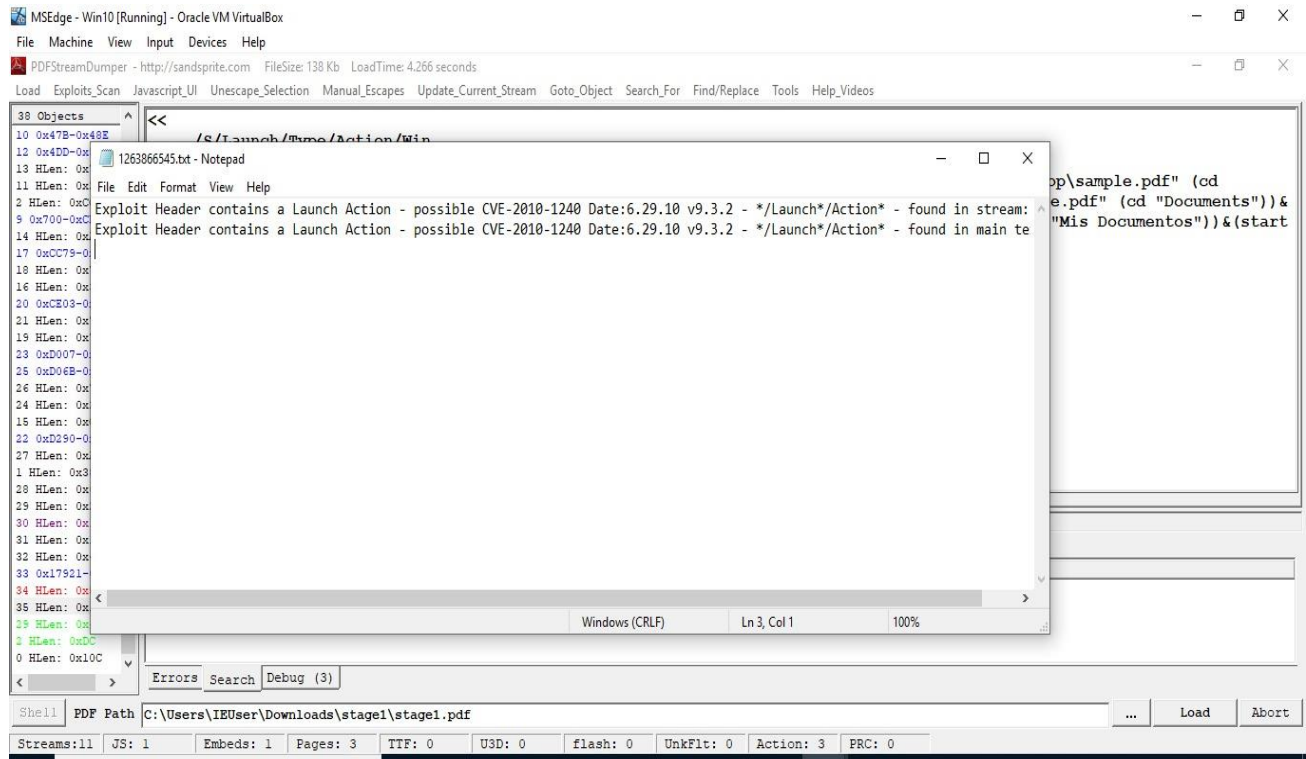
2. Determine whether the file is compressed or not.

Answer: Yes, the file is compressed. As there are filters in the given pdf file (analyzed using pdf stream dumper), we can say that the file is compressed.



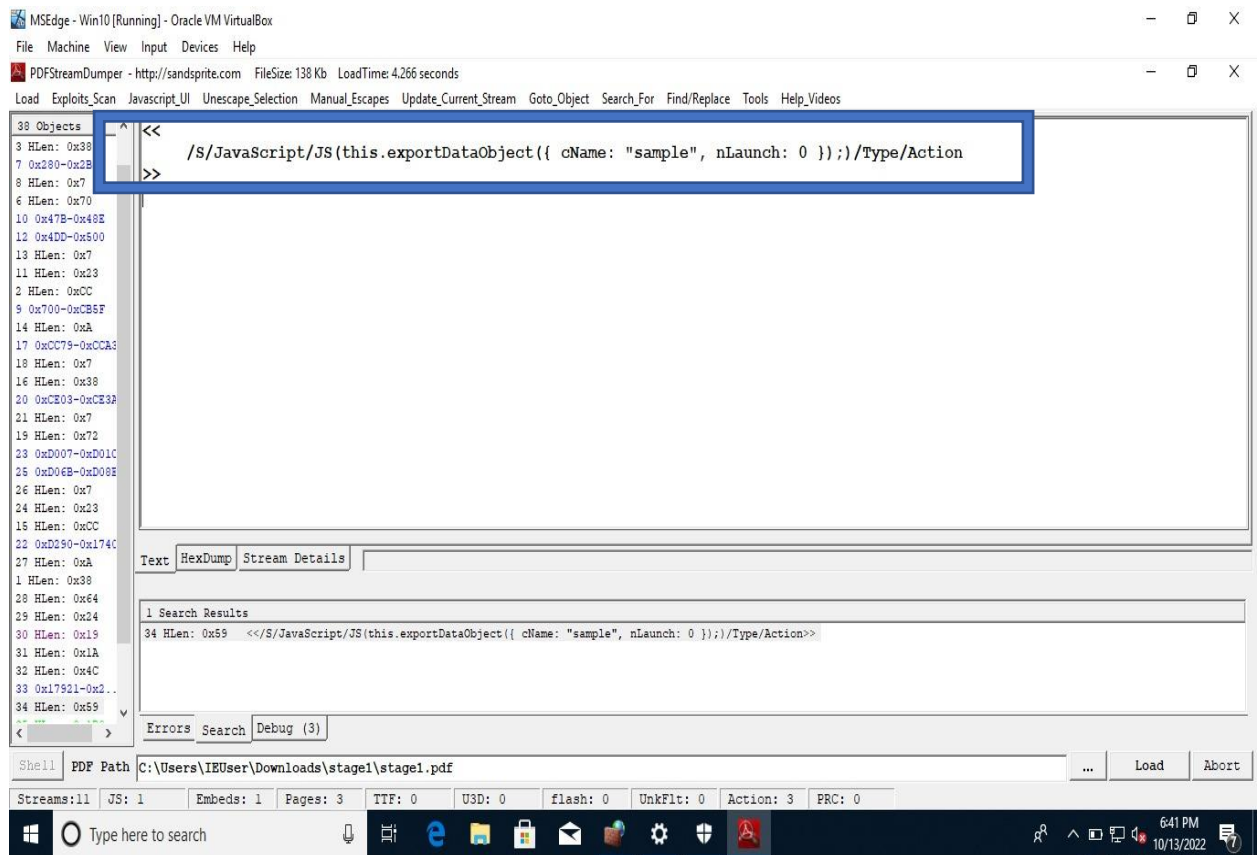
3. Determine whether the file is obfuscated or not.

Answer: Yes, the file is obfuscated. As the header contains the launch action, we can write that the file is obfuscated.



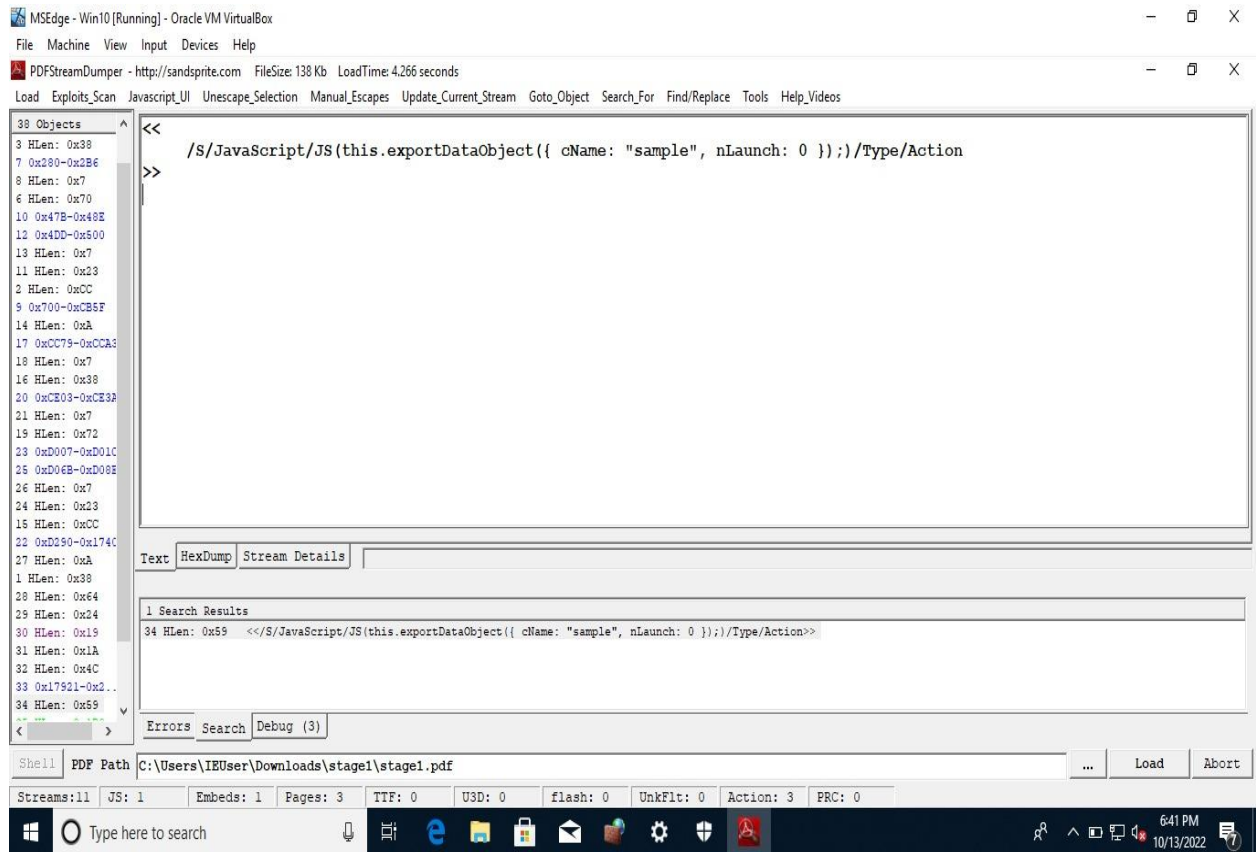
4. Find and Extract JavaScript.

Answer: The extract JavaScript is shown below.



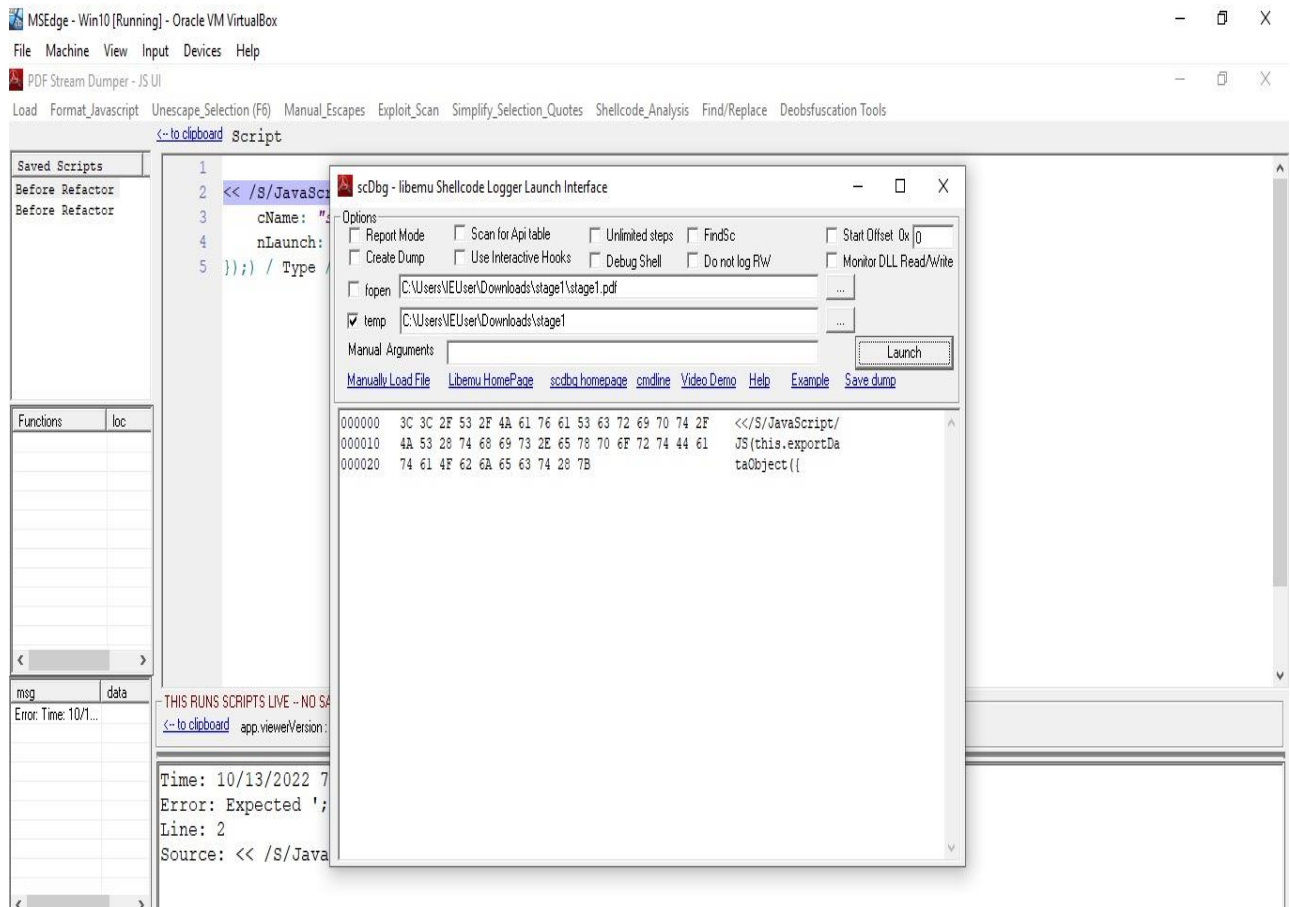
5. De-obfuscate JavaScript.

Answer: The De-obfuscated JavaScript is shown below:



6. Extract the shell code.

Answer: The extracted shell code is:



MSEdge - Win10 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Select C:\Windows\SYSTEM32\cmd.exe

```
/i          enable interactive hooks (file and network)
/las int    log at step ex. -las 100
/laa hexnum log at address or api ex. -laa 0x401020 or -laa ReadFile
/lookup api shows the address of WinAPI function ex. -lookup GetProcAddress
/mm         enabled Memory Monitor (logs access to key addresses)
/mdll       Monitor Dll - log direct access to dll memory (hook detection/patches)
/min steps  min number of steps (decimal) to trigger record in findsc mode (def 200)
/nc         no color (if using sending output to other apps)
/noseh      Disables support for seh and UnhandledExceptionFilter
/norw       Disables display of read/write file hooks
/o hexnum   base offset to use (default: 0x401000)
/patch fpath load patch file <fpath> into libemu memory
/r          show analysis report at end of run (includes -mm)
/redirect ip:port redirect connect to ip (port optional)
/s int      max number of steps to run (def=2000000, -1 unlimited)
/sigs       show signatures (can be used with -disasm)
/t int      MS to delay between steps (v1-2) or api (v0)
/temp folder use folder as temp path for interactive mode file writes
/u          unlimited steps (same as -s -1)
/v          verbosity, can be used up to 4 times, ex. /v /v /vv
/- /+       increments or decrements GetFileSize, can be used multiple times
/va 0xBase-0xSize VirtualAlloc memory at 0xBase of 0xSize
/raw 0xBase-fpath Raw Patch Mode: load fpath into mem at 0xBase (not PE aware)
/llo dllName-0xBase LoadLibrary Override: returns 0xBase for LoadLibrary/GetModuleHandle
/wint 0xBase-0xVal Write 32bit integer 0xValue at 0xBase
/wstr 0xBase-Str Write string at base ex. 0x401000-0x9090EB15CCBB or "0xBase-ascii string"
/dllmap     show the name, base, size, and version of all built in dlls
/nofile     assumes you have loaded shellcode manually with -raw, -wstr, or -wint
/bswap      byte swaps -f and -wstr input buffers
/eswap      endian swaps -f and -wstr input buffers
/conv path  outputs converted shellcode to file (%u,%x,bswap,eswap..)
/ida        connects to last opened IDA instance on startup
/[reg] value sets init register value ex: -eax 0x20 -ebx 20 -ecx base -reg base

in the dbg> shell enter ? to see supported commands
```

MSEdge - Win10 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

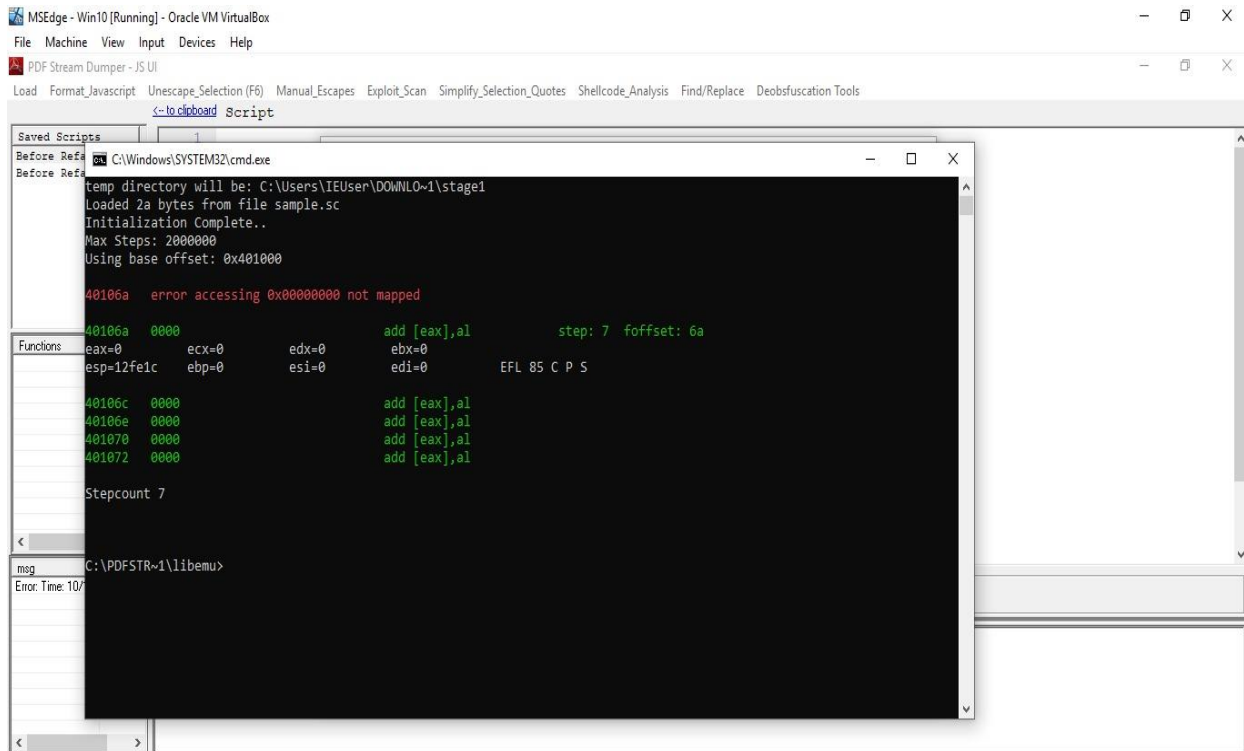
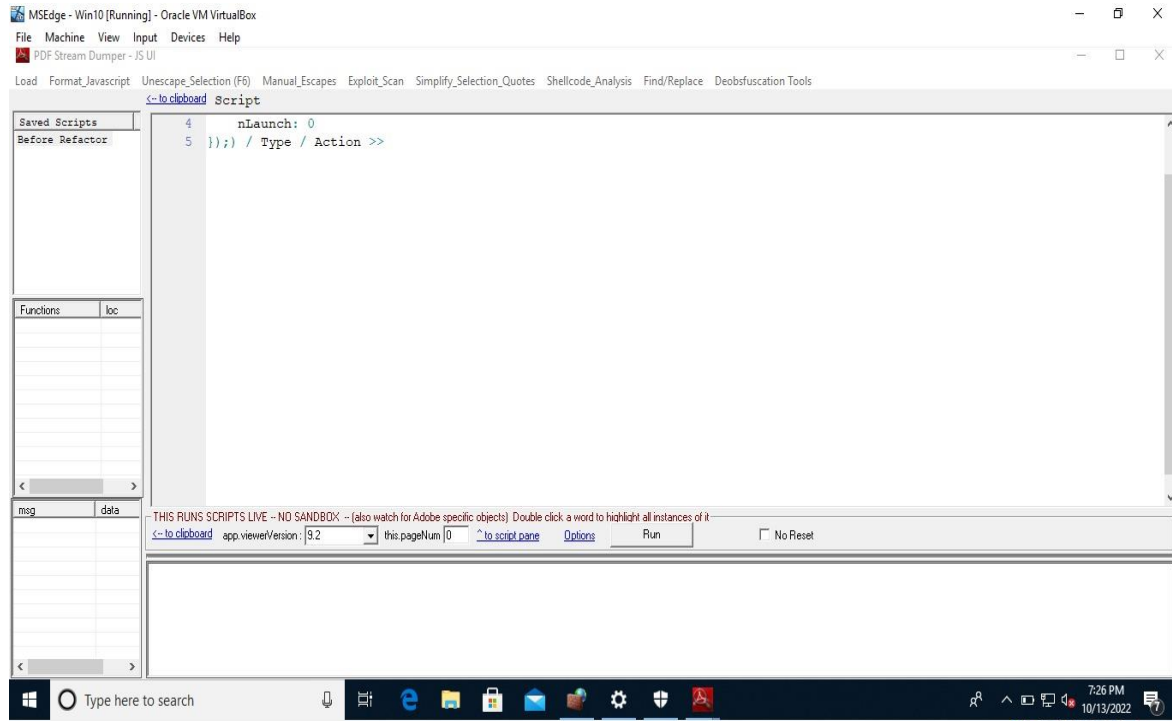
Select C:\Windows\SYSTEM32\cmd.exe

```
sctdbg is an adaption of the libemu library and sctest project
Libemu Copyright (C) 2007 Paul Baecher & Markus Koetter
sctdbg developer: David Zimmer <dzzie@yahoo.com>
Compile date: Jun 29 2015 13:05:50

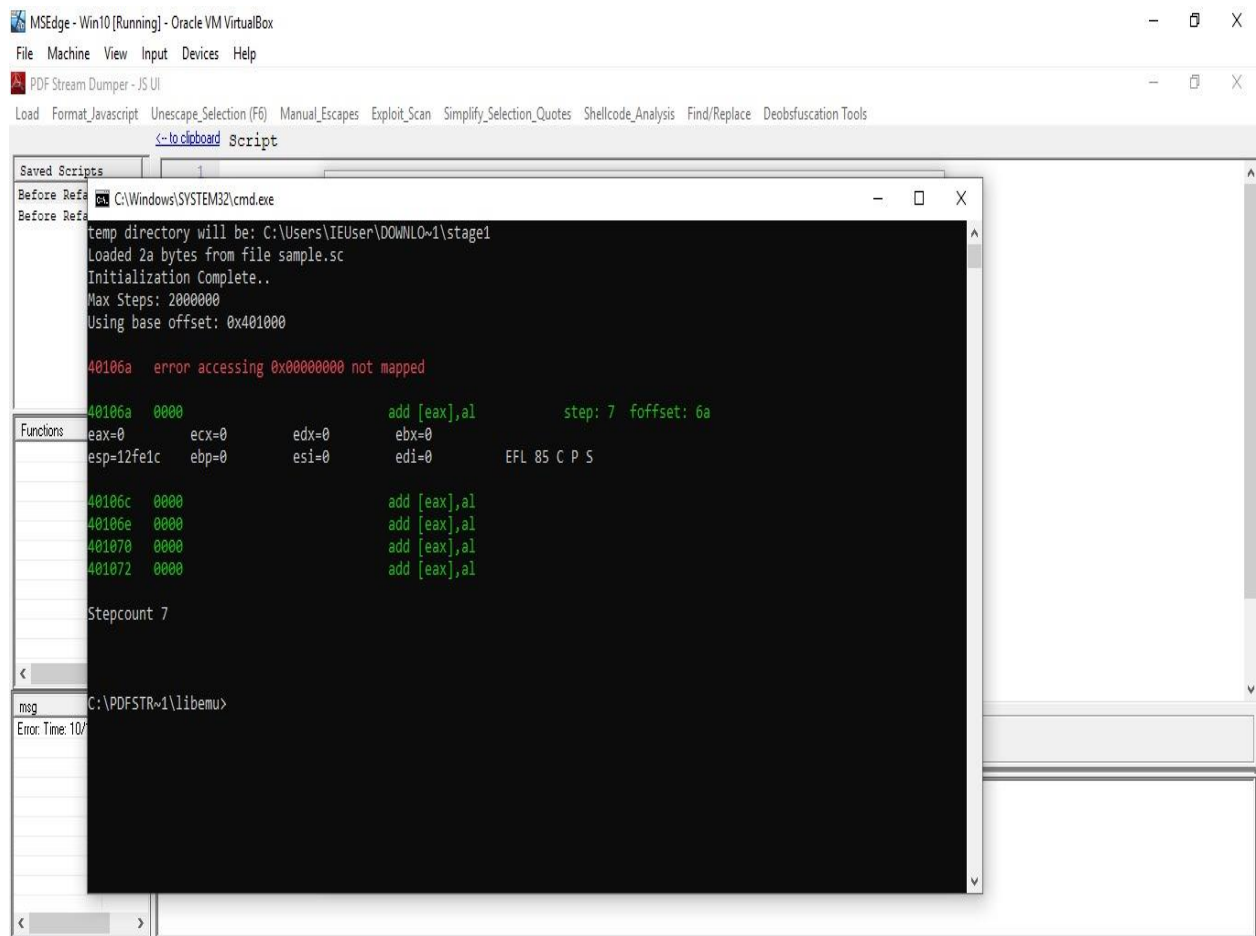
/f fpath      load shellcode from file - accepts binary, %u, %x, %x, hex blob
/api          scan memory and try to find API table
/auto         running as part of an automation run
/ba hexnum    break above - breaks if eip > hexnum
/bp varies    set breakpoint on file offset, virtual addr or api name (max 10)
/bs int       break on step (shortcut for -las <int> -vvv)
/b0           break if 00 00 add [eax],al
/cmd "string data" data to use for GetCommandLineA (use \" to embed quotes)
/cfo          CreateFileOverRide - if /fopen use handle else open real arg
/d            dump unpacked shellcode
/dir folder   process *.sc in <folder> supports: -r (1 report), -v (report mode), -u
/disasm int   Disasm int lines (can be used with /foff)
/dump         view hexdump (can be used with /foff)
/e int        verbosity on error (3 = debug shell)
/findsc       detect possible shellcode buffers (brute force) (supports -dump, -disasm)
/fopen file   Opens a handle to <file> for use with GetFileSize() scanners
/foff hexnum  starts execution at file offset (also supports virtual addresses)
/h            show this help
/hex          show hex dumps for hook reads/writes (paged)
/hooks        dumps a list all implemented api hooks
/i            enable interactive hooks (file and network)
/las int      log at step ex. -las 100
/laa hexnum   log at address or api ex. -laa 0x401020 or -laa ReadFile
/lookup api   shows the address of WinAPI function ex. -lookup GetProcAddress
/mm           enabled Memory Monitor (logs access to key addresses)
/mdll         Monitor Dll - log direct access to dll memory (hook detection/patches)
/min steps    min number of steps (decimal) to trigger record in findsc mode (def 200)
/nc           no color (if using sending output to other apps)
/noseh        Disables support for seh and UnhandledExceptionFilter
/norw         Disables display of read/write file hooks
```


7. Create a shellcode executable.

Answer: The created shellcode is shown below:



- Answer:** Before the execution of shell code, we got the data as Ip address, I-port or mac address etc. But due to the change of shell code we get an error.



9. What is the secret code?

Answer: The secret code is hocuspocus.

The screenshot shows the PDFStreamDumper application interface. The main window displays a JavaScript payload for a PDF's /S/Type/Action/Win entry. The payload is a long string of code designed to traverse the file system and find a file named 'sample.pdf'. A blue box highlights the text 'secret code is hocuspocus)' within the payload. Below the main window, the 'Search' tab is active, showing one search result for the string '<</S/JavaScript/JS(this.exportDataObject({ cName: "sample", nLaunch: 0 });)/Type/Action>>'. The bottom status bar shows the PDF path as 'C:\Users\IEUser\Downloads\stage1\stage1.pdf' and various PDF metadata fields.

MSEdge - Win10 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

PDFStreamDumper - http://sandsprite.com FileSize: 138 Kb LoadTime: 3.593 seconds

Load Exploits_Scan Javascript_UI Unescape_Selection Manual_Escapes Update_Current_Stream Goto_Object Search_For Find/Replace Tools Help_Videos

38 Objects

- 10 0x47B-0x48E
- 12 0x4DD-0x500
- 13 HLen: 0x7
- 11 HLen: 0x23
- 2 HLen: 0x0C
- 9 0x700-0xC85F
- 14 HLen: 0xA
- 17 0x0C79-0x0CA3
- 18 HLen: 0x7
- 16 HLen: 0x38
- 20 0xC03-0xC33
- 21 HLen: 0x7
- 19 HLen: 0x72
- 23 0xD007-0xD01C
- 25 0xD06B-0xD08E
- 26 HLen: 0x7
- 24 HLen: 0x23
- 15 HLen: 0x0C
- 22 0xD390-0x174C
- 27 HLen: 0xA
- 1 HLen: 0x38
- 28 HLen: 0x64
- 29 HLen: 0x24
- 30 HLen: 0x19
- 31 HLen: 0x1A
- 32 HLen: 0x4C
- 33 0x17921-0x2...
- 34 HLen: 0x59
- 35 HLen: 0x1B0
- 33 HLen: 0x43
- 3 HLen: 0x0C
- 0 HLen: 0x10C

```
<<
/S/Type/Action/Win
<<
  /F(cmd.exe)/D(c:\windows\system32)/P(/Q /C %HOMEDRIVE%&cd %HOMEPATH%&(if exist "Desktop\sample.pdf" (cd
"Desktop"))&(if exist "My Documents\sample.pdf" (cd "My Documents"))&(if exist "Documents\sample.pdf" (cd "Documents"))&
(if exist "Escritorio\sample.pdf" (cd "Escritorio"))&(if exist "Mis Documentos\sample.pdf" (cd "Mis Documentos"))&(start
sample.pdf)
>>
secret code is hocuspocus)
>>
```

Text HexDump Stream Details

1 Search Results

34 HLen: 0x59 <</S/JavaScript/JS(this.exportDataObject({ cName: "sample", nLaunch: 0 });)/Type/Action>>

Errors Search Debug (3)

Shell PDF Path C:\Users\IEUser\Downloads\stage1\stage1.pdf ... Load Abort

Streams: 11 JS: 1 Embeds: 1 Pages: 3 TTF: 0 USD: 0 flash: 0 UnkFlt: 0 Action: 3 PRC: 0