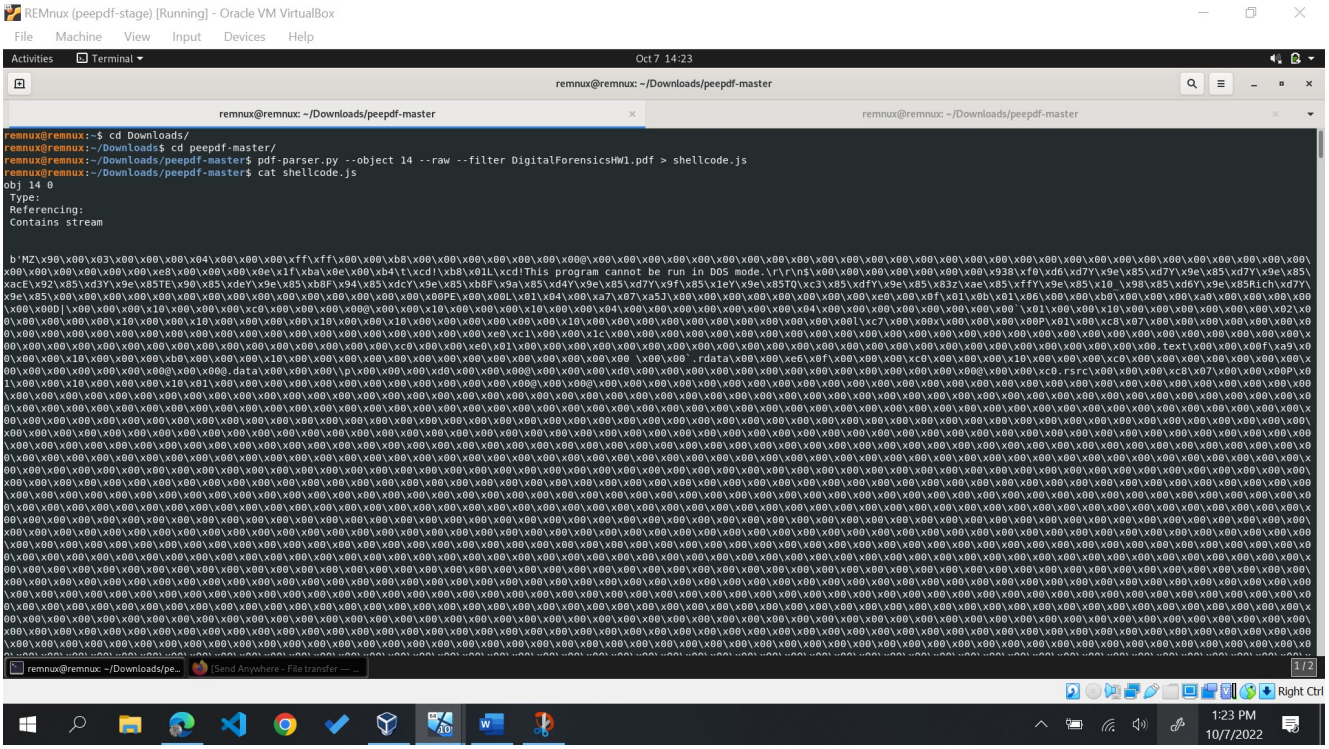
PDF analysis

1. Report the number of objects in the file - 18
2. Determine whether the file is compressed or not. - NO
3. Determine whether the file is obfuscated or not. -YES
4. Find and Extract JavaScript.
   14 and **15** Objects have js, JavaScript extracted by the peepdf.py extract command is given below

```
this.exportDataObject({
    cName: "sample",
    nLaunch: 0
});
```

5. De-obfuscate JavaScript.
   By using command –raw –filter the obfuscated js is stored in shellcode.js



6. Extract the shell code.

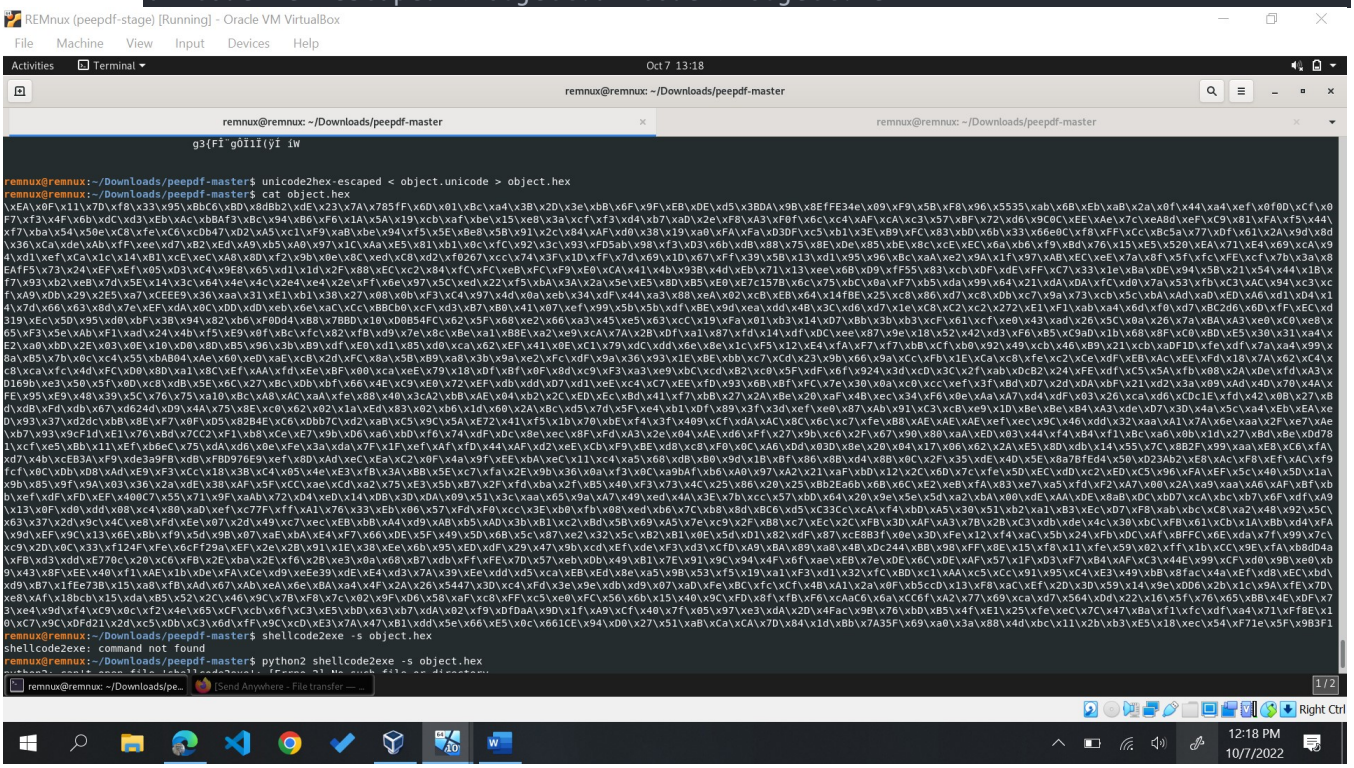
   The process of extracting shellcode is via following steps

   1. Convert to Unicode format

      ```
      cat object.raw  | tr '\' '%' > object.unicode
      ```

   2. Convert the Unicode to hex

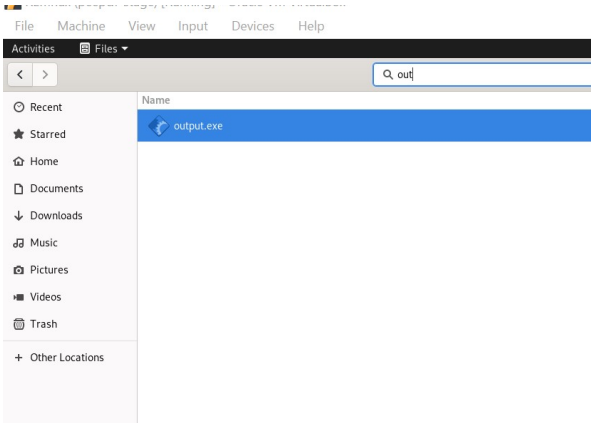
      ```
      unicode2hex-escaped < object.unicode > object.hex
      ```



7. Create a shell code executable

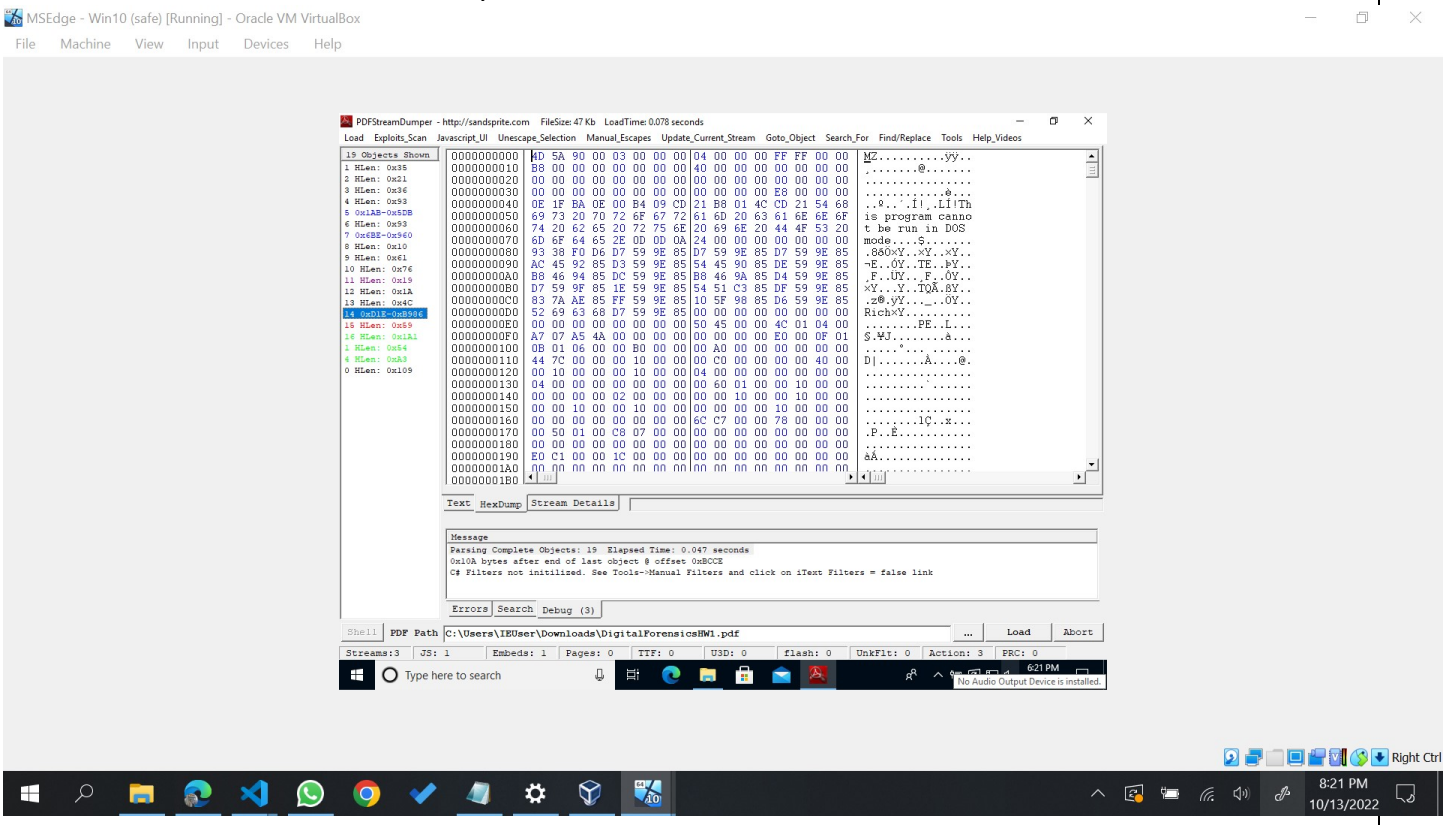
   Convert Hex file from the previous step to shell code

   ```
   shellcode2exe -s object.hex
   ```

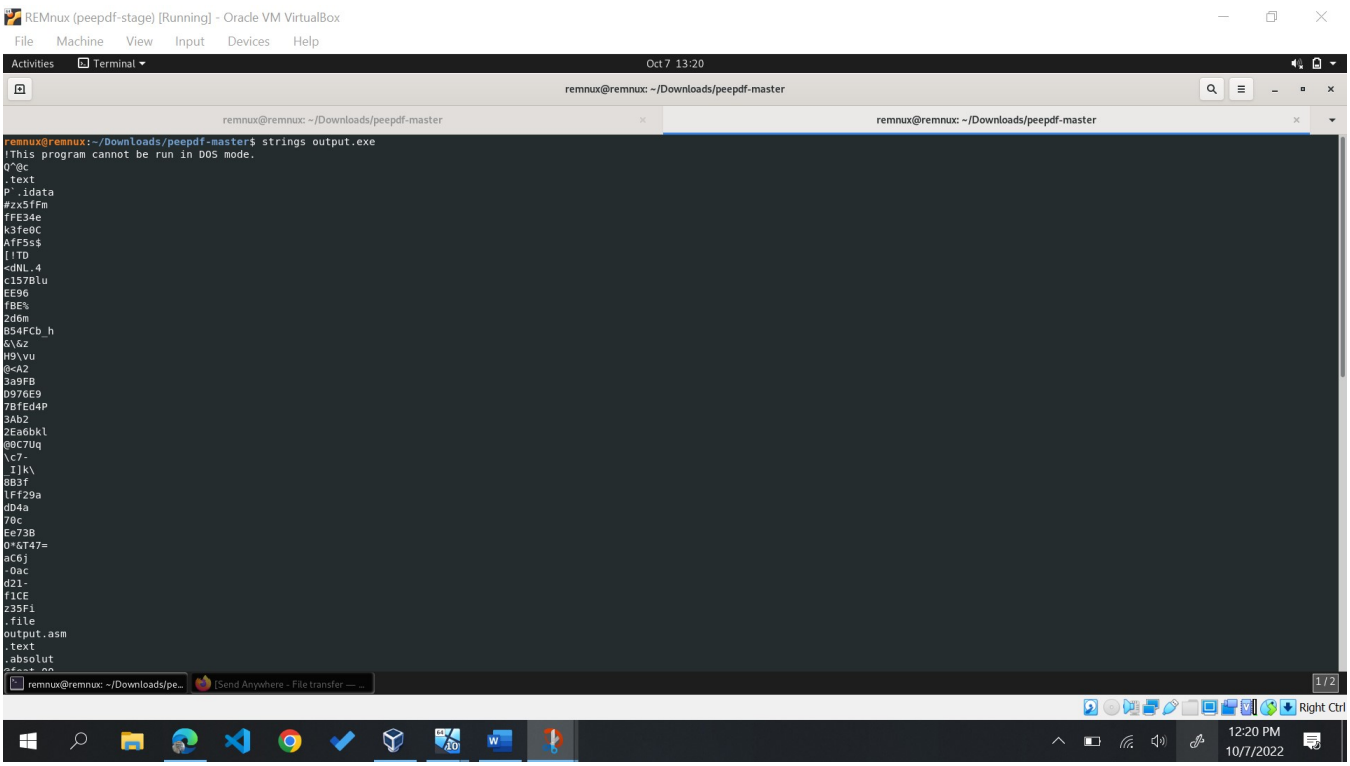


8. Analyze shell code and determine what is does or even execute it using sctest or spider monkey.

    1. Running PDF stream dumper
       Screenshot below of the analysis

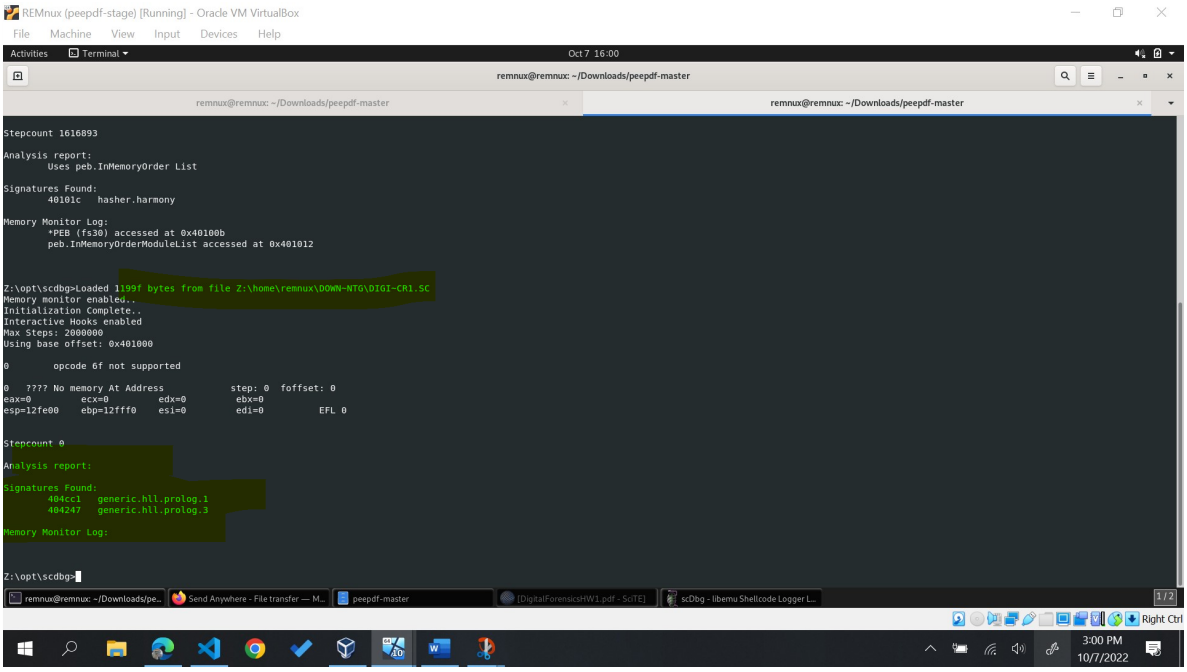


    2. Screenshot below of the hex code converted into exe files
       And a strings were extracted to print out strings in the exe file



3. After Printing out the strings we check for emulation of the exe
4. Running scdb test to emulate shell commands and found below signatures *generic.hll.prolog.1* and *generic.hll.prolog.3*
   And no memory monitor logs were found.

Screenshot below



9. What is the secret code? - SKY IS RED
   Secret code is embedded via Metasploit launch message and the exploit must be **exploit/windows/fileformat/adobe_pdf_embedded_exe**