CS 365: Digital Forensics
Spring 2020
Assignment #2

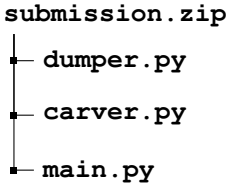
Arun Dunna
adunna@cs.umass.edu

Revision 1.0 - January 31, 2020

Due: February 7, 11:55pm

Submission Instructions

1. Your programming solutions for this assignment are to be written in **Python 3.6**.
2. Your programming solutions are to be submitted to the corresponding Gradescope programming assignment by the deadline **February 7, 11:55pm**. These should be submitted in a **ZIP file** containing your Python code files, in the following format:

   This will match the format of the assignment code distributable.

   Alternatively, you can upload each Python file individually in Gradescope instead of using a ZIP file.

Prerequisites

1. Ensure that you are enrolled in the course on Gradescope. The assignment submission will open a few days before the deadline. If you have not been automatically enrolled, you can use the following enrollment code: **9DZ53K**
2. Setup and test your own environment for executing **Python 3.6** code. For example, this can be in an IDE such as PyCharm, or in a terminal with the **`python3.6`** command.
3. Obtain the assignment distributable from the course website: **`asgn02distrib.zip`**

1 Programming

The assignment distributable contains skeleton code in two files: **`dumper.py`** and **`carver.py`**. It also contains **`main.py`**, but this is to test the program; **DO NOT EDIT THIS FILE**. Edit the dumper and carver files to complete following functions, further described in the docstrings of each function in the skeleton code. You may assume that the files you are reading (passed into your functions as strings) exist.

1.1 Hex Dumping [50 points]

(50 points) `hex_dump(inputFile)`**:** return a hexdump of the input file; example on ***main.py***:

```
"00000000  69 6d 70 6f 72 74 20 61  72 67 70 61 72 73 65 0a  |import argparse.|
00000010  69 6d 70 6f 72 74 20 64  75 6d 70 65 72 0a 69 6d  |import dumper.im|
00000020  70 6f 72 74 20 63 61 72  76 65 72 0a 66 72 6f 6d  |port carver.from|
00000030  20 6f 73 20 69 6d 70 6f  72 74 20 70 61 74 68 0a  | os import path.|
00000040  0a 69 66 20 5f 5f 6e 61  6d 65 5f 5f 20 3d 3d 20  |.if __name__ == |
00000050  22 5f 5f 6d 61 69 6e 5f  5f 22 3a 0a 0a 20 20 20  |"__main__":..   |
00000060  20 70 20 3d 20 61 72 67  70 61 72 73 65 2e 41 72  | p = argparse.Ar|
00000070  67 75 6d 65 6e 74 50 61  72 73 65 72 28 29 0a 20  |gumentParser(). |
00000080  20 20 20 70 2e 61 64 64  5f 61 72 67 75 6d 65 6e  |   p.add_argumen|
00000090  74 28 22 66 69 6c 65 22  2c 20 74 79 70 65 3d 73  |t("file", type=s|
000000a0  74 72 2c 20 68 65 6c 70  3d 22 6e 61 6d 65 2f 70  |tr, help="name/p|
000000b0  61 74 68 20 6f 66 20 74  68 65 20 69 6e 70 75 74  |ath of the input|
000000c0  20 66 69 6c 65 22 29 0a  20 20 20 20 70 2e 61 64  | file").    p.ad|
000000d0  64 5f 61 72 67 75 6d 65  6e 74 28 22 2d 63 22 2c  |d_argument("-c",|
000000e0  20 61 63 74 69 6f 6e 3d  22 73 74 6f 72 65 5f 74  | action="store_t|
000000f0  72 75 65 22 2c 20 68 65  6c 70 3d 22 63 61 72 76  |rue", help="carv|
00000100  65 20 66 69 6c 65 20 66  6f 72 20 55 54 46 2d 38  |e file for UTF-8|
00000110  20 63 68 61 72 61 63 74  65 72 73 22 29 0a 20 20  | characters").  |
00000120  20 20 70 2e 61 64 64 5f  61 72 67 75 6d 65 6e 74  |  p.add_argument|
00000130  28 22 2d 64 22 2c 20 61  63 74 69 6f 6e 3d 22 73  |("-d", action="s|
00000140  74 6f 72 65 5f 74 72 75  65 22 2c 20 68 65 6c 70  |tore_true", help|
00000150  3d 22 64 75 6d 70 20 68  65 78 20 76 69 65 77 20  |="dump hex view |
00000160  6f 66 20 66 69 6c 65 22  29 0a 20 20 20 20 70 2e  |of file").    p.|
00000170  61 64 64 5f 61 72 67 75  6d 65 6e 74 28 22 2d 73  |add_argument("-s|
00000180  22 2c 20 68 65 6c 70 3d  22 73 65 61 72 63 68 20  |", help="search |
00000190  66 6f 72 20 66 69 72 73  74 20 6f 63 63 75 72 72  |for first occurr|
000001a0  65 6e 63 65 20 6f 66 20  63 6f 64 65 20 70 6f 69  |ence of code poi|
000001b0  6e 74 22 29 0a 20 20 20  20 61 72 67 73 20 3d 20  |nt").    args = |
000001c0  70 2e 70 61 72 73 65 5f  61 72 67 73 28 29 0a 0a  |p.parse_args()..|
000001d0  20 20 20 20 69 66 20 70  61 74 68 2e 65 78 69 73  |    if path.exis|
000001e0  74 73 28 61 72 67 73 2e  66 69 6c 65 29 3a 0a 20  |ts(args.file):. |
000001f0  20 20 20 20 20 20 20 69  66 20 6e 6f 74 20 70 61  |       if not pa|
00000200  74 68 2e 69 73 66 69 6c  65 28 61 72 67 73 2e 66  |th.isfile(args.f|
00000210  69 6c 65 29 3a 0a 20 20  20 20 20 20 20 20 20 20  |ile):.          |
00000220  20 20 72 61 69 73 65 20  45 78 63 65 70 74 69 6f  |  raise Exceptio|
00000230  6e 28 22 47 69 76 65 6e  20 70 61 74 68 20 69 73  |n("Given path is|
00000240  20 6e 6f 74 20 61 20 66  69 6c 65 21 22 29 0a 20  | not a file!"). |
00000250  20 20 20 65 6c 73 65 3a  0a 20 20 20 20 20 20 20  |   else:.       |
00000260  20 72 61 69 73 65 20 45  78 63 65 70 74 69 6f 6e  | raise Exception|
00000270  28 22 47 69 76 65 6e 20  70 61 74 68 20 64 6f 65  |("Given path doe|
00000280  73 20 6e 6f 74 20 65 78  69 73 74 21 22 29 0a 0a  |s not exist!")..|
00000290  20 20 20 20 69 66 20 61  72 67 73 2e 64 3a 0a 20  |    if args.d:. |
000002a0  20 20 20 20 20 20 20 70  72 69 6e 74 28 64 75 6d  |       print(dum|
000002b0  70 65 72 2e 68 65 78 5f  64 75 6d 70 28 61 72 67  |per.hex_dump(arg|
000002c0  73 2e 66 69 6c 65 29 29  0a 0a 20 20 20 20 69 66  |s.file))..    if|
```

```
000002d0  20 61 72 67 73 2e 73 20  61 6e 64 20 61 72 67 73  | args.s and args|
000002e0  2e 73 2e 73 74 61 72 74  73 77 69 74 68 28 22 55  |.s.startswith("U|
000002f0  2b 22 29 20 61 6e 64 20  6c 65 6e 28 61 72 67 73  |+") and len(args|
00000300  2e 73 29 20 3c 3d 20 38  3a 0a 20 20 20 20 20 20  |.s) <= 8:.      |
00000310  20 20 70 72 69 6e 74 28  63 61 72 76 65 72 2e 75  |  print(carver.u|
00000320  74 66 38 5f 73 65 61 72  63 68 28 61 72 67 73 2e  |tf8_search(args.|
00000330  66 69 6c 65 2c 20 61 72  67 73 2e 73 29 29 0a 0a  |file, args.s))..|
00000340  20 20 20 20 69 66 20 61  72 67 73 2e 63 3a 0a 20  |    if args.c:. |
00000350  20 20 20 20 20 20 70 72  69 6e 74 28 63 61 72 20  |       print(car|
00000360  76 65 72 2e 75 74 66 38  5f 63 61 72 76 65 28 61  |ver.utf8_carve(a|
00000370  72 67 73 2e 66 69 6c 65  29 29 0a                 |rgs.file)).|
0000037b"
```

1.2 UTF-8 Carving [50 points]

(10 points) `utf8_search(inputFile, codePoint)`: search for the UTF-8 codepoint in the input file

(40 points) `utf8_carve(inputFile)`: extract any/all valid UTF-8 characters from the input file