Demos for Lecture 09
Updated for Fall 2019

MEMORY LAYOUT [slide #6]

Run ./locate64 multiple times

* See overall structure seen in Slide #6
* But heap and stack addresses vary from one run to another due to
  randomization
* Code stays fixed (code was compiled to NOT use position-independent
  code)

STACK LIMIT [slide #7]

Run ./runaway with different commandline values
* ./runaway 63 works
* ./runaway 64 segfaults

BUFFER OVERFLOW [slide #13]

Method #1

./bufdemo-nsp

When prompts for string, type

01234567890123456789012 OK
012345678901234567890123 Segfaults

Method #2

Same effect with

```
echo 01234567890123456789012 | ./bufdemo-nsp
echo 012345678901234567890123 | ./bufdemo-nsp
```

STACK SMASHING [slide #22]

```
cat smash-hex.txt | ./hexify | ./bufdemo-nsp
```

STACK CANARIES [slide #29]

```
echo 01234567 | ./bufdemo-sp            (OK)
echo 012345678 | ./bufdemo-sp           (Smashing detected)
```

Note that this particular version allows overflowing by one byte,
since LSB of canary == 0x00.